Regression Analysis Of Population Drinking Dataset

Gaurav Tomar

# Contents

1. [Introduction 2](#_TOC_250004)
2. [Exploratory Data Analysis 2](#_TOC_250003)
3. [Regression Analysis 6](#_TOC_250002)
   1. [Fitting a Linear Model . . . . . . . . . . . . . . . . . . . . . . . . 6](#_TOC_250001)
   2. Checking Model Assumptions . . . . . . . . . . . . . . . . . . . . 7
   3. Detecting Influential Points 12
   4. Remedies For Influential Points 16
   5. Collinearity 18
   6. Remedies For Collinearity 20
   7. Model Selection 21
   8. Shrinkage Methods 31
   9. [Robust Regression Methods 33](#_TOC_250000)

# Introduction

## About The Data

Here we have 46 observations from different places of the following quantiles :-

Urban Population Late Births

Wine Consumption Per Capita Liquor Consumption Per Capita Cirrhosis Death Rate

Here our response variable is Cirrhosis Death Rate and others are all covariates. To get an overview of the data, we first perform the exploratory data analysis.

# Exploratory Data Analysis

## Loading the Dataset

To get an overview of the data, we first load it in R and print first few values:-

**library**(MASS) **library**(lattice) **library**(olsrr) **library**(car) **library**(L1pack)

X = **read.csv**(file = "D:\\PG files\\Projects\\Regression-Analysis-Project-main\\population\_drinking1.txt", header=TRUE,sep = "\t")

**names**(X) <- **c**("Ind","Ind\_1","Urban population", "Late births", "Wine consumption per capita", "Liquor consumption per cap

**head**(X)

Ind Ind\_1 Urban population Late births Wine consumption per capita

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 1 | 1 1 | 44 | 33.2 | | 5 |
| 2 | 2 1 | 43 | 33.8 | | 4 |
| 3 | 3 1 | 48 | 40.6 | | 3 |
| 4 | 4 1 | 52 | 39.2 | | 7 |
| 5 | 5 1 | 71 | 45.5 | | 11 |
| 6 | 6 1  Liquor consumptio | 44 37.5 9  n per capita Cirrhosis death rate | | | |
| 1 | 30 | | | 41.2 | |
| 2 | 41 | | | 31.7 | |
| 3 | 38 | | | 39.4 | |
| 4 | 48 | | | 57.5 | |
| 5 | 53 | | | 74.8 | |
| 6 | 65 | | | 59.8 | |

Here we have “Cirrhosis death rate” as the response and “Urban population”, “Late births”, “Wine consumption per capita”, “Liquor consumption per capita” as the covariates.

## Type of the covariates

To know type of each covariates, we use the str() function :-

**str**(X)

*'*data.frame*'*: 46 obs. of 7 variables:

$ Ind : int 1 2 3 4 5 6 7 8 9 10 ...

$ Ind\_1 : int 1 1 1 1 1 1 1 1 1 1 ...

$ Urban population : int 44 43 48 52 71 44 57 34 70 54 ...

$ Late births : num 33.2 33.8 40.6 39.2 45.5 37.5 44.2 31.9 45.6 45.9 ...

$ Wine consumption per capita : int 5 4 3 7 11 9 6 3 12 7 ...

$ Liquor consumption per capita: int 30 41 38 48 53 65 73 32 56 57 ...

$ Cirrhosis death rate : num 41.2 31.7 39.4 57.5 74.8 59.8 54.3 47.9 77.2 56.6 ...

So the dataset contains no factor covariate hence we can perform multiple linear regression here. For ease of indexing, we name the columns as “I”, “1”, “A1”, “A2”, “A3”, “A4”, “Y”.

**names**(X) <- **c**("I","1","A1","A2","A3","A4","Y")

## 5-number Summary of Covariates

To get an idea of the values of each covariate we calculate the 5-number summary for each of them :-

**summary**(X[,-**c**(1,2)])

Min. : 28.00

1st Qu.: 48.90

Median : 57.65

Mean : 63.49

3rd Qu.: 75.70

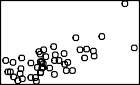
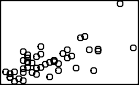
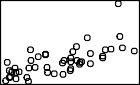
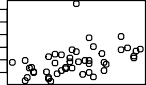
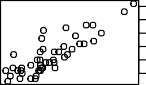
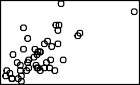
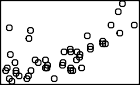
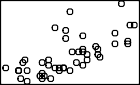
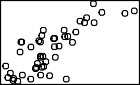
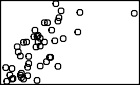
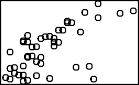
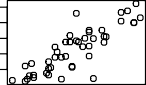
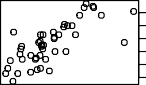
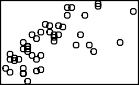
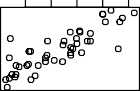
Max. :129.90

|  |  |  |  |
| --- | --- | --- | --- |
| A1 | A2 | A3 | A4 |
| Min. :27.00 | Min. :31.20 | Min. : 2.00 | Min. : 26.00 |
| 1st Qu.:44.25 | 1st Qu.:35.62 | 1st Qu.: 6.25 | 1st Qu.: 41.50 |
| Median :55.00 | Median :42.25 | Median :10.00 | Median : 56.00 |
| Mean :56.26 | Mean :41.48 | Mean :11.59 | Mean : 57.50 |
| 3rd Qu.:65.00 | 3rd Qu.:45.83 | 3rd Qu.:15.75 | 3rd Qu.: 68.75 |
| Max. :87.00 | Max. :56.10 | Max. :31.00 | Max. :149.00 |
| Y |  |  |  |

## Pairwise Scatterplots

To get an idea of the relationship between covariates and response, we make pairwise scatterplots using pairs() function :-

**pairs**(X[,-**c**(1,2)])

35 40 45 50 55 40 80 120

A1

A2

A3

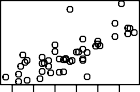
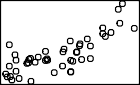
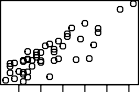
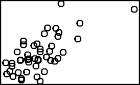
A4

35

45

55

30 50 70

Y

30 50 70

40 80

140

40 80 120

5 15 25

5 10 20 30

40 60 80 120

This plot clearly indicates linear relationship between the covariates and re- sponse also. This might lead to the problem of multicollinearity which we will formally diagnose.

## Correlation Between Covariates

We calculate the correlation between the covariates and response to get even bet- ter idea of linear dependence between them :-

|  |  |  |  |
| --- | --- | --- | --- |
| A1 | A2 | A3 | A4 |
| A1 1.0000000 | 0.8432812 | 0.6786230 | 0.4402957 |
| A2 0.8432812 | 1.0000000 | 0.6398407 | 0.6863643 |
| A3 0.6786230 | 0.6398407 | 1.0000000 | 0.6759206 |
| A4 0.4402957 | 0.6863643 | 0.6759206 | 1.0000000 |

We can see that the correlations are high between many predictors which can lead to problem of multicollinearity.

## Partial Residual Plots

To get an idea of the nature of relationship between the covariates and response, we make the partial residual plot for all the covariates :-

**crPlots**(**lm**(Y~A1+A2+A3+A4,data = X))

0

50

Component + Residual Plots

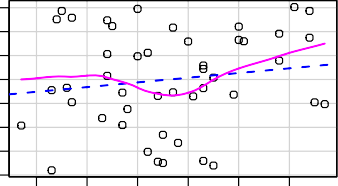
Component+Residual(Y)

−20 −10 0 5

15

Component+Residual(Y)

−30 −10 10



30 40 50 60 70 80 35 40 45 50 55

A1 A2

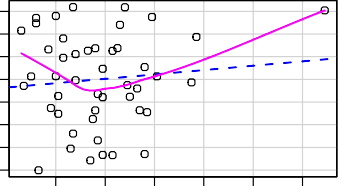
Component+Residual(Y)

−20 0 20 40

Component+Residual(Y)

−20 −10 0 5

15



5 10 15 20 25 30 40 60 80 100 120 140

A3 A4

## Conclusion

The plot indicates the linear relationship between the covariates and response.

Hence, we will fit the usual multiple linear regression model with no higher order polynomial terms.

Later we will see if other models with interaction terms are better or not.

## Boxplots of Covariates and Response

We draw the boxplots for differnet covariates to get idea of presence of outlier / high leavarage points :-

**boxplot**(X[,-**c**(1,2)])

100

150

A1 A2 A3 A4 Y

Here also we get some indication of possible presence of those points.

# Regression Analysis

## Fitting a Linear Model

**Fitting a Linear Model to the Dataset**

We fit a linear model of the form :-

***Y*** *n*×1 = ***X****n*×*p****β****p*×1 + ***ϵ****n*×1

where *n* = 46 (total number of observed responses) and *p* = 5 where columns of ***X*** = **1***n* ***x***1 ***x***2 ***x***3 ***x***4 and ***β*** = *β*0 *β*4 each corresponding to the 4 different covariates.

è é 1 · · · 2

We fit a linear model based on the given dataset in R and then verify the different assumptions of it.

