---

title: "Lab 7 - Solutions"

output:

html\_document:

df\_print: paged

---

```{r setup, include=FALSE}

knitr::opts\_chunk$set(echo = TRUE)

```

```{r}

library(tidyverse)

```

## Loading data

```{r}

drill.data <- read.csv("DrillData.csv", header = T, stringsAsFactors=T)

drill.data

```

## Covariance and correlation

To find out the covariance between depth and drilling time

```{r}

#Input is 2 vectors

cov(drill.data$Depth,drill.data$DrillingTime)

```

Alternative

```{r}

drill.data |> select(Depth, DrillingTime) |> cov()

```

The number returned is positive so there is a positive correlation.

To find the correlation coefficient

```{r}

cor(drill.data$Depth,drill.data$DrillingTime)

```

#Alternative using index numbers

```{r}

cor(drill.data[2],drill.data[3])

```

```{r}

drill.data |> select(Depth, DrillingTime) |> cor()

```

The correlation coefficient is close to 1 so the correlation is strong.

To use the Spearman method.

```{r}

cor(drill.data$Depth,drill.data$DrillingTime, method="spearman")

drill.data |> select(Depth, DrillingTime) |> cor(method="spearman")

```

Note that the result is different from the default (Persons) correlation coefficient value.

To obtain multiple correlation coefficients using the Pearsons correlation coefficient.

```{r}

# For a single dataframe, all columns in range are compared

cor(drill.data[2:3], method="pearson")

```

To obtain multiple correlation coefficients using the Spearman correlation coefficient.

```{r}

cor(drill.data[2:3], method="spearman")

```

## scatterplot using the data

```{r}

p <- ggplot(drill.data,aes(x=Depth,y=DrillingTime))

p <- p + geom\_point()

p <- p + labs(title = "Time vs Drilling Time", x="Depth (m)", y = "Drilling time (days)")

p

```

The variables have a strong linear correlation and the graph looks reasonably linear, so we shall proceed to a linear regression.

## Linear regression

```{r}

lm.output <- lm(formula = DrillingTime~Depth,data=drill.data)

summary(lm.output)

```

So the equation is

DrillingTime = -15.398101 + 0.024246 \* Depth

To look at the residuals

```{r}

lm.output$residuals

```

## Predictions and residuals

```{r}

newdata <- data.frame(Depth=1700)

predict(lm.output,newdata)

```

Plotting residuals against response variable.

First, we create the dataframe.

```{r}

lmData <-

data.frame(residuals = lm.output$residuals,

Depth=drill.data$Depth,

DrillingTime= drill.data$DrillingTime)

```

Now the data can be plotted.

Note: in the plot below, theme\_classic() produces a white background and text = element\_text(size = 20) increases the size of the fonts.

```{r}

p <- ggplot(lmData, aes(x=DrillingTime, y = residuals))

p <- p + geom\_point(size=2, colour="red")

p <- p + theme\_classic()

p <- p + theme(text = element\_text(size = 20))

p

```

The residuals are the differences between the data values and the fitted line. Note that three points have large negative residuals. The remaining residuals are positive or close to zero.

To obtain the residuals against the Depth

```{r}

p <- ggplot(lmData, aes(x=Depth, y = residuals))

p <- p + geom\_point(size=2, colour="red")

p <- p + theme\_classic()

p <- p + theme(text = element\_text(size = 20))

p

```

Again there are 3 residuals which are quite large and negative.

And for the Q-Q plot

```{r}

p <- ggplot(lmData, aes(sample = residuals))

p <- p + stat\_qq(size=2) +

stat\_qq\_line( alpha = 0.9, color = "red",

linetype = "dashed")

p <- p + theme\_classic()

p <- p + theme(text = element\_text(size = 20))

p

```

In this q-q plot, if all points were lying close to the straight line of slope 1 the residuals could be considered normally distributed data. We can see that the residuals are not normally distributed in this case. Some points are far from the line.

The Shapiro-Wilk test also confirms this.

```{r}

shapiro.test(lmData$residuals)

```

The p-value is smaller than 0.05, indicating that it is not reasonable to assume that the residuals are normally distributed.

## Scatterplot with LR line

Note that the below code has se set to true (se=T) so the grey confidence band appears.

```{r}

p <- ggplot(drill.data,aes(x=Depth,y=DrillingTime))

p <- p + geom\_point(col="blue")

p<- p + stat\_smooth(method="lm",se=T,col="red")

p <- p + labs(title ="Linear Fit to Drilling Time vs. Depth", x="Depth (m)" , y ="Drilling Time (days)")

p

```

You will see that a warning message appears stating the formula used (regress y on x).

We can explicitly state that we are regressing y on y by saying formula = y ~ x. Also, we can remove the grey confidence band (setting se=F).

```{r}

p <- ggplot(drill.data,aes(x=Depth,y=DrillingTime))

p <- p + geom\_point(col="blue")

p<- p + stat\_smooth(method="lm",se=F,col="red", formula = y ~ x)

p <- p + labs(title ="Linear Fit to Drilling Time vs. Depth", x="Depth (m)" , y ="Drilling Time (days)")

p

```

# Exercises

## Exercise 1

Building the linear model of Depth on DrillingTime.

```{r}

lm.output2 <- lm(formula = Depth ~ DrillingTime,data=drill.data)

summary(lm.output2)

```

So the equation is

DrillingTime = 782.164 + 36.153 \* Depth

For Depth = 1700 we previously obtained DrillingTime = 25.82071.

