Mini Project 6

CS6313.001

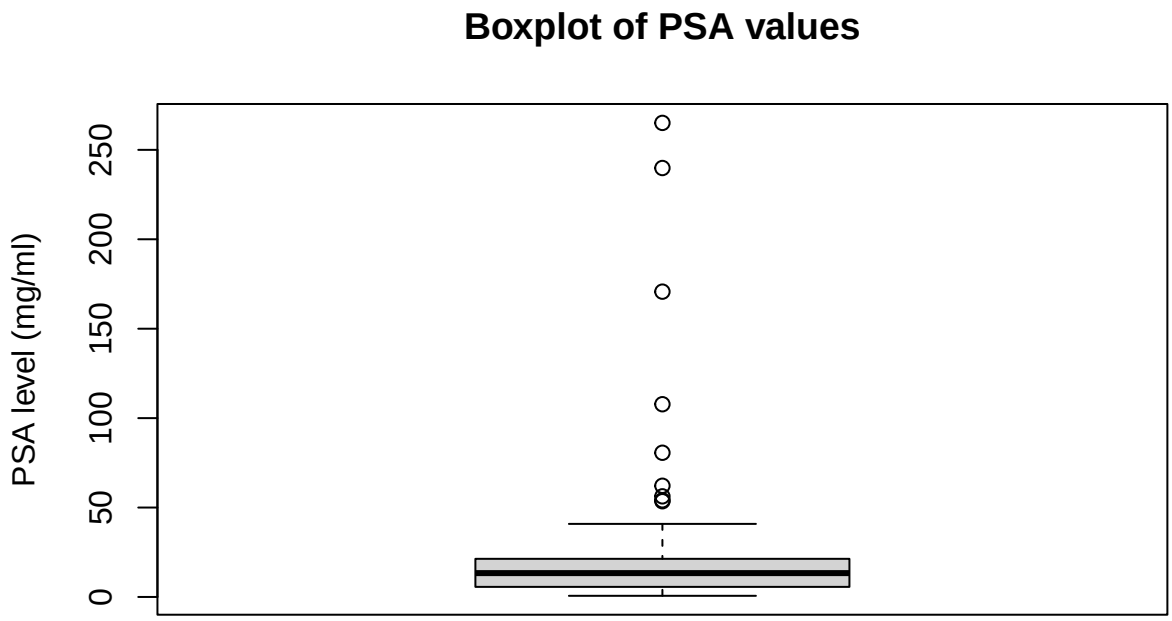
Rutvij Shah (rds190000)

