

COMPUTER EXERCISE 1  
NHUT CAO – 906939

1.2 Homework

- a) Read file "tobacco.txt", set seed

Fit1 is the linear regression model. Summary(fit1) returns the summary as in the console  
Constant term is the Intercept

```
1 # 1.2 homework
2
3 tobacco = read.table("tobacco.txt", header=T, sep = "\t")
4 tobacco.matrix <- as.matrix(tobacco)
5 # colnames(tobacco)
6 set.seed(123)
7
8 # (a)
9 # Linear regression model
10 # constant term = Intercept
11 fit1 <- lm(ILL~CONSUMPTION, data=tobacco)
12 summary(fit1)
13
```

Residuals:

Min	1Q	Median	3Q	Max
-169.016	-32.813	0.004	45.804	136.914

Coefficients:

	Estimate	Std. Error	t value	Pr(> t )
(Intercept)	65.74886	48.95871	1.343	0.21217
CONSUMPTION	0.22912	0.06921	3.310	0.00908 **

---

Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 84.13 on 9 degrees of freedom

Multiple R-squared: 0.549, Adjusted R-squared: 0.4989

F-statistic: 10.96 on 1 and 9 DF, p-value: 0.009081

[View Coefficients](#)

b)

X is a matrix here

	Intercept	V2
1	1	220
2	1	250
3	1	310
4	1	510
5	1	380
6	1	455
7	1	1280
8	1	460
9	1	530
10	1	1115
11	1	1145

```
exercise_1.R x b_coefficient x X x
7
8 # (a)
9 # Linear regression model
10 # constant term = Intercept
11 fit1 <- lm(ILL~CONSUMPTION, data=tobacco)
12 summary(fit1)
13
14 # (b)
15 consumption <- as.matrix(tobacco$CONSUMPTION)
16
17 n <- nrow(tobacco) # get number of rows
18 Intercept <- rep(1,n)
19 X = cbind(Intercept, consumption) # add constant term to the matrix
20
21 # estimate the regression coefficient with least squares method
22 #and give interpretations for it
23 b_coefficient <- solve(t(X) %*% X) %*% t(X)%*%(tobacco$ILL)
24
25 # (c)
26 summary(fit1)$r.squared
27 # The coefficient of determination of the model is 54.9%, which corresponds to
28 # "multiple R-squared" in the output
29
30
31
32
29:1 (Top Level) ↕

Console Terminal x R Markdown x Jobs x
R 4.1.1 · ~/MS-C2128/Week 1/Computer exercises 1-20211010/ ↗
> # (b)
> consumption <- as.matrix(tobacco$CONSUMPTION)
> n <- nrow(tobacco) # get number of rows
> Intercept <- rep(1,n)
> X = cbind(Intercept, consumption) # add constant term to the matrix
> # estimate the regression coefficient with least squares method
> #and give interpretations for it
> b_coefficient <- solve(t(X) %*% X) %*% t(X)%*%(tobacco$ILL)
> # (c)
> summary(fit1)$r.squared
[1] 0.54904
> View(b_coefficient)
>
```

We got “coefficient” by least squares method, Which is similar to what we got from the summary function, the intercept is the constant term, value of 2 is the coefficient, which is 0.2291153:

	V1
Intercept	65.7488570
2	0.2291153

c) The coefficient of determination of the model: 54.9%

The screenshot shows the R Studio environment. The script editor on the left contains the following code:

```

12 summary(fit1)
13
14
15 # (b)
16 consumption <- as.matrix(tobacco$CONSUMPTION)
17
18 n <- nrow(tobacco) # get number of rows
19 Intercept <- rep(1,n)
20 X = cbind(Intercept, consumption) # add constant term to the matrix
21
22 # estimate the regression coefficient with least squares method
23 #and give interpretations for it
24 b_coefficient <- solve(t(X) %*% X) %*% t(X)%*(tobacco$ILL)
25
26
27 # (c)
28 summary(fit1)$r.squared
29 # The coefficient of determination of the model is 54.9%, which corresponds to
30 # "multiple R-squared" in the output
31
32

