COMPUTER EXERCISE 1
NHUT CAO – 906939

1.2 Homework

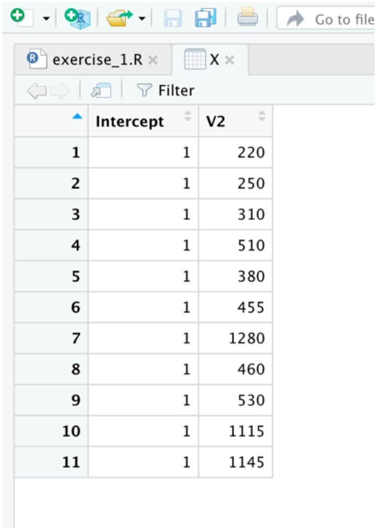
a) Read file “tobacco.txt”, set seed
Fit1 is the linear regression model. Summary(fit1) returns the summary as in the console
Constant term is the Intercept

```
# 1.2 Homework

tobacco = read.table("tobacco.txt", header=T, sep = "\t")
tobacco.matrix <- as.matrix(tobacco)
# colnames(tobacco)
set.seed(123)

# (a)
# Linear regression model
# constant term  = Intercept
fit1 <- lm(ILL~CONSUMPTION, data=tobacco)
summary(fit1)

```

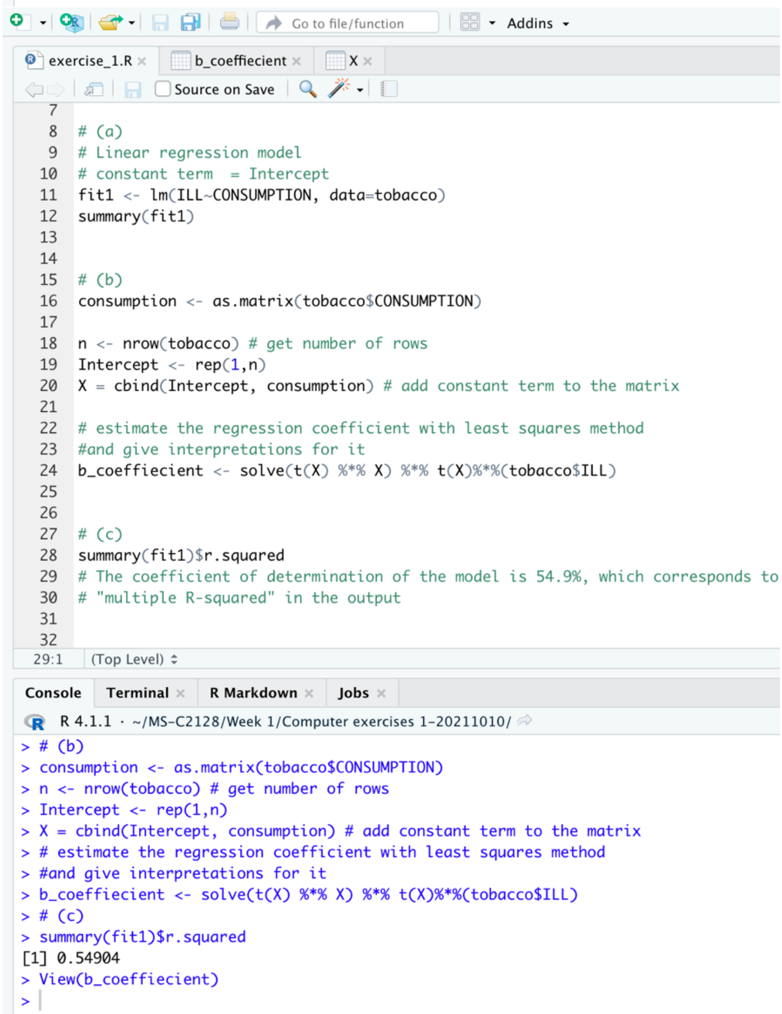
```
Residuals:
     Min       1Q   Median       3Q      Max
-169.016  -32.813    0.004   45.804  136.914

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(b)
```

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




```
# (a)
# Linear regression model
# constant term  = Intercept
fit1 <- lm(ILL~CONSUMPTION, data=tobacco)
summary(fit1)


# (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_coeffiecient <- solve(t(X) %*% X) %*% t(X)%*%(tobacco$ILL)


# (c)
summary(fit1)$r.squared
# The coefficient of determination of the model is 54.9%, which corresponds to
# "multiple R-squared" in the output
```

29:1 (Top Level)

