Lab 1: Exploratory Data Analysis

Sarah Wright

January 18, 2022

This lab will be due *Thursday, January 27th* by the end of day. Adapted from "Start teaching with R," created by R Pruim, N J Horton, and D Kaplan, 2013 and "Interactive and Dynamic Graphics for Data Analysis," by Dianne Cook and Deborah F. Swayne.

Introduction

One of the most important components of data science is exploratory data analysis. One definition, which comes from this article (though it's probably not the original source) explains exploratory data analysis as the following:

"Exploratory Data Analysis refers to the critical process of performing initial investigations on data so as to discover patterns, spot anomalies, to test hypotheses and to check assumptions with the help of summary statistics and graphical representations."

Before you begin your exploratory analysis, you may already have a particular question in mind. For example, you might work for an online retailer and want to develop a model to predict which purchased items will be returned. Or, you may not have a particular question in mind. Instead, you might just be asked to look at browsing data for several customers and figure out some way to increase purchases. In either case, before you construct a fancy model, you need to explore and understand your data. This is how you gain new insights and determine if an idea is worth pursuing.

Understanding your data

Today we will be working with the `TIPS` dataset which is in the `regclass` package. The data in the `TIPS` dataset is information recorded by one waiter about each tip he received over a period of a few months working in a restaurant. We would like to use this data to address the question, *"What factors affect tipping behavior?"* Please prepare your answers to the following questions using the Lab Submission Guidelines posted on our course Moodle page.

1. Install the `regclass` package by either typing `install.packages("regclass")` in the console or by clicking "Tools > Install Packages" and selecting the package.

Once you have done this, the R chunk below will load the package and dataset. Notice that a bunch of unnecessary output is included when you knit the document. Change the R chunk options so that this is not displayed.

Edit the code below to display only necessary output. You may need to Google to find the exact code you need!

```
#install.packages("regclass")
library(regclass)
```

```
## Warning: package 'regclass' was built under R version 4.0.2
```

```
## Loading required package: bestglm
```

```
## Warning: package 'bestglm' was built under R version 4.0.2
```

```
## Loading required package: leaps
```

```
## Warning: package 'leaps' was built under R version 4.0.2
```

```
## Loading required package: VGAM
```

```
## Warning: package 'VGAM' was built under R version 4.0.2
```

```
## Loading required package: stats4
```

```
## Loading required package: splines
```

```
## Loading required package: rpart
```

```
## Loading required package: randomForest
```

```
## Warning: package 'randomForest' was built under R version 4.0.2
```

```
## randomForest 4.6-14
```

```
## Type rfNews() to see new features/changes/bug fixes.
```

```
## Important regclass change from 1.3:
## All functions that had a . in the name now have an _
## all.correlations -> all_correlations, cor.demo -> cor_demo, etc.
```

```
library(dplyr)
```

```
## Warning: package 'dplyr' was built under R version 4.0.2
```

```
##
## Attaching package: 'dplyr'
```

```
## The following object is masked from 'package:randomForest':
##
##     combine
```

```
## The following objects are masked from 'package:stats':
##
##     filter, lag
```

```
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```r
library(mosaic)
```

```
## Warning: package 'mosaic' was built under R version 4.0.2
```

```
## Registered S3 method overwritten by 'mosaic':
##   method                           from
##   fortify.SpatialPolygonsDataFrame ggplot2
```

```
##
## The 'mosaic' package masks several functions from core packages in order to add
## additional features.  The original behavior of these functions should not be affected by this.
```

```
##
## Attaching package: 'mosaic'
```

```
## The following object is masked from 'package:Matrix':
##
##     mean
```

```
## The following object is masked from 'package:ggplot2':
##
##     stat
```

```
## The following objects are masked from 'package:dplyr':
##
##     count, do, tally
```

```
## The following objects are masked from 'package:VGAM':
##
##     chisq, logit
```

```
## The following objects are masked from 'package:stats':
##
##     binom.test, cor, cor.test, cov, fivenum, IQR, median, prop.test,
##     quantile, sd, t.test, var
```

```
## The following objects are masked from 'package:base':
##
##     max, mean, min, prod, range, sample, sum
```

```r
library(Stat2Data)
```

```
## Warning: package 'Stat2Data' was built under R version 4.0.2
```

```r
#library(tidyverse)

data("TIPS")
```

When exploring a new dataset, it's important to first understand the basics. What format is our data in? What types of information are included in the dataset? How many observations are there? (These are rhetorical questions!)

