Hochschule Anhalt
Anhalt University of Applied Sciences

Economic Statistics and Empirical Methods

Homework Assignment Group 6 (A1)

Submitted by:

Ankush Bohora, 5021206

Gaurav Bhatia, 5027013

Lecturer: Prof. Dr. Achim Wübker

Dr. Mohammed Abujarad

Date: February 23, 2021

Exercise 1

(a) Define the variable v1 as the vector (3.7, −4.2).

Code:

```
# Defining the variable v1 as the vector (3.7, −4.2)

v1 <- c (3.7, -4.2)
v1
```

Output:

```
> v1 <- c (3.7,-4.2)
> v1
[1]  3.7 -4.2
```

(b) Define the variable v2 as the vector (5, 10, 15, ..., 40, 45, 50).

Code:

```
# Defining the variable v2 as the vector (5, 10, 15, ..., 40, 45, 50)

v2 <- seq (5, 50, 5)
v2
```

Output:

```
> v2 <- seq(5,50,5)
> v2
 [1]  5 10 15 20 25 30 35 40 45 50
```

The seq function generates values from 5, increasing every next value by 5 up to the 50.

(c) Define the variable v3 as the vector (3, 7, −4, 2, 5, 10, 15, ..., 40, 45, 50). You may use v1 and v2 for this.

Code:

```
# Defining the vector v3 by combining the vectors v1 and v2.

v3 <- c (v1, v2)
v3
```

Output:

```
> v3 <- c(v1,v2)
> v3
 [1]  3.7 -4.2  5.0 10.0 15.0 20.0 25.0 30.0 35.0 40.0 45.0 50.0
```

The vectors v1 and v2 are combined to obtain the vector v3.

(d) Define the variable v4 as the vector (0.5, 0.6, 0.7, . . . , 1.8, 1.9, 2.0).

Code:

```
# Defining the vector v4

v4 <- seq (0.5, 2, 0.1)
v4
```

Output:

```
 v4 <- seq(0.5,2,0.1)
 v4
[1] 0.5 0.6 0.7 0.8 0.9 1.0 1.1 1.2 1.3 1.4 1.5 1.6 1.7 1.8 1.9 2.0
```

Using the sequence function for defining the vector v4, starting from value of first element 0.5, incrementing by 0.1, up to last element 2.

(e) Sum over all elements of v1. Sum over all elements of v2.

Code:

```
# Sum of all the elements of v1 and of v2.

sum(v1)
sum(v2)
```

Output:

```
> sum(v1)
[1] -0.5
> sum(v2)
[1] 275
```

The sum function gives the sum of all the elements of the vector. The sum of all the elements of v1 is -0.5, and of v2 is 275.

(f) What is the product of all elements of the vector v4?

Code:

```
# Product of all the elements of v4

prod(v4)
```

Output:

```
> prod(v4)
[1] 10.13709
```

The product of all elements of a vector can be obtained by prod function. The product of all elements of v4 is 10.13709.

Exercise 2

The following scores were achieved by 20 students in a written exam:
50, 63.5, 41, 72.5, 74, 71.5, 98, 61.5, 11.5, 69, 47, 50, 42.5, 51.5, 82, 52, 39, 23, 73, 69.
Build classes of width 25 points starting at 0.
calculate for the classified data,
(a) arithmetic mean,
(b) empirical variance
(c) median,
(d) draw the histogram for these classified data.

Code:

```
#Following scores were achieved by 20 students in a written exam, the data set of these numerical values is defined by examscores.

examscores <- c (50, 63.5, 41, 72.5, 74, 71.5, 98, 61.5, 11.5, 69, 47, 50, 42.5, 51.5, 82, 52, 39, 23, 73, 69)

examscores

#Building classes of width 25 points starting at 0.

breaks <- seq (0, 100, by= 25)

classes <- cut (examscores, breaks, labels = c ("class 0-25", "class 25-50", "class 50-75", "class 75-100"), right = T)

examscore.classes <- table(classes)

examscore.classes
```

