CWRU DSCI351-351m-451: Lab Exercise LE3

Inference, Exploratory Data Analysis

Prof.:Roger French, TAs: Raymond Wieser, Sameera Nalin Venkat, Mingxuan Li

03 October, 2021

Contents

3.0.1 LE3, 10 points, 2 questions.

Use Cntrl + Shift + O to see the summary of questions.

3.0.1.1 Lab Exercise (LE) 3

3.1 LE3-Q1. Inference: (4.5 points)

Inference Guide

There is a useful Inference Cheat Sheet in your readings folder

- ois4_extra_inference_guide.pdf

3.1.1 Twitter users and News:

3.1.1.1 LE3-Q1.1 (OIS Exercise 5.8) (0.5 points)

A poll conducted in 2013 found that

- 52% of U.S. adult Twitter users get at least some news on Twitter.[@mitchell_twitter_2013]
- The standard error for this estimate was 2.4%,
  - and a normal distribution may be used to model the sample proportion.

Construct a 99% confidence interval for

- the fraction of U.S. adult Twitter users
- who get some news on Twitter,

and interpret the confidence interval (CI) in context.

```
library(tidyverse)
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.1 --
```

```
## v ggplot2 3.3.5     v purrr   0.3.4
## v tibble  3.1.4     v dplyr   1.0.7
## v tidyr   1.1.4     v stringr 1.4.0
## v readr   2.0.2     v forcats 0.5.1
```

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

```
p_hat = 0.52        # point_estimate from a given sample
se  = 2.4/100       # sample error that comes from Sqrt(pq/n)

cl = 0.99
ci = 2.58

solLwr = p_hat - (2.58 * se)
solUpr = p_hat + (2.58 * se)
```

ANSWER (interpret the CI) -> The confidence intervel is (45.808 , 58.192) This means that with 99% confidence we can say that the true population parameter: percentage of all adult twitter users who get their news from twitter lie in between (45.808 to 58.192 percentage)

3.1.1.2 LE3-Q1.2 (OIS Exercise 5.9) (0.5 points)

Identify each of the following statements as true or false. Provide an explanation to justify each of your answers.

(a) The data provide statistically significant evidence that

- more than half of U.S. adult Twitter users
  - get some news through Twitter.
- Use a significance level of $\alpha = 0.01$.

ANSWER -> p_val = 0.797 : There is high chance that 50% occurs in this simulation and therefore, we accept the null hypothesis.

```
H0 <- " More than half of twitter users get some news from twitter : p >50% "
Ha <- " It could be less "

se  = 2.4/100
z = (0.52 - 0.50) /se
```

```
p_val = pnorm(z)
```

(b) Since the standard error is 2.4%,

- we can conclude that 97.6% of all U.S. adult Twitter users
  - were included in the study.

ANSWER -> Standard error implies the difference between the point estimate parameters of all the samples collected, it only talks about the collection data which could be much less than the actual population size

(c) If we want to reduce the standard error of the estimate,

- we should collect less data.

ANSWER -> No, standard error is inversely proportional to sqrt(observations)

(d) If we construct a 90% confidence interval

- for the percentage of U.S. adults Twitter users
  - who get some news through Twitter,
- this confidence interval will be wider
  - than a corresponding 99% confidence interval.

ANSWER -> No, When we decrease the confidence level, we also decrease the width of the confidence intervel.

3.1.2 Gifted children:

3.1.2.1 LE3-Q1.3 (0.5 points) Researchers investigating characteristics of gifted children

- collected data from schools in a large city
  - on a random sample
- of thirty-six children
  - who were identified as gifted children
  - soon after they reached the age of four.

The following histogram shows

- the distribution of the ages (in months)
- at which these children first counted to 10 successfully.

Also provided are some sample statistics.[@graybill_regression_1994]

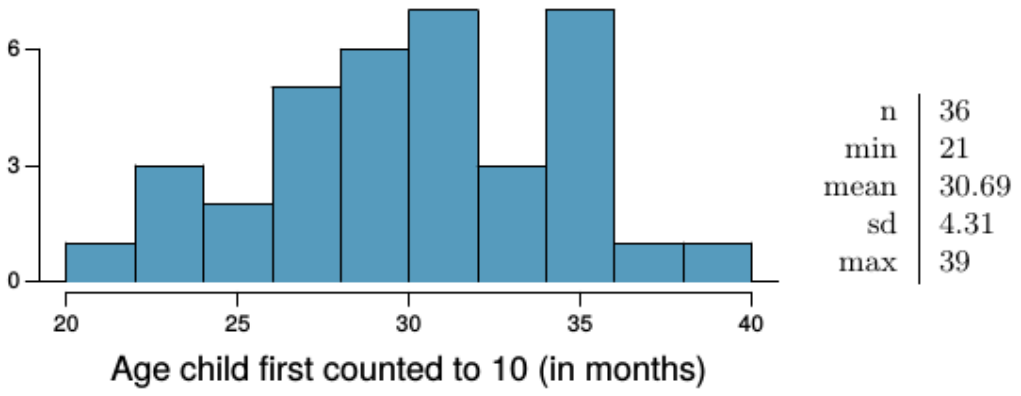


Figure 1: Age Child First Counted to 10

(a) Are conditions for inference satisfied?

ANSWER -> There are n = 36 childrean, and in the sample distribution above, we cannot conclusively say that the distribution is approaching a bell curve. The other condition for inference, at least 10 observations of the point estimate, in this particular question, none of the observations have a frequency of 10 or above. So there is not much ground for inference.

(b) Suppose you read online that children

- first count to 10 successfully when they are 32 months old, on average.

Perform a hypothesis test to evaluate

- if these data provide convincing evidence that
- the average age at which gifted children fist count to 10 successfully
  - is less than the general average of 32 months.
- Use a significance level of 0.10.

ANSWER -> since Pval = 0.034 < 0.10, we have to neglect the null hypothesis, and favour the alternate.

```
H0 <- "Average age for gifted and average children is same"
HA <- " Average age for gifted is less than general average"
n = 36
sd = 4.31
pe = 30.69

se = sd /sqrt(n)

z = (pe - 32) / se
pval = pnorm(z)
```

(c) Interpret the p-value in context

- of the hypothesis test
- and the data.

ANSWER -> The frequency of occurence of 32 months in the sample simulation is lower than significance value, that means that in a sample of gifted children most likely the value that occurs is something lower than 32.

(d) Calculate a 90% confidence interval

- for the average age at which gifted children
  - first count to 10 successfully.

ANSWER ->(29.508, 31.871)

```
n = 36
sd = 4.31
pe = 30.69

se = sd /sqrt(n)
cl = 0.90
z = 1.645

solLwr = pe - (z * se)
solUpr = pe + (z * se)
```

(e) Do your results from

- the hypothesis test and

- the confidence interval agree?

Explain.

ANSWER -> Yes, they agree, both point to the left side of 32 months ,i.e. less than 32 months.

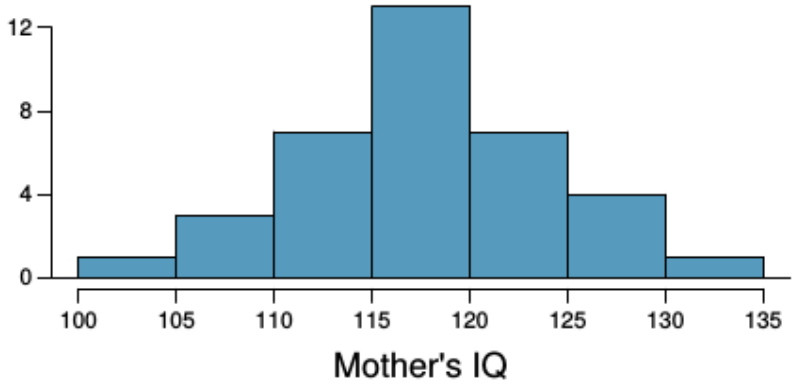
3.1.2.2 LE3-Q1.4 (0.5 points)

Gifted children Part I describes a study on gifted children.

In this study, along with variables on the children,

- the researchers also collected data
  - on the motherâ€™s and fatherâ€™s IQ
  - of the 36 randomly sampled gifted children.

The histogram below shows the distribution of motherâ€™s IQ. Also provided are some sample statistics.



n	36
min	101
mean	118.2
sd	6.5
max	131

Figure 2: Mother's IQ

(a) Perform a hypothesis test

- to evaluate if these data provide convincing evidence
  - that the average IQ of mothers of gifted children
- is different than the average IQ for the population at large,
  - which is 100.
- Use a significance level of 0.10.

ANSWER -> since p-val = 1 » 0.10, there is a very high chance of getting 100 in this distribution, which means we have to agree with the null hypothesis, the average IQ of mum's of gifted children is 100.

```
p_hat = 118.2 # point estimate
H0 <- "There is nothing special about the IQ of mothers,and population mean of
children's mum is same as of the general population mu = 100"
HA <- "Maybe there is an influence"

sd = 6.5
se  = sd /sqrt(36)

z = (p_hat - 100 ) / se
p_val <- pnorm(z)
```

(b) Calculate a 90% confidence interval

- for the average IQ of mothers of gifted children. ANSWER -> (116.41, 119.982)

```
p_hat = 118.2       # point_estimate from a given sample
sd = 6.5
se  = sd/sqrt(36)     # sample error that comes from Sqrt(pq/n)

cl = 0.90
z = 1.645

solLwr = p_hat - (z * se)
solUpr = p_hat + (z * se)
```

(c) Do your results from

- the hypothesis test
- and the confidence interval agree?

