MINI PROJECT 6

GROUP NO : 28

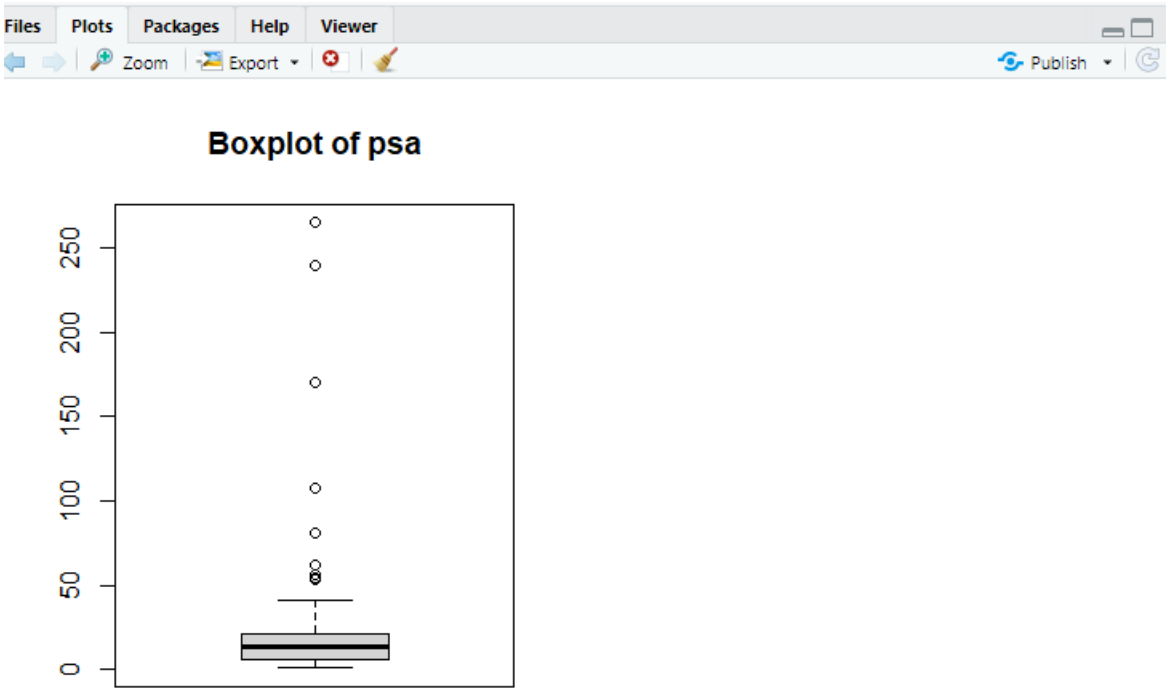
JEEVAN DSOUZA

SAMARTH SAIRAM

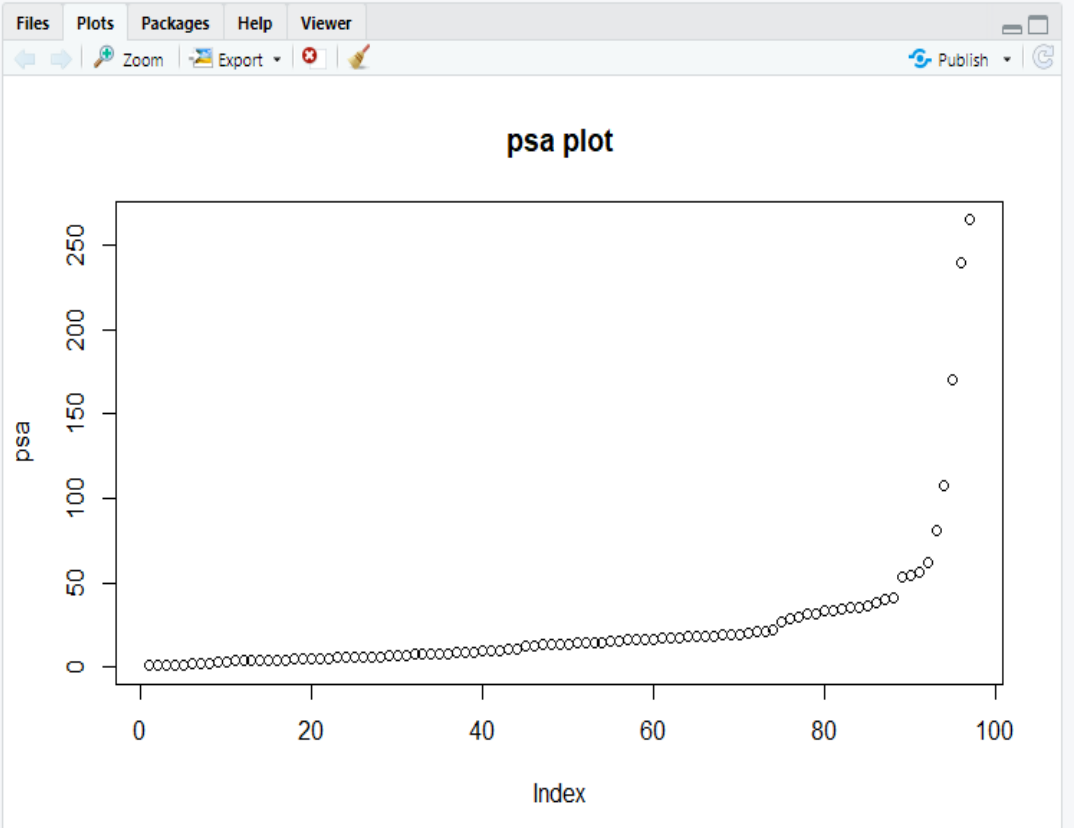
CONTRIBUTIONS

Each group member rendered contributions equally in both analysis and design of the solution for the given problem statement.

```
Pro_can = read.csv("prostate_cancer.csv")

Pro_can

psa = Pro_can$psa


#scatterplot for PSA

plot(psa,main='psa plot')


par(mfrow=c(1,2))

#boxplot of psa to check outliers

boxplot(psa,main='Boxplot of psa')
```




```
qqnorm(psa)

qqline(psa,col='blue')
```

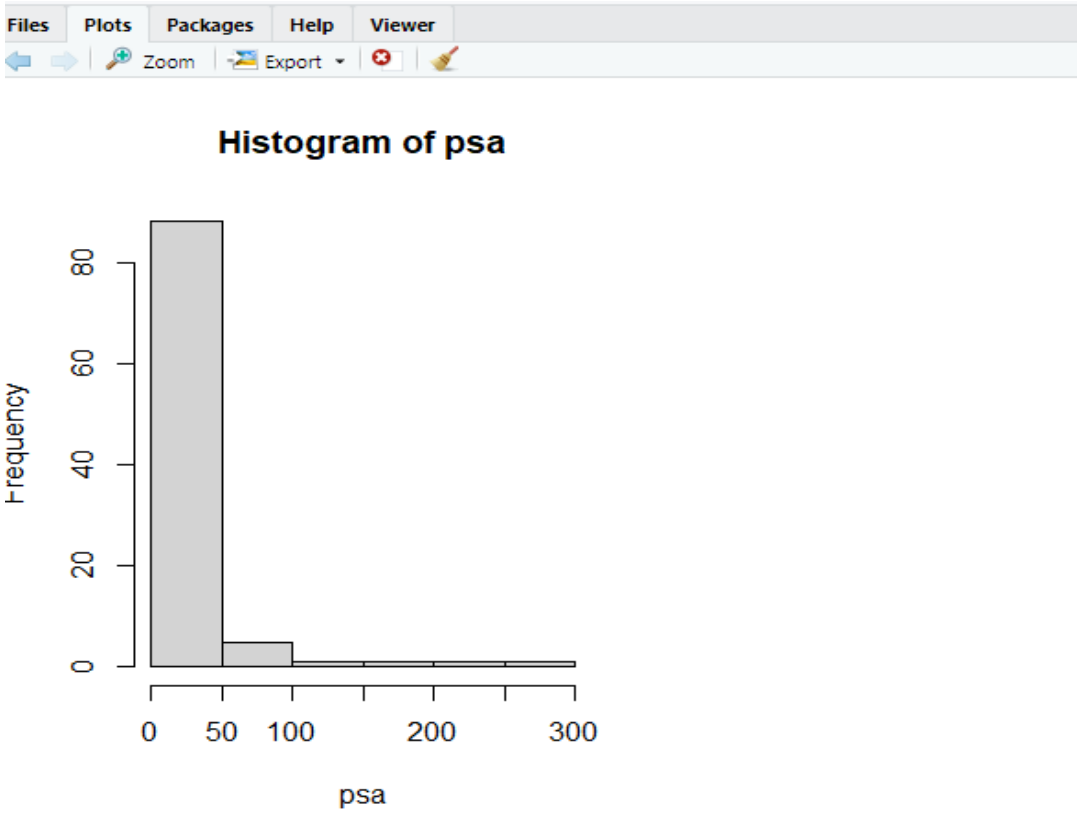


Observation:- In the above plots we see that there are many outliers in the data and also the data doesn't fit qq plot

