Experiments on Decision-making under Risk

Wenting Zhou

Doctor of Philosophy

University of York
Economics

March 2015

Abstract

This thesis consists of three chapters. They are on three different topics, but they are related in the sense that they are concerned with decision-making under risk or uncertainty.

In Chapter 1, we use an analogy of walking through fog to explain the problem under study. Fog has the property that vision becomes less and less clear the further ahead that you try to look, and after a certain distance you simply cannot see anything. In the experiment, the subject's aim is to travel across the foggy and hilly terrain with minimum energy expenditure. This problem, although set in a non-economic context, obviously has relevance to many economic problems.

Chapter 2 is about elicitation methods for discovering subjects' risk preferences. The concern of the experiment is to compare four different methods used for eliciting the level of risk aversion. We carried out an experiment in four parts, corresponding to the four different methods and our methodology involves fitting four different preference functionals. Our results show that the inferred level of risk aversion is more sensitive to the elicitation method than to the assumed-true preference functional. Experimenters should worry most about context.

Chapter 3 is about the interrelationship of decisions which come in a series. The validity of Random Lottery Incentive mechanism has been investigated in two main ways: first, just two decision problems, and second, many problems. This chapter combines and extends these two ways by investigating a cognitively less-demanding hypothesis than that all previous decisions are taken into account, but allows for an indirect effect of previous decisions on current ones. Reassuringly we find little effect and hence our results complement the previous evidence indicating that the Random Lottery Incentive mechanism is robust and can safely be used.

Table of Contents

List of Figures

List of Tables

Acknowledgments

I wish to express my profound and immense gratitude towards my supervisor, Professor John Hey, for his kind help, inspiring suggestions, guidance and encouragement throughout my PhD. I have learned a lot from him and he has been a true role model for my development as a young researcher, and without his support this thesis could have not been successfully accomplished. I also very appreciate the financial support for our experiments from Professor Daniela Di Cagno.

Second, I would like to thank my parents and grandparents, though they may not have always understood the finer points of behavioural economics and experimental economics, they have always been proud of my achievements and supportive of my goals.

Thirdly, I would also like to acknowledge the special support given me by my boyfriend, Dr Hao Lan, whose boundless generosity and thoughtfulness has kept me in one piece for the best part of my PhD study.

Finally, my great appreciation is given to all my fellow PhD students and other friends, especially from Zhechun He, Lisha Li, and Eryang Wang, for their support of my doctoral study and life especially at the final stage of my PhD.

Declaration of Authorship

I, Wenting Zhou, declare that this thesis titled, ‘Experiments on Decision-making under Risk’ and the work presented in it are my own. I confirm that:

- This work was done wholly or mainly while in candidature for a research degree at this University.
- Where I have consulted the published work of others, this is always clearly attributed.
- Where I have quoted from the work of others, the source is always given.
- I have acknowledged all main sources of help.
- Where the thesis is based on work done by myself jointly with others, I have made reference to that.

Chapter 3 of this thesis has previously been published in the following paper:

Hey, J., & Zhou, W. (2014). Do past decisions influence future decisions?. *Applied Economics Letters*, 21(3), 152-157.

Chapter 1. Walking through the Fog[1]

1.1 Introduction

Consider a typical dynamic problem - that of planning the optimal lifetime path of consumption conditional on income. If future incomes are certain the optimising process is conceptually straightforward (though it may be technically complex). If the future incomes are *uncertain*, there are two paths that can be followed. If future incomes are *risky*, and probability distributions are specified for incomes at each date, then the Decision-Maker (DM) can optimise by maximising the *expected* utility[2] of consumption over the cycle; the DM could solve the problem by backward induction. If future incomes are *ambiguous*, but can be characterised in some way (by sets of possible probabilities for example) then it could be assumed that one of the relatively new preference functionals for decision-making under ambiguity would be employed; once again backward induction could be used.

Note that all these methods assume that the DM either knows the distributions of future incomes or is prepared to make some assumptions about them. Now consider a situation where the DM is *told nothing* about incomes after a certain point in the future. In principle it could be modelled by assuming that the DM translates this information into some

---

[1] This chapter is a joint work with John Hey, and financially supported by Daniela Di Cagno (LUISS, Rome, Italy).

[2] Or some other objective if the preferences are not Expected Utility.

distributions. But it depends on how 'being told nothing' is interpreted. One way of interpreting this is as saying that the distribution of such incomes is unbounded. But taking expectations in such a case may lead to an expected utility, under all actions, of minus or plus infinity. In such a case, attempts to optimise over the complete future would be impossible. An alternative story would be that the DM could be modelled as looking only a short way forward – perhaps just as far as he or she can 'see' – and rolling this short horizon forwards as time passes. Clearly this does not use backward induction from the true horizon and cannot be considered the optimal strategy – either objectively or subjectively so.

In such a situation, an individual whom we would call rational in some sense will still aim to work out a strategy to maximise his or her utility. Here in this chapter we are interested in what they do – in a problem where there is 'no' information sufficiently 'far away'.

In this chapter, we investigate, in a situation where full information is not available, human decision-making. We cannot observe human behaviour directly in field, mainly because we cannot control the environment while trying to do so might raise ethical issues, as well as implying time problems. Thus, we have designed a laboratory experiment which retains the nature of this problem, and which can be finished within a reasonable time-slot without any ethical issues.

Our experiment also incorporates a further important extension, one in which backward induction may not only be computationally difficult or not lead to a solution, but one in which 'backwards' is not defined. The economic problem mentioned above is set in a one-dimensional (with the one dimension being time) world, in which 'backwards' is clearly defined, with the decision-maker starting in some time period t_1 and stopping in time period t_2. Consider however a two-dimensional problem. Label the two dimensions x and y. Suppose the decision-maker starts at some point in this 2-dimensional world (x_1,y_1) and the problem is to get to some other point (x_2,y_2) with some objective in mind. Passing through points on the way costs money (which is defined by the nature of the problem) and the objective is to get to the destination with the minimum expenditure. Now clearly if (x_1,y_1) and (x_2,y_2) do not share the same x values or the same y values (and even if they do) there is no obvious way 'backwards' from (x_2,y_2) to (x_1,y_1). Hence finding the optimal way from the start to the destination is not a backward induction problem – we shall explain in more detail later.

An example of such a problem is one where the decision-maker is a firm and wants to move from some initial factor combination to a new one, but can only do this in steps of a given size at any one time. Production has to occur in the meantime, and certain factor combinations are less efficient than others. There are costs to changing the factor inputs. This problem cannot be solved by backward induction, simply because there is no obvious meaning to the word 'backward'. It may be better to pass through an intermediate stage of production where one or both of the factor inputs are outside the range from the initial factor combination to the final one, than go 'directly to the final one.

The experiment we have designed is called "Walking through the Fog". In the experiment, the subject is asked to take a series of decisions. These decisions are not independent. A previous decision determines the option set for the next decision. His or her payoff will be shown at the end of the experiment, and is dependent on all the decisions that he or her has made. More specifically, in the experiment the subject has to travel across a map and reach a destination. She or he consumes energy while travelling. And the expenditure of energy is dependent on how hilly is the route that she or he has chosen. His or her aim is to reach the destination with the least energy expenditure. Or in other words, his or her aim is to find a flattest route from current position to the destination. She or he can find such a route if she or he can see the whole map clearly. However, to make the experiment interesting and relevant to the problem that we outlined, *the terrain is foggy*. The subject can vaguely see the terrain around him or her. But if the distance is too far, due to the fog, she or he can see nothing. Although this experiment is based in a non-economic context, it can be easily applied to many economic problems. We can gather the data which share the same nature with the economic problem we are interested in.

There is a good example in industrial organisation. Suppose a supermarket provides discount on a specific day each week to attract customers. Thus, on that day the profit brought by big demand covers the loss from discount. However, if the supermarket provides discount on more than one day in a week, the loss cannot be covered by the profit. And if it does not provide discount at all, it will suffer a loss since the customers might be attracted by competitors. Given the background, the supermarket faces a decision-making problem every working day – to discount, or not to discount?

In order to make a decision which maximises profit, the supermarket has to consider about the dimension of time. Discount today or postpone it? How its competitors will react to this decision in the future? Will such reactions enhance or weaken its profit in the future? The supermarket also has to consider about the dimension of space. What other events happen in the same time while it is making decision? For example, is its competitors providing discount now? Is there any important event just in the market?

The supermarket has to make a decision every working day with considering both of the two dimensions. Since it cannot predict the future, it may have some vague ideas about the near coming days, but know very little if the future is too far. For example, suppose now is Monday, the supermarket might have some expectation about Tuesday, and some vague information about Wednesday. However, it might have little prediction about Sunday.

In section 2, the design of the experiment is introduced. There we also talk about the implementation of the experiment. In section 3, we discuss the algorithm to find the optimal route without fog. Data from the experiment is analysed in section 4. We build several strategies and to see which one fits the subjects best. In section 5, we conclude, and discuss our plan of future research.

1.2 The Experiment

1.2.1 Introduction

With full information, it is possible for a person to work out an optimal strategy to maximise his or her utility in a dynamic decision-making problem. Here “full information” means accurate information about the future, or an appropriate distribution of the possible outcomes in the future. However, in the real world, not only the accurate information, but also the distribution of the possible outcomes is unavailable in many situations. Or even if the distribution is obtained, with huge variance, it is uninformative. For example, a possible value is equally likely to be between zero and infinity. We cannot have any useful inference from such information, since the expected value of that outcome is infinite.

Nevertheless, rational people still try to maximise their utility or payoff by some strategy. Indeed we found from our experiment that some of them have achieved an outcome which is not too far from the optimal strategy with full information.

We are interested in the question as to whether, in the situation without full information, what people do to try and maximise their utility or payoff. We are curious about what kind of strategies they employ, and what kind of outcomes these strategies lead to. The seemingly direct way of getting data for this topic is observing people making decisions in the field. But it is unrealistic. First, in field, some series of dynamic decision-making can run through the whole life span. It is impossible to observe a person's whole life closely and record every decision made by him or her for examination. Secondly, even if some kind of series of decisions can be observed in a relatively short time span, a person may not want to reveal all the decisions made, due to ethical issues or other reasons. Thirdly, even if a person would like to reveal all his or her decisions in a specific time span, some important decisions which affect the outcome heavily may still be excluded unwittingly. For example, a small thing ten years ago might be a vital reason for today's outcome; but it may have been ignored not only because of the passage of time but also because how small it is.

We turn to a laboratory experiment since it is impossible to gather the data in the field for our research question. A laboratory experiment has several advantages. First, the experiment can be run in about one hour. A group of subjects can do the experiment simultaneously. Thus we can gather a considerable amount of data in a relatively short time span. This is much more efficient than gathering data in field. Second, the experiment can be designed in a specific context without any ethical issues or problems of personal privacy. Third, the experiment can be designed to rule out all outside interference, so that the outcome of the experiment is only affected by the decisions made in the experiment.

In order to build a one-to-one mapping from the field topic to a laboratory experiment, we construct a story which is a metaphor of the topic. In the story, a person is travelling across hilly countryside. She or he consumes energy while moving uphill or downhill. Although the absolute difference is the same, uphill and downhill consume different amount of energy. She or he has an incentive to reach the destination with the minimum energy expenditure. In other words, she or he wants to follow a cheapest/optimal route to reach the destination. If she or he can see the whole terrain clearly, it is possible for him or her to work out an optimal route. Unfortunately, the terrain is foggy. She or he can only have some vague information of the terrain around. And if the distance is too far, she or he can see nothing at all. We would like to observe, under such a situation, people's behaviour of trying to identify the optimal route.

The experiment was implemented in Visual Studio. We use maps from the real world for our experiment. There are four journeys each with a different map in a session. The payoff for each journey is the endowment minus the total energy expenditure of that journey. Thus, in order to maximise his or her payoff, a rational subject's aim is to find the optimal route with the least energy expenditure. To mimic the fog, the subject cannot see the terrain of the map freely. The maps are divided into squares. Each square contains a number which denotes the height of that area. The subject can see the height of the square which shares a boundary (one square away) with his or her current square. However, if the square is two square-away from the current square, she or he can only see a range. For the squares which are three squares away from the current square, the range is even wider. For the further squares, the subject can see nothing.

There are four treatments in the experiment. Each subject only experiences one treatment. Each treatment contains four journeys with four different maps. But the four maps are the same four across treatments. The difference between treatments is the distance the subject can see and the width of the ranges. Table 1.1 provides a summary. If the range is denoted as ∞ it means that in that treatment the subject can see nothing at that distance.

Ranges Across Treatments			
Treatment	One square away	Two squares away	Three squares away
1	0	20	50
2	0	20	∞
3	0	40	100
4	0	40	∞

Table 1.1: Ranges across Treatments

In the experiment, once the destination of a journey is reached, the subject can start the next journey. The experiment ends when all the four journeys are finished. Then one of the four payoffs is randomly selected for determining the payoff to the subject. The payoff can never be negative. If the energy expenditure exceeds the endowment, the payoff is zero. The rest of this section contains details of the experiment.

1.2.2 The Background Story

In the story, the subject is supposed to be walking through countryside. His or her aim is to reach a specific destination with energy expenditure as low as possible. Unfortunately, the terrain is hilly. Thus, the subject may have to move uphill or downhill. Such activities cause energy expenditure. To make matters worse, the terrain is foggy. So the subject cannot directly see and plan a flattest route with the least energy expenditure from his or her current position to the destination.

Due to the thick fog, the subject does not have even a rough idea about how the terrain between his or her current position and what the destination looks like. She or he can only vaguely see the terrain around him or her. If the distance is too far, she or he cannot see anything, because the fog is so thick.

Given the conditions above, regardless of energy expenditure, reaching the destination itself looks like an impossible mission. But luckily, the subject has several items of equipment for helping. First, she or he has a map without contours. So she or he knows on the map where his or her starting point is and where the destination is. Secondly, she or he has a compass. So she or he knows to which direction the destination is. Combining these two tools, she or he can always check where his or her current position is, and how far away the destination is from the current position, without knowing the terrain between the two points.

Before the journey, the subject knows that, as a human-being, she or he has to spend energy while moving. To simplify, only moving uphill or downhill consumes energy. Moving through a flat route does not consume any energy. And as is the case, she or he spends more energy moving uphill than moving downhill. Given that she or he cannot see the terrain around clearly, and even cannot see anything if the distance is too far, she or he has to find an optimal route with the least energy expenditure possible to reach the destination. As we have described before, in this way, the subject can attempt to maximise his or her utility/payoff.

The subject has to fulfil an aim, which involves a lot of decision making. In our story, walking from the start position to the destination with the least energy expenditure is the aim. "With the least energy" implies "maximise utility", since using energy means negative experience.

The thick fog simulates the condition that, in real world, people cannot predict the future. She or he only has some vague information of the near future. In the experiment, the subject cannot see the terrain between his or her current position and the destination. Thus she or he cannot directly pick the flattest route with the least energy expenditure. However, in real world, people have to make series of decisions under this condition. And in the experiment, the subject has to make series of decisions of where to move to at every step.

As a result, the story is an abstraction of economic problem. By observing and analysing subjects' behaviour in the story, we can infer something about their behaviour in real world. Our experiment is based on such a story.

1.2.3 Experiment Design

1.2.3.1 Programming

1.2.3.1.1 Moving Rules

The experimental interface is programmed in Visual Studio. In the experiment, there are four different maps for travelling. These maps are modified from different parts of the real world, though the subjects do not know which parts they are. The map has been divided into 200 by 200 squares. Every square contains a non-negative integer which denotes the height of that square.

When the subject moves from one square to another, she or he spends energy. As we explained in the story, moving up uses more energy than moving down. More precisely, moving up is using energy equal to twice the difference between the heights of the two squares, while moving down is using energy equal to the difference between the heights of the two squares.

For example, if the height of the subject's current square is 50, and the height of the square to which the subject has decided to move is 55, then the subject is moving up, since 50 is smaller than 55. His or her energy expenditure is 10, which is twice the difference between the heights of the two squares.

If the height of the subject's current square is 55, and the height of the square to which the subject has decided to move is 50, then the subject is moving down, since 55 is larger than 50. His or her energy expenditure is 5, which is the difference between the heights of the two squares.

1.2.3.1.2 Decision Making

In the experiment, the subject's aim is to move from the start position to the destination with the least energy expenditure. Thus, for each journey, the subject has to decide which square to move to at every step. The four journeys are completely independent. Energy expenditure of one journey does not affect the others.

The subject starts from the centre square which is in the 101st row and 101st column. The square is shown in the centre of the screen. And she or he has to move between squares till they reach the destination. While travelling across the map, the subject can only move to the squares which share a boundary with his or her current square. In other words, she or he can only move to the adjacent square up, down, left or right relative to the current square.

Figure 1.1 is an illustration of the moving rule. At the beginning of each journey, the subject has to click the square with "Start" on it. Then she or he has four options, up, down, left, and right. That is, at this step, she or he can only move to the four squares with "*" on them. She or he has to decide which one to move to.

Figure 1.1: An Example for the Rule of Moving

From Figure 1.1, there is no useful information for helping to make decision. However, in the experiment, once the subject clicks the "Start" button, she or he will have some information about the true heights of nearby squares. Figure 1.2 is an example.

In Figure 1.2, the subject's current position is in the centre of the picture. It is denoted by a dark-grey square. And the subject can see that its height is 370. And the subject can see three-squares away from his or her current position.

The four green squares in Figure 1.2 are squares to which the subject can move. They are disabled until ten seconds have elapsed. The subject is forced to wait at least ten seconds before they can go to the next step. Such a design is to restrict the subject from just fast clicking without thinking. While green, the squares are selectable. And the subject can make a decision for this step. She or he can find the exact height of the four squares.

She or he might want to consider a bit further. Unfortunately, she or he cannot get the exact height of those squares which are not adjacent to his or her current position. But she or he still has some vague information about the range of the true value.

From Figure 1.2, if a square is two squares away from current position, the subject can see a range which contains the true height of that square. The interval of the range is 20. More precisely, in this example the range contains 20 integers. For example, if the range is 1-20, it contains 20 integers (all heights are given to the nearest integer). The true height of the square is one of the 20 integers all with the same probability. Subjects are told this.

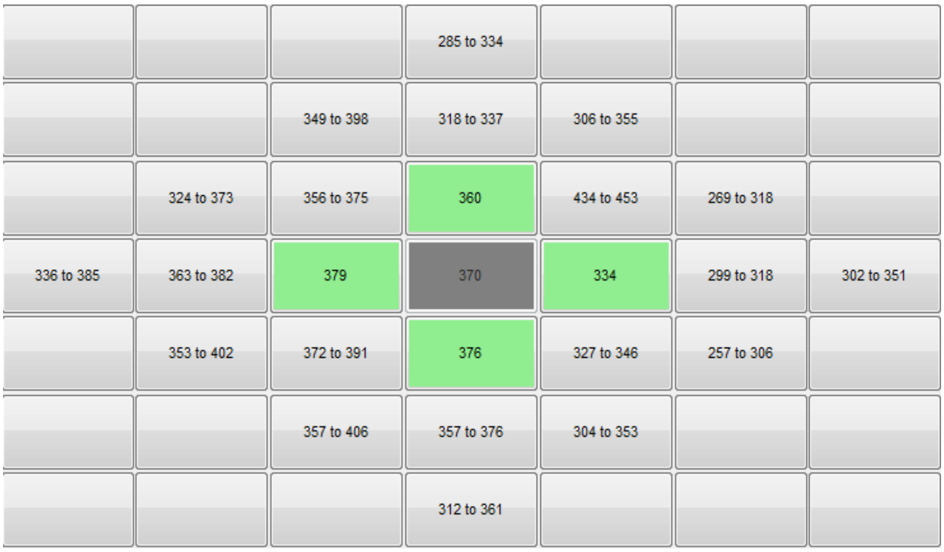


Figure 1.2: A Screenshot for the Experiment Interface in Treatment 1

If a square is three squares away from current position, the interval of the range in this example is 50, since the fog is increasing as the distance is increasing. Further, in this current example, if a square is more than three squares away from the current position, the subject is told nothing about its height; the distance is too far.

We should note that in the experiment, information is not always like this; it depends upon the treatment. There were four of them. Figure 1.2 is from treatment one. In treatment one, subject can see three squares away, with intervals of width zero, 20, and 50. In treatment two, subject can see only two squares away, with intervals of width zero and 20. In treatment three, subject can see three squares away, with intervals of width zero, 40, and 100. And in treatment four, subject can see only two squares away, with intervals of width zero and 40.

Thus, the four treatments differ with information quality and quantity. Table 1.2 illustrates. Treatment one has information with both higher quality and higher quantity. And treatment four has information with both lower quality and lower quantity. Or in other words, treatment four is much foggier than treatment one.

		Quality	
		High	Low
Quantity	High	T1	T3
	Low	T2	T4

Table 1.2: Information Quantity and Quality across Treatments

1.2.3.1.3 Interface

Figure 1.3 is the screen display at the beginning of a journey. The main body is the map. Due to space limitations, subjects are only shown a part of the map. Here it is an 11 by 11 matrix. As the subject moves across the map, the boundaries roll. The subject always stays in the centre of the matrix. In other words, if the subject is moving towards the destination, actually it is the destination moving towards to him or her.

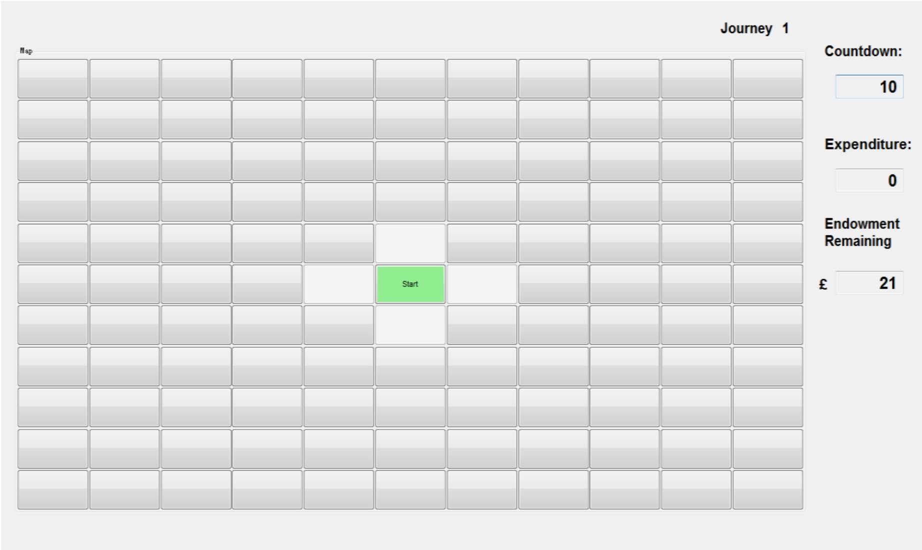


Figure 1.3: A Screenshot of the Interface before Starting

On the top left of the screen, it shows the number of the current journey. There are four journeys with different maps in a session of the experiment. The four maps are the same four across treatments.

At the upper-right corner, there is a timer. After one click, the subject has to wait 10 seconds to make the next click. If she or he clicks before the timer has reached zero, nothing happens.

Below the countdown, there is a box showing how much energy expenditure has been spent on this journey. Energy expenditure is independent from journey to journey. Once a new journey starts, the box shows energy expenditure from zero.

Under the expenditure box, there is the endowment-remaining box. In each session, endowments vary from journey to journey (because the journeys vary in difficulty). Across the four treatments, the four endowments are the same. From journey one to journey four the endowments are £21.00, £25.50, £25.50, and £36.00. The 'endowment remaining' is the endowment minus the energy expenditure, with an exchange rate of one point of energy expenditure equalling 1.5 pence[3]; the endowment remaining cannot be negative - if the energy expenditure converted to pounds exceeds the endowment, the endowment remaining will stay zero.

At the end of each journey, the endowment remaining is the payoff of this journey.

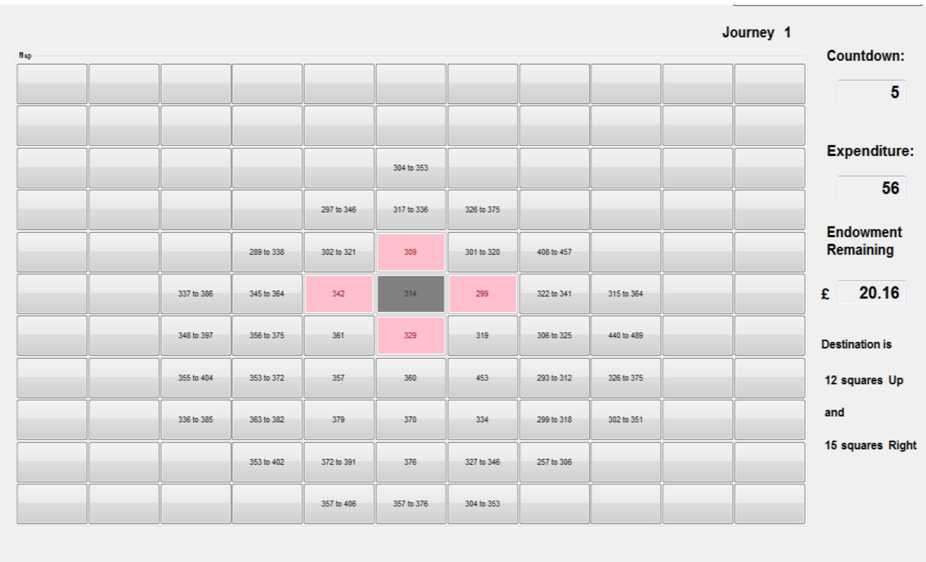


Figure 1.4: A Screenshot of the Interface after Starting

Figure 1.4 is an example of an ongoing journey in treatment one. On the map, the subject can see three squares away with intervals zero, 20, and 50.

[3] We did the experiment twice. For the one in May 2013, the exchange rate of one point of energy expenditure equalling 1 penny. For the one in Nov 2013, the exchange rate of one point of energy expenditure equalling 1.5 pence.

Once she or he knows the height of a specific square, the height stays on that square and will not be replaced by vague information.

The current position is always in the centre of the screen. The adjacent squares are red because the countdown has not reached zero. Thus the subject cannot make a decision. There are still five seconds to elapse. Once the countdown reaches zero, the four squares turn to green and clickable.

The expenditure box shows that, up to now, the subject has spent 56 points of energy, which equals 84 pence, or £0.84.

Since the initial endowment for journey one is £21. The endowment left is £21 minus £0.84, which is £20.16. It is shown in the endowment-remaining box.

The destination is not shown on the map of Figure 1.4, since it is too far away from the current position and out of the 11 by 11 matrix. However, the subject can always check how far away and in which direction the destination is. This information is given below the expenditure-remaining box; it shows that the destination is 12 squares up from and 15 squares to the right of the current position.

In Figure 1.5, the destination square has appeared on the map, since now the current position is close enough to the destination. The current position is still in the centre of the map. So actually, it is the destination moving towards the subject, who is always staying in the centre of the map.

Once the destination square is reached, the journey is finished. A message box shows with the payoff of this journey. Then the subject can click a "Start" button to enter the next journey.

Once journey four is finished, the whole experiment is finished. The screen turns out to one like Figure 1.6. The payoff of each of the four journeys are shown. One of them is randomly selected to be constitute the payoff for the experiment as a whole.

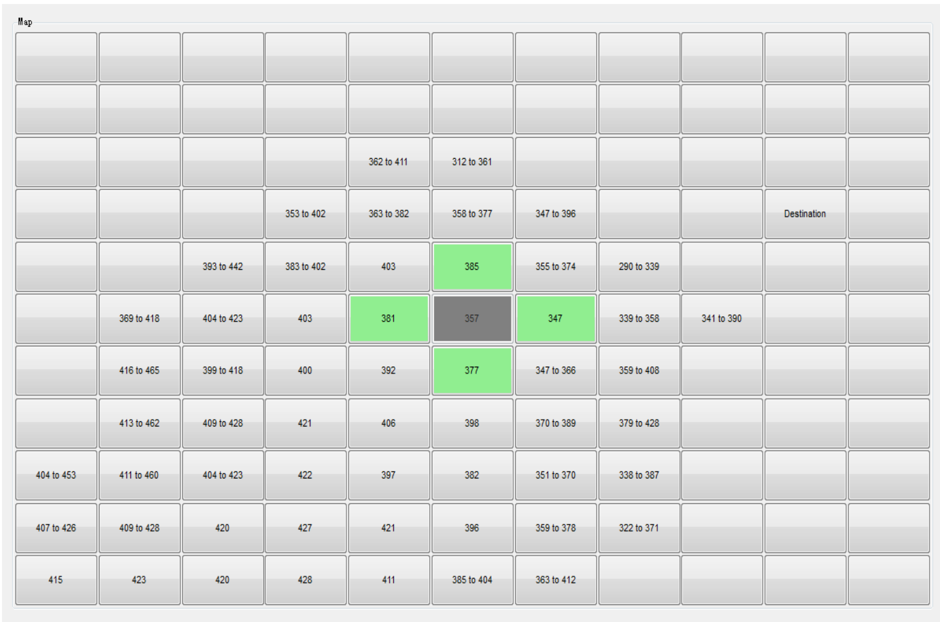


Figure 1.5: A Screenshot for a Nearly Ending Journey

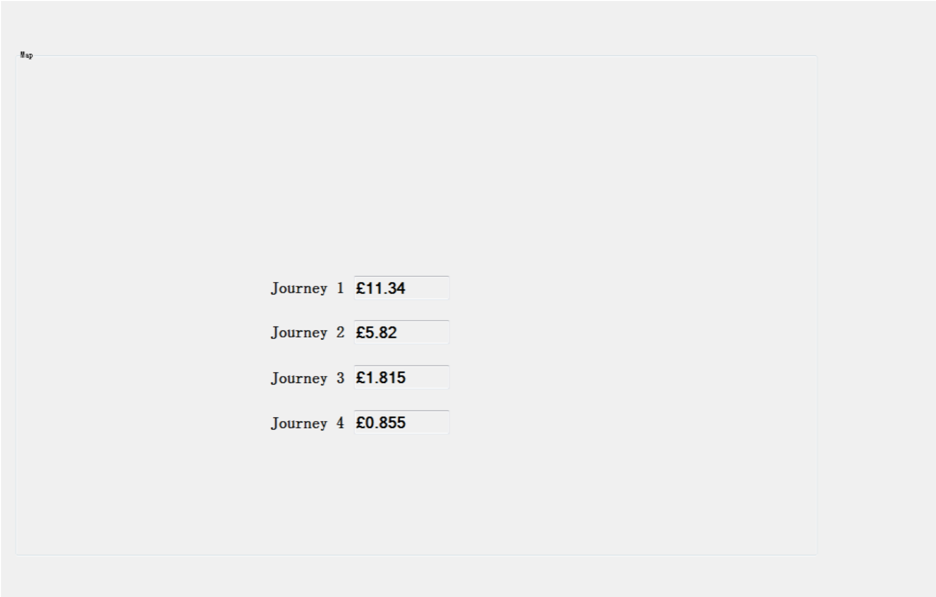


Figure 1.6: A Screenshot for the Payoff Summary after the Experiment

1.2.3.2 Data for the maps

The maps of the four journeys come from different parts of the real world. Real data is used instead of randomly-generated data.

The experiment is a metaphor of a series of decision making. In the real world, events in a sequence have some kind of relationship between each other. They are not completely independent. Thus, if we use data which is randomly generated by computer and the elements of which are completely independent of each other, it would become a bad metaphor. For example, there might be a peak next to an abyss. But this case seldom occurs in the real world.

We considered adding some correlations on randomly-generated data. However, it is impossible, because this process will cause bias. We tried to put some parameters which relate to adjacent elements while generating a new element. However, a computer cannot generate the whole map simultaneously. If there is an order of data generating, there is a bias. Since new elements always depend on the existing adjacent elements. And it is a one-direction relationship between elements.

Thus we decided to use real data to build the map. There is innate correlation of geographic elements. And such correlation is bidirectional rather than depending on a specific direction.

We get the data from CGIAR CSI (Consultative Group for International Agricultural Research - Consortium for Spatial Information). The name and version of the database is SRTM (the NASA Shuttle Radar Topographic Mission) 90m Digital Elevation Database v4.1. And the pictures of the four maps are shown in Figure 1.7. From left to right and from top to bottom, they are journeys one, two, three, and four.

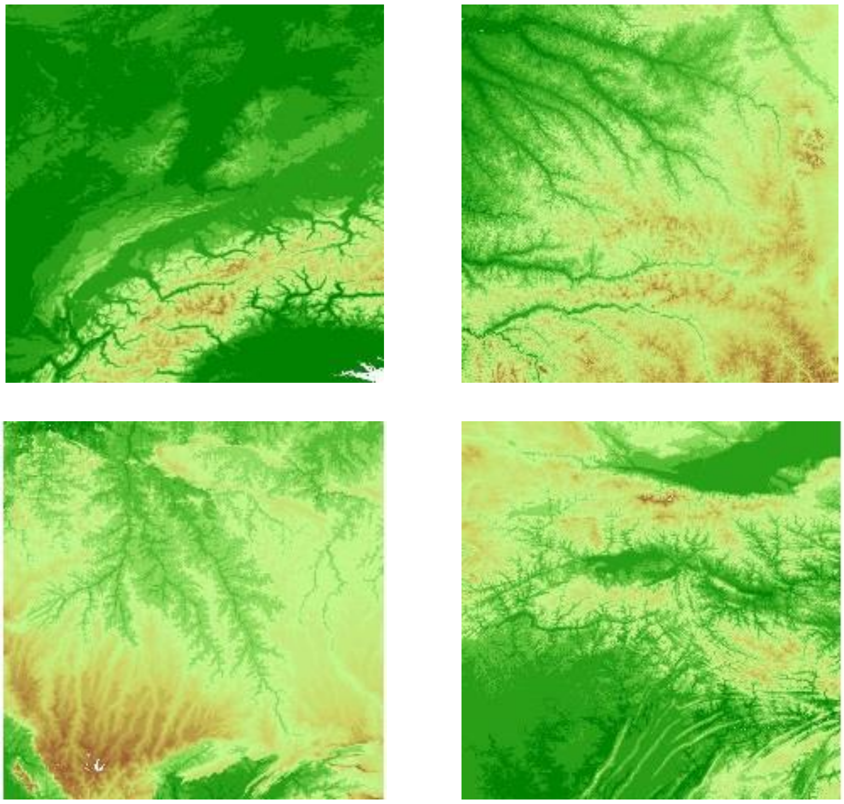


Figure 1.7: The Four Maps used in the Experiment

We deliberately choose maps which are quite hilly. The size of a map is five degrees of latitude by five degrees of longitude. Each map is divided into 6000 by 6000 squares. The width and length of each square is approximately 90 metres.

However, the 6000 by 6000 matrix is too large to our experiment. First, limited by time, subject can only go through a very small part of the map. Thus the large matrix is a waste and it is unnecessary. Secondly, to run a 6000 by 6000 matrix is a computational burden for computers.

In order to modify the data for our experiment, we did some aggregation. We treat the 6000 by 6000 matrix as a 200 by 200 matrix with each element is a 30 by 30 matrix. Then we calculate the average value of the 30 by 30 matrix. In this way, we reduce the large matrix to a 200 by 200 matrix with each element of an integer. This method simplifies the map. And the interrelationship of each element is retained.

In the experiment, we did not tell the subject which parts of the world the maps are. Thus we avoid providing them any hints of the journey.

1.2.4 Experiment Implementation

In the experiment there were four sessions each with four different treatments. There were 12 subjects in each session; each session lasted about one hour, though subjects were allowed to go at their own speed.

Since the experiment is an individual experiment, subjects were not allowed to communicate with others. The results were anonymous and kept as private information.

Before subjects entered the laboratory, they were asked to randomly draw a piece of paper which contained a number. This number was their seat number. In this way, subjects were randomly seated. Every seat was equipped with a computer, a set of Instructions (see the Appendix), and a pen.

After all subjects had been seated, the experimenter read the instruments from the front of the lab. Then the subjects were given five minutes to read the instructions and finish a set of control questions. During this period, if they had any question they could ask the experimenter; the experimenter would come and answer it privately. The experimenter also checked the answers of the control questions, in order to make sure that every subject had understood instructions.

Once all the subjects finished the control questions and had been checked by the experimenter, the experiment started. During the experiment, subjects were not allowed to communicate with each other. If they had any questions, they were free to ask the experimenter privately.

When the subject finished the experiment, she or he let the experimenter know. The experimenter approached him or her and brought an opaque bag. There were four balls in the bag numbered from 1 to 4. The subject randomly drew a ball from the bag, and she or he was paid according to the number on the ball. For example, if she or he drew a ball with number three, she or he was paid according to his or her payoff on journey three. In addition, each subject received a show-up fee of £2.5.

1.3 Optimal Route (without Fog)

1.3.1 Introduction

Without fog full information about the map can be obtained. However, the optimal route in this particular problem cannot be calculated by backward induction; first, because there is no obvious ‘backwards’ in this two-dimensional problem, and second because it is not necessarily the case that the optimal move at any stage is in a direction directly towards the destination. It might be better to go round some high peak rather than go through it.

In principle, the optimal route from the starting square to the destination can be found. It can be separated into two parts. First, the minimum energy expenditure from the starting square to the destination can be identified. Second, based on the minimum expenditure, the optimal route can be identified. The most difficult task is the first step. Since the matrix is big and there are huge numbers of possible routes, we cannot calculate the minimum expenditure manually. Unfortunately, in the context of our experiment, there is no existing algorithm to identify the minimum energy expenditure. However, we can borrow the idea from some of the algorithms of computer science. Then an algorithm can be built based on them to identify our optimal route.

In computer science, there is a general algorithm for “finding the shortest path”. This was constructed by Dijkstra (1959). At a first glance, “finding the shortest path” has no relation with our problem, since we are “finding the optimal route with the least energy expenditure”. However, these two problems are essentially similar.

In our context, we are looking for a route for which the energy expenditure is the least. The total energy expenditure is calculated by aggregating the energy expenditure between each two neighbouring squares on the route. Abstractly, we can treat each square as a node, and the energy expenditure is a value *v* related to the two neighbouring nodes. For finding an optimal route, it is for identifying a route with the minimum sum of *v* among all possible routes from the initial node to the destination node. It can be expressed as follows.

$$\text{Energy Expenditure of Optimal Route} = \text{Min}(\sum_{i=1}^{n-1} v_i) \qquad (1.1)$$

On a route from the starting square to the destination, there are *n* squares[4] in total. The energy expenditure is calculated from the starting square towards the destination. Once the energy expenditure of moving from one square to the other has been calculated, both the two squares are marked as visited. v_i is the energy expenditure of moving from one square to its unvisited neighbour square. Especially, v_1 is the energy expenditure of

[4] Here the number *n* includes the starting square and the destination.

moving from the starting square to one of its neighbour square, and v_{n-1} is the energy expenditure of moving to the destination from one of its neighbour square.

In computer science, for finding the shortest path, it is actually identifying a route following which the distance between the initial node and the end node is minimised. The distance is the sum of the distance between each the two neighbour nodes on the route. It can be exhibited as follows.

$$\text{Distance of Shortest Path} = \text{Min}(\sum_{i=1}^{n-1} v_i) \qquad (1.2)$$

On a route from the initial node to the end node, denote by *n* the total number of nodes passed through. The distance is calculated from the initial node towards the end node. Once the distance between two nodes has been calculated, both the two nodes are marked as visited. Denote by v_i the distance between one node and its unvisited neighbour node. In particular, v_1 is the distance between the initial node and one of its neighbour nodes, and v_{n-1} is the distance between one of the neighbour nodes of the end node and the end node itself.

Although the problem of finding the shortest path looks different from the problem of identifying the optimal route, they are essentially similar. From equation (1.1) and equation (1.2), it is easy to see that the difference between the two problems is the definition of v_i. Thus, we can borrow some ideas from the algorithm of finding the shortest path for solving our problem.

1.3.2 Review of Algorithms for Finding the Shortest Path

1.3.2.1 Dijkstra's Algorithm

Dijkstra (1959) has constructed an algorithm for finding the shortest route in computer science. It can be decomposed into several steps. First, all the nodes except the initial node are marked as unvisited. The tentative distances from the initial node to each of them are signed as infinite. Second, the initial node is marked as the current node, with the tentative distance from itself as 0. Third, if a node is marked as the current node, the new tentative distance for each of its unvisited neighbour nodes is calculated by the tentative distance of the current node adding the distance between the current node and that node. The old tentative distance assigned on that node is replaced by the new one if and only if the new tentative distance is smaller. Fourth, as soon as all the neighbour nodes of the current node are considered, the current node is marked as visited. A visited node is never visited again. Fifth, if the destination is marked as the current node or the smallest tentative distance among all unvisited node is equal to infinity, the algorithm is finished. In the second case, there is no connection between the current node and any of its unvisited neighbour nodes. Sixth, the unvisited node with the smallest tentative distance is marked as the current node. The third step is repeated till the destination is marked as visited.

Here is an example to illustrate Dijkstra's Algorithm. In Figure 1.8, each circle denotes a node, and each line denotes the path between two nodes. The number along with each line denotes the distance between the two connected nodes. n_1 is the initial node and n_6 is the destination.

Figure 1.9 shows the map after the second step of Dijkstra's Algorithm. All the numbers or signs inside the brackets denote the tentative distance from the initial node to that node. The initial node, n_1, is marked as the current step, with the distance to itself as 0. All the other nodes are marked as unvisited, with the tentative distance as infinite.

Figure 1.10 shows the map after the third step of Dijkstra's Algorithm for the first round. Since n_1 is the current node, its neighbour nodes are n_2, n_3, and n_4. The new tentative distance from the initial node to n_2 equals the tentative distance assigned to n_1 adding the distance between n_1 and n_2. The result is 15, which is smaller than the tentative distance assigned on n_2. Thus for n_2 the old tentative distance ∞ is replaced by the new tentative distance 15. In the same way, the tentative distances assigned on n_3 and n_4 can be replaced by 10 and 6 respectively. Since all the neighbour nodes of n_1 have been considered, n_1 is marked as visited. And since neither of the two conditions in the fifth step is satisfied, the algorithm goes on. According to the sixth step, the node n_4 with the smallest tentative distance among all the unvisited nodes is marked as the current node. The algorithm is not finished, thus we go back to the third step.

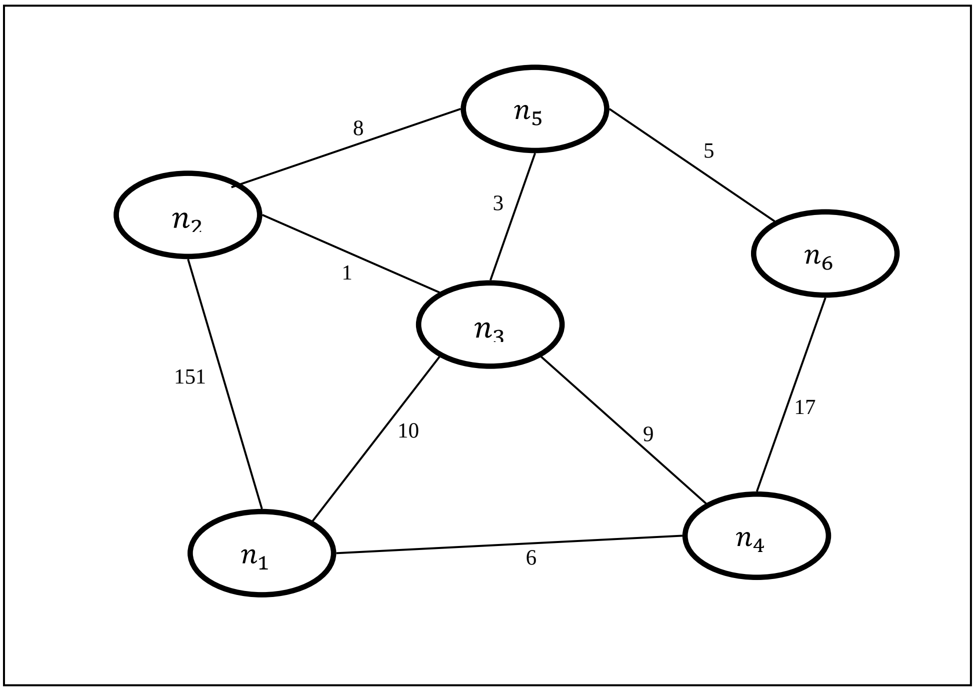


Figure 1.8: An Example of Dijkstra's Algorithm (Initial)

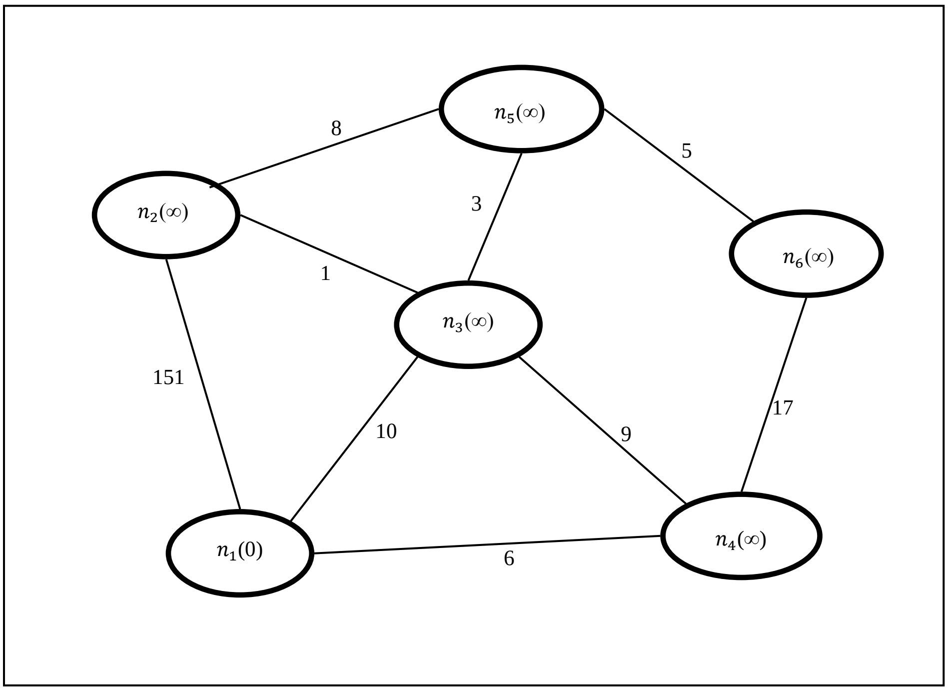


Figure 1.9: An Example of Dijkstra's Algorithm (After the Second Step)

Figure 1.11 shows the map after the third step of Dijkstra's Algorithm for the second round. n_4 is marked as the current node, and its unvisited neighbour nodes are n_3 and n_6. The new tentative distance for n_3 is the sum of the tentative distance assigned to n_4 and the distance between n_4 and n_3. It is 19, and it is greater than the old tentative distance assigned to n_3. Thus, the old tentative distance is not replaced by the new one. But for n_6, the old tentative distance ∞ is replaced by the new one, which is 23. And then, since there is no more unconsidered neighbour nodes of the current node, n_4 is marked as visited. Neither of the two conditions in the fifth step is satisfied, and the algorithm goes on. According to the sixth step, n_3 is marked as the new current node.

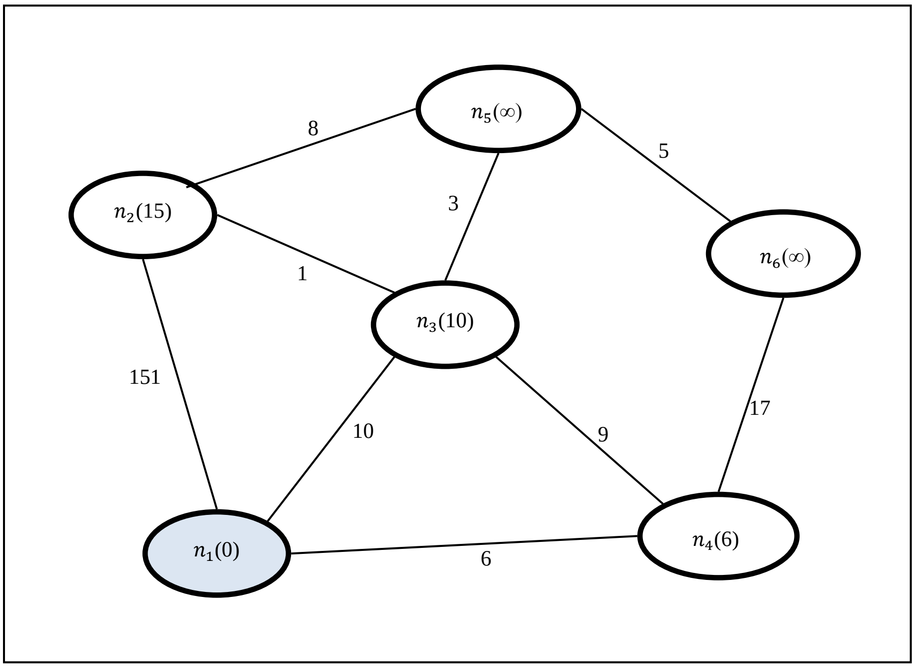


Figure 1.10: An Example of Dijkstra's Algorithm (After the Third Step - 1)

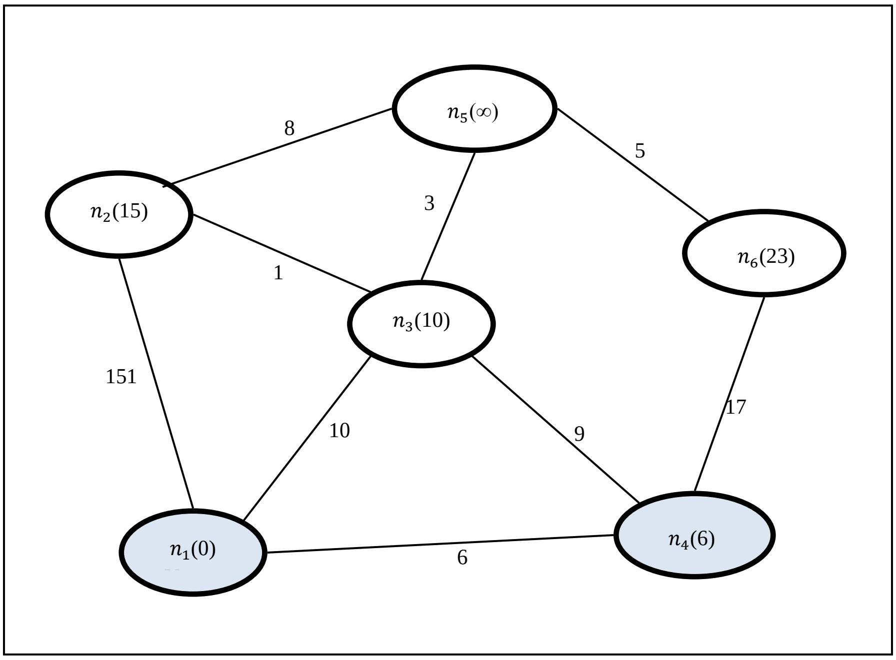


Figure 1.11: An Example of Dijkstra's Algorithm (After the Third Step - 2)

Figure 1.12 shows the map after the third step of Dijkstra's Algorithm for the third round. n_3 is the current node with assigned tentative distance 10. Its unvisited neighbour nodes are n_2 and n_5. The new tentative distance for n_2 is the tentative distance of n_3 adding the distance between n_3 and n_5, which is 11. It is smaller than the old tentative distance of n_2. Thus, the tentative distance of n_2 is replaced by 11. And in the same way, the tentative distance of n_5 is replaced by 13. Since both of the unvisited neighbour nodes of n_3 are visited, n_3 is marked as visited. Neither of the two conditions in the fifth step is satisfied, and the algorithm goes on. According to the sixth step, n_2 is marked as the new current node.