Explain.

ANSWER -> No, The hypothesis test yielded that mother of gifted children has an Iq = 100, and the confidence intervel suggest that is between (116.41, 119.982), and does not include 100

3.1.3 Spray Paint

3.1.3.1 LE3-Q1.5 (0.75 points) Suppose the area that can be painted using a single can of spray paint

- is slightly variable
- and follows a nearly normal distribution
  - with a mean of 25 square feet
  - and a standard deviation of 3 square feet.

(a) What is the probability that

- the area covered by a can of spray paint
- is more than 27 square feet?

ANSWER -> 0.747

```
pe = 25
sd = 3

pnorm(27, mean=25, sd=3)
```

```
## [1] 0.7475075
```

(b) Suppose you want to spray paint

- an area of 540 square feet
- using 20 cans of spray paint.

On average, how many square feet

- must each can be able to cover
- to spray paint all 540 square feet?

ANSWER -> 27

```
540/20
```

```
## [1] 27
```

(c) What is the probability

- that you can cover a 540 square feet area
- using 20 cans of spray paint?

ANSWER ->

```
pe = 25
n_of_cans = 20
area_cov = 25*20
```

(d) If the area covered by a can of spray paint

- had a slightly skewed distribution,
- could you still calculate the probabilities in parts (a) and (c)
  - using the normal distribution?

ANSWER -> No, I guess not

3.1.4 Fuel efficiency of Prius

3.1.4.1 LE3-Q1.6 (0.75 points) Fueleconomy.gov,

- the official US government source
  - for fuel economy information,
- allows users to share gas mileage information on their vehicles.

The histogram below shows

- the distribution of gas mileage in miles per gallon (MPG)
  - from 14 users who drive a 2012 Toyota Prius.
- The sample mean is 53.3 MPG
  - and the standard deviation is 5.2 MPG.

Note that these data are user estimates

- and since the source data cannot be verified,
- the accuracy of these estimates are not guaranteed.[@noauthor_gas_nodate]

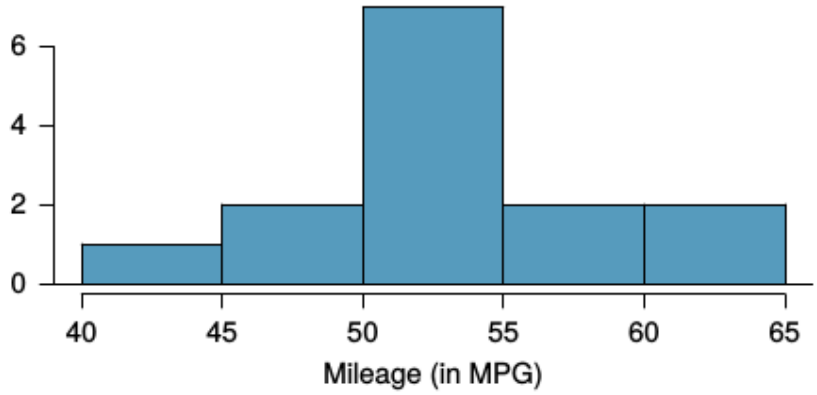


Figure 3: Mileage in MPG

(a) We would like to use these data to evaluate

- the average gas mileage of all 2012 Prius drivers.

Do you think this is reasonable?

- Why or why not?

ANSWER -> The number of observations of each milege group here is less than 10, so if we are interested in any population parameter, we have not ensured enough observations to satisfy central limit theorem.

(b) The EPA claims that a 2012 Prius gets 50 MPG

- (city and highway mileage combined).

Do these data provide strong evidence against this estimate

- for drivers who participate on fueleconomy.gov?
- Note any assumptions you must make as you proceed with the test.

ANSWER -> p-val = 0.9912, which is a very high value, and there is a good chance that this occurence has not been by chance, we favour null hypthesis for 50 for this simulation. Though it is important to reiterate that one cannot follow this tactic because central limit theorem is not satisfied.

```
H0 <- " There is nothing special going on, The population mean = 50MPG "
H1 <- " Maybe the observed value is indicative of a higher mean and maybe
the mean is actually different"

pt_es_m = 53.3
sd = 5.2

se = sd/sqrt(14)

z = (pt_es_m - 5 )/se
pval <- pnorm(z)
```

(c) Calculate a 95% confidence interval

- for the average gas mileage of a 2012 Prius
- by drivers who participate on fueleconomy.gov.

ANSWER ->(50.57, 56.02)

```
cl = 0.95
z = 1.96

pt_es_m = 53.3
sd = 5.2

se = sd /sqrt(14)

solLwr = pt_es_m - (z * se)
solUpr = pt_es_m + (z * se)
```

3.1.5 Diamonds

3.1.5.1 LE3-Q1.7 (OIS Exercise 7.24) (0.5 points) Prices of diamonds are determined by what is known as the 4 Cs:

- cut,
- clarity,
- color,
- and carat weight.

The prices of diamonds go up

- as the carat weight increases,
- but the increase is not smooth.

For example, the difference between the size

- of a 0.99 carat diamond and

  - a 1 carat diamond is undetectable to the naked human eye,
- but the price of a 1 carat diamond tends to be much higher
  - than the price of a 0.99 diamond.

In this question we use two random samples of diamonds,

- 0.99 carats and 1 carat,
- each sample of size 23,

and compare the average prices of the diamonds.

In order to be able to compare equivalent units,

-we first divide the price for each diamond - by 100 times its weight in carats.

That is, for a 0.99 carat diamond, we divide the price by 99.

For a 1 carat diamond, we divide the price by 100.

The distributions and some sample statistics are shown below.[@wickham_ggplot2:_2016]

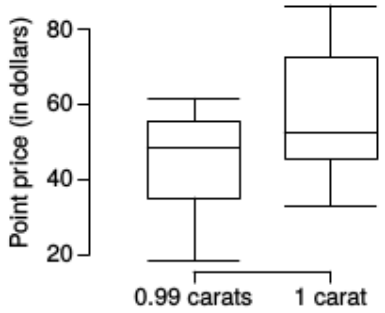


Figure 4: Point Price

	0.99 carats	1 carat
Mean	$ 44.51	$ 56.81
SD	$ 13.32	$ 16.13
n	23	23

Figure 5: Sample Statistics

(a) Conduct a hypothesis test to evaluate

- if there is a difference between the average standardized prices
- of 0.99 and 1 carat diamonds.

Make sure to

- state your hypotheses clearly,
- check relevant conditions,
- and interpret your results in context of the data.

ANSWER -> Since the p_val = 0.991, it means there is a high chance of occurence of Mu_diff = 0, which supports null hypothesis

```
H0 <- " There is no difference between the two diamonds prices: mu1 - mu2 = 0"
H1 <- " There maybe a difference between them: mu1 - mu2 != 0"
```

```
n1 = 23
n2 = n1
sd1 = 13.32
sd2 = 16.13
mu1 = 44.51
mu2 = 56.81
p_est = mu2 - mu1

se <- sqrt( (sd1^2/n1) + (sd2^2/n2) )
z = (p_est - 0)/ se

p_val = pnorm(z)
```

3.1.5.2 LE3-Q1.8 (OIS Exercise 7.26) (0.5 points)

We discussed diamond prices

- (standardized by weight)
- for diamonds with weights 0.99 carats and 1 carat.

See the table for summary statistics,

- and then construct a 95% confidence interval
- for the average difference
  - between the standardized prices of 0.99 and 1 carat diamonds.

You may assume the conditions for inference are met.

ANSWER ->(44.75, 61.849) : 95% intervel

```
cl = 0.95
z = 1.96

n1 = 23
n2 = n1
sd1 = 13.32
sd2 = 16.13
mu1 = 44.51
mu2 = 56.81
p_est = mu2 - mu1

se = sqrt( (sd1^2/n1) + (sd2^2/n2) )

solLwr = pt_es_m - (z * se)
solUpr = pt_es_m + (z * se)
```

3.2 LE3-Q2. Acrylic Hardcoats: (5.5 points)

```
# dt1 = read.csv("./data/acryhc-key.csv")
# dt1.1 = dt1 %>%  distinct(Product.Name)
# dt1.2 = dt1 %>%  distinct(Exposure)

###HF, ASTMG155, ASTMG154, and mASTMG154 have data on Steps 0-4 while the two outdoor
##exposures (1x and 5x) have data on Steps 0-3.
```

```
# dt2 = read.csv("./data/acryhc-exp.csv")

# Samples were exposed to a varying number of hours for
# the different exposure conditions, which can be seen in the file acryhc-exp.csv (in the ./data/ folde
# The numbers given are the hours per step and are not cumulative exposure times as listed.
```

```
# dt3 = read.csv("./data/color/step0/")
# dt4 = read.csv("./data/FTIR/step0/step0.csv")
```

In this assignment, you will work on degradation data

- from outdoor exposure of hard-coat acrylic films(9006,9013,9025)
- on substrate films of
  - polyester(PET) films
  - and urethane(TPU) films.

