Tutorial 1

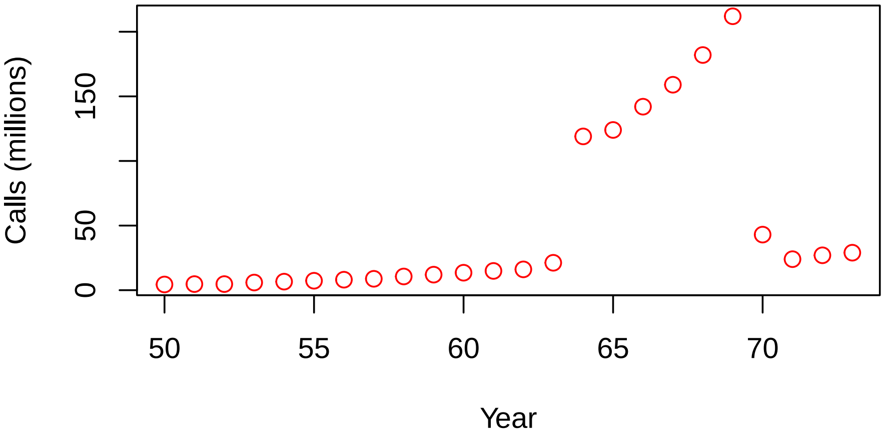
ST2137-2420

Material

This tutorial covers material from chapter 1 of the course textbook. It provides practice on basic R functions. There are many ways to code each question, so try them more than once. The following R functions may be helpful: `combn`, `median`, `union`, `intersect`.

Dataset: `phones`

The dataset `phones` is available from the `MASS` package (which is installed by default with R). It contains two numeric variables, in a list format. Here is a plot of the two variables:



1. Create a dataframe `df1` with the following columns.

```
df1 <- data.frame(x=phones$year, y=phones$calls)
head(df1)
```

```
   x   y
1 50 4.4
2 51 4.7
3 52 4.7
4 53 5.9
5 54 6.6
6 55 7.3
```

2. Write this dataframe to a csv file in the `data/` folder named `phones-2420.csv`.

```
write.csv(df1, "data/phones-2420.csv", row.names=FALSE)
```

These are the first few lines of the file:

```
"x","y"
50,4.4
51,4.7
52,4.7
53,5.9
...
```

3. Answer the following queries about the data:

    1. How many rows are there in the dataset?
    2. How many observations between 100 and 200 million calls?
    3. What are the largest 3 and smallest 3 number of calls?
    4. In which year was the largest number of calls made?

```r
# number of rows:
NROW(df1)
```

```
[1] 24
```

```r
# num. observations between 100 and 200
sum(df1$y > 100 & df1$y < 200)
```

```
[1] 5
```

```r
# smallest 3 calls
head(sort(df1$y), n=3)
```

```
[1] 4.4 4.7 4.7
```

```r
# largest 3 calls
tail(sort(df1$y), n=3)
```

```
[1] 159 182 212
```

```r
# year with largest number of calls.
df1$x[df1$y == max(df1$y)]
```

```
[1] 69
```

4. In R, matrix multiplication is carried out with the `%*%` operator. For instance, if we have

$$x_{1,2} = \begin{bmatrix}1 & 1\end{bmatrix} \quad y_{2,2} = \begin{bmatrix}0 & 0.5\\ 0.5 & 0\end{bmatrix}$$

Then $x \times y$ is computed as

```r
x_mat = matrix(c(1,1), nrow=1)
y_mat = matrix(c(0, 0.5, 0.5, 0), nrow=2)
# solve(y_mat) # computes inverse of a square matrix
# t(y_may)     # returns transpose of a matrix
```

1. Create a 24×2 matrix $\mathbf{X}$ with the first column all ones, and the second column containing the `year` vector from the `phones` data. Now create a 24×1 matrix $\mathbf{y}$ containing the `calls` column.
2. Compute the estimate of the slope and intercept for a least-squares best fit to the above data, storing it as `beta_hat`.

