practical_4

Soumarya Basak

16/05/2022

Problem 1

a

Importing the data

```
library(readr)
df<- read.csv("C:\\Users\\souma\\Dropbox\\Mstat_CU\\Sem 2\\Regression_analysis_1\\Data Sets\\exp_inc_wel
colnames(df)<- c("y","x1","x2")
colnames(df)
```

```
## [1] "y"  "x1" "x2"
```

Fit the model

```
model1<- lm(y~ x1+x2,data=df)
summary(model1)
```

```
##
## Call:
## lm(formula = y ~ x1 + x2, data = df)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -22.637  -5.657   2.974   8.076   9.237
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 18.165539  11.360774   1.599    0.154
## x1           0.632614   1.384489   0.457    0.662
## x2          -0.009891   0.135714  -0.073    0.944
##
## Residual standard error: 11.45 on 7 degrees of freedom
## Multiple R-squared:  0.9104, Adjusted R-squared:  0.8848
## F-statistic: 35.57 on 2 and 7 DF,  p-value: 0.0002151
```

All the coefficients are insignificant, but the adj.r-square is very high,so there may be some multicollinearity.

Correlation Matrix

```
cor(df[,-1])
```

```
##           x1        x2
```

```
## x1 1.0000000 0.9989624
## x2 0.9989624 1.0000000
```

The correlation between `x1` and `x2` is very high, so there is a high multicollinearity.

VIF

```
library(car)
```

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

```
vif(model1)
```

```
##       x1       x2
## 482.1275 482.1275
```

Thumb Rule : The Vif's are quite large (» 10), so tehere is a multicollinearity in between the variables.

CN

```
# For the  Design Matrix
d<-model.matrix(model1)
d_t<- t(d) %*% d # X'X


# Finding the eigen values of matrix
lambda<-eigen(d_t)$values
lambda
```

```
## [1] 3.403227e+07 6.795204e+01 1.016483e+00
```

```
# CN
cn<- sqrt(max(lambda)/min(lambda))
cn
```

```
## [1] 5786.226
```

The CN is too large, so there there is multicolliearity in the data.

Remadial Measures

b: adding two obsns. . . . . .

adding 11th and 12th observation

```
a11<- c(160,120,3000)  # 11th obns
a12<- c(85,255,920)    # 12th obsn
df1<- rbind(df,a11,a12)
dim(df1)
```

```
## [1] 12  3
```

Fit the model in new data

```
model2<- lm(y~. , data=df1)
summary(model2)
```

```
##
```

```
## Call:
## lm(formula = y ~ ., data = df1)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -22.561  -5.413   2.400   6.428   9.639
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 17.852113  10.173655   1.755   0.1132
## x1           0.100630   0.054652   1.841   0.0987 .
## x2           0.042541   0.004735   8.984 8.66e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 10.28 on 9 degrees of freedom
## Multiple R-squared:  0.9289, Adjusted R-squared:  0.9131
## F-statistic:  58.8 on 2 and 9 DF,  p-value: 6.809e-06
```

Check for multicollinearity

```
vif(model2)
```

```
##       x1       x2
## 1.203552 1.203552
```

```
# design matrix
d1<- model.matrix(model2)
dd<- t(d1) %*% d1

# eigen value of matrix
lambda1<-eigen(dd)$values

#CN
cn2<- sqrt(max(lambda1)/min(lambda1))
cn2
```

```
## [1] 6554.492
```

```
# correlation matrix
cor(df1[,-1])
```

```
##           x1        x2
## x1 1.0000000 0.4112492
## x2 0.4112492 1.0000000
```

Comment: The VIF becomes low, the correlation between the variables becomes low, but the CN is very large.

Comment: As the Vifs are low and correlation betweeen the variables is not such high their might not be the problem of multicollinearity.
SO we need to re fit the model with out insignificant parameters

```
model_2.1<- lm(y~x1+ x2 -1, data=df1)
summary(model_2.1)
```

```
##
## Call:
```

```
## lm(formula = y ~ x1 + x2 - 1, data = df1)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -17.882  -1.589   2.537   5.871  19.980
##
## Coefficients:
##    Estimate Std. Error t value Pr(>|t|)
## x1 0.157099   0.048548   3.236  0.00893 **
## x2 0.046237   0.004661   9.921 1.71e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 11.3 on 10 degrees of freedom
## Multiple R-squared:  0.9921, Adjusted R-squared:  0.9905
## F-statistic: 624.3 on 2 and 10 DF,  p-value: 3.167e-11
```

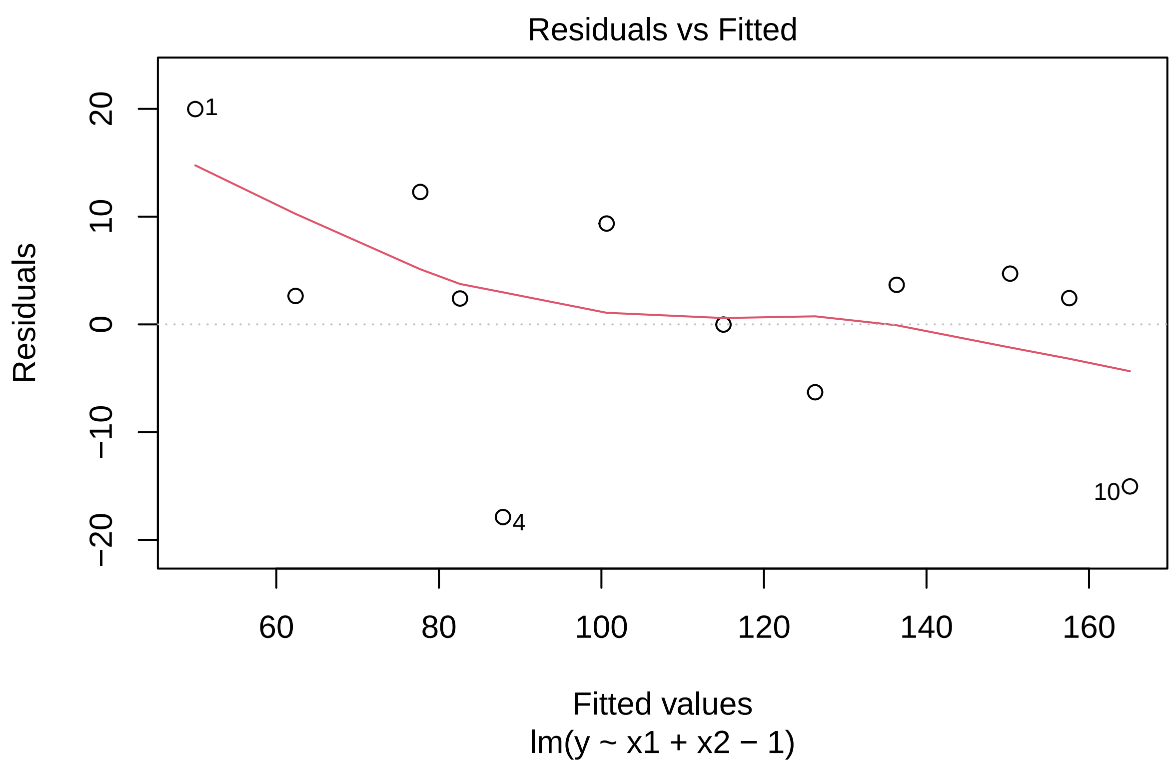
```
### Further checking for multicollinearity
vif(model_2.1)
```

