GDP_Model_Output.R

lalone

2022-08-29

```
### GDP Data Import In R ###
# setwd("E:/GDP")
data=read.csv(file.choose(),header = T)
#View(data)
df=data[,-1]
dim(df)
```

```
## [1] 59 17
```

```
head(df)
```

```
##              x1    x2           x3   x4    x5    x6    x7    x8     x9
## 1 3.985662e+12  8857 7.900065e+11 2127 26295 36161 31252 12905 408739
## 2 3.989018e+12  9367 8.574691e+11 2431 27219 38495 32596 13731 420953
## 3 3.909672e+12 10479 9.198878e+11 2730 28233 40527 33693 15510 429594
## 4 4.001130e+12 10789 1.006896e+12 3227 31680 43489 34735 17242 451446
## 5 4.370205e+12 10945 1.076520e+12 3523 34225 46693 35688 19093 485193
## 6 3.887638e+12 12231 1.086518e+12 3892 36509 46981 36766 19773 467155
##      x10  x11          x12          x13  x14          x15          x16
## 1 385761 8889 6.024061e+12 393950147076 6128 293837064043 495391527474
## 2 396844 8938 6.126761e+12 423982005990 5204 293290899649 447061619964
## 3 404119 8901 6.206577e+12 511559938963 4508 291590307387 464538221660
## 4 424527 9149 6.437464e+12 628618037771 3680 316095783962 480295306195
## 5 456327 9627 6.822025e+12 656720472996 6218 295511469528 496528052096
## 6 436650 9003 6.828064e+12 720379534332 4695 254541237910 441630878920
##               y
## 1 7.370436e+12
## 2 7.644819e+12
## 3 7.868898e+12
## 4 8.340588e+12
## 5 8.962207e+12
## 6 8.725984e+12
```

```
str(df)
```

```
## 'data.frame':    59 obs. of  17 variables:
##  $ x1 : num  3.99e+12 3.99e+12 3.91e+12 4.00e+12 4.37e+12 ...
##  $ x2 : num  8857 9367 10479 10789 10945 ...
##  $ x3 : num  7.90e+11 8.57e+11 9.20e+11 1.01e+12 1.08e+12 ...
##  $ x4 : num  2127 2431 2730 3227 3523 ...
##  $ x5 : num  26295 27219 28233 31680 34225 ...
##  $ x6 : num  36161 38495 40527 43489 46693 ...
##  $ x7 : num  31252 32596 33693 34735 35688 ...
##  $ x8 : num  12905 13731 15510 17242 19093 ...
##  $ x9 : num  408739 420953 429594 451446 485193 ...
##  $ x10: num  385761 396844 404119 424527 456327 ...
##  $ x11: num  8889 8938 8901 9149 9627 ...
##  $ x12: num  6.02e+12 6.13e+12 6.21e+12 6.44e+12 6.82e+12 ...
##  $ x13: num  3.94e+11 4.24e+11 5.12e+11 6.29e+11 6.57e+11 ...
##  $ x14: num  6128 5204 4508 3680 6218 ...
##  $ x15: num  2.94e+11 2.93e+11 2.92e+11 3.16e+11 2.96e+11 ...
##  $ x16: num  4.95e+11 4.47e+11 4.65e+11 4.80e+11 4.97e+11 ...
##  $ y  : num  7.37e+12 7.64e+12 7.87e+12 8.34e+12 8.96e+12 ...
```

```
y=df[,17]
head(y)
```

```
## [1] 7.370436e+12 7.644819e+12 7.868898e+12 8.340588e+12 8.962207e+12
## [6] 8.725984e+12
```

```
x=df[,-17]
head(x)
```

```
##             x1    x2           x3   x4    x5    x6    x7    x8     x9
## 1 3.985662e+12  8857 7.900065e+11 2127 26295 36161 31252 12905 408739
## 2 3.989018e+12  9367 8.574691e+11 2431 27219 38495 32596 13731 420953
## 3 3.909672e+12 10479 9.198878e+11 2730 28233 40527 33693 15510 429594
## 4 4.001130e+12 10789 1.006896e+12 3227 31680 43489 34735 17242 451446
## 5 4.370205e+12 10945 1.076520e+12 3523 34225 46693 35688 19093 485193
## 6 3.887638e+12 12231 1.086518e+12 3892 36509 46981 36766 19773 467155
##      x10  x11          x12          x13  x14          x15          x16
## 1 385761 8889 6.024061e+12 393950147076 6128 293837064043 495391527474
## 2 396844 8938 6.126761e+12 423982005990 5204 293290899649 447061619964
## 3 404119 8901 6.206577e+12 511559938963 4508 291590307387 464538221660
## 4 424527 9149 6.437464e+12 628618037771 3680 316095783962 480295306195
## 5 456327 9627 6.822025e+12 656720472996 6218 295511469528 496528052096
## 6 436650 9003 6.828064e+12 720379534332 4695 254541237910 441630878920
```

```
### Log trancformation all varibles
y=log(y)
x=cbind(log(x$x1),log(x$x2),log(x$x3),log(x$x4),log(x$x5),log(x$x6),log(x$x7)
,

log(x$x8),log(x$x9),log(x$x10),log(x$x11),log(x$x12),log(x$x13),log(x$x14),
        log(x$x15),log(x$x16))
```

```
## Warning in log(x$x14): NaNs produced
```

```
head(x)
```

```
##           [,1]     [,2]     [,3]     [,4]     [,5]     [,6]     [,7]
## [1,] 29.01372 9.088963 27.39531 7.662468 10.17713 10.49574 10.34984
## [2,] 29.01457 9.144948 27.47725 7.796058 10.21167 10.55828 10.39194
## [3,] 28.99447 9.257129 27.54752 7.912057 10.24825 10.60972 10.42505
## [4,] 29.01760 9.286282 27.63789 8.079308 10.36344 10.68026 10.45550
## [5,] 29.10583 9.300638 27.70475 8.167068 10.44071 10.75135 10.48257
## [6,] 28.98882 9.411729 27.71400 8.266678 10.50531 10.75750 10.51233
##          [,8]     [,9]    [,10]    [,11]    [,12]    [,13]    [,14]
## [1,] 9.465370 12.92083 12.86297 9.092570 29.42678 26.69949 8.720624
## [2,] 9.527411 12.95028 12.89130 9.098067 29.44369 26.77296 8.557183
## [3,] 9.649240 12.97060 12.90946 9.093919 29.45663 26.96073 8.413609
## [4,] 9.755104 13.02021 12.95873 9.121400 29.49316 27.16679 8.210668
## [5,] 9.857077 13.09230 13.03096 9.172327 29.55118 27.21052 8.735204
## [6,] 9.892073 13.05442 12.98689 9.105313 29.55206 27.30304 8.454253
##         [,15]    [,16]
## [1,] 26.40629 26.92861
## [2,] 26.40443 26.82596
## [3,] 26.39862 26.86431
## [4,] 26.47931 26.89767
## [5,] 26.41197 26.93091
## [6,] 26.26273 26.81374
```

