Scatterplot and Correlation in R

Scatterplot

In these notes, we are going to produce some scatter plots regarding seatshares in legislatures and how many cabinet positions parties receive. We will also plot some college football data. The necessary variables are in the data set `gamson.csv` and `CollegeFB.csv`.

```
# load data
#### make sure you change the path to where you put the data folder
gamson <- read.csv("~/Documents/GitHub/polisci209_fall2017/img/images/data/gamson.csv")
#### make sure you change the path to where you put the data folder

# quick look at data
tibble::glimpse(gamson)
```

```
## Observations: 826
## Variables: 2
## $ seat_share      <dbl> 0.02424242, 0.46060607, 0.51515150, 0.47204968...
## $ portfolio_share <dbl> 0.09090909, 0.36363637, 0.54545456, 0.45454547...
```

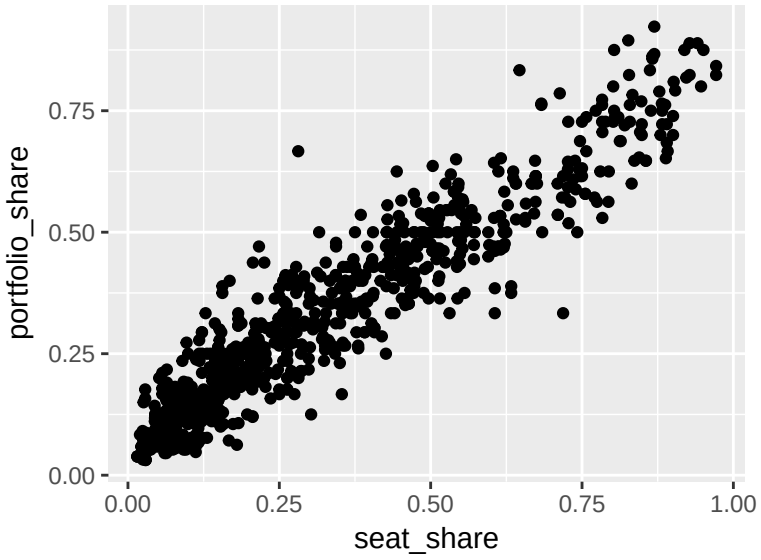
You can see that the data frame `gamson` has two variables: `seat_share`, and `portfolio`.

Of course, we're going to use ggplot, and we know we'll use `gamson` as the data frame.

In a scatterplot, we usually plot the variable *doing the causing* along the x-axis and the variable *being caused* along the y-axis. It makes sense here that seat shares cause portfolio shares because portfolio shares are determined well after seat shares are determined. Based on this rule, we can quickly figure out what the x and y aesthetics will be: `x = seat_share` and `y = portfolio_share`.

All that's left is the geometry. To create a scatterplot, we use `geom_point()`.

```
# load packages
library(ggplot2)

# create scatterplot
ggplot(gamson, aes(x = seat_share, y = portfolio_share)) +
  geom_point()
```



The Size and Color Aesthetics

Let's read in a different data set. This one is about rushers in college football in 2016/17.

```
# load data
cfb <- read.csv("~/Documents/GitHub/polisci209_fall2017/img/images/data/CollegeFB.csv")
#### make sure you change the path to where you put the data folder

# quick look at data
tibble::glimpse(cfb)
```

```
## Observations: 202
## Variables: 7
## $ Name     <fctr> Donnel Pumphrey, San Diego St. (Mountain West), D'On...
## $ Year     <fctr> Sr., Jr., Jr., Jr., Jr., Jr., Sr., Sr., Jr., So., Sr...
## $ Position <fctr> RB, RB, RB, RB, RB, RB, RB, RB, RB, QB, RB, QB, RB, ...
## $ Games    <dbl> 14, 11, 14, 12, 13, 13, 13, 13, 11, 13, 13, 13, 13, 1...
## $ Attempts <dbl> 349, 323, 349, 229, 288, 314, 258, 237, 253, 260, 232...
## $ Yards    <dbl> 2133, 2028, 1860, 1773, NA, 1709, 1629, 1621, 1603, 1...
## $ TDs      <dbl> 17, 15, 22, 17, 19, 23, 18, 27, 13, 21, 17, 18, 15, 1...
```

```
### let's see how many different values of position there are
unique(cfb$Position)
```

```
## [1] RB QB LB DB WR
## Levels: DB LB QB RB WR
```

