Poverty and Economic Decision-Making

Gov 50 - Section 4

Introduction

Do changes in one's financial circumstances affect one's decision-making process and cognitive capacity? In an experimental study, researchers randomly selected a group of US respondents to be surveyed before their payday and another group to be surveyed after their payday. Under this design, the respondents of the `Before Payday` group are more likely to be financially strained than those of the `After Payday` group. The researchers were interested in investigating whether or not changes in people's financial circumstances affect their decision making and cognitive performance. Other researchers have found that scarcity induce an additional mental load that impedes cognitive capacity. This exercise is based on:

Carvalho, Leandro S., Meier, Stephen, and Wang, Stephanie W. (2016). "Poverty and economic decision-making: Evidence from changes in financial resources at payday." *American Economic Review*, Vol. 106, No. 2, pp. 260-284.

In this study, the researchers administered a number of decision-making and cognitive performance tasks to the `Before Payday` and `After Payday` groups. We focus on the *numerical stroop task*, which measures cognitive control. In general, taking more time to complete this task indicates less cognitive control and reduced cognitive ability. They also measured the amount of cash the respondents have, the amount in their checking and saving accounts, and the amount of money spent. The data set is in the CSV file `poverty.csv`. The names and descriptions of variables are given below:

Name	Description
`treatment`	Treatment conditions: `Before Payday` and `After Payday`
`cash`	Amount of cash respondent has on hand
`accts_amt`	Amount in checking and saving accounts
`stroop_time`	Log-transformed average response time for cognitive stroop test
`income_less20k`	Binary variable: `1` if respondent earns less than 20k a year and `0` otherwise

Question 1: Summary Statistics

First, read in the experiment's data from `poverty.csv` into an object named `dat`, using `read_csv()`.

We may think that a respondent's financial situation is important in this experiment. Let's get some summary statistics for `cash` and `accts_mt` to help us understand these variables better.

Use the `summary()` function to learn about the distribution of cash on hand and money in the bank in our sample. Instead of using the pipe `|>`, be sure to use the `$` to tell R which column you want to summarize. You can use the following format: `summary(dataset$variable)`

Then, use the `group_by()` and `summarize()` functions to look at the `mean()`, `median()`, and `sd()` (standard deviation) of the cash variable, for the two groups (Before Payday and After Payday).

What do you notice about these variables that may be important for the study? Do you think the mean or the median is a better summary statistic of the central tendency of these variables? And is there significant variation in the spread before and after payday?

Answer 1

```r
library(tidyverse)
```

```
## -- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 --
## v dplyr     1.1.4     v readr     2.1.5
## v forcats   1.0.0     v stringr   1.5.1
## v ggplot2   3.5.0     v tibble    3.2.1
## v lubridate 1.9.3     v tidyr     1.3.1
## v purrr     1.0.2
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
## i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become errors
```

```r
dat <- read_csv("data/poverty.csv")
```

```
## Rows: 2670 Columns: 5
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr (1): treatment
## dbl (4): cash, accts_amt, stroop_time, income_less20k
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```r
summary(dat$cash)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.    NA's
##     0.0    15.0    49.5   169.0   136.2  9000.0     226
```

```r
summary(dat$accts_amt)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.    NA's
##       0     176    1000    6211    5000   95000     433
```

```r
# Method 1: na.rm in each function
dat |> group_by(treatment) |>
  summarize(mean_cash = mean(cash, na.rm = T),
            median_cash = median(cash, na.rm = T),
            sd_cash = sd(cash, na.rm = T))
```

```
## # A tibble: 2 x 4
##   treatment     mean_cash median_cash sd_cash
##   <chr>             <dbl>       <dbl>   <dbl>
## 1 After Payday       188.          50    525.
## 2 Before Payday      151.          40    365.
```

```r
# Method 2: filter by !is.na(var)
dat |> filter(!is.na(cash)) |>
  group_by(treatment) |>
  summarize(mean_cash = mean(cash),
            median_cash = median(cash),
            sd_cash = sd(cash))
```

```
## # A tibble: 2 x 4
##   treatment     mean_cash median_cash sd_cash
##   <chr>             <dbl>       <dbl>   <dbl>
## 1 After Payday       188.           50    525.
## 2 Before Payday      151.           40    365.
```

These variables have a very wide range with a large gap between the two measures of central tendency. The value of the mean is much higher than the median, telling us that we likely have a few very large outliers at the high end of cash on hand and amount of money in the bank. In this case, the median is generally going to be a better measure of central tendency because it is not impacted by the magnitude of the outliers. Unsurprisingly, all three variables increase after payday, but standard deviation increases by nearly $200.

