A Quick Guide to Data Analysis Using R

Paul Testa

11/28/2016

# Overview

This document provides a quick guide to get you set up to use R, a statistical programming languange, for empirical quantitative analysis. Specifically, we'll cover the following

* Downloading and installing R and RStudio (a GUI that makes R nice and pretty).
* Basics of coding in R and R Markdown
* Setting up R to do your analysis
  + Installing packages in R and loading libraries in R that do specific things
  + Downloading and loading data
* Cleaning and recoding data
* Describing and Analyzing data to test hypothesis
* Producing Figures and Tables
* Additional Resources

This may seem like a lot. In some ways it is. But think of it like cooking. You could do some cooking in a microwave (Excel) but to make a really nice dinner (your honors thesis), you need a good kitchen (R), the right tools (libraries in R), the best ingredients (interesting data), properly prepared (cleaned and recoded) to let you produce a delicious meal (convincing empirical analysis). Learning a little programming will help you do this in way that will save time and reduce errors. So let's begin!

# Downloading and Installing R and R Studio

First we'll get our kitchen set up by downloading R and then RStudio.

* You can download R here: <https://cran.r-project.org/>
* Next, download the desktop version of RStudio here: <https://www.rstudio.com/products/rstudio/download/>

Make sure you pick the right version for your computer (i.e. Mac or PC)

# Basics of coding in R and R Markdown

This document is produced using R Mardown, a format that allows you to write code in R and text in markdown and compile it into a nice pretty document like an html file (or a word document or, with some additional software, a pdf). We'll cover more of how to write code and text in R Markdown in class. Essentially each .Rmd doucment is composed of the following:

An (optional) YAML header surrounded by ---s

---  
title: "A Quick Guide to Data Analysis Using R"  
author: "Paul Testa"  
date: "11/28/2016"  
output: html\_document  
---

R code chunks surrounded by ```s

# Here's an example of an actual code chunk using R as a calculator  
2+2

## [1] 4

Text mixed with simple text formatting (like this like)

Think of your .Rmd file as a recipe that lays out all the steps in your analysis:

* Setup
* Preparation
* Analysis
* Presentation

In class, we'll run code sequentially, executing it line-by-line in the console. When we're done, we'll "compile" our document. This essentially starts at the top of the file runs all the code, saves any output and combines it with our text to produce the final html file you're looking at.

In general, it's a good idea to compile often (as a way of checking for errors in your code). You do this by clicking the "knit" button in the script window (top left) or by clicking command+shift+k (cntrl+shift+k on a PC).

# Setup

## Clearing your workspace

In general, it's a good idea to clear out your workspace and start fresh, so typically we'll write this line of code

rm(list=ls())

## Installing packages and loading libraries

Next, we'll install packages that contain data, code and functions that as we'll see below, do useful things.

One package we'll need is the rmarkdown which turns this .Rmd file into a html or pdf or word document.

install.packages("rmarkdown")

You only have to instatll a pacakge once, but each time you open R you'll need to load packages using the library function. For example, the "foreign" package is useful for loading different types of data.

This next bit of code combines the install and library commands. Essetially, it installs the pacman package if it's not already installed and uses that the p\_load function from that package to install (if not installed) and load other packages you'll use in your analysis.

# Load and if necessary install packages  
if (!require("pacman")){ install.packages("pacman") }

## Loading required package: pacman

pacman::p\_load("knitr","mosaic","plyr","ggplot2","readstata13","car","lmtest","sandwich","gtable","grid","texreg","htmlTable")

## Setting your working directory

Computers (for now...) a very obedient. They do exactly what we tell them to do. Most of the time this is great, but some of the time it's a pain, particularly when we're writing code and make a mistake.