Output:

```
> examscores <- c(50, 63.5, 41, 72.5, 74, 71.5, 98, 61.5, 11.5, 69, 47, 50, 42.5, 51.5, 82, 52, 39, 23, 73, 69)
> examscores
 [1] 50.0 63.5 41.0 72.5 74.0 71.5 98.0 61.5 11.5 69.0 47.0 50.0 42.5 51.5 82.0 52.0 39.0 23.0 73.0 69.0
>
> breaks <- seq(0, 100, by= 25)
> breaks
[1]   0  25  50  75 100
>
> classes <- cut(examscores, breaks, labels = c("class 0-25", "class 25-50", "class 50-75", "class 75-100"), right = T)
>
> examscore.classes <- table(classes)
> examscore.classes
classes
  class 0-25  class 25-50  class 50-75 class 75-100
           2            6           10            2
```

The exam scores for the students are defined under numeric vector examscores. The cut function divides the range of examscores into intervals. The cut function is used to cut the exmascores (classify the data) into the number of intervals such as 0-25, 26-50, 51-75 and 76-100. The examscore.classes shows the number of students in each class as per the exam scores they have obtained.

Code:

(a) arithmetic mean for the classified data using mean function

mean (examscores)

Output:
```
> mean(examscores)
[1] 57.075
```

The arithmetic mean of the classified data is 57.075.

(b) empirical variance for the classified data using var function

var (examscores)

Output:
```
> var(examscores)
[1] 416.9283
```

The empirical variance of the classified data is 416.9283.

(c) median for the classified data using the median function

median (examscores)

Output:
```
> median(examscores)
[1] 56.75
```

The sample median of the classified data is 56.75.

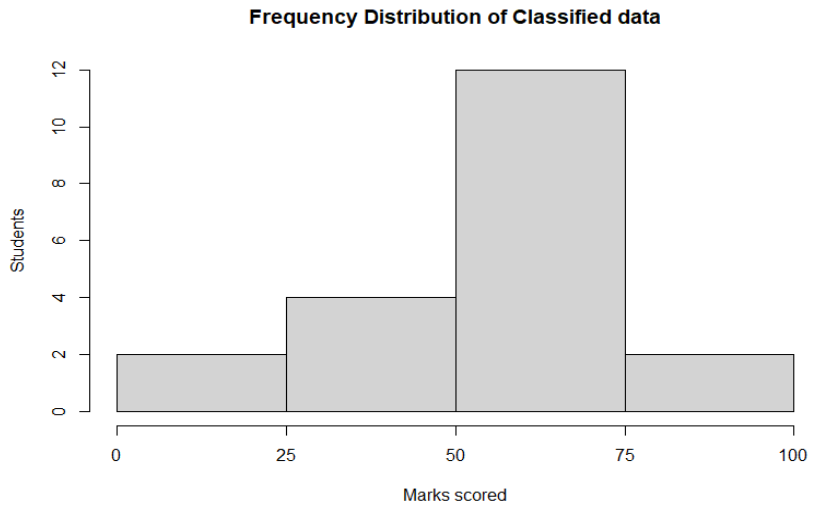
(d) histogram for the classified data

histogram for these classified data.

hist (examscores, breaks, right = FALSE, main = "Frequency Distribution of Classified data", xaxt = 'n', xlab = "Marks scored", ylab = "Students")

axis (side = 1, at =seq (0,100,25), labels= seq (0,100,25))

The generic function hist computes a histogram of the given data values. Based on the marks scored by the students, the histogram is computed which also shows the different classes, and the number of students in each class.



Exercise 3

Calculate the following for the variables turnover and gross value,

(a) empirical variance.
(b) empirical covariance.
(c) empirical correlation.
(d) In addition, a test is to be carried out to find out whether there is a non-zero correlation between the two variables. α is equal to 0.05.

Code:

```
#defining the two variables turnover and gross value

turnover <- c (2970, 552, 299, 1100, 3463, 2343, 3630, 3224, 2000, 5008)
grossvalue <- c (23273, 5083, 2807, 5258, 20442, 15076, 28360, 19812, 13379, 20403)
```

The variables are defined as numeric vectors.

```
# (a) Empirical variance for turnover and gross value using var function

var(turnover)
var(grossvalue)
```

Output:

```
> var(grossvalue)
[1] 74659536
> var(turnover)
[1] 2232919
```

The variance is computed using var function. The var for turnover is 2232919, and for gross value is 74659536.

```
# (b) Empirical covariance between turnover and gross value

cov (turnover, grossvalue)
```

Output:

```
> cov(turnover, grossvalue)
[1] 11334761
```

The covariance between the two variables can be computed using the cov function. The covariance between the two variables is 11334761.

```
# (c) Empirical correlation between turnover and gross value

cor (turnover, grossvalue)
```

output:

```
> cor(turnover, grossvalue)
[1] 0.8778761
```

Correlation between two variables can be computed using cor function. The correlation between turnover and gross value is 0.8778 as seen above.

