R for Statistics

Timothy Roeper

Spring 2019

Contents

Introduction

RStudio

- Console
  - Execute commands and see output.
- Scripts
  - Write a series of commands that can be executed all at once.
- Plots/Help/Files
  - See which files are in your working directory.
  - Plots are outputted here
  - Information on R functions
- Environment/History
  - View all the objects in your **workspace**.
  - View your entire command history

Installing packages

Installing packages

```
install.packages("tidyverse")
install.packages("AER")
```

You only have to install packages once, but every time you start a new R session, you need to load the packages you plan to use

```
library(tidyverse)
library(AER)
```

Loading Our Data

Later we will learn how to load our own data, but for now we will simply load some pre-built data from the AER package.

```
data("CPSSW8")
summary(CPSSW8)
```

```
##     earnings         gender           age              region
##  Min.   : 2.003   male  :34348   Min.   :21.00   Northeast:12371
##  1st Qu.:11.058   female:27047   1st Qu.:33.00   Midwest  :15136
##  Median :16.250                  Median :41.00   South    :18963
##  Mean   :18.435                  Mean   :41.23   West     :14925
##  3rd Qu.:23.558                  3rd Qu.:49.00
##  Max.   :72.115                  Max.   :64.00
##    education
##  Min.   : 6.00
##  1st Qu.:12.00
##  Median :13.00
##  Mean   :13.64
##  3rd Qu.:16.00
##  Max.   :20.00
```

The data we loaded is from the Current Population Survey, which interviews a large number of households in the U.S. every month. As you can see from the summary, the data set contains five variables: earnings, gender, age, region, and education. Earnings, age, and education are all numeric variables, meaning that each observation corresponds to a number. Gender and region are factors or categorical variables, which each observation corresponds to one of a small set of categories (ie South, Northeast, Midwest, and West for region).

View Data

```
View(CPSSW8)
```

R will only display the first 1,000 rows of a data set using this function. To specify a specific set of rows:

```
CPSSW8[6980:6982, ]
```

```
##      earnings gender age    region education
## 6980 18.02885 female  58 Northeast        16
## 6981 12.50000   male  62 Northeast        12
## 6982 17.30769 female  45 Northeast        12
```

View Data

To view the first 6, or last 6 rows, use `head()` or `tail()`.

```
tail(CPSSW8)
```

```
##        earnings gender age region education
## 61390 10.416667   male  24   West        11
## 61391  6.778846 female  51   West        13
## 61392  6.153846   male  52   West        14
## 61393 12.019231 female  44   West        12
## 61394 13.942307   male  39   West        12
## 61395 11.923077   male  24   West        10
```

```
head(CPSSW8)
```

```
##    earnings gender age region education
## 1 20.673077   male  31  South        14
## 2 24.278847   male  50  South        12
## 3 10.149572   male  36  South        12
## 4  8.894231 female  33  South        10
## 5  6.410256 female  56  South        10
## 6 16.666666 female  52  South        12
```

Individual columns

You can use the `$` to look at a single column:

```
summary(CPSSW8$earnings)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   2.003  11.058  16.250  18.435  23.558  72.115
```

```
head(CPSSW8$earnings)
```

```
## [1] 20.673077 24.278847 10.149572  8.894231  6.410256 16.666666
```

Add new columns

```
CPSSW8$age_squared <- CPSSW8$age^2
head(CPSSW8)
```

```
##    earnings gender age region education age_squared
## 1 20.673077   male  31  South        14         961
## 2 24.278847   male  50  South        12        2500
## 3 10.149572   male  36  South        12        1296
## 4  8.894231 female  33  South        10        1089
## 5  6.410256 female  56  South        10        3136
## 6 16.666666 female  52  South        12        2704
```

Delete Columns

```
CPSSW8$age_squared <- NULL
head(CPSSW8)
```

```
##    earnings gender age region education
## 1 20.673077   male  31  South        14
```

```
## 2 24.278847   male  50  South        12
## 3 10.149572   male  36  South        12
## 4  8.894231 female  33  South        10
## 5  6.410256 female  56  South        10
## 6 16.666666 female  52  South        12
```

Data Manipulation

`dplyr` functions

function	Purpose
filter()	Select rows based on criteria
arrange()	order rows
select()	choose columns
distinct()	unique values of column(s)
mutate() and transmute()	Create new columns
summarise()	Find a summary statistic
sample_n()	Take a random sample

Selecting rows using filter()

