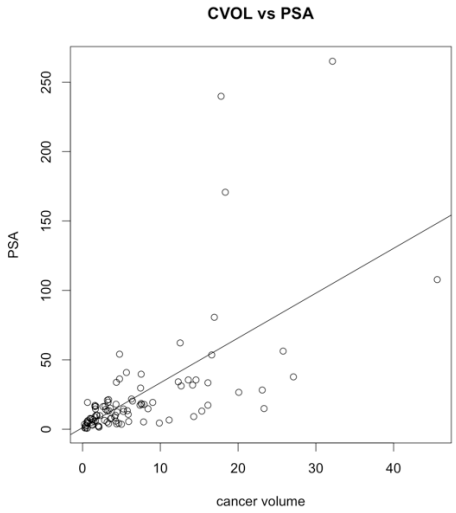
Statistical Methods for Data Science
Mini Project 5

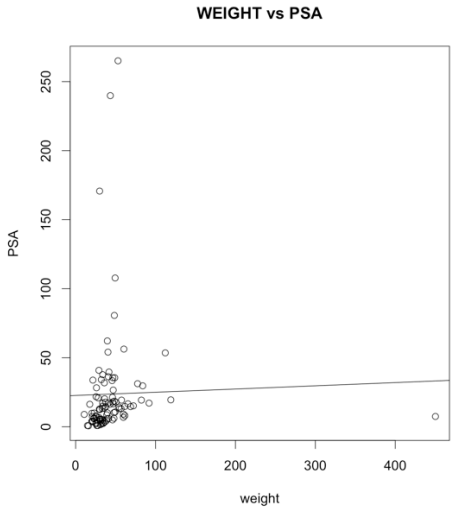
Charu Arora
Cxa150730

1.
PSA -> Response variable

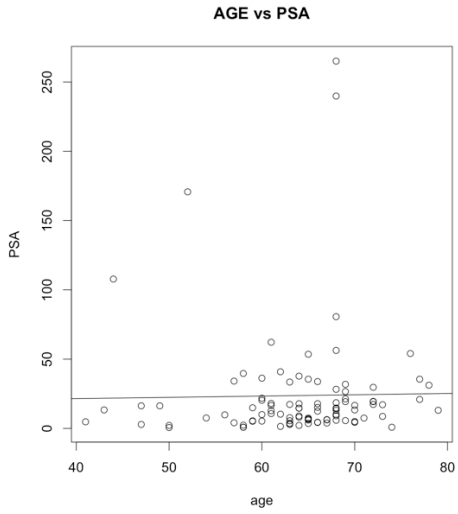
Scatterplots with other quantitative variables



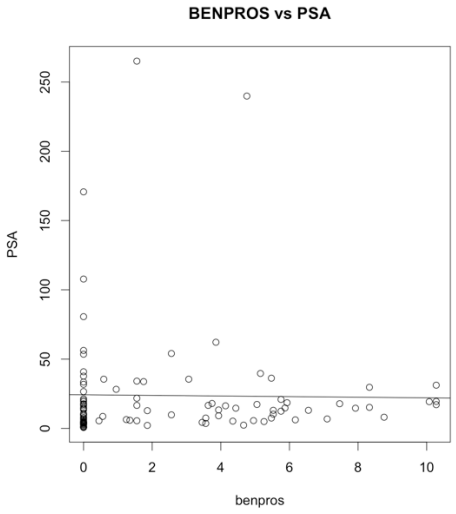
> cor(sysBP$cancervol,sysBP$psa)
[1] 0.6241506



