STA 206 - Project Code

Yemisi Obasemo and Monica Orme

12/2/2022

Exploratory Data Analysis will be conducted on the dataset to serve as a guide for model building and selection.

Let's verify the variables' classes

```
sapply(prostate, class)
```

```
##                     PSA_level                 cancer_volume
##                     "numeric"                     "numeric"
##                        weight                           age
##                     "numeric"                     "integer"
## benign_prostatic_hyperplasia     seminal_vesicle_invasion
##                     "numeric"                     "integer"
##          capsular_penetration                 gleason_score
##                     "numeric"                     "integer"
```

The classes of seminal vehicle invasion and gleason score should be factor since these are categorical variables.

```
prostate$seminal_vesicle_invasion  = as.factor(prostate$seminal_vesicle_invasion)
prostate$gleason_score = as.factor(prostate$gleason_score)

sapply(prostate, class)
```

```
##                     PSA_level                 cancer_volume
##                     "numeric"                     "numeric"
##                        weight                           age
##                     "numeric"                     "integer"
## benign_prostatic_hyperplasia     seminal_vesicle_invasion
##                     "numeric"                      "factor"
##          capsular_penetration                 gleason_score
##                     "numeric"                      "factor"
```

Let's also check if there are any missing values in the dataset

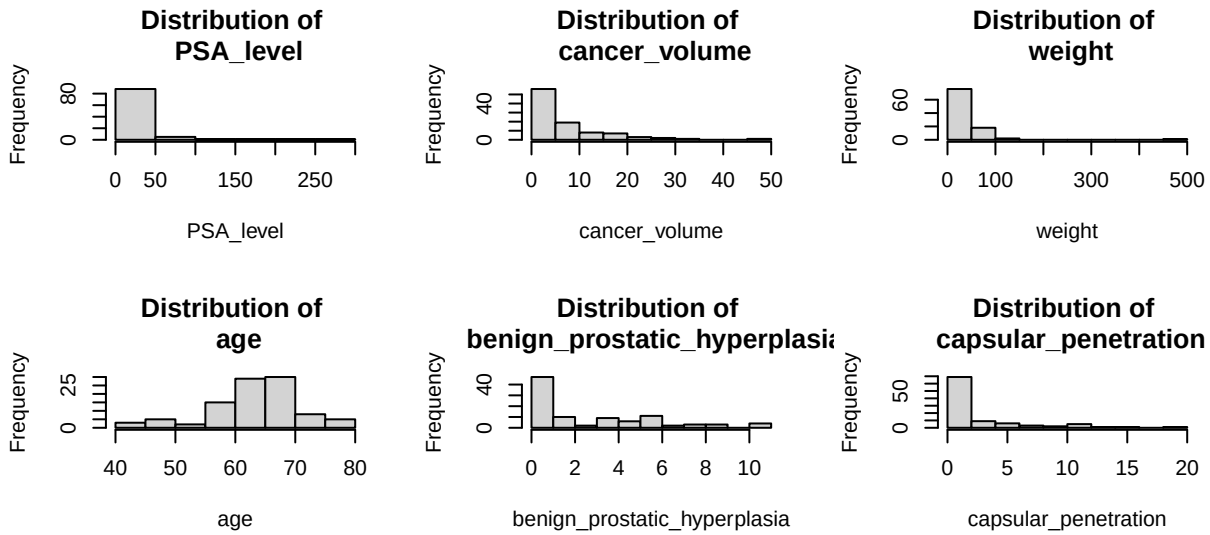
```
sum(is.na(prostate))
```

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

There are no missing values in the dataset since none of the values are NA.

Now, let's verify the distributions of the variables

```
par(mfrow=c(3, 3))

for (i in 1:dim(prostate)[2]) {
  if (class(prostate[,i]) == "numeric" || class(prostate[,i]) == "integer") {
    hist(prostate[,i], xlab= names(prostate)[i],
         main = paste("Distribution of \n", names(prostate)[i]))
  }
}

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



It can be noted that the distribution of the PSA levels, cancer volume, weight, and capsular penetrationare right skewed. The distribution of age is roughly symmetric. The distribution of benign prostatic hyperplasia appears to be bimodal.

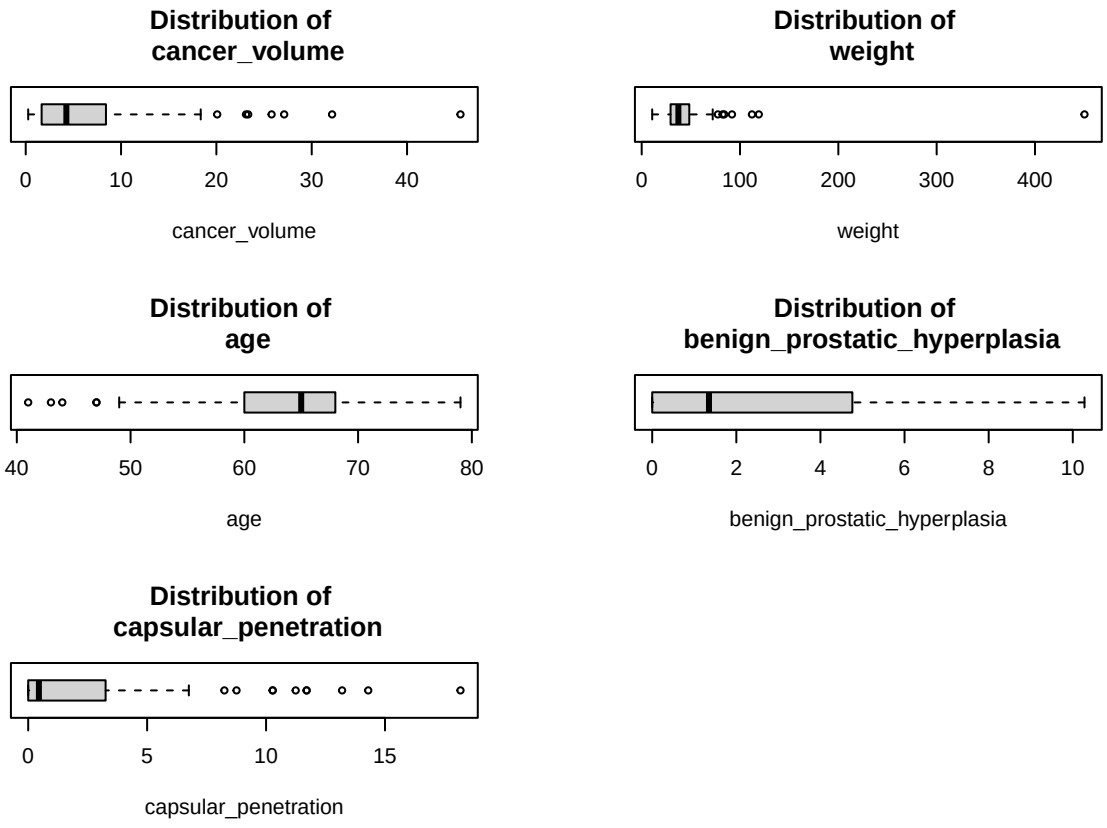
Since there are many zeros for benign prostatic hyperplasia (BPH), we'll add a categorical variable to account for this where 1 indicates the presence of BPH (BPH levels greater than 0) and 0 for absence (BPH levels equal 0).

```
# prostate["prostate_bph_presence"] = as.factor(0)

bph_presence = rep(0, nrow(prostate))
bph_presence[prostate$benign_prostatic_hyperplasia>0] = 1
bph_presence = as.factor(bph_presence)


prostate["bph_presence"] = bph_presence
```

```
# create columns

# labels = prostate$benign_prostatic_hyperplasia>0
# prostate$prostate_bph_presence[labels,] = as.factor(1)
# head(prostate$prostate_bph_presence)
```

Let's also look at the boxplots of the quantitative variables:

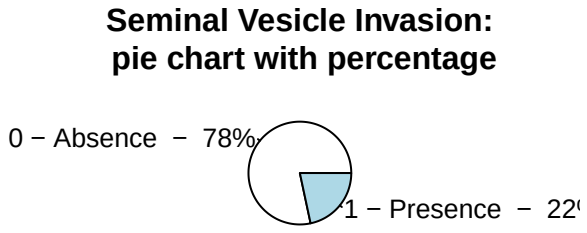
```
par(mfrow=c(3, 2))

for (i in 2:dim(prostate)[2]) {
  if (class(prostate[,i]) == "numeric" || class(prostate[,i]) == "integer") {
    boxplot(prostate[,i], xlab= names(prostate)[i],
         main = paste("Distribution of \n", names(prostate)[i]),
         horizontal = T)
  }
}

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

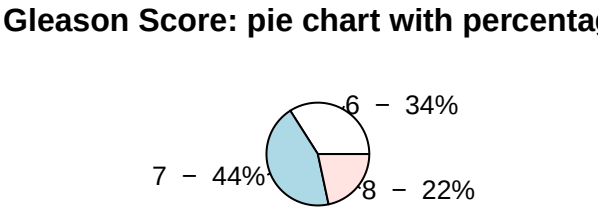


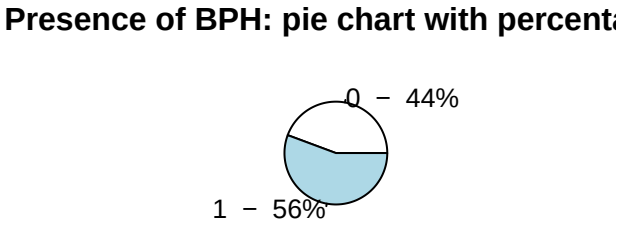
In the boxplots, we can note that the most of the variables with right-skewed distributions contained outliers.