(d) correlation test to find out whether there is a non-zero correlation between the two variables. α is equal to 0.05.

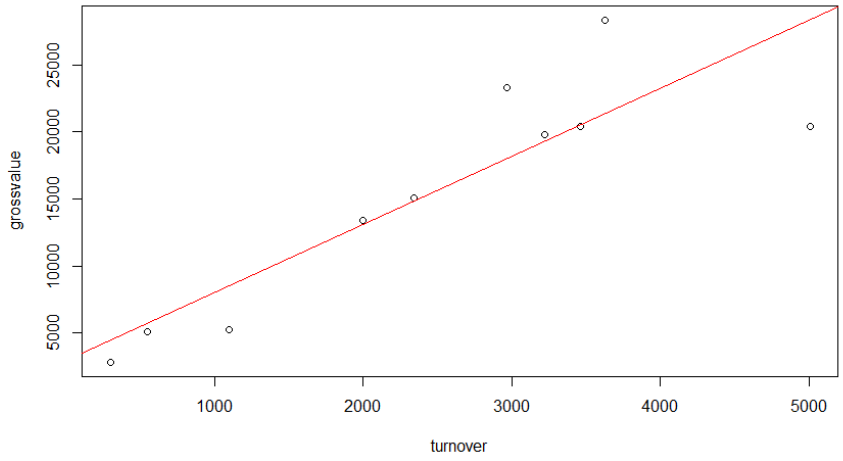
Code:

```
#mod1 represents the linear model
plot (turnover, grossvalue)
mod1<- lm (grossvalue~turnover)
abline (mod1, col="red")

test <- cor.test (turnover, grossvalue)
test
```

Output:

```
> test <- cor.test (turnover, grossvalue)
> test

        Pearson's product-moment correlation

data:  turnover and grossvalue
t = 5.1849, df = 8, p-value = 0.0008377
alternative hypothesis: true correlation is not equal to 0
95 percent confidence interval:
 0.5550383 0.9708698
sample estimates:
      cor
0.8778761
```

The test function is used to carry out the correlation test in order to check if the null hypothesis is true. Based on the output, the **p-value** is **0.0008377**. Since the **p-value < 0.05**, we can reject the null hypothesis and accept the alternative hypothesis. Therefore, the correlation between the two variables is likely to be true and is not equal to 0.



The correlation coefficient measures the strength and direction of a linear relationship between the two variables on a scatterplot. Furthermore, the model shows a positive correlation (positive uphill linear pattern) between the two variables. Also, as the coefficient of correlation (0.8778 in this example) approaches -1 or 1, the strength of the relationship increases, and the data points tend to fall closer to a line.

Exercise 4

The data for the 50 patients is stored in file magnets.txt. Download this file to your computer and store it in the working directory of R. Read the content of the file into an R data frame.

Code:

```
# Importing the data for 50 patients into a R data frame
getwd ()
setwd ("D:/Econstat")
getwd ()

# Read data from the file
magnets <- read.csv('magnets.txt')
view (magnets)
```

Using setwd and getwd we move to working directory and using read.csv we read the data from the file. The data in the magnets.txt file has been saved in the working directory Econstat. The content of the file has been read into a R data frame.

(a) What is the sample average of the change in score between the patient's rating before the application of the device and the rating after the application?

Code:

```
# (a) sample mean of change in score between the patient's rating before the application of the device and the rating after the application

mean (magnets$change)
```

Output:

```
> mean(magnets$change)
[1] 3.74
```

The sample average of change i.e. change in score between the patient's rating before the application of the device and the rating after the application, is 3.74.

(b) Is the variable active a factor or a numeric variable?

Code:

```
# variable active
str (magnets$active)
summary (magnets$active)
```

Output:

```
> str(magnets$active)
 chr [1:50] "\"1\"" "\"1\"" "\"1\"" "\"1\"" "\"1\"" "\"1\"" "\"1\"" "\"1\"" "\"1\"" "\"1\"" "\"1\"" "\"1\""
> summary(magnets$active)
   Length     Class      Mode
       50 character character
```

The str function compactly displays the internal structure of R object and is an alternative to summary function. The output shows that the data type of the variable active is not factor but is character.