There is a document of background information

- "2108-351-351m-451-LE3-Q2-AcrylicHardcoatDegradation.pdf"
- Which discusses how the study is performed
- What the exposures are
- What the "key files" of metadata for the study contain
- And what the samples are

There is also a csv file

- "2108-351-351m-451-LE3-Q2-example-dataframe.csv"
- that has a possible column and row structure
- Like you may want to use as you make your dataframe

These data are taken step-wise in time,

- where the samples are exposed
  - to real world conditions
  - or lab-based (accelerated) exposure conditions
- of temperature, humidity and solar irradiance
- and then are evaluated using non-destructive techniques
  - such as optical spectroscopy.

The data can be divided into three parts or types,

- they are color, FTIR, dose.
- Using tidyverse commands
- You will want to wrangle these data one by one.
  - And assemble them into a final data frame for analysis.

Step 1 : build a color files dataframe,

- that is the begining of your analysis dataframe

Step 2 : build an FTIR dataframe

Step 3 : quantify the peak heights at specified x-axis wavenumbers

Step 4 : add the FTIR peak heights into your master datafram.

[1] read the acryhc_exp and acryhc_key files (located in ./data/ folder)

```
exp = read.csv("./data/acryhc-exp.csv")
key = read.csv("./data/acryhc-key.csv")
```

[2] read all color datafiles (in the ./data/color folder)

```
#read color step 0

# step 0 is associated with each product name - 8

# step0_sample = read.csv("./data/color/step0/9013-PET.csv")

files <- list.files(path = "./data/color/step0/" )
# Define dat_total
step0 <- NULL


i <- files[1]
# We can now use a for loop through all of the file names we want to read
for (i in files) {

unit_data <- read.csv(paste0("./data/color/step0/", i) )

# print(colnames(unit_data))

# # rbind data to organize it
step0 <- rbind(step0, unit_data)

step0_clean <- left_join(step0, key,
              by = c("ID" = "Sample.Number"))
step0_clean <- na.omit(step0_clean)
}

step0_clean <- step0_clean %>%
  add_column(step = 0)
```

```
# using acryhc_key files to check the ID in color file and filled the missing information


#read color step 1

files <- list.files(path = "./data/color/step1/" )
# Define dat_total
step1 <- NULL

i <- files[1]
# We can now use a for loop through all of the file names we want to read
for (i in files) {


unit_data <- read.csv(paste0("./data/color/step1/", i) )


# # rbind data to organize it
step1 <- rbind(step1, unit_data)
step1_clean <- left_join(step1, key,
              by = c("ID" = "Sample.Number"))
step1_clean <- na.omit(step1_clean)
}
```

```
step1_clean <- step1_clean %>%
  add_column(step = 1)
```

4 ### try to combine step0 and step1

5 combined <- rbind(step0, step1)

```
# using acryhc_key files to check the ID in color file and filled the missing information

library(tidyverse)

#read color step 2

files <- list.files(path = "./data/color/step2/" )
# Define dat_total
step2 <- NULL

i <- files[1]
# We can now use a for loop through all of the file names we want to read
for (i in files) {


unit_data <- read.csv(paste0("./data/color/step2/", i) )
# # rbind data to organize it
step2 <- rbind(step2, unit_data)
step2_clean <- left_join(step2, key,
              by = c("ID" = "Sample.Number"))
}

step2_clean <- step2_clean %>%
  add_column(step = 2)

# using acryhc_key files to check the ID in color file and filled the missing information


#read color step 3
files <- list.files(path = "./data/color/step3/" )
# Define dat_total
step3 <- NULL

i <- files[1]
# We can now use a for loop through all of the file names we want to read
for (i in files) {

unit_data <- read.csv(paste0("./data/color/step3/", i) )
# # rbind data to organize it
step3 <- rbind(step3, unit_data)
step3_clean <- left_join(step3, key,
              by = c("ID" = "Sample.Number"))

step3_clean <- na.omit(step3_clean)
}
```

```r
step3_clean <- step3_clean %>%
  add_column(step = 3)


# using acryhc_key files to check the ID in color file and filled the missing information


#read color step 4

files <- list.files(path = "./data/color/step4/" )
# Define dat_total
step4 <- NULL

i <- files[1]
# We can now use a for loop through all of the file names we want to read
for (i in files) {

unit_data <- read.csv(paste0("./data/color/step4/", i) )
# # rbind data to organize it
step4 <- rbind(step4, unit_data)
step4_clean <- left_join(step4, key,
              by = c("ID" = "Sample.Number"))
step4_clean <- na.omit(step4_clean)

}

step4_clean <- step4_clean %>%
  add_column(step = 4)



# using acryhc_key files to check the ID in color file and filled the missing information

## only those ids must appear that are in the key table

color_clean <- rbind(step0_clean, step1_clean, step2_clean, step3_clean, step4_clean)

## rename the columns


color_clean <- color_clean %>%
  dplyr::rename(sample = ID  , L = L., a = a., b = b., YI = YI.E313..D65.10.,
         Haze = Haze...D65.10, material = Product.Name)

## rename got confused, and wanted specifically deplyr added




# build big color data frame
```

```
#clean the color data frame, remove NA, the ID with wrong name will have NA in its row
```