```
x=as.data.frame.matrix(x)
summary(x)
```

```
##        V1              V2               V3              V4        
##  Min.   :28.97   Min.   : 9.089   Min.   :27.40   Min.   : 7.662  
##  1st Qu.:29.26   1st Qu.: 9.672   1st Qu.:28.06   1st Qu.: 8.980  
##  Median :29.69   Median :10.656   Median :28.85   Median :10.138  
##  Mean   :29.69   Mean   :10.652   Mean   :28.93   Mean   :10.102  
##  3rd Qu.:30.07   3rd Qu.:11.313   3rd Qu.:29.67   3rd Qu.:11.007  
##  Max.   :30.55   Max.   :12.823   Max.   :30.78   Max.   :12.565  
##                                                                   
##        V5              V6              V7              V8        
##  Min.   :10.18   Min.   :10.50   Min.   :10.35   Min.   : 9.465  
##  1st Qu.:10.76   1st Qu.:11.12   1st Qu.:10.83   1st Qu.:10.436  
##  Median :11.35   Median :11.90   Median :11.89   Median :11.336  
##  Mean   :11.63   Mean   :12.15   Mean   :12.08   Mean   :11.434  
##  3rd Qu.:12.27   3rd Qu.:12.95   3rd Qu.:12.95   3rd Qu.:12.039  
##  Max.   :13.87   Max.   :14.72   Max.   :14.86   Max.   :14.343  
##                                                                  
##        V9             V10             V11              V12       
##  Min.   :12.92   Min.   :12.86   Min.   : 9.091   Min.   :29.43  
##  1st Qu.:13.39   1st Qu.:13.30   1st Qu.: 9.221   1st Qu.:29.79  
##  Median :14.05   Median :13.95   Median : 9.543   Median :30.41  
##  Mean   :14.25   Mean   :14.15   Mean   : 9.770   Mean   :30.50  
##  3rd Qu.:14.86   3rd Qu.:14.75   3rd Qu.:10.067   3rd Qu.:31.08  
##  Max.   :16.45   Max.   :16.33   Max.   :11.436   Max.   :32.02  
##                                                                  
##       V13             V14              V15             V16       
##  Min.   :26.70   Min.   : 6.912   Min.   :26.26   Min.   :26.73  
##  1st Qu.:27.71   1st Qu.: 8.896   1st Qu.:27.12   1st Qu.:27.06  
##  Median :28.64   Median : 9.776   Median :27.98   Median :28.12  
##  Mean   :28.57   Mean   : 9.979   Mean   :28.38   Mean   :28.55  
##  3rd Qu.:29.31   3rd Qu.:11.338   3rd Qu.:29.69   3rd Qu.:29.75  
##  Max.   :30.34   Max.   :12.524   Max.   :31.01   Max.   :31.20
##                  NA's   :5
```

```
x$V14=ifelse(is.na(x$V14),median(x$V14,na.rm = T),x$V14)
summary(x)
```

```
##        V1              V2                V3              V4        
##  Min.   :28.97   Min.   : 9.089   Min.   :27.40   Min.   : 7.662  
##  1st Qu.:29.26   1st Qu.: 9.672   1st Qu.:28.06   1st Qu.: 8.980  
##  Median :29.69   Median :10.656   Median :28.85   Median :10.138  
##  Mean   :29.69   Mean   :10.652   Mean   :28.93   Mean   :10.102  
##  3rd Qu.:30.07   3rd Qu.:11.313   3rd Qu.:29.67   3rd Qu.:11.007  
##  Max.   :30.55   Max.   :12.823   Max.   :30.78   Max.   :12.565  
##        V5              V6              V7              V8        
##  Min.   :10.18   Min.   :10.50   Min.   :10.35   Min.   : 9.465  
##  1st Qu.:10.76   1st Qu.:11.12   1st Qu.:10.83   1st Qu.:10.436  
##  Median :11.35   Median :11.90   Median :11.89   Median :11.336  
##  Mean   :11.63   Mean   :12.15   Mean   :12.08   Mean   :11.434  
##  3rd Qu.:12.27   3rd Qu.:12.95   3rd Qu.:12.95   3rd Qu.:12.039  
##  Max.   :13.87   Max.   :14.72   Max.   :14.86   Max.   :14.343  
##        V9             V10             V11               V12       
##  Min.   :12.92   Min.   :12.86   Min.   : 9.091   Min.   :29.43  
##  1st Qu.:13.39   1st Qu.:13.30   1st Qu.: 9.221   1st Qu.:29.79  
##  Median :14.05   Median :13.95   Median : 9.543   Median :30.41  
##  Mean   :14.25   Mean   :14.15   Mean   : 9.770   Mean   :30.50  
##  3rd Qu.:14.86   3rd Qu.:14.75   3rd Qu.:10.067   3rd Qu.:31.08  
##  Max.   :16.45   Max.   :16.33   Max.   :11.436   Max.   :32.02  
##       V13             V14               V15             V16       
##  Min.   :26.70   Min.   : 6.912   Min.   :26.26   Min.   :26.73  
##  1st Qu.:27.71   1st Qu.: 8.929   1st Qu.:27.12   1st Qu.:27.06  
##  Median :28.64   Median : 9.776   Median :27.98   Median :28.12  
##  Mean   :28.57   Mean   : 9.962   Mean   :28.38   Mean   :28.55  
##  3rd Qu.:29.31   3rd Qu.:11.084   3rd Qu.:29.69   3rd Qu.:29.75  
##  Max.   :30.34   Max.   :12.524   Max.   :31.01   Max.   :31.20  
```

```
sum(is.na(x))
```

```
## [1] 0
```

```
df1=data.frame(y,x)
```

```
model=lm(y~.,data = df1)
summary(model)
```

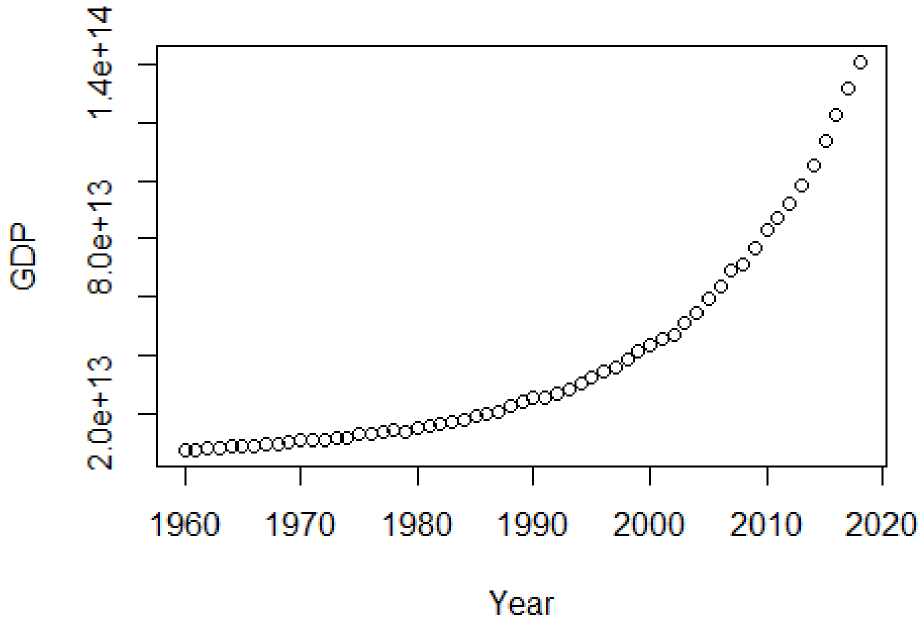
```
## 
## Call:
## lm(formula = y ~ ., data = df1)
## 
## Residuals:
##        Min         1Q     Median         3Q        Max 
## -0.0160260 -0.0057336  0.0002305  0.0050085  0.0264007 
```

```
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  2.932268   2.600900   1.127 0.265971