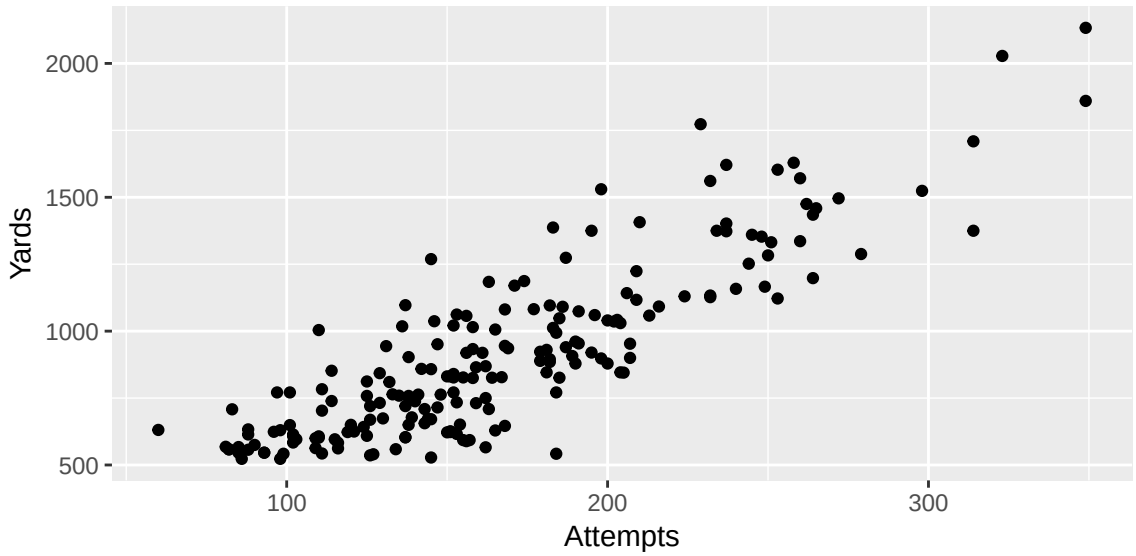
The data set includes the player's name, year in college, position, games played, rushing attempts, rushing yards, and total touchdowns scored. First, let's calculate a 'yard per attempt' and 'TD per attempt' statistic.

```
cfb$yards_carry <- cfb$Yards/cfb$Attempts
cfb$TDs_carry <- cfb$TDs/cfb$Attempts

### take another look at the data, now with the additional variables
tibble::glimpse(cfb)
```

```
## Observations: 202
## Variables: 9
## $ Name        <fctr> Donnel Pumphrey, San Diego St. (Mountain West), D...
## $ Year        <fctr> Sr., Jr., Jr., Jr., Jr., Jr., Sr., Sr., Jr., So.,...
## $ Position    <fctr> RB, RB, RB, RB, RB, RB, RB, RB, RB, QB, RB, QB, R...
## $ Games       <dbl> 14, 11, 14, 12, 13, 13, 13, 13, 11, 13, 13, 13, 13...
## $ Attempts    <dbl> 349, 323, 349, 229, 288, 314, 258, 237, 253, 260, ...
## $ Yards       <dbl> 2133, 2028, 1860, 1773, NA, 1709, 1629, 1621, 1603...
## $ TDs         <dbl> 17, 15, 22, 17, 19, 23, 18, 27, 13, 21, 17, 18, 15...
## $ yards_carry <dbl> 6.111748, 6.278638, 5.329513, 7.742358, NA, 5.4426...
## $ TDs_carry   <dbl> 0.04871060, 0.04643963, 0.06303725, 0.07423581, 0....
```