To further understand the relationships that exist, we can create pie charts (with class percentage) for each categorical variable.

```r
# pie chart for seminal_vesicle_invasion
n <- nrow(prostate)
labels <- c("0 - Absence", "1 - Presence")
percent <- round(100*table(prostate$seminal_vesicle_invasion)/n)
lab <- paste(labels, " - ",percent)
lab <- paste(lab,'%',sep='')
# lab
par(mfrow=c(2, 2))
pie(table(prostate$seminal_vesicle_invasion), labels = lab,
main='Seminal Vesicle Invasion:\n pie chart with percentage')


# pie chart for gleason_score
n <- nrow(prostate)
labels <- c("6", "7", "8")
percent <- round(100*table(prostate$gleason_score)/n)
lab <- paste(labels, " - ",percent)
lab <- paste(lab,'%',sep='')
# lab

pie(table(prostate$gleason_score), labels = lab,
main='Gleason Score: pie chart with percentage')


# pie chart for bph_presence
n <- nrow(prostate)
labels <- c("0", "1")
percent <- round(100*table(prostate$bph_presence)/n)
lab <- paste(labels, " - ",percent)
lab <- paste(lab,'%',sep='')
# lab

pie(table(prostate$bph_presence), labels = lab,
main='Presence of BPH: pie chart with percentage')
par(mfrow=c(2, 2))
```





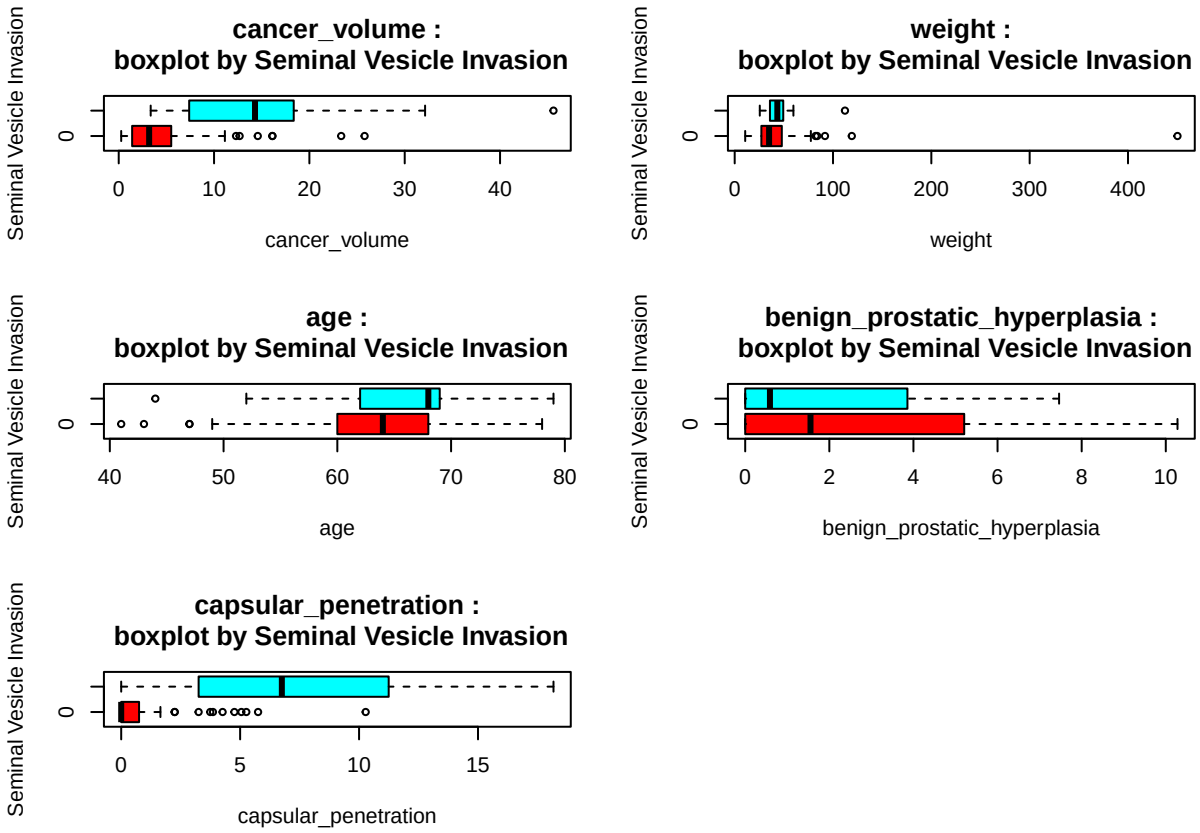


We can note that in the pie charts, most of the participants in this study don't have presence of seminal vesicle invasion by 78% and the remaining 22% have it. 44% of the individuals in the study have a gleason score of 7, 34% have a gleason score of 6, and 22% have a gleason score of 8. This means that more than half of the participants (66%) in this study have a moderate or worse prognosis.

We can also create boxplots of the distributions of each quantitative separated by the levels of the categorical variables.

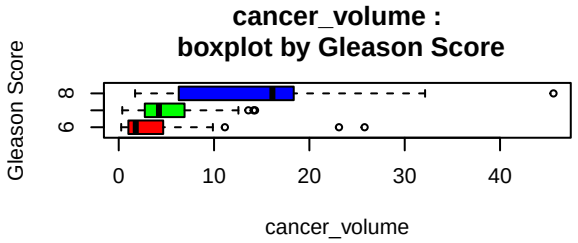
Let's get the boxplots on the basis on of the seminal invasion variable:

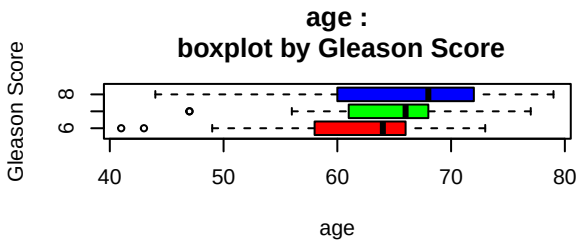
```
par(mfrow=c(3, 2))
quant_pred_columns = c(2, 3, 4, 5, 7)
#

for (i in quant_pred_columns) {
  boxplot(prostate[,i]~prostate$seminal_vesicle_invasion,
        main = paste(names(prostate)[i], ":\n boxplot by Seminal Vesicle Invasion"),
        ylab = "Seminal Vesicle Invasion", xlab = names(prostate)[i],
        col = rainbow(2), horizontal = T)

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

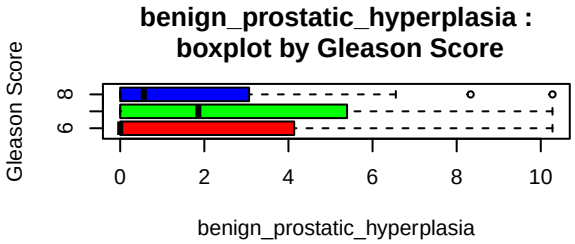


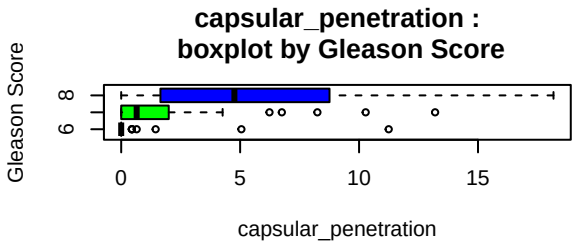
Now, let's get the boxplots on the basis on of the gleason scores:

```
par(mfrow=c(3, 2))
for (i in quant_pred_columns) {
  boxplot(prostate[,i]~prostate$gleason_score,
        main = paste(names(prostate)[i], ":\n boxplot by Gleason Score"),
        ylab = "Gleason Score", xlab = names(prostate)[i],
        col = rainbow(3), horizontal = T)

}

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

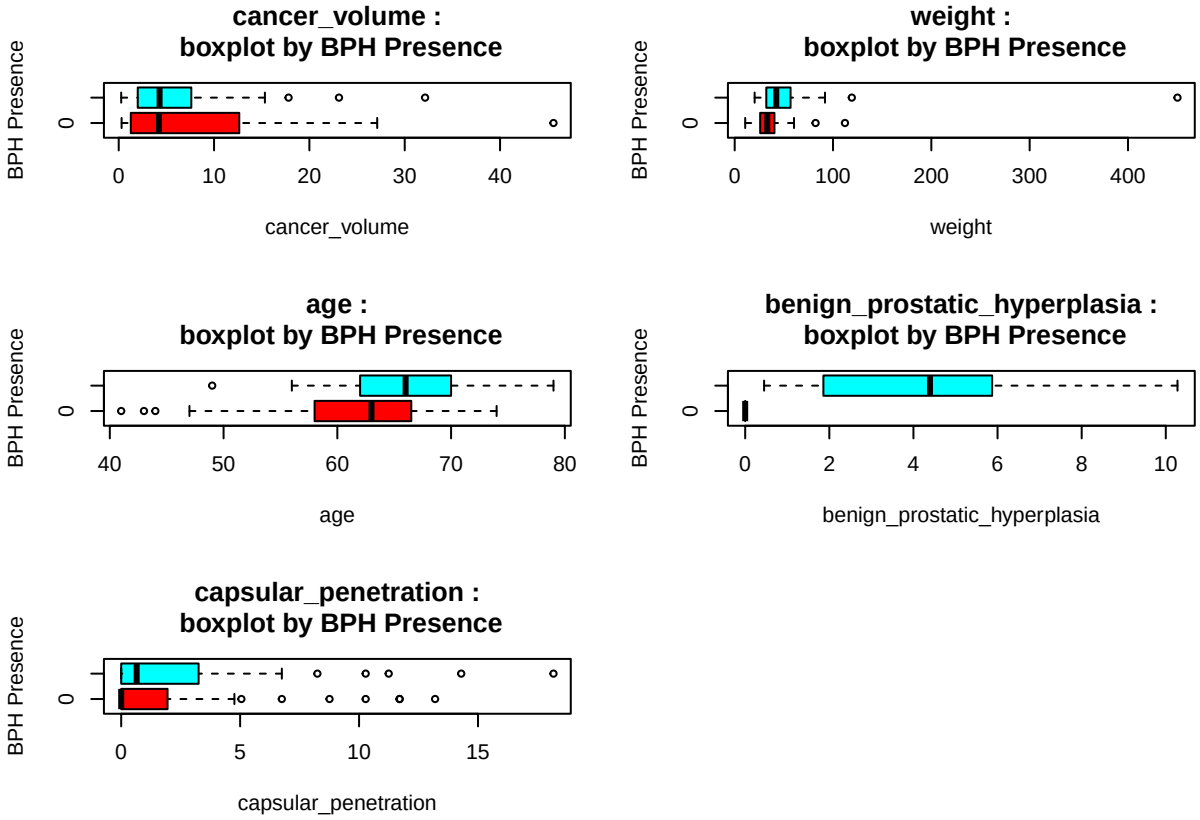






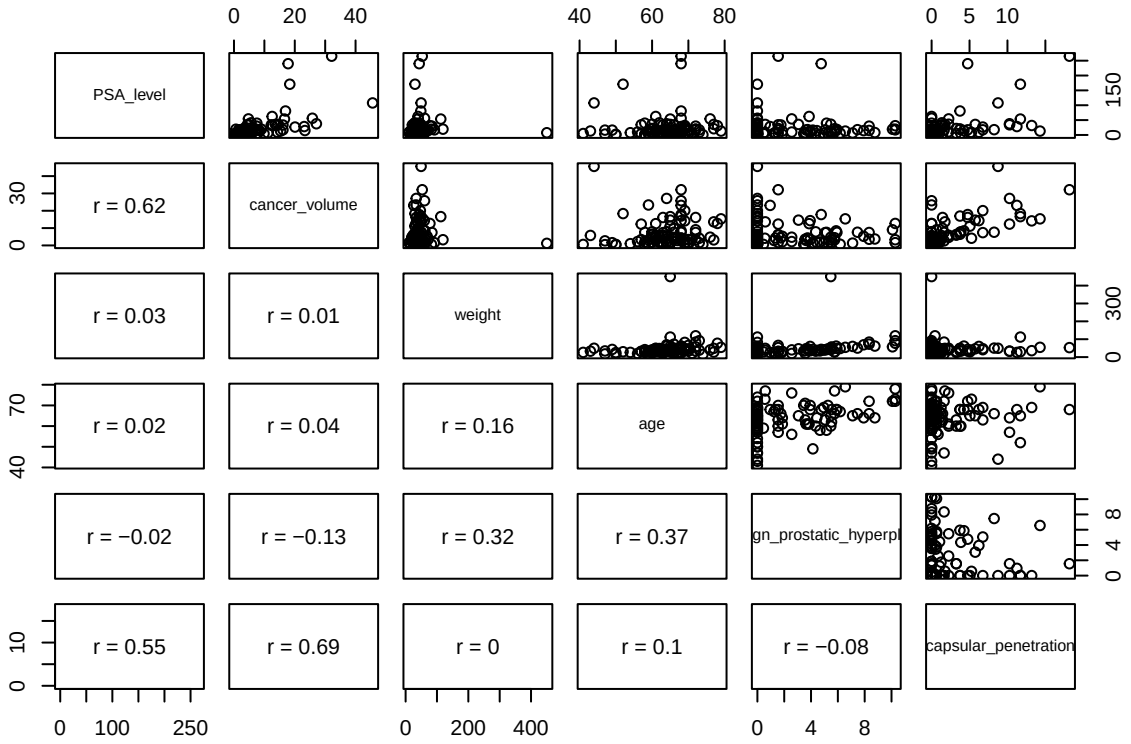



```
par(mfrow=c(3, 2))

for (i in quant_pred_columns) {
  boxplot(prostate[,i]~prostate$bph_presence,
        main = paste(names(prostate)[i], ":\n boxplot by BPH Presence"),
        ylab = "BPH Presence", xlab = names(prostate)[i],
        col = rainbow(2), horizontal = T)

}

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



Let's verify the relationship the PSA levels have with the other quantitative variables in the model:

```
panel.cor <- function(x, y){
  #usr <- par("usr")
  #on.exit(par(usr))
  par(usr = c(0, 1, 0, 1))
  r <- round(cor(x, y, use="complete.obs"), 2)
  txt <- paste0("r = ", r)
  #cex.cor <- 0.8/strwidth(txt)
  text(0.5, 0.5, txt)
}
pairs(prostate[,c(1, quant_pred_columns)], lower.panel = panel.cor)
```



We can see that the relationship between PSA levels and most of the other quantitative variables aren't linear. To address the skewness, we may consider a transformation of the PSA levels. Let's apply the log transformation:

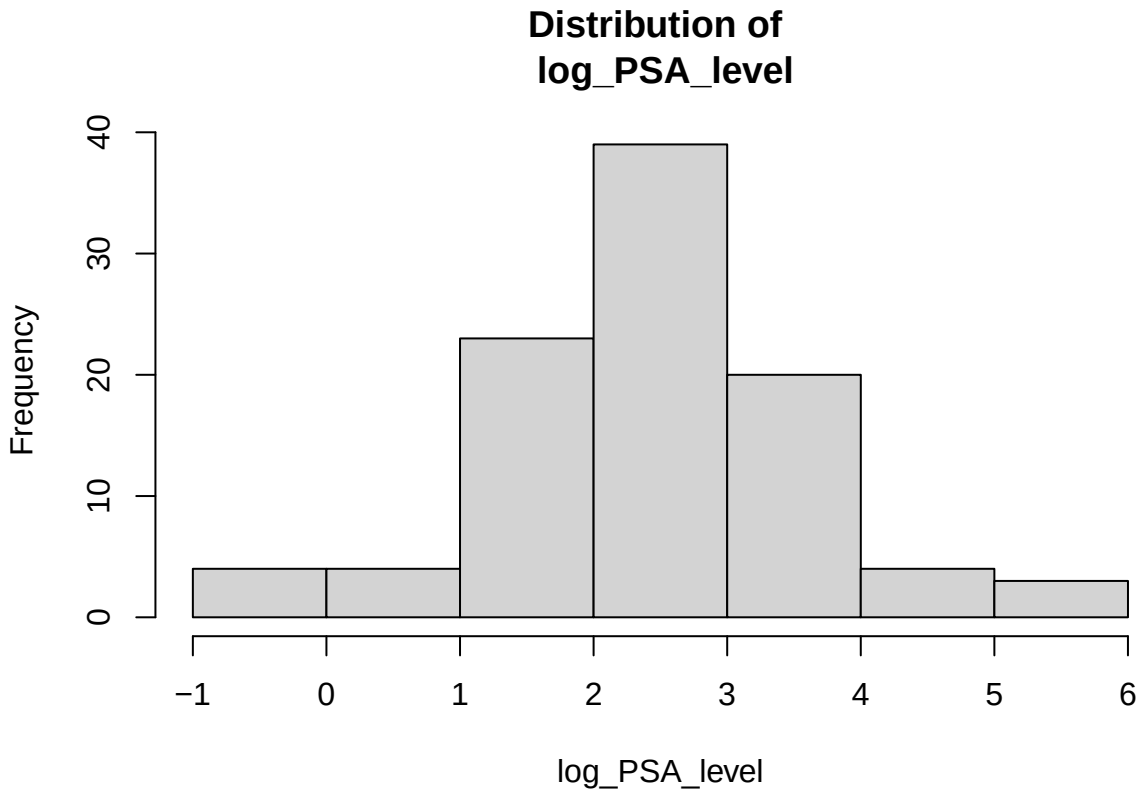
```
prostate["log_PSA_level"] = log(prostate$PSA_level)
head(prostate)
```

```
##   PSA_level cancer_volume weight age benign_prostatic_hyperplasia
## 1     0.651        0.5599 15.959  50                           0
## 2     0.852        0.3716 27.660  58                           0
## 3     0.852        0.6005 14.732  74                           0
## 4     0.852        0.3012 26.576  58                           0
## 5     1.448        2.1170 30.877  62                           0
## 6     2.160        0.3499 25.280  50                           0
##   seminal_vesicle_invasion capsular_penetration gleason_score bph_presence
## 1                        0                    0             6            0
## 2                        0                    0             7            0
## 3                        0                    0             7            0
## 4                        0                    0             6            0
## 5                        0                    0             6            0
## 6                        0                    0             6            0
##   log_PSA_level
## 1    -0.4292456
## 2    -0.1601688
## 3    -0.1601688
## 4    -0.1601688
```

```
## 5     0.3701833
## 6     0.7701082
```

Let's now plot a histogram of the log transformed PSA levels:

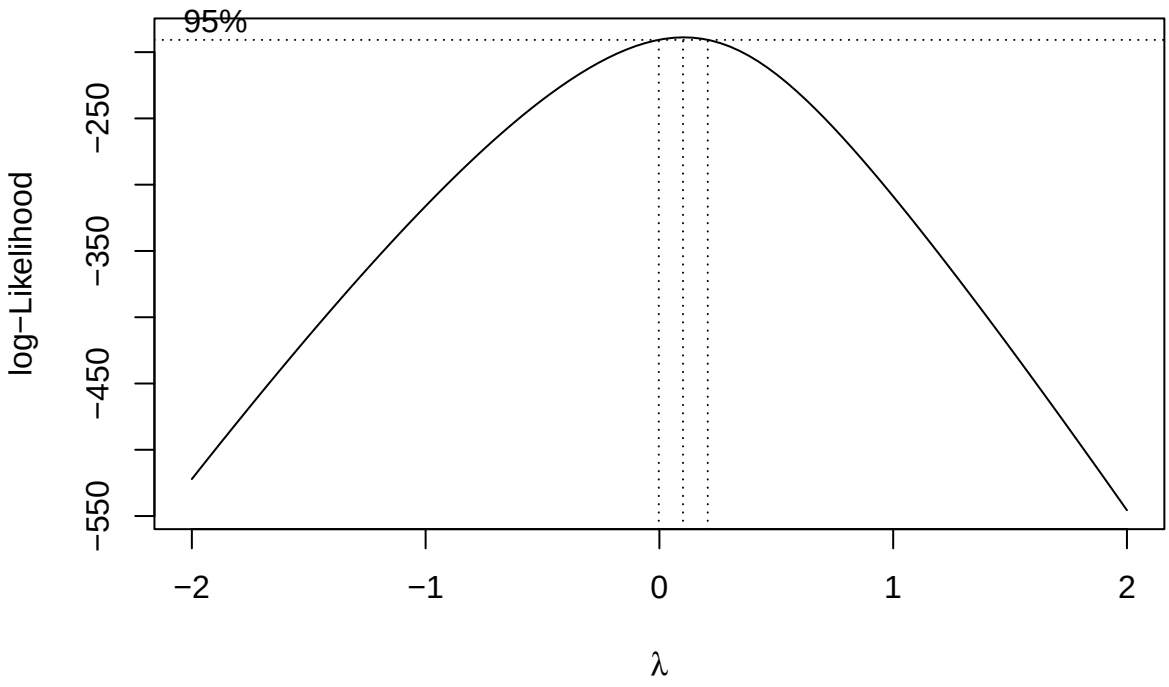
```
hist(prostate[,10], xlab= names(prostate)[10],
         main = paste("Distribution of \n", names(prostate)[10]))
```



We can see that the distribution of the log-transformed PSA levels is symmetric.

Now, let's determine which transformation of Y should be used according to the Box-Cox procedure:

```
library(MASS)
lm_bc = boxcox(lm(PSA_level~.-log_PSA_level, data = prostate))
```

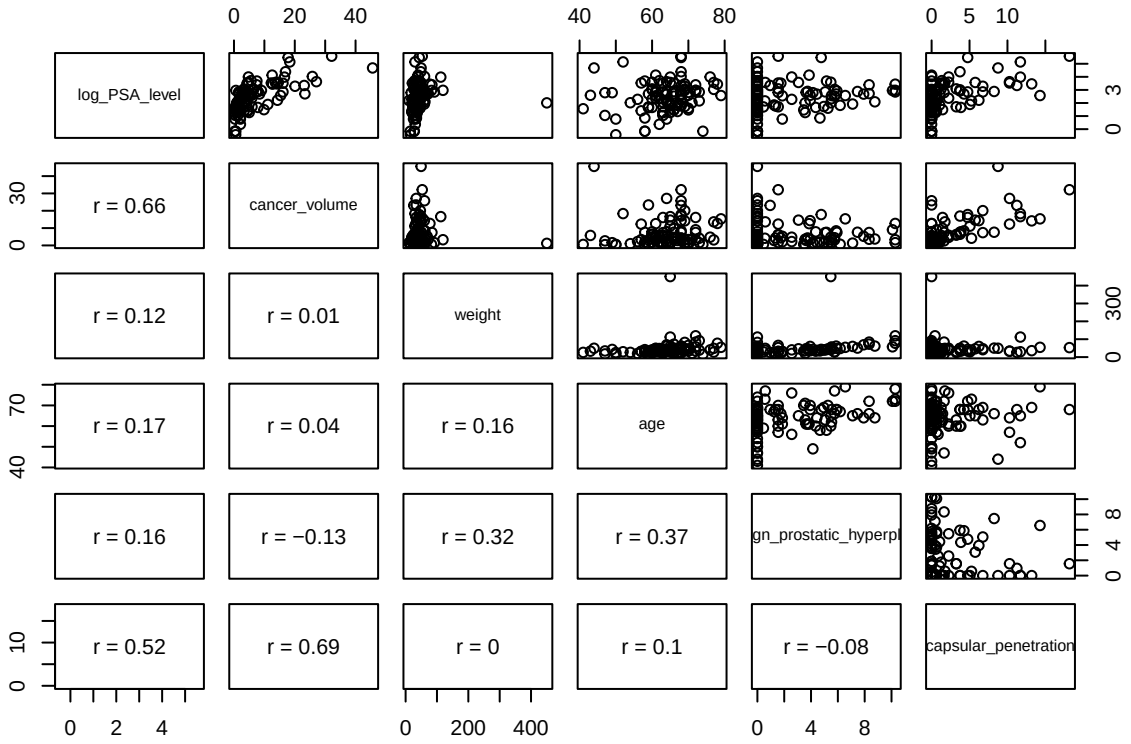