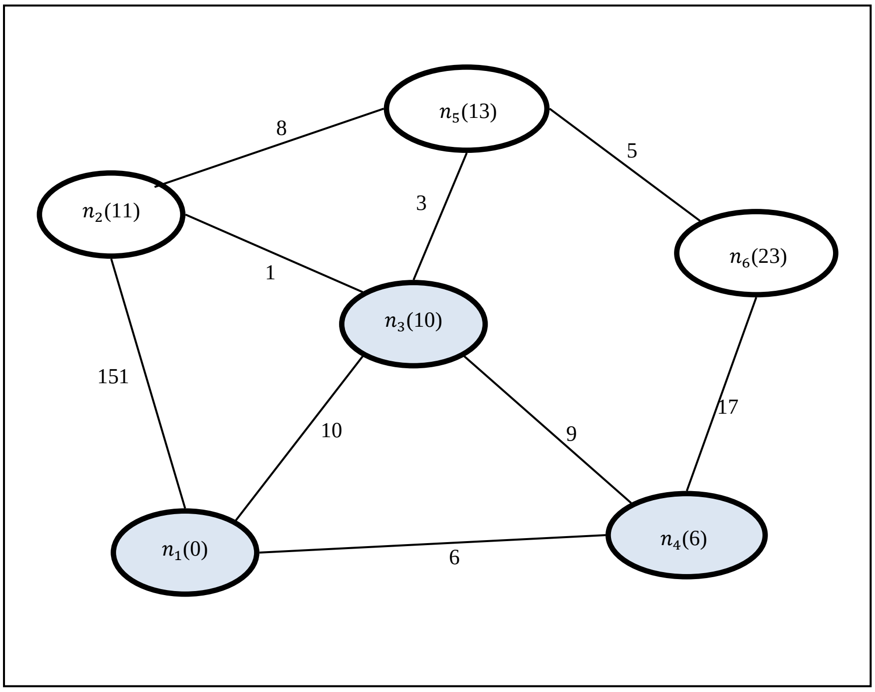


Figure 1.12: An Example of Dijkstra's Algorithm (After the Third Step - 3)

Figure 1.13 shows the map after the third step of Dijkstra's Algorithm for the fourth round. n_2 is marked as the current node with tentative distance 11. The only unvisited neighbour node of it is n_5, with tentative distance *13.* The new tentative distance of n_5 is the sum of the tentative distance of n_2 and the distance between n_2 and n_5. The result is 19 and it is greater than 13. Thus, the tentative distance of n_5 is not replaced by the new one. Since there are no unconsidered neighbour nodes, n_2 is marked as visited. Neither of the two conditions in the fifth step are satisfied, and the algorithm goes on. According to the sixth step, n_5 is marked as the new current node.

Figure 1.14 shows the map after the third step of Dijkstra's Algorithm for the fifth round. n_5 is the current node, of which the tentative distance is

13. The only unvisited neighbour node of it is n_6, with tentative distance 23. Since the distance between n_5 and n_6 is 5, the new tentative distance for n_6 is 18. It is smaller than 18. Thus the old tentative distance of n_6 is replaced. Since there are no unvisited neighbour nodes other than n_6, n_5 is marked as visited. And n_6 is the current node.

Since n_6 is the destination and is also the current node, the first condition in the fifth step is satisfied. The algorithm is finished. We can easily check the steps and find the shortest path from the initial node to the destination. It is shown in figure 1.15. And the path is n_1-n_3-n_5-n_6.

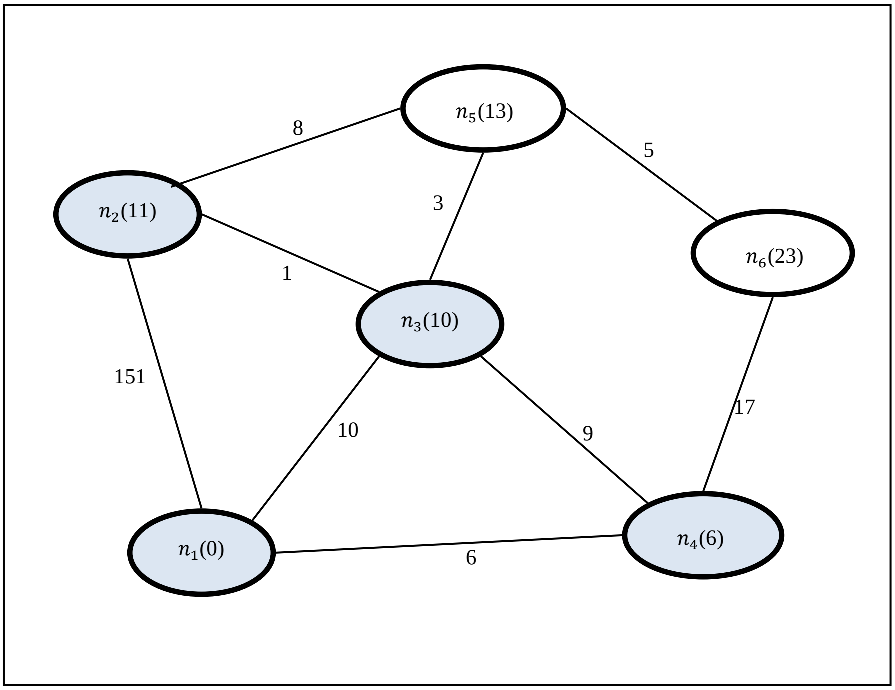


Figure 1.13: An Example of Dijkstra's Algorithm (After the Third Step - 4)

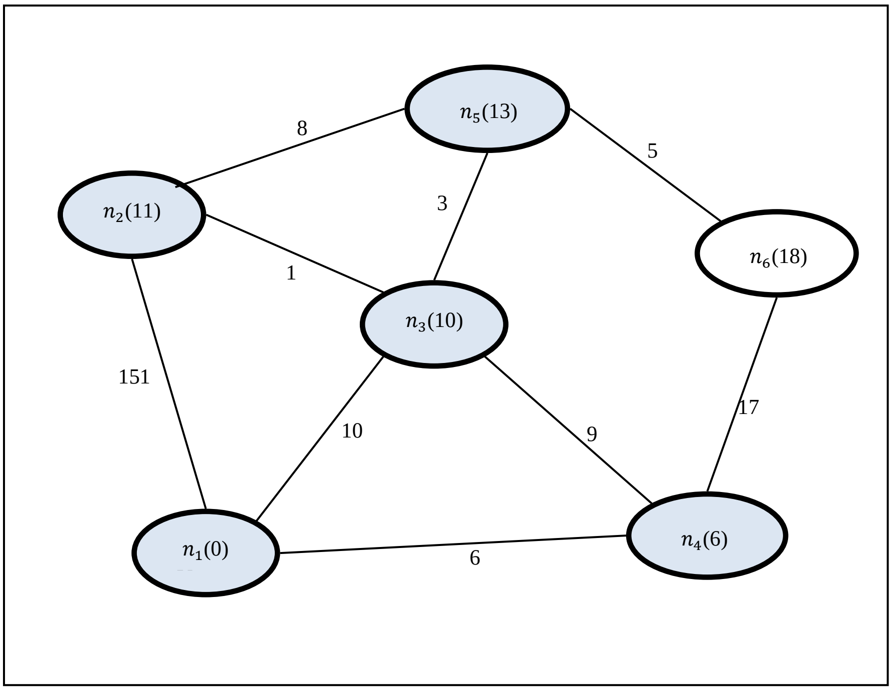


Figure 1.14: An Example of Dijkstra's Algorithm (After the Third Step - 5)

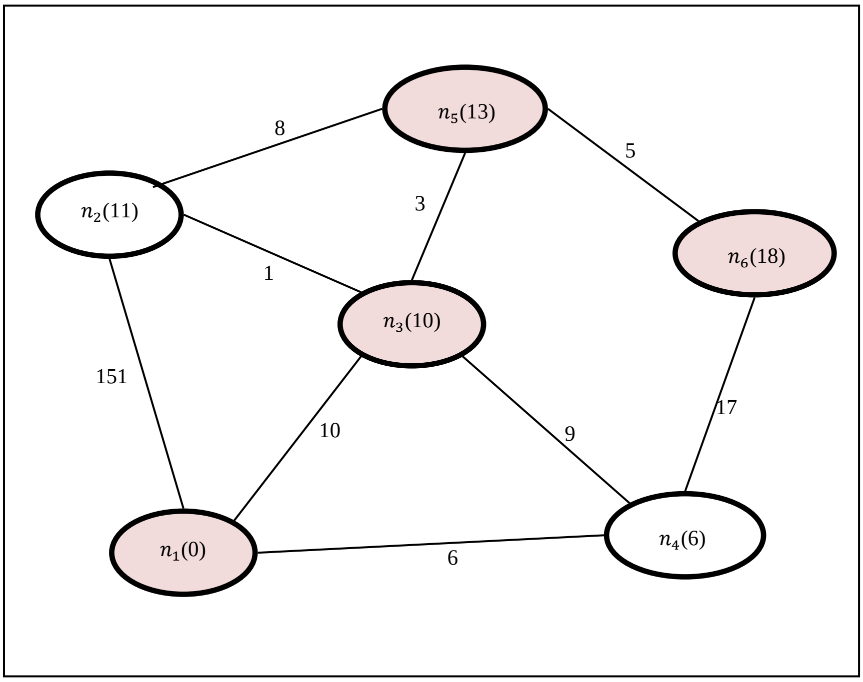


Figure 1.15: An Example of Dijkstra's Algorithm (Finished)

1.3.2.2 Other Related Algorithms

In computer science, there are several algorithms which share the same essential properties with Dijkstra's Algorithm. But they are different in details and thus suitable for special cases. Here we have a review on them one after one.

1.3.2.3 Bellman-Ford Algorithm

Bellman-Ford Algorithm (Ford, 1956 and Bellman, 1958) is one of such algorithms. If we treat the distance between two nodes is a kind of path cost, in Dijkstra's Algorithm, all the path costs are non-negative. If there is negative path cost on the map, Dijkstra's Algorithm cannot identify it. Compared with Dijkstra's Algorithm, Bellman-Ford Algorithm can identify negative path costs. But it requires more calculation time since there is a final scan at the end of the algorithm.

If there is a cycle with negative cost on the map, no cheapest path from the initial node to the destination exists. Since any path can be cheaper after one more walk around the negative-cost cycle. But the negative path cannot be identified by following Dijkstra's Algorithm since this algorithm visit each node at most once.

Although the Bellman-Ford Algorithm follows Dijkstra's Algorithm to obtain the potential cheapest path, once the path has been identified, a final scan of the whole map is applied to check if there is any negative cycle. In the scan, for each node *i* in the map, if the sum of the tentative cost of *i* and the cost between *i* and its neighbour node *j* is less than the tentative cost of *i*, a negative path cost is identified. It can be shown as follows.

$$if\ c_i + v_{ij} < c_i$$

The path between node i and j has negative path cost.

Here c_i denotes a node *i* on the map, and v_{ij} denotes the path cost between *i* and one of its neighbour nodes *j*.

Figure 1.16 shows an example to show how Bellman-Ford Algorithm identifies negative path cost. This map is modified from the map we used for illustrating Dijkstra's Algorithm. The only modification is that the path cost between n_3 and n_5 is -3 instead of 3.

It is obvious to see that, by Dijkstra's Algorithm, a route with "the minimum path cost" can be identified. The route and minimum path cost is illustrated in Figure 1.16. With this algorithm, a visited node will never be visited again. However, in the figure, we can see that, the minimum path cost for this map is actually infinite. Since every time visiting one of n_3 and n_5 from the other, the minimum path cost can be reduced by three. It is an infinite loop. Thus Dijkstra's Algorithm identifies an incorrect route with an incorrect minimum path cost.

By the Bellman-Ford Algorithm, there is a final scan after finding the possible route with "the minimum path cost". In the final scan, if the tentative cost assigned on one node plus the cost to one of its neighbour node is less than the tentative cost itself, there is a path with negative cost between the two nodes. In the figure, we can see that, the tentative cost for n_3 is ten, and the cost between n_3 and its neighbour node n_5 is minus three. The sum of these two numbers is seven, which is less than ten. And for n_5, the tentative cost is seven, and the cost between n_5 and its neighbour node n_3 is minus three. The sum of these two numbers is four, which is less than seven. Thus, a path with negative cost between n_3 and n_5 is identified.

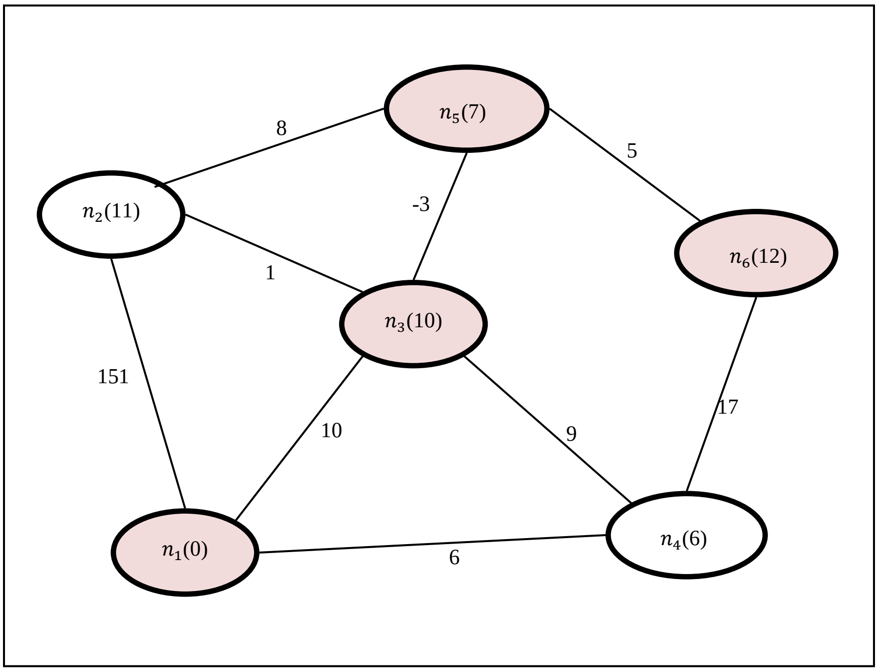


Figure 1.16: A Map with Negative Path Cost

1.3.2.4 A* Algorithm

A* Algorithm (Hart et al 1968) is also a computer science algorithm for finding the minimum-cost path, or in the distance scenario, finding the shortest path. It cannot identify a path with negative cost, but it has higher efficiency than Dijkstra's Algorithm.

The higher efficiency comes from a heuristic estimate of the cost from node *i* to the destination. Unlike Dijkstra's algorithm and Bellman-Ford algorithm, A* algorithm takes into account not only the forward path, but also the cost of travelled path. Thus, the tentative cost in this algorithm denotes the tentative cost from the initial node to the destination rather than to node *i*. It is composed of two elements, and can be presented as follows.

$$t_i = d_i + h_i$$

Suppose node *i* is under consideration, then t_i denotes the tentative cost for the whole route calculated at node *i*. d_i denotes the actual cost from the initial node to the node *i*. h_i denotes the heuristic-estimate cost from node *i* to the destination. Here h_i has to be an admissible heuristic. It means that, h_i should not overestimate the cost from node *i* to the destination. For instance, in the distance scenario, h_i should no more than the minimum actual cost from node *i* to the destination. Usually, it equals to the linear distance from node *i* to the destination.

The A* algorithm can be decomposed into several steps. First, an empty set is constructed. It can be called a consideration set. Second, all the neighbour nodes of the initial node are put into the consideration set. Third, there is a calculation of the tentative cost for each node in the consideration set. Here the tentative cost for node *i* is the actual cost from the initial node to the node plus the heuristic-estimate cost from that node to the destination. Fourth, the node with the smallest tentative node is removed from the consideration set. And all its neighbour nodes are added into this set and their tentative costs are calculated. Fifth, if the destination is removed from the set or the smallest tentative cost in the set is infinite, the algorithm is finished. In the second case, there is no connection between any node in the set and the destination. Sixth, the third step is repeated till either case in the fifth step appears.

The example we used for illustrating Dijkstra's Algorithm can also be used for illustrating the A* algorithm. In Figure 1.17, all the neighbour nodes of the initial node n_1 is added in to the consideration set. They are n_2, n_3, and n_4. Their tentative cost can be calculated as follows.

$$t_2 = d_2 + h_2 = 15 + 1.5 = 16.5$$

$$t_3 = d_3 + h_3 = 10 + 1 = 11$$

$$t_4 = d_4 + h_4 = 6 + 17 = 23$$

Here for t_4, the heuristic-estimate cost is also the actual cost from n_4 to the destination n_6.

Then according to the third step, the node with the smallest tentative cost should be removed from the consideration set. Here n_3 is removed. Then all its neighbour nodes are added into the set. Here n_3 has only one neighbour node outside the set, which is n_5. The tentative cost of n_5 can be calculated as follows.

$$t_5 = d_5 + h_5 = 13 + 5 = 18$$

Here for t_5, the heuristic-estimate cost is also the actual cost from n_5 to the destination n_6.

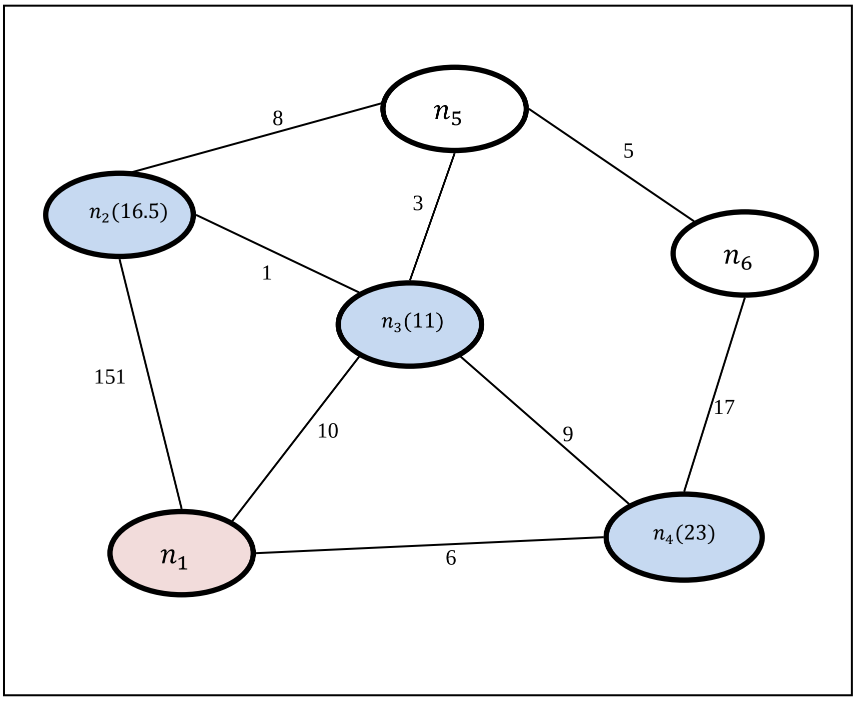


Figure 1.17: An Example of the A* Algorithm (After the Second Step)

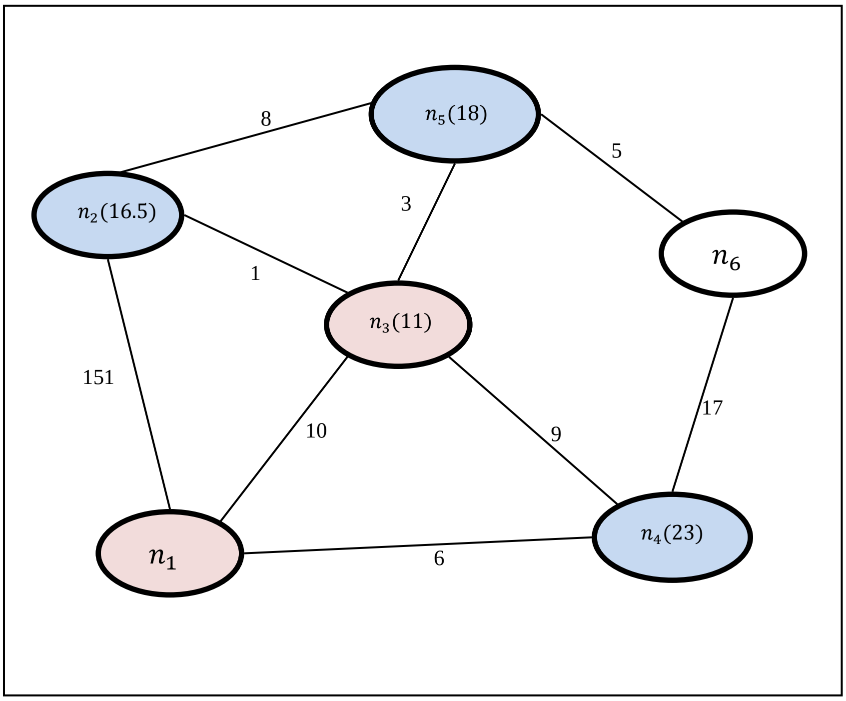


Figure 1.18: An Example of the A* Algorithm (After the third Step-1)

Figure 1.18 illustrates the scenario after the third step for the first iteration. Now there are three nodes, n_2, n_4, n_5, in the consideration set. Neither of the two cases in the fifth step is satisfied, thus it goes to the sixth step and then repeat the third step once more.

This time the node with the smallest tentative cost is n_2. It is removed from the set. And it has no neighbour node outside the set. It is obvious to see that, the route through n_2 to n_5 costs more than the route through n_3 to n_5. n_2 can be abandoned since it is not on the optimal route.

Now follow the third step again, the node with the smallest tentative cost is n_5. It is removed from the set. n_5 has only one neighbour node, which is n_6, the destination. n_6 is added to the set. Neither of the two cases in the fifth step is satisfied, thus it goes to the sixth step and then repeat the third step once more.

Figure 1.19 illustrates the scenario after the second iteration of the third step. It is obvious to see that the destination has the smallest tentative cost. Then the destination is removed from the consideration set. The first case in the fifth step is satisfied, and the algorithm is finished.

Figure 1.20 illustrates the finished version of this example. The optimal route and the smallest cost is exactly the same as Figure 1.15. The A* algorithm gets the same results as Dijkstra's algorithm, but is more efficient.

In A* algorithm, the problem is solved after the third iteration. In Dijkstra's algorithm, the problem is solved after the sixth iteration.

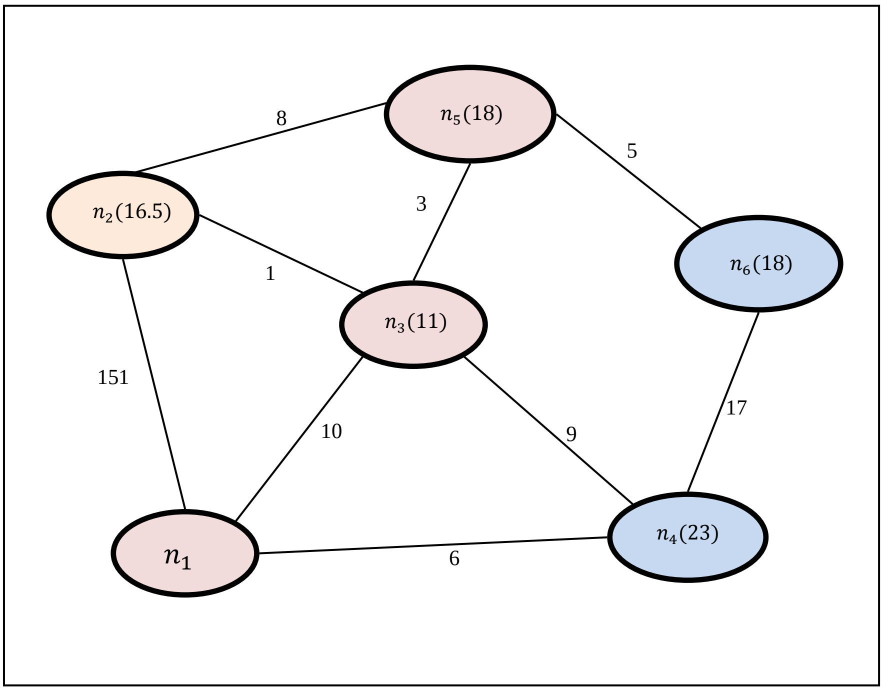


Figure 1.19: An Example of the A* Algorithm (After the third Step-2)

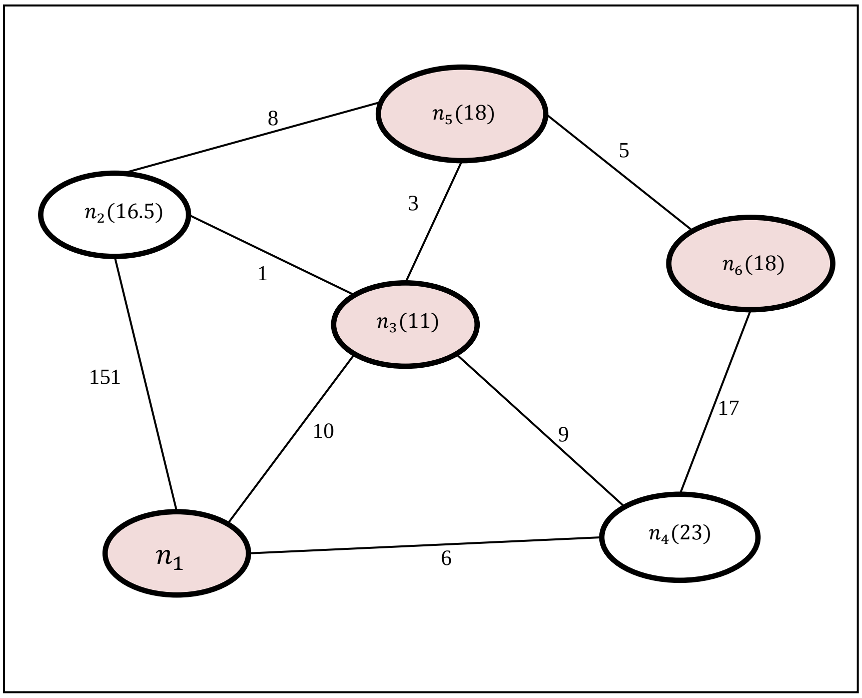


Figure 1.20: An Example of the A* Algorithm (Finished)

1.3.2.5 Other Algorithms for the Shortest Path Problem

Besides Dijkstra's algorithm, the Bellman-Ford algorithm, and the A* algorithm, there are some other algorithms developed to solve the shortest path problem. For instance, there are Floyd–Warshall algorithm (Floyd, 1962 and Warshall, 1962), Johnson's algorithm (Johnson, 1977), Viterbi algorithm (Viterbi, 1967), and so on.

These algorithms are developed from the three algorithms we have mentioned above. They are designed for solving specific computer science problems. We have explained the three algorithms in details, and have illustrated the essential of solving the shortest path problem sufficiently before go into our optimal problem. Discussing more about other algorithms will not add more help for solving *our* problem, which is a crucially different one.

1.3.3 Our Algorithm

1.3.3.1 Impossibility of Identifying All the Possible Routes

In our problem, there is seemingly a straightforward way to find the optimal route with the smallest energy expenditure. We only need to identify all the possible routes from the starting square to the destination. Then we calculate the energy expenditure for each route, and select the one with the smallest energy expenditure.

However, this mission is impossible in our case. For a 200 by 200 map, the number of all the possible routes is huge, even if for each route, each square can be passed at most once. Here is an example to illustrate the amount of the possible routes as the dimension of the map increases. In order to simplify the example, we put two restrictions here. First, on each route, a square can never be passed more than once. Second, if square A shares a boundary with square B, moving from A to B is always a direct move. Specifically, moving from A to B is always A to B, rather than A to other squares, and then to B.

These two restrictions can reduce the number of routes to be analysed without distorting the result of finding the optimal route. Since all the moving costs are non-negative in our problem, revisiting a square will cause unnecessary energy expenditure. It is weakly dominated by visiting a square at most once. Similarly, moving to a neighbour square in a roundabout way also may cause unnecessary energy expenditure.

Figure 1.21 is a 2 by 2 map. The number in each square only denotes the label of that square. Square 1 is the starting square and square 4 is the destination. Taken the two restrictions above into consideration, all the possible routes can be identified. There are only two possible routes. One is 1-2-4 and the other is 1-3-4.

3	4
1	2

Figure 1.21: A 2 by 2 Map

Figure 1.22 is a 3 by 3 map. Square 1 is the starting square, and square 9 is the destination. There are six possible routes in total. They are 1-2-3-6-9, 1-2-5-6-9, 1-2-5-8-9, 1-4-5-6-9, 1-4-5-8-9, and 1-4-7-8-9.

7	8	9
4	5	6
1	2	3

Figure 1.22: A 3 by 3 Map

Now we increase the size of the map to be 4 by 4. It is illustrated in Figure 1.23.

Square 1 is the starting square and square 16 is the destination. It is still possible to enumerate all the possible routes under the two restrictions. There are twenty possible routes in total. They are listed in Table 1.3.

13	14	15	16
9	10	11	12
5	6	7	8
1	2	3	4

Figure 1.23: A 4 by 4 Map

It is obvious to see that the number of the possible routes is exponentially growing as the size of the matrix increasing. For a 5 by 5 matrix, there are 70 possible routes. It is already difficult to enumerate each route. And for a 6 by 6 matrix, it is even difficult to calculate the number of the possible routes manually.

Our map is a 200 by 200 matrix. Obviously the number of all the possible routes is huge. It is theoretically possible but practically difficult to identify each route, calculate their energy expenditure, and choose the optimal one. This task is even overwhelming for a personal computer to calculate.

This is the reason for why we cannot identify all the possible routes and select the optimal one directly. We have to seek for some efficient algorithm to solve our optimal problem. Here the efficiency means that, by such an algorithm, an optimal route can be identified by personal computer in a tolerable time span.

Step 1	Step 2	Step 3	Step 4	Step 5	Step 6	Step 7
1	2	3	4	8	12	16
1	2	3	7	8	12	16
1	2	3	7	11	12	16
1	2	3	7	11	15	16
1	2	6	7	8	12	16
1	2	3	7	11	12	16
1	2	3	7	11	15	16
1	2	3	10	11	12	16
1	2	3	10	11	15	16
1	2	3	10	14	15	16
1	5	6	7	8	12	16
1	5	6	7	11	12	16
1	5	6	7	11	15	16
1	5	6	10	11	12	16
1	5	6	10	11	15	16
1	5	6	10	14	15	16
1	5	6	10	11	12	16
1	5	6	10	11	15	16
1	5	6	10	14	15	16
1	5	6	13	14	15	16

Table 1.3: Possible Routes for a 4 by 4 Matrix

Nevertheless, there is no ready algorithm in economics for solving our problem. We have to borrow some ideas from computer science. They have some algorithms to solve the shortest path problem, which share some similarities with our problem.

1.3.3.2 Relationship between the Shortest Path Problem and Our Problem

As we have mentioned before, although the contexts of the shortest path problem and our problem are not the same, they can be regarded as one-to-one mapping.

In the first place, the aims of the two problems are similar. For solving the shortest path problem, a route from the initial node to the destination with the smallest distance has to be identified. For solving our optimal route problem, a route from the starting square to the destination with the smallest energy expenditure has to be identified.

What is more, in both the problems, moving is restricted and not free. In the shortest path problem, moving can only happen between two neighbour nodes. Here neighbour means there is a path between that node and the current node. And distance can be regarded as a kind of cost of moving from one node to the other. In our optimal route problem, moving can only happens between two neighbour squares. Here neighbour means that square sharing a bound with the current square. And energy is expended while moving.

In addition, for the two problems, the ways of calculating the expenditure are the same. For the shortest path problem, the distance for the whole journey is the sum of the distance between each two neighbour nodes on the route. And for our optimal route problem, the total expenditure of energy is the sum of the energy expenditure of each step moving on the route.

1.3.3.3 Differences between the Shortest Path Problem and Our Problem

However, we cannot directly use the shortest path problem to solve our problem. First, none of the algorithms reviewed above can go backward. Algorithms for shortest path problem only consider the path in the area between the initial node and the destination. They start from the initial node and goes towards the destination directly. But on our map, part of the optimal route is possibly outside the area between the starting square and the destination. For instance, the subject might have to go backwards from the starting square and move several squares in the opposite direction to that towards the destination, since going directly toward the destination might be extremely costly.

Figure 1.24 illustrates an example. A is the starting square and B is the destination. Following the algorithm of shortest path problem, all the possible routes are constrained in the 3 by 3 matrix with the bold boundary. Thus, the optimal route should also be constrained in this area. However, in our problem, the optimal route might be out of the area. Our scenario is travelling across hilly country. Suppose H and I are two high mountains, but the path A-C-D-E-F-G-B is a flat path with zero energy expenditure. Thus a rational person should go one-square backward and follow the flat path, rather than go directly toward the destination through H or I. And in this case, A-C-D-E-F-G-B is the optimal route which cannot be identified by the algorithms of shortest path problem.

			B	
	H		G	
	A	I	F	
	C	D	E	

Figure 1.24: Difference between Shortest Path Algorithm and Our Algorithm

Second, the way of calculating the expenditure of moving from one square to the other is different. In the shortest path problem, the moving cost is regardless of the direction. The expenditure of moving from node A to node B is the same as the expenditure of moving from node B to node A. However, in our problem, the direction matters. The moving cost depends on the relative amount of the value on the current square and the value on the moving-to square. The expenditure of moving from A to B is usually different from the expenditure of moving from B to A. As we have mentioned, our story is travelling across the country. The terrain is hilly and as is common sense, going uphill consumes more energy than going downhill.

For instance, square A is assigned with a value x, square B is assigned with a value y, and x is greater than y. In our problem, moving from A to B is downhill, and the energy expenditure is the absolute difference between x and y. But moving from B to A is uphill, and the energy expenditure is twice the absolute difference between x and y.

Third, even with the most efficient algorithm, A*, the calculation load of a 200 by 200 matrix is overwhelming for a personal computer using MATLAB. It can take several days if the variance of the values on a map is large enough, and could end with a crash as the computer overloads and runs out of storage space. Thus, we need to produce a more efficient method which can be run on a personal computer within a tolerable time span.

1.3.3.4 The Idea of Our Algorithm

As we have mentioned above, the shortest path problem is a kind of one-to-one mapping of our problem, but we cannot use the algorithm directly. We have to borrow some ideas from those algorithms and produce our own algorithm for finding the optimal route.

The key ideas of our algorithm are “expanding” and “updating”. These can solve not only the “backward steps” but also the workload of calculation. We also modify the way of calculating the moving cost between two squares. In our context, the relative amount of the value on the current square and the value on the moving-to square does affect the expenditure.

In our algorithm, the map is an *n* by *n* matrix. Each value in the matrix denotes the height of that area. The subject can only move to the squares sharing bounds with the current square. If the value on the current square is less than the value of the moving-to square, the subject goes uphill with the energy expenditure twice the absolute difference of the two squares. If the value on the current square is greater than the value of the moving-to square, the subject goes downhill with the energy expenditure just the absolute difference of the two squares.

Our algorithm can be decomposed into several steps. First, it starts with a 3 by 3 matrix. The destination is in the middle of the matrix. More specifically, the destination is in the second row and the second column of this matrix.

Second, the optimal costs of moving from any of the four squares sharing boundaries with the destination can be calculated. They are just the cost of moving from that square to the destination. The four squares are marked as considered.

Third, based on the current information, for all the unconsidered squares, the optimal cost from each square to the destination is calculated. In the 3 by 3 matrix, the unconsidered squares are the four squares in the four corners. Each of them share boundaries with two considered squares. The minimum energy expenditure from an unconsidered square *i* to the destination can be illustrated as follows.

$$E_1 = E(i,x) + E(x,d)$$

$$E_2 = E(i,y) + E(y,d)$$

$$E(i,d) = min(E_1, E_2)$$

Here *x* and *y* denote the two squares sharing boundaries with the square *i*, and *d* denotes the destination. *E(i,x)* is the expenditure of moving from *i* to *x*; *E(x,d)* is the expenditure of moving from *x* to the destination. Obviously, there are two possible routes for moving from *i* to the destination. The expenditure for moving from each of the routes is calculated and are denoted by E_1 and E_2 respectively. Then the minimum expenditure from *i* to the destination is the smaller of E_1 and E_2.

Fourth, after calculating the minimum expenditure from each square in the 3 by 3 matrix to the destination, we expand the matrix two rows and two columns wider. We expand the matrix symmetrically. That is, a new row and a new column is added to be the first row and the first column. And a new row and a new column is added to be the last row and the last column. Now all the minimum expenditures calculated for the old matrix are regarded as only *tentative* optimal expenditures. Since new information is now to be incorporated, the true optimal expenditure for those squares might be changed.

We have discussed that the workload of going through the whole matrix to identify every possible route, and then selecting the one with the minimum energy expenditure is overwhelming for a personal computer, especially when the matrix is big. Using tentative optimal values can save a lot of workload of calculation. However, the new rows and columns have no tentative optimal values assigned. Thus, fifth, we have to assign tentative optimal values on them before we do the optimal calculation. In this step, we treat different types of squares differently.

The first type is the second square in the first row, first column, last row, and last column respectively. They are chosen as an entry point of assigning tentative optimal values on the new squares. And they have to be adjusted later. In this step, the moving paths of the four squares are considered compulsory at this stage of the algorithm (they will be updated if necessary later). This procedure is illustrated in Figure 1.25.

	x_1			
x_2	y_1		y_2	x_3
		d		
	y_3			
	x_4			

Figure 1.25: The First Type Squares in Fifth Step

x_1, x_2, x_3, x_4 are all the first type squares. In order to calculate the tentative optimal values, the algorithm makes it compulsory to move from x_1 or x_2 to y_1, from x_3 to y_2, and from x_4 to y_3. Thus the tentative optimal expenditure from *i* to *d* can be calculated by the following expression:

$$E(x, d) = E(x, y) + E(y, d)$$

Here *E(x,y)* is the actual expenditure from *x* to *y*, and *E(y,d)* is the tentative optimal expenditure from *y* to *d*, which has been calculated in the third step.

The second type squares can be separated into two parts. One part starts from the third column and ends at the penultimate column of the first row and the last row respectively. The other part starts from the third row and ends at the penultimate row of the first column and the last column respectively. They are illustrated in Figure 1.26.

	z	x^*	x	
z		y	y	z
x^*	y	d	y	x^*
x	y	y	y	x
	z	x^*	x	

Figure 1.26: The Second Type Squares in Fifth Step

In Figure 1.26, x^* and *x* are both what we call type two squares. The only difference between x^* and *x* is that, x^* can be reached from z. Or more specifically, there are two choices of moving from x^*, moving to *y* or to *z*. And there are also two ways of moving from *x*, moving to x^* or moving to *y*. The tentative optimal expenditure of x^* can be expressed as follows.

$$E_1 = E(x^*, y) + E(y, d)$$

$$E_2 = E(x^*, z) + E(z, d)$$

$$E(x^*, d) = min(E_1, E_2)$$

Here $E(x^*, y)$ and $E(x^*, z)$ are the actual expenditures of moving from x^* to *y* or *z* respectively. $E(y, d)$ and $E(z, d)$ are the tentative optimal expenditures from *y* or *z* to the destination respectively. And in the same way, we can calculate the tentative optimal expenditure from *x.* We just have to replace *z* with x^*.

The third type of squares is the four squares in the corners. They are illustrated in Figure 1.27.

X	*y*		*y*	*x*
z				*z*
		d		
Z				*z*
X	*y*		*y*	*x*

Figure 1.27: The Third Type Squares in Fifth Step

All of the *x* are the third type squares. Moving from any *x*, there are two possible moves, to *y* or to *z*. Thus the tentative optimal expenditure from *x* can be expressed as follows:

$$E_1 = E(x, y) + E(y, d)$$

$$E_2 = E(x, z) + E(z, d)$$

$$E(x, d) = min(E_1, E_2)$$

Here $E(x, y)$ and $E(x, z)$ are the actual expenditures of moving from *x* to *y* or *z* respectively. And $E(y, d)$ and $E(z, d)$ are the tentative optimal expenditures of moving from *y* or *z* to *d* respectively.

After all the three types of squares have been considered, all the new squares have been assigned a tentative optimal value. Now, in the sixth step, we have to adjust the tentative values for the whole matrix. There are also three types of squares here. The first type squares are the squares from the old matrix. Since new information comes, the map is changed. The tentative optimal expenditures of these squares might be also changed. It is illustrated in Figure 1.28.

	v			
u	*x*	*y*		
	z	*d*		

Figure 1.28: The First Type Squares for Adjusting

The values adjusted in this step are the tentative optimal expenditures of the squares with bold boundaries. In the example, we can see that, now from square *x*, there are four directions, *u, v, y ,z* to move to. Thus the tentative optimal expenditure can be adjusted as follows.

$$E_1 = E(x,u) + E(u,d)$$

$$E_2 = E(x,v) + E(v,d)$$

$$E_3 = E(x,y) + E(y,d)$$

$$E_4 = E(x,z) + E(z,d)$$

$$E_5 = E(x,d)$$

$$E^*(x,d) = min(E_1, E_2, E_3, E_4, E_5)$$

Here *E(x,u), E(x,v), E(x,y)*, and *E(x,z)* are the actual expenditure of moving from *x* to *u, v, y, z* respectively. *E(u,d), E(v,d), E(y,d), E(z,d)* and *E(x,d)* are the tentative optimal expenditures of moving from *u, v, y, z, x* to the destination respectively. And $E^*(x,d)$ denotes the adjusted value of *E(x,d)*.

The second type squares for adjusting are the squares on the boundaries of the new matrix, except the squares in the four corners. They are illustrated in Figure 1.29.

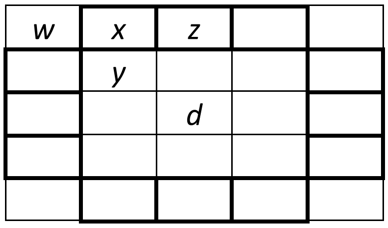


Figure 1.29: The Second Type Squares for Adjusting

The values adjusted in this step are the tentative optimal expenditures of the squares with bold boundaries in Figure 1.30. In the illustrated example, from square *x*, there are three possible squares, *w, y, z,* to move to. Thus the tentative optimal expenditure can be adjusted as follows.

$$E_1 = E(x,w) + E(u,w)$$

$$E_2 = E(x,y) + E(y,d)$$

$$E_3 = E(x,z) + E(z,d)$$

$$E_4 = E(x,d)$$

$$E^*(x,d) = min(E_1, E_2, E_3, E_4)$$

Here *E(x,w), E(x,y),* and *E(x,z)* are the actual expenditure of moving from *x* to *w, y, z* respectively. *E(w,d), E(y,d), E(z,d)* and *E(x,d)* are the tentative optimal expenditure of moving from *w, y, z, x* to the destination respectively. And $E^*(x,d)$ denotes the adjusted value of *E(x,d)*.

The third type squares for adjusting are the squares in the four corners. They can be illustrated in Figure 1.30.

The values adjusted in this step are the tentative optimal expenditures of the squares with bold boundaries in Figure 1.30.

x	*y*			
z				
		d		

Figure 1.30: The Third Type Squares for Adjusting

In the illustrated example, from square *x* there are two possible squares, *y*, and *z*, to move to.

Thus the tentative optimal expenditures can be adjusted as follows.

$$E_1 = E(x,y) + E(y,d)$$

$$E_2 = E(x,z) + E(z,d)$$

$$E_3 = E(x,d)$$

$$E^*(x,d) = min(E_1, E_2, E_3)$$

Here *E(x,y)* and *E(x,z)* are the actual expenditures of moving from *x* to *y* and *z* respectively. *E(y,d), E(z,d)* and *E(x,d)* are the tentative optimal expenditures of moving from *y*, *z* and *x* to the destination respectively. And $E^*(x,d)$ denotes the adjusted value of *E(x,d).*

After all the tentative optimal expenditures in the matrix have been adjusted, we can go onto the seventh step. This step is the most important one. Each square on the map is assigned an optimisation equation. By solving the simultaneous equations of optimisation, the actual optimal energy expenditure for each square of the current matrix can be calculated. In the equations set, there are three different types of equations, corresponding to three different types of square positions in the matrix. They are just the three types of squares we have examined in the sixth step.

They are squares in the middle of the matrix, squares on the boundaries, and squares in the corners. Thus, we can use Figure 1.28, Figure 1.29, and Figure 1.30 to describe the three types of equations.

The first type of equations is for the squares with bold boundaries in Figure 1.28. The equation *F(x)* for these can be illustrated as follows.

$$E_1 = E(x,u) + E(u,d)$$

$$E_2 = E(x,v) + E(v,d)$$

$$E_3 = E(x,y) + E(y,d)$$

$$E_4 = E(x,z) + E(z,d)$$

$$F(x) = E(x,d) - min(E_1, E_2, E_3, E_4)$$

The second type of equations is for the squares with bold boundaries in Figure 1.29. The equation *F(x)* can be illustrated as follows.

$$E_1 = E(x,w) + E(u,w)$$

$$E_2 = E(x,y) + E(y,d)$$

$$E_3 = E(x,z) + E(z,d)$$

$$F(x) = E(x,d) - min(E_1, E_2, E_3)$$

And the third type of equations is for the squares with bold boundaries in Figure 1.30. The equation *F(x)* for these can be illustrated as follows.

$$E_1 = E(x,y) + E(y,d)$$

$$E_2 = E(x,z) + E(z,d)$$

$$F(x) = E(x,d) - min(E_1, E_2)$$

Once all the *F(x)* in the equations are set equal zero, the simultaneous functions are solved. And then the *E(x,d)* is the actual optimal energy expenditure from square x to the destination for the current map.

In order to solving the simultaneous equations, there is a starting value for each $E(x,d)$. Then the computer adjust the starting value and searches for the set of $E(x,d)$ to make all the $F(x)$ equal to zero simultaneously. If the map is large enough, and if the variance of the heights on the map is big, the workload of calculation is overwhelming for a personal computer. If the start value of $E(x,d)$ is not chosen carefully, the calculation project may end up without convergence. That means the simultaneously equations cannot be solved by the computer. Thus, the starting value is very important. An appropriate set of starting value eases the calculation burden. Actually, all we were doing before the seventh step is trying to find an appropriate set for the starting value. At the end of the sixth step, the adjusted tentative optimal values are close to, if they are not the same as, the actual optimal values. Using them as the starting values, the simultaneous equations can be solved very quickly. More specifically, a 41 by 41 matrix can be solved within five minutes.

At the end of the seventh step, the actual optimal energy expenditure from each square to the destination of the current map is identified. In the eighth step, if the current map is not the whole map to be examined, the algorithm repeats from the fourth step to the seventh step. Once the current map is the whole map to be examined, the algorithm is finished.

We have mentioned that, in our context, sometimes the optimal route goes away from the destination. So we cannot only examine a map with the starting square at one corner and the destination at the centre. We have to examine a map on which the starting square and the destination are not on the boundary. There has to be sufficient space for possible roundabout steps.

Here is an example to illustrate this scenario. Figure 1.31 is a map with heights assigned. The square in the fourth row and the second column is the starting square, and the square in the middle of the matrix is the destination.

If we only estimate the 3 by 3 matrix with the bold boundaries, the optimal energy expenditure from the starting square to the destination is 3. However, if we estimate the whole map, there is a flat route from the starting square to the destination with zero energy expenditure. But it requires going away from the destination at the first step.

This kind of route cannot be identified if the map we examined is not large enough. Thus, in our algorithm, the starting square is never on the boundary of the map. Since the destination is always the centre of the map, we do not have to worry about it.

2	7	1	4	3
5	3	1	1	1
3	2	1	2	1
6	1	9	3	1
8	1	1	1	1

Figure 1.31: Map with Heights for Example

1.3.4 Finding the Optimal Route

1.3.4.1 Programming of the Algorithm

We have described above our algorithm for finding the optimal route with the minimum energy expenditure. It is obvious that the task cannot be done by hand. Thus we have to ask the computer to do it for us. We use MATLAB to do the programming. All the algorithms and equations are translated to code and then run on a personal computer.

Each of our four maps is a 200 by 200 matrix. However, due to time limitation, it is impossible for a subject to travel across the whole map. We deliberately set 10 seconds as a compulsory thinking time for each step to avoiding random clicking. Thus, if a subject travels across the whole map for each of the four journeys, it will at least cost more than four hours. It is too long for a laboratory experiment. Thus, we only use part of the map. For every journey, the destination is 15-squares up and 15-squares right of the starting square. By this way, the experiment can be limited in a tolerable time span.

If we put the destination in the centre and the starting square in the bottom-let corner of a matrix, it is a 31 by 31 matrix. But we have discussed above that if we only examine a matrix with the starting square on the boundary, the true optimal route of the map may not be identified. Since the optimal route may go out of the matrix.

Thus, we have to make sure that outside the row and column which contains the starting square, there is enough space to be examined to allow the optimal route going away from the destination at some point. Here we expand five rows down and five columns left to the starting square. As in our algorithm, the matrix is expanding symmetrically. So we have to also expand five rows up and five rows right to the matrix. Now the matrix becomes a 41 by 41 matrix. This is the map we examined in the MATLAB program.

The program outputs a 41 by 41 matrix. The number in each cell of the matrix measures the minimum energy expenditure from that cell to the destination. For obtaining the energy expenditure of the optimal route, we only need to find the position of the starting square and see the value assigned on it.

With the minimum energy expenditure, we can identify the optimal route. This process is also programmed by MATLAB. And it can be described as follows. First, the starting square is marked as the current square. Second, the four squares sharing boundaries with the current square is examined one after one by the following function.

$$E(c,d) = E(c,i) + E(i,d)$$

$E(c,d)$ is the minimum energy expenditure from the current square to the destination. $E(c,i)$ is the energy expenditure from the current square to square *i* which shares a boundary with it. And $E(i,d)$ is the minimum energy expenditure from square *i* to the destination.

If a square *i* satisfies this function, the square is a step on the optimal route. The starting square is marked as visited. And square *i* is marked as the current square. The process is repeated till the destination is marked as visited. Then the optimal route is identified.

It is possible that two squares sharing boundary with the current square can both be a step on the optimal route. This makes the situation complicated. The optimal route may not be unique on a map. But as we have obtained the minimum energy expenditure, we can control the optimal payoff of each journey. We do not have to examine a subject's moving track to see if she or he deviates from a specific optimal route. We only have to see if his or her payoff of that journey deviates from the optimal payoff. This is more efficient for the analysis of the results.

1.3.4.2 Optimal Routes

As mentioned in the experimental design, there are four different maps in any one session of the experiment. Figures 1.32 to Figure 1.35 show the optimal routes for the four maps separately. If the subject follows the green squares one after one, she or he can reach the destination with the least energy expenditure. These routes were identified by our algorithm which we have introduced above, given that the full information is provided to MATLAB.

At a first glance the four optimal routes have one thing in common – they zigzag. None of the optimal routes is straight to the destination (which is always to the up and right of the initial square in our experiment). Each of the routes sometimes goes in the opposite direction to that towards the destination. As we have mentioned before, the optimal route may not be unique. But we have checked the maps one after one and deliberately modified some of the heights slightly to make sure that there is no “flat” optimal route. Here the word “flat” means that go directly to the destination without changes of directions.

For Figure 1.32, the optimal route is within the 15 by 15 matrix in which the initial square is in the down-left corner and the destination is the upper-left corner. However, even in this case, the optimal route is not straight. It requires subjects to go down at some point, which is opposite to the position of the destination, if they want to obtain least energy expenditure.

In Figure 1.33, the optimal route is more zigzagged than that in Figure 1.32. Moreover, some part of the route is outside the bounds of the 15 by 15 matrix. Subjects have to go in the opposite direction at the very beginning if they follow the optimal route. And when they approach the destination, they also have to go a little bit beyond the destination and turn down to it at some point.

In Figure 1.34, the optimal route is also outside the 15 by 15 matrix for some part. It is clear to see that, subjects have to keep being outside of the matrix for about half way, and then turn to the same direction with the destination, approaching it in a zigzag manner.

Since in Figure 1.35, the variance of the heights on the map is much larger than in the other three maps, the optimal route is not so zigzag. But subjects still have to go a little bit out of the 15 by 15 matrix and then go back to the destination in the end of their journey, if they are following the optimal route.

