Data Summaries / Prediction, Homework

Data Science Team

```r
if (!require(pacman)) {install.packages(pacman)}
```

```
## Loading required package: pacman
```

```r
pacman::p_load(ggplot2,readr,tidyr,dplyr)

# the code in this section loads libraries, installing them as necessary
# you may need to add more libraries
```

Due Tuesday July 19 at 10am

Question 1: Mean and Median

a. Generate a numerical variable with 50 values. Do this with one of the functions to generate random numbers in R such as rnorm() (if you look up the help for this function you will see other options). Here's an example of how to use rnorm:

```r
x <- rnorm(20)
```

Calculate the range (min, max) of these values and call it [a,b]. Calculate the arithmetic mean and median of the values.

b. Now, choose one of the 50 values you have generated (it might as well be the first one x_1), and study what happens to the mean and median of the dataset as you change that value. You are going to change x_1 to take on, one by one, the values $y_1, \ldots, y_{100}$, where $y_1 = a - 2(b - a)$ and $y_{100} = a + 2(b - a)$, and the rest of the y_i are equispaced in between (you might find the function `seq()` and its `length.out` argument useful to generate y). This will lead you to have 100 different datasets (with the remaining values $x_2, \ldots, x_{50}$ fixed). Calculate the mean and median for each of the datasets and show with a graphical display how mean and median change as the values of x_1 change.

Hint: the "easy" way to do this is with the `sapply` function. sapply is one of several "apply" functions in R. Apply is possibly the most useful general purpose function in R. It allows you to write a function and then repeatedly call that function on a list of values without writing a loop. Here is an example that computes the mean of a vector *leaving out* a different one of the entries every time. You can modify this example to obtain the code you need for question b.

```r
loo.mean <- function(j,x) {
  # x : input vector
  # j : element to leave out
  return(mean(x[-j]))
}

z <- rnorm(20)
# sapply wants a list; the list should correspond
# to the *first* argument of the function you are "applying"
```

```r
id.lo <- as.list(seq(20))


mns <- sapply(id.lo,loo.mean,x=z)
```

c. Describe what one can learn from these calculations about the different behavior of median and mean.

Question 2: Diversity index

Recall that the Gini index for diversity for a variable with possible values $v_1, \dots, v_m$ recorded with relative frequencies $p_1, \dots, p_m$

$$\begin{array}{cccc} v_1 & v_2 & \cdots & v_m \\ p_1 & p_2 & \cdots & p_m \end{array}$$

is defined as follows:

$$D = 1 - \sum_{i=1}^{m} p_i^2$$

Show via simulation, that means by a large enough number of trials (e.g. 10,000), that

$$D \leq 1 - \frac{1}{m}$$

(Note: Instead of the simulation, a mathematical derivation is also fine).

Question 3: Linear Regression

I collect a set of data ($n = 100$ observations) containing a single predictor and a quantitative response. I then fit a linear regression model $Y = \beta_0 + \beta_1 x + \epsilon$ to the data, as well as a separate cubic regression, i.e. $Y = \beta_0 + \beta_1 X + \beta_2 X^2 + \beta_3 X^3 + \epsilon$.

(a). Suppose that the **true** relationship between X and Y is linear, i.e. $Y = \beta_0 + \beta_1 X + \epsilon$. Consider the training residual sum of squares (RSS) for the linear regression, defined by

$$RSS_{\text{train}} = \sum_{i=1}^{n} (y_i^{\text{train}} - \hat{y}_i)^2$$

and also the training RSS for the cubic regression. Would we expect one to be lower than the other, would we expect them to be the same, or is there not enough information to tell? Justify your answer. What would your answer be if we were considering the *test* rather than the *training* RSS?

(b). Suppose that the **true** relationship between X and Y is **not** linear, but we don't know how far it is from linear. Consider the training RSS for the linear regression, and also the training RSS for the cubic regression. Would we expect one to be lower than the other, would we expect them to be the same, or is there not enough information to tell? Justify your answer. What would your answer be if we were considering the *test* rather than the *training* RSS?

(c). You can work out the answers to this part by hand on a piece of paper and attach to the file that you upload. Consider the fitted values that result from performing linear regression without an intercept. In this setting, the ith fitted value takes the form

$$\hat{y}_i = x_i \hat{\beta},$$

where

$$\hat{\beta} = \frac{\sum_{i=1}^n x_i y_i}{\sum_{i=1}^n x_i^2}$$

Show that we can write

$$\hat{y}_i = \sum_{j=1}^n a_{i,j} y_j.$$

What is $a_{i,j}$?

(d). You can work out the answers to this part by hand on a piece of paper and attach to the file that you upload. Now, in the setting of part (c), consider the case where we include an intercept, so the model is $y = \beta_0 + \beta_1 x + \epsilon$. For this question, you can use the following formulae:

$$\hat{\beta}_1 = \frac{\sum_{i=1}^n (x_i - \bar{x})(y_i - \bar{y})}{\sum_{i=1}^n (x_i - \bar{x})^2}$$
$$\hat{\beta}_0 = \bar{y} - \hat{\beta}_1 \cdot \bar{x}.$$

Show that in the case of a linear regression model with intercept, the regression line passes through the point $(\bar{x}, \bar{y})$. Do you think this also holds for the regression without an intercept (briefly explain)?

Question 4: Multiple Linear Regression

This question involves the use of multiple linear regression on the Auto data set. You can get this from the ISLR package in R (install from CRAN)

```
pacman::p_load(ISLR)
data(Auto)
```

(a). Produce a bivariate plot matrix and the matrix of correlations between the variables. You will need to exclude the name variable, which is qualitative. You can calculate the matrix of correlations using the function `cor()`. You can produce the bivariate plot matrix using e.g. `GGally` or base R graphics which includes all of the variables in the data set except `name`. (The function `ggpairs` in `GGally` is a nicer alternative to the more limited `pairs` function in basic R.)

```
pacman::p_load(dplyr)
if (!require(GGally)) {install.packages(GGally,type='source')}
```

(b). Use the lm() function to perform a multiple linear regression with mpg as the response and all other variables except name as the predictors. Use the summary() function to print the results. Comment on the output. For instance:

- Is there a relationship between the predictors and the response?
- Which predictors appear to have a statistically significant relationship to the response?
- What does the coefficient for the `year` variable suggest?
- Interpret the coefficient on weight. What does the sign mean? What does the magnitude mean? (in this case you should be able to interpret it exactly – look up the units for the different variables here.)

(c). Use the `plot` function to produce diagnostic plots of the linear regression fit (or `autoplot` which requires libraries `tidyverse` and `ggfortify`). Comment on any problems you see with the fit. Do the residual plots suggest any unusually large outliers? Does the leverage plot identify any observations with unusually high leverage?

(d). Try a few different transformations of the response, such as $\log(Y)$, $\sqrt{Y}$, Y^2. Comment on your findings.