```
#lm_bc$x[which.max(lm_bc$y)]
```

Since the log-likelihood is maximized when lambda = 0 approximately, so we should use the log-transformation on Y.

Now, let's plot the scatterplot matrix of the quantitative variables (excluding PSA levels).

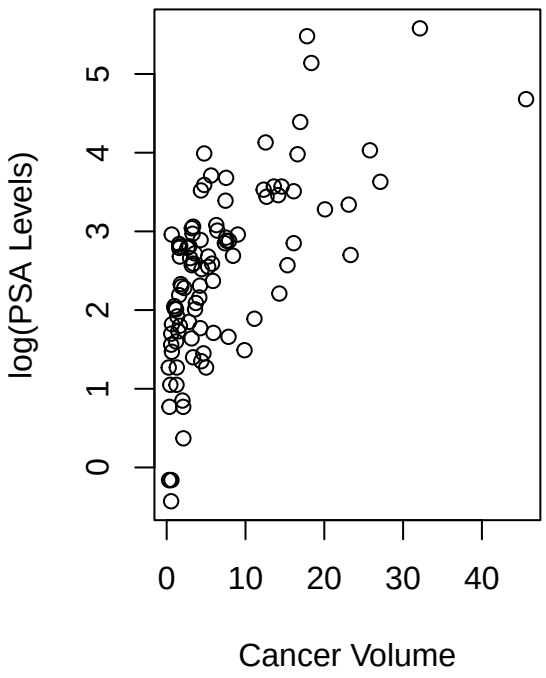
```
pairs(prostate[,c(10, quant_pred_columns)], lower.panel = panel.cor)
```

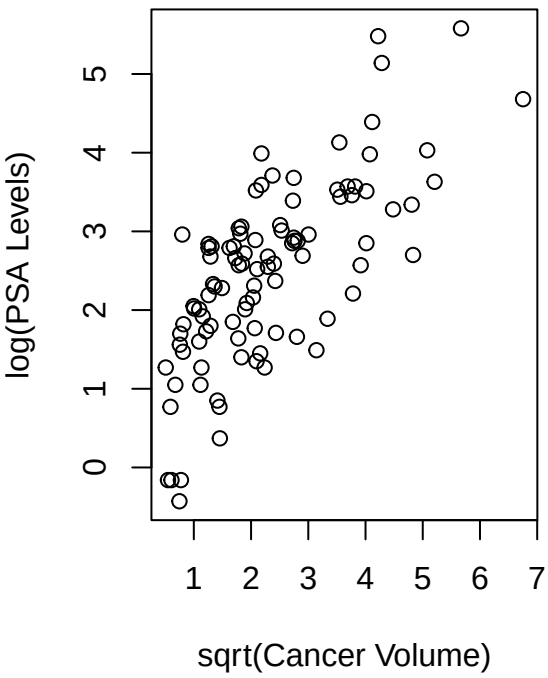


Most of the variables have a linear relationship with log(Y). However, cancer volume doesn't. Also, Most variables have correlation with each other (besides PSA levels). However,the correlation between cancer volume and capsular penetration is high at roughly 0.69.

Since the plot of log(Y) vs. cancer volume is increasing and concaves downward, this suggests that a square root transformation of X is appropriate.

```
par(mfrow=c(1,2))
plot(prostate$cancer_volume,prostate$log_PSA_level, ylab = "log(PSA Levels)",
     xlab = "Cancer Volume")
plot(sqrt(prostate$cancer_volume),prostate$log_PSA_level, ylab = "log(PSA Levels)",
     xlab = "sqrt(Cancer Volume)")
```






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

The trend is more linear after applying the square root transformation to cancer volume.

We can compare the first order models (without interaction) based on the transformation of the cancer volume predictor.

```
fit_1 = lm(log_PSA_level~cancer_volume+weight+age+benign_prostatic_hyperplasia+bph_presence+seminal_ves
summary(fit_1)
```

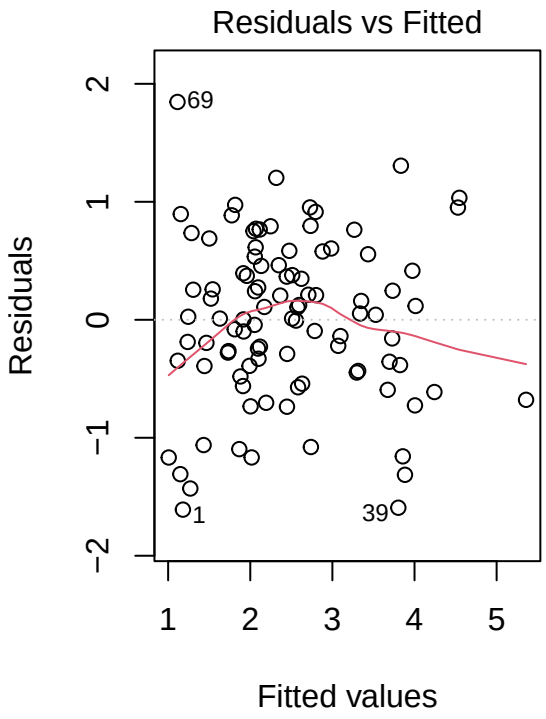
```
##
## Call:
## lm(formula = log_PSA_level ~ cancer_volume + weight + age + benign_prostatic_hyperplasia +
##     bph_presence + seminal_vesicle_invasion + capsular_penetration +
##     gleason_score, data = prostate)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -1.7668 -0.4819  0.0858  0.4890  1.6086
##
## Coefficients:
##                               Estimate Std. Error t value Pr(>|t|)
## (Intercept)                   1.516413   0.716743   2.116  0.03723 *
## cancer_volume                 0.070720   0.015381   4.598 1.44e-05 ***
## weight                        0.001515   0.001839   0.824  0.41230
```

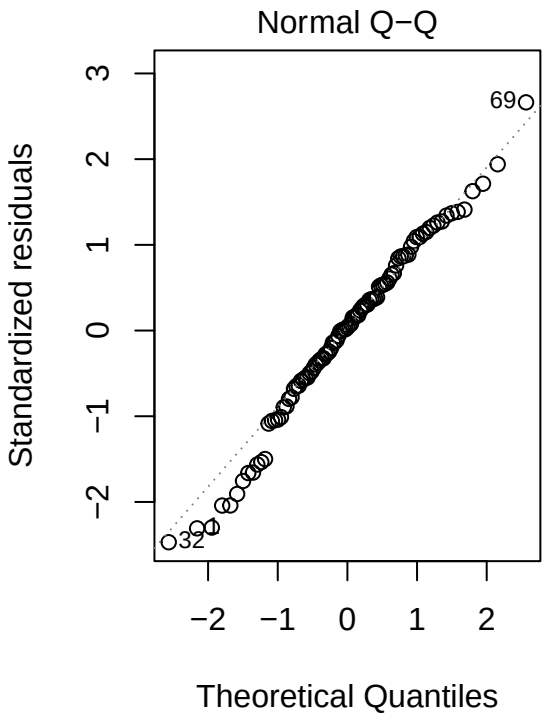
```
## age                          -0.004852   0.011753  -0.413  0.68075
## benign_prostatic_hyperplasia  0.042511   0.041975   1.013  0.31398
## bph_presence1                 0.383503   0.243963   1.572  0.11959
## seminal_vesicle_invasion1     0.790127   0.268379   2.944  0.00415 **
## capsular_penetration         -0.035501   0.033231  -1.068  0.28833
## gleason_score7                0.299326   0.187356   1.598  0.11375
## gleason_score8                0.752814   0.263871   2.853  0.00541 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.765 on 87 degrees of freedom
## Multiple R-squared:  0.6015, Adjusted R-squared:  0.5603
## F-statistic: 14.59 on 9 and 87 DF,  p-value: 4.106e-14
```

```
fit_2 = lm(log_PSA_level~I(sqrt(cancer_volume))+weight+age+benign_prostatic_hyperplasia+bph_presence+se
summary(fit_2)
```

```
##
## Call:
## lm(formula = log_PSA_level ~ I(sqrt(cancer_volume)) + weight +
##     age + benign_prostatic_hyperplasia + bph_presence + seminal_vesicle_invasion +
##     capsular_penetration + gleason_score, data = prostate)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.60907 -0.39191  0.02564  0.46287  1.84616
##
## Coefficients:
##                               Estimate Std. Error t value Pr(>|t|)
## (Intercept)                   1.336175   0.681033   1.962  0.05296 .
## I(sqrt(cancer_volume))        0.509993   0.087967   5.798 1.06e-07 ***
## weight                        0.001873   0.001742   1.075  0.28539
## age                          -0.011357   0.011131  -1.020  0.31042
## benign_prostatic_hyperplasia  0.034346   0.039819   0.863  0.39076
## bph_presence1                 0.394717   0.230603   1.712  0.09052 .
## seminal_vesicle_invasion1     0.731401   0.254611   2.873  0.00511 **
## capsular_penetration         -0.046871   0.031321  -1.496  0.13814
## gleason_score7                0.229795   0.177168   1.297  0.19804
## gleason_score8                0.664539   0.250809   2.650  0.00957 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.7244 on 87 degrees of freedom
## Multiple R-squared:  0.6427, Adjusted R-squared:  0.6058
## F-statistic: 17.39 on 9 and 87 DF,  p-value: 4.449e-16
```

```
par(mfrow=c(1,2))
plot(fit_2, which = c(1:2))
```






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

We can note that there may be extreme cases that are affecting the results of the regression model, so we'll use Cook's distance to identify influencing cases.

