case study 1 markdown

Case Study 1

```
# Load Libraries for plotting data
library(ggplot2)
library(corrplot)
library(reshape2)
library(faraway)
library(usmap)
library(tidyverse)
library(broom)
library(lmtest)
library(skimr)
library(ggfortify)
library(car)
```

```
# read in the dataset
prostate = read.table('prostate.txt', col.names = c('ID', 'psalevel', 'cancervolume', 'prostateweight',

# remove unneeded predictors
prostate = subset(prostate, select = -c(ID, prostateweight))
head(prostate)
```

```
##   psalevel cancervolume age hyperplasia svi capsular gleason
## 1    0.651       0.5599  50           0   0        0       6
## 2    0.852       0.3716  58           0   0        0       7
## 3    0.852       0.6005  74           0   0        0       7
## 4    0.852       0.3012  58           0   0        0       6
## 5    1.448       2.1170  62           0   0        0       6
## 6    2.160       0.3499  50           0   0        0       6
```

```
prostate.num=subset(prostate, select = -c(svi))
skim_without_charts(prostate)
```

Table 1: Data summary

Name	prostate
Number of rows	97
Number of columns	7
Column type frequency:	
numeric	7

Table 1: Data summary

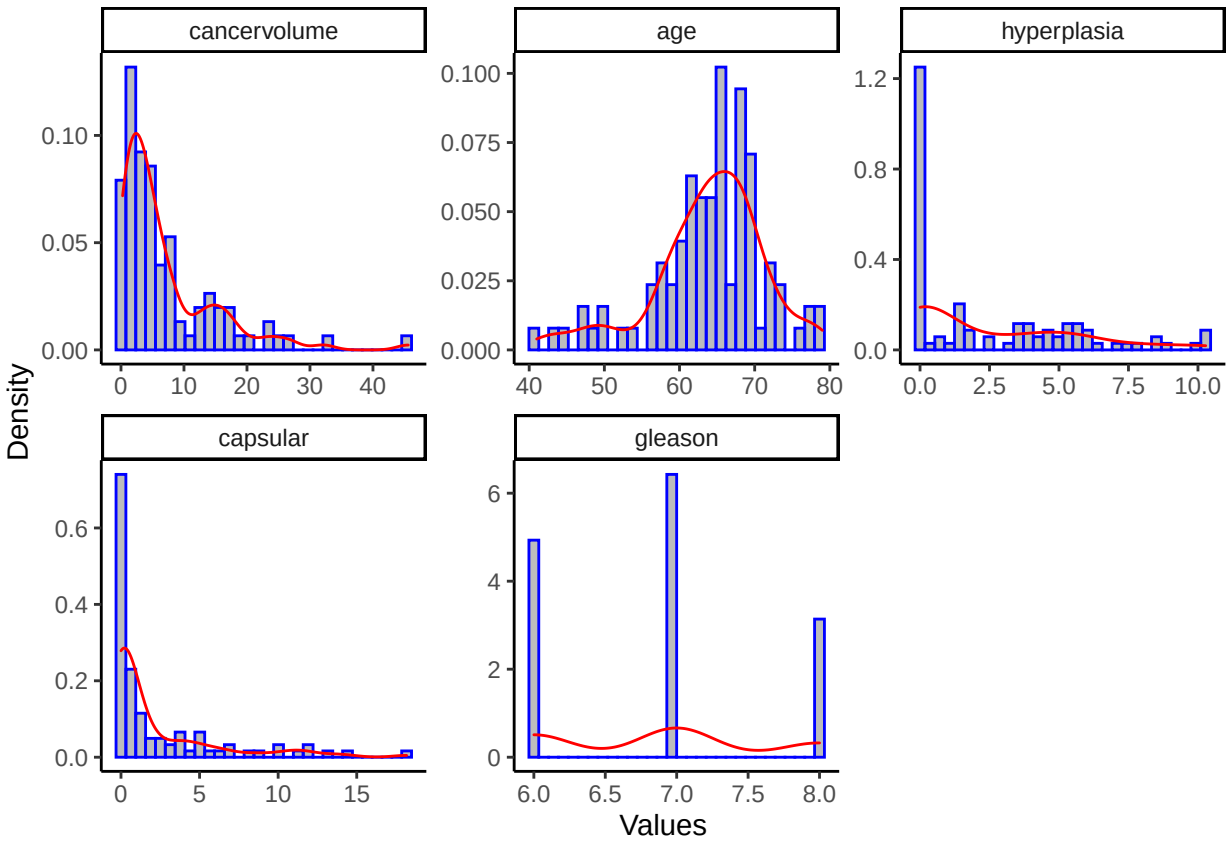
Group variables	None

Variable type: numeric

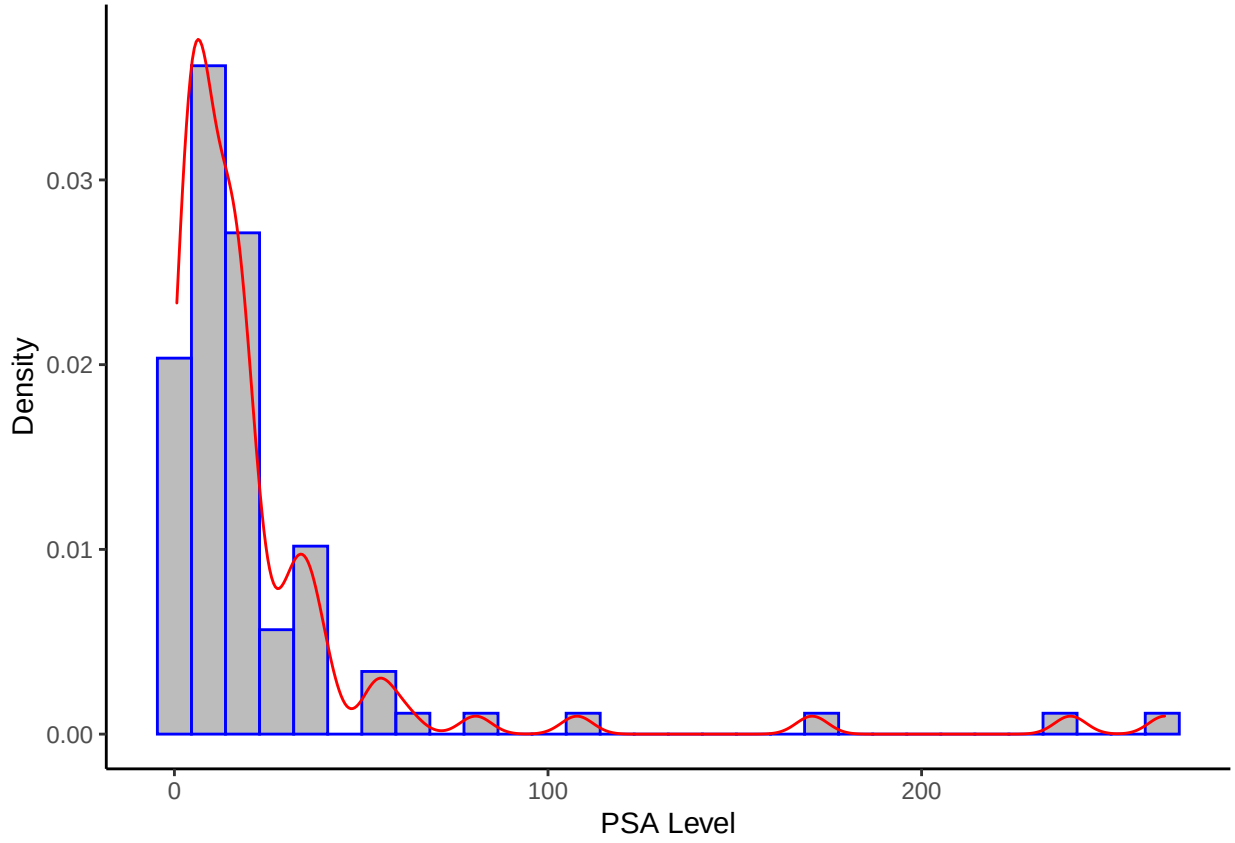
skim_variable	n_missing	complete_rate	mean	sd	p0	p25	p50	p75	p100
psalevel	0	1	23.73	40.78	0.65	5.64	13.33	21.33	265.07
cancervolume	0	1	7.00	7.88	0.26	1.67	4.26	8.41	45.60
age	0	1	63.87	7.45	41.00	60.00	65.00	68.00	79.00
hyperplasia	0	1	2.53	3.03	0.00	0.00	1.35	4.76	10.28
svi	0	1	0.22	0.41	0.00	0.00	0.00	0.00	1.00
capsular	0	1	2.25	3.78	0.00	0.00	0.45	3.25	18.17
gleason	0	1	6.88	0.74	6.00	6.00	7.00	7.00	8.00

```
#Density graph of each predictors
ggplot(melt(subset(prostate.num, select=-c(psalevel))), aes(value)) +
  geom_histogram(aes(y = ..density.. ), alpha = 0.4, color = "blue", bins=30) +
  geom_density(color = "red") + facet_wrap(variable~., scales = "free", ncol = 3) +
  xlab("Values") + ylab("Density") + theme_classic ()
```

```
## No id variables; using all as measure variables
```



```
#Density graph of y-variable (psalevel)
ggplot(prostate.num, aes(x = psalevel)) + geom_histogram(aes(y = ..density.. ), alpha = 0.4, color ="blu
  geom_density(color = "red") + xlab("PSA Level") + ylab("Density") + theme_classic ()
```




```
# create full linear model
prostate.full = lm(psalevel ~ ., data = prostate)
summary(prostate.full)
```