## V1           0.238026   0.101139   2.353 0.023356 *
## V2          -0.013942   0.033502  -0.416 0.679421
## V3           0.057369   0.071840   0.799 0.429039
## V4          -0.056473   0.044593  -1.266 0.212347
## V5           0.046575   0.036678   1.270 0.211134
## V6           0.172876   0.088793   1.947 0.058249 .
## V7           0.186637   0.040083   4.656 3.22e-05 ***
## V8          -0.193376   0.068071  -2.841 0.006912 **
## V9          -0.197695   0.749596  -0.264 0.793273
## V10          0.311880   0.958190   0.325 0.746428
## V11         -0.046873   0.212910  -0.220 0.826817
## V12          0.328581   0.080497   4.082 0.000196 ***
## V13          0.230715   0.047211   4.887 1.54e-05 ***
## V14          0.001913   0.002278   0.840 0.405749
## V15          0.047546   0.022437   2.119 0.040039 *
## V16         -0.064755   0.012422  -5.213 5.32e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.009465 on 42 degrees of freedom
## Multiple R-squared:  0.9999, Adjusted R-squared:  0.9999
## F-statistic: 3.12e+04 on 16 and 42 DF,  p-value: < 2.2e-16
```

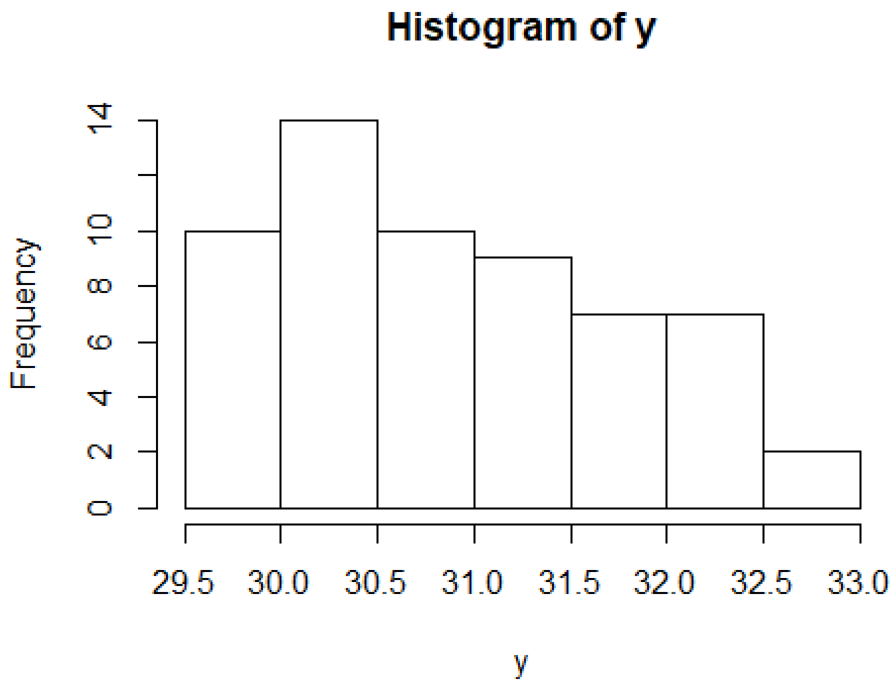
```r
### Normality check
shapiro.test(residuals(model))# H0: data is normal
```

```
##
##  Shapiro-Wilk normality test
##
## data:  residuals(model)
## W = 0.96091, p-value = 0.05548
```

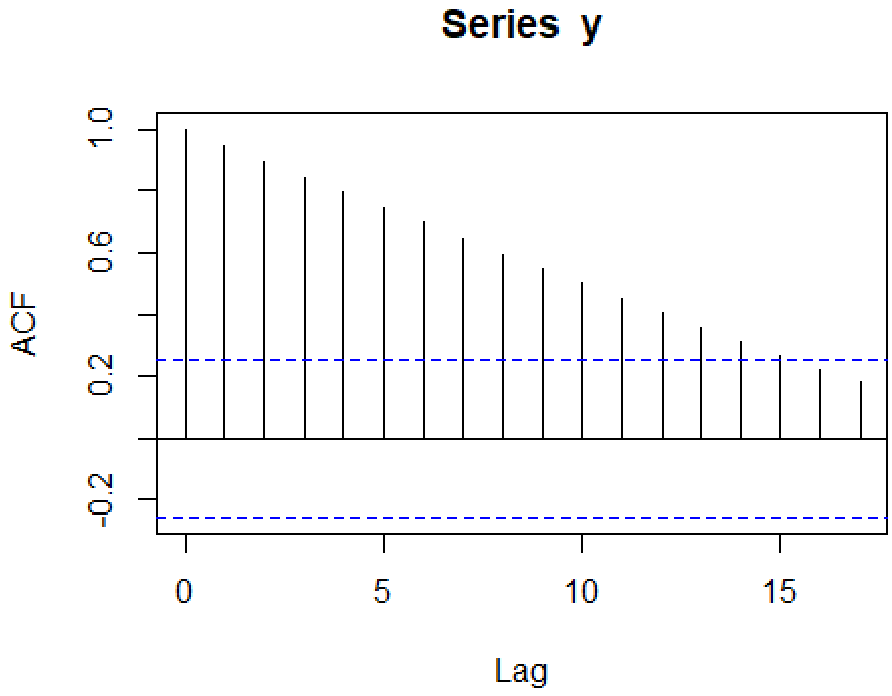
```r
plot(data[,1],data[,18],ylab="GDP",xlab = "Year")
```



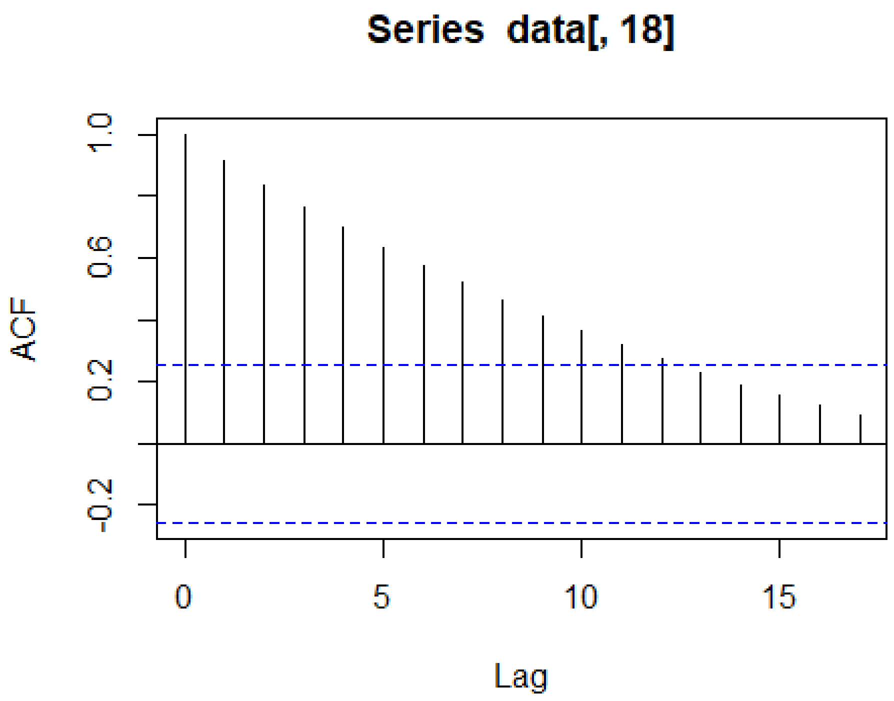
```
hist(y)

# To check Autocorrelation
library(zoo)

## Warning: package 'zoo' was built under R version 3.6.3

##
## Attaching package: 'zoo'

## The following objects are masked from 'package:base':
##
##     as.Date, as.Date.numeric
```



```
library(lmtest)

## Warning: package 'lmtest' was built under R version 3.6.3

acf(y)
```

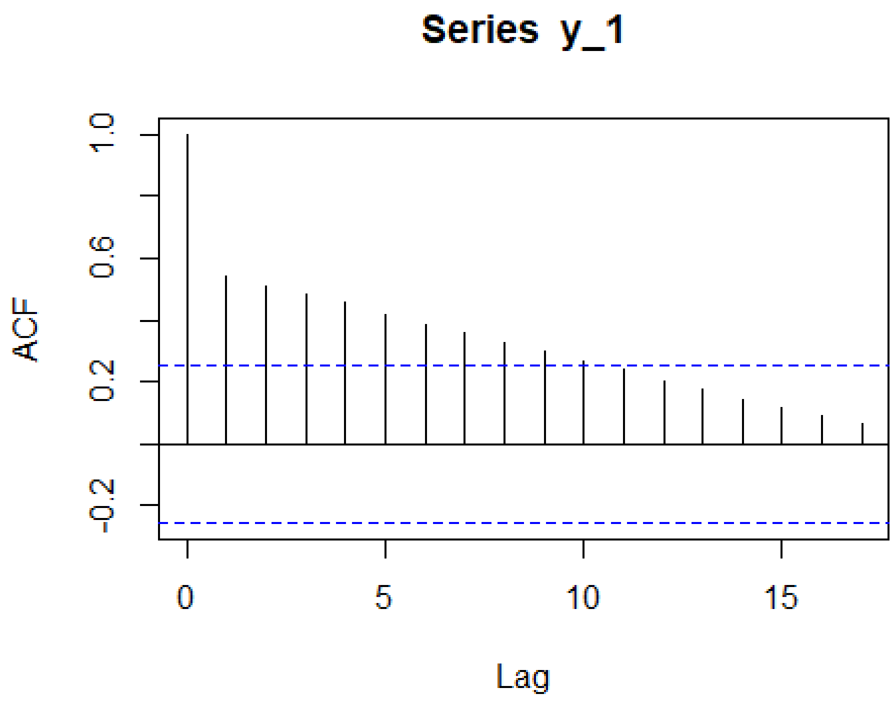


```
dwtest(model) # Ho: data is no autocorrelation

##
##  Durbin-Watson test
##
## data:  model
## DW = 1.6311, p-value = 0.001123
## alternative hypothesis: true autocorrelation is greater than 0

