Karthik Ragunath Ananda Kumar - Mini Project 6 - Solo Group 10

Solution 1:

Data Extraction

==========

Utility function to read csv

```
read_csv_func = function(x){
  df = read.csv(x, header=TRUE) # Read CSV
  return (df)
}
```

```
prostate_cancer_df =
  read_csv_func("/Users/karthik_ragunath/Desktop/Stats/prostate_cancer.csv")
```

Printing all columns in dataset

```
names(prostate_cancer_df)
```

```
## [1] "subject"   "psa"       "cancervol" "weight"    "age"       "benpros"
## [7] "vesinv"    "capspen"   "gleason"
```

Getting class type of columns

```
sapply(prostate_cancer_df, class)
```

```
##   subject       psa cancervol    weight       age   benpros    vesinv   capspen
## "integer" "numeric" "numeric" "numeric" "integer" "numeric" "integer" "numeric"
##   gleason
## "integer"
```

Encoding Categorical Variable

```
prostate_cancer_df$vesinv = factor(prostate_cancer_df$vesinv)
```

Encoded Categorical Variable

```
names(prostate_cancer_df)
```

```
## [1] "subject"   "psa"       "cancervol" "weight"    "age"       "benpros"
## [7] "vesinv"    "capspen"   "gleason"
```

Getting class type of columns

```
sapply(prostate_cancer_df, class)
```

```
##   subject       psa cancervol    weight       age   benpros    vesinv   capspen
## "integer" "numeric" "numeric" "numeric" "integer" "numeric"  "factor" "numeric"
##   gleason
## "integer"
```

Custom utility function to extract specific column from dataframe

```
extract_column_func = function(df, column_name, keep_char=FALSE){
  column_names_list = names(df)
  column_index = match(column_name, column_names_list)
  column_data = df[, column_index]
  return (column_data)
}
```

Getting Induvidual Columns

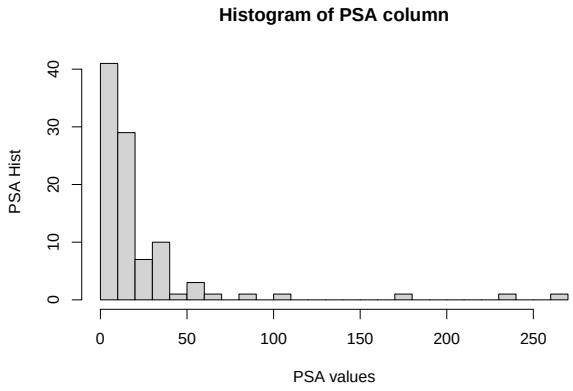
```
psa = extract_column_func(prostate_cancer_df, "psa")
cancervol = extract_column_func(prostate_cancer_df, "cancervol")
weight = extract_column_func(prostate_cancer_df, "weight")
age = extract_column_func(prostate_cancer_df, "age")
benpros = extract_column_func(prostate_cancer_df, "benpros")
vesinv = extract_column_func(prostate_cancer_df, "vesinv")
capspen = extract_column_func(prostate_cancer_df, "capspen")
gleason = extract_column_func(prostate_cancer_df, "gleason")
```

Exploratory Data Analysis

==========

Histogram of PSA column

```
hist(psa, xlab="PSA values", main="Histogram of PSA column", ylab="PSA Hist",
     breaks=30)
```



From historgram of PSA column, we can see that the PSA distribution represents
that of a random variable following exponential distribution
We can also see that most of the people have very low PSA values.

Getting Number of Samples

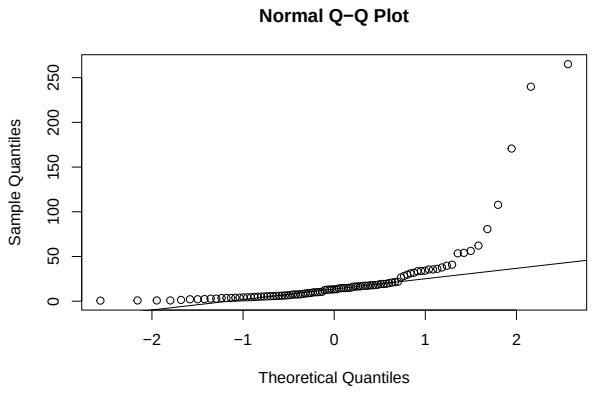
```
nrow(prostate_cancer_df)
```

```
## [1] 97
```

Since Number of Samples is quite large, checking if the distribution

approximates to Normal Curve

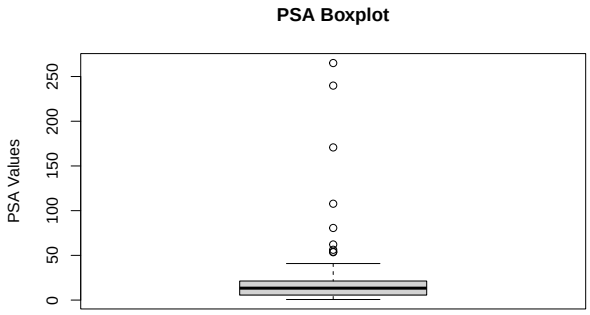
```
qqnorm(psa)
qqline(psa)
```



The curve deviates from Normal Distribution line and hence doesn't approximate to Normal Curve

Plotting Box-Plot of PSA values

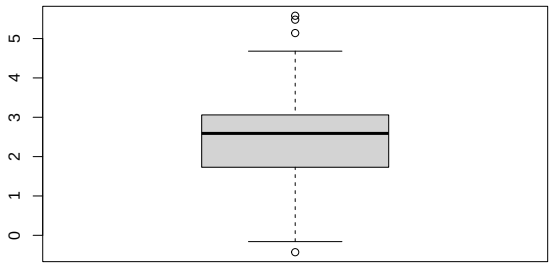
```
boxplot(psa, main="PSA Boxplot", ylab="PSA Values")
```



From the box plot, it is clear that we need to perform some transformation since there are too many outliers with our current dataset.

Checking box plot again after performing LOG TRANSFORMATION

```
boxplot(log(psa))
```

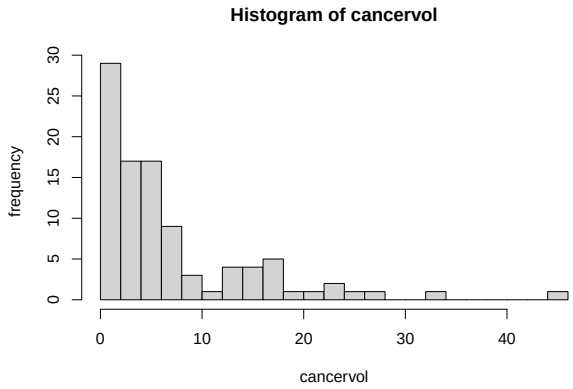


Now, we could see that the number of outliers has reduced drastically and also the distribution is more symmetric. Hence, we would use log transformed output.

Lets perform exploratory analysis on predictor columns to understand which columns could be used for model building

Histogram of cancervol

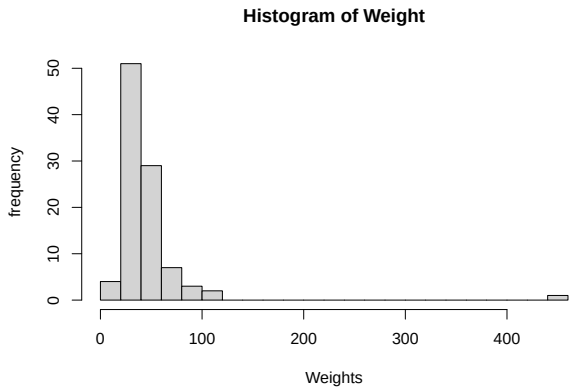
```
hist(cancervol, main="Histogram of cancervol", xlab="cancervol",
     ylab="frequency", breaks=30)
```



Histogram of cancervol closely resembles histogram of psa

Histogram of Weight column

```
hist(weight, main="Histogram of Weight", xlab="Weights", ylab="frequency",
     breaks=30)
```



Histogram of weight column also closely resembles that of histogram of PSA column

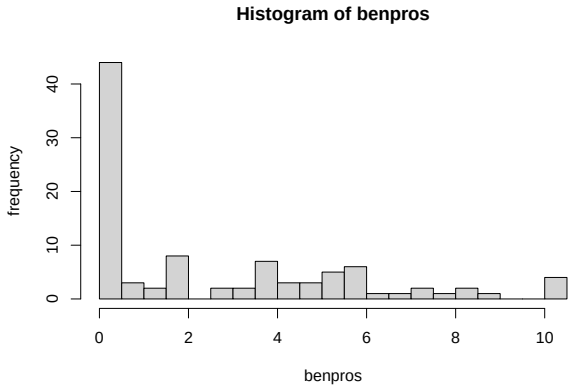
Histogram of Age

```
hist(age, main="Histogram of Age", xlab="Age", ylab="frequency", breaks=30)
```

Histogram of Age resembles normal distribution

Histogram of benpros

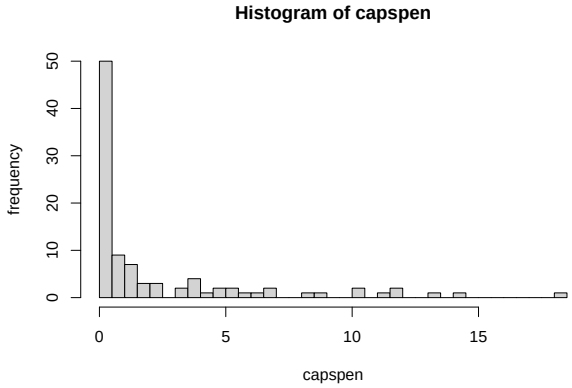
```
hist(benpros, main="Histogram of benpros", xlab="benpros", ylab="frequency",
     breaks=30)
```



Histogram of benpros column also appears to be exponentially and somewhat resembles that of PSA column.

Histogram of capspen

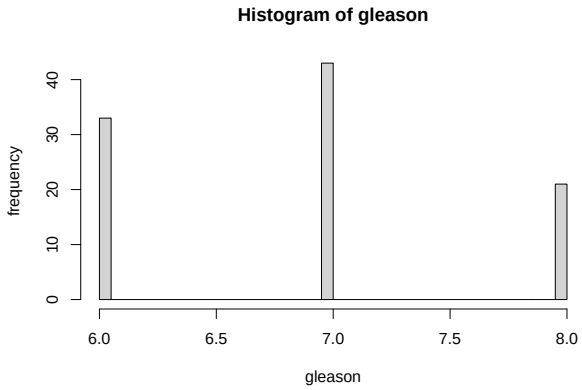
```
hist(capspen, main="Histogram of capspen", xlab="capspen", ylab="frequency",
     breaks=30)
```



Histogram of capspen column also appears to be exponentially and somewhat resembles that of PSA column.

Histogram of gleason

```
hist(gleason, main="Histogram of gleason", xlab="gleason", ylab="frequency",
     breaks=30)
```



Gleason appears to have only 3 possible values from its histogram.

Computing correlation between column values

```
prostate_cancer_cor = cor(prostate_cancer_df[c(2,3,4,5,6,8,9)])
round(prostate_cancer_cor, 4)
```

```
##               psa cancervol  weight    age benpros capspen gleason
## psa        1.0000    0.6242  0.0262 0.0172 -0.0165  0.5508  0.4296
## cancervol  0.6242    1.0000  0.0051 0.0391 -0.1332  0.6929  0.4814
## weight     0.0262    0.0051  1.0000 0.1643  0.3218  0.0016 -0.0242
## age        0.0172    0.0391  0.1643 1.0000  0.3663  0.0996  0.2259
```

```
## benpros   -0.0165   -0.1332  0.3218 0.3663  1.0000 -0.0830  0.0268
## capspen    0.5508    0.6929  0.0016 0.0996 -0.0830  1.0000  0.4616
## gleason    0.4296    0.4814 -0.0242 0.2259  0.0268  0.4616  1.0000
```

The most important factor to consider here is the first row which exhibits correlation between psa (our response variable) with the predictor variables.

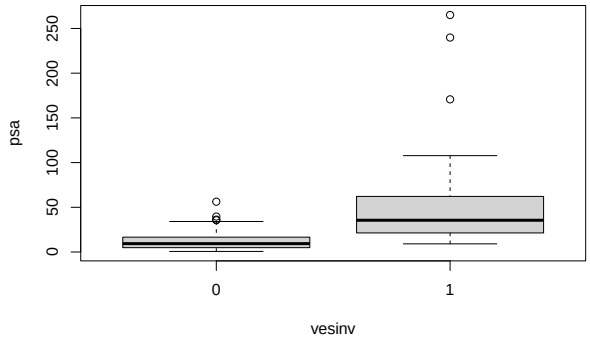
From the correlation table psa exhibits significant linear trend with the following columns:
1. cancervol
2. capsen
3. gleason

Also, the above three predictor variables also displays strong correlation among themselves. Therefore we must avoid overfitting of data which could be caused due to strong correlation between predictor variables themselves

To understand importance of categorical varible column (vesinv), lets do

box plot of psa and vesinv

```
boxplot(psa~vesinv)
```



This boxplot indicates that psa values varies significantly with distinct vesinv values.
Therfore, the categorical variable could be useful in our model building.

Lets make sure that our correlation analysis still holds with log(psa)

values as response variable

```
prostate_cancer_log_transformed_df = prostate_cancer_df
prostate_cancer_log_transformed_df$psa = log(psa)
```

```
prostate_cancer_log_transformed_cor = cor(
  prostate_cancer_log_transformed_df[c(2,3,4,5,6,8,9)])
round(prostate_cancer_log_transformed_cor, 4)
```

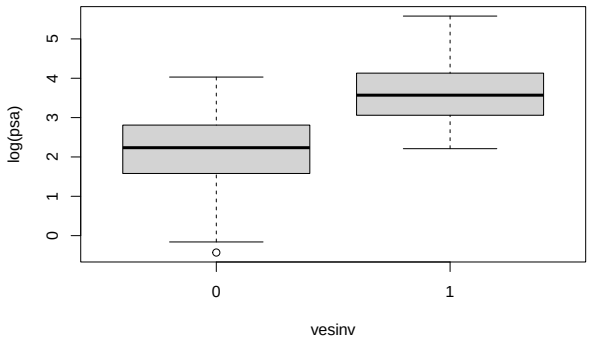
```
##              psa cancervol  weight    age benpros capspen gleason
## psa       1.0000    0.6571  0.1217 0.1699  0.1574  0.5180  0.5390
## cancervol 0.6571    1.0000  0.0051 0.0391 -0.1332  0.6929  0.4814
## weight    0.1217    0.0051  1.0000 0.1643  0.3218  0.0016 -0.0242
## age       0.1699    0.0391  0.1643 1.0000  0.3663  0.0996  0.2259
## benpros   0.1574   -0.1332  0.3218 0.3663  1.0000 -0.0830  0.0268
## capspen   0.5180    0.6929  0.0016 0.0996 -0.0830  1.0000  0.4616
## gleason   0.5390    0.4814 -0.0242 0.2259  0.0268  0.4616  1.0000
```

From correlation table, we could see that log(psa) still exhibits strong correlation with cancervol, carspen and gleason predictor variables as before.

Lets also understand the importance of categorical varible column (vesinv)

for predicting log(psa)

```
boxplot(log(psa)~vesinv)
```



This boxplot indicates that log(psa) values varies significantly with distinct vesinv values.
Therfore, the categorical variable could be useful in our model building.

Model Fitting

=======

Initializing the target Variable as log(psa)

```
log_psa = log(psa)
```

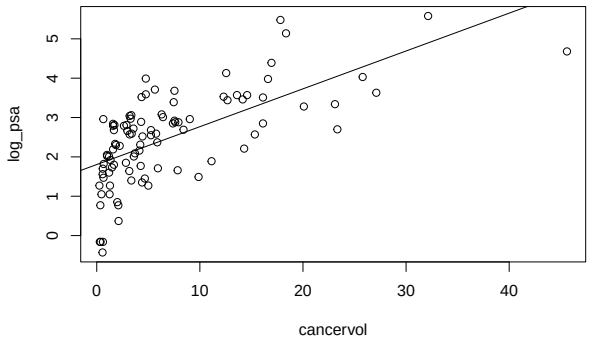
Fitting models with induvidual variables

We are considering only out three variables of interest - cancervol,

capspen, gleason

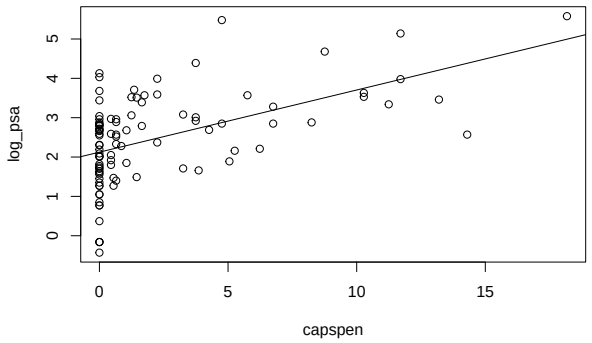
Relation between log(psa) and cancervol

```
plot(cancervol, log_psa)
fit1 = lm(log_psa~cancervol)
abline(fit1)
```



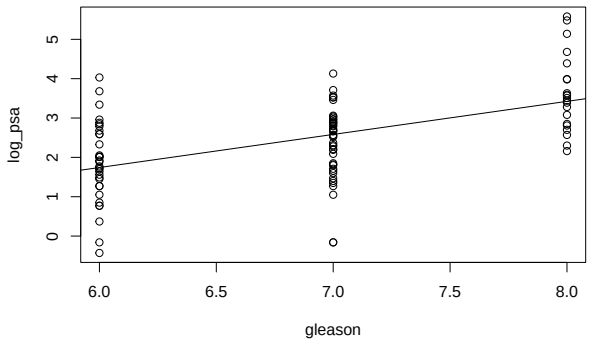
Relation between log(psa) and capspen

```
plot(capspen, log_psa)
fit2 