2. In R, datasets are usually stored in a 2-dimensional structure called a *data frame*. You can get an idea of the structure of a dataset using the syntax `str(dataset)` and you can peak at the first few rows and columns with `head(dataset)`. Use these functions in the R chunk below to better understand the data. How many tips are recorded in this dataset? Which days of the week did the waiter work? (Answer these questions in the chunk below.)

```
str(TIPS) #basic info about the data set
```

```
## 'data.frame':    244 obs. of  8 variables:
##  $ TipPercentage: num  5.94 16.1 16.7 14 14.7 18.6 22.8 11.6 13 21.9 ...
##  $ Bill         : num  17 10.3 21 23.7 24.6 ...
##  $ Tip          : num  1.01 1.66 3.5 3.31 3.61 4.71 2 3.12 1.96 3.23 ...
##  $ Gender       : Factor w/ 2 levels "Female","Male": 1 2 2 2 1 2 2 2 2 2 ...
##  $ Smoker       : Factor w/ 2 levels "No","Yes": 1 1 1 1 1 1 1 1 1 1 ...
##  $ Weekday      : Factor w/ 4 levels "Friday","Saturday",..: 3 3 3 3 3 3 3 3 3 3 ...
##  $ Time         : Factor w/ 2 levels "Day","Night": 2 2 2 2 2 2 2 2 2 2 ...
##  $ PartySize    : int  2 3 3 2 4 4 2 4 2 2 ...
```

```
head(TIPS, 10) #view the first few rows of the dataset
```

```
##    TipPercentage  Bill  Tip Gender Smoker Weekday  Time PartySize
## 1           5.94 16.99 1.01 Female     No  Sunday Night         2
## 2          16.10 10.34 1.66   Male     No  Sunday Night         3
## 3          16.70 21.01 3.50   Male     No  Sunday Night         3
## 4          14.00 23.68 3.31   Male     No  Sunday Night         2
## 5          14.70 24.59 3.61 Female     No  Sunday Night         4
## 6          18.60 25.29 4.71   Male     No  Sunday Night         4
## 7          22.80  8.77 2.00   Male     No  Sunday Night         2
## 8          11.60 26.88 3.12   Male     No  Sunday Night         4
## 9          13.00 15.04 1.96   Male     No  Sunday Night         2
## 10         21.90 14.78 3.23   Male     No  Sunday Night         2
```

```
count(TIPS)
```

```
##     n
## 1 244
```

```
summary(TIPS)
```

```
##  TipPercentage        Bill             Tip           Gender    Smoker
##  Min.   : 3.56   Min.   : 3.07   Min.   : 1.000   Female: 87   No :151
##  1st Qu.:12.88   1st Qu.:13.35   1st Qu.: 2.000   Male  :157   Yes: 93
##  Median :15.45   Median :17.80   Median : 2.900
##  Mean   :16.08   Mean   :19.79   Mean   : 2.998
##  3rd Qu.:19.12   3rd Qu.:24.13   3rd Qu.: 3.562
##  Max.   :71.00   Max.   :50.81   Max.   :10.000
##      Weekday      Time       PartySize
##  Friday  :19   Day  : 68   Min.   :1.00
##  Saturday:87   Night:176   1st Qu.:2.00
##  Sunday  :76               Median :2.00
##  Thursday:62               Mean   :2.57
##                            3rd Qu.:3.00
##                            Max.   :6.00
```

```
From the TIPS dataset we can see that:
  Tips are recorded in this dataset:
    244 tips are recorded in the data set
 Days of the week did the waiter works:
    Friday, Saturday, Sunday, Thursday
```

Often, a dataset will come with a *code book* which gives more complete information about the structure of the data, the meaning of variables, and how the data were collected. In this case, most of the column names are pretty self explanatory.