[3] read all FTIR files (in the ./data/ftir folder)

```
library(ggplot2)
library(dplyr)
#read step 0 files
library(tidyverse)
library(purrr)

files <- list.files(path = "./data/FTIR/step0/" )
# Define dat_total
f_step0 <- NULL

f_step0 <- files %>%
  map(~read_csv(file.path('./data/FTIR/step0/',.))) %>%
  reduce(rbind)
```

```
## Rows: 1798 Columns: 17
```

```
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## dbl (17): Wavenumber, sa22089.14, sa22088.00, sa22087.14, sa22086.00, sa2208...
```

```
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```
f_step0_t <-  t(f_step0)
colnames(f_step0_t) <- f_step0_t[1,]
f_step0_t <- f_step0_t[-1,]
f_step0_t <- as.data.frame(f_step0_t)
f_step0_t <- tibble::rownames_to_column(f_step0_t, "Sample_ID")
f_step0_t <- f_step0_t %>%
  add_column(step = 0, .after = "Sample_ID")
```

```
#define the peak wavenumber

## Step 3 : quantify the peak heights at specified x-axis wavenumbers

#read FTIR step 1 files

temp1 <- read.csv("./data/FTIR/step1/1x-5x.csv")
temp2 <- read.csv("./data/FTIR/step1/HF-ASTMG155-ASTMG154-mASTMG154.csv")


files <- list.files(path = "./data/FTIR/step1/" )
# Define dat_total
f_step1 <- NULL
# We can now use a for loop through all of the file names we want to read
files_f1 <- files %>%
  map(~read_csv(file.path('./data/FTIR/step1/',.)))
```

```
## Rows: 1798 Columns: 17
```

```
## -- Column specification --------------------------------------------------------
## Delimiter: ","
```

```
## dbl (17): Wavenumber, sa22087.10, sa22089.10, sa22085.05, sa22082.00, sa2207...
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
## Rows: 1798 Columns: 33
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## dbl (33): Wavenumber, sa22088.11, sa22086.11, sa22084.01, sa22078.01, sa2208...
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```
# Add Exposure from files to avoid duplicates
# files_f1[[1]] <- files_f1[[1]] %>%
#   add_column(exposure_from = "1x-5x.csv")
#
# files_f1[[2]] <- files_f1[[2]]%>%
#   add_column(exposure_from = "HF-ASTMG155-ASTMG154-mASTMG154.csv")




# f_step1 <- merge(files_f1[1], files_f1[2],
#                  by= c("Wavenumber","exposure_from"), all =TRUE)
f_step1 <- merge(files_f1[1], files_f1[2],
                 by= c("Wavenumber"), all =TRUE)


# f_step1 %>%
#   select(exposure_from.x, exposure_from.y)


f_step1_t <- t(f_step1)
colnames(f_step1_t) <- f_step1_t[1,]
f_step1_t <- f_step1_t[-1,]
f_step1_t <- as.data.frame(f_step1_t)
f_step1_t <- tibble::rownames_to_column(f_step1_t, "Sample_ID")
f_step1_t <- f_step1_t %>%
  add_column(step = 1 , .after = "Sample_ID")
```

```
#read FTIR step 2 files

temp3 <- read.csv("./data/FTIR/step2/1x-5x.csv")
temp4 <- read.csv("./data/FTIR/step2/HF-ASTMG155-ASTMG154-mASTMG154.csv")

files <- list.files(path = "./data/FTIR/step2/" )
files_2 <- files %>%
  map(~read_csv(file.path('./data/FTIR/step2/',.)))
```

```
## Rows: 1798 Columns: 38
## -- Column specification --------------------------------------------------------
## Delimiter: ","
```

```
## dbl (38): Wavenumber, sa22088.12, sa22086.12, sa22084.02, sa22078.02, sa2208...
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
## Rows: 1798 Columns: 33
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## dbl (33): Wavenumber, sa22088.12, sa22086.12, sa22084.02, sa22078.02, sa2208...
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```
# # Add Exposure from files to avoid duplicates
# files_2[[1]] <- files_2[[1]] %>%
#   add_column(exposure_from = "1x-5x.csv")
#
# files_2[[2]] <- files_2[[2]]%>%
#   add_column(exposure_from = "HF-ASTMG155-ASTMG154-mASTMG154.csv")

f_step2 <- merge(files_2[1], files_2[2], by= c("Wavenumber"), all =TRUE)

f_step2_t = t(f_step2)
colnames(f_step2_t) <- f_step2_t[1,]
f_step2_t <- f_step2_t[-1,]
f_step2_t <- as.data.frame(f_step2_t)
f_step2_t <- tibble::rownames_to_column(f_step2_t, "Sample_ID")

f_step2_t <- f_step2_t %>%
  add_column(step = 2,.after = "Sample_ID")


#### Here there are duplicates of ids.
```

```
#read FTIR step 3 files

ftir_30 <- read.csv("./data/FTIR/step3/1x-5x.csv")
ftir_31 <- read.csv("./data/FTIR/step3/HF-ASTMG155-ASTMG154-mASTMG154.csv")

## the condition for column binding is not satisfied.

ftir_30_t = t(ftir_30)
ftir_31_t = t(ftir_31)

## postprocessing on transpose

colnames(ftir_30_t) <- ftir_30_t[1,]
ftir_30_t <- ftir_30_t[-1,]
ftir_30_t <- as.data.frame(ftir_30_t)
# ftir_30_t <- ftir_30_t %>%
#   add_column(ID = rownames(ftir_30_t))

ftir_30_t <- tibble::rownames_to_column(ftir_30_t, "Sample_ID")
```

```
colnames(ftir_31_t) <- ftir_31_t[1,]
ftir_31_t <- ftir_31_t[-1,]
ftir_31_t <- as.data.frame(ftir_31_t)
ftir_31_t <- tibble::rownames_to_column(ftir_31_t, "Sample_ID")



## They do nt have same column values, and hence have to be joined,
## but preserving all the data


f_step3 <- full_join( ftir_30_t, ftir_31_t,by = "Sample_ID" )
dim(f_step3)
```

```
## [1]  116 2699
```

```
f_step3 <- f_step3 %>%
  add_column(step = 3,.after = "Sample_ID")
```

```
#read FTIR step 4 files


f_step4 <- read.csv("./data/FTIR/step4/step4-acryhc-ftiratr.csv") ## single file
f_step4_t = t(f_step4)
colnames(f_step4_t) <- f_step4_t[1,]
f_step4_t <- f_step4_t[-1,]
f_step4_t <- as.data.frame(f_step4_t)
f_step4_t <- tibble::rownames_to_column(f_step4_t, "Sample_ID")

f_step4_t <- f_step4_t %>%
  add_column(step = 4)


#merge all FTIR data to one data frame.
ftir_clean <- full_join(f_step0_t, f_step1_t)
```

```
## Joining, by = c("Sample_ID", "step", "650.612", "652.476", "654.341", "656.205", "658.069", "659.933
```

```
ftir_clean <- full_join(ftir_clean, f_step2_t)
```

```
## Joining, by = c("Sample_ID", "step", "650.612", "652.476", "654.341", "656.205", "658.069", "659.933
```

```
ftir_clean <- full_join(ftir_clean, f_step3)
```

```
## Joining, by = c("Sample_ID", "step", "650.612", "652.476", "654.341", "656.205", "658.069", "659.933
```

```
ftir_clean <- full_join(ftir_clean, f_step4_t)
```

```
## Joining, by = c("Sample_ID", "step", "3335.09", "3346.27", "3350", "3353.73", "3364.91", "648.387",
```

```
## let us check what is happening in our big dataframe for the four wavenumbers.
```

```
library(dplyr)
library(matrixStats)
```

```
##
## Attaching package: 'matrixStats'
```

```
## The following object is masked from 'package:dplyr':
##
##     count
```

```
wavenumbers <- c(1250,1700,2900,3350)
col_names <- colnames(ftir_clean)
col_names <- as.double(col_names) # Gave NA for sample id
```

```
## Warning: NAs introduced by coercion
```

```
col_names <- na.omit(col_names)

get_range <- function(min, max, col_names) {
  lst_range <- NULL
  for (i in 3:length(col_names)){
    if(col_names[[i]]> min & col_names[[i]] < max){
    lst_range <- c(lst_range, col_names[i])
   }
  }
  lst_range
}

get_mod_range <- function(old_range) {
  lst_range_a <- c("sampleID", "step")
  old_range <- as.character(old_range)
  for (i in 1 : length(old_range)) {
    nc <- paste("a", old_range[i], sep ="_")
    lst_range_a <- c(lst_range_a, nc)
}
  lst_range_a

}

get_range_dt <- function(ftir_clean, old_range, lst_range_a) {
  old_range <- as.character(old_range)
  col_range_dt <- ftir_clean %>%
  select(Sample_ID, step, all_of(old_range))
  colnames(col_range_dt) <- lst_range_a ## put the new columns for finding max
  col_range_dt[is.na(col_range_dt)] <- 0 ## the exercise gives na otherwise
  col_range_dt
}

################################################################################

lst1250 <- get_range(1225, 1275, col_names)
lst1250_a <- get_mod_range(lst1250)
col_1250_dt <- get_range_dt(ftir_clean, lst1250, lst1250_a)
col_1250_dt_n <- col_1250_dt %>%
  rowwise() %>%
  mutate(ftir1250 = max(a_1226.66, a_1228.52, a_1230.38, a_1232.25,
                        a_1234.11, a_1235.98, a_1237.84, a_1239.71,
                        a_1241.57, a_1243.43, a_1245.3, a_1247.16,
                        a_1249.03, a_1250.89, a_1252.75, a_1254.62,
                        a_1256.48, a_1258.35, a_1260.21, a_1262.08,
                        a_1263.94, a_1265.8, a_1267.67, a_1269.53,
```

```
                          a_1271.4,  a_1273.26, a_1225.97, a_1229.7,
                          a_1233.43, a_1237.15, a_1240.88,a_1244.61,
                          a_1248.33, a_1252.06, a_1255.78, a_1259.51,
                          a_1263.24, a_1266.96,a_1270.69, a_1274.42)) %>%
  select(sampleID, ftir1250, step)

##############################################################################
lst1700 <- get_range(1675, 1725, col_names)
lst1700_a <- get_mod_range(lst1700)
col_1700_dt <- get_range_dt(ftir_clean, lst1700, lst1700_a)

col_1700_dt_n <- col_1700_dt %>%
  rowwise() %>%
  mutate(ftir1700 = max(
    a_1675.93 , a_1677.8  ,a_1679.66 , a_1681.52 , a_1683.39 , a_1685.25 ,
    a_1687.12,a_1688.98 , a_1690.85 , a_1692.71 , a_1694.57 , a_1696.44 ,
    a_1698.3 , a_1700.17 , a_1702.03,a_1703.9,a_1705.76 , a_1707.62 ,
    a_1709.49 , a_1711.35 , a_1713.22 , a_1715.08 , a_1716.94, a_1718.81 ,
    a_1720.67 , a_1722.54 , a_1724.4 , a_1676.86 , a_1680.59 , a_1684.32 ,
    a_1688.04, a_1691.77 , a_1695.49 , a_1699.22 , a_1702.95 , a_1706.67 ,
    a_1710.4 , a_1714.13 , a_1717.85,a_1721.58)) %>%
  select(sampleID, ftir1700, step)

##############################################################################

lst2900 <- get_range(2875, 2925, col_names)
lst2900_a <- get_mod_range(lst2900)
col_2900_dt <- get_range_dt(ftir_clean, lst2900, lst2900_a)
col_2900_dt_n <- col_2900_dt %>%
  rowwise() %>%
  mutate(ftir2900 = max(
    a_2876.49 ,  a_2878.35 ,  a_2880.22 ,  a_2882.08 ,  a_2883.95 ,
    a_2885.81 ,  a_2887.67, a_2889.54 ,  a_2891.4 , a_2893.27 ,  a_2895.13 ,
    a_2896.99 ,  a_2898.86 ,  a_2900.72 ,  a_2902.59, a_2904.45 ,  a_2906.32 ,
    a_2908.18 ,  a_2910.04 ,  a_2911.91 ,  a_2913.77 ,  a_2915.64 ,  a_2917.5,
    a_2919.37 ,  a_2921.23 ,  a_2923.09 ,  a_2924.96 ,  a_2876.75 ,  a_2880.48 ,
    a_2884.2 , a_2887.93,a_2891.66 ,  a_2895.38 ,  a_2899.11 ,  a_2902.84 ,
    a_2906.56 ,  a_2910.29 ,  a_2914.02 ,  a_2917.74 , a_2921.47 )) %>%
  select(sampleID, ftir2900, step)

##############################################################################
lst3350 <- get_range(3325, 3375, col_names)
lst3350_a <- get_mod_range(lst3350)
col_3350_dt <- get_range_dt(ftir_clean, lst3350, lst3350_a)

col_3350_dt_n <- col_3350_dt %>%
  rowwise() %>%
  mutate(ftir3350 = max(a_3325.77,  a_3327.63,  a_3329.49,  a_3331.36,
                        a_3333.22,  a_3335.09,  a_3336.95, a_3338.81,
                        a_3340.68,  a_3342.54,  a_3344.41,  a_3346.27,
                        a_3348.14,  a_3350 , a_3351.86, a_3353.73,  a_3355.59,
                        a_3357.46,  a_3359.32,  a_3361.19,  a_3363.05,
                        a_3364.91,  a_3366.78, a_3368.64,  a_3370.51,
```

```
                        a_3372.37,  a_3374.23,  a_3327.64,  a_3331.37,
                        a_3338.82,  a_3342.55, a_3357.45,  a_3361.18,
                        a_3368.63,  a_3372.36
    )) %>%
  select(sampleID, ftir3350, step)


################################################################

  ftir_wav_des  <- full_join(col_1250_dt_n, col_1700_dt_n)
```

```
## Joining, by = c("sampleID", "step")
```

```
  ftir_wav_des <- full_join(ftir_wav_des, col_2900_dt_n)
```

```
## Joining, by = c("sampleID", "step")
```

```
  ftir_wav_des <- full_join(ftir_wav_des, col_3350_dt_n)
```

```
## Joining, by = c("sampleID", "step")
```