Thus we use logarithmic function for a better fit

#Histogram plot

Hist(psa)

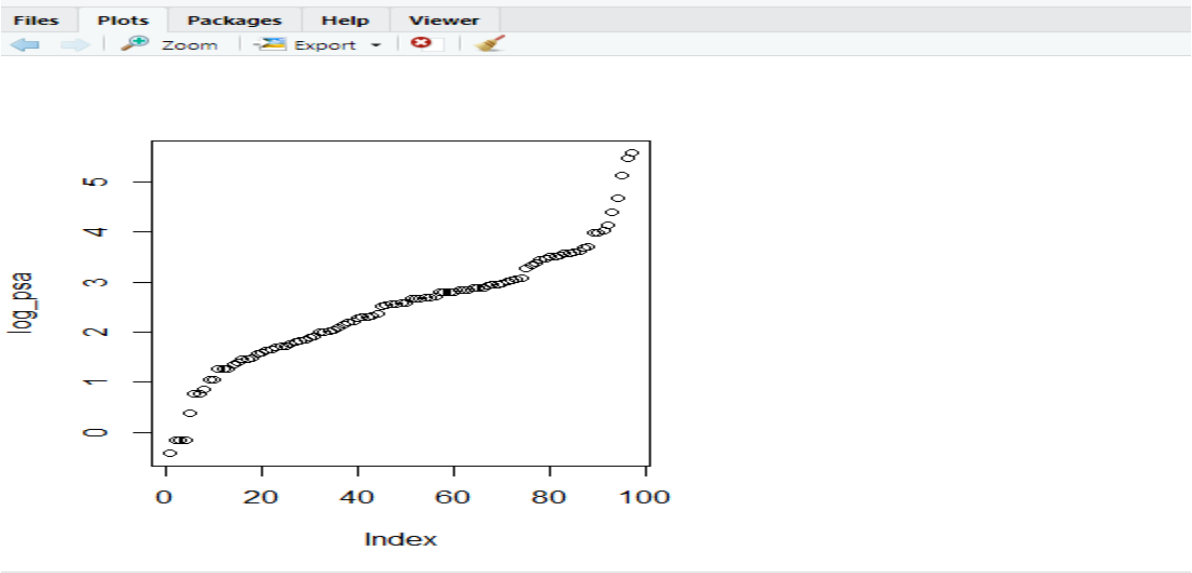


#Take the log transformation

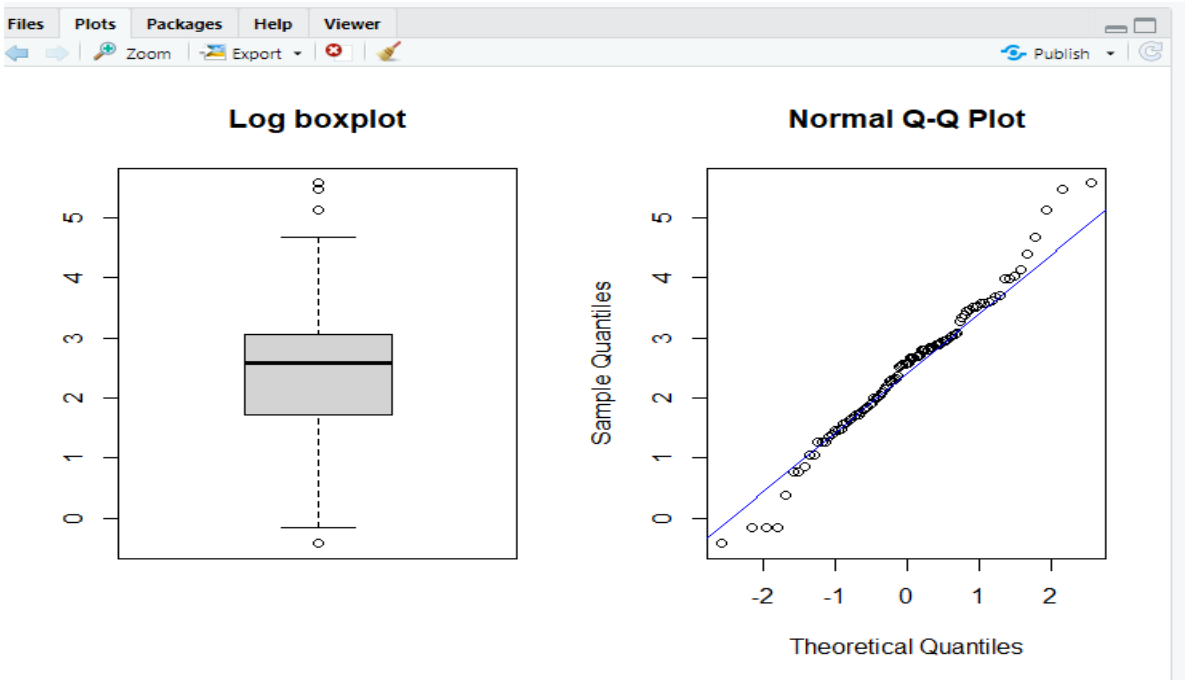
log_psa = log(psa)

#See the plot after applying the transformation

plot(log_psa)




```
par(mfrow=c(1,2))
boxplot(log_psa,main='Log boxplot')
qqnorm(log_psa)
qqline(log_psa,col='blue')
```

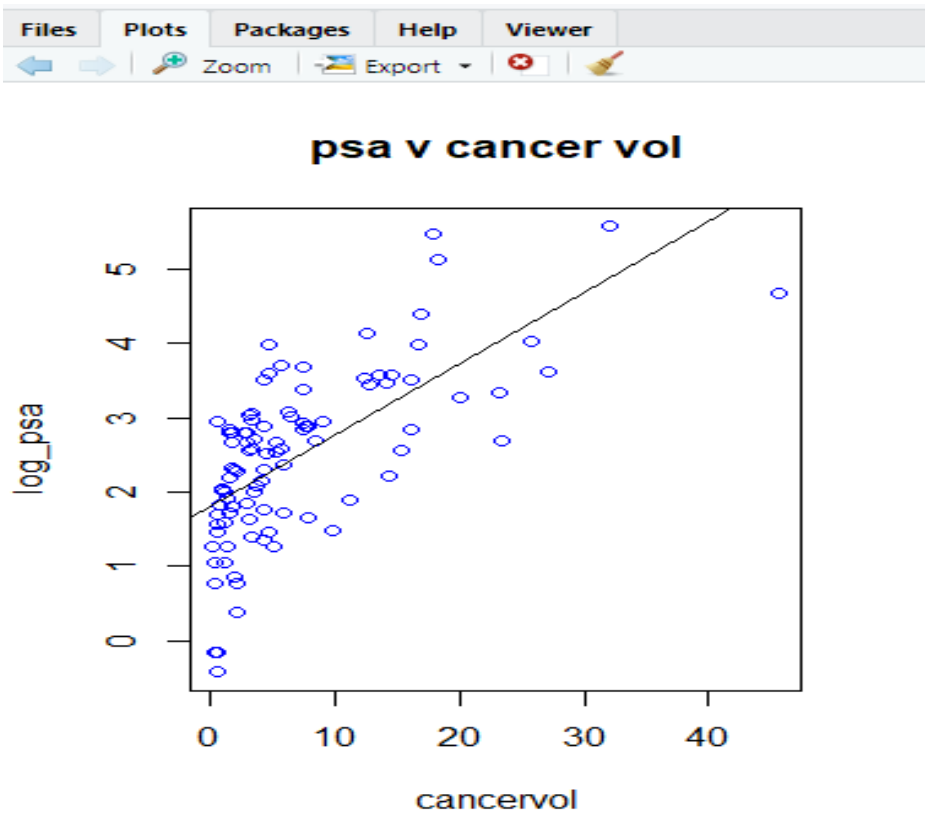


Observation: Now the boxplot and qqplot seem to be correct.