For example, say we want to load a dataset (more on that below). Generally, we have to tell R exactly where to look to find this data. This can be tediuous, so usually, I like to include a line of code that sets the working driectory for R to where my code (.Rmd file) and data (.csv, .rda, .dta etc. files are):

setwd("~/Documents/\_Brown/\_Teaching/thesis\_guide")

We'll talk more about this in class. And R studio's drop down menu will also do this for you if you click: Session -> Set working directory -> To Source File Location

## Loading Data

Ok, let's load some data and get down to business. We'll work with some data from the [2016 National Elections Pilot Study](http://www.electionstudies.org/studypages/anes_pilot_2016/anes_pilot_2016.htm). In class we'll download the data, save it into the folder with our code and load it directly. For now, we'll load it from the web:

nes<-read.csv(url("https://raw.github.com/PTesta/thesis\_guide/master/anes\_pilot\_2016.csv"))

## Looking at your data

As you'll see in the "Environment" panel in the upper right hand of R Studio, there's now an object called "nes" in our workspace. Next the name column in the Environment panel is a type column. You'll see that nes is a "data.frame" which is basically R's standard way for handling big spreadsheets of data. Each row in nes, corresponds to an individual survey respondent, each column corresponds to a variable or attribute of that respondent. We index rows by and columns by so that if we wanted to look at the first 5 characteristics of the first respondent we could type:

nes[1,1:5]

## version caseid weight weight\_spss follow  
## 1 ANES 2016 Pilot Study version 20160223 1 0.95116 0.5421612 1

Which is just a bunch of adminstrative data. If we wanted to see the partisan identification of the 73 respondent we would type

nes[73, "pid7"]

## [1] 1

Looking at the codebook, available here <http://www.electionstudies.org/studypages/anes_pilot_2016/anes_pilot_2016_CodebookUserGuide.pdf>

We see that a value of 1 on the "pid7" variable corresponds to someone who considers themselves to be a Strong Democrat.

There are lots of ways to look at the data. If we were interested in the education and party ID of the first ten respodents we could type

nes[1:10, c("educ","pid7")]

## educ pid7  
## 1 5 1  
## 2 6 4  
## 3 5 6  
## 4 5 1  
## 5 2 5  
## 6 6 4  
## 7 3 1  
## 8 2 7  
## 9 4 4  
## 10 6 1

Or if we wanted to know only the party ID of the 11 respondent we could also type

nes$pid7[11]

## [1] 6

Where the syntax says, in this object "nes" select the variable "$" with the name "pid7" and show me the eleventh element "[11]"

Finally, if we wanted to see how many observations and variables we had we could use the dim() function, to tell us the dimensions of our data.frame

dim(nes)

## [1] 1200 594

So we have 1200 observations and 594 variables.

# Preparation

In this section, we'll cover some best practices for preparing your data for analysis. The basic steps are as follows:

1. Identify your variables (columns in the data frame) of interest from a codebook
2. Look at the codebook and raw data to understand what values your data can take
3. Create a copy of the variable with a meaningful name
4. Recode any problematic or

## Recoding data

Let's look at people's feelings toward Donald Trump. The data contain a variable called fttrump. From the [questionaire](http://www.electionstudies.org/studypages/anes_pilot_2016/anes_pilot_2016_qnaire.pdf), we see that this variable contains people's responses to a "feeling thermometer" that asks them:

We'd like to get your feelings toward some of our political leaders and other people who are in the news these days. We'll show the name of a person or group and we'd like you to rate that person or group using something we call the feeling thermometer.  
Ratings between 50 degrees and 100 degrees mean that you feel favorable and warm toward the person. Ratings between 0 degrees and 50 degrees mean that you don't feel favorable toward the person and that you don't care too much for that person. You would rate the person at the 50 degree mark if you don't feel particularly warm or cold toward the person.  
If we come to a person whose name you don't recognize, you don't need to rate that person. Just click Next and we'll move on to the next one.

First, we'll use the table() function to take a quick look at the frequency of each unique value of "fttrump" (i.e. how many people felt really warmly toward him, how many felt lukewarm, etc).

table(nes$fttrump)

##   
## 0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17   
## 162 93 51 46 23 31 33 12 9 6 18 12 5 9 8 2 4 9   
## 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35   
## 6 7 14 7 10 1 1 2 2 2 5 1 10 12 4 4 5 6   
## 36 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54   
## 7 2 7 15 17 2 2 2 5 2 5 5 5 20 18 6 3 5   
## 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72   
## 6 9 4 2 5 18 10 6 3 6 10 10 3 4 10 18 12 11   
## 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90   
## 4 3 18 3 2 7 2 16 5 2 2 8 6 7 7 12 14 20   
## 91 92 93 94 95 96 97 98 99 100 998   
## 12 6 5 12 20 8 13 10 12 54 3

Ugh, That's a lot of numbers. The key thing to note though are the three people who have a value of 998. These aren't people using a different temperature scale. Instead their people who didn't respond, or skipped the question.