(c) Compute the average value of the variable change for the patients that received an active magnet and average value for those that received an inactive placebo.

Code:

```
#Activemagnnets represents patients that received an active magnet
Activemagnets <- subset (magnets, magnets$active == '"1"')
Activemagnets

#Inactiveplacebo represents patients that received an inactive placebo
Inactiveplacebo <- subset (magnets, magnets$active == '"2"')
Inactiveplacebo

#The mean for Activemagnets and Inactiveplacebo
mean (Activemagnets$change)
mean (Inactiveplacebo$change)
```

Output:

```
> mean(Activemagnets$change)
[1] 5.034483
> mean(Inactiveplacebo$change)
[1] 1.952381
```

The average value of the variable change for the patients that received an active magnet is 5.034483 and average value for those that received an inactive placebo is 1.952381.

(d) Compute the sample standard deviation of the variable change for the patients that received an active magnet and the sample standard deviation for those that received an inactive placebo.

Code:

```
#SD for patients that received an active magnet
sd (Activemagnets$change)

#SD for patients that received an inactive placebo
sd (Inactiveplacebo$change)
```

Output:

```
> sd(Activemagnets$change)
[1] 3.26762
> sd(Inactiveplacebo$change)
[1] 2.673503
```

The standard deviation of the variable change for the patients that received an active magnet is 3.26762 and standard deviation for those that received an inactive placebo is 2.673503.

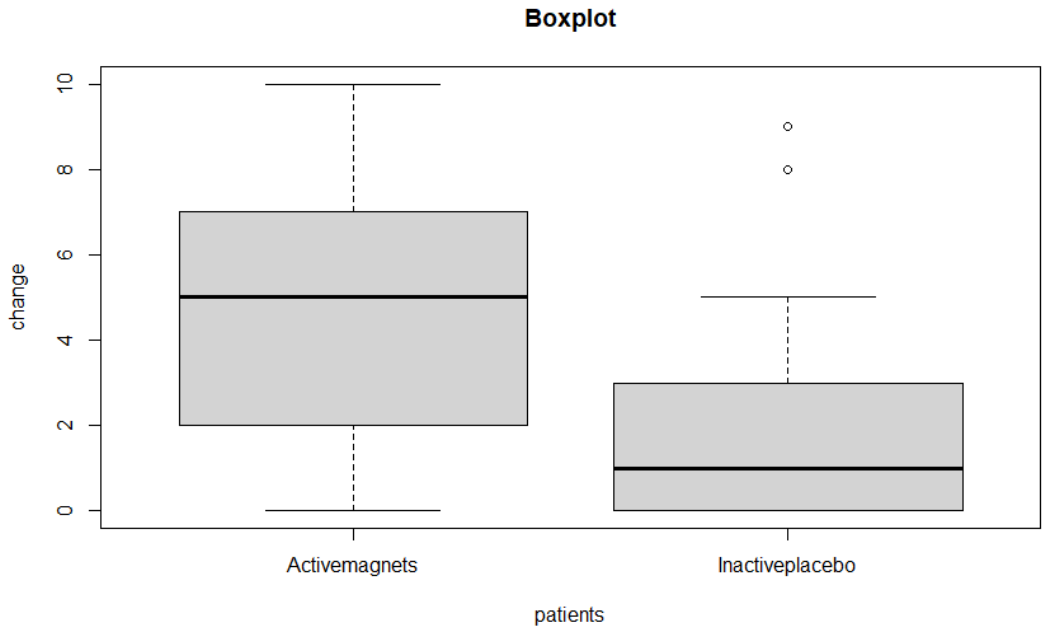
(e) Produce a boxplot of the variable change for the patients that received an active magnet and for patients that received an inactive placebo. What is the number of outliers in each subsequence?