29 November 2021

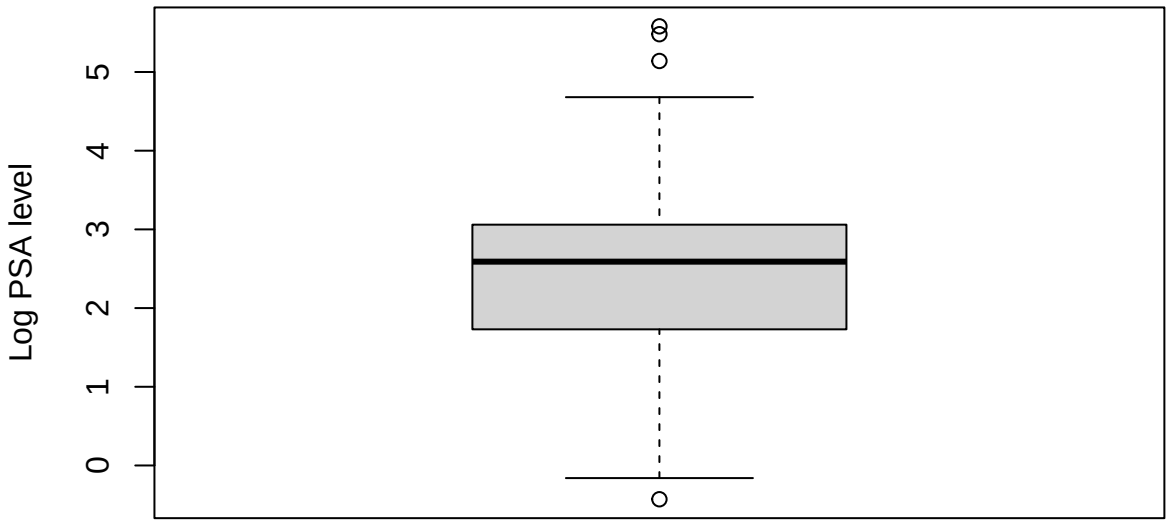
Question 1

Part (a)

```
library(modeest)
cancer_data = read.csv("~/Downloads/prostate_cancer.csv")
attach(cancer_data)
# Investigating distribution of response variable
boxplot(psa, main="Boxplot of PSA values", ylab="PSA level (mg/ml)")
```




```
# Large number of outliers suggest the need for a transformation to prevent
# data loss
boxplot(log(psa), main="Boxplot for log of PSA values", ylab="Log PSA level")
```




```
# This suggests that log transformation does a fair job for our purposes, thus
response_var <- log(psa)

# calculating central tendencies for indicator variables
mean_cancervol = mean(cancervol)
mean_weight = mean(weight)
mean_age = mean(age)
mean_benpros = mean(benpros)
mean_capspen = mean(capspen)
mean_gleason = mean(gleason)
mod_vesinv = mfv(vesinv)
```

```
# Checking the linear relation of each indicator with the response

#1 cancervol
cor(response_var, cancervol)
```

```
## [1] 0.6570739
```

```
#2 weight
cor(response_var, weight)
```

```
## [1] 0.1217208
```

```
#3 age
cor(response_var, age)
```

```
## [1] 0.1699068
```

```
#4 benpros
cor(response_var, benpros)
```

```
## [1] 0.1574016
```

```
#5 capspen
cor(response_var, capspen)
```

```
## [1] 0.5180231
```

```r
#6 gleason
cor(response_var, gleason)
```

```
## [1] 0.5390167
```

```r
#7 vesinv
# since vesinv is a qualitative var, we will use as.factor()
cor(response_var, vesinv)
```

```
## [1] 0.5663641
```

We see that PSA's highest linear correlations are with cancervol, vesinv, gleason and capspen (in that order).

Thus, we will gradually add each to the linear model and determine its significance using ANOVA.

```r
#1
f1 <- lm(response_var ~ cancervol)
summary(f1)
```

```
##
## Call:
## lm(formula = response_var ~ cancervol)
##
## 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
```

F1 is statistically significant since p-val « 0.01. Next we will add vesinv.

```r
#2
f2 <- lm(response_var ~ cancervol + as.factor(vesinv))
summary(f2)
```

```
##
## Call:
## lm(formula = response_var ~ cancervol + as.factor(vesinv))
##
## 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 ***
## cancervol            0.07249    0.01335   5.431 4.38e-07 ***
## as.factor(vesinv)1   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
```

Adding `vesinv` certainly improves the R-Squared value and reduces the residual standard error thus we will build on F2.

```
#3
f3 <- lm(response_var ~ cancervol + as.factor(vesinv) + gleason)
summary(f3)
```

```
##
## Call:
## lm(formula = response_var ~ cancervol + as.factor(vesinv) + gleason)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -2.16928 -0.44558  0.08431  0.60719  1.64082
##
## Coefficients:
##                     Estimate Std. Error t value Pr(>|t|)
## (Intercept)        -0.72120    0.85749  -0.841   0.4025
## cancervol           0.05981    0.01352   4.425 2.62e-05 ***
## as.factor(vesinv)1  0.62117    0.24962   2.488   0.0146 *
## gleason             0.38491    0.12966   2.969   0.0038 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.8055 on 93 degrees of freedom
## Multiple R-squared:  0.5277, Adjusted R-squared:  0.5125
## F-statistic: 34.64 on 3 and 93 DF,  p-value: 4.022e-15
```