## Features of the fitted model

We fit the linear model specified before in the dataset using lm() function and to get an idea about the estimates we use the summary() function :-

**colnames**(X)=**c**("I","1","A1","A2","A3","A4","Y")

**attach**(X)

reg <- **lm**(Y~A1+A2+A3+A4)

**summary**(reg)

Call:

lm(formula = Y ~ A1 + A2 + A3 + A4)

Residuals:

Min 1Q

Median 0.1507

3Q

Max

-18.8723 -6.7803

7.3252 16.4419

Coefficients:

Estimate Std. Error t value Pr(>|t|) (Intercept) -13.96310 11.40035 -1.225 0.2276

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

Residual standard error: 10.61 on 41 degrees of freedom Multiple R-squared: 0.8136,Adjusted R-squared: 0.7954 F-statistic: 44.75 on 4 and 41 DF, p-value: 1.951e-14

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| A1 | 0.09829 | 0.24407 | 0.403 | 0.6893 |
| A2 | 1.14838 | 0.58300 | 1.970 | 0.0556 . |
| A3 | 1.85786 | 0.40096 | 4.634 | 3.61e-05 \*\*\* |
| A4 | 0.04817 | 0.13336 | 0.361 | 0.7198 |
| --- |  |  |  |  |

## Explanation of the fitted model

As we can see only the coefficients *β*2*, β*3 for covariates “A2”,“A3” are statistically significant.

This does not imply that other covariates are insignificant since there maybe many problems that are hidden in the model.

So, before concluding anything we verify all the assumptions of a linear model.

## Obs vs Fitted Values

40

60

80

We plot the observed vs fitted values to get some idea about prediction :-

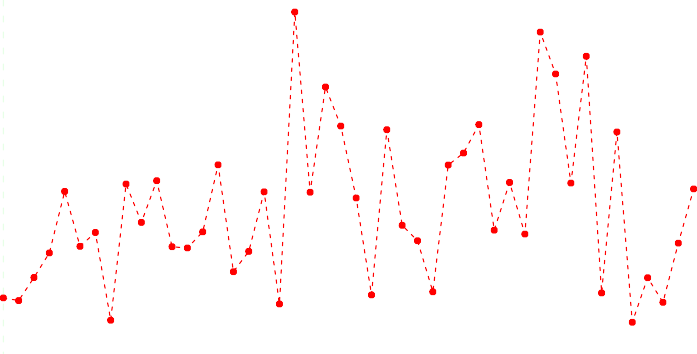
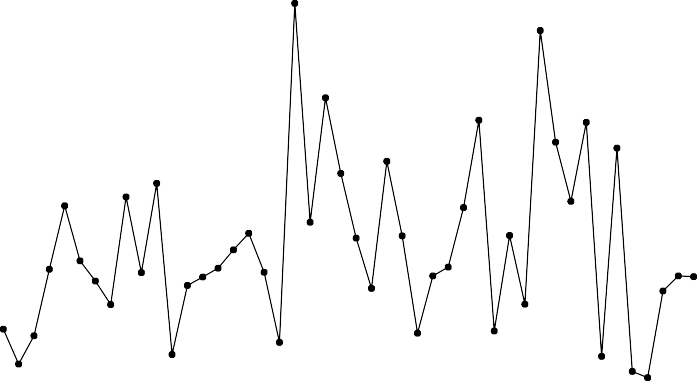
**plot**(1:**nrow**(X),X$Y,type = "o",pch = 20,ylab = "observed & predicted values",xlab = "Index")

**lines**(1:**nrow**(X),reg$fitted.values,type = "o",pch = 20,col = "red",lty = 2) **abline**(v = 1:**nrow**(X),lty = 2,col = **rgb**(0,1,0,alpha = 0.3)) **legend**("topleft",legend = **c**("Obs","Pred"),fill = **c**("black","red"))

observed & predicted values

100

120



Obs Pred

0 10 20 30 40

Index

The fit is good except a few observations.

There may be many reasons for this which we will eventually look into.

## Checking Model Assumptions QQ-plot of residuals

We now plot the sorted residuals (quantiles) against the population quantiles of a normal distribution :-

resi<-**residuals**(reg) **qqnorm**(resi,ylab="Residuals",main="") **qqline**(resi)

5

10

15

−2 −1 0 1 2

Residuals

−20 −15 −10 −5 0

Theoretical Quantiles

We can see the qq-plot indicates ligh tailed residuals with possible deviation from normality.

There maybe some outlier points present which we will verify later.

## Shapiro-Wilk Test

We test the following hypothesis *H*0 : residuals are normally distributed against

*H*1 : *H*0 is false using Shapiro-Wilk test in R as :-

**shapiro.test**(resi)

Shapiro-Wilk normality test

data: resi

W = 0.95987, p-value = 0.1133

Though the p-value is more than 0*.*1 but this doesn’t give strong evidence in favour of *H*0 so we will further check for presence of correlation between the errors and other issues also.

## Checking Homoskedasticity Assumptions

(*y*) plots :-

First to check homoskedasticity assumption, we make the residuals (*ε*) vs fitted

**plot**(**fitted**(reg),**residuals**(reg),xlab="Fitted",ylab="Residuals") **abline**(h=0)

^

^

0

100

200

5

10

15

40 60 80 100 120

Residuals

−20 −15 −10 −5 0

Fitted

We can see the plot doesn’t give indication of presence of heteroskedasticity, hence we will perform confirmatory tests.

https://online.stat.psu.edu/stat462/node/117/#:~:text=When%20conducting%20a%20residual%

*bi* **vs** *y*^*i* **plot**

A standard technique to detect presence of heteroskedasticity is to plot the quan-

tities *b* = *e*2 against the fitted values *y* .

*i* 1 *i* ^*i*

We mak−e *h*t*i*he plot using R :-

A = **as.matrix**(X[,-1])

H = A%\*%**solve**(**t**(A)%\*%A)%\*%**t**(A) H\_i = **diag**(H)

e\_i = **residuals**(reg) b\_i = e\_i^2/(1-H\_i)

**plot**(**fitted**(reg),b\_i,pch = 20,main = **bquote**("Plot of" ~ b[i]^2 ~ "vs fitted values"

Plot of b2 vs fitted values

300

400

500

i

40 60 80 100 120

This plot gives no indication of any heteroskedasticity present in the residuals.

## Breusch-Pagan Test

We perform the Breusch–Pagan test for testing the homoskedasticity assumptions using R :-

**library**(lmtest) **bptest**(reg)

studentized Breusch-Pagan test

data: reg

BP = 2.6929, df = 4, p-value = 0.6105

We can see that the p-value of the outcome is satisfactorily high so we can safely assume the error variances to be equal.

## ACF plot

If the errors in the model are truely independent, then we will expect the sample autocorrelation coefficients for different lags *k* to be insignificant.

**acf**(resi,ylab = "",xlab = "",main = "")

**title**(xlab="Lag", ylab="ACF", line=2)

0 5 10 15

Lag

ACF

0.2 0.4 0.6 0.8 1.0

−0.2

This plot clearly gives indication of no presence of any type of correlation between the residuals.

## PACF plot

Similarly, we plot the sample partial autocorrelation coefficients for different lags and got the same kind of observations indicating no presence of correlations.

0.1

0.2

0.3

**pacf**(resi,ylab = "",xlab = "",main = "")

**title**(xlab="Lag", ylab="PACF", line=2)

5 10 15

Lag

PACF

−0.3 −0.2 −0.1 0.0

## Durbin-Watson Test

To test the null hypothesis *H*0 : errors are uncorrelated against *H*1 : errors are correlated, we perform Durbin-Watson test which gives the following results :-

**require**(lmtest) **dwtest**(Y~A1+A2+A3+A4,data=X)

Durbin-Watson test

data: Y ~ A1 + A2 + A3 + A4 DW = 2.5494, p-value = 0.9734

alternative hypothesis: true autocorrelation is greater than 0

Since the test gives high p-value we can accept *H*0 hence the assumption of uncorrelated residuals can be assumed to be satishfied.

## Breusch–Godfrey test

To check whether residuals are uncorrelated for higher orders, we perform the Breusch–Godfrey test upto order 20.

**require**(lmtest) **bgtest**(reg,order = 20)

Breusch-Godfrey test for serial correlation of order up to 20

data: reg

LM test = 24.951, df = 20, p-value = 0.2033

Here also the p-value is fairly high favouring the null assumption.

## Detecting Influential Points Hat Matrix Diagonals

stud

−3 −2 −1 0

1

2

3

To detect high leverage points, we compute the hat matrix diagonals *hi* of the matrix ***H*** = ***X*** 1***X****T* ***X***2−1 ***X****T* :-

*n*

hat\_d <- **hatvalues**(reg) **head**(**sort**(hat\_d,dec=TRUE))

36

20

12

38

30

39

0.4994692 0.3042241 0.3015417 0.2998329 0.2078749 0.1721364

We find out if there is any diagonal element with value *>* 2*p* as they should

be looked at more closely.

n = **nrow**(X);p = 5 hat\_d[hat\_d > 2\*(p/n)]

12

20

36

38

0.3015417 0.3042241 0.4994692 0.2998329

Hence we will apply other procedures also to confirm whether these points are influential or not.