[4] assemble the color data with the FTIR peak value data into a dataframe for analysis

```
# combine color and FTIR when the ID and step are the same, add the FTIR peak value to the color files
combo <- full_join(color_clean, ftir_wav_des, by = c("sample" = "sampleID", "step" = "step"))
# step0_clean <- left_join(step0, key,
#                by = c("ID" = "Sample.Number"))
```

[5] calculate the Exposure Photodose of light (Photodose = Irradiance * Time)

```
# calculate dose

# make a column for dose : with no values




combo_with_dose <- combo %>%              # irradiance addition
  mutate(irradiance = case_when( # from look up table
      Exposure == "mASTMG154"  ~ 0.21835,
      Exposure == "ASTMG154"  ~ 0.21835,
      Exposure == "ASTMG155"  ~ 0.09382,
      Exposure == "1x"  ~ 0.04599,
      Exposure == "5x"  ~ 2.31772,
      Exposure == "HF" ~ 0,
      Exposure == "baseline"  ~ 0
        ))

uniExp <-combo_with_dose %>%
  distinct(Exposure)

for (i in colnames(exp)){
  name = paste("cum",i,sep="_")
  exp[,name] <- cumsum(exp[, i])
}

exp <- exp %>%
```

```
  dplyr::rename(step = Steps)


 combo_with_dose <- full_join(combo_with_dose, exp)
```

```
## Joining, by = "step"
```

```
 combo_with_dose <- combo_with_dose %>%
   mutate( dose =  case_when(
       Exposure == "mASTMG154"  ~ irradiance * cum_mASTG154,
      Exposure == "ASTMG154"  ~ irradiance * cum_ASTMG154,
      Exposure == "ASTMG155"  ~ irradiance * cum_ASTMG155,
      Exposure == "1x"  ~ irradiance *  cum_X1x,
      Exposure == "5x"  ~ irradiance * cum_X5x,
      Exposure == "HF" ~ irradiance * cum_HF,
      Exposure == "baseline"  ~ irradiance * cum_baseline
     ))
```

[6] Finally tidy you dataframe (rename and adjust the order of the columns)

```
final <- combo_with_dose %>%
  select( -15:-30)

## I did much of the column renaming in an earlier cell
final <- final %>%
    select(sample, step, dose, L, a, b, YI, Haze, material, Exposure,
           Step.Number.Retained, ftir1250, ftir1700,ftir2900, ftir3350)
```

```
exposure_type_lookup_MJ_m2 <- read.table(text =
 "Exposure standard weight Full UV
 mASTMG154 QUV.340  8/12  0.30435 0.21835
 ASTMG154 QUV.340  1  0.30435 0.21835
 baseline Baseline  1  0  0
 HF    HF     1 0  0
 ASTMG155 QSUN.TUV  1  1.40655 0.09382
 1x   AM1.5   1 3.6 0.04599
 5x   X50x    1 181.44 2.31772", header = TRUE) %>% data.frame() %>%
  transform(weight = sapply(weight, function(x) eval(parse(text = x))))
```

5.0.1 LE3-Q2.1: What are the dimensions of your data frame? (1.5 points)

```
dim(final)
```

```
## [1] 940  15
```

ANSWER ->

5.0.2 LE3-Q2.2: Show the head and tail of your data frame (1.5 points)

```
head(final)
```

```
##       sample step dose     L     a    b   YI Haze material  Exposure
## 1 sa22068.00    0    0 95.79 -0.18 0.96 1.68  1.4 9006-PET  baseline
## 2 sa22068.01    0    0 95.81 -0.17 0.95 1.67  2.1 9006-PET mASTMG154
## 3 sa22068.02    0    0 95.78 -0.17 0.96 1.68  2.2 9006-PET mASTMG154
## 4 sa22068.03    0    0 95.82 -0.19 0.95 1.65  2.1 9006-PET mASTMG154
```

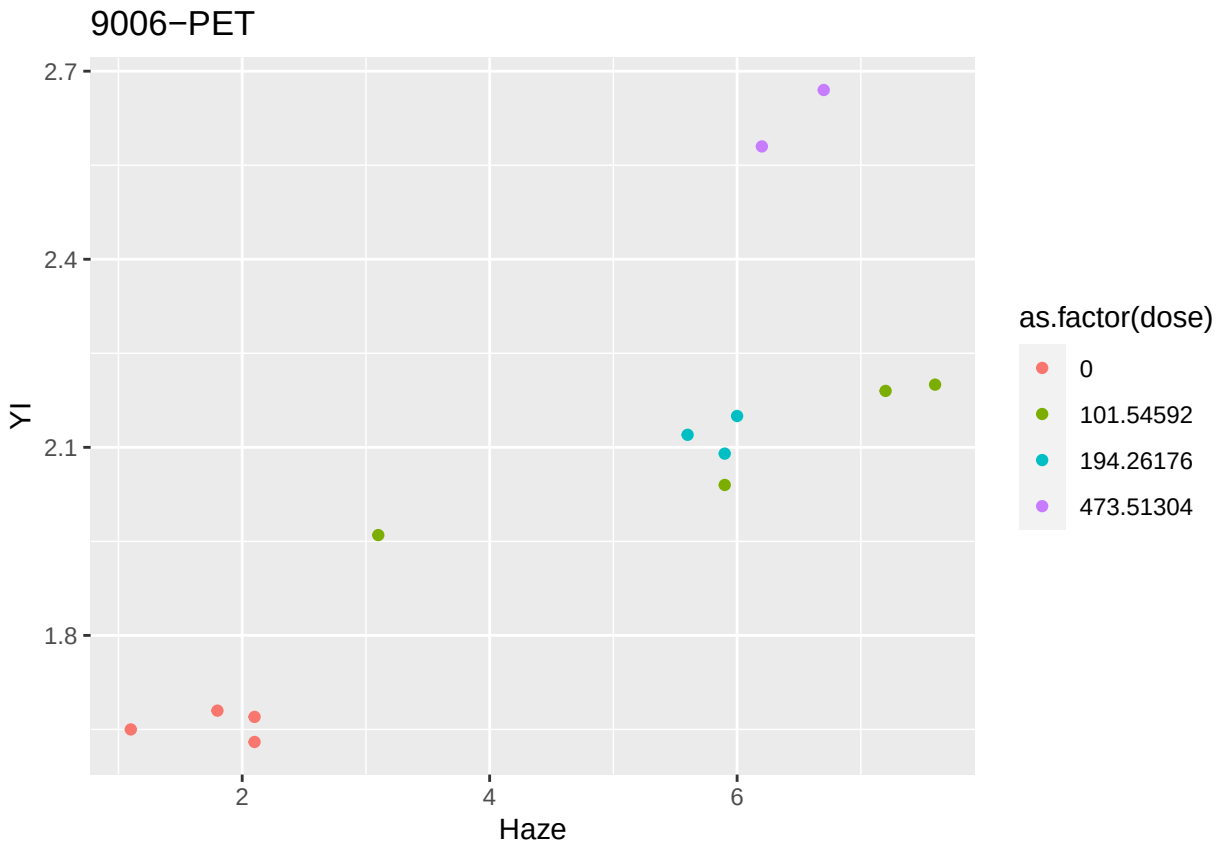
```
## 5 sa22068.04    0    0 95.82 -0.17 0.94 1.65  1.7 9006-PET mASTMG154
## 6 sa22068.06    0    0 95.82 -0.17 0.95 1.66  1.7 9006-PET  ASTMG154
##   Step.Number.Retained  ftir1250 ftir1700  ftir2900  ftir3350
## 1                    0 0.0962835  0.14324 0.0269307 0.0106506
## 2                    1        NA       NA        NA        NA
## 3                    2        NA       NA        NA        NA
## 4                    3        NA       NA        NA        NA
## 5                    4        NA       NA        NA        NA
## 6                    1        NA       NA        NA        NA
```

```
tail(final)
```

