

Dataset_S&Z.dta

Task:

- = 1: Task P
- = 2: Task AC ($P \cdot D$)
- = 3: Task AC ($P \cdot D + A$)
- = 4: Task AC ($P \cdot D_1 \cdot D_2$)
- = 5: Task NC ($P \cdot D$)
- = 6: Task NC ($P \cdot D + A$)
- = 7: Task NC ($P \cdot D_1 \cdot D_2$)
- = 8: Task PC ($P \cdot D$)
- = 9: Task PC ($P \cdot D + A$)
- = 10: Task PC ($P \cdot D_1 \cdot D_2$)

Participants: Participants' ID

excess_price: excess price = the price of the chosen offer – the price of the optimal offer

max_time: Time used

optimal_or_not:

- =1 if the chosen offer is optimal
- =0 if chosen offer isn't optimal

Number_of_offers_detected: Number of offers inspected in a task

Number_of_CSoffers_detected: Number of common standard offers inspected in a task

Number_of_noCSoffers_detected: Number of individuated standard offers inspected in a task

final_decision_cs_or_not:

- =1 if the chosen offer is a common standard offer
- =0 if the chosen offer is an individuated standard offer

task_type:

- =1 if the task is the baseline task
- =2 if the task is an AC task
- =3 if the task is a NC task
- =4 if the task is a PC task

actions_index: the sequence of each click.

max_index: the number of clicks in total for each subject in each task.

action_proceed_10sections: 10 clicking time intervals (from 1 to 10)

best_cs_inthe_basket_or_not :

- =1 if the best common standard offer is in the basket at moment i
- =0 if the best common standard offer isn't in the basket at moment i

best_nocs_inthe_basket_or_not :

=1 if the best individuated standard offer is in the basket at moment i

=0 if the best individuated standard offer isn't in the basket at moment i

From noBest_CS_good_in_the_basket2 to noBest_CS_good_in_the_basket24

=1 if a common standard offer that is not the best common standard offer, but is the (from 2nd to 24th) cheapest offer among 24 offers, is in the basket at moment i

=0 if a common standard offer that is not the best common standard offer, but is the (from 2nd to 24th) cheapest offer among 24 offers, is not in the basket at moment i

From noBest_noCS_good_in_the_basket2 to noBest_noCS_good_in_the_basket24

=1 if an individuated standard offer that is not the best individuated standard offer, but is the (from 2nd to 24th) cheapest offer among 24 offers, is in the basket at moment i

=0 if an individuated standard offer that is not the best individuated standard offer, but is the (from 2nd to 24th) cheapest offer among 24 offers, is not in the basket at moment i

From random_good_cs_or_not1 to random_good_cs_or_not24: 1 = the (from 1st to 24th) cheapest offer is a common standard offer; 0 = otherwise

twoclick_ratio: the ratio of two-click inspections to total inspections (include repeated inspections)

oneclick_ratio: the ratio of one-click inspections to total inspections (include repeated inspections)

Inspected_offer_code: The code of the offer (from 1 to 24, price from low to high) that has been

inspected at momentst i (it equals 0 if no offer has been inspected at momentst i)

Inspected_offer_cs_or_not

=1 if the inspected offer at momemnt i is a common standard offer

=0 else

best_cs: the rank of the best cs offer

best_nocs: the rank of the best IS offer

final_earnings: Based on the final choice, how much money a subject would earn.

offer_price: The final price of the offer.

final_decision: This variable measures the final prices of the offers chosen by subjects in the relevant task. It uses a linear transformation of the final price scale; the value 1 denotes the lowest of the 24 final prices (£0.27 in all tasks) and the value 24 denotes the highest of these prices (£8.90).

group: 1 = if the subject inspected 7 or 8 common standard offers in the first 8 inspections; 0 = otherwise

offer_type:

1= $P \cdot D$

2= $P \cdot D + A$

3= $P \cdot D_1 \cdot D_2$

4= P

From cs_rank1 to cs_rank8: The rank of the CS offer (within 24 offers) if it is the i th best CS offer of the 8 CS offers.

From cs_rank2_in_basket_or_not to cs_rank8_in_basket_or_not: Whether or not cs_rank`i' offer is in the basket. 1=YES 0=Otherwise.

From nocs_rank1 to nocs_rank8: The rank of the IS offer (within 24 offers) if it is the i th best IS offer of the 16 IS offers.

From nocs_rank2_in_basket_or_not to nocs_rank16_in_basket_or_not: Whether or not nocs_rank`i' offer is in the basket. 1=YES 0=Otherwise.

Dataset_Regression (include demographic variables).dta

Dependent variable:

final_decision

This variable measures the final prices of the offers chosen by subjects in the relevant task. It uses a linear transformation of the final price scale; the value 1 denotes the lowest of the 24 final prices (£20.27 in all tasks) and the value 24 denotes the highest of these prices (£28.90).

Independent variables:

task_position: The position of the task in the series of ten tasks faced by the subject.

Number_of_offers inspected: From 1 to 24. This variable represents the number of offers subjects inspected in each task.

time_spent: The time (in seconds) spent on the task.

subject_click_type: {Subjects are classified into three types.

1= One-click subjects. Subjects who inspected more than one offer and had a ratio of one-click inspections to total inspections greater than 0.8.

2= Two-click subjects. Subjects who inspected more than one offer and had a ratio of two-click inspections to total inspections greater than 0.8.

3= Subjects are neither-type.

eco_relevance: A dummy variable which takes the value 1 if the subject's field of study includes economics or mathematics and 0 otherwise.

Gender: 0 = male, 1 = female.

Age: Subject's age.

group1: In each PC task, subjects are classified into two groups.

1 = "subjects inspected 7 or 8 common standard offers in their first 8 inspections;"

0 = "subjects inspected less than 7 common standard offers in their first 8 inspections".

Offer type:

1= P*D (Baseline)

2= P*D+A

3= P*D1*D2

4= P