## Externally Studentized Residuals

We plot the externally studentized residuals using the formula *t*2 = *r*2 3  *n*−*p*−1 4

:-

*i*

*i*

*n*−*p*−*r*2

*i*

stud <- **rstudent**(reg)

**plot**(stud,ylim = **c**(-3,3),pch=20,col = "blue")

**abline**(h=**c**(0))

0 10 20 30 40

Index

If the assumptions are correct i.e. ***ϵ*** ∼ *Nn* (**0***, σ*2***I***) then we should get that

*ti* ∼ *tn*−*p*−1.

Hence the significant externally studentized residuals will have values|*ti*| *>*

*tn*−*p*−1; *α*

2

*i*

⇐⇒ *t*2 *> Fn*−*p*−1;*α* :-

**ols\_plot\_resid\_stud\_fit**(reg)

Deleted Studentized Residual vs Predicted Values

2

*Threshold: abs(2)*

*12*

1

Deleted Studentized Residual

Observation

0

normal

outlier

−1

−2

−3

50 75 100

Predicted Value

We can see from the plot that one residual is significant hence we treat that as an outlier.

## DFBETAS

After outliers , we check for presence of high leavarage points, which can be de-

(*i*)*j*

*S*(*i*)

*i*

tected using DFBETAS measure for different parameters *DFBETAS* =

Σ *c*

*ij*

2

*j*+1*,i*

*β*^ñ*j* −*β*^

where ***C*** = ((*cij*)) = 1***X****T* ***X***2−1 ***X*** .

2

We will consider the points for which *DFBETASij >* √ . In the next slide we plot the values for all the 5 coefficients *βi, i* = 0*, ...,* 4.

| | *n*

**par**(mfrow = **c**(2,3)) DFBETAS = **dfbetas**(reg) **for**(i **in** 1:5)

{

**plot**(DFBETAS[,i],main=**bquote**("DFBETAS for" ~ beta[**.**(i-1)]),ylab="",ylim=**c**(-1.5,1.5),xlab="",pch=

**abline**(h=**c**(-2/**sqrt**(n),2/**sqrt**(n)))

ind = **which**(**abs**(DFBETAS[,i]) > 2/**sqrt**(n)) *# beta\_0*

text = **text**(ind,DFBETAS[ind,i],pos = 3,labels = ind)

}

DFBETAS for 0

0.5

1.5

0.5

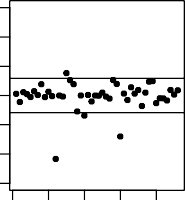
1.5

0.5

1.5

DFBETAS for 1

DFBETAS for 2

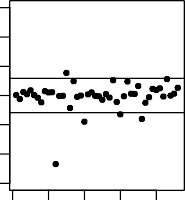


15

20

30

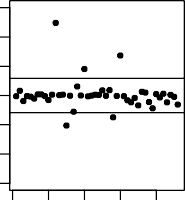
12



15

20 30 36

12



12

30

20

15

28

0 10 20 30 40

−1.5

−0.5

−1.5

−0.5

−1.5

−0.5

0 10 20 30 40

0 10 20 30 40

DFBETAS for 3 DFBETAS for 4

−1.5

−0.5

0.5

1.5

−1.5

−0.5

0.5

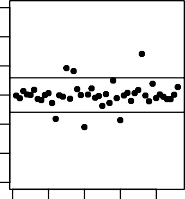
1.5



20

38

1178



36

1157

12 20 30

0 10 20 30 40 0 10 20 30 40

## DFFITS

To notice the change in fitted values, we plot the DFFITS values for all the

1 2

points where *DFFITSi*

|*DFFITSi*| *>* 2ñ *p* .

*n*

= *ti*

*h* 1*/*2 and we will check for the points for which

1−*hi*

*i*

DFFITS

**ols\_plot\_dffits**(reg)

Influence Diagnostics for Y

*Th3r6eshold: 0.66*

1

*20*

0

−1

*30*

*12*

0 10 20 30 40

Observation

## COVRATIO

We also plot the COVRATIO values which are defined as

*COV RAT IOi* =

*n* − *p* − 1 +

*n* − *p*

A

2

*i*

*t*

*n* − *p*

B−*p*

(1 − *hi*)−1

3*p* and we consider the points to have high fluence for which |*COV RAT IO* − 1| *>*

COVRATIO  1

0.0

0.2

0.4

0.6

.

*n*

COVRATIO = **covratio**(reg)

**plot**(**abs**(COVRATIO-1),ylab=**expression**(**abs**(COVRATIO-1)),pch = 20)

**abline**(h = 3\*p/n)

ind = **which**(**abs**(COVRATIO-1) >= 3\*p/n)

**text**(ind,**abs**(COVRATIO[ind]-1),pos = 1,labels = ind)

36

38

24

0 10 20 30 40

Index

## Cook’s *D*

( ( )−*β*)*T* ***XT X***(*β*(*i*)−*β*)

Lastly, we calculate the Cook’s Distance *D* = *β*^ *i* ^ ^ ^ = *r*2 *hi* for

all the

*n* points.

*i*

*pS*2

*i p*(1−*hi*)

We flag the points as suspicious for which *Di >*  4 here *n* = 46*, p* = 5. Whose

*n*

value equals to 0*.*087.

We plot the values and see if such suspicious points exists or not.

COOKSD = **cooks.distance**(reg) **plot**(COOKSD,pch=20)

**abline**(h = 3\***mean**(COOKSD))

ind = **which**(COOKSD > 3\***mean**(COOKSD)) *# beta\_0*

text = **text**(ind,COOKSD[ind],pos = 1,labels = ind)

0.3

0.4

36

12

20

30

0 10 20 30 40

COOKSD

0.0

0.1

0.2

Index

We can clearly notice that the points 12,20,30,36 have significant values of *Di*.

So we will investigate them further.

## Conclusion

From all the diagnostics performed for finding influential observations, we can make the following table of our findings :-

|  |  |
| --- | --- |
| Diagnostic Measures | Points Detected |
| *hi* | 12*,* 20*,* 36*,* 38 |
| *ti* | 12 |
| *DFBETAS* | 12*,* 15*,* 17*,* 18*,* 20*,* 28*,* 30*,* 36 |
| *DFFITS* | 12*,* 20*,* 30*,* 36 |
| *COV RAT IO* | 24*,* 36*,* 38 |
| *Cook*′*s D* | 12*,* 20*,* 30*,* 36 |

Hence, from the table, we conclude the points 12*,* 20*,* 30*,* 36 to be influential points and we we will later remove them from the model and see the changes occuring in all aspects of the fitted linear models.

## Remedies For Influential Points Removing Influential Points

We remove the influential points and then again fit a linear model with all the covariates and write the summary output of the fitted model here :-

**summary**(**lm**(Y~A1+A2+A3+A4,data=X[-**c**(12,20,30,36),]))

Call:

lm(formula = Y ~ A1 + A2 + A3 + A4, data = X[-c(12, 20, 30, 36),

])

Residuals:

Min

1Q Median

3Q

Max

-16.6468 -5.0683 -0.3998 6.1885 16.7016

Coefficients:

Estimate Std. Error t value Pr(>|t|) (Intercept) 20.2021 11.7358 1.721 0.09353 .

A1 0.8783 0.2582 3.401 0.00162 \*\*

A2 -0.7104 0.6205 -1.145 0.25960

A3 1.2890 0.3960 3.255 0.00243 \*\*

A4 0.1489 0.1312 1.134 0.26392

---

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

Residual standard error: 8.672 on 37 degrees of freedom Multiple R-squared: 0.8236,Adjusted R-squared: 0.8045 F-statistic: 43.18 on 4 and 37 DF, p-value: 1.867e-13

This model has increased value of *R*2 = 0*.*8045.

*adj*

## Improvements in the fitted model

From the output,we can see that the *R*2 value has increased from that of the full

*adj*

model which was = 0*.*7954.

Also we can see that the model indicates the estimates of the intercept term

1 ^ 2 1 ^ ^ 2

*β*0 , variables A1 & A3 *β*1*, β*3 to be signifcant.

Whether covariate A2 is signifcant or not, will be verified later.