Different surveys have different conventions for how to label these responses. The NES typically uses 998 or 999. Some other surveys will leave them blank or code them as something else. For our purposes, we want to tell R to treat these data as "missing" or "not avaialbe".

1. First we'll create a new variable, called "ft\_trump" that's just a copy of the original varialbe
2. Next, we'll set the values of this new variable that equal 998 to NA (where NA is R's way of knowing the data are "not availabe")
3. Then we'll compare our recoded variable to our raw variable to make sure everything looks good.

# Create copy of fttrump for recoding  
nes$ft\_trump<-nes$fttrump  
# Recode 998s to NAs  
nes$ft\_trump[nes$ft\_trump==998]<-NA  
# Compare raw data to recoded data  
summary(nes$fttrump)

## Min. 1st Qu. Median Mean 3rd Qu. Max.   
## 0.00 2.00 30.00 40.78 72.00 998.00

summary(nes$ft\_trump)

## Min. 1st Qu. Median Mean 3rd Qu. Max. NA's   
## 0.00 2.00 30.00 38.38 72.00 100.00 3

Note the different outputs of the summary function (which gives you the mininum 25th percentile, median [50th percentile], mean, 75th percentile, and maximum)

Let's do some more recoding. Specifically let's look at:

* Gender
* Education
* Income
* Age
* PID
* Racial identity

# gender  
# 1= Male, 2=Female  
table(nes$gender)

##   
## 1 2   
## 570 630

# female01  
# 0=male, 1=female  
nes$female01<-ifelse(nes$gender==1,0,1)  
with(nes,table(female01,gender))

## gender  
## female01 1 2  
## 0 570 0  
## 1 0 630

# educ  
# 1= No Hs 6=Post-grad  
table(nes$educ)

##   
## 1 2 3 4 5 6   
## 102 411 257 106 202 122

# education  
nes$education<-nes$educ-1  
  
# faminc  
# 1-16  
table(nes$faminc)

##   
## 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 31 97   
## 75 122 149 128 105 99 64 77 83 50 46 29 11 5 5 3 2 146   
## 98   
## 1

# No cluse what 31, 97, 98 mean, make them NA  
  
# income  
# 0-15  
nes$income<-nes$faminc-1  
nes$income[nes$income>15]<-NA  
table(nes$income,useNA="ifany")

##   
## 0 1 2 3 4 5 6 7 8 9 10 11 12 13 14   
## 75 122 149 128 105 99 64 77 83 50 46 29 11 5 5   
## 15 <NA>   
## 3 149

# Age  
# birthyr  
#table(nes$birthyr)  
#age  
nes$age<-2016-nes$birthyr  
summary(nes$age)

## Min. 1st Qu. Median Mean 3rd Qu. Max.   
## 19.00 34.00 48.00 48.06 61.25 95.00

# Race  
# race  
# 1 White   
# 2 Black  
# 3 Hispanic   
# 4 Asian  
# 5 Native American   
# 6 Mixed  
# 7 Other  
# 8 Middle Eastern  
  
table(nes$race)

##   
## 1 2 3 4 5 6 7 8   
## 875 135 113 23 7 30 15 2

# Give race meaningful levels  
  
nes$race\_f<-recode(nes$race,"1='White';  
 2='Black';  
 3='Hispanic';  
 4='Asian';  
 5='Native American';  
 6='Mixed';  
 7='Other';  
 8='Middle Eastern'")  
  
with(nes, table(race\_f,race))

## race  
## race\_f 1 2 3 4 5 6 7 8  
## Asian 0 0 0 23 0 0 0 0  
## Black 0 135 0 0 0 0 0 0  
## Hispanic 0 0 113 0 0 0 0 0  
## Middle Eastern 0 0 0 0 0 0 0 2  
## Mixed 0 0 0 0 0 30 0 0  
## Native American 0 0 0 0 7 0 0 0  
## Other 0 0 0 0 0 0 15 0  
## White 875 0 0 0 0 0 0 0