Further improvements in R-Squared values couples with a drop of residual standard error evidence that "gleason" does improve our prediction.

```
f4 <- update(f3, . ~ . + capspen)
summary(f4)
```

```
##
## Call:
## lm(formula = response_var ~ cancervol + as.factor(vesinv) + gleason +
##     capspen)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -2.1747 -0.4497  0.1049  0.6215  1.6135
##
## Coefficients:
##                     Estimate Std. Error t value Pr(>|t|)
## (Intercept)        -0.79386    0.86660  -0.916  0.36203
## cancervol           0.06452    0.01522   4.238 5.35e-05 ***
## as.factor(vesinv)1  0.70675    0.28024   2.522  0.01339 *
## gleason             0.39566    0.13100   3.020  0.00327 **
## capspen            -0.02348    0.03455  -0.680  0.49852
```

```
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.8078 on 92 degrees of freedom
## Multiple R-squared:  0.5301, Adjusted R-squared:  0.5097
## F-statistic: 25.95 on 4 and 92 DF,  p-value: 2.075e-14
```

We reject f4 since it has a negative effect on both R-Squared values and on residual standard errors. Also, the p-value » 0.05. We will resume with f3

```
f5 <- update(f3, . ~ . + benpros)
summary(f5)
```

```
##
## Call:
## lm(formula = response_var ~ 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
```

R-Squared has improved from f3, residual error has further reduced and the p-val for benpros is «0.01 suggesting it has a significant influence on the model. Thus we will continue with f5.

```
f6 <- update(f5, . ~ . + age)
summary(f6)
```

```
##
## Call:
## lm(formula = response_var ~ cancervol + as.factor(vesinv) + gleason +
##     benpros + age)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -1.8798 -0.4865  0.1186  0.5484  1.5883
##
## Coefficients:
##                     Estimate Std. Error t value Pr(>|t|)
## (Intercept)        -0.535503   0.985225  -0.544  0.58809
## cancervol           0.064696   0.012944   4.998 2.79e-06 ***
## as.factor(vesinv)1  0.689452   0.238993   2.885  0.00489 **
```

```
## gleason             0.338677   0.126217   2.683  0.00866 **
## benpros             0.093520   0.028198   3.317  0.00131 **
## age                -0.002408   0.011650  -0.207  0.83674
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.7646 on 91 degrees of freedom
## Multiple R-squared:  0.5836, Adjusted R-squared:  0.5607
## F-statistic: 25.51 on 5 and 91 DF,  p-value: 5.349e-16
```

```
f7 <- update(f5, . ~ . + weight)
summary(f7)
```

```
##
## Call:
## lm(formula = response_var ~ cancervol + as.factor(vesinv) + gleason +
##     benpros + weight)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -1.8693 -0.4930  0.0882  0.4993  1.5000
##
## Coefficients:
##                     Estimate Std. Error t value Pr(>|t|)
## (Intercept)        -0.736294   0.819897  -0.898  0.37154
## cancervol           0.064321   0.012897   4.987 2.92e-06 ***
## as.factor(vesinv)1  0.679486   0.237041   2.867  0.00516 **
## gleason             0.340550   0.123925   2.748  0.00723 **
## benpros             0.084442   0.027687   3.050  0.00300 **
## weight              0.001362   0.001805   0.754  0.45271
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.7624 on 91 degrees of freedom
## Multiple R-squared:  0.586,  Adjusted R-squared:  0.5632
## F-statistic: 25.76 on 5 and 91 DF,  p-value: 4.14e-16
```