```

The console output on the bottom left shows the results of the least squares method:

```

R 4.1.1 ~ /MS-C2128/Week 1/Computer exercises 1-20211010/
estimate std. error t value Pr(>|t|)
(Intercept) 65.74886 48.95871 1.343 0.21217
CONSUMPTION 0.22912 0.06921 3.310 0.00908 **
---
Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 84.13 on 9 degrees of freedom
Multiple R-squared: 0.549, Adjusted R-squared: 0.4989
F-statistic: 10.96 on 1 and 9 DF, p-value: 0.009081

```

The environment pane on the right shows the following objects:

- `b_coefficient`: num [1:2, 1] 65.749 0.229
- `consumption`: int [1:11, 1] 220 250 310 510 380 455 1280 460 ...
- `fit1`: List of 12
- `tobacco`: 11 obs. of 15 variables
- `tobacco.matrix`: num [1:11, 1:15] 32.9 43.1 63.1 123.3 85.5 ...
- `X`: num [1:11, 1:2] 1 1 1 1 1 1 1 1 1 ...

The 'Values' section shows:

- `Intercept`: num [1:11] 1 1 1 1 1 1 1 1 1 ...
- `n`: 11L

d) & e)

Here the level of significance is 1% = 0.01

For F-test: the p-value is less than the significance level, therefore we can reject the null hypothesis and conclude that this model is statistically significant.

For t-test: the CONSUMPTION variable has p-value is also lesser than the level of significance, therefore it is statistically significant.

Compare those p-values, we see that the p-value ( $\Pr(>|t|) = 0.00908$ ) is almost equal to the p-value in part d, which is 0.009081. Both the p-value represents the probability of obtaining the test results, under the assumption of the null hypothesis. In this case, we use them to compare with the level of significance to conclude whether it's statistically significance or not.

The screenshot shows the RStudio environment with the following components:

- Source Editor:** Contains R code for a hypothesis test.
 

```

33 # (d)
34 #F-stat: 10.96 on 1 and 9 DF, p-value: 0.009081
35 p_value <- summary(fit1)$coefficients[,4] [2]
36 alpha = 0.01 # level of significance
37 p_value < alpha
38 #return True, so we can reject the null hypothesis
39 #this model is statistically significant according to the F-test
40
41
42 # (e)
43 cons_value <- summary(fit1)$coefficients[,3][2]
44 cons_value > alpha
45 # the variable CONSUMPTION is statistically significance as its t-value is much bigger
46 # than the significance level
47
48 # compare t-value and p-value
49 cons_value < p_value # False
50
51 # inextricably linked, The larger the absolute value of the t-value, the smaller
52 # the p-value, and the greater the evidence against the null hypothesis (statistically significance)
53
49:29 (Top Level) >
      
```
- Console:** Shows the execution of the code.
 

```

R 4.1.1 ~ ./MS-C2128/Week 1/Computer exercises 1-20211010/ >
> #and give interpretations for t
> b_coefficient <- solve(t(X) %*% X) %*% t(X)%*(tobacco$ILL)
> View(b_coefficient)
> # (c)
> summary(fit1)$r.squared
[1] 0.54904
> # (d)
> #F-stat: 10.96 on 1 and 9 DF, p-value: 0.009081
> p_value <- summary(fit1)$coefficients[,4] [2]
> alpha = 0.01 # level of significance
> p_value < alpha
CONSUMPTION
TRUE
> # (e)
> cons_value <- summary(fit1)$coefficients[,3][2]
> cons_value > alpha
CONSUMPTION
TRUE
> # compare t-value and p-value
> cons_value < p_value # False
CONSUMPTION
FALSE
>
      
```
- Environment:** Lists objects in the global environment.
 