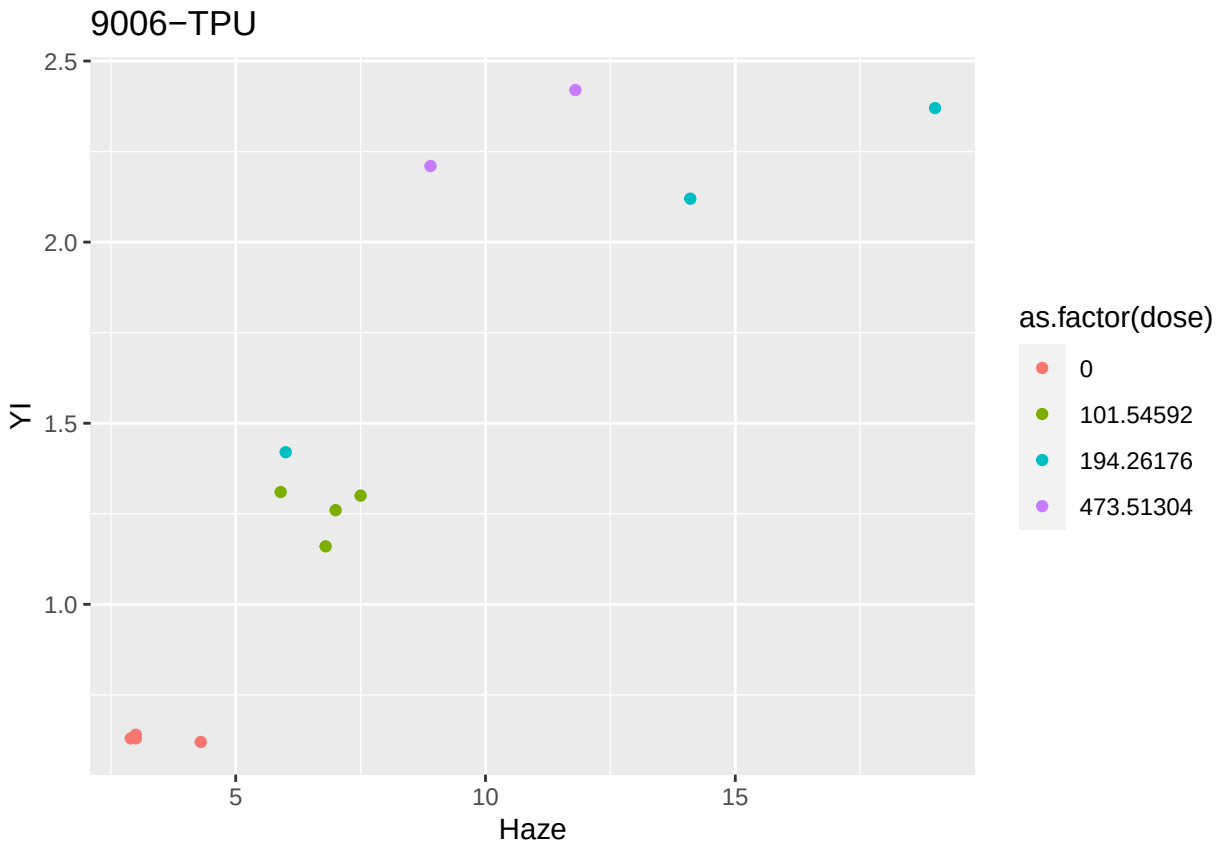
```
##           sample step dose  L  a  b YI Haze material Exposure
## 935 sa22059.54.1    4   NA NA NA NA NA   NA     <NA>     <NA>
## 936   sa22056.55    4   NA NA NA NA NA   NA     <NA>     <NA>
## 937   sa22056.04    4   NA NA NA NA NA   NA     <NA>     <NA>
## 938   sa22074.05    4   NA NA NA NA NA   NA     <NA>     <NA>
## 939   sa22068.05    4   NA NA NA NA NA   NA     <NA>     <NA>
## 940   sa22070.00    4   NA NA NA NA NA   NA     <NA>     <NA>
##     Step.Number.Retained ftir1250 ftir1700  ftir2900  ftir3350
## 935                   NA 0.383492 0.528912 0.1259940 0.0602495
## 936                   NA 0.146153 0.168702 0.0246690 0.0261215
## 937                   NA 0.188405 0.220954 0.0293698 0.0322278
## 938                   NA 0.122258 0.124313 0.0278608 0.0266694
## 939                   NA 0.116885 0.108653 0.0245639 0.0214671
## 940                   NA 0.162649 0.156280 0.0351721 0.0345523
```

5.0.3 LE3-Q2.3: Plot the YI and Haze as a function of Dose for each material for the 1x exposure. Do you notice any difference between substrates and the coatings on the substrates? (2.5 points)

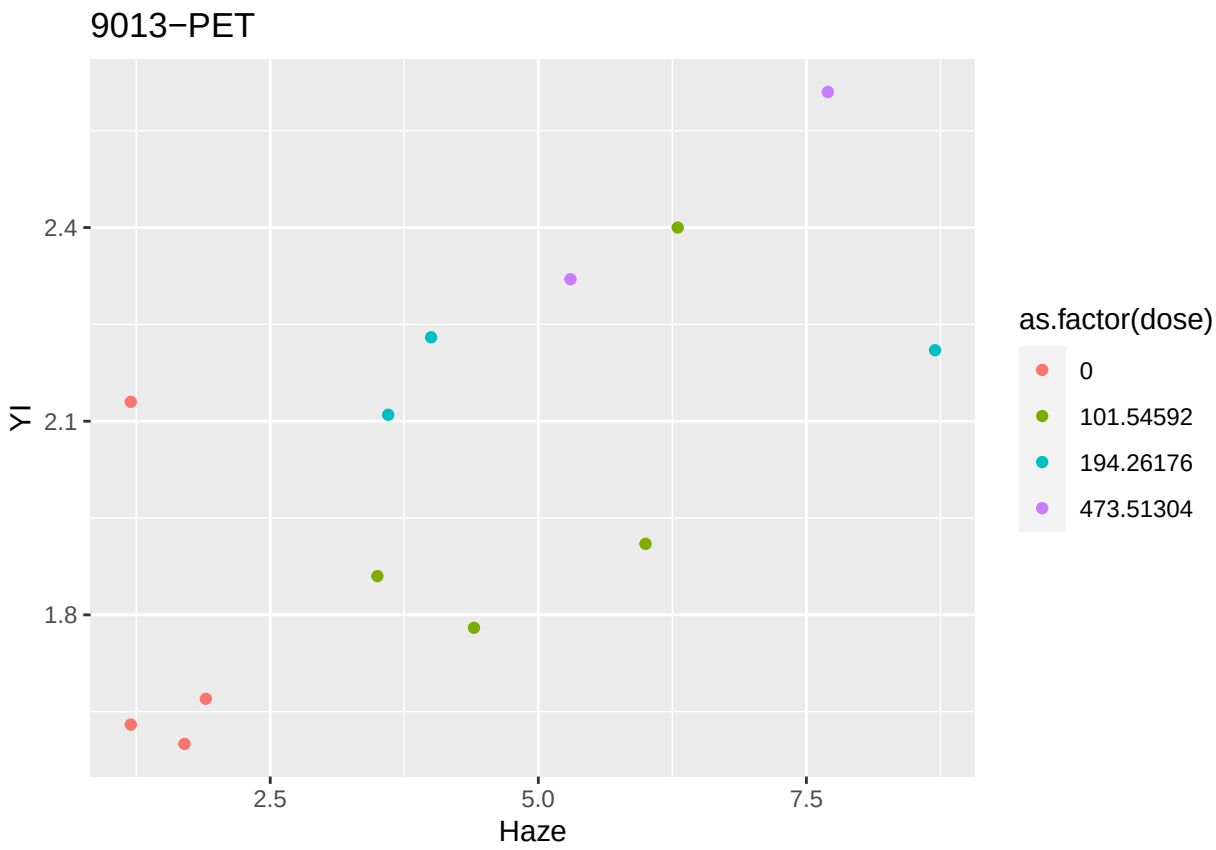
```
ggplot(data = final %>%
      filter(Exposure == "1x", material =="9006-PET"),  aes(x = Haze, y = YI)) +
      geom_point( aes(color = as.factor(dose))) + ggtitle("9006-PET")
```



