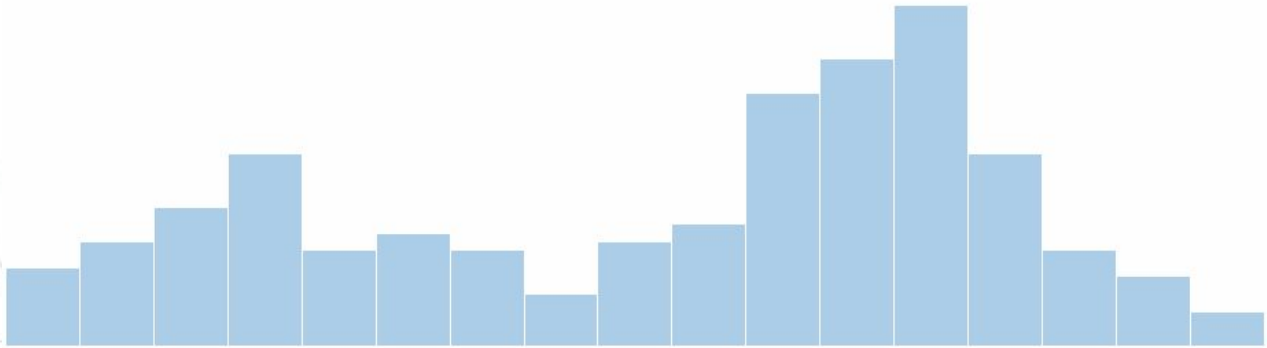
Introduction to

Jamie Ford, NISD
Paulina Cano, CI:NOW

What is R?

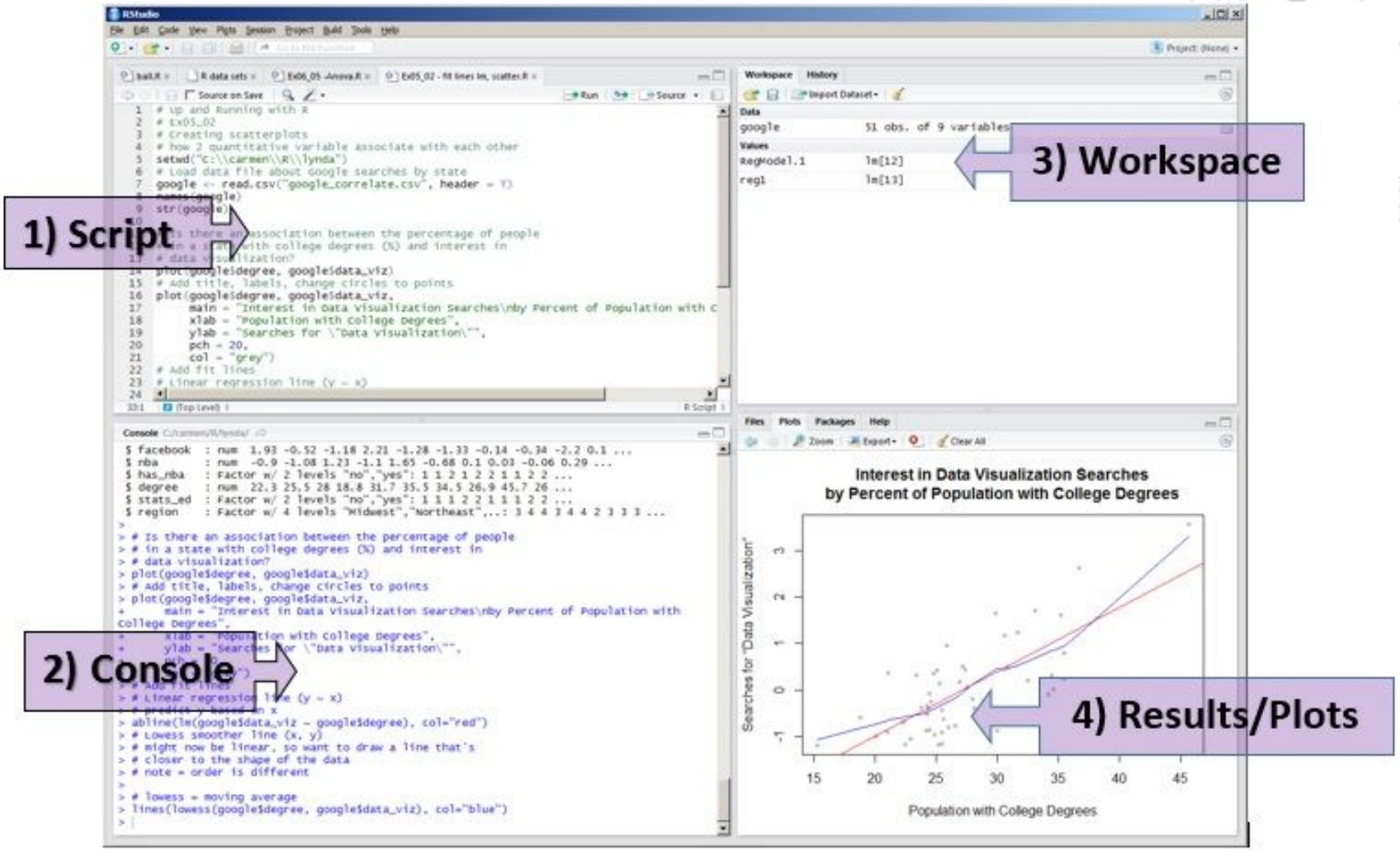
- One of the most widely used data analysis software, used by statisticians, analysts, data scientists, etc.
- Powerful statistical programming language with unique data visualizations
- More than 14,000 libraries approved on CRAN (plus others on GitHub, etc.)
- R has more than 2 million users worldwide and is growing rapidly
- R can be downloaded online for free along with **Rstudio**



How does R compare to other statistical software?

	SPSS	STATA	SAS	R
Ease of learning	✓✓	✓	✓	✗
Good user interface	✓✓	✓	✓	✗
Programming Capabilities	✗	✓	✓	✓✓
Support from company	✓	✓	✓	✗
Price	✗	✗	✗	✓
Advanced Visualization capabilities	✗	✗	✗	✓✓
Handle complex models	✗	✓✓	✓✓	✓✓
Handle large sets of data	✓	✓✓	✓✓	✓✓

Rstudio


1) Script
2) Console
3) Workspace
4) Results/Plots
Interest in Data Visualization Searches
by Percent of Population with College Degrees
Searches for "Data Visualization"
Population with College Degrees