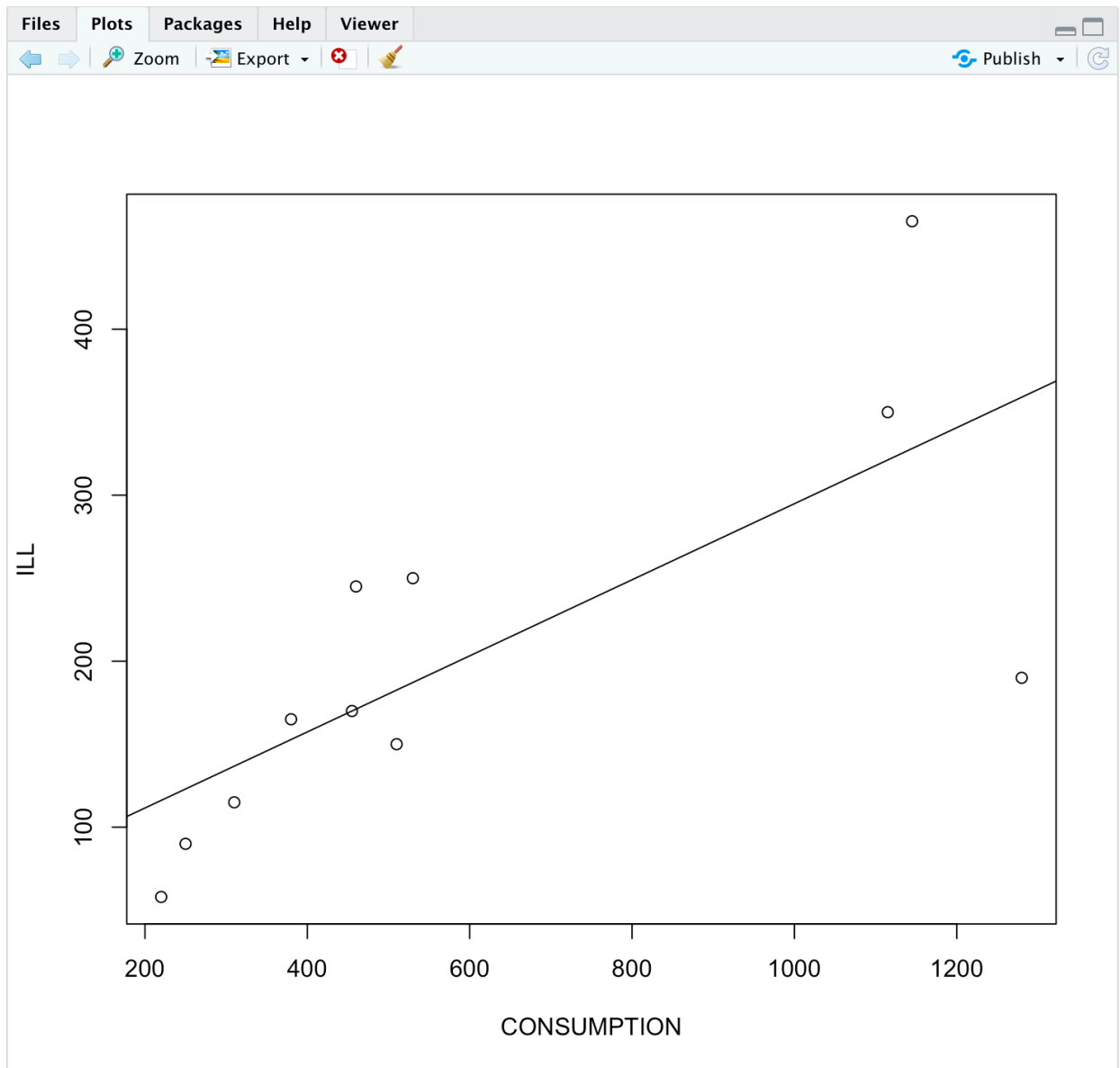
Object	Class	Attributes
b_coefficient	num	[1:2, 1] 65.749 0.229
consumption	int	[1:11, 1] 220 250 310 510 380 455 1280 460 ...
fit1	List of	12
tobacco	11 obs. of 15 variables	
tobacco.matrix	num	[1:11, 1:15] 32.9 43.1 63.1 123.3 85.5 ...
X	num	[1:11, 1:2] 1 1 1 1 1 1 1 1 1 ...
alpha	0.01	
cons_value	Named num	3.31
Intercept	num	[1:11] 1 1 1 1 1 1 1 1 1 ...
n	11L	
p_value	Named num	0.00908

