Descriptive Statistics in R

Joseph Nathan Cohen

9/8/2019

Contents

Introduction

In this module, we will learn how to begin an analysis in R. We will learn about:

- Our goals in descriptive analysis
- Matching data and descriptive methods
- Summarizing single, discrete variables
- Summarizing single, continuous variables
- Summarizing the relationship between two discrete variables
- Summarizing the relationship between one discrete and one continuous variable
- Summarizing the relationship between two continuous variables

Descriptive Analysis

Our goal in this program is to teach you how to test and refine your beliefs about how people think and act through observational data. We want you to learn to watch how people think and behave in the real world, and use what you see to sharpen your worldview about them. If you want to understand education, we might press you to go out and watch how teachers and students at real world schools think and act. If you were trying to understand how to create an influential advertisement, we would encourage you to get information about how people react to different kinds of advertising or marketing experiments. If you were trying to understand how parents treat their children, you might ask both parents and children how they think about and act towards each other. The main point is that we are training you in a method that involves basing your beliefs about the world by watching real world events.

When you are making these observations, it is a good idea to keep records. Records helps us deal with the fact that memories can fade or become contorted over time. They allow other people to reexamine what we think we saw, and maybe catch those circumstances where we clearly missed something, saw something that didn't happen, or perhaps looked at an event with a biased or distorted lens. Moreover, records can be used to use mathematics to test our ideas.

One problem in this process is that we are often faced with more data than our minds can process. For example, imagine I want to study the economic circumstances of people in Queens. I assemble a random sample of 300 households in Queens, and ask them how much money they earn.[1] If you were to ask "How much do people in Queens earn?", the most exact and detailed answer would be the 300 number sequence depicted in Table 1 below. Go take a look at it and then come back. I'll wait.

73569	63664	10128	8000	18542	12197	187555	80971	75000	8000
157858	8000	8000	8000	8000	8000	8000	8000	106701	15335
58416	8000	15927	70342	54614	8000	84499	8000	27954	28122
48705	34211	31780	31361	35063	127586	8000	8000	8000	8000
8000	108202	42775	8000	8000	30899	26310	83228	8000	86352
72466	8000	48924	8000	40361	8000	8000	117903	180771	76882
45282	157106	67646	131567	8000	13278	8000	33061	43151	44692
42888	181179	102927	162706	66484	37512	92960	8000	133919	8000
33423	48376	8000	35689	168634	8000	157417	8000	81205	8000
76395	156817	95501	168962	8000	229968	55889	8000	114875	59473
157111	8000	8000	21605	12289	62450	56621	130824	8000	177284
96565	70602	92438	57145	150283	142651	53858	50541	124672	158284
18047	8000	168068	8000	30761	103544	172127	60732	22706	31576
23609	99374	8000	8000	88712	8000	8000	8000	100484	90915
156136	8000	135922	33923	60456	173453	8000	69327	20804	8000
75277	14640	8000	58614	99505	47566	8000	85300	149375	25513
85408	202234	148697	130448	82307	59428	83104	126413	8000	98134
70184	8000	177260	8000	103021	133165	8000	24006	42911	83736
8000	124362	168071	143977	15406	8000	76654	8000	237352	35126
181365	222231	194333	100744	8000	161508	108633	52873	136504	41129
91264	8000	199449	161545	8000	8000	107871	8000	157119	47016
161404	144667	198210	8000	25494	47011	68437	137538	62147	119051
30965	159231	120709	95956	35144	8000	8000	8000	122039	211641
8000	94049	34046	49210	127393	74969	81328	88684	162556	9618
10117	162351	123251	220905	8000	80315	36697	8000	26249	8000
251279	13045	36890	8000	167743	26970	8000	139438	23132	33826
153293	8000	8000	28148	161881	61989	15534	25426	8000	80182
9037	68151	31267	57578	163212	11203	8000	70284	60804	93851
8000	160602	8000	259435	8000	8000	119147	129983	8000	53198
8000	42939	87856	42711	145664	116072	142900	65380	113676	47919

Table 1: Three Hundred Household Incomes

So, now you've seen the information. What can you tell us about how much money people earn? You have all the information after all! The question isn't easily answered, because I gave you 300 discrete pieces of information. That's far more than your mind can process. We need some more cognitively-accessible way of describing how these data are distributed. (We use the term *distribution* to describe all the scores that we observe in a data series or variable.)

[1] Assume a mean income of \$54,373, a standard deviation of \$75,000, and a minimum guaranteed income of \$8,000. This is in fact a far more egalitarian society that it our real world.

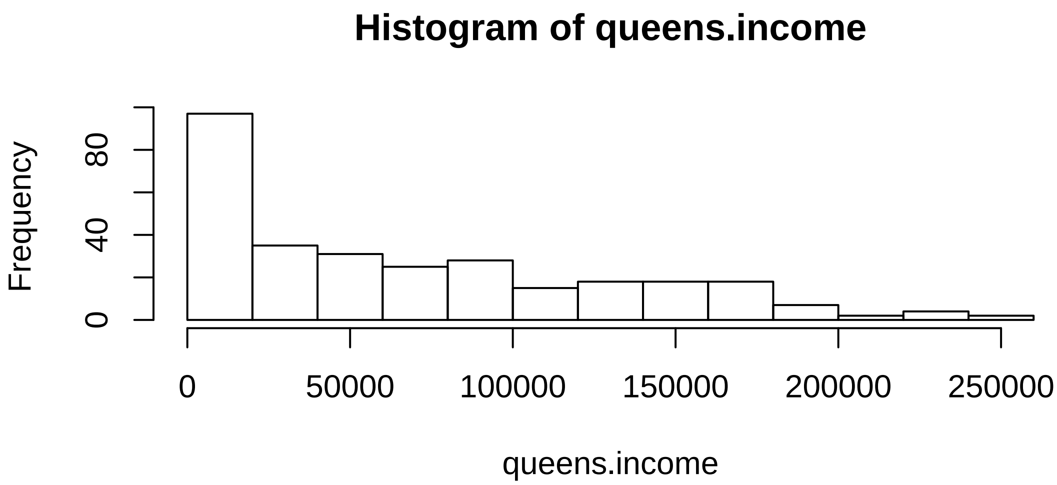


Figure 1: Histogram of 300 Household Income Observations

You've already learned how to summarize this distribution – by using *summary statistics*. Here's an example, in which the data from this table was coded into an object called "queens.income":

```
summary(queens.income)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##    8000    8000   49876   68440  109894  259435
```

```
sd(queens.income)
```

```
## [1] 61455.76
```