#Initializing all the parameters to be used

```
cancervol = Pro_can$cancervol

weight = Pro_can$weight

Age = Pro_can$age

Benpros = Pro_can$benpros

Vesinv = Pro_can$vesinv

Capspen = Pro_can$capspen

Gleason = Pro_can$gleason


psa_cancervol <- lm(log_psa~cancervol, data=Pro_can)

psa_cancervol

summary(psa_cancervol)

plot(cancervol,log_psa,col='blue',main='psa v cancer vol')

abline(psa_cancervol)
```

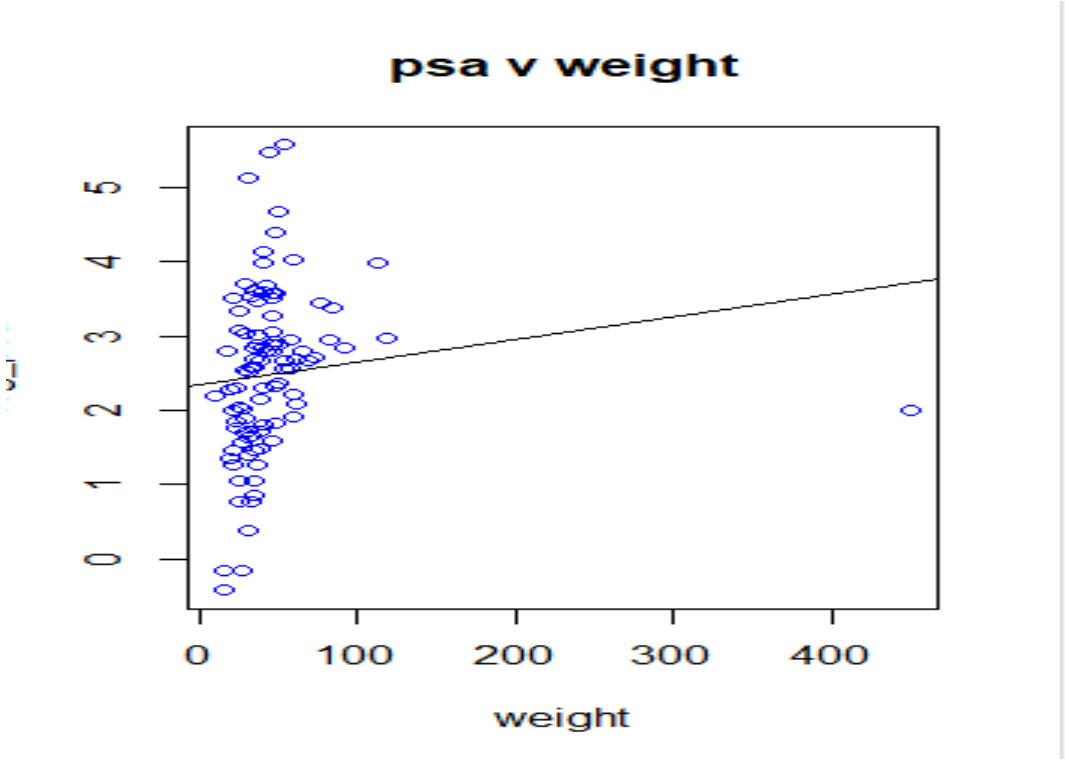
```
> psa_cancervol

Call:
lm(formula = log_psa ~ cancervol, data = Pro_can)

Coefficients:
(Intercept)    cancervol
    1.80549      0.09619

> summary(psa_cancervol)

Call:
lm(formula = log_psa ~ cancervol, data = Pro_can)

Residuals:
    Min      1Q  Median      3Q     Max
-2.2886 -0.6590  0.1493  0.5769  1.9610

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  1.80549    0.11899  15.174  < 2e-16 ***
cancervol    0.09619    0.01132   8.496 2.69e-13 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.8742 on 95 degrees of freedom
Multiple R-squared:  0.4317,      Adjusted R-squared:  0.4258
F-statistic: 72.18 on 1 and 95 DF,  p-value: 2.688e-13
```




```
psa_weight <- lm(log_psa~weight, data=Pro_can)
psa_weight
summary(psa_weight)
plot(weight,log_psa,col='blue',main='psa v weight')
abline(psa_weight)
```

```
> psa_weight

Call:
lm(formula = log_psa ~ weight, data = Pro_can)

Coefficients:
(Intercept)      weight
   2.338901    0.003072
```

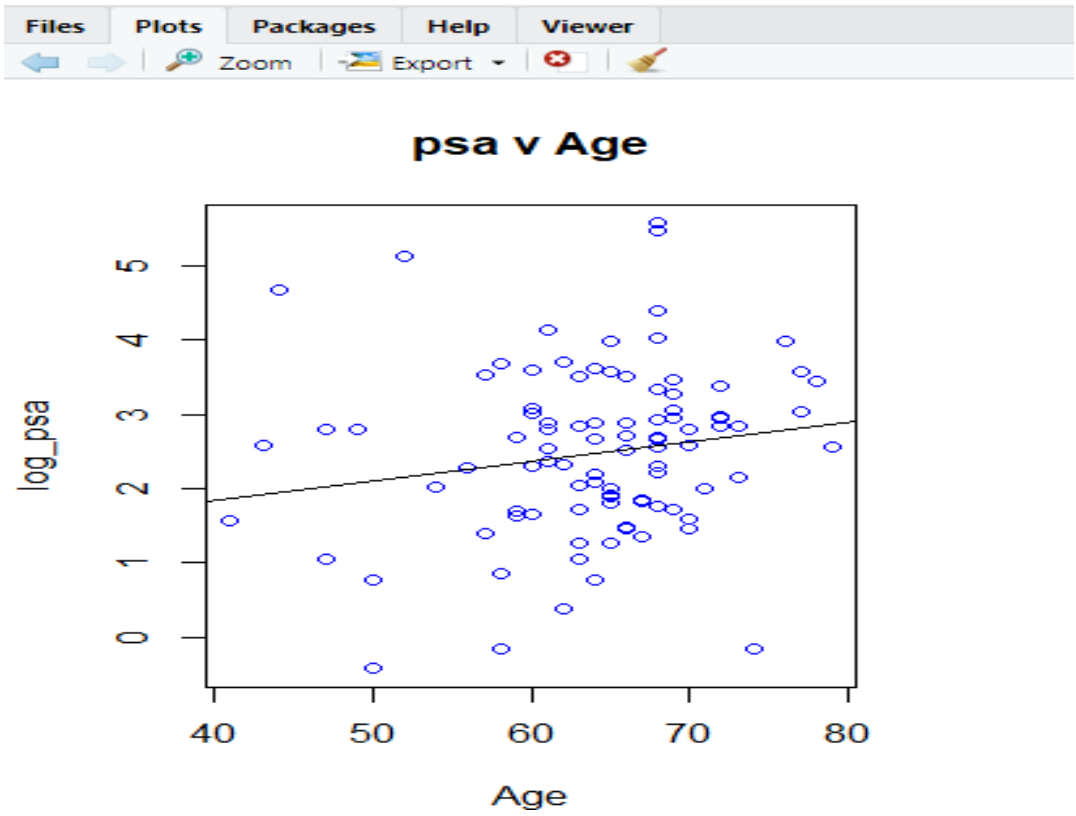
```
> summary(psa_weight)

Call:
lm(formula = log_psa ~ weight, data = Pro_can)

Residuals:
    Min      1Q  Median      3Q     Max
-2.8172 -0.7291  0.1300  0.6144  3.0783

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept) 2.338901   0.165328  14.147   <2e-16 ***
weight      0.003072   0.002570   1.195    0.235
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 1.151 on 95 degrees of freedom
Multiple R-squared:  0.01482,    Adjusted R-squared:  0.004446
F-statistic: 1.429 on 1 and 95 DF,  p-value: 0.235
```




```
psa_Age <- lm(log_psa~Age, data=Pro_can)
psa_Age
summary(psa_Age)
plot(Age,log_psa,col='blue',main='psa v Age')
abline(psa_Age)
```

```
> psa_Age

Call:
lm(formula = log_psa ~ Age, data = Pro_can)

Coefficients:
(Intercept)          Age
    0.79721      0.02633
```