Variable	Description
`TipPercentage`	the gratuity, as a percentage of the bill
`Bill`	the cost of the meal in US dollars
`Tip`	the tip in US dollars
`Gender`	gender of the bill payer
`Smoker`	whether the party included smokers
`Weekday`	day of the week
`Time`	time the bill was paid
`PartySize`	size of the party

3. Even though the column names are self-explanatory, we might have more questions about the data. For example, we might conjecture that people tip differently for breakfast and lunch, but our data only tells us if the bill was paid at "Day" or "Night." State another reasonable conjecture about a factor that might affect tipping behavior. What additional information would be helpful to explore that conjecture?

   On average do people tip differently based on the bill payers gender?

```
TIPS %>% group_by(Tip, Gender) %>% count()
```

```
## # A tibble: 139 x 3
## # Groups:   Tip, Gender [139]
##      Tip Gender     n
##    <dbl> <fct>  <int>
##  1  1    Female     3
##  2  1    Male       1
##  3  1.01 Female     1
##  4  1.1  Female     1
##  5  1.17 Male       1
##  6  1.25 Female     1
##  7  1.25 Male       2
##  8  1.32 Male       1
##  9  1.36 Female     1
## 10  1.44 Male       2
## # ... with 129 more rows
```

```
tally(~Gender, data = TIPS) #number of male compared to female workers
```

```
## Gender
## Female   Male
##     87    157
```

```
    From this we gather that there are more males pay the bill compared to females
```

Numerical Summaries

Now we'd like to start looking closely at the dataset to develop some ideas about what factors might affect tipping. The basic descriptive statistics have obvious names, like `mean, median, sd, IQR, quantile`, etc. A quick shortcut function is `summary()`, which computes several numerical summaries all at once.We can apply these functions to an entire data frame or a specific column of the data fame.

4. Use some of these summaries to answer the following. How many smokers are in the dataset? How fancy do you think restaurant is? Is it possible to tell from this summary how many different shifts the waiter worked? Why or why not?

```
summary(TIPS)
```

```
##  TipPercentage        Bill             Tip            Gender    Smoker
##  Min.   : 3.56   Min.   : 3.07   Min.   : 1.000   Female: 87   No :151
##  1st Qu.:12.88   1st Qu.:13.35   1st Qu.: 2.000   Male  :157   Yes: 93
##  Median :15.45   Median :17.80   Median : 2.900
##  Mean   :16.08   Mean   :19.79   Mean   : 2.998
##  3rd Qu.:19.12   3rd Qu.:24.13   3rd Qu.: 3.562
##  Max.   :71.00   Max.   :50.81   Max.   :10.000
##      Weekday      Time       PartySize
##  Friday  :19   Day  : 68   Min.   :1.00
##  Saturday:87   Night:176   1st Qu.:2.00
##  Sunday  :76               Median :2.00
##  Thursday:62               Mean   :2.57
##                            3rd Qu.:3.00
##                            Max.   :6.00
```

```
TIPS %>% group_by(Time)
```

```
## # A tibble: 244 x 8
## # Groups:   Time [2]
##    TipPercentage  Bill   Tip Gender Smoker Weekday Time  PartySize
##            <dbl> <dbl> <dbl> <fct>  <fct>  <fct>   <fct>     <int>
##  1          5.94 17.0   1.01 Female No     Sunday  Night         2
##  2         16.1  10.3   1.66 Male   No     Sunday  Night         3
##  3         16.7  21.0   3.5  Male   No     Sunday  Night         3
##  4         14    23.7   3.31 Male   No     Sunday  Night         2
##  5         14.7  24.6   3.61 Female No     Sunday  Night         4
##  6         18.6  25.3   4.71 Male   No     Sunday  Night         4
##  7         22.8   8.77  2    Male   No     Sunday  Night         2
##  8         11.6  26.9   3.12 Male   No     Sunday  Night         4
##  9         13    15.0   1.96 Male   No     Sunday  Night         2
## 10         21.9  14.8   3.23 Male   No     Sunday  Night         2
## # ... with 234 more rows
```

```
  How many smokers are in the dataset?
      There are 93 smokers in the data set
  How fancy do you think restaurant is?
      Not very the bill mean is around 20 dollars and the median is only 17.
  Is it possible to tell from this summary how many different shifts the waiter worked?  Why or why not
      We can tell what day and time the waiter worked, so I would say yes. For example on Sunday the wa
```