Using `filter()` to select observations

You can create a new data set with just men:

```
just_men <- CPSSW8 %>% filter(gender == "male")
head(just_men)
```

```
##   earnings gender age region education
## 1 20.67308   male  31  South        14
## 2 24.27885   male  50  South        12
## 3 10.14957   male  36  South        12
## 4 34.61538   male  30   West        16
## 5 11.05769   male  41  South        12
## 6 19.23077   male  37  South        13
```

You could also create a dataset of just women in their 40s who live in the South:

```
women_40s_south <- CPSSW8 %>% filter(gender == "female" &
                                     age >= 40 & age < 50 &
                                     region == "South")
head(women_40s_south)
```

```
##   earnings gender age region education
## 1 12.01923 female  44  South        13
## 2 10.20408 female  47  South         8
## 3 30.21978 female  45  South        12
## 4 15.68826 female  43  South        14
## 5 17.30769 female  41  South        18
## 6 14.42308 female  41  South        16
```

Exercise

Create a data set of men in their 20s who live in the northeast and have 16 years of education. Find their average earnings using `summary()`. For your reference below is the code I used to find women in their 40s in the south.

```
women_40s_south <- CPSSW8 %>% filter(gender == "female" &
                                     age >= 40 & age < 50 &
                                     region == "South")
```

Exercise Answer:

```
men_20s_NE_college <- CPSSW8 %>% filter(gender == "male" &
                                        age >= 20 & age <= 30 &
                                        region == "Northeast" &
                                        education == 16)
summary(men_20s_NE_college)
```

```
##     earnings         gender         age             region      education
##  Min.   : 3.846   male  :249   Min.   :22.0   Northeast:249   Min.   :16
##  1st Qu.:13.846   female:  0   1st Qu.:25.0   Midwest  :  0   1st Qu.:16
##  Median :18.462                Median :27.0   South    :  0   Median :16
##  Mean   :20.035                Mean   :26.8   West     :  0   Mean   :16
##  3rd Qu.:24.038                3rd Qu.:29.0                   3rd Qu.:16
##  Max.   :60.096                Max.   :30.0                   Max.   :16
```

Ordering observations with arrange()

You can order the observations by a column:

```
youngest_first <- CPSSW8 %>% arrange(age)
head(youngest_first)
```

```
##    earnings gender age    region education
## 1  4.395605 female  21     South        13
## 2 11.057693   male  21   Midwest        12
## 3  5.325444   male  21     South        12
## 4 17.393162   male  21 Northeast        12
## 5  8.653846 female  21 Northeast        14
## 6  2.403846   male  21   Midwest        13
```

```
tail(youngest_first)
```

```
##       earnings gender age    region education
## 61390 19.23077 female  64   Midwest        16
## 61391 18.75000   male  64   Midwest         8
## 61392 17.78846   male  64 Northeast        12
## 61393  7.39645   male  64   Midwest        12
## 61394 11.05769   male  64     South        12
## 61395 10.00000 female  64     South        11
```

Arrange the data by multiple variables

Order the observations

- First by education
- Second by earnings

```
educ_then_earnings <- CPSSW8 %>% arrange(education, earnings)
head(educ_then_earnings)
```

```
##   earnings gender age  region education
## 1 2.136752 female  42    West         6
## 2 2.403846 female  39 Midwest         6
## 3 2.428571 female  35   South         6
## 4 2.604167   male  31 Midwest         6
## 5 2.958580 female  45   South         6
## 6 3.205128 female  36   South         6
```

```
tail(educ_then_earnings)
```

```
##       earnings gender age    region education
## 61390 60.09615 female  54      West        20
## 61391 60.09615   male  49     South        20
## 61392 60.09615   male  51     South        20
## 61393 60.57692   male  61 Northeast        20
## 61394 65.38461   male  33      West        20
## 61395 67.85714 female  41     South        20
```

If you want to order variables in the opposite direction, simply put a `-` sign before the variable.

```
educ_then_earnings <- CPSSW8 %>% arrange(-education, -earnings)
head(educ_then_earnings)
```

```
##   earnings gender age    region education
## 1 67.85714 female  41     South        20
## 2 65.38461   male  33      West        20
## 3 60.57692   male  61 Northeast        20
## 4 60.09615   male  33 Northeast        20
## 5 60.09615 female  35      West        20
## 6 60.09615 female  54      West        20
```

```
tail(educ_then_earnings)
```

```
##       earnings gender age  region education
## 61390 3.205128 female  36   South         6
## 61391 2.958580 female  45   South         6
## 61392 2.604167   male  31 Midwest         6
## 61393 2.428571 female  35   South         6
## 61394 2.403846 female  39 Midwest         6
## 61395 2.136752 female  42    West         6
```