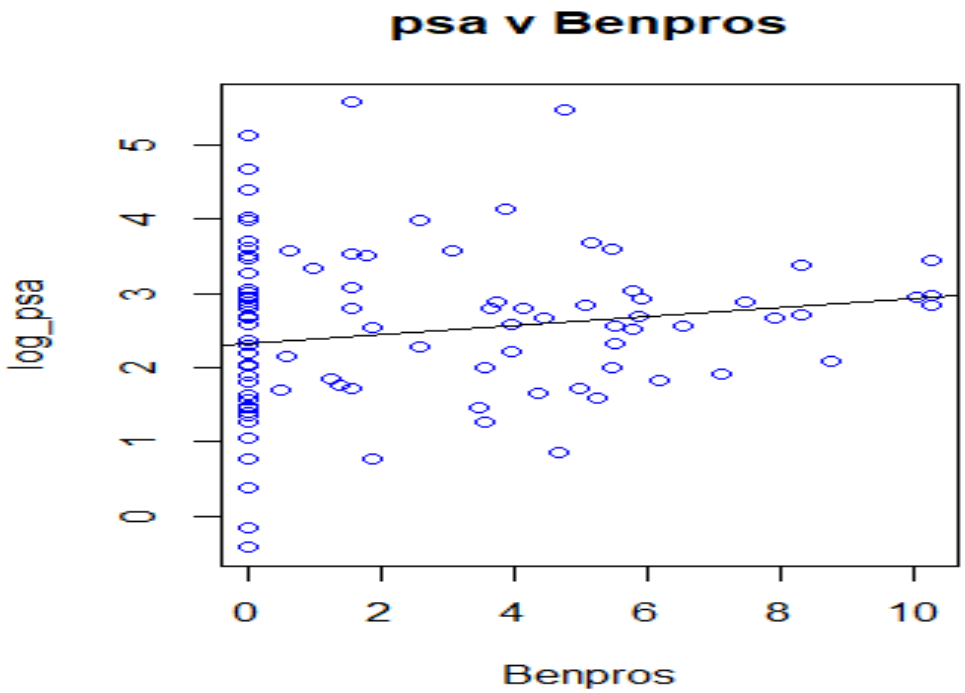
```
> summary(psa_Age)

Call:
lm(formula = log_psa ~ Age, data = Pro_can)

Residuals:
    Min      1Q  Median      3Q     Max
-2.90564 -0.71115  0.07247  0.66617  2.99249

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  0.79721    1.00729   0.791   0.4307
Age          0.02633    0.01567   1.680   0.0961 .
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 1.143 on 95 degrees of freedom
Multiple R-squared:  0.02887,    Adjusted R-squared:  0.01865
F-statistic: 2.824 on 1 and 95 DF,  p-value: 0.09615
```




```
psa_Benpros<- lm(log_psa~Benpros, data=Pro_can)

psa_Benpros

summary(psa_Benpros)

plot(Benpros,log_psa,col='blue',main='psa v Benpros')

abline(psa_Benpros)
```

```
> psa_Benpros

Call:

lm(formula = log_psa ~ Benpros, data = Pro_can)

Coefficients:

(Intercept)     Benpros

    2.32682     0.05991
```

```
> summary(psa_Benpros)

Call:
lm(formula = log_psa ~ Benpros, data = Pro_can)

Residuals:
    Min      1Q  Median      3Q     Max
-2.75607 -0.76149 -0.01686  0.63318  3.16016

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  2.32682    0.15191  15.317   <2e-16 ***
Benpros      0.05991    0.03856   1.554    0.124
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 1.145 on 95 degrees of freedom
Multiple R-squared:  0.02478,    Adjusted R-squared:  0.01451
F-statistic: 2.413 on 1 and 95 DF,  p-value: 0.1236



psa_Vesinv <- lm(log_psa~Vesinv, data=Pro_can)
psa_Vesinv
summary(psa_Vesinv)
plot(Vesinv,log_psa,col='blue',main='psa v vesinv vol')
abline(psa_Vesinv)
```



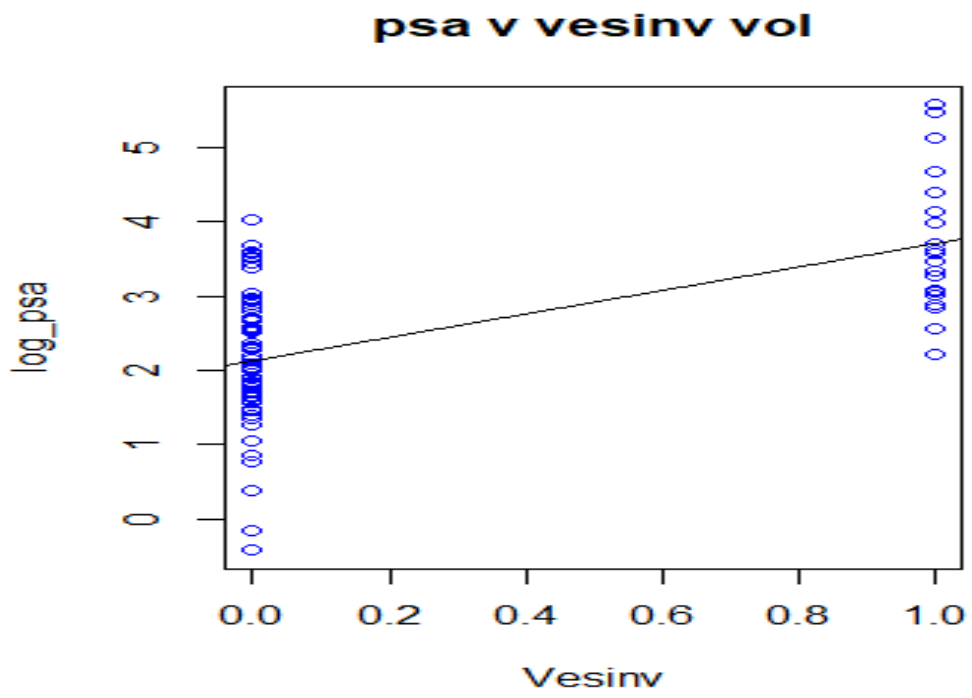

```
psa_Vesinv <- lm(log_psa~Vesinv, data=Pro_can)
psa_Vesinv
summary(psa_Vesinv)
plot(Vesinv,log_psa,col='blue',main='psa v cancer vol')
abline(psa_Vesinv)
```

Multiple R-squared: 0.02478,   Adjusted R-squared: 0.01451

F-statistic: 2.413 on 1 and 95 DF, p-value: 0.1236

```
> psa_Vesinv

Call:
lm(formula = log_psa ~ Vesinv, data = Pro_can)

Coefficients:
```