# Create White/Nonwhite indicator  
nes$white01<-ifelse(nes$race==1,1,0)  
nes$nonwhite01<-ifelse(nes$race==1,0,1)  
  
table(nes$nonwhite01,nes$race\_f)

##   
## Asian Black Hispanic Middle Eastern Mixed Native American Other White  
## 0 0 0 0 0 0 0 0 875  
## 1 23 135 113 2 30 7 15 0

# PID  
# pid7  
# 1=strong dem 7= strong rep, 8 = not sure  
table(nes$pid7)

##   
## 1 2 3 4 5 6 7 8   
## 300 146 109 205 111 116 158 41

# pid  
nes$pid<-nes$pid7-1  
# recode notsures  
nes$pid[nes$pid>6]<-NA  
table(nes$pid,nes$pid7,useNA="ifany")

##   
## 1 2 3 4 5 6 7 8 <NA>  
## 0 300 0 0 0 0 0 0 0 0  
## 1 0 146 0 0 0 0 0 0 0  
## 2 0 0 109 0 0 0 0 0 0  
## 3 0 0 0 205 0 0 0 0 0  
## 4 0 0 0 0 111 0 0 0 0  
## 5 0 0 0 0 0 116 0 0 0  
## 6 0 0 0 0 0 0 158 0 0  
## <NA> 0 0 0 0 0 0 0 41 14

# dem,rep,ind indicator  
nes$dem01<-ifelse(nes$pid<3,1,0)  
nes$rep01<-ifelse(nes$pid>3,1,0)  
nes$ind01<-ifelse(nes$pid==3,1,0)  
  
  
# race\_ident  
# 1 Extremelyimportant; 5 Not important at all  
table(nes$race\_ident)

##   
## 1 2 3 4 5   
## 321 228 262 147 242

# reverse code so that higher values are more important  
nes$racial\_imp<-(nes$race\_ident-5)\*-1

# Analysis

Now we're ready to start cooking. Specifically, let's consider how support for trump varies conditionally on people's demographics and racial identity. We'll start by describing the data generally, and then move toward more nuanced tests of this association.

summary(lm(ft\_trump~racial\_imp,nes))

##   
## Call:  
## lm(formula = ft\_trump ~ racial\_imp, data = nes)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -40.700 -35.412 -8.123 33.588 64.453   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 35.5469 1.9022 18.687 <2e-16 \*\*\*  
## racial\_imp 1.2882 0.7202 1.789 0.0739 .   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 36.5 on 1195 degrees of freedom  
## (3 observations deleted due to missingness)  
## Multiple R-squared: 0.002671, Adjusted R-squared: 0.001836   
## F-statistic: 3.2 on 1 and 1195 DF, p-value: 0.0739

summary(lm(ft\_trump~racial\_imp\*white01,nes))

##   
## Call:  
## lm(formula = ft\_trump ~ racial\_imp \* white01, data = nes)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -53.284 -31.651 -8.596 32.033 77.670   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 41.351 5.220 7.922 5.32e-15 \*\*\*  
## racial\_imp -4.755 1.622 -2.931 0.00344 \*\*   
## white01 -8.701 5.581 -1.559 0.11925   
## racial\_imp:white01 9.914 1.820 5.448 6.19e-08 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 35.24 on 1193 degrees of freedom  
## (3 observations deleted due to missingness)  
## Multiple R-squared: 0.07178, Adjusted R-squared: 0.06945   
## F-statistic: 30.75 on 3 and 1193 DF, p-value: < 2.2e-16

## Descriptive Statistics

First, let's create a table with some descriptive statistics. We'll save the output of the summary function to objects begun by "sum\_" for each variable. For variables without NAs we'll add an additional NA element using the c() function.

sum\_trump<-summary(nes$ft\_trump)  
sum\_white<-c(summary(nes$white01),NA)  
sum\_female<-c(summary(nes$female01),NA)  
sum\_educ<-c(summary(nes$education),NA)  
sum\_income<-summary(nes$income)  
sum\_pid<-summary(nes$pid)  
sum\_raceimp<-c(summary(nes$racial\_imp),NA)

