Mini project #6

Group Member: Chaoran Li, Wenting Wang

Contribution of each member:

Firstly, we discussed the mathematical models and code details together. Then, we divided the project into two part and finished our respective work. Chaoran Li worked on coding and Wenting Wang worked on analyzing. Then, we checked and reviewed our report together. Each member makes contribution to this project as the details shown in table 1.

	Question1
Chaoran li	50%
Wenting wang	50%

Table 1: Member contribution table

Question 1:

Build a "reasonably good" linear model for PSA level in prostate.cancer.csv data set.

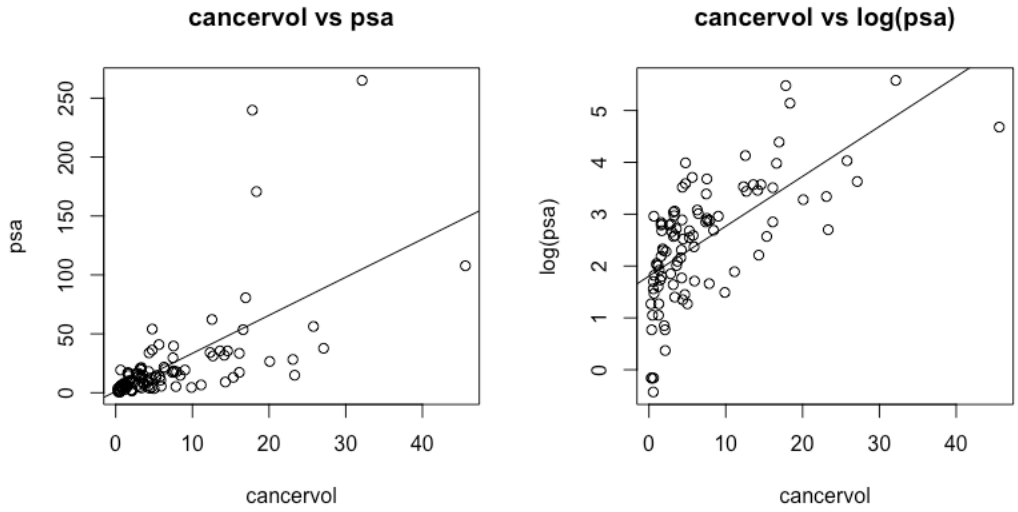
1) Firstly, we need to prepare and explore the data. Because the vesinv is quantitative variable, we need to convert it to factor with 1 dummy variable.

```
> # Question 1
> # Note that vesinv is a qualitative variable.
> # You can treat gleason as a quantitative variable.
> # 1) Load and prepare data
> prostate_cancer <- read.csv(
+   file=file.path("./Mini Project 6/prostate_cancer.csv"))
> str(prostate_cancer)
'data.frame':   97 obs. of  9 variables:
 $ subject  : int  1 2 3 4 5 6 7 8 9 10 ...
 $ psa      : num  0.651 0.852 0.852 0.852 1.448 ...
 $ cancervol: num  0.56 0.372 0.601 0.301 2.117 ...
 $ weight   : num  16 27.7 14.7 26.6 30.9 ...
 $ age      : int  50 58 74 58 62 50 64 58 47 63 ...
 $ benpros  : num  0 0 0 0 0 ...
 $ vesinv   : int  0 0 0 0 0 0 0 0 0 0 ...
 $ capspen  : num  0 0 0 0 0 0 0 0 0 0 ...
 $ gleason  : int  6 7 7 6 6 6 6 7 6 ...
```

```
> index = prostate_cancer$subject
> pas = prostate_cancer$psa
> logpsa = log(prostate_cancer$psa)
> cancervol = prostate_cancer$cancervol
> weight = prostate_cancer$weight
> age = prostate_cancer$age
> benpros = prostate_cancer$benpros
> vesinv = prostate_cancer$vesinv
> capspen = prostate_cancer$capspen
> gleason = prostate_cancer$gleason
> table(vesinv)
vesinv
 0  1
76 21
> # vesinv is a qualitative variable with 2 values
> # Automatically represent with 1 dummy variable: factor(vesinv)
> vesinv.factor1 = ifelse(vesinv == 1, 1, 0)
```