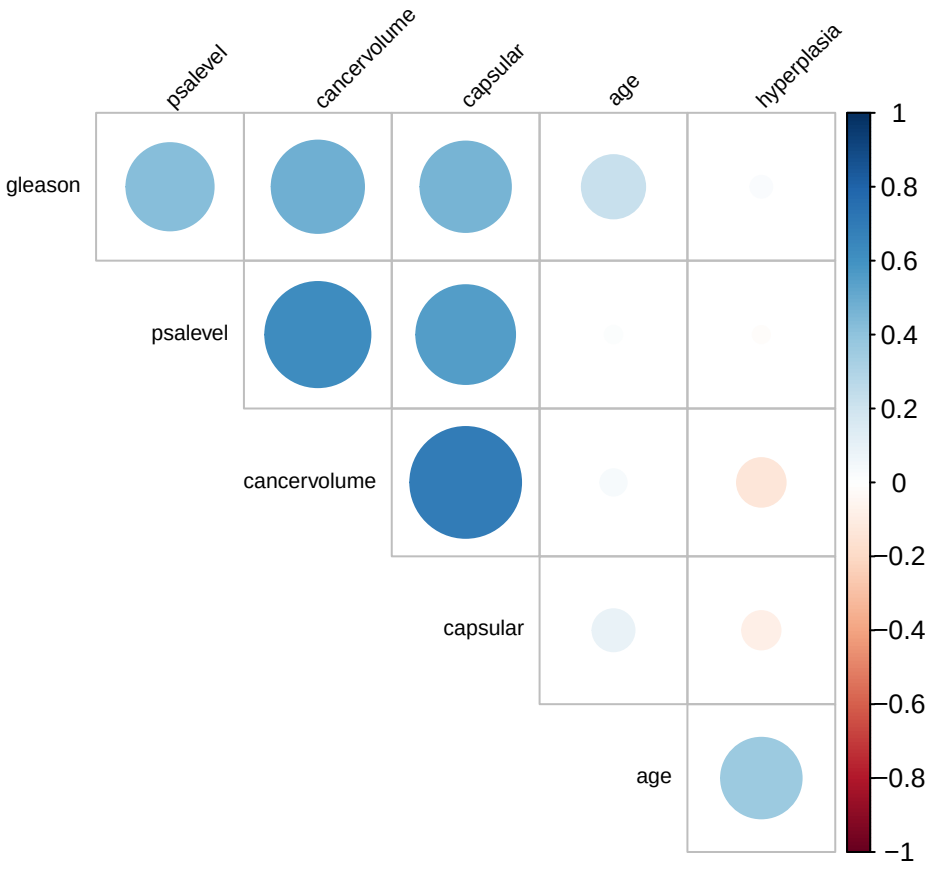
```
##
## Call:
## lm(formula = psalevel ~ ., data = prostate)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -61.491  -8.199  -0.080   5.923 167.267
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  -14.7460    40.1894  -0.367 0.714545
## cancervolume   2.0375     0.5894   3.457 0.000836 ***
## age           -0.5327     0.4724  -1.128 0.262448
## hyperplasia    1.3518     1.1434   1.182 0.240209
## svi           19.6441    10.8303   1.814 0.073038 .
## capsular       1.0974     1.3265   0.827 0.410273
## gleason        6.9942     5.1489   1.358 0.177741
```

```
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 31 on 90 degrees of freedom
## Multiple R-squared:  0.4584, Adjusted R-squared:  0.4223
## F-statistic:  12.7 on 6 and 90 DF,  p-value: 2.481e-10
```

```
# create correlation matrix
round(cor(prostate), dig = 2)
```

```
##              psalevel cancervolume  age hyperplasia   svi capsular gleason
## psalevel         1.00         0.62 0.02       -0.02  0.53     0.55    0.43
## cancervolume     0.62         1.00 0.04       -0.13  0.58     0.69    0.48
## age              0.02         0.04 1.00        0.37  0.12     0.10    0.23
## hyperplasia     -0.02        -0.13 0.37        1.00 -0.12    -0.08    0.03
## svi              0.53         0.58 0.12       -0.12  1.00     0.68    0.43
## capsular         0.55         0.69 0.10       -0.08  0.68     1.00    0.46
## gleason          0.43         0.48 0.23        0.03  0.43     0.46    1.00
```

```
#visualization (heat map) of correlation matrix
corrplot(cor(prostate.num), type = "upper", order = "hclust", diag = FALSE, tl.col = "black", tl.srt = 4
```




```
# compute condition number
x = model.matrix(prostate.full)[,-1]
dim(x)
```

```
## [1] 97  6
```

```
x = x - matrix(apply(x,2, mean), dim(x)[1],dim(x)[2], byrow=TRUE)
x = x / matrix(apply(x, 2, sd), dim(x)[1],dim(x)[2], byrow=TRUE)
head(x)
```

```
##   cancervolume        age hyperplasia        svi   capsular    gleason
## 1   -0.8170143 -1.8624260  -0.8362184 -0.5229409 -0.5934898 -1.1847841
## 2   -0.8409076 -0.7878962  -0.8362184 -0.5229409 -0.5934898  0.1672636
## 3   -0.8118626  1.3611634  -0.8362184 -0.5229409 -0.5934898  0.1672636
## 4   -0.8498407 -0.7878962  -0.8362184 -0.5229409 -0.5934898 -1.1847841
## 5   -0.6194346 -0.2506313  -0.8362184 -0.5229409 -0.5934898 -1.1847841
## 6   -0.8436611 -1.8624260  -0.8362184 -0.5229409 -0.5934898 -1.1847841
```

```
e = eigen(t(x) %*% x)
condition_number = sqrt(e$val[1]/e$val[6])
condition_number
```

```
## [1] 3.152639
```

```
# Since condition number is < 30, collinearity not present

#VIFs = round(vif(x), dig = 2)
#sqrt(VIFs)

# vif(x) sometimes produced an error: "$ operator is invalid for atomic vectors" that we determined aro
df_x = as.data.frame(x)
df_x$psalevel = prostate$psalevel
VIFs = vif(lm(psalevel ~ ., data = df_x))
# VIF is less than 5 for all predictors, confirming a lack of collinearity.
round(sqrt(VIFs), dig = 2)
```

```
## cancervolume          age  hyperplasia          svi     capsular      gleason
##         1.47         1.11         1.10         1.42         1.59         1.20
```

```
# SE is less than 2 times larger than it would have been without collinearity for all variables.

# remove and perform anova on most insignificant variable from model until all variables are significan

prostate.red1 = lm(psalevel ~ cancervolume + age + hyperplasia + svi + gleason, data = prostate)
summary(prostate.red1)
```

```
##
## Call:
## lm(formula = psalevel ~ cancervolume + age + hyperplasia + svi +
##     gleason, data = prostate)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -54.839  -8.758   0.206   5.181 163.883
##
```

```
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  -18.4353    39.8719  -0.462   0.6449
## cancervolume   2.2595     0.5238   4.313 4.07e-05 ***
## age           -0.5261     0.4715  -1.116   0.2674
## hyperplasia    1.3714     1.1412   1.202   0.2326
## svi           23.6477     9.6720   2.445   0.0164 *
## gleason        7.4688     5.1080   1.462   0.1471
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 30.94 on 91 degrees of freedom
## Multiple R-squared:  0.4543, Adjusted R-squared:  0.4243
## F-statistic: 15.15 on 5 and 91 DF,  p-value: 8.245e-11
```