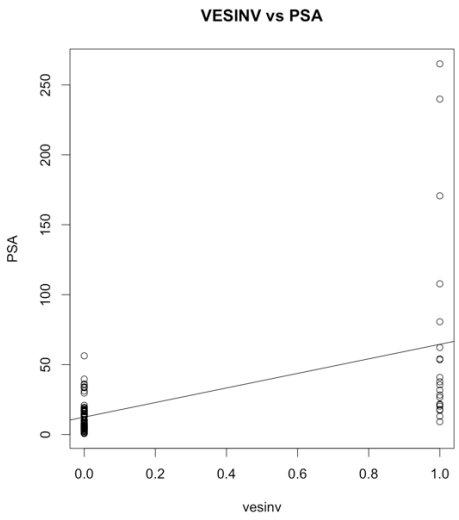
> cor(sysBP$weight,sysBP$psa)
[1] 0.02621343



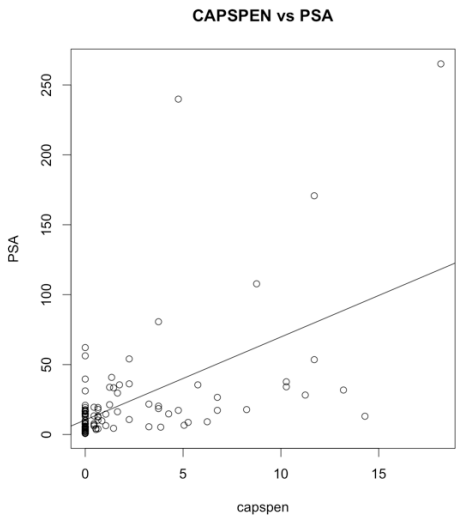
> cor(sysBP$age,sysBP$psa)
[1] 0.01719938



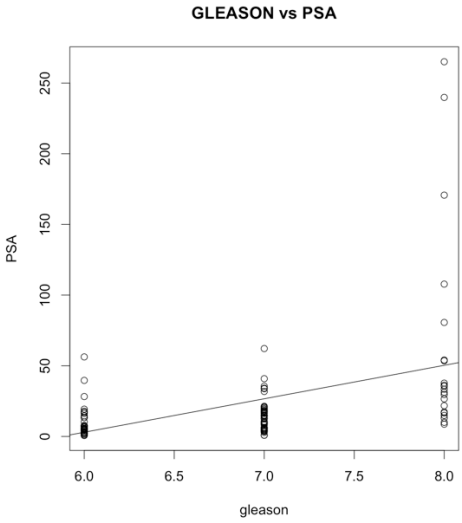
> cor(sysBP$benpros,sysBP$psa)
[1] -0.01648649



> cor(sysBP$vesinv,sysBP$psa)
[1] 0.5286188

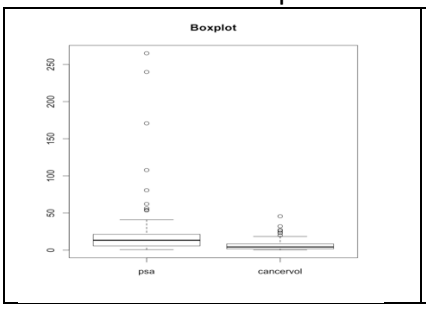


> cor(sysBP$capspen,sysBP$psa)
[1] 0.5507925



> cor(sysBP$gleason,sysBP$psa)
[1] 0.4295798

The variable that can be used to predict the PDA level effectively is cancervol as it has a strong positive correlation with PSA. Cancervol increases linearly with PSA. It has also the highest correlation with PSA as compared to the other quantitative variables.
We see in the scatterplot that there are a couple of outliers.



From the boxplot it is visible that individually also they both have some outliers.