R
LIVE

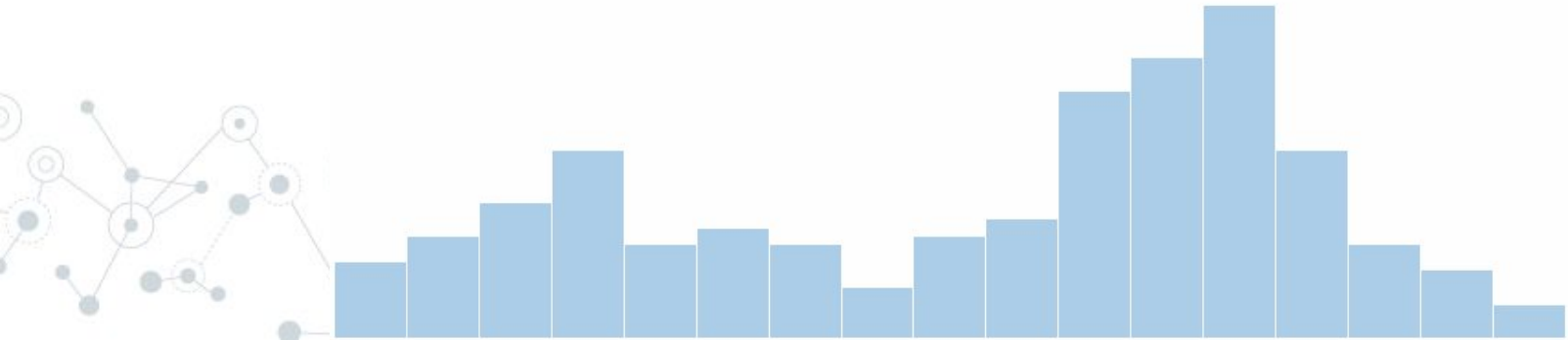
Is R right for you?

Advantages

- Open-Source
- Community support
- Automation
- Flexibility
- Dynamic output

Disadvantages

- Steep learning curve
- Programming and capacity limitations when compared to Python or similar
- Some libraries may not be updated
- Not standardized



Using R to Work with Census Data

R allows you to download census data directly.

Steps:

1. Request a free Census Bureau API key https://api.census.gov/data/key_signup.html

2. Download a few packages: **tigris** (shapefiles), **tidycensus** (Census and ACS data with feature geometries) and **sf**, (simple features is use to represent geographic vector data).

3. Load variables of interest.

```
nc_pop <-
  get_acs(geography = "county",
          variables = "B01003_001",
          state = "NC",
          geometry = TRUE)
```

4. Your are now ready to interact with the data

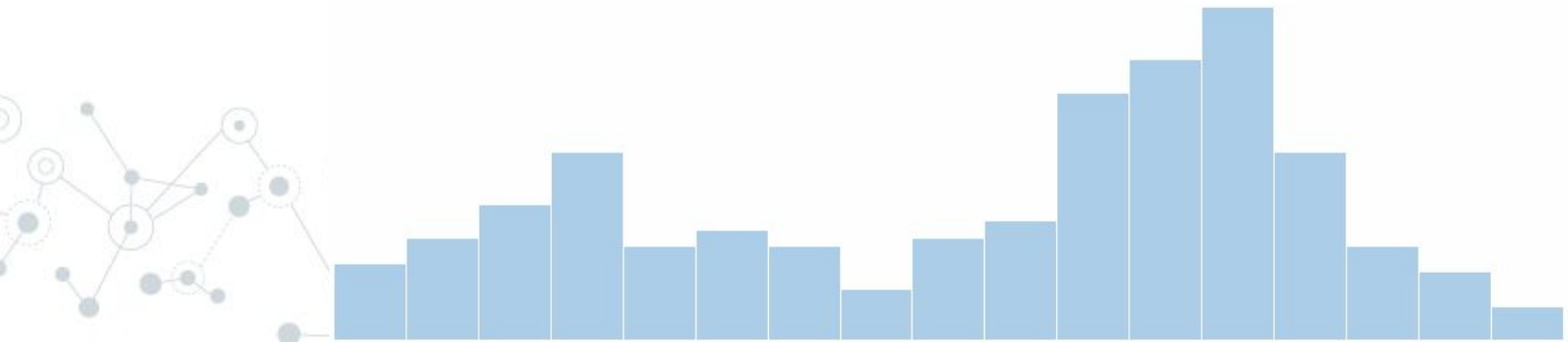
United States™ Census Bureau

Continuation of Census Data and R

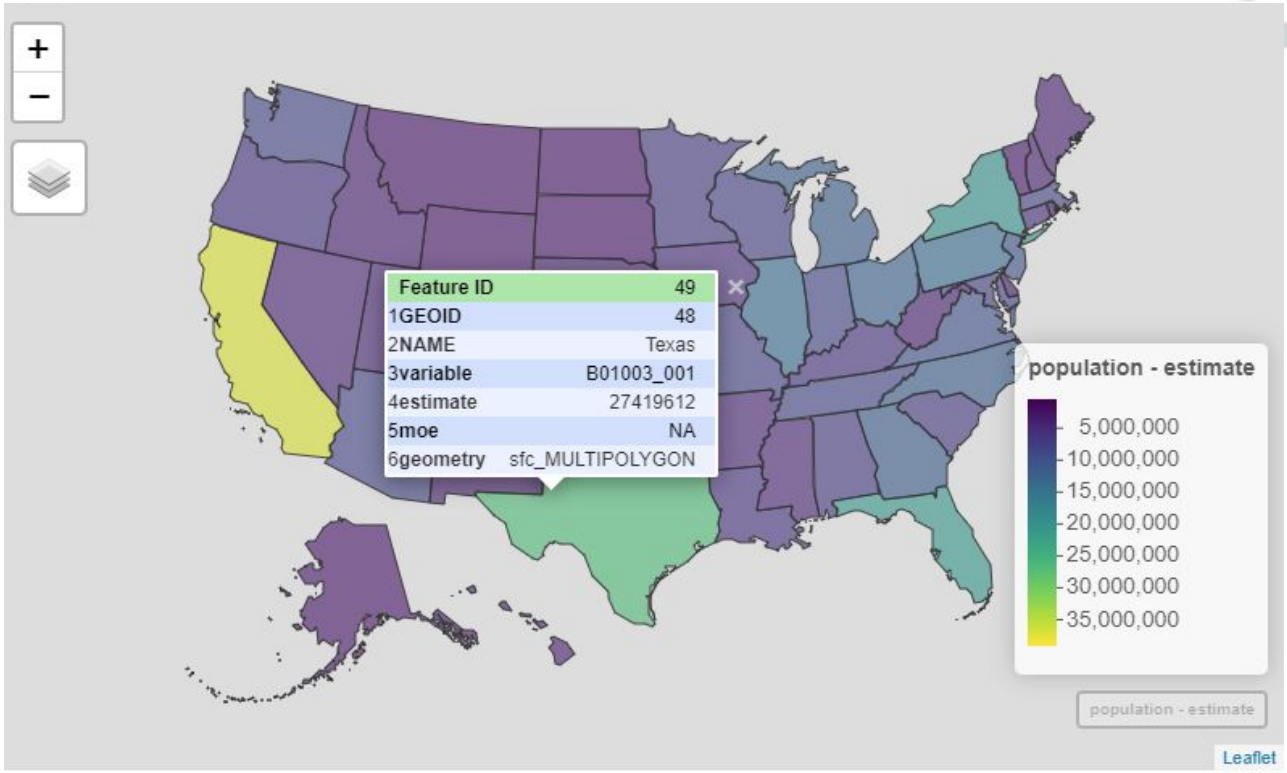
If we install the **leafview** and **mapview** packages, we can visualize the data:

```
population <- get_acs(geography = "state",
                      variables = "B01003_001",
                      geometry = TRUE,
                      shift_geo = TRUE)
```

```
mapviewOptions(legend.pos = "bottomright")
mapviewOptions(leafletWidth = 800)
#mapviewOptions()
#mapviewOptions(default = TRUE)
mapview(population, zcol = "estimate", native.crs = TRUE, crs = 5070)
```



