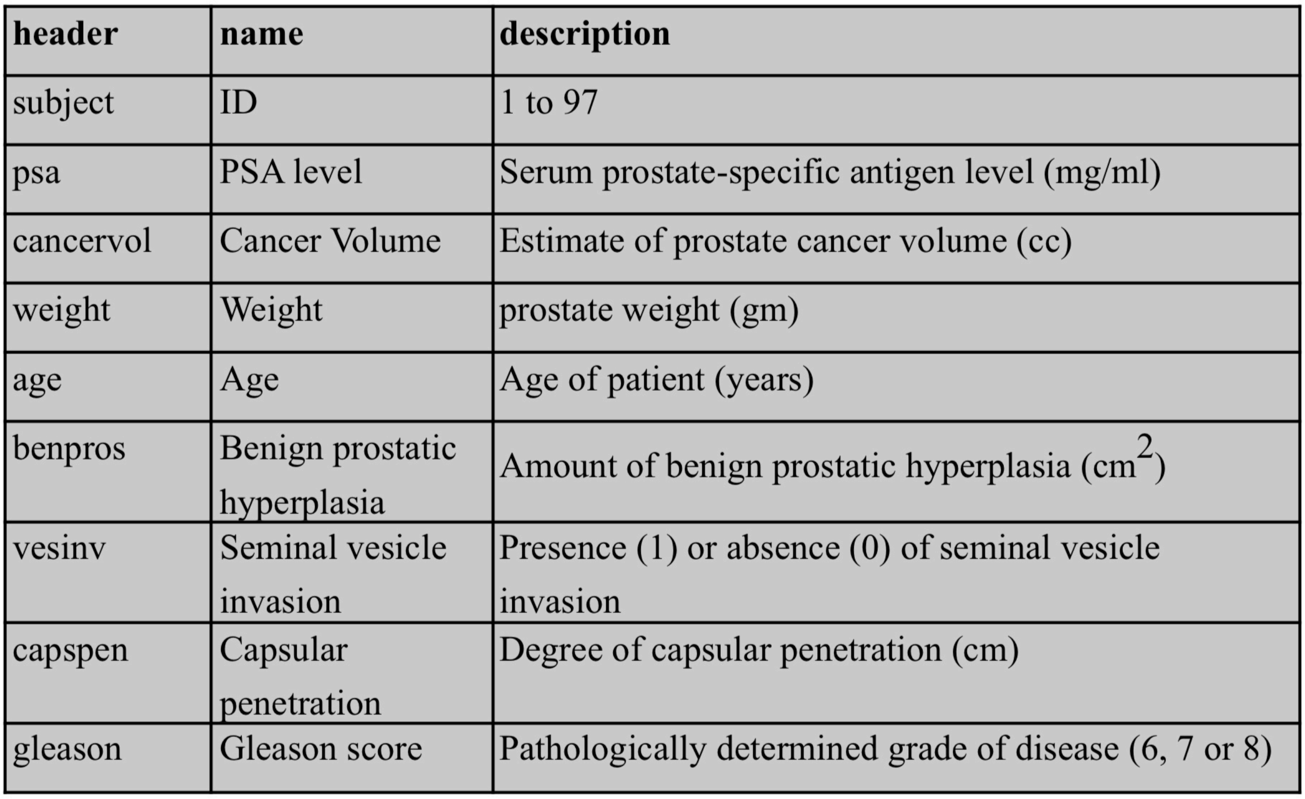
Mini Project 6 Report

Jiadao Zou: jxz172230
Houyi Liu: hxl163630

Question

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.



header	name	description
subject	ID	1 to 97
psa	PSA level	Serum prostate-specific antigen level (mg/ml)
cancervol	Cancer Volume	Estimate of prostate cancer volume (cc)
weight	Weight	prostate weight (gm)
age	Age	Age of patient (years)
benpros	Benign prostatic hyperplasia	Amount of benign prostatic hyperplasia (cm^2)
vesinv	Seminal vesicle invasion	Presence (1) or absence (0) of seminal vesicle invasion
capspen	Capsular penetration	Degree of capsular penetration (cm)
gleason	Gleason score	Pathologically determined grade of disease (6, 7 or 8)

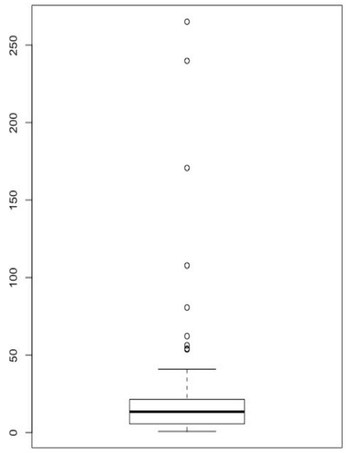
Figure 1: List of variables in the prostate cancer data