These statistics might not give us a full cognizance of every observation in its individual glory, but it does give us a sense of what typical observations look like. We get a sense that a middling income in our sample is about $50 thousand. We know at least 25% of the sample earns the minimum income of $8,000, and about 25% earn more than $110,000. We know the biggest earner in our sample took in almost $260 thousand per year. The typical observation deviated plus/minus $61,455 from the mean.

Alternatively, we can depict the results of our 300 observations graphically in a histogram (see Figure 1 below).

```
hist(queens.income)
```

This gives us a different look at the data, but it's still reducing the complexity (from 300 observations to a dozen or so bars). It's another way to describe all of the income data that we collected in a succinct way.

These are examples of the kind of jobs we do when performing *descriptive analysis*. We are reducing the complexity of our data to cognitively-digestible pieces of information. When our data are summarized, people can more easily make sense of them and interpret their meaning.

The above two examples – the numerical and graphical summaries of our fictitious income data – illustrate *univariate* statistics – descriptions of how one particular variable or data series is distributed. Today, we will also explore the production and interpretation of *bivariate* statistics, which summarize the relationship between two variables. For example, Figure 2 (below) depicts the relationship between education and income in our (fictional) data. The figure is describing the average income by educational level. It is conveying evidence that the two variables are related, at least in our sample – more educated people make more money,

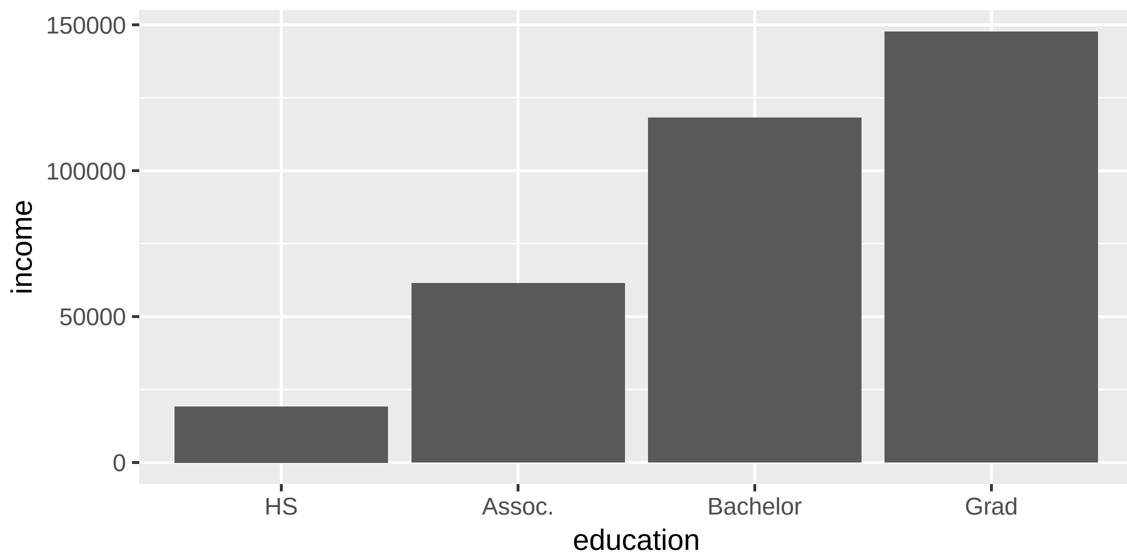


Figure 2: Bar Chart of Income-Education in Our Fictional Data

on average.

Matching Variable Types and Descriptors

Variable Types

Descriptive analysis involves matching your variables to the appropriate descriptor. Doing so requires that you be able to differentiate between two types of variables:

- ***Discrete Variables.*** Discrete variables can only take a finite number of values. They can take non-numerical values (e.g., race, education level) or numerical ones (e.g., points on a Likert scale). You cannot take fractions of these numbers.
- ***Continuous Variables.*** These are more natural numbers than can be divisible and have no natural lower and upper bounds. For example, height, distance, or money are often measured as continuous variables.

Note: There are times that we treat discrete variables as continuous ones. For example, a survey question in which respondents are asked to rate their feelings about a particular restaurant on a scale from zero to 100 might be considered discrete in a technical sense (if the survey does not allow answers with decimal places, and does not allow responses outside of the 0 - 100 scale). However, we conventionally treat these types of variables as continuous. As a rule of thumb, we treat variables with a smaller set of possible responses as discrete, whereas we can treat discrete variables with large numbers of potential responses along an ordinal scale as continuous.

Choosing The Appropriate Descriptor Operation

Variables and their relationships can be described numerically or graphically. The chart below gives recommended operations for different combinations of discrete and continuous variables:

Variable Set	Numerical Descriptor	Graphical Descriptor
One Discrete	Frequency Table	Histogram
One Continuous	Central Tendency and Dispersion Metrics	Histogram
Two Discrete	Cross-Tabulation	Bar Chart
A Discrete & Continuous	Summary Statistics Table	Bar Chart or Box Plot
Two Continuous	Correlations	Scatterplot

We will show how to implement and interpret these operations below.

