

Lab 1: Introduction to R and RStudio

Overall Goals

- Start RStudio.
- Learn the different screen areas in RStudio.
- Load a built-in dataset.
- Look at the help for the dataset.
- View the dataset in tabular form.
- Perform basic operations on dataset
 - Number of rows
 - Sort rows based on column
 - Simple graphs
 - Simple numerical summaries

Specific R commands learned

- `data` for loading built-in data.
- `?` for asking for information on a function or data.
- View for a tabular look at dataset.
- `nrow` for number of rows.
- ...

Lab 1

- Start RStudio (TODO: weblink)
- Panes:
 - Console
 - Environment/History
 - Outputs/Help/File
 - Document (Later)
- Create new project for counties
- Use `library(...)`, `data(...)` to load the package that has the counties data (need to create it).
- (Not for student use: `read.table("https://skiadas.github.io/AppliedStatsCourse/site/datasets/countyComplete.txt", sep = ";")`)
- See the counties set on the right, click on it
- Data view comes up -> document pane (also View command)
- Sort view to find 3 most/least populous counties based on 2010
- How many counties? (3143, on left side)
- number of counties per state:
 - Numerically: `tally(~state, data=counties)`
 - * Use upper arrow to repeat, then add: `%>% sort()`

- * Then add: `%>% barchart()`
- Graphically: `bargraph(~state, data=counties, horizontal = TRUE)`
- * zoom graph
- Population of each state:
 - `sum(~pop2010|state, data=counties) %>% sort() %>% barchart()`
 - Maybe skip for later?
- Histogram of population by county:
 - `histogram(~pop2010, data=counties)`
 - `histogram(~pop2010, data=counties %>% filter(pop2010 <= 2e6))`
 - Add “breaks = 40”
 - `favstats (~pop2010, data=counties)`
- Histogram of percent of female population:
 - `histogram(~female, data=counties, breaks=40)`
 - `favstats (~female, data=counties)`
 - `bwplot(state ~female, data=counties)`
 - * Find lowest median, highest median
 - * Find state with lowest outlier
 - Work with black and/or asian states
- Compare populations 2000, 2010:
 - `xyplot(pop2010~pop2000, data=counties)`
 - `xyplot(pop2010~pop2000, data=counties %>% filter(pop2000 < 2e6))`
 - `tally (~name, data=counties) %>% sort() %>% tail(10)`
- County names:
 - `tally (~name, data=counties) %>% sort() %>% tail(10)`
- Union counties:
 - `counties %>% filter(name=="Union County") %>% select(state)`
 - `summary`

Lab 3 or something: Introduce markdown

TODO: Consider using the `%in%` operator somewhere

To find percent of whites in state: - white count per county - add white counts across states - divide white count by state pop count

```
counties %>% transmute(state = state, pop = pop2010, whitepop = pop2010 * white / 100) %>%
group_by(state) %>% summarize(pop = sum(pop), whitepop = sum(whitepop)) %>% transmute(state =
state, white = whitepop/pop * 100) %>% arrange(white)
```

Easier: `sum(whitepop2010/100~state, data=counties)/sum(pop2010~state, data=counties) 100`

```
sum(~black*pop2010/100, data=counties %>% filter(!is.na(black)))/sum(~pop2010, data=counties
%>% filter(!is.na(black)))
```