Answer

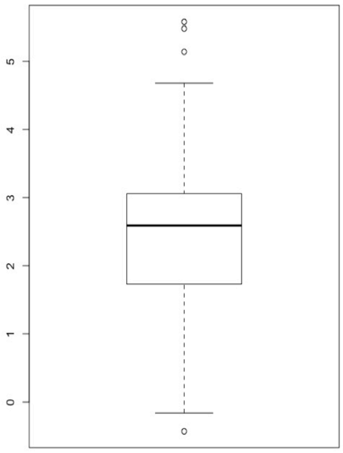
Analyse

- *First we draw the boxplot of the PSA*

```r
# Read data
data <- read.csv("prostate_cancer.csv")
# boxplot drawing
boxplot(data$psa, main="Distribution Graph of PSA levels")
```

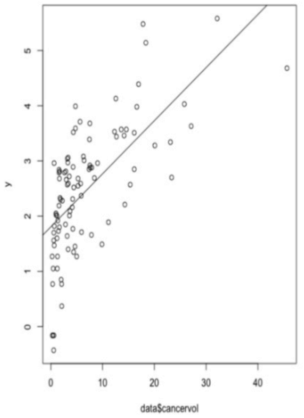


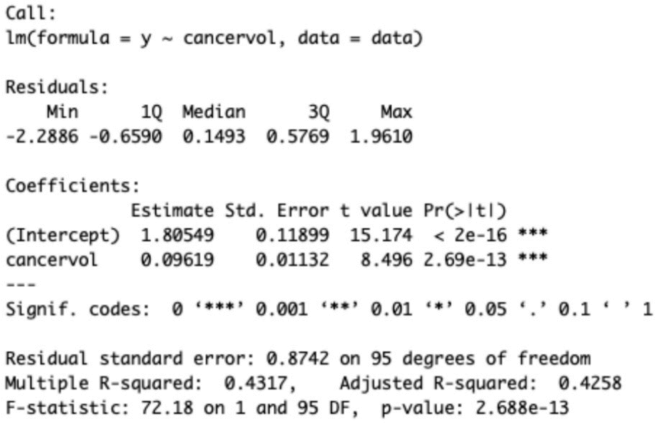
> ***As we could see, the original distribution is not good cause there are many outliers and two tails are not balanced.***

- > *Then, we plot the Natural Log distribution of PSA and have a look*
  >
  > ```
  > boxplot(log(data$psa), main="Distribution Graph of Log of PSA levels")
  > ```
  >
  > 
  >
  >
  > ***This time, natural log transformation makes the distribution less skewed and reduce the number of outliers.***
  >
  > ***Therefore, we should use transformed Distribution of PSA levels.***
  >
  > ```
  > y <- log(data$psa)
  > ```

- > *Next, we try to fit the response to each predictors left. Also, notice that "vesinv" is a qualitative variable and "gleason" is a quantitative value.*

- > - *Transformed PSA level && "cancervol"*
  >
  > ```
  > plot(data$cancervol, y, main="Transformed PSA level && cancervol)
  > fit1 <- lm(y ~ cancervol, data = data)
  > summary(fit1)
  > ```

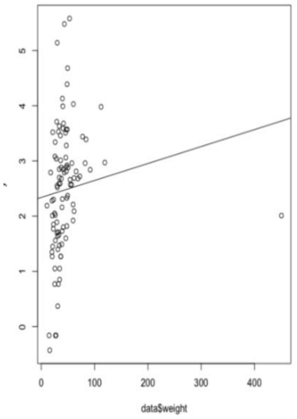


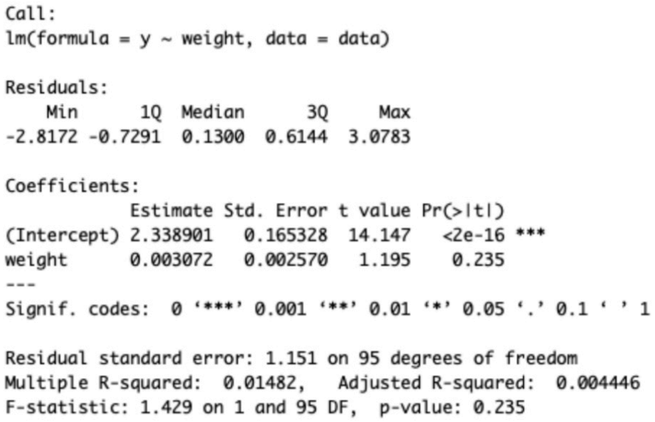


```
Call:
lm(formula = y ~ cancervol, data = data)

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
```

- *Transformed PSA level && "Weight"*

```
plot(data$weight, y, main="Transformed PSA level && weight)
fit2 <- lm(y ~ weight, data = data)
summary(fit2)
```