Data for this Module

For this module, we will use an extract from the *American National Election Survey of 2016*, a major survey studying people's political attitudes and voting behavior. You can download that number from the class site. The data is in a file called "ANES Data.csv", and the codebook is "ANES Extract Codebook.txt"

Recall that we are working with unweighted data today. Our descriptive statistics are not describing estimates of these relationships among the general population. Instead, we are describing what was observed in the survey sample.

Univariate Analysis

Univariate analysis involves the description of a single variable's distribution. We perform these types of descriptive statistics when we want to show whether something is rare or commonplace, to show what constitutes a "typical" score, or to illustrate how scores are spread out.

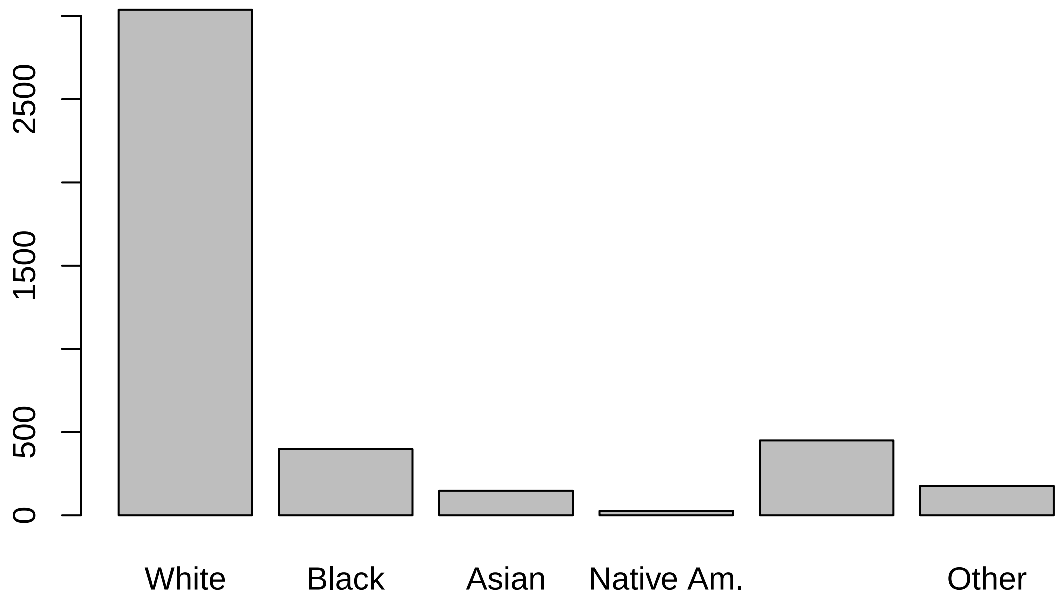
Describing a Single Discrete Variable

Numeric Summary: Frequency Table

I recommend describing the distribution of a single discrete variable by using a *frequency table*, a table that describes the proportion of people who fell into each of this variable's categories. One way to generate such a table is by using the **freq()** command, which is part of the ***descr*** package. This command also generates a histogram.

Let's try it on the race variable of our ANES extract:

```
library(descr)
freq(data$race)
```




```
## data$race
##             Frequency  Percent Valid Percent
## White            3038  71.1309       71.6848
## Black             398   9.3187        9.3912
## Asian             148   3.4652        3.4922
## Native Am.         27   0.6322        0.6371
## Hispanic          450  10.5362       10.6182
## Other             177   4.1442        4.1765
## NA's               33   0.7727
## Total            4271 100.0000      100.0000
```

There may be times when you want to create distinct objects with frequency counts or percentages. You might do this when you want to construct a nice figure or table using a cool R package, for example. If you want to create an object with frequency counts, use the **table()** function:

```
table(data$race)
```

```
##
##      White      Black      Asian Native Am.   Hispanic      Other
##       3038        398        148         27        450        177
```

To get proportions, use the **prop.table()** function on the table you produce using **table()**:

```
prop.table(table(data$race))
```

```
##
##      White      Black      Asian Native Am.   Hispanic      Other
## 0.71684757 0.09391222 0.03492213 0.00637093 0.10618216 0.04176498
```

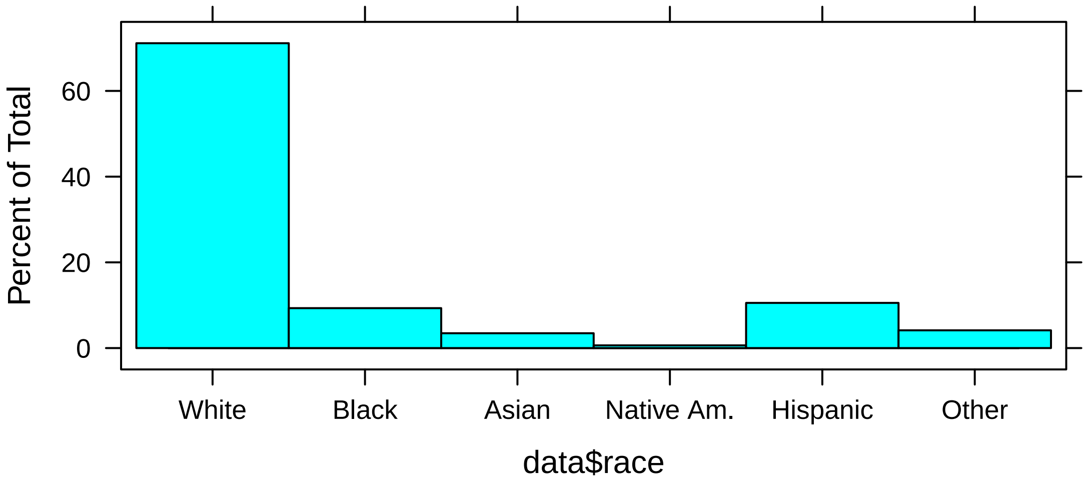


Figure 3: ANES Extract Racial Composition

Exercise. Load the ANES data and check out its codebook. Find a discrete variable and create a frequency table of it. What insights do you glean from it?