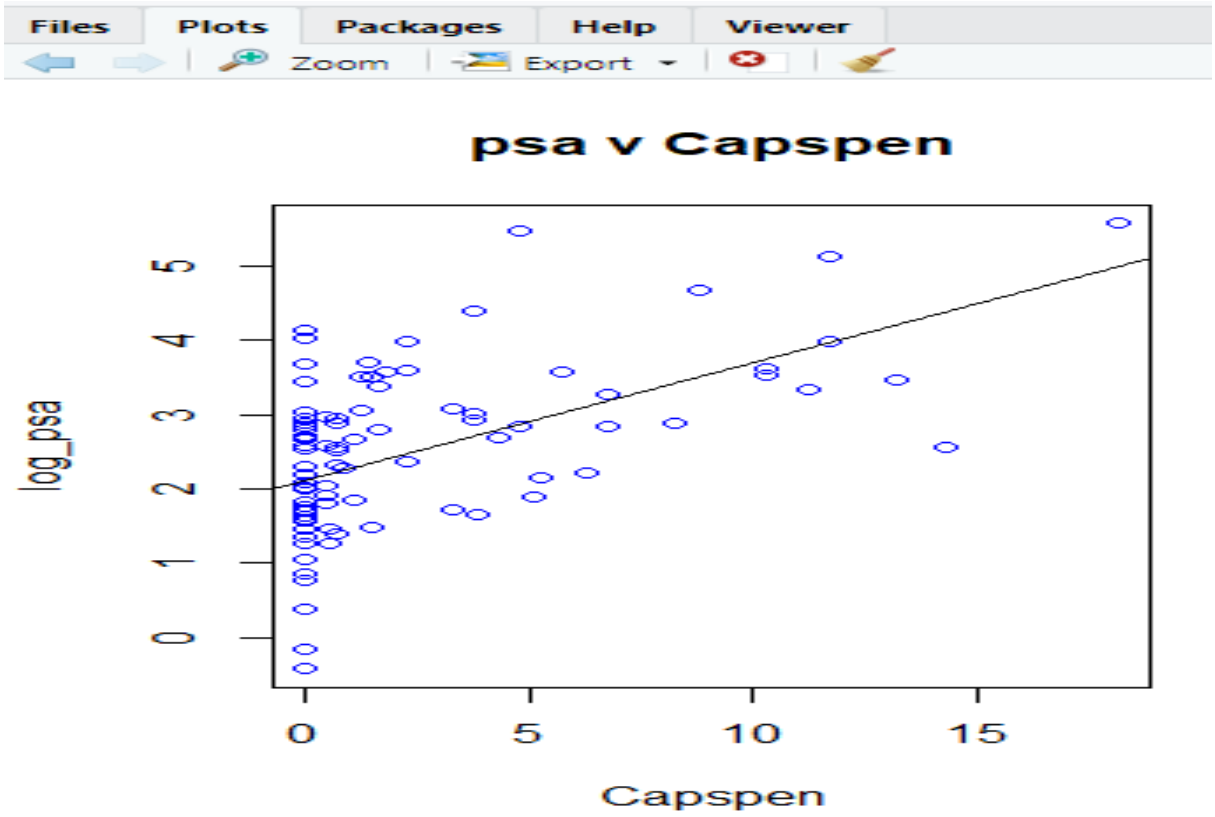
```
(Intercept)      Vesinv
     2.137       1.578

> summary(psa_Vesinv)

Call:
lm(formula = log_psa ~ Vesinv, data = Pro_can)

Residuals:
    Min      1Q  Median      3Q     Max
-2.56623 -0.63526 -0.00524  0.67302  1.89302

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)   2.1370     0.1096  19.492  < 2e-16 ***
Vesinv        1.5783     0.2356   6.698 1.48e-09 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.9558 on 95 degrees of freedom
Multiple R-squared:  0.3208,      Adjusted R-squared:  0.3136
F-statistic: 44.86 on 1 and 95 DF,  p-value: 1.481e-09
```




```
psa_Capspen <- lm(log_psa~Capspen, data=Pro_can)

psa_Capspen

summary(psa_cancervol)

plot(Capspen,log_psa,col='blue',main='psa v Capspen')

abline(psa_Capspen)
```

```
> psa_Capspen

Call:

lm(formula = log_psa ~ Capspen, data = Pro_can)

Coefficients:

(Intercept)     Capspen

      2.124       0.158

> summary(psa_cancervol)
```

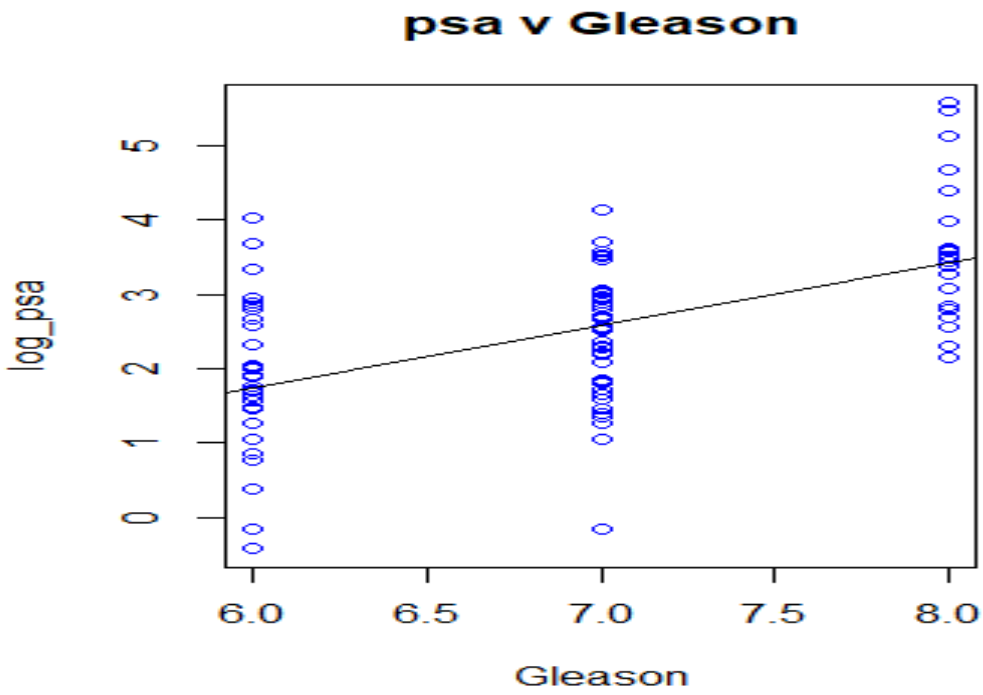
```
Call:
lm(formula = log_psa ~ cancervol, data = Pro_can)

Residuals:
    Min      1Q  Median      3Q     Max
-2.2886 -0.6590  0.1493  0.5769  1.9610

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  1.80549    0.11899  15.174  < 2e-16 ***
cancervol    0.09619    0.01132   8.496 2.69e-13 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.8742 on 95 degrees of freedom
Multiple R-squared:  0.4317,      Adjusted R-squared:  0.4258
F-statistic: 72.18 on 1 and 95 DF,  p-value: 2.688e-13
```




```
psa_Gleason <- lm(log_psa~Gleason, data=Pro_can)

psa_Gleason

summary(psa_Gleason)

plot(Gleason,log_psa,col='blue',main='psa v Gleason')

abline(psa_Gleason)
```

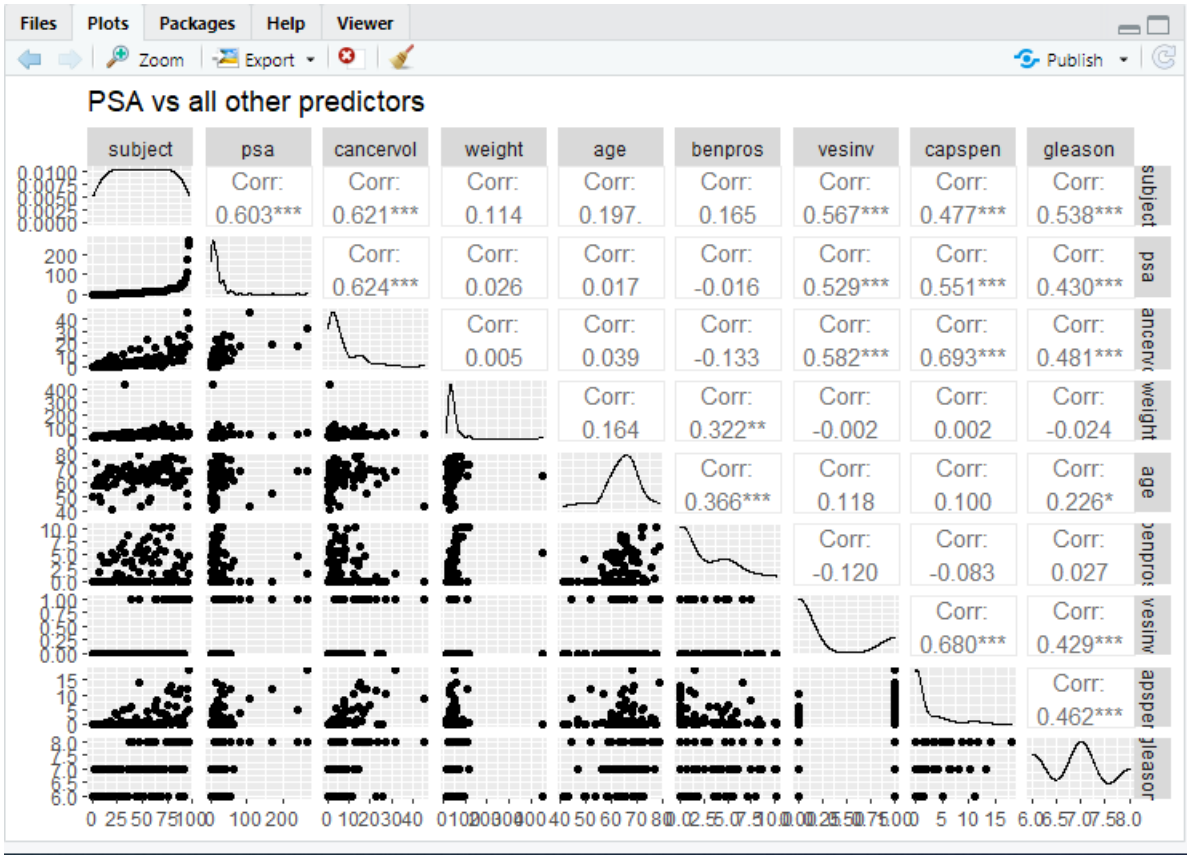
```
> psa_Gleason

Call:

lm(formula = log_psa ~ Gleason, data = Pro_can)

Coefficients:

(Intercept)      Gleason
```

```
   -3.3026       0.8408


> summary(psa_Gleason)


Call:

lm(formula = log_psa ~ Gleason, data = Pro_can)


Residuals:

    Min      1Q  Median      3Q     Max

-2.7428 -0.6134  0.0773  0.4773  2.2881


Coefficients:

            Estimate Std. Error t value Pr(>|t|)

(Intercept)  -3.3026     0.9322  -3.543 0.000616 ***

Gleason       0.8408     0.1348   6.237 1.23e-08 ***

---

Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1


Residual standard error: 0.9768 on 95 degrees of freedom

Multiple R-squared:  0.2905,       Adjusted R-squared:  0.2831

F-statistic:  38.9 on 1 and 95 DF,  p-value: 1.228e-08
```