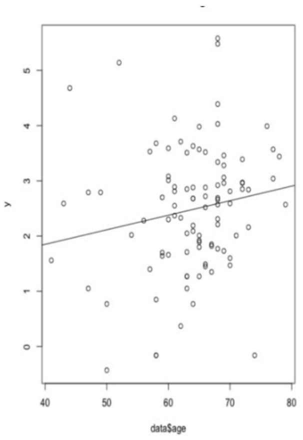


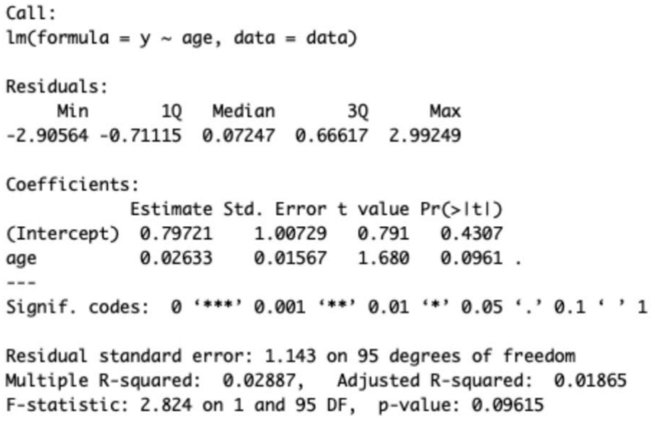


```
Call:
lm(formula = y ~ weight, data = data)

Residuals:
    Min      1Q  Median      3Q     Max 
-2.8172 -0.7291  0.1300  0.6144  3.0783 

Coefficients:
            Estimate Std. Error t value Pr(>|t|)    
(Intercept) 2.338901   0.165328  14.147   <2e-16 ***
weight      0.003072   0.002570   1.195    0.235    
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 1.151 on 95 degrees of freedom
Multiple R-squared:  0.01482,	Adjusted R-squared:  0.004446 
F-statistic: 1.429 on 1 and 95 DF,  p-value: 0.235
```

- *Transformed PSA level && "Age"*

```
plot(data$age, y, main="Transformed PSA level && Age)
fit3 <- lm(y ~ age, data = data)
summary(fit3)
```

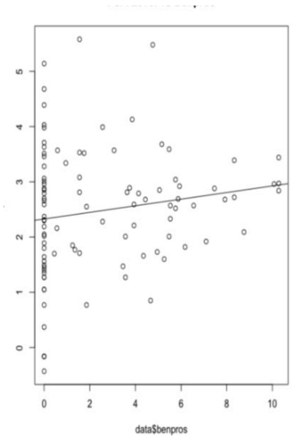


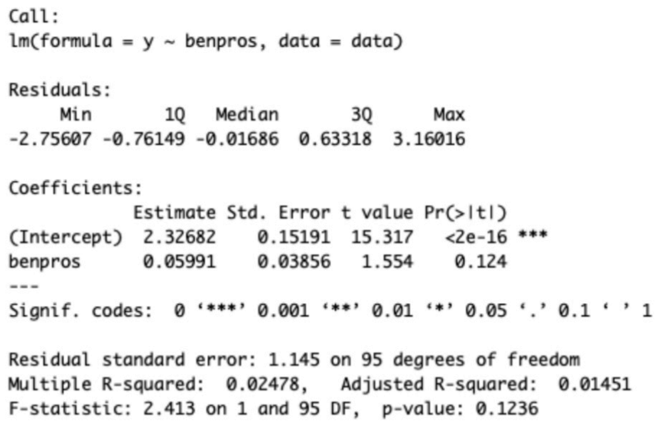


```
Call:
lm(formula = y ~ age, data = data)

Residuals:
     Min       1Q   Median       3Q      Max 
-2.90564 -0.71115  0.07247  0.66617  2.99249 

Coefficients:
            Estimate Std. Error t value Pr(>|t|)  
(Intercept)  0.79721    1.00729   0.791   0.4307  
age          0.02633    0.01567   1.680   0.0961 .
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 1.143 on 95 degrees of freedom
Multiple R-squared:  0.02887,	Adjusted R-squared:  0.01865 
F-statistic: 2.824 on 1 and 95 DF,  p-value: 0.09615
```

- *Transformed PSA level && "Benpros"*

```
plot(data$benpros, y, main="Transformed PSA level && Benpros)
fit4 <- lm(y ~ benpros, data = data)
summary(fit4)
```