Selecting columns with select()

Using `select()` to select columns

`select()` allows one to select specific columns of the dataset

```
earnings_educ <- CPSSW8 %>% select(earnings, education)
head(earnings_educ)
```

```
##    earnings education
## 1 20.673077        14
## 2 24.278847        12
## 3 10.149572        12
## 4  8.894231        10
## 5  6.410256        10
## 6 16.666666        12
```

To get rid of a column and keep the rest, just put - in front of the variable.

```
no_region <- CPSSW8 %>% select(-region)
head(no_region)
```

```
##    earnings gender age education
## 1 20.673077   male  31        14
## 2 24.278847   male  50        12
## 3 10.149572   male  36        12
## 4  8.894231 female  33        10
## 5  6.410256 female  56        10
## 6 16.666666 female  52        12
```

`select()` helper functions

Three well named functions:

1. `starts_with()`
2. `ends_with()`
3. `contains()`

Example

```
ends_with_on <- CPSSW8 %>% select(ends_with("on"))
head(ends_with_on)
```

```
##   region education
## 1  South        14
## 2  South        12
## 3  South        12
## 4  South        10
## 5  South        10
## 6  South        12
```

Finding summary statistics with summarise()

Suppose we wanted to find the average earnings, standard deviation of earnings, and the 5th and 95 percentiles.

```
CPSSW8 %>%  summarise("Avg_Earnings" = mean(earnings),
                      "Std_dev_earnings" = sd(earnings),
                      "95th percentile" = quantile(earnings, probs = .95),
                      "5th percentile" = quantile(earnings, probs = .05))
```

```
##   Avg_Earnings Std_dev_earnings 95th percentile 5th percentile
## 1     18.43512         10.12717        38.46154           6.25
```

Functions to use with `summarise()`

function	Result
`mean(x)`	mean of a sample, $\bar{X} = \sum_{i=1}^{n} X_i/n$
`sd(x)`	standard deviation of a sample $s_x = \sqrt{\sum_{i=1}^{n}(X_i - \bar{X})^2/(n-1)}$
`var(x)`	variance of a sample $s_x^2 = \sum_{i=1}^{n}(X_i - \bar{X})^2/(n-1)$
`cov(x, y)`	Sample covariance of two variables $s_{x,y} = \sum_{i=1}^{n}(X_i - \bar{X})(Y_i - \bar{Y})/(n-1)$
`cor(x, y)`	Correlation coefficient $\rho_{x,y} = s_{x,y}/(s_x \times s_y)$
`quantile(x, probs)`	Percentile of variable `x`
`length(x)`	Number of observations in the variable `x`
`sum(x)`	Sum of all the observations in `x`

Summarising multiple variables with multiple functions

Suppose you want the mean, median, and standard deviation of the earnings, education, age and education. You could use `summarise()` to do this, but it would be tedious and repetitive. Much more efficient is to use `summarise_at()`:

```
CPSSW8 %>% summarise_at(vars(earnings, education, age), funs(mean, median, sd))
```

```
##   earnings_mean education_mean age_mean earnings_median education_median
## 1      18.43512       13.64432  41.2314           16.25               13
##   age_median earnings_sd education_sd   age_sd
## 1         41    10.12717     2.460676 10.58667
```

Another example `summarise_at()`

Suppose we want the 25th and 75th, percentiles of each variable for men and women, and we want to name each one "low" and "high"

```
CPSSW8 %>% summarise_at(vars(earnings, education, age),
                        funs("low" = quantile(., probs = .25),
                             "high" = quantile(., probs = .75)))
```

```
##   earnings_low education_low age_low earnings_high education_high age_high
## 1     11.05769            12      33      23.55769             16       49
```

Alternative: `summarise_all()`

Instead of choosing columns to summarise using `summarise_at()`. Choose columns you want using `select()`, and then use `summarise_all()`.

```
vars_for_summ <- CPSSW8 %>% select(earnings, education, age)
vars_for_summ %>% summarise_all(funs(mean, median, sd))
```

```
##   earnings_mean education_mean age_mean earnings_median education_median
## 1      18.43512       13.64432  41.2314           16.25               13
##   age_median earnings_sd education_sd   age_sd
## 1         41    10.12717     2.460676 10.58667
```

Exercise using `filter()` and `summarise()`

Exercise

Find the average earnings and average age of women with 16 years of education.