## Histogram of Residuals

We also plot the histogram of the residual values and can notice it’s symmetric about 0 and seems to be normally distributed :-

reg\_out = **lm**(Y~A1+A2+A3+A4,data = X[-**c**(12,20,30,36),])

**hist**(**rstandard**(reg\_out),breaks = 5,main = "Histogram Of Residual Values")

**Histogram Of Residual Values**

Frequency

0 2 4 6 8 10

−3 −2 −1 0 1 2 3

rstandard(reg\_out)

## QQ-plot

−2

For checking the assumptions for the residuals we again make the quantile-quantile plot of the standardized residuals :-

reg\_out = **lm**(Y~A1+A3+A4,data = X[-**c**(12,20,30,36),]) **qqnorm**(**rstandard**(reg\_out)) **qqline**(**rstandard**(reg\_out))

**Normal Q−Q Plot**

Sample Quantiles

−1

0

1

2

−2 −1 0 1 2

Theoretical Quantiles

## Checking Other Assumptions

We again perform both the Shapiro-Wilk test and Durbin-Watson test for checking normality and presence of correlation between the residuals, respectively. We write down the observations in the following table :-

|  |  |  |
| --- | --- | --- |
| Tests | Model with  influential points | Model without  influential points |
| Shapiro-Wilk | 0.1133 | 0.8429 |
| Durbin-Watson | 0.9734 | 0.6943 |
| Breusch-Pagan | 0.6105 | 0.5742 |
| Breusch-  Godfrey | 0.2033 | 0.7973 |

Hence we can see considerable improvement in the normality assumptions of the residuals whereas the uncorrelated & homoskedasticity assumptions are more or less remains equally acceptable.

Hence, the model can be considered to be better than the previous model as a result of removing the influential points.

## Collinearity Multicollinearity

Next we consider the problem of multicollinearity that may be present in our dataset as suspected from the pairwise scatterplots.

We calculate the condition number for the scaled and centred model matrix

***X***∗ which is *κ* (***X***∗) = ñ

*λmax λmin*

We calculate *κ* (***X***∗) using R :-

where *λi*’s are the eigenvalues of ***X***∗*T* ***X***∗ = ***R****xx*.

X\_mdl = **model.matrix**(reg)[,-1] **kappa**(**scale**(X\_mdl))

[1] 8.624355

Hence, the square of condition number *κ*2 (***X***∗) 74*.*379 is an upper bound for the VIFs which is quite large !

≈

## VIF

Now, to determine whether some covariate with corresponding column ***x****j*, can be predicted accurately using other covariates or not, we compute the variance

inflation factors VIF*j* =

*j*

1

1−*R*2

*j*

where *R*2 is the coefficient of determination of the

regression of ***x***∗(*j*) on the columns of ***X***∗(*j*).

We calculate the VIF*j* values for *j* = 1*,* 2*,* 3*,* 4 in R :-

**require**("faraway")

**vif**(**lm**(Y~A1+A2+A3+A4,data = X[-**c**(12,20,30,36),]))*#high implies collinearity*

A1

A2

A3

A4

9.261059 9.360894 2.857795 2.751179

For the variables A1 and A2, we can see that the VIF values are greater than 5 and even close to 10! So we can interpret this as “the standard error of and

*β*2 would be √9*.*26 3*.*043 and √9*.*361 3*.*059 times more (respectively) than it

*β*1

 ≈ ≈

would have been without the presence of collinearity”.

## Effect of Influential Points on Collinearity

This is a very interesting observation that we have made in the dataset.

If we remove the influential points and then calculate the VIF values, we get

:-

**require**("faraway")

**vif**(**lm**(Y~A1+A2+A3+A4,data = X[-**c**(12,20,30,36),]))*#high implies collinearity*

A1

A2

A3

A4

9.261059 9.360894 2.857795 2.751179

But if we do the same without removing those points, we get :-

**require**("faraway")

**vif**(**lm**(Y~A1+A2+A3+A4,data = X[]))*#high implies collinearity*

A1

A2

A3

A4

5.910333 6.748416 3.080737 3.488172

So, as we can see the VIF values increased after removal of the influential points.

This is intuitive from the fact that actual linear dependence between the co- variates was being slightly nullified by the presence of such influential points.

## Demonstrating Effect of Collinearity

To demonstrate how collinearity can affect the estimates badly, we delibarately introduce some random noise in the response observations (***δ*** *N* (0*,* 1)) and then fit a linear model and see the changes in the estimate.

**set.seed**(2124)

lmod\_per = **lm**(Y+10\***rnorm**(**nrow**(X)-4,s = 5) ~ A1+A2+A3+A4,data = X[-**c**(12,20,30,36),]) reg$coefficients

∼

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| (Intercept)  -13.96310010 | A1 0.09828590 | A2 1.14837707 | A3 1.85786103 | A4 0.04817018 |
| lmod\_per$coefficients | | | | |
| (Intercept) | A1 | A2 | A3 | A4 |
| 1.304038 | 2.353402 | -2.501280 | -4.056390 | 1.437409 |

Hence, we can clearly see the how the estimates change a lot for introducing random noise in the response.

## Remedies For Collinearity Dealing with Collinearity

To deal with the collinearity present in the dataset, we first try to remove one of the correlated covariates “A1” or “A2” and see what improvements are observed in the variation inflation factors :-

|  |  |  |
| --- | --- | --- |
| Model | Condition Number (*κ*) | *R*2  *adj* |
| *Y* = *β*0 + *β*1A1 + *β*2A2 + *β*3A3 + *β*4A4 | 8*.*512 | 0*.*795 |
| *Y* = *β*0 + *β*2A2 + *β*3A3 + *β*4A4 | 5*.*625 | 0*.*755 |
| *Y* = *β*0 + *β*1A1 + *β*3A3 + *β*4A4 | 3*.*951 | 0*.*803 |

We also check for other assumptions between the models :-

|  |  |  |  |
| --- | --- | --- | --- |
| Tests | Full Model | Without “A1” | Without “A2” |
| Shapiro-Wilk | 0.1133 | 0.563 | 0.2461 |
| Durbin-Watson | 0.9734 | 0.7376 | 0.7638 |
| Breusch-Pagan | 0.6105 | 0.3742 | 0.9423 |

## Dealing with Collinearity

Hence, the model with covariates “A1”, “A3”, “A4” seems to be a much better model in terms of both prediction and accuracy of the estimates of ***β***. Also if we calculate the VIF values for the last model, we get them to be considerably small

:-

**vif**(reg\_out)

A1 A3

A4

2.360578 2.491333 1.677148

This is also relatable from the fact that initially covariate “A2” had the max- imum VIF value and the condition number also decreased significantly due to its removal.

## Added Variable Plot

For better understanding the contribution of a covariate in the regression model, we make a scatter plot of ***e***(*i*) = (***I P*** *i*) ***Y*** against (***I P*** *i*) ***x***(*i*) where ***e***(*i*) are the residuals of the model with variable Ai excluded and ***x***(*i*) is the column of observations of Ai. This is also called the added variable plot.

− −

**avPlots**(**lm**(Y~A1+A2+A3+A4,data = X[-**c**(12,20,30,36),]))

Added−Variable Plots

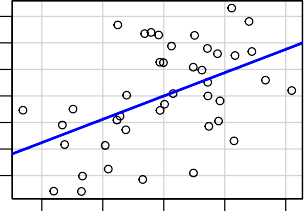
Y | others

−15 −5 5 15

Y | others

−15 −5 5

15

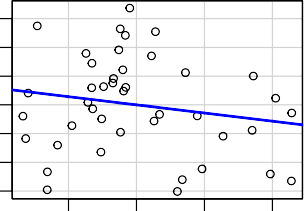


16

45

13

18



16

39

18

15

−10 −5 0 5 10 −2 0 2 4

A1 | others A2 | others

Y | others

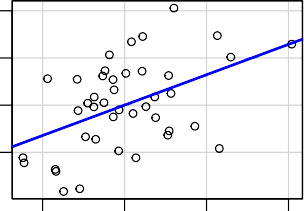
−10 0 10

20

Y | others

−15 −5 5

15

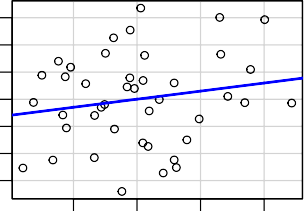


16

39

38

18



16

17

24

18

−5 0 5 10 −10 0 10 20

A3 | others A4 | others

## Conclusion

From the 4 plots, we can see that slopes of the fitted lines for the added variable plots of A1 & A3 are much more signifcant than other two plots for A2 & A4.

We can make some important conclusions from here.

This indicates that once predictor A1, A3, A4 is included, A2 can be excluded from the model for the high collinearity present between them.

Similar can be said for A4.

Now, for much better conclusions, we perform further model selection proce- dures based on several criterias.

## Model Selection Stepwise Selection

We perform the stepwise selection algorithm which performs a forward selection (FS) followed by a backward elimination (BE) using the AIC criterion and get to an optimum model.