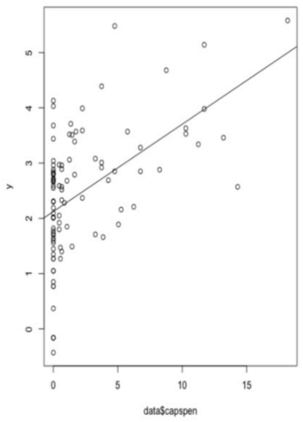


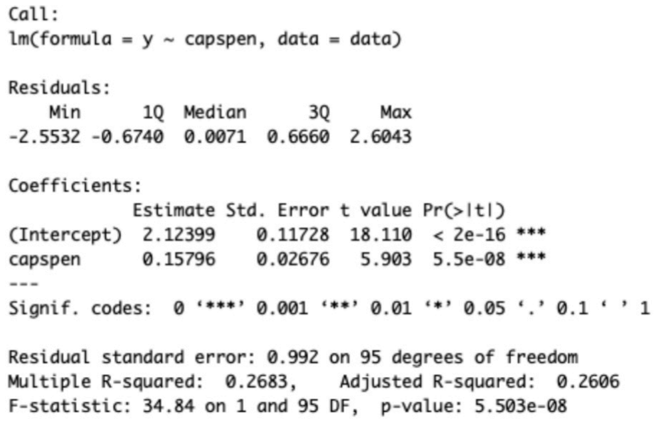


```
Call:
lm(formula = y ~ benpros, data = data)

Residuals:
     Min       1Q   Median       3Q      Max 
-2.75607 -0.76149 -0.01686  0.63318  3.16016 

Coefficients:
            Estimate Std. Error t value Pr(>|t|)    
(Intercept)  2.32682    0.15191  15.317   <2e-16 ***
benpros      0.05991    0.03856   1.554    0.124    
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 1.145 on 95 degrees of freedom
Multiple R-squared:  0.02478,	Adjusted R-squared:  0.01451 
F-statistic: 2.413 on 1 and 95 DF,  p-value: 0.1236
```

- *Transformed PSA level && "Capspen"*

```
plot(data$capsen, y, main="Transformed PSA level && Capsen)
fit5 <- lm(y ~ capsen, data = data)
summary(fit5)
```

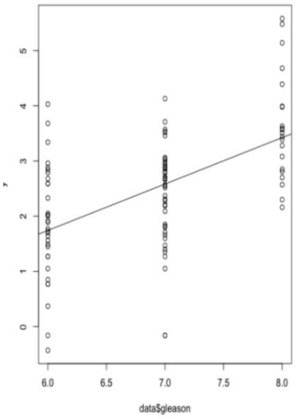


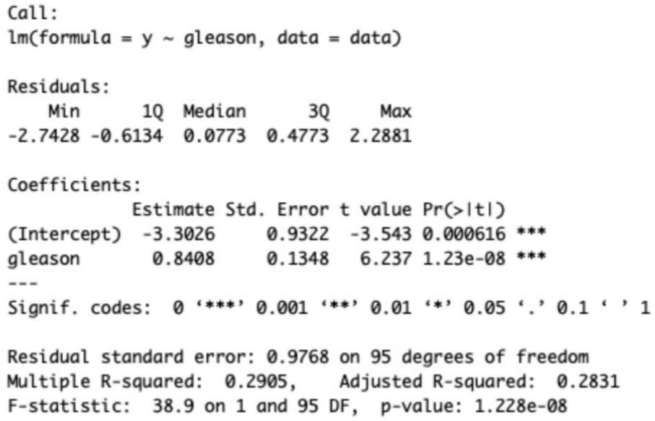


```
Call:
lm(formula = y ~ capspen, data = data)

Residuals:
    Min      1Q  Median      3Q     Max 
-2.5532 -0.6740  0.0071  0.6660  2.6043 

Coefficients:
            Estimate Std. Error t value Pr(>|t|)    
(Intercept)  2.12399    0.11728  18.110  < 2e-16 ***
capspen      0.15796    0.02676   5.903  5.5e-08 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.992 on 95 degrees of freedom
Multiple R-squared:  0.2683,	Adjusted R-squared:  0.2606 
F-statistic: 34.84 on 1 and 95 DF,  p-value: 5.503e-08
```

- *Transformed PSA level && "Gleason"*

```
plot(data$gleason, y, main="Transformed PSA level && Gleason)
fit6 <- lm(y ~ gleason, data = data)
summary(fit6)
```

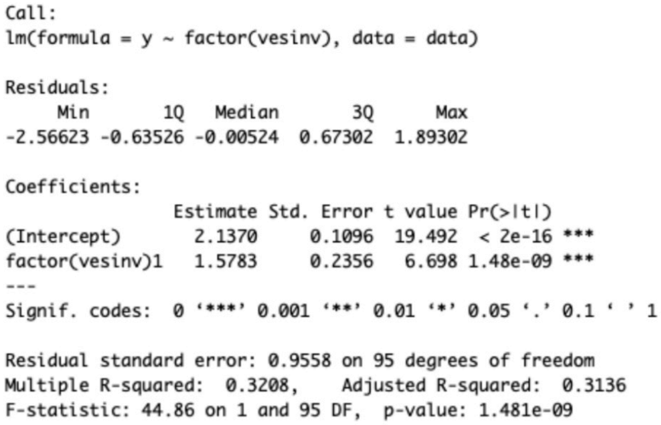




```
Call:
lm(formula = y ~ gleason, data = data)

Residuals:
    Min      1Q  Median      3Q     Max 
-2.7428 -0.6134  0.0773  0.4773  2.2881 

Coefficients:
            Estimate Std. Error t value Pr(>|t|)    
(Intercept)  -3.3026     0.9322  -3.543 0.000616 ***
gleason       0.8408     0.1348   6.237 1.23e-08 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.9768 on 95 degrees of freedom
Multiple R-squared:  0.2905,	Adjusted R-squared:  0.2831 
F-statistic:  38.9 on 1 and 95 DF,  p-value: 1.228e-08
```