```
D=cooks.distance(fit_2)
cooks_criteria = 4/(n - length(fit_2$coefficients))
#which(D>1)

which(D>1)
```

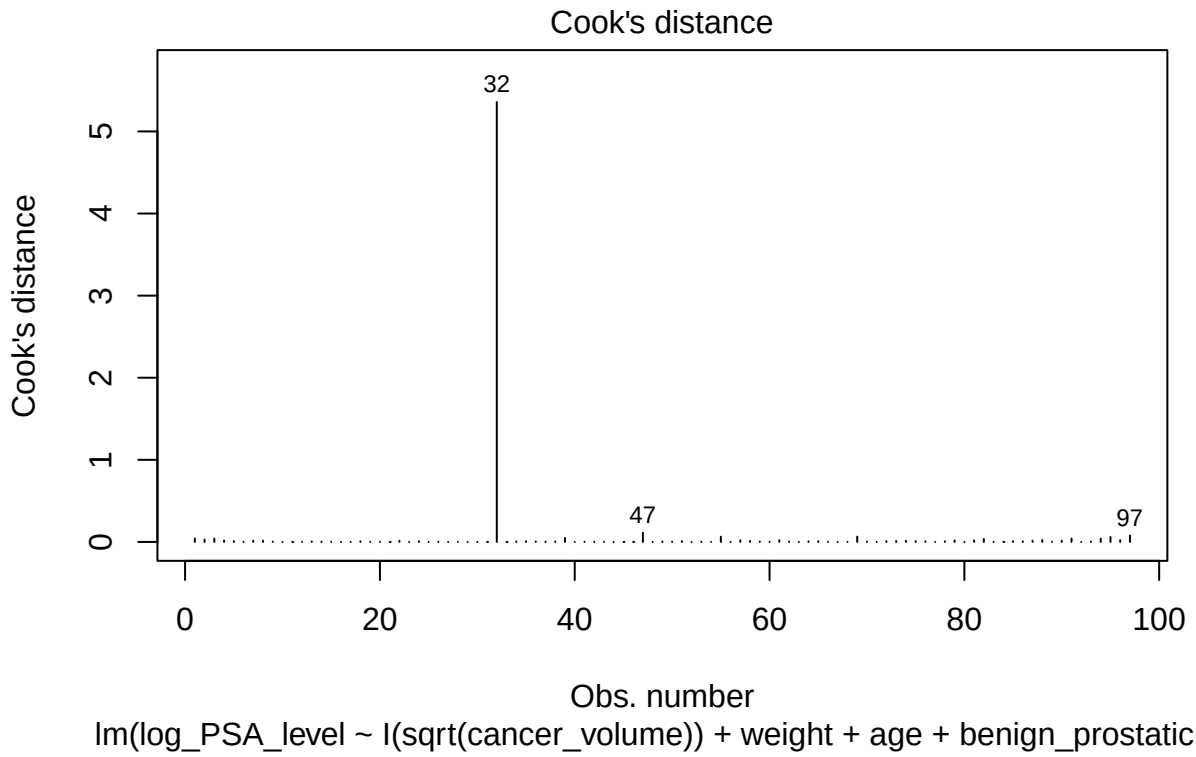
```
## 32
## 32
```

```
which(D>cooks_criteria)
```

```
## 32 39 47 55 69 95 97
## 32 39 47 55 69 95 97
```

It can be noted that there are a few observations that appear to be a influential cases based on Cook's distance. We'll investigate this further to determine if it has a significant affect on the model and if we should actually remove it.

```r
plot(fit_2, which = 4)
```



It can be noted that the 32nd observation (from the original dataset) is much more influential compared to the other ones in this study based on Cook's distance.

We can also look at the percentage change in the fitted values when we remove the 32nd observation.

Hence, we'll remove this from the analysis since this individual doesn't represent a typical member of the population.

```r
prostate_no_32 = prostate[-which(rownames(prostate)==32),]

fit_2_no_32 = lm(log_PSA_level~I(sqrt(cancer_volume))+weight+age+benign_prostatic_hyperplasia+bph_prese
summary(fit_2_no_32)
```

```
##
## Call:
## lm(formula = log_PSA_level ~ I(sqrt(cancer_volume)) + weight +
##     age + benign_prostatic_hyperplasia + bph_presence + seminal_vesicle_invasion +
##     capsular_penetration + gleason_score, data = prostate_no_32)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.76477 -0.37462 -0.01455  0.43365  1.47357
##
## Coefficients:
```

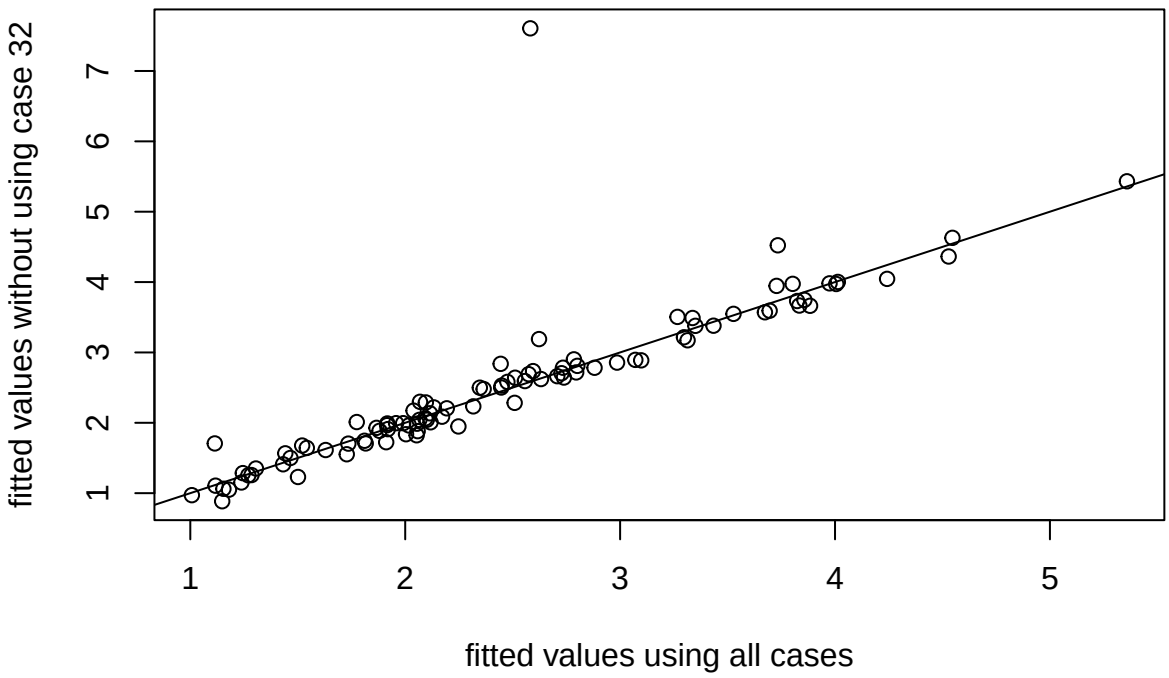
```
##                                Estimate Std. Error t value Pr(>|t|)
## (Intercept)                    1.283697   0.660847   1.943  0.05535 .
## I(sqrt(cancer_volume))         0.489454   0.085698   5.711 1.57e-07 ***
## weight                         0.014278   0.005156   2.769  0.00688 **
## age                           -0.016554   0.010987  -1.507  0.13555
## benign_prostatic_hyperplasia -0.028584   0.045849  -0.623  0.53464
## bph_presence1                  0.589548   0.236380   2.494  0.01454 *
## seminal_vesicle_invasion1      0.711679   0.247066   2.881  0.00501 **
## capsular_penetration          -0.050118   0.030404  -1.648  0.10292
## gleason_score7                 0.236898   0.171855   1.378  0.17163
## gleason_score8                 0.593628   0.244844   2.425  0.01742 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.7025 on 86 degrees of freedom
## Multiple R-squared:  0.6672, Adjusted R-squared:  0.6324
## F-statistic: 19.16 on 9 and 86 DF,  p-value: < 2.2e-16
```

```
# per_change=abs((fit3$fitted-predict.lm(fit3.no3, fat[,1:2]))/fit3$fitted)*100
per_change=abs((fit_2$fitted-predict.lm(fit_2_no_32, prostate[,-c(1,10)]))/fit_2$fitted)*100
summary(per_change)
```

```
##      Min.   1st Qu.    Median      Mean   3rd Qu.      Max.
##    0.1812    1.7958    3.6450    7.4418    6.5848 194.5747
```

The percentage difference in the predictions of the response range between .18% and 194.6%. Based on this range, case #32 (from the original dataset) has a notable influence on the predictions.

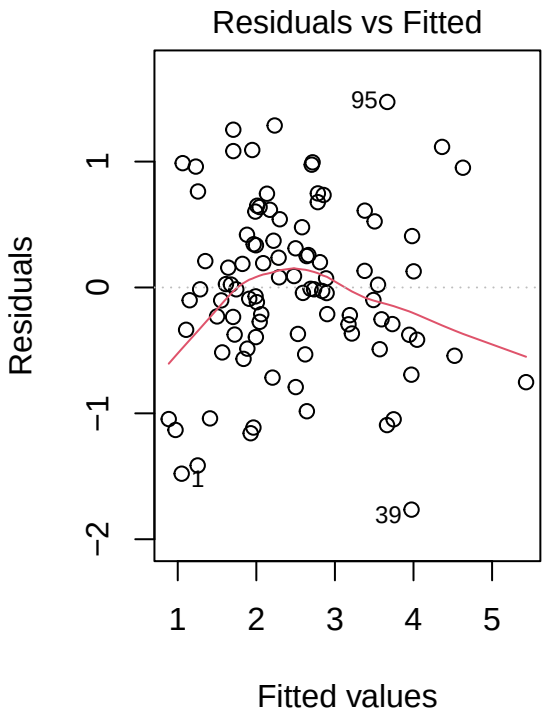
```
plot(fit_2$fitted.values, predict(fit_2_no_32, prostate[,-c(1,10)]), xlab="fitted values using all cases
abline(0,1)
```

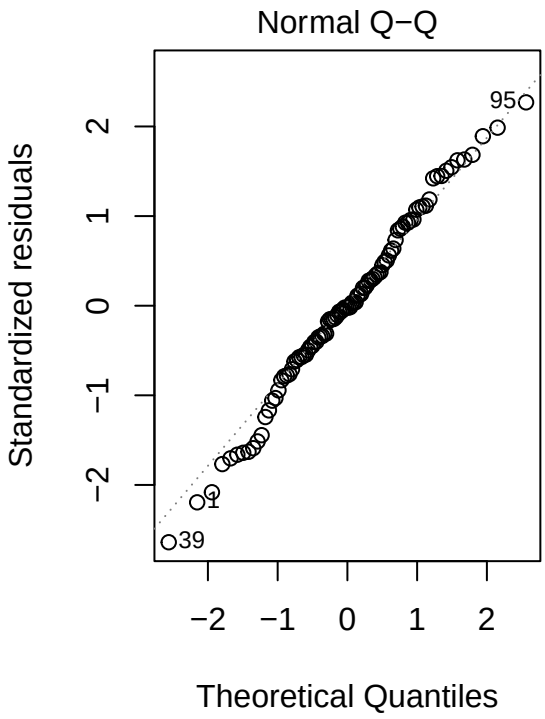


It can be noted that most of the points are close to a straight line (with slight deviations) and there is one observation that noticeably deviates from that line. Hence, the 32nd observation will be removed from this analysis since the attributes from this study are significantly different from the rest.

```
fit_3 = fit_2_no_32
```

~~~

```
par(mfrow=c(1,2))
plot(fit_3, which = c(1:2))
```
~~~






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

We can see that the qq plot of the residuals appears to more closely follow a Normal distribution.

Since the transformed response has a linear relationship with the square root of cancer volume, we'll add this variable to our dataset.

