My grades for **In-Class Lab 1**

Q1

1 / 1

Please upload your group's PDF lab report here.

Module 1 In-Class Lab #1

due Thursday September 23 by 11:59PM ET

General Instructions

1. Take the first 5 minutes to introduce yourselves, telling everyone your name, your degree program, and one thing you're excited to do on the weekend.

2. Decide amongst yourselves on the role that each student will perform and add the names to the role below:
   - Timekeeper: Jialong Yuan
   - Submission Manager: Kaibo Li
   - Live Coder: Feihao Qu
   - Moderator: Weihan Wang
3. Get set up for conducting your roles by making sure everyone can see the shared screen (shared by the LIVE CODER) and that everyone knows who will be contributing verbally and/or by chat and can see/hear each other (MODERATOR should keep track).
   - The SUBMISSION MANAGER should use this time to access the Crowdmark link and create a group submission link by adding the group members to the group.
4. To get help at any time, anyone in the group (or the HELP FINDER if there is one) can tag the instructor by typing @Katherine Daignault or the TA (TBD) in the chat with a question. The TIME-KEEPER should keep track of how long the group spends on each part so that the group will be able to finish the lab during class time.

Submission Instructions

All students will receive an email from Crowdmark which will be used to submit the knitted PDF you produce in your group. YOU WILL NEED TO CREATE YOUR GROUP BEFORE SUBMITTING. To do this:

1. The SUBMISSION MANAGER on the team should use the emailed link to access the assignment page on Crowdmark.

2. There will be an option to add group members to the submission.

3. Using the names you've entered for the group roles above, search for your teammates and add them to your group.

4. All teammembers will receive an email from Crowdmark stating they have been added to the group.

5. At the end of the lab (or before the submission deadline), the SUBMISSION MANAGER should upload the PDF you create from this document to Crowdmark using the group submission link that was created. This will submit the lab for everyone :)

1

Lab 1 - Motivating the Regression Line of Best Fit

Summary

This lab is meant to get you playing with R and learning some coding practices and tools from others. It is also meant to have you perform a small exploratory data analysis and comment on the results. Lastly it will motivate the interpretation of the line of best fit that a linear regression model fits to data.

The data this week are 200 observations from a cleaning company. The information in this dataset is:

- Case: the observation number (just an identifier, won't be used)
- Crews: the number of members in a cleaning crew sent to a job
- Rooms: the number of rooms cleaned by that cleaning crew

Part 1: Load the data and perform some data explorations

In this part, you will need to:

1. Load the dataset (cleaning_sim.csv) that can be downloaded from Quercus (ensure that you allow the code to be printed in the report). Don't forget to give your dataset a meaningful name.

```
library(tidyverse)
```

```
## Warning in system("timedatectl", intern = TRUE): running command 'timedatectl'
## had status 1

## -- Attaching packages --------------------------------------- tidyverse 1.3.0 --

## v ggplot2 3.3.3     v purrr   0.3.4
## v tibble  3.1.2     v dplyr   1.0.6
## v tidyr   1.1.3     v stringr 1.4.0
## v readr   1.4.0     v forcats 0.5.1

## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```
data <- read_csv("cleaning_sim.csv")
```

```
##
## -- Column specification --------------------------------------------------------
## cols(
##   Case = col_double(),
##   Crews = col_double(),
##   Rooms = col_double()
## )
```

2. Find the mean and standard deviation of the numerical variables and ensure that the result prints in the report. Also make a boxplot of each of the variables.

```
#  mean and standard deviation of Crews
mean(data$Crews)
```

```
## [1] 5.5
```

```
sd(data$Crews)
```

```
## [1] 2.879489
```

```
#  mean and standard deviation of Rooms
mean(data$Rooms)
```

```
## [1] 20.245
```

```
sd(data$Rooms)
```

```
## [1] 11.05968
```

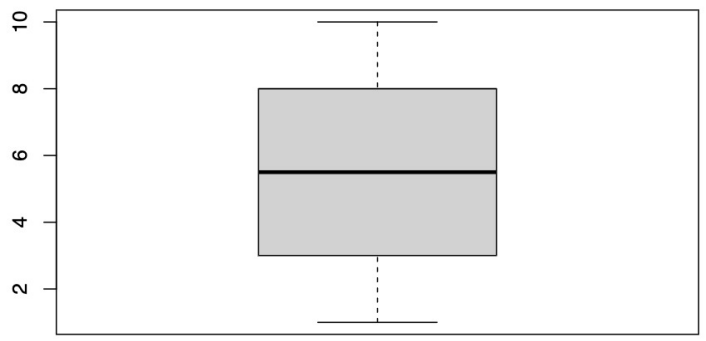
```
#  mean and standard deviation of Case
mean(data$Case)
```

```
## [1] 100.5
```