Question 2: Proportion table

Now let's look at the outcome variable `stroop_time`. Use `case_when` to make a new variable called `stroop_3way` that breaks the continuous variable `stroop_time` into three categories : under 7 seconds, between 7 and 7.5 seconds, and more than 7.5 seconds. Then create a pivot table that shows the proportions of the sample that fall into these three categories.

- Hint: Create a table of counts by first mutating to create the `stroop_3way` variable for the groups, then use `group_by()` and `summarize()` to get a new column named `count` with the counts for each group. Then use `mutate()` to add a column of proportions and `select(-count)` to drop the `count` column, so we're just left with proportions. Finally, use `pivot_wider()` to get the final pivot table.

Answer 2

```
dat <- dat |>
  mutate(stroop_3way = case_when(stroop_time < 7 ~ "Under 7",
                                 stroop_time >= 7 & stroop_time < 7.5 ~ "Between",
                                 stroop_time >= 7.5 ~ "More than 7.5"))

table(dat$stroop_3way) |> prop.table()
```

```
##
##       Between More than 7.5       Under 7
##    0.33707865    0.63857678    0.02434457
```

Question 3: Examining and plotting variables with multiple groupings

Next let's look at the distribution of cash on hand by treatment status and income.

First, use `mutate()` to create a new variable called `income_less20k_rec` that recodes the current income variable `income_less20k` to have more informative values. Replace the numerical values from `income_less20k` with "Lower income" and "Higher income" for this new variable. **Save this variable into the dataset (keep the name `dat`)**. Then create a table that shows average cash on hand for each combination of the income indicator and the treatment assignment.

Then create a bar plot of the data in the table using `ggplot`. The y-axis should show cash on hand and the x-axis should show the income indicator. The plot should contain double bars, with one bar corresponding to before payday and one bar corresponding to after payday. Make sure to set informative labels and a title too!

- Hint: To create the double bars, use `geom_col(position = "dodge")`. You will also have to set the `fill()` argument in the `aes()` function in order to specify that there should be separate bars for before/after payday.

Answer 3

```
dat <- dat |>
  mutate(income_less20k_rec = ifelse(income_less20k,
```

