Mini Project : 6

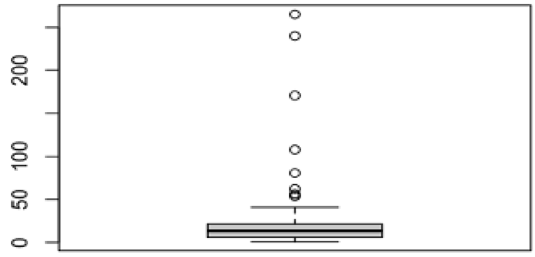
Members :
1. Puru Jaiswal (NetID : PXJ200018)
2. Sara Tabassi (NetID : SXT200083)

Both the team members worked together to finish the project. Collaborated to learn R and then write the code. Both of us worked on each of the problems independently and then reviewed each other's answers to determine which solution was optimal.
Both members worked efficiently to complete the project requirements.

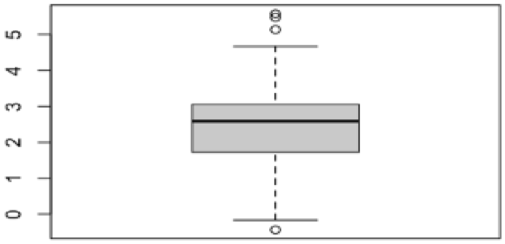
Section 1:
Consider the prostate cancer dataset available on eLearning as prostate cancer.csv. It consists of data on 97 men with advanced prostate cancer. A description of the variables is given in Figure 1. We would like to understand how PSA level is related to the other predictors in the dataset. Note that vesinv is a qualitative variable. You can treat gleason as a quantitative variable. Build a "reasonably good" linear model for these data by taking PSA level as the response variable. Carefully justify all the choices you make in building the model. Be sure to verify the model assumptions. In case a transformation of response is necessary, try the natural log transformation. Use the final model to predict the PSA level for a patient whose quantitative predictors are at the sample means of the variables and qualitative predictors are at the most frequent category.

Section 1

We first looked at a boxplot of psa (left) and log(psa) (right). Notice that log(psa) has far fewer outliers and generally has a better distribution to fit the data, Therefore, we choose log(psa) as our response variable.



psa



log(psa)

“cancervol” has the highest significance with **p = 2.69e-13**, therefore we chose it first.
We then chose the next best predictor, factor(vesinv), and added that to our model. We did an anova test of **y ~ cancervol with y ~ cancervol + factor(vesinv)**, and saw that there was a significant difference with **p =0.002953**, so we keep factor(vesinv) in our model. We then looked at capspen and compared that to y ~ cancervol + factor(vesinv), but noticed there was not a significant difference, with p = 0.7616.