- *Transformed PSA level && "vesinv"*

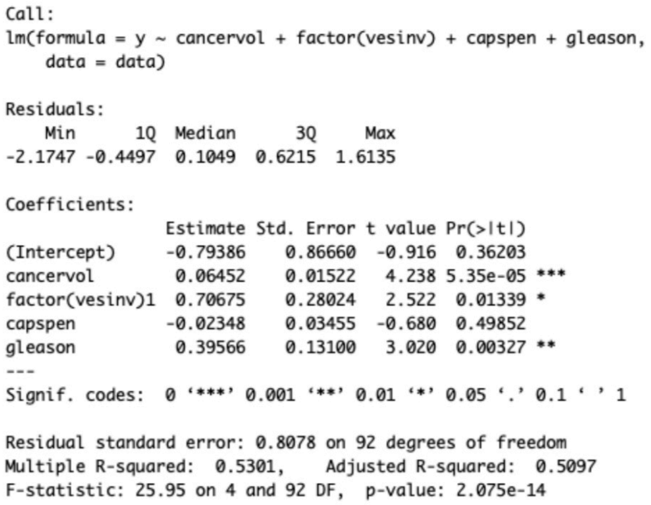
```
plot(data$vesinv, y, main="Transformed PSA level && Vesinv)
fit7 <- lm(y ~ vesinv, data = data)
summary(fit7)
```



```
Call:
lm(formula = y ~ factor(vesinv), data = data)

Residuals:
     Min       1Q   Median       3Q      Max 
-2.56623 -0.63526 -0.00524  0.67302  1.89302 

Coefficients:
                Estimate Std. Error t value Pr(>|t|)    
(Intercept)       2.1370     0.1096  19.492  < 2e-16 ***
factor(vesinv)1   1.5783     0.2356   6.698 1.48e-09 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.9558 on 95 degrees of freedom
Multiple R-squared:  0.3208,	Adjusted R-squared:  0.3136 
F-statistic: 44.86 on 1 and 95 DF,  p-value: 1.481e-09
```

- *From the above summary:*
  - ***As we have seen, features: {cancervol, capspen, gleason and vesinv} are significant predictors because their t-test p-values are $\leq$ 0.05.***
  - *Build a linear model with above significant predictors*

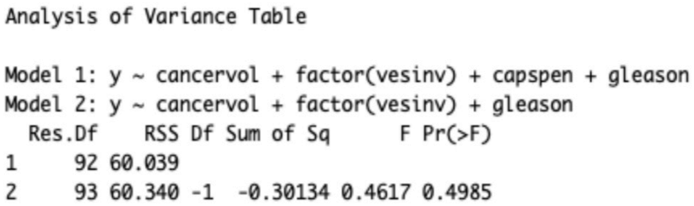
```
fit8 <- lm(y ~ cancervol + factor(vesinv) + capspen + gleason, data=data)
summary(fit8)
```



```
Call:
lm(formula = y ~ cancervol + factor(vesinv) + capspen + gleason,
    data = data)

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 ***
factor(vesinv)1  0.70675    0.28024   2.522  0.01339 *  
capspen         -0.02348    0.03455  -0.680  0.49852    
gleason          0.39566    0.13100   3.020  0.00327 ** 
---
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 could see that "capspen" is not significant. To verify it, we should use ANOVA table:

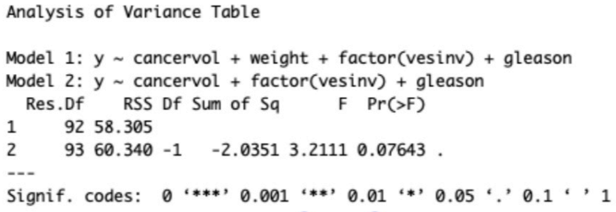
```
fit9 <- lm(y ~ cancervol + factor(vesinv) + gleason, data=data)
anova(fit8, fit9)
```



```
Analysis of Variance Table

Model 1: y ~ cancervol + factor(vesinv) + capspen + gleason
Model 2: y ~ cancervol + factor(vesinv) + gleason
  Res.Df    RSS Df Sum of Sq      F Pr(>F)
1     92 60.039                           
2     93 60.340 -1  -0.30134 0.4617 0.4985
```

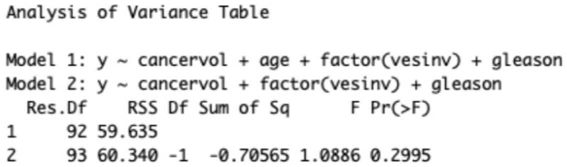
Since the P−value is $\gg 0.05$, it indicates "capspen" is insignificant. ***Also, we want to make sure the features we drop at the very beginning are really unimportant.***