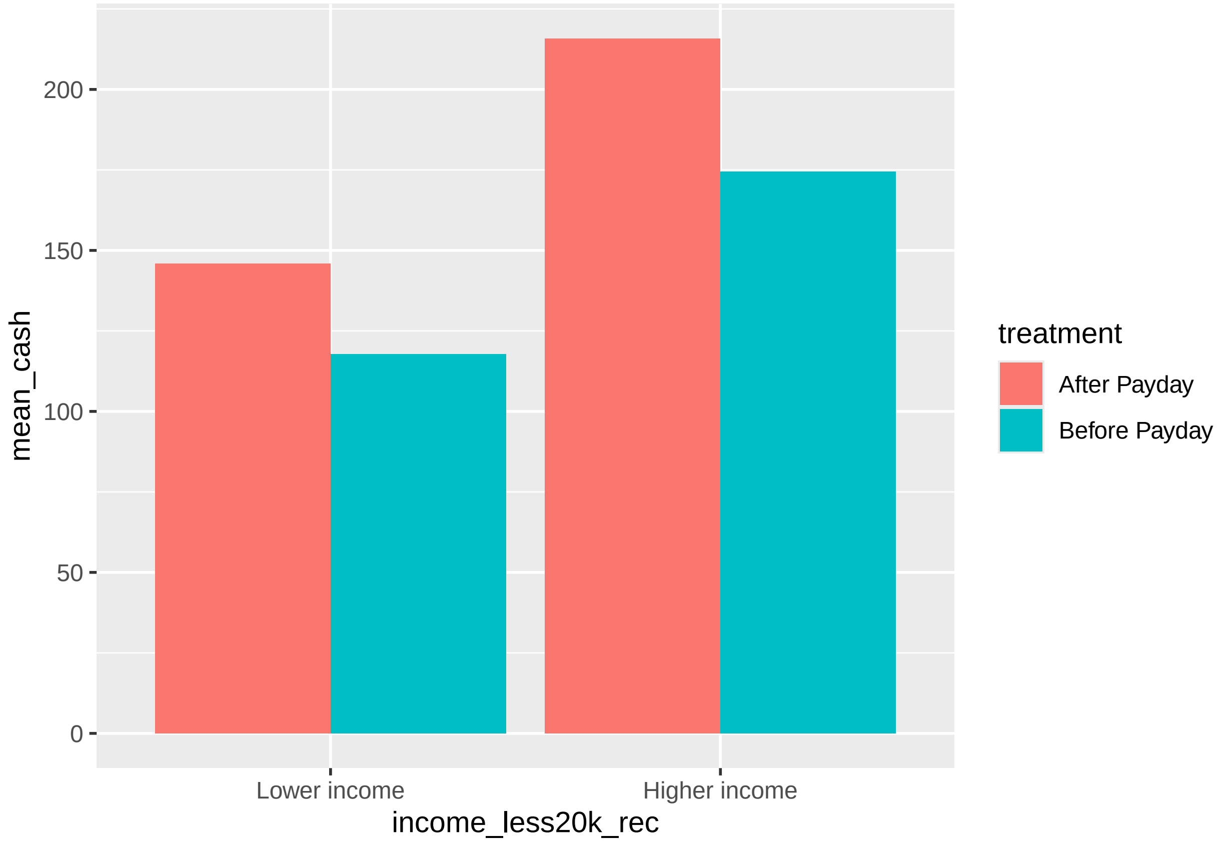
```r
                                     "Lower income", "Higher income") |>
            factor(levels = c("Lower income", "Higher income")))

# dat$income_less20k_rec <- factor(dat$income_less20k_rec, levels = c("Lower income", "Higher income"))

plot_table1 <- dat |> group_by(income_less20k_rec, treatment) |>
  summarize(mean_cash = mean(cash, na.rm = T))
```

```
## `summarise()` has grouped output by 'income_less20k_rec'. You can override
## using the `.groups` argument.
```

```r
plot_table1
```

```
## # A tibble: 4 x 3
## # Groups:   income_less20k_rec [2]
##   income_less20k_rec treatment     mean_cash
##   <fct>              <chr>             <dbl>
## 1 Lower income       After Payday       146.
## 2 Lower income       Before Payday      118.
## 3 Higher income      After Payday       216.
## 4 Higher income      Before Payday      174.
```

```r
ggplot(plot_table1, mapping = aes(x = income_less20k_rec, y = mean_cash,
                        fill = treatment)) +
  geom_bar(stat = "identity", position = "dodge")
```



Question 4: Average Treatment Effect

Now let's calculate the Average Treatment Effect (ATE), both overall and then within income groups, using the same $20k income cutoff as Question 3. What is the effect of payday on stroop test results? Do these effects change when looking at higher and lower income groups?

- Hint: We will use `group_by()`, `summarize()`, `pivot_wider()`, and `mutate()` to find the ATE both times.

Answer 4

```
## Overall ATE
dat |>
  group_by(treatment) |>
  summarize(stroop_time = mean(stroop_time, na.rm = TRUE)) |>
  pivot_wider(names_from = treatment,
              values_from = stroop_time) |>
  mutate(ATE = `After Payday` - `Before Payday`)
```

```
## # A tibble: 1 x 3
##   `After Payday` `Before Payday`    ATE
##            <dbl>           <dbl>  <dbl>
## 1           7.55            7.54 0.0114
```

```
## Income Group ATE
dat |>
  group_by(income_less20k_rec, treatment) |>
  summarize(stroop_time = mean(stroop_time, na.rm = TRUE)) |>
  pivot_wider(names_from = treatment,
              values_from = stroop_time) |>
  mutate(ATE = `After Payday` - `Before Payday`)
```

```
## `summarise()` has grouped output by 'income_less20k_rec'. You can override
## using the `.groups` argument.
```

```
## # A tibble: 2 x 4
## # Groups:   income_less20k_rec [2]
##   income_less20k_rec `After Payday` `Before Payday`     ATE
##   <fct>                       <dbl>           <dbl>   <dbl>
## 1 Lower income                 7.55            7.55 0.00560
## 2 Higher income                7.55            7.53 0.0157
```

The overall average treatment effect is .011 seconds – the difference is higher for higher income individuals, at .0157 seconds, and lower for lower income individuals, at .006 seconds. We cannot say whether these differences are statistically significant at this point in the course.