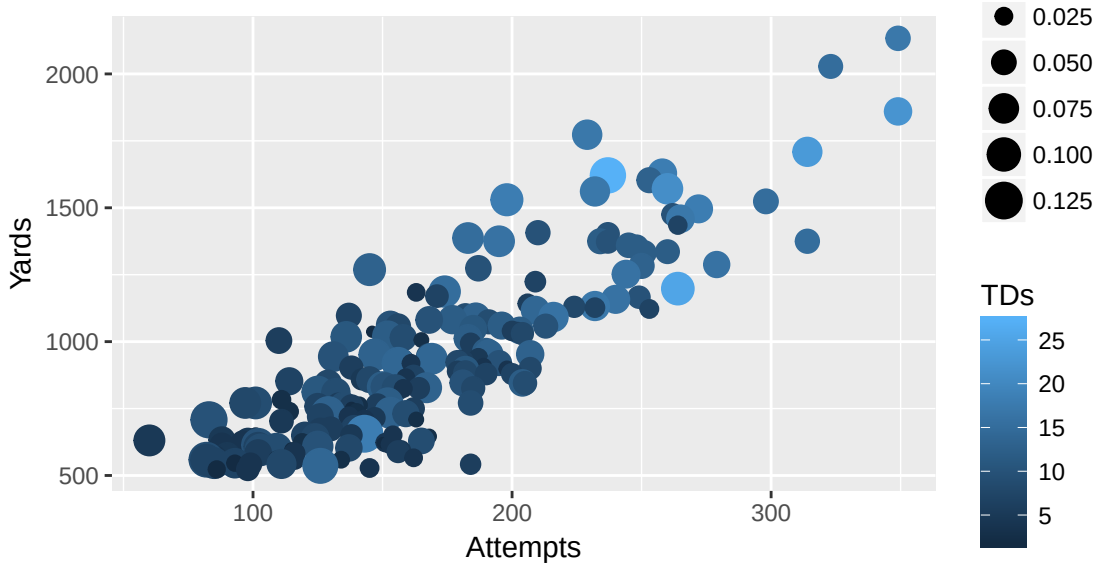
Let's explain the number of total yards in a season with the number of rushing attempts. Seems like those two should be related.

```
# create scatterplot
ggplot(cfb, aes(x = Attempts, y = Yards)) + geom_point()
```



Now maybe we would also like to see how these two variables are related to total number of touchdowns or TDs per carry. Since a scatterplot only has two spatial aesthetics (horizontal and vertical positioning), we'll have to use other aesthetics. Color and size are two options.

```
# create scatterplot with color and size aesthetics
ggplot(cfb, aes(x = Attempts, y = Yards, color = TDs, size = TDs_carry)) + geom_point()
```



Review Exercises

1. In creating a scatterplot, what variable do we typically place along the x- and y-axes?
2. What geometry creates a scatterplot?
3. Experiment with the x, y, size, and color aesthetics for the cfb data.
4. Many authors argue that district magnitude (the number of legislative seats in a districts) causes turnout. In particular, they argue that increasing district magnitude leads to an increase in turnout. The taiwan data set has information that might be useful in testing this hypothesis. Using the data set `taiwan.csv`, create the appropriate scatterplot and evaluate whether the data are consistent with the claim that district magnitude has a large, positive effect on turnout.

Correlation

In order to compute a correlation in R, we use the `cor()` function. The first argument `x` is the first of the two variables for which we would like to calculate a correlation. The second argument `y` is the second of the two variables. `cor()` is not designed to work with data frames, so we have to use the `data$variable` syntax.

Remember that the textbook defines a correlation as $r =$ average of (x in standard units)$\times$(y in standard units). For reasons similar to the SD, most computer programs, including R, divide by the number of entries $ - 1$ rather than the number of entries. For this reason, you'll see small differences between a correlation computed by R and a correlation computed by hand, especially when the number of observations is small. For practical purposes, though, the two approaches are equivalent, especially when we have many observations.

```
# calculate a correlation
cor(gamson$seat_share, gamson$portfolio_share)
```

```
## [1] 0.9423176
```

For the data frame `cfb` we have some missing data in the variables `Yards` and `Attempts`. If you were to feed this data to R to calculate the correlation, it will return `NA`. In order to drop the incomplete pairs (either x is missing, y is missing, or both), we just supply the argument `use = "pairwise.complete.obs"` to the `cor()` function.

```
# calculate a correlation
cor(cfb$Yards, cfb$Attempts) # returns NA
```

```
## [1] NA
```

```
cor(cfb$Yards, cfb$Attempts, use = "pairwise.complete.obs") # returns NA
```

```
## [1] 0.8625797
```

Review Exercises

1. What function do we use to calculate a correlation in R?
2. If some observations are missing, what argument to we use to drop those observations?
3. Does `cor()` take a `data` argument? If not, how do we calculate correlations for variables in data frames?
4. Using the data set `taiwan.csv`, calculate a correlation to assess whether the data are consistent with the claim that district magnitude has a large, positive effect on turnout.

Correlation Matrix

Sometimes we want to calculate correlations for many variables at the same time. Each of the correlations is computed in the usual way, except they are presented in a correlation matrix. The `cor` function allows us to compute a correlation matrix quickly if the first argument is a data frame rather than a vector. If the first argument is a data frame, then `cor()` computes correlations between every variables in the data frame.

Note: every variable in the data frame must be numeric.