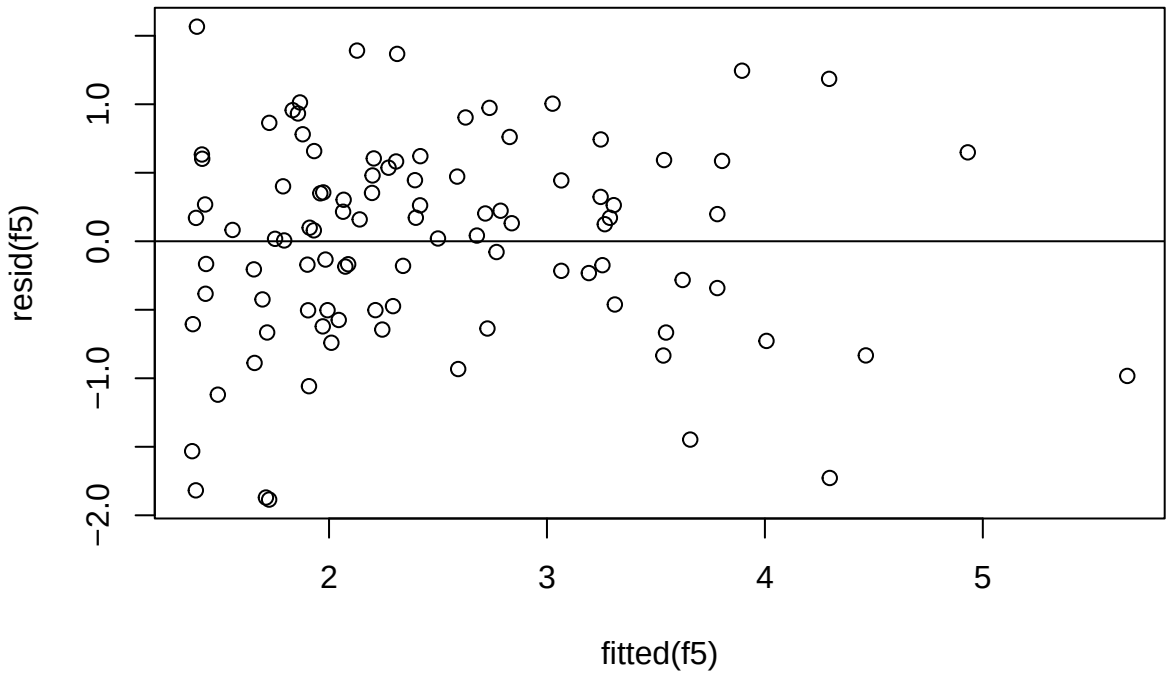
The p-values for both age & weight are »0.05 and hence are rejected.

Thus f5 seems to be our “reasonably good” linear model for predicting PSA level based on the data available.

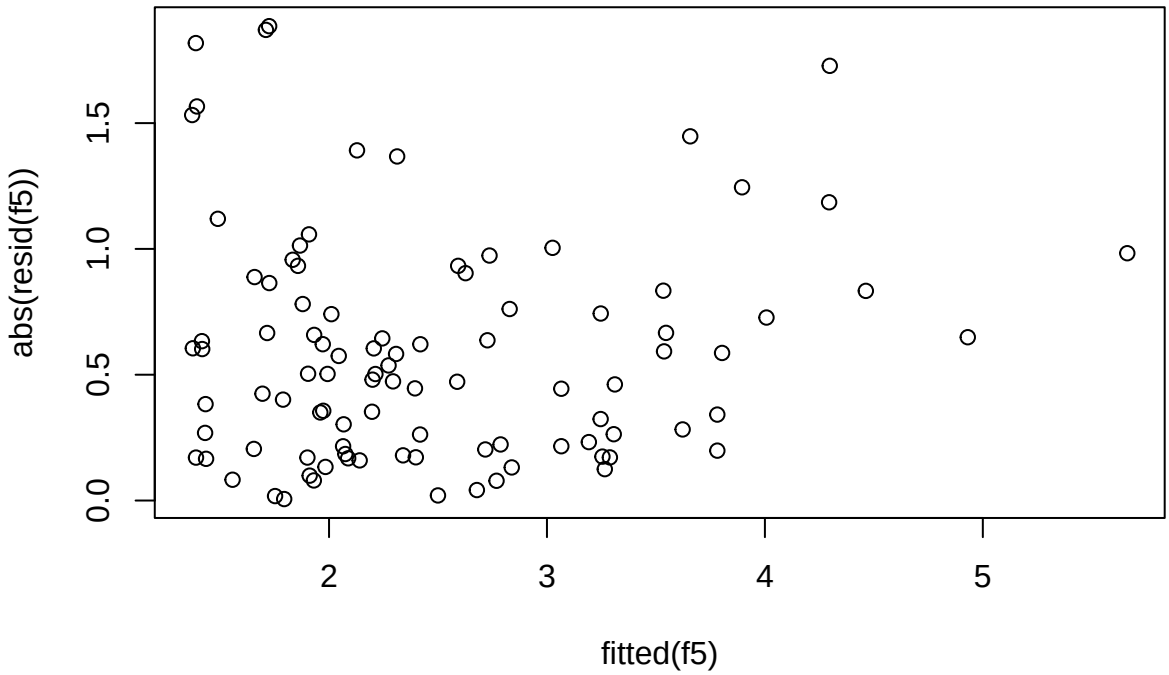
Lastly, we will check the assumptions of our model.