We get the following sequence of models in the selection procedure :-

|  |  |
| --- | --- |
| Model | AIC Value |
| *Y* = *β*0 | 372.1803 |
| *Y* = *β*0 + *β*3A3 | 319.7608 |
| *Y* = *β*0 + *β*1A1 + *β*3A3 | 305.0896 |

We give the final model as an output we get in R :-

Stepwise Selection Method

Candidate Terms:

1 . A1

2 . A2

3 . A3

4 . A4

Step 0: AIC = 372.1803

Y ~ 1

Variables Entered/Removed:

Enter New Variables

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Variable | DF | AIC | Sum Sq | RSS | R-Sq | Adj. R-Sq | |
| A1 | 1 | 319.761 | 11455.113 | 4316.877 | 0.726 | 0.719 | |
| A3 | 1 | 324.405 | 10950.375 | 4821.615 | 0.694 | 0.687 | |
| A2 | 1 | 340.276 | 8736.385 | 7035.605 | 0.554 | 0.543 | |
| A4 | 1 | 355.432 | 5678.829 | 10093.161 | 0.360 | 0.344 | |
| - A1 added  Step 1 : AIC = 319.7608 Y ~ A1  Enter New Variables | | | | | | |  |
| Variable | DF | AIC | Sum Sq | RSS | R-Sq | Adj. R-Sq |  |
| A3 | 1 | 305.090 | 12869.415 | 2902.575 | 0.816 | 0.807 |  |
| A4 | 1 | 318.972 | 11732.403 | 4039.587 | 0.744 | 0.731 |  |
| A2 | 1 | 321.033 | 11529.316 | 4242.673 | 0.731 | 0.717 |  |
| - A3 added |  |  |  |  |  |  |  |
| Step 2 : AIC = Y ~ A1 + A3 | | 305.0896 | | |  |  |  |
|  | | Remove Existing Variables | | |  |  |  |
| Variable DF | | AIC Sum Sq RSS | | | R-Sq | Adj. R-Sq |  |
| A3 1 | | 319.761 11455.113 4316.877 | | | 0.726 | 0.719 |  |
| A1 1 | | 324.405 10950.375 4821.615 | | | 0.694 | 0.687 |  |
| Enter New Variables  Variable DF AIC Sum Sq RSS R-Sq Adj. R-Sq | | | | | | |  |
| A2 | 1 | 306.749 | 12892.877 2879.113 | | 0.817 | 0.803 |  |
| A4 | 1 | 306.775 | 12891.077 2880.913 | | 0.817 | 0.803 |  |
| No more variables to be added or removed. | | | | | | |  |

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Final Model Output  Model Summary | | | | | | | | | |  |  | | |
| R | | | 0.903 | | RMSE | | | 8.627 | |  |
| R-Squared | | | 0.816 | | Coef. Var | | | 14.066 | |  |
| Adj. R-Squared | | | 0.807 | | MSE | | | 74.425 | |  |
| Pred R-Squared | | | 0.788 | | MAE | | | 6.755 | |  |
| RMSE: Root Mean Square Error MSE: Mean Square Error  MAE: Mean Absolute Error  ANOVA  Sum of | | | | | | | | | |  |
|  | | | Squares | DF | | Mean Square | | F | | Sig. |
| Regression | | | 12869.415 | 2 | | 6434.707 | | 86.459 | | 0.0000 |
| Residual | | | 2902.575 | 39 | | 74.425 | |  | |  |
| Total | | | 15771.990 | 41 | |  | |  | |  |
| Parameter Estimates  model Beta Std. Error Std. Beta t Sig | | | | | | | | | | | lower upper | | |
| (Intercept) | | | 9.924 | 5.106 | | 1.944 | | | 0.059 | | -0.403 | 20.251 | |
| A1 | | | 0.640 | 0.126 | | 0.521 | | 5.078 | 0.000 | | 0.385 | 0.895 | |
| A3 | | | 1.516 | 0.348 | | 0.447 | | 4.359 | 0.000 | | 0.813 | 2.219 | |
| Stepwise Summary | | | | | | | | | | | | | |
|  | Variable | Method | | AIC | | RSS | Sum Sq | | R-Sq | | Adj. R-Sq | |  |
|  | A1 | addition | | 319.761 | | 4316.877 | 11455.113 | | 0.72629 | | 0.71945 | |  |
|  | A3 | addition | | 305.090 | | 2902.575 | 12869.415 | | 0.81597 | | 0.80653 | |  |
|  |  |  | |  | |  |  | |  | |  | |  |

Hence, this method gives the model containing covariates “A1”, “A3” as the optimum one.

## Best Subset Selection

We use different criterions for chosing optimal model among all the 15 possible linear models and plot the diagrams for all of them one by one.

X\_m = X[,-**c**(1,2)]

X1 <- X\_m[-**c**(12,20,30,36),]

**names**(X1) <- **c**("V1","V2","V3","V4","V5")

models<-**list**() models[["V1"]]<-**lm**(V5~V1,X1)

models[["V2"]]<-**lm**(V5~V2,X1)

models[["V3"]]<-**lm**(V5~V3,X1)

models[["V4"]]<-**lm**(V5~V4,X1)

models[["V12"]]<-**lm**(V5~V1+V2,X1)

models[["V13"]]<-**lm**(V5~V1+V3,X1)

models[["V14"]]<-**lm**(V5~V1+V4,X1)

models[["V23"]]<-**lm**(V5~V2+V3,X1)

models[["V24"]]<-**lm**(V5~V2+V4,X1)

models[["V34"]]<-**lm**(V5~V3+V4,X1) models[["V123"]]<-**lm**(V5~V1+V2+V3,X1)

models[["V124"]]<-**lm**(V5~V1+V2+V4,X1) models[["V134"]]<-**lm**(V5~V1+V3+V4,X1) models[["V234"]]<-**lm**(V5~V2+V3+V4,X1) models[["V1234"]]<-**lm**(V5~V1+V2+V3+V4,X1)

mnames<-**factor**(**names**(models),levels = **names**(models))

## Coefficient of determination (*R*2)

R2 <- **sapply**(models, **function**(fit) **summary**(fit)$r.squared)

**dotplot**(R2 ~ mnames, type = "o", pch = 16,auto.key=**list**(space="right"),xlab="Models",ylab=**expressi**

0.8

0.7

0.6

R2

0.5

0.4

V1 V2 V3 V4 V12 V13 V14 V23 V24 V34 V123 V124 V134 V234 V1234

Models

2 2

*R* **&** *R*

*adj*

adj.R2 <- **sapply**(models, **function**(fit) **summary**(fit)$adj.r.squared)

**dotplot**(R2 + adj.R2 ~ mnames, type = "o", pch = 16,auto.key=**list**(space="right"),xlab="Models")

0.8

0.7

0.6

R2 + adj.R2

R2

adj.R2

0.5

0.4

V1 V2 V3 V4 V12 V13 V14 V23 V24 V34 V123 V124 V134 V234V1234

Models

## Mallow’s *Cp*

100

80

60

40

Cp

20

0

V1 V2 V3 V4 V12 V13 V14 V23 V24 V34 V123 V124 V134 V234 V1234

## Akaike information criterion (AIC)

AIC <- **sapply**(models, **function**(fit) **AIC**(fit))

**dotplot**(AIC ~ mnames, type = "o", pch = 16,xlab="Models")

350

340

330

AIC

320

310

V1 V2 V3 V4 V12 V13 V14 V23 V24 V34 V123 V124 V134 V234 V1234

Models

## Bayesian information criterion (BIC)

BIC <- **sapply**(models, **function**(fit) **BIC**(fit))

**dotplot**(BIC ~ mnames, type = "o", pch = 16,xlab="Models")

360

350

340

BIC

330

320

310

V1 V2 V3 V4 V12 V13 V14 V23 V24 V34 V123 V124 V134 V234 V1234

Models

## Leave-One-Out CV

CV = **NULL**

**for**(i **in** 1:15)

{

X\_mdl\_mat = **model.matrix**(models[[i]]) **head**(X\_mdl\_mat)

Y\_vec = X1$V5

H = X\_mdl\_mat%\*%**solve**(**t**(X\_mdl\_mat)%\*%X\_mdl\_mat)%\*%**t**(X\_mdl\_mat) h = **diag**(H)

n\_h = **nrow**(X1)

CV[i] = (1/n\_h)\***sum**((Y\_vec-H%\*%Y\_vec)^2/(1-h)^2)

}

**dotplot**(CV ~ mnames, type = "o", pch = 16,xlab="Models")

250

200

150

CV

100

V1 V2 V3 V4 V12 V13 V14 V23 V24 V34 V123 V124 V134 V234 V1234

Models

## Values of different measures for all the models

We list down the values of *R*2*, R*2 *,* Mallow’s *Cp, AIC, BIC, CV* (1) values in one

*adj*

table for all the 15 models for better comparison :-

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| V1 | R2  0.7262947 | adj.R2  0.7194521 | Cp  19.406493 | AIC  319.7608 | BIC  324.9738 | CV  112.46225 |
| V2 | 0.5539177 | 0.5427657 | 55.560548 | 340.2757 | 345.4888 | 185.08170 |
| V3 | 0.6942925 | 0.6866498 | 26.118573 | 324.4050 | 329.6180 | 127.88888 |
| V4 | 0.3600579 | 0.3440593 | 96.220397 | 355.4325 | 360.6455 | 261.15078 |
| V12 | 0.7309995 | 0.7172046 | 20.419718 | 321.0326 | 327.9832 | 119.82317 |
| V13 | 0.8159665 | 0.8065289 | 2.598883 | 305.0896 | 312.0402 | 79.76883 |
| V14 | 0.7438759 | 0.7307413 | 17.719047 | 318.9724 | 325.9231 | 110.28385 |
| V23 | 0.7675782 | 0.7556591 | 12.747757 | 314.8939 | 321.8445 | 100.10949 |
| V24 | 0.5635089 | 0.5411247 | 55.548917 | 341.3629 | 348.3135 | 188.38242 |
| V34 | 0.7084529 | 0.6935017 | 25.148610 | 324.4131 | 331.3638 | 128.75523 |
| V123 | 0.8174541 | 0.8030425 | 4.286879 | 306.7487 | 315.4370 | 84.42043 |
| V124 | 0.7730732 | 0.7551579 | 13.595257 | 315.8890 | 324.5773 | 105.20886 |
| V134 | 0.8173399 | 0.8029194 | 4.310820 | 306.7749 | 315.4633 | 83.10493 |