Graphical Summary: Histogram

We will learn how to make nice graphs next week. For the time being, we will work with simpler ones. To construct a histogram with discrete data, I recommend the **histogram()** function in the ***lattice*** pacakge

```
library(lattice)
histogram(data$race)
```

Exercise. Choose a discrete variable from the ANES codebook. Generate a historgram. What insights do you glean from the resulting figure?

Describing a Single Continuous Variable

Numeric Summary: Central Tendency Metrics

Central tendency metrics try to describe the "typical" observation. The two most common metrics are mean and median. The main difference between the two is that the mean incorporates the effect of very large positive or negative scores, while the median does not.

You can get either score by using the **summary()** function in the base package. For this analysis, we will treat the "feeling thermometer" questions as continuous variables:

```
summary(data$feel.whites)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.    NA's
##    0.00   52.00   70.00   71.75   85.00  100.00     698
```

You can also use the **mean()** and **median()** functions if you want to create objects or perform calculations using the mean or median of a data series. Remember that R will report these values as missing if there are any missing values in the variable. You can tell R to skip these missing observations by using the "na.rm" option:

```r
mean(data$feel.whites, na.rm = T)
```

```
## [1] 71.74531
```

```r
median(data$feel.whites, na.rm = T)
```

```
## [1] 70
```

Numeric Summary: Dispersion Metrics

Dispersion metrics Measure the spread of variables. Two useful metrics are:

- the *standard deviation*, which measures the average observation's deviation from the sample mean
- *percentile scores*, which return values at different percentiles in the distribution

Standard Deviation. To get a standard deviation, you can use the **sd()** operation:

```r
sd(data$feel.police, na.rm = T)
```

```
## [1] 22.48235
```

This statistic suggest that the typical score on people's "feeling thermometer" towards police is +/- 22.5 points about the sample mean.

Percentile Scores. The **quantile()** command can be used to get percentile scores of a continuous variable. For the 10th, 50th (median) and 90th percentile scores:

```r
quantile(data$feel.police, probs = c(0.1, 0.5, 0.9), na.rm = T)
```

```
## 10% 50% 90%
##  41  85 100
```

So 10% of the sample gave police a "feeling thermometer" below 41, and at least 10% gave them a 100. The median was 85. Compare this figure with attitudes towards Congress

```r
quantile(data$feel.congress, probs = c(0.1, 0.5, 0.9), na.rm = T)
```

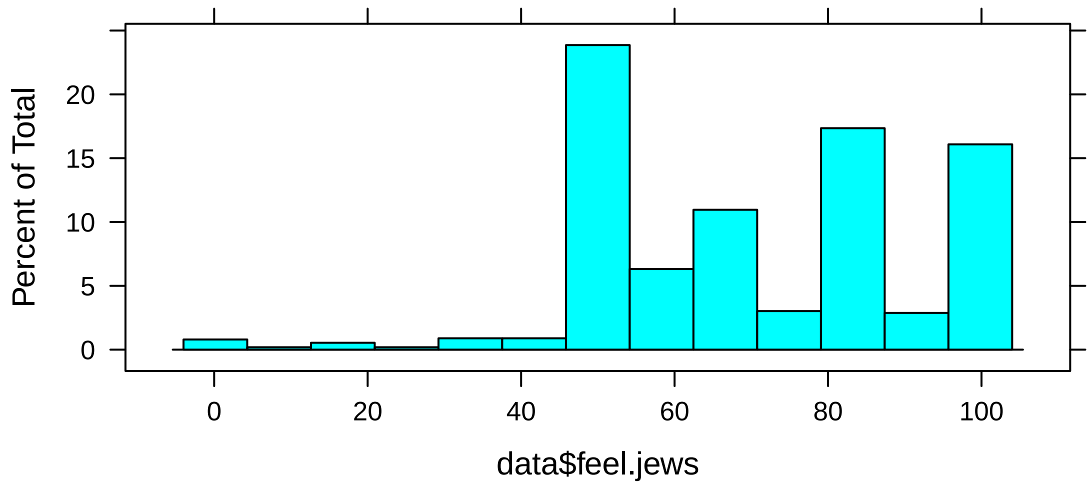
```
## 10% 50% 90%
##  15  41  70
```

Exercise. Which groups do you think attract the warmest or coldest feelings among Americans in general. Test your intuition by examining the ANES data.

Graphical Summary: Histogram

As with discrete metrics, we can depict the distribution of discrete variables using **histogram()**:

```
histogram(data$feel.jews)
```

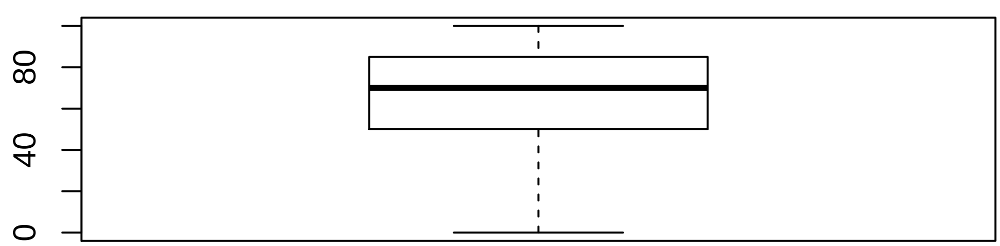


Graphical Summary: Box Plots

Box plots are another way to describe a distribution. The top and bottom edges of the box represents the 75th and 25th percentile scores. The line in the middle of the box is the median. The lines (whiskers) coming out of the top and bottom of the boxes show the distance to the minimum and maximum (excluding outliers). Outliers are depicted as dots.

