Task A

Task A.1

```
#!/bin/bash
cut -f 1 $1 | perl -p -e 's/\s+/\n/g;' | perl -p -e 's/&gt;/>/g;s/&lt;/</g;' | grep -e ')' -e '(' -e ':p' -e
    ':D' -e '\^' -e '-_' -e '<' -e '>' | sort | uniq -c | sort -r | head -40 | sed 's/^[ \t]*//g' | sed 's/ /,/
    g' > potential_emoticon.csv
```

Figure1. Screenshot of tweet2emo.sh

In this task, I used 'cut -f 1 $1' this command to extracting the first line of data in msgraw_sample.txt first. Then I used two 'perl' commands to tokenise each line of text by converting space characters to newlines and convert embedded HTML escapes for '>' and '<' back to their original form. After then I used 'grep', 'uniq', 'sort' and 'sed' commands to extract 40 candidate emoticons and their counts for the tweets.

4954	:)
3446	()
3105	:D
2114	(
2012	)
1434	:(
1428	:p
1169	(cont)
1031	;)
980	<3
723	-_-
610	^^
505	:))
473	(^o^)
394	(:
382	(_)
367	:-)
363	(')
321	(^^)
313	(**)
312	:'(
264	(@
260	>
241	=))
228	=)
213	:')
210	(>_<)
200	-__-
187	(*^^*)
177	><
174	(^O^)
167	(*)
141	;-)
133	(;)
130	<

Figure2. Potential Emoticon

Then I picked 20 emoticons and saved them to 'emoticon.csv'

Task A.2

```
for E in `cat emoticon.csv | tr ',' '\t' | cut -f 2`; do
echo $E
echo "~~~~~These are the most frequent words co-occurring with $E~~~~~" | cat >> result.txt
./emoword.py $E < msgraw_sample.txt | grep -v -e '^the$' -e '^in$' -e '^is$' -e '^at$' -e '^at$' -e '^which$' -e
    '^on$' | sort | uniq -c | perl -p -e 's/^\s+//; s/ /\t/; ' | sort -r | head -20 >> result.txt
done
```

Figure3. Screenshot of emword.sh

In this task, I used 'cat emoticon.csv | tr ',' '\t' | cut -f 2' to get the emoticons in the 'emoticon.csv'. Then I used for loop to run over my emoticon list and call the python file getting the co-occurring words with them. After then I excluded the stop words and count the 20 most commonly co-occurring words.

```
~~~~~These are the most frequent words co-occurring with :)~~~~~
96      I'm
94      with
94      ...
9       yuk
9       you,
9       win
9       why
9       what's
9       weer
9       wake
9       va
9       uur
9       um
9       two
9       thing
9       that's
9       tanggal
9       talk
9       t
9       sudah
~~~~~These are the most frequent words co-occurring with :D~~~~~
94      me
94      !
9       yes
9       wkwkwk
9       wait
9       tgl
```