V234 0.7684411 0.7501601 14.566777 316.7376 325.4260 105.04886

V1234 0.8235897 0.8045183 5.000000 307.3127 317.7387 85.14023

## Conclusion

Now, we write down the optimals models we get from different model selection criterions with corresponding values :-

|  |  |  |
| --- | --- | --- |
| Criterions | Optimum Model | Value |
| *R*2  *adj* | *Y* = *β*0 + *β*1A1 + *β*3A3 | 0*.*8065 |
| Mallow’s *Cp* | *Y* = *β*0 + *β*1A1 + *β*3A3 | 2*.*598 |
| AIC | *Y* = *β*0 + *β*1A1 + *β*3A3 | 305*.*089 |
| BIC | *Y* = *β*0 + *β*1A1 + *β*3A3 | 312*.*0402 |
| *CV* (1) | *Y* = *β*0 + *β*1A1 + *β*3A3 | 79*.*76883 |

Hence, clearly this indicates among all the linear models, *Y* = *β*0+*β*1A1+*β*3A3 is optimum based on several criterions.

This is also intuitive from the fact that here we are removing the covariates which had linear dependence.

In terms of terminology of the given dataset, the optimum predictors of *Y* = Cirrhosis death rate are A1 = Urban population & A3 = Wine consumption per capita.

So the optimum fitted model can be written as :-

*Y* = 9*.*9241 + 0*.*6397A1 + 1*.*5159A3

Also we observed that all the covariates in the model are signifcant.

## Obs vs Fitted Values

We plot the observed vs fitted values obtained using this model :-

X1 = X[-**c**(12,20,30,36),]

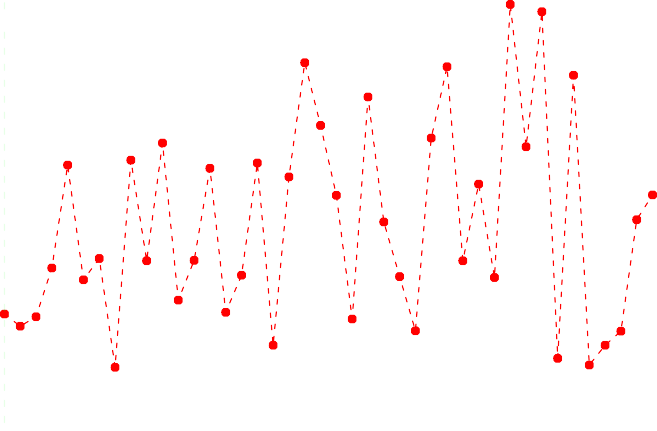
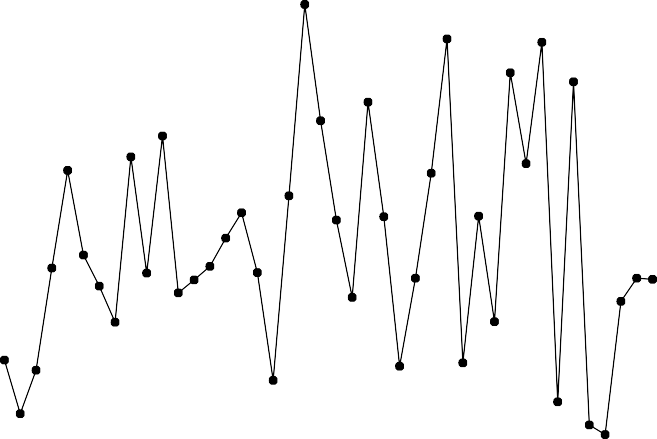
mod\_opt = **lm**(Y ~ A1+A3, data = X[-**c**(12,20,30,36),])

**plot**(1:**nrow**(X1),X1$Y,type = "o",pch = 20,ylab = "observed & predicted values",xlab = "Index")

**lines**(1:**nrow**(X1),mod\_opt$fitted.values,type = "o",pch = 20,col = "red",lty = 2) **abline**(v = 1:**nrow**(X1),lty = 2,col = **rgb**(0,1,0,alpha = 0.3)) **legend**("topleft",legend = **c**("Obs","Pred"),fill = **c**("black","red"))

80

100



Obs Pred

0 10 20 30 40

observed & predicted values

40

60

Index

Here we can see the prediction is accurate compared to the full model.

## Conclusion

Again we perform all the diagonistic tests for this final model and find the following

:-

|  |  |
| --- | --- |
| Tests | p-values |
| Shapiro-Wilk | 0.6282 |
| Durbin-Watson | 0.645 |
| Breusch-Pagan | 0.2723 |
| Breusch-Godfrey | 0.9345 |

All the assumptions seem to be satishfied here. Hence we can really consider this to be a good model.

## Models with Interaction Terms

One class of models that we have not considered yet are those with interaction terms (upto second order). Since there can be too many of them, we will not perform best subset selection here. Rather we again perform stepwise regression for choosing an optimal one among them.

In this class, we get the following model :-

mod\_wt\_out = **lm**(Y ~ A1+A2+A3+A4, data = X[-**c**(12,20,30,36),])

STEP\_REG = **stepAIC**(mod\_wt\_out,scope = **list**(upper = ~(A1+A2+A3+A4)^2, lower = ~1),trace = TRUE)

Start: AIC=186.12

Y ~ A1 + A2 + A3 + A4

Df Sum of Sq RSS AIC

+ A1:A4 1 186.85 2595.5 185.20

+ A2:A4 1 173.31 2609.0 185.42

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| - A4 | 1 | 96.77 | 2879.1 185.56 | |
| - A2 | 1 | 98.57 | 2880.9 185.58 | |
| + A3:A4 | 1 | 153.19 | 2629.1 185.74 | |
| + A1:A2 | 1 | 134.69 | 2647.7 186.04 | |
| <none> |  |  | 2782.3 186.12 | |
| + A2:A3 | 1 | 77.78 | 2704.6 186.93 | |
| + A1:A3 | 1 | 22.44 | 2759.9 187.78 | |
| - A3 | 1 | 796.75 | 3579.1 194.70 | |
| - A1 | 1 | 869.80 | 3652.1 195.55 | |
| Step: AIC=185.2  Y ~ A1 + A2 + A3 + A4 + A1:A4  Df Sum of Sq RSS AIC  - A2 1 104.35 2699.8 184.86  <none> 2595.5 185.20 | | | | |
| - A1:A4 1 | | 186.85 | 2782.3 | 186.12 |
| + A1:A3 1 | | 56.52 | 2539.0 | 186.28 |
| + A1:A2 1 | | 7.45 | 2588.0 | 187.08 |
| + A3:A4 1 | | 4.07 | 2591.4 | 187.14 |
| + A2:A3 1 | | 3.27 | 2592.2 | 187.15 |
| + A2:A4 1 | | 0.45 | 2595.0 | 187.19 |
| - A3 1 | | 674.06 | 3269.6 | 192.90 |
| Step: AIC=184.86  Y ~ A1 + A3 + A4 + A1:A4  Df Sum of Sq RSS AIC | | | | |
| <none> | | 2699.8 | | 184.86 |
| + A2 1 | | 104.35 2595.5 | | 185.20 |
| - A1:A4 1 | | 181.07 2880.9 | | 185.58 |
| + A1:A3 1 | | 78.30 2621.5 | | 185.62 |
| + A3:A4 1 | | 0.02 2699.8 | | 186.86 |
| - A3 1 | | 1005.28 3705.1 | | 196.15 |
| STEP\_REG | |  | |  |

This model has an adjusted *R*2 value equal to 0*.*8299. But the main problem is this model has very high vif values and many of the predictors are not signifcant. So we don’t consider these type of models.

Call:

lm(formula = Y ~ A1 + A3 + A4 + A1:A4, data = X[-c(12, 20, 30, 36), ])

Coefficients:

(Intercept) A1 A3 36.182148 0.121091 1.366505

A4

-0.448029

A1:A4

0.008963

## Conclusion

Hence we conclude our final multiple linear regression model is :-

*Y* = 9*.*9241 + 0*.*6397A1 + 1*.*5159A3

Obviously this model also has some drawback and there is no such “best” model that we can have but this performs more or less better than most of the models hence, it is a good one.

Next we use other types of regression models with different interpretations.

## Shrinkage Methods Ridge Regression

An alternate approach to deal with collinearity is fitting a ridge regression model as it can improve the accuracy of the predictions. The ridge estimate of the model

parameters is ***β***^ (*k*) = 1***X****T* ***X*** + *k****I***2−1 ***X****T* ***Y*** where *k* is the ridge parameter.