Use the **boxplot()** command.

```
boxplot(data$feel.asians)
```



Bivarate Descriptives

Bivariate statistics examine the relationship between two variables. You should base your choice of descriptor on the combination of variables with which you are working:

Two Discrete Variables

Numerical Summary: Cross-Tabulation

If you want to create an clean object with simple cross tables, use the functino **table()**

```
table(data$vote16, data$education)
```

```
##
##           Less than HS High School College Grad School
##   Clinton           62         550     328         342
##   Trump             43         677     304         144
##   Other              6          97      68          22
```

And you can get proportions using **prop.table()**. This will give you each cell as a percentage of all observations.

```
prop.table(table(data$vote16, data$education))
```

```
##
##           Less than HS High School     College Grad School
##   Clinton  0.023458191 0.208096860 0.124101400 0.129398411
##   Trump    0.016269391 0.256148316 0.115020810 0.054483541
##   Other    0.002270148 0.036700719 0.025728339 0.008323874
```

If you want to express these proportions as percentages of rows:

```
prop.table(table(data$vote16, data$education), 1)
```

```
##
##           Less than HS High School    College Grad School
##   Clinton   0.04836193  0.42901716 0.25585023  0.26677067
##   Trump     0.03681507  0.57962329 0.26027397  0.12328767
##   Other     0.03108808  0.50259067 0.35233161  0.11398964
```

Or as columns:

```
prop.table(table(data$vote16, data$education), 2)
```

```
##
##           Less than HS High School    College Grad School
##   Clinton   0.55855856  0.41540785 0.46857143  0.67322835
##   Trump     0.38738739  0.51132931 0.43428571  0.28346457
##   Other     0.05405405  0.07326284 0.09714286  0.04330709
```

Exercise. Were this survey's more religious respondents more likely to vote for Donald Trump, Hillary Clinton, or another candidate?

Graphical Summary: Paneled Bar Chart

To create a paneled bar chart, create a table of proportions (see above). For example, let's say I want to create a paneled bar chart that describes support for Donald Trump among different racial groups:

```
tab.trump.race <- prop.table(table(data$vote16, data$race))
round(tab.trump.race, 4)
```

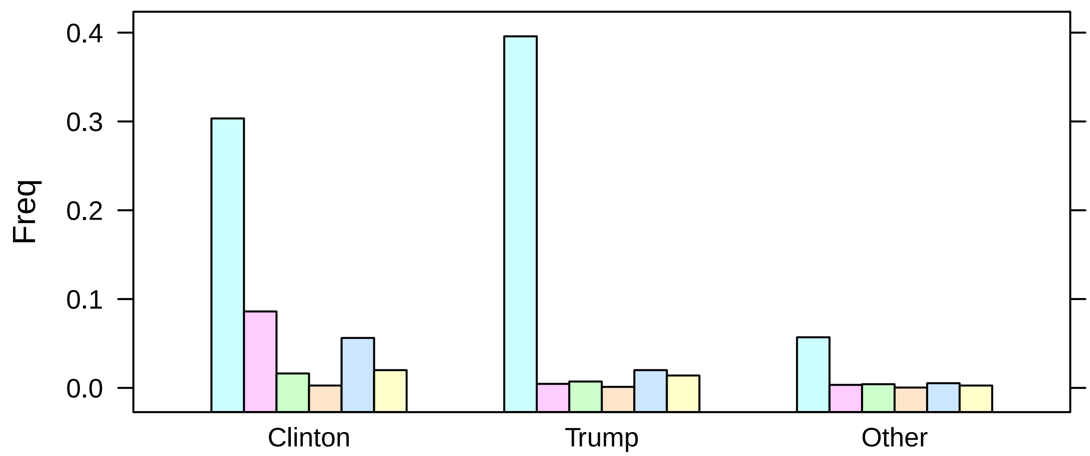


Figure 4: Paneled Bar Chart 1

```
##
##            White  Black  Asian Native Am. Hispanic  Other
##   Clinton 0.3034 0.0860 0.0162     0.0026   0.0562 0.0200
##   Trump   0.3958 0.0045 0.0072     0.0011   0.0200 0.0140
##   Other   0.0570 0.0034 0.0042     0.0004   0.0053 0.0026
```

Once you have the cross-tab that you want to graph, follow these steps:

First, transform the cross-tab into a data frame:

```
tab.trump.race.2 <- data.frame(tab.trump.race)
head(tab.trump.race.2)
```

```
##      Var1  Var2        Freq
## 1 Clinton White 0.303396226
## 2   Trump White 0.395849057
## 3   Other White 0.056981132
## 4 Clinton Black 0.086037736
## 5   Trump Black 0.004528302
## 6   Other Black 0.003396226
```

Next, use the operation **barchart()** on the resulting data table. We use the variable names of our new data frame. If we want the race distribution across Presidential candidates (Figure 4):

```
barchart(Freq~Var1, tab.trump.race.2, groups = Var2)
```

For Presidential votes across races (Figure 5):

```
barchart(Freq~Var2, tab.trump.race.2, groups = Var1)
```

Graphical Summary: Stacked Bar Chart

This graph is simpler. It's a bit ugly and non-informative here, but I'll include it FYI. You use the raw counts cross-tab object:

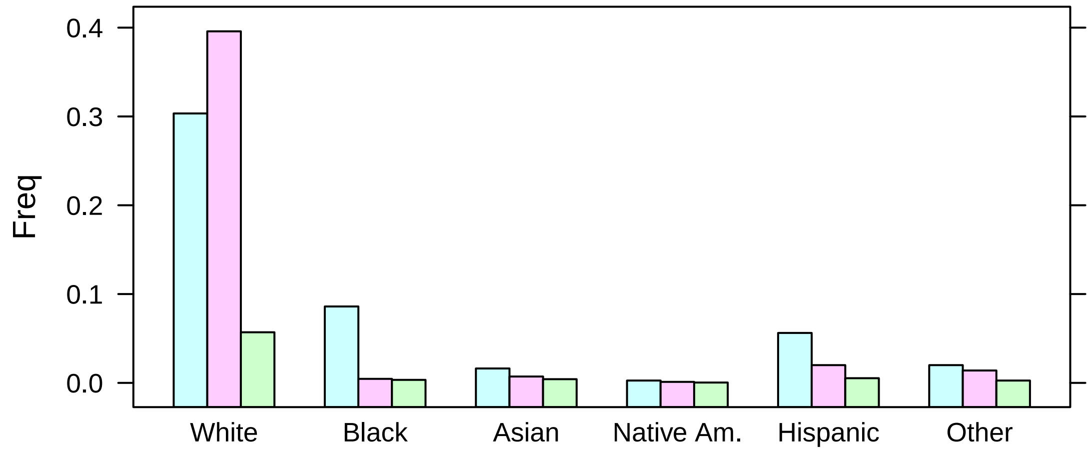
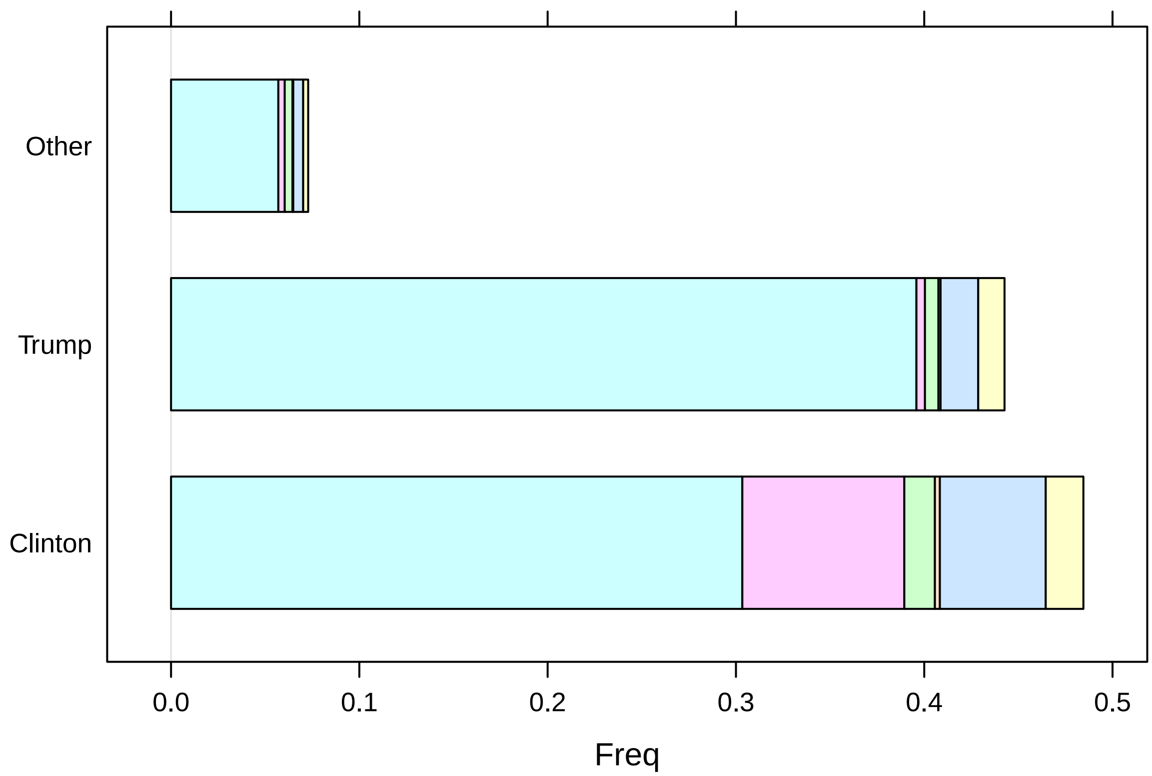


Figure 5: Paneled Bar Chart 2

```r
barchart(tab.trump.race)
```



These are crude. We can add labels, etc. with subcommands, but our task here is quick plots that are not for publication. We'll develop formatted graphs in the visualization chapter, using the ***ggplot2*** package.

One Discrete and One Continuous Variable

Numerical Summary: Tables of Summary Statistics

The most common way to describe a relationship between a discrete and continuous variable is by a table of summary statistics, in which we present measures of central tendency or dispersion across the discrete variable's categories. This can be done using the **aggregate()** command.

To get the mean "feeling thermometer" score towards police across voting groups:

```
aggregate(feel.police ~ vote16, data = data, mean)
```

```
##    vote16 feel.police
## 1 Clinton    69.78521
## 2   Trump    85.13787
## 3   Other    73.25128
```

For the median:

```
aggregate(feel.police ~ vote16, data = data, median)
```

```
##    vote16 feel.police
## 1 Clinton          70
## 2   Trump          85
## 3   Other          76
```