330	346	384	376	356	362	355	351	333	358	335	314	317	313	294	275	270	271
337	355	375	376	365	379	363	362	339	349	333	336	323	302	296	292	264	268
342	346	363	375	375	380	383	360	353	355	356	327	307	318	305	298	290	265
313	329	332	348	349	389	382	363	359	373	355	342	338	336	318	309	288	262
306	309	318	327	352	363	384	381	364	378	356	363	347	333	305	294	285	271
332	338	327	318	330	348	369	377	364	374	363	343	322	313	306	317	303	281
314	332	343	349	336	332	363	380	377	372	360	353	340	323	337	325	299	299
300	307	314	333	343	337	350	380	381	372	379	366	359	346	346	328	312	298
305	323	312	328	357	353	365	386	387	377	380	358	344	334	320	303	294	294
318	325	342	335	356	375	365	384	381	378	385	379	357	334	316	317	316	319
331	330	333	347	349	339	369	374	381	378	386	370	359	358	341	338	336	328
313	310	317	334	320	324	353	349	362	377	384	375	356	348	356	359	339	337
291	305	325	319	306	431	436	435	349	375	372	360	360	329	326	337	321	310
291	290	306	310	298	305	311	336	354	378	373	354	329	316	314	306	301	296
277	278	287	388	397	308	421	317	337	356	358	338	326	338	341	327	321	313
308	283	269	293	303	300	409	330	350	357	351	339	343	364	342	320	304	298
325	296	287	284	407	325	328	321	329	344	345	346	355	358	327	309	324	309
322	324	408	276	295	308	300	304	323	343	356	352	361	371	338	330	326	304
341	333	301	277	297	295	314	317	327	334	361	373	367	351	326	318	312	310
348	339	301	272	297	321	330	349	346	348	362	379	367	357	341	309	306	322
342	327	298	279	292	296	300	315	323	341	358	376	355	333	325	314	336	329
342	323	298	281	310	317	304	307	319	343	367	386	362	344	320	334	354	345
339	326	319	286	291	312	321	326	331	323	356	374	351	344	324	342	367	373

Figure 1.32: Optimal Route for Journey 1

330	346	384	376	356	362	355	351	333	358	335	314	317	313	294	275	270	271
337	355	375	376	365	379	363	362	339	349	333	336	323	302	296	292	264	268
342	346	363	375	375	380	383	360	353	355	356	327	307	318	305	298	290	265
313	329	332	348	349	389	382	363	359	373	355	342	338	336	318	309	288	262
306	309	318	327	352	363	384	381	364	378	356	363	347	333	305	294	285	271
332	338	327	318	330	348	369	377	364	374	363	343	322	313	306	317	303	281
314	332	343	349	336	332	363	380	377	372	360	353	340	323	337	325	299	299
300	307	314	333	343	337	350	380	381	372	379	366	359	346	346	328	312	298
305	323	312	328	357	353	365	386	387	377	380	358	344	334	320	303	294	294
318	325	342	335	356	375	365	384	381	378	385	379	357	334	316	317	316	319
331	330	333	347	349	339	369	374	381	378	386	370	359	358	341	338	336	328
313	310	317	334	320	324	353	349	362	377	384	375	356	348	356	359	339	337
291	305	325	319	306	431	436	435	349	375	372	360	360	329	326	337	321	310
291	290	306	310	298	305	311	336	354	378	373	354	329	316	314	306	301	296
277	278	287	388	397	308	421	317	337	356	358	338	326	338	341	327	321	313
308	283	269	293	303	300	409	330	350	357	351	339	343	364	342	320	304	298
325	296	287	284	407	325	328	321	329	344	345	346	355	358	327	309	324	309
322	324	408	276	295	308	300	304	323	343	356	352	361	371	338	330	326	304
341	333	301	277	297	295	314	317	327	334	361	373	367	351	326	318	312	310
348	339	301	272	297	321	330	349	346	348	362	379	367	357	341	309	306	322
342	327	298	279	292	296	300	315	323	341	358	376	355	333	325	314	336	329
342	323	298	281	310	317	304	307	319	343	367	386	362	344	320	334	354	345
339	326	319	286	291	312	321	326	331	323	356	374	351	344	324	342	367	373

Figure 1.33: Optimal Route for Journey 2

133	131	126	123	125	132	133	131	134	129	125	139	129	105	88	90	103	110	121	145	170
123	121	114	112	113	123	130	121	116	116	104	101	93	88	91	90	97	125	137	206	201
118	115	109	106	106	104	104	109	103	101	101	99	91	90	98	98	97	163	187	190	168
119	125	117	111	106	104	113	125	127	107	100	99	97	96	134	187	151	147	173	223	212
124	132	125	126	123	114	123	113	106	102	111	115	124	142	206	198	125	201	201	219	205
160	164	168	150	124	112	123	119	133	135	171	229	253	245	173	126	130	187	211	261	200
197	197	191	154	121	113	135	170	254	239	291	254	205	207	173	197	144	161	222	287	228
175	153	136	125	128	175	231	158	171	243	372	248	295	277	233	258	175	174	185	267	244
147	145	149	220	256	271	208	153	196	213	360	256	324	273	268	211	187	194	191	300	305
277	250	180	223	235	201	199	186	265	302	306	252	348	326	254	201	192	197	201	292	314
332	304	262	205	246	176	241	198	277	384	318	293	335	299	218	190	182	207	224	245	325
397	360	339	274	217	191	327	288	349	367	435	322	283	205	203	251	181	190	206	221	327
411	369	351	299	233	287	376	396	439	430	415	363	299	247	221	270	177	189	178	276	306
365	349	325	314	318	344	423	415	403	387	380	343	337	322	232	286	183	219	206	305	316
305	333	333	339	341	361	396	384	369	356	344	295	298	302	353	298	187	232	237	300	299
240	348	356	351	390	378	416	390	380	348	337	265	288	281	322	243	202	204	219	286	280
243	264	382	376	404	363	397	378	387	342	332	245	296	270	339	262	249	178	205	255	244
244	244	374	399	396	322	368	344	362	287	277	225	281	252	323	350	287	180	180	206	215
184	257	358	388	386	302	287	328	318	235	278	232	265	258	308	320	281	200	190	202	192
222	326	340	347	291	257	236	268	291	287	259	254	257	299	271	309	310	236	161	175	216
205	263	266	299	256	271	219	238	292	334	234	265	240	326	257	312	270	231	179	172	230
196	205	240	235	220	248	209	191	223	294	253	222	178	236	248	238	230	256	214	226	193
167	177	188	157	157	155	160	177	207	225	250	177	168	222	226	204	205	214	201	229	244
175	159	145	140	139	154	205	200	195	194	195	184	193	228	210	216	200	178	211	220	253
137	151	186	177	200	182	209	205	174	175	174	213	191	192	237	181	193	223	247	279	248
144	181	259	275	280	197	185	233	158	168	168	164	186	192	233	214	181	221	240	267	249
214	161	246	245	253	203	151	195	163	168	205	179	199	259	223	197	173	206	286	233	226

Figure 1.34: Optimal Route for Journey 3

199	201	198	196	198	197	199	199	213	216	220	235	352	398	367	473	618	467
204	205	202	200	201	201	202	203	211	219	252	329	383	416	529	560	727	811
209	208	205	204	204	207	208	213	231	239	251	298	376	527	681	847	1048	1051
217	213	207	206	208	212	213	232	245	263	328	475	562	622	709	893	1137	1054
225	216	208	210	214	216	222	262	286	358	416	458	511	629	864	965	1145	1102
226	215	211	213	221	222	254	268	348	459	593	669	573	706	1017	1117	919	1015
228	217	215	220	226	252	290	367	477	584	686	819	790	945	1031	985	814	767
230	219	218	228	243	323	377	472	754	849	942	966	1050	840	823	724	651	701
236	222	223	236	284	346	363	497	709	800	862	764	848	778	802	832	598	820
238	226	226	292	335	351	427	476	550	622	874	695	656	620	859	793	632	789
242	229	267	329	338	362	376	412	522	693	679	679	576	667	652	564	666	905
248	232	338	320	353	293	338	430	539	566	453	462	472	488	501	650	768	823
254	236	282	295	291	314	305	373	453	451	428	462	496	468	414	481	492	604
259	247	238	239	257	295	312	355	337	358	371	394	402	433	460	463	429	429
283	258	247	240	246	271	385	303	396	423	415	438	422	430	446	416	435	386
307	276	262	252	248	260	281	323	440	447	384	395	358	328	314	316	332	333
339	309	279	273	264	259	270	373	423	444	342	287	281	293	306	338	336	294
369	356	338	314	315	276	281	268	268	272	295	341	379	350	398	444	444	456
384	361	352	328	347	292	412	417	385	330	392	398	482	429	492	509	552	504
441	410	352	336	316	317	506	643	519	412	370	469	527	549	567	583	552	561
642	636	563	510	396	406	621	671	520	446	398	435	458	429	470	456	517	564
507	447	395	432	488	551	617	591	513	480	471	474	555	513	546	564	567	588
441	391	399	476	587	701	729	669	597	624	596	530	537	538	586	629	699	675

Figure 1.35: Optimal Route for Journey 4

1.4 Empirical results and discussion (With Fog)

The calculated optimal route is without fog. However, we have discussed before, in the real world, full information is not always available. The experiment is aimed to investigate how people in a world with fog behave and how far their behaviour is from the optimal route without fog.

In section 1.4, we introduced our algorithm to identify the optimal route without fog. Those optimal routes, as shown in Figure 1.32 to Figure 1.35, are quite zigzagged. Sometimes, the optimal steps on the route are away from the destination. Obviously, without fog, it is possible to work out the optimal route. However, with fog, it is impossible to obtain an optimal strategy. An optimal route cannot be identified, since information is not sufficient.

Given the task of reaching the destination with the least expenditure of energy, subjects cannot work out an optimal strategy to follow. But they do have some way to try to approach the optimal strategy as closely as possible, which are revealed in their behaviour. What we have done is to observe their behaviour, or more precisely, their decisions on each step, and to try and describe their underlying strategies.

In a word, optimal routes in the four maps cannot be identified when there is fog. Subjects cannot go straight to the destination if they would like to maximise their utility/payoff. Thus, intuitively, the payment they received would be quite different from the optimal payment. However, we found that, some of the subjects get payments which are very close to or even equal to the optimal payoff. In the experiment May 2013, the optimal payment was £10 for each map, and in the experiment November 2013, it was £15.

In the following section, the data from the experiment is analysed. Section 1.4.1 describes a brief summary of the average payoff across journeys and treatments. In section 1.4.2, four possible strategies are tested to see how much of the subjects' behaviour can be explained.

1.4.1 At First Glance

We implemented the experiment twice, one in May 2013 and the other in November 2013. The only difference between the two experiments was the payoff. In the second experiment, we scaled up the payoff. So the optimal payoff was £15 instead of £10. Because the scales of payoff are different, we summarise the payoff of the two experiments separately.

Table 1.4 is a brief summary of the average payoff of the experiment carried out in May 2013. The optimal payoff for each journey was £10. It can be seen from the table that there is an obvious difference between the average payoff with fog (£6.992) and the optimal payoff without fog (£10). The average payoff for all of the four treatments is a little under 70% of the optimal payoff.

There is an interesting phenomenon which can be observed from Table 1.4. In treatment 1, for which both information quality and quantity are high, the average payoff is £6.535 – the lowest of the four treatments. However, in treatment 4, for which both information quality and information quantity are low, the average payoff is £7.177 – the highest of the four treatments. This is counter-intuitive.

Table 1.5 is a brief summary of the average payoffs in the experiment carried out in November 2013. In this experiment, we used the exactly same map, but scaled up the payoffs. The optimal payoff here is £15 for each journey.

We can see that, in Table 1.5, the result is consistent with Table 1.4. In treatment 1, both the information quantity and quality are high, but the average payoff is the lowest out of the four treatments. However, in treatment 4, with both low information quality and quantity, the average payoff is the highest out of the four.

If we go a little bit more into detail, we can see that, in treatment 2, with high information quality but low information quantity, the average payoff is lower than treatment 3, with low information quality but high information quantity.

One of the possible interpretations of this counter-intuitive phenomenon might be that too much information makes subjects confused. The mathematical-computational ability of human brain is limited. For example, treatment 1 provides information with both high quantity and quality. If a subject wants to take all the available information into account the workload of calculation is huge. It exceeds the computation ability of an ordinary person. The subject may directly give up calculating, or end up with a wrong result due to the difficulty of processing so much information. Thus, his or her behaviour may deviate from the optimal choice. In treatment 4, information is provided with both low quantity and quality. The workload for computation is not as heavy as in treatment 1. The subject can use such vague information to work out a decision which may deviated from the optimal decision, but which leads to an acceptable outcome. Such a decision made in treatment 4 is possibly better than the decision made in treatment 1. Since quality of calculation in treatment 4 is better than in treatment 1.

Payoff Summary (£) Experiment May 2013					
	J1	J2	J3	J4	Average
T1	6.743	6.323	7.448	5.628	6.535
T2	8.533	6.277	7.858	5.703	7.092
T3	7.905	7.198	8.083	5.468	7.163
T4	8.337	7.243	7.623	5.503	7.177
Average	7.879	6.760	7.753	5.575	6.992

Table 1.4: Payoff Summary for Experiment May 2013

Payoff Summary (£) Experiment November 2013					
	J1	J2	J3	J4	Average
T1	11.704	10.279	10.004	8.115	10.025
T2	13.05	9.8963	11.783	7.3575	10.522
T3	11.914	10.455	12.105	7.9838	10.614
T4	12.314	9.825	11.686	8.7341	10.640
Average	12.245	10.114	11.394	8.048	10.450

Table 1.5: Payoff Summary for Experiment November 2013

1.4.2 Strategy Tests

From the experiment, we observed subjects' behaviour, and have recorded each step of their decisions.

If the subject follows the optimal route, in each journey she or he is expected to be paid £10 in the first experiment and £15 in the second experiment. But from Tables 1.4 and 1.5, we can see that the average payoff for any of the four journeys is different from that of the optimal payoff. Thus at least the majority of the subjects are not following the optimal route. It is not surprising that they do not, since due to the fog, they do not have full information about the terrain.

But they must have their own strategies to make decisions when they were travelling across the map. We have observed their decisions and have tried to identify the strategies that they might have been following. What we have done is to construct some possible strategies (which seem to be suggested by their behaviour), and then to explore which one fits the subjects' behaviour best.

1.4.2.1 The Myopic Strategy

The first strategy we test is called the "Myopic Strategy" (MS). In this strategy, the subject is assumed to be very myopic. She or he understands the algorithm of the energy expenditure of moving. But she or he ignores all the information more than one square away from his or her current position. Thus she or he moves to and only moves to the square with the minimum energy expenditure. All the four possible directions are taken into consideration. Once a square is visited, it will never be visited again, because the expenditure of moving is non-negative; non-necessary energy expenditure may occur if a square is revisited. For example, the total expenditure of moving from A to B, then back to A, and then to C, is at least as much as the expenditure of moving from A to C directly.

We have examined the subjects' decisions step by step. If in a specific step, the subject's moving decision follows the MS, the step is regarded as "fitting" this strategy. In some steps, there may be more than one square that meets the minimum moving cost. That step is regarded as fitting this strategy if and only if the subject moves to one of those qualified squares. And the last step of a journey is unconditionally marked as fitting. Since if the subject's current position is next to the destination, his or her best choice is just move to the destination, no matter how much the expenditure of moving is to the other squares around.

Table 1.6 shows the result of the MS Test across journeys and treatments. The figures in this table record the percentage of moves that are consistent with the Myopic Strategy. On average, the subjects' behaviour in journey 2 fits the MS best out of the four journeys, which is 57.19%; and the data in journey 3 fits this strategy worst, with just 51.44% of the decisions consistent with the Myopic Strategy. Similarly, the subjects' behaviour in treatment 4 fits the MS best out of the four treatments; and their behaviour in treatment 2 fits this strategy worst (just 53.73% consistent with it). More specifically, on average, journey 2 in treatment 4 fits the MS best, at 60.19%, while journey 3 in treatment 2 fits this strategy worst, at 50.19%.

The Myopic Strategy					
	J1	J2	J3	J4	Average
T1	55.28%	55.62%	52.15%	57.75%	55.20%
T2	56.60%	56.46%	50.19%	51.67%	53.73%
T3	53.28%	57.00%	52.34%	58.16%	55.19%
T4	59.34%	60.19%	50.99%	56.17%	56.67%
Average	55.98%	57.19%	51.44%	55.93%	55.14%

Table 1.6: The Average Fit for Myopic Strategy

1.4.2.2 The Two Direction Myopic Strategy

The second strategy we seem to have observed in behaviour and which we test is called the "Two Direction Myopic Strategy" (TDMS). This strategy is very similar to the Myopic Strategy. The only difference between them is that, in TDMS, only two instead of four directions are taken into account. The subject is assumed to never move down or left which is opposite to the destination.

As we have done in MS, we have examined the subjects' decisions step by step. A specific step is regarded as "fitting" TDMS if and only if the subject's decision can be explained by this strategy. It is possible that in some steps, moving to either of the two directions consumes the same amount of energy. In this case, the step is regarded as fitting TDMS for the subject moves to either of the two directions. And the last step of each journey is unconditionally marked as fitting.

Table 1.7 shows the result of the TDMS Test across journeys and treatments. On average, the subjects' behaviour in journey 2 out of the four journeys fits this strategy best, at 82.56%. The data in journey 4 fits this strategy worst, at 70.50%. Similarly, it is obvious that the subjects' behaviour in treatment 4 out of the four treatments fits the TDMS best, at 79.13%. The subjects' behaviour in treatment 2 fits this strategy worst, at 75.20%. More specifically, on average, journey 2 in treatment 3 fits the TDMS best, at 84.76%, while journey 4 in treatment 4 fits the strategy worst, at 70.25%.

The Two-Direction-Myopic Strategy					
	J1	J2	J3	J4	Average
T1	77.01%	79.46%	78.26%	70.59%	76.33%
T2	77.44%	81.76%	73.78%	67.82%	75.20%
T3	75.41%	84.76%	81.28%	73.30%	78.69%
T4	79.41%	84.59%	82.28%	70.25%	79.13%
Average	77.23%	82.56%	78.75%	70.50%	77.26%

Table 1.7: The Average Fit for Two-Direction-Myopic Strategy

1.4.2.3 The Minimum Difference Strategy

The third strategy we seem to have observed in behaviour and which we test is called the "Minimum Difference Strategy" (MDS). In this strategy, the subject is assumed to be even more naive than in the Myopic Strategy. She or he does not use the algorithm to calculate the energy expenditure of moving. She or he only cares about the difference between the heights of his or her current position and the squares that she or he can move to. And in this strategy, she or he ignores all the information more than one square away from the current position. Thus, she or he moves to and only moves to the square with the minimum height difference of his or her current position. All the four possible directions are taken into account. And once a square is visited, it will never be visited again.

As we did for MS and TDMS, we have examined the subjects' decisions step by step. If in a specific step, the subject's decision follows the MDS, the step is regarded as "fitting" this strategy. In some steps, from the current position, there may be more than one possible squares that fit the strategy.

That step is regarded as “fitting” if and only if the subject moves to one of those qualified squares. And the last step of each journey is unconditionally regarded as fitting. Since if the subject’s current position is next to the destination, the only best choice is to move to the destination directly.

Table 1.8 shows the average fit of the MDS Test across journeys and treatments. On average, the subjects’ behaviour in journey 1 fits the MDS best out of the four journeys, at 58.27%. In journey 3 it fits the strategy worst, at 48.96%. Similarly, the subjects’ behaviour in treatment 4 fits the MDS best out of the four treatments, at 55.68%, while behaviour in treatment 2 fits this strategy worst, at 52.16%. More specifically, on average, journey 1 in treatment 4 fits the MDS best, at 61.54%, while journey 3 in treatment 2 fits this strategy worst, at 45.51%.

The Minimum-Difference Strategy					
	J1	J2	J3	J4	Average
T1	56.59%	54.00%	50.91%	51.10%	53.15%
T2	60.76%	52.75%	45.51%	49.60%	52.16%
T3	54.75%	55.03%	49.73%	54.06%	53.39%
T4	61.54%	57.32%	49.82%	54.02%	55.68%
Average	58.27%	54.67%	48.96%	52.11%	53.50%

Table 1.8: The Average Fit for Minimum-Difference Strategy

1.4.2.4 The Two Direction Minimum Difference Strategy

The fourth strategy we test is called the “Two Direction Minimum Difference Strategy” (TDMDS). This strategy is very similar to the Minimum Difference Strategy. The only difference between them is that, in TDMDS, only two instead of four directions are taken into account. The subject is assumed to never move down or left, which is in the opposite direction to the destination.

As we have done in MDS, we have examined the subjects' decisions step by step. A specific step is regarded as "fitting" TDMDS if and only if the subject's decision can be explained by this strategy. It is possible that in some steps, the heights of the two possible directions are the same. That is the differences between the current position and either of the two possible moving-to squares are the same. In this case, the step is regarded as fitting TDMDS if the subject moves to either of the two directions. And the last step of each journey is unconditionally regarded as fitting.

Table 1.9 shows the result of the TDMDS Test across journeys and treatments. On average, the subjects' behaviour in journey 2 out of the four journeys fits this strategy best, at 78.39%; and the data in journey 4 fits this strategy worst, at 72.20%. Similarly, the subjects' behaviour in treatment 4 out of the four treatments fits the TDMDS best, at 77.87%, while in treatment 2 fits this strategy worst, at 72.75%. More specifically, on average, journey 2 in treatment 3 fits the TDMDS best, at 81.02%, while journey 3 in treatment 2 fits this strategy worst, at 68.85%.

The Two-Direction-Minimum-Difference Strategy					
	J1	J2	J3	J4	Average
T1	74.39%	75.65%	75.79%	71.68%	74.38%
T2	75.71%	76.40%	68.85%	70.06%	72.75%
T3	73.79%	81.02%	77.40%	73.96%	76.54%
T4	77.45%	80.93%	79.80%	73.30%	77.87%
Average	75.24%	78.39%	75.27%	72.20%	75.28%

Table 1.9: The Average Fit for Two-Direction-Minimum-Difference Strategy

1.4.3 Discussion

We have observed the subjects' decisions and tested their behaviour against four different possible strategies. On average, none of them can explain the subjects' behaviour more than 90%. In a specific journey of a specific treatment, one strategy can at most explain about 85% of the subjects' decisions. However, in some journeys of some treatments, the strategies only 'explain' about 45% of the subjects' behaviour. It means that each of the four strategies have limited ability to explain the subjects' behaviour. One possible reason is that the strategies assume that subjects are extremely myopic: they simply ignore all information which is beyond the squares one squares away from their current position. In the experiment, they might actually have taken more information into account and applied some more sophisticated strategies at least in some specific steps.

Table 1.10 shows that, the two-direction strategies, TDMS, and TDMDS, fit subjects' behaviour better than the four-direction strategies, MS and MDS.

And compared with TDMDS, TDMS fits subjects' behaviour much better. In the table, TDMS is the best fitting strategy for journey 1, 2, and 3 across all the four treatments. TDMDS is the best fitting strategy just for journey 4, but also across all the four treatments.

One possible explanation of this phenomenon can be stated below. Subjects actually calculated the expenditure for each step in the experiment, but they were too tired to do careful computation in the last journey. In TDMS, subjects are assumed to understand the algorithm of the energy expenditure of moving. They know that moving uphill is more expensive than moving downhill. However, in TDMDS, subjects are assumed to only take into account the absolute difference between the heights of the current position and the squares that can be moved to. They do not use the algorithm to calculate the energy expenditure of moving carefully. In the experiment, journeys come one after one, in the order of 1 to 4. In the first three journeys, TDMS fits better than TDMDS. It implies that subjects were calculating the energy expenditure of moving carefully. But in journey 4, TDMDS fits better than TDMS. It might because subjects had done too much calculation and got bored or tired. They adjusted the strategy to roughly estimate the expenditure rather than work out the accurate result. In different treatments, subjects were different. But for all the treatments, TDMS is consistently fitting better than TDMDS for journey 1, 2, and 3, while TDMDS is consistently fitting better than TDMS for journey 4. It increases the credibility of the above explanation.

Table 1.11 shows that the four-direction strategies, MS, and MDS, fit subjects' behaviour worse than the two-direction strategies, TDMS and TDMDS. Compared with MS, MDS fits subjects' behaviour even worse. In the table, MS is the worst fitting strategy for journey 1 across all the four treatments. And TDMDS is the worst fitting strategy for journey 2, 3, and 4, also across all the four treatments.

Overall, MDS fits subjects' behaviour worse than MS. One possible explanation is a possible filtration of behaviour. Those subjects considering four directions rather than two are more likely to be less lazy (or more sophisticated) than others. Thus they are more likely to calculate energy expenditure of moving carefully rather than just roughly estimating it. However, in journey 1, MS fits worse than MDS. It might because the subjects were not getting used to the workload of computation at the beginning of the experiment, and hence they only do rough estimation for the four directions. As practiced in journey 1, they adjusted their strategy to do the accurate calculation in the rest journeys.

Compared with the Two Direction Myopic Strategy, on average the Myopic Strategy fits each journey in each treatment worse. It is not surprising, since MS takes all the four directions into account. But in the experiment, if a subject has decided to ignore all the information one squares away from his or her current square, it implies that she or he is myopic. And she or he is very likely to also ignore the directions which do not directly lead to the destination. In the same way, the Two Direction Minimum Difference Strategy fits all the four journeys in all the four treatments better than MS.

Compared with the Minimum Difference Strategy, on average, MS fits all the four journeys in treatment 1 worse. But it fits all the four journeys in the other three treatments better than MDS. This is very reasonable. As we have said before, the subject might adjust his or her strategies map after map. She or he might try some simpler strategies in the early journeys, and then try more sophisticated ones later. MDS is simpler than MS, because it only care about the absolute difference between two squares. In journey 1, the first journey, the subject is very likely to try the simplest calculation strategy. As the experiment goes on, she or he might discover that the simple algorithm does not lead to good pay off, since the algorithm for the energy expenditure is not so simple. And then she or he adjust his or her strategy to care more about the actually energy expenditure for each possible step rather than the absolute difference between two squares.

Compared with MDS, TDMS fits all the four journeys in all the four treatments better. As we said in the comparison between MS and TDMS, if a subject employs a myopic strategy, she or he is very likely to ignore the directions which are in the opposite of the destination. Thus it is not surprising that TDMS fits better than MDS.

Compared with TDMDS, TDMS fits all the journeys in treatment 4 worse, but fits all the journeys in the other three treatments better. The only difference between TDMS and TDMDS is that, TDMDS only cares about the absolute difference between two squares. But TDMS takes energy expenditure into account. As we have mentioned above, the subject is probably adjusting his or her strategy journey after journey. TDMS fits worse than TDMDS in journey 4 maybe because the subject is already tired of calculating after the three long journeys. And she or he goes back to the simple calculation strategy.

And compared with TDMDS, MDS fits worse in all the journeys of all the treatments. The possible reason is very similar to the comparison of MS and TDMS: a myopic subject probably just ignores the directions which do not directly lead to the destination.

All the four strategies are myopic, and only take the very immediate information into account. But from the analysis results, it is obvious that the myopic strategies cannot explain the entire subjects' behaviour. The subject must have employed a more sophisticated strategy on at least some specific steps.

	J1	J2	J3	J4
T1	TDMS	TDMS	TDMS	TDMDS
T2	TDMS	TDMS	TDMS	TDMDS
T3	TDMS	TDMS	TDMS	TDMDS
T4	TDMS	TDMS	TDMS	TDMDS

Table 1.10: Best Fitting Strategies across Journeys and Treatments

	J1	J2	J3	J4
T1	MS	MDS	MDS	MDS
T2	MS	MDS	MDS	MDS
T3	MS	MDS	MDS	MDS
T4	MS	MDS	MDS	MDS

Table 1.11: Worst Fitting Strategies across Journeys and Treatments

1.5 Conclusion and Further Research

1.5.1 Conclusion

Our purpose is to investigate, without full information human behaviour in making a series of dynamic decisions. As is almost tautological, we can say that a rational person always tries to maximise his or her utility. An optimal strategy maximising utility can be worked out with full information. However, in the real world, information is not always sufficient. For example, theoretically, people can maximise their utility, given that they know the appropriate distribution of their income of every period in the future. In fact, people might have some rough ideas about their income of the very near future, but they cannot predict their income if the future is far away enough. Thus, in this situation, backward induction does not work. Nevertheless, given this constraint, rational people still want to maximise their utility. Then they must have other strategies to try to solve this problem.

We are interested in what people actually do. It is almost impossible to gather data from daily life. It is costly and inefficient to gather a series of decisions made by a specific person from the field. Even though it is possible, we cannot guarantee that we have gathered all the decisions the person made in this series, and all the information available when the decision was taken. Sometimes a decision might be implicit, and sometimes people might not be willing to report his or her every decision due to privacy or ethical issues.

Thus we can only gather data from laboratory experiments. We have designed an experiment called “fog”, which retains all the characters of the situation we are interested in. In the experiment, the subject is required to travel across four maps one after one. These maps are divided into squares. She or he starts from a square, and can only move to those squares which shares bounds with his or her current square. A journey ends once she or he arrives at the “destination” square. The four journeys are independent of each other. At the start of each journey, the subject is endowed with some money. While travelling, she or he has to consume some energy for each step. The expenditure of energy is dependent on the height of his or her current position and the height of the square she or he is moving to. When she or he arrives at the destination, she or he receives the payoff, which is the endowment minus the total energy consumed. In the experiment the subject could not receive a negative payoff; if the energy expenditure exceeded his or her endowment, the payoff is zero.

If the subject knows all the heights of the squares on the map, an optimal route can be calculated by our algorithm. However, in order to mimic the fact that people cannot predict the future precisely, the subject does not know the exact height of all the squares on the map. In order to reflect that people might have some vague information in the very near future, a range is presented on those squares which are not far away from the subject. The true value of the height is uniformly distributed in the range. And if the square is far away from the current position enough, the subject can see nothing.

We have four treatments in total. In all four treatments, the subject can see the exact height of the squares which share bounds with his or her current position. In treatment 1, the subject can see the height of the squares which are two squares away with range 20, and the height of the squares which are three squares away with range 50. In treatment 2, the subject can see the height of the squares which are two squares away with range 20, and cannot see anything if the square is further than that. In treatment 3, the subject can see the height of the squares which are two squares away with range 40, and the height of the squares which are three squares away with range 100. And in treatment 4, the subject can see the height of the squares which are two squares away with range 40, and can see nothing if the square is further than that.

We implemented this experiment twice. We recorded the subjects' decision of moving to which square in each step. And based on these results, we have done some analyses.

Four different strategies have been built to see which one can explain the subjects' behaviour better. The first one is called the Myopic Strategy, in which the subject considers all four possible directions to move, but only one step away. She or he moves to the square with the least energy expenditure. The second one is called Two Direction Myopic Strategy, in which the subject considers only the directions which are towards the destination and only one step away as well. She or he moves to the square with the least energy expenditure and which is towards to the destination. The third is called Minimum Difference Strategy. It is similar to MS, but considers the absolute difference between two squares instead of the energy expenditure. And the fourth one is called Two Direction Minimum Difference Strategy. It is similar to TDMS, but considers the absolute difference instead of the energy expenditure as well.

None of the four strategies can explain the subjects' behaviour 100%. More precisely, on average, they can only explain about 65% of the subjects' behaviour. And the percentage of the explanation varies across treatments and journeys. The best fitting one is Two-Direction-Myopic Strategy for journey 2 in treatment 3, which explains 84.76% of the subjects' behaviour. The worst fitting one is Minimum-Difference Strategy for journey 3 in treatment 2, which explains 45.51% of the subjects' behaviour. It is obviously that subject do use some forward strategies to maximise his or her utility. And she or he is very likely to adjust their strategies map after map.

1.5.2 Further Research

All four strategies discussed in this chapter are under the hypothesis that the subject is myopic. She or he only takes the information on the squares which shares bounds with his or her current square into account. This kind of information is immediately available, and is accurate, since the height on these squares is not distorted by fog. We can say that this kind of information is cheap but with high quality. Putting heavy weight on such information can be regarded as a sort of wise choice based on rational considerations.

However, the possible strategies are not restricted to these four. In the following we list some possible strategies that we are going to examine in the future.

The first strategy is the Two Squared Sophisticated Strategy (TSS). In this strategy, the subject is assumed to be sophisticated. She or he takes not only the squares sharing bounds with the current square but also the squares two squares away from the current square into account. The expected height for the square with fog is the middle number of the range, since the true height of that square is equally distributed within the range.

The second strategy is the All Information Sophisticated Strategy (AIS). In this strategy, the subject takes all information into account. If the subject is in the treatments which they can only see two-squares away, the strategy is exactly same as the TSS. If the subject is in the treatments which they can see three squares away, they take the squares sharing boundary with the current square, the squares two squares away, and the squares three squares away into account.

The third strategy is the Risk-Averse Strategy (RAS). In this strategy, the subject is assumed to be extremely risk-averse. She or he put heavy weight on the possible worst outcome, and thus chooses a path which will not lead to the undesirable outcome.

Chapter 2. Context Matters[1]

2.1 Introduction

Risk attitude is a crucial factor influencing economic behaviour. As a consequence, experimenters are interested in eliciting the risk-attitude of their subjects. This can be done in two ways: either *directly*, using the context of a particular experiment to estimate the risk-aversion that best explains behaviour; or *indirectly*, eliciting risk aversion in a separate part of the experiment, and using the elicited value to explain behaviour in the main experiment. This chapter is focused on the latter approach.

Economic theory posits that decisions under risk depend on how people evaluate, and hence decide between, risky lotteries. By these we mean lotteries where the outcomes are risky, and where the probabilities are known. Clearly how people evaluate lotteries depends not only on the lotteries, but also on the preference functionals of the decision-maker (DM). In the literature there are a number of proposed preference functionals, the best-known of which is the Expected Utility functional. All of these embody the idea of an underlying utility function *u(.)*; it is the degree of concavity of this when it is defined over money that indicates the degree of risk-aversion. It is this that we are trying to elicit.

There are a number of methods that are used in the literature to elicit risk aversion. Possibly the most popular is that known as the Holt-Laury *Price*

[1] This chapter is a joint work with John Hey, financially supported by Daniela Di Cagno.

List, introduced by Holt and Laury (2002), and which we will refer to as HL. While the detail may vary from application to application, the basic idea is simple: subjects are presented with an ordered list of pairwise choices and have to choose one of each pair. The list is ordered in that one of the two choices is steadily getting better or steadily getting worse as one goes through the list. There are many variants on the basic theme: sometimes one of the two choices is a certainty, and that is getting better or worse through the list; sometimes either just one or both of the choices are risky choices and one of them is getting better or worse through the list. Because of the ordered nature of the list, subjects *should* choose the option on one side up to a certain point thereafter choosing the option on the other side. Some experimenters force subjects to have a unique switch point; others leave it up to subjects. A rational subject never switches more than once, since one side of the list getting better or worse steadily. However, they may switch more than once due to an implementation error. Thus in our experiment, we force the subject to switch at most once.

A second method is to give a set of *Pairwise Choices,* but separately (not in a list) and not ordered. We will refer to this as PC. Typically the pairwise choices are presented in a random order. This has been used by Hey and Orme (1996), amongst many others. Some argue that this method, whilst being similar to that of Price Lists, avoids some potential biases associated with ordered lists. Frequently the pairwise choices are chosen such that they are distributed randomly over one or more Marschak-Machina triangles. Such a triangle is used to represent lotteries over a set of three outcomes. Two of the probabilities in the lotteries are plotted on the vertical and horizontal axes, while the third is the residual from one. The points at the vertices are certainties while points properly inside the triangle are lotteries.

A method which is elegant from a theoretical point of the view is the Becker-DeGroot-Marschak mechanism proposed by Becker *et al* (1964). This we will later denote by LC (Lottery Choice) because of the way that we implement it. The method centres on eliciting the value to a subject of a lottery – if we know the value that a subject places on a lottery with monetary outcomes, we can deduce the individual's attitude to risk over money. There are two variants of this mechanism that are used in the literature: one where the DM is told that they own the lottery, and hence have the right to play it out or to sell it; and one where the DM is offered the chance to buy the lottery, and, if so, to then play out the lottery. The subject's valuation of the lottery as a potential seller is the *minimum* price for which they would be willing to sell it, while the subject's valuation of the lottery as potential buyer is the *maximum* price for which they would be willing to buy it. Here we describe the mechanism as it relates to a potential buyer – the mechanism is the same, *mutatis mutandis*, if it relates to a potential buyer. The subject is asked to state a number; then a random device is activated, which produces a random number between the lowest amount in the lottery and the highest amount. If the random number is less than the stated number, then the subject buys the lottery at a price equal to the random number (and then plays out the lottery); if the random number is greater, then nothing happens and the subject stays as he or she was. If[2] the subject's preference functional is the expected utility functional then it can be shown that this mechanism is incentive compatible and reveals the subject's true evaluation of the lottery. The problem is that subjects do seem to have difficulty in understanding this mechanism, and a frequent criticism is that subjects understate their evaluation when acting as potential buyers and overstate it when acting as potential sellers.

The Allocation method, which we shall denote by AL, was originally pioneered by Loomes (1991). It was then revived by Andreoni and Miller (2002) in a social choice context, and later by Choi et al. (2007) in a risky choice context. This method involves giving the subject some experimental money to allocate between various states of the world, with specified probabilities for the various states, and, in some implementations, with given exchange rates between experimental money and real money for each of the states. This method seems easier for subjects to understand than BDM.

One clear difference between the methods is the information that the answers give. Pairwise Choices (on which Price Lists are built) merely tell us

[2] Though this is not necessarily true with other preference functionals.

which of two lotteries is preferred, but not by how much. In contrast both the LC and AL give us a continuous measure, which is (should be) the outcome of an optimising decision. This suggests that the latter two might be more informative. A discussion of the various methods can be found in Charness *et al* (2013). Other methods are also discussed there, including *the Balloon Analogue Risk Task* (Lejuez *et al* 2002), the *Gneezy and Potters* method (Gneezy and Potter 1997) – which is close to our allocation method – and the *Eckel and Grossman* method (Eckel and Grossman 2002) – which is a generalisation of the Pairwise Choice method to a decision over several lotteries. This can be further extended by asking subjects to *rank* the various lotteries in the list. This has also been used by Carbone and Hey (1994), Bateman *et al* (2015) and Loomes and Pogrebna (2014). We call this the *Ranking* method. It can be considered a special – discretised – case of the allocation method.

This chapter is a follow-up, and complement to, the paper by Loomes and Pogrebna (2014), in which the authors compare three of the elicitation methods described above – specifically Holt-Laury price lists, Ranking and Allocations. This chapter complements theirs, not only in the elicitation methods we consider, but also in that our experimental design (and crucially the numbers of problems asked for each method), as well as the data analysis, are completely different. We also consider a slightly different set of elicitation methods.

The purpose of this chapter is to report on the results of an experiment in which subjects were asked to perform each of the four methods described above. The chapter is organised as follows. In the next section we describe how our experiment was organised and how the various methods were implemented in it, giving more detail about each of the methods. As we adopt an econometric methodology of fitting preference functionals to the data, we specify in section 3 the preference functionals that we fit to the data and describe the functional forms that we assume, and the parameters in them that we estimate. In section 4, we describe how we analysed the data, detailing the stochastic assumptions that we make. Section 5 contains the results and section 6 concludes.

2.2 The Experimental Design and Implementation

2.2.1 Introduction

Our experiment was in four parts and different subjects took the parts in different orders. In total there are 24 different possible orders. Thus we had 24 subjects in each session to make sure that each order could be assigned to a subject. This design avoids the possibility of the experimental results being affected by a fixed presented order. Here we describe the four parts of the experiment. All parts of the experiment concerned lotteries. The complete set of tasks is attached in the appendices.

Throughout the experiment, the lotteries are visually on the subjects' computer screens in two dimensions, with the amount of money on the vertical axis and the chances on the horizontal axis. This is a different presentation than that used by Loomes and Pogrebna (2014). Theirs is more appropriate in their setting; ours is more appropriate in ours as we wanted finer divisions (in steps of 0.01 rather than 0.1).

If a particular lottery was chosen to pay out at the end of the experiment, the subject would draw a disk from a bag of 100 disks numbered from 1 to 100. The subject was paid the amount of money corresponding to the number on the disk. Let us give an example. Take the lottery shown in Figure 2.1; this represents a lottery where there is a 1 in 50 chance of gaining £5 and a 1 in 50 chance of gaining £15. If this was played out at the end of the experiment, if the numbered disk was between 1 and 50 inclusive, the subject was paid £5; if it was between 51 and 100 inclusive the subject was paid £15. One of the suggested advantages of this way of portraying lotteries is that the area of each bar on the graph indicates the expected value of the lottery.

In the experiment, for each problem, the confirm button is not enabled until five seconds has elapsed. Forcing subjects to wait at least five seconds before they can make a decision somewhat reduces the possiblity that they are just clicking without thinking.

At the end of the experiment, for each subject, one of the four parts is randomly selected to be real. And one problem is randomly selected from that part to be played out. Different subjects are paid according to different problems in different parts.

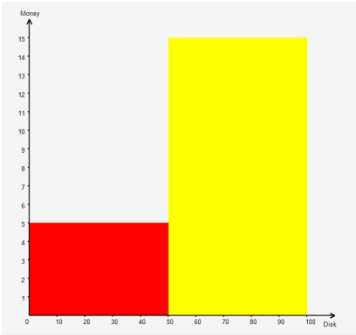


Figure 2.1: An Example of a Lottery in the Experiment

2.2.2 Holt-Laury Price List (HL)

The first part of the experiment presented a set of 48 *Holt-Laury price lists*, each containing 10 pairwise choices. In contrast, Loomes and Pogrebna (2014) had just 5 such lists. We used more because pre-experimental simulations indicated that to achieve accuracy in estimation we needed more. And we had a different objective to that of Loomes and Pogrebna (2014).

For each list, the subject has to choose one lottery in each pair which she or he wishes to be played out. When all the ten pairwise choices in the list have been assigned a decision, a confirm button appears on the screen. Then the subject can submit the decision and proceed to the next problem.

For each pair of lotteries, the left-hand side is always a risky lottery, and the right-hand side is always a certainty. Within a list, one side of the pairs is fixed. The other side is changed gradually through the list in a specific pattern. For example, if the risky lottery is fixed in the list, the certainty is steadily getting better from the first pair to the last. And if the certainty is fixed, the risky lottery is steadily getting worse through the list.

It is obvious to see that, in the list, the risky lottery higher up the list is always no worse than the risky lottery lower down; and the certainty lower down the list is always no worse than the certainty higher up. Thus, a rational subject switches at most once in a list. And the direction is always from a lottery to a certainty, if she or he switches at some point in the list.

Making decisions in a list can be abstracted as choosing a switch point. Or we can say that a subject's decisions reveal his or her certainty equivalence for a specific lottery. If she or he switches somewhere in a list, the certainty equivalence of the lottery is somewhere between the two pairwise choices that come just before and after the switch point. If the subject chooses all the lotteries in a list, it means that even the best certainty is not as good as the lottery, or even the worst lottery is not worse than the certainty. If the subject chooses all the certainties in a list, it means that even the worst certainty is better than the lottery, or even the best lottery is not better than the certainty.

According to the nature of HL, we have no reason not to force a subject to switch at most once in a list. In the experiment, if a subject selects the certainty in one pair of lotteries, the certainties in all the pairs after that pair are automatically selected. And similarly, if the lottery is selected in one pair, the lotteries in all the pairs after that pair are automatically selected. His kind of design can reduce the subject's manual mistakes. And it can also free the subject from clicking the pairs one by one.

An example is shown in Figures 2.2a and 2.2b; Figure 2.2a showing how it was first seen by the subject and Figure 2.2b showing it after its possible completion by a subject. These are screen shots from the experimental software; they appeared full-screen in the experimental interface.

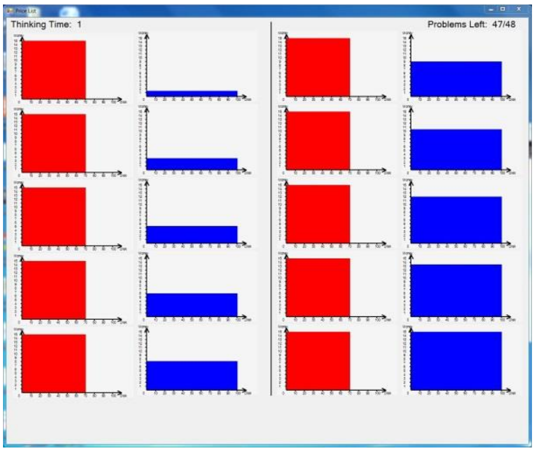


Figure 2.2a: Example of HL-1

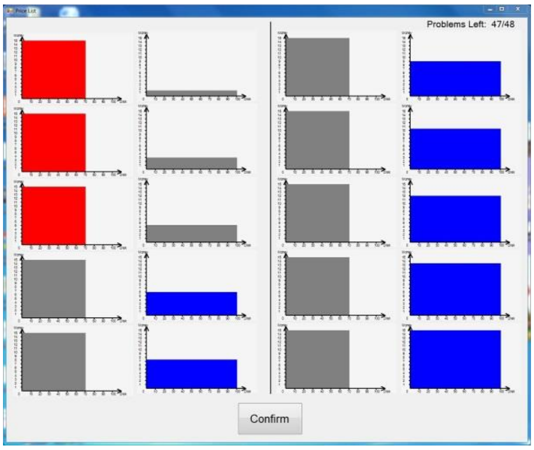


FIGURE 2.2b: Example of HL-2

Figure 2.2a gives an example of a Price List: in this example the thing that is staying constant is the lottery on the left (which is a 70% chance of £15 and a 30% chance of £0); the thing that is changing is that on the right – in this case a certainty – which increases from a certainty of £1.50 to a certainty of £15. Subjects were asked, for each pair in the list, to click on the preferred item; when doing so, the item on the other side turned grey. Figure 2.2b shows a possible set of responses – with the lottery being preferred until the certainty became £6. To avoid problems with subjects switching at several points within the list, the software forced subjects to choose a unique switching point. In contrast, Loomes and Pogrebna (2014) allowed subjects to switch at several points. But they write "the proportion of inconsistent participants in our data set ranged between 1.1% and 5.6%". It is not clear what to do with such subjects.

The 48 Price Lists spanned a variety of cases; details are attached in the appendices. We denote this method HL.

2.2.3 Pairwise Choices (PC)

The second part asked subjects to respond to 80 *pairwise choice* problems. Again the problems were chosen after pre-experimental simulations. The objective was to get a set of problems which would enable us to identify accurately the preference functional and its parameters.

In each problem, subjects face a pair of lotteries. Sometimes they are both risky lotteries, and sometimes one of them is risky and the other is a certainty. Subjects have to decide which one they prefer to be played out.

The decision-making implementation is straightforward. Subjects only need to click the “Left” or “Right” button. In order to avoid the situation where a subject’s decision is affected by the position of the lotteries presented, we randomized the two lotteries in a pair to be left or right. For example, in one problem, lottery A is on the left and lottery B is on the right for subject one. In the same problem for subject two, lottery A can be on the right and lottery B on the left.

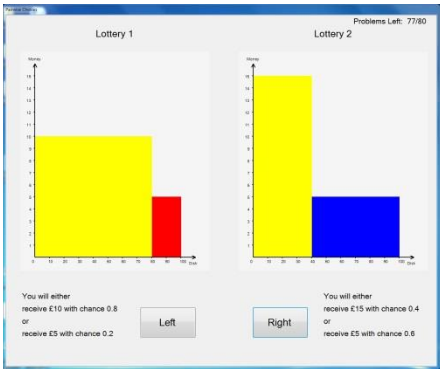


Figure 2.3: Example of PC

An example is shown in Figure 2.3. In this pairwise choice, subjects had to choose between a lottery which give an 80% chance of £10 and a 20% chance of £5 and a lottery which gives a 40% chance of £15 and a 60% chance of £5. The set of 80 pairwise choice problems spanned lotteries with outcomes of £0, £5, £10 and £15 with probabilities of 0, 0.2, 0.4, 0.6, 0.8 and 1.0; details are attached in the appendices. We denote this method PC.

2.2.4 Lottery Choices (LC)

The third part asked subjects to respond to 54 *Becker-DeGroot-Marschak* problems. Again, the problems were chosen after pre-experimental simulations. Typically, subjects are shown a lottery and asked to state their maximum willingness-to-pay or minimum willingness-to-accept for the lottery. Many experimenters have reported confusion among subjects in understanding this mechanism, so we adopted a new way of implementing it.

Suppose that we want to find the subject's certainty equivalent of a lottery which pays £x with probability p and £y with probability $1-p$, where $x>y$. The subject is asked to choose a number £z. We want z to be the certainty equivalent. To obtain this in an incentive-compatible way[3], we could tell the subjects that a random number Z will be generated from a uniform distribution over the interval (y,x) and that they will be paid Z if $Z>z$ and will get to play out the lottery if $Z\leq z$. The optimal choice of z is the subject's certainty equivalent for the lottery. Consider the implications in terms of what they are choosing: their choice of z implies the choice of a lottery, which is a *compound* of the original lottery and the uniform distribution.

[3] This works, as before, with Expected Utility preferences and may work with other preferences.

To illustrate this, consider the lottery in Figure 2.4a, where the payoffs are £5 and £15. If they state *z=5* they get to play out the lottery; if they state *z=15*, they are opting for the lottery in Figure 2.4b – that is a uniform distribution over (5,15); if they state some number in between, they are opting for the lottery in Figure 2.4c. As *z* is varied from 5 to 15, the lottery in Figure 2.4c varies from that in Figure 2.4a to that in Figure 2.4b. We simply asked them to choose their preferred lottery; they did this by moving the slider below the graph and then clicking on 'Confirm'. The implied value of *z* given by a choice of the lottery in Figure 2.4c is 11; this is the observed certainty equivalent.

Figure 2.4a: Example for LC-1

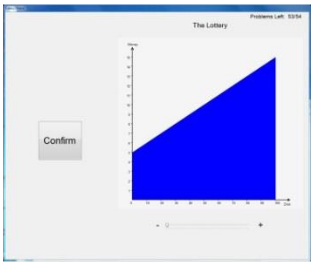


FIGURE 2.4b: Example for LC-2

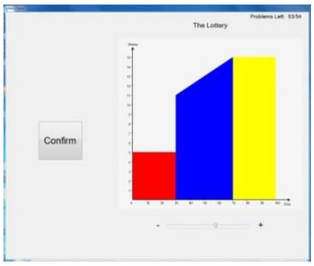


FIGURE 2.4c: Example for LC-3

We feel that this is a clearer way of implementing the Becker-DeGroot-Marschak mechanism. We denote it by LC – Lottery Choice – as they are choosing their preferred lottery. The 54 LC problems spanned lotteries with outcomes of £0, £5, £10 and £15 with probabilities ranging from 0.0 to 1.0 in steps of 0.1; details are attached in the appendices.

2.2.5 Allocation (AL)

The fourth part asked subjects to respond to 81 *allocation* problems. Loomes and Pogrebna (2014) had just 13 problems, but theirs were over three states and the exchange rates were always 1 to 1. We adopted a two-way allocation with non-unitary exchange rates, partly because it is easier for subjects to understand, but more crucially because the econometric analysis of the data is simpler. The number of problems was again chosen after pre-experimental simulations.

The subject is endowed with 100 tokens at the beginning of each problem. She or he has to split the endowment to two risky states, red and yellow, with given chances. For each state, there is an exchange rate between tokens and money. The exchange rate for red varies across problems. But the exchange rate for yellow is always equal to one. The chances assigned to red and yellow also vary across problems, but they always sum to 100. That is, if a problem is played out, the real state will be either red or yellow.

An example is shown in Figure 2.5.

Figure 2.5: Example of AL

In this example, the two states (red and yellow) have probabilities 0.7 and 0.3 respectively. Subjects have 100 tokens to allocate, and the exchange rates between tokens and money are 1 token = 17.5p for red, and 1 token = 10p for yellow. They made their allocation with the slider, with the figure showing the implied amounts of money (and their probabilities). There are three buttons below the bar to free the subject from moving the cursor. By clicking the buttons, the subject can allocate all the endowments to red or yellow, or split them equally. “Equal” here means the endowment is divided 50-50 between red and yellow. It does not mean that the monetary values of red and yellow are equal. The 81 allocation problems spanned probabilities ranging from 0 to 1 in steps of 0.1 with varying exchange rates; details are attached in the appendices. We denote this method AL.

2.2.6 Experimental Implementation

We implemented this experiment in four sessions during October and November 2014. The four sessions were all the same. There were 24 subjects in each session. Each subject was assigned a specific order for the four elicitation methods, which was different from the order experienced by the other 23 subjects.

Before the experiment, the subject was randomly allocated a specific integer between 1 and 24 by drawing a disk from an opaque bag. Then the subject was led to a desk with a number that matched the number on the disk she or he had drawn. On the desk, there was a computer and a piece of colour-printed instructions. There was also a pen and blank sheets of paper for the subject to write or draw drafts during the experiment.

Once all the subjects were seated, the experimenter briefly introduced the experiment. Then the subjects were given 15 minutes to read the instructions and to answer the control questions. During this time, the experimenter walked among the subjects and answered their questions privately. But the subjects were forbidden to talk to each other.

The experimenter checked the answers to the control questions for every subject to make sure she or he understood the experiment. Once all the subjects had correct answers to the control questions and they had no further questions, the experiment was started. There was a “Start” button showing on each subject’s computer screen after the experimenter had changed a value on the central computer. The subjects clicked the button and started the experiment.

During the experiment, the subjects could not communicate with anyone else. If they had any questions, they put up their hands or went to the control room. The experimenter would answer their questions privately.

Once a subject had finished the experiment, she or he had to notify the experimenter. Then the subject was taken to a separate room and was paid there. Since there were four elicitation methods, the subject had to randomly draw a disk from four in an opaque bag. The number on the disk denoted which elicitation method would be played out. For the numbers on the four disks, one to four denotes AL, LC, PC, and HL respectively.

There were four separate bags prepared for the four different methods. The number of disks in the bags matched the number of problems in each method. Once the played-out method had been determined, the subject randomly drew a disk from the corresponding bag to decide which problem was going to be played out.

This information and the subject's number was input into a "replay" project, and then the subject's decision on that specific problem was shown on the screen. Then the subject was asked to randomly draw a disk from an opaque bag. There was an integer from 1 to 100 on the disk denoting the real state which is played out for the problem. The subject's payoff depended on his or her decision and the real state played out. She or he also received a show-up fee, which was £2.50.

2.3 Functional forms assumed

While we are primarily interested in the differences between the different elicitation methods, in order to understand these differences we need to model behaviour and hence estimate the attitude to risk. To model the behaviour, we[4] need to choose preference functionals. We do not know

[4] By this we mean when using our methodology; that of Loomes and Pogrebna differs. We shall

the preference functionals of our subjects, so we have to choose a set of such functionals and use our data to find the best-fitting one(s). We chose the most popular in the literature, namely Expected Utility (EU) and Rank Dependent expected utility (RD).

Under the EU hypothesis, the utility of a lottery is simply the sum of the products of the probabilities and the utility of the possible outcomes respectively. A person simply perceives the probabilities as they are stated in the lottery. Or, in other words, a person assigns the same weight to all the probabilities. That weight is one.