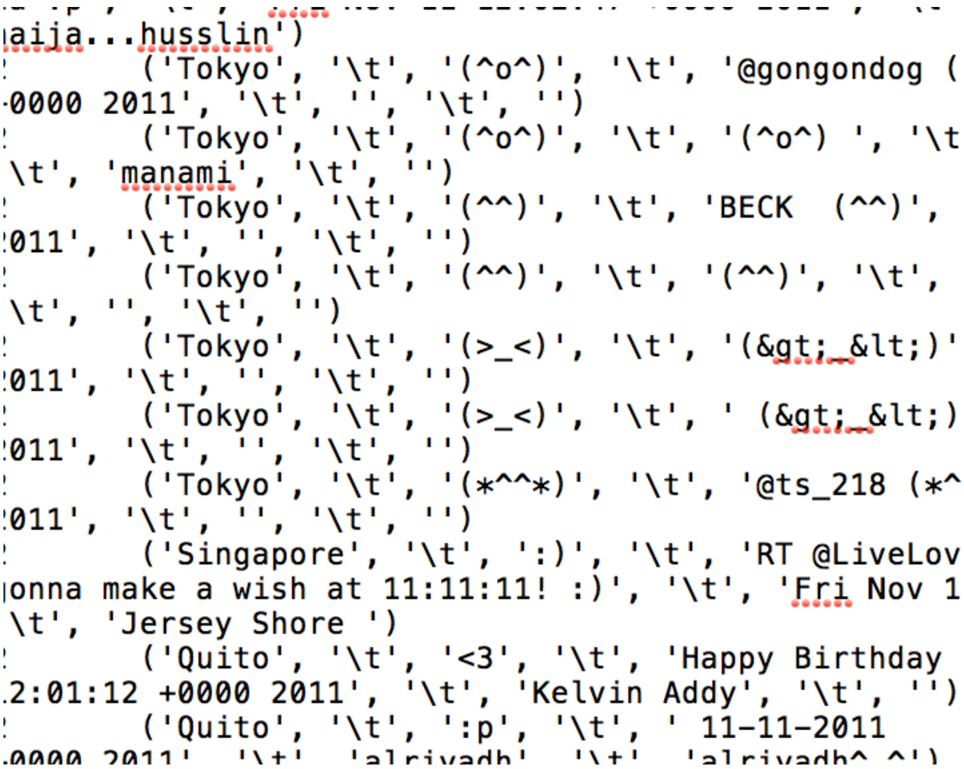
Figure4. Screenshot of Result

Task A.3

```
for E in `cat emoticon.csv | tr ',' '\t' | cut -f 2`; do
echo $E
echo "~~~~~These are the most frequent words co-occurring with $E~~~~~" | cat >> test.txt
./py3_emodata.py $E < msgraw_sample.txt | sort | uniq -c | perl -p -e 's/^\s+//; s/ /\t/; ' | sort -r | head -20
    >> test.txt
done
```

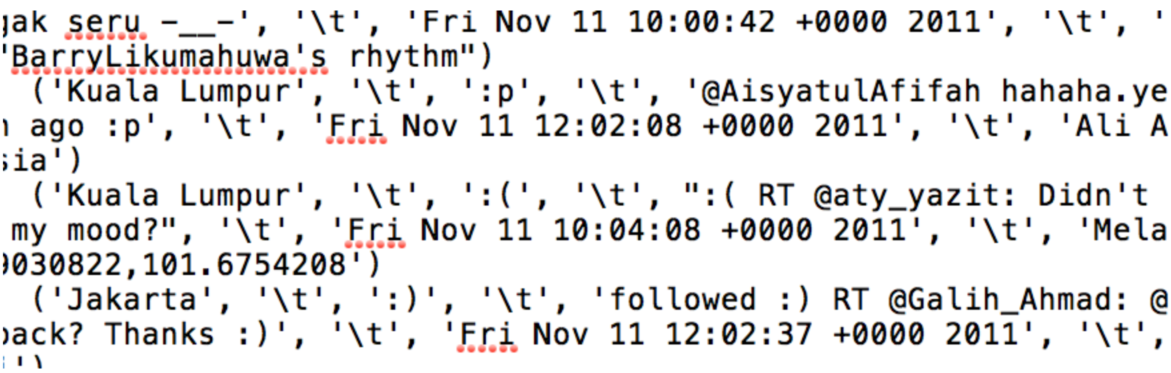
Figure5. Screenshot of Task3 Code

In this task, I basically used the same code as task a.2. I got the 20 most commonly co-occurring information with the emoticons in ‘emoticon.csv’. It is clear from the result that Tokyo, Kuala Lumpur, Jakarta, Singapore and Quito appear with emoticons many times.



```
aija...husslin')
        ('Tokyo', '\t', '(^o^)', '\t', '@gongondog (
0000 2011', '\t', '', '\t', '')
        ('Tokyo', '\t', '(^o^)', '\t', '(^o^) ', '\t
\t', 'manami', '\t', '')
        ('Tokyo', '\t', '(^^)', '\t', 'BECK  (^^)',
011', '\t', '', '\t', '')
        ('Tokyo', '\t', '(^^)', '\t', '(^^)', '\t',
\t', '', '\t', '')
        ('Tokyo', '\t', '(>_<)', '\t', '(&gt;_&lt;)'
011', '\t', '', '\t', '')
        ('Tokyo', '\t', '(>_<)', '\t', ' (&gt;_&lt;)
011', '\t', '', '\t', '')
        ('Tokyo', '\t', '(*^^*)', '\t', '@ts_218 (*^
011', '\t', '', '\t', '')
        ('Singapore', '\t', ':)', '\t', 'RT @LiveLov
onna make a wish at 11:11:11! :)', '\t', 'Fri Nov 1
\t', 'Jersey Shore ')
        ('Quito', '\t', '<3', '\t', 'Happy Birthday
2:01:12 +0000 2011', '\t', 'Kelvin Addy', '\t', '')
        ('Quito', '\t', ':p', '\t', ' 11-11-2011
```