```
anova(prostate.red1, prostate.full)
```

```
## Analysis of Variance Table
##
## Model 1: psalevel ~ cancervolume + age + hyperplasia + svi + gleason
## Model 2: psalevel ~ cancervolume + age + hyperplasia + svi + capsular +
##     gleason
##   Res.Df   RSS Df Sum of Sq      F Pr(>F)
## 1     91 87138
## 2     90 86480  1    657.61 0.6844 0.4103
```

```
prostate.red2 = lm(psalevel ~ cancervolume + hyperplasia + svi + gleason, data = prostate)
summary(prostate.red2)
```

```
##
## Call:
## lm(formula = psalevel ~ cancervolume + hyperplasia + svi + gleason,
##     data = prostate)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -58.262  -9.596   0.477   5.428 164.429
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  -43.4846    32.9960  -1.318   0.1908
## cancervolume   2.2995     0.5233   4.394 2.97e-05 ***
## hyperplasia    0.9001     1.0616   0.848   0.3987
## svi           22.5019     9.6301   2.337   0.0216 *
## gleason        6.3942     5.0231   1.273   0.2062
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 30.99 on 92 degrees of freedom
## Multiple R-squared:  0.4468, Adjusted R-squared:  0.4228
## F-statistic: 18.58 on 4 and 92 DF,  p-value: 3.206e-11
```

```r
anova(prostate.red2, prostate.red1)
```

```
## Analysis of Variance Table
##
## Model 1: psalevel ~ cancervolume + hyperplasia + svi + gleason
## Model 2: psalevel ~ cancervolume + age + hyperplasia + svi + gleason
##   Res.Df   RSS Df Sum of Sq      F Pr(>F)
## 1     92 88330
## 2     91 87138  1    1192.3 1.2452 0.2674
```

```r
prostate.red3 = lm(psalevel ~ cancervolume + hyperplasia + svi, data = prostate)
summary(prostate.red3)
```

```
##
## Call:
## lm(formula = psalevel ~ cancervolume + hyperplasia + svi, data = prostate)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -55.591  -7.018  -0.071   4.065 166.850
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)   -2.0080     5.2227  -0.384  0.70151
## cancervolume   2.5160     0.4965   5.067 2.04e-06 ***
## hyperplasia    1.0601     1.0576   1.002  0.31881
## svi           25.1405     9.4357   2.664  0.00909 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 31.09 on 93 degrees of freedom
## Multiple R-squared:  0.4371, Adjusted R-squared:  0.4189
## F-statistic: 24.07 on 3 and 93 DF,  p-value: 1.295e-11
```

```r
anova(prostate.red3, prostate.red2)
```

```
## Analysis of Variance Table
##
## Model 1: psalevel ~ cancervolume + hyperplasia + svi
## Model 2: psalevel ~ cancervolume + hyperplasia + svi + gleason
##   Res.Df   RSS Df Sum of Sq      F Pr(>F)
## 1     93 89886
## 2     92 88330  1    1555.8 1.6204 0.2062
```

```r
prostate.red4 = lm(psalevel ~ cancervolume + svi, data = prostate)
summary(prostate.red4)
```

```
##
## Call:
## lm(formula = psalevel ~ cancervolume + svi, data = prostate)
##
```

```
## Residuals:
##     Min      1Q  Median      3Q     Max
## -55.145  -7.535  -1.129   4.256 170.018
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)     1.060      4.231   0.251   0.8027
## cancervolume    2.477      0.495   5.003 2.62e-06 ***
## svi            24.647      9.423   2.616   0.0104 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 31.09 on 94 degrees of freedom
## Multiple R-squared:  0.431,  Adjusted R-squared:  0.4189
## F-statistic:  35.6 on 2 and 94 DF,  p-value: 3.098e-12
```

```
anova(prostate.red4, prostate.red3)
```

```
## Analysis of Variance Table
##
## Model 1: psalevel ~ cancervolume + svi
## Model 2: psalevel ~ cancervolume + hyperplasia + svi
##   Res.Df   RSS Df Sum of Sq      F Pr(>F)
## 1     94 90857
## 2     93 89886  1    970.94 1.0046 0.3188
```

```
model.final = lm(psalevel ~ cancervolume + svi, data = prostate)
summary(model.final)
```

```
##
## Call:
## lm(formula = psalevel ~ cancervolume + svi, data = prostate)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -55.145  -7.535  -1.129   4.256 170.018
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)     1.060      4.231   0.251   0.8027
## cancervolume    2.477      0.495   5.003 2.62e-06 ***
## svi            24.647      9.423   2.616   0.0104 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 31.09 on 94 degrees of freedom
## Multiple R-squared:  0.431,  Adjusted R-squared:  0.4189
## F-statistic:  35.6 on 2 and 94 DF,  p-value: 3.098e-12
```