Code:

```
# boxplot of the variable change for the patients that received an active magnet and for patients that received an inactive placebo

name <- c ("Activemagnets", "Inactiveplacebo")
boxplot <- boxplot (magnets$change ~ magnets$active, names=name, main=
          "Boxplot", xlab="patients", ylab="change")

#boxplot$out gives the outlier values and the length of outliers provides the number of outliers

outliers <- boxplot$out
outliers
length (outliers)
```

Output:

```
> outliers <- boxplot$out
> outliers
[1] 8 9
> length (outliers)
[1] 2
```



It can be observed that the inactive placebo subsequence has 2 outliers i.e. 2 data points are located outside the whiskers of the box plot. This is as per the outcome and can be seen in the Boxplot as well. And for the active magnets subsequence there are no outliers. But the overall spread in Activemagnets boxplot shown by extreme values at the end of the whiskers indicates wider distribution of data.

Exercise 5

The manager of the purchasing department of a large company would like to develop a regression model to predict the average amount of time it takes to process a given number of invoices. Over a 30-day period, data are collected on the number of invoices processed and the total time taken (in hours). The data are available in the file invoices.txt. The following model was fit to the data:
Y = α + mx + ϵ;
Where *Y* is the processing time and *x* is the number of invoices. Complete the following tasks.

(a) Construct a scatter plot of processing time versus the number of invoices. Does the plot suggest a linear relationship?

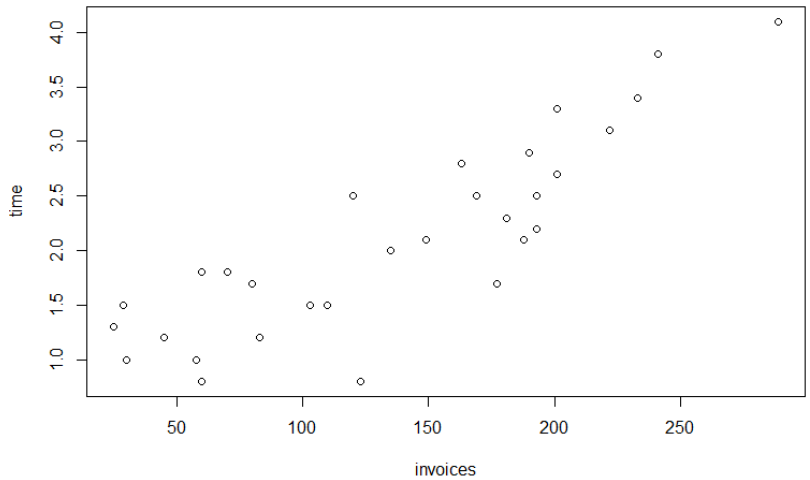
Code:

```
# Importing the dataset
invoices = read.csv ('invoices.csv')
# Compactly displaying the internal structure of invoices.
str (invoices)

# Create Data Frame for tightly coupled collections of variables.
invo <- data.frame (invoices)
inv <- invoices$Invoices
t <- invoices$Time

# Create Scatterplot displays the relationship between time & Number of invoices.
plot (inv,t, xlab = "invoices", ylab = "time")
# Correlation of Invoices and Time.
cor (inv,t)
```

Output:

```
> # Correlation of  Invoices and Time.
> cor(inv,t)
[1] 0.8673299
```

The scatter plot is as shown below for time vs invoices:



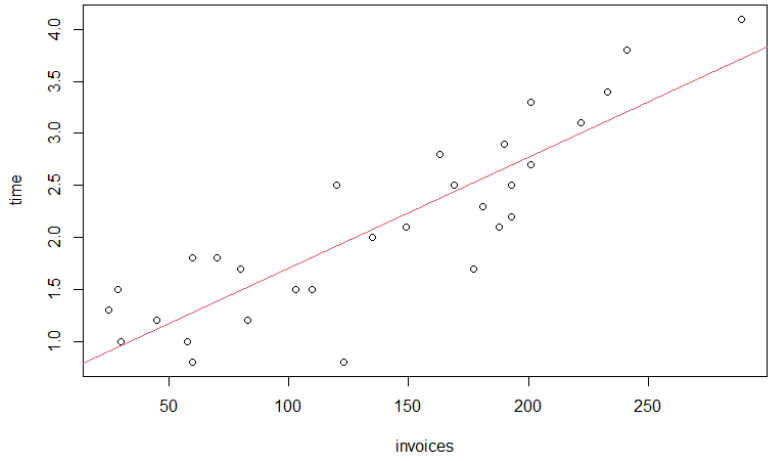
With 0.867 Correlation and the scatter plot shows a strong uphill linear relationship. As the correlation coefficient is close to 1, it shows a strong linear relationship.