Under the RD hypothesis, people are assumed to attach non-unitary weights to (de-)cumulative probabilities. A low probability is likely to be over-weighted, while a high probability is likely to be under-weighted. The weight assigned to a specific probability also depends on the amount of the related outcome and all the other possible outcomes in the lottery. RD can be regarded as a generalized form of EU. EU is the special case of RD in which all the weights are equal to one. Below is a mathematical illustration of EU and RD.

Let us denote by V the value to a subject of a 3-outcome lottery which pays x_i with probability p_i (for $i=1,2,3$), and let us order the payoffs so that $x_1 \geq x_2 \geq x_3$, then we have

discuss these methodological differences later.

for EU: $u(V) = p_1u(x_1) + p_2u(x_2) + p_3u(x_3)$

for RD: $u(V) = w(p_1)u(x_1) + [w(p_1+p_2)-w(p_1)]u(x_2) + [1-w(p_1+p_2)]p_3u(x_3)$

In these expressions, *u(.)* is the underlying utility function of the subject and *w(.)* is the rank-dependent weighting function.

We need to specify the utility function *u(.)* which appears in both these functionals. We adopt both the constant Relative Risk aversion (RR) form and the constant Absolute Risk aversion (AR) form. These are given by:

for RR: $u(x) = x^{1-r}/(1-r), \quad r \neq 1; \ln(x), r=1$

for AR: $u(x) = -\exp(-rx), r \neq 0; x, r=0.$

Here risk aversion measures the degree of reluctance of a person to accept a risky lottery rather than a certainty. If a person is more risk-averse, she or he is more likely to accept a certainty rather than a risky lottery, even when the expected value of the lottery is greater than the certainty. Or, in other words, to a risk-averse person, a risky lottery brings utility less than a certainty which is equal to the expected payoff.

We note that in both cases $r=0$ corresponds to risk-neutrality and increases in r imply greater risk aversion, but there is no mapping between the r for RR and that for AR. This is because the r in RR is a measure of *relative* risk aversion, while the r in AR is a measure of *absolute* risk aversion.

In fitting the RD specifications, we also need to specify a *weighting function* for the probabilities. This we take to be of the Quiggin form

$w(p) = p^g/[p^g+(1-p)^g]^{1/g}$

In the results that follow, for all four elicitation methods, we fit the four possible combinations of the two preference functionals and the two utility functions, using the obvious notation RREU**,** RRRD, AREU and ARRD. Essentially we want to see which of these best explains the data and we also want to see whether the estimated parameters differ across the elicitation methods; we do this on a subject-by-subject basis, as it is clear that subjects are different.

2.4 Our stochastic assumptions and econometric methodology

We should comment on our econometric methodology, and contrast it to that used by others. We treat subjects as different, so we analyse subject-by-subject[5]. We also use simultaneously *all* the responses of the subjects on *all* problems of a particular elicitation method (and use them for estimation), rather than compare responses on particular problems. The latter is what Loomes and Pogrebna (2014) and many others have done. So, for example, in their Table 1 on their page 578, they look at the distribution of responses[6] for particular decision tasks and compare these distributions across tasks. They note that the distributions are different across tasks, sometimes significantly so. This could be the case because of noise in subjects' responses but they present no way of modelling this

[5] We could fit a mixture model to the data, thus estimating *distributions* of the relevant parameters over all subjects.

[6] Which imply particular levels of risk-aversion.

noise, though the use of a statistical test (in this case a Mann-Whitney test) does necessarily involve some implicit assumption about stochastics.

Why we prefer our approach is because of inference and statistical considerations. We can illustrate this very simply: suppose we wish to test an hypothesis that $\mu_1 = \mu_2 = \mu_3$ where these are means of some variable(s) across some population(s) and we base our test on sample means m_1, m_2 and m_3. Testing separately whether m_1 is significantly different from m_2, that m_2 is significantly different from m_3 and that m_3 is significantly different from m_1 is **not** the same as testing whether m_1, m_2 and m_3 are significantly different from each other. For given levels of significance, the power is different. Another difference between our methodology as compared to others' is in the number of problems we present to our subjects: pre-experimental simulations show that one needs large numbers of observations to get precise estimates; there is a lot of noise in subjects' behaviour.

Our econometric methodology is to fit, for each of RREU, RRRD, AREU and ARRD, the models to the decisions of the subjects, for each of the four elicitation methods, and hence obtain estimates of the risk aversion index (and also the other parameters). We do this by maximum likelihood, using MATLAB. To do this we need to make assumptions about the stochastic nature of the data. This arises from errors made by the subjects. We largely follow convention.

- HL: we assume that the subject calculates the utility difference between the two lotteries for each pair in the list, but makes an error in the calculation. Further, embodying the fact that the list is presented as a list, we assume that the subject *makes the same error* for each pair; we further assume that this error has a normal distribution with mean 0 and precision (the inverse of the standard deviation) *s.* Then the switch-point decision is taken on the basis of where this utility difference plus error changes from positive to negative or *vice versa*.
- PC: we assume that on each problem the subject calculates the utility difference between the two lotteries, but makes an error in this calculation; we further assume that this error is independent across problems and has a normal distribution with mean 0 and precision (the inverse of the standard deviation) *s.* So that the decision is taken on the basis of the sign of the correct utility difference plus error.
- LC: we assume that the subject calculates the certainty equivalent of the lottery, but makes an error in this calculation; we further assume that this error has a normal distribution with mean 0 and precision (the inverse of the standard deviation) *s.* So that the observed certainty equivalent is the correct subject's true equivalent plus error.
- AL: we assume that the subject calculates the optimal allocation that he or should make, but makes an error in this calculation; we

further assume that this error has a normal distribution with mean 0 and precision (the inverse of the standard deviation) *s.* So that the observed allocation is the optimal allocation plus error.

We note that the *s* in the PL and PC stories are on a different scale than the *s* in the PC and AL stories – the former being on utilities and the latter on money.

2.5 Results

We had 96 subjects, who each completed all four parts of the experiment. For each subject and for each elicitation method, we attempted to fit the four models RREU, RRRD, AREU and ARRD to their decisions; so for each subject 16 models were estimated more details for estimation. This implies a total of 1,536 estimations. In certain cases the estimation did not converge. This was for a variety of reasons which we discuss below. In the table below we enumerate these cases by elicitation method. It will be seen from this table that the 'worst offender' is PC. There were a total of 20 subjects where convergence was not obtained on at least one method. As the point of the chapter is to compare different elicitation methods, we exclude all these 20 subjects from the analysis that follows (though an online appendix repeats parts of the analyses with all 96 subjects).

Method(s)	Number of times not converged
Just LC	3
Just PC	9
Just HL	5
Both AL and LC	1
Both AL and PC	1
Both LC and PC	1

These cases of non-convergence took several forms: (1) where the subject was clearly either risk-neutral or risk-loving – in which cases the implied parameters are not unique; (2) where the estimation hit the bounds imposed by us on the parameters; (3) where the subject appeared not to understand the tasks, or where the subject appeared to be responding randomly.

Regarding the bounds, the problematic parameter was often the g in the probability weighting function for RD. We imposed a lower limit of 0.3. Below that the weighting function is not monotonically increasing, and an upper limit which varied from subject to subject.

We present our results in two main parts. First we present some summary statistics; these are in Table 2.1. Then we compare the elicited/estimated parameters across preference functionals for given elicitation methods; finally we compare the elicited/estimated parameters across elicitation methods for given preference functionals.

Table 2.1 presents some summary statistics. It is very clear from this that the estimated parameters vary widely across the different elicitation methods. For example, using AL the risk-aversion index elicited in the RR specifications is, on average, much higher that found with the other methods, and also has a much higher spread. It may well be that the elicitation method is affecting the way that subjects process the problems. For example the allocation method may be focussing subjects' minds on what outcome they might obtain for different states of the world. And here we have to emphasize that, the r value for the RR specifications cannot be compared with the r value for the AR specifications.

Stats	Method	RREU		RRRD			AREU		ARRD		
		r	*s*	*r*	*g*	*s*	*r*	*s*	*r*	*g*	*s*
Mean	PC	0.502	1.666	0.375	1.105	2.135	0.175	0.127	0.134	1.112	0.164
	AL	3.144	0.161	2.059	1.120	0.168	0.078	0.148	0.052	1.021	0.151
	LC	0.192	0.594	0.028	0.959	0.635	0.094	0.564	0.043	1.027	0.598
	HL	0.182	0.955	-0.022	0.824	0.963	0.068	0.110	0.026	0.741	0.136
Median	PC	0.535	1.358	0.399	0.850	1.515	0.157	0.111	0.109	0.870	0.137
	AL	1.054	0.078	0.947	1.005	0.087	0.027	0.068	0.024	0.924	0.070
	LC	0.329	0.539	0.237	0.907	0.580	0.073	0.530	0.041	0.904	0.549
	HL	0.174	0.777	0.004	0.648	0.748	0.044	0.097	0.018	0.646	0.121
Standard Deviation	PC	0.275	1.440	0.320	0.714	2.998	0.131	0.077	0.154	0.701	0.131
	AL	10.367	0.646	8.450	0.428	0.655	0.231	0.661	0.186	0.403	0.667
	LC	0.956	0.193	1.092	0.401	0.192	0.189	0.164	0.183	0.535	0.166
	HL	0.285	1.136	0.308	0.622	1.240	0.089	0.144	0.087	0.314	0.149

Table 2.1: Summary statistics

The parameters of the various specifications are the risk-aversion index *r* (for both the EU and the RD functionals), the weighting function parameter *g* (for the RD functional) and the precision parameter *s* for all functionals. Some of these parameters are comparable, and in Table 2.2.1 we show the relationships between them for those that are comparable. We can compare *g* across RRRD and ARRD, and similarly we can compare *s* across the various preference functionals. Also we can compare the *r* between RREU and RRRD, and between AREU and ARRD, though clearly if the true preference functional is Rank Dependent then assuming Expected Utility preferences may lead to bias. Table 2.2.1 reports the correlation (ρ) between the estimated parameters for different elicitation methods and the intercept (α) and slope (β) of a linear regression of one against the other; if they were consistently producing the same estimates then α should be zero and β should be one. The table shows that the estimated values of *s*, across preference functionals, are generally not too far apart. For example, the estimated values of *s* (the precision parameter) using the AL method are very close whether we fit RREU or RRRD. This is less true for the estimated values of *g* (the weighting function parameter), though they are very similar using the LC method whether we fit RRRD or ARRD. However this is not always the case: for example, there are big differences between the estimated values of *g* using the PC method depending on whether we fit RRRD or ARRD. The estimated *r* values differ more markedly across the elicitation methods, though once again there are cases (using LC and comparing RREU and RRRD) where the fit is good. Even though it is difficult to summarise a whole table in one sentence, one could say that the correlations are all positive and reasonably large, and certainly larger than in Table 2.3 (comparisons across elicitation methods), which we shall come to shortly.

Parameter	Method	x	y	α	β	ρ
r	PC	RREU	RRRD	-0.121***	0.988	0.849
r	PC	AREU	ARRD	-0.048***	1.035	0.875
s	PC	RREU	RRRD	0.538***	0.789***	0.795
s	PC	AREU	ARRD	0.046***	0.844***	0.879
g	PC	RRRD	ARRD	0.624***	0.442***	0.451
r	AL	RREU	RRRD	0.185**	0.639***	0.801
r	AL	AREU	ARRD	0.009***	0.528***	0.700
s	AL	RREU	RRRD	0.001	1.058***	0.987
s	AL	AREU	ARRD	-0.001	1.037***	0.997
g	AL	RRRD	ARRD	0.373***	0.579***	0.617
r	LC	RREU	RRRD	-0.179***	1.082	0.923
r	LC	AREU	ARRD	-0.040***	0.884**	0.913
s	LC	RREU	RRRD	0.078***	0.938	0.941
s	LC	AREU	ARRD	0.055**	0.963	0.950
g	LC	RRRD	ARRD	0.095	0.975	0.731
r	HL	RREU	RRRD	-0.174***	0.835**	0.773
r	HL	AREU	ARRD	-0.030***	0.829***	0.848
s	HL	RREU	RRRD	-0.111*	1.139**	0.890
s	HL	AREU	ARRD	0.008	1.210**	0.847
g	HL	RRRD	ARRD	0.421***	0.388***	0.769
The hypotheses being tested are α=0 and β=1. * denotes significantly different at 10%; ** at 5% and *** at 1%.						

Table 2.2.1: A comparison of the estimated coefficients across Preference Functionals (part 1)

Some of the parameters are not comparable. Crucially the r parameter differs in both what it measures and its scale between the Constant Absolute Risk Aversion specification and the Constant Relative Risk Aversion specification; moreover there is no precise mapping between them. However increases in either imply a higher risk-aversion so that they should be positively related. Table 2.2.2 shows the results. Again the correlations are reasonably high.

We can also show the results graphically. We show here just a subset – the full set can be found in the appendices. Figure 2.6 shows the scatter of the estimated *r* values using the AL method across preference functionals. This figure suggests that getting the functional form wrong does not upset our estimation of the risk-aversion of the subjects. (Deck *et al* (2008) also present such scatters and make the same point, though their risk-aversion indices are not estimated.) However, Figure 2.7 suggests that if we get the utility function wrong then the estimate of the probability weighting parameter *g* may be quite seriously wrong.

Figure 2.8 shows the scatter of the estimated *s* values using the AL method across the preference functionals. The relationships are almost always the 45 degree line (as Table 2.2.1 shows). This means that if whether we assume RR or AR preferences we get almost the same estimate of the noise in subjects' responses.

Parameter	Method	x	y	α	β	ρ
r	PC	RREU	AREU	-0.036**	0.422***	0.889
r	PC	RREU	ARRD	-0.073***	0.410***	0.731
r	PC	RRRD	AREU	0.056***	0.318***	0.779
r	PC	RRRD	ARRD	-0.009	0.380***	0.788
s	PC	RREU	AREU	0.044***	0.051***	0.816
s	PC	RREU	ARRD	0.084***	0.043***	0.637
s	PC	RRRD	AREU	0.053***	0.040***	0.622
s	PC	RRRD	ARRD	0.074***	0.043***	0.678
r	AL	RREU	AREU	0.007***	0.020***	0.979
r	AL	RREU	ARRD	0.012***	0.012***	0.746
r	AL	RRRD	AREU	0.014***	0.019***	0.724
r	AL	RRRD	ARRD	0.008***	0.019***	0.968
s	AL	RREU	AREU	0.002	0.809***	0.901
s	AL	RREU	ARRD	0.002	0.834 ***	0.893
s	AL	RRRD	AREU	0.004	0.732 ***	0.873
s	AL	RRRD	ARRD	0.004	0.759 ***	0.872
r	LC	RREU	AREU	0.026**	0.260***	0.880
r	LC	RREU	ARRD	-0.022*	0.247***	0.857
r	LC	RRRD	AREU	0.090***	0.137***	0.790
r	LC	RRRD	ARRD	0.039***	0.143***	0.851
s	LC	RREU	AREU	0.208***	0.600***	0.705
s	LC	RREU	ARRD	0.286***	0.525***	0.609
s	LC	RRRD	AREU	0.202***	0.571***	0.668
s	LC	RRRD	ARRD	0.229***	0.581***	0.671
r	HL	RREU	AREU	0.019***	0.267***	0.857
r	HL	RREU	ARRD	-0.016**	0.231***	0.757
r	HL	RRRD	AREU	0.072***	0.204***	0.706
r	HL	RRRD	ARRD	0.031***	0.228***	0.806
s	HL	RREU	AREU	0.033***	0.070***	0.723
s	HL	RREU	ARRD	0.048***	0.085***	0.618
s	HL	RRRD	AREU	0.050***	0.048***	0.638
s	HL	RRRD	ARRD	0.060***	0.070***	0.648

The hypotheses being tested are α=0 and β=0.
* denotes significantly different at 10%; ** at 5% and *** at 1%.

Table 2.2.2: A comparison of the estimated coefficients across Preference Functionals (part 2)

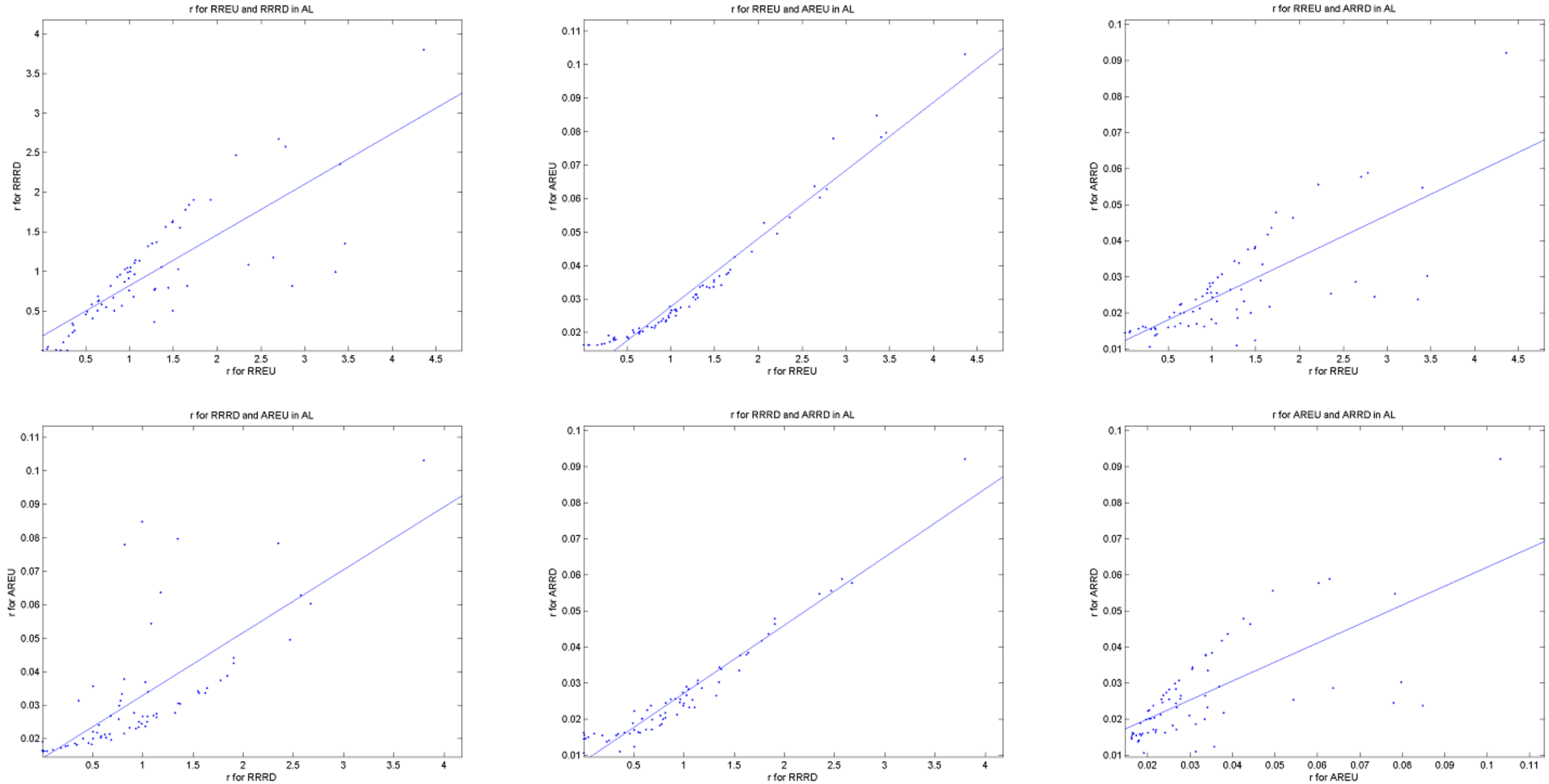


Figure 2.6: estimates of *r* using AL across preference functionals

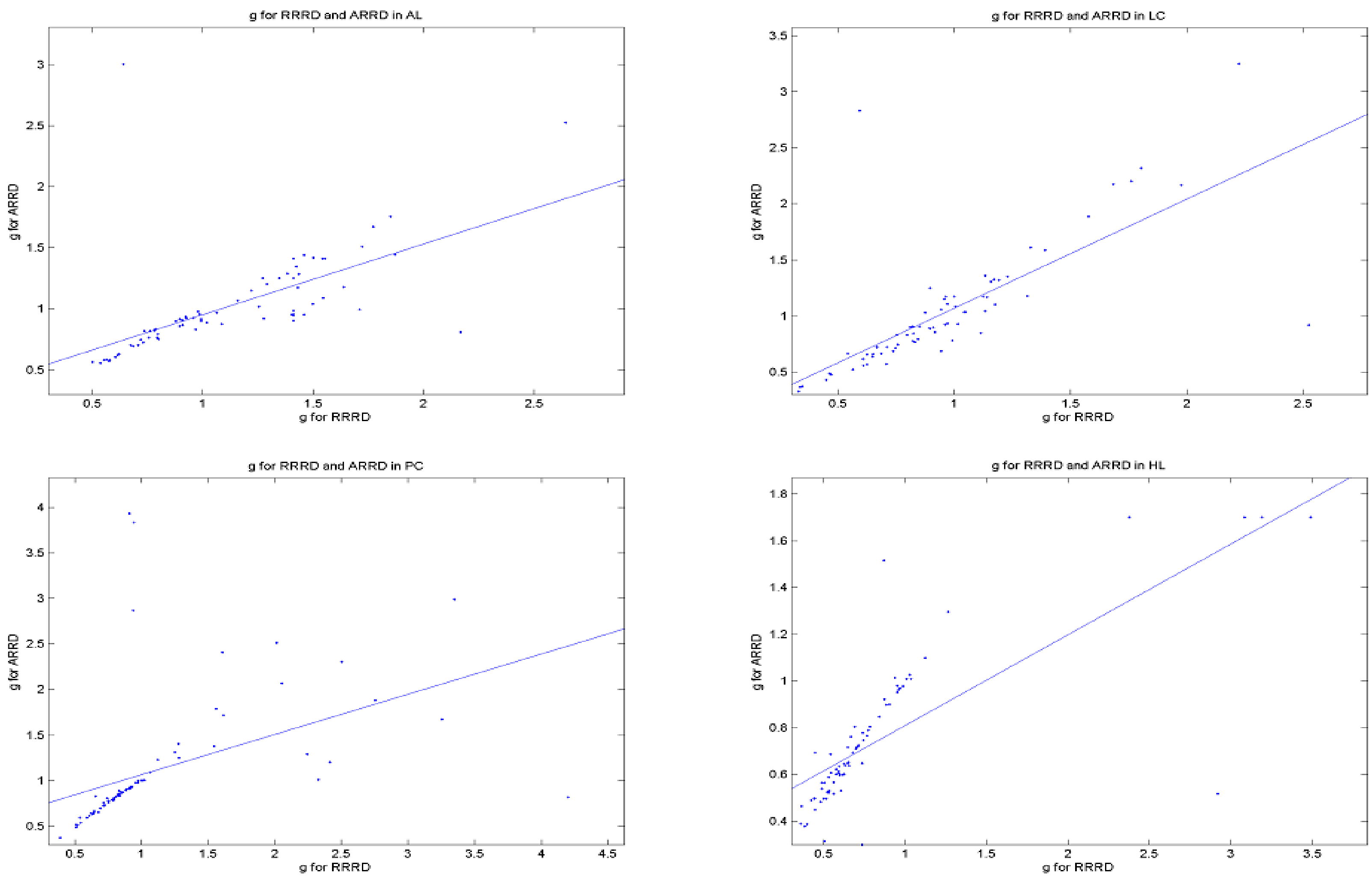


Figure 2.7: estimates of *g* across preference functional

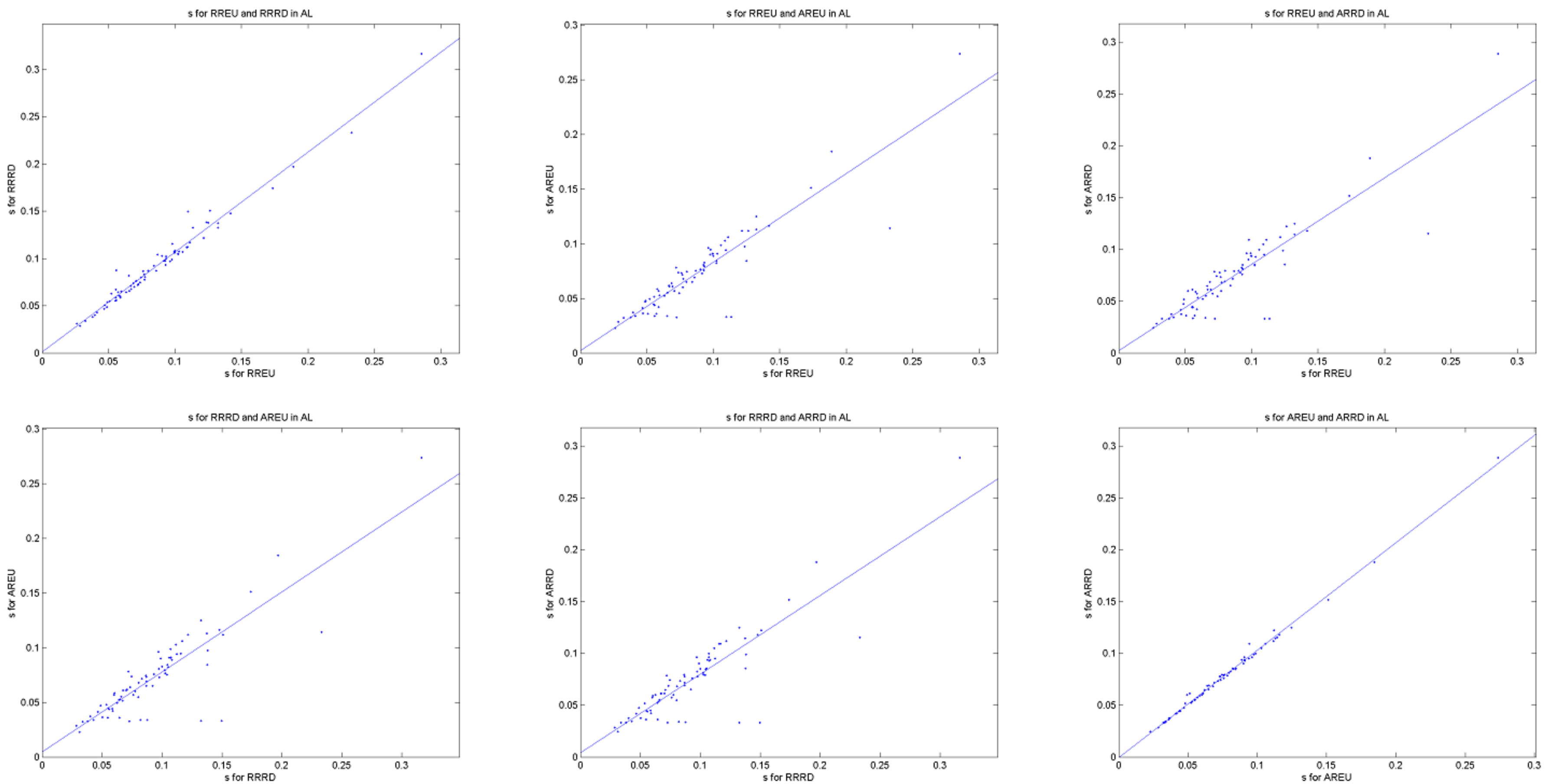


Figure 2.8: estimates of *s* using AL across preference functionals

We continue to analyse the different elicitation methods across preference functionals, and now consider those where the estimated parameters are **not** comparable. Table 2.2.2 gives details. Here we include the estimated intercept and slope of the regression of an estimated parameter using a particular elicitation method for each preference functional against each of the others. Because of the lack of a relationship between the two parameters, other than monotonic increasingness, the obvious test to carry out is that the slope is positive; significance is good and this is reported in the *β* column. We also include a test of whether the intercept is significantly different from 0 – as it should be. Here significance is weak.

Finally, and crucially, we compare the estimated parameters across elicitation methods. Table 2.3 gives a summary, while Figures 2.9 to 2.12 present a subset of graphical comparisons; the full set of 10 sets of comparisons can be found in the appendices. Let us start with Figure 2.9 which shows the 6 scatters for the estimated *r* value for the RRRD functional; each scatter being the estimated values using one elicitation method plotted against the estimated values for another elicitation method, for all the non-excluded subjects. For example, the scatter at the top left compares AL with LC. In a perfect world these points would lie on the 45^{o} line. As the scales on the two axes differ, it helps to fit a regression line to the scatter. Table 2.3 gives the intercept (-0.013) and slope (0.224) of this line. While the intercept is not significantly different from zero, the slope is significantly different from one, as the asterisks indicate. As can be seen, the risk-aversion index elicited by AL is generally greater than that elicited by LC. This could result from what might be called a built-in bias with the allocation method – subjects tend to make allocations to avoid large differences in their payoff depending on which state occurs, while the 'BDM' mechanism does not make so explicit the possible consequences of their actions. Indeed generally the risk-aversion elicited under allocation is generally higher than for the other three methods.

Parameter	PF	*x*	*y*	*α*	*β*	*ρ*
r	RREU	AL	LC	-0.013	0.224***	0.417
r	RREU	AL	PC	0.484***	0.022***	0.073
r	RREU	AL	HL	0.003	0.132***	0.417
r	RREU	LC	PC	0.471***	0.098***	0.197
r	RREU	LC	HL	0.113***	0.253***	0.507
r	RREU	PC	HL	0.163**	0.038***	0.037
s	RREU	AL	LC	0.566***	0.240	0.056
s	RREU	AL	PC	1.589***	-2.760*	-0.157
s	RREU	AL	HL	0.554***	2.583	0.281
s	RREU	LC	PC	1.674***	-0.549***	-0.126
s	RREU	LC	HL	0.473***	0.517**	0.252
s	RREU	PC	HL	0.900***	-0.074***	-0.147
r	RRRD	AL	LC	-0.230**	0.340***	0.426
r	RRRD	AL	PC	0.334***	0.046***	0.101
r	RRRD	AL	HL	-0.194***	0.161***	0.377
r	RRRD	LC	PC	0.366***	0.060***	0.122
r	RRRD	LC	HL	-0.045	0.193***	0.412
r	RRRD	PC	HL	-0.071	0.129***	0.134
s	RRRD	AL	LC	0.620***	0.089*	0.022
s	RRRD	AL	PC	2.067***	-3.640*	-0.157
s	RRRD	AL	HL	0.445***	3.518**	0.322
s	RRRD	LC	PC	1.036**	1.080	0.196
s	RRRD	LC	HL	0.424**	0.557	0.212
s	RRRD	PC	HL	0.822***	-0.014***	-0.030
g	RRRD	AL	LC	0.916***	0.045***	0.048
g	RRRD	AL	PC	0.706***	0.365***	0.220
g	RRRD	AL	HL	0.602***	0.200***	0.138
g	RRRD	LC	PC	1.075***	0.030***	0.017

g	RRRD	LC	HL	0.764***	0.062***	0.040
g	RRRD	PC	HL	0.767***	0.052***	0.059
r	AREU	AL	LC	-0.040	3.693***	0.391
r	AREU	AL	PC	0.171***	0.276	0.039
r	AREU	AL	HL	-0.007	2.232**	0.468
r	AREU	LC	PC	0.159***	0.174***	0.252
r	AREU	LC	HL	0.042***	0.279***	0.592
r	AREU	PC	HL	0.064***	0.020***	0.029
s	AREU	AL	LC	0.493***	0.982	0.228
s	AREU	AL	PC	0.124***	0.032***	0.016
s	AREU	AL	HL	0.097***	-0.124***	-0.124
s	AREU	LC	PC	0.161***	-0.061***	-0.130
s	AREU	LC	HL	0.027*	0.107***	0.460
s	AREU	PC	HL	0.104***	-0.122***	-0.249
r	ARRD	AL	LC	-0.098***	4.694***	0.415
r	ARRD	AL	PC	0.121***	0.656	0.060
r	ARRD	AL	HL	-0.038*	2.369**	0.377
r	ARRD	LC	PC	0.129***	0.108***	0.128
r	ARRD	LC	HL	0.017*	0.200***	0.420
r	ARRD	PC	HL	0.026*	-0.001***	-0.002
s	ARRD	AL	LC	0.531***	0.911	0.217
s	ARRD	AL	PC	0.149***	0.020***	0.011
s	ARRD	AL	HL	0.114***	-0.001***	-0.001
s	ARRD	LC	PC	0.173***	-0.037***	-0.085
s	ARRD	LC	HL	0.034	0.134***	0.409
s	ARRD	PC	HL	0.137***	-0.153***	-0.201
g	ARRD	AL	LC	0.928***	0.104***	0.078
g	ARRD	AL	PC	1.078***	0.042***	0.024
g	ARRD	AL	HL	0.665***	0.065***	0.084
g	ARRD	LC	PC	1.132***	-0.019***	-0.015
g	ARRD	LC	HL	0.612***	0.115***	0.196
g	ARRD	PC	HL	0.720***	0.010***	0.021

The hypotheses being tested are α=0 and β=1.
* denotes significantly different at 10%; ** at 5% and *** at 1%.

Table 2.3: A comparison of the estimated coefficients across Elicitation Method

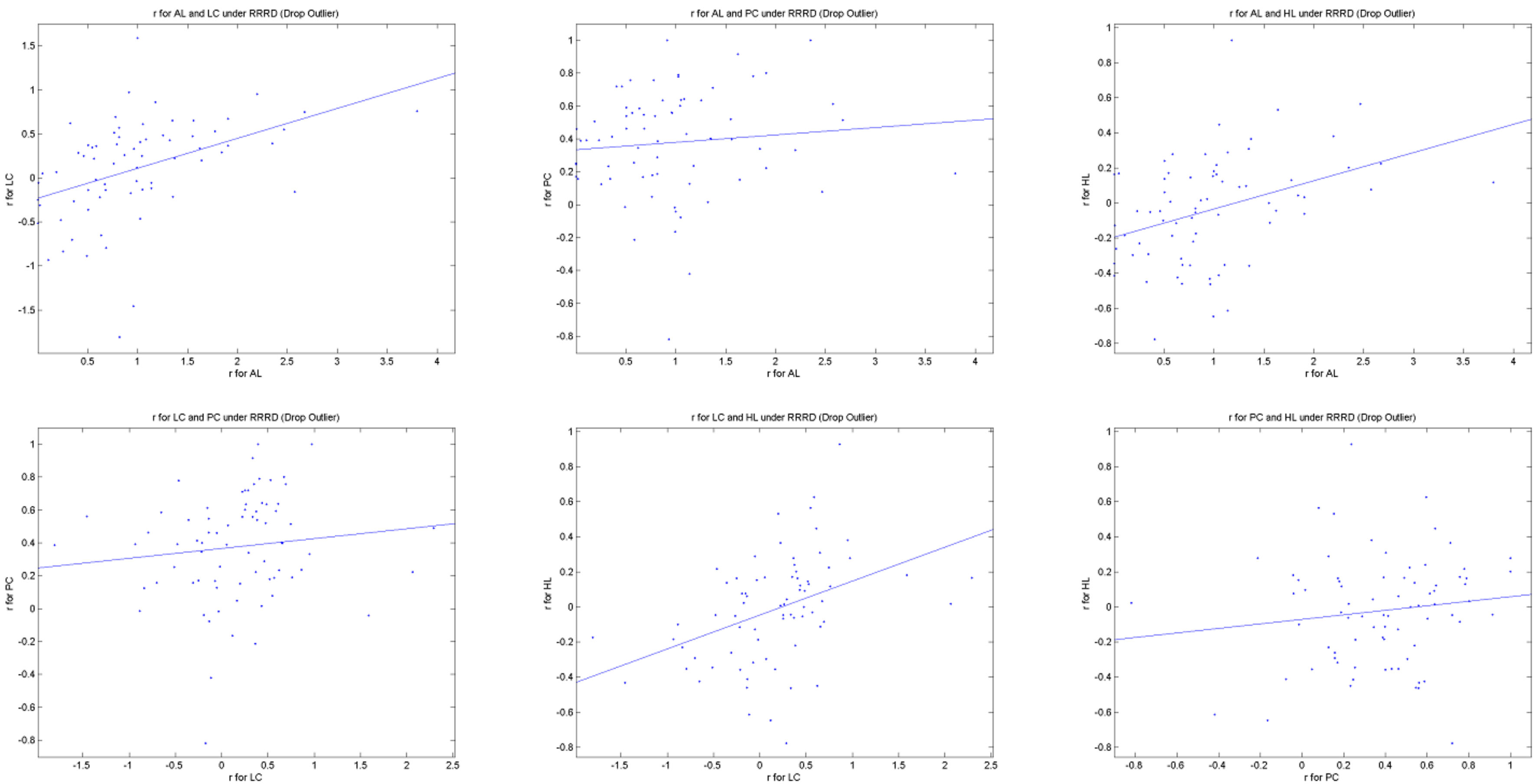


Figure 2.9: estimates of *r* in RRRD across elicitation methods

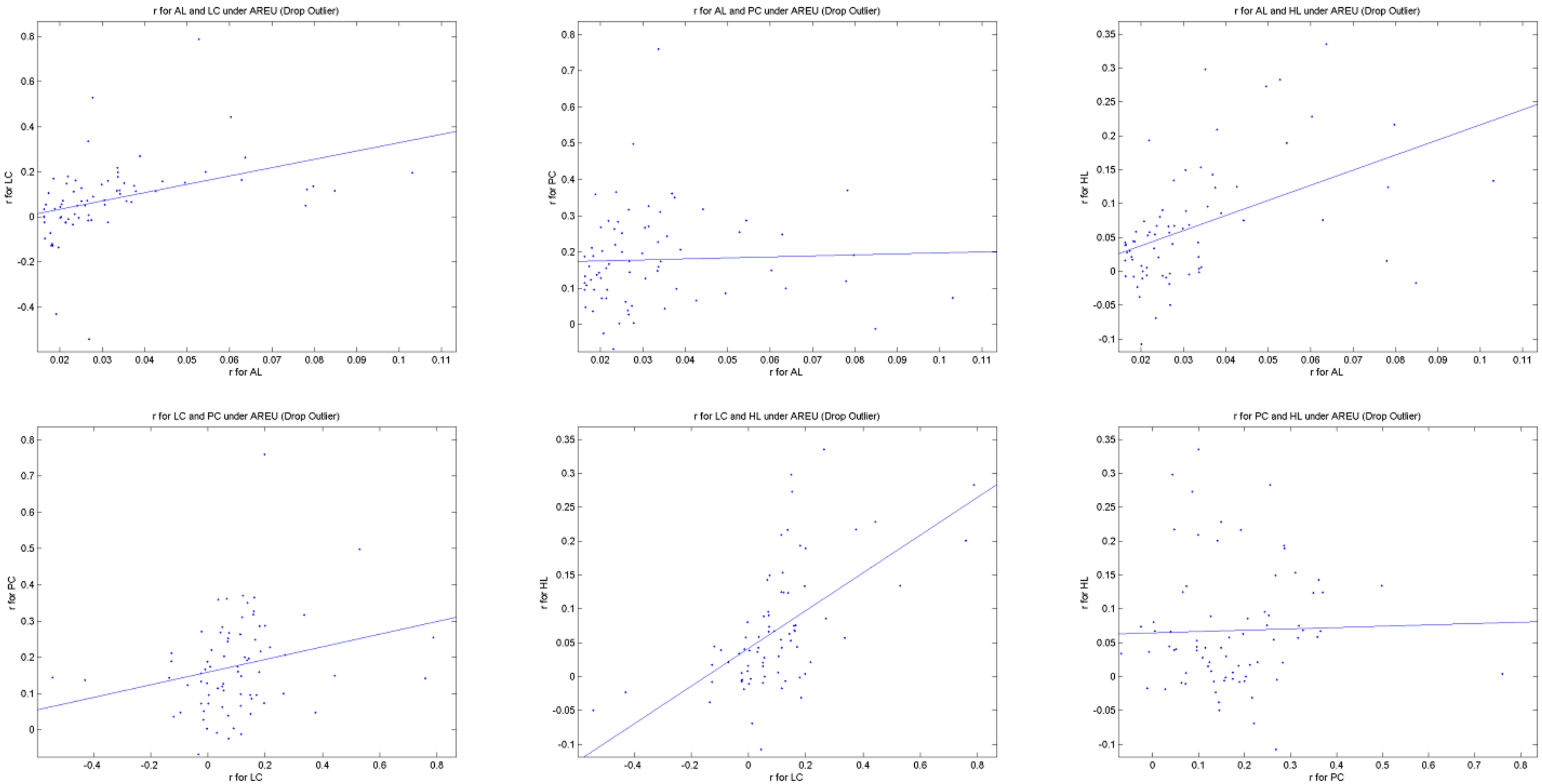


Figure 2.10: estimates of *r* in AREU across elicitation methods

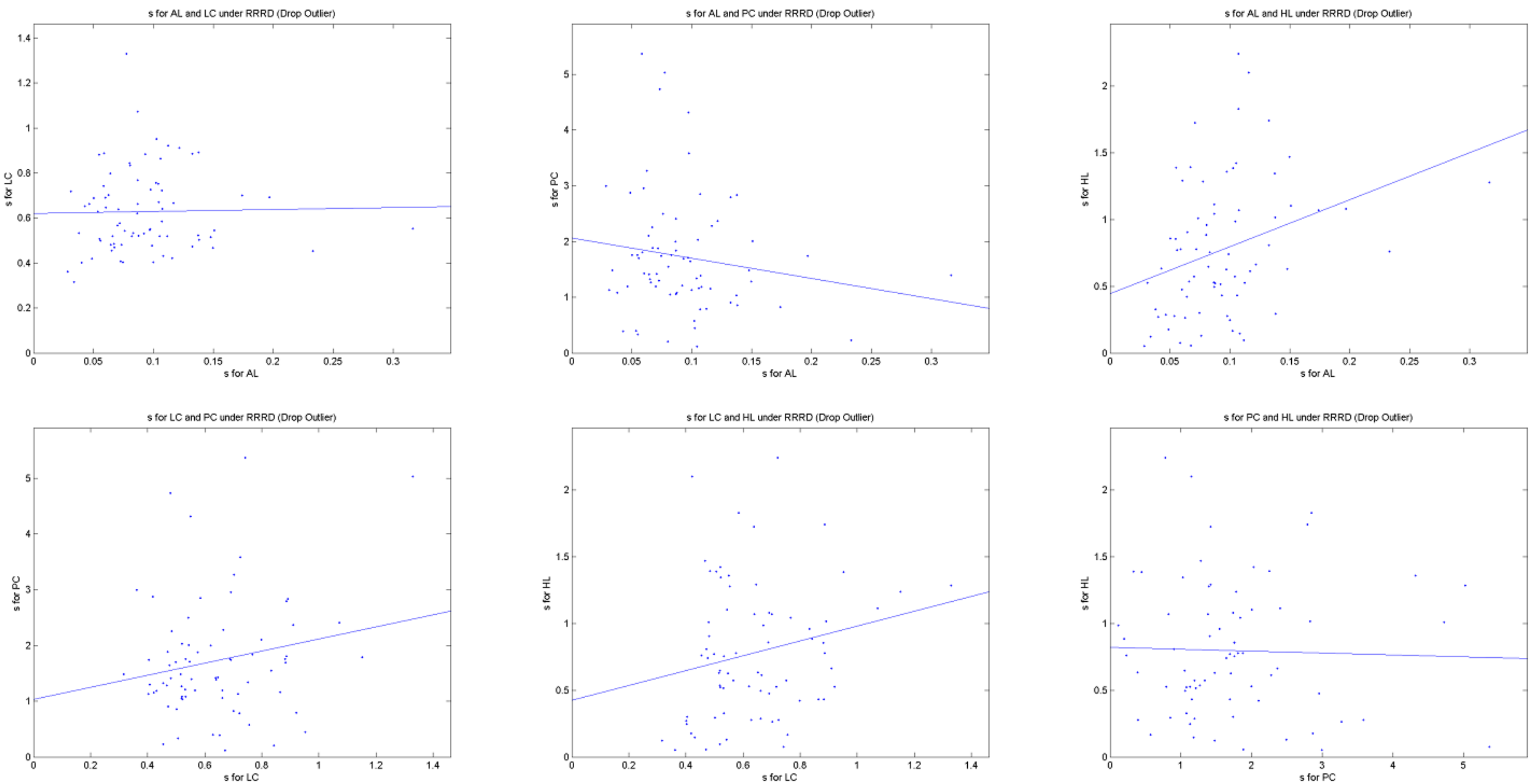


Figure 2.11: estimates of *s* in RRRD across elicitation method

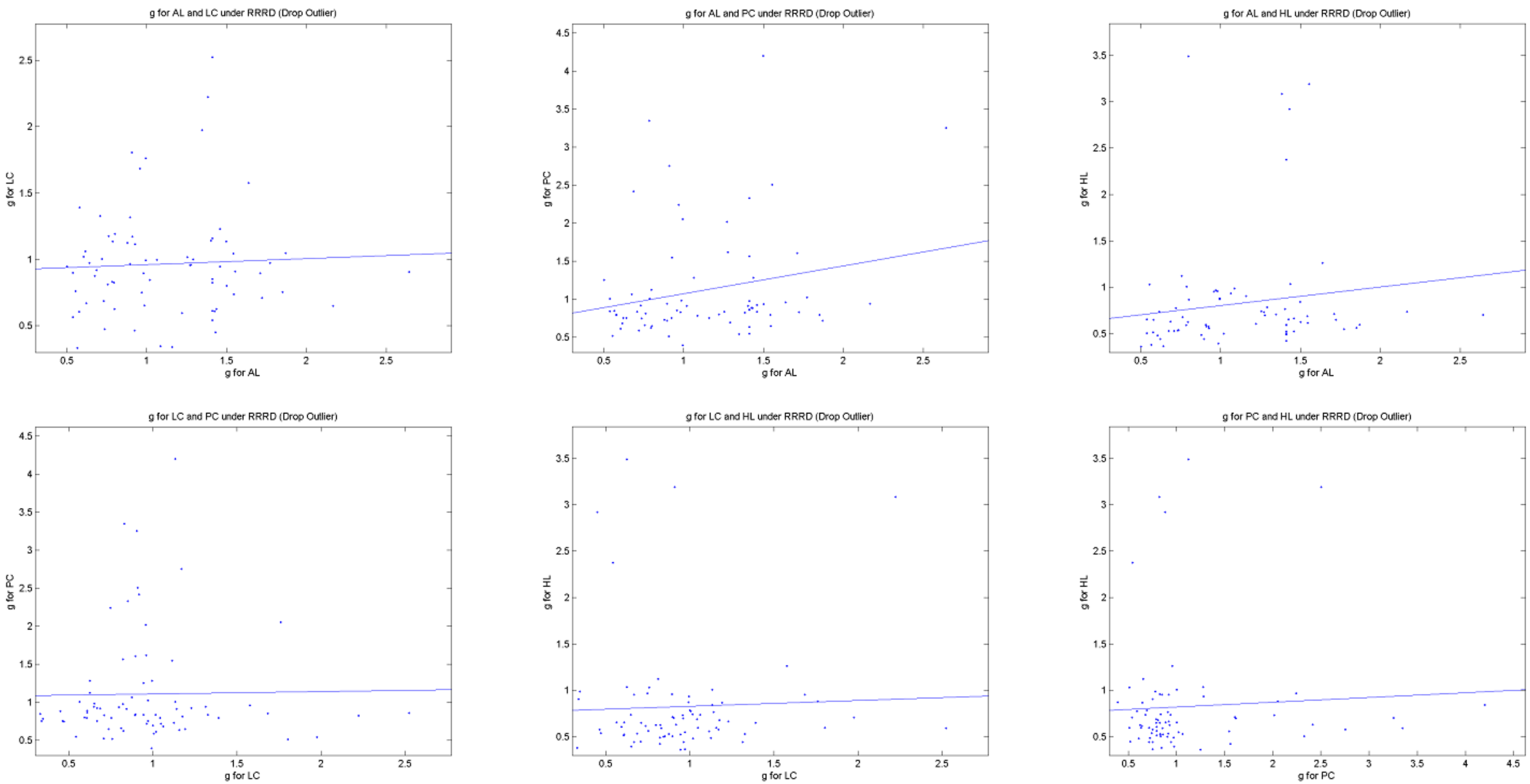


Figure 2.12: estimates of *g* in RRRD across elicitation methods

Relative to the other comparisons, the one discussed above is one of the best. Examine Figure 2.10 which shows the 6 scatters for the estimated *r* value for the AREU functional. Here, as Table 2.3 shows, most of the intercept values are significantly different from zero and most of the slope values from 1, so the different elicitation methods are getting usually significantly different estimates. Once again the allocation method seems to be inducing more risk-averse behaviour. Figure 2.11, comparing different estimates of the precision parameter *s* for RRRD across the different methods, shows that sometimes the relationship is negative. Here the precision seems to be lower with the allocation method, and possibly highest on the Pairwise Choice method, though a direct comparison does not make much sense as the error on PC is on the utility difference (between the two lotteries) while the error on AL is on the difference in the amounts of implied money.

Figure 2.12 is arguably the worst, usually showing very little relationship (and sometimes a negative one) between the elicited *g* values. Here the *g* value for the LC method seems to have the largest variation in the estimated values and that for PC having the smallest.

Finally, even though the different elicitation methods seem to disagree on the estimates of the parameters, we should ask whether at least they agree on the best-fitting preference functional. Table 2.4 gives the results, with the criterion for the 'best-fitting' functional being either the raw log-likelihood, or either the Akaike or Bayes information criterion (both of which correct the log-likelihood for the number of parameters involved in the fitting). Table 2.4 shows that correcting for degrees of freedom does make a big difference. But here again, the different elicitation methods disagree: it is clear that AL puts the AR specifications first, while the other methods suggest that RR fits better. Moreover there is no general agreement as to which of EU and RD is the best.

Method	PF	LL	BIC	AIC
PC	RREU	1	25	33
	RRRD	38	20	12
	AREU	1	10	14
	ARRD	37	21	17
AL	RREU	1	0	1
	RRRD	74	2	2
	AREU	0	24	37
	ARRD	2	50	36
LC	RREU	2	14	18
	RRRD	49	14	12
	AREU	0	17	22
	ARRD	27	31	24
HL	RREU	1	40	40
	RRRD	49	6	6
	AREU	0	19	22
	ARRD	27	11	8

Table 2.4: Best-fitting preference functional

2.6 Conclusions

One clear conclusion that emerges from our results is that the elicitation method – the context – *does* matter to the estimated risk-aversion index; there are big differences in the estimated risk attitudes across the elicitation methods. The choice of the preference functional seems to be less important, though if the best functional is RD then assuming it to be EU can lead to mis-elicitation. The choice of the utility function seems to be even less important.

This seems to send a clear message: risk-aversion should be elicited in the context in which it is to be interpreted. This suggests that one should estimate the risk-aversion index along with the other parameters of the model being fitted to the data; eliciting them in another context could lead to mis-interpretations of the data. As Loomes and Pogrebna (2014) write "In the short run, one recommendation is that researchers who wish to take some account of and/or make some adjustment for risk attitude in their studies should take care to pick an elicitation procedure as similar as possible to the type of decision they are studying…". We would even go as far by suggesting modifying "as similar as possible" to "in the same decision problem". They follow up with a comment about the number of tasks that are posed to subjects; often, given the noise in subjects' responses, there are far too few. We would not go as one experiment which asked 2400 pairwise choice problems (over two sessions) but 1 or 2 is surely too few even if we want to elicit just 1 or 2 parameters.

In summary, our results suggest something particularly worrying: namely, that subjects do not have a stable preference functional for making decisions under risk. This conclusion would undermine much of economics.

To check whether that is true we could investigate more carefully the stochastic component of decision-making. Or we could take up Loomes and Pogrebna's call to understand better "how contextual or procedural factors interact with that process [of decision-making]." Context does seem to matter, though it is not clear whether this is because of cognitive factors or because of error.

Chapter 3. Do Past Decisions Influence Future Decisions?[1]

3.1 Introduction

In many, if not most, experiments on individual choice, the Random Lottery Incentive (RLI) mechanism is used. This involves each subject being asked a number of questions, one of which is chosen at random at the end of the experiment, with the payoff to the subject being determined by the subject's response on that question. A rather superficial reason for the use of this mechanism is that it can appear to save the experimenter money while making the payoffs appear salient to the subjects, but the real reason is that it avoids income effects and cross-task contamination.

However, as pointed out by Holt (1986) and others, this requires the *separation* by the subject of the various questions. If the subject has Expected Utility (EU) preferences, this is guaranteed through the Independence Axiom, but if the preferences are not EU, there is a potential inconsistency in the use of the RLI mechanism. This can be illustrated very simply, though rather artificially, with the following example. Suppose a subject is asked to choose between (L) £4 with certainty and (R) a 50-50 gamble between £10 and £0. Suppose that the subject is sufficiently risk-averse that L is preferred. However, now suppose that the subject is offered the same choice twice and is told that his or her answer on a

[1] This chapter is a joint work with John Hey and has been published in *Applied Economics Letters*, 21:3, 152-157, 2013.

randomly-chosen one of the two choices will be played out for real. Having chosen L on the first occasion, the subject may argue as follows: “if I choose L again on the second choice, I will get £4 for certain, but if I choose R, I will get £4 with probability ½, £10 with probability ¼ and £0 with probability ¼”. Obviously if the subject has EU preferences the subject will choose L again, but if the preferences are not EU the subject *could* prefer R. So the subject may choose L the first time, and R the second: hence, when making inferences from the data, the experimenter needs to take into account whether the subject separates or not.