Predicting the depth for DrillingTime = 25.82071

```{r}

newdata2 <- data.frame(DrillingTime = 25.82071)

predict(lm.output2,newdata2)

```

So for DrillingTime = 25.82071 the estimated depth is 1715.673

## Exercise 2

Loading the data.

```{r}

loginData <- read.csv("loginTimes.csv", header=T, stringsAsFactors = T)

```

Checking correlations

```{r}

cor(loginData[1:6])

```

Looking for values close to 1 or -1 which are not in the diagonal. Users and timeToLogin seem to have a good correlation ( 0.794681253) . TimeBetweenMessages and timeToLogin are negatively correlated (-0.41082579), but not strongly.

Checking the relationship between users and timeToLogin.

```{r}

p <- ggplot(loginData,aes(x=users,y=timeToLogin))

p <- p + geom\_point()

p <- p + labs(title = "Users vs. time to login", x="number of users", y = "login time (s)")

p

```

The points are roughly arranged around a line, but the relationship does not appear to be very strong. It can be seen that the association is positive (larger values in one variable are associated with larger values in the other variable).

Checking the relationship between variables TimeBetweenMessages and timeToLogin.

```{r}

p <- ggplot(loginData,aes(x=timeBetweenMessages,y=timeToLogin))

p <- p + geom\_point()

p <- p + labs(title = "Time between messages vs login time", x="Time between messages (s)", y = "login time (s)")

p

```

There is no clear relationship between the two variables.

a. The plot does not show a linear correlation between variables TimeBetweenMessages and timeToLogin.

b. The prediction for time to login if time between messages is 200 should not be made using linear regression as the relation between variables is not linear.

c. A linear model should not be used (see above). Also, 20 is not in the range for time between messages so this prediction should not be made even if a linear model were suitable.

## Exercise 3 - Anscombe dataset

Checking data

```{r}

anscombe

```

Set 1

```{r}

ans.plot1 <- ggplot(data=anscombe,aes(x=x1,y=y1))

ans.plot1 + geom\_point() + stat\_smooth(method="lm", formula = y ~ x) + labs(title="Set 1")

```

Set 2

```{r}

ans.plot2 <- ggplot(data=anscombe,aes(x=x2,y=y2))

ans.plot2 + geom\_point() + stat\_smooth(method="lm", formula = y ~ x) + labs(title="Set 2")

```

Set 3

```{r}

ans.plot3 <- ggplot(data=anscombe,aes(x=x3,y=y3))

ans.plot3 + geom\_point() + stat\_smooth(method="lm", formula = y ~ x) + labs(title="Set 3")

```

Set 4

```{r}

ans.plot4 <- ggplot(data=anscombe,aes(x=x4,y=y4, ))

ans.plot4 + geom\_point() + stat\_smooth(method="lm", formula = y ~ x) + labs(title="Set 4")

```

the 2 sets were discussed during class time so the observations are as explained in the class.

## Exercise 4

Loading the data.

```{r}

Advertising <- read.csv("Advertising.csv", header=T, stringsAsFactors = T)

```

Checking correlations

```{r}

cor(Advertising[1:5])

```

Sales and TV have a strong correlation (0.78222442). Sales and radio also have a correlation (0.57622257), but this is weaker.

```{r}

p <- ggplot(Advertising,aes(x=sales,y=TV))

p <- p + geom\_point()

p <- p + labs(title = "Sales vs TV advertisng", x="sales", y = "TV expenditure")

p

```

Note that the unit of measures are not given.

The relationship does not appear to be linear. The relation is positive but, as the TV expenditure increases, the spread of values for sales widens.

Plotting sales vs radio expediture

```{r}

p <- ggplot(Advertising,aes(x=sales,y=radio))

p <- p + geom\_point()

p <- p + labs(title = "Sales vs radio advertisng", x="sales", y = "radio expenditure")

p

```

No linear pattern can be seen.

## Exercise 5

Creating the dataset.

```{r}

Time <- c(9.2,13.3,8,11.9,12.1,8.9,10.8,7.2,14.9,7.3,6.3,10.4,13.8,7.6)

Recall <- c(61,53,57,50,53,54,54,6,46,87,88,52,45,72)

irTool <- data.frame(Time=Time, Recall=Recall)

```

checking correlation using the Spearman correlation coefficient as Recall is a discrete variable.

```{r}

cor(irTool$Time, irTool$Recall)

cor(irTool$Time, irTool$Recall, method="spearman")

```

there is a negative correlation between the 2 variables.

Plotting sales vs radio expediture

```{r}

p <- ggplot(irTool,aes(x=Time,y=Recall))

p <- p + geom\_point()

p <- p + labs(title = "Time vs recall", x="Time (mins)", y = "% recall")

p

```

It can be seen that a higher time time taken is generally associated with a lower recall score. The relationship does not match a straight line. Use Spearman correlation coefficient to check strength of association.