# above check the dwtest and H0 is rejected i.e in data autocorrelation.
bgtest(model)

##
##  Breusch-Godfrey test for serial correlation of order up to 1
##
## data:  model
## LM test = 2.819, df = 1, p-value = 0.09315

acf(data[,18])
```



```
# To remove autocorrelation
## AR(1) model lag=1
e=model$residuals
e=as.vector(e);e

##  [1] -0.0003454586 -0.0001594235 -0.0020026815 -0.0124960798  0.0077913564
##  [6] -0.0057579570 -0.0015038902  0.0056898221  0.0057701704  0.0027961667
## [11] -0.0007312502  0.0039025365  0.0029682326  0.0068336828  0.0207029984
## [16]  0.0126672123 -0.0050059478 -0.0009330813 -0.0091018124 -0.0025434773
## [21] -0.0084446962 -0.0093482161 -0.0017450636 -0.0048960909 -0.0160260097
## [26] -0.0116239890  0.0014766119  0.0048431303  0.0036898111  0.0011666817
## [31] -0.0030840331 -0.0094747600  0.0083199028 -0.0074767762  0.0048351459
## [36]  0.0079182745 -0.0059670851  0.0051739642 -0.0028380975  0.0054827200
## [41]  0.0069282047 -0.0125238554  0.0015638955  0.0009645678  0.0264007274
## [46]  0.0085310904  0.0043566902 -0.0088182036 -0.0126445966  0.0083430573
## [51] -0.0098961971  0.0002304768  0.0021791883 -0.0104412427 -0.0037233498
## [56] -0.0057093353  0.0035250507  0.0056542303  0.0045570577

e1=e[1:58]
e2=e[2:59]
d=(sum(e1^2)+sum(e2^2)-(2*sum(e1*e2)))/(sum(e^2))
d

## [1] 1.631067

rho=sum(e1*e2)/sum(e^2)
rho
```

```
## [1] 0.181691

x=as.vector(x)
class(x)

## [1] "data.frame"

yone=y[1]*sqrt(1-rho^2)
y_1=y[2:59]-rho*y[1:58]
y_1=append(y_1,yone,after = 0)
end(y_1)

## [1] 59  1

acf(y_1)
```



```
z=rho*x[1:58,1:16]
w=x[2:59,1:16]
x_11=x[1,1:16]*sqrt(1-rho^2)
x_12=w-z

x_1=rbind(x_11,x_12)
head(x_1)

##         V1       V2       V3       V4        V5        V6        V7
## 1 28.53081 8.937684 26.93933 7.534931 10.007742 10.321042 10.177572
## 2 23.74303 7.493565 22.49977 6.403856  8.362576  8.651302  8.511472
## 3 23.72279 7.595573 22.55515 6.495583  8.392878  8.691378  8.536922
## 4 23.74956 7.604345 22.63276 6.641758  8.501426  8.752572  8.561366
## 5 23.83359 7.613404 22.68320 6.699130  8.557767  8.810841  8.582898
## 6 23.70055 7.721886 22.68029 6.782795  8.608330  8.804075  8.607740
##         V8       V9      V10      V11      V12      V13      V14      V15
## 1 9.307825 12.70577 12.64888 8.941230 28.93699 26.25509 8.575475 25.96678
## 2 7.807638 10.60268 10.55421 7.446029 24.09710 21.92190 6.972724 21.60664
## 3 7.918195 10.61765 10.56723 7.440882 24.10698 22.09632 6.858845 21.60117
## 4 8.001923 10.66357 10.61320 7.469116 24.14115 22.26827 6.681991 21.68292
## 5 8.084662 10.72665 10.67648 7.515050 24.19254 22.27456 7.243399 21.60092
## 6 8.101130 10.67566 10.61928 7.438783 24.18288 22.35914 6.867145 21.46391
##        V16
## 1 26.48041
## 2 21.93327
## 3 21.99027
## 4 22.01666
## 5 22.04384
## 6 21.92064
```

```
dim(x_1)
```

```
## [1] 59 16
```

```
data1=data.frame(y_1,x_1)
model1=lm(y_1~.,data = data1)
summary(model1)
```

```
##
## Call:
## lm(formula = y_1 ~ ., data = data1)
##
## Residuals:
##        Min         1Q     Median         3Q        Max
## -0.0166482 -0.0060177 -0.0003647  0.0052934  0.0247282
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  0.023739   0.071663   0.331 0.742093
## V1           0.312450   0.058929   5.302 3.97e-06 ***
## V2           0.004188   0.033619   0.125 0.901446
## V3           0.124774   0.055024   2.268 0.028559 *
## V4          -0.081000   0.046711  -1.734 0.090245 .
## V5           0.069720   0.032932   2.117 0.040219 *
## V6           0.165498   0.089546   1.848 0.071625 .
## V7           0.188220   0.040604   4.636 3.44e-05 ***
## V8          -0.118624   0.034725  -3.416 0.001421 **
## V9          -0.080842   0.751778  -0.108 0.914877
## V10         -0.095147   0.839526  -0.113 0.910306
## V11          0.049537   0.163589   0.303 0.763524
## V12          0.384841   0.071545   5.379 3.09e-06 ***
## V13          0.187372   0.039379   4.758 2.32e-05 ***
## V14          0.002347   0.002170   1.082 0.285519
## V15          0.054495   0.019743   2.760 0.008524 **
## V16         -0.057096   0.013568  -4.208 0.000133 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.009355 on 42 degrees of freedom
## Multiple R-squared:  0.9999, Adjusted R-squared:  0.9999
## F-statistic: 3.101e+04 on 16 and 42 DF,  p-value: < 2.2e-16
```

```
dwtest(model1)
```

```
##
##  Durbin-Watson test
##
## data:  model1
## DW = 1.8205, p-value = 0.01682
## alternative hypothesis: true autocorrelation is greater than 0
```

```
## AR(2) model lag=2e=model$residuals
e_1=model1$residuals;e_1
```

```
##             1             2             3             4             5
##  0.0002288402 -0.0013130392 -0.0030241504 -0.0098815432  0.0082669852
##             6             7             8             9            10
## -0.0049807497 -0.0039971881  0.0056501350  0.0069680329  0.0031284950
##            11            12            13            14            15
## -0.0004147739  0.0055062150  0.0025399810  0.0053343328  0.0184678678
##            16            17            18            19            20
##  0.0132197654 -0.0051095732 -0.0006075532 -0.0087915551 -0.0010257976
##            21            22            23            24            25
## -0.0092041570 -0.0105283392 -0.0009383554 -0.0077684361 -0.0166482220
##            26            27            28            29            30
## -0.0074524873  0.0047026811  0.0049282741 -0.0003646896 -0.0023139527
##            31            32            33            34            35
## -0.0065973423 -0.0086839344  0.0102099623 -0.0054379686  0.0071840703
##            36            37            38            39            40
##  0.0062002883 -0.0075102132  0.0076635304 -0.0032600954  0.0066856945
##            41            42            43            44            45
##  0.0052523946 -0.0130257599  0.0068646476  0.0025673097  0.0247281620
##            46            47            48            49            50
##  0.0041637045  0.0024980111 -0.0067065919 -0.0127863821  0.0101754756
##            51            52            53            54            55
## -0.0108781416  0.0016869430  0.0006137210 -0.0128422452 -0.0005622780
##            56            57            58            59
## -0.0024934927  0.0041556985  0.0040768888  0.0014809005
```

```
sum(is.na(e_1))
```

```
## [1] 0
```

```
e_1=as.vector(e_1);e_1
```

```