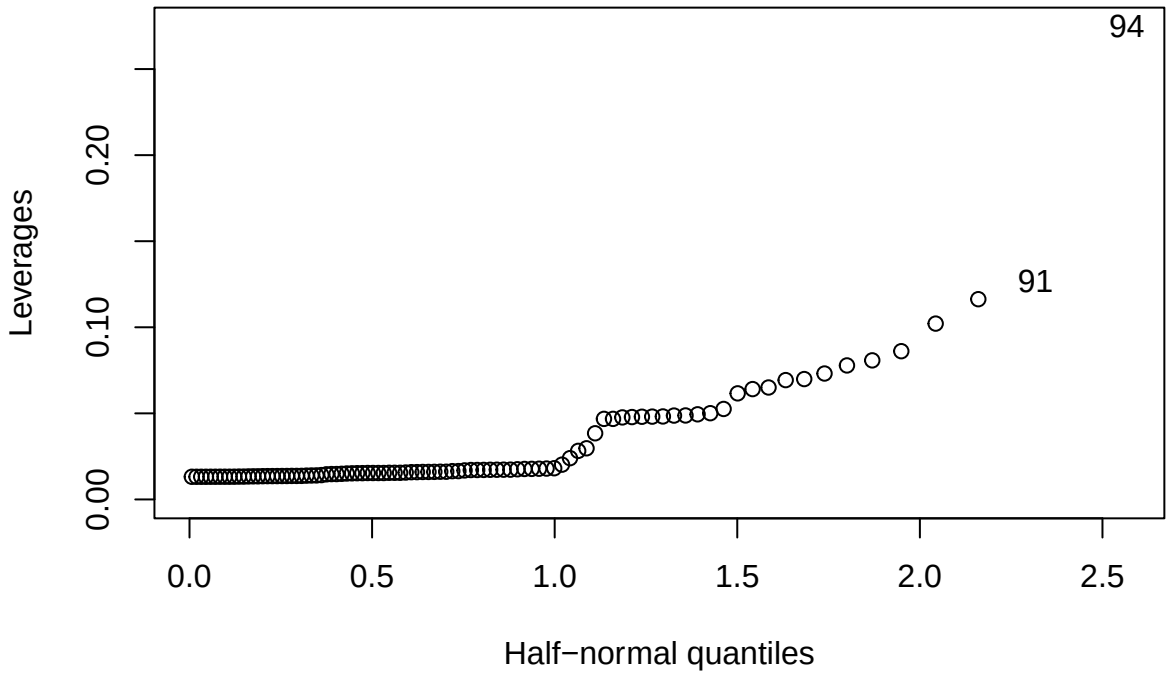
```
#all predictors are now statistically significant

# Model Diagnostics
```

```r
#High leverage points
diag_hat = lm.influence(prostate.red4)$hat
high_lev = diag_hat[diag_hat > (2 * dim(prostate)[2] / dim(prostate)[1])]
high_lev
```

```
##        94
## 0.2747004
```

```r
halfnorm(diag_hat, ylab='Leverages')
```




```r
# Calculate the IQR for the dependent variable
IQR_y = IQR(prostate$psalevel)

# Define a range with its lower limit being (Q1 - IQR) and upper limit being (Q3 + IQR)
QT1_y = quantile(prostate$psalevel,0.25)
QT3_y = quantile(prostate$psalevel,0.75)
lower_lim_y = QT1_y - IQR_y
upper_lim_y = QT3_y + IQR_y
vector_lim_y = c(lower_lim_y,upper_lim_y)

#vector_lim_y

highlev = prostate[diag_hat>(2 * dim(prostate)[2] / dim(prostate)[1]),]
```

```r
# Select only the observations with leverage points outside the range

highlev_lower = highlev[highlev$psalevel < vector_lim_y[1], ]
highlev_upper = highlev[highlev$psalevel > vector_lim_y[2], ]
rbind(highlev_lower,highlev_upper)
```

```
##    psalevel cancervolume age hyperplasia svi capsular gleason
## 94   107.77      45.6042  44           0   1   8.7583       8
```

```r
# Outliers
jackknife = rstudent(prostate.red4)
critical_value = qt(0.05/(2*dim(prostate) [1]), prostate.red4$df.residual - 1)
critical_value
```

```
## [1] -3.598447
```