R – code of scatterplots and boxplot

```
sysBP <- read.csv(file="/Users/charuarora/Downloads/prostate_cancer.csv",header=TRUE,sep=",");

plot(sysBP$cancervol,sysBP$psa,xlab = "cancer volume", ylab = "PSA", main = "CVOL vs PSA")
abline(lm(sysBP$psa~sysBP$cancervol))
cor(sysBP$cancervol,sysBP$psa)

plot(sysBP$weight,sysBP$psa,xlab = "weight", ylab = "PSA", main = "WEIGHT vs PSA")
abline(lm(sysBP$psa~sysBP$weight))
cor(sysBP$weight,sysBP$psa)

plot(sysBP$age,sysBP$psa,xlab = "age", ylab = "PSA", main = "AGE vs PSA")
abline(lm(sysBP$psa~sysBP$age))
```

```
plot(sysBP$benpros,sysBP$psa,xlab = "benpros", ylab = "PSA", main = "BENPROS vs PSA")
abline(lm(sysBP$psa~sysBP$benpros))
cor(sysBP$benpros,sysBP$psa)

plot(sysBP$vesinv,sysBP$psa,xlab = "vesinv", ylab = "PSA", main = "VESINV vs PSA")
abline(lm(sysBP$psa~sysBP$vesinv))
cor(sysBP$vesinv,sysBP$psa)

plot(sysBP$capspen,sysBP$psa,xlab = "capspen", ylab = "PSA", main = "CAPSPEN vs PSA")
abline(lm(sysBP$psa~sysBP$capspen))
cor(sysBP$capspen,sysBP$psa)

plot(sysBP$gleason,sysBP$psa,xlab = "gleason", ylab = "PSA", main = "GLEASON vs PSA")
abline(lm(sysBP$psa~sysBP$gleason))
cor(sysBP$gleason,sysBP$psa)

boxplot(sysBP$psa,sysBP$cancervol,main = "Boxplot",names=c("psa","cancervol"))
```

2. Fitting a simple linear regression model.

```
> x<-sysBP$cancervol
> y<-sysBP$psa
> cancer.reg<-lm(y~x)
```

The function lm fits a linear model to data and we specify the model using the formula where PSA (response variable) is on the left side separated by ~ from the other variable.
To create a summary of the fitted model:

```
> summary(cancer.reg)
```

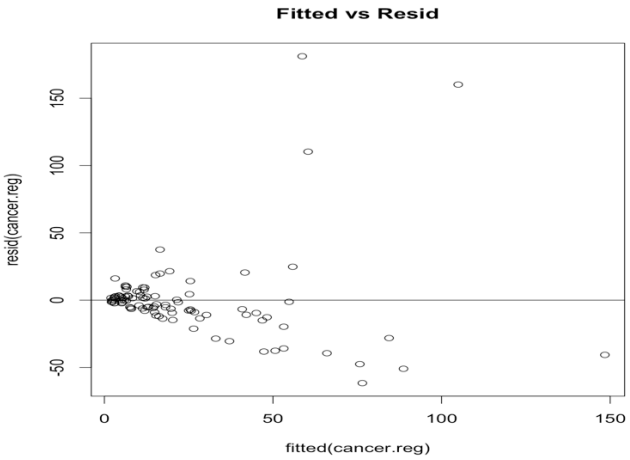
```
Call:
lm(formula = y ~ x)

Residuals:
    Min      1Q  Median      3Q     Max
-61.619  -9.023  -1.586   3.151 181.183

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)   1.1249     4.3596   0.258    0.797
x             3.2299     0.4148   7.786 8.47e-12 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 32.03 on 95 degrees of freedom
Multiple R-squared:  0.3896,  Adjusted R-squared:  0.3831
F-statistic: 60.63 on 1 and 95 DF,  p-value: 8.468e-12
```

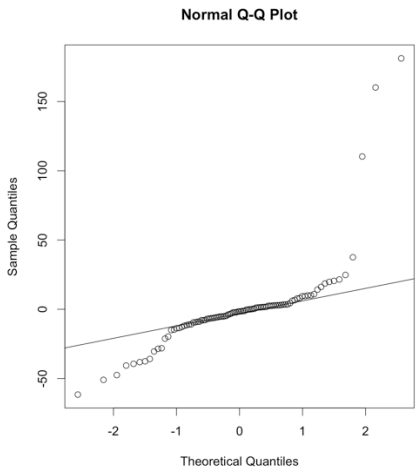
The estimates for the model intercept is 1.1249 and the coefficient measuring the slope of the relationship with cancervol is 3.2299.