```
prostate_1 = prostate
prostate_1["sqrt_cancer_volume"] = sqrt(prostate_1$cancer_volume)
```

```
names(prostate_1)
```

```
##  [1] "PSA_level"                    "cancer_volume"
##  [3] "weight"                       "age"
##  [5] "benign_prostatic_hyperplasia" "seminal_vesicle_invasion"
##  [7] "capsular_penetration"         "gleason_score"
##  [9] "bph_presence"                 "log_PSA_level"
## [11] "sqrt_cancer_volume"
```

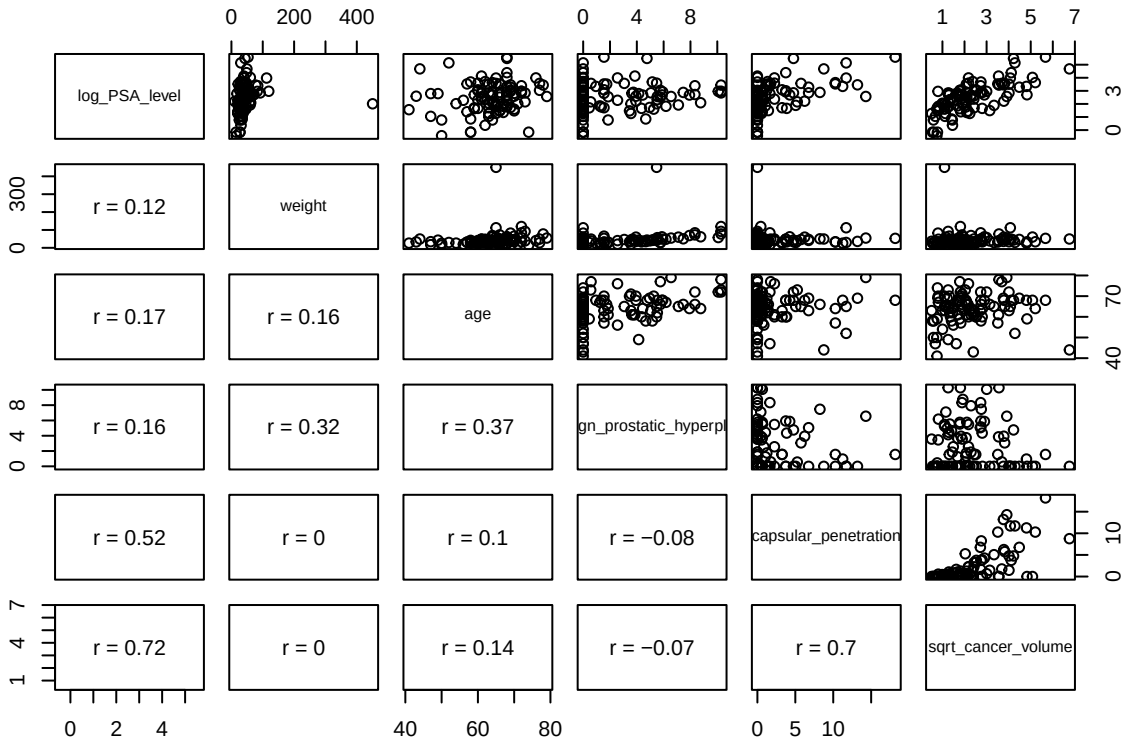
```
# remove the original response and untransformed cancer volume variable
prostate_2 = prostate_1[,c(-1,-2)]
names(prostate_2)
```

```
## [1] "weight"                       "age"
```

```
## [3] "benign_prostatic_hyperplasia" "seminal_vesicle_invasion"
## [5] "capsular_penetration"         "gleason_score"
## [7] "bph_presence"                 "log_PSA_level"
## [9] "sqrt_cancer_volume"
```

Now, let's plot the scatterplot matrix of the quantitative variables (excluding PSA levels).

```
pairs(prostate_2[,c(8, c(1,2,3,5,9))], lower.panel = panel.cor)
```

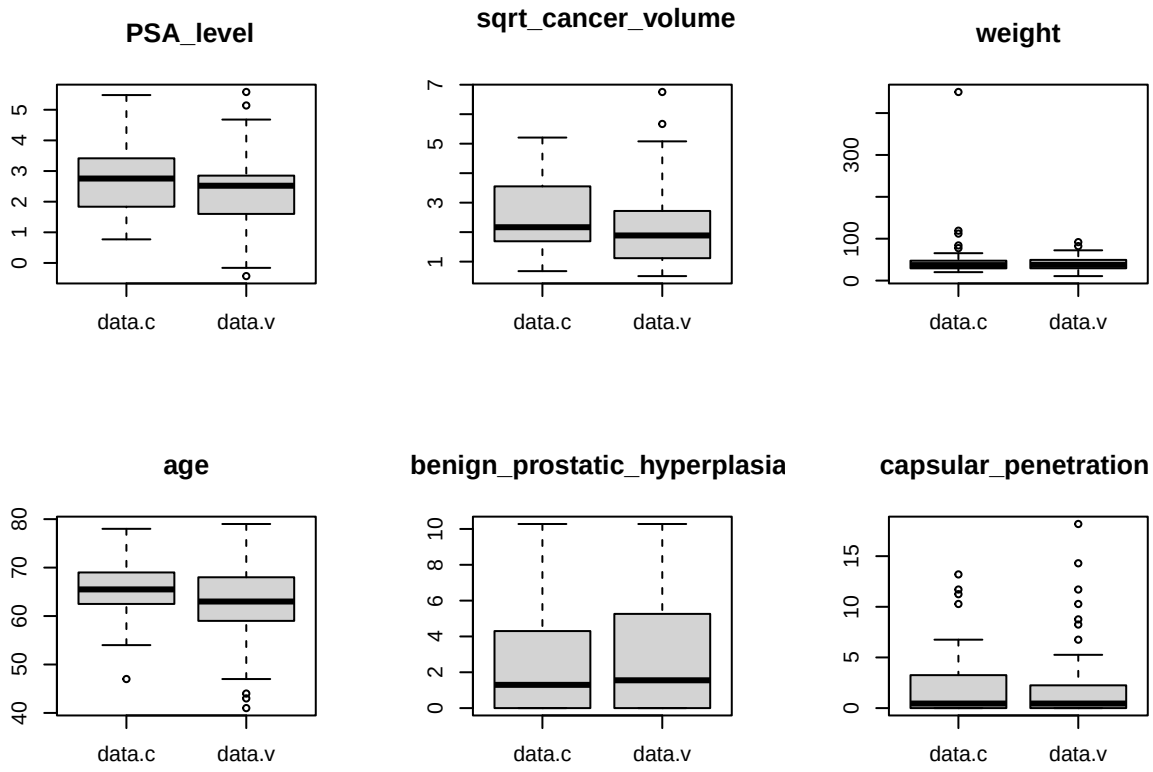



```
set.seed(10) ## set seed for random number generator
##so everyone gets the same split of the data.
n=nrow(prostate_2) ## number of cases in data (96)
 index=sample(1:n, size=n/2, replace=FALSE)
## randomly sample 183 cases to form the training data.
data.c=prostate_2[index,] ## get the training data set.
data.v=prostate_2[-index,] ## the remaining
```

We draw a box plot to check the distribution of variables in the validation and training data

```
 par(mfrow=c(2,3))
boxplot(data.c$log_PSA_level, data.v$log_PSA_level , main=" PSA_level" ,names=c("data.c","data.v"))
boxplot(data.c$sqrt_cancer_volume
  ,data.v$sqrt_cancer_volume
  ,main=" sqrt_cancer_volume
```

```
" ,names=c("data.c","data.v"))
boxplot(data.c$weight,data.v$weight,main="weight",names=c("data.c","data.v"))
boxplot(data.c$age,data.v$age,main="age",names=c("data.c","data.v"))
boxplot(data.c$benign_prostatic_hyperplasia ,data.v$benign_prostatic_hyperplasia ,main="benign_prostati
boxplot(data.c$capsular_penetration ,data.v$capsular_penetration ,main="capsular_penetration ",names=c(
```




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

```
library(leaps)
```

```
## Warning: package 'leaps' was built under R version 4.1.3
```

```
fit = lm(log_PSA_level~.,data=data.c)
fit_summ= summary(fit)

sub_set=regsubsets(log_PSA_level~.,data=data.c,nbest=1,nvmax=9,method="exhaustive")
sum_sub=summary(sub_set)
n=nrow(data.c)
## number of coefficients in each model: p
p.m=as.integer(as.numeric(rownames(sum_sub$which))+1)
sse=sum_sub$rss
aic=n*log(sse/n)+2*p.m
bic=n*log(sse/n)+log(n)*p.m
res_sub=cbind(sum_sub$which,sse,sum_sub$rsq,sum_sub$adjr2,sum_sub$cp, aic, bic)
```

```r
fit0=lm(log_PSA_level~1,data=data.c) ##fit the model with only intercept
fit_full = lm(log_PSA_level ~., data=data.c)
 sse1=sum(fit0$residuals^2)
 p=1
c1=(sse1/fit_summ$sigma^2)-(n-2*p)
aic1=n*log(sse1/n)+2*p
bic1=n*log(sse1/n)+log(n)*p
 none=c(1,rep(1,9),sse1,0,0,c1,bic1,aic1)
 res_sub=rbind(none,res_sub) ##combine the results with other models
 colnames(res_sub)=c(colnames(sum_sub$which),"sse", "R^2", "R^2_a", "Cp", "aic", "bic")
res_sub
```

```
##      (Intercept) weight age benign_prostatic_hyperplasia
## none           1      1   1                            1
## 1              1      0   0                            0
## 2              1      0   0                            0
## 3              1      0   0                            1
## 4              1      0   0                            1
## 5              1      0   0                            1
## 6              1      0   0                            1
## 7              1      0   1                            1
## 8              1      1   1                            1
## 9              1      1   1                            1
##      seminal_vesicle_invasion1 capsular_penetration gleason_score7
## none                         1                    1              1
## 1                            0                    0              0
## 2                            1                    0              0
## 3                            1                    0              0
## 4                            1                    0              0
## 5                            1                    1              0
## 6                            1                    1              1
## 7                            1                    1              1
## 8                            1                    1              1
## 9                            1                    1              1
##      gleason_score8 bph_presence1 sqrt_cancer_volume      sse       R^2
## none              1             1                  1 45.10856 0.0000000
## 1                 0             0                  1 25.86614 0.4265802
## 2                 0             0                  1 23.31628 0.4831075
## 3                 0             0                  1 19.72765 0.5626629
## 4                 1             0                  1 17.83121 0.6047045
## 5                 1             0                  1 17.24384 0.6177258
## 6                 1             0                  1 16.86587 0.6261049
## 7                 1             0                  1 16.64506 0.6309999
## 8                 1             0                  1 16.62430 0.6314603
## 9                 1             1                  1 16.60791 0.6318236
##          R^2_a        Cp         aic         bic
## none 0.0000000 57.211391   0.8890059  -0.9821951
## 1    0.4141146 15.183454 -25.6767818 -21.9343798
## 2    0.4601345 11.349197 -28.6583694 -23.0447663
## 3    0.5328445  5.138171 -34.6806414 -27.1958374
## 4    0.5679328  2.799001 -37.5320337 -28.1760286
## 5    0.5722170  3.455053 -37.1398157 -25.9126096
## 6    0.5713886  4.590228 -36.2036413 -23.1052343
```

```
## 7    0.5664248  6.085017 -34.8361928 -19.8665847
## 8    0.5558624  8.037498 -32.8961198 -16.0553107
## 9    0.5446239 10.000000 -30.9434622 -12.2314521
```

Deciding our final model

```
fit1 = lm(log_PSA_level ~., data=data.c)
library(MASS)
step.f=stepAIC(fit0,scope=list(upper=fit1, lower=~1), direction="both", k=2)
```