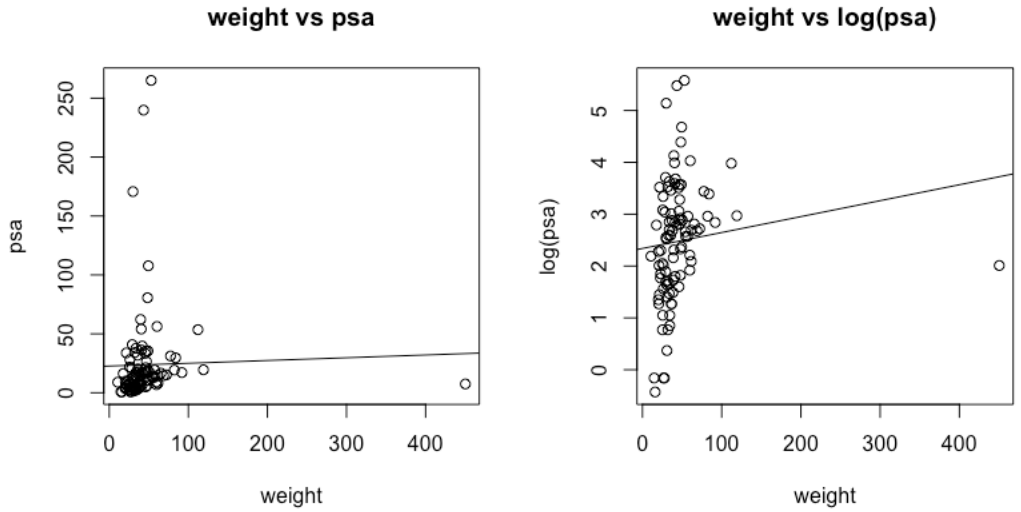
2) Then, we analyze the data with simple linear regression for both psa and the log transformation log(psa).

a) cancervol: From the two plots below, we can find that, as univariate regression, log(psa) performs much better positive linear trend.



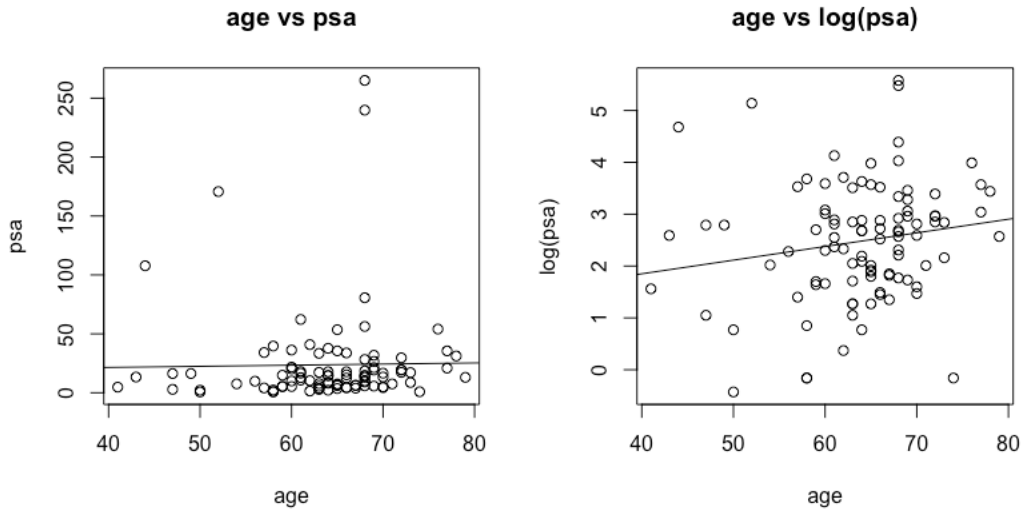

```
> # 2) Analyze the data with simple linear regression first
> # cancervol vs psa
> plot(cancervol, psa, type="p", main="cancervol vs psa")
> abline(lm(psa ~ cancervol))
> plot(cancervol, logpsa, type="p", ylab="log(psa)", main="cancervol vs log(psa)")
> abline(lm(logpsa ~ cancervol))
> paste("cancervol vs psa: ", cor(psa, cancervol),
+       "; log(psa): ", cor(logpsa, cancervol))
[1] "cancervol vs psa:  0.624150588319316 ; log(psa):  0.657073936076788"
```

b) weight: From the two plots below, we can find that, as univariate regression, both psa and log(psa) do not perform very good linear trend, but log(psa) is better.



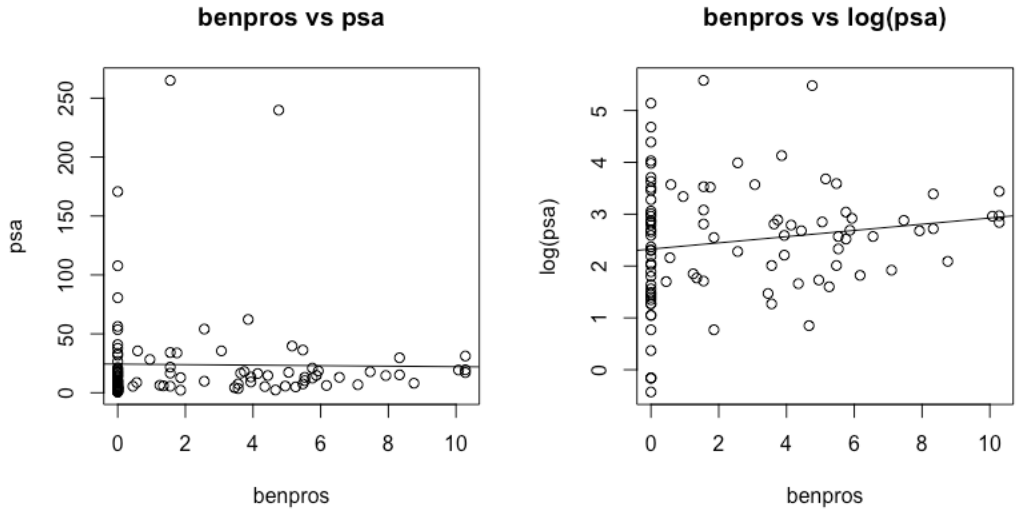

```
> # weight vs psa
> plot(weight, psa, type="p", main="weight vs psa")
> abline(lm(psa ~ weight))
> plot(weight, logpsa, type="p", ylab="log(psa)", main="weight vs log(psa)")
> abline(lm(logpsa ~ weight))
> paste("weight vs psa: ", cor(psa, weight),
+       "; log(psa): ", cor(logpsa, weight))
[1] "weight vs psa:  0.0262134297405694 ; log(psa):  0.121720757056228"
```

c) age: From the two plots below, we can find that, as univariate regression, log(psa) performs much better positive linear trend.



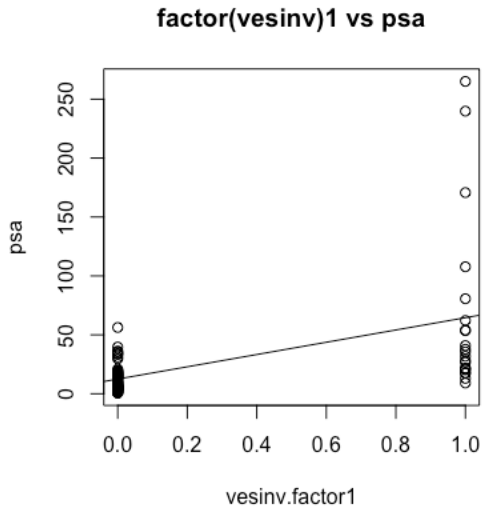
```
> # age vs psa
> plot(age, psa, type="p", main="age vs psa")
> abline(lm(psa ~ age))
> plot(age, logpsa, type="p", ylab="log(psa)", main="age vs log(psa)")
> abline(lm(logpsa ~ age))
> paste("age vs psa: ", cor(psa, age),
+       "; log(psa): ", cor(logpsa, age))
[1] "age vs psa:  0.0171993776381882 ; log(psa):  0.169906822489551"
```

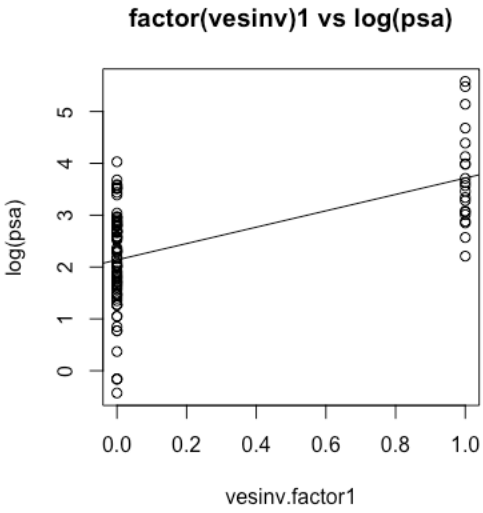
d) benpros: From the two plots below, we can find that, as univariate regression, psa shows negative correlation, but log(psa) shows positive linear relationship.




```
> # benpros vs psa
> plot(benpros, psa, type="p", main="benpros vs psa")
> abline(lm(psa ~ benpros))
> plot(benpros, logpsa, type="p", ylab="log(psa)", main="benpros vs log(psa)")
> abline(lm(logpsa ~ benpros))
> paste("benpros vs psa: ", cor(psa, benpros),
+       "; log(psa): ", cor(logpsa, benpros))
[1] "benpros vs psa:  -0.0164864930981332 ; log(psa):  0.157401578697896"
```

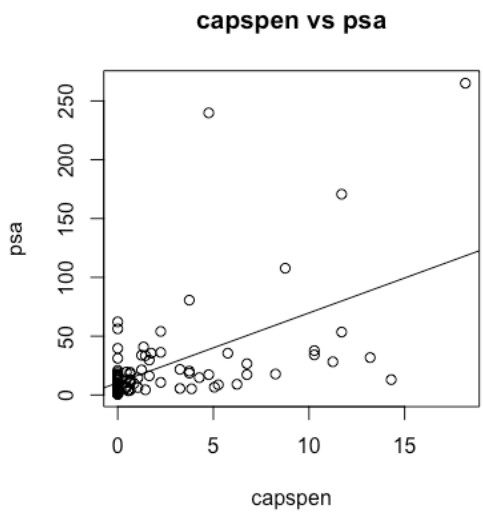
e) factor(vesinv)1: Because vesinv is a qualitative variable, there exist only two possible values i.e. 0 or 1. And there are more people have Seminal vesicle invasion, who are the people with dummy variable 0.

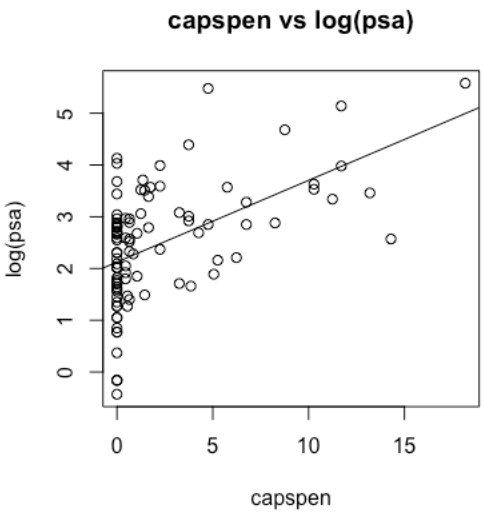





```
> # factor(vesinv)1 vs psa
> plot(vesinv.factor1, psa, type="p", main="factor(vesinv)1 vs psa")
> abline(lm(psa ~ vesinv.factor1))
> plot(vesinv.factor1, logpsa, type="p", ylab="log(psa)",
+      main="factor(vesinv)1 vs log(psa)")
> abline(lm(logpsa ~ vesinv.factor1))
> paste("factor(vesinv)1 vs psa: ", cor(psa, vesinv.factor1),
+       "; log(psa): ", cor(logpsa, vesinv.factor1))
[1] "factor(vesinv)1 vs psa:  0.528618784780521 ; log(psa):  0.566364086881402"
```

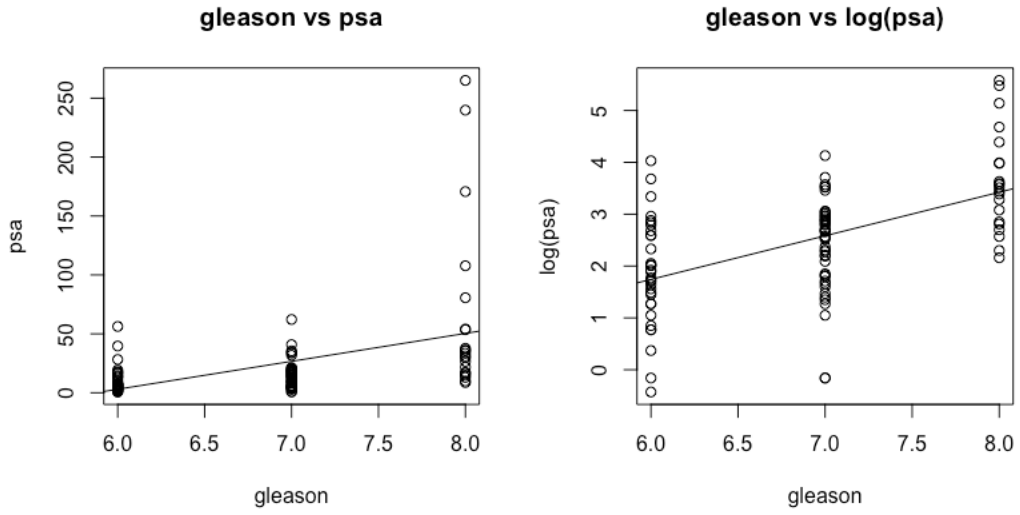
f) capspen: From the two plots below, we can find that, as univariate regression, both psa and log(psa) show linear relationship, but psa shows better positive linear relationship.





```
> # capspen vs psa
> plot(capspen, psa, type="p", main="capspen vs psa")
> abline(lm(psa ~ capspen))
> plot(capspen, logpsa, type="p", ylab="log(psa)", main="capspen vs log(psa)")
> abline(lm(logpsa ~ capspen))
> paste("capspen vs psa: ", cor(psa, capspen),
+       "; log(psa): ", cor(logpsa, capspen))
[1] "capspen vs psa:  0.550792516672696 ; log(psa):  0.518023107610481"
```

g) gleason: Although there only exist three value for gleason, we still can find that, as univariate regression, log(psa) performs much better positive linear trend.




```
> # gleason vs psa
> plot(gleason, psa, type="p", main="gleason vs psa")
> abline(lm(psa ~ gleason))
> plot(gleason, logpsa, type="p", ylab="log(psa)", main="gleason vs log(psa)")
> abline(lm(logpsa ~ gleason))
> paste("gleason vs psa: ", cor(psa, gleason),
+       "; log(psa): ", cor(logpsa, gleason))
[1] "gleason vs psa:  0.429579750396728 ; log(psa):  0.539016748795237"
```

Above all, nearly all regressions with log(pas) are better than that with psa. Hence, we would use log(pas) afterwards.

3) Next, we explore multiple linear regression

a) Start with full model:

```
> # 3) Multiple linear regression
> # Start with full model
> fit1 = lm(logpsa ~ cancervol + weight + age +
+             benpros + factor(vesinv) + capspen + gleason)
> summary(fit1)

Call:
lm(formula = logpsa ~ cancervol + weight + age + benpros + factor(vesinv) +
    capspen + gleason)

Residuals:
     Min       1Q   Median       3Q      Max
-1.88309 -0.46629  0.08045  0.47380  1.53219

Coefficients:
                Estimate Std. Error t value Pr(>|t|)
(Intercept)     -0.685796   0.998754  -0.687  0.49409
cancervol        0.069454   0.014624   4.749 7.77e-06 ***
weight           0.001380   0.001822   0.757  0.45079
age             -0.002799   0.011724  -0.239  0.81186
benpros          0.087470   0.029605   2.955  0.00401 **
factor(vesinv)1  0.782623   0.268339   2.917  0.00448 **
capspen         -0.026521   0.032860  -0.807  0.42177
gleason          0.358153   0.127976   2.799  0.00629 **
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.7679 on 89 degrees of freedom
Multiple R-squared:  0.5893,    Adjusted R-squared:  0.557
F-statistic: 18.24 on 7 and 89 DF,  p-value: 7.694e-15
```

From the Coefficients, we find the P-value of age is 0.81186, which fail to reject the Null Hypothesis $H_0: \beta_{age} = 0$. Thus, we can try to drop the most impossible predictor ‘age’.

b) Drop age

```
> # Drop age
> fit2 = update(fit1, . ~ . - age)
> summary(fit2)

Call:
lm(formula = logpsa ~ cancervol + weight + benpros + factor(vesinv) +
    capspen + gleason)

Residuals:
     Min       1Q   Median       3Q      Max
-1.87226 -0.46558  0.08206  0.46484  1.50784

Coefficients:
                Estimate Std. Error t value Pr(>|t|)
(Intercept)     -0.817763   0.827461  -0.988  0.32567
cancervol        0.069705   0.014510   4.804 6.17e-06 ***
weight           0.001353   0.001809   0.748  0.45643
benpros          0.085103   0.027750   3.067  0.00286 **
factor(vesinv)1  0.777115   0.265941   2.922  0.00440 **
capspen         -0.026656   0.032682  -0.816  0.41688
gleason          0.352362   0.124995   2.819  0.00592 **
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.7638 on 90 degrees of freedom
Multiple R-squared:  0.589,     Adjusted R-squared:  0.5616
F-statistic:  21.5 on 6 and 90 DF,  p-value: 1.619e-15
```

‘Weight’ has a P-value of 0.45643 far more large than 0.05. Hence, we can drop the predictor ‘weight’.

c) Drop weight

```
> # Drop weight
> fit3 = update(fit2, . ~ . - weight)
> summary(fit3)

Call:
lm(formula = logpsa ~ cancervol + benpros + factor(vesinv) +
    capspen + gleason)

Residuals:
     Min       1Q   Median       3Q      Max
-1.88954 -0.48197  0.08813  0.48409  1.57370

Coefficients:
                Estimate Std. Error t value Pr(>|t|)
(Intercept)     -0.73258    0.81760  -0.896 0.372608
cancervol        0.07029    0.01445   4.863 4.82e-06 ***
benpros          0.09198    0.02612   3.522 0.000672 ***
factor(vesinv)1  0.78233    0.26520   2.950 0.004041 **
capspen         -0.02680    0.03260  -0.822 0.413237
gleason          0.34568    0.12437   2.779 0.006617 **
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.762 on 91 degrees of freedom
Multiple R-squared:  0.5865,    Adjusted R-squared:  0.5637
F-statistic: 25.81 on 5 and 91 DF,  p-value: 3.931e-16
```

‘Capspen’ has a P-value of 0.413237 far more large than 0.05. Hence, we can drop the predictor ‘capspen’. Besides, ‘capspen’ is the only predictor which prefers psa better than log(psa). If we drop it, the final model would have better linear trend.

d) Drop capspen

```
> # Drop capspen (which prefer pas better than log(psa))
> fit4 = update(fit3, . ~ . - capspen)
> summary(fit4)

Call:
lm(formula = logpsa ~ cancervol + benpros + factor(vesinv) +
    gleason)

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 ***
benpros          0.09136    0.02606   3.506 0.000705 ***
factor(vesinv)1  0.68421    0.23640   2.894 0.004746 **
gleason          0.33376    0.12331   2.707 0.008100 **
---
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
```

Reject all H_0 here. Compare existing models:

```
> # Reject all H0 here
> anova(fit1, fit2, fit3, fit4)
Analysis of Variance Table

Model 1: logpsa ~ cancervol + weight + age + benpros + factor(vesinv) +
    capspen + gleason
Model 2: logpsa ~ cancervol + weight + benpros + factor(vesinv) + capspen +
    gleason
Model 3: logpsa ~ cancervol + benpros + factor(vesinv) + capspen + gleason
Model 4: logpsa ~ cancervol + benpros + factor(vesinv) + gleason
  Res.Df    RSS Df Sum of Sq      F Pr(>F)
1     89 52.477
2     90 52.510 -1  -0.03360 0.0570 0.8119
3     91 52.837 -1  -0.32642 0.5536 0.4588
4     92 53.229 -1  -0.39230 0.6653 0.4169
```

We stop here. For it seems that we dropped 3 parameters which are clear not important and the other 4 parameters show good linear trend.

4) Verify our result by model selection with BIC

We would use all three kind of model selections and check the results with the model we got above.

a) Forward selection:

```
> # 4) Verify by model selection with BIC
> nullmd = lm(logpsa ~ 1)
> fullmd = lm(logpsa ~ cancervol + weight + age +
+               benpros + factor(vesinv) + capspen + gleason)
> forward = step(nullmd, scope=list(lower=nullmd, upper=fullmd),
+                direction="forward", k=log(length(logpsa)))

Start:  AIC=31.3
logpsa ~ 1

                 Df Sum of Sq     RSS     AIC
+ cancervol       1    55.164  72.605 -18.949
+ factor(vesinv)  1    40.984  86.785  -1.645
+ gleason         1    37.122  90.647   2.579
+ capspen         1    34.286  93.482   5.566
<none>                         127.769  31.299
+ age             1     3.688 124.080  33.033
+ benpros         1     3.166 124.603  33.441
+ weight          1     1.893 125.876  34.426

Step:  AIC=-18.95
logpsa ~ cancervol

                 Df Sum of Sq    RSS     AIC
+ gleason         1    8.2468 64.358 -26.070
+ benpros         1    7.8034 64.802 -25.404
+ factor(vesinv)  1    6.5468 66.058 -23.541
<none>                        72.605 -18.949
+ age             1    2.6615 69.944 -17.997
+ weight          1    1.7901 70.815 -16.796
+ capspen         1    0.9673 71.638 -15.675

Step:  AIC=-26.07
logpsa ~ cancervol + gleason

                 Df Sum of Sq    RSS     AIC
+ benpros         1    6.2827 58.075 -31.459
+ factor(vesinv)  1    4.0178 60.340 -27.748
<none>                        64.358 -26.070
+ weight          1    2.0334 62.325 -24.609
+ age             1    0.9611 63.397 -22.954
+ capspen         1    0.1685 64.190 -21.749

Step:  AIC=-31.46
logpsa ~ cancervol + gleason + benpros

                 Df Sum of Sq    RSS     AIC
+ factor(vesinv)  1    4.8466 53.229 -35.337
<none>                        58.075 -31.459
+ weight          1    0.4006 57.675 -27.556
+ capspen         1    0.1863 57.889 -27.196
+ age             1    0.0059 58.070 -26.894

Step:  AIC=-35.34
logpsa ~ cancervol + gleason + benpros + factor(vesinv)

          Df Sum of Sq    RSS     AIC
<none>                 53.229 -35.337
+ capspen  1   0.39230 52.837 -31.480
+ weight   1   0.33060 52.898 -31.367
+ age      1   0.02497 53.204 -30.808
```

2) Backward elimination:

```
> # When scope is missing, default for direction is "backward"
> backward = step(fullmd, k=log(length(logpsa)))
Start:  AIC=-22.99
logpsa ~ cancervol + weight + age + benpros + factor(vesinv) +
    capspen + gleason

                 Df Sum of Sq    RSS      AIC
- age             1    0.0336 52.510 -27.5062
- weight          1    0.3383 52.815 -26.9451
- capspen         1    0.3841 52.861 -26.8610
<none>                       52.477 -22.9936
- gleason         1    4.6180 57.095 -19.3871
- factor(vesinv)  1    5.0155 57.492 -18.7141
- benpros         1    5.1469 57.624 -18.4927
- cancervol       1   13.2994 65.776  -5.6572

Step:  AIC=-27.51
logpsa ~ cancervol + weight + benpros + factor(vesinv) + capspen +
    gleason

                 Df Sum of Sq    RSS      AIC
- weight          1    0.3264 52.837 -31.4798
- capspen         1    0.3881 52.898 -31.3666
<none>                       52.510 -27.5062
- gleason         1    4.6365 57.147 -23.8734
- factor(vesinv)  1    4.9820 57.492 -23.2887
- benpros         1    5.4873 57.998 -22.4398
- cancervol       1   13.4654 65.976  -9.9381

Step:  AIC=-31.48
logpsa ~ cancervol + benpros + factor(vesinv) + capspen + gleason

                 Df Sum of Sq    RSS     AIC
- capspen         1    0.3923 53.229 -35.337
<none>                       52.837 -31.480
- gleason         1    4.4852 57.322 -28.151
- factor(vesinv)  1    5.0526 57.889 -27.196
- benpros         1    7.2024 60.039 -23.659
- cancervol       1   13.7311 66.568 -13.646

Step:  AIC=-35.34
logpsa ~ cancervol + benpros + factor(vesinv) + gleason

                 Df Sum of Sq    RSS     AIC
<none>                       53.229 -35.337
- gleason         1    4.2389 57.468 -32.479
- factor(vesinv)  1    4.8466 58.075 -31.459
- benpros         1    7.1115 60.340 -27.748
- cancervol       1   14.7580 67.987 -16.174
```

3) Stepwise selection:

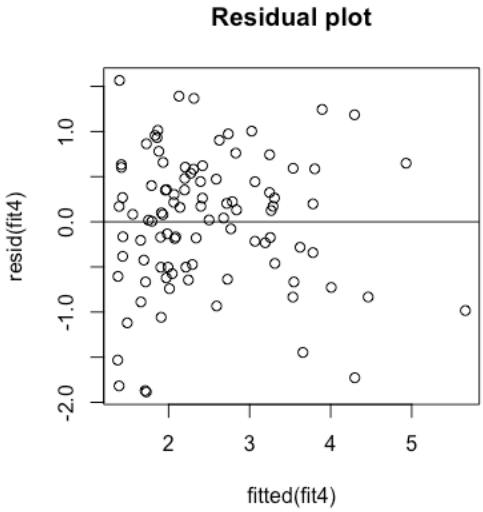
```
> # When scope is announced, default for direction is "both"
> both = step(nullmd, scope=list(lower=nullmd, upper=fullmd),
+             k=log(length(logpsa)))
Start:  AIC=31.3
logpsa ~ 1

                 Df Sum of Sq     RSS     AIC
+ cancervol       1    55.164  72.605 -18.949
+ factor(vesinv)  1    40.984  86.785  -1.645
+ gleason         1    37.122  90.647   2.579
+ capspen         1    34.286  93.482   5.566
<none>                         127.769  31.299
+ age             1     3.688 124.080  33.033
+ benpros         1     3.166 124.603  33.441
+ weight          1     1.893 125.876  34.426

Step:  AIC=-18.95
logpsa ~ cancervol

                 Df Sum of Sq     RSS     AIC
+ gleason         1     8.247  64.358 -26.070
+ benpros         1     7.803  64.802 -25.404
+ factor(vesinv)  1     6.547  66.058 -23.541
<none>                         72.605 -18.949
+ age             1     2.662  69.944 -17.997
+ weight          1     1.790  70.815 -16.796
+ capspen         1     0.967  71.638 -15.675
- cancervol       1    55.164 127.769  31.299

Step:  AIC=-26.07
logpsa ~ cancervol + gleason

                 Df Sum of Sq    RSS      AIC
+ benpros         1    6.2827 58.075 -31.4590
+ factor(vesinv)  1    4.0178 60.340 -27.7480
<none>                        64.358 -26.0697
+ weight          1    2.0334 62.325 -24.6093
+ age             1    0.9611 63.397 -22.9545
+ capspen         1    0.1685 64.190 -21.7493
- gleason         1    8.2468 72.605 -18.9492
- cancervol       1   26.2887 90.647   2.5788

Step:  AIC=-31.46
logpsa ~ cancervol + gleason + benpros

                 Df Sum of Sq    RSS     AIC
+ factor(vesinv)  1    4.8466 53.229 -35.337
<none>                        58.075 -31.459
+ weight          1    0.4006 57.675 -27.556
+ capspen         1    0.1863 57.889 -27.196
+ age             1    0.0059 58.070 -26.894
- benpros         1    6.2827 64.358 -26.070
- gleason         1    6.7262 64.802 -25.404
- cancervol       1   29.9589 88.034   4.317
```

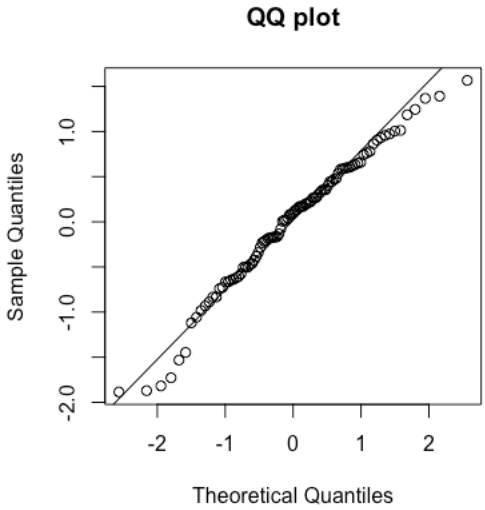
```
Step:  AIC=-35.34
logpsa ~ cancervol + gleason + benpros + factor(vesinv)

                 Df Sum of Sq    RSS     AIC
<none>                        53.229 -35.337
- gleason         1    4.2389 57.468 -32.479
+ capspen         1    0.3923 52.837 -31.480
- factor(vesinv)  1    4.8466 58.075 -31.459
+ weight          1    0.3306 52.898 -31.367
+ age             1    0.0250 53.204 -30.808
- benpros         1    7.1115 60.340 -27.748
- cancervol       1   14.7580 67.987 -16.174
```

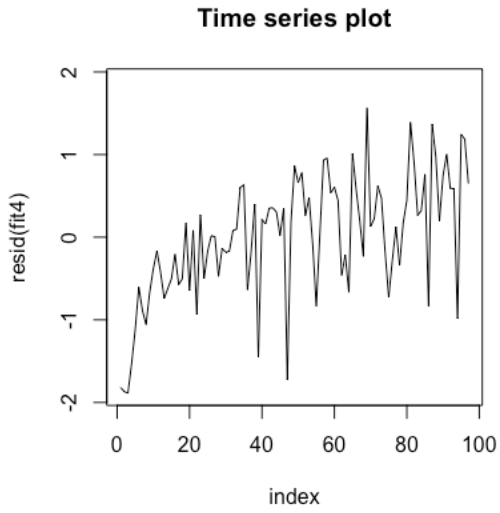
All three model selections choose the model as fit4 which is chosen manually by us. For there exist a clear boundary in this question and we can clearly find out where to stop.

5) Verify the model assumptions:

```
> # 5) Verify the model assumptions
> # Residual plot
> plot(fitted(fit4), resid(fit4), main="Residual plot")
> abline(h=0)
> # QQ Plot
> qqnorm(resid(fit4), main="QQ plot")
> qqline(resid(fit4))
> # Time series plot
> maxabs = max(abs(resid(fit4)))
> plot(index, resid(fit4), type='l', main="Time series plot",
+      ylim=maxabs*c(-1, 1))
>
```







Assumptions:

a) Errors have mean zero and constant variance.

Proved by residual plot. The horizontal line nearly split all data points in half. The points are scattered around zero and have less pattern. Hence, errors have nearly mean zero and constant variance.

b) Errors are normally distributed.

Proved by QQ plot. The QQ plot nearly fits the QQ line which means that the errors are nearly normal distributed.

c) Errors are independent.

Proved by time series plot. In all, the time series plot shows a positive trend while index increasing. But this trend is quite tiny. We can roughly announced that errors are independent.

6) Predict the PSA level

From the QQ plot of residuals above, we see that residuals hold normalization very well. Thus, we can conclude that our model assumptions hold and we can choose model fit4 as our 'reasonably good' model.

PSA = exp(-0.65013 + 0.06488*(cancervol) + 0.09136*(benpros) +

0.68421*(factor(vesinv)1) + 0.33376*(gleason))

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.

```
> # 6) Predict the PSA level for a "common average" patient
> summary(fit4)

Call:
lm(formula = logpsa ~ cancervol + benpros + factor(vesinv) +
    gleason)

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 ***
benpros          0.09136    0.02606   3.506 0.000705 ***
factor(vesinv)1  0.68421    0.23640   2.894 0.004746 **
gleason          0.33376    0.12331   2.707 0.008100 **
---
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

> # Get mode by names(sort(-table(vesinv)))[1]. Return a string here.
> psa.predict = exp(-0.65013 + 0.06488*mean(cancervol) + 0.09136*mean(benpros) +
+                   0.68421*(ifelse(names(sort(-table(vesinv)))[1]=='1', 1, 0)) +
+                   0.33376*mean(gleason))
> print(mean(cancervol))
[1] 6.998682
> print(mean(benpros))
[1] 2.534725
> print(ifelse(names(sort(-table(vesinv)))[1]=='1', 1, 0))  # mode
[1] 0
> print(mean(gleason))
[1] 6.876289
> print(psa.predict)
[1] 10.28357
>
```

The mean of cancerval = 6.998682

The mean of benpros = 2.534725

The mode of vesinv = 0

The mean of gleason = 6.876289

Thus, the PSA level for a patient whose quantitative predictors are at the sample means of the variables and qualitative predictors is 10.28357.