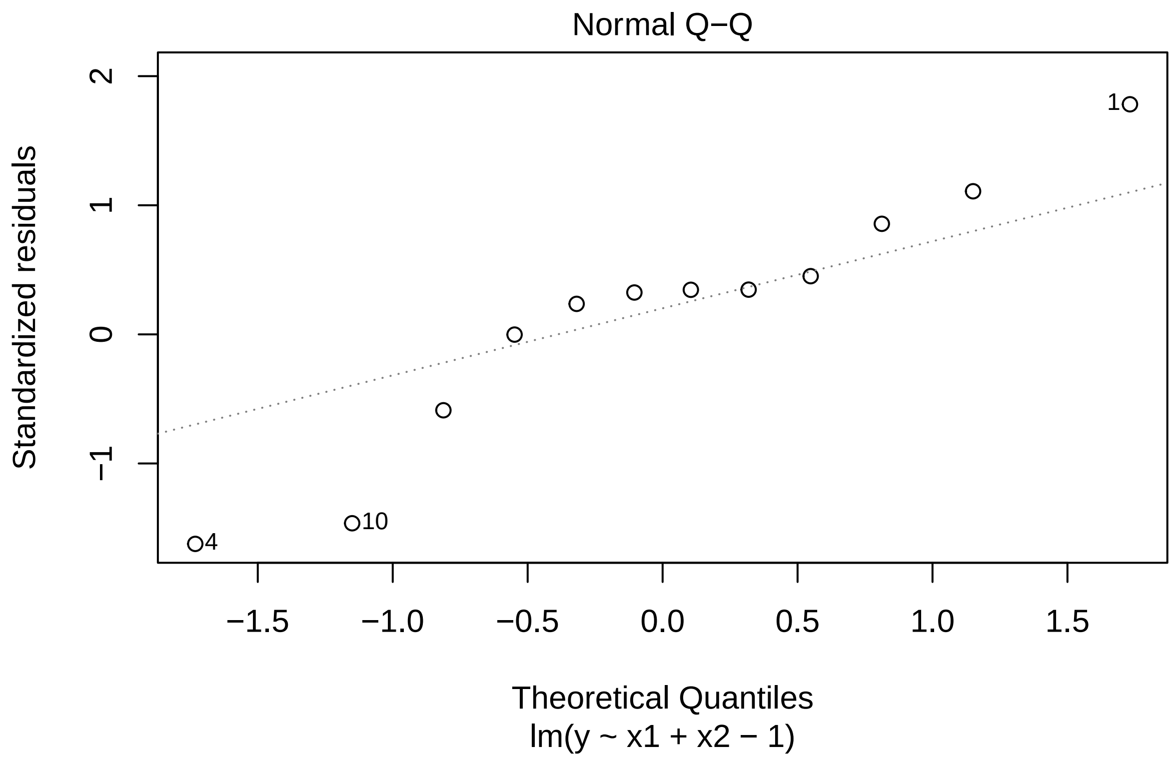
```
## Warning in vif.default(model_2.1): No intercept: vifs may not be sensible.
```

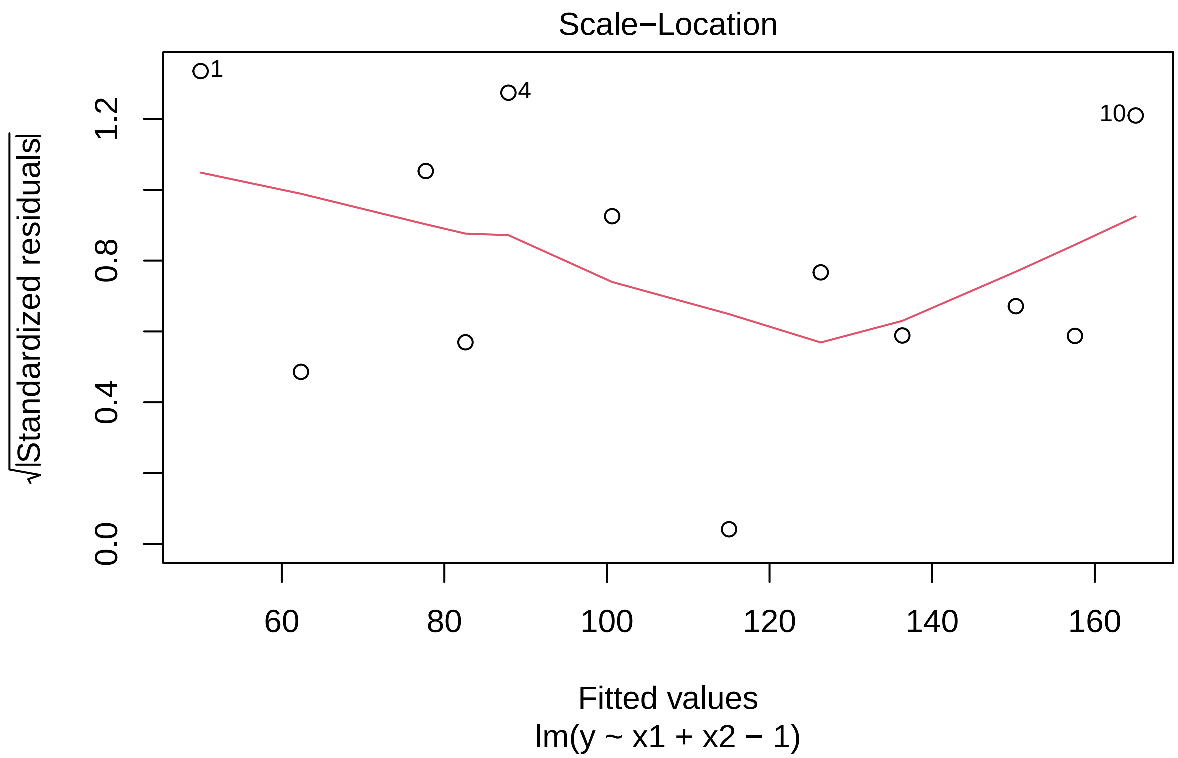
```
##       x1       x2
## 7.404982 7.404982
```

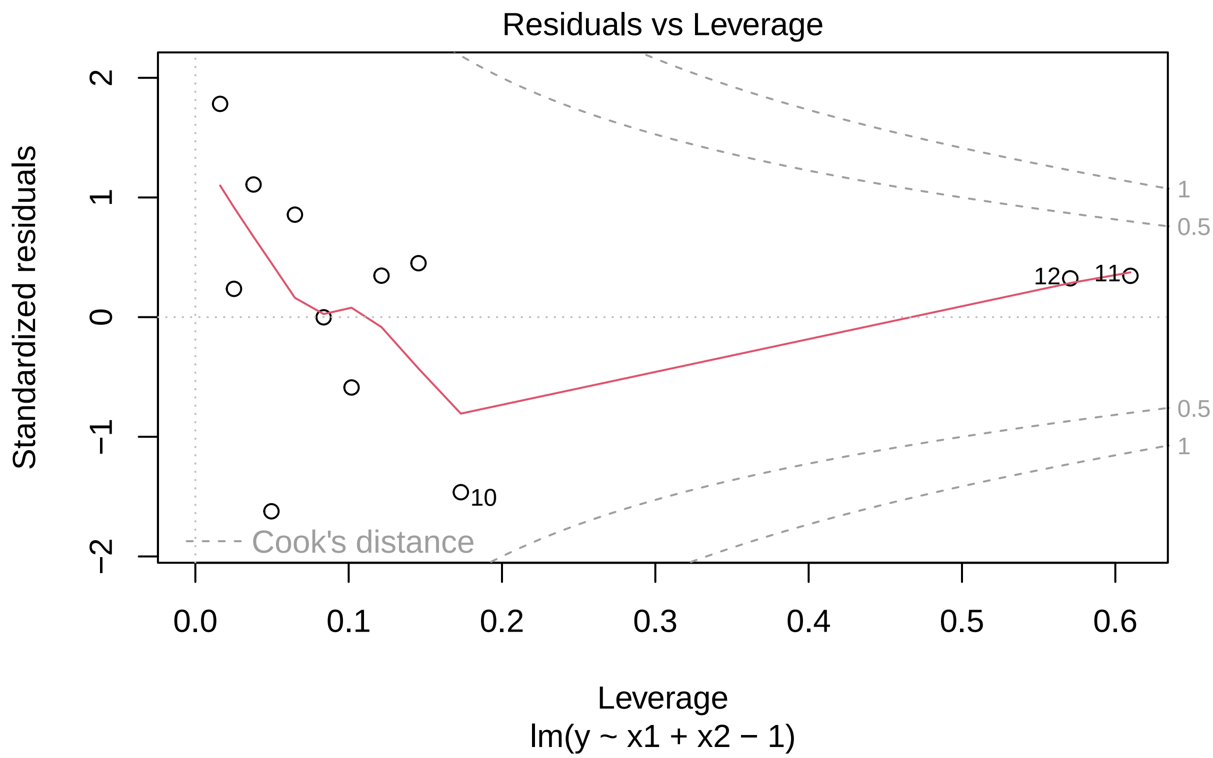
The vif is still low, so no multicolineasrity is there.

```
plot(model_2.1)
```