As we start to explore different questions, we might want to know things about interactions between variables. For instance, are tips larger during the day or at night? Or does gender or smoking status matter for how much people spend and how much they tip? You can calculate statistics within groups by including grouping variables and using `aggregate` like this:

```
aggregate(Tip ~ Time, data = TIPS, FUN = median)
```

```
##    Time  Tip
## 1   Day 2.25
## 2 Night 3.00
```

```
aggregate(cbind(Bill, TipPercentage) ~ Gender + Smoker, data = TIPS, FUN = mean)
```

```
##   Gender Smoker     Bill TipPercentage
## 1 Female     No 18.10519      15.69296
## 2   Male     No 19.79124      16.06701
## 3 Female    Yes 17.97788      18.21606
## 4   Male    Yes 22.28450      15.27967
```

The ~ (tilde) symbol appears in a lot of functions. In R, a **formula** is an expression involving ~ that provides slots for laying out how you want to relate variables: `y ~ x` means "y versus x", "y depends on x", "y explained by" x, or "break down y by x". In the top case above, you're saying "break Tip down by Time" or "perform this function on the Tip, conditioned on Time."

5. Calculate the variance of the tip percentage broken down by day of the week. HINT: Use `FUN = var` in the aggregate function. Do you notice anything unusual? Explore the data and determine a possible cause for this.

```
aggregate(TipPercentage ~ Weekday, data = TIPS, FUN = var)
```

```
##    Weekday TipPercentage
## 1   Friday      22.54667
## 2 Saturday      26.33058
## 3   Sunday      71.82457
## 4 Thursday      14.96456
```

For categorical variables, we can create tables as follows:

```
table(TIPS$Smoker, TIPS$Gender)
```

```
##
##       Female Male
##   No      54   97
##   Yes     33   60
```

```
xtabs(~ Smoker + Gender, data = TIPS)
```

```
##       Gender
## Smoker Female Male
##    No      54   97
##    Yes     33   60
```

6. Which day of the week has the highest *percentage* of tables that are smokers?

```
prop.table(xtabs(~Smoker + Weekday, data = TIPS))
```

```
##        Weekday
## Smoker     Friday   Saturday     Sunday   Thursday
##    No  0.01639344 0.18442623 0.23360656 0.18442623
##    Yes 0.06147541 0.17213115 0.07786885 0.06967213
```

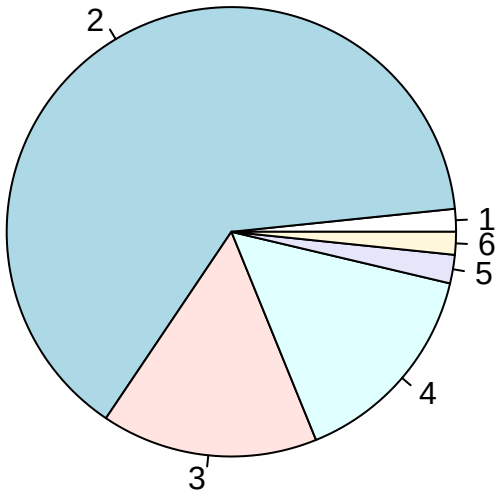
```
  The higest percentage of smokers is on Thursday
```

