**Mini Project 4 (Solution)**

**Mini Project Duo Group # 12**

**Contribution of each group member**

Chetan Siddappareddy – 50%

Ankit Sahu – 50%

Both of us have contributed equally to the project. We learnt R through collaboration and then write the R scripts for the corresponding and report all the findings.

Section 1 for explanation (and R code snippets part wise) and Section 2 for R code (from local R Studio).

**Section 1**

**Problem 1**

Reading the data given(gpa.csv)

data = read.csv("gpa.csv")

Storing the gpa and act in separate variable and then plotting the scatter plot.

# getting the gpa scores of the student in variable gpa

gpa = as.numeric(data$gpa)

# getting the act scores of the student in variable act

act = as.numeric(data$act)

plot(gpa, act, main = "Scatter Plot GPA vs ACT", xlab = "GPA", ylab="ACT");

Chart, scatter chart

Description automatically generated

Using Regression model

#Correlation

#abline() can be used to add vertical, horizontal or regression lines to a graph.

#lm() function -> is used to fit linear models -> regression

abline(lm(act~gpa))

From the plot generated, it can be infer that the value of GPA increases the value of the act score also. Now finding the correlation.

> cor(gpa, act)

[1] 0.2694818

Using some other sample for correlation:

> library(boot)

>

> cov.npar = function(ank, iters){

+ gpa = ank$gpa[iters]

+ act = ank$act[iters]

+ result = cor(gpa, act)

+ return (result)

+ }

>

> cov.npar.boot = boot(data, cov.npar, R=999, sim="ordinary", stype="i")

>

> print(cov.npar.boot)

Calculating the correlation estimate :

ORDINARY NONPARAMETRIC BOOTSTRAP

Call:

boot(data = data, statistic = cov.npar, R = 999, sim = "ordinary",

stype = "i")

Bootstrap Statistics :

original bias std. error

t1\* 0.2694818 0.003158357 0.1080157

From above, we can conclude that correlation estimate is 0.2694818

Chart, histogram

Description automatically generated

Using boot.ci for 95% CI (confidence interval):

#Confidence Interval(with boot.ci)

> print(boot.ci(cov.npar.boot))

BOOTSTRAP CONFIDENCE INTERVAL CALCULATIONS

Based on 999 bootstrap replicates

CALL :

boot.ci(boot.out = cov.npar.boot)

Intervals :

Level Normal Basic

95% ( 0.0546, 0.4780 ) ( 0.0597, 0.4693 )

Level Percentile BCa

95% ( 0.0696, 0.4793 ) ( 0.0630, 0.4683 )

Calculations and Intervals on Original Scale

Now, calculating 95% CI using percentile bootstrap

> print(sort(cov.npar.boot$t)[c(25,975)])

[1] 0.06963114 0.47930259

So, 95% CI using percentile bootstrap is (0.06963, 0.4793)

**Problem 2**

1. Exploration using the box plot (5 value analysis).

Filtering for remote and local also is done.

voltages = read.csv("VOLTAGE.csv")

print(voltages)

#for remote location 0 and local location 1

> remote = voltages$voltage[voltages$location==0]

> local = voltages$voltage[voltages$location==1]

> print(remote)

[1] 9.98 10.26 10.05 10.29 10.03 8.05 10.55 10.26 9.97 9.87 10.12 10.05 9.80 10.15 10.00

[16] 9.87 9.55 9.95 9.70 8.72 9.84 10.15 10.02 9.80 9.73 10.01 9.98 8.72 8.80 9.84

> print(local)

[1] 9.19 9.63 10.10 9.70 10.09 9.60 10.05 10.12 9.49 9.37 10.01 8.82 9.43 10.03 9.85

[16] 9.27 8.83 9.39 9.48 9.64 8.82 8.65 8.51 9.14 9.75 8.78 9.35 9.54 9.36 8.68

The box plot below shows that the voltage is higher for local (right) than the remote(left). Also, the remote location voltage is left skewed.

The QQ plots for the remote and the local locations also confirms the dissimilarity.

qqnorm(remote, main="QQ Norm for remote voltage location")

> qqline(remote)

> qqnorm(local, main="QQ Norm for local voltage location")

> qqline(local)

Chart, box and whisker chart

Description automatically generated

Chart

Description automatically generated

Chart, scatter chart

Description automatically generated

We need to find variance to show dissimilarity between them.

> local\_variance = var(local)

> remote\_variance = var(remote)

> print(local\_variance)

[1] 0.229322

> print(remote\_variance)

[1] 0.2925895

As we can see from the R output, both population variances are different. So, we find the confidence interval for the population means of the voltages at the two locations. I am assuming that the populations are normally distributed and hence we can use the Welch’s two sample t-test for finding the CI.

> t.test(remote,local, alternative = "two.sided", conf.level = 0.95,var.equal = FALSE)

Welch Two Sample t-test

data: remote and local

t = 2.8911, df = 57.16, p-value = 0.005419

alternative hypothesis: true difference in means is not equal to 0

95 percent confidence interval:

0.1172284 0.6454382

sample estimates:

mean of x mean of y

9.803667 9.422333

Since, the CI does not contain zero value then we can conclude that the difference in population means of local and remote voltages cannot be zero. Therefore, we cannot establish the manufacturing process locally. We can perform manual calculations of the CI for the difference of population means to verify the assumptions.