Normal Q−Q
Standardized residuals
2
1
0
−1
1
10
4
−1.5
−1.0
−0.5
0.0
0.5
1.0
1.5
Theoretical Quantiles
lm(y ~ x1 + x2 − 1)


Scale−Location
√|Standardized residuals|
0.0
0.4
0.8
1.2
60
80
100
120
140
160
1
4
10
Fitted values
lm(y ~ x1 + x2 − 1)



Comment: So, the last model is free from multicollinearity with all the significant parameters with a good Adj R^2 value.
so this model is good enough to work.

c: PCA

PCA

Here we will work with the first data set

```
pca1<- princomp(df[,-1])
summary(pca1)
```

```
## Importance of components:
##                           Comp.1       Comp.2
## Standard deviation     588.8342894 2.603776e+00
## Proportion of Variance   0.9999804 1.955297e-05
## Cumulative Proportion    0.9999804 1.000000e+00
```

```
prcomp(df[,-1])
```

```
## Standard deviations (1, .., p=2):
## [1] 620.685840   2.744621
##
## Rotation (n x k) = (2 x 2):
##           PC1         PC2
## x1 0.09745891 -0.99523955
## x2 0.99523955  0.09745891
```

so we will use the PC1, component

```
df$z1<- (0.09745891*df$x1+0.99523955*df$x2)
model_pca<- lm(y~z1 ,data=df )
summary(model_pca)
```

```
##
## Call:
## lm(formula = y ~ z1, data = df)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -22.106  -5.275   2.635   7.375  10.310
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 17.921019  10.769705   1.664    0.135
## z1           0.051810   0.005838   8.875 2.05e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 10.87 on 8 degrees of freedom
## Multiple R-squared:  0.9078, Adjusted R-squared:  0.8963
## F-statistic: 78.76 on 1 and 8 DF,  p-value: 2.054e-05
```

```
model_pca_u<- lm(y~z1-1 ,data=df )
summary(model_pca_u)
```

```
##
## Call:
## lm(formula = y ~ z1 - 1, data = df)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -17.367  -4.347   4.083   8.838  20.336
##
## Coefficients:
##    Estimate Std. Error t value Pr(>|t|)
## z1 0.061016   0.002038   29.93 2.53e-10 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 11.89 on 9 degrees of freedom
## Multiple R-squared:  0.9901, Adjusted R-squared:  0.989
## F-statistic: 896.1 on 1 and 9 DF,  p-value: 2.532e-10
```

The model becomes

$$y = 0.061016 \times Z_1 = 0.061016 \times (0.09745891 * x_1 + 0.9952395 * 5x_2)$$

Sir's algorithm

```
#design matrix
d
```

```
##   (Intercept) x1  x2
## 1           1 80 810
```

```
## 2            1 100 1009
## 3            1 120 1273
## 4            1 140 1425
## 5            1 160 1633
## 6            1 180 1876
## 7            1 200 2052
## 8            1 220 2201
## 9            1 240 2435
## 10           1 260 2686
## attr(,"assign")
## [1] 0 1 2
```

```
# X'X
d_t
```

```
##             (Intercept)      x1       x2
## (Intercept)          10    1700    17400
## x1                 1700  322000  3294300
## x2                17400 3294300 33710326
```

```
# eigen values of X'X
l<- eigen(d_t)$values
# eigen vectors of X'X
L<- eigen(d_t)$vectors
L
```

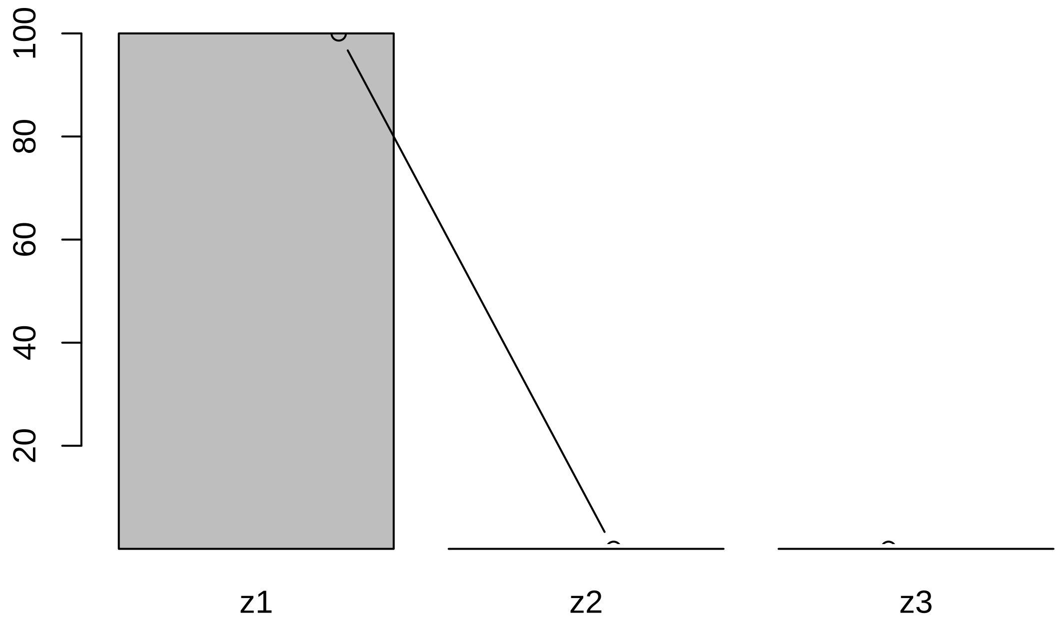
```
##               [,1]         [,2]         [,3]
## [1,] -0.000513714  0.005902328  0.999982449
## [2,] -0.097260635 -0.995241903  0.005824382
## [3,] -0.995258813  0.097255936 -0.001085334
```

```
# note: L is a otthogonal matrix
```

```
## Define Z
z=d %*% L
z
```

```
##          [,1]       [,2]        [,3]
## 1   -813.941 -0.8361417  0.58681260
## 2  -1013.943 -1.3870485  0.48731881
## 3  -1278.636  4.3836805  0.31727832
## 4  -1431.861 -0.7382552  0.26879522
## 5  -1640.820 -0.4138586  0.15953342
## 6  -1884.613  3.3144958  0.01228494
## 7  -2061.724  0.5267025 -0.06224618
## 8  -2211.963 -4.8870011 -0.10747328
## 9  -2446.798 -2.0339501 -0.24495375
## 10 -2698.553  2.4724517 -0.40088491
```

```
## Variability of Z_i
pc_var<- c( var(z[,1]) , var(z[,2]), var(z[,3]) )

## plot of variability explained
barplot((pc_var/sum(pc_var))*100, names.arg = c("z1","z2","z3"),ylim = c(1,100) )
lines((pc_var/sum(pc_var))*100,type = 'b')
```