(1) Are the residuals randomly distributed?

```
plot(fitted(f5), resid(f5))
abline(h=0)
```



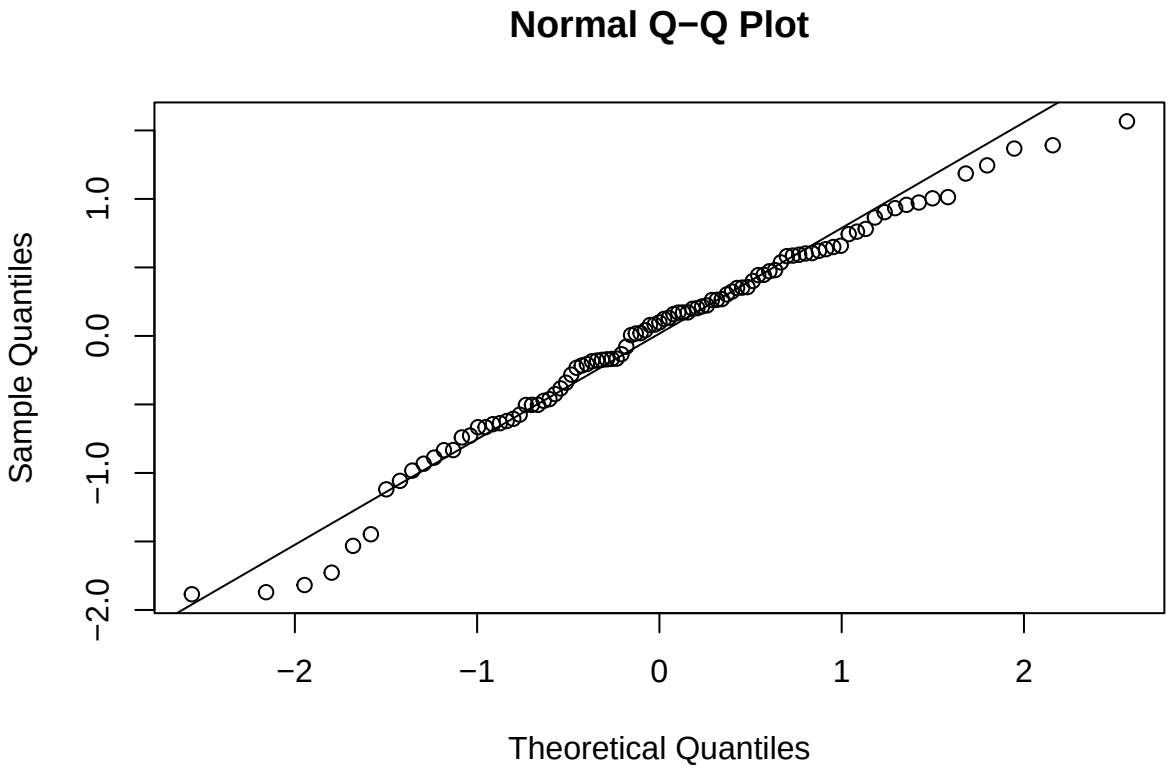

```
plot(fitted(f5), abs(resid(f5)))
```



Yes, the residues are fairly random in their distribution and hence a linear regression model's use can be justified.