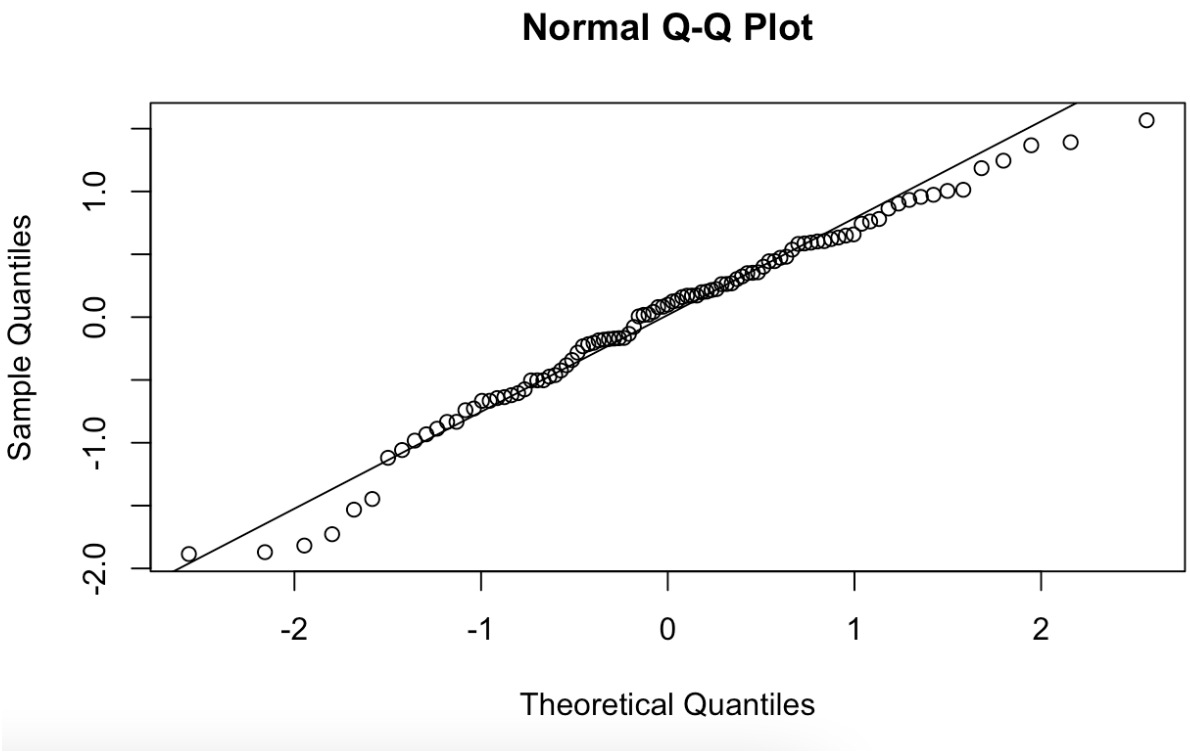
Therefore, we do not add capspen to our model. We then added gleason and did an Anova test, and noticed that p=0.003804, so we kept it in our model. We continued this process and noticed that the only other predictor that added significance to our model was benpros.
So, we ended up with a model of log(psa) ~ cancervol + factor(vesinv) + gleason + benpros.

Summary of the model:

```
           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    ***

 factor(vesinv)1 0.68421             0.23640        2.894 0.004746 **


 gleason            0.33376      0.12331        2.7070.008100    **

 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
```

Based on the summary all the predictors are significant.
We then further verified our model by performing three tests on the dataset: a forward selection, backwards elimination, and both direction model building.
All three of these gave us the same model: **log(psa) ~ cancervol + factor(vesinv) + gleason + benpros**, which is what we got above.

Finally, we looked at a qqplot to see how well our model fit our data:



The **qqplot** confirms that our model matches the data we have.

Prediction based on the model:

cancervol (sample mean) = 6.998682
benpros (sample mean) = 2.534725
gleason (sample mean) = 6.876289
vesinv (sample mode) = 0

log(psa) = -0.65013 + 0.06488 * 6.998682 + 2.534725 * 0.09136 + 6.876289 * 0.33376 + 0 * 0.68421 = 2.3305