So the first component explains the most of the variability, so we will take z1 variable for fitting the model.

```
model_p <- lm(df$y~ z[,1])
summary(model_p)
```

```
##
## Call:
## lm(formula = df$y ~ z[, 1])
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -22.106  -5.275   2.635   7.375  10.310
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 17.920992  10.769713   1.664    0.135
## z[, 1]      -0.051810   0.005838  -8.875 2.05e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 10.87 on 8 degrees of freedom
## Multiple R-squared:  0.9078, Adjusted R-squared:  0.8963
## F-statistic: 78.76 on 1 and 8 DF,  p-value: 2.054e-05
```

The intercept is insignificant, so need to drop that

```
model_pu<- lm(df$y~ 0+ z[,1])
summary(model_pu)
```

```
##
## Call:
## lm(formula = df$y ~ 0 + z[, 1])
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -17.367  -4.347   4.083   8.838  20.336
##
## Coefficients:
##         Estimate Std. Error t value Pr(>|t|)
## z[, 1] -0.061016   0.002038  -29.93 2.53e-10 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 11.89 on 9 degrees of freedom
## Multiple R-squared:  0.9901, Adjusted R-squared:  0.989
## F-statistic: 896.1 on 1 and 9 DF,  p-value: 2.532e-10
```

So, the model is

$$y = -0.061016 \times Z_1$$

```
## Finding estimaton of beta
eta<- -0.061016
beta = L[,1] * eta
round(beta,5)
```

```
## [1] 0.00003 0.00593 0.06073
```

The final model

$$y = 0.00003 + 0.00593 \times X_1 + 0.06073 \times X_2$$

Ridge Regression

```
## Removing the z1 variable
df<- df[,-4]

library(glmnet)
```

```
## Loading required package: Matrix
```

```
## Loaded glmnet 4.1-4
```

```
aa<- 10^seq(2, -3, by = -0.1)
ridge_reg = glmnet(df[,-1], df[,1], nlambda = 25, alpha = 0, family = 'gaussian', lambda = aa)
summary(ridge_reg)
```

```
##           Length Class     Mode
## a0         51    -none-    numeric
## beta      102    dgCMatrix S4
## df         51    -none-    numeric
## dim         2    -none-    numeric
## lambda     51    -none-    numeric
## dev.ratio  51    -none-    numeric
## nulldev     1    -none-    numeric
## npasses     1    -none-    numeric
```

```
## jerr         1    -none-    numeric
## offset       1    -none-    logical
## call         7    -none-    call
## nobs         1    -none-    numeric
```

```
cv_ridge <- cv.glmnet(as.matrix(df[,-1]), df[,1], alpha = 0, lambda = aa)
```

```
## Warning: Option grouped=FALSE enforced in cv.glmnet, since < 3 observations per
## fold
```

```
optimal_lambda <- cv_ridge$lambda.min
optimal_lambda
```

```
## [1] 1.995262
```

so, the optimal choice of c is 1.995262

Estimation of Beta

```
#X'X
d_t
```

```
##             (Intercept)      x1       x2
## (Intercept)          10    1700    17400
## x1                 1700  322000  3294300
## x2                17400 3294300 33710326
```

```
# I matrix
I = diag(rep(1,3))

#beta
beta= (solve(d_t +  (optimal_lambda*I))) %*% t(d) %*% df[,1]

beta
```

```
##                    [,1]
## (Intercept) 6.129046444
## x1          0.547649530
## x2          0.004625073
```

The final model

$$y = 6.129046444 + 0.547649530 \times x_1 + 0.004625073 \times x_2$$

Ridge Regression Sir's Algorithm

```
# Choice of C
```

Problem 2

Importing data

```
df_4<- read.csv("C:\\Users\\souma\\Dropbox\\Mstat_CU\\Sem 2\\Regression_analysis_1\\Data Sets\\problem_
colnames(df_4)[1]<-"pt"
```

```
cor(df_4[,-1])
```

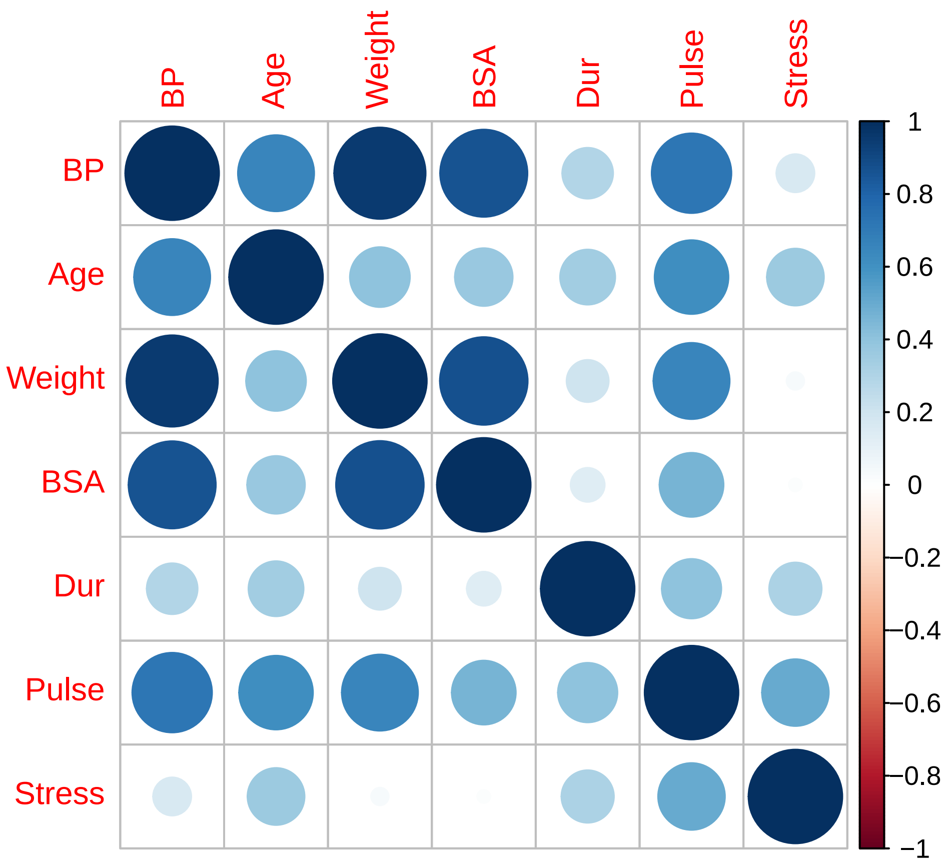
```
##                BP        Age      Weight          BSA         Dur       Pulse       Stress
```

```
## BP     1.0000000 0.6590930 0.95006765 0.86587887 0.2928336 0.7214132 0.16390139
## Age    0.6590930 1.0000000 0.40734926 0.37845460 0.3437921 0.6187643 0.36822369
## Weight 0.9500677 0.4073493 1.00000000 0.87530481 0.2006496 0.6593399 0.03435475
## BSA    0.8658789 0.3784546 0.87530481 1.00000000 0.1305400 0.4648188 0.01844634
## Dur    0.2928336 0.3437921 0.20064959 0.13054001 1.0000000 0.4015144 0.31163982
## Pulse  0.7214132 0.6187643 0.65933987 0.46481881 0.4015144 1.0000000 0.50631008
## Stress 0.1639014 0.3682237 0.03435475 0.01844634 0.3116398 0.5063101 1.00000000
```