##  [1]  0.0002288402 -0.0013130392 -0.0030241504 -0.0098815432  0.0082669852
##  [6] -0.0049807497 -0.0039971881  0.0056501350  0.0069680329  0.0031284950
## [11] -0.0004147739  0.0055062150  0.0025399810  0.0053343328  0.0184678678
## [16]  0.0132197654 -0.0051095732 -0.0006075532 -0.0087915551 -0.0010257976
## [21] -0.0092041570 -0.0105283392 -0.0009383554 -0.0077684361 -0.0166482220
## [26] -0.0074524873  0.0047026811  0.0049282741 -0.0003646896 -0.0023139527
## [31] -0.0065973423 -0.0086839344  0.0102099623 -0.0054379686  0.0071840703
## [36]  0.0062002883 -0.0075102132  0.0076635304 -0.0032600954  0.0066856945
## [41]  0.0052523946 -0.0130257599  0.0068646476  0.0025673097  0.0247281620
## [46]  0.0041637045  0.0024980111 -0.0067065919 -0.0127863821  0.0101754756
## [51] -0.0108781416  0.0016869430  0.0006137210 -0.0128422452 -0.0005622780
## [56] -0.0024934927  0.0041556985  0.0040768888  0.0014809005
```

```
e11=e_1[1:58];e11
```

```
##  [1]  0.0002288402 -0.0013130392 -0.0030241504 -0.0098815432  0.0082669852
##  [6] -0.0049807497 -0.0039971881  0.0056501350  0.0069680329  0.0031284950
## [11] -0.0004147739  0.0055062150  0.0025399810  0.0053343328  0.0184678678
## [16]  0.0132197654 -0.0051095732 -0.0006075532 -0.0087915551 -0.0010257976
## [21] -0.0092041570 -0.0105283392 -0.0009383554 -0.0077684361 -0.0166482220
## [26] -0.0074524873  0.0047026811  0.0049282741 -0.0003646896 -0.0023139527
## [31] -0.0065973423 -0.0086839344  0.0102099623 -0.0054379686  0.0071840703
## [36]  0.0062002883 -0.0075102132  0.0076635304 -0.0032600954  0.0066856945
## [41]  0.0052523946 -0.0130257599  0.0068646476  0.0025673097  0.0247281620
## [46]  0.0041637045  0.0024980111 -0.0067065919 -0.0127863821  0.0101754756
## [51] -0.0108781416  0.0016869430  0.0006137210 -0.0128422452 -0.0005622780
## [56] -0.0024934927  0.0041556985  0.0040768888
```

```
e21=e_1[2:59];e21
```

```
##  [1] -0.0013130392 -0.0030241504 -0.0098815432  0.0082669852 -0.0049807497
##  [6] -0.0039971881  0.0056501350  0.0069680329  0.0031284950 -0.0004147739
## [11]  0.0055062150  0.0025399810  0.0053343328  0.0184678678  0.0132197654
## [16] -0.0051095732 -0.0006075532 -0.0087915551 -0.0010257976 -0.0092041570
## [21] -0.0105283392 -0.0009383554 -0.0077684361 -0.0166482220 -0.0074524873
## [26]  0.0047026811  0.0049282741 -0.0003646896 -0.0023139527 -0.0065973423
## [31] -0.0086839344  0.0102099623 -0.0054379686  0.0071840703  0.0062002883
## [36] -0.0075102132  0.0076635304 -0.0032600954  0.0066856945  0.0052523946
## [41] -0.0130257599  0.0068646476  0.0025673097  0.0247281620  0.0041637045
## [46]  0.0024980111 -0.0067065919 -0.0127863821  0.0101754756 -0.0108781416
## [51]  0.0016869430  0.0006137210 -0.0128422452 -0.0005622780 -0.0024934927
## [56]  0.0041556985  0.0040768888  0.0014809005
```

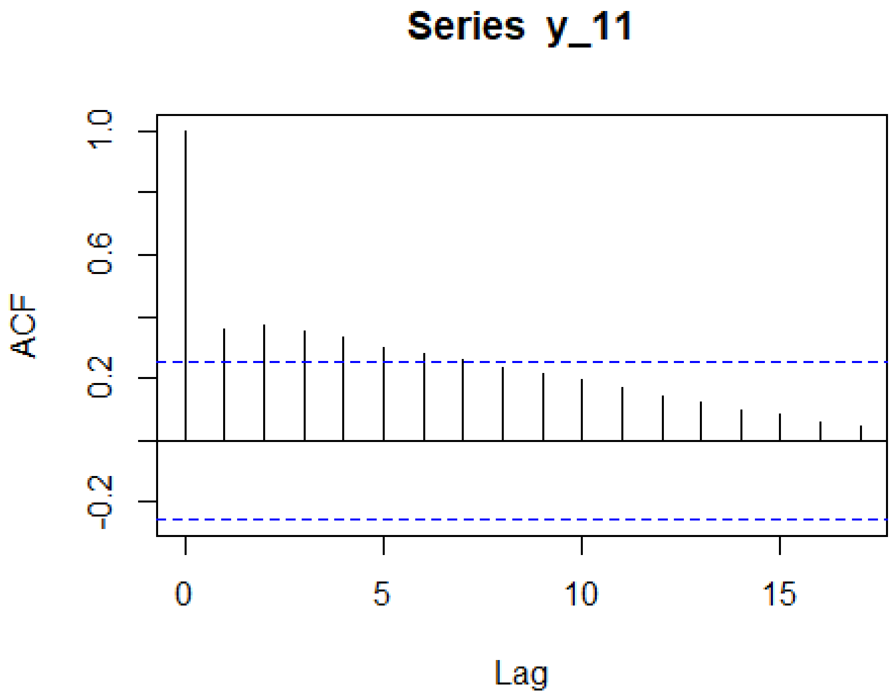
```
d=(sum(e11^2)+sum(e21^2)-(2*sum(e11*e21)))/(sum(e_1^2))
d
```

```
## [1] 1.820518
```

```
rho1=sum(e11*e21)/sum(e_1^2)
rho1
```

```
## [1] 0.08943536
```

```
#x=as.vector(x)
#class(x)
y_1one=y_1[1]*sqrt(1-rho1^1)
y_11one=y_1[2:59]-rho1*y_1[1:58]
y_11=append(y_11one,y_1one,after = 0)
end(y_11)

## [1] 59  1

acf(y_11)
```



```
x_one=x_1[1,1:16]*sqrt(1-rho1^2)
z1=rho1*x_1[1:58,1:16]
w1=x_1[2:59,1:16]
x_1one=w1-z1
dim(x_1one)

## [1] 58 16

x_11=rbind(x_one,x_1one)
dim(x_11)

## [1] 59 16

data2=data.frame(y_11,x_11)
dim(data2)

## [1] 59 17
```

```
model2=lm(y_11~.,data = data2)
summary(model2)
```

```
##
## Call:
## lm(formula = y_11 ~ ., data = data2)
##
## Residuals:
##       Min        1Q    Median        3Q       Max
## -0.033810 -0.006402 -0.001714  0.004655  0.030075
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  3.658363   0.060670  60.300  < 2e-16 ***
## V1           0.199759   0.078530   2.544  0.01474 *
## V2           0.011335   0.045074   0.251  0.80268
## V3           0.073164   0.073112   1.001  0.32270
## V4          -0.040954   0.063517  -0.645  0.52258
## V5           0.062005   0.044533   1.392  0.17114
## V6           0.235133   0.119056   1.975  0.05487 .
## V7           0.103163   0.055530   1.858  0.07022 .
## V8          -0.248347   0.046270  -5.367 3.21e-06 ***
## V9           0.077047   0.992487   0.078  0.93849
## V10          0.094782   1.098882   0.086  0.93167
## V11         -0.003449   0.215505  -0.016  0.98731
## V12          0.163510   0.096899   1.687  0.09894 .
## V13          0.340179   0.052534   6.475 8.22e-08 ***
## V14          0.001157   0.002767   0.418  0.67783
## V15          0.028733   0.025676   1.119  0.26947
## V16         -0.059350   0.018437  -3.219  0.00248 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.01223 on 42 degrees of freedom
## Multiple R-squared:  0.9999, Adjusted R-squared:  0.9998
## F-statistic: 1.983e+04 on 16 and 42 DF,  p-value: < 2.2e-16
```

```
dwtest(model2)
```

```
##
##  Durbin-Watson test
##
## data:  model2
## DW = 2.0103, p-value = 0.1016
## alternative hypothesis: true autocorrelation is greater than 0
```

```
###Again AR(3)
#e_11=model2$residuals;e_11
#sum(is.na(e_11))
#e_11=as.vector(e_11);e_11
#e111=e_11[1:57];e111
```

```
#e211=e_11[2:58];e211
#d2=(sum(e111^2)+sum(e211^2)-(2*sum(e111*e211)))/(sum(e_11^2))
#d2
#rho3=sum(e111*e211)/sum(e_11^2)
#rho3
#x=as.vector(x)
#class(x)
#y_111one=y_11[1]*sqrt(1-rho3^2)
#y_111=y_11[2:58]-rho3*y_11[1:57]
#y_111=append(y_111,y_111one,after = 0)
#end(y_111)
#acf(y_111)
#z2=rho3*x_11[1:57,1:16]
#w2=x_11[2:58,1:16]
#x_121=w2-z2
#x_11one=x_11[1,1:16]*sqrt(1-rho3^2)
#x_111=rbind(x_11one,x_121)
#dim(x_111)
#data3=data.frame(y_111,x_111)
#model3=lm(y_111~.,data = data3)
#summary(model3)
#dwtest(model3)
### Check Heteroscedasticity

library(lmtest)
gqtest(model2)
```

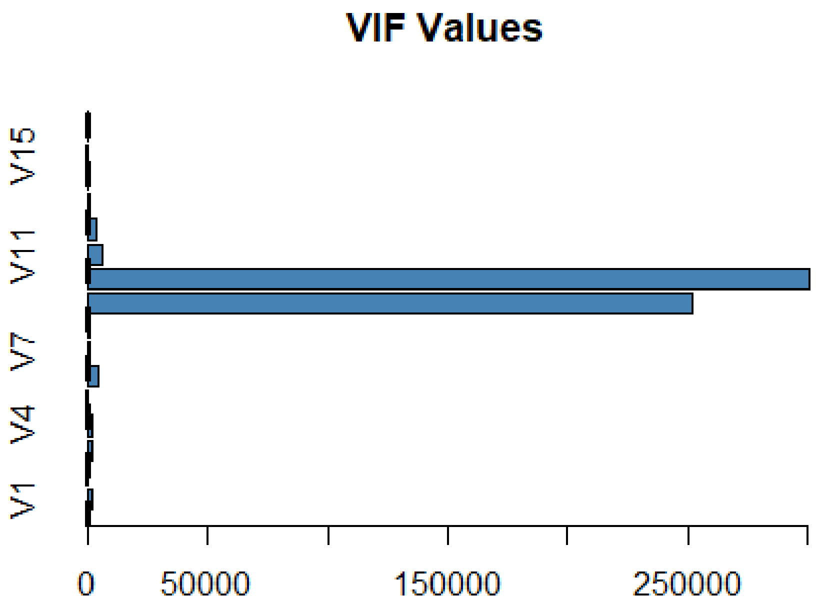
```
## 
##  Goldfeld-Quandt test
## 
## data:  model2
## GQ = 0.42874, df1 = 13, df2 = 12, p-value = 0.928
## alternative hypothesis: variance increases from segment 1 to 2
```

```
#white.test(x_11,y_11)# H0:data is homoscedasticity

### To check multicollinearity
#install.packages('car')
library("car")
```

```
## Loading required package: carData
```

```
car::vif(model2) # The VIF > 5. i.e there is multicollinearity
```

```
##           V1           V2           V3           V4           V5 
## 1.921787e+03 5.139554e+02 2.080581e+03 1.503807e+03 4.998877e+02 
##           V6           V7           V8           V9          V10 
## 4.717581e+03 1.267128e+03 8.010229e+02 2.520421e+05 3.009495e+05 
##          V11          V12          V13          V14          V15 
## 5.813778e+03 3.478407e+03 9.893064e+02 3.760357e+00 4.285390e+02
##          V16
## 2.234663e+02