psa = e^2.3305 = 10.283

Section 2

```
#read from file
pr_cancer_data=read.csv(file="./prostate_cancer.csv",sep=",", header=T)
str(pr_cancer_data)

#fetching the column data
subject = pr_cancer_data$subject
psa = pr_cancer_data$psa
weight = pr_cancer_data$weight
cancervol = pr_cancer_data$cancervol
age = pr_cancer_data$age
benpros = pr_cancer_data$benpros
vesinv = pr_cancer_data$vesinv
capspen = pr_cancer_data$capspen
gleason = pr_cancer_data$gleason

#Distributions of psa and log(psa)
boxplot(psa)
boxplot(log(psa))

#Based on the boxplots log(psa) has a better distribution
y=log(psa)

plot(weight, y)
fit1 <- lm(y ~ weight, data = pr_cancer_data)
abline(fit1)
summary(fit1)

plot(cancervol, y)
fit2 <- lm(y ~ cancervol, data = pr_cancer_data)
abline(fit2)
summary(fit2)

plot(age, y)
fit3 <- lm(y ~ age, data = pr_cancer_data)
abline(fit3)
summary(fit3)

plot(benpros, y)
fit4 <- lm(y ~ benpros, data = pr_cancer_data)
```

```
abline(fit4)
summary(fit4)

plot(factor(vesinv), y)
fit5 <- lm(y ~ factor(vesinv), data = pr_cancer_data)

summary(fit5)

plot(capspen, y)
fit6 <- lm(y ~ capspen, data = pr_cancer_data)
abline(fit6)
summary(fit6)

plot(gleason, y)
fit7 <- lm(y ~ gleason, data = pr_cancer_data)
abline(fit7)
summary(fit7)

#start with cancervol and compare to cancervol + vesinv (two best significance levels)
fit8 = lm(y ~ cancervol+factor(vesinv), data = pr_cancer_data)
anova(fit2, fit8)

#Result of anova(fit2, fit8): p = 0.002953, so vesinv is significant
fit9 = lm(y ~ capspen+factor(vesinv)+cancervol, data = pr_cancer_data)
anova(fit8, fit9)

#Result of anova(fit8, fit9): P = 0.7616, so capspen is not significant
fit10 = lm(y ~cancervol + factor(vesinv) + gleason, data = pr_cancer_data)
anova(fit8, fit10)

#Result of anova(fit8, fit10): p = 0.003804, so gleason is significant
fit11 = lm(y ~ cancervol + factor(vesinv) + gleason + age + benpros + weight , data =
pr_cancer_data)
anova(fit10,fit11)

#Result of anova(fit10,fit11): p = 0.007466, so at least one of age + benpros + weight is
significant
fit12 = lm(y ~ cancervol + factor(vesinv) + gleason + age, data = pr_cancer_data)
anova(fit10,fit12)

#Result of anova(fit10,fit12): P = 0.2995, so age is not significant
fit13 = lm(y ~ cancervol + factor(vesinv) + gleason + benpros, data = pr_cancer_data)
anova(fit10,fit13)
```

```
#Result of anova(fit10,fit13): P = 0.0007054, so benpros is significant

fit14 = lm(y ~ cancervol + factor(vesinv) + gleason + benpros + weight, data=pr_cancer_data)
anova(fit13,fit14)

#Result of anova(fit13,fit14): P = 0.4527, so weight is not significant

#Model: y ~ cancervol + gleason + benpros + vesinv

#Compare our model to forward selection method
fit10.forward <- step(lm(y ~ 1, data = pr_cancer_data),
                      scope = list(upper = ~cancervol + age + weight + benpros + factor(vesinv) +
capspen + gleason),
                      direction = "forward")

#Result: y ~ cancervol + gleason + benpros + vesinv
#compare our model to backward selection method
fit11.backward <- step(lm(y ~ cancervol + age + weight + benpros + factor(vesinv) + capspen +
gleason, data = pr_cancer_data),
                       scope = list(lower = ~1), direction = "backward")
#result: y ~ cancervol + benpros + vesinv + gleason

#compare our model to both selection method
fit12.both <- step(lm(y ~ 1, data = pr_cancer_data),
                   scope = list(lower = ~1, upper = ~cancervol + age + weight + benpros +
factor(vesinv) + capspen + gleason),
                   direction = "both")

#result: cancervol + gleason + benpros + vesinv
#  residual plot
plot(fitted(fit13), resid(fit13))
abline(h = 0)
#  plot of absolute residuals
plot(fitted(fit13), abs(resid(fit13)))
#  normal QQ plot
qqnorm(resid(fit13))
qqline(resid(fit13))
#qq plot looks accurate
#All vars in our model are signifcant

summary(fit13)
get.mode <- function(k) {
  u = unique(k)
  u[which.max(tabulate(match(k, u)))]
}

#Get the means of cancervolm benpros, gleason, and mode of vesinv.

cancervol.mean = mean(pr_cancer_data$cancervol)
benpros.mean = mean(pr_cancer_data$benpros)
temp.vesinv.list = as.numeric(unlist(factor(pr_cancer_data$vesinv)))

vesinv.mode = get.mode(temp.vesinv.list)

gleason.mean = mean(pr_cancer_data$gleason)

#Model : y ~ cancervol + factor(vesinv) + gleason + benpros

#Find prediction using sample means
prediction = (fit13$coefficients[1] + cancervol.mean * fit13$coefficients[2]
        +0*fit13$coefficients[3]+gleason.mean*fit13$coefficients[4]
        +benpros.mean*fit13$coefficients[5])
print(prediction)
print(exp(prediction))
```