Graphical Summaries

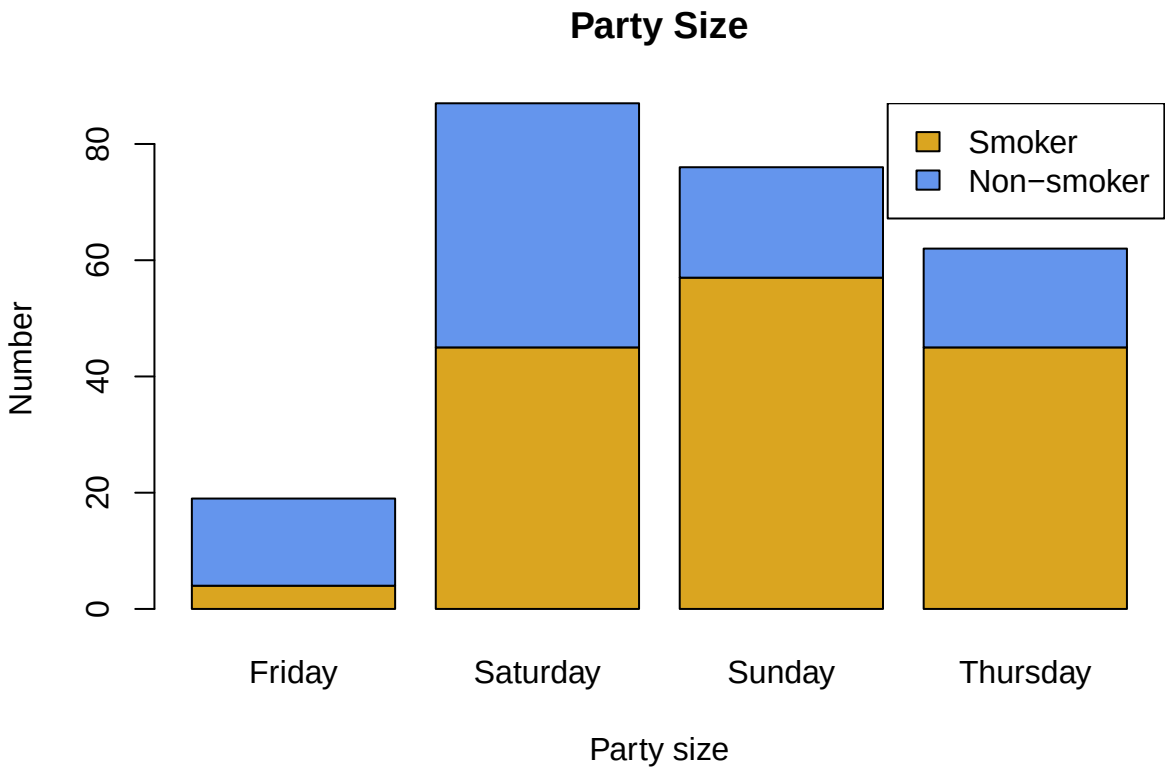
Graphical summaries are a key tool in exploratory data analysis to help you understand your data. They also help you communicate insights about your data to others.

For example, we might want to display relationships about some of our categorical variables. So we could start by graphing different party sizes in our dataset. (if you are familiar with the ggplot2 package, you could also use pie charts and graphs from there, too!)

```
party_size_table <- table(TIPS$PartySize)
pie(x = party_size_table, labels = 1:6)
```