If we don't want to compute correlations for every variable in the data frame, then we can use the `select()` function from the package dplyr to create a new data frame that includes only certain variables from the original data frame. The first argument to `select()` is the original data frame. The remaining arguments are the variables we want to keep.

```
# load package
library(dplyr)  # for select()
```

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

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

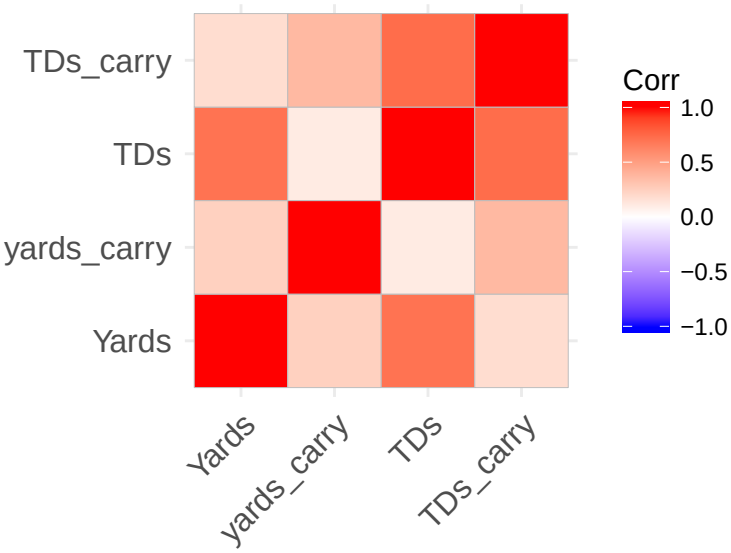
```
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
# keep only the miles, average_heart_rate, and minutes variables
numeric_vars <- select(cfb, Yards, yards_carry, TDs, TDs_carry)

# calculate the correlations between every variables in our new data frame
cor(numeric_vars, use = "pairwise.complete.obs")
```

```
##                Yards yards_carry       TDs TDs_carry
## Yards       1.0000000   0.2436051 0.7007229 0.1805367
## yards_carry 0.2436051   1.0000000 0.1068071 0.3657091
## TDs         0.7007229   0.1068071 1.0000000 0.7290759
## TDs_carry   0.1805367   0.3657091 0.7290759 1.0000000
```

We can also store these correlations as an object and create a ggplot that graphically communicates the correlations. The function `ggcorrplot()` in the package ggcorrplot does this automatically without us having to specify a data frame, aesthetics, or geometry. `ggcorrplot()` needs only one argument–the output of the `cor()` function.

```
# calculate the correlations between every variables in our new data frame
cors <- cor(numeric_vars, use = "pairwise.complete.obs")

# use ggcorrplot()
library(ggcorrplot)  # for ggcorrplot()
ggcorrplot(cors)
```



The data frame `health.csv` contains a lot of variables for which we might like to calculate correlations, so let's take a look.