Example

Finding the average earnings and education of 60 year old men.

```
men_age60 <- CPSSW8 %>% filter(gender == "male" & age == 60)
men_age60 %>% summarise("Avg_Earnings" = mean(earnings), "Avg_Educ"= mean(education))
```

```
##   Avg_Earnings Avg_Educ
## 1     20.93144 13.69947
```

Exercise using `filter()` and `summarise()`: Answer

Exercise Find the average earnings and average age of women with 16 years of education. For your reference, below is an example of finding the average earnings and education of 60 year old men.

Exercise Answer

```
college_women <- CPSSW8 %>% filter(gender == "female" & education == 16)
college_women %>% summarise("Avg_Age" = mean(age), "Avg_Earnings" = mean(earnings))
```

```
##    Avg_Age Avg_Earnings
## 1 40.28928     20.34774
```

Exercise using `filter()` and `summarise()`: Alternative Answer

Exercise Find the average earnings and average age of women with 16 years of education. For your reference, below is an example of finding the average earnings and education of 60 year old men.

Alternative Answer

```
CPSSW8 %>% filter(gender == "female" & education == 16) %>%
  summarise("Avg_Age" = mean(age), "Avg_Earnings" = mean(earnings))
```

```
##    Avg_Age Avg_Earnings
## 1 40.28928     20.34774
```

Manipulating data by sub-group using group_by()

Understanding what %>% does

You most likely have noticed each of these `dplyr` examples involve `%>%`. This expression roughly translates to "and then". So for example

```
CPSSW8 %>% filter(gender == "male") %>% select(gender, earnings) %>% head()
```

Roughly translates to: Take the CPSSW8 dataset **and then** keep only those observations that are male **and then** select the gender and earnings columns **and then** display the first six observations.

Finding statistics by group using `group_by()`

```
CPSSW8 %>% group_by(gender) %>% summarise("Avg_Earnings"= mean(earnings))
```

```
## # A tibble: 2 x 2
##   gender Avg_Earnings
##   <fct>         <dbl>
## 1 male           20.1
## 2 female         16.3
```

```
CPSSW8 %>% group_by(region) %>% summarise("Avg_Earnings"= mean(earnings))
```

```
## # A tibble: 4 x 2
##   region    Avg_Earnings
##   <fct>            <dbl>
## 1 Northeast         19.8
## 2 Midwest           18.1
## 3 South             17.6
## 4 West              18.6
```

Grouping by multiple variables

What if you wanted to divide the data into groups by both age and region, and find the mean earnings and education for each group?

```
CPSSW8 %>% group_by(gender, region) %>%
  summarise("Avg_Earnings"= mean(earnings), "Avg_Educ" = mean(education))
```

```
## # A tibble: 8 x 4
## # Groups:   gender [?]
##   gender region    Avg_Earnings Avg_Educ
##   <fct>  <fct>            <dbl>    <dbl>
## 1 male   Northeast         21.5     13.8
## 2 male   Midwest           19.9     13.7
## 3 male   South             19.3     13.4
## 4 male   West              20.1     13.4
## 5 female Northeast         17.7     14.0
## 6 female Midwest           15.9     13.9
## 7 female South             15.7     13.6
## 8 female West              16.5     13.7
```

Outputting Results to Excel

To output a summarization table to excel:

```
table <- CPSSW8 %>% group_by(gender, region) %>%
  summarise("Avg_Earnings"= mean(earnings), "Avg_Educ" = mean(education))
write_excel_csv(table, "my_table_output.csv")
```

Using `mutate()` to create new columns (variables) in a dataset

Suppose hypothetically you wanted to create a new variable called "experience" that is equal to a person's age, minus their education level minus 6 (most people begin school around age 6). To create a new dataset that included this variable, you would have to use `mutate()`

```
new_CPS <- CPSSW8 %>% mutate("experience" = age - education - 6)
head(new_CPS)
```

```
##    earnings gender age region education experience
## 1 20.673077   male  31  South        14         11
## 2 24.278847   male  50  South        12         32
## 3 10.149572   male  36  South        12         18
## 4  8.894231 female  33  South        10         17
## 5  6.410256 female  56  South        10         40
## 6 16.666666 female  52  South        12         34
```