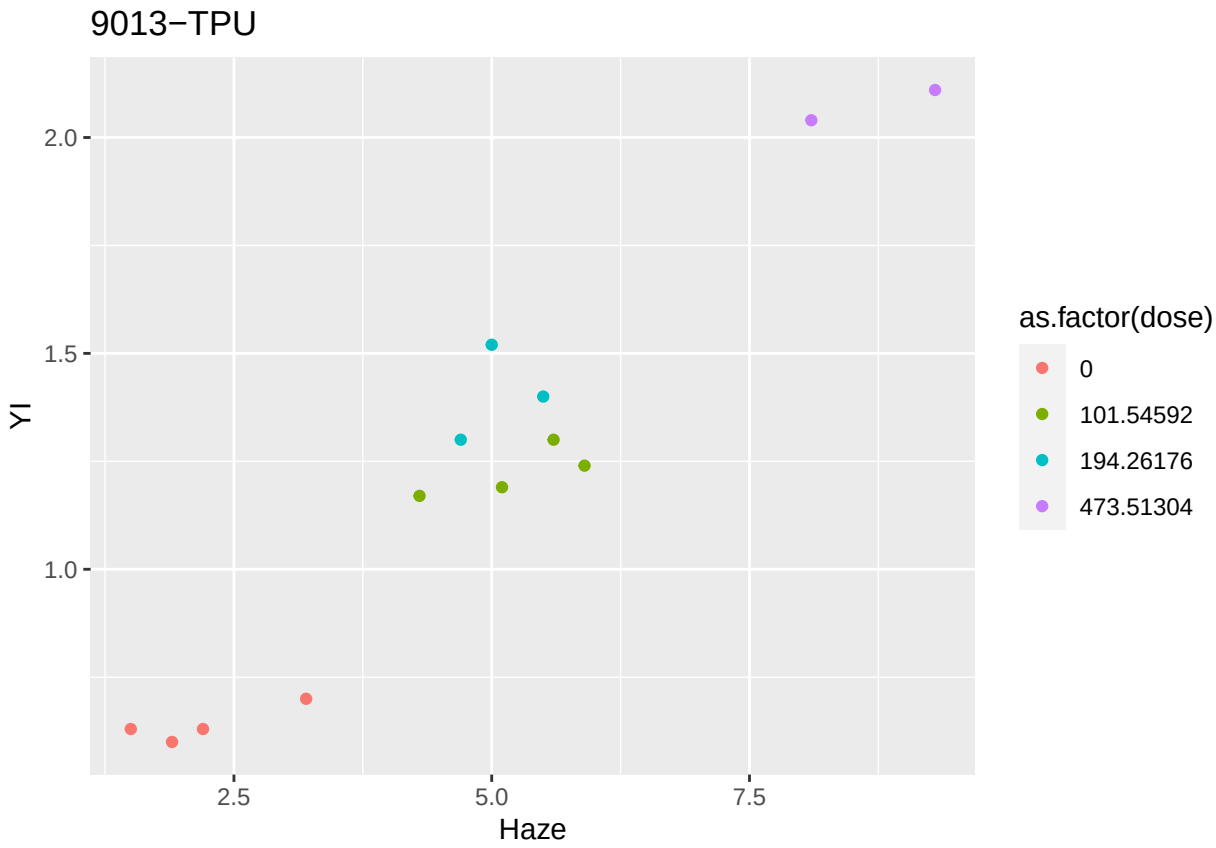

```
ggplot(data = final %>%
      filter(Exposure == "1x", material =="9006-TPU"),  aes(x = Haze, y = YI)) +
      geom_point( aes(color = as.factor(dose))) + ggtitle("9006-TPU")
```



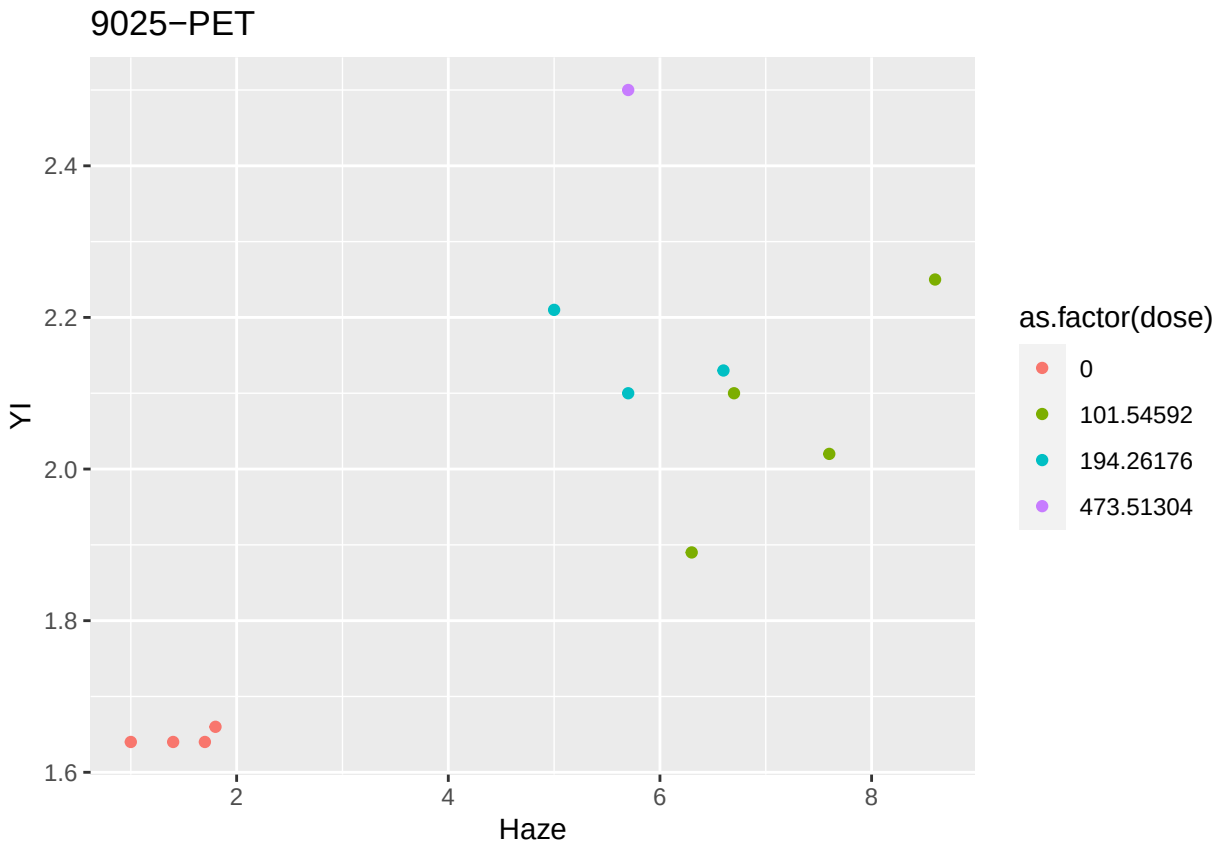

```
ggplot(data = final %>%
         filter(Exposure == "1x", material =="9013-PET"),  aes(x = Haze, y = YI)) +
       geom_point( aes(color = as.factor(dose)))+ ggtitle("9013-PET")
```



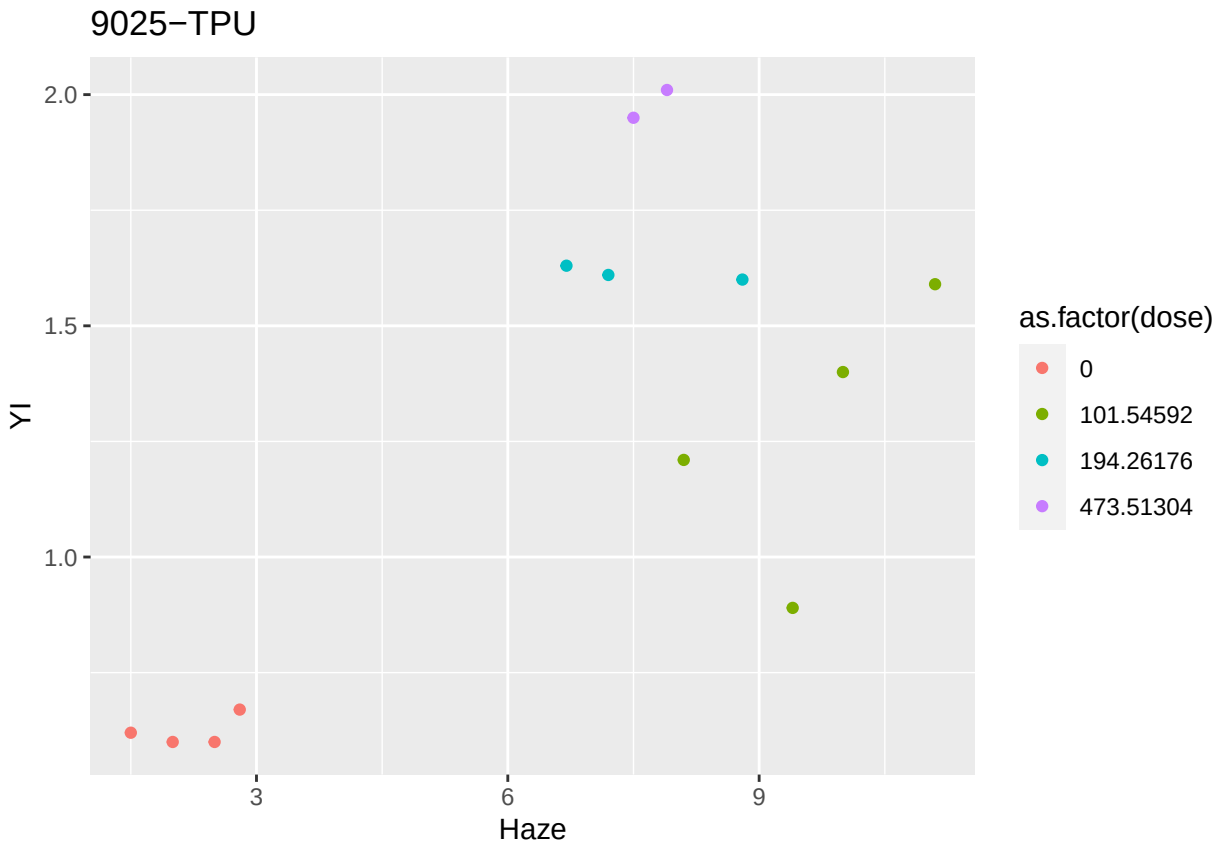

```
ggplot(data = final %>%
         filter(Exposure == "1x", material =="9013-TPU"),  aes(x = Haze, y = YI)) +
       geom_point( aes(color = as.factor(dose))) + ggtitle("9013-TPU")
```