```
## Start:  AIC=-0.98
## log_PSA_level ~ 1
##
##                                Df Sum of Sq    RSS      AIC
## + sqrt_cancer_volume            1   19.2424 25.866 -25.6768
## + seminal_vesicle_invasion      1   14.4977 30.611 -17.5926
## + gleason_score                 2   12.0824 33.026 -11.9473
## + capsular_penetration          1    7.9857 37.123  -8.3345
## + age                           1    3.9886 41.120  -3.4259
## <none>                                      45.109  -0.9822
## + benign_prostatic_hyperplasia  1    1.3138 43.795  -0.4010
## + bph_presence                  1    0.6328 44.476   0.3397
## + weight                        1    0.0325 45.076   0.9832
##
## Step:  AIC=-25.68
## log_PSA_level ~ sqrt_cancer_volume
##
##                                Df Sum of Sq    RSS      AIC
## + seminal_vesicle_invasion      1    2.5499 23.316 -28.6584
## + benign_prostatic_hyperplasia  1    2.0067 23.859 -27.5530
## + gleason_score                 2    2.6779 23.188 -26.9227
## + age                           1    1.2977 24.568 -26.1475
## + bph_presence                  1    1.2122 24.654 -25.9808
## <none>                                      25.866 -25.6768
## + weight                        1    0.4520 25.414 -24.5229
## + capsular_penetration          1    0.0684 25.798 -23.8038
## - sqrt_cancer_volume            1   19.2424 45.109  -0.9822
##
## Step:  AIC=-28.66
## log_PSA_level ~ sqrt_cancer_volume + seminal_vesicle_invasion
##
##                                Df Sum of Sq    RSS     AIC
## + benign_prostatic_hyperplasia  1    3.5886 19.728 -34.681
## + gleason_score                 2    2.5572 20.759 -30.234
## + bph_presence                  1    1.6199 21.696 -30.115
## + age                           1    1.5360 21.780 -29.929
## + capsular_penetration          1    1.1723 22.144 -29.134
## <none>                                      23.316 -28.658
## + weight                        1    0.4423 22.874 -27.578
## - seminal_vesicle_invasion      1    2.5499 25.866 -25.677
## - sqrt_cancer_volume            1    7.2946 30.611 -17.593
##
## Step:  AIC=-34.68
```

```
## log_PSA_level ~ sqrt_cancer_volume + seminal_vesicle_invasion +
##     benign_prostatic_hyperplasia
##
##                                Df Sum of Sq    RSS     AIC
## + gleason_score                 2    2.3567 17.371 -36.787
## <none>                                      19.728 -34.681
## + capsular_penetration          1    0.6691 19.059 -34.337
## + age                           1    0.1521 19.576 -33.052
## + bph_presence                  1    0.0115 19.716 -32.709
## + weight                        1    0.0044 19.723 -32.691
## - benign_prostatic_hyperplasia  1    3.5886 23.316 -28.658
## - seminal_vesicle_invasion      1    4.1318 23.860 -27.553
## - sqrt_cancer_volume            1    6.2421 25.970 -23.485
##
## Step:  AIC=-36.79
## log_PSA_level ~ sqrt_cancer_volume + seminal_vesicle_invasion +
##     benign_prostatic_hyperplasia + gleason_score
##
##                                Df Sum of Sq    RSS     AIC
## <none>                                      17.371 -36.787
## + capsular_penetration          1    0.5051 16.866 -36.204
## + age                           1    0.1945 17.176 -35.328
## + bph_presence                  1    0.0331 17.338 -34.879
## + weight                        1    0.0034 17.367 -34.797
## - gleason_score                 2    2.3567 19.728 -34.681
## - sqrt_cancer_volume            1    2.5026 19.873 -32.327
## - benign_prostatic_hyperplasia  1    3.3882 20.759 -30.234
## - seminal_vesicle_invasion      1    3.9366 21.308 -28.983
```

```
step.f$anova
```

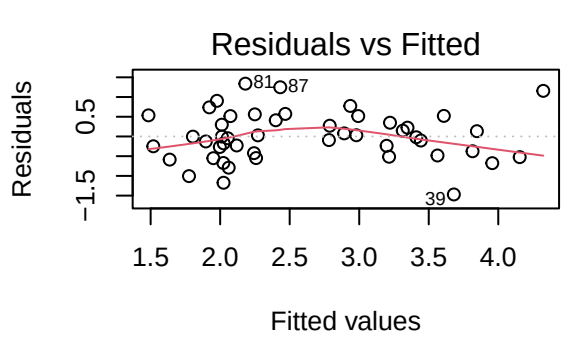
```
## Stepwise Model Path
## Analysis of Deviance Table
##
## Initial Model:
## log_PSA_level ~ 1
##
## Final Model:
## log_PSA_level ~ sqrt_cancer_volume + seminal_vesicle_invasion +
##     benign_prostatic_hyperplasia + gleason_score
##
##
##                              Step Df  Deviance Resid. Df Resid. Dev         AIC
## 1                                                    47   45.10856  -0.9821951
## 2           + sqrt_cancer_volume  1 19.242418        46   25.86614 -25.6767818
## 3     + seminal_vesicle_invasion  1  2.549863        45   23.31628 -28.6583694
## 4 + benign_prostatic_hyperplasia  1  3.588630        44   19.72765 -34.6806414
## 5                + gleason_score  2  2.356694        42   17.37095 -36.7872780
```

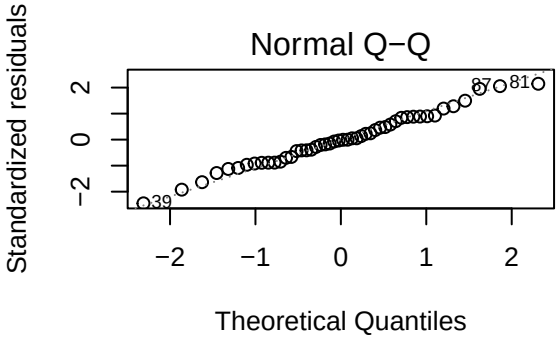
Fitting the best model based on the variable selection process using AIC. We would choose another"good" model based on the adjusted R_squared and compare both

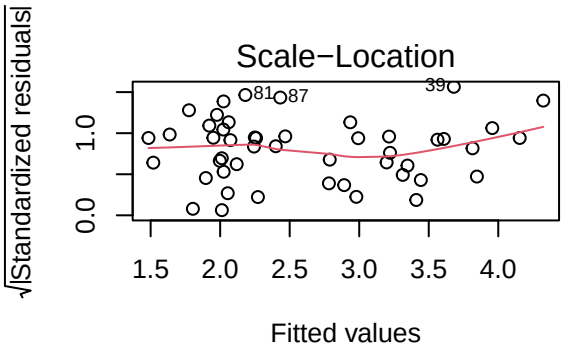
```
Model_final1 = lm(log_PSA_level ~ sqrt_cancer_volume + seminal_vesicle_invasion +
    benign_prostatic_hyperplasia + gleason_score, data = data.c)
summary(Model_final1)
```

```
##
## Call:
## lm(formula = log_PSA_level ~ sqrt_cancer_volume + seminal_vesicle_invasion +
##     benign_prostatic_hyperplasia + gleason_score, data = data.c)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.47054 -0.43833 -0.01212  0.43671  1.33867
##
## Coefficients:
##                              Estimate Std. Error t value Pr(>|t|)
## (Intercept)                   1.21095    0.28813   4.203 0.000135 ***
## sqrt_cancer_volume            0.27208    0.11061   2.460 0.018099 *
## seminal_vesicle_invasion1     0.83690    0.27127   3.085 0.003592 **
## benign_prostatic_hyperplasia  0.09201    0.03215   2.862 0.006535 **
## gleason_score7                0.24185    0.22926   1.055 0.297499
## gleason_score8                0.68956    0.28986   2.379 0.021986 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.6431 on 42 degrees of freedom
## Multiple R-squared:  0.6149, Adjusted R-squared:  0.5691
## F-statistic: 13.41 on 5 and 42 DF,  p-value: 7.866e-08
```

```
par(mfrow=c(2,2))
plot(Model_final1, which = c(1:3))
par(mfrow=c(1,1))
```







From the QQplot and fitted versus Residual plot, we can see a linear relationship between the log PSA level and the chosen X variables and we can say that the final model is a good model

#Data Validation

The "best" model based on forward selection, backward elimination and forward or Backward stepwise with BIC is "log_PSA_level ~ sqrt_cancer_volume + seminal_vesicle_invasion + benign_prostatic_hyperplasia + gleason_score.

```
PRESS_none <- sum((fit0$residuals/(1-influence(fit0)$hat))^2)
PRESS_full <- sum((fit_full$residuals/(1-influence(fit_full)$hat))^2)
PRESS_none
```

```
## [1] 47.04849
```

```
PRESS_full
```

```
## [1] 27.32679
```

```
PRESS_final1 <- sum((Model_final1$residuals/(1-influence(Model_final1)$hat))^2)
PRESS_final1
```

```
## [1] 22.08743
```

Now, let's conduct model validation on the final Model. The "best" model based on R_a^2 and AIC is "log_PSA_level ~ sqrt_cancer_volume + seminal_vesicle_invasion + benign_prostatic_hyperplasia + gleason_score".

The validation data is used to re-run this model.

```
train1 = lm(log_PSA_level ~ sqrt_cancer_volume + seminal_vesicle_invasion +
    benign_prostatic_hyperplasia + gleason_score, data =data.c)
valid1 = lm(log_PSA_level ~ sqrt_cancer_volume + seminal_vesicle_invasion +
    benign_prostatic_hyperplasia + gleason_score, data =data.v)

mod_sum1 = cbind(coef(summary(train1))[,1], coef(summary(valid1))[,1],
coef(summary(train1))[,2], coef(summary(valid1))[,2])
colnames(mod_sum1) = c("Train Est1","Valid Est1","Train s.e1.","Valid s.e1.")
```

And we have the following comparison:

```
mod_sum1
```

```
##                               Train Est1 Valid Est1 Train s.e1. Valid s.e1.
## (Intercept)                   1.21095161 0.72558271   0.2881334  0.27247089
## sqrt_cancer_volume            0.27208481 0.55928274   0.1106114  0.11834660
## seminal_vesicle_invasion1     0.83689816 0.27523053   0.2712690  0.45596768
## benign_prostatic_hyperplasia 0.09201267 0.09061299   0.0321479  0.04098196
## gleason_score7                0.24184981 0.06129269   0.2292620  0.27426810
## gleason_score8                0.68955502 0.62178729   0.2898624  0.46898977
```

Most of the estimated coefficients as well as their standard errors agree somewhat closely on the two data sets.

We can also examine the SSE and adjusted R squares using both the training data and validation data.

```
sse_t1 <- sum(train1$residuals^2)
sse_v1 <- sum(valid1$residuals^2)
Radj_t1 <- summary(train1)$adj.r.squared
Radj_v1 <- summary(valid1)$adj.r.squared
train_sum1 <- c(sse_t1,Radj_t1)
valid_sum1 <- c(sse_v1,Radj_v1)
criteria1 <- rbind(train_sum1,valid_sum1)
colnames(criteria1) <- c("SSE1","R2_adj1")
criteria1
```

```
##                SSE1   R2_adj1
## train_sum1 17.37095 0.5690635
## valid_sum1 29.13088 0.5956635
```

The SSE values are quite far, but the adjusted R squares are very close.

Now what we'd like to do is find the SSE/n under the training model, and compare it to the $MSPE_v$ when we apply our training model to the validation data.