Console | Terminal | R Markdown | Jobs

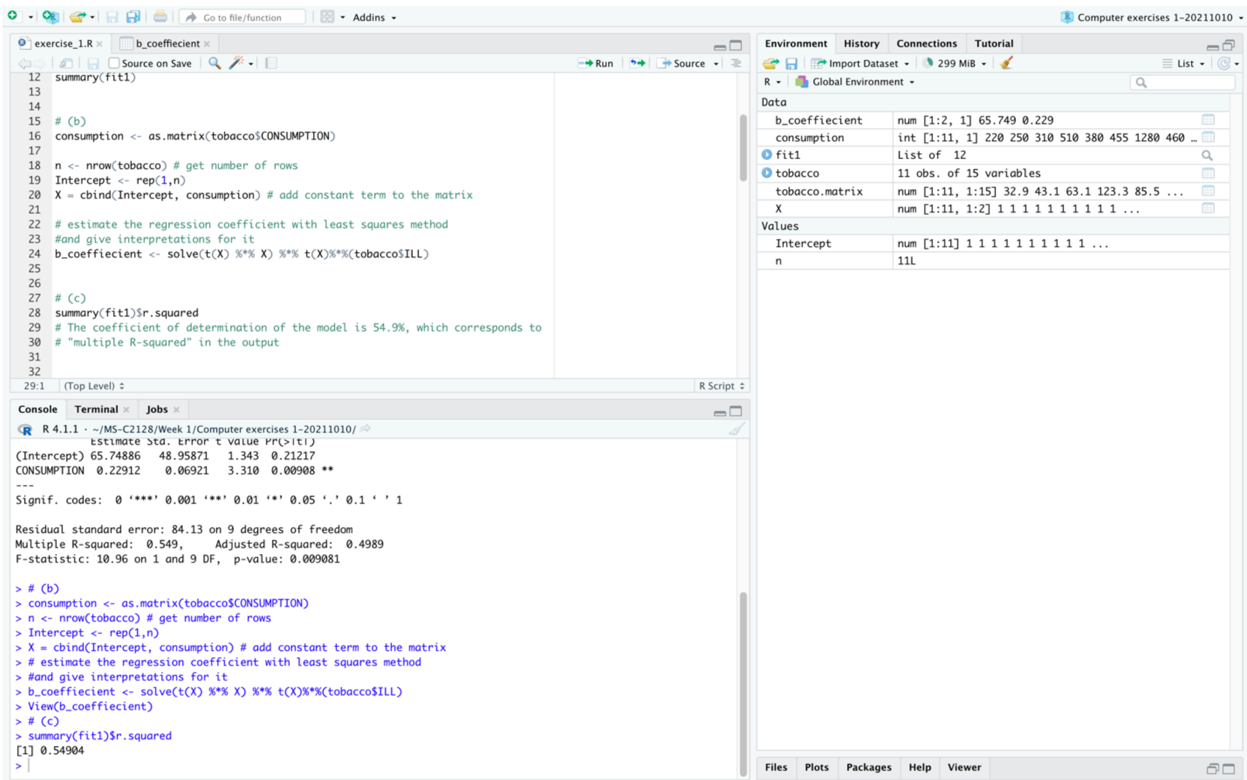
R 4.1.1 · ~/MS-C2128/Week 1/Computer exercises 1-20211010/

```
> # (b)
> consumption <- as.matrix(tobacco$CONSUMPTION)
> n <- nrow(tobacco) # get number of rows
> Intercept <- rep(1,n)
> X = cbind(Intercept, consumption) # add constant term to the matrix
> # estimate the regression coefficient with least squares method
> #and give interpretations for it
> b_coeffiecient <- solve(t(X) %*% X) %*% t(X)%*%(tobacco$ILL)
> # (c)
> summary(fit1)$r.squared
[1] 0.54904
> View(b_coeffiecient)
>
```