Example Output



https://map-rfun.library.duke.edu/02_choropleth.html

Using R for Survey Data - Jamie

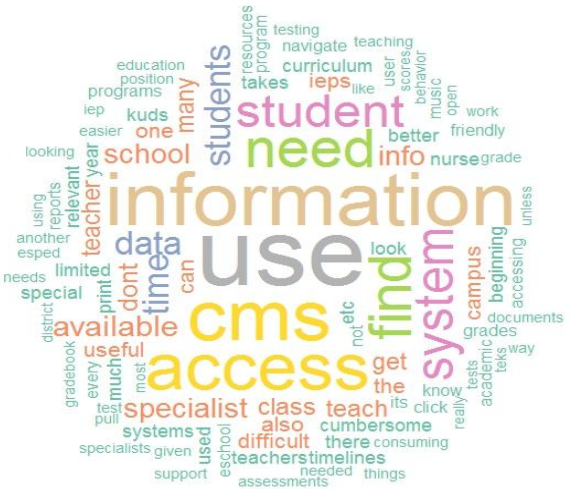
Using libraries **gmodels** and **wordcloud**, R can analyze frequencies, cross-tabs and text.

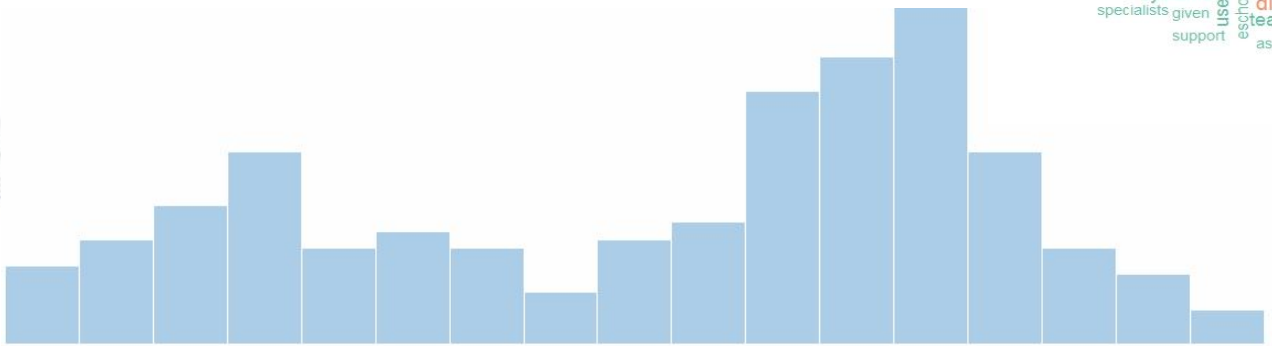
```
> CrossTable(survey$Q7,survey$Q2, prop.t=FALSE, prop.r=TRUE, prop.c=FALSE, prop.chisq=FALSE, format="SPSS",digits=1)

   Cell Contents
|-------------------------|
|                   Count |
|             Row Percent |
|-------------------------|

Total Observations in Table:  45

             | survey$Q2
   survey$Q7 |         A |         B |         C |         D | Row Total |
-------------|-----------|-----------|-----------|-----------|-----------|
           A |         1 |         7 |         2 |         0 |        10 |
             |     10.0% |     70.0% |     20.0% |      0.0% |     22.2% |
-------------|-----------|-----------|-----------|-----------|-----------|
           B |        12 |        14 |         6 |         0 |        32 |
             |     37.5% |     43.8% |     18.8% |      0.0% |     71.1% |
-------------|-----------|-----------|-----------|-----------|-----------|
           C |         2 |         0 |         1 |         0 |         3 |
             |     66.7% |      0.0% |     33.3% |      0.0% |      6.7% |
-------------|-----------|-----------|-----------|-----------|-----------|
Column Total |        15 |        21 |         9 |         0 |        45 |
-------------|-----------|-----------|-----------|-----------|-----------|
```





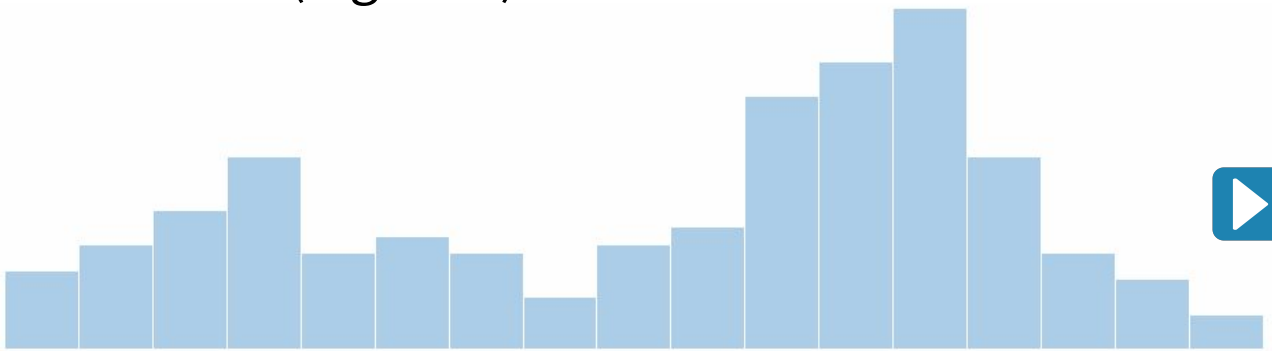
Data Manipulation

Derive new variables

Join multiple data sets of data together

Create summaries of your dataset

Pull information directly from websites and/or public data sets (e.g. ACS)