Using `mutate()` for summary statistics

Suppose you wanted to find the z-score of earnings for each observation

$$\text{z-score of observation} = \frac{\text{observation} - \text{mean of all observations}}{\text{standard deviation of all observations}}$$

One way to do this is to create two new columns, the mean of earnings, and the standard deviation:

```
new_cols <- CPSSW8 %>% mutate("mean_earn" = mean(earnings), "sd_earn" = sd(earnings),
                              "z-score" = (earnings - mean_earn) / sd_earn)
head(new_cols)
```

```
##    earnings gender age region education mean_earn  sd_earn    z-score
## 1 20.673077   male  31  South        14  18.43512 10.12717  0.2209858
```

```
## 2 24.278847   male  50  South        12  18.43512 10.12717  0.5770349
## 3 10.149572   male  36  South        12  18.43512 10.12717 -0.8181498
## 4  8.894231 female  33  South        10  18.43512 10.12717 -0.9421075
## 5  6.410256 female  56  South        10  18.43512 10.12717 -1.1873857
## 6 16.666666 female  52  South        12  18.43512 10.12717 -0.1746242
```

Combining `mutate()` with `group_by()`

We wanted to find each observation's z-score given the person's gender and level of education.

```
new_cols <- CPSSW8 %>% group_by(gender, education) %>%
  mutate("mean_earn" = mean(earnings), "sd_earn" = sd(earnings),
         "z-score" = (earnings - mean_earn) / sd_earn)
head(new_cols)
```

```
## # A tibble: 6 x 8
## # Groups:   gender, education [4]
##   earnings gender   age region education mean_earn sd_earn `z-score`
##      <dbl> <fct>  <int> <fct>      <int>     <dbl>   <dbl>     <dbl>
## 1    20.7  male      31 South         14     20.4     9.26    0.0307
## 2    24.3  male      50 South         12     16.8     8.29    0.906
## 3    10.1  male      36 South         12     16.8     8.29   -0.800
## 4     8.89 female    33 South         10      9.72    4.76   -0.174
## 5     6.41 female    56 South         10      9.72    4.76   -0.695
## 6    16.7  female    52 South         12     12.8     6.28    0.610
```

Testing for Statistical Significance

t test of earnings by gender

```
t.test(earnings ~ gender, data = CPSSW8)
```

```
##
##  Welch Two Sample t-test
##
## data:  earnings by gender
## t = 47.243, df = 61085, p-value < 2.2e-16
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  3.592782 3.903797
## sample estimates:
##   mean in group male mean in group female
##             20.08639             16.33810
```

- The differences we observe in average earnings between men and women is statistically significant at the .001 level.

Analysis of variance of earnings by region

```
earnings_by_region <- aov(earnings ~ region, data = CPSSW8)
summary(earnings_by_region)
```

```
##                Df  Sum Sq Mean Sq F value Pr(>F)
## region          3   37147   12382   121.4 <2e-16 ***
## Residuals   61391 6259397     102
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

- The differences in earnings we observed in earnings across the four regions is statistically significant at the .001 level.

Regression

Estimating a Regression

$$\text{Earnings} = \beta_0 + \beta_1 \text{Education} + \beta_2 \text{Male} + \epsilon$$

```
earnings_reg <- lm(earnings ~ education + gender, data = CPSSW8)
coeftest(earnings_reg)
```

```
##
## t test of coefficients:
##
##               Estimate Std. Error t value  Pr(>|t|)
## (Intercept)  -4.104045   0.204343 -20.084 < 2.2e-16 ***
## education     1.787144   0.014670 121.819 < 2.2e-16 ***
## genderfemale -4.188481   0.072714 -57.602 < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

R can handle putting categorical variables, such as `gender` in the regression equation.

Interpreting the output `summary()`

```
summary(earnings_reg)
```

```
##
## Call:
## lm(formula = earnings ~ education + gender, data = CPSSW8)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -29.441  -5.939  -1.413   4.347  50.046
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  -4.10405    0.20434  -20.08   <2e-16 ***
## education     1.78714    0.01467  121.82   <2e-16 ***
## genderfemale -4.18848    0.07271  -57.60   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 8.934 on 61392 degrees of freedom
## Multiple R-squared:  0.2219, Adjusted R-squared:  0.2218
## F-statistic:  8752 on 2 and 61392 DF,  p-value: < 2.2e-16
```

Other R Resources

- DataCamp (not free, but worth the investment) https://www.datacamp.com/courses
- R for Data Science http://r4ds.had.co.nz/
- R bloggers http://www.r-bloggers.com/