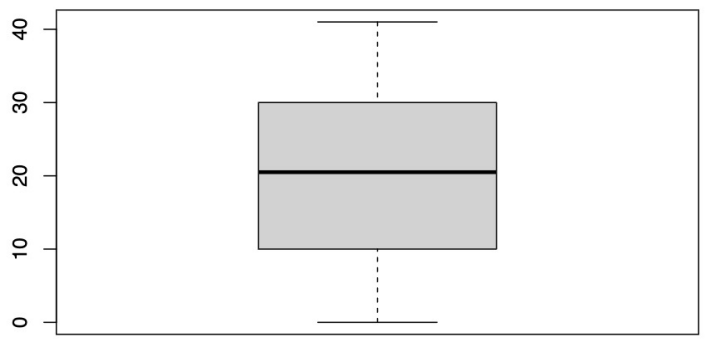
```
sd(data$Case)
```

```
## [1] 57.87918
```

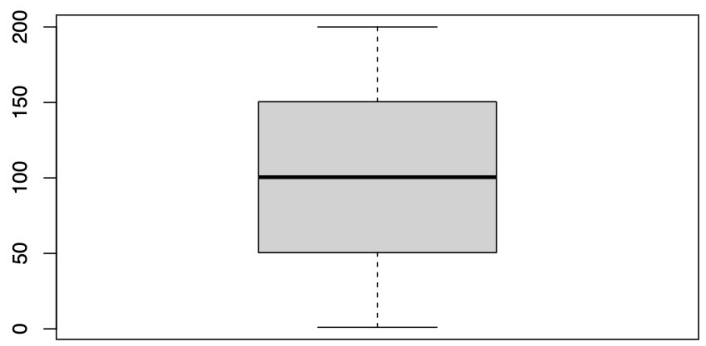
```
# Boxplot for Crews Rooms and Case
boxplot(data$Crews)
```




```
boxplot(data$Rooms)
```



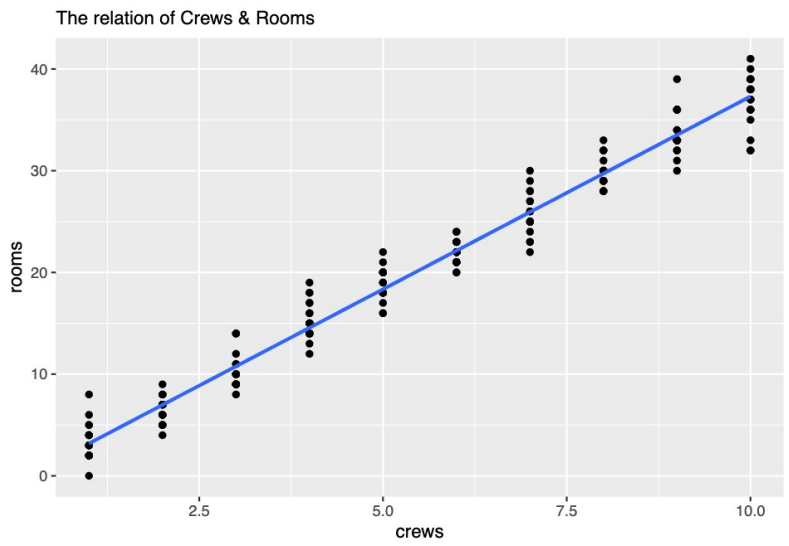

```
boxplot(data$Case)
```



3. Produce a scatterplot of the two variables, with Rooms as the y-axis and Crews on the x-axis, with appropriate title and axis labels.

```
#plot code
ggplot(data=data, aes(x=Crews , y=Rooms)) + geom_point() + geom_smooth(method=lm , se=FALSE) + labs(x='
```

```
## `geom_smooth()` using formula 'y ~ x'
```



It is easy to see there is a positive linear relationship.