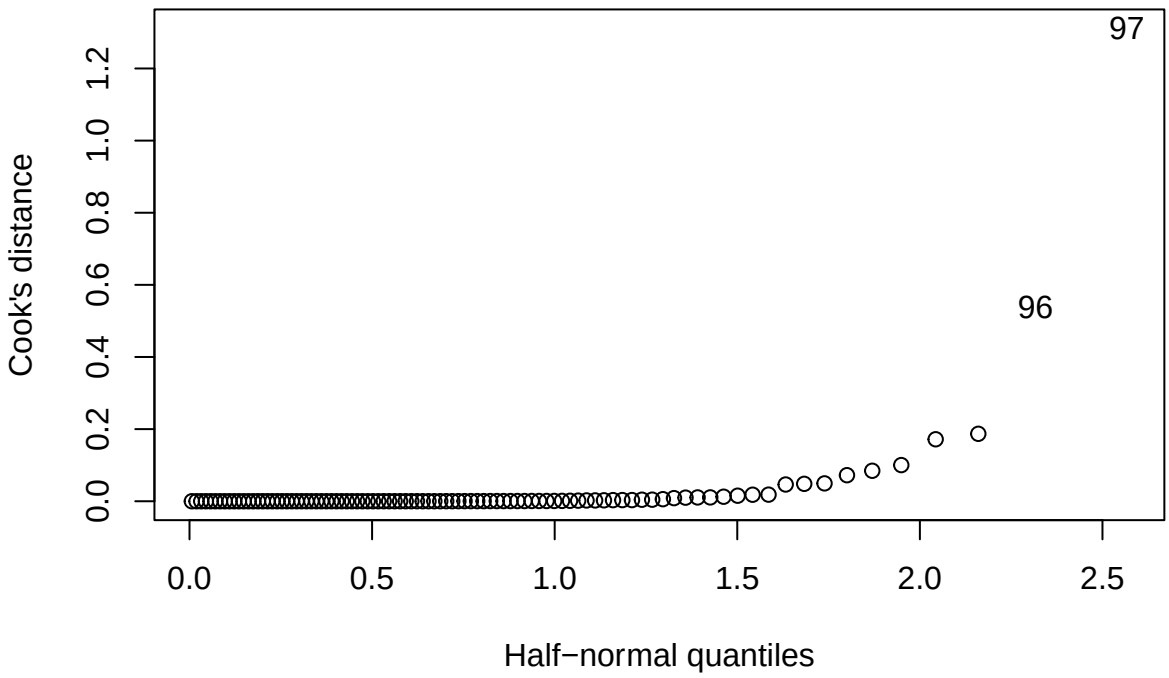
```r
outliers = jackknife[abs(jackknife) > abs(critical_value)]
outliers
```

```
##       96       97
## 6.836494 6.584143
```

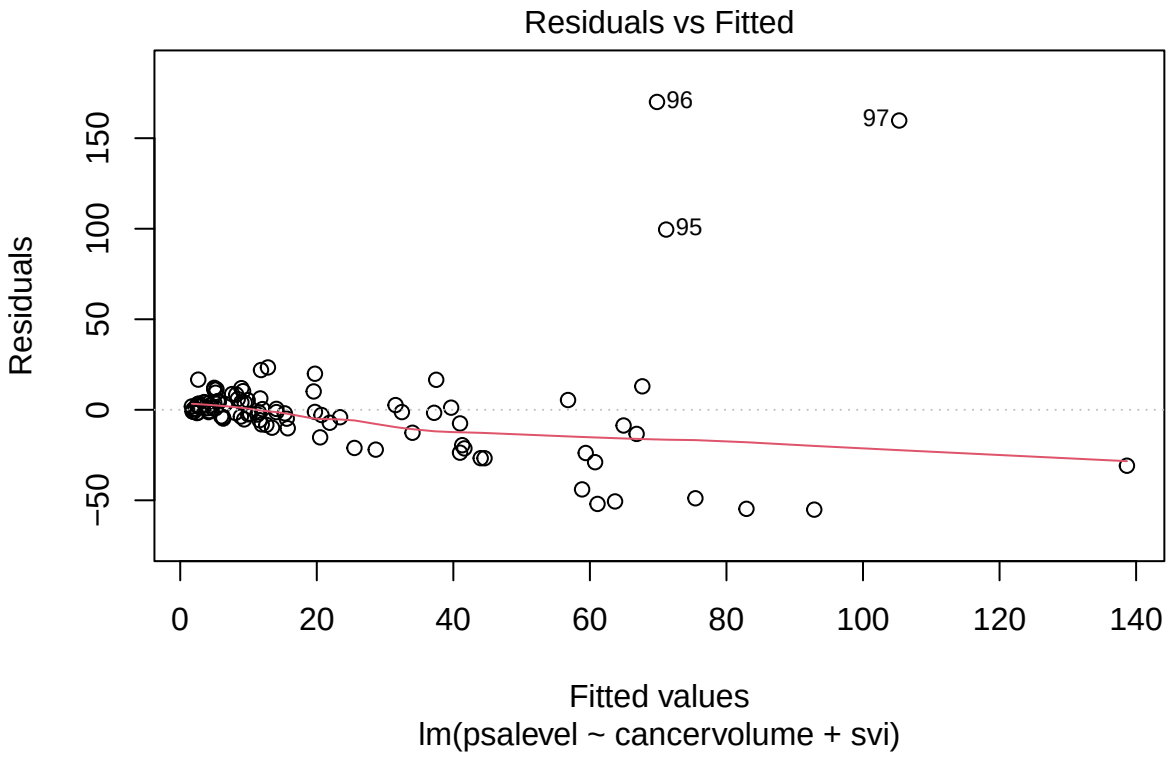
```r
# High Influential Points
cook = cooks.distance(prostate.red4)
cook[cook > 1]
```

```
##       97
## 1.311253
```

```r
halfnorm(cook, labs = row.names(prostate), ylab= "Cook's distance")
```