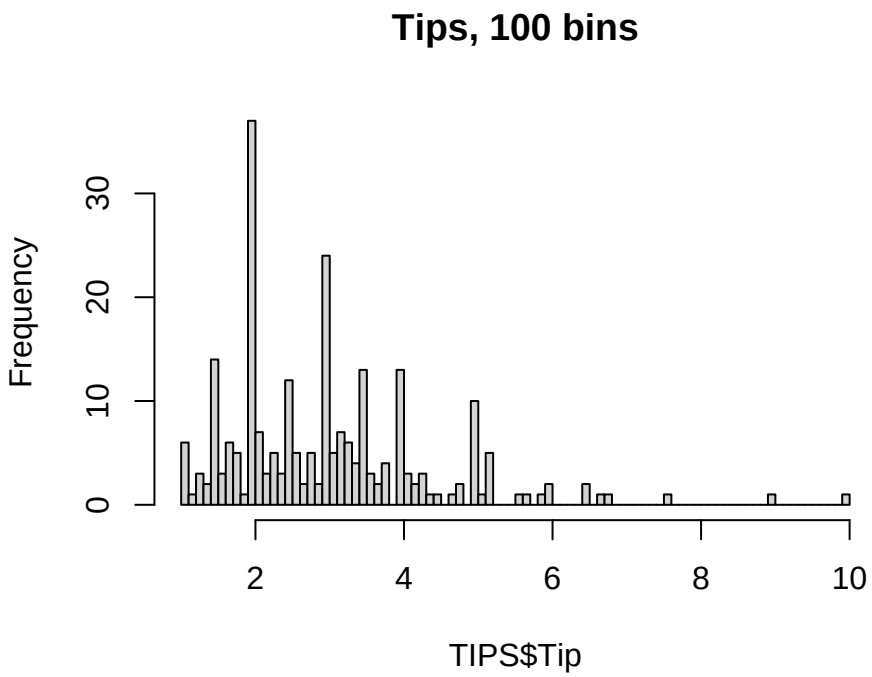


Or we could explore the question about the percentage of tables that are smokers on different days of the week visually (the command `ECHO = FALSE` makes it so code doesn't print). (You could also use ggplot2 here!) Warning: If you have errors running the legend function, it may be caused by showing your chunk output inline. Use the cog near the Knit button to select, "Chunk Output in Console" to have your console viewer show the barplot with a legend.



We might summarize a numerical variable with a histogram. For example, here is a histogram of all of the tips in the dataset.

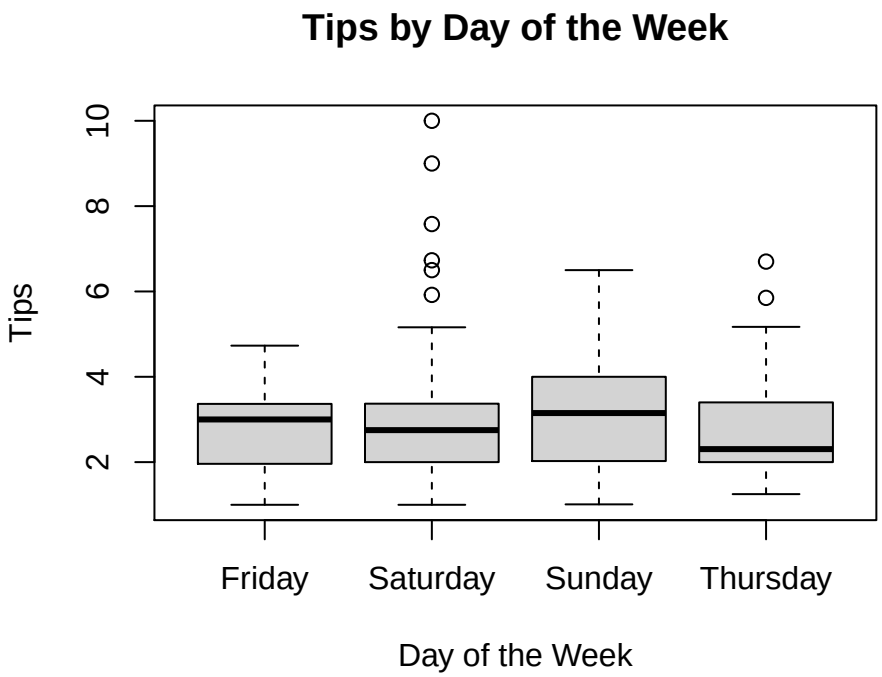
```
hist(TIPS$Tip, breaks = 100, main = "Tips, 100 bins")
```