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%



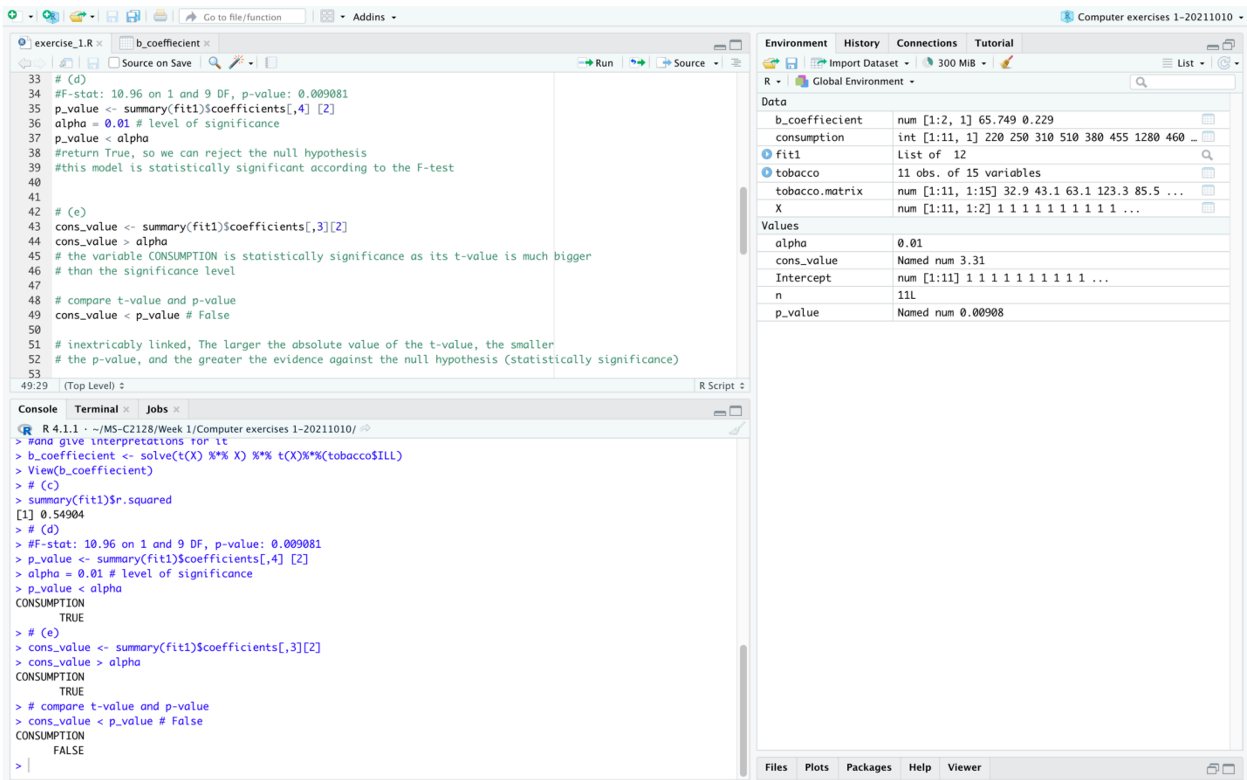
d) & e)

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.