Figure6. Screenshot of Task3 Result



```
jak seru -__-', '\t', 'Fri Nov 11 10:00:42 +0000 2011', '\t', '
'BarryLikumahuwa's rhythm")
  ('Kuala Lumpur', '\t', ':p', '\t', '@AisyatulAfifah hahaha.ye
 ago :p', '\t', 'Fri Nov 11 12:02:08 +0000 2011', '\t', 'Ali A
sia')
  ('Kuala Lumpur', '\t', ':(', '\t', ":( RT @aty_yazit: Didn't
 my mood?", '\t', 'Fri Nov 11 10:04:08 +0000 2011', '\t', 'Mela
030822,101.6754208')
  ('Jakarta', '\t', ':)', '\t', 'followed :) RT @Galih_Ahmad: @
ack? Thanks :)', '\t', 'Fri Nov 11 12:02:37 +0000 2011', '\t',
```

Figure7. Screenshot of Task3 Result

Task B

Task B.1

The following diagrams are plot histograms of X1, X2, X3 and X4 in train.csv.

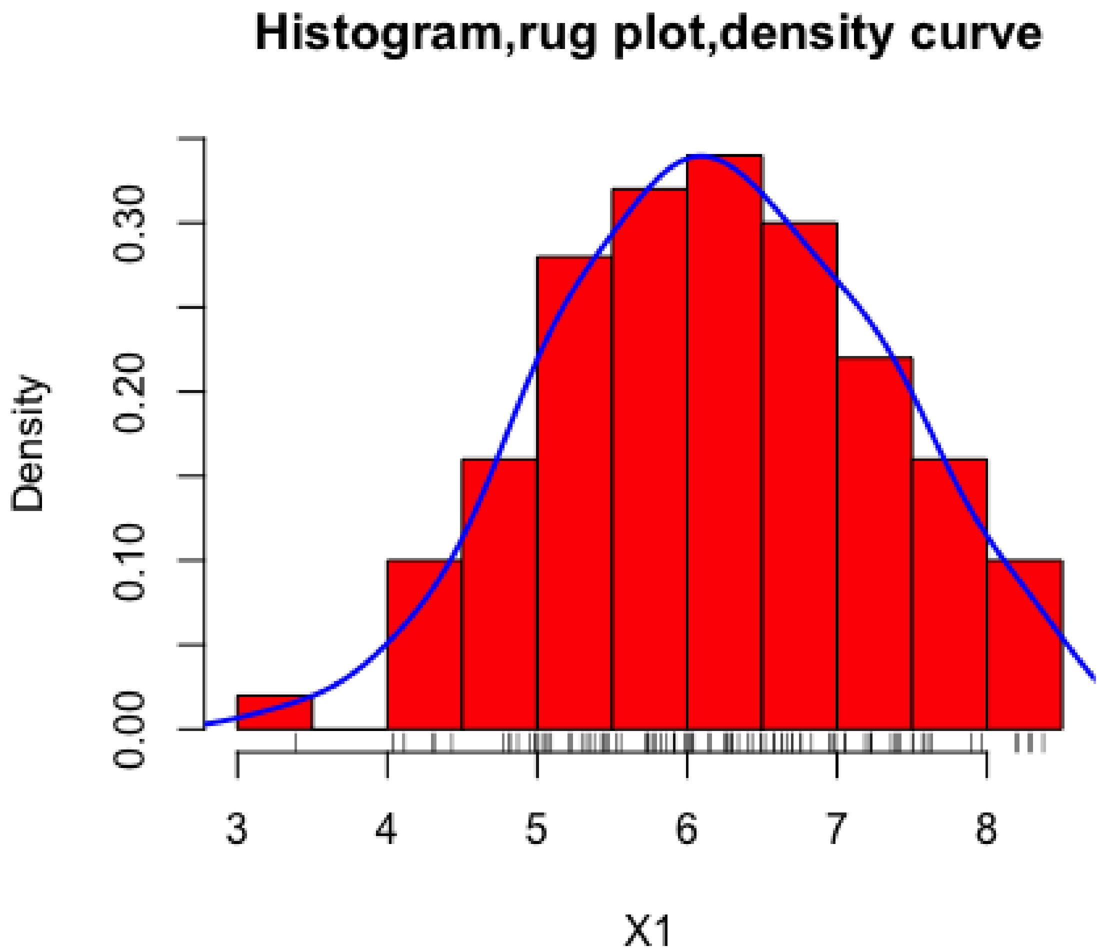


Figure1. Histogram, Rug Plot and Density Curve of X1

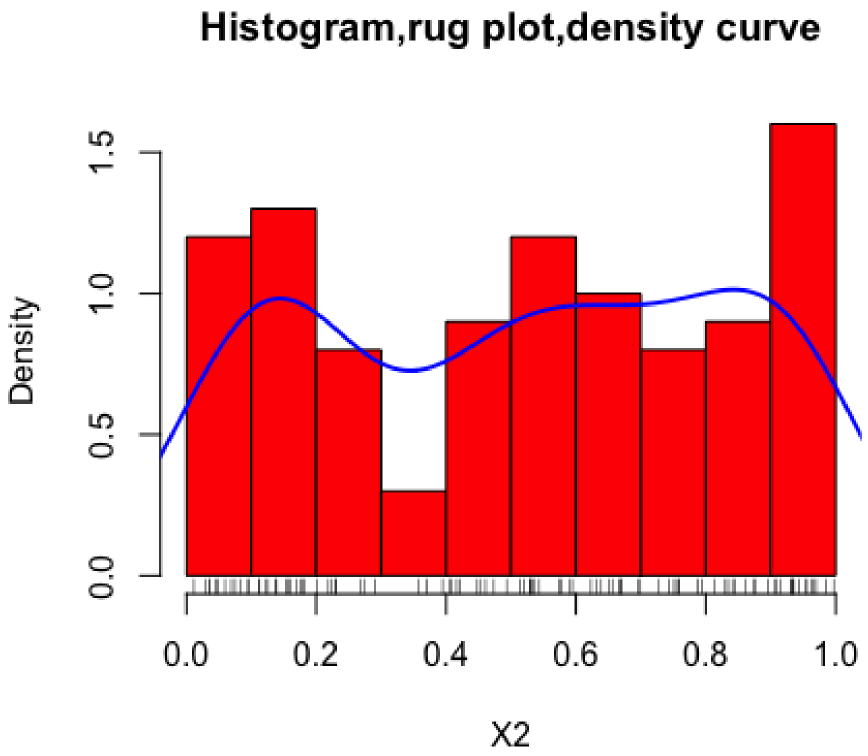


Figure2. Histogram, Rug Plot and Density Curve of X2