(2) Are the residues approximately normally distributed? (Yes!)

```
qqnorm(resid(f5))
qqline(resid(f5))
```



Part (b)

Prediction of PSA value for means of indicator (or mode if qualitative).

```
c1 <- c(mean_cancervol)
c2 <- c(mod_vesinv)
c3 <- c(mean_gleason)
c4 <- c(mean_benpros)
a_row_df = data.frame(c1, c2, c3, c4)
names(a_row_df) <- c("cancervol", "vesinv", "gleason", "benpros")
log_psa = predict.lm(f5, a_row_df)
pred_psa = exp(log_psa)
cat(paste("The predicted value of PSA is = ", round(pred_psa, 4)))
```

```
## The predicted value of PSA is =  10.2835
```

Part (c)

Verification with model selection using backward elimination algorithm

```
full = lm(response_var ~ gleason
       + capspen
       + benpros
       + age
       + weight
       + cancervol
       + as.factor(vesinv))

null = lm(response_var ~ 1)
```

```
f5.auto.back_elm <- step( full, scope=list( lower=null, upper=full ), direction="backward" )
```

```
## Start:  AIC=-43.59
## response_var ~ gleason + capspen + benpros + age + weight + cancervol +
##     as.factor(vesinv)
##
##                      Df Sum of Sq    RSS     AIC
## - age                 1    0.0336 52.510 -45.529
## - weight              1    0.3383 52.815 -44.968
## - capspen             1    0.3841 52.861 -44.884
## <none>                            52.477 -43.591
## - gleason             1    4.6180 57.095 -37.410
## - as.factor(vesinv)   1    5.0155 57.492 -36.737
## - benpros             1    5.1469 57.624 -36.516
## - cancervol           1   13.2994 65.776 -23.680
##
## Step:  AIC=-45.53
## response_var ~ gleason + capspen + benpros + weight + cancervol +
##     as.factor(vesinv)
##
##                      Df Sum of Sq    RSS     AIC
## - weight              1    0.3264 52.837 -46.928
## - capspen             1    0.3881 52.898 -46.815
## <none>                            52.510 -45.529
## - gleason             1    4.6365 57.147 -39.322
## - as.factor(vesinv)   1    4.9820 57.492 -38.737
## - benpros             1    5.4873 57.998 -37.888
## - cancervol           1   13.4654 65.976 -25.386
##
## Step:  AIC=-46.93
## response_var ~ gleason + capspen + benpros + cancervol + as.factor(vesinv)
##
##                      Df Sum of Sq    RSS     AIC
## - capspen             1    0.3923 53.229 -48.211
## <none>                            52.837 -46.928
## - gleason             1    4.4852 57.322 -41.025
## - as.factor(vesinv)   1    5.0526 57.889 -40.069
## - benpros             1    7.2024 60.039 -36.532
## - cancervol           1   13.7311 66.568 -26.520
##
## Step:  AIC=-48.21
## response_var ~ gleason + benpros + cancervol + as.factor(vesinv)
##
##                      Df Sum of Sq    RSS     AIC
## <none>                            53.229 -48.211
## - gleason             1    4.2389 57.468 -42.778
## - as.factor(vesinv)   1    4.8466 58.075 -41.758
## - benpros             1    7.1115 60.340 -38.047
## - cancervol           1   14.7580 67.987 -26.473
```

```
auto.log_psa = predict.lm(f5.auto.back_elm, a_row_df)
auto.pred_psa = exp(auto.log_psa)

cat(paste("\n\nThe auto predicted value of PSA is = ", round(auto.pred_psa, 4)))
```

```
##
##
## The auto predicted value of PSA is =  10.2835
```

We've verified that the most reasonable models produced are the same with both the manual and automated methods. And the predicted value generated by both attests to the fact along with the formula for linear regression.

```
library(equatiomatic)
extract_eq(f5, use_coefs = T)
```

Thus, the final model is encapsulated by

$$\hat{\log(\text{psa})} = -0.65 + 0.06(\text{cancervol}) + 0.68(\text{as. factor}(\text{vesinv})_1) + 0.33(\text{gleason}) + 0.09(\text{benpros})$$