```
fit10 <- lm(y ~ cancervol + weight + factor(vesinv) + gleason, data=data)
anova(fit10, fit9)
```



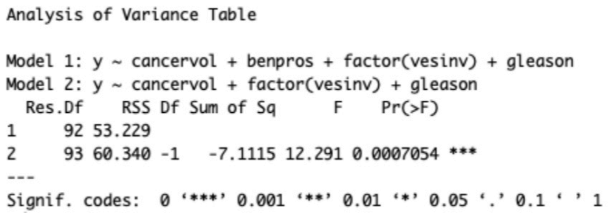
```
Analysis of Variance Table

Model 1: y ~ cancervol + weight + factor(vesinv) + gleason
Model 2: y ~ cancervol + factor(vesinv) + gleason
  Res.Df    RSS Df Sum of Sq      F  Pr(>F)  
1     92 58.305                              
2     93 60.340 -1   -2.0351 3.2111 0.07643 .
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

```
fit11 <- lm(y ~ cancervol + age +factor(vesinv) + gleason, data=data)
anova(fit11, fit9)
```



```
Analysis of Variance Table

Model 1: y ~ cancervol + age + factor(vesinv) + gleason
Model 2: y ~ cancervol + factor(vesinv) + gleason
  Res.Df    RSS Df Sum of Sq      F Pr(>F)
1     92 59.635                           
2     93 60.340 -1  -0.70565 1.0886 0.2995
```

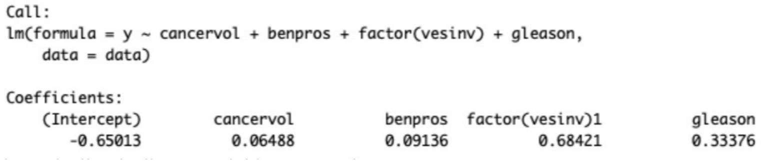
```
fit12 <- lm(y ~ cancervol + benpros + factor(vesinv) + gleason, data=data)
anova(fit12, fit9)
```



```
Analysis of Variance Table

Model 1: y ~ cancervol + benpros + factor(vesinv) + gleason
Model 2: y ~ cancervol + factor(vesinv) + gleason
  Res.Df    RSS Df Sum of Sq      F    Pr(>F)
1     92 53.229
2     93 60.340 -1   -7.1115 12.291 0.0007054 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

From the above images, anova shows "benpros" is important. Which means the feature we should use to construct the linear model are {cancervol, benpros, vesinv, gleason}, and we could see the final model as below:

```
fit12
```



```
Call:
lm(formula = y ~ cancervol + benpros + factor(vesinv) + gleason,
    data = data)