Next we'll stack each of these vectors of length 7 on top of each other to create a object called "tab\_sum"

tab\_sum<-rbind(sum\_trump,sum\_white,sum\_female,sum\_educ,sum\_income,sum\_pid,  
sum\_raceimp)  
tab\_sum

## Min. 1st Qu. Median Mean 3rd Qu. Max. NA's  
## sum\_trump 0 2 30 38.3800 72 100 3  
## sum\_white 0 0 1 0.7292 1 1 NA  
## sum\_female 0 0 1 0.5250 1 1 NA  
## sum\_educ 0 1 2 2.2180 4 5 NA  
## sum\_income 0 2 4 4.6120 7 15 149  
## sum\_pid 0 0 3 2.5770 4 6 55  
## sum\_raceimp 0 1 2 2.1990 4 4 NA

Finally, we can use the kable() and row.names() functions to create a pretty table with

tab\_sum\_md<-tab\_sum  
row.names(tab\_sum\_md)<- c("FT Trump", "Prop. White","Prop. Female","Education","Income","Party ID","Imp of Racial ID")  
kable(tab\_sum\_md)

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  | Min. | 1st Qu. | Median | Mean | 3rd Qu. | Max. | NA's |
| FT Trump | 0 | 2 | 30 | 38.3800 | 72 | 100 | 3 |
| Prop. White | 0 | 0 | 1 | 0.7292 | 1 | 1 | NA |
| Prop. Female | 0 | 0 | 1 | 0.5250 | 1 | 1 | NA |
| Education | 0 | 1 | 2 | 2.2180 | 4 | 5 | NA |
| Income | 0 | 2 | 4 | 4.6120 | 7 | 15 | 149 |
| Party ID | 0 | 0 | 3 | 2.5770 | 4 | 6 | 55 |
| Imp of Racial ID | 0 | 1 | 2 | 2.1990 | 4 | 4 | NA |

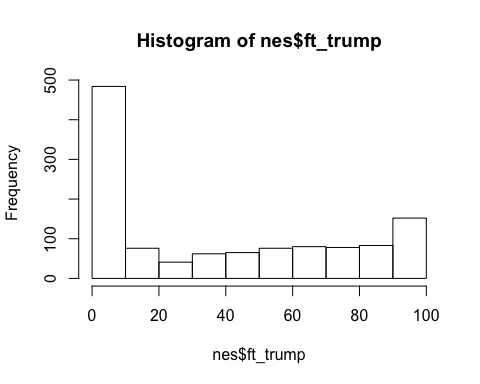
## Descriptive Figures

Next, let's see how to produce some descriptive figures.

### Histograms

Distribution of feelings toward trump:

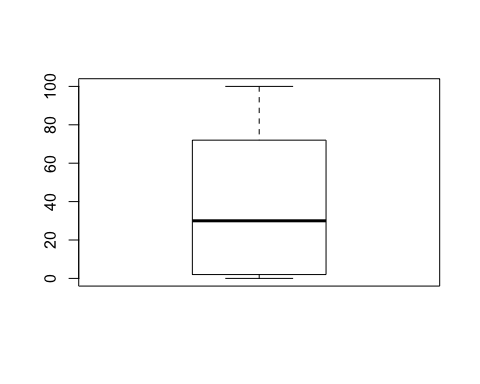
hist(nes$ft\_trump)



## Boxplots

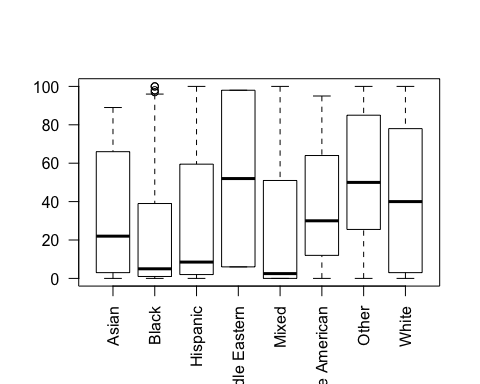
Another way of viewing the distribution of feelings toward Trump

boxplot(nes$ft\_trump)