f)

Response variable ILL, explanatory variable CONSUMPTION

```
53  
54 # (f)  
55 plot(tobacco[,5], tobacco[,14], xlab = "CONSUMPTION", ylab = "ILL")  
56 abline(fit1)  
57
```

Below is the scatterplot:



g)

Concept of confidence interval:

Confidence interval displays the probability that a parameter may fall between a pair of values around the mean. It measures the degree of uncertainty or certainty in a sampling method.

FACT: "A level  $(1 - \alpha)$  confidence interval for a parameter  $\theta$  is a random interval that contains the true (non-random) parameter value  $\theta$  with probability  $(1 - \alpha)$ . "

h) For the 99%, the value of the constant term is much wider as it allows one to be more confident that the unknown population parameter is contained within the interval.

```
58 # (h)
59 confint(fit1,level=0.95)
60
61 confint(fit1,level=0.99)
```

```
> confint(fit1,level=0.95)
              2.5 %      97.5 %
(Intercept) -45.00344053 176.5011546
CONSUMPTION  0.07254024  0.3856904
> confint(fit1,level=0.99)
              0.5 %      99.5 %
(Intercept) -93.358900306 224.8566143
CONSUMPTION  0.004178126  0.4540525
> |
```

i)

```
63 # (i)
64
65 k <- 2000
66 bootmat <- matrix(NA, nrow=k, ncol=2)
67 y <- tobacco$ILL
68
69 set.seed(123)
70 for(i in 1:(k-1)){
71   ind <- sample(1:n,replace = TRUE)
72   Xtmp <- X[ind,]
73   ytmp <- y[ind]
74   btmp <- solve(t(Xtmp)%*%Xtmp)%*%t(Xtmp)%*%ytmp
75   bootmat[i,] <- t(btmp)
76 }
77
78 #b_original <- solve(t(X)%*%X)%*%t(X)%*%y
79 bootmat[k,] <- t(b_coefficient)
80
81 q_consumption <- quantile(bootmat[,2], probs = c(0.025,0.975))
82
83
84
```

The screenshot shows the RStudio environment with three tabs: 'exercise\_1.R', 'q\_consumption', and 'b\_coefficient'. The 'q\_consumption' tab is active, displaying a table with three rows of bootstrap results. The first row shows the mean and standard deviation of the bootstrap distribution. The second and third rows show the 2.5% and 97.5% quantiles, respectively, which define the confidence interval.

Name	Type	Value
q_consumption	double [2]	0.0524 0.4386
2.5%	double [1]	0.05241902
97.5%	double [1]	0.4385944

Compare this to h), we see that this confidence intervals is quite close from those two intervals, especially with the 99% confidence intervals.

j)

Bootstrap approach is a useful alternative to the traditional method of hypothesis testing as it is quite simple, and it mitigates some of the pitfalls encountered within the traditional approach. If the sample size is really large, it cannot necessarily be assumed that the theoretical sampling distribution is normal. This then makes it difficult to determine the standard error of the estimate, and harder to draw reasonable conclusions from the data in the traditional way.

The bootstrapping approach will always work because it does not assume any underlying distribution of the data.

Bootstrapping is a straightforward way to derive the estimates of standard errors and confidence intervals, and it is convenient since it avoids the cost of repeating the experiment to get other groups of sampled data.