Coefficients:
    (Intercept)        cancervol          benpros  factor(vesinv)1          gleason
       -0.65013          0.06488          0.09136          0.68421          0.33376
```

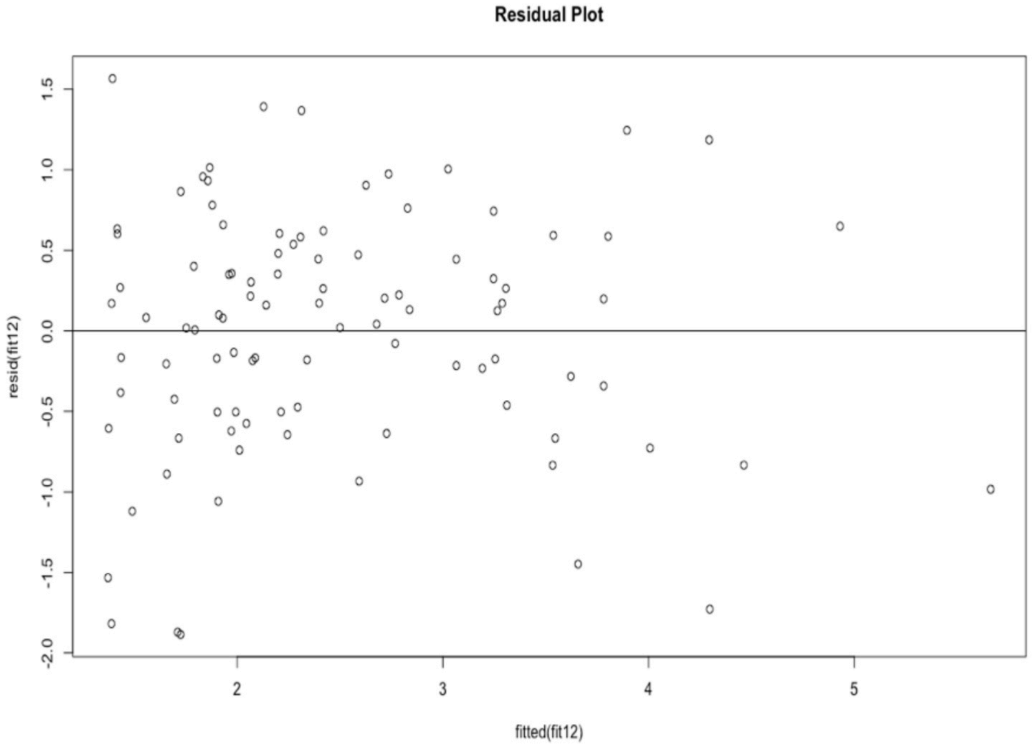
So, the mathematics question:

$$ln(PSA) = -0.65013 + 0.06488 * cancervol + 0.09136 * benpros + 0.6842(vesinv = 1) + 0.33376 * gleason$$

- ***Analyse our model***

  - *Now, plot the residual graph for the linear model we build:*

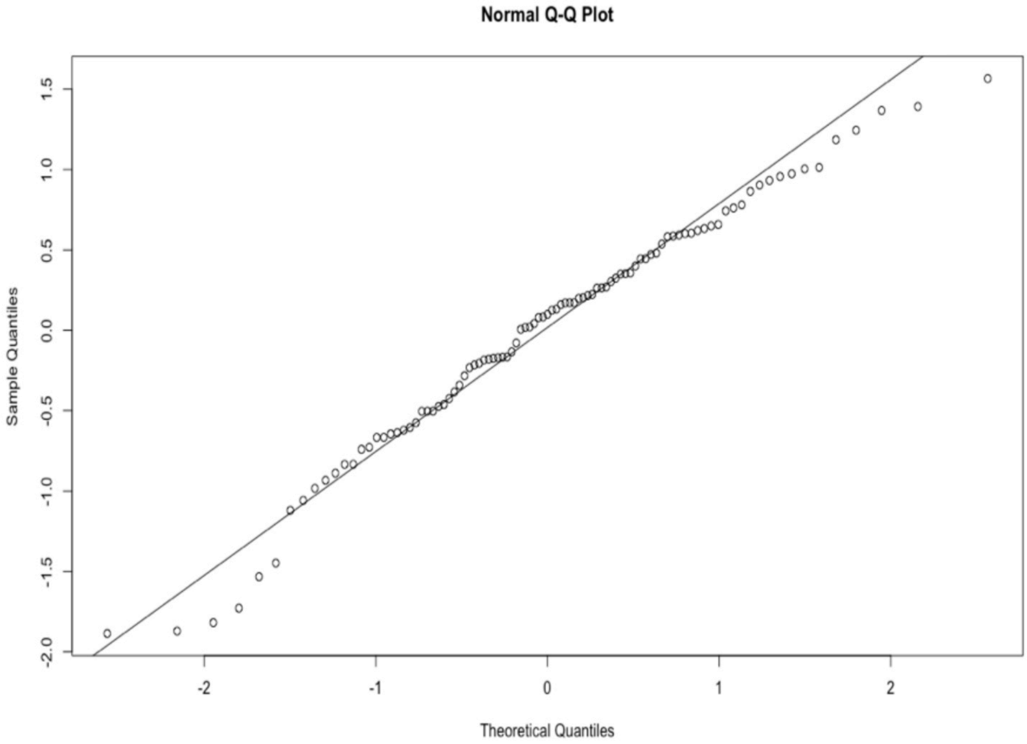
```
plot(fitted(fit12), resid(fit12), main="Residual Plot")
abline(h=0)
```



Resident points are scattered around zero and there is no obvious pattern of these points.

  - *Now, plot the Normal Q–Q plot for the linear model we build:*

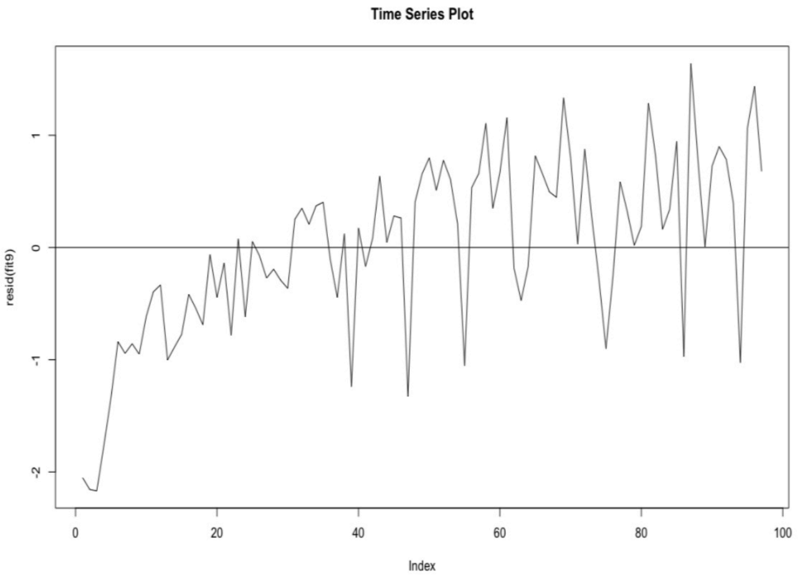
```
qqnorm(resid(fit12))
qqline(resid(fit12))
```



From above plot, It’s good if residuals are lined well on the straight dashed line. In other words, residual points are approximately around the straight line which means the errors are normally distributed.

- *Then, we take a look at time series plot:*