Question 5: Missing data

Now let's look to see how our average treatment effect and case count (how many observations we have) changes based on what we decide to do with missing data.

First make a table that shows the mean outcome and case count for `stroop_time` by treatment status.

Then replicate that table but first drop all cases that have missing values using `drop_na()`.

Finally, replicate the table again, but this time drop only cases that have missing values for the `cash` variable (you will want to use a combination of `filter()` and `is.na()`). You will need to use the `!` operator to filter your data down to the set of observations for which `cash` is **not** missing. Remember - we can use the `!` operator to negate logical statements (e.g., `!=` means “not equal to”).

Compare the averages for `Before Payday` and `After Payday` and the accompanying case counts across the three tables. What do you notice?

Answer 5

```
full <- dat |>
  group_by(treatment)|>
  summarize(avg_stroop_time = mean(stroop_time),
            count = n()) |>
  knitr::kable()

any_NA <- dat |> drop_na() |>
  group_by(treatment)|>
  summarize(avg_stroop_time = mean(stroop_time),
            count = n()) |>
  knitr::kable()

cash_NA <- dat |> filter(!is.na(cash)) |>
  group_by(treatment)|>
  summarize(avg_stroop_time = mean(stroop_time),
            count = n()) |>
  knitr::kable()


full
```

treatment	avg_stroop_time	count
After Payday	7.550519	1316
Before Payday	7.539096	1354

```
any_NA
```

treatment	avg_stroop_time	count
After Payday	7.545226	1093
Before Payday	7.540923	1144

```
cash_NA
```

treatment	avg_stroop_time	count
After Payday	7.544245	1202
Before Payday	7.543365	1242

There are slight variations in the case counts and treatment effects across the three tables. We get the largest case count and largest average treatment effect when using the full dataset. In the two tables in which we drop cases, the case count is lower and the treatment effect is smaller (ate = .004 when all cases with missing data are dropped and ate = .001 when only cases missing the `cash` variable are dropped).

Question 6: BONUS QUESTION - Examining central tendency, range and spread by subgroup

Let's take another look at the distribution of financial resources by income level. Create a table that groups by the low income indicator variable `income_less20k`. Look at the table at the beginning of this activity to get more information about this variable. This indicator variable should be in the rows and you should have the following summary statistics for `accts_amt` in the columns:

- mean
- median
- standard deviation
- minimum value
- maximum value

After creating the table, plot the distribution of `accts_amt` for each income group using `geom_histogram()` and `facet_wrap()`. Add vertical lines for the mean and median on each graph using `geom_vline()` - be sure to make the mean and median lines different colors.

Does the table or the plot give you more detailed information about the distribution of `accts_amt`? When might it be helpful to provide more detailed information and when might it be helpful to provide less detailed information to your audience?