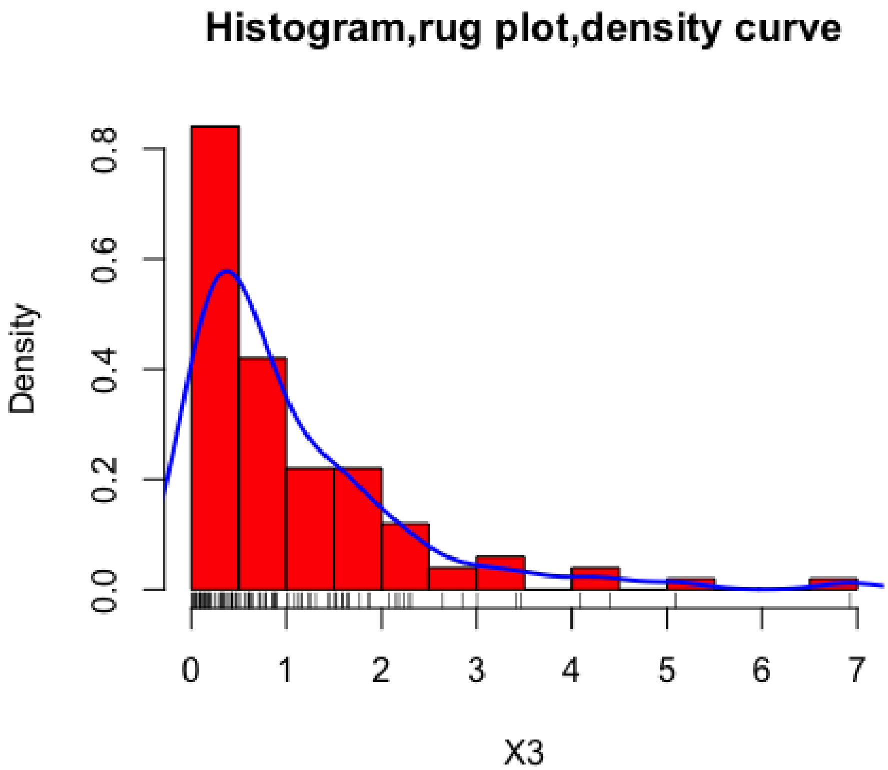


Figure3. Histogram, Rug Plot and Density Curve of X3

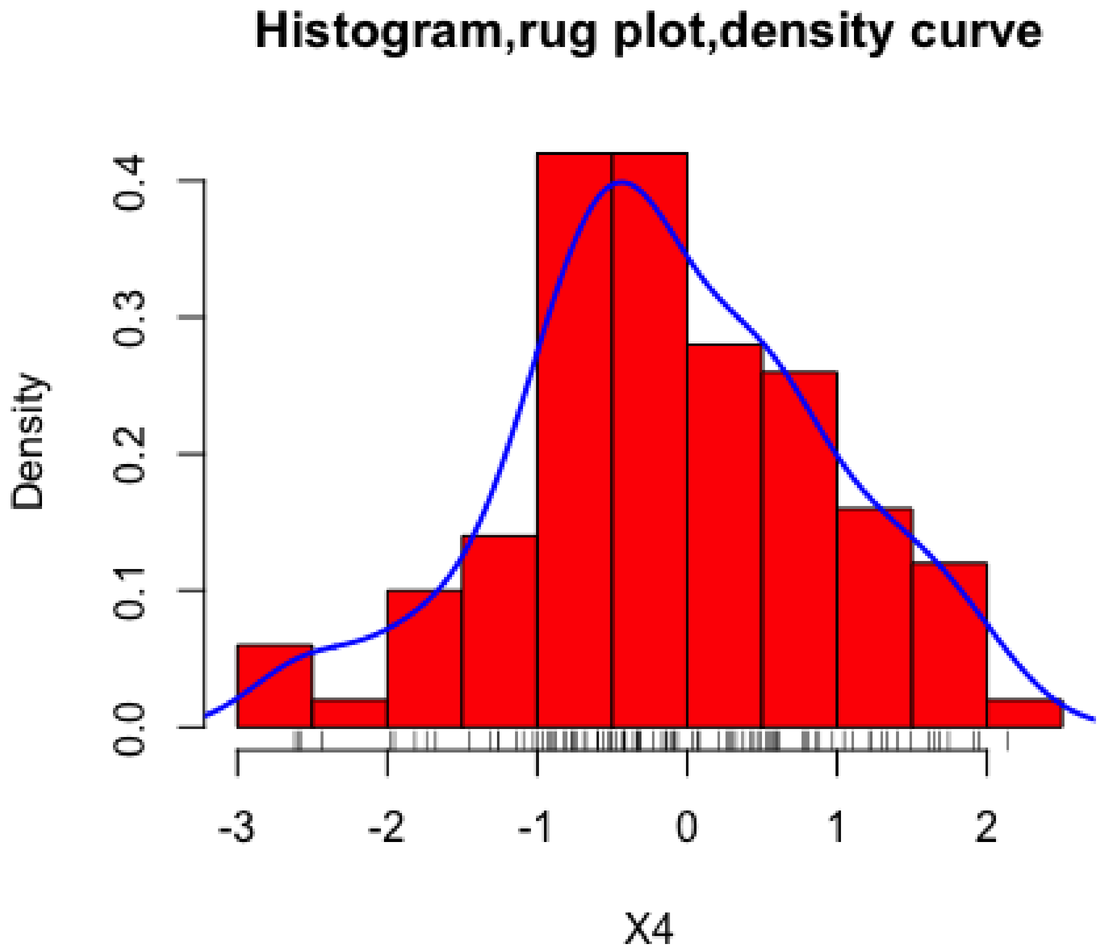


Figure4. Histogram, Rug Plot and Density Curve of X4

It is apparent from the chart that the data in figure 1, 3 and 4 are quite unstable and are undergoing major fluctuation. In contrast, the data in figure 2 fluctuate in a relatively small range. Thus, variable X2 is most likely the sample drawn from normal distributions.