While independence is sufficient for separation, it is not necessary: subjects could have non-EU preferences and still separate – making the RLI mechanism still valid. It is therefore an empirical issue. Not surprisingly, it has already been investigated. The main studies are Starmer and Sugden (1991), Cubitt *et al* (1998), Hey and Lee (2005a) and Hey and Lee (2005b).[2] The first two of these take a very simple experimental setting in which subjects were presented with at most two questions, and the focus was on whether there was cross-question contamination. The conclusion was that there was not, and hence that the RLI was valid. However, in many experiments subjects are presented with many more than two questions (on the grounds that, because of the presence of noise in subjects’ behaviour, many observations are needed to elicit preferences accurately). This was the setting for the two Hey and Lee papers, and the hypotheses

[2] Lee (2008) is also relevant.

being tested there were: (1) that the subjects, in answering each question, considered the experiment *as a whole*; (2) that in answering any question, subjects took into account their answers to *all* the preceding questions. Once again, the conclusions were that they did not, and hence that the RLI was valid.

However that is not the end of the story. Wakker[3] writes: "The random-lottery incentive system has since become the almost exclusively used incentive system for individual choice. Unfortunately, more than half of the referees of economic journals will embark on yet another discussion from scratch of this issue." He also points to the evidence discussed above.
The problem is that not all of the various possible forms of contamination have been investigated. In particular, in experiments with many questions, the two hypotheses considered by Hey and Lee are cognitively very demanding and hence possibly unrealistic. Something simpler might be cognitively more plausible. This is what we test here: we test the hypothesis that in answering any question the subject takes into account their decision on the immediately preceding question. We call this the Contamination Hypothesis, to distinguish it from the usual Separation Hypothesis. Under our hypothesis any decision is affected by all previous decisions, not directly as in Hey and Lee, but indirectly through the immediately preceding one.

[3] http://people.few.eur.nl/wakker/miscella/debates/randomlinc.htm

3.2 The hypothesis and the data

We suppose that every subject faces N decisions in one experiment ($n=1,2,\ldots,N$). The N decisions are presented sequentially. In each decision, the subject has to choose between two lotteries, Left (L) and Right (R). The n^{th} decision is denoted by d_n. Our Contamination Hypothesis clearly does not apply to the first decision since there is no previous decision. However on decision n ($n>1$) the subject may be affected by d_{n-1}. We measure such an effect with a parameter α and we hypothesise that he or she weighs the present decision with α ($0 \leq \alpha \leq 1$), and the previous decision with $(1-\alpha)$. Hence our hypothesis assumes that the subject is thinking that he or she is facing a choice between $[d_{n-1},(1-\alpha);L_n,\alpha]$ and $[d_{n-1},(1-\alpha);R_n,\alpha]$ instead of a choice between L_n and R_n (where $[a,p_a;b,p_b]$ denotes a lottery which yields a with probability p_a and b with probability p_b). We interpret α as a behavioural parameter, indicating the weight that the subject attaches to the previous decision when taking the present decision. Note that if α takes the value 1 then the subject separates completely.

We should emphasise that under this hypothesis, there *is* contamination of any decision by *all* previous decisions – but the contamination is *indirect* (through the previous decision, which, in turn is contaminated by the one before that, and so on) and not *direct*, as in Hey and Lee.

We use the data from Hey (2001). In that experiment there were 53 subjects; they were asked to complete 5 experimental sessions; each session contained 100 pairwise-lottery questions; there were four possible outcomes (-£25, £25, £75 and £125), though each lottery contained at most three of these four outcomes; subjects were given a show-up fee of £25. The pairwise-lotteries were randomly ordered across sessions and across subjects. Subjects had to indicate their preferred lottery in each pair.

They were told that one of the 500 pairwise-lotteries would be randomly selected at the end of the experiment; their preferred lottery on that question would be played out; and they would be paid accordingly. We have 25,600 observations.

3.3 The econometric method

We confine the technical detail and the mathematics to an Unpublished Appendix available online at the EXEC site. Given the nature of our data and our hypotheses, we have to fit preference functionals. Clearly the functional has to be non-EU[4]; the natural choice is Rank Dependent Expected Utility (RDEU)[5] which now seems to be the most widely accepted of all non-EU functions. RDEU involves a *weighting function* (weighting the probabilities); we choose that of Wu and Gonzalez (1996), which seems to be generally accepted as being empirically valid. Further, in view of the fact that subjects make errors when taking decisions, and in order to fit

[4] For otherwise there could be no contamination.

[5] See Wakker (2010), though we should note that he prefers to call it Rank Dependent Utility (or even Prospect Theory).

preference functionals, we have to assume some stochastic process; here we follow Hey (2001) and others in assuming that a normally distributed random variable with zero mean is added to the true difference in the valuations of the two lotteries in each decision-problem to determine the perceived best lottery. We use the constrained maximum-likelihood routine in GAUSS to fit the data under the two hypotheses, by estimating the parameters for both hypotheses and α for the Contamination Hypothesis. In so doing we obtain the maximised log-likelihoods under the two hypotheses: LL_S and LL_C. We can then test whether the Contamination Hypothesis fits the data significantly better than the Separation Hypothesis.

3.4 Results

The results can be summarised very quickly in Figure 3.1, which is a histogram of the differences between the LL_S and LL_C. Using a standard likelihood-ratio test, the Contamination Hypothesis fits significantly better if $(LL_C - LL_S)$ is larger than 1.96 (at the 5% level) and 3.32 (at the 1% level). The figure shows clearly that there are just 3 subjects in the former category and 2 in the latter. Indeed one might even argue that, at these levels of significance, such numbers being significant can be expected (5% of 53 is 2.65 and 1% of 53 is 0.5), even if the Separation Hypothesis is true. The RLI survives this test of its validity.

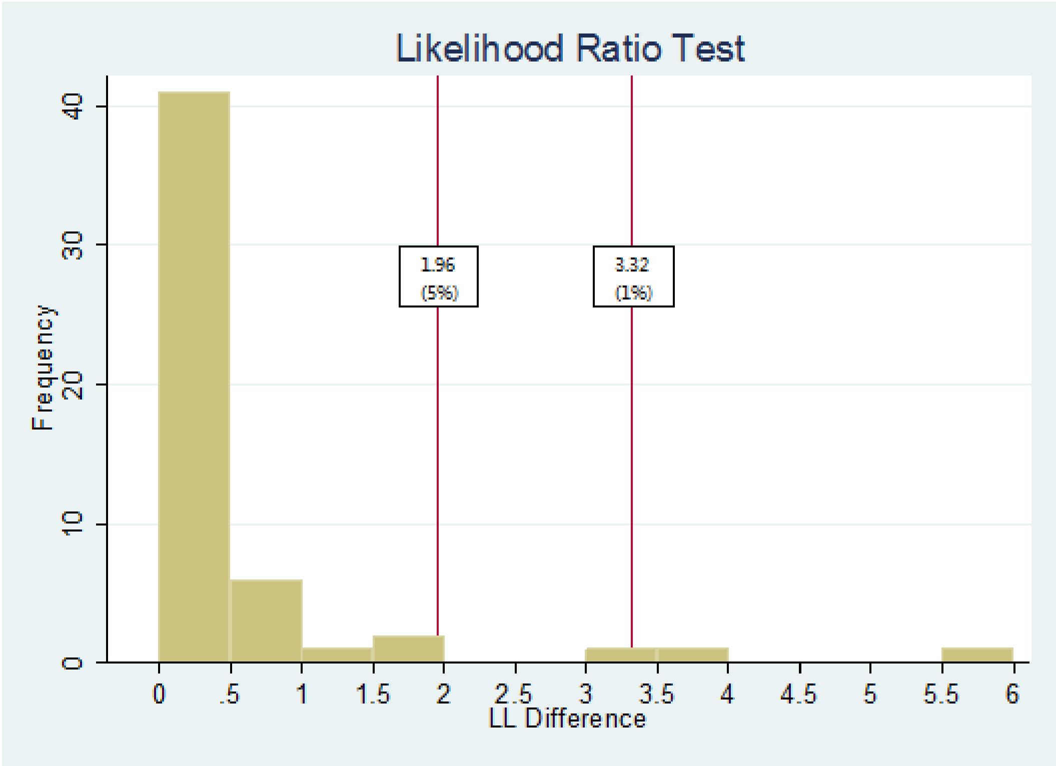


Figure 3.1: the results

3.5 Conclusions

This chapter tests for the existence of a possible form of contamination in experiments which use the Random Lottery Incentive mechanism. This mechanism is one in which subjects are posed a (large) set of decision problems, and where their payment is determined by their decision *on a randomly chosen* one of these problems. Critics have argued that one cannot necessarily infer the same things from the resulting data set as one could from a data set in which subjects were asked just one question: subjects' responses might be contaminated by their responses on other questions. We test here a particularly appealing and cognitively simple possible form of contamination: where the subjects' decision on one problem might be directly affected by their decision on the immediately preceding question, and hence indirectly on all preceding questions.

We test this Contamination Hypothesis against the usual Separation Hypothesis, and find that for just 3 (2) of the 53 subjects in the experiment, the former fits significantly better than the latter at the 5% (1%) level. It seems that the subjects do not suffer from this form of contamination. This provides further support for the continued use of the Random Lottery Incentive mechanism.

Appendix A.

Experimental Instruction for Chapter 1

These are the instructions given to the subjects in the experiment in November 2013.

We have done the experiment twice. One in May 2013 and the other in November 2013. The instructions for the two sessions were very similar. The only difference between them is the exchange rate of energy expenditure and money. In May 2013, one unit of energy expenditure is equal to one penny. In November 2013, one unit of energy expenditure is equal to 1.5 pence.

We have four treatments in the experiment. Here the instructions are only for treatment 1, since the instructions for the four treatments are very similar. The differences across treatments are the information quantity and quality.

In treatment 1, subjects can see three squares away with ranges 0, 20, 50.

In treatment 2, subjects can see two squares away with ranges 0, 20.

In treatment 3, subjects can see three squares away with ranges 0, 40, 100.

In treatment 4, subjects can see two squares away with ranges 0, 40.

Instructions

(Treatment 1)

You are about to take part in an economics experiment which is designed to observe individuals' decision making processes when they have vague information about the future. If you read the following instructions carefully, you can – depending on your decisions – earn money in addition to your £2.50 show-up fee, which you cannot lose. You will get paid in cash at the end of the experiment. You will fill a demographic survey. This survey does **NOT** affect your payoff.

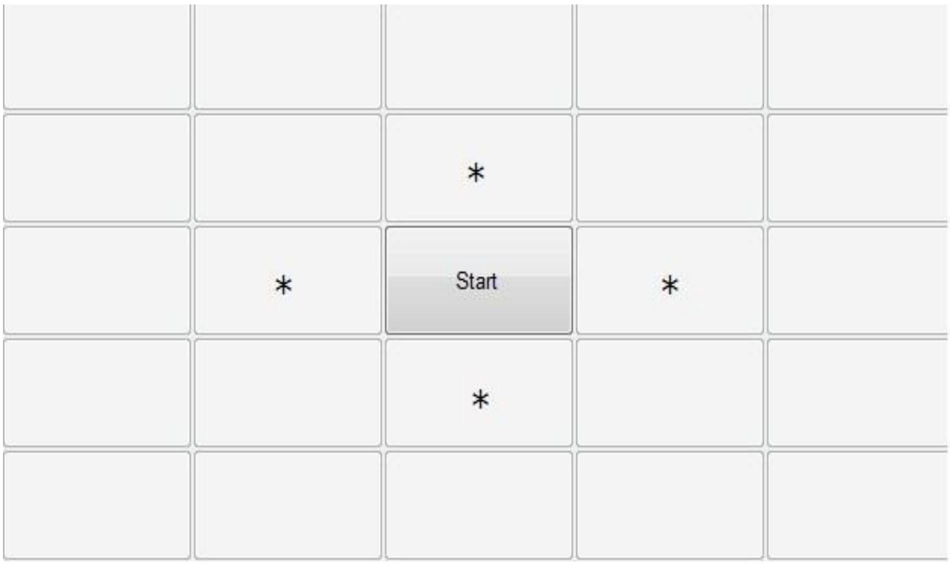
This is an individual experiment. Please do not communicate with other subjects during the experiment. If you have any questions, please let the experimenter know and the experimenter will answer them privately. We fear that if you violate this rule we will have to exclude you from the experiment.

The Background

The experiment is based on a story in which you are walking through hilly country and it is foggy. You are trying to get to a destination. You know where and how far away the destination is. But due to the thick fog, you cannot see the terrain around you clearly and hence you cannot immediately see the best way to get to your destination. You only have some vague information about the near terrain. Crucially both walking up and down hill uses energy – though going down less than going up. Your aim is to arrive at the destination with the least expenditure of energy.

The Experiment

This experiment involves four *journeys* with four different maps, each of which is divided into *squares*. These maps are modified from different parts of the real world. But you do not know which parts they are. In the experiment, each map contains 200 by 200 squares. You have to move between squares till you reach the destination. The moving rule is that you can only move to squares which are share a boundary with your current square. In other words, you can only move to the adjacent square up, down, left or right relative to your current square.



As the picture above showed, supposing your current square is the square with “Start” on it, you can and only can move to one of the squares with * on them.

When you start a journey, you will see a square with “Start” written on it.

This is your starting square. You may not see the “Destination” square on your screen because sometimes it is too far away and off the screen. But it will appear when you get near to it. You can always check the position of the destination square relative to your current square from information which will be given at the left-top of your screen. When you are moving across the terrain, you are always the centre, and the destination is getting closer to you (if you are getting closer to it).

On each journey, you click the “Start” button and hence start the journey. Then you will see a number on the “Start” square. It is the height of that square. As you will notice, for simplicity all heights are expressed as an integer. Because the area is hilly, different squares have different heights. That means that you may have to move up or down. You have to spend energy when moving. At each square you have to decide which square you want to move to next. Remember that you can only move to the adjacent square up, down, left or right. The software does not allow you to actually move to your chosen square until at least 10 seconds have elapsed; this is to ensure that you spend some time thinking about your decision.

Because of the fog, you cannot see the heights of other squares clearly. But you will be given some information which depends on how far away are other squares. If a square is adjacent to (that is, one square away from) the square that you are currently on, you will be told its height precisely; if a square is two squares away from the square that you are currently on, an interval of width 20 will be specified, and you will be told that the height is somewhere in that interval (with all integer heights in the interval equally likely); if a square is three squares away from the square that you are currently on, an interval of width 50 will be specified, and you will be told that the height is somewhere in that interval (with all integer heights in the interval equally likely); if a square is more than three squares away from the square that you are currently on, you will be told nothing about its height. The fog is that thick!

When you move from one square to another, you spend energy. Moving up uses more energy than moving down. More precisely, moving up uses energy equal to **twice** the difference between the heights of the two squares, while moving down uses energy equal to the difference between the heights of the two squares. The software keeps a record of the energy you spend in the “expenditure” box on the right-top screen.

You will have an endowment of cash at the beginning of each journey. This endowment varies from journey to journey since the maps differ. The payoff of any one journey is the endowment on that journey minus the energy you spend, where **each unit of energy costs 1.5 penny**. If your expenditure of energy on any journey exceeds your endowment, your payoff for that journey will be zero.

We will record the payoff of all four journeys. Your payment for the experiment as a whole will be the payoff on a randomly chosen one of the four journeys, plus the showup fee of £2.50.

Examples

- **Your expenditure when you are moving up or down**

  You have to spend energy when you are moving up or down. Moving-up expenditure is **twice** the difference between the heights of the two squares. Moving-down expenditure is equal to the difference between the heights of the two squares.

  1. If the height of your current square is 300, and the height of the square to which you moving is 400, you are moving up. Your energy expenditure is 2(400 – 300) = 200.

     It costs 200 * 1.5= 300 (pence).
  2. If the height of your current square is 400, and the height of the square to which you moving is 300, you are moving down. Your energy expenditure is (400 - 300) = 100.

     It costs 100 * 1.5=150 (pence).

Control Questions:

These questions are designed to help you test your understanding of the experiment.

1. If the height of your current square is 500, and then you move to another square with a height of 600, are you moving up or down? And how much is your expenditure for this move?
2. If the height of your current square is 600, and then you move to another square with a height of 500, are you moving up or down? And how much is your expenditure for this move?
3. If your payoffs on journeys one to four are £10.25, £12.89, £9.01, £5.33 and the random process selects journey two as the payoff journey, what is your payment from this experiment? (Do remember to include the £2.50 show-up fee.)

Answers:

1. Moving up, because 500 (the height of the current square) is less than 600 (the height of the next square), the expenditure is 2(600 – 500) which is 200. And it costs 200*1.5=300 (pence).
2. Moving down, because 600 (the height of the current square) is greater than 500 (the height of the next square), the expenditure is (600 – 500) which is 100. And it costs 100*1.5=150 (pence).
3. £15.39.

Appendix B.

Experimental Instruction for Chapter 2

Instructions

Preamble

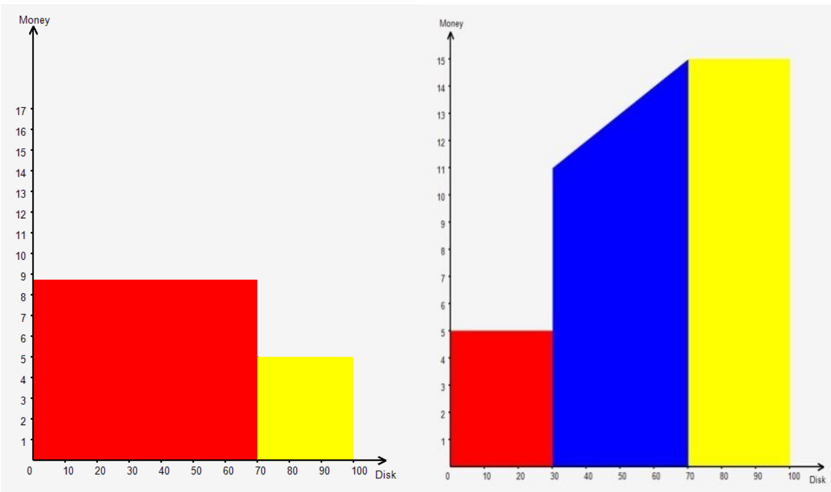
Welcome to this experiment. These instructions are to help you to understand what you are being asked to do during the experiment and how you will be paid. The experiment is simple and gives you the chance to earn a considerable amount of money, which will be paid to you in cash after you have completed the experiment. The payment described below is *in addition* to a participation fee of £2.50 that you will be paid independently of your answers.

The Experiment

The experiment is interested in *your* preferences under risk. There are no right or wrong answers. It is in four *parts*. Each of the four parts consists of a series of *problems*. At the end of all four parts, one of the four parts will be randomly selected, then one of the problems on that part will be randomly selected, and then you will play out that problem. This will always imply playing out a lottery. The outcome of playing out this lottery will lead to a *payoff* to you, and we shall pay this to you in cash, plus the participation fee of £2.50, immediately after you have completed the experiment. How all this will be done will be explained below.We start by describing a generic lottery. Then we describe the four parts; you will not necessarily get them in the order that they are described here.

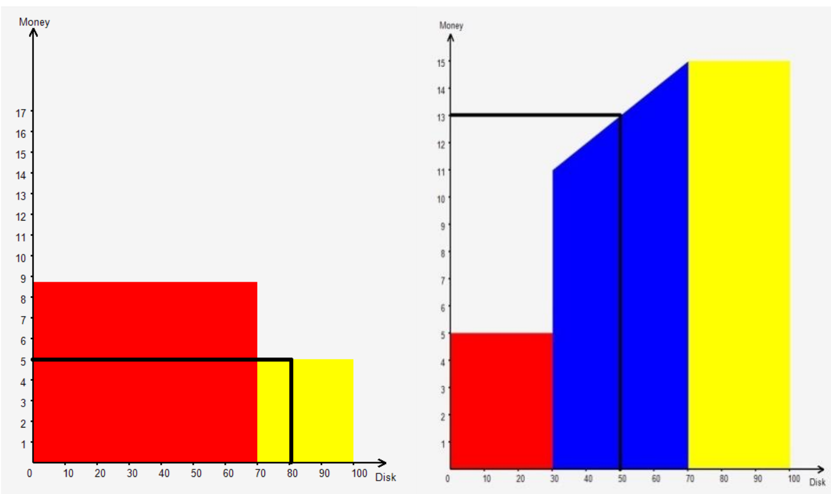
A Generic Lottery

We describe now what we mean by a ‘Generic Lottery’. Here we represent each lottery visually. The visual representation will be one of the following two forms.



It is simplest to explain these in terms of the implications for your payment if one of these is randomly selected to be played out at the end of the experiment. What we will do in all cases is to ask you to draw – without looking – a disk out of a bag containing 100 disks numbered from 1 to 100. You can check that the bag contains all these disks before you do the drawing. The number on the disk that you draw will determine a point on the horizontal axis; your payment would be the amount on the vertical axis implied by that point through the figure. So, for example, in the left-hand lottery, if the number on the disk that you draw is between 1 and 70 inclusive you would get £8.75; if it is between 71 and 100 inclusive you would get £5.This implies that the chance of you getting paid £8.75 is 0.7 and the chance of you getting paid £5 is 0.3. This will also be written in words. If the right-hand lottery is to be played out, if the number on the disk that you draw is between 1 and 30 you would get £5; if it is between 31 and 70 inclusive you would get between £11 and £15 – the precise amount depending upon the number on the disk drawn; if it is between 71 and 100 inclusive you would get £15.

Let us give specific examples.In the left-hand lottery, suppose the number on the disk that you drawis 80, then you would receive £5. In the right-hand lottery, suppose the number on the disk that you draw is 50, you would receive £13.



We now describe the four parts of the experiment. Remember that you might not get them in the order presented here.

Part 1: Pairwise Choices

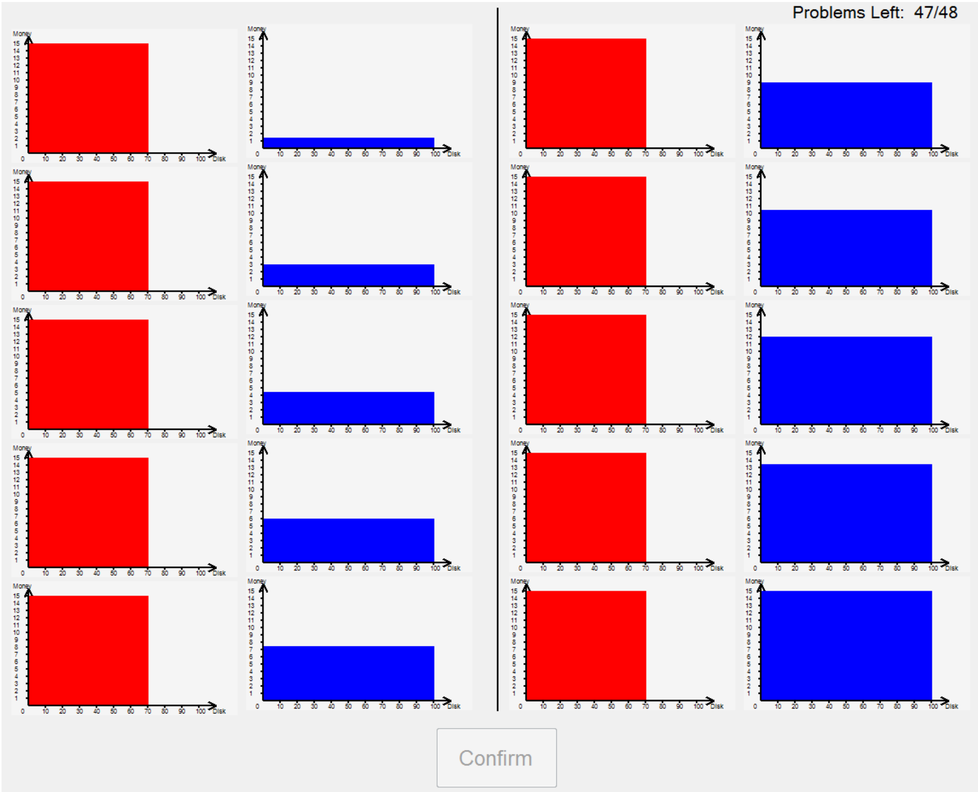
Here each problem is a simple *pairwise choice*, an example of which is pictured below. In each problem you have to decide which of two lotteries you prefer. If this problem on this part is chosen for payment at the end of the experiment, then the lottery that you chose will be the one that is played out.

In the example below, the left-hand lottery is certain, and the right-hand lottery is risky; in some problems both lotteries are risky; in some lotteries one of the amounts will be £0 and thus not appear in the figure. In the figure we show the amounts of money you might win on the vertical axis and the disk number on the horizontal axis. The implications are written in words underneath the figure. So the left-hand lottery would lead to a payoff of £10 with certainty; the right-hand lottery would lead to a payoff of £15 with chance 0.4 or to a payoff of £0 with chance 0.6; you will be asked to click on the lottery that you would prefer to have played out.

In this part you will be asked to express your preference over a total of 80 such problems. In the upper-right corner of the screen you will be told how many problems remain. In each problem, you cannot take a decision until at least five seconds have elapsed, but you can take as long as you like.

Part 2: Lists

In some ways this part is similar to Part 1, though here the pairwise choices are structured. Each problem is in the form of a list. One such list is shown in the figure below. In each list there is a set of pairwise choice problems, presented in exactly the same way as in Part 1. But, as you will, see there is a pattern: one of the two lotteries in any pair is the same throughout the list – here the left-hand lottery is always £15 with chance 0.7 and £0 with chance 0.3. The other lottery is changing through the list – in the sense that the chance of getting the higher amount of money is increasing, or the amount of money is increasing.

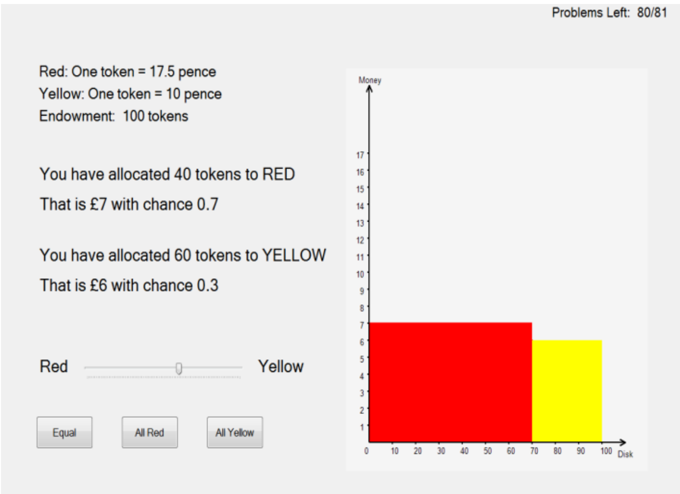


In this particular list the left-hand lottery is always the same (£15 with chance 0.7 and £0 with chance 0.3), while the right-hand lottery is a certainty with the amount of money going up from £1.50 to £15 through the list. As in Part 1, in each pair you are asked to specify which lottery you prefer. You do this by clicking on the preferred lottery; you will see that when you do this, the other lottery becomes greyed-out. However, because one of the lotteries is getting better through the list, we impose some structure on your answers. If you say that you prefer the certainty at one point, we force you to say that you also prefer the certainty further down the list. You will understand this as you click through the list. When, in each pair, one of the lotteries has been indicated as preferred by you (and the other in the pair greyed-out) the 'Confirm' button will become active, allowing you to record your preferences for that list and move onto to the next list.

There are a total of 48 lists in this part of the experiment. In the upper-right corner of the screen you will be told how many problems remain. In each problem, you cannot take a decision until at least five seconds have elapsed, but you can take as long as you like.

Part 3: Allocations

In each problem in this part, you will be given a quantity of *tokens* to allocate between two risky colours with stated chances. For each colour you will be told the *exchange rate* between tokens and money. An example of such a problem is shown below.



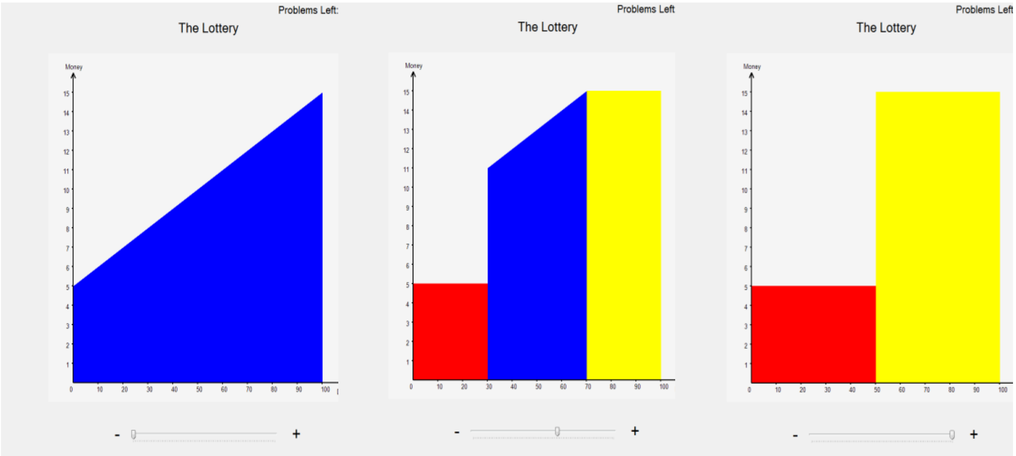
Here there are100 tokens to allocate; the chance of red happening is 0.7 and that of yellow 0.3. You have to decide how to allocate the 100 tokens between red and yellow; shown is an example of allocation but you may prefer a different one. Your chosen allocation implies a lottery – as pictured above. If this problem were to be played out at the end of the experiment, this lottery would be played out.

There will be a total of 81 problems in this part. In the upper-right corner of the screen you will be told how many problems remain. In each problem, you cannot take a decision until at least five seconds have elapsed, but you can take as long as you like.

Part 4: Lottery Choices

In this part, in each problem you will be asked to choose a lottery. The choice set is the continuum between two 'extreme' lotteries illustrated in the figure below: the left-hand lotterycan give any payment *between*£5 and £15, with all payments being equally likely; the right-hand lottery consists of a simple lottery with two possible outcomes, here £5 with chance 0.5 and £15 with chance 0.5. Your chosen lottery can be any mixture of these two 'extreme' lotteries. As you move the slider bar from the extreme left to the extreme right you will see the mixture lottery moves from one of the two extremes to the other. The implied payments are between £5 and £15, with the chances indicated in the figure. One such problem is shown below.

In the example below, if you accept a mixture lottery like this, you would be paid £5 with chance of 0.3, £15 with chance of 0.3, and between £11 and £15 depending on the number on the disk that you randomly draw at the end of the experiment.



There will be a total of 54 problems in this part. In the upper-right corner of the screen you will be told how many problems remain. In each problem, you cannot take a decision until at least five seconds have elapsed, but you can take as long as you like.

The Payment Procedure

When you have completed the experiment, one of the experimenters will come to you. The experimenter will have a record of your decisions in each part of the experiment. You will then be asked to go into an adjoining room for payment. There will be another experimenter, who has on their computer all the decisions that you took. Then the following procedure will be followed.

1. First you will draw – without looking – a disk out of a bag containing disks numbered from 1 to 4. The number on the disk will determine on which part of the experiment your payment will be determined.

2. *If the number on the disk is 1,* then one of your answers in part 1 will determine your payment. You will draw – without looking – a disk from a bag containing disks numbered from 1 to 80 (the number of problems in part 1). This will determine the problem to be played out. The experimenter will then retrieve from the computer your decision on that problem. This will be a lottery. This will then be played out as described above (with you drawing – without looking – a disk out of a bag containing numbered disks from 1 to 100).
3. *If the number on the disk is 2,* then one of your answers in part 2 will determine your payment. You will draw – without looking – a disk from a bag containing disks numbered from 1 to 48 (the number of problems in part 2). This will determine the problem to be played out. This will be a list. In each list there are 10 pairwise choices. You will then draw – without looking – a disk from a bag containing disks numbered from 1 to 10. This will determine a particular pairwise choice in that list. The experimenter will then retrieve from the computer your decision on that pairwise choice. This may be a certainty or a lottery. If it is a certainty, you will receive that amount of money. It is a lottery it will then be played out as described above (with you drawing – without looking – a disk out of a bag containing numbered disks from 1 to 100).
4. *If the number on the disk is 3,* then one of your answers in part 3 will determine your payment. You will draw – without looking – a

disk from a bag containing disks numbered from 1 to 81 (the number of problems in part 3). This will determine the problem to be played out. The experimenter will then retrieve from the computer your decision on that problem. This will be a lottery. This will then be played out as described above (with you drawing – without looking – a disk out of a bag containing numbered disks from 1 to 100).

5. *If the number on the disk is 4,* then one of your answers in part 4 will determine your payment. You will draw – without looking – a disk from a bag containing disks numbered from 1 to 54 (the number of problems in part 4). This will determine the problem to be played out. The experimenter will then retrieve from the computer your decision on that problem. This will be a lottery. This will then be played out as described above (with you drawing – without looking – a disk out of a bag containing numbered disks from 1 to 100).

The show-up fee of £2.50 will be added to the payment as described above. You will be paid in cash, be asked to sign a receipt and then you are free to go.

If you have any questions, please ask one of the experimenters.

Thank you for your participation.

Appendix C.

Problem Lists for Chapter 2

1. Allocation

ProblemNo	Endow	ProbX	ProbY	ExX	ExY
1	100	0.1	0.9	0.5	1
2	100	0.1	0.9	0.57	1
3	100	0.1	0.9	0.67	1
4	100	0.1	0.9	0.8	1
5	100	0.1	0.9	1	1
6	100	0.1	0.9	1.25	1
7	100	0.1	0.9	1.5	1
8	100	0.1	0.9	1.75	1
9	100	0.1	0.9	2	1
10	100	0.2	0.8	0.5	1
11	100	0.2	0.8	0.57	1
12	100	0.2	0.8	0.67	1
13	100	0.2	0.8	0.8	1
14	100	0.2	0.8	1	1
15	100	0.2	0.8	1.25	1
16	100	0.2	0.8	1.5	1
17	100	0.2	0.8	1.75	1
18	100	0.2	0.8	2	1
19	100	0.3	0.7	0.5	1
20	100	0.3	0.7	0.57	1
21	100	0.3	0.7	0.67	1
22	100	0.3	0.7	0.8	1
23	100	0.3	0.7	1	1
24	100	0.3	0.7	1.25	1
25	100	0.3	0.7	1.5	1
26	100	0.3	0.7	1.75	1
27	100	0.3	0.7	2	1
28	100	0.4	0.6	0.5	1
29	100	0.4	0.6	0.57	1
30	100	0.4	0.6	0.67	1
31	100	0.4	0.6	0.8	1
32	100	0.4	0.6	1	1
33	100	0.4	0.6	1.25	1
34	100	0.4	0.6	1.5	1
35	100	0.4	0.6	1.75	1
36	100	0.4	0.6	2	1
37	100	0.5	0.5	0.5	1
38	100	0.5	0.5	0.57	1
39	100	0.5	0.5	0.67	1
40	100	0.5	0.5	0.8	1
41	100	0.5	0.5	1	1

42	100	0.5	0.5	1.25	1
43	100	0.5	0.5	1.5	1
44	100	0.5	0.5	1.75	1
45	100	0.5	0.5	2	1
46	100	0.6	0.4	0.5	1
47	100	0.6	0.4	0.57	1
48	100	0.6	0.4	0.67	1
49	100	0.6	0.4	0.8	1
50	100	0.6	0.4	1	1
51	100	0.6	0.4	1.25	1
52	100	0.6	0.4	1.5	1
53	100	0.6	0.4	1.75	1
54	100	0.6	0.4	2	1
55	100	0.7	0.3	0.5	1
56	100	0.7	0.3	0.57	1
57	100	0.7	0.3	0.67	1
58	100	0.7	0.3	0.8	1
59	100	0.7	0.3	1	1
60	100	0.7	0.3	1.25	1
61	100	0.7	0.3	1.5	1
62	100	0.7	0.3	1.75	1
63	100	0.7	0.3	2	1
64	100	0.8	0.2	0.5	1
65	100	0.8	0.2	0.57	1
66	100	0.8	0.2	0.67	1
67	100	0.8	0.2	0.8	1
68	100	0.8	0.2	1	1
69	100	0.8	0.2	1.25	1
70	100	0.8	0.2	1.5	1
71	100	0.8	0.2	1.75	1
72	100	0.8	0.2	2	1
73	100	0.9	0.1	0.5	1
74	100	0.9	0.1	0.57	1
75	100	0.9	0.1	0.67	1
76	100	0.9	0.1	0.8	1
77	100	0.9	0.1	1	1
78	100	0.9	0.1	1.25	1
79	100	0.9	0.1	1.5	1
80	100	0.9	0.1	1.75	1
81	100	0.9	0.1	2	1

Key:

ProblemNo: Problem Number

Endow: the endowment (by tokens)

ProbX: the probability of red

ProbY: the probability of yellow

ExX: the exchange rate of red

ExY: the exchange rate of yellow

2. Lottery Choices

ProblemNo	AmountX	ProbX	AmountY
1	0	0.1	5
2	0	0.2	5
3	0	0.3	5
4	0	0.4	5
5	0	0.5	5
6	0	0.6	5
7	0	0.7	5
8	0	0.8	5
9	0	0.9	5
10	0	0.1	10
11	0	0.2	10
12	0	0.3	10
13	0	0.4	10
14	0	0.5	10
15	0	0.6	10
16	0	0.7	10
17	0	0.8	10
18	0	0.9	10
19	0	0.1	15
20	0	0.2	15
21	0	0.3	15
22	0	0.4	15
23	0	0.5	15
24	0	0.6	15
25	0	0.7	15
26	0	0.8	15
27	0	0.9	15
28	5	0.1	10
29	5	0.2	10
30	5	0.3	10
31	5	0.4	10
32	5	0.5	10
33	5	0.6	10
34	5	0.7	10
35	5	0.8	10
36	5	0.9	10
37	5	0.1	15
38	5	0.2	15
39	5	0.3	15
40	5	0.4	15
41	5	0.5	15

42	5	0.6	15
43	5	0.7	15
44	5	0.8	15
45	5	0.9	15
46	10	0.1	15
47	10	0.2	15
48	10	0.3	15
49	10	0.4	15
50	10	0.5	15
51	10	0.6	15
52	10	0.7	15
53	10	0.8	15
54	10	0.9	15

Key:

ProblemNo: Problem Number

AmoutX: the first outcome of the lottery

ProbX: the probability of the first outcome of the lottery

AmountY: the second outcome of the lottery

3. Pairwise Choices

ProblemNo	AmountXL	ProbXL	AmountYL	AmoutXR	ProbXR	AmountYR
1	10	0.6	5	10	0.8	0
2	10	0.4	5	10	0.8	0
3	10	0.4	5	10	0.6	0
4	10	0.2	5	10	0.8	0
5	10	0.2	5	10	0.6	0
6	10	0.2	5	10	0.4	0
7	10	0	5	10	0.8	0
8	10	0	5	10	0.6	0
9	10	0	5	10	0.4	0
10	10	0	5	10	0.2	0
11	5	1	0	10	0.8	0
12	5	1	0	10	0.6	0
13	5	1	0	10	0.4	0
14	5	1	0	10	0.2	0
15	5	0.8	0	10	0.6	0
16	5	0.8	0	10	0.4	0
17	5	0.8	0	10	0.2	0
18	5	0.6	0	10	0.4	0
19	5	0.6	0	10	0.2	0
20	5	0.4	0	10	0.2	0
21	15	0.6	5	15	0.8	0
22	15	0.4	5	15	0.8	0
23	15	0.4	5	15	0.6	0
24	15	0.2	5	15	0.8	0
25	15	0.2	5	15	0.6	0
26	15	0.2	5	15	0.4	0
27	15	0	5	15	0.8	0
28	15	0	5	15	0.6	0
29	15	0	5	15	0.4	0
30	15	0	5	15	0.2	0
31	5	1	0	15	0.8	0
32	5	1	0	15	0.6	0
33	5	1	0	15	0.4	0
34	5	1	0	15	0.2	0
35	5	0.8	0	15	0.6	0
36	5	0.8	0	15	0.4	0
37	5	0.8	0	15	0.2	0
38	5	0.6	0	15	0.4	0
39	5	0.6	0	15	0.2	0
40	5	0.4	0	15	0.2	0

41	15	0.6	10	15	0.8	0
42	15	0.4	10	15	0.8	0
43	15	0.4	10	15	0.6	0
44	15	0.2	10	15	0.8	0
45	15	0.2	10	15	0.6	0
46	15	0.2	10	15	0.4	0
47	15	0	10	15	0.8	0
48	15	0	10	15	0.6	0
49	15	0	10	15	0.4	0
50	15	0	10	15	0.2	0
51	10	1	0	15	0.8	0
52	10	1	0	15	0.6	0
53	10	1	0	15	0.4	0
54	10	1	0	15	0.2	0
55	10	0.8	0	15	0.6	0
56	10	0.8	0	15	0.4	0
57	10	0.8	0	15	0.2	0
58	10	0.6	0	15	0.4	0
59	10	0.6	0	15	0.2	0
60	10	0.4	0	15	0.2	0
61	15	0.6	10	15	0.8	5
62	15	0.4	10	15	0.8	5
63	15	0.4	10	15	0.6	5
64	15	0.2	10	15	0.8	5
65	15	0.2	10	15	0.6	5
66	15	0.2	10	15	0.4	5
67	15	0	10	15	0.8	5
68	15	0	10	15	0.6	5
69	15	0	10	15	0.4	5
70	15	0	10	15	0.2	5
71	10	1	5	15	0.8	5
72	10	1	5	15	0.6	5
73	10	1	5	15	0.4	5
74	10	1	5	15	0.2	5
75	10	0.8	5	15	0.6	5
76	10	0.8	5	15	0.4	5
77	10	0.8	5	15	0.2	5
78	10	0.6	5	15	0.4	5
79	10	0.6	5	15	0.2	5
80	10	0.4	5	15	0.2	5

Key:

ProblemNo: Problem Number

AmountXL: the first outcome of the left lottery

ProbXL: the probability of the first outcome of the left lottery

AmountYL: the second outcome of the left lottery

AmountXR: the first outcome of the right lottery

ProbXR: the probability of the first outcome of the right lottery

AmountYR: the second outcome of the right lottery

4. Holt-Laury Price Lists

ListNo	AmountX L	ProbXL	AmountY L	AmountX R	ProbX R	AmountY R
1	5	1	0	2.5	1	0
1	5	0.9	0	2.5	1	0
1	5	0.8	0	2.5	1	0
1	5	0.7	0	2.5	1	0
1	5	0.6	0	2.5	1	0
1	5	0.5	0	2.5	1	0
1	5	0.4	0	2.5	1	0
1	5	0.3	0	2.5	1	0
1	5	0.2	0	2.5	1	0
1	5	0.1	0	2.5	1	0
2	10	1	0	5	1	0
2	10	0.9	0	5	1	0
2	10	0.8	0	5	1	0
2	10	0.7	0	5	1	0
2	10	0.6	0	5	1	0
2	10	0.5	0	5	1	0
2	10	0.4	0	5	1	0
2	10	0.3	0	5	1	0
2	10	0.2	0	5	1	0
2	10	0.1	0	5	1	0
3	15	1	0	7.5	1	0
3	15	0.9	0	7.5	1	0
3	15	0.8	0	7.5	1	0
3	15	0.7	0	7.5	1	0
3	15	0.6	0	7.5	1	0
3	15	0.5	0	7.5	1	0
3	15	0.4	0	7.5	1	0
3	15	0.3	0	7.5	1	0
3	15	0.2	0	7.5	1	0
3	15	0.1	0	7.5	1	0
4	10	1	5	7.5	1	0
4	10	0.9	5	7.5	1	0
4	10	0.8	5	7.5	1	0
4	10	0.7	5	7.5	1	0
4	10	0.6	5	7.5	1	0
4	10	0.5	5	7.5	1	0
4	10	0.4	5	7.5	1	0

4	10	0.3	5	7.5	1	0
4	10	0.2	5	7.5	1	0
4	10	0.1	5	7.5	1	0
5	15	1	5	10	1	0
5	15	0.9	5	10	1	0
5	15	0.8	5	10	1	0
5	15	0.7	5	10	1	0
5	15	0.6	5	10	1	0
5	15	0.5	5	10	1	0
5	15	0.4	5	10	1	0
5	15	0.3	5	10	1	0
5	15	0.2	5	10	1	0
5	15	0.1	5	10	1	0
6	15	1	10	12.5	1	0
6	15	0.9	10	12.5	1	0
6	15	0.8	10	12.5	1	0
6	15	0.7	10	12.5	1	0
6	15	0.6	10	12.5	1	0
6	15	0.5	10	12.5	1	0
6	15	0.4	10	12.5	1	0
6	15	0.3	10	12.5	1	0
6	15	0.2	10	12.5	1	0
6	15	0.1	10	12.5	1	0
7	5	0.5	0	0.5	1	0
7	5	0.5	0	1	1	0
7	5	0.5	0	1.5	1	0
7	5	0.5	0	2	1	0
7	5	0.5	0	2.5	1	0
7	5	0.5	0	3	1	0
7	5	0.5	0	3.5	1	0
7	5	0.5	0	4	1	0
7	5	0.5	0	4.5	1	0
7	5	0.5	0	5	1	0
8	10	0.5	0	1	1	0
8	10	0.5	0	2	1	0
8	10	0.5	0	3	1	0
8	10	0.5	0	4	1	0
8	10	0.5	0	5	1	0
8	10	0.5	0	6	1	0
8	10	0.5	0	7	1	0
8	10	0.5	0	8	1	0

8	10	0.5	0	9	1	0
8	10	0.5	0	10	1	0
9	15	0.5	0	1.5	1	0
9	15	0.5	0	3	1	0
9	15	0.5	0	4.5	1	0
9	15	0.5	0	6	1	0
9	15	0.5	0	7.5	1	0
9	15	0.5	0	9	1	0
9	15	0.5	0	10.5	1	0
9	15	0.5	0	12	1	0
9	15	0.5	0	13.5	1	0
9	15	0.5	0	15	1	0
10	10	0.5	5	5.5	1	0
10	10	0.5	5	6	1	0
10	10	0.5	5	6.5	1	0
10	10	0.5	5	7	1	0
10	10	0.5	5	7.5	1	0
10	10	0.5	5	8	1	0
10	10	0.5	5	8.5	1	0
10	10	0.5	5	9	1	0
10	10	0.5	5	9.5	1	0
10	10	0.5	5	10	1	0
11	15	0.5	5	6	1	0
11	15	0.5	5	7	1	0
11	15	0.5	5	8	1	0
11	15	0.5	5	9	1	0
11	15	0.5	5	10	1	0
11	15	0.5	5	11	1	0
11	15	0.5	5	12	1	0
11	15	0.5	5	13	1	0
11	15	0.5	5	14	1	0
11	15	0.5	5	15	1	0
12	15	0.5	10	10.5	1	0
12	15	0.5	10	11	1	0
12	15	0.5	10	11.5	1	0
12	15	0.5	10	12	1	0
12	15	0.5	10	12.5	1	0
12	15	0.5	10	13	1	0
12	15	0.5	10	13.5	1	0
12	15	0.5	10	14	1	0
12	15	0.5	10	14.5	1	0

12	15	0.5	10	15	1	0
13	5	0.2	0	0.5	1	0
13	5	0.2	0	1	1	0
13	5	0.2	0	1.5	1	0
13	5	0.2	0	2	1	0
13	5	0.2	0	2.5	1	0
13	5	0.2	0	3	1	0
13	5	0.2	0	3.5	1	0
13	5	0.2	0	4	1	0
13	5	0.2	0	4.5	1	0
13	5	0.2	0	5	1	0
14	5	0.3	0	0.5	1	0
14	5	0.3	0	1	1	0
14	5	0.3	0	1.5	1	0
14	5	0.3	0	2	1	0
14	5	0.3	0	2.5	1	0
14	5	0.3	0	3	1	0
14	5	0.3	0	3.5	1	0
14	5	0.3	0	4	1	0
14	5	0.3	0	4.5	1	0
14	5	0.3	0	5	1	0
15	5	0.4	0	0.5	1	0
15	5	0.4	0	1	1	0
15	5	0.4	0	1.5	1	0
15	5	0.4	0	2	1	0
15	5	0.4	0	2.5	1	0
15	5	0.4	0	3	1	0
15	5	0.4	0	3.5	1	0
15	5	0.4	0	4	1	0
15	5	0.4	0	4.5	1	0
15	5	0.4	0	5	1	0
16	5	0.6	0	0.5	1	0
16	5	0.6	0	1	1	0
16	5	0.6	0	1.5	1	0
16	5	0.6	0	2	1	0
16	5	0.6	0	2.5	1	0
16	5	0.6	0	3	1	0
16	5	0.6	0	3.5	1	0
16	5	0.6	0	4	1	0
16	5	0.6	0	4.5	1	0
16	5	0.6	0	5	1	0

17	5	0.7	0	0.5	1	0
17	5	0.7	0	1	1	0
17	5	0.7	0	1.5	1	0
17	5	0.7	0	2	1	0
17	5	0.7	0	2.5	1	0
17	5	0.7	0	3	1	0
17	5	0.7	0	3.5	1	0
17	5	0.7	0	4	1	0
17	5	0.7	0	4.5	1	0
17	5	0.7	0	5	1	0
18	5	0.8	0	0.5	1	0
18	5	0.8	0	1	1	0
18	5	0.8	0	1.5	1	0
18	5	0.8	0	2	1	0
18	5	0.8	0	2.5	1	0
18	5	0.8	0	3	1	0
18	5	0.8	0	3.5	1	0
18	5	0.8	0	4	1	0
18	5	0.8	0	4.5	1	0
18	5	0.8	0	5	1	0
19	10	0.2	0	1	1	0
19	10	0.2	0	2	1	0
19	10	0.2	0	3	1	0
19	10	0.2	0	4	1	0
19	10	0.2	0	5	1	0
19	10	0.2	0	6	1	0
19	10	0.2	0	7	1	0
19	10	0.2	0	8	1	0
19	10	0.2	0	9	1	0
19	10	0.2	0	10	1	0
20	10	0.3	0	1	1	0
20	10	0.3	0	2	1	0
20	10	0.3	0	3	1	0
20	10	0.3	0	4	1	0
20	10	0.3	0	5	1	0
20	10	0.3	0	6	1	0
20	10	0.3	0	7	1	0
20	10	0.3	0	8	1	0
20	10	0.3	0	9	1	0
20	10	0.3	0	10	1	0

21	10	0.4	0	1	1	0
21	10	0.4	0	2	1	0
21	10	0.4	0	3	1	0
21	10	0.4	0	4	1	0
21	10	0.4	0	5	1	0
21	10	0.4	0	6	1	0
21	10	0.4	0	7	1	0
21	10	0.4	0	8	1	0
21	10	0.4	0	9	1	0
21	10	0.4	0	10	1	0
22	10	0.6	0	1	1	0
22	10	0.6	0	2	1	0
22	10	0.6	0	3	1	0
22	10	0.6	0	4	1	0
22	10	0.6	0	5	1	0
22	10	0.6	0	6	1	0
22	10	0.6	0	7	1	0
22	10	0.6	0	8	1	0
22	10	0.6	0	9	1	0
22	10	0.6	0	10	1	0
23	10	0.7	0	1	1	0
23	10	0.7	0	2	1	0
23	10	0.7	0	3	1	0
23	10	0.7	0	4	1	0
23	10	0.7	0	5	1	0
23	10	0.7	0	6	1	0
23	10	0.7	0	7	1	0
23	10	0.7	0	8	1	0
23	10	0.7	0	9	1	0
23	10	0.7	0	10	1	0
24	10	0.8	0	1	1	0
24	10	0.8	0	2	1	0
24	10	0.8	0	3	1	0
24	10	0.8	0	4	1	0
24	10	0.8	0	5	1	0
24	10	0.8	0	6	1	0
24	10	0.8	0	7	1	0
24	10	0.8	0	8	1	0
24	10	0.8	0	9	1	0
24	10	0.8	0	10	1	0
25	15	0.2	0	1.5	1	0