7. Notice that there are a few "spikes" in the histogram above. What do you think is causing this?
   Saturdays Tips

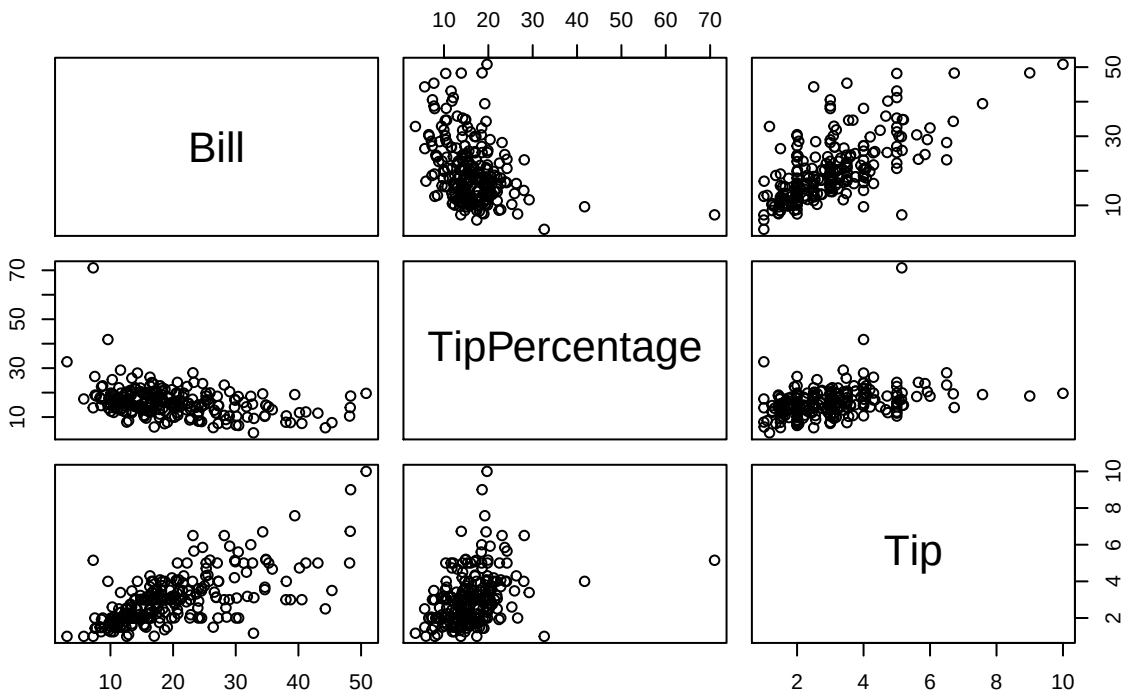
We can also summarize this numerical data broken down by one of the categorical variables using boxplots.

```
boxplot(Tip ~ Weekday, data = TIPS, main = "Tips by Day of the Week",
   xlab = "Day of the Week", ylab = "Tips")
```



Or we can visualize the relationship between a lot of our numerical variables at once.

```
pairs(~ Bill + TipPercentage + Tip, data = TIPS, main = "Scatterplot Matrix for TIPS")
```



8) Are there any clear linear relationships in the scatterplot above? What do you think is the explanation for these relationships?

   There is not a clear relationship between any of the variables There is a weak positive relationship between the Bill & Tip and Tip Percentage & Tip

There are lots of other interesting graphical summaries available for interpreting and displaying data. In addition, there are lots of R packages that allow you to draw these graphics and to further customize some of the ones we discussed here. In your projects, you are welcome to use any of these that you think are appropriate.

9) State a reasonable conjecture about tipping behavior that you would like to explore in the dataset. For example, you might think that people on dates tip more or that the waiter gets smaller tips when he has too many tables. Give *at least* one numerical and one graphical summary to explore this conjecture. Is there any evidence to support your conjecture?

   On average people in groups tip more.Based on the results of the EDA, tables with less than 2 people give less tips compared to tables with more than 2 people.

```
TIPS %>% group_by(PartySize) %>% count()
```

```
## # A tibble: 6 x 2
## # Groups:   PartySize [6]
##   PartySize     n
##       <int> <int>
## 1         1     4
```

```
## 2         2   156
## 3         3    38
## 4         4    37
## 5         5     5
## 6         6     4
```