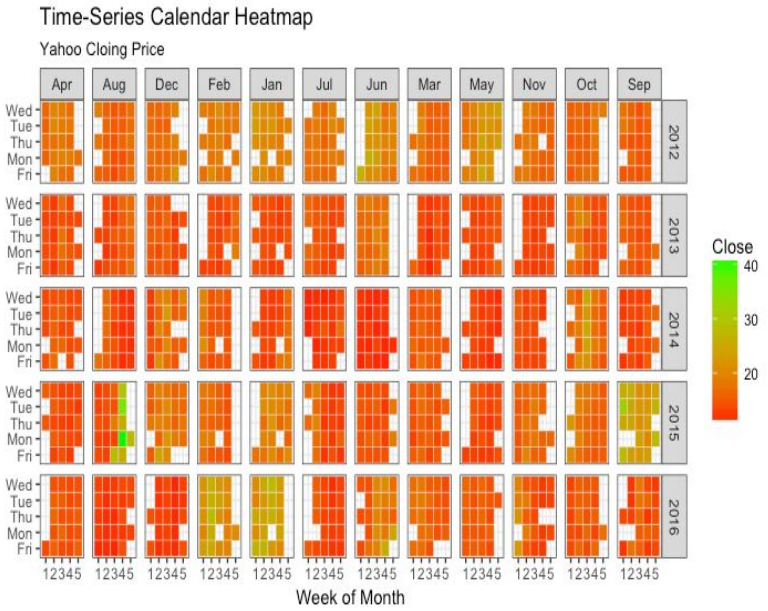
R-LIVE


R
LIVE

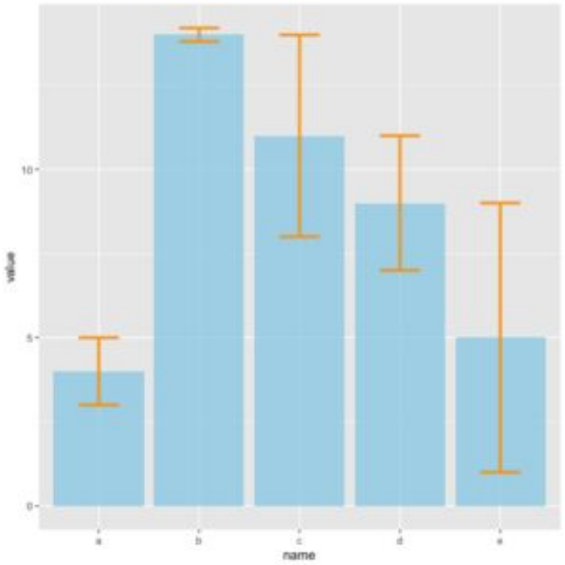
Data Visualization

R has several packages that enable visualizing data:

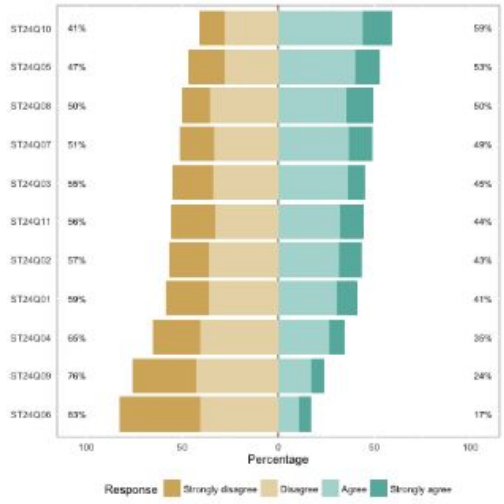
- BaseR
- Ggplot2
- Leaflet (interactive)
- Plotly (interactive)
- Other specialized (various models, EDA, GIS, network, etc.)