25	15	0.2	0	3	1	0
25	15	0.2	0	4.5	1	0
25	15	0.2	0	6	1	0
25	15	0.2	0	7.5	1	0
25	15	0.2	0	9	1	0
25	15	0.2	0	10.5	1	0
25	15	0.2	0	12	1	0
25	15	0.2	0	13.5	1	0
25	15	0.2	0	15	1	0
26	15	0.3	0	1.5	1	0
26	15	0.3	0	3	1	0
26	15	0.3	0	4.5	1	0
26	15	0.3	0	6	1	0
26	15	0.3	0	7.5	1	0
26	15	0.3	0	9	1	0
26	15	0.3	0	10.5	1	0
26	15	0.3	0	12	1	0
26	15	0.3	0	13.5	1	0
26	15	0.3	0	15	1	0
27	15	0.4	0	1.5	1	0
27	15	0.4	0	3	1	0
27	15	0.4	0	4.5	1	0
27	15	0.4	0	6	1	0
27	15	0.4	0	7.5	1	0
27	15	0.4	0	9	1	0
27	15	0.4	0	10.5	1	0
27	15	0.4	0	12	1	0
27	15	0.4	0	13.5	1	0
27	15	0.4	0	15	1	0
28	15	0.6	0	1.5	1	0
28	15	0.6	0	3	1	0
28	15	0.6	0	4.5	1	0
28	15	0.6	0	6	1	0
28	15	0.6	0	7.5	1	0
28	15	0.6	0	9	1	0
28	15	0.6	0	10.5	1	0
28	15	0.6	0	12	1	0
28	15	0.6	0	13.5	1	0
28	15	0.6	0	15	1	0
29	15	0.7	0	1.5	1	0
29	15	0.7	0	3	1	0

29	15	0.7	0	4.5	1	0
29	15	0.7	0	6	1	0
29	15	0.7	0	7.5	1	0
29	15	0.7	0	9	1	0
29	15	0.7	0	10.5	1	0
29	15	0.7	0	12	1	0
29	15	0.7	0	13.5	1	0
29	15	0.7	0	15	1	0
30	15	0.8	0	1.5	1	0
30	15	0.8	0	3	1	0
30	15	0.8	0	4.5	1	0
30	15	0.8	0	6	1	0
30	15	0.8	0	7.5	1	0
30	15	0.8	0	9	1	0
30	15	0.8	0	10.5	1	0
30	15	0.8	0	12	1	0
30	15	0.8	0	13.5	1	0
30	15	0.8	0	15	1	0
31	10	0.2	5	5.5	1	0
31	10	0.2	5	6	1	0
31	10	0.2	5	6.5	1	0
31	10	0.2	5	7	1	0
31	10	0.2	5	7.5	1	0
31	10	0.2	5	8	1	0
31	10	0.2	5	8.5	1	0
31	10	0.2	5	9	1	0
31	10	0.2	5	9.5	1	0
31	10	0.2	5	10	1	0
32	10	0.3	5	5.5	1	0
32	10	0.3	5	6	1	0
32	10	0.3	5	6.5	1	0
32	10	0.3	5	7	1	0
32	10	0.3	5	7.5	1	0
32	10	0.3	5	8	1	0
32	10	0.3	5	8.5	1	0
32	10	0.3	5	9	1	0
32	10	0.3	5	9.5	1	0
32	10	0.3	5	10	1	0
33	10	0.4	5	5.5	1	0
33	10	0.4	5	6	1	0
33	10	0.4	5	6.5	1	0

33	10	0.4	5	7	1	0
33	10	0.4	5	7.5	1	0
33	10	0.4	5	8	1	0
33	10	0.4	5	8.5	1	0
33	10	0.4	5	9	1	0
33	10	0.4	5	9.5	1	0
33	10	0.4	5	10	1	0
34	10	0.6	5	5.5	1	0
34	10	0.6	5	6	1	0
34	10	0.6	5	6.5	1	0
34	10	0.6	5	7	1	0
34	10	0.6	5	7.5	1	0
34	10	0.6	5	8	1	0
34	10	0.6	5	8.5	1	0
34	10	0.6	5	9	1	0
34	10	0.6	5	9.5	1	0
34	10	0.6	5	10	1	0
35	10	0.7	5	5.5	1	0
35	10	0.7	5	6	1	0
35	10	0.7	5	6.5	1	0
35	10	0.7	5	7	1	0
35	10	0.7	5	7.5	1	0
35	10	0.7	5	8	1	0
35	10	0.7	5	8.5	1	0
35	10	0.7	5	9	1	0
35	10	0.7	5	9.5	1	0
35	10	0.7	5	10	1	0
36	10	0.8	5	5.5	1	0
36	10	0.8	5	6	1	0
36	10	0.8	5	6.5	1	0
36	10	0.8	5	7	1	0
36	10	0.8	5	7.5	1	0
36	10	0.8	5	8	1	0
36	10	0.8	5	8.5	1	0
36	10	0.8	5	9	1	0
36	10	0.8	5	9.5	1	0
36	10	0.8	5	10	1	0
37	15	0.2	5	6	1	0
37	15	0.2	5	7	1	0
37	15	0.2	5	8	1	0
37	15	0.2	5	9	1	0

37	15	0.2	5	10	1	0
37	15	0.2	5	11	1	0
37	15	0.2	5	12	1	0
37	15	0.2	5	13	1	0
37	15	0.2	5	14	1	0
37	15	0.2	5	15	1	0
38	15	0.3	5	6	1	0
38	15	0.3	5	7	1	0
38	15	0.3	5	8	1	0
38	15	0.3	5	9	1	0
38	15	0.3	5	10	1	0
38	15	0.3	5	11	1	0
38	15	0.3	5	12	1	0
38	15	0.3	5	13	1	0
38	15	0.3	5	14	1	0
38	15	0.3	5	15	1	0
39	15	0.4	5	6	1	0
39	15	0.4	5	7	1	0
39	15	0.4	5	8	1	0
39	15	0.4	5	9	1	0
39	15	0.4	5	10	1	0
39	15	0.4	5	11	1	0
39	15	0.4	5	12	1	0
39	15	0.4	5	13	1	0
39	15	0.4	5	14	1	0
39	15	0.4	5	15	1	0
40	15	0.6	5	6	1	0
40	15	0.6	5	7	1	0
40	15	0.6	5	8	1	0
40	15	0.6	5	9	1	0
40	15	0.6	5	10	1	0
40	15	0.6	5	11	1	0
40	15	0.6	5	12	1	0
40	15	0.6	5	13	1	0
40	15	0.6	5	14	1	0
40	15	0.6	5	15	1	0
41	15	0.7	5	6	1	0
41	15	0.7	5	7	1	0
41	15	0.7	5	8	1	0
41	15	0.7	5	9	1	0
41	15	0.7	5	10	1	0

41	15	0.7	5	11	1	0
41	15	0.7	5	12	1	0
41	15	0.7	5	13	1	0
41	15	0.7	5	14	1	0
41	15	0.7	5	15	1	0
42	15	0.8	5	6	1	0
42	15	0.8	5	7	1	0
42	15	0.8	5	8	1	0
42	15	0.8	5	9	1	0
42	15	0.8	5	10	1	0
42	15	0.8	5	11	1	0
42	15	0.8	5	12	1	0
42	15	0.8	5	13	1	0
42	15	0.8	5	14	1	0
42	15	0.8	5	15	1	0
43	15	0.2	10	10.5	1	0
43	15	0.2	10	11	1	0
43	15	0.2	10	11.5	1	0
43	15	0.2	10	12	1	0
43	15	0.2	10	12.5	1	0
43	15	0.2	10	13	1	0
43	15	0.2	10	13.5	1	0
43	15	0.2	10	14	1	0
43	15	0.2	10	14.5	1	0
43	15	0.2	10	15	1	0
44	15	0.3	10	10.5	1	0
44	15	0.3	10	11	1	0
44	15	0.3	10	11.5	1	0
44	15	0.3	10	12	1	0
44	15	0.3	10	12.5	1	0
44	15	0.3	10	13	1	0
44	15	0.3	10	13.5	1	0
44	15	0.3	10	14	1	0
44	15	0.3	10	14.5	1	0
44	15	0.3	10	15	1	0
45	15	0.4	10	10.5	1	0
45	15	0.4	10	11	1	0
45	15	0.4	10	11.5	1	0
45	15	0.4	10	12	1	0
45	15	0.4	10	12.5	1	0
45	15	0.4	10	13	1	0

45	15	0.4	10	13.5	1	0
45	15	0.4	10	14	1	0
45	15	0.4	10	14.5	1	0
45	15	0.4	10	15	1	0
46	15	0.6	10	10.5	1	0
46	15	0.6	10	11	1	0
46	15	0.6	10	11.5	1	0
46	15	0.6	10	12	1	0
46	15	0.6	10	12.5	1	0
46	15	0.6	10	13	1	0
46	15	0.6	10	13.5	1	0
46	15	0.6	10	14	1	0
46	15	0.6	10	14.5	1	0
46	15	0.6	10	15	1	0
47	15	0.7	10	10.5	1	0
47	15	0.7	10	11	1	0
47	15	0.7	10	11.5	1	0
47	15	0.7	10	12	1	0
47	15	0.7	10	12.5	1	0
47	15	0.7	10	13	1	0
47	15	0.7	10	13.5	1	0
47	15	0.7	10	14	1	0
47	15	0.7	10	14.5	1	0
47	15	0.7	10	15	1	0
48	15	0.8	10	10.5	1	0
48	15	0.8	10	11	1	0
48	15	0.8	10	11.5	1	0
48	15	0.8	10	12	1	0
48	15	0.8	10	12.5	1	0
48	15	0.8	10	13	1	0
48	15	0.8	10	13.5	1	0
48	15	0.8	10	14	1	0
48	15	0.8	10	14.5	1	0
48	15	0.8	10	15	1	0

Key:

ListNo: List Number

AmountXL: the first outcome of the left lottery

ProbXL: the probability of the first outcome of the left lottery

AmountYL: the second outcome of the left lottery

AmountXR: the first outcome of the right lottery

ProbXR: the probability of the first outcome of the right lottery

AmountYR: the second outcome of the right lottery

Appendix D.

Estimates of Parameters in Chapter 2

(Full Set, without Outliers)