For an optimal choice of *k*, we calculate estimates of prediction errors of the ridge predictors for different choices of *k* over a set of trial values. This can be

*n* [ *T* ( )]2

expressed as *CV*

(1) = 1 Σ

*Yi*−***x****i* ^***β*** *k*

and choose the *k*

for which this quantity

*k*

is minimum.

*n i*=1

[1−*aii*(*k*)]2

*opt*

We plot the *CVk* (1) values for different choices of *k* :-

CV(1)

83.6 83.8 84.0 84.2 84.4 84.6 84.8 85.0

k\_grid = 10^**seq**(-2,1/2,length.out = 100) PE = **NULL**

X\_R = **as.matrix**(X[-**c**(12,20,30,36),**c**(2,3,4,5,6)])

Y\_R = **as.matrix**(X[-**c**(12,20,30,36),**c**(7)])

n = **nrow**(X\_R)

**for**(i **in** 1:**length**(k\_grid))

{

k = k\_grid[i]

beta\_k = **solve**(**t**(X\_R)%\*%X\_R + k\***diag**(**rep**(1,5)))%\*%**t**(X\_R)%\*%Y\_R A\_k = X\_R%\*%**solve**(**t**(X\_R)%\*%X\_R + k\***diag**(**rep**(1,5)))%\*%**t**(X\_R) Y\_ft\_R = X\_R%\*%beta\_k

A\_K\_diag = **diag**(A\_k)

PE[i] = (1/n)\***sum**((Y\_R-Y\_ft\_R)^2/(1-A\_K\_diag)^2)

}

**plot**(k\_grid,PE,type = "l",ylab = "CV(1)",xBlab = "k") k\_opt = k\_grid[**which**(PE == **min**(PE))]

**abline**(v = k\_opt)

0.0 0.5 1.0 1.5 2.0 2.5 3.0

k\_grid

We find that the *CVk* (1) is minimum for *k* 0*.*3898 hence, we calculate the corresponding ridge estimates.

≈

The fitted model then becomes :-

40

60

80

*Y* = 11*.*777 + 0*.*767A1 + −0*.*322A2 + 1*.*324A3 + 0*.*114A4

This model has estimated prediction error 83*.*612.

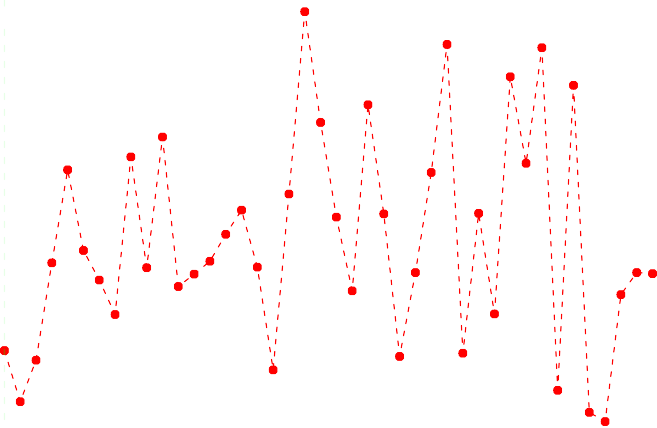
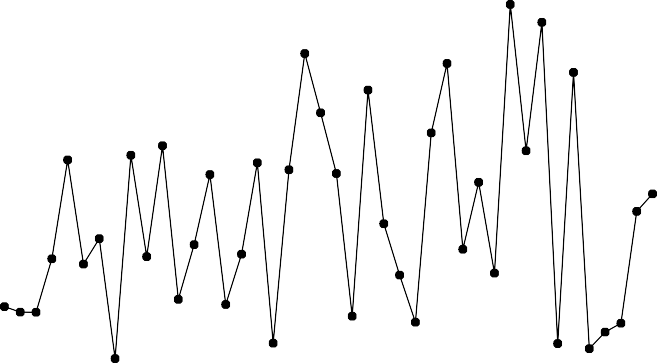
≈

This is close to the optimum OLS model that we have fitted.

We plot the observed & fitted values with the same index in the x-axis and get the following output :-

observed & predicted values

100



Obs Pred

0 10 20 30 40

Index

## Lasso Regression

Another efficient way of model selection is using the Lasso Regression method. Here we minimize the sum of squares ||***Y*** − ***Xβ***||2 subject to the constraint *j* | *j*| ≤

Σ *β λ* for some *λ >* 0.

Using R, we find the Lasso Estimates of ***β*** where the value of *λ* is chosen using

*k*-fold cross-validation criteria.

The optimum value of *λ* chosen by the criteria approximately equals 0*.*501 and the model is :-

≈

**library**(glmnet)

x <- **as.matrix**(X[-**c**(12,20,30,36),**c**(3,4,5,6)]) y <- X[-**c**(12,20,30,36),7]

lambdas <- 10^**seq**(-1, 5, by = 0.1)

lasso\_reg <- **cv.glmnet**(x,y, alpha = 1, lambda = lambdas, standardize = TRUE, nfolds = 10) lambda\_best <- lasso\_reg$lambda.min

lasso\_model <- **glmnet**(x,y, alpha = 1, lambda = 5, standardize = TRUE)

**c**(lasso\_model$a0,**t**(lasso\_model$beta))

$s0

[1] 25.2888

[[2]]

1 x 4 sparse Matrix of class "dgCMatrix"

A1 A2 A3 A4 s0 0.4579335 . 1.013119 .

*β*0

 

*β*1

25*.*28

 

0*.*457

Hence the Lasso Estimates of the parameters ***β*** = *β*2 =  0 .

 

*β*4

*β*3

1*.*013

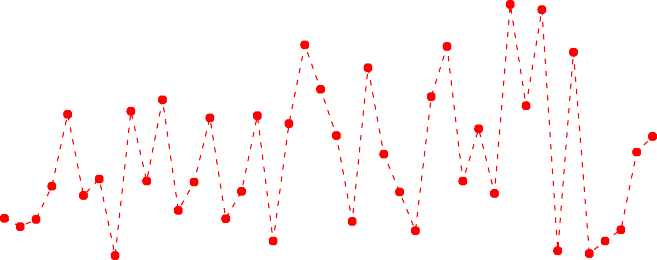
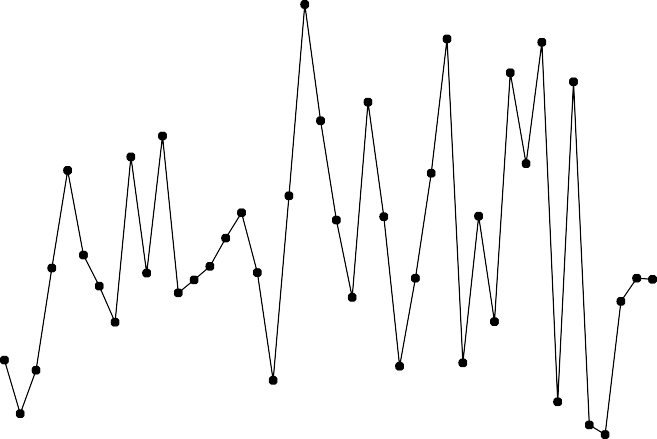
0

As we can see from the output, here also, the variables “A2”, “A4” has been dropped and this also gives strong evidence in favour of the optimum linear model.

Here also make the observed and fitted plot for different index values :-

80

100



0 10 20 30 40

1:nrow(X\_R)

## Robust Regression Methods

We have detected influential points in our dataset, and also removed them to get better models.

Now, we demonstrate using different robust regression methods how they can be used even if we have outliers in our dataset.

So we perform the rest of the methods using the full dataset, without removing any observation.

## Least Absolute Deviation

***b***

Here we minimize the quantity Σ*i* |*ei* (***b***)| i.e.

Y\_R

40

60

*ei* (***b***) = *Yi* − ***x****T* ***b***.