(b) Fit a regression line predicting processing time from the number of invoices.

Code:

```
# Fit a regression line predicting processing time from the number of invoices
# lm is used to fit linear models

fit <- lm (t~inv, data= invoices)
abline (fit, col=2)
```

Output:



The regression line in red can be seen in the above scatter plot. This regression line best predicts the processing time based on the number of invoices. Most of the data points lie near the regression line.

(c) Using the regression equation from subtask (b) to find a point estimate and a 95% prediction interval for the time taken to process 130 invoices.

Code:

```
# find a point estimate and a 95% prediction interval for the time taken to process 130
invoices.
# lm is used to fit linear models
# mod2 represent Model and PredI represent Prediction Interval.

mod2 <- lm(t~ inv)
invo1 <- data.frame(inv = 130)
PredI <- predict (mod2, invo1, interval = "prediction", level = 0.95)
PredI
```

Output:

```
> invo1 <- data.frame(inv = 130)
> PredI <- predict(mod2, invo1,interval = "prediction", level = 0.95)
> PredI
       fit      lwr     upr
1 2.024975 1.091841 2.95811
```

Models for lm are specified symbolically. A typical model has the form response ~ terms where response is the (numeric) response vector (time in our problem) and terms are a series of terms (invoices in our problem) which specifies a linear predictor for response. The Point estimate is 2.024975, and 95% prediction interval [1.091841, 2.95811].

(d) Find a 95% confidence interval for the start-up time, i.e., *α*.

Code:

```
# Find a 95% confidence interval for the start-up time
# ConfiI represents Confidence Interval.

invo2 <- data.frame (inv = 0)
ConfiI <- predict (mod2, invo2, interval = "confidence", level = .95)
ConfiI
```

Output:

```
> invo2<- data.frame(inv = 0)
> ConfiI <- predict(mod2,invo2,interval = "confidence", level = .95)
> ConfiI
       fit       lwr      upr
1 0.6421855 0.2768888 1.007482
```

At 95% Confidence interval for start-up time is [**0.2768888, 1.007482**].

(e) Can you at significance level α = 0.05 reject the hypothesis that the line passes through (0, 0)?

Code:
summary(mod2)

Output:

```
> summary(M)

Call:
lm(formula = t ~ inv)

Residuals:
     Min       1Q   Median       3Q      Max
-1.15052 -0.26537  0.04945  0.38986  0.59434

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept) 0.642185   0.178332   3.601  0.00121 **
inv         0.010637   0.001154   9.221 5.59e-10 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.4481 on 28 degrees of freedom
Multiple R-squared:  0.7523,    Adjusted R-squared:  0.7434
F-statistic: 85.02 on 1 and 28 DF,  p-value: 5.592e-10
```

Since the p value of intercept is p-value: 5.592e-10 < α=0.05, we can reject the null hypothesis that the line passes through (0,0). Since the p-value is significantly small, we can confidently reject the null hypothesis and accept the alternative that the line does not pass through (0,0).

(f) Suppose that a best practice benchmark for the average processing time for an additional invoice is 0.01 hours (or 0.6 minutes). Test the null hypothesis H_0: m = 0.01 against a two-sided alternative. Interpret your result. What is a 90% confidence interval for the slope m in the regression model?

Code:

```
# Computes confidence intervals for one or more parameters in a fitted model.
confint (mod2, level = .90)
```

Output:

```
> confint(mod2,level = .90)
                    5 %       95 %
(Intercept) 0.338819283 0.94555170
inv         0.008674459 0.01259923
```

The 90% confidence interval for the slope m in the regression model is [0.008674459, 0.01259923]. Since as per null hypothesis H_0: m = 0.01 is in between the confidence interval, thus in this case we cannot reject the null hypothesis.

(g) How large a part of the processing time ($\Sigma^n_{i=1}(y_i - y)^2$) is not explained by the number of invoices?

The residuals standard error is 0.4481. So, total sum of square is $(0.4481)^2$ = 0.20079361, which is the processing time, and is not explained by the number of invoices.

(h) Describe any weaknesses in your model.

Small data set is not enough to provide best fitted & accurate line between number of invoices and time.