Figure D.1: Estimates of r Using AL across Preference Functionals

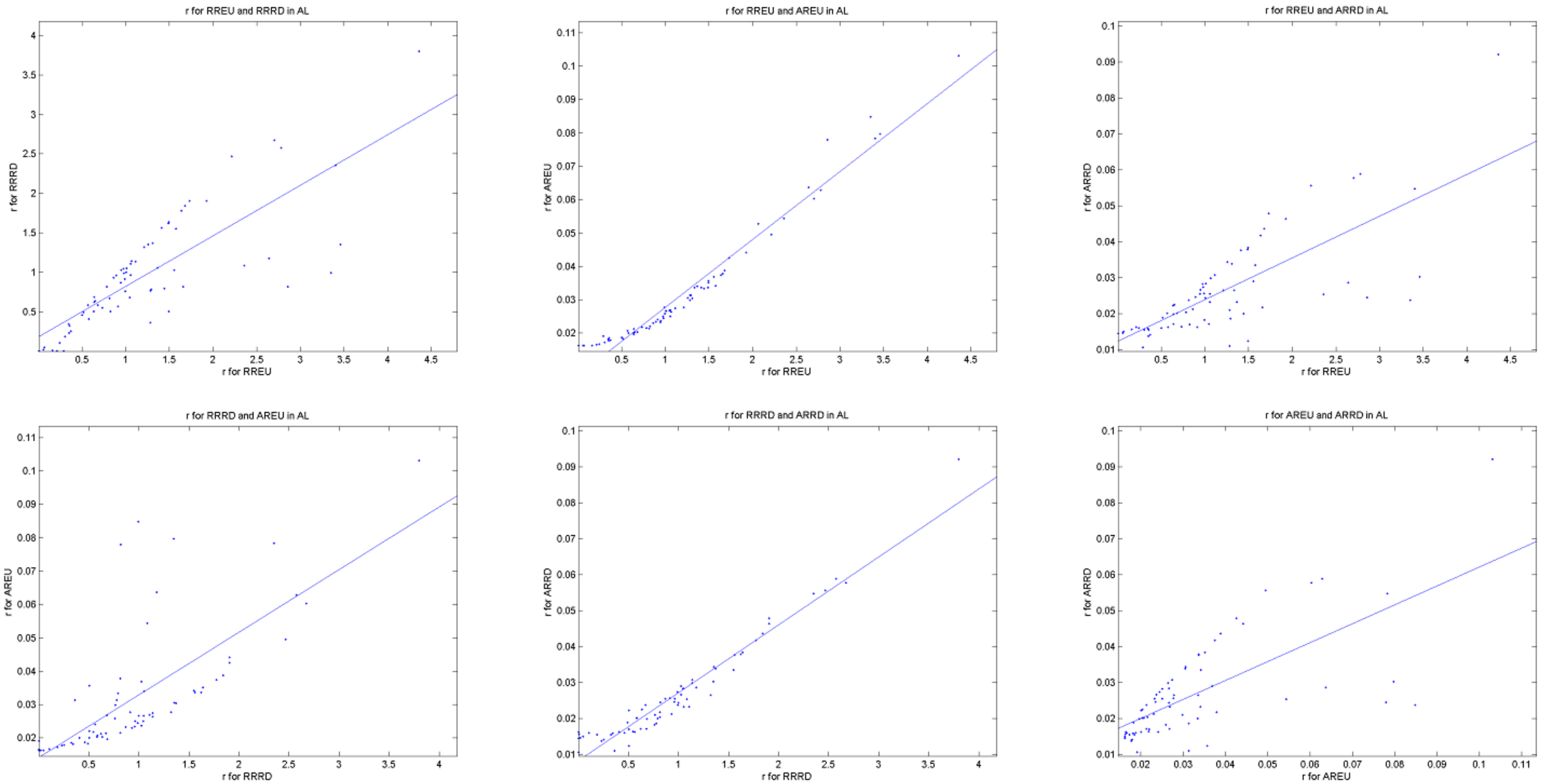


Figure D.2: Estimates of r Using LC across Preference Functionals

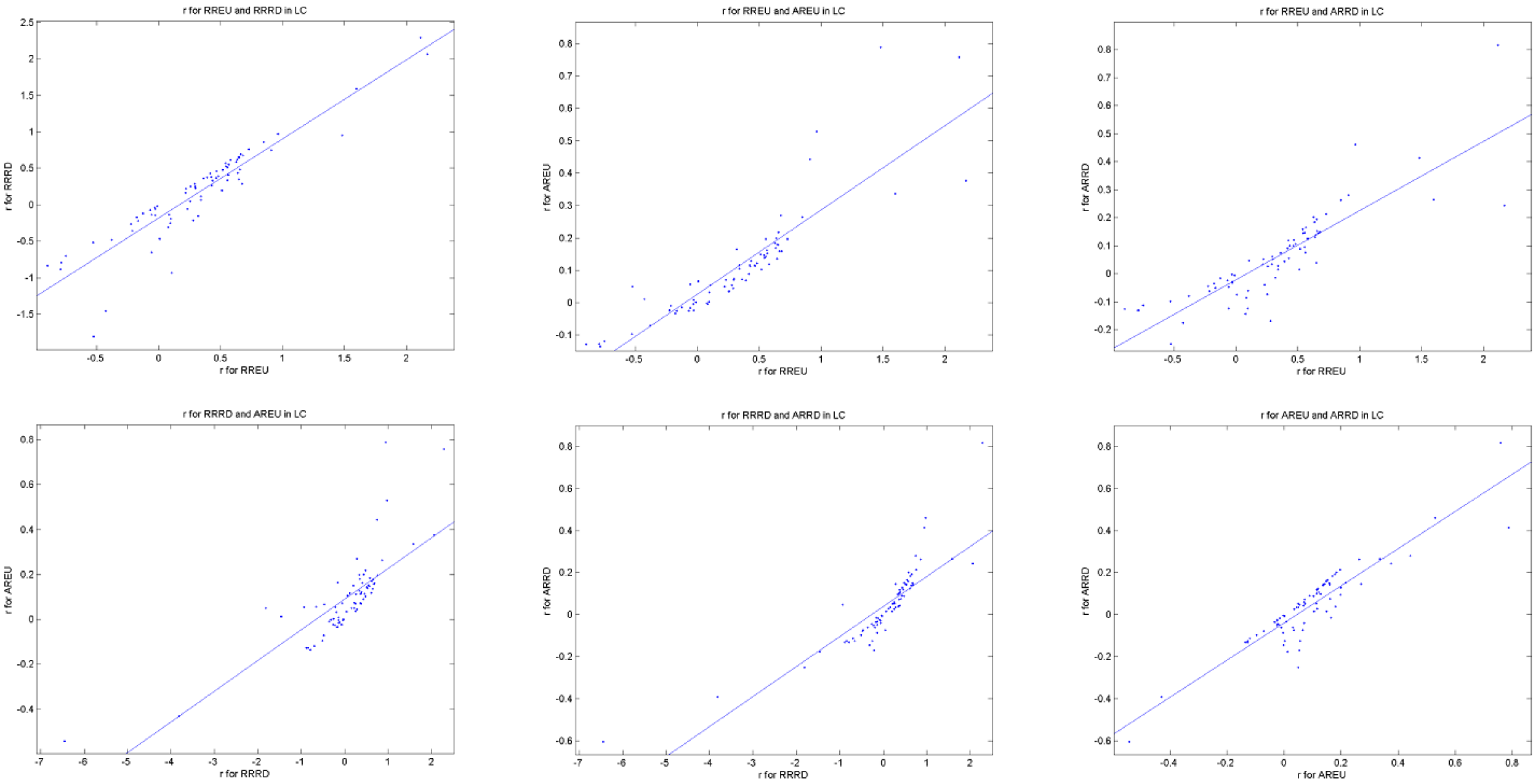


Figure D.3: Estimates of r Using PC across Preference Functionals

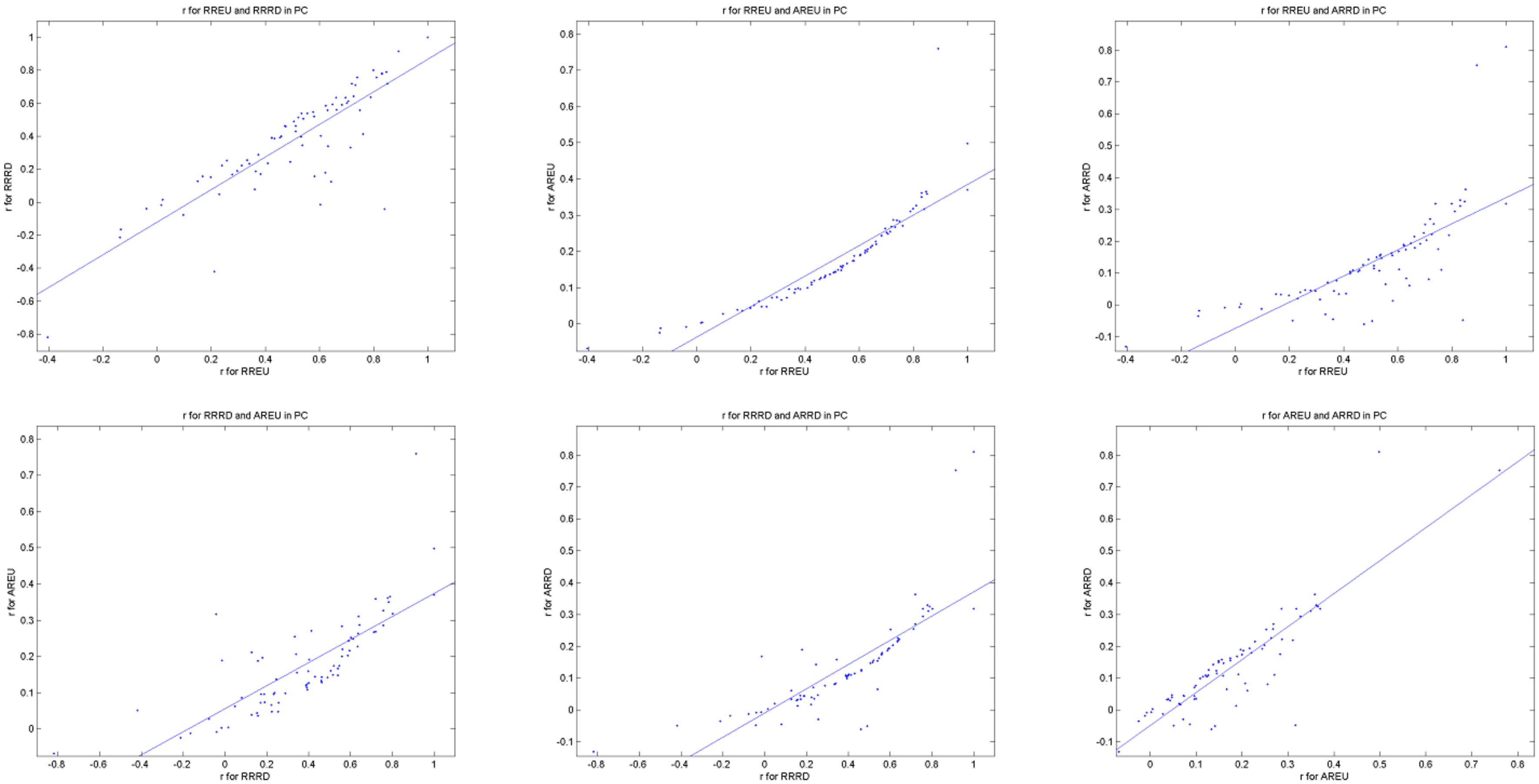


Figure D.4: Estimates of r Using HL across Preference Functionals

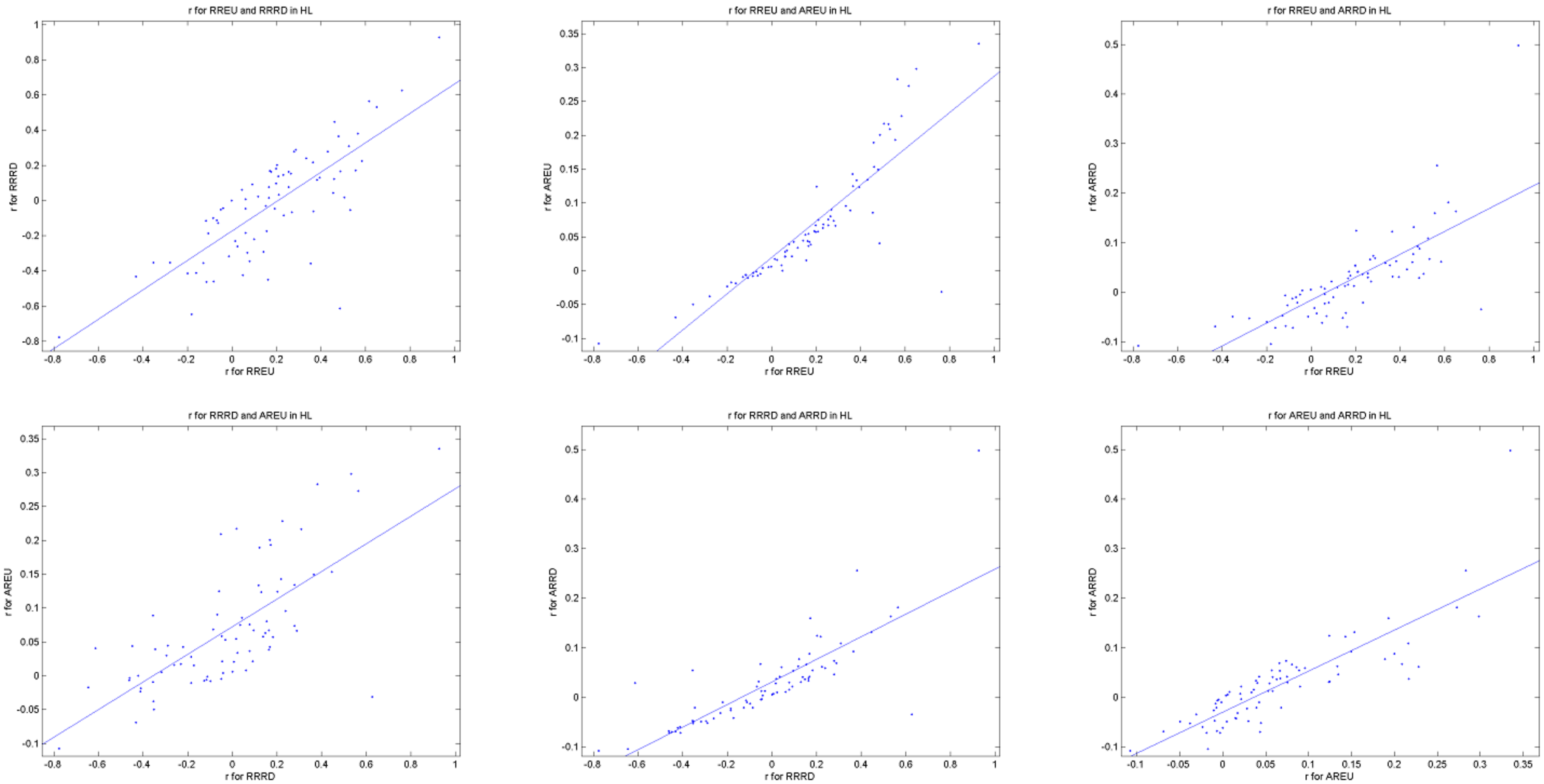


Figure D.5: Estimates of s Using AL across Preference Functionals

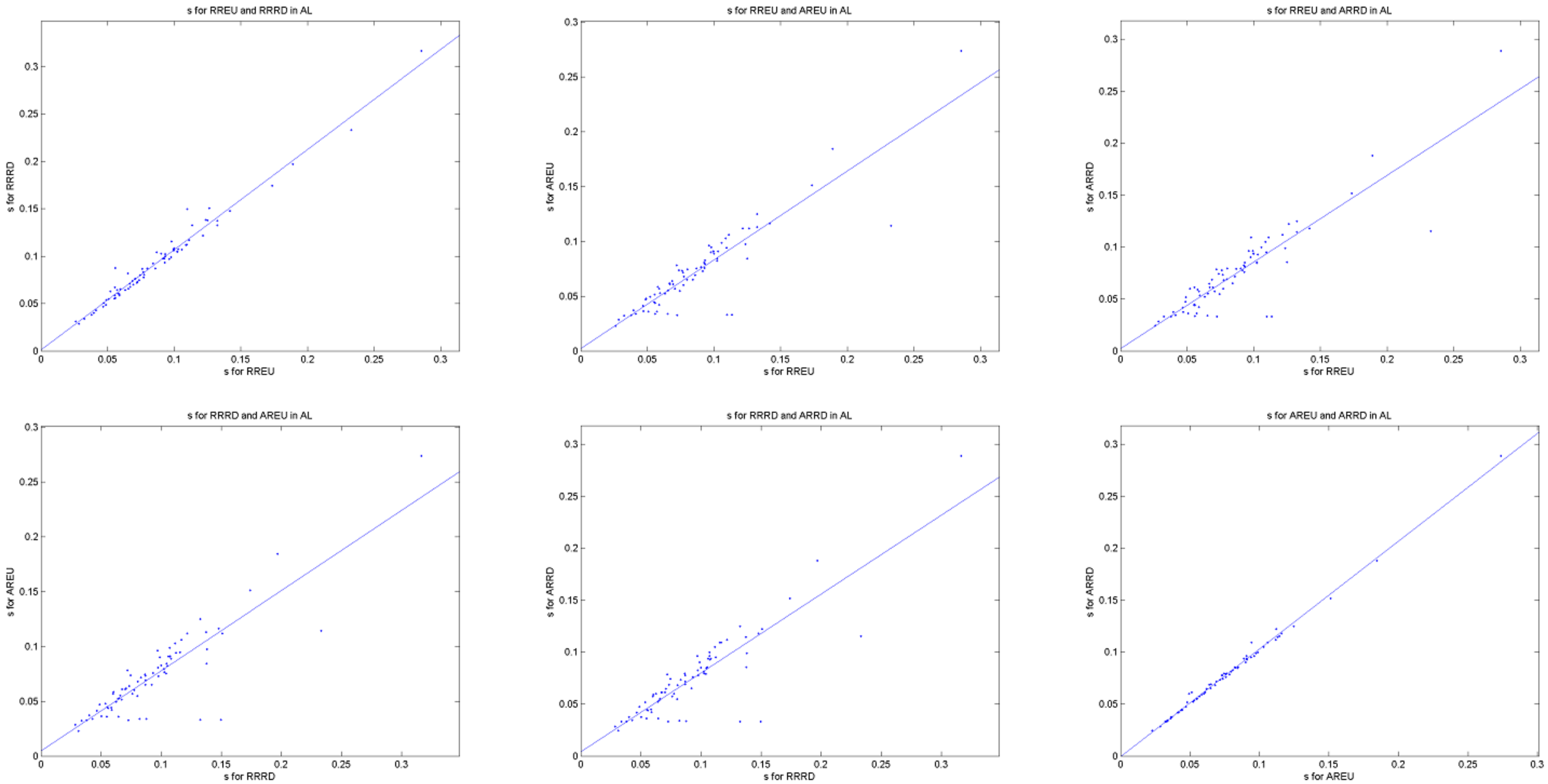


Figure D.6: Estimates of s Using LC across Preference Functionals

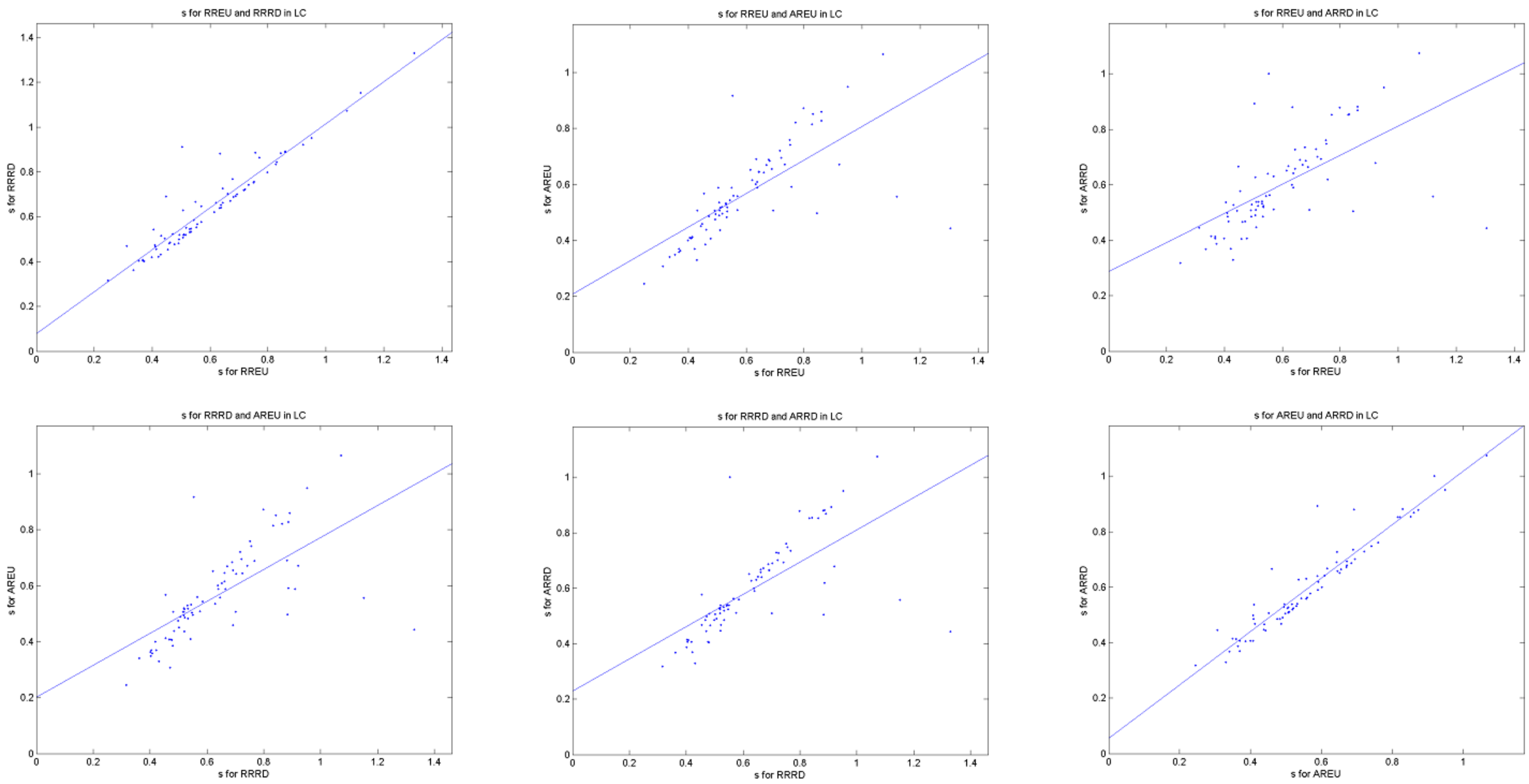


Figure D.7: Estimates of s Using PC across Preference Functionals

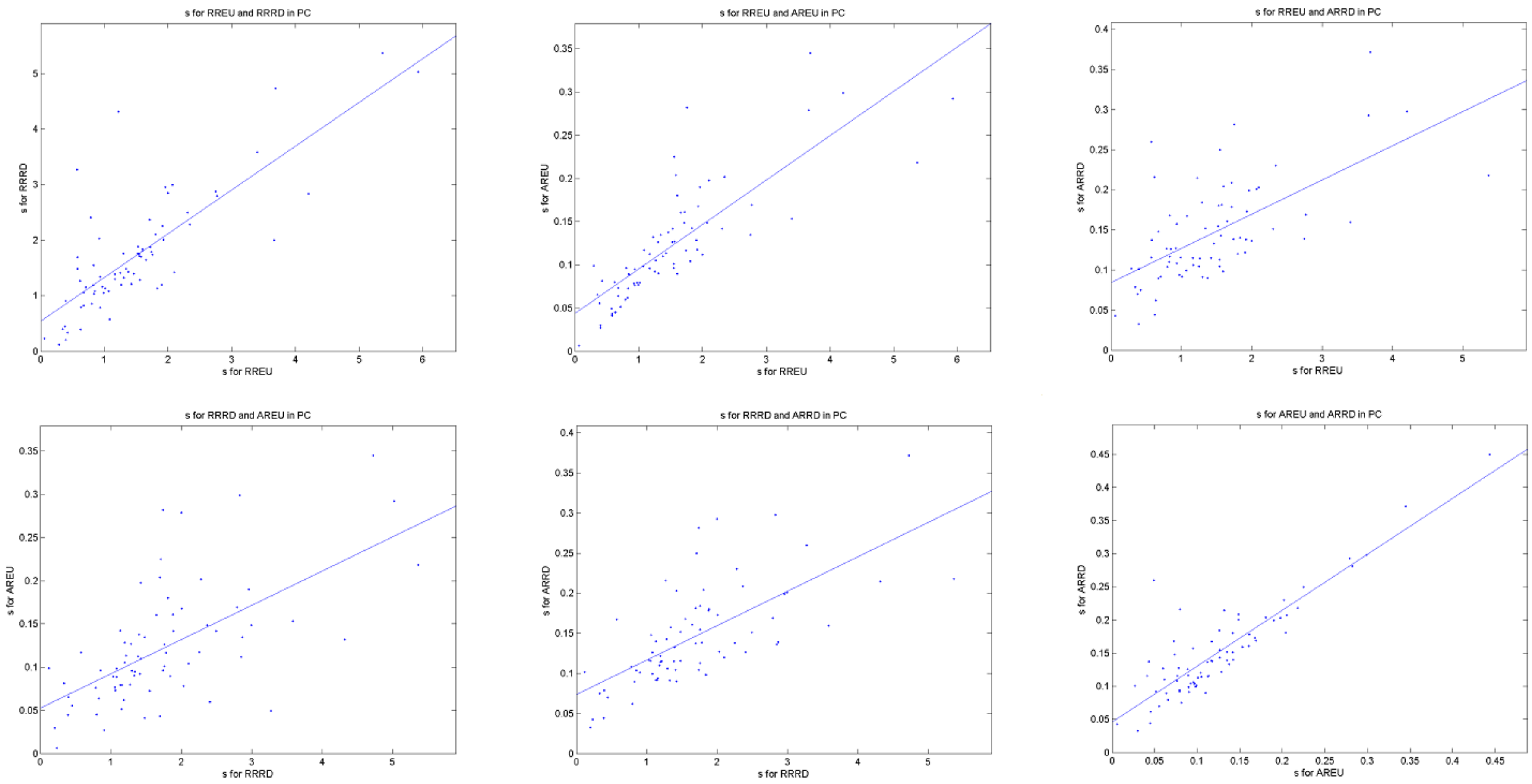


Figure D.8: Estimates of s Using HL across Preference Functionals

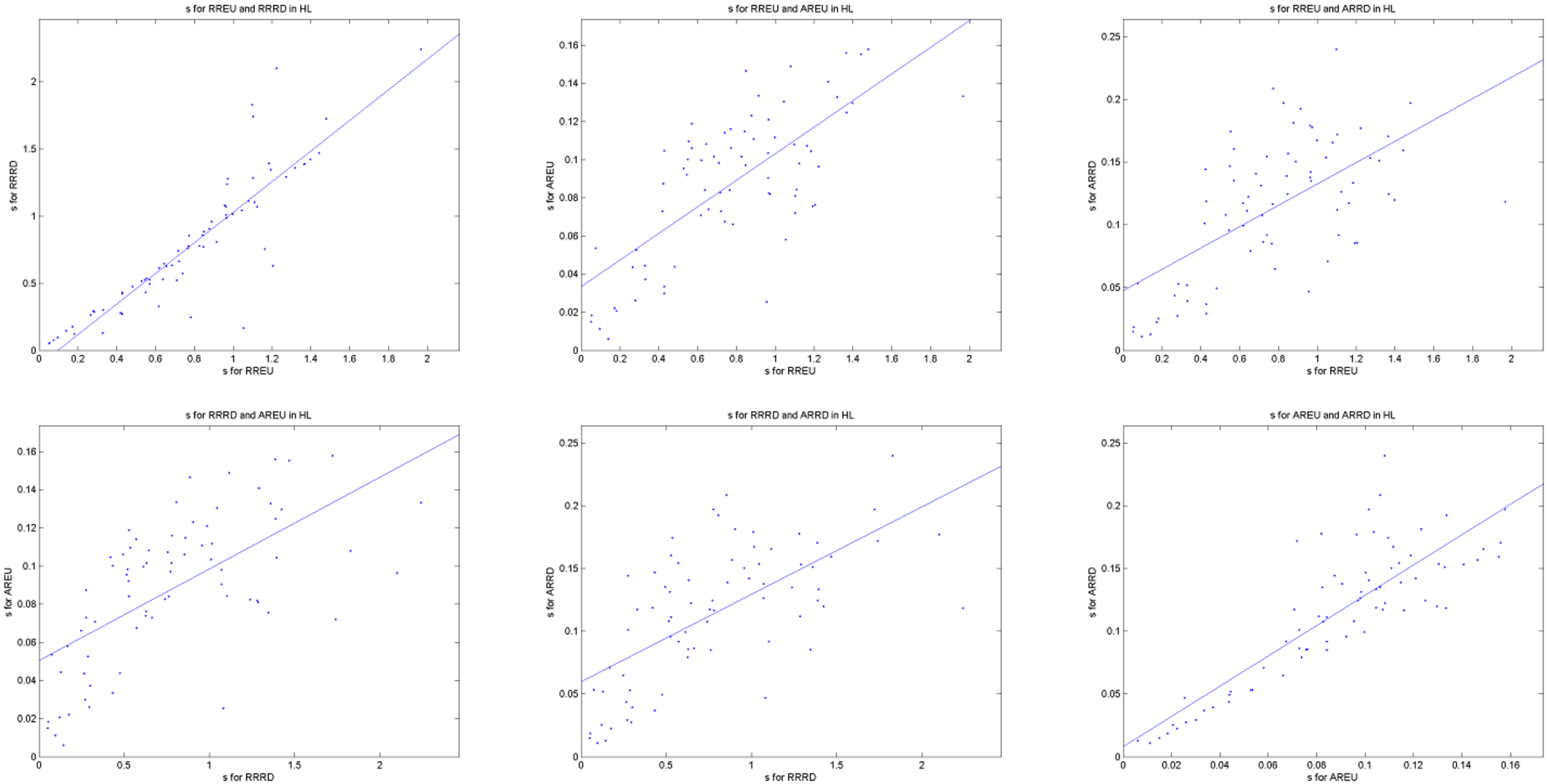


Figure D.9: Estimates of g across Preference Functionals

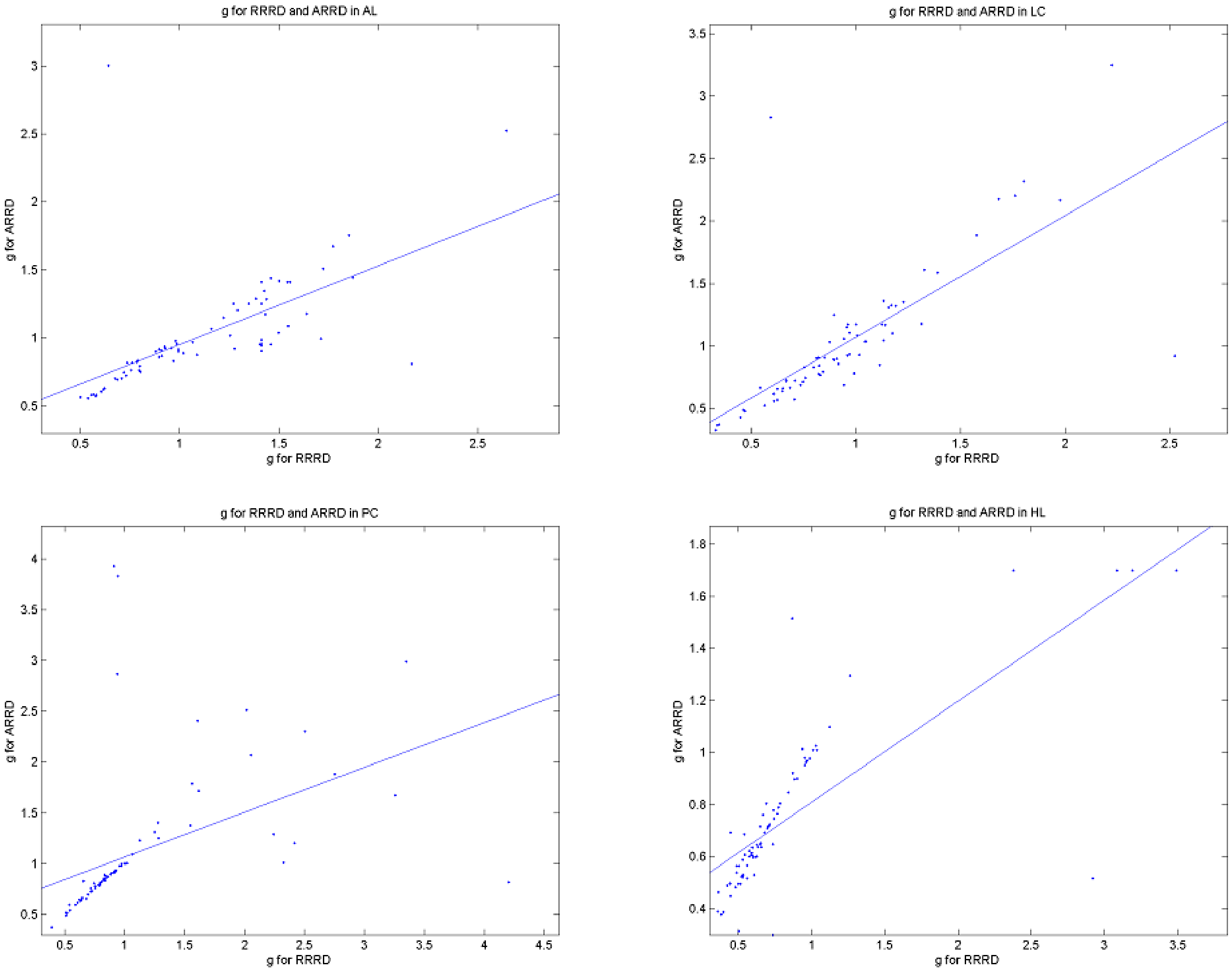


Figure D.10: Estimates of r in RREU across Elicitation Methods

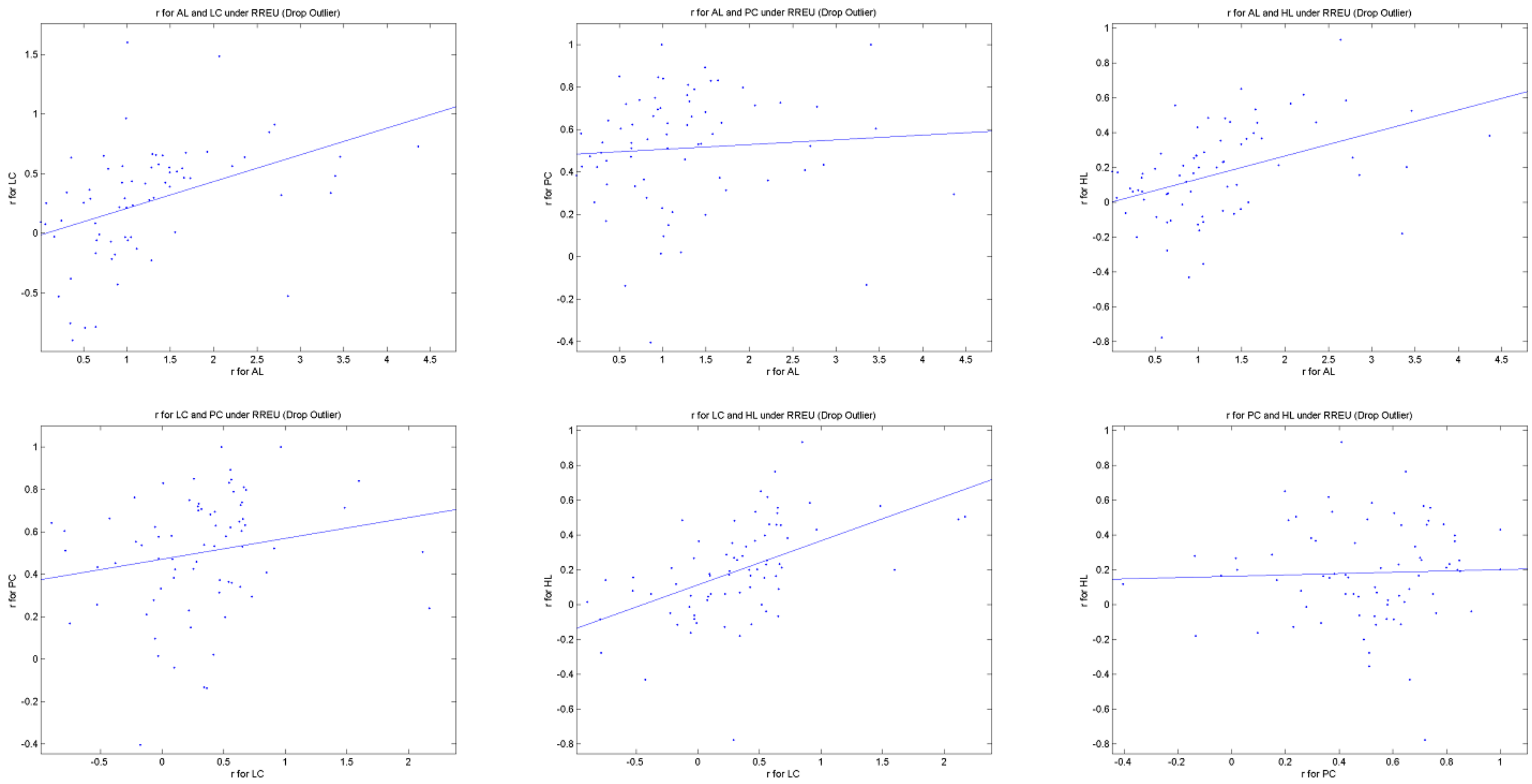


Figure D.11: Estimates of r in RRRD across Elicitation Methods

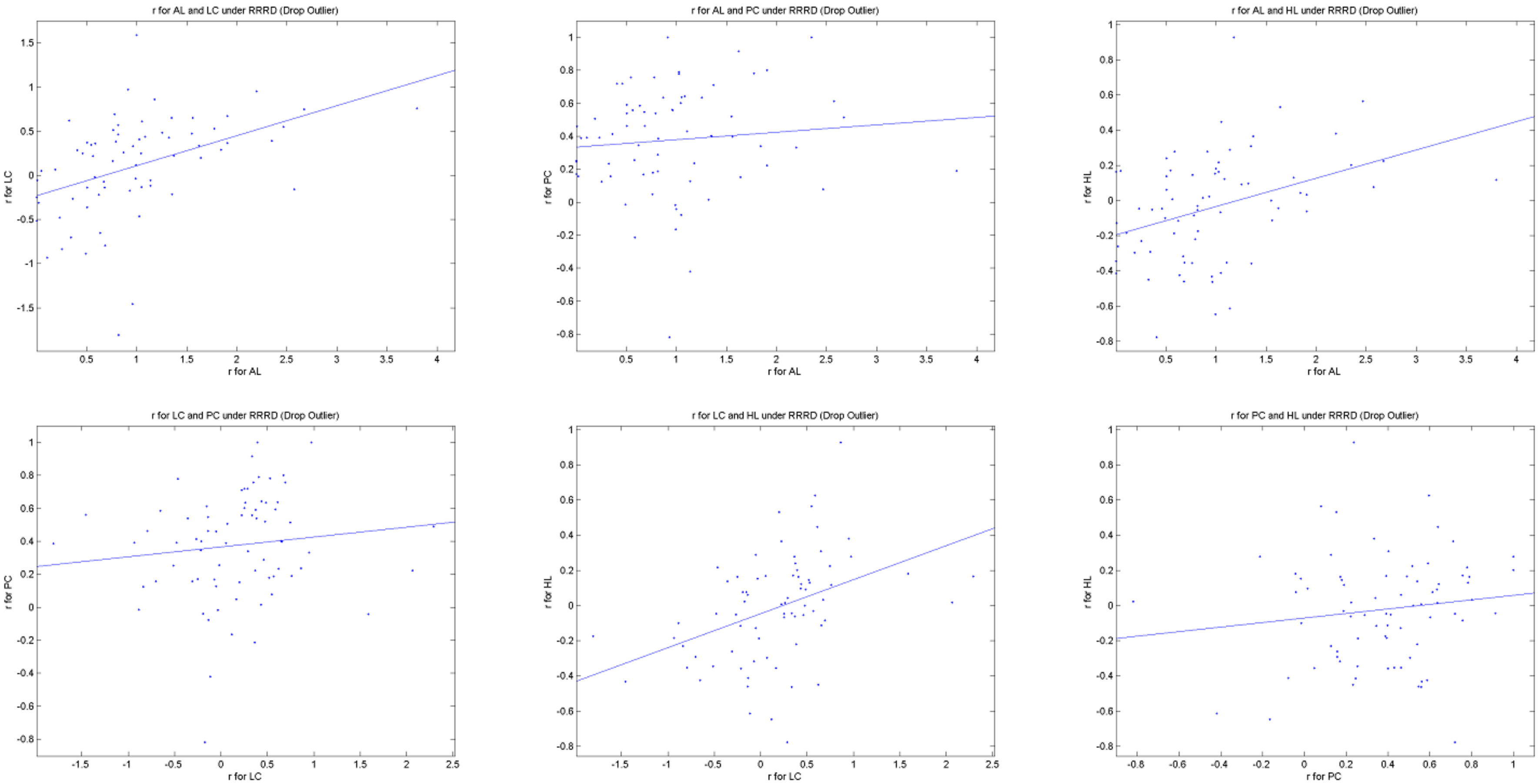


Figure D.12: Estimates of r in AREU across Elicitation Methods

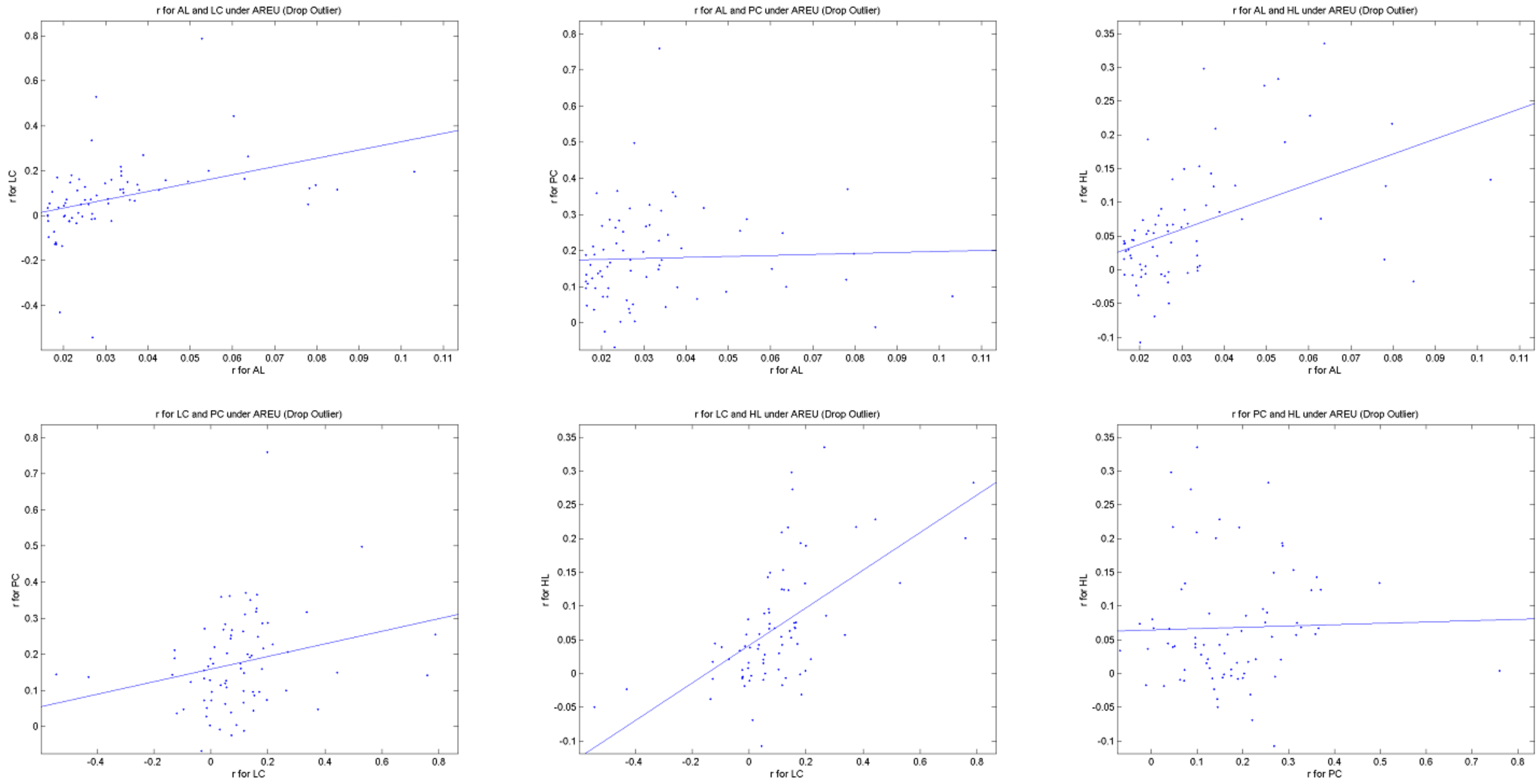


Figure D.13: Estimates of r in ARRD across Elicitation Methods

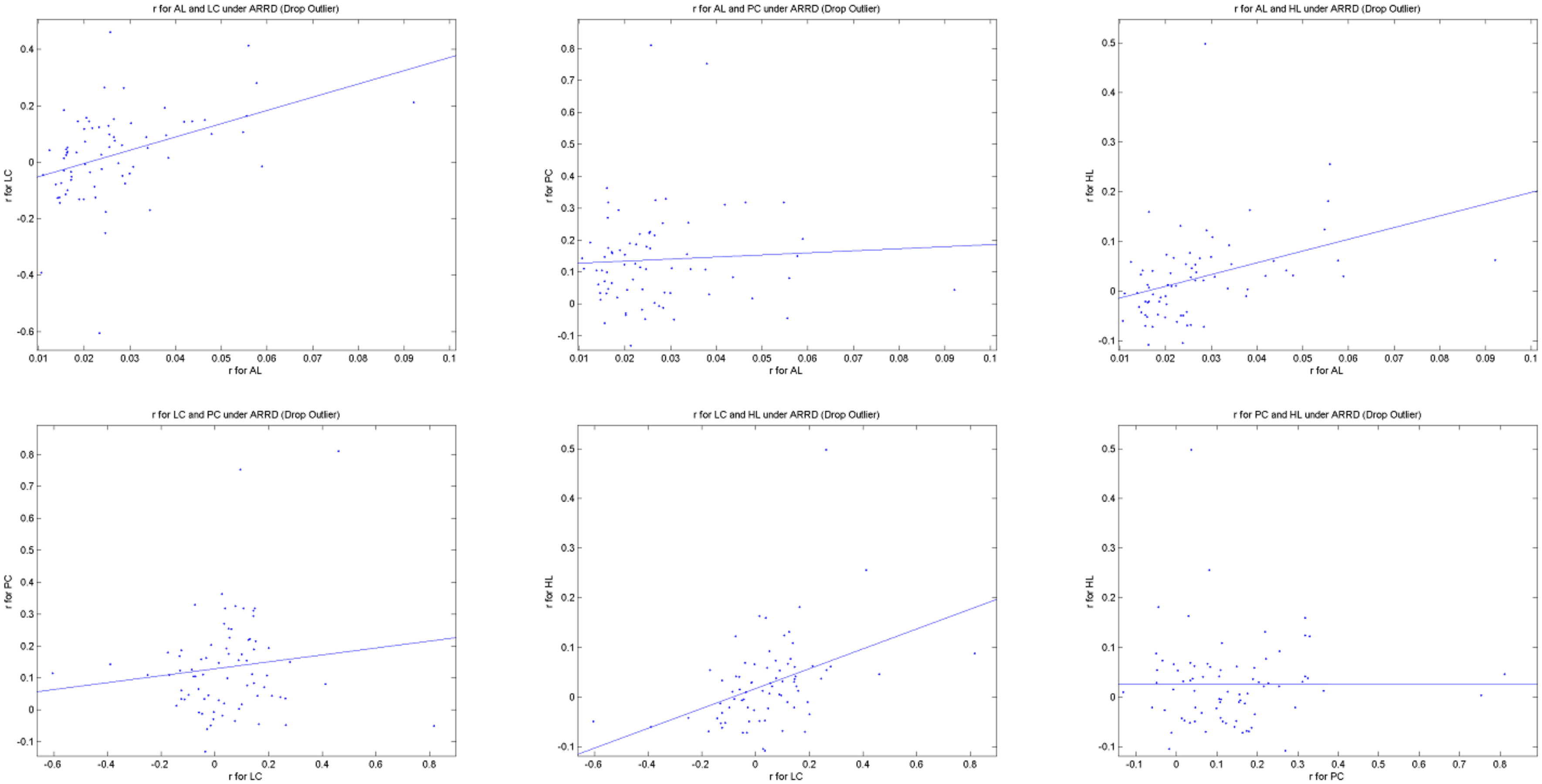


Figure D.14: Estimates of s in RREU across Elicitation Methods

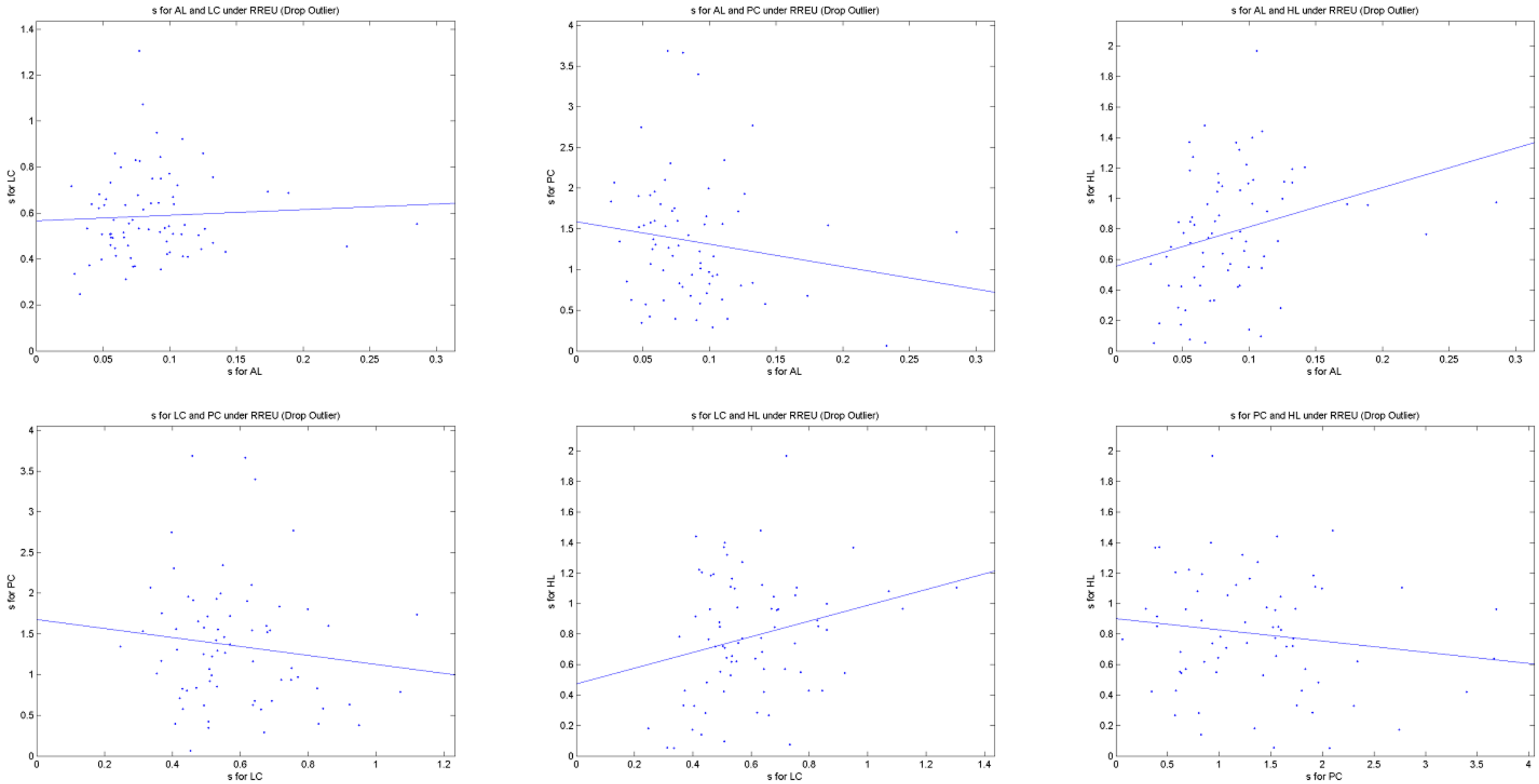


Figure D.15: Estimates of s in RRRD across Elicitation Methods

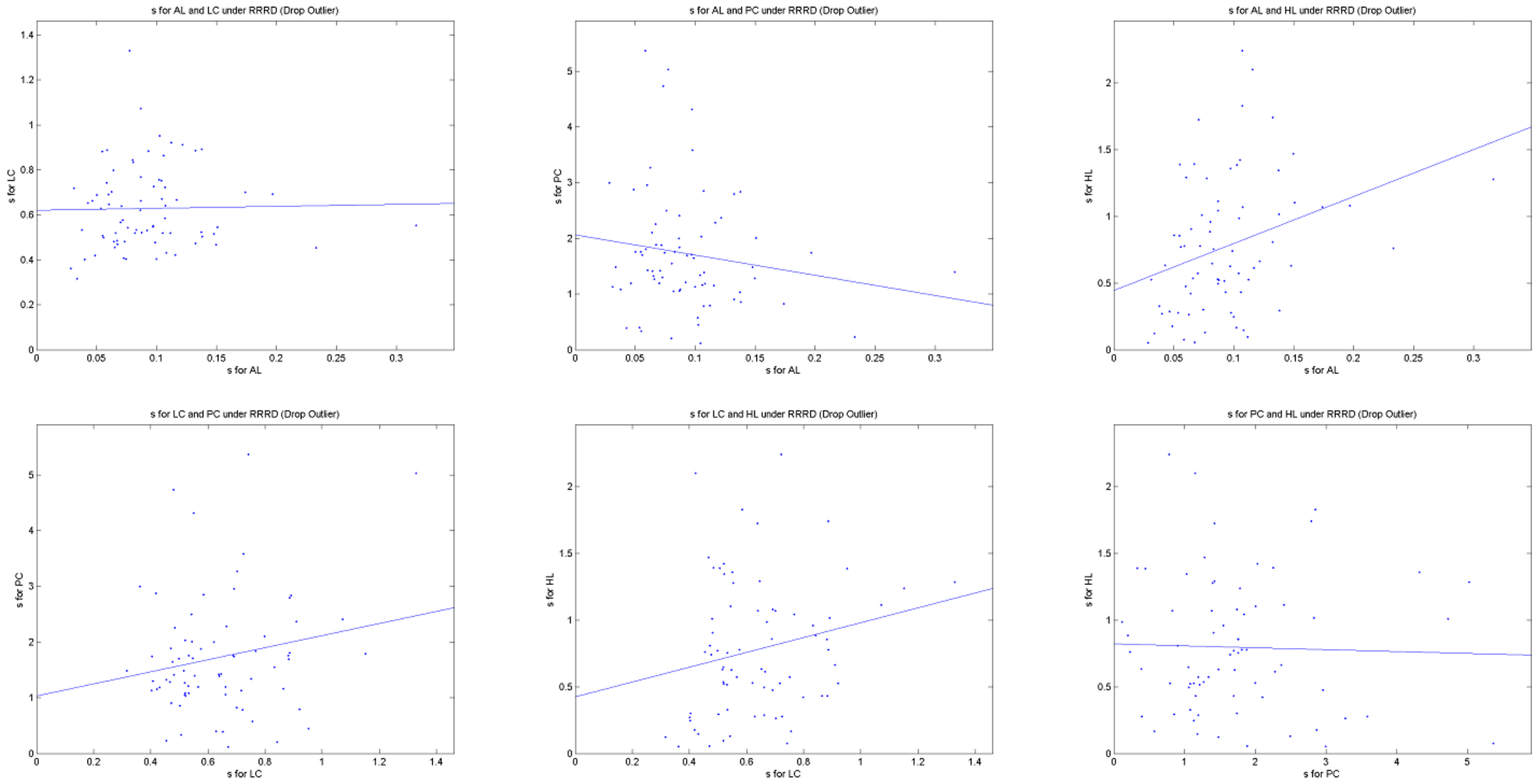


Figure D.16: Estimates of s in AREU across Elicitation Methods

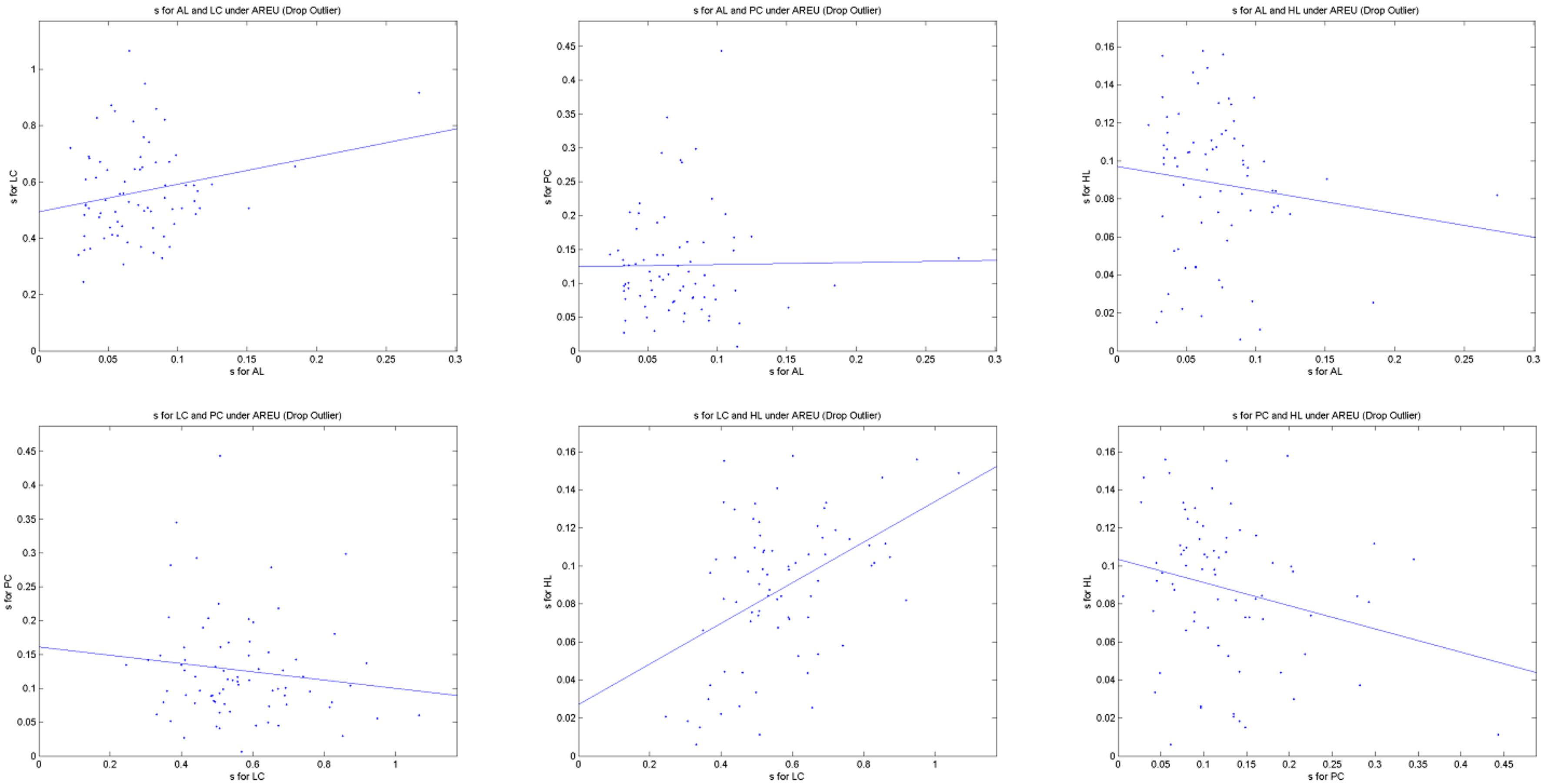


Figure D.17: Estimates of s in ARRD across Elicitation Methods

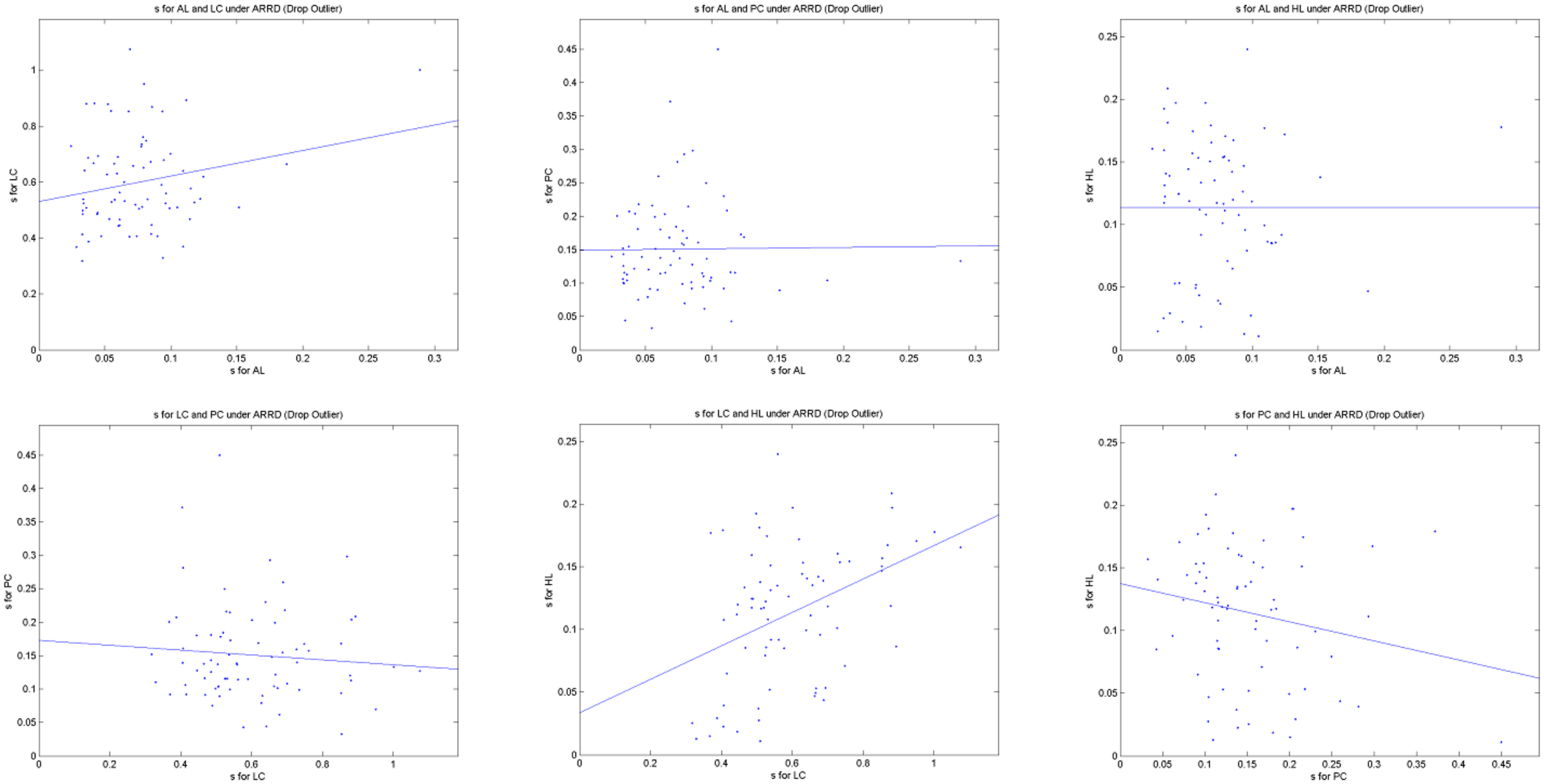


Figure D.18: Estimates of g in RRRD across Elicitation Methods

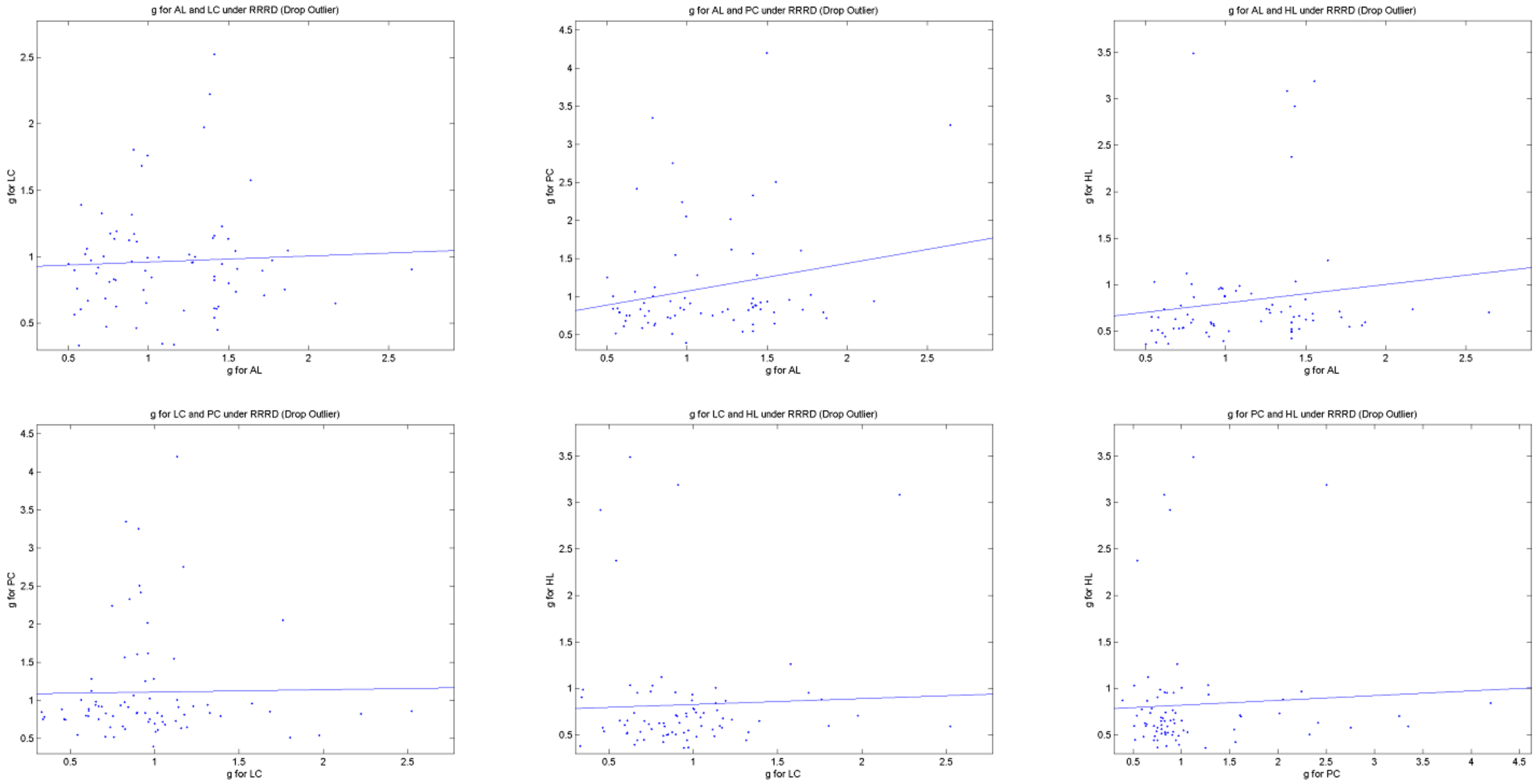
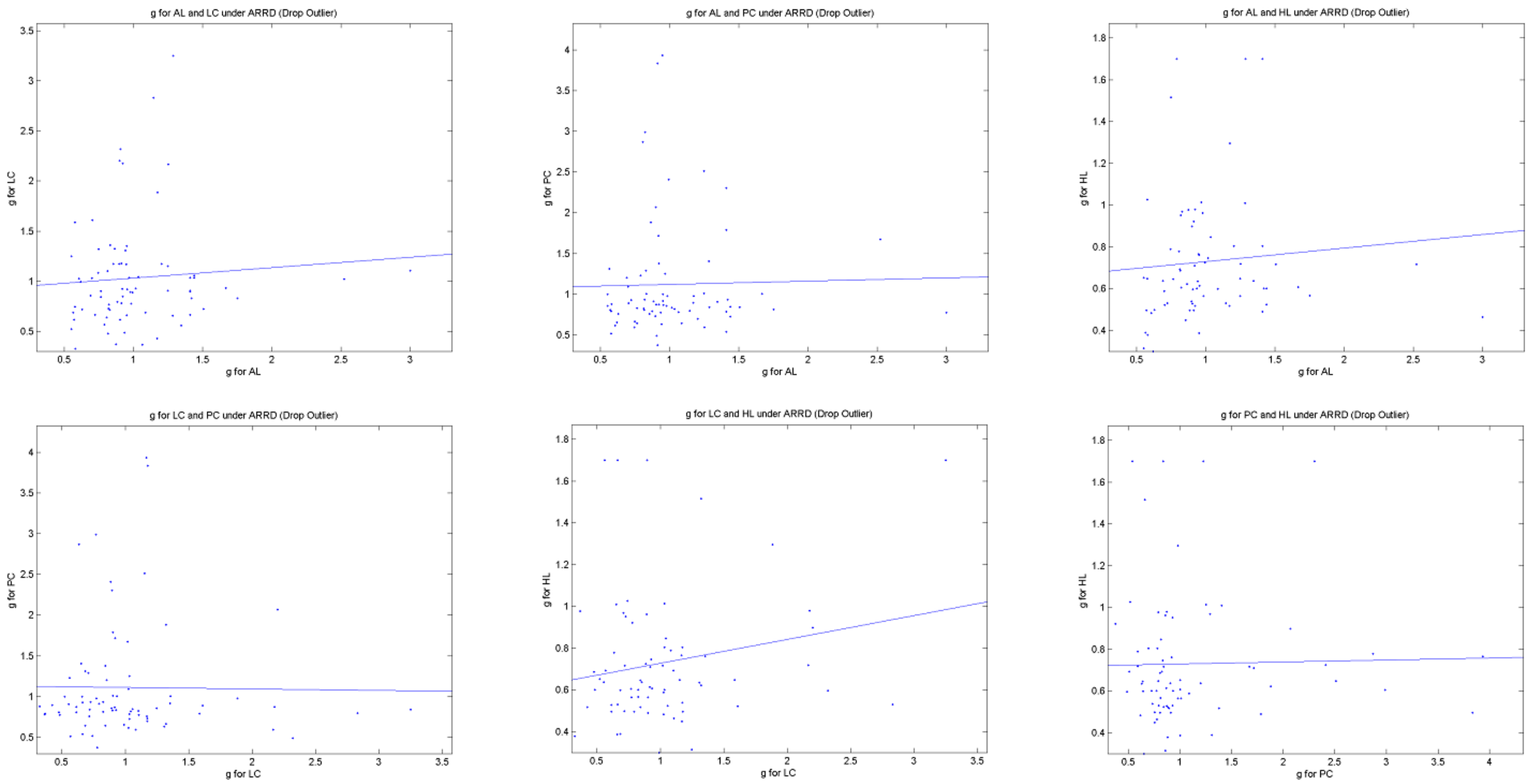


Figure D.19: Estimates of g in ARRD across Elicitation Methods



Appendix E.

Estimates of Parameters in Chapter 2

(Full Set, with Outliers)

Figure E.1: Estimates of r Using AL across Preference Functionals (with Outliers)

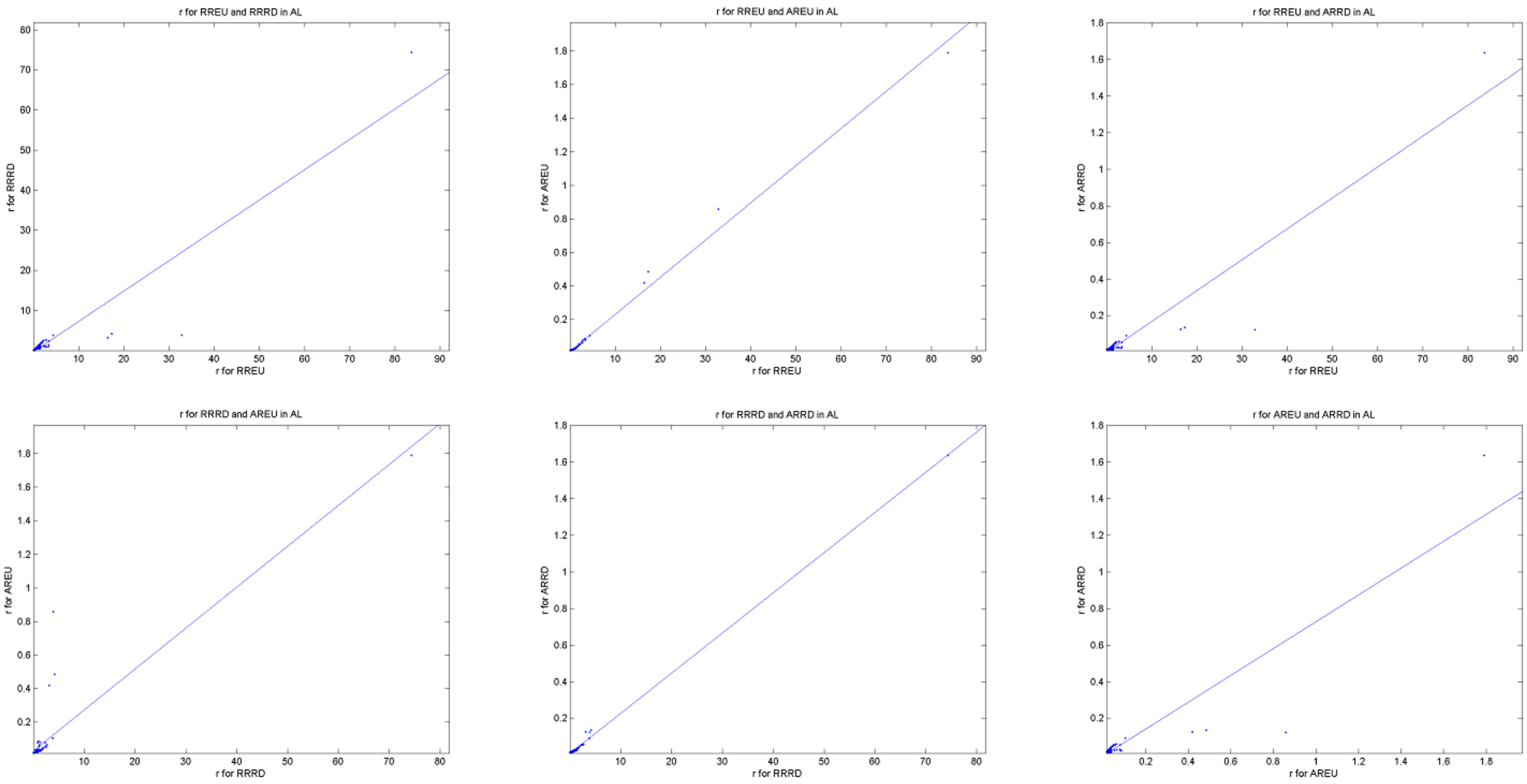


Figure E.2: Estimates of r Using LC across Preference Functionals (with Outliers)

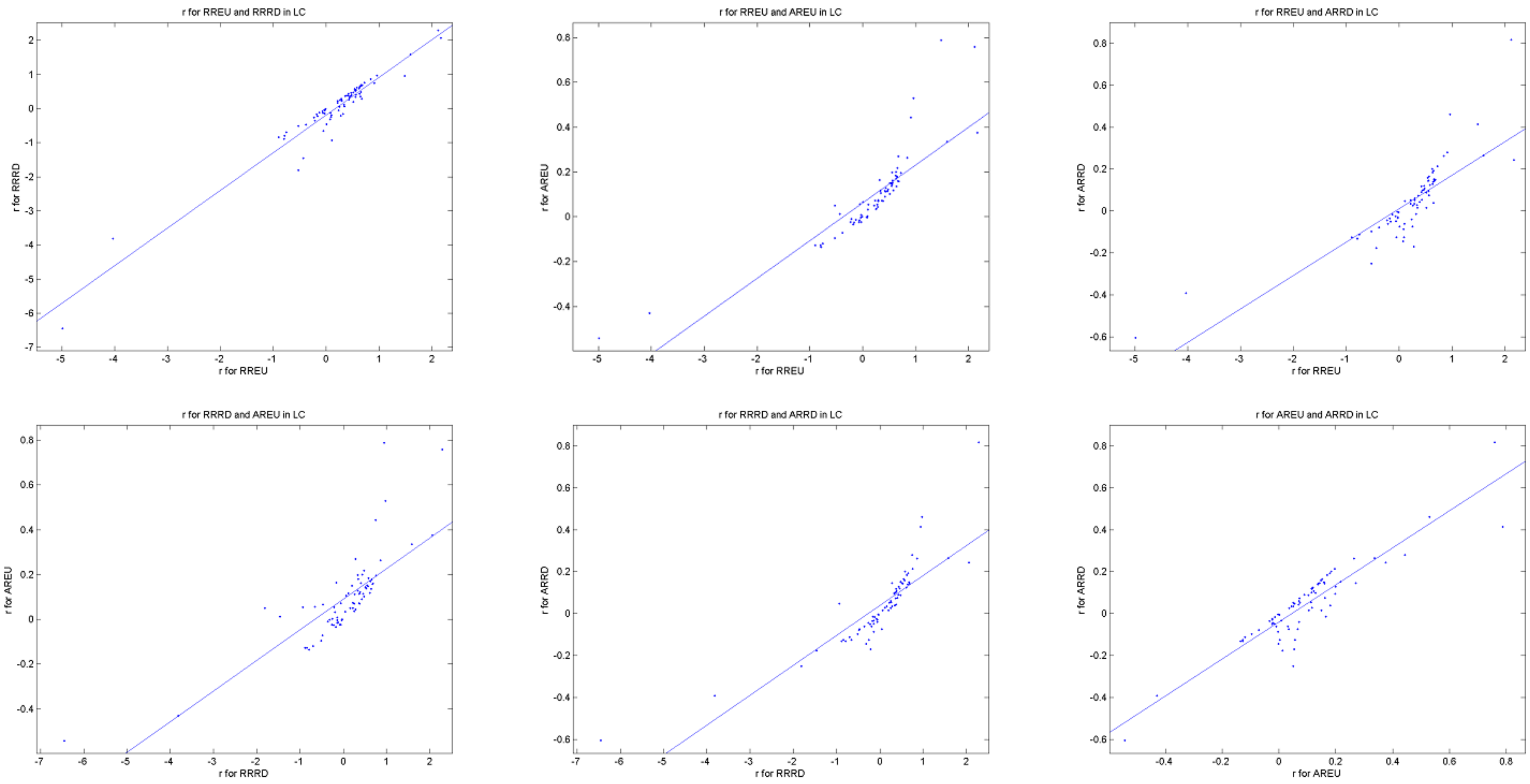


Figure E.3: Estimates of r Using PC across Preference Functionals (with Outliers)

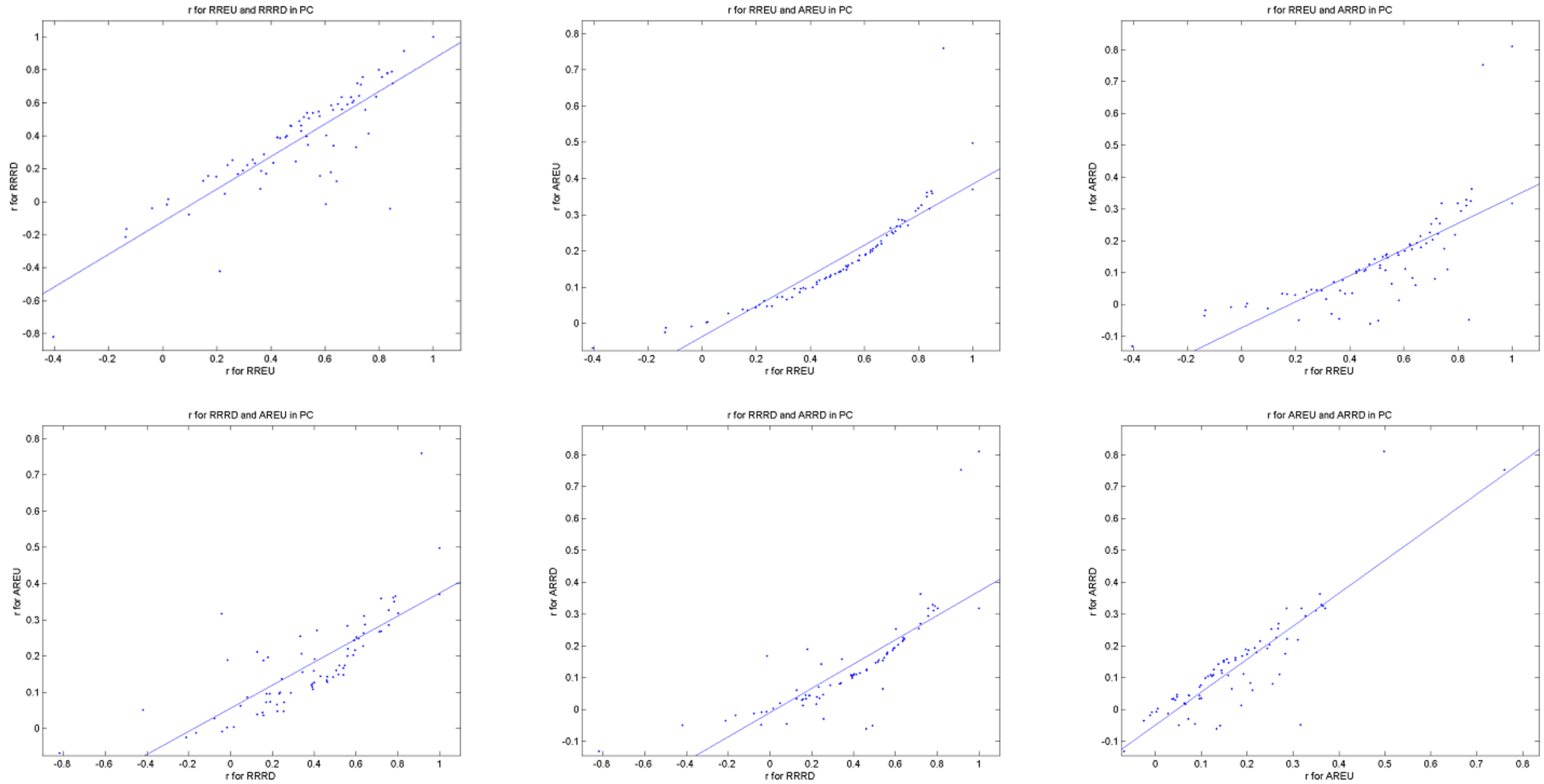


Figure E.4: Estimates of r Using HL across Preference Functionals (with Outliers)

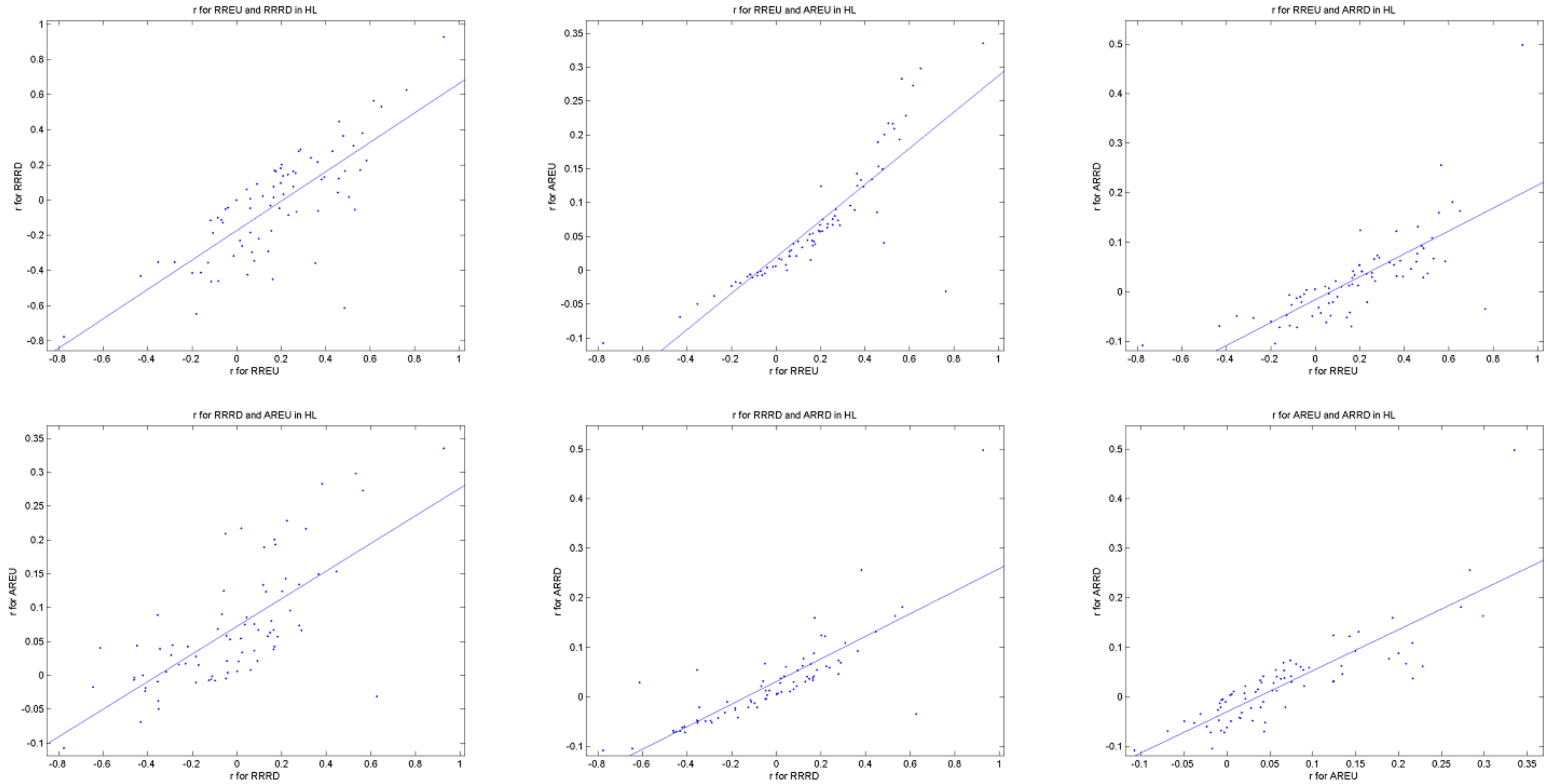


Figure E.5: Estimates of s Using AL across Preference Functionals (with Outliers)

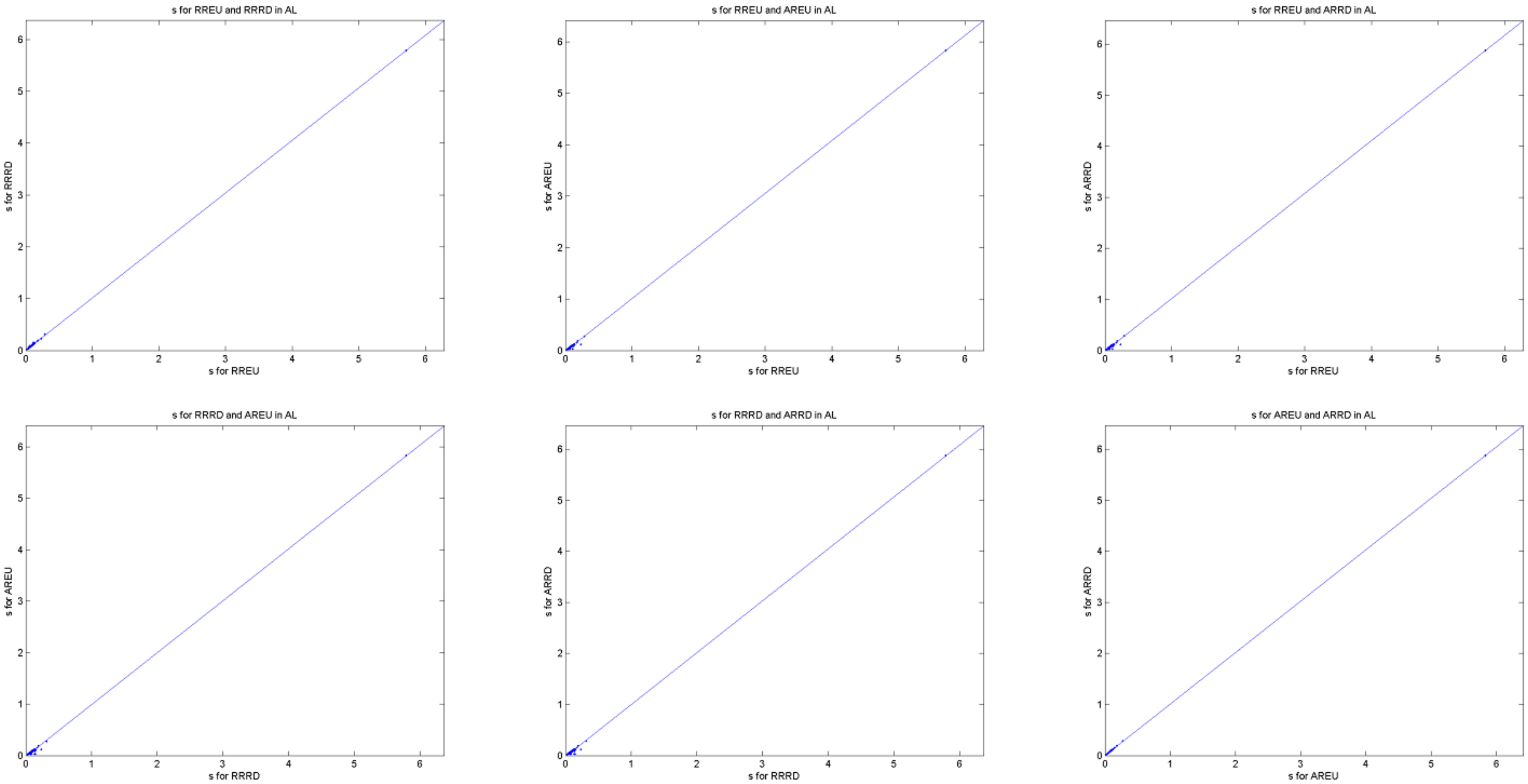


Figure E.6: Estimates of s Using LC across Preference Functionals (with Outliers)

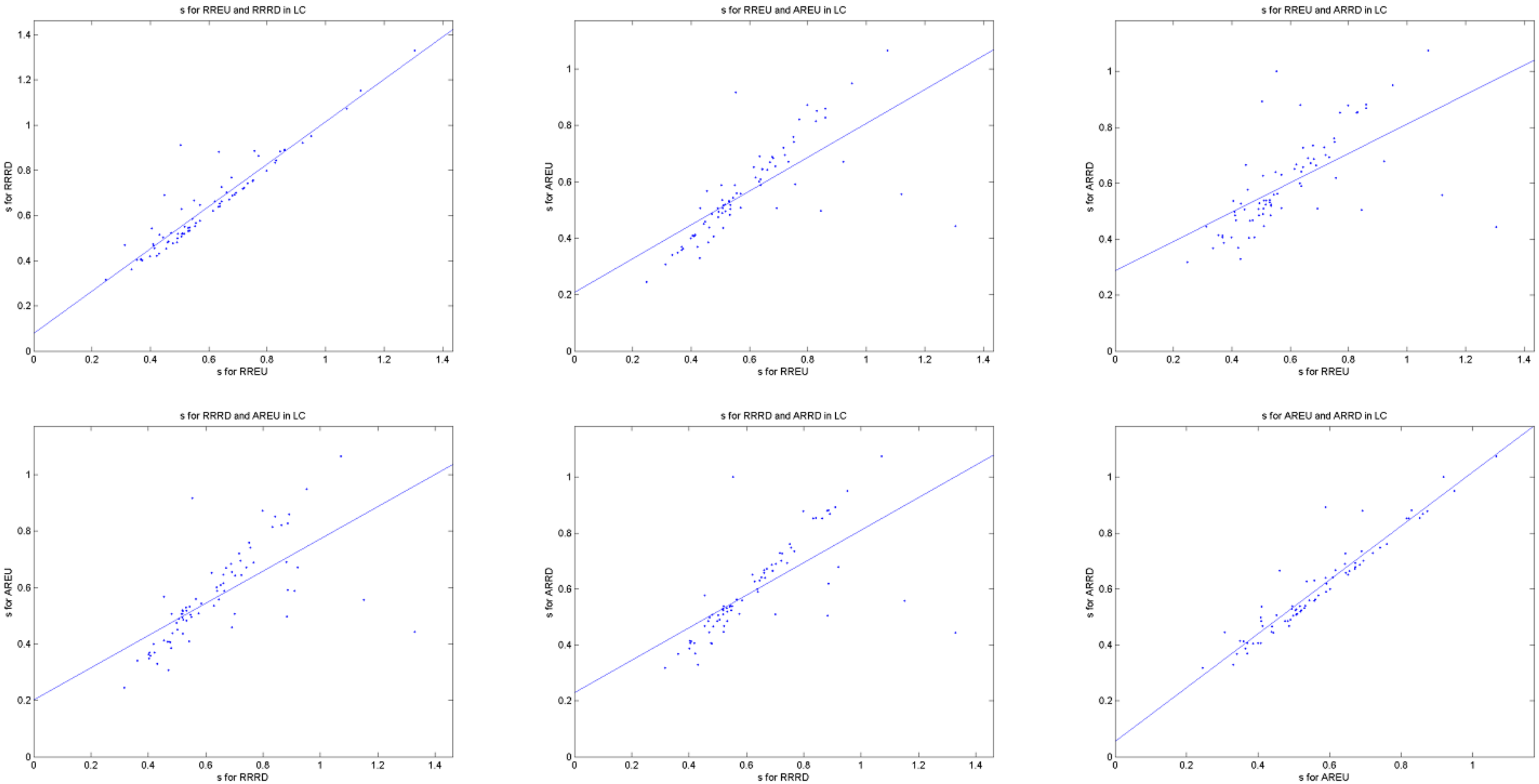


Figure E.7: Estimates of s Using PC across Preference Functionals (with Outliers)