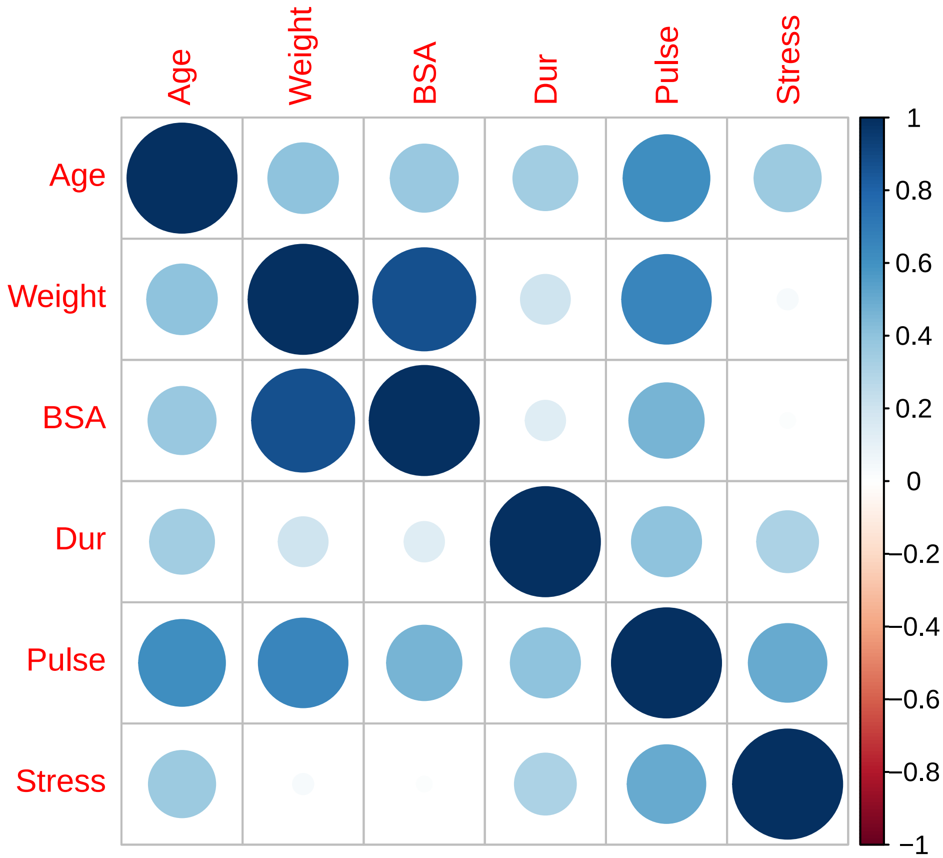
```
library(corrplot)
```

```
## corrplot 0.92 loaded
```

```
corrplot(cor(df_4[,-1]))
```




```
corrplot(cor(df_4[,c(-1,-2)]))
```



From the plot we observe that the `weight` and `BSA` are highly correlated, so, they may cause multicollinearity.

Model fitting

```
m_4<- lm(BP ~Age+Weight+BSA+Dur+Pulse+Stress, data = df_4)
summary(m_4)
```

```
##
## Call:
## lm(formula = BP ~ Age + Weight + BSA + Dur + Pulse + Stress,
##     data = df_4)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.93213 -0.11314  0.03064  0.21834  0.48454
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) -12.870476   2.556650  -5.034 0.000229 ***
## Age           0.703259   0.049606  14.177 2.76e-09 ***
## Weight        0.969920   0.063108  15.369 1.02e-09 ***
## BSA           3.776491   1.580151   2.390 0.032694 *
## Dur           0.068383   0.048441   1.412 0.181534
## Pulse        -0.084485   0.051609  -1.637 0.125594
## Stress        0.005572   0.003412   1.633 0.126491
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
##
## Residual standard error: 0.4072 on 13 degrees of freedom
## Multiple R-squared:  0.9962, Adjusted R-squared:  0.9944
## F-statistic: 560.6 on 6 and 13 DF,  p-value: 6.395e-15
```

```
summary(influence.measures(m_4))
```

```
## Potentially influential observations of
##   lm(formula = BP ~ Age + Weight + BSA + Dur + Pulse + Stress,      data = df_4) :
##
##    dfb.1_ dfb.Age dfb.Wght dfb.BSA dfb.Dur dfb.Puls dfb.Strs dffit cov.r
## 7   0.68  -0.26   -0.61    -0.13    1.14_*  0.71    -0.57    -1.96  0.71
## 11 -0.10   0.33    0.24    -0.24    0.16   -0.35    -0.02    -0.51  2.68_*
## 13 -0.07  -0.04    0.10    -0.11   -0.18    0.02     0.05    -0.25  2.87_*
## 15  0.05  -0.01   -0.04     0.02   -0.03    0.02    -0.03    -0.08  2.92_*
## 16  0.05  -0.20    0.33    -0.23    0.10   -0.27     0.39     0.53  2.64_*
## 19 -0.69   0.66   -0.06     0.41   -1.03_* -0.10    -0.80    -2.14  0.01
## 20 -0.18   0.26   -0.02     0.00   -0.08   -0.01     0.03     0.33  3.86_*
##    cook.d hat
## 7   0.47   0.54
## 11  0.04   0.45
## 13  0.01   0.42
## 15  0.00   0.40
## 16  0.04   0.45
## 19  0.33   0.25
## 20  0.02   0.57
```

Vif

```
library(car)
vif(m_4)
```

```
##      Age   Weight      BSA      Dur    Pulse   Stress
## 1.762807 8.417035 5.328751 1.237309 4.413575 1.834845
```

CN

```
d_matrix<- model.matrix(m_4)
x<- t(d_matrix) %*% d_matrix

l<- eigen(x)$values

cn<- sqrt(max(l)/min(l))
cn
```

```
## [1] 4016.581
```

The CN is too large, so there exists multicollieanrity .

Fit a model without `BSA`

```
m_4.1<- lm(BP ~Age+Weight+Dur+Pulse+Stress, data = df_4)
summary(m_4.1)
```

```
##
```

```
## Call:
## lm(formula = BP ~ Age + Weight + Dur + Pulse + Stress, data = df_4)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.02600 -0.18526 -0.00077  0.21934  0.72533
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) -15.116781   2.748758  -5.499 7.83e-05 ***
## Age           0.731940   0.055646  13.154 2.85e-09 ***
## Weight        1.098958   0.037773  29.093 6.37e-14 ***
## Dur           0.064105   0.055965   1.145   0.2712
## Pulse        -0.137444   0.053885  -2.551   0.0231 *
## Stress        0.007429   0.003841   1.934   0.0736 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.4708 on 14 degrees of freedom
## Multiple R-squared:  0.9945, Adjusted R-squared:  0.9925
## F-statistic: 502.5 on 5 and 14 DF,  p-value: 2.835e-15
```

```r
# ------- Lets calculated vif for the model ---------

vif(m_4.1)
```

```
##      Age   Weight      Dur    Pulse   Stress
## 1.659637 2.256150 1.235620 3.599913 1.739641
```

Fit a model without `Pluse`

```r
m_4.2<- lm(BP ~Age+Weight+Dur+Stress, data = df_4)
summary(m_4.2)
```

```
##
## Call:
## lm(formula = BP ~ Age + Weight + Dur + Stress, data = df_4)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.11359 -0.29586  0.01515  0.27506  0.88674
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) -15.869829   3.195296  -4.967 0.000169 ***
## Age           0.683741   0.061195  11.173 1.14e-08 ***
## Weight        1.034128   0.032672  31.652 3.76e-15 ***
## Dur           0.039889   0.064486   0.619 0.545485
## Stress        0.002184   0.003794   0.576 0.573304
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.5505 on 15 degrees of freedom
## Multiple R-squared:  0.9919, Adjusted R-squared:  0.9897
## F-statistic: 458.3 on 4 and 15 DF,  p-value: 1.764e-15
```

This is a Failure

Fit a model without stress

```r
m_4.3<- lm(BP ~Age+Weight+Dur, data = df_4)
summary(m_4.3)
```

```
##
## Call:
## lm(formula = BP ~ Age + Weight + Dur, data = df_4)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.03592 -0.29671  0.05216  0.32551  0.85934
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) -16.09486    3.10435  -5.185 9.04e-05 ***
## Age           0.69526    0.05661  12.280 1.47e-09 ***
## Weight        1.03121    0.03159  32.639 4.54e-16 ***
## Dur           0.04821    0.06152   0.784    0.445
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.5388 on 16 degrees of freedom
## Multiple R-squared:  0.9917, Adjusted R-squared:  0.9901
## F-statistic: 637.6 on 3 and 16 DF,  p-value: < 2.2e-16
```

Again a Bad model, as the VIF for this model has less VIFs so, there is no multicollinearity, so the parameter is ingnificant not for multicollinearity.