Answer 6

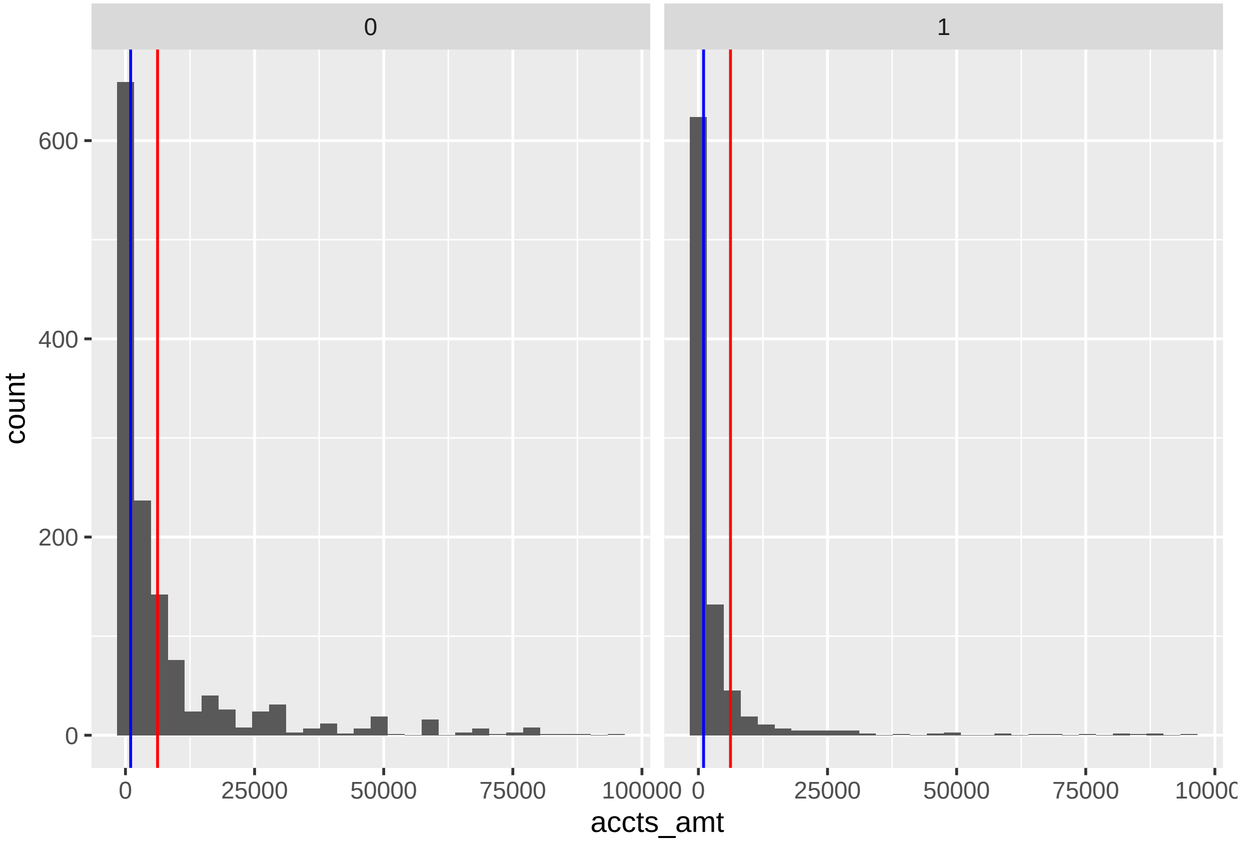
```r
tab2 <- dat |> group_by(income_less20k) |>
  summarize(mean_acct = mean(accts_amt, na.rm=TRUE),
            med_acct = median(accts_amt, na.rm=TRUE),
            sd_acct = sd(accts_amt, na.rm=TRUE),
            min_acct = min(accts_amt, na.rm=TRUE),
            max_acct = max(accts_amt, na.rm=TRUE))
tab2
```

```
## # A tibble: 2 x 6
##   income_less20k mean_acct med_acct sd_acct min_acct max_acct
##            <dbl>     <dbl>    <dbl>   <dbl>    <dbl>    <dbl>
## 1              0     8024.     1860  14975.        0    95000
## 2              1     3399.      500  10273.        0    94000
```

```r
plot2 <- dat |>
  ggplot(mapping = aes(x = accts_amt)) +
  geom_histogram() +
  geom_vline(xintercept = mean(dat$accts_amt, na.rm=T), color = "red") +
  geom_vline(xintercept = median(dat$accts_amt, na.rm=T), color = "blue") +
  facet_wrap(~income_less20k)

plot2
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```

```
## Warning: Removed 433 rows containing non-finite outside the scale range
## (`stat_bin()`).
```



We get more information about the distribution from the plot. All of the data is represented instead of just the summary statistics. If we are invested in our audience understanding this variable deeply, we likely want to show them the plots. If we just want to give our audience a broad idea of what these variables look like, we may only give them the basic summary statistics of central tendency and spread (usually the mean and standard deviation).