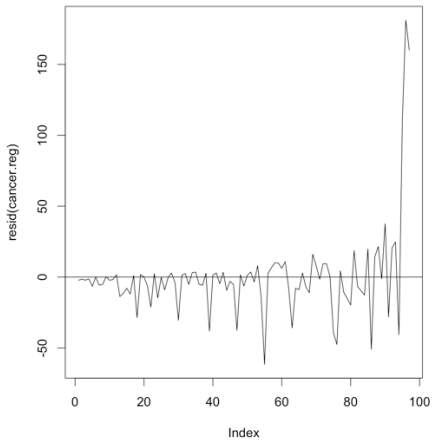
Key Assumptions:

- Errors are constant
- Errors are independent
- Errors follow a normal distribution

A plot of the residuals against fitted values is used to determine whether there are any systematic patterns, such as over estimation for most of the large values or increasing spread as the model fitted values increase.

<table>
<tr>
<td>Testing key assumptions:
From the following plots, it is clear that the errors are not constant which violates our key assumptions. We see a pattern in the residuals plots.
From the Q-Q plot we see that the normality is also not a reasonable assumptions.
Therefore, we take the log transformation as all our key assumptions are not satisfied.</td>
<td> 

> plot(fitted(cancer.reg), resid(cancer.reg),main = 'Fitted vs Resid')
> abline(h=0)</td>
</tr>
<tr>
<td> 

> qqnorm(resid(cancer.reg))
> qqline(resid(cancer.reg))</td>
<td> 

> plot(resid(cancer.reg), type="l")
> abline(h=0)</td>
</tr>
</table>

The regression model with log transformation:

R- code for modelling the linear model for log(PSA) and log(cancervol)

> logpsa = log(sysBP$psa)

> newcancer.reg = lm(logpsa~logcancervol)
> summary(newcancer.reg)

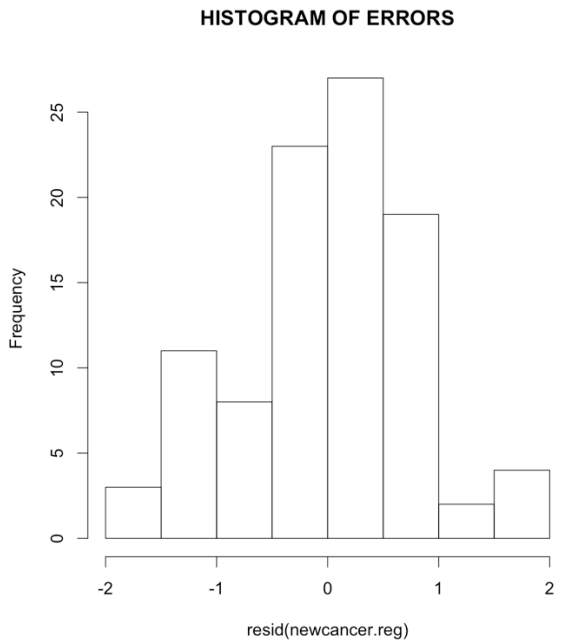
```
Call:
lm(formula = logpsa ~ logcancervol)

Residuals:
    Min      1Q  Median      3Q     Max
-1.6778 -0.4187  0.1012  0.5035  1.9022

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept)   1.50923    0.12198   12.37   <2e-16 ***
logcancervol  0.71827    0.06822   10.53   <2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.7879 on 95 degrees of freedom
Multiple R-squared:  0.5385,  Adjusted R-squared:  0.5336
F-statistic: 110.8 on 1 and 95 DF, p-value: < 2.2e-16
```

The estimated intercept, b_0=1.50923 and slope is b_1=0.71827.
From the, R-squared value, logcancervol explains 53.36% of the total variability
Since the p-value for log cancervol is between 0 and 0.001, logcancervol is a significant predictor for the PSA value.