Observation: To understand the correlation between the parameters PSA vs all other predictors

```
install.packages("GGally")

library(GGally)

ggpairs(data=Pro_can, columns=c(1:9), title="PSA vs all other predictors")
```



<u>Observation: We see that Cancervol,gleason,capspen and vesinv are highly correlated to PSA</u>

Next we have full model our NULL hypothesis help to predict response and Alternate Hypothesis : Atleast one of the predictors help predict response

fit1 <- lm(log_psa ~ cancervol+as.factor(Vesinv)+Capspen+Gleason+weight+Age+Benpros)

summary(fit1)

> summary(fit1)

Call:

lm(formula = log_psa ~ cancervol + as.factor(Vesinv) + Capspen +

Gleason + weight + Age + Benpros)

Residuals:

```
    Min      1Q  Median      3Q     Max
-1.88309 -0.46629  0.08045  0.47380  1.53219
```

Coefficients:

```
                  Estimate Std. Error t value Pr(>|t|)
(Intercept)       -0.685796   0.998754  -0.687  0.49409
cancervol          0.069454   0.014624   4.749 7.77e-06 ***
as.factor(Vesinv)1  0.782623   0.268339   2.917  0.00448 **
Capspen           -0.026521   0.032860  -0.807  0.42177
Gleason            0.358153   0.127976   2.799  0.00629 **
weight             0.001380   0.001822   0.757  0.45079
Age               -0.002799   0.011724  -0.239  0.81186
Benpros            0.087470   0.029605   2.955  0.00401 **
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

Residual standard error: 0.7679 on 89 degrees of freedom

Multiple R-squared: 0.5893, Adjusted R-squared: 0.557

F-statistic: 18.24 on 7 and 89 DF, p-value: 7.694e-15

From the above values we see that cancervol,gleason,vesinv and benpros are significant predictors as value is less than 0.05 . Thus we reject null hypothesis.

Same null and alternate hypothesis predictors as 1st model

```
fit2 <- update(fit1,.~. - Capspen -Age -weight)
summary(fit2)
```

```
> summary(fit2)

Call:
lm(formula = log_psa ~ cancervol + as.factor(Vesinv) + Gleason +
    Benpros)

Residuals:
     Min       1Q   Median       3Q      Max
-1.88531 -0.50276  0.09885  0.53687  1.56621

Coefficients:
                   Estimate Std. Error t value Pr(>|t|)
(Intercept)       -0.65013    0.80999  -0.803 0.424253
cancervol          0.06488    0.01285   5.051 2.22e-06 ***
as.factor(Vesinv)1  0.68421    0.23640   2.894 0.004746 **
Gleason            0.33376    0.12331   2.707 0.008100 **
Benpros            0.09136    0.02606   3.506 0.000705 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.7606 on 92 degrees of freedom
Multiple R-squared:  0.5834,      Adjusted R-squared:  0.5653
F-statistic: 32.21 on 4 and 92 DF,  p-value: < 2.2e-16
```

We see here all are significant predictors thus we reject null hypothesis and also we see adjusted R squared value also increases validating the correctness

As we see Capspen is one the most important parameter of the model we add it back and check our null and alternate hypothesis.

```
fit3 <- update(fit2, .~. + Capspen)
```

```
summary(fit3)
```

```
> summary(fit3)

Call:
lm(formula = log_psa ~ cancervol + as.factor(Vesinv) + Gleason +
    Benpros + Capspen)

Residuals:
     Min       1Q   Median       3Q      Max
-1.88954 -0.48197  0.08813  0.48409  1.57370

Coefficients:
                  Estimate Std. Error t value Pr(>|t|)
(Intercept)       -0.73258    0.81760  -0.896 0.372608
cancervol          0.07029    0.01445   4.863 4.82e-06 ***
as.factor(Vesinv)1  0.78233    0.26520   2.950 0.004041 **
Gleason            0.34568    0.12437   2.779 0.006617 **
Benpros            0.09198    0.02612   3.522 0.000672 ***
Capspen           -0.02680    0.03260  -0.822 0.413237
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.762 on 91 degrees of freedom
Multiple R-squared:  0.5865,      Adjusted R-squared:  0.5637
F-statistic: 25.81 on 5 and 91 DF,  p-value: 3.931e-16
```

From the results we see that significant predictors are vesinv,gleason ,cancervol and benpros and the adjusted R value decreases which indicates that capspen is not an optimal predictor to predict response variable

```
anova(fit1)
```

```
> anova(fit1)
Analysis of Variance Table

Response: log_psa
                 Df Sum Sq Mean Sq F value    Pr(>F)
cancervol         1 55.164  55.164 93.5572 1.522e-15 ***
as.factor(Vesinv)  1  6.547   6.547 11.1034  0.001256 **
Capspen           1  0.066   0.066  0.1114  0.739372
Gleason           1  5.954   5.954 10.0971  0.002042 **
weight            1  2.041   2.041  3.4624  0.066083 .
Age               1  0.374   0.374  0.6344  0.427866
Benpros           1  5.147   5.147  8.7291  0.004007 **
Residuals        89 52.477   0.590
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

anova (fit1,fit2,fit3)

anova(fit2,fit3)

```
> anova (fit1,fit2,fit3)
Analysis of Variance Table

Model 1: log_psa ~ cancervol + as.factor(Vesinv) + Capspen + Gleason +
    weight + Age + Benpros
Model 2: log_psa ~ cancervol + as.factor(Vesinv) + Gleason + Benpros
Model 3: log_psa ~ cancervol + as.factor(Vesinv) + Gleason + Benpros +
```

```
   Capspen
 Res.Df    RSS Df Sum of Sq      F Pr(>F)
1     89 52.477
2     92 53.229 -3  -0.75232 0.4253 0.7353
3     91 52.837  1   0.39230 0.6653 0.4169
>
> anova(fit2,fit3)
Analysis of Variance Table
```

Just ANOVA doesn't predict every significant thing so we use AIC score

```
Model 1: log_psa ~ cancervol + as.factor(Vesinv) + Gleason + Benpros
Model 2: log_psa ~ cancervol + as.factor(Vesinv) + Gleason + Benpros +
   Capspen
 Res.Df    RSS Df Sum of Sq      F Pr(>F)
1     92 53.229
2     91 52.837  1    0.3923 0.6757 0.4132
AIC_Fwd <- step(lm(log_psa ~ 1), scope = formula(fit1),k=2,trace=0, direction = "forward")
BIC_fwd <- step(lm(log_psa ~ 1), scope = formula(fit1), k = log(32), trace = 0,direction = "forward")
AIC_Bwd <- step(fit1, k = 2, trace = 0, direction = "backward")
BIC_BWD <- step(fit1, k = log(32), trace = 0, direction = "backward")

R_square_Adjst <- data.frame("Method"=c("AIC_Fwd", "BIC_fwd", "AIC_Bwd", "BIC_BWD"),
                             "Adj.r.square"=c(summary(AIC_Fwd)$adj.r.square, summary(BIC_fwd)$adj.r.square,
                                              summary(AIC_Bwd)$adj.r.square, summary(BIC_BWD)$adj.r.square))

summary(AIC_Fwd)

> summary(AIC_Fwd)

Call:
```