Task B.2

Model 1: Y~X1+X2+X3+X4

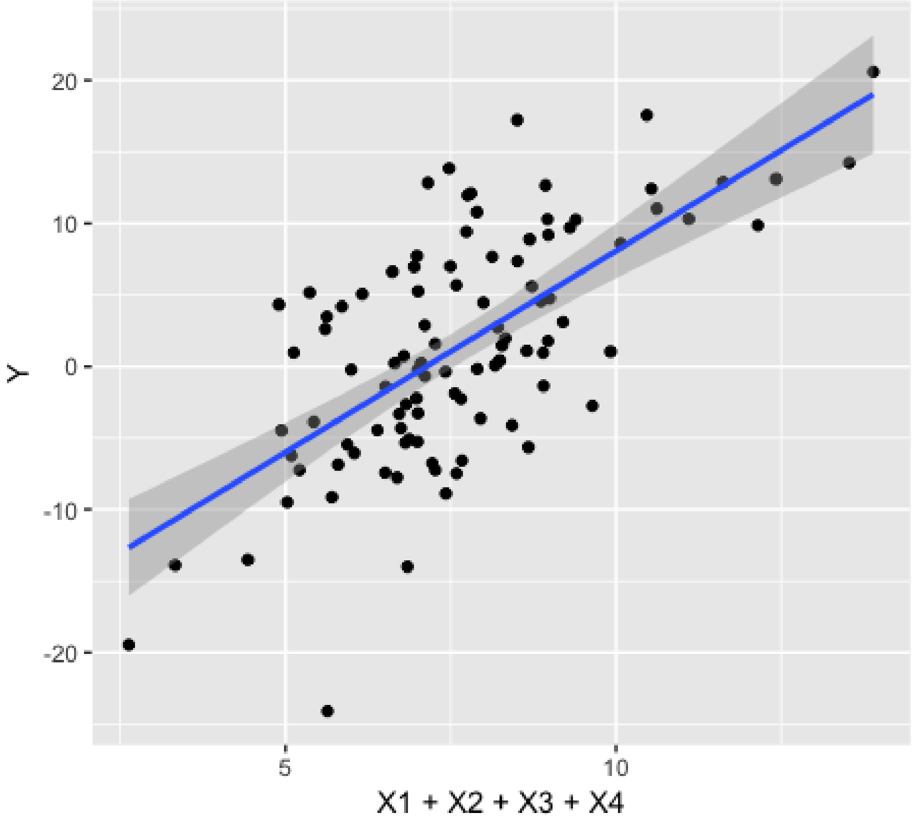


Figure5. Linear Regression of Model 1

```
Call:
lm(formula = Y ~ (X1 + X2 + X3 + X4), data = train)

Residuals:
    Min      1Q  Median      3Q     Max 
-4.5110 -1.3386 -0.0158  1.5315  4.7958 

Coefficients:
            Estimate Std. Error t value Pr(>|t|)    
(Intercept)   4.7394     1.3259   3.575 0.000554 ***
X1           -0.2850     0.1945  -1.465 0.146156    
X2           -5.5824     0.6609  -8.447 3.42e-13 ***
X3            2.1597     0.1760  12.273  < 2e-16 ***
X4            6.9379     0.1951  35.568  < 2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 2.037 on 95 degrees of freedom
Multiple R-squared:  0.9402,	Adjusted R-squared:  0.9376 
F-statistic: 373.1 on 4 and 95 DF,  p-value: < 2.2e-16
```

Figure6. Summary of Model 1 Using Data from Train.csv

Model 2: Y~X2+X3+X4

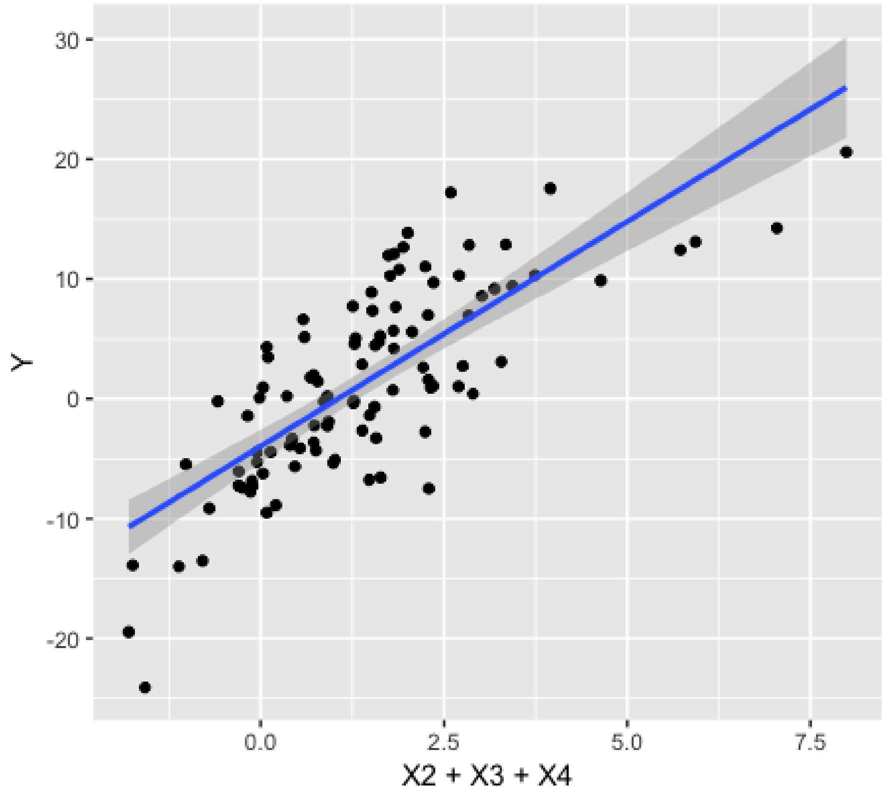


Figure7. Linear Regression of Model 2

```
Call:
lm(formula = Y ~ (X2 + X3 + X4), data = train)

Residuals:
    Min      1Q  Median      3Q     Max 
-4.7054 -1.4289 -0.0285  1.5845  4.6968 

Coefficients:
            Estimate Std. Error t value Pr(>|t|)    
(Intercept)   2.8978     0.4247   6.823 7.96e-10 ***
X2           -5.4905     0.6618  -8.296 6.70e-13 ***
X3            2.1826     0.1763  12.378  < 2e-16 ***
X4            6.9213     0.1959  35.333  < 2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 2.049 on 96 degrees of freedom
Multiple R-squared:  0.9388,	Adjusted R-squared:  0.9369 
F-statistic: 490.9 on 3 and 96 DF,  p-value: < 2.2e-16
```

Figure8. Summary of Model 2 Using Data from Train.csv

In this task, I drew two linear regression diagrams for two models first. Then I put the data of train.csv into two different formulas to see the summaries of each formula. Obvious from these figures is that the Multiple R-squared value of model 1 is higher than model 2. Multiple R-squared value is the square of the correlation coefficient between the actual value and the predicted value. The closer the value is to 1, the better the fitting of the model is.