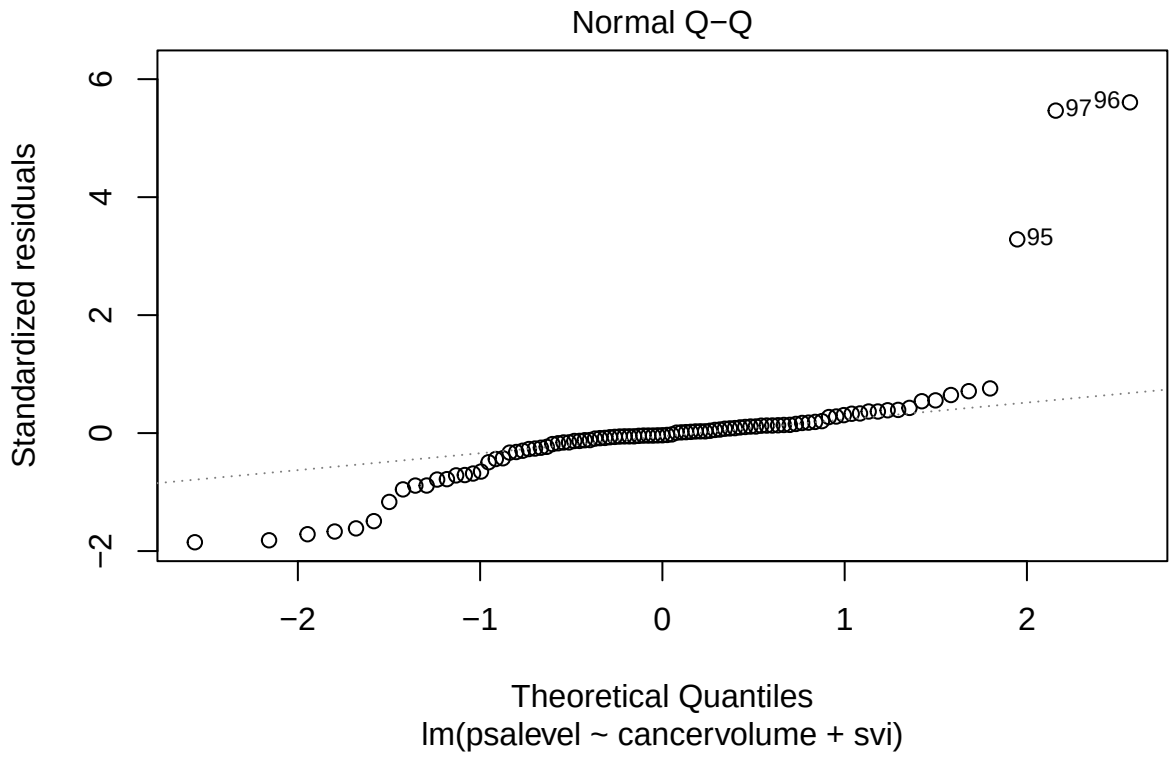

```
# Const Variance
plot(prostate.red4, which = 1)
```




```
bptest(prostate.red4)
```

```
##
##  studentized Breusch-Pagan test
##
## data:  prostate.red4
## BP = 21.674, df = 2, p-value = 1.966e-05
```

```
# Normality Assumption
plot(prostate.red4, which = 2)
```




```
ks.test(resid(prostate.red4), y= pnorm)
```

```
##
##  One-sample Kolmogorov-Smirnov test
##
## data:  resid(prostate.red4)
## D = 0.39551, p-value = 3.542e-14
## alternative hypothesis: two-sided
```