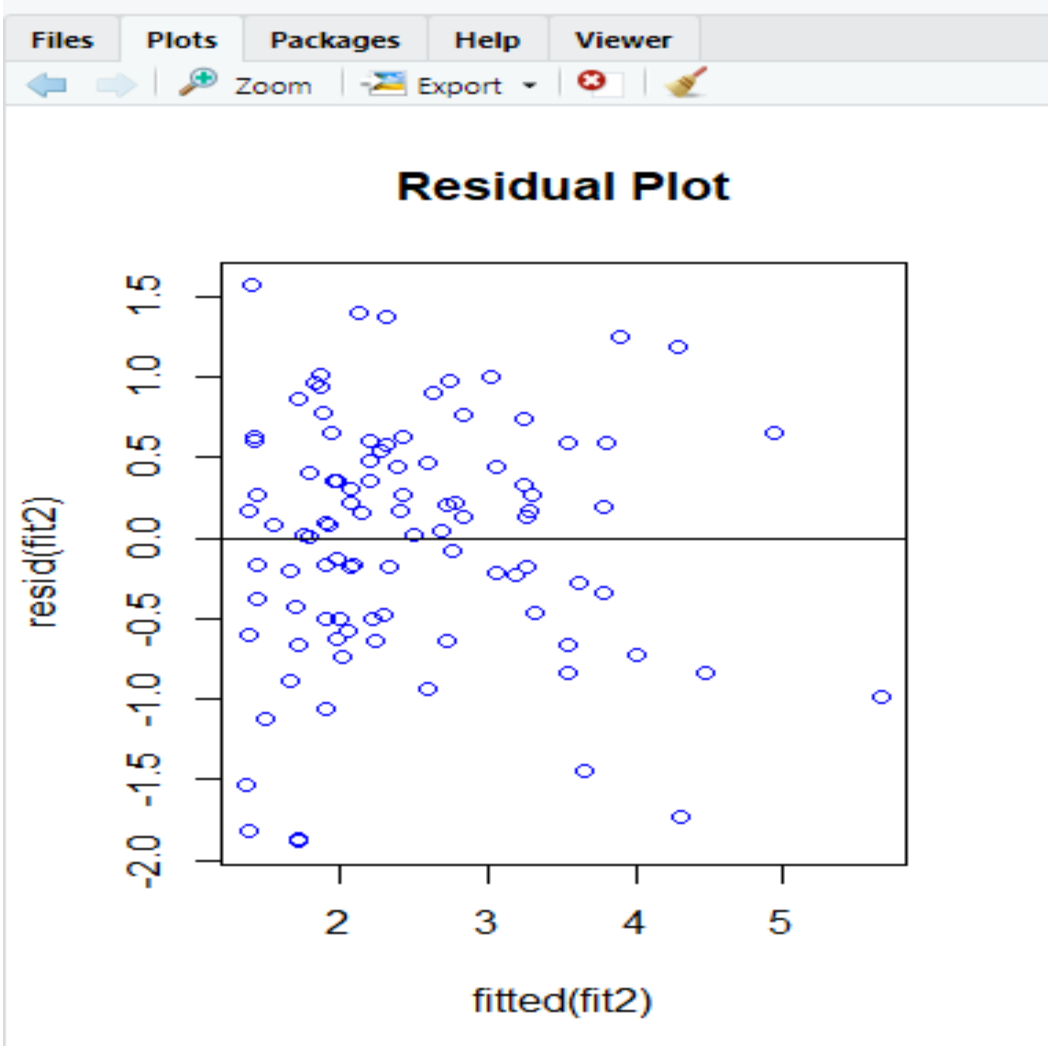
```
lm(formula = log_psa ~ cancervol + Gleason + Benpros + as.factor(Vesinv))

Residuals:
    Min      1Q  Median      3Q     Max
-1.88531 -0.50276  0.09885  0.53687  1.56621

Coefficients:
                  Estimate Std. Error t value Pr(>|t|)
(Intercept)       -0.65013    0.80999  -0.803 0.424253
cancervol          0.06488    0.01285   5.051 2.22e-06 ***
Gleason            0.33376    0.12331   2.707 0.008100 **
Benpros            0.09136    0.02606   3.506 0.000705 ***
as.factor(Vesinv)1  0.68421    0.23640   2.894 0.004746 **
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.7606 on 92 degrees of freedom
Multiple R-squared:  0.5834,      Adjusted R-squared:  0.5653
F-statistic: 32.21 on 4 and 92 DF,  p-value: < 2.2e-16

Fst_glm <- glm(fit2)
Scd_glm <- glm(fit1)
Thd_glm <- glm(fit3)

Fst_glm$aic
Scd_glm$aic
Thd_glm$aic

> Fst_glm <- glm(fit2)
```

```
> Scd_glm <- glm(fit1)
> Thd_glm <- glm(fit3)
>
> Fst_glm$aic
[1] 229.0635
> Scd_glm$aic
[1] 233.6828
> Thd_glm$aic
[1] 230.346
```

We see that fit2 or Fst_glm linear model has lowest AIC score , we see that it is the best mode of all.

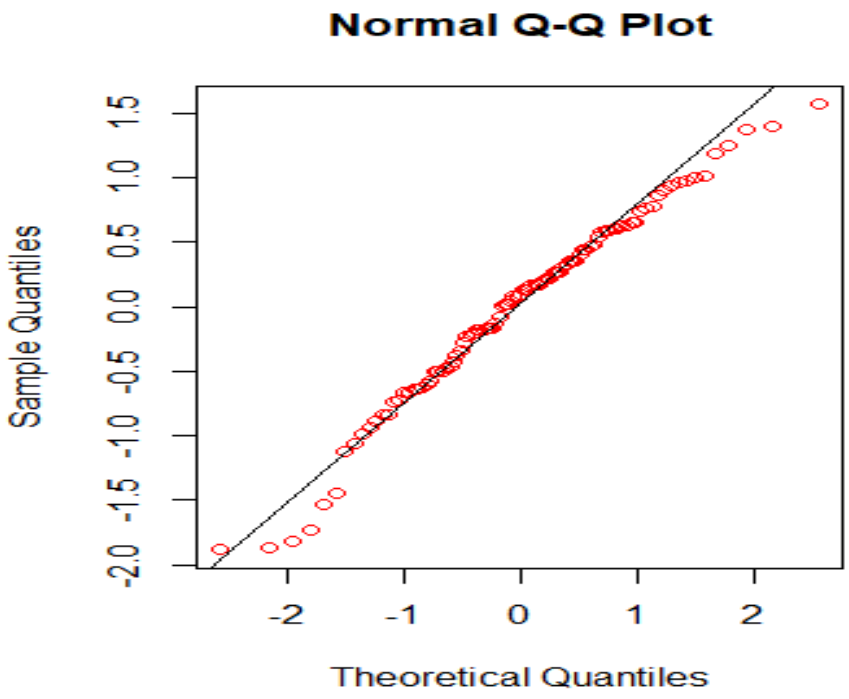
We perform now the model evaluation whether model 2 is good one or not.

```
plot(fitted(fit2),resid(fit2), main = "Residual Plot",col="blue")
abline(h=0)
```



From the plot we see mean is zero and no much change in the vertical spread so standard deviation is constant thus the linear model is good estimate.

```
qqnorm(resid(fit2),col="red")
qqline(resid(fit2))
```

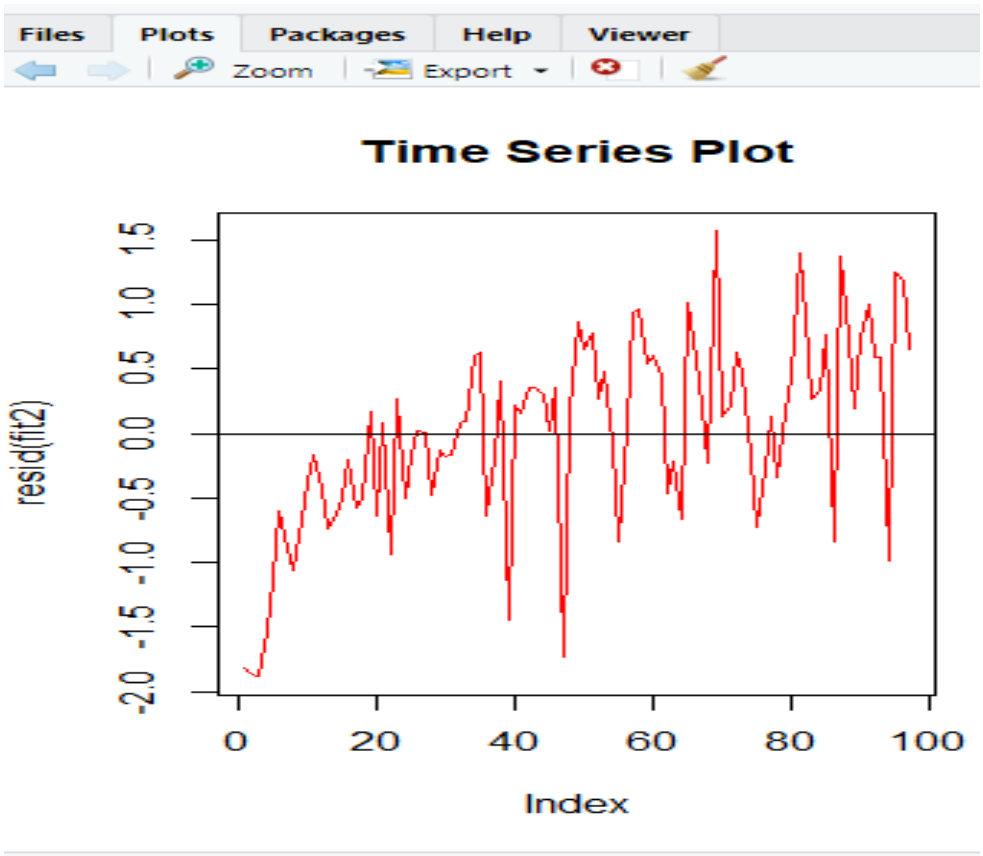



```
plot(resid(fit2),main = "Time Series Plot",col="red",type = "l")
abline(h=0)
```