# In above data violate the problem of multicollinearity.
# To remove the multicollinearity we use Principal Component Analysis

#create a vector of VIF values
vif_values <- vif(model2)

#create a horizontal bar chart to display each VIF value
barplot(vif_values, main = "VIF Values", horiz = TRUE, col = "steelblue")

#add a vertical line at 5
abline(v = 5, lwd = 3, lty = 2)
```



```
###########################################################################
### Principal Component Analysis ###
str(data2)

## 'data.frame':    59 obs. of  17 variables:
##  $ y_11: num  27.8 21.7 22.1 22.2 22.2 ...
##  $ V1  : num  28.4 21.2 21.6 21.6 21.7 ...
##  $ V2  : num  8.9 6.69 6.93 6.93 6.93 ...
##  $ V3  : num  26.8 20.1 20.5 20.6 20.7 ...
##  $ V4  : num  7.5 5.73 5.92 6.06 6.11 ...
##  $ V5  : num  9.97 7.47 7.64 7.75 7.8 ...
##  $ V6  : num  10.28 7.73 7.92 7.98 8.03 ...
##  $ V7  : num  10.14 7.6 7.78 7.8 7.82 ...
##  $ V8  : num  9.27 6.98 7.22 7.29 7.37 ...
##  $ V9  : num  12.65 9.47 9.67 9.71 9.77 ...
##  $ V10 : num  12.6 9.42 9.62 9.67 9.73 ...
##  $ V11 : num  8.91 6.65 6.77 6.8 6.85 ...
##  $ V12 : num  28.8 21.5 22 22 22 ...
##  $ V13 : num  26.1 19.6 20.1 20.3 20.3 ...
##  $ V14 : num  8.54 6.21 6.24 6.07 6.65 ...
##  $ V15 : num  25.9 19.3 19.7 19.8 19.7 ...
##  $ V16 : num  26.4 19.6 20 20 20.1 ...
```

```
R=round(cor(data2[,-1]),2)
R
```

```
##       V1   V2   V3   V4   V5   V6   V7   V8   V9  V10  V11  V12  V13  V14
## V1  1.00 0.53 0.92 0.39 0.57 0.54 0.52 0.47 0.67 0.67 0.69 0.98 0.90 0.44
## V2  0.53 1.00 0.81 0.98 0.98 0.99 0.99 0.99 0.98 0.98 0.95 0.69 0.83 0.80
## V3  0.92 0.81 1.00 0.72 0.84 0.82 0.81 0.77 0.90 0.90 0.90 0.98 1.00 0.68
## V4  0.39 0.98 0.72 1.00 0.95 0.97 0.98 0.97 0.93 0.92 0.87 0.57 0.74 0.79
## V5  0.57 0.98 0.84 0.95 1.00 1.00 0.99 0.98 0.99 0.99 0.98 0.73 0.84 0.82
## V6  0.54 0.99 0.82 0.97 1.00 1.00 1.00 0.99 0.99 0.99 0.96 0.71 0.83 0.82
## V7  0.52 0.99 0.81 0.98 0.99 1.00 1.00 0.99 0.98 0.98 0.95 0.69 0.82 0.82
## V8  0.47 0.99 0.77 0.97 0.98 0.99 0.99 1.00 0.97 0.96 0.95 0.64 0.78 0.80
## V9  0.67 0.98 0.90 0.93 0.99 0.99 0.98 0.97 1.00 1.00 0.99 0.81 0.90 0.81
## V10 0.67 0.98 0.90 0.92 0.99 0.99 0.98 0.96 1.00 1.00 0.99 0.81 0.91 0.80
## V11 0.69 0.95 0.90 0.87 0.98 0.96 0.95 0.95 0.99 0.99 1.00 0.82 0.89 0.78
## V12 0.98 0.69 0.98 0.57 0.73 0.71 0.69 0.64 0.81 0.81 0.82 1.00 0.97 0.57
## V13 0.90 0.83 1.00 0.74 0.84 0.83 0.82 0.78 0.90 0.91 0.89 0.97 1.00 0.68
## V14 0.44 0.80 0.68 0.79 0.82 0.82 0.82 0.80 0.81 0.80 0.78 0.57 0.68 1.00
## V15 0.78 0.92 0.96 0.86 0.94 0.93 0.93 0.89 0.97 0.97 0.95 0.89 0.96 0.78
## V16 0.80 0.90 0.97 0.83 0.93 0.92 0.91 0.87 0.96 0.96 0.95 0.91 0.97 0.77
##      V15  V16
## V1  0.78 0.80
## V2  0.92 0.90
## V3  0.96 0.97
## V4  0.86 0.83
## V5  0.94 0.93
## V6  0.93 0.92
## V7  0.93 0.91
## V8  0.89 0.87
## V9  0.97 0.96
## V10 0.97 0.96
## V11 0.95 0.95
## V12 0.89 0.91
## V13 0.96 0.97
## V14 0.78 0.77
## V15 1.00 0.99
## V16 0.99 1.00
```

```
a=as.matrix((data2[,-1]))
head(a)
```

```
##         V1       V2       V3       V4       V5        V6        V7
## 1 28.41648 8.901867 26.83137 7.504736 9.967638 10.279682 10.136787
## 2 21.19137 6.694220 20.09044 5.729967 7.467530  7.728236  7.601237
## 3 21.59932 6.925384 20.54287 5.922852 7.644968  7.917646  7.775695
## 4 21.62791 6.925032 20.61553 6.060823 7.750806  7.975255  7.797863
## 5 21.70954 6.933306 20.65903 6.105122 7.797439  8.028052  7.817210
## 6 21.56899 7.040979 20.65161 6.183656 7.842963  8.016074  7.840125
##         V8        V9       V10      V11      V12      V13      V14
## 1 9.270525 12.654857 12.598189 8.905399 28.82103 26.14988 8.541110
## 2 6.975190  9.466332  9.422955 6.646366 21.50911 19.57376 6.205773
## 3 7.219916  9.669392  9.623311 6.774943 21.95184 20.13573 6.235237
## 4 7.293756  9.713977  9.668113 6.803638 21.98513 20.29207 6.068567
## 5 7.369007  9.772946  9.727284 6.847047 22.03346 20.28299 6.645792
## 6 7.378075  9.716321  9.664423 6.766672 22.01921 20.36700 6.219329
##        V15      V16
## 1 25.86272 26.37429
## 2 19.28430 19.56499
## 3 19.66877 20.02866
## 4 19.75101 20.04995
## 5 19.66170 20.07477
## 6 19.53202 19.94914
```

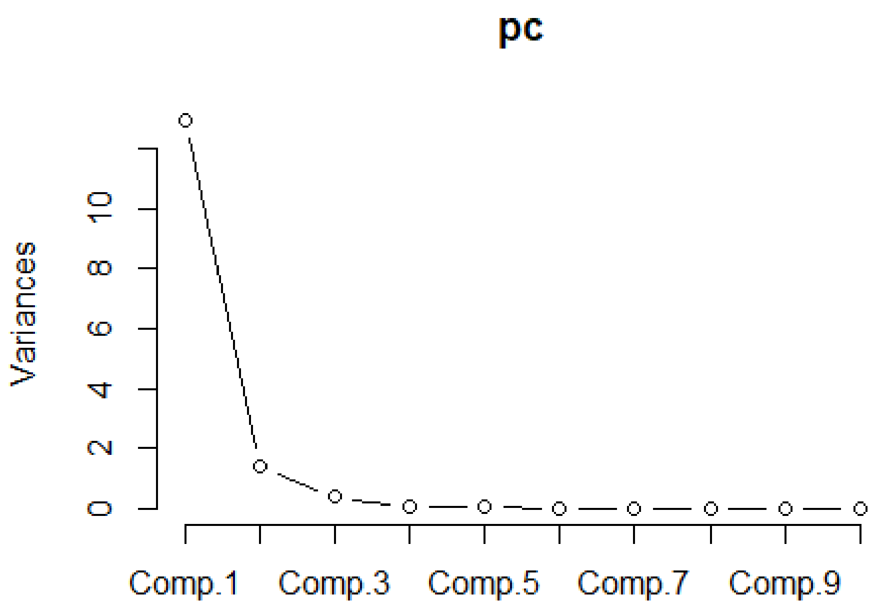
```
dim(a)
```

```
## [1] 59 16
```

```
pc<-princomp(a)
summary(pc)
```

```
## Importance of components:
##                           Comp.1     Comp.2     Comp.3      Comp.4
## Standard deviation     3.597098 1.17756593 0.60614159 0.255427817
## Proportion of Variance 0.872458 0.09349975 0.02477354 0.004399227
## Cumulative Proportion  0.872458 0.96595777 0.99073131 0.995130541
##                            Comp.5       Comp.6       Comp.7       Comp.8
## Standard deviation     0.22028116 0.1041100962 0.0655955129 0.0562893554
## Proportion of Variance 0.00327186 0.0007308456 0.0002901271 0.0002136449
## Cumulative Proportion  0.99840240 0.9991332460 0.9994233730 0.9996370179
##                              Comp.9      Comp.10      Comp.11      Comp.12
## Standard deviation     0.0489055408 3.633910e-02 3.022681e-02 1.851103e-02
## Proportion of Variance 0.0001612709 8.904065e-05 6.160622e-05 2.310474e-05
## Cumulative Proportion  0.9997982888 9.998873e-01 9.999489e-01 9.999720e-01
##                             Comp.13      Comp.14      Comp.15      Comp.16
## Standard deviation     1.699034e-02 9.620437e-03 5.679942e-03 1.082555e-03
## Proportion of Variance 1.946454e-05 6.240647e-06 2.175343e-06 7.902055e-08
## Cumulative Proportion  9.999915e-01 9.999977e-01 9.999999e-01 1.000000e+00
```

```
screeplot(pc,type = "line")
```



```
E=eigen(R)
head(E)