> mean\_remote = mean(remote)

> mean\_local = mean(local)

> print(mean\_local)

[1] 9.422333

> print(mean\_remote)

[1] 9.803667

> ci = (mean\_remote - mean\_local) +c(-1,1)\*qt(0.025,58)\*sqrt((var(local) + var(remote))/30)

> print(ci)

[1] 0.6453556 0.1173110

1. We showed in the (a) that the two distributions of the voltages, remote and local are dissimilar, which led to the conclusion that the population means would be different. From the (b) we concluded the same. So we can say that the manufacturing cannot be established locally.

**Problem 3**

We will load the “VAPOR.csv” file and then analyze the data provided on the theoretical and experimental values. As per the problem, I am interested in calculating the difference between the experimental values and the theoretical values at the given set of temperatures. Thus, we calculate the difference from the given data as a paired sample.

> vapor = read.csv("VAPOR.csv")

> theoretical = vapor$theoretical

> experimental = vapor$experimental

>

> diff = theoretical-experimental

>

> print(diff)

[1] 0.006 0.007 -0.015 0.014 -0.022 0.008 0.000 0.002 -0.026 0.029 0.008 0.000 -0.010 0.010

[15] -0.010 0.010

Using histogram for looking at the difference.

> hist(diff)

By looking at the histogram, I can infer that the data is normally distributed. Now I will calculate the confidence interval of the mean of the difference in the data observation in both of the cases and if the confidence interval that we calculated includes a zero, we can say that the theoretical model for vapor pressure is a good model of reality.

> ci = mean(diff) + c(1,-1)\*qt(0.975, 15)\* (sd(diff)/sqrt(16))

> print(ci)

[1] 0.008262694 -0.006887694

Chart, histogram

Description automatically generated

**Section 2**

**####################################**

**R CODE FOR PROBLEM 1:**

**####################################**

**# Solution for Problem 1**

**setwd("/Users/sahuankit010/Desktop/Repo/CS-6313-Stats/Mini Projects/MP4")**

**getwd()**

**data = read.csv("gpa.csv")**

**print(data)**

**# getting the gpa scores of the student in variable gpa**

**gpa = as.numeric(data$gpa)**

**# getting the act scores of the student in variable act**

**act = as.numeric(data$act)**

**plot(gpa, act, main = "Scatter Plot GPA vs ACT", xlab = "GPA", ylab="ACT");**

**#Correlation**

**#abline() can be used to add vertical, horizontal or regression lines to a graph.**

**#lm() function -> is used to fit linear models -> regression**

**abline(lm(act~gpa))**

**cor(gpa, act)**

**#Estimates::**

**library(boot)**

**cov.npar = function(ank, iters){**

**gpa = ank$gpa[iters]**

**act = ank$act[iters]**

**result = cor(gpa, act)**

**return (result)**

**}**

**cov.npar.boot = boot(data, cov.npar, R=999, sim="ordinary", stype="i")**

**print(cov.npar.boot)**

**plot(cov.npar.boot)**

**# Now doing the Point Estimation of bootstrap**

**#mean**

**print(mean(cov.npar.boot$t))**

**#Confidence Interval(with boot.ci)**

**print(boot.ci(cov.npar.boot))**

**#Percentile CI**

**print(sort(cov.npar.boot$t)[c(25,975)])**

**Graphical user interface, application

Description automatically generated with medium confidenceGraphical user interface, text, application

Description automatically generatedGraphical user interface, text, application, email

Description automatically generatedGraphical user interface, text

Description automatically generated**

**R CODE FOR PROBLEM 2:**

**####################################**

**# Solution for Problem 2**

**a)**

**setwd("/Users/sahuankit010/Desktop/Repo/CS-6313-Stats/Mini Projects/MP4")**

**getwd()**

**voltages = read.csv("VOLTAGE.csv")**

**print(voltages)**

**#for remote location 0 and local location 1**

**remote = voltages$voltage[voltages$location==0]**

**local = voltages$voltage[voltages$location==1]**

**print(remote)**

**print(local)**

**boxplot(remote, local)**

**qqnorm(remote, main="QQ Norm for remote voltage location")**

**qqline(remote)**

**qqnorm(local, main="QQ Norm for local voltage location")**

**qqline(local)**

**local\_variance = var(local)**

**remote\_variance = var(remote)**

**print(local\_variance)**

**print(remote\_variance)**

**t.test(remote,local, alternative = "two.sided", conf.level = 0.95,var.equal = FALSE)**

**mean\_remote = mean(remote)**

**mean\_local = mean(local)**

**print(mean\_local)**

**print(mean\_remote)**

**ci = (mean\_remote - mean\_local) +c(-1,1)\*qt(0.025,58)\*sqrt((var(local) + var(remote))/30)**

**print(ci)**

**Graphical user interface, application

Description automatically generatedText

Description automatically generated**

**Graphical user interface, text, application, email

Description automatically generated**

**R CODE FOR PROBLEM 3:**

**####################################**

**# Solution for Problem 3**

**vapor = read.csv("VAPOR.csv")**

**print(vapor)**

**theoretical = vapor$theoretical**

**experimental = vapor$experimental**

**diff = theoretical-experimental**

**print(diff)**

**hist(diff)**

**ci = mean(diff) + c(1,-1)\*qt(0.975, 15)\* (sd(diff)/sqrt(16))**

**print(ci)**

**Graphical user interface, text, application

Description automatically generated**