Initially, we assumed that the Residual error is independent and identically distributed coming from a normal distribution with mean = 0 and standard deviation of sigma squared.

To validate this assumption, we plotted QQ Plot of fitted model.

From the QQ Plot, we observe that data is almost distributed normally.



We see that final model to predict PSA level whose quantitative predictors are at the sample mean of variables

lm(formula = log_psa ~ cancervol + as.factor(Vesinv) + Gleason + Benpros)

table(as.factor(Vesinv))

table(Gleason)

mean(Benpros)

mean(cancervol)

Call:

lm(formula = log_psa ~ cancervol + as.factor(Vesinv) + Gleason +

Benpros)

Coefficients:

(Intercept)	cancervol	as.factor(Vesinv)1	Gleason	Benpros
-0.65013	0.06488	0.68421	0.33376	0.09136

```
> table(as.factor(Vesinv))

 0  1
76 21
> table(Gleason)
Gleason
 6  7  8
33 43 21
> mean(Benpros)
[1] 2.534725
> mean(cancervol)
[1] 6.998682
```

We see here cancervol and benpros are 6.998 and 2.534 gleason value is 7 and vesinv is 0

Predicting the model with best linear model is fit 2

summary(fit2)

```
Call:
lm(formula = log_psa ~ cancervol + as.factor(Vesinv) + Gleason +
    Benpros)

Residuals:
     Min       1Q   Median       3Q      Max
-1.88531 -0.50276  0.09885  0.53687  1.56621
```

Coefficients:

	Estimate	Std. Error	t value	Pr(>|t|)	
(Intercept)	-0.65013	0.80999	-0.803	0.424253	
cancervol	0.06488	0.01285	5.051	2.22e-06	***
as.factor(Vesinv)1	0.68421	0.23640	2.894	0.004746	**
Gleason	0.33376	0.12331	2.707	0.008100	**
Benpros	0.09136	0.02606	3.506	0.000705	***

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

Residual standard error: 0.7606 on 92 degrees of freedom

Multiple R-squared: 0.5834, Adjusted R-squared: 0.5653

F-statistic: 32.21 on 4 and 92 DF, p-value: < 2.2e-16

predicted value= -0.65013 + 6.998682*(0.06488) + 7*(0.33376) + 0.09136*(2.534725)

= 2.371837

Hence, the actual value of PSA = exp(2.371837) = 10.71706

Section 2 code

```
Pro_can = read.csv("prostate_cancer.csv")
Pro_can
psa = Pro_can$psa

#scatterplot for PSA
plot(psa,main='psa plot')

par(mfrow=c(1,2))
```

```
#boxplot of psa to check outliers
boxplot(psa,main='Boxplot of psa')

#qqplot
qqnorm(psa)
qqline(psa,col='blue')


#Histogram plot
hist(psa)

#Take the log transformation
log_psa = log(psa)

#See the plot after applying the transformation
plot(log_psa)

par(mfrow=c(1,2))
boxplot(log_psa,main='Log boxplot')
qqnorm(log_psa)
qqline(log_psa,col='blue')

#Initializing all the parameters to be used
cancervol = Pro_can$cancervol
weight = Pro_can$weight
Age = Pro_can$age
Benpros = Pro_can$benpros
Vesinv = Pro_can$vesinv
```

```
Capspen = Pro_can$capspen
Gleason = Pro_can$gleason

psa_cancervol <- lm(log_psa~cancervol, data=Pro_can)
psa_cancervol
summary(psa_cancervol)
plot(cancervol,log_psa,col='blue',main='psa v cancer vol')
abline(psa_cancervol)

psa_weight <- lm(log_psa~weight, data=Pro_can)
psa_weight
summary(psa_weight)
plot(weight,log_psa,col='blue',main='psa v weight')
abline(psa_weight)

psa_Age <- lm(log_psa~Age, data=Pro_can)
psa_Age
summary(psa_Age)
plot(Age,log_psa,col='blue',main='psa v Age')
abline(psa_Age)

psa_Benpros<- lm(log_psa~Benpros, data=Pro_can)
psa_Benpros
summary(psa_Benpros)
plot(Benpros,log_psa,col='blue',main='psa v Benpros')
abline(psa_Benpros)

psa_Vesinv <- lm(log_psa~Vesinv, data=Pro_can)
```

```
psa_Vesinv
summary(psa_Vesinv)
plot(Vesinv,log_psa,col='blue',main='psa v vesinv vol')
abline(psa_Vesinv)

psa_Capspen <- lm(log_psa~Capspen, data=Pro_can)
psa_Capspen
summary(psa_cancervol)
plot(Capspen,log_psa,col='blue',main='psa v Capspen')
abline(psa_Capspen)


psa_Gleason <- lm(log_psa~Gleason, data=Pro_can)
psa_Gleason
summary(psa_Gleason)
plot(Gleason,log_psa,col='blue',main='psa v Gleason')
abline(psa_Gleason)

install.packages("GGally")
library(GGally)
ggpairs(data=Pro_can, columns=c(1:9), title="PSA vs all other predictors")

#Now let's try and fit the whole model

fit1  <- lm(log_psa ~ cancervol+as.factor(Vesinv)+Capspen+Gleason+weight+Age+Benpros)
summary(fit1)

fit2 <- update(fit1,.~. - Capspen -Age -weight)
```

```
summary(fit2)

fit3 <- update(fit2, .~. + Capspen)
summary(fit3)

anova(fit1)
anova(fit2)
anova(fit3)

anova (fit1,fit2,fit3)

anova(fit2,fit3)

AIC_Fwd <- step(lm(log_psa ~ 1), scope = formula(fit1),k=2,trace=0, direction = "forward")
BIC_fwd <- step(lm(log_psa ~ 1), scope = formula(fit1), k = log(32), trace = 0,direction = "forward")
AIC_Bwd <- step(fit1, k = 2, trace = 0, direction = "backward")
BIC_BWD <- step(fit1, k = log(32), trace = 0, direction = "backward")

R_square_Adjst <- data.frame("Method"=c("AIC_Fwd", "BIC_fwd", "AIC_Bwd", "BIC_BWD"),
                             "Adj.r.square"=c(summary(AIC_Fwd)$adj.r.square, summary(BIC_fwd)$adj.r.square,
                                              summary(AIC_Bwd)$adj.r.square, summary(BIC_BWD)$adj.r.square))

summary(AIC_Fwd)

Fst_glm <- glm(fit2)
```

```
Scd_glm <- glm(fit1)
Thd_glm <- glm(fit3)

Fst_glm$aic
Scd_glm$aic
Thd_glm$aic

plot(fitted(fit2),resid(fit2), main = "Residual Plot",col="blue")
abline(h=0)

qqnorm(resid(fit2),col="red")
qqline(resid(fit2))

plot(resid(fit2),main = "Time Series Plot",col="red",type = "l")
abline(h=0)
lm(formula = log_psa ~ cancervol + as.factor(Vesinv) + Gleason + Benpros)
table(as.factor(Vesinv))
table(Gleason)
mean(Benpros)
mean(cancervol)

summary(fit2)
```