ECON 340
Economics Research Methods

Div Bhagia

Lecture 7: Data Analysis in R

So far

```
# Load Packages
library(tidyverse)

# Import data
data <- read.csv("caschool.csv")
```

You can clear your environment before starting by using the broom on the top-right. Or add `rm(list=ls())` command on top of your R-script.

Tabulating Variables

- Variable `gr_span` reports the grade span of a school district (K-6 or K-8)

```
table(data$gr_span)
```

```
##
## KK-06 KK-08
##    61   359
```

- So 61 school districts go up to grade 6 while 359 go up to grade 8

Dplyr Syntax

- `dplyr` is a `TidyVerse` package that provides several useful functions for data manipulation
- However, `dplyr` uses slightly different syntax from base R.
- One key operator utilized by this package is the pipe operator %>%
- You can use shortcut Cmd + Shift + M (Mac) and Ctrl + Shift + M (Windows) for %>%
- You can think of this operator as standing for “then” in the code

Dplyr Syntax

For example, to tabulate data:

```
data %>% count(gr_span)
```

```
##   gr_span   n
## 1   KK-06  61
## 2   KK-08 359
```

Some Useful `dplyr` Functions

- `mutate()` adds new variables that are functions of existing variables
- `select()` picks variables based on their names
- `filter()` picks cases based on their values
- `summarise()` reduces multiple values down to a single summary
- `arrange()` changes the ordering of rows
- `group_by()` performs subsequent calculations within-group (and ungroup() when done)

Select Variables

```
data %>% select(computer, enrl_tot)
```

```
##     computer enrl_tot
## 1         67      195
## 2        101      240
## 3        169     1550
## 4         85      243
## 5        171     1335
## 6         25      137
## 7         28      195
## 8         66      888
## 9         35      379
```

Finding Correlation

```
# Base R
cor(data$computer, data$enrl_tot)
```

```
## [1] 0.9288821
```

```
# Tidy way
data %>% select(computer, enrl_tot) %>% cor()
```

```
##              computer  enrl_tot
## computer   1.0000000 0.9288821
## enrl_tot   0.9288821 1.0000000
```

Filter Observations

```
data %>% select(gr_span, computer) %>%
  filter(gr_span=="KK-06")
```

```
##    gr_span computer
## 1    KK-06        0
## 2    KK-06      742
## 3    KK-06      324
## 4    KK-06      669
## 5    KK-06      196
## 6    KK-06      560
## 7    KK-06     1048
## 8    KK-06      505
```

And and *Or* in R

To select schools in Orange county with enrollment over 5000

```
data1 <- data %>%
  filter(county=="Orange" & enrl_tot>=5000)
```

To select schools that are either in Orange country or in LA county

```
data2 <- data %>%
  filter(county=="Orange" | county=="Los Angeles")
```

Summarize Variables

```
# Calculating mean
data %>% summarise(mean(computer))
```

```
##   mean(computer)
## 1       303.3833
```

```
# Standard deviation and median
data %>% summarise(sd = sd(computer),
                   med = median(comp_stu))
```

```
##         sd       med
## 1 441.3413 0.1254644
```

Creating New Variables

```
data <- data %>%
  mutate(log_enrl = log(enrl_tot))
```

- The code takes `data` and adds a new column `log_enrl`, which is the log of `enrl_tot`
- It then updates the original data with this new column.

Creating New Variables

```
data <- data %>%
  mutate(hcomp = ifelse(comp_stu>=median(comp_stu),1,0))
```

- Syntax: `ifelse(test_expression, x, y)`
- The returned vector has element from `x` if the corresponding value of `test_expression` is `TRUE` and `y` if it is `FALSE`
- So here `hcomp` takes value 1 whenever computers per student are above the median, and 0 otherwise. What should be the output from `mean(data$hcomp)`?

Combining `group_by()` and `summarise()`

```
data %>%
  group_by(hcomp) %>%
  summarise(mean(comp_stu))
```

```
## # A tibble: 2 x 2
##   hcomp `mean(comp_stu)`
##   <dbl>            <dbl>
## 1     0           0.0881
## 2     1           0.184
```

Excercise for you

Find the county with the highest average number of computers per student (`comp_stu`) (Hint: Use `group_by(county)` and `summarise()`)