```
plot(resid(fit9), type="1", main="Tine Series Plot")
abline(h=0)
```

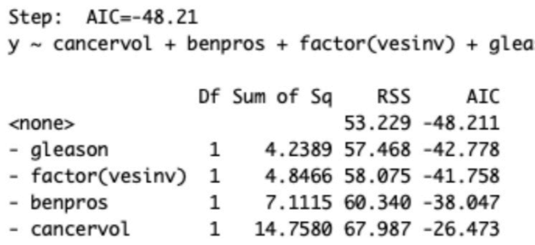


From that series plot, we could not see there is obvious pattern between time interval and residual points. That shows out model is good because errors are independent.

- ***Final step, comparing our model with AIC generated models***

  - *Backward AIC:*

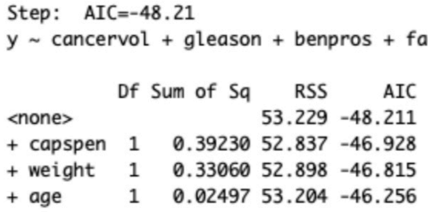
```
fit13.backward <- step(lm(y ~ cancervol + weight + age + benpros + factor(vesinv) + capspen + gleason, data=data), scope = list(lower = ~-1), direction = "backward")
```



```
Step:  AIC=-48.21
y ~ cancervol + benpros + factor(vesinv) + gleason

                 Df Sum of Sq    RSS     AIC
<none>                        53.229 -48.211
- gleason         1    4.2389 57.468 -42.778
- 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
```

- *Forward AIC:*

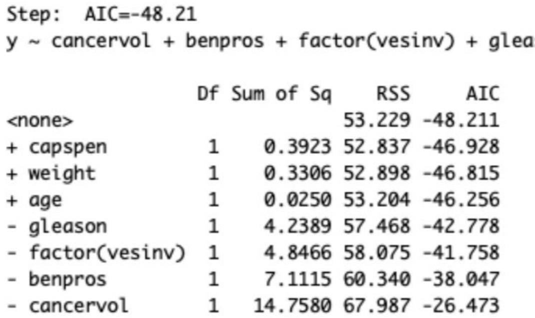
```
fit13.forward <- step(lm(y ~ 1, data=data), scope = list(upper = ~cancervol + weight + age + benpros + factor
(vesinv) + capspen + gleason), direction = "forward")
```



```
Step:  AIC=-48.21
y ~ cancervol + gleason + benpros + factor(vesinv)

          Df Sum of Sq    RSS     AIC
<none>                 53.229 -48.211
+ capspen  1   0.39230 52.837 -46.928
+ weight   1   0.33060 52.898 -46.815
+ age      1   0.02497 53.204 -46.256
```

- *Both AIC:*

```
fit13.both <- step(lm(y ~ cancervol + weight + age + benpros + factor(vesinv) + capspen + gleason, data=data)
), scope = list(upper = ~cancervol + weight + age + benpros + factor(vesinv) + capspen + gleason), direction
= "both")
```

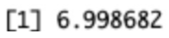


```
Step:  AIC=-48.21
y ~ cancervol + benpros + factor(vesinv) + gleason

                 Df Sum of Sq    RSS     AIC
<none>                        53.229 -48.211
+ capspen         1    0.3923 52.837 -46.928
+ weight          1    0.3306 52.898 -46.815
+ age             1    0.0250 53.204 -46.256
- gleason         1    4.2389 57.468 -42.778
- 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
```

- ***Result of above three different stepwise model selection methods agree with our model.***

- ***Predicting:***
  - *concervol:*

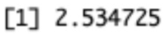
```
concervol <- mean(data$concervol)
concervol
```



```
[1] 6.998682
```

  - *benpros:*

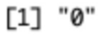
```
benpros <- mean(data$benpros)
benpros
```



```
[1] 2.534725
```

  - *vesinv:*

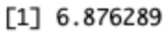
```
vesinv.t <- table(factor(data$vesinv))
vesinv <- names(which.max(vesinv.t))
vesinv
```



```
[1] "0"
```

- *gleason*

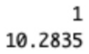
```
gleason <- mean(data$gleason)
gleason
```



```
[1] 6.876289
```

- *Predicting the response by current predictor and arguments*

```
arguments <- data.frame(cancervol: cancervol, benpros: benpros, vesinv: vesinv, gleason: gleason)
PSA_log_response <- predict(fit12, arguments)
exp(PSA_log_response)
```



```
      1 
10.2835 
```

So the predicted PSA level is 10.2835.