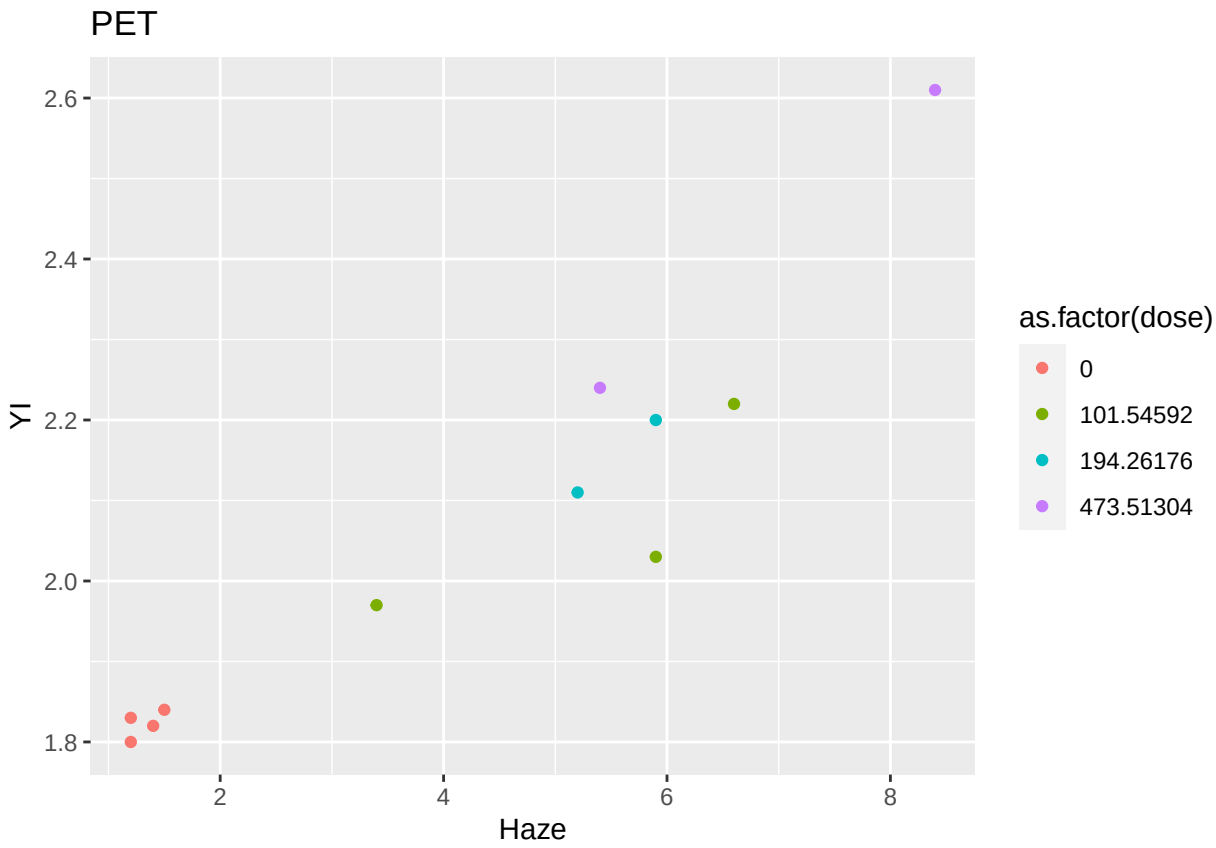

```
ggplot(data = final %>%
         filter(Exposure == "1x", material =="9025-PET"),  aes(x = Haze, y = YI)) +
       geom_point( aes(color = as.factor(dose))) + ggtitle("9025-PET")
```



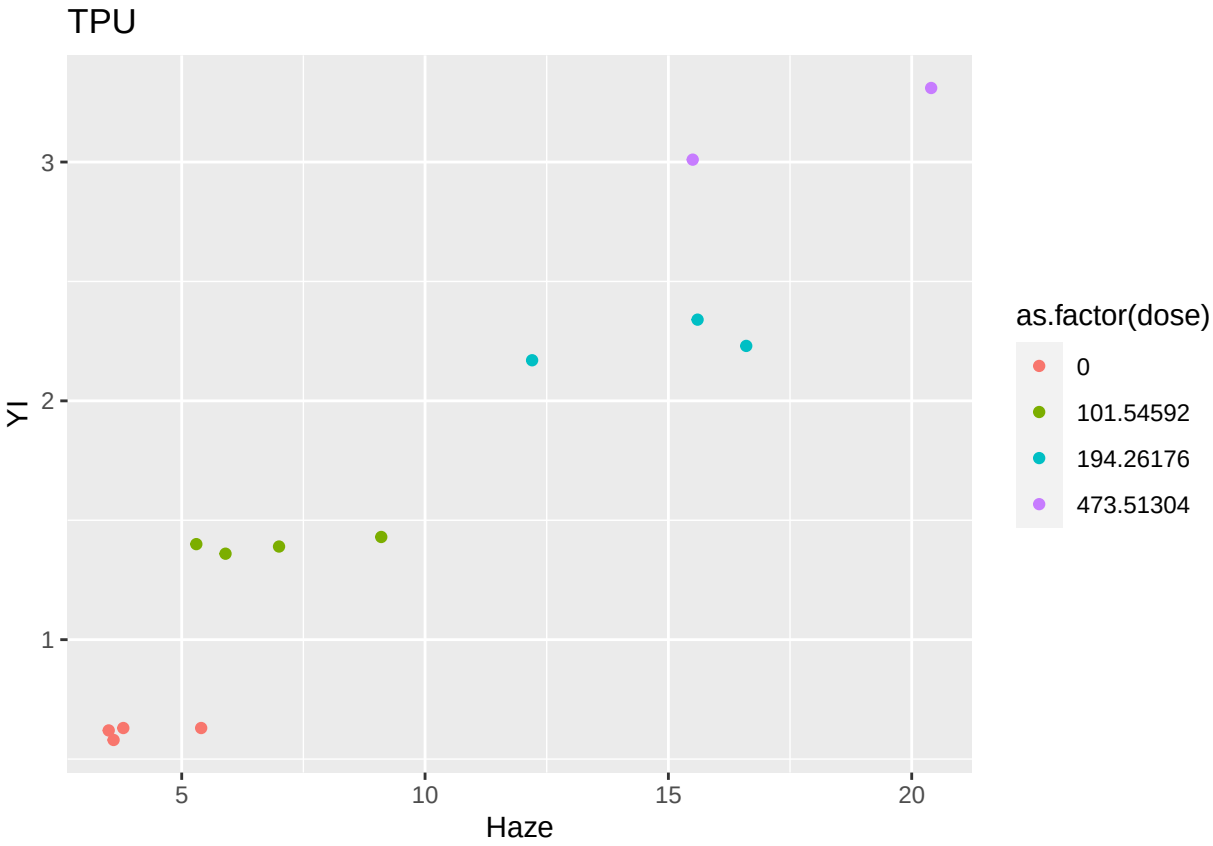

```
ggplot(data = final %>%
         filter(Exposure == "1x", material =="9025-TPU"),  aes(x = Haze, y = YI)) +
       geom_point( aes(color = as.factor(dose))) + ggtitle("9025-TPU")
```




```
ggplot(data = final %>%
         filter(Exposure == "1x", material =="PET"),  aes(x = Haze, y = YI)) +
       geom_point( aes(color = as.factor(dose)))+ ggtitle("PET")
```




```
ggplot(data = final %>%
         filter(Exposure == "1x", material =="TPU"),  aes(x = Haze, y = YI)) +
       geom_point( aes(color = as.factor(dose)))+ ggtitle("TPU")
```



ANSWER ->In general both Haze and YI increases with dose, 9025-TPU, 9025-PET, are the ones not following this trend strictly

5.0.3.1 Links

http://www.r-project.org

http://rmarkdown.rstudio.com/