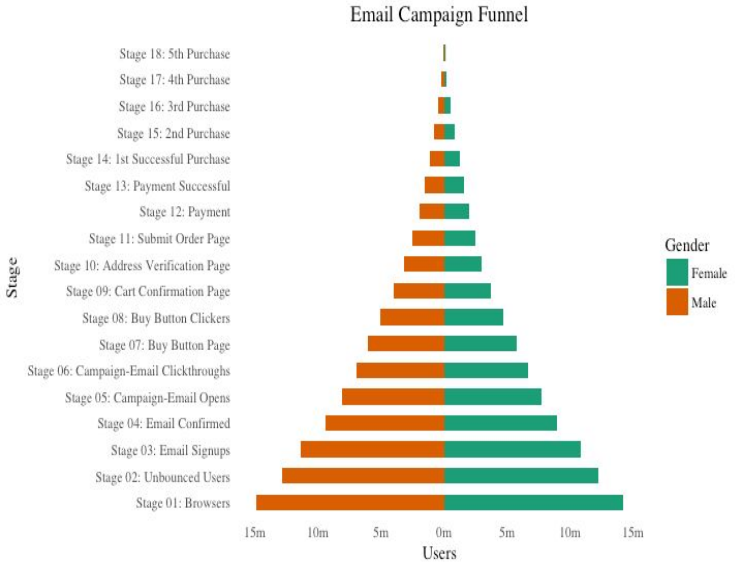


Bar Plots

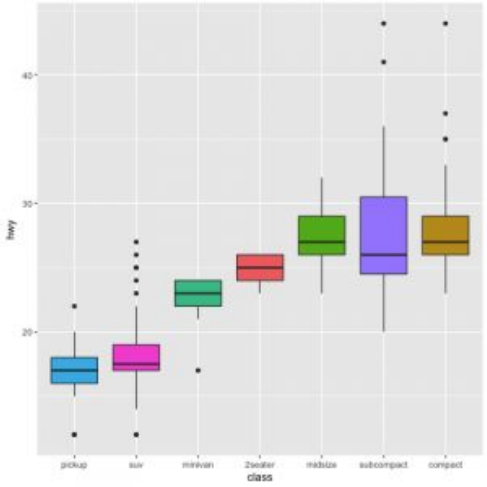


#4 Error bars on barplot

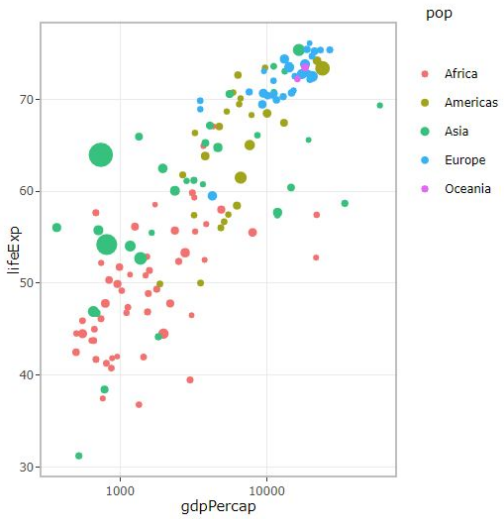




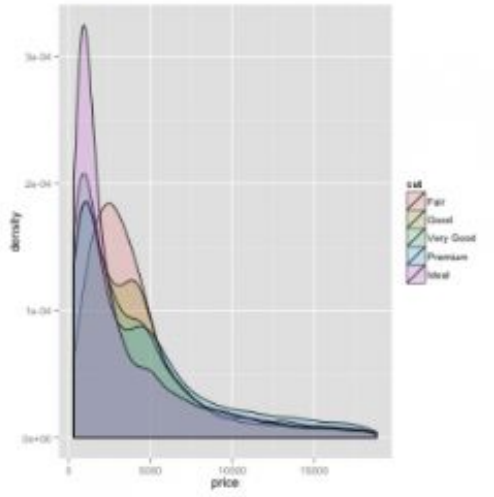
Data Visualization



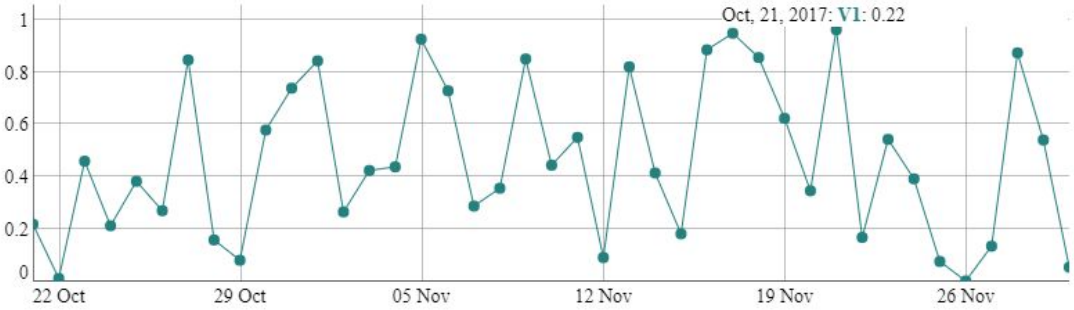
Boxplots



Bubble Graphs

