Lab 3 Activity

For this activity we will be looking at some data about the 1907 Romanian peasant revolt! The data is called `Chirot` and is stored in the `carData` package. Run `help("Chirot")` to find out more about the data.

1. Load the `carData` and create a separate `data.frame` object called `rev_int` that only contains the the `intensity` and `inequality` variables from the `chirot` data.

2. Find the correlation between `inequality` and `intesity`. Is it significantly different from 0? What is the 95% confidence interval?

3. Run a linear regression with `inequality` predicting `intensity`. Report the estimated regression coefficient in APA style. Do you notice any similarities between the regression coefficient of `inequality` and the correlation you calculated in the previous question? (yes, what is the same?)

4. According to the previous regression model, what happens to the `intensity` once `inequality` increases by 3 units?

5. What is the predicted value of `intensity` if `inequality` is 10? Make sure there is no rounding error in the answer (i.e., do not copy and paste numbers from the output).

6. Create a QQplot that evaluates the normality of residuals. What do you think?

- Afterwards, create a scatterplot with `inequality` on the x-axis and `intensity` on the y-axis to check whether a linear relationship is appropriate for the two variables. What do you think?

Some R Practice: `for` loops

Loops are used to repeat an operation a certain number of times based on certain criteria. There are different types of loops, but the most used type of loop across computer languages is the `for` loop.

Here is an example of a `for` loop:

```r
for(i in 1:10){

  print(2*i)
}
```

```
## [1] 2
## [1] 4
## [1] 6
## [1] 8
## [1] 10
## [1] 12
## [1] 14
## [1] 16
## [1] 18
## [1] 20
```

Did you see what happened? the loop ran *10 times*, determined by the `1:10` part. Remember that `1:10` actually represents:

```r
1:10
```

```
##  [1]  1  2  3  4  5  6  7  8  9 10
```

Now, the loop was repeated 10 times, but every time `i`, the *index*, was substituted for one of the numbers in `1:10`. So every time the loop multiplied 2 by a number from 1 to 10.

This is a simple example, but loops are used in many ways. One way that loops are often used is to save a series of results to some object. The task requires you to do one such operation:.

Task

- You have the following matrix `dat_ex` and the empty object `SDs`.

```r
dat_ex <- cbind(c(1,4,6,77,3,12),
                c(87,789,35,72,33,4),
                c(134,455,66,776,397,1289))

# create an empty object

SDs <- c()
```

Create a `for` loop that stores the standard deviations of each of the columns of `dat_ex` into the `SDs` object. At the end of the `for` loop, the `SDs` object should look like this:

```
SDs
```

```
## [1]  29.55278 304.69788 454.17210
```

To solve this task, it is important to know how to index vectors and matrices as discussed in the R practice from Lab 1.

Note: use the `sd()` function calculate the standard deviation.