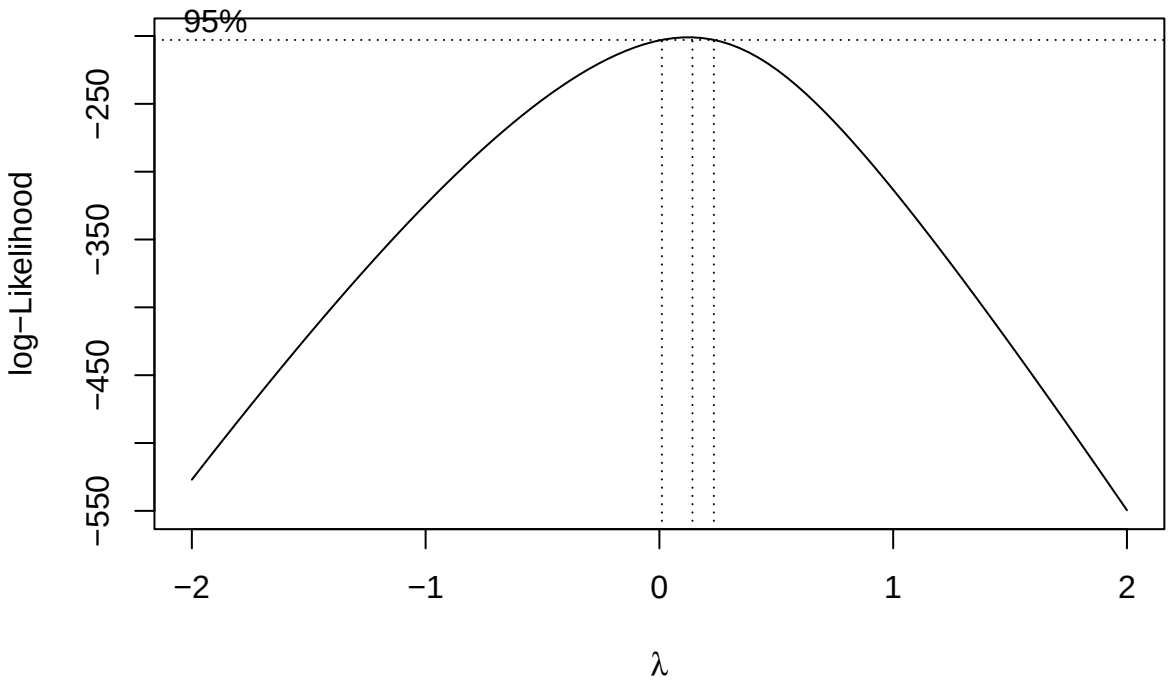
```
# p-values for both tests are less than .05, so we must transform the model.

# Box-Cox Transformation?
library(MASS)
```

```
##
## Attaching package: 'MASS'
```

```
## The following object is masked from 'package:dplyr':
##
##     select
```

```
boxc =  boxcox(prostate.red4, plotit = T)
```




```
boxc
```

```
## $x
##   [1] -2.00000000 -1.95959596 -1.91919192 -1.87878788 -1.83838384 -1.79797980
##   [7] -1.75757576 -1.71717172 -1.67676768 -1.63636364 -1.59595960 -1.55555556
##  [13] -1.51515152 -1.47474747 -1.43434343 -1.39393939 -1.35353535 -1.31313131
##  [19] -1.27272727 -1.23232323 -1.19191919 -1.15151515 -1.11111111 -1.07070707
##  [25] -1.03030303 -0.98989899 -0.94949495 -0.90909091 -0.86868687 -0.82828283
##  [31] -0.78787879 -0.74747475 -0.70707071 -0.66666667 -0.62626263 -0.58585859
##  [37] -0.54545455 -0.50505051 -0.46464646 -0.42424242 -0.38383838 -0.34343434
##  [43] -0.30303030 -0.26262626 -0.22222222 -0.18181818 -0.14141414 -0.10101010
##  [49] -0.06060606 -0.02020202  0.02020202  0.06060606  0.10101010  0.14141414
##  [55]  0.18181818  0.22222222  0.26262626  0.30303030  0.34343434  0.38383838
##  [61]  0.42424242  0.46464646  0.50505051  0.54545455  0.58585859  0.62626263
##  [67]  0.66666667  0.70707071  0.74747475  0.78787879  0.82828283  0.86868687
##  [73]  0.90909091  0.94949495  0.98989899  1.03030303  1.07070707  1.11111111
##  [79]  1.15151515  1.19191919  1.23232323  1.27272727  1.31313131  1.35353535
##  [85]  1.39393939  1.43434343  1.47474747  1.51515152  1.55555556  1.59595960
##  [91]  1.63636364  1.67676768  1.71717172  1.75757576  1.79797980  1.83838384
##  [97]  1.87878788  1.91919192  1.95959596  2.00000000
##
## $y
##   [1] -527.0110 -518.0859 -509.2056 -500.3719 -491.5865 -482.8511 -474.1674
##   [8] -465.5375 -456.9632 -448.4469 -439.9909 -431.5974 -423.2691 -415.0088
##  [15] -406.8192 -398.7035 -390.6647 -382.7065 -374.8323 -367.0459 -359.3513
##  [22] -351.7528 -344.2547 -336.8618 -329.5789 -322.4113 -315.3641 -308.4431
```

```
##  [29] -301.6542 -295.0034 -288.4972 -282.1422 -275.9453 -269.9139 -264.0554
##  [36] -258.3780 -252.8898 -247.5999 -242.5176 -237.6533 -233.0177 -228.6228
##  [43] -224.4819 -220.6089 -217.0203 -213.7334 -210.7676 -208.1445 -205.8868
##  [50] -204.0198 -202.5698 -201.5634 -201.0280 -200.9898 -201.4714 -202.4942
##  [57] -204.0727 -206.2149 -208.9253 -212.1958 -216.0156 -220.3650 -225.2180
##  [64] -230.5462 -236.3161 -242.4932 -249.0431 -255.9312 -263.1244 -270.5929
##  [71] -278.3074 -286.2428 -294.3760 -302.6860 -311.1553 -319.7674 -328.5087
##  [78] -337.3671 -346.3316 -355.3936 -364.5446 -373.7778 -383.0871 -392.4671
##  [85] -401.9132 -411.4213 -420.9876 -430.6091 -440.2828 -450.0061 -459.7767
##  [92] -469.5925 -479.4515 -489.3522 -499.2928 -509.2718 -519.2880 -529.3402
##  [99] -539.4270 -549.5471
```

```
lambda = boxc$x[which.max(boxc$y)]
lambda
```