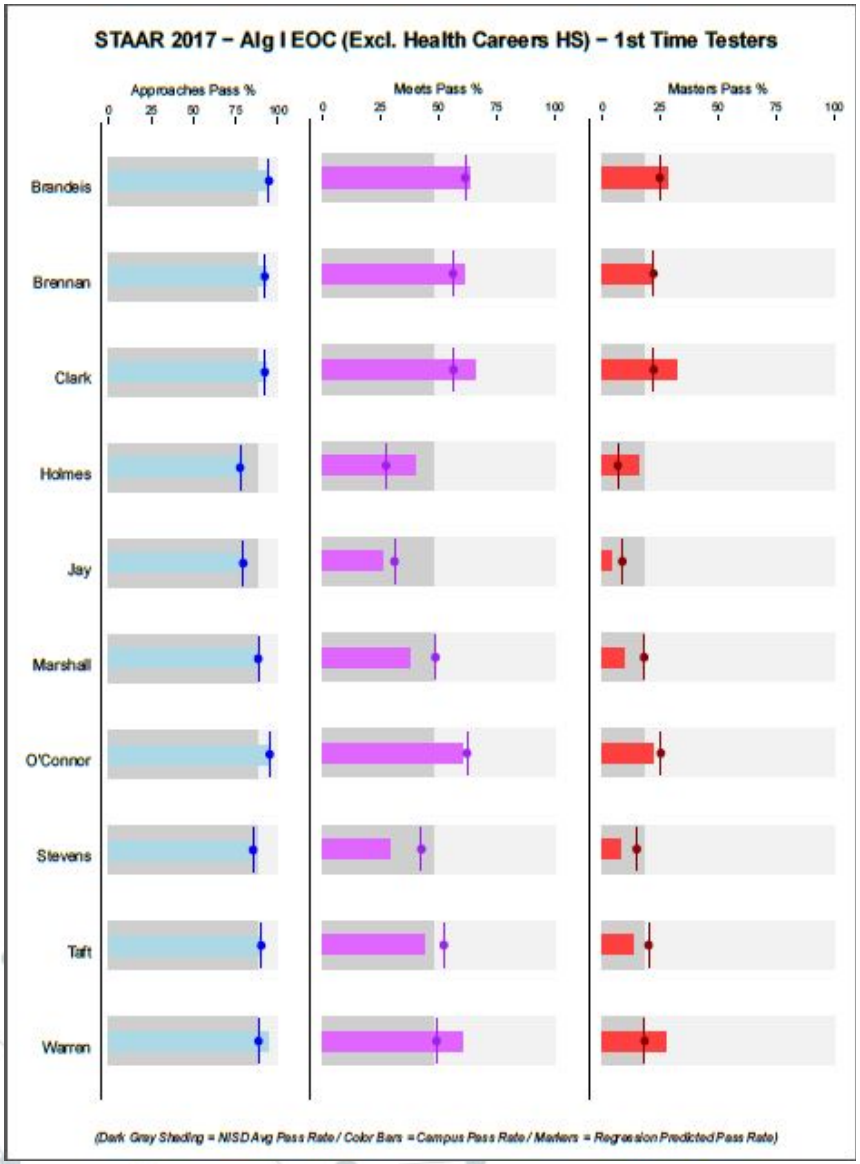


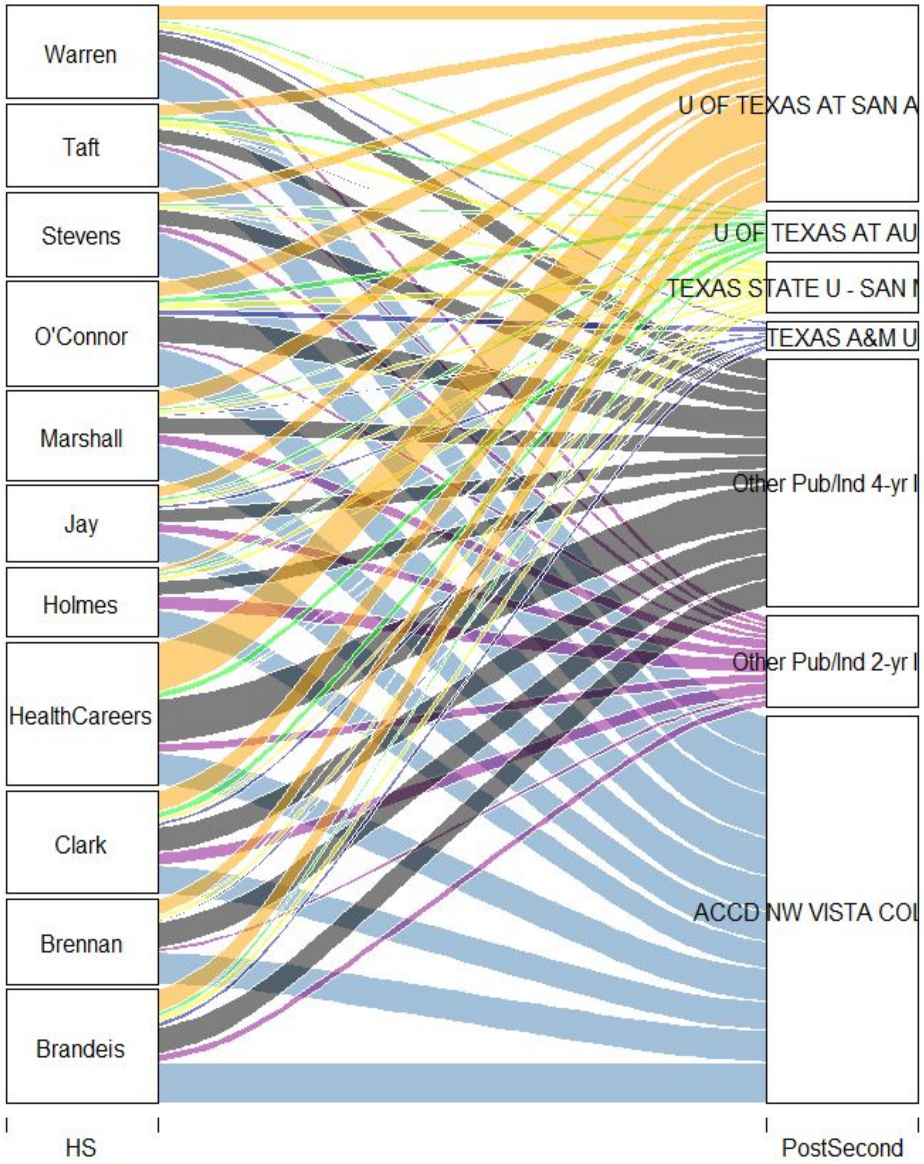
Density Graphs



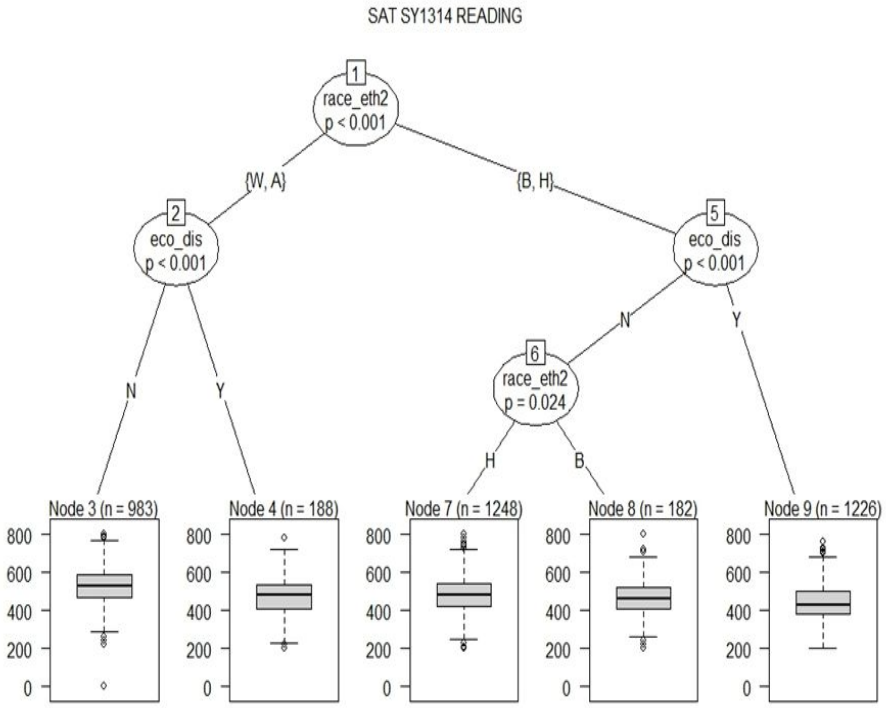
Time Series Graph

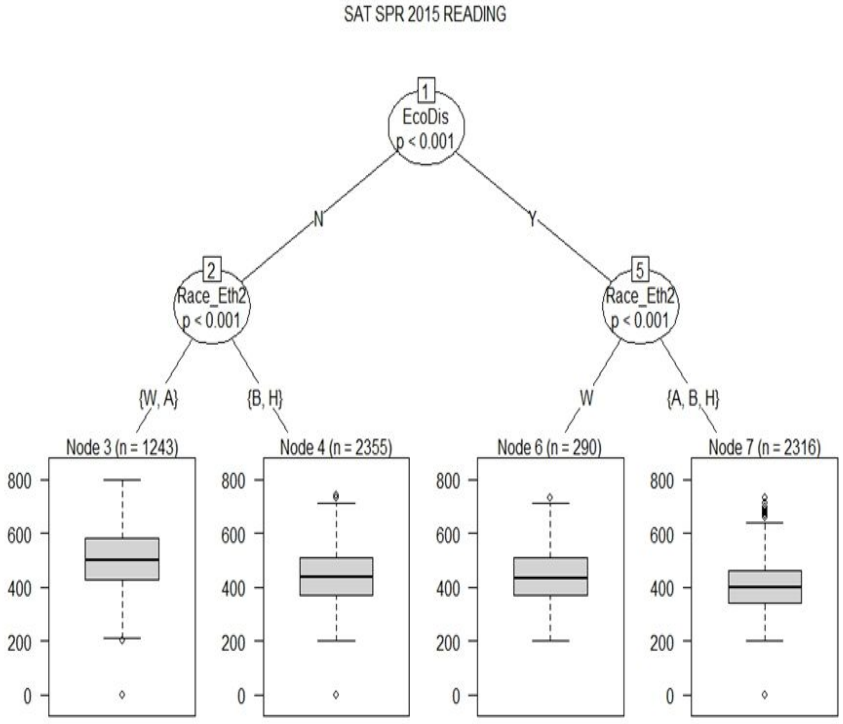
EXAMPLES OF PROJECTS
Visualizations of STAAR Results & College Enrollment Flows