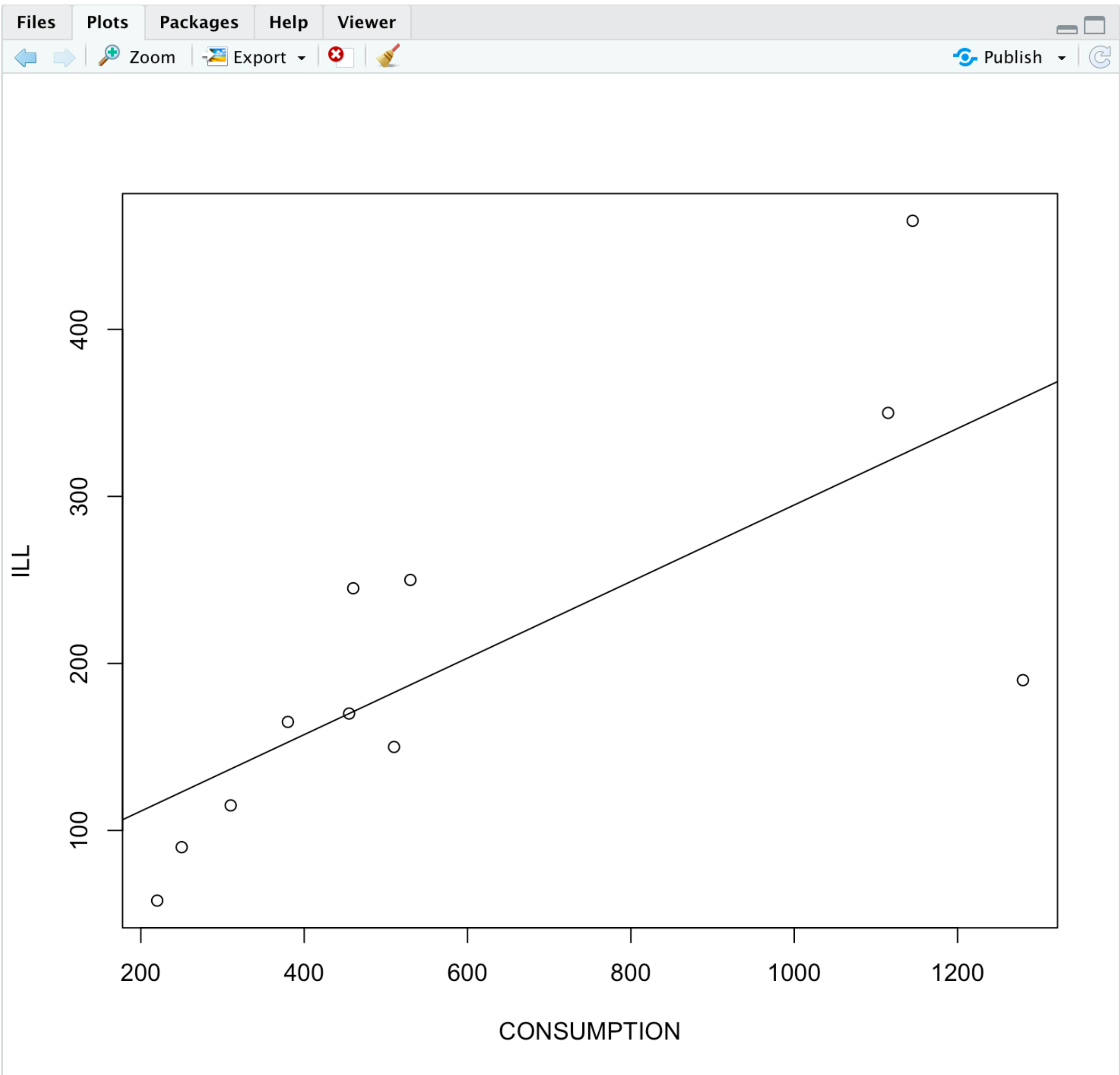


f)

Response variable ILL, explanatory variable CONSUMPTION

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

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 − α) confidence interval for a parameter θ is a random interval that contains the true (non-random) parameter value θ with probability (1 − α). "

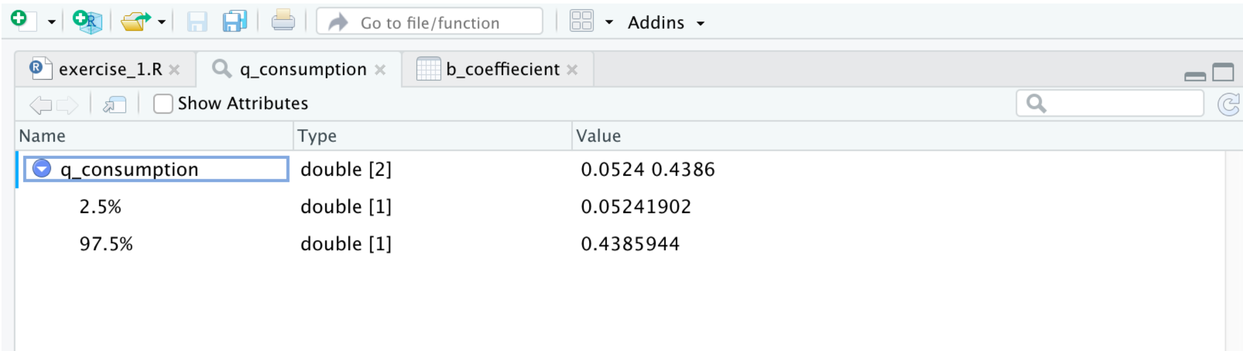
h) For the 99%, the value of the constant term is much wider as it allows one to be more confident that the unknown population parameter is contained within the interval.

```
# (h)
confint(fit1,level=0.95)

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)

```
# (i)

k <- 2000
bootmat <- matrix(NA, nrow=k,ncol=2)
y <- tobacco$ILL

set.seed(123)
for(i in 1:(k-1)){
  ind <- sample(1:n,replace = TRUE)
  Xtmp <- X[ind,]
  ytmp <- y[ind]
  btmp <- solve(t(Xtmp)%*%Xtmp)%*%t(Xtmp)%*%ytmp
  bootmat[i,] <- t(btmp)
}

#b_original <- solve(t(X)%*%X)%*%t(X)%*%y
bootmat[k,] <- t(b_coeffiecient)

q_consumption <- quantile(bootmat[,2], probs = c(0.025,0.975))
```



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.