## $values
##  [1] 13.9987996549  1.5403982482  0.3119358327  0.0971847981  0.0397756212
##  [6]  0.0165550834  0.0141048632  0.0110459094  0.0068984524  0.0038822155
## [11] -0.0004635094 -0.0031041970 -0.0045205603 -0.0085686519 -0.0107496968
## [16] -0.0131740636
##
## $vectors
##             [,1]        [,2]          [,3]        [,4]         [,5]
##  [1,] -0.1921632 -0.55720355 -0.0396417912 -0.07858384  0.303898265
##  [2,] -0.2574918  0.19198691  0.1400755057  0.12746051  0.396361611
##  [3,] -0.2491186 -0.29124168 -0.0128930092  0.12854694  0.027481212
##  [4,] -0.2427372  0.30308825  0.1213999577  0.54209218  0.176875644
##  [5,] -0.2610823  0.15218781  0.0648879816 -0.23368118 -0.278817812
##  [6,] -0.2599578  0.18328960  0.0839547666 -0.04670208 -0.139524379
##  [7,] -0.2582351  0.20063109  0.0812136362  0.09558472 -0.140914447
##  [8,] -0.2523807  0.24531022  0.1394296089 -0.15291845  0.290220537
##  [9,] -0.2663439  0.05852127  0.0672258416 -0.14621670  0.109878171
## [10,] -0.2660131  0.05200004  0.0898604592 -0.14986887  0.001118251
## [11,] -0.2621290  0.02083094  0.0918567130 -0.61047405  0.046600373
## [12,] -0.2268438 -0.42616878  0.0008608973 -0.04724351  0.047025720
## [13,] -0.2499500 -0.27052544  0.0160865204  0.32405962  0.142112193
## [14,] -0.2178543  0.18560549 -0.9507978762 -0.02962061  0.107615351
## [15,] -0.2633290 -0.08829330 -0.0154959458  0.20862203 -0.382835261
## [16,] -0.2617573 -0.12322040 -0.0325570816  0.09856973 -0.567815630
##               [,6]        [,7]         [,8]         [,9]        [,10]
##  [1,] -0.303214781 -0.17107008  0.079689232  0.039609928  0.158152208
##  [2,]  0.212794132  0.15549765  0.345700093  0.043949327  0.484623676
##  [3,]  0.189156128 -0.16307997 -0.270975629  0.342528507 -0.272320131
##  [4,] -0.222516967 -0.08555259 -0.216803457  0.342948812  0.048696813
##  [5,] -0.182078724  0.07993500 -0.100842494  0.547888961 -0.073458264
##  [6,] -0.357493212  0.31663416 -0.123962600 -0.218545097 -0.008190975
##  [7,] -0.213859839 -0.13271933  0.033605015 -0.340270260 -0.371656212
##  [8,]  0.274204178 -0.35171960 -0.327171132 -0.405624488 -0.124906350
##  [9,] -0.208384890 -0.11222233 -0.033548021 -0.057590211  0.317248322
## [10,]  0.003941390  0.42613361  0.458144380 -0.020046803 -0.273296904
## [11,]  0.272463762 -0.10991576 -0.001078605  0.211039774 -0.035139801
## [12,] -0.329081473  0.13354986 -0.160517338 -0.234586892  0.023934311
## [13,]  0.400809619  0.34506038  0.001185226 -0.043591586 -0.314843251
## [14,] -0.002385519  0.02298889  0.002040845 -0.006815051 -0.009371329
## [15,]  0.033880565 -0.54424967  0.551035407 -0.037173508 -0.009884403
## [16,]  0.328382010  0.15738389 -0.282079861 -0.163354377  0.477438840
##              [,11]        [,12]       [,13]        [,14]        [,15]
##  [1,]  0.074975748 -0.119294947  0.56767137  0.145554147  0.186702224
##  [2,]  0.172837471 -0.240053109 -0.02290646 -0.223950033 -0.326055455
##  [3,]  0.020267890  0.350690483  0.08962624 -0.476874517 -0.378521214
##  [4,]  0.127148946  0.048038308 -0.09362593  0.011540213  0.435333743
##  [5,] -0.184975659 -0.438776126  0.06946567  0.288182577 -0.319038173
##  [6,] -0.344078530  0.022378077  0.17104152 -0.405116979  0.232639885
##  [7,]  0.634421564 -0.008724555  0.10611446  0.186871890 -0.221046046
##  [8,] -0.321667545 -0.223346020  0.15971628 -0.003127668 -0.054657017
##  [9,] -0.211626857  0.633766348 -0.21079608  0.425063270 -0.216608410
## [10,] -0.032970409  0.263583196  0.17878753 -0.080215264  0.017530404
## [11,]  0.307100802  0.042529130 -0.23826044 -0.069392120  0.474564254
## [12,]  0.068699090 -0.267312287 -0.61196367 -0.168851344 -0.101712486
## [13,] -0.193892288 -0.078515989 -0.12600275  0.432571335  0.136128105
## [14,]  0.008907204 -0.018120421 -0.01597948 -0.008796797  0.007143505
## [15,] -0.271905893 -0.086630403 -0.15701305 -0.096241433  0.113449586
## [16,]  0.181250187  0.038320864  0.20278783  0.046053981  0.063600683
##             [,16]
##  [1,]  0.01461777
##  [2,]  0.18335566
##  [3,]  0.07918588
##  [4,] -0.26195081
##  [5,] -0.02068231
##  [6,]  0.46096512
##  [7,]  0.20364510
##  [8,] -0.28486125
##  [9,]  0.08988299
## [10,] -0.56580166
## [11,]  0.18044594
## [12,] -0.25582596
## [13,]  0.30470364
## [14,] -0.01228742
## [15,]  0.04285852
## [16,] -0.18626395
```

```
end(E)
```

```
## [1] 2 1
```

```
EV=round(E$values,3)
EV
```

```
##  [1] 13.999  1.540  0.312  0.097  0.040  0.017  0.014  0.011  0.007  0.004
## [11]  0.000 -0.003 -0.005 -0.009 -0.011 -0.013
```

```
end(EV)
```

```
## [1] 16  1
```

```
for(i in 1:3)
{
  var=(EV/(sum(EV)))*100
}
var
```

```
##  [1] 87.49375  9.62500  1.95000  0.60625  0.25000  0.10625  0.08750
##  [8]  0.06875  0.04375  0.02500  0.00000 -0.01875 -0.03125 -0.05625
## [15] -0.06875 -0.08125
```

```
CV=cumsum(var)
CV
```

```
##  [1]  87.49375  97.11875  99.06875  99.67500  99.92500 100.03125 100.11875
##  [8] 100.18750 100.23125 100.25625 100.25625 100.23750 100.20625 100.15000
## [15] 100.08125 100.00000
```

```
wt=E$vectors
head(wt)
```

```
##            [,1]       [,2]        [,3]        [,4]        [,5]       [,6]
## [1,] -0.1921632 -0.5572035 -0.03964179 -0.07858384  0.30389826 -0.3032148
## [2,] -0.2574918  0.1919869  0.14007551  0.12746051  0.39636161  0.2127941
## [3,] -0.2491186 -0.2912417 -0.01289301  0.12854694  0.02748121  0.1891561
## [4,] -0.2427372  0.3030882  0.12139996  0.54209218  0.17687564 -0.2225170
## [5,] -0.2610823  0.1521878  0.06488798 -0.23368118 -0.27881781 -0.1820787
## [6,] -0.2599578  0.1832896  0.08395477 -0.04670208 -0.13952438 -0.3574932
##             [,7]        [,8]        [,9]        [,10]       [,11]
## [1,] -0.17107008  0.07968923  0.03960993  0.158152208  0.07497575
## [2,]  0.15549765  0.34570009  0.04394933  0.484623676  0.17283747
## [3,] -0.16307997 -0.27097563  0.34252851 -0.272320131  0.02026789
## [4,] -0.08555259 -0.21680346  0.34294881  0.048696813  0.12714895
## [5,]  0.07993500 -0.10084249  0.54788896 -0.073458264 -0.18497566
## [6,]  0.31663416 -0.12396260 -0.21854510 -0.008190975 -0.34407853
##            [,12]       [,13]       [,14]      [,15]       [,16]
## [1,] -0.11929495  0.56767137  0.14555415  0.1867022  0.01461777
## [2,] -0.24005311 -0.02290646 -0.22395003 -0.3260555  0.18335566
## [3,]  0.35069048  0.08962624 -0.47687452 -0.3785212  0.07918588
## [4,]  0.04803831 -0.09362593  0.01154021  0.4353337 -0.26195081
## [5,] -0.43877613  0.06946567  0.28818258 -0.3190382 -0.02068231
## [6,]  0.02237808  0.17104152 -0.40511698  0.2326399  0.46096512
```

```
dim(wt)
```

```
## [1] 16 16
```

```
comp=c(1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16)
comp
```

```
##  [1]  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16
```

```
df=data.frame(comp,round(EV,3),round(var,3),round(CV,3))
df
```