Task B.3

```
> resultTrue
             Y
1   -4.4944460
2   -0.6746411
3    1.0679766
4   -0.7327577
5    1.4219150
6    7.6269002
7   -4.1776235
8    1.2716331
9   -2.2498374
10   2.6642682
11  -9.2557306
12  -6.7190949
13   6.6350402
14  -6.2589780
15  -3.4942142
16  -8.1242119
17  10.0274544
18  -3.2295611
19  -5.1422472
20  -1.3302640
```

Figure9. Actual Result in Test.csv

```
> resultModel1
        1         2         3         4         5 
-5.247517  2.332055  1.647435 -1.239928  1.088981 
        6         7         8         9        10 
 6.588221 -2.971900  2.277291 -1.973435  6.887364 
       11        12        13        14        15 
-8.097834 -7.101084  3.925810 -6.169646 -4.685280 
       16        17        18        19        20 
-6.512826 10.336682 -2.126811 -3.900154  1.823841 
```

Figure10. Predicted Result Using Model 1

```
> resultModel2
        1         2         3         4         5 
-4.912605  2.277259  1.912049 -1.256946  0.990325 
        6         7         8         9        10 
 5.930054 -3.014530  2.254797 -1.992130  6.637339 
       11        12        13        14        15 
-7.880344 -7.042298  3.940002 -5.974210 -4.410152 
       16        17        18        19        20 
-6.736941  9.995940 -1.926359 -4.172230  1.651235 
```

Figure11. Predicted Result Using Model 2

```
Call:
lm(formula = model1, data = test)

Residuals:
    Min      1Q  Median      3Q     Max 
-3.0707 -0.9109  0.1641  0.9078  2.6194 

Coefficients:
            Estimate Std. Error t value Pr(>|t|)    
(Intercept)   2.3323     2.9703   0.785  0.44455    
X1            0.3322     0.5046   0.658  0.52024    
X2           -7.1553     1.2249  -5.842 3.24e-05 ***
X3            1.3408     0.4356   3.078  0.00765 ** 
X4            7.4889     0.6756  11.086 1.27e-08 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 1.501 on 15 degrees of freedom
Multiple R-squared:  0.9341,	Adjusted R-squared:  0.9166 
F-statistic: 53.19 on 4 and 15 DF,  p-value: 1.104e-08
```

Figure12. Summary of Model 1 Using Data from Test.csv

```
Call:
lm(formula = model2, data = test)

Residuals:
    Min      1Q  Median      3Q     Max 
-3.1408 -1.0496  0.2258  0.8251  2.8045 

Coefficients:
            Estimate Std. Error t value Pr(>|t|)    
(Intercept)   4.1914     0.9057   4.628 0.000279 ***
X2           -6.8646     1.1221  -6.118 1.48e-05 ***
X3            1.2655     0.4128   3.066 0.007392 ** 
X4            7.3217     0.6148  11.909 2.30e-09 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 1.475 on 16 degrees of freedom
Multiple R-squared:  0.9322,	Adjusted R-squared:  0.9195 
F-statistic: 73.38 on 3 and 16 DF,  p-value: 1.438e-09
```

Figure13. Summary of Model 2 Using Data from Test.csv

In this task, I got the predicted values by two different formulas. We can see the difference by comparing the actual and the predicted value. Then I put the data of test.csv into two different formulas to see the summaries of each formula. We can then calculate the Mean Squared Error(MSE) of each model.

MSE of model 1 = 1.501^2 = 2.25
MSE of model 2 = 1.475^2 = 2.18

It is clear from the result that the model 2 has smaller MSE. It reflects the degree of deviation from the predicted data to the true value. The smaller the value, the higher the accuracy of the prediction. Thus, model 2 is better. We know from the task B.2 that model 1 has higher R square. However, there are many other criteria for evaluating the model. The high R square does not indicate that the model is always suitable.