Fitting a model removing dur keeping stress

```r
m_4.4<-lm(BP ~Age+Weight+Stress, data = df_4)
summary(m_4.4)
```

```
##
## Call:
## lm(formula = BP ~ Age + Weight + Stress, data = df_4)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -1.0252 -0.3277  0.0368  0.2274  0.8901
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) -16.196316   3.090002  -5.242 8.07e-05 ***
## Age           0.691179   0.058833  11.748 2.80e-09 ***
## Weight        1.036206   0.031865  32.518 4.82e-16 ***
## Stress        0.002710   0.003625   0.748    0.465
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.5397 on 16 degrees of freedom
```

```
## Multiple R-squared:  0.9917, Adjusted R-squared:  0.9901
## F-statistic: 635.4 on 3 and 16 DF,  p-value: < 2.2e-16
```

Fitting model without BSA and Stress

```
m_4.5<- lm(BP~Age+Weight+Pulse+Dur, data = df_4)
summary(m_4.5)
```

```
##
## Call:
## lm(formula = BP ~ Age + Weight + Pulse + Dur, data = df_4)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.87309 -0.25007 -0.00217  0.30303  0.89738
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) -15.96851    2.95072  -5.412 7.21e-05 ***
## Age           0.74032    0.06033  12.271 3.19e-09 ***
## Weight        1.06556    0.03654  29.164 1.26e-14 ***
## Pulse        -0.08165    0.04950  -1.650    0.120
## Dur           0.07448    0.06058   1.229    0.238
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.512 on 15 degrees of freedom
## Multiple R-squared:  0.993,  Adjusted R-squared:  0.9911
## F-statistic: 530.3 on 4 and 15 DF,  p-value: 5.957e-16
```

```
#-------VIf-------
vif(m_4.5)
```

```
##      Age   Weight    Pulse      Dur
## 1.649586 1.784753 2.568296 1.224274
```

Fitting MOdel without BSA, Stress, DUR

```
m_4.6<- lm(BP ~Age+Weight+Pulse, data = df_4)
summary(m_4.6)
```

```
##
## Call:
## lm(formula = BP ~ Age + Weight + Pulse, data = df_4)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.71174 -0.45422 -0.01909  0.41745  0.88743
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) -16.69000    2.93761  -5.681 3.40e-05 ***
## Age           0.75018    0.06074  12.350 1.36e-09 ***
## Weight        1.06135    0.03695  28.722 3.40e-15 ***
## Pulse        -0.06566    0.04852  -1.353    0.195
```

```
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.5201 on 16 degrees of freedom
## Multiple R-squared:  0.9923, Adjusted R-squared:  0.9908
## F-statistic: 684.7 on 3 and 16 DF,  p-value: < 2.2e-16
```

```
#------VIF--------
vif(m_4.6)
```

```
##      Age   Weight    Pulse
## 1.620404 1.769065 2.390933
```

Again fit the model without Pluse

```
m_4.6<- lm(BP ~Age+Weight, data = df_4)
summary(m_4.6)
```

```
##
## Call:
## lm(formula = BP ~ Age + Weight, data = df_4)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.89968 -0.35242  0.06979  0.35528  0.82781
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) -16.57937    3.00746  -5.513 3.80e-05 ***
## Age           0.70825    0.05351  13.235 2.22e-10 ***
## Weight        1.03296    0.03116  33.154  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.5327 on 17 degrees of freedom
## Multiple R-squared:  0.9914, Adjusted R-squared:  0.9904
## F-statistic: 978.2 on 2 and 17 DF,  p-value: < 2.2e-16
```

So this is a significant model, lets check is there still multicollinearity or not

```
#--------vif-----------------
vif(m_4.6)
```

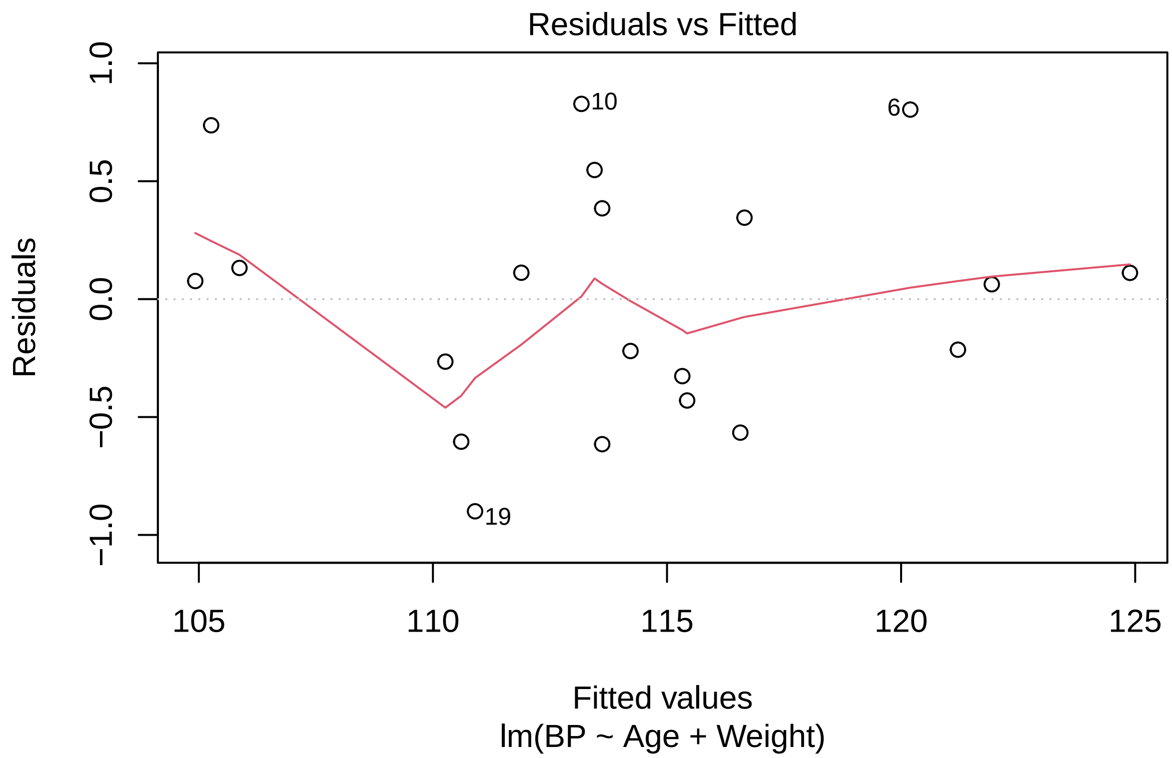
```
##      Age   Weight
## 1.198945 1.198945
```