```
##    comp round.EV..3. round.var..3. round.CV..3.
## 1     1       13.999        87.494       87.494
## 2     2        1.540         9.625       97.119
## 3     3        0.312         1.950       99.069
## 4     4        0.097         0.606       99.675
## 5     5        0.040         0.250       99.925
## 6     6        0.017         0.106      100.031
## 7     7        0.014         0.088      100.119
## 8     8        0.011         0.069      100.188
## 9     9        0.007         0.044      100.231
## 10   10        0.004         0.025      100.256
## 11   11        0.000         0.000      100.256
## 12   12       -0.003        -0.019      100.238
## 13   13       -0.005        -0.031      100.206
## 14   14       -0.009        -0.056      100.150
## 15   15       -0.011        -0.069      100.081
## 16   16       -0.013        -0.081      100.000
```

```
### THe above first five princ comp explains more variability then we select the first five PCs.
## Using Eigenvalue criteria (EV>0.7)
wt=data.frame(wt)
head(wt)
```

```
##           X1         X2          X3          X4          X5         X6
## 1 -0.1921632 -0.5572035 -0.03964179 -0.07858384  0.30389826 -0.3032148
## 2 -0.2574918  0.1919869  0.14007551  0.12746051  0.39636161  0.2127941
## 3 -0.2491186 -0.2912417 -0.01289301  0.12854694  0.02748121  0.1891561
## 4 -0.2427372  0.3030882  0.12139996  0.54209218  0.17687564 -0.2225170
## 5 -0.2610823  0.1521878  0.06488798 -0.23368118 -0.27881781 -0.1820787
## 6 -0.2599578  0.1832896  0.08395477 -0.04670208 -0.13952438 -0.3574932
##            X7          X8          X9         X10         X11         X12
## 1 -0.17107008  0.07968923  0.03960993  0.158152208  0.07497575 -0.11929495
## 2  0.15549765  0.34570009  0.04394933  0.484623676  0.17283747 -0.24005311
## 3 -0.16307997 -0.27097563  0.34252851 -0.272320131  0.02026789  0.35069048
## 4 -0.08555259 -0.21680346  0.34294881  0.048696813  0.12714895  0.04803831
## 5  0.07993500 -0.10084249  0.54788896 -0.073458264 -0.18497566 -0.43877613
## 6  0.31663416 -0.12396260 -0.21854510 -0.008190975 -0.34407853  0.02237808
##           X13         X14        X15         X16
## 1  0.56767137  0.14555415  0.1867022  0.01461777
## 2 -0.02290646 -0.22395003 -0.3260555  0.18335566
## 3  0.08962624 -0.47687452 -0.3785212  0.07918588
## 4 -0.09362593  0.01154021  0.4353337 -0.26195081
## 5  0.06946567  0.28818258 -0.3190382 -0.02068231
## 6  0.17104152 -0.40511698  0.2326399  0.46096512
```

```
head(a)
```

```
##         V1       V2       V3       V4       V5        V6        V7
## 1 28.41648 8.901867 26.83137 7.504736 9.967638 10.279682 10.136787
## 2 21.19137 6.694220 20.09044 5.729967 7.467530  7.728236  7.601237
## 3 21.59932 6.925384 20.54287 5.922852 7.644968  7.917646  7.775695
## 4 21.62791 6.925032 20.61553 6.060823 7.750806  7.975255  7.797863
## 5 21.70954 6.933306 20.65903 6.105122 7.797439  8.028052  7.817210
## 6 21.56899 7.040979 20.65161 6.183656 7.842963  8.016074  7.840125
##         V8        V9       V10      V11      V12      V13      V14
## 1 9.270525 12.654857 12.598189 8.905399 28.82103 26.14988 8.541110
## 2 6.975190  9.466332  9.422955 6.646366 21.50911 19.57376 6.205773
## 3 7.219916  9.669392  9.623311 6.774943 21.95184 20.13573 6.235237
## 4 7.293756  9.713977  9.668113 6.803638 21.98513 20.29207 6.068567
## 5 7.369007  9.772946  9.727284 6.847047 22.03346 20.28299 6.645792
## 6 7.378075  9.716321  9.664423 6.766672 22.01921 20.36700 6.219329
##        V15      V16
## 1 25.86272 26.37429
## 2 19.28430 19.56499
## 3 19.66877 20.02866
## 4 19.75101 20.04995
## 5 19.66170 20.07477
## 6 19.53202 19.94914
```

```
dim(a)
```

```
## [1] 59 16
```

```
Z=a%*%as.matrix(wt[1:16,1:2])
head(Z)
```

```
##          X1        X2
## 1 -64.19568 -33.67923
## 2 -47.96681 -25.08903
## 3 -49.07764 -25.58985
## 4 -49.27047 -25.62772
## 5 -49.52001 -25.53049
## 6 -49.35987 -25.47112
```

```
dim(Z)

## [1] 59  2

Y=data2[,1]
end(Y)

## [1] 59  1

data3=data.frame(Y,Z)
str(data3)

## 'data.frame':    59 obs. of  3 variables:
##  $ Y : num  27.8 21.7 22.1 22.2 22.2 ...
##  $ X1: num  -64.2 -48 -49.1 -49.3 -49.5 ...
##  $ X2: num  -33.7 -25.1 -25.6 -25.6 -25.5 ...

head(data3)

##          Y        X1        X2
## 1 27.80198 -64.19568 -33.67923
## 2 21.67609 -47.96681 -25.08903
## 3 22.13241 -49.07764 -25.58985
## 4 22.18339 -49.27047 -25.62772
## 5 22.23996 -49.52001 -25.53049
## 6 22.19470 -49.35987 -25.47112

#data2=data.frame(Y,Z123)

lm.fit=lm(Y~.,data=data3)
lm.fit

##
## Call:
## lm(formula = Y ~ ., data = data3)
##
## Coefficients:
## (Intercept)           X1           X2
##      3.9627      -0.2293      -0.2713

sum=summary(lm.fit)   ## from here we can see the significance of the PC's
sum

##
## Call:
## lm(formula = Y ~ ., data = data3)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.09933 -0.01142  0.00395  0.01905  0.04095
##
## Coefficients:
```

```
##                Estimate Std. Error t value Pr(>|t|)    
## (Intercept)  3.962663   0.093246    42.5   <2e-16 ***
## X1          -0.229284   0.001049  -218.5   <2e-16 ***
## X2          -0.271277   0.003143   -86.3   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.02832 on 56 degrees of freedom
## Multiple R-squared:  0.9991, Adjusted R-squared:  0.999 
## F-statistic: 2.953e+04 on 2 and 56 DF,  p-value: < 2.2e-16
```

```
betas=t(as.vector(sum$coefficients[,1]))

intercept=betas[1];intercept

## [1] 3.962663

beta=betas[-1];beta

## [1] -0.2292841 -0.2712765

##### to find the final coefficient of real variables
beta=as.matrix(beta)
betan=as.matrix(wt[,1:2])%*%(beta)
betan
```

Variables	Co-efficient
Intercept	3.962663
X1	0.1952162068
X2	0.0069572380
X3	0.1361259687
X4	-0.0265649391
X5	0.0185770422
X6	0.0098820345
X7	0.0047826913
X8	-0.0086800145
X9	0.0451929684
X10	0.0468861877
X11	0.0544510661
X12	0.1676212602
X13	0.1306967470
X14	-0.0003998758
X15	0.0843290563
X16	0.0934435963