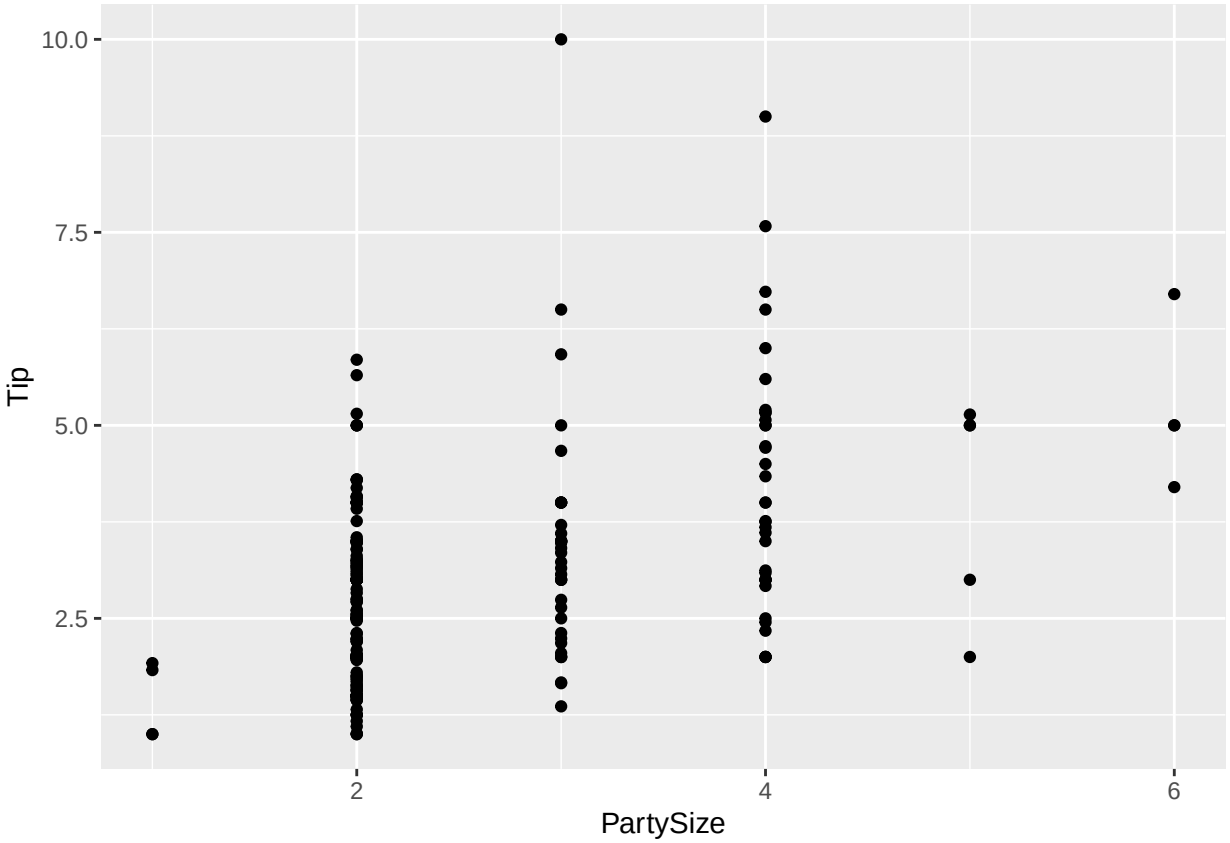
```
TIPS %>% group_by(Tip) %>% count()
```

```
## # A tibble: 123 x 2
## # Groups:   Tip [123]
##      Tip     n
##    <dbl> <int>
##  1  1        4
##  2  1.01     1
##  3  1.1      1
##  4  1.17     1
##  5  1.25     3
##  6  1.32     1
##  7  1.36     1
##  8  1.44     2
##  9  1.45     1
## 10  1.47     1
## # ... with 113 more rows
```

```
TIPS %>% group_by(Tip) %>% count()
```

```
## # A tibble: 123 x 2
## # Groups:   Tip [123]
##      Tip     n
##    <dbl> <int>
##  1  1        4
##  2  1.01     1
##  3  1.1      1
##  4  1.17     1
##  5  1.25     3
##  6  1.32     1
##  7  1.36     1
##  8  1.44     2
##  9  1.45     1
## 10  1.47     1
## # ... with 113 more rows
```

```
gf_point(Tip~ PartySize, data = TIPS)
```



It's okay if your conjecture is not supported or if you are just wrong–that's often the case in exploratory data analysis!