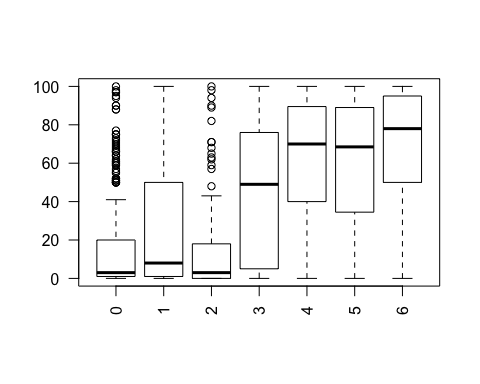
Now by race

boxplot(nes$ft\_trump~nes$race\_f,las=2)



Now by party id

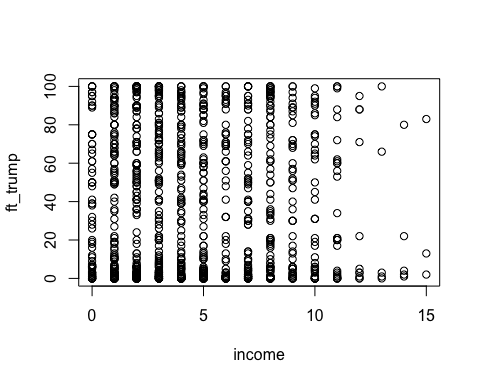
boxplot(nes$ft\_trump~nes$pid,las=2)



## Scatterplots

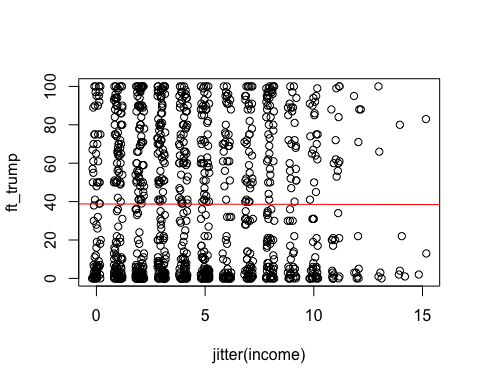
How do feelings toward trump vary by income

with(nes,plot(income,ft\_trump))

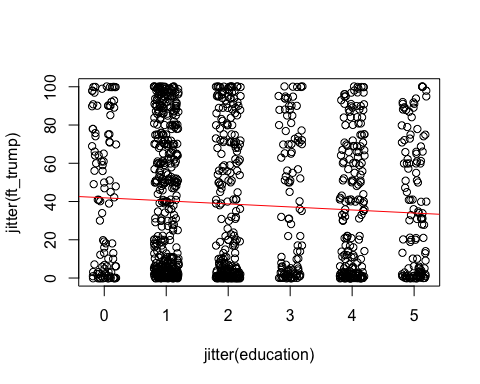


Hmm... hard to say. Let's jitter the income variable (add some random noise to each value) add a line to summarize this bivariate relationship

with(nes,plot(jitter(income),ft\_trump))  
abline(lm(ft\_trump~income,nes),col="red")



with(nes,plot(jitter(education),jitter(ft\_trump)))  
abline(lm(ft\_trump~education,nes),col="red")



How did we choose those lines, and not some other line to summarize the data? We used a procedure Ordinary Least Squares regression, to choose an intercept and slope that would minimize the sum of squared residuals--the vertical distance between our line (what our model predicts) and the dots (what we observe).

Why did we decide to minimize the sum of squared residuals? Well, if we think about our line as being a prediction of the typical value of feelings toward Trump, we want a prediction that on average is neither too high, nor too low conditional on people's income or education.