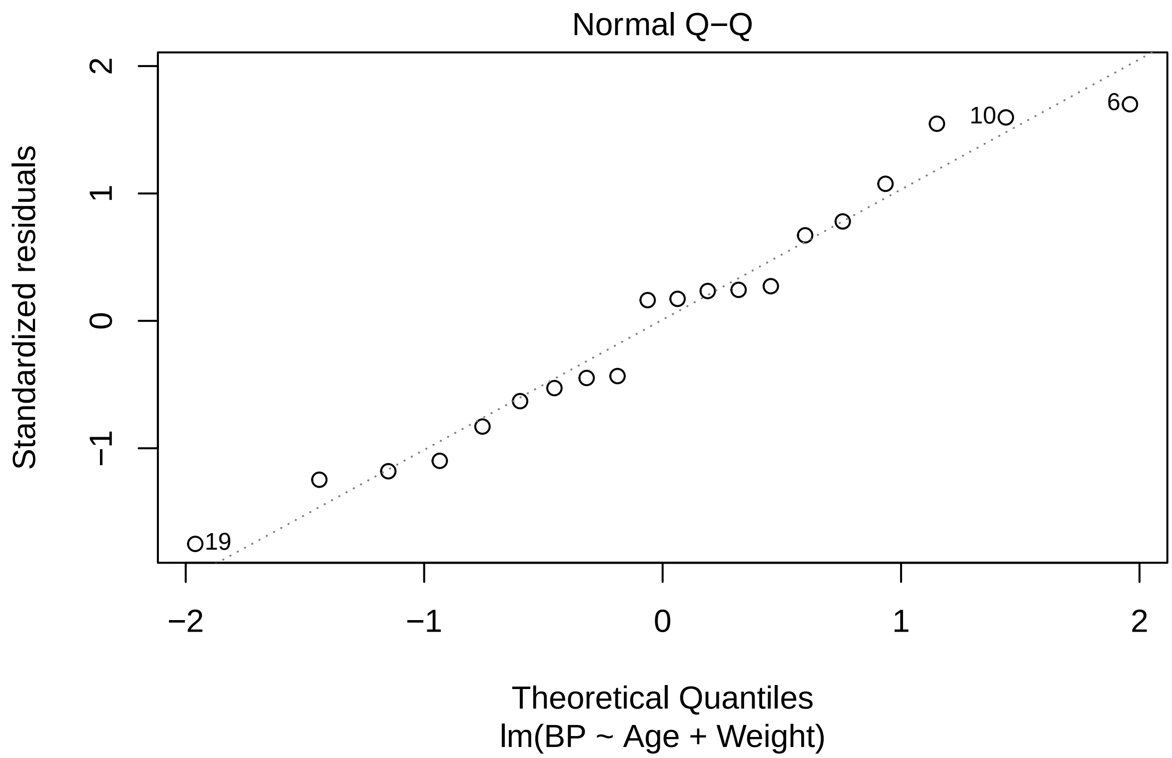
The Vifs are also low

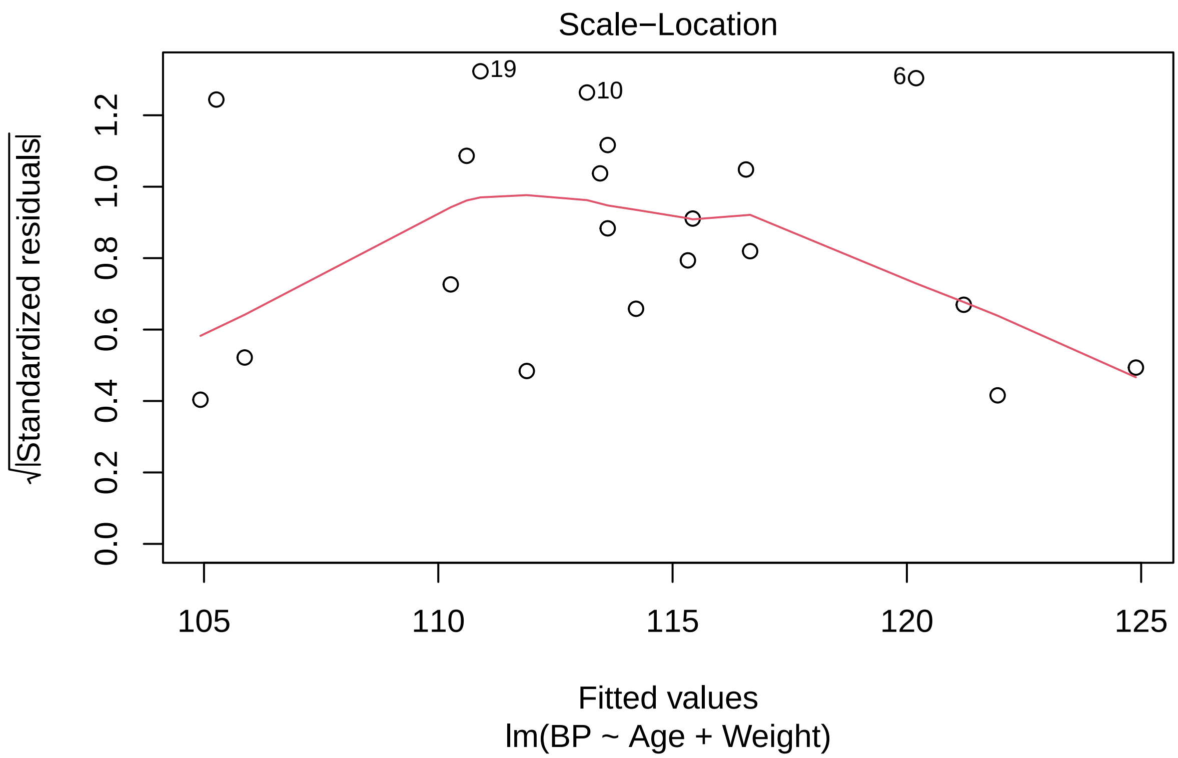
So, the data is good enough to work with further.

As we would consider this model, the model will be good enough if it statisfies the asuumptions

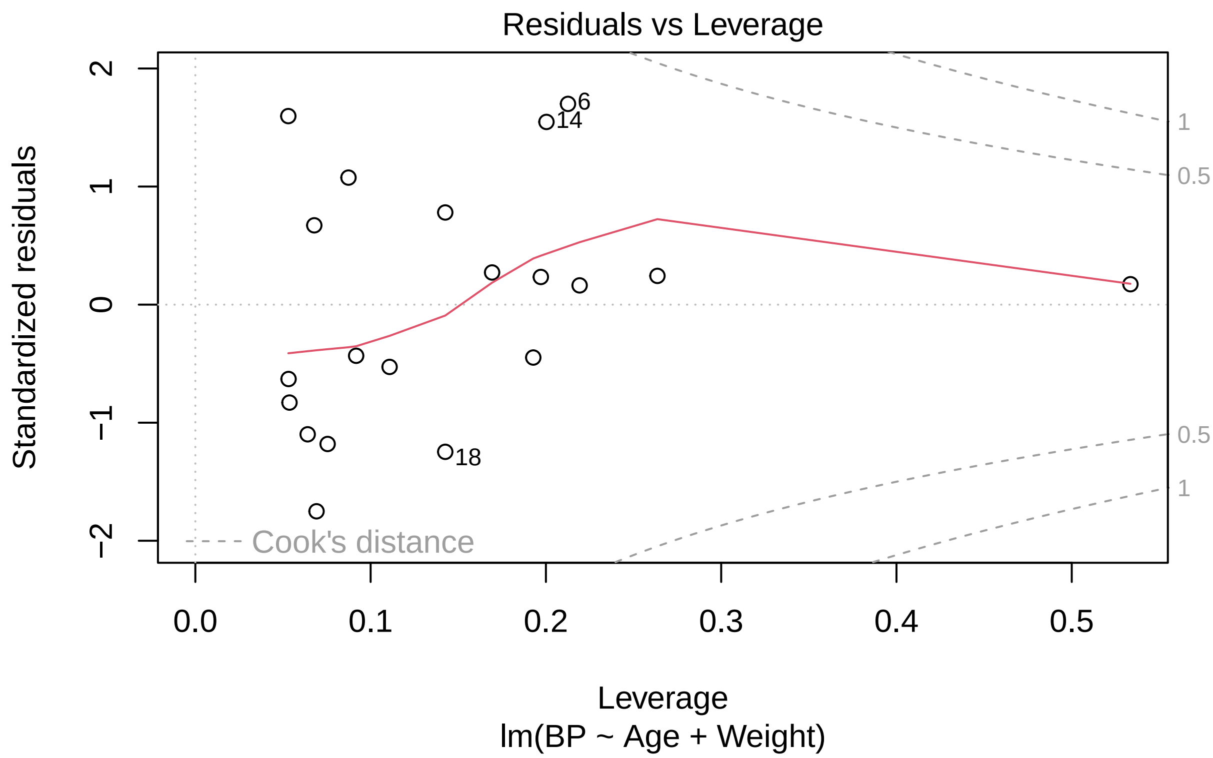
```
# ------------------- Residual Analysis -----------------

plot(m_4.6)
```