EXAMPLES OF PROJECTS
Decision Trees using CTREE



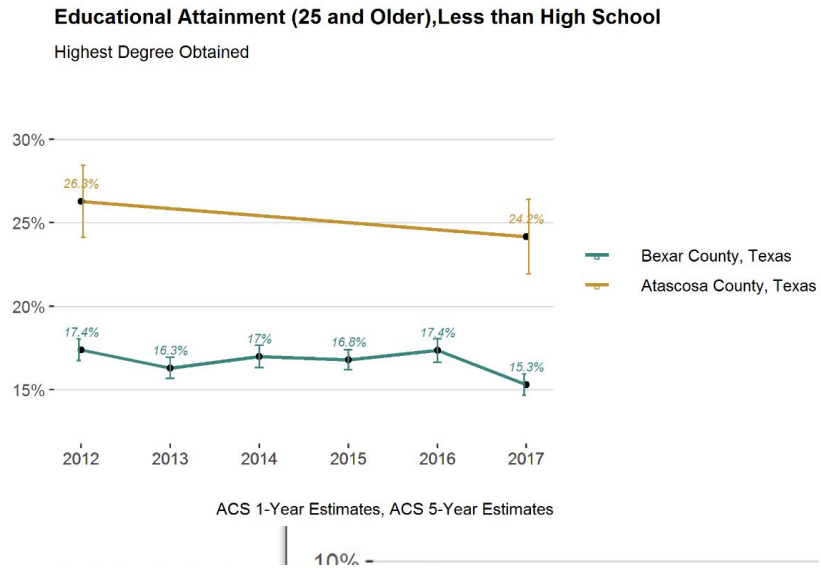


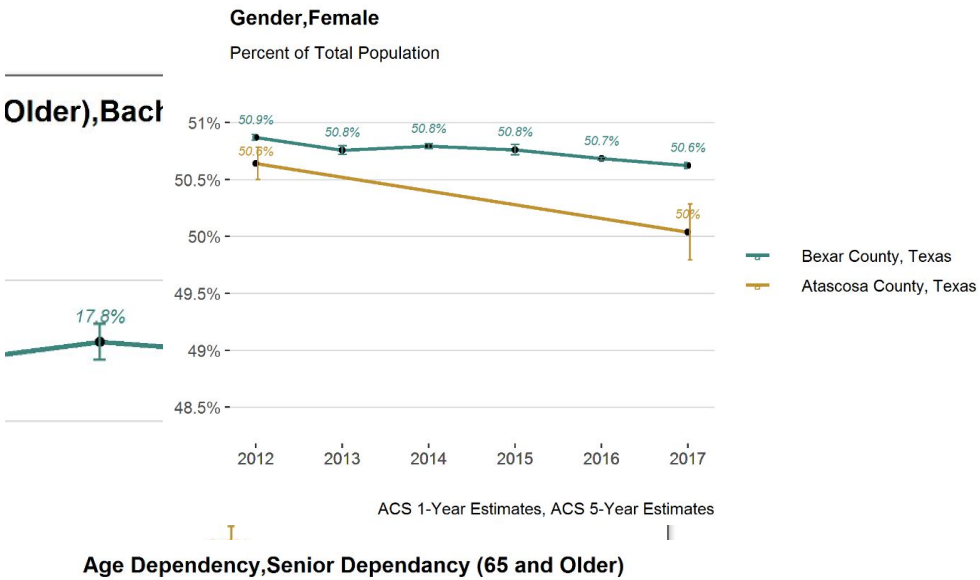
EXAMPLES OF PROJECTS

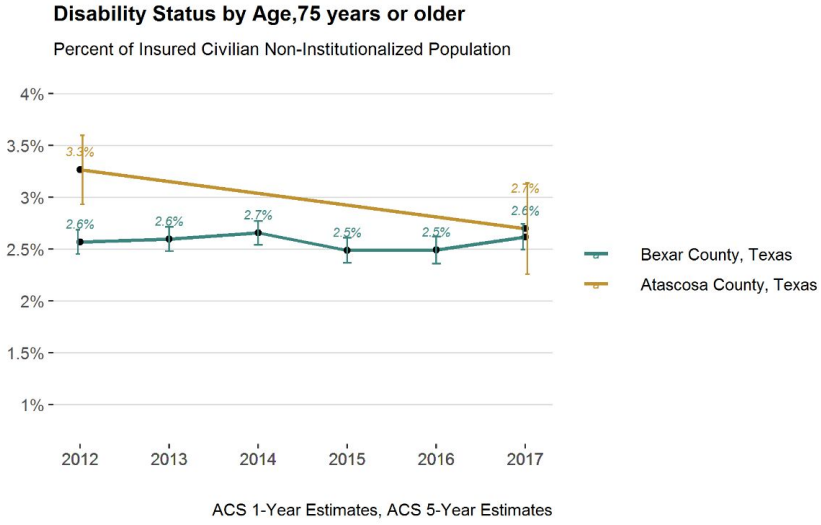
Automation and customization of over 200 Trendlines

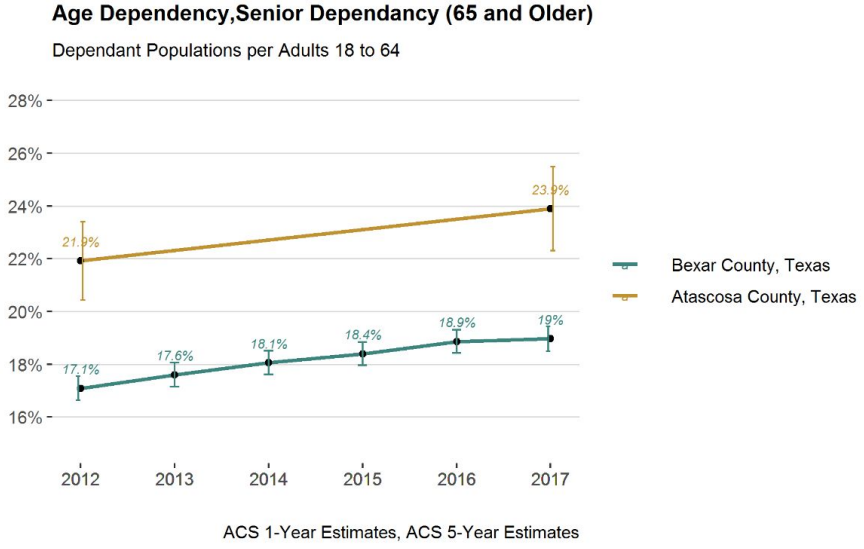
Geoid	Title	Subtitle	Source	Year	Estimate	Margin of Error (Moe)	Min Moe	Max Moe
Atascosa County	Educational Attainment (25 and Older), Bachelors Degree	Highest Degree Obtained	ACS 1-Year Estimates, ACS 5-Year Estimates	2012	8.15	1.38	6.77	9.53
Atascosa County	Educational Attainment (25 and Older), Bachelors Degree	Highest Degree Obtained	ACS 1-Year Estimates, ACS 5-Year Estimates	2017	9.61	1.64	7.97	11.25
Bexar County	Educational Attainment (25 and Older), Bachelors Degree	Highest Degree Obtained	ACS 1-Year Estimates, ACS 5-Year Estimates	2012	16.5	0.59	15.91	17.10
Bexar County	Educational Attainment (25 and Older), Bachelors Degree	Highest Degree Obtained	ACS 1-Year Estimates, ACS 5-Year Estimates	2013	17	0.58	16.42	17.58

EXAMPLE OF PROJECT:
AUTOMATION AND CUSTOMIZATION OF TRENDLINES









```r
trendpercent$geoid <- factor(trendpercent$GEOID, levels = c("Bexar County, Texas","Atascosa County, Texas"))
# create graphing function
trend.graph <- function(trendpercent, na.rm = TRUE, ...){

  # create list of title in data to loop over
  title_list <- unique(trendpercent$CHARTTITLE )

  # create for loop to produce ggplot2 graphs
  for (i in seq_along(title_list)) {

    # create plot for each title in trend and adds values for x and y plotting
    plot <-
      ggplot(subset(trendpercent, trendpercent$CHARTTITLE ==title_list[i]),
             aes(x=YEAR, y=ESTIMATE, ymin=minmoe, ymax=maxmoe, group=(GEOID), color=GEOID)) + theme_hc() +

             geom_line(aes(colour=GEOID),size = 1.0) +  geom_point(color="black")+

#labels in bars, paste % symbol, adjust color and position
 geom_text_repel(aes(label = paste0(round(ESTIMATE,1),"%")), size=2.5 , fontface='italic', force=1,
 segment.color = 'transparent', min.segment.length = 0, segment.size = 0, point.padding = .3,
 nudge_x = 0, nudge_y = .1, direction = "y")+

#add the % symbol to the values in axis and adjust limits to be automated and adds the function "expand".
scale_y_continuous(labels=function(ESTIMATE) paste0(ESTIMATE,"%"), expand = expand_scale(mult = c(.1,.3),
add = c(1.5,    0)),breaks=pretty_breaks(n=5),limits=c(min(trendpercent$minmoe[trendpercent$CHARTTITLE ==title_list[i]])
,max(trendpercent$maxmoe[trendpercent$CHARTTITLE ==title_list[i]])))+

 scale_x_continuous(breaks=c(trendpercent$YEAR))+

 #specify colors for lines
 scale_color_manual(values=c("Bexar County, Texas"="#3c857f", "Atascosa County, Texas"="#c09231"))+

 #modify legend
 theme(legend.position="right", legend.title = element_blank(),legend.spacing.x = unit(.6, 'cm'),

 #remove panel, set background color and reorder items in legend
 panel.grid.major = element_blank(), panel.background = element_rect(fill = "white")) +

 #error bars
 geom_errorbar(aes(ymin=minmoe, ymax=maxmoe), width=.1, position = position_dodge(0.1)) +

 #automates title and specified x and y labels as blank
  ggtitle(title_list[i])+labs( x="", y="", caption = trendpercent$SOURCEID [trendpercent$CHARTTITLE ==title_list[i]],
  subtitle = trendpercent$CHARTSUBTITLE[trendpercent$CHARTTITLE ==title_list[i]]) +
  theme(plot.title=element_text (hjust = 0, face='bold', size=12, margin=margin(0,0,10,0))) +
  theme(plot.subtitle = element_text(hjust=0,    size=10,margin=margin(0,0,20,0)))+
  theme(plot.margin = unit(c(.5,.9,.5,.5), "cm"))#top, right, bottom, left

  #Export as images
  ggsave(paste0(title_list[i],".png"))

  # print plots to screen
  print(plot)
  }}
 # run graphing function
 trend.graph(trendpercent, ESTIMATE)
```