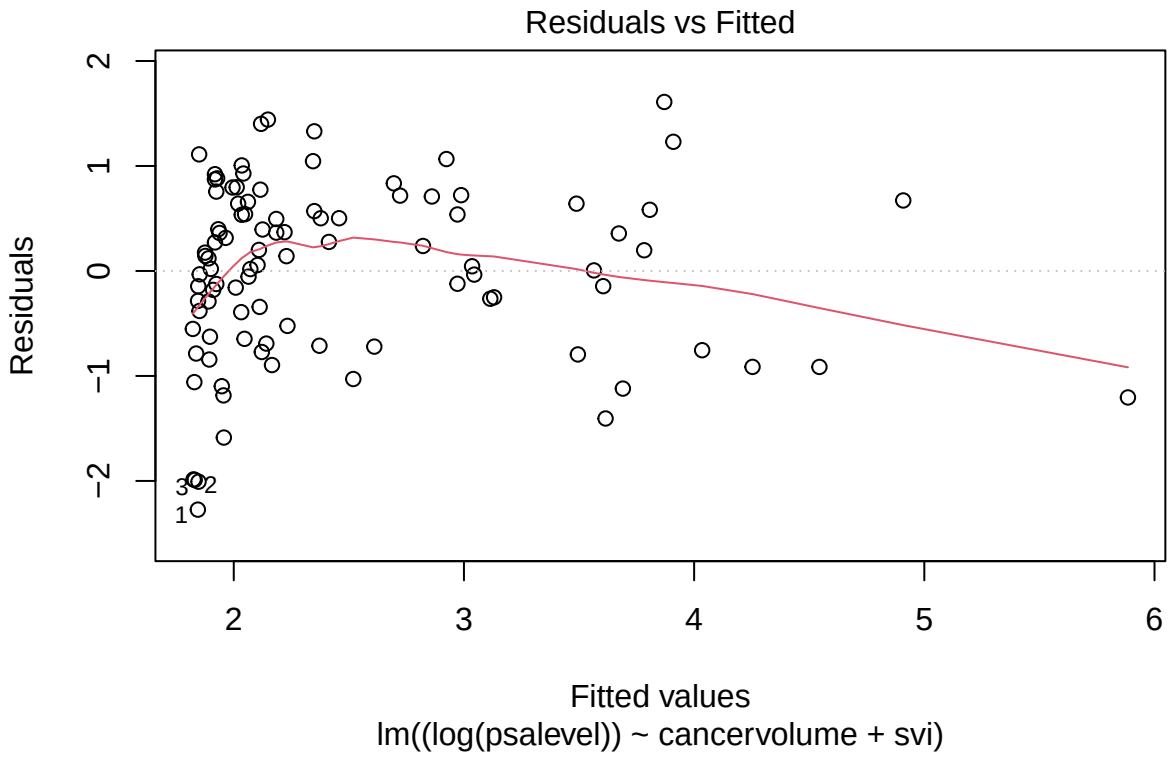
```
## [1] 0.1414141
```

```
prostate.transformed.lm = lm((log(psalevel)) ~ cancervolume + svi, data = prostate)

summary(prostate.transformed.lm)
```

```
##
## Call:
## lm(formula = (log(psalevel)) ~ cancervolume + svi, data = prostate)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -2.2733 -0.6265  0.1197  0.6409  1.6097
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)   1.80346    0.11410  15.806  < 2e-16 ***
## cancervolume  0.07249    0.01335   5.431 4.38e-07 ***
## svi           0.77552    0.25408   3.052  0.00295 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.8383 on 94 degrees of freedom
## Multiple R-squared:  0.483,  Adjusted R-squared:  0.472
## F-statistic: 43.91 on 2 and 94 DF,  p-value: 3.425e-14
```

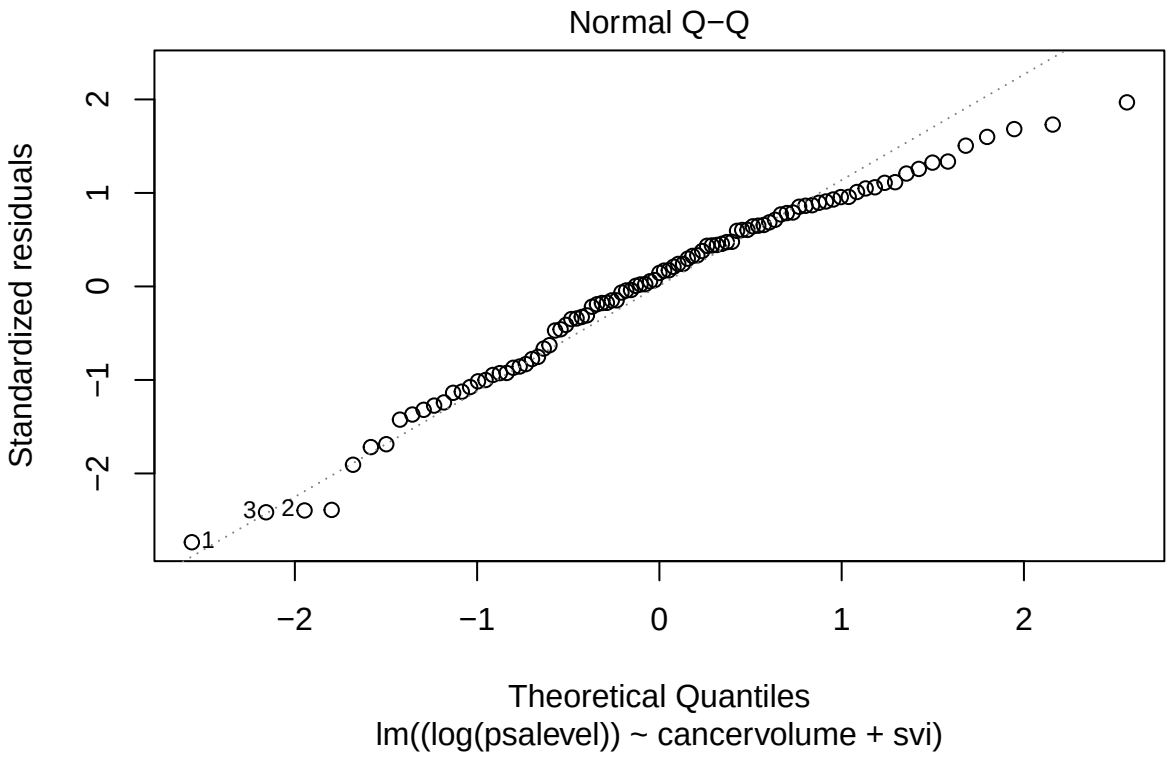
```
# Const Variance
plot(prostate.transformed.lm, which = 1)
```




```
bptest(prostate.transformed.lm)
```

```
##
##  studentized Breusch-Pagan test
##
## data:  prostate.transformed.lm
## BP = 0.02737, df = 2, p-value = 0.9864
```

```
# Normality Assumption
plot(prostate.transformed.lm, which = 2)
```




```
ks.test(resid(prostate.transformed.lm), y= pnorm)
```

```
##
##  One-sample Kolmogorov-Smirnov test
##
## data:  resid(prostate.transformed.lm)
## D = 0.083849, p-value = 0.4773
## alternative hypothesis: two-sided
```

```
#both tests are satisfied, we have our final model.
```