Residuals vs Fitted
Residuals
1.0
0.5
0.0
−0.5
−1.0
105
110
115
120
125
10
6
19
Fitted values
lm(BP ~ Age + Weight)


Normal Q−Q
Standardized residuals
Theoretical Quantiles
lm(BP ~ Age + Weight)
2
1
0
−1
−2
−1
0
1
2
19
10
6



Scale−Location
√|Standardized residuals|
0.0
0.2
0.4
0.6
0.8
1.0
1.2
19
10
6
105
110
115
120
125
Fitted values
lm(BP ~ Age + Weight)



Stepwise Regression Model

FOrward stepwise

```r
library(MASS)
int_only<- lm(BP~ 1, data=df_4)
s_m_f<- step(int_only, direction = "forward", scope = formula(m_4))
```

```
## Start:  AIC=68.64
## BP ~ 1
##
##          Df Sum of Sq    RSS    AIC
## + Weight  1    505.47  54.53 24.060
## + BSA     1    419.86 140.14 42.938
## + Pulse   1    291.44 268.56 55.946
## + Age     1    243.27 316.73 59.247
## <none>                560.00 68.644
## + Dur     1     48.02 511.98 68.851
## + Stress  1     15.04 544.96 70.099
##
## Step:  AIC=24.06
## BP ~ Weight
##
##          Df Sum of Sq    RSS     AIC
## + Age     1    49.704  4.824 -22.443
```

```
## + Stress  1     9.660 44.868  22.160
## + Pulse   1     8.940 45.588  22.478
## + Dur     1     6.095 48.433  23.689
## <none>                54.528  24.060
## + BSA     1     2.814 51.714  25.000
##
## Step:  AIC=-22.44
## BP ~ Weight + Age
##
##          Df Sum of Sq    RSS     AIC
## + BSA     1   1.76778 3.0561 -29.572
## + Pulse   1   0.49557 4.3284 -22.611
## <none>                4.8239 -22.443
## + Dur     1   0.17835 4.6456 -21.196
## + Stress  1   0.16286 4.6611 -21.130
##
## Step:  AIC=-29.57
## BP ~ Weight + Age + BSA
##
##          Df Sum of Sq    RSS     AIC
## + Dur     1   0.33510 2.7210 -29.894
## <none>                3.0561 -29.572
## + Stress  1   0.21774 2.8384 -29.050
## + Pulse   1   0.04111 3.0150 -27.842
##
## Step:  AIC=-29.89
## BP ~ Weight + Age + BSA + Dur
##
##          Df Sum of Sq    RSS     AIC
## <none>                2.7210 -29.894
## + Pulse   1   0.12307 2.5980 -28.820
## + Stress  1   0.12077 2.6003 -28.802
```

Coefficients

```r
s_m_f$coefficients
```

```
##  (Intercept)       Weight          Age          BSA          Dur
## -12.85206440   0.89700637   0.68335254   4.86037186   0.06652958
```

Backward Stepwise

```r
s_m_b<- step(m_4, direction = "backward", scope = formula(int_only))
```

```
## Start:  AIC=-30.55
## BP ~ Age + Weight + BSA + Dur + Pulse + Stress
##
##          Df Sum of Sq    RSS     AIC
## <none>                 2.156 -30.551
## - Dur     1     0.330  2.486 -29.698
## - Stress  1     0.442  2.598 -28.820
## - Pulse   1     0.444  2.600 -28.802
## - BSA     1     0.947  3.103 -25.267
## - Age     1    33.331 35.486  23.468
```

```
## - Weight  1    39.172 41.328  26.516
```

Both side

```r
s_m<- step(int_only, direction = "both", scope = formula(m_4))
```

```
## Start:  AIC=68.64
## BP ~ 1
##
##          Df Sum of Sq    RSS    AIC
## + Weight  1    505.47  54.53 24.060
## + BSA     1    419.86 140.14 42.938
## + Pulse   1    291.44 268.56 55.946
## + Age     1    243.27 316.73 59.247
## <none>                560.00 68.644
## + Dur     1     48.02 511.98 68.851
## + Stress  1     15.04 544.96 70.099
##
## Step:  AIC=24.06
## BP ~ Weight
##
##          Df Sum of Sq    RSS     AIC
## + Age     1     49.70   4.82 -22.443
## + Stress  1      9.66  44.87  22.160
## + Pulse   1      8.94  45.59  22.478
## + Dur     1      6.09  48.43  23.689
## <none>                 54.53  24.060
## + BSA     1      2.81  51.71  25.000
## - Weight  1    505.47 560.00  68.644
##
## Step:  AIC=-22.44
## BP ~ Weight + Age
##
##          Df Sum of Sq    RSS     AIC
## + BSA     1     1.768   3.06 -29.572
## + Pulse   1     0.496   4.33 -22.611
## <none>                  4.82 -22.443
## + Dur     1     0.178   4.65 -21.196
## + Stress  1     0.163   4.66 -21.130
## - Age     1    49.704  54.53  24.060
## - Weight  1   311.910 316.73  59.247
##
## Step:  AIC=-29.57
## BP ~ Weight + Age + BSA
##
##          Df Sum of Sq    RSS     AIC
## + Dur     1     0.335  2.721 -29.894
## <none>                 3.056 -29.572
## + Stress  1     0.218  2.838 -29.050
## + Pulse   1     0.041  3.015 -27.842
## - BSA     1     1.768  4.824 -22.443
## - Age     1    48.658 51.714  25.000
## - Weight  1    65.303 68.359  30.581
##
```

```
## Step:  AIC=-29.89
## BP ~ Weight + Age + BSA + Dur
##
##          Df Sum of Sq    RSS     AIC
## <none>                 2.721 -29.894
## - Dur     1     0.335  3.056 -29.572
## + Pulse   1     0.123  2.598 -28.820
## + Stress  1     0.121  2.600 -28.802
## - BSA     1     1.925  4.646 -21.196
## - Age     1    42.021 44.742  24.104
## - Weight  1    62.878 65.599  31.756
```

```
s_m$coefficients
```

```
##  (Intercept)       Weight          Age          BSA          Dur
## -12.85206440   0.89700637   0.68335254   4.86037186   0.06652958
```

```
summary(s_m)
```

```
##
## Call:
## lm(formula = BP ~ Weight + Age + BSA + Dur, data = df_4)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.86420 -0.26320  0.08341  0.25020  0.58272
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) -12.85206    2.64804  -4.853 0.000211 ***
## Weight        0.89701    0.04818  18.618 8.88e-12 ***
## Age           0.68335    0.04490  15.220 1.58e-10 ***
## BSA           4.86037    1.49220   3.257 0.005305 **
## Dur           0.06653    0.04895   1.359 0.194184
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.4259 on 15 degrees of freedom
## Multiple R-squared:  0.9951, Adjusted R-squared:  0.9938
## F-statistic:   768 on 4 and 15 DF,  p-value: < 2.2e-16
```

```
vif(s_m)
```

```
##   Weight      Age      BSA      Dur
## 4.484932 1.320201 4.344272 1.154968
```

```
## CN
a<- model.matrix(s_m)
a<- t(a)%*%a
a<- eigen(a)$values
cn<- sqrt(max(a)/min(a))
cn
```

```
## [1] 3015.973
```

Multicollinearity is still there.