*i*

The estimated values of ***β***

**library**("L1pack")

rmodel\_l <- **lad**(formula = Y~A1+A2+A3+A4,data = X) rmodel\_l$coefficients

^

*LAD*

equals :-

***β***^ *LAD* = argmin Σ*i* |*ei* (***b***)|where

observed & predicted values

40

60

80

100

120

(Intercept)

A1

A2

A3

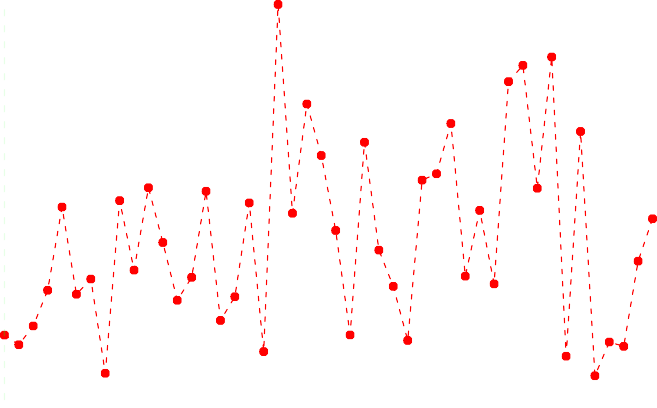
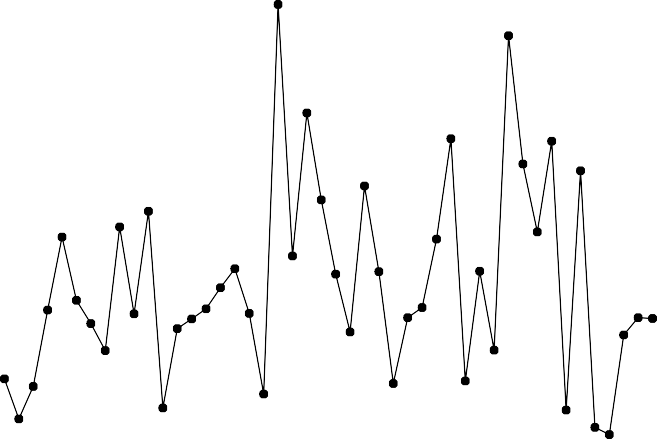
A4

-7.19977780 0.42727435 0.61749765 1.96154792 -0.02363183

We plot the observed vs fitted values obtained using this model :-

**plot**(1:**nrow**(X),X$Y,type = "o",pch = 20,ylab = "observed & predicted values",xlab = "Index")

**lines**(1:**nrow**(X),rmodel\_l$fitted.values,type = "o",pch = 20,col = "red",lty = 2) **abline**(v = 1:**nrow**(X),lty = 2,col = **rgb**(0,1,0,alpha = 0.3)) **legend**("topleft",legend = **c**("Obs","Pred"),fill = **c**("black","red"))



Obs Pred

0 10 20 30 40

Index

The plot shows here the predicted values are more or less accurate for all the observations.

## Least Median Square

Here we minimize the median of the squared residuals ***β***^ *LMS* = argmin med *e*2 (***b***)

.

***b***

*i*

*i*

Using R, we get the estimated value of ***β***

rmodel\_l <- **lqs**(Y~A1+A2+A3+A4,data = X,method = "lms") rmodel\_l$coefficients

^

*LMS*

as :-

(Intercept) 100.67646579

1.43011898 -3.48160404

A1

A2

3.15085397 -0.04186281

A3

A4

We plot the observed vs fitted values obtained using this model :-

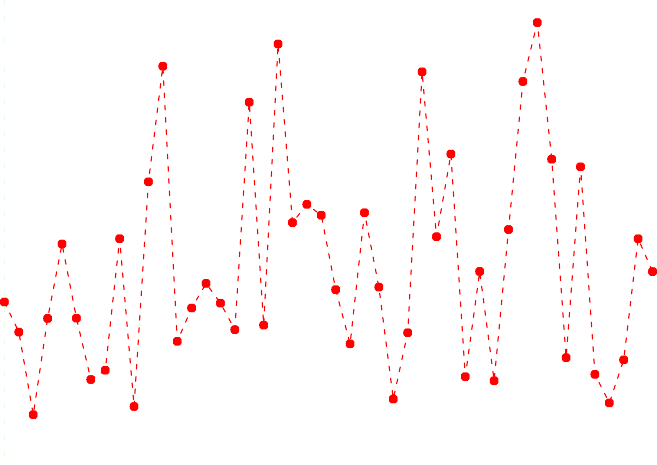
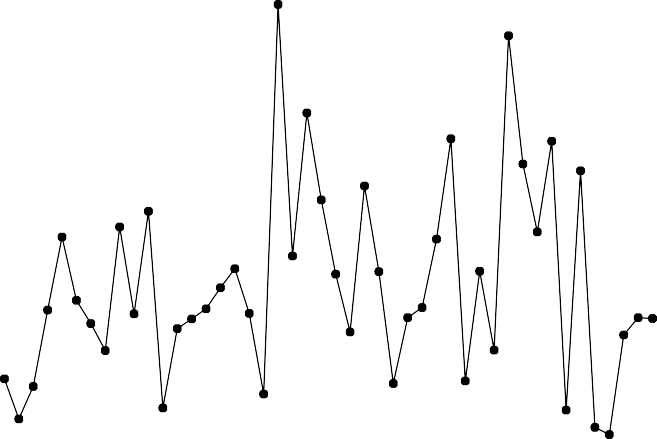
**par**(mfrow = **c**(1,1))

**plot**(1:**nrow**(X),X$Y,type = "o",pch = 20,ylab = "observed & predicted values",xlab = "Index")

**lines**(1:**nrow**(X),rmodel\_l$fitted.values,type = "o",pch = 20,col = "red",lty = 2) **abline**(v = 1:**nrow**(X),lty = 2,col = **rgb**(0,1,0,alpha = 0.3)) **legend**("topleft",legend = **c**("Obs","Pred"),fill = **c**("black","red"))

100

120



Obs Pred

0 10 20 30 40

observed & predicted values

40

60

80

Index

We can see the prediction is quite accurate at some places where as it is bad at some others possibly due to the presence of outliers.

## Least Trimmed Squares Estimate

Lastly, we compute the LTS estimates of ***β*** where we minimize the trimmed mean

*h*

1

***b***

*h i*=1

*i*

of the squared residuals ***β***^ *LMS* = argmin  Σ *e*2 (***b***) for some appropriate choice of

*h*. (Here we choose *h* = [*n/*2] + 1)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| rmodel\_l<-**lqs**(Y~A1+A2+A3+A4,data = X,method = "lts") rmodel\_l$coefficients | | | | |
| (Intercept) | A1 | A2 | A3 | A4 |
| -21.7304500 | 0.2861684 | 1.2215938 | 2.4980685 | -0.1567162 |

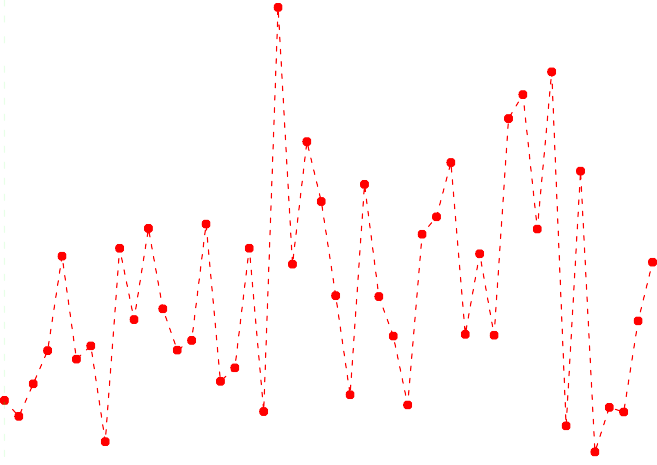
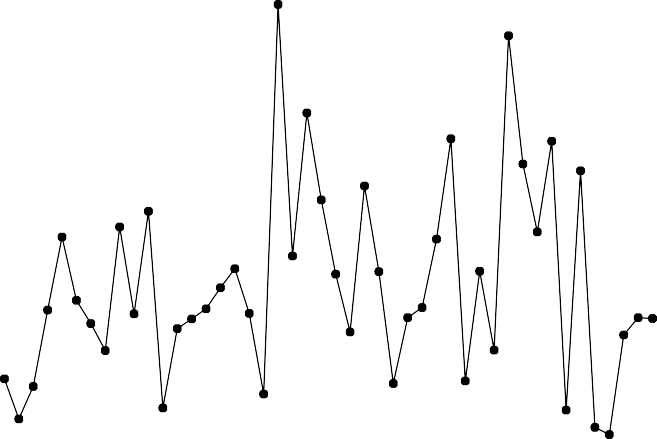
We plot the observed vs fitted values obtained using this model :-

**plot**(1:**nrow**(X),X$Y,type = "o",pch = 20,ylab = "observed & predicted values",xlab = "Index")

**lines**(1:**nrow**(X),rmodel\_l$fitted.values,type = "o",pch = 20,col = "red",lty = 2) **abline**(v = 1:**nrow**(X),lty = 2,col = **rgb**(0,1,0,alpha = 0.3)) **legend**("topleft",legend = **c**("Obs","Pred"),fill = **c**("black","red"))

100

120



Obs Pred

0 10 20 30 40

observed & predicted values

40

60

80

Index

We can see that this method performs better in terms of prediction compared to LMS estimates.

## Comparative Study Of All Models

Finally, as a measure of comparison of different models, we use the root mean

square error *RMSE* = ó 1 Σ (*y* )2where *y* denotes the fitted values using

*n*

^

− *y* ^

*n*

*i*=1

*i*

*i*

*i*

different regression models. We calculate this measure for all the “good” models we have found so far :-

|  |  |
| --- | --- |
| Methods | *RMSE* values |
| OLS model with “A1” & “A3” | 8.313178 |
| Ridge Model | 8.195655 |
| Lasso Model | 9.889319 |
| LAD Model | 10.59494 |
| LMS Model | 11.6656 |
| LTS Model | 11.0904 |

Hence, we can conclude the Ridge and the OLS model with influential points removed with covariates “A1” & “A3” performs more or less better than the others in terms of prediction accuracy.