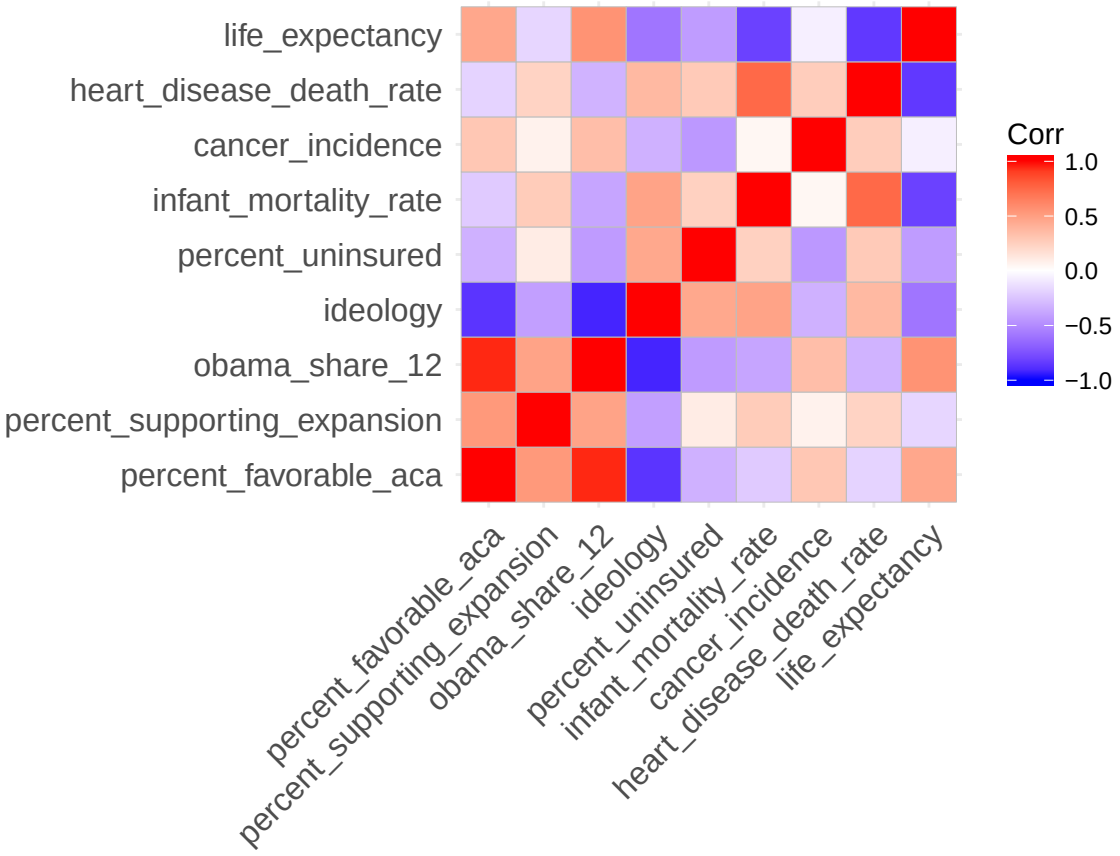
```
# load data
health <- read.csv("~/Documents/GitHub/polisci209_fall2017/img/images/data/health.csv")
#### make sure you change the path to where you put the data folder
```

```r
# quick look
tibble::glimpse(health)
```

```
## Observations: 50
## Variables: 17
## $ state                          <fctr> Alabama, Alaska, Arizona, Arkans...
## $ state_abbr                     <fctr> AL, AK, AZ, AR, CA, CO, CT, DE, ...
## $ gov_party                      <fctr> Repubican Governor, Repubican Go...
## $ sen_party                      <fctr> Republican Senate, Republican Se...
## $ house_party                    <fctr> Republican House, Republican Hou...
## $ percent_favorable_aca          <dbl> 38.2711, 37.4428, 39.6722, 36.162...
## $ percent_supporting_expansion   <dbl> 57.7616, 47.4247, 53.2125, 54.438...
## $ obama_share_12                 <dbl> 38.7838, 42.6847, 45.3866, 37.845...
## $ ideology                       <dbl> 0.2440440, 0.0472331, 0.1048640, ...
## $ percent_uninsured              <int> 14, 19, 18, 18, 19, 15, 8, 10, 21...
## $ infant_mortality_rate          <dbl> 9.2, 6.5, 6.4, 7.6, 5.1, 6.2, 6.1...
## $ cancer_incidence               <dbl> 472.9, 451.4, 387.1, 426.7, 434.0...
## $ heart_disease_death_rate       <dbl> 236.0, 151.5, 146.7, 222.5, 161.9...
## $ life_expectancy                <dbl> 75.4, 78.3, 79.6, 76.0, 80.8, 80....
## $ leg_party                      <fctr> Unified Republican Legislature, ...
## $ health_score                   <dbl> -2.0999900, 0.0484103, 0.6444630,...
## $ health_score_cat               <fctr> Bottom Tercile, Middle Tercile, ...
```

Let's use the `select()` function to create a new data frame with only numeric variables and then plot a correlation matrix for all of them.

```r
# keep only the miles, average_heart_rate, and minutes variables
numeric_vars <- select(health, percent_favorable_aca, percent_supporting_expansion,
                       obama_share_12, ideology, percent_uninsured,
                       infant_mortality_rate, cancer_incidence,
                       heart_disease_death_rate, life_expectancy)

# calculate the correlations between every variables in our new data frame
cors <- cor(numeric_vars, use = "pairwise.complete.obs")

# use ggcorrplot()
ggcorrplot(cors)
```



Review Exercises

1. How can we use the `cor()` function to compute many correlations at once?
2. What does the `select()` function do? What is the first argument? What are the subsequent arguments? What does it output?
3. What function can we use to plot a correlation matrix? What argument does it take?
4. Take a look at the very last figure in this document–the correlation matrix for the health data set. Why is the diagonal completely red?
5. Looking at that same figure, how do the measures of state health (i.e., infant mortality rate, cancer incidence, heart disease death rate, and life expectancy) correlate with support for the ACA? Which correlations are in the expected direction?