$$(\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{y}$$

3. Compute the fitted y-values, storing them as `y_hat`.

$$\mathbf{X}(\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{y}$$

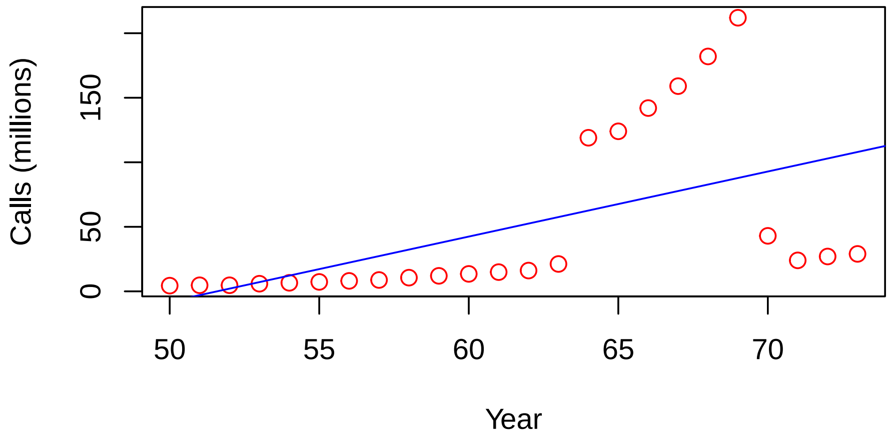
```r
X <- cbind(1, df1$x)
y <- matrix(df1$y, ncol=1)
beta_hat <- solve(t(X) %*% X) %*% t(X) %*% y
y_hat <- X %*% beta_hat
```

5. The `lm` function performs the above computation for us in R. Inspect the output object and retrieve the parameters and the fitted values.

```r
lm_output <- lm(y ~ x, data=df1)
lm_output$coefficients["x"]
```

```
       x 
5.041478 
```

Here is a plot of the line fitted to the data:



6. The fit of the line has been affected by the anomalous points. Here is an algorithm to compute a fitted line that is not so affected by those points:

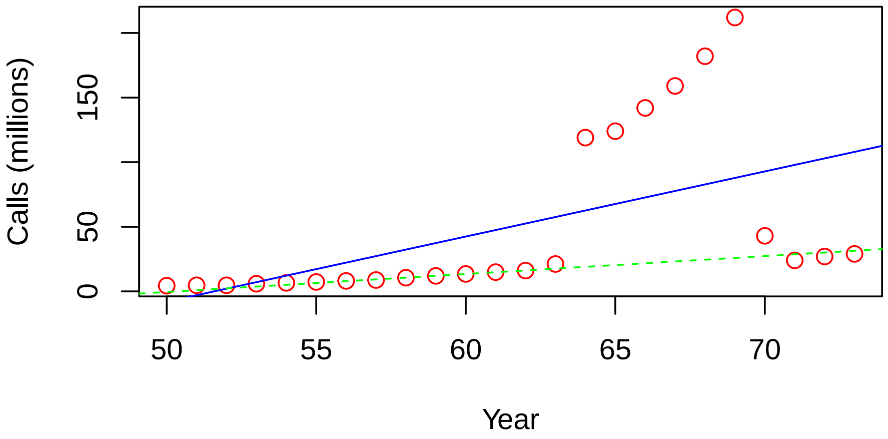
   a. Generate all pairwise combinations of observations. (see `combn`)
   b. For each pair of points, compute the gradient.
   c. Compute the median over all these gradients. This returns the fitted slope.

   Write a for-loop that will compute this median. Compare it to the earlier slope. Here is a plot with the new slope in green:

```r
all_combn <- combn(24, 2)
all_slopes <- rep(0.0, length=276)
for(ii in 1:276) {
  y_vec <- df1$y[all_combn[,ii]]
  x_vec <- df1$x[all_combn[,ii]]
  all_slopes[ii] <- (y_vec[2] - y_vec[1])/(x_vec[2] - x_vec[1])
}
fit_slope <- median(all_slopes)
fit_intercept <- median(df1$y) - median(df1$x)*fit_slope

## plotting code (not asked for in this question)
plot(x=phones$year, y=phones$calls, col="red", cex=1.2,
     xlab="Year", ylab="Calls (millions)")
```

```
abline(lm_output, col="blue")
abline(a=fit_intercept, b=fit_slope, col="green", lty=2)
```



7. The file `phones.json` contains corrected readings for particular years. The following commands will read the data into R as a list. Replace the data in `df1` at the appropriate years with the corrected call values. *Hint: read up on `match()` function.*

```
library(jsonlite)
corrected_data <- read_json("data/phones.json", TRUE)

matched_rows <- match(corrected_data$year, df1$x)
df1$y[matched_rows] <- corrected_data$corrected_calls
```