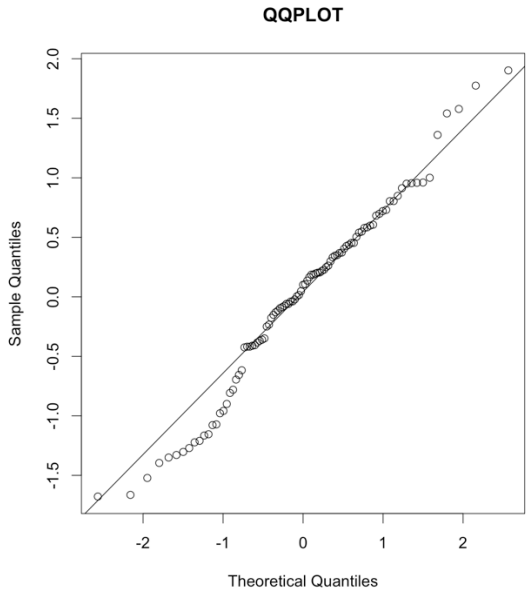
> anova(newcancer.reg)

```
Analysis of Variance Table

Response: logpsa
             Df Sum Sq Mean Sq F value    Pr(>F)
logcancervol  1 68.801  68.801  110.84 < 2.2e-16 ***
Residuals    95 58.968   0.621
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

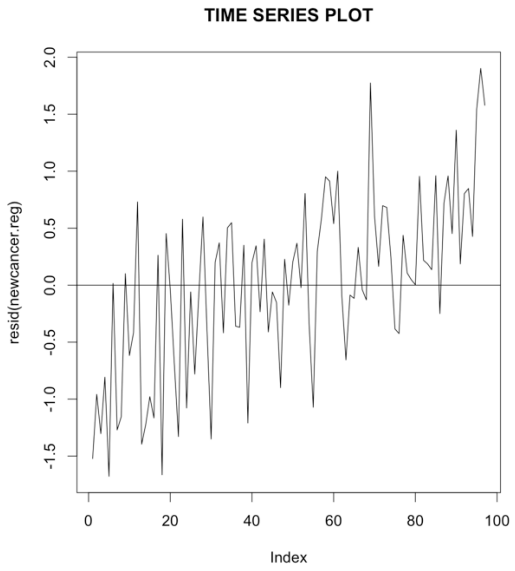
Our key assumptions are satisfied as there are no trends in the above given plots.



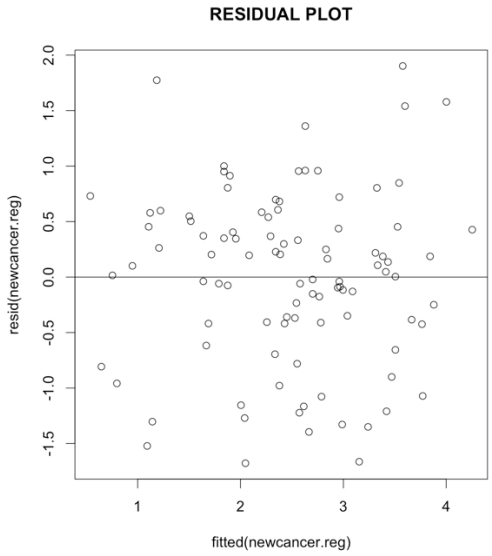
> hist(resid(newcancer.reg), main="HISTOGRAM OF ERRORS")



> qqnorm(resid(newcancer.reg), main="QQPLOT")
> qqline(resid(newcancer.reg))



> plot(resid(newcancer.reg), type="l", main ="TIME SERIES PLOT")
> abline(h=0)



> plot(fitted(newcancer.reg),resid(newcancer.reg), main="RESIDUAL PLOT")
> abline(h=0)

3.PSA value for a patient whose predictor variables are at the sample medians of the variable.

Using the old method,

> x.new <- data.frame(x=median(x))
> predict(cancer.reg,x.new)

1
14.89438

```
> new_predict <- exp(1)^predict(newcancer.reg,data.frame(logcancervol = median(logcancervol)))
> new_predict
       1
12.81632
```