Standard deviation:

```
aggregate(feel.police ~ vote16, data = data, sd)
```

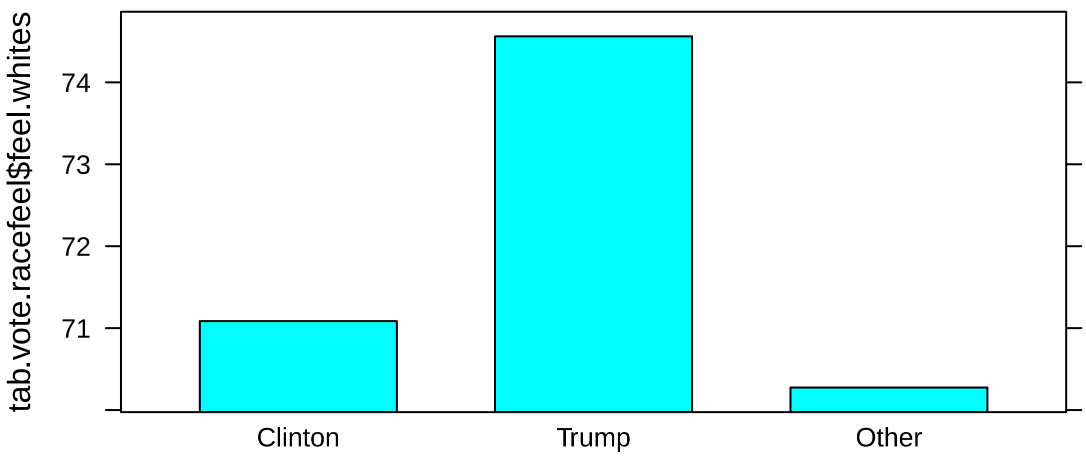
```
##    vote16 feel.police
## 1 Clinton    22.71561
## 2   Trump    16.42254
## 3   Other    21.33951
```

Exercise. Were there big differences between Trump and Clinton voters' feelings towards trans people? What do we learn when comparing group medians? What do we learn when comparing group standard deviations?

Graphical Summary: Bar Charts

When creating figures with this combination of variables, create an object using the results of your **aggregate()** operation, and then use the operation **barchart()** on it. For example, to graph differences in mean feelings towards white people across voting groups:

```
tab.vote.racefeel <- aggregate(feel.whites ~ vote16, data=data, mean)
barchart(tab.vote.racefeel$feel.whites ~ tab.vote.racefeel$vote16)
```



Beware interpreting this graph! As we will learn next week, the truncated axis (and lack of a zero point on this axis) distorts our impression of what this graph says.

Two Continuous Variables

Numerical Summary: Correlations

The **cor()** operation gives correlations:

```
cor(data$feel.blm, data$feel.police, use="pairwise.complete.obs")
```

```
## [1] -0.266921
```

The negative correlation suggests that people who view Black Lives Matter more positively tend to have more negative views of police, and vice-versa.

You can produce a correlation matrix for multiple variables. For example, let's say I wanted to make a matrix of correlations between "feeling thermometer" variables for BLM, police, the Rich, Christian Fundamentalists, Muslims, and Jews. My first step is to identify the column position of the variables containing these data.

```
names(data)
```

```
##  [1] "id"                   "weight"               "religious"
##  [4] "age.group"            "education"            "race"
##  [7] "vote16"               "income"               "feel.dempres"
## [10] "feel.reppres"         "feel.fundamentalists" "feel.feminists"
## [13] "feel.liberals"        "feel.unions"          "feel.poor"
## [16] "feel.bigbiz"          "feel.cons"            "feel.scotus"
## [19] "feel.lgb"             "feel.congress"        "feel.rich"
## [22] "feel.muslims"         "feel.christians"      "feel.jews"
## [25] "feel.teaparty"        "feel.police"          "feel.trans"
## [28] "feel.scientists"      "feel.blm"             "feel.asians"
## [31] "feel.hisp"            "feel.blacks"          "feel.undoc"
## [34] "feel.whites"
```

It looks like we are dealing with variables 11, 21, 22, 24, 26, and 29:

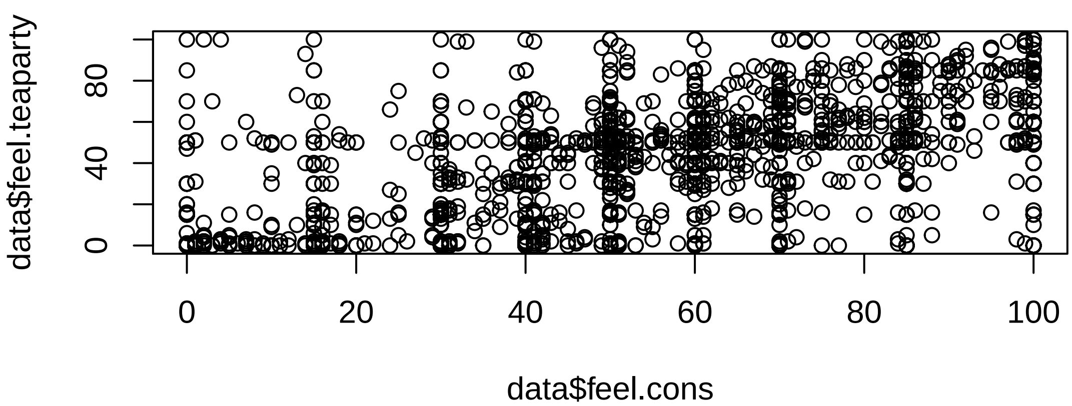
```
cor(data[,c(11, 21, 22, 24, 26, 29)], use = "pairwise.complete.obs")
```

```
##                       feel.fundamentalists   feel.rich feel.muslims
## feel.fundamentalists             1.0000000  0.26610842  -0.13402544
## feel.rich                        0.2661084  1.00000000   0.08980237
## feel.muslims                    -0.1340254  0.08980237   1.00000000
## feel.jews                        0.1058170  0.30268196   0.40542789
## feel.police                      0.2656590  0.32017615  -0.03445646
## feel.blm                        -0.1335612 -0.08468782   0.50445238
##                       feel.jews feel.police    feel.blm
## feel.fundamentalists 0.1058170  0.26565902 -0.13356120
## feel.rich            0.3026820  0.32017615 -0.08468782
## feel.muslims         0.4054279 -0.03445646  0.50445238
## feel.jews            1.0000000  0.24334457  0.15499624
## feel.police          0.2433446  1.00000000 -0.26692099
## feel.blm             0.1549962 -0.26692099  1.00000000
```