```
#Get MSPE_v from new data
newdata<- data.v[ , -8]
log_PSA_level.hat1 <- predict(train1, newdata)
```

Now that we have the fitted values for the validation set (using the estimated coefficients from the training set), we can find $MSPE_v$. We have that:

$$MSPE_v = \frac{\sum_{j=1}^{m}(Y_j - \hat{Y}_j)^2}{m}$$

Where Y_j is the j-th observation from the validation set, $\hat{Y}_j$ is the j-th fitted value, and m is the number of observations in the validation set.

```
MSPE1 <- mean((data.v$log_PSA_level -log_PSA_level.hat1 )^2)
MSPE1
```

```
## [1] 0.7004003
```

```
SSE_over_N1 = sse_t1/n
SSE_over_N1
```

```
## [1] 0.3618948
```

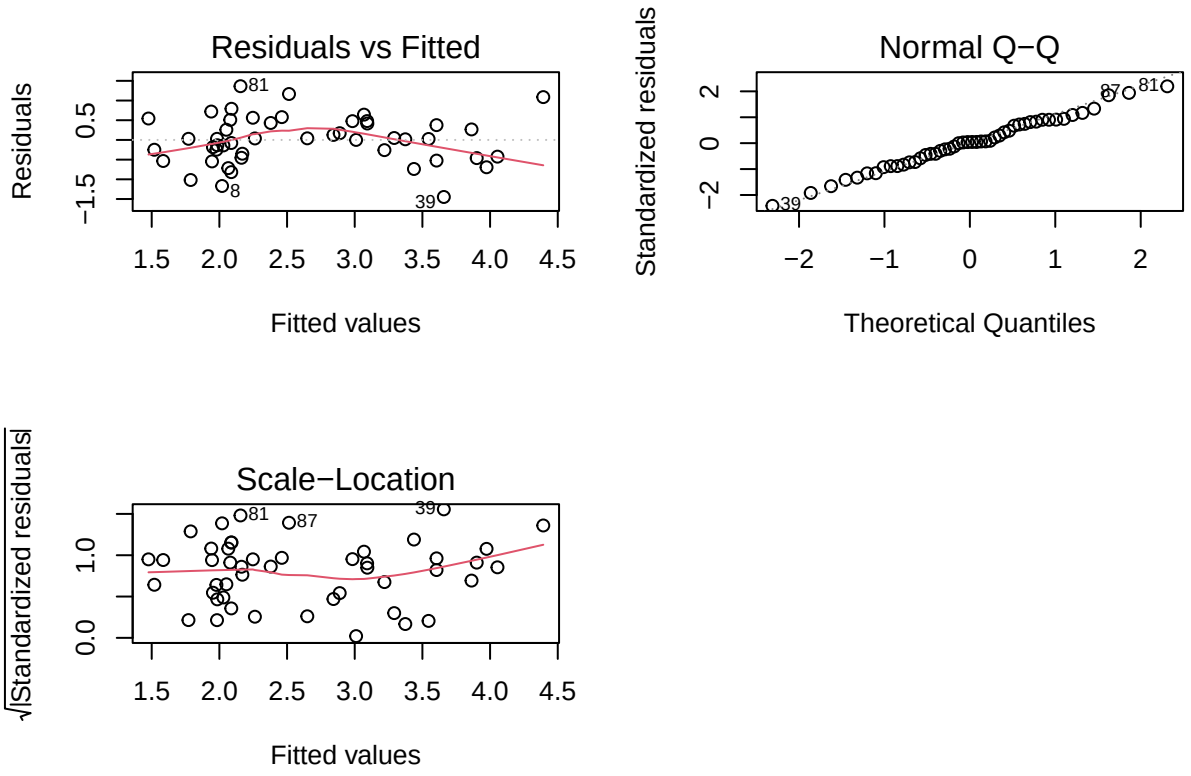
The MSPE is close to the SSE divided by n, so it doesn't overfit the data as much.

Based on the R2_adjusted, we would chose the model below and proceed to compare

```
Model_final2 = lm(log_PSA_level ~ sqrt_cancer_volume + seminal_vesicle_invasion +
    benign_prostatic_hyperplasia + capsular_penetration + gleason_score  , data = data.c)
summary(Model_final2)
```

```
##
## Call:
## lm(formula = log_PSA_level ~ sqrt_cancer_volume + seminal_vesicle_invasion +
##     benign_prostatic_hyperplasia + capsular_penetration + gleason_score,
##     data = data.c)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -1.4483 -0.4318  0.0259  0.4421  1.3638
##
## Coefficients:
##                              Estimate Std. Error t value Pr(>|t|)
## (Intercept)                   1.13546    0.29532   3.845 0.000413 ***
## sqrt_cancer_volume            0.33893    0.12573   2.696 0.010140 *
## seminal_vesicle_invasion1     0.97124    0.29646   3.276 0.002146 **
## benign_prostatic_hyperplasia  0.08699    0.03238   2.687 0.010377 *
## capsular_penetration         -0.04687    0.04230  -1.108 0.274286
## gleason_score7                0.21998    0.22949   0.959 0.343401
## gleason_score8                0.66489    0.28993   2.293 0.027034 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.6414 on 41 degrees of freedom
## Multiple R-squared:  0.6261, Adjusted R-squared:  0.5714
## F-statistic: 11.44 on 6 and 41 DF,  p-value: 1.747e-07
```

```
par(mfrow=c(2,2))
plot(Model_final2, which = c(1:3))
par(mfrow=c(1,1))
```



From the QQplot and fitted versus Residual plot, we can see a linear relationship between the log PSA level and the chosen X variables and we can say that the final model is a good model

#Data Validation

The "best" model based on forward selection, backward elimination and forward or Backward stepwise with BIC is "log_PSA_level ~ sqrt_cancer_volume + seminal_vesicle_invasion + benign_prostatic_hyperplasia + capsular_penetration + gleason_score.

```
PRESS_none <- sum((fit0$residuals/(1-influence(fit0)$hat))^2)
PRESS_full <- sum((fit_full$residuals/(1-influence(fit_full)$hat))^2)
PRESS_none
```

```
## [1] 47.04849
```

```
PRESS_full
```

```
## [1] 27.32679
```

```
PRESS_final2 <- sum((Model_final2$residuals/(1-influence(Model_final2)$hat))^2)
PRESS_final2
```

```
## [1] 22.47607
```

Now, let's conduct model validation on the final Model. The "best" model based on R_a^2 and AIC is "log_PSA_level ~ sqrt_cancer_volume + seminal_vesicle_invasion + benign_prostatic_hyperplasia + capsular_penetration + gleason_score".

The validation data is used to re-run this model.

```
train2 = lm(log_PSA_level ~ sqrt_cancer_volume + seminal_vesicle_invasion +
    benign_prostatic_hyperplasia + capsular_penetration +gleason_score, data =data.c)
valid2 = lm(log_PSA_level ~ sqrt_cancer_volume + seminal_vesicle_invasion +
    benign_prostatic_hyperplasia + capsular_penetration + gleason_score, data =data.v)

mod_sum2 = cbind(coef(summary(train2))[,1], coef(summary(valid2))[,1],
coef(summary(train2))[,2], coef(summary(valid2))[,2])
colnames(mod_sum2) = c("Train Est2","Valid Est2","Train s.e.2","Valid s.e.2")
```

And we have the following comparison:

```
mod_sum2
```

```
##                               Train Est2  Valid Est2 Train s.e.2 Valid s.e.2
## (Intercept)                   1.13545838  0.68231775  0.29532119  0.28624706
## sqrt_cancer_volume            0.33892559  0.58966284  0.12572802  0.13199138
## seminal_vesicle_invasion1     0.97124078  0.39928821  0.29646044  0.51423252
## benign_prostatic_hyperplasia  0.08699184  0.09149602  0.03237966  0.04135697
## capsular_penetration         -0.04686808 -0.02830124  0.04229674  0.05253867
## gleason_score7                0.21998164  0.08946019  0.22949283  0.28146025
## gleason_score8                0.66489101  0.68592116  0.28993499  0.48766607
```

Most of the estimated coefficients as well as their standard errors agree somewhat closely on the two data sets.

We can also examine the SSE and adjusted R squares using both the training data and validation data.

```
sse_t2 <- sum(train2$residuals^2)
sse_v2 <- sum(valid2$residuals^2)
Radj_t2 <- summary(train2)$adj.r.squared
Radj_v2 <- summary(valid2)$adj.r.squared
train_sum2 <- c(sse_t2,Radj_t2)
valid_sum2 <- c(sse_v2,Radj_v2)
criteria2 <- rbind(train_sum2,valid_sum2)
colnames(criteria2) <- c("SSE2","R2_adj2")
criteria2
```

```
##                SSE2   R2_adj2
## train_sum2 16.86587 0.5713886
## valid_sum2 28.93100 0.5888768
```

The SSE values are quite far, but the adjusted R squares are very close.

Now what we'd like to do is find the SSE/n under the training model, and compare it to the $MSPE_v$ when we apply our training model to the validation data.

```
#Get MSPE_v from new data
newdata <- data.v[ , -8]
log_PSA_level.hat2 <- predict(train2, newdata)
```

Now that we have the fitted values for the validation set (using the estimated coefficients from the training set), we can find $MSPE_v$. We have that:

$$MSPE_v = \frac{\sum_{j=1}^{m}(Y_j - \hat{Y}_j)^2}{m}$$

Where Y_j is the j-th observation from the validation set, $\hat{Y}_j$ is the j-th fitted value, and m is the number of observations in the validation set.

```
MSPE2 <- mean((data.v$log_PSA_level -log_PSA_level.hat2 )^2)
MSPE2
```

```
## [1] 0.6911011
```

```
SSE_over_N2 = sse_t2/n
SSE_over_N2
```

```
## [1] 0.3513722
```

The MSPE is close to the SSE divided by n, so it doesn't overfit the data as much.

```
confint(Model_final1,parm=names(Model_final1$coefficients),level=.95)
```

```
##                                  2.5 %    97.5 %
## (Intercept)                  0.62947488 1.7924283
## sqrt_cancer_volume           0.04886199 0.4953076
## seminal_vesicle_invasion1    0.28945524 1.3843411
## benign_prostatic_hyperplasia 0.02713558 0.1568898
## gleason_score7              -0.22081965 0.7045193
## gleason_score8               0.10458909 1.2745209
```

```
confint(valid1,parm=names(valid1$coefficients),level=.95)
```

```
##                                   2.5 %    97.5 %
## (Intercept)                  0.176092796 1.2750726
## sqrt_cancer_volume           0.320614082 0.7979514
## seminal_vesicle_invasion1   -0.644315940 1.1947770
## benign_prostatic_hyperplasia 0.007964983 0.1732610
## gleason_score7              -0.491821638 0.6144070
## gleason_score8              -0.324020712 1.5675953
```

```r
Model1_alldata = lm(log_PSA_level ~ sqrt_cancer_volume + seminal_vesicle_invasion +
    benign_prostatic_hyperplasia + gleason_score, data = prostate_2)

confint(Model1_alldata,parm=names(Model1_alldata$coefficients),level=.95)
```

```
##                                   2.5 %    97.5 %
## (Intercept)                   0.52655546 1.2927517
## sqrt_cancer_volume            0.28430456 0.6006456
## seminal_vesicle_invasion1     0.13893828 1.0519124
## benign_prostatic_hyperplasia  0.03334809 0.1330253
## gleason_score7               -0.16771527 0.5284372
## gleason_score8                0.09708291 1.0866090
```