Create an Index

Y Axis
X Axis
Group
Min Moe
Max Moe

Labels

Limits and breaks

Color of lines

Legend

Background

Error bars

Title, subtitle, source

Save as an image

Resources

News & Tutorials

R-bloggers

Blogs related to R and its applications

https://www.r-bloggers.com/

R Graph Gallery

Examples of visualizations with code samples

https://www.r-graph-gallery.com/

Troubleshooting

Rdocumentation

Manuals and information for packages
https://www.rdocumentation.org/

Stackoverflow

Developers share knowledge

https://stackoverflow.com/

Quick Tips

Rtips

List of common tasks performed in R

http://pj.freefaculty.org/R/Rtips.html.

ImpatientR

Introduction to basic functions

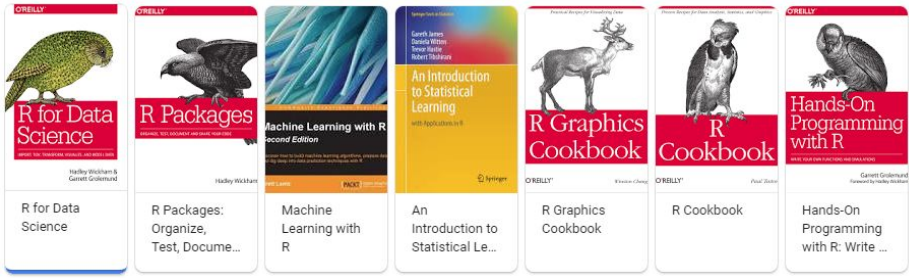
https://www.burns-stat.com/documents/tutorials/why-use-the-r-language/

YaRrr! The Pirates Guide to R

Intro to basic analytical tools in R, from basic coding and analyses, to data wrangling, plotting, and statistical inference.

https://bookdown.org/ndphillips/YaRrr/

Books



Thanks!

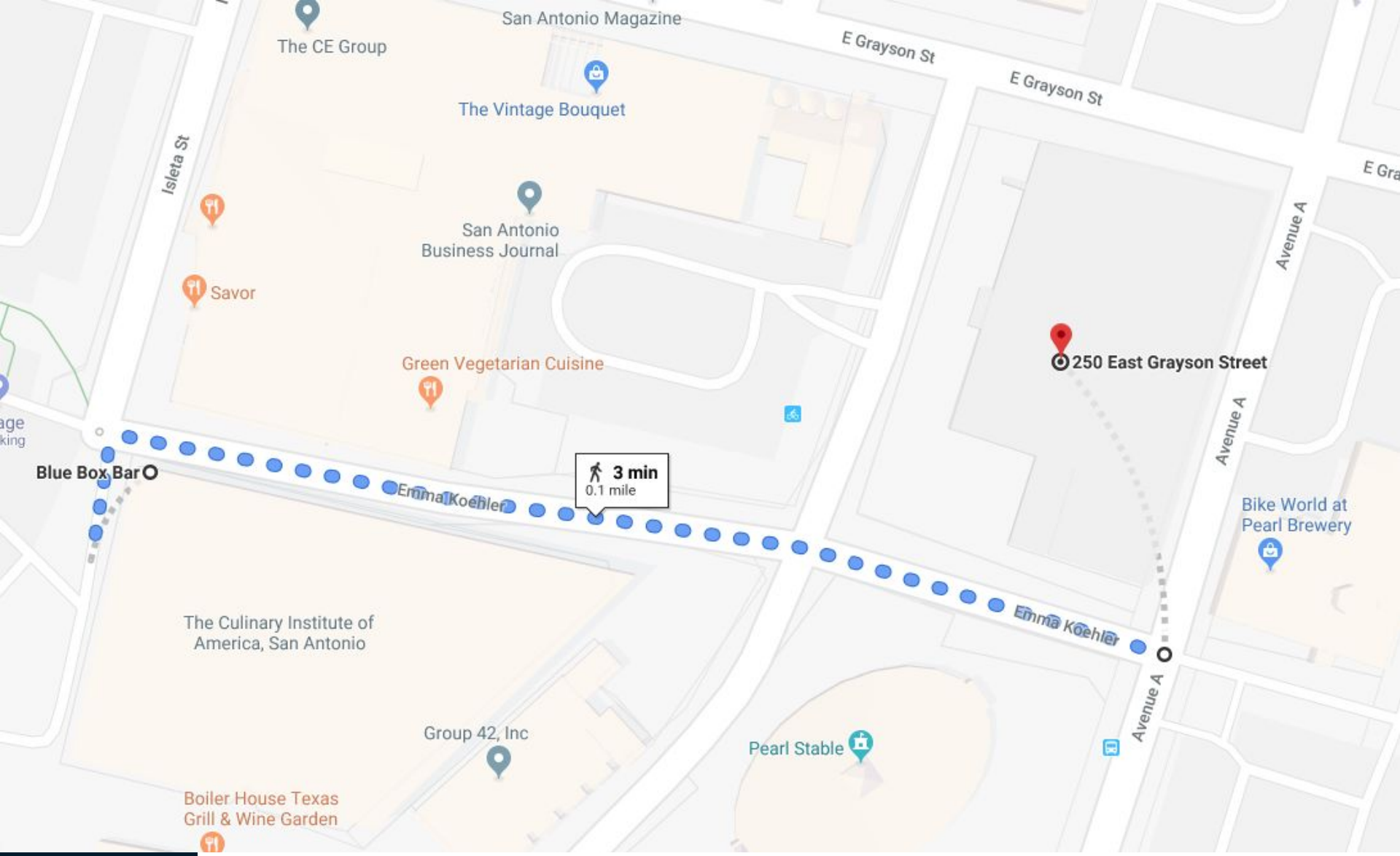
Questions?

Paulina Cano:

paulina.canomccutcheon@uth.tmc.edu

Jamie Ford:

jamie.ford@nisd.net



EST. 2012
BLUE BOX
BAR

Data Drinks @ 4:30 PM