Graphical Summary: Scatterplot

One way to depict the relationship between two continuous variables graphically is through a scatterplot. You can make a basic scatterplot using the **plot()** command:

```
plot(data$feel.cons, data$feel.teaparty)
```



Special Functions to Manipulate Your Variables

Subsetting

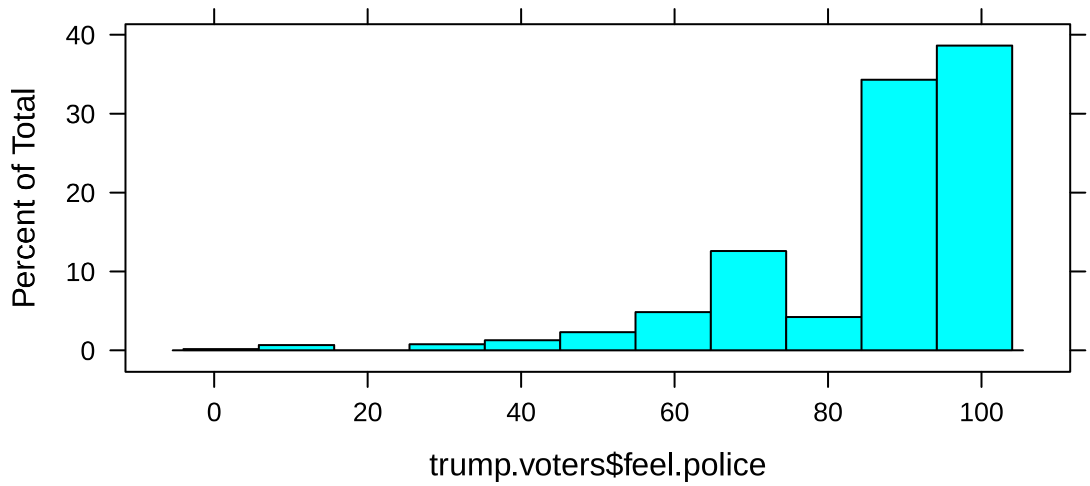
subsetting means extracting part of your data set. Sometimes, we only want to look at a particular subset of our sample. To create a new data frame that includes only observations that meet a particular set of conditions, use the **subset()** function.

For example, if we only wanted to focus on Trump voters, we could use the "vote16" variable in this set:

```r
trump.voters <- subset(data, vote16 == "Trump")
summary(trump.voters$feel.police)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.    NA's 
##    0.00   80.50   85.00   85.14  100.00  100.00       3
```

```r
histogram(trump.voters$feel.police)
```

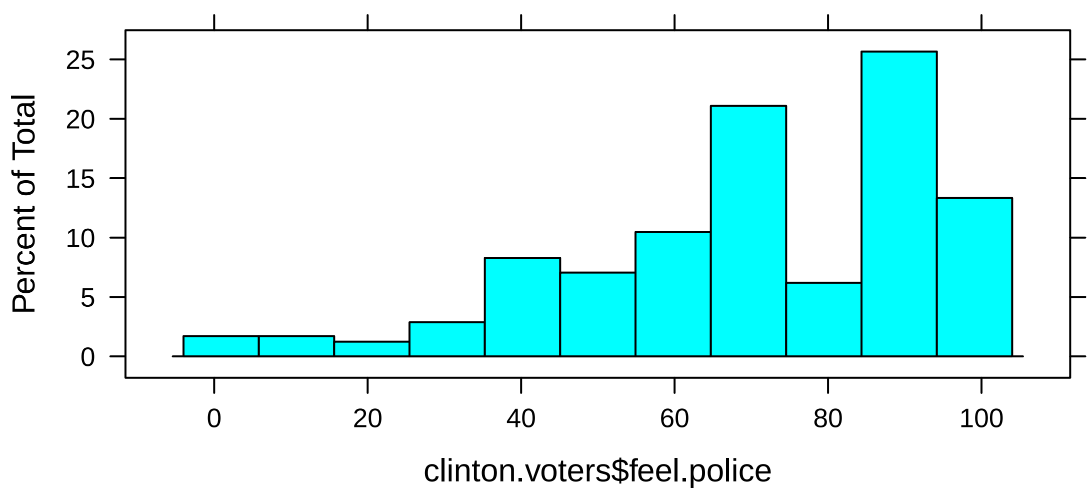


Trump voters view the police very positively. This was less true of Clinton supporters:

```r
clinton.voters <- subset(data, vote16 == "Clinton")
summary(clinton.voters$feel.police)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.    NA's 
##    0.00   60.00   70.00   69.79   85.00  100.00       5
```

```r
histogram(clinton.voters$feel.police)
```



Cutting

Cutting is an operation that transforms a continuous variable into a discrete one. For example, maybe we wanted to divide our "feeling thermometer" metrics into three categories: dislike (<40), neutral (40 - 60), and like (>60). We would use the **cut()** function:

```r
#Look at the original variable
summary(data$feel.blm)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.    NA's
##    0.00   16.00   50.00   48.35   70.00  100.00     681
```

```r
#Cutting Variable
#Note I set bounds outside of min & max
data$blm.groups <- cut(data$feel.blm,
                       breaks = c(-1,40,60, 101),
                       labels = c("Dislike", "Neutral", "Like"))

prop.table(table(data$blm.groups))
```

```
##
##   Dislike   Neutral      Like
## 0.4038997 0.2562674 0.3398329
```