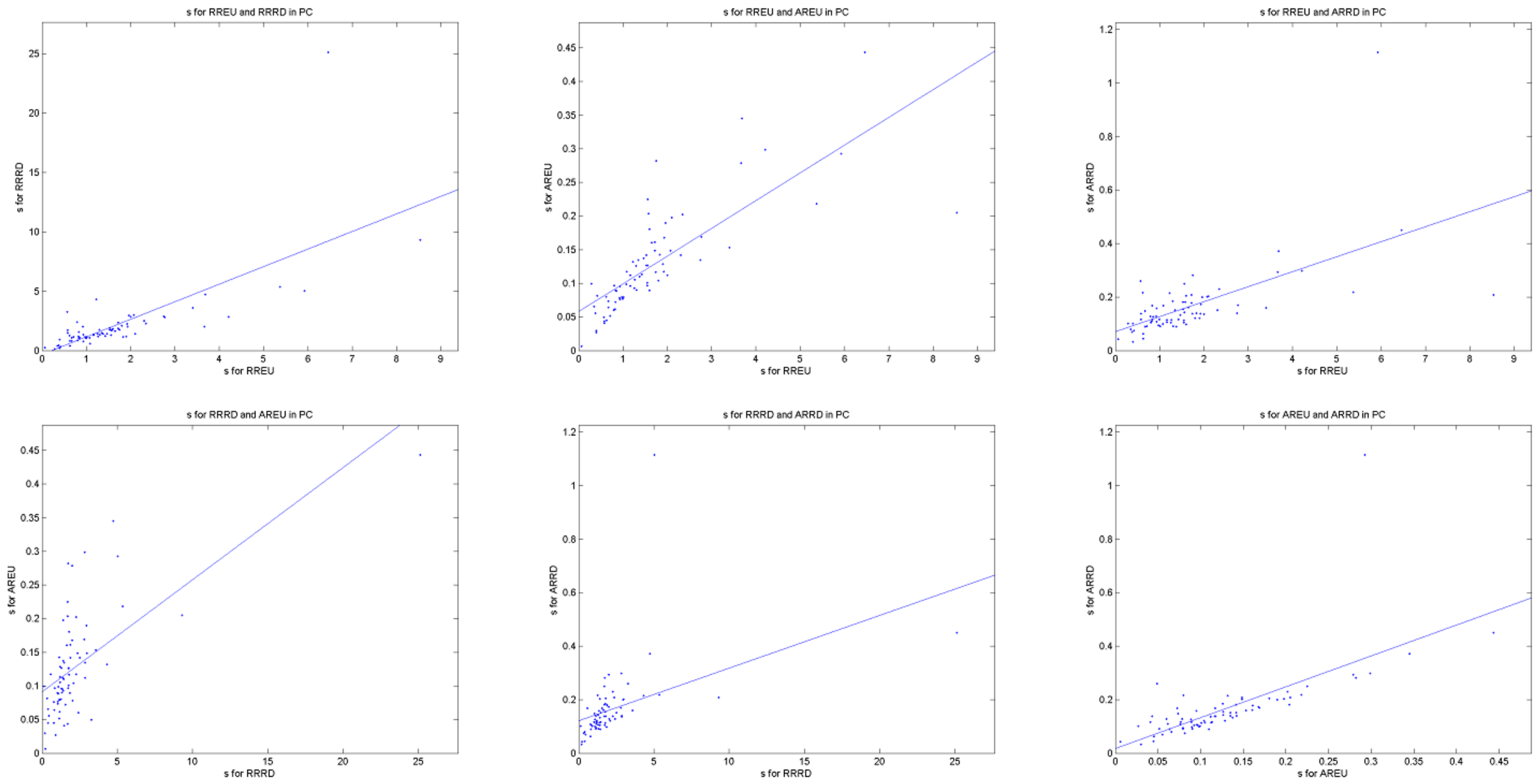


Figure E.8: Estimates of s Using HL across Preference Functionals (with Outliers)

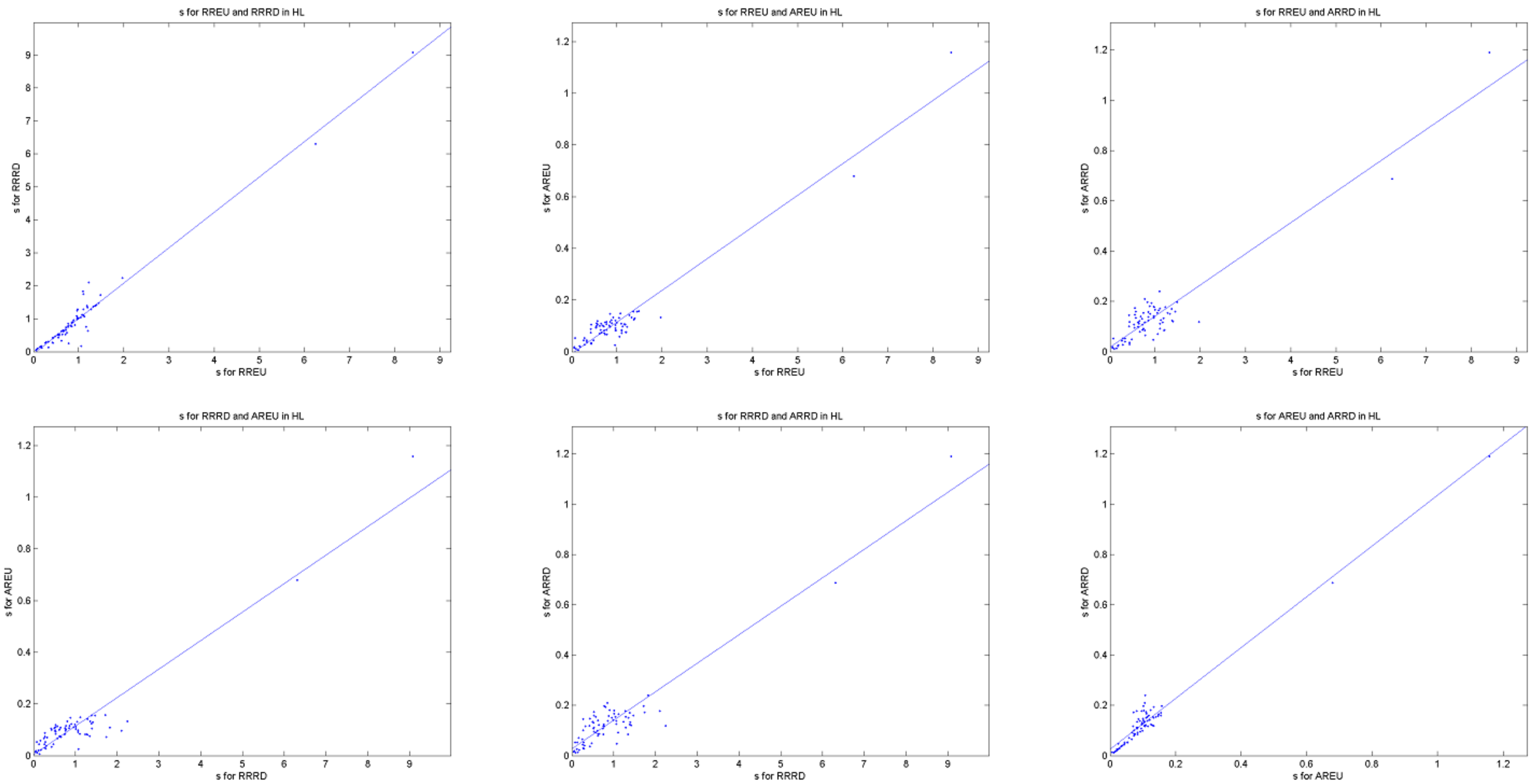


Figure E.9: Estimates of g across Preference Functionals (with Outliers)

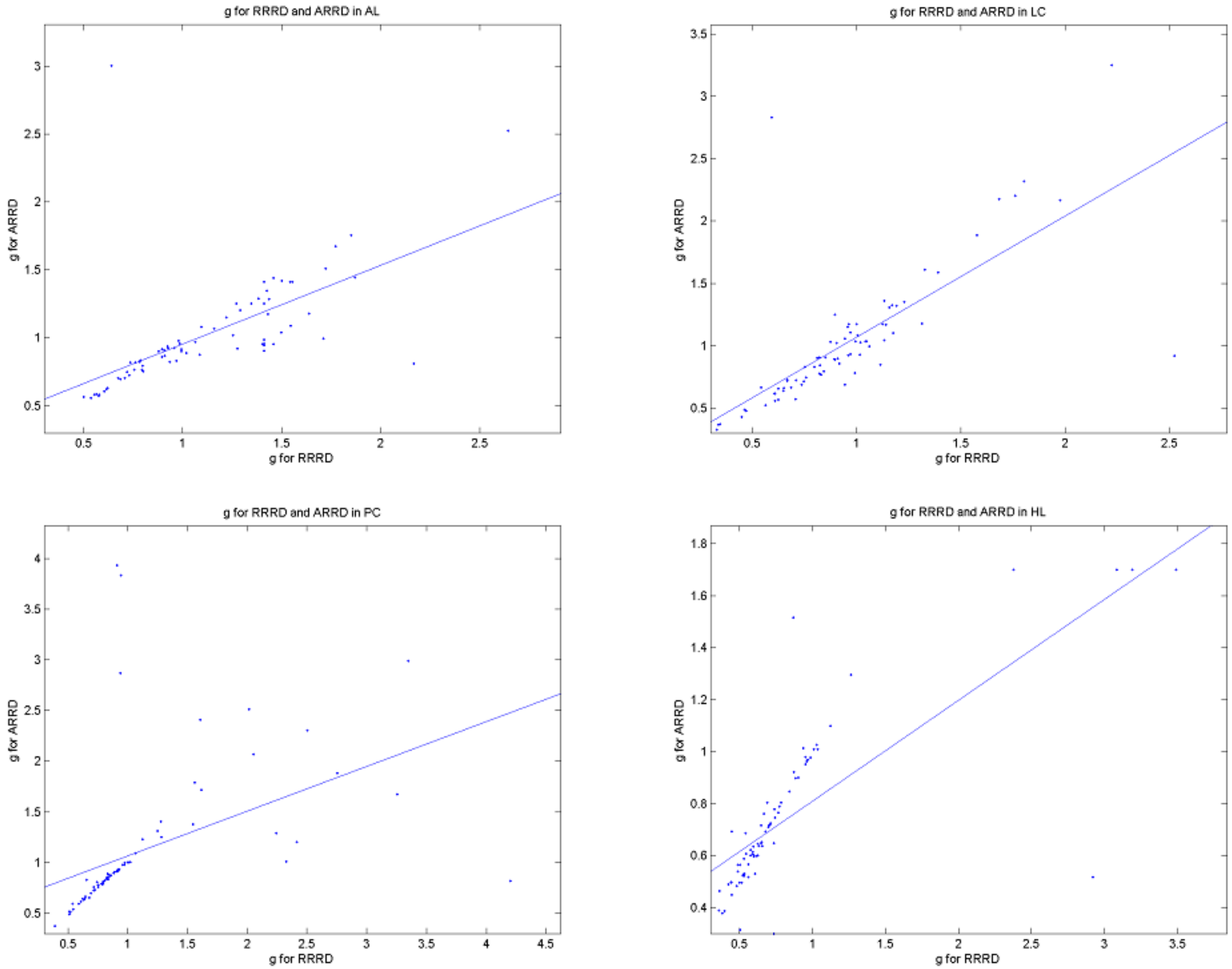


Figure E.10: Estimates of r in RREU across Elicitation Methods

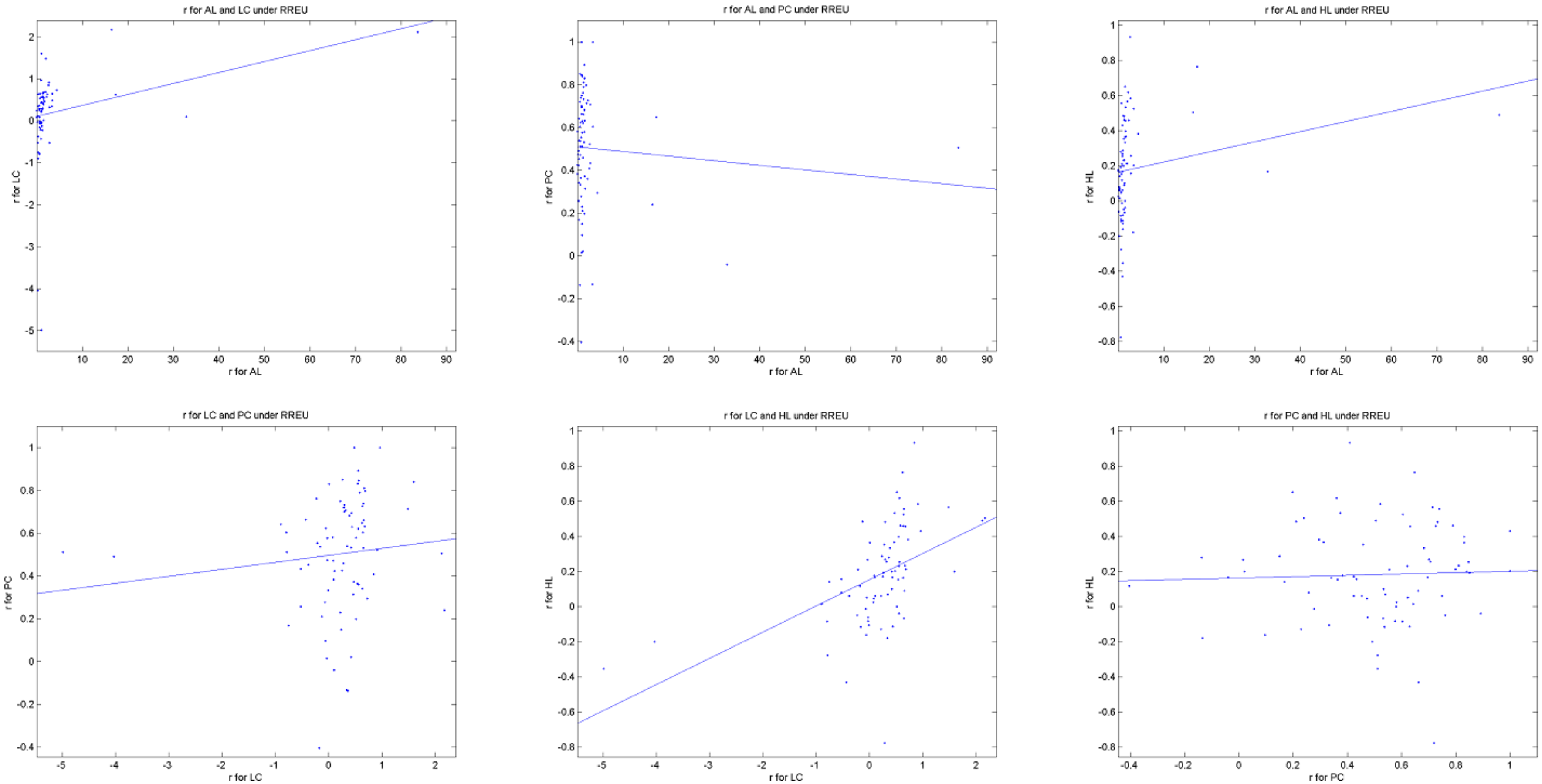


Figure E.11: Estimates of r in RRRD across Elicitation Methods

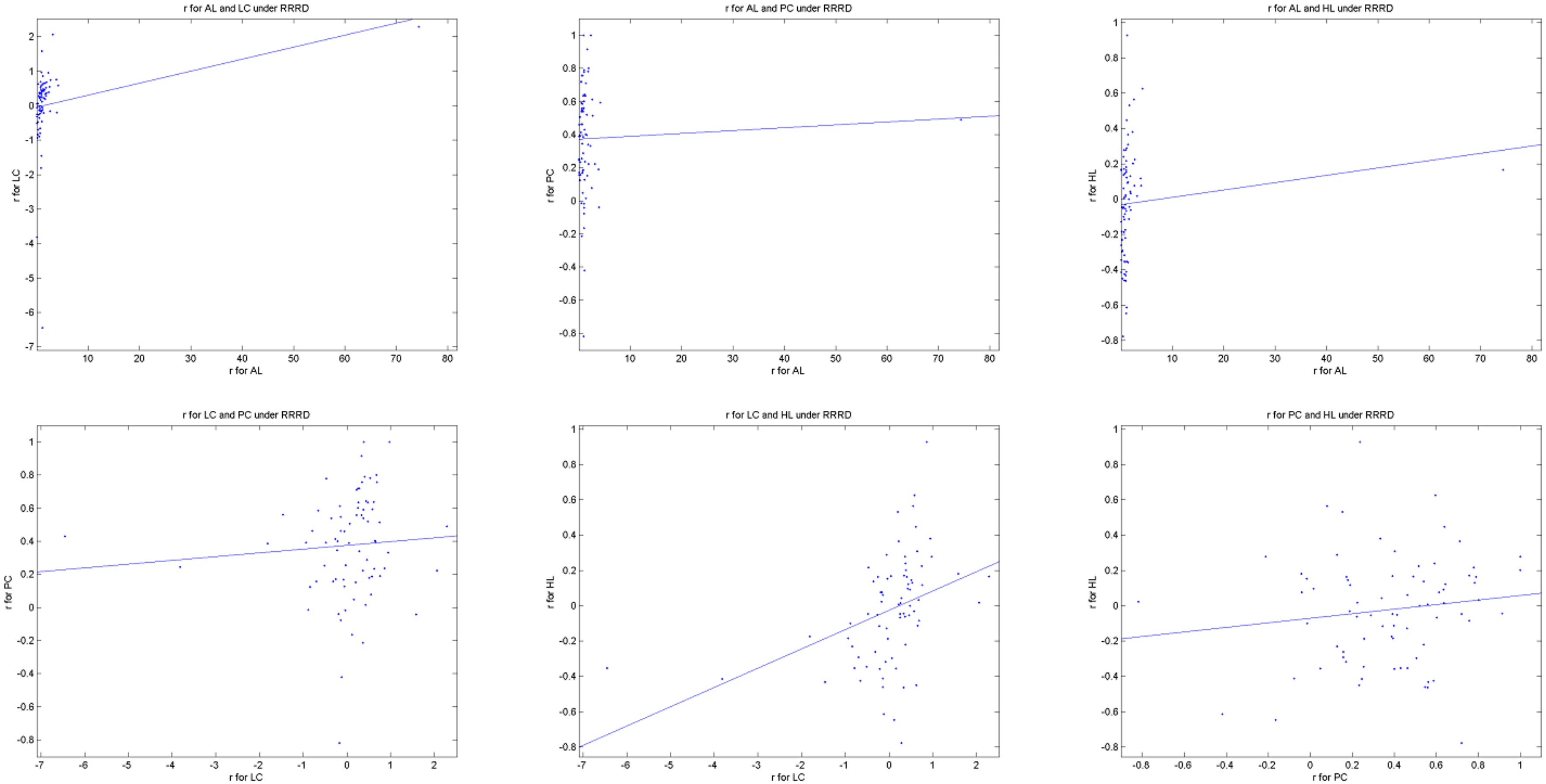


Figure E.12: Estimates of r in AREU across Elicitation Methods

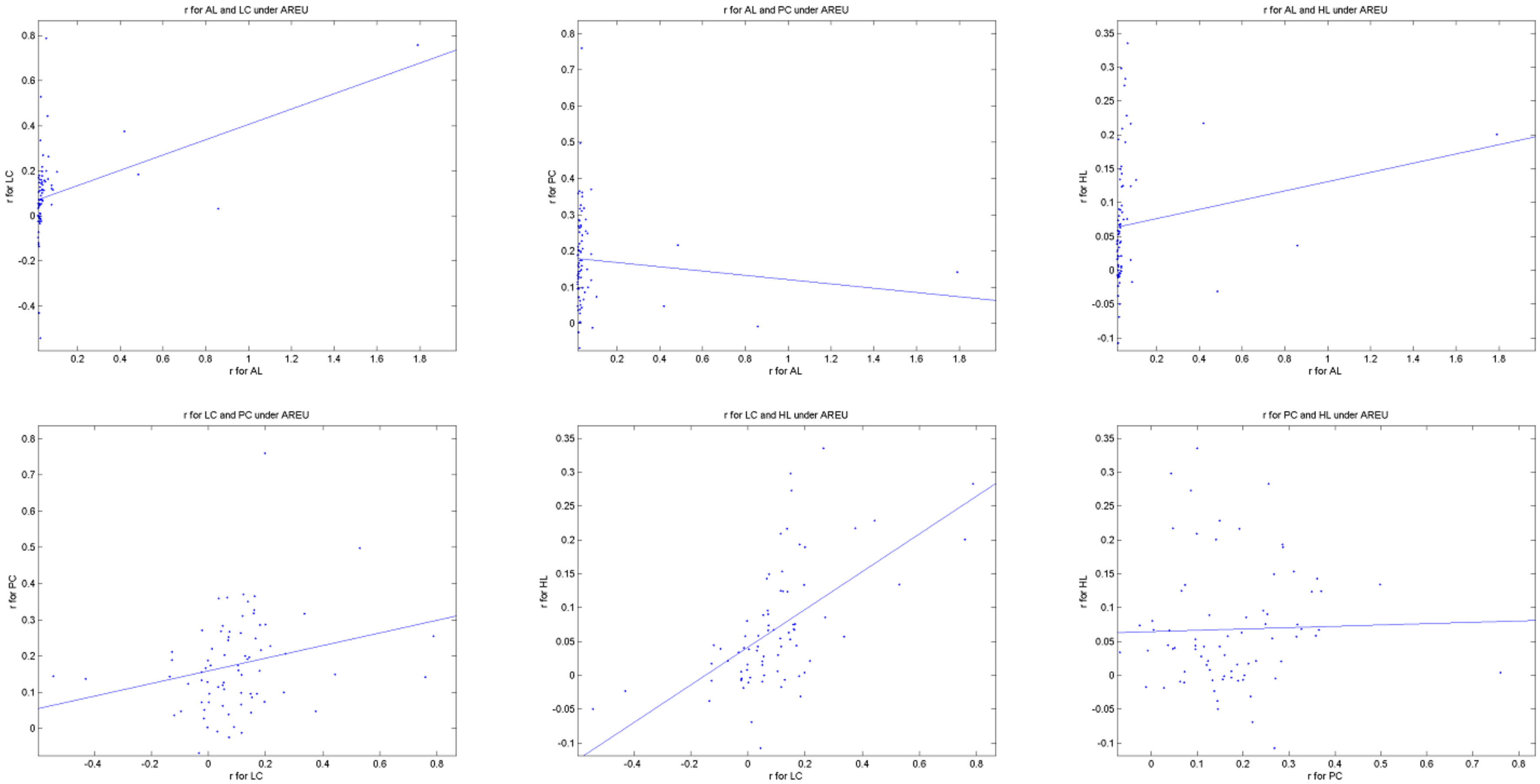


Figure E.13: Estimates of r in ARRD across Elicitation Methods

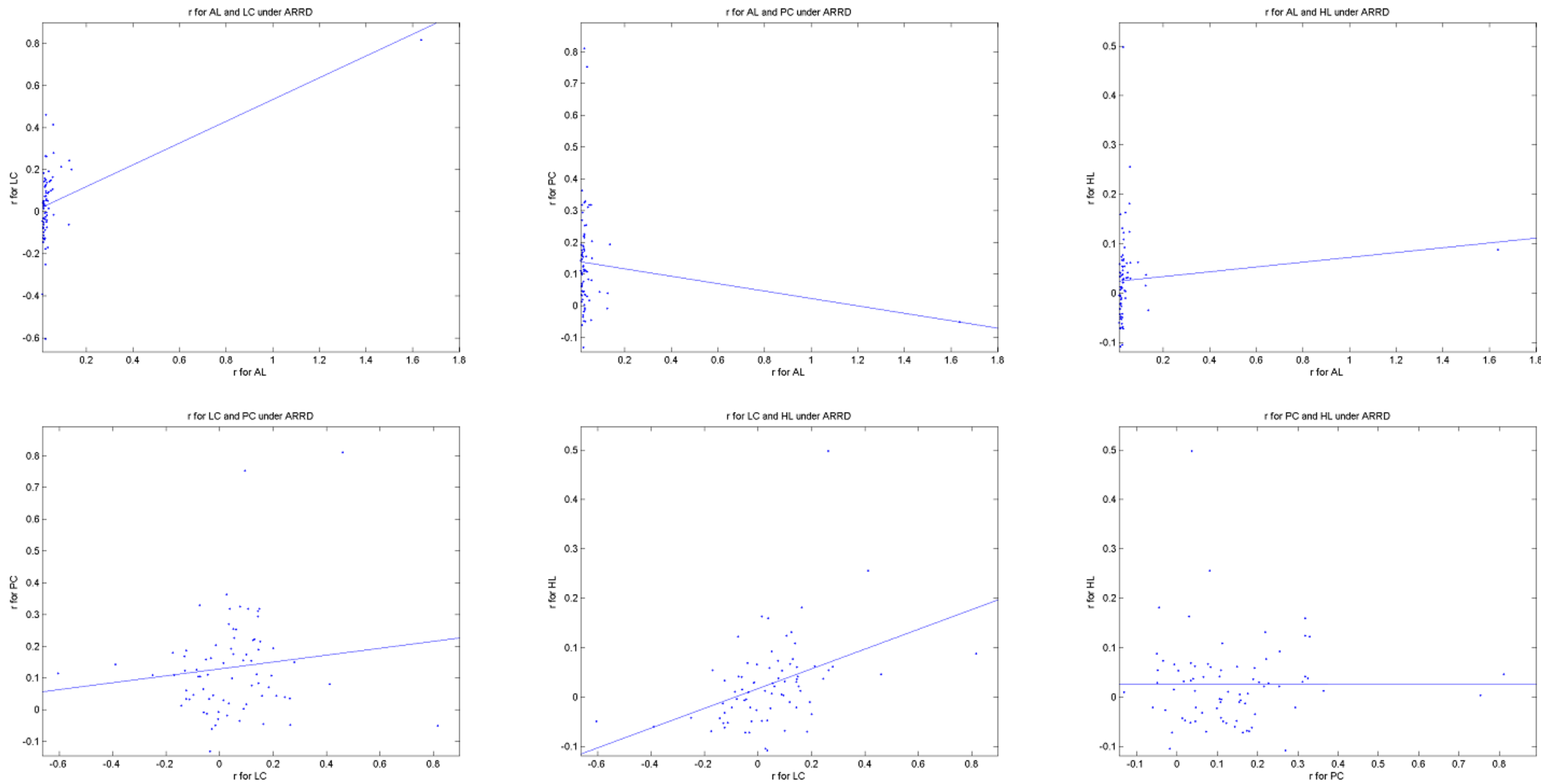


Figure E.14: Estimates of s in RREU across Elicitation Methods

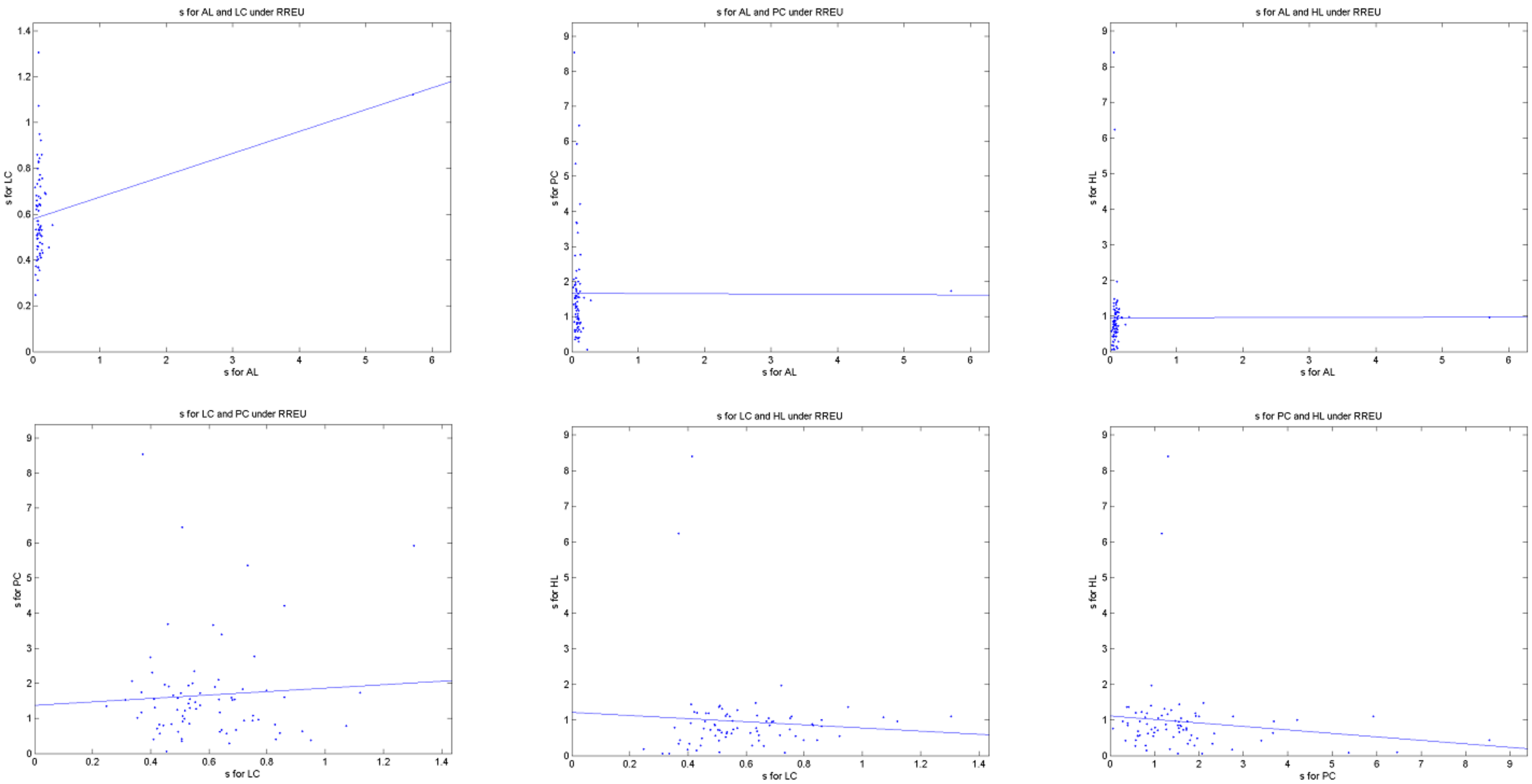


Figure E.15: Estimates of s in RRRD across Elicitation Methods

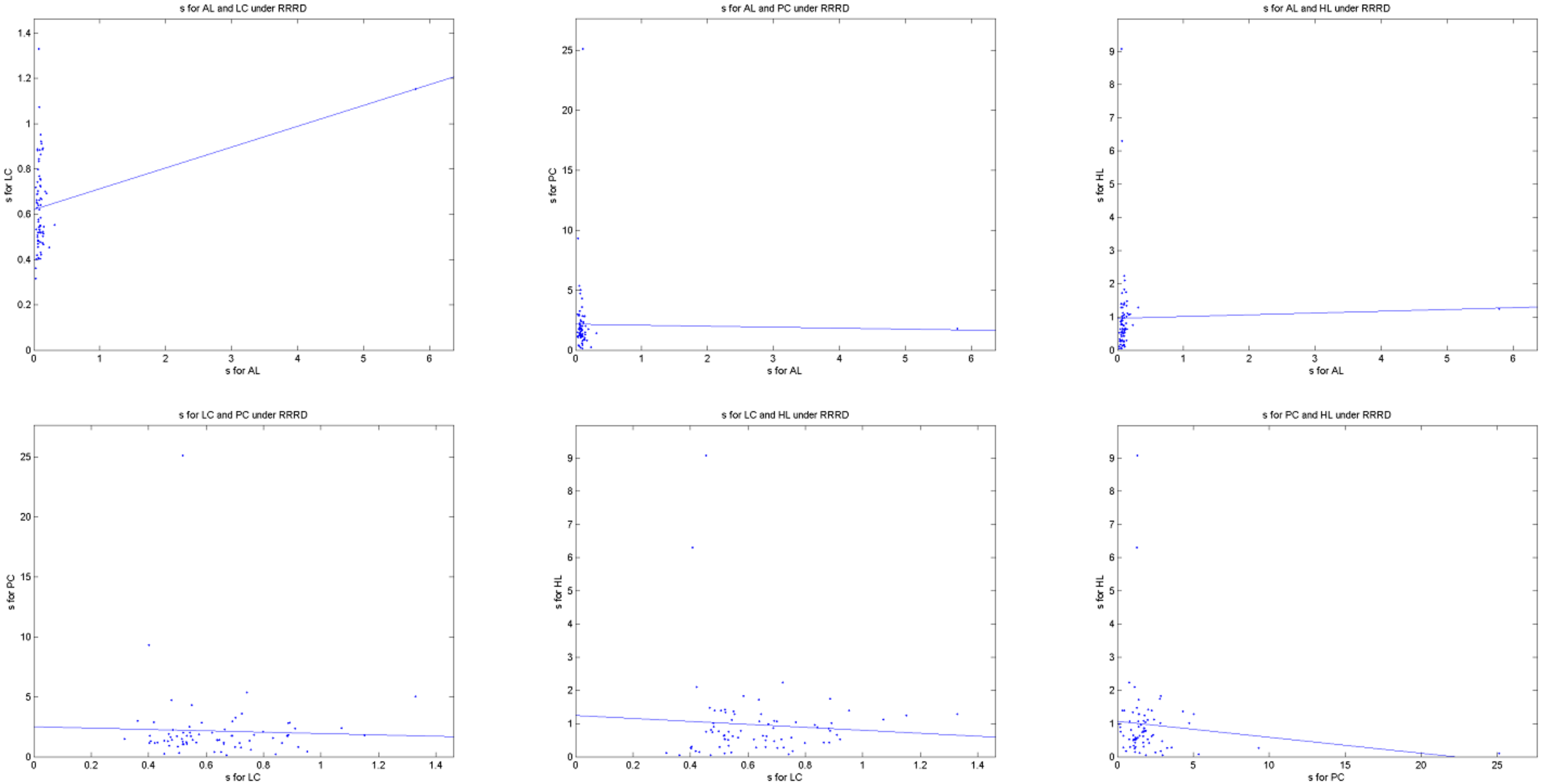


Figure E.16: Estimates of s in AREU across Elicitation Methods

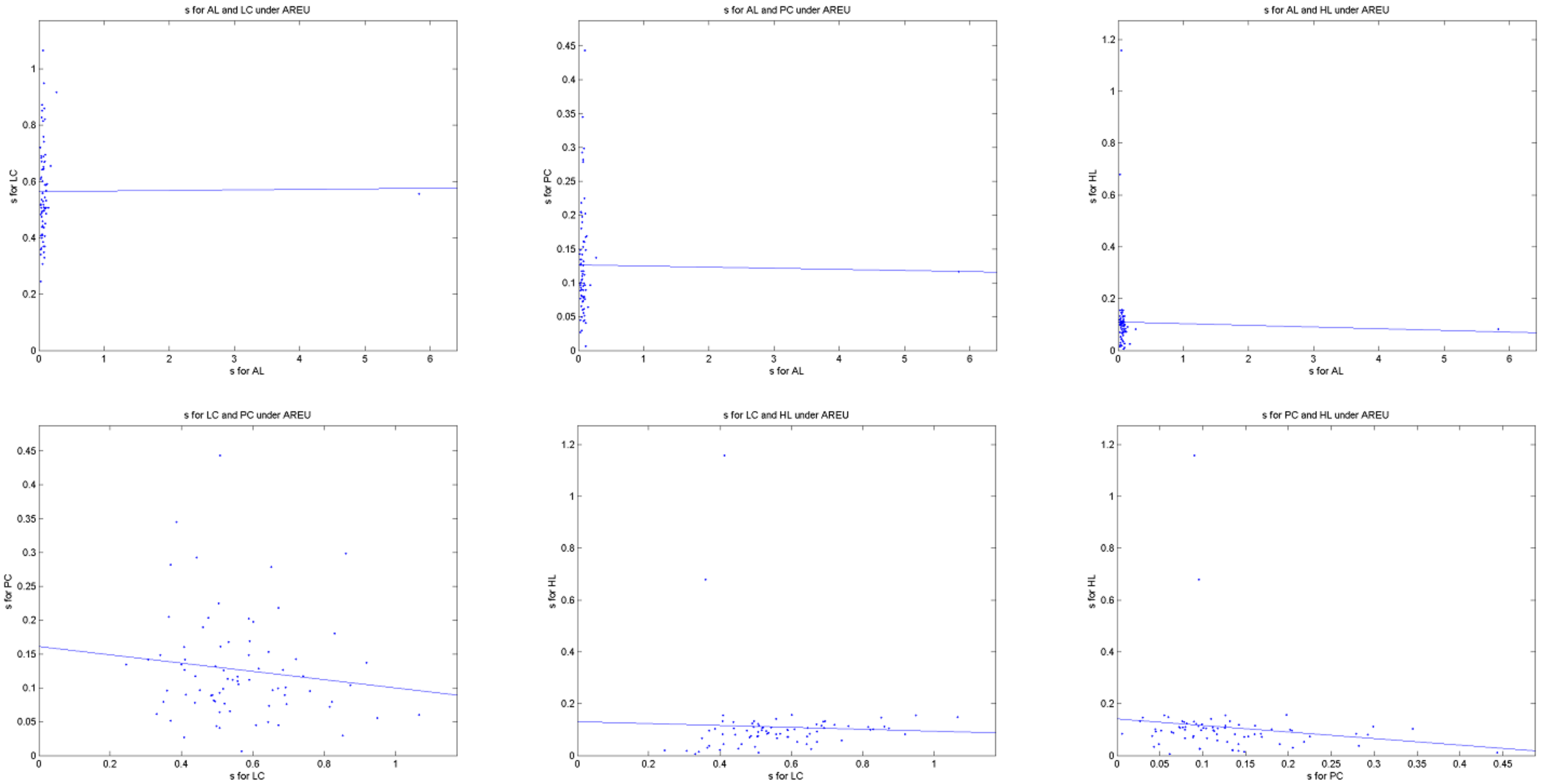


Figure E.17: Estimates of s in ARRD across Elicitation Methods

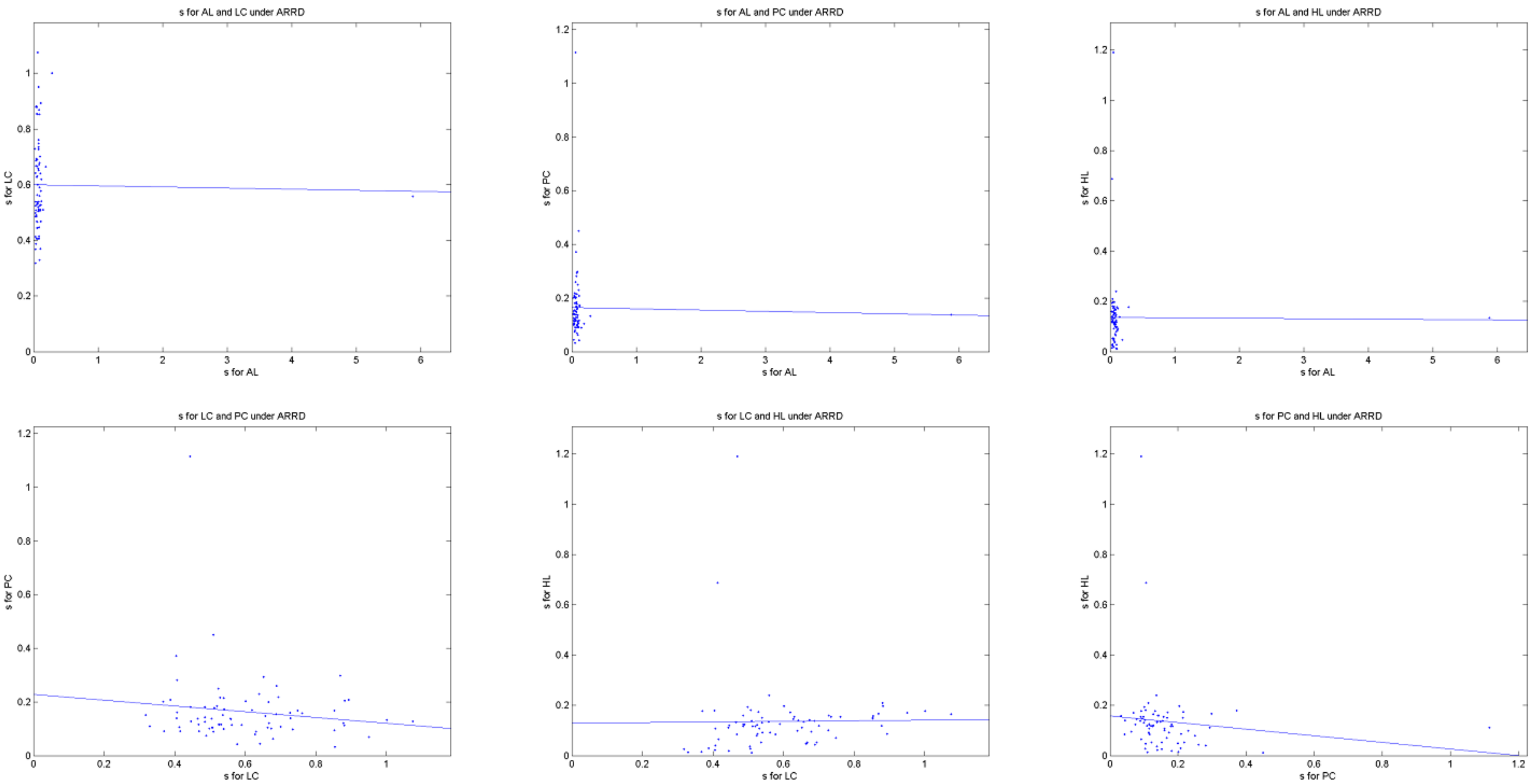


Figure E.18: Estimates of g in RRRD across Elicitation Methods

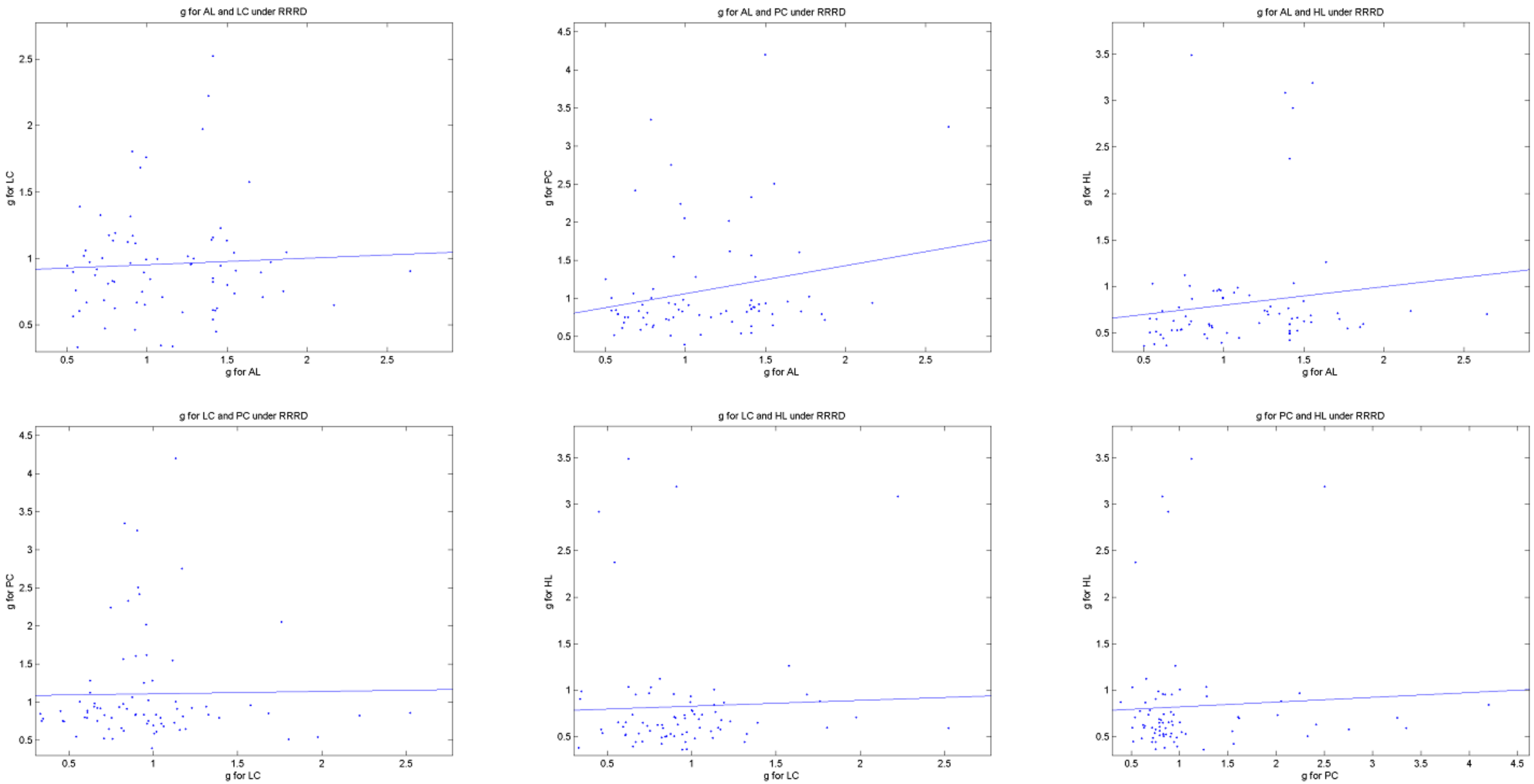
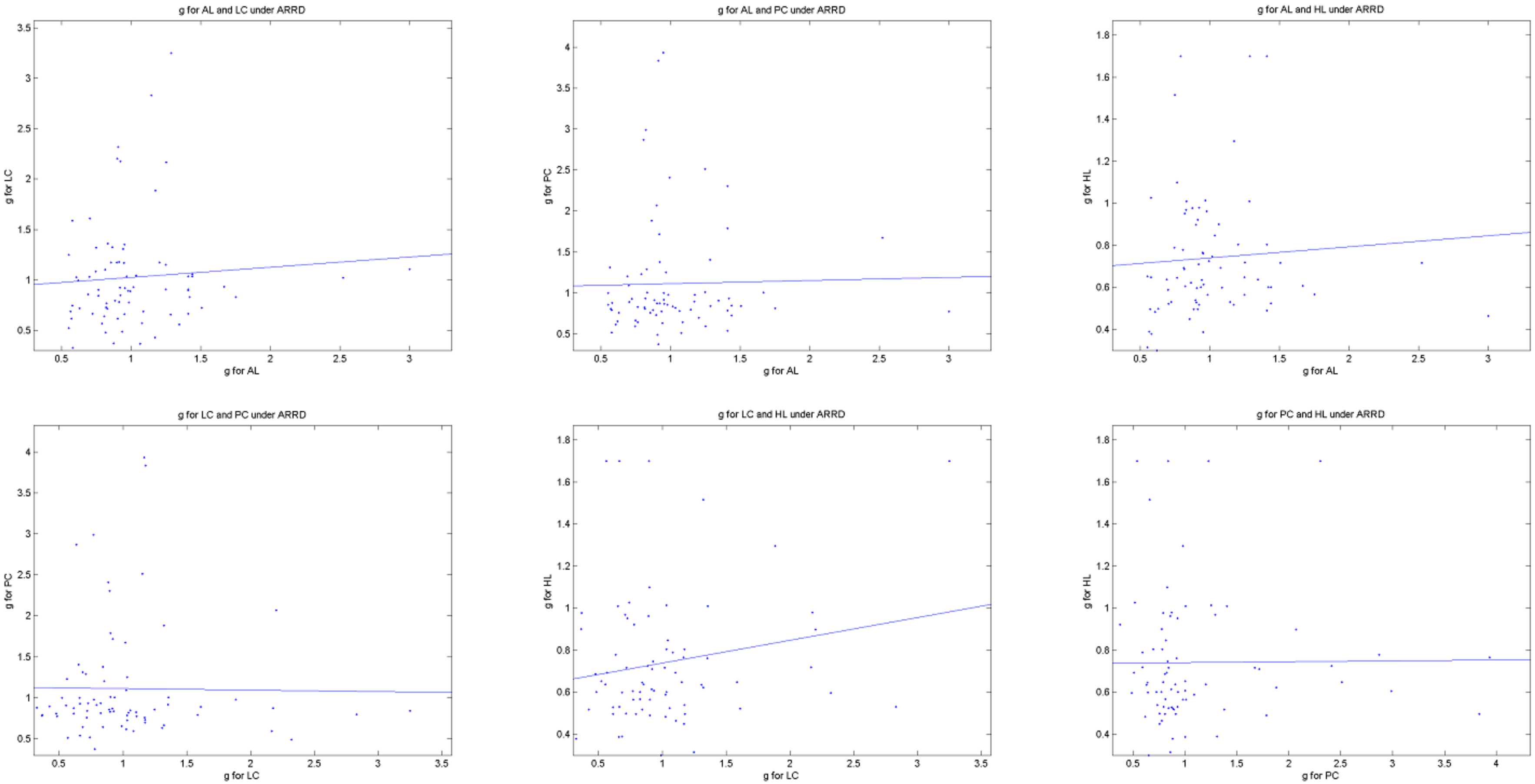


Figure E.19: Estimates of g in ARRD across Elicitation Methods



References

Andreoni J and Miller J (2002), “Giving according to garp: an experimental test of the consistency of preferences for altruism”, *Econometrica*, *70*, 737–753.

Bateman I, Day B, Loomes G and Sugden R (2015), “Ranking Versus Choice in the Elicitation of Preferences”, Working paper, University of Warwick.

Becker GM, DeGroot MH and Marschak J (1964), “Measuring Utility by a Single-Response Sequential Method”, *Behavioral Science*, 9, 226-232.

Bellman, R. (1956). On a routing problem (No. RAND-P-1000). RAND CORP SANTA MONICA CA.

Carbone E and Hey JD (1994), “Estimation of Expected Utility and Non-Expected Utility Preference Functionals Using Complete Ranking Data”, Munier B and Machina M J (eds), *Models and Experiments on Risk and Rationality*, Kluwer Academic Publishers, 119-139.

Charness G, Gneezy U and Imas A (2013) “Experimental methods: Eliciting risk preferences”, *Journal of Economic Behavior & Organization*, 87, 43-51.

Choi S, Fisman R, Gale D and Kariv S (2007), “Consistency and heterogeneity of individual behavior under uncertainty”, *American Economic Review*, *97* (4), 1921–1938.

Cubitt R, Starmer C and Sugden R (1998), "On the Validity of the Random Lottery Incentive System", *Experimental Economics*, 1, 115-131.

Deck C, Lee J, Reyes J and Rosen C (2008). “Measuring risk attitudes controlling for personality traits,” mimeo, available at:

Dijkstra, E. W. (1959). A note on two problems in connexion with graphs.Numerische mathematik, 1(1), 269-271.

Eckel CC and Grossman PJ (2002), “Sex differences and statistical stereotyping in attitudes toward financial risk” *Evolution and Human Behavior* 23, 281–295.

Floyd, R. W. (1962). Algorithm 97: shortest path. Communications of the ACM,5(6), 345.

Floyd, R. W. (1962). Algorithm 97: shortest path. Communications of the ACM,5(6), 345.

Ford Jr, L. R. (1956). Network flow theory (No. P-923). RAND CORP SANTA MONICA CA.

Gneezy U, Potters J (1997), “An experiment on risk taking and evaluation periods”, *Quarterly Journal of Economics*, 112, 631–645.

Hart, P. E., Nilsson, N. J., & Raphael, B. (1968). A formal basis for the heuristic determination of minimum cost paths. Systems Science and Cybernetics, IEEE Transactions on, 4(2), 100-107.

Hey J D (2001), "Does Repetition Improve Consistency?", *Experimental Economics*, 4, 5-54.

Hey J D and Lee J (2005a), "Do Subjects Separate (or Are They Sophisticated)?", *Experimental Economics*, 8, 233-265.

Hey J D. and Lee J (2005b), "Do subjects remember the past?", *Applied Economics*, 37, 9-18.

Hey JD and Orme CE (1994), “Investigating Generalisations of Expected Utility Theory Using Experimental Data”, *Econometrica,* 62*,* 1291-1326.

Holt C (1986), "Preference Reversals and the Independence Axiom", *American Economic Review*, 76, 508-515.

Holt CA and Laury S (2002), “Risk Aversion and Incentive Effects”, *American Economic Review*, 92, 1644-1655.

http://www.econ.canterbury.ac.nz/research/pdf/Paper_Deck.pdf.

Johnson, D. B. (1977). Efficient algorithms for shortest paths in sparse networks. Journal of the ACM (JACM), 24(1), 1-13.

Lee J (2008), "The Effect of the Background Risk in a Simple Chance Improving Decision Model", *Journal of Risk and Uncertainty*, 36, 19-41.

Lejuez CW, Aklin WM, Zvolensky MJ and Pedulla CM (2003), "Evaluation of the Balloon Analogue Risk Task (BART) as a predictor of adolescent real-world risk-taking behaviours", *Journal of Adolescence,* 26, 475–479.

Loomes G (1991), "Evidence of a new violation of the independence axiom", *Journal of Risk and Uncertainty*, 4, 91–108.

Loomes G and Pogrebna G (2014), "Measuring Risk Attitudes when Preferences are Imprecise", *Economic Journal*, 124, 569–593.

Starmer C and Sugden R (1991), "Does the Random-Lottery Incentive System Elicit True Preferences? An Experimental Investigation", *American Economic Review*, 81, 971-978.

Viterbi, A. J. (1967). Error bounds for convolutional codes and an asymptotically optimum decoding algorithm. *IEEE Transactions on Information Theory*, 13(2), 260-269.

Wakker P (2010), *Prospect Theory*, Cambridge University Press.

Wu G and Gonzalez R (1996), "Curvature of the probability weighting function", *Management Science,* 42, 1676–1690.