Part 2: Looking at the distribution of Rooms within each value of Crews

For this section, we are going to work with the fact that we have discrete values of Crews, and can therefore directly look at the distribution of Rooms at each unique value of Crews. To do this, we will:

1. Create a new variable that lists the unique values of your Crews variable.

```
# help(unique)
new_crews <-unique(data$Crews, incomparables = FALSE,MARGIN = 1, fromLast = FALSE)
new_crews
```

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

2. Find the mean of Rooms at each unique value of Crews. This will require a **for loop** which has been started for you. To run, you will need to un-comment the code by deleting the number signs, and to replace the following variables in the code below to match the variable names you have:

   - unique_variable: this is the variable from the previous step
   - data$...: replace data with whatever you called your dataset at the start

```
ymeans <- NULL
 for(i in new_crews){
   values <- data$Rooms[data$Crews==i]
   ymeans <- c(ymeans, mean(values))
 }
ymeans
```

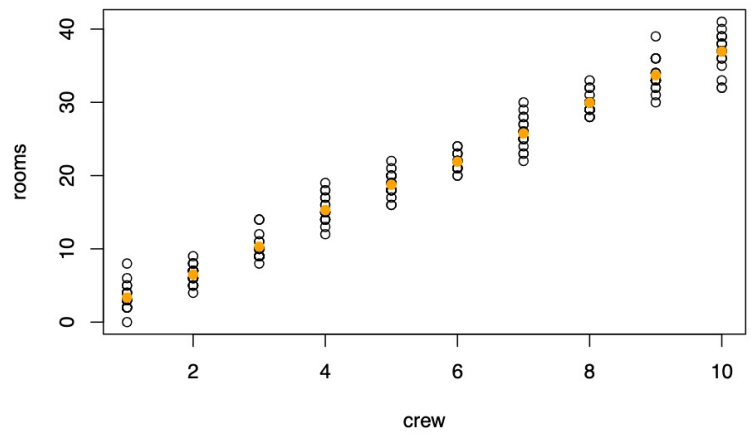
```
##  [1]  3.35  6.50 10.30 15.30 18.75 21.90 25.75 29.95 33.70 36.95
```

```
help(c)
```

Yes. Firstly, it creats ymeans to save data that will be produced in the for loop. In the for loop, it selects Rooms corresponding to the unique value of Crews. Then, the means of each value of Crews from 1-10 is calculated and stored into ymeans.

3. Now let's take our previous scatterplot and add these means:

```
plot(data$Crews,data$Rooms,xlab="crew",ylab="rooms")
points(ymeans~new_crews, col="orange", pch=19)
```



The means of Rooms increased as Crew increases.

Part 3: Digging deeper into the change in means as X increases

This last part will look more deeply into whether there is a consistent trend associated with how the mean of Rooms changes as Crews increases.

1. Let's investigate the difference between these means as Crews increases. First let's sort our means into increasing order and then take the difference between the subsequent values:

```
sorted <- sort(ymeans)

diffs <- NULL
for(i in 1:(length(sorted)-1)){
  diffs <- c(diffs, sorted[i+1]-sorted[i])
}
diffs
```

```
## [1] 3.15 3.80 5.00 3.45 3.15 3.85 4.20 3.75 3.25
```

2. Find the mean of these differences.

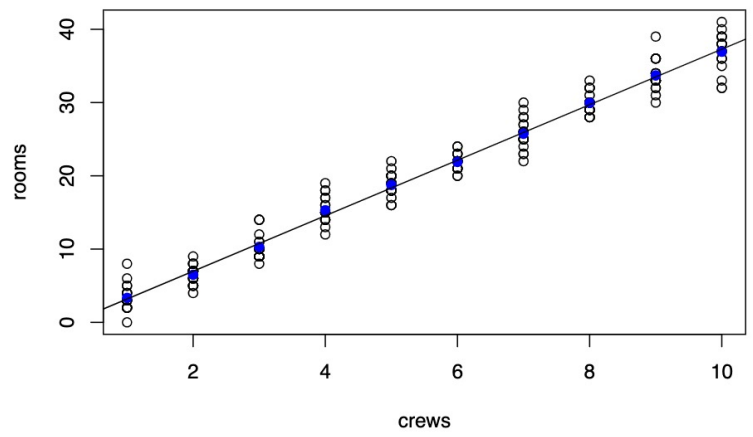
```
# mean
mean(diffs)
```

```
## [1] 3.733333
```

This value represents the mean of Crews is 3.73

3. A line of best fit has been found for these data, which has a slope of 3.79 and an intercept of -0.61, i.e. $y = -0.61 + 3.79x$. Add a line with these properties to the scatterplot from earlier that also has the means.

```
plot(data$Crews,data$Rooms,xlab="crews",ylab="rooms")
points(ymeans~new_crews, col="blue", pch=19)
abline(a =-0.61, b =3.79)
```



The line of best fit aligns with each mean for different Crew values. The trend mean values of Crew match the line of best fit for the entire dataset.