## Hypothesis Testing and Confidence intervals

Let's clarify our question about the relationship between race, the importance of one's racial identity and feelings toward Trump. For now, let's focus on the distinction between non-Whites and Whites. Given the election, we might expect that whites feel more warmly toward Trump than non-Whites. A simple way to test this would be to look at the difference in mean feeling thermometer responses between whites and non-whites

ft\_wh<-with(nes,mean(ft\_trump[white01==1],na.rm=T))  
ft\_wh

## [1] 42.50171

ft\_nonwh<-with(nes,mean(ft\_trump[white01==0],na.rm=T))  
ft\_nonwh

## [1] 27.17391

A difference of

ft\_nonwh-ft\_wh

## [1] -15.3278

Seems pretty big, but how do we know it's not just a feature of this particular sample or how certain are we that the true difference isn't something else? Let's formalize this concept of something else into a hypothesis, namely that the true difference (had we surveyed everyone in the US) is actually 0. We can use a difference of means or t-test to assess whether the data are consistent or inconsistent with this hypothesis

t.test(ft\_trump~white01,data=nes)

##   
## Welch Two Sample t-test  
##   
## data: ft\_trump by white01  
## t = -6.9466, df = 645.1, p-value = 9.163e-12  
## alternative hypothesis: true difference in means is not equal to 0  
## 95 percent confidence interval:  
## -19.66063 -10.99497  
## sample estimates:  
## mean in group 0 mean in group 1   
## 27.17391 42.50171

The p-value from this test tells us that if this hypotheis were true, the chances of seeing a test statistic (a standardized difference of means) as large in absolute value as the one we did (t = -6.9466) is low. Similarly, the confidence interval of -19.66063 -10.99497 describes a range of plausible values (hyptoheses we would fail to reject given the data) for the actually difference of means in the population.

## Bivariate Linear Regression with a Dichotmous Predictor

Now let's delve a little deeper into our linear model that we used to descriptively describe the relationship between feelings toward Trump and income and education above. Specifically let's model feelings toward Trump with our indicator of whether you're non-White or not, using lm(), R's function for OLS regression. We'll save the output of lm() to an object called m1 so we can apply the summary function to it

m1<-lm(ft\_trump~white01,nes)  
summary(m1)

##   
## Call:  
## lm(formula = ft\_trump ~ white01, data = nes)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -42.502 -34.502 -8.174 32.498 72.826   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 27.174 2.001 13.581 < 2e-16 \*\*\*  
## white01 15.328 2.340 6.549 8.56e-11 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 35.91 on 1195 degrees of freedom  
## (3 observations deleted due to missingness)  
## Multiple R-squared: 0.03465, Adjusted R-squared: 0.03384   
## F-statistic: 42.9 on 1 and 1195 DF, p-value: 8.56e-11

Those numbers should look familiar. The intercept is the mean among Whites, and the coefficient on nonwhite01 is the difference between the mean among Whites and the mean among non-whites.

The test stastic is slightly different from the t-test because of some assumptions about equal or unequal variances in these two groups, which we won't concern ourselves with, but with some adjustments, the linear model can produce the same results

coeftest(m1,vcovHC(m1,"HC2"))

##   
## t test of coefficients:  
##   
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 27.1739 1.8167 14.9581 < 2.2e-16 \*\*\*  
## white01 15.3278 2.2065 6.9466 6.133e-12 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

## Bivariate Linear Regression with a Continuous Predictor

So we've seen that linear regression essentially tells us how the mean of our outcome varies linearly with changes in our predictor. Let's consider how feelings toward Trump vary with the importance of ones racial identity

m2<-lm(ft\_trump~racial\_imp,nes)  
summary(m2)

##   
## Call:  
## lm(formula = ft\_trump ~ racial\_imp, data = nes)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -40.700 -35.412 -8.123 33.588 64.453   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 35.5469 1.9022 18.687 <2e-16 \*\*\*  
## racial\_imp 1.2882 0.7202 1.789 0.0739 .   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 36.5 on 1195 degrees of freedom  
## (3 observations deleted due to missingness)  
## Multiple R-squared: 0.002671, Adjusted R-squared: 0.001836   
## F-statistic: 3.2 on 1 and 1195 DF, p-value: 0.0739

Recall, racial identity is an ordinal variable. OLS treats it's as continous variable with interval properties (the change from 0 to 1 is the same from 3 to 4). For now we'll ignore this somewhat strong assumption and focus on the result, which is a weakly positive replatiship between feelings and racial identity. Specifically, someone who says their racial identity is not at all important, has generally cool feeling toward Trump 35.54. Someone who feels very strongly about their racial identity has a warmer feeling toward Trump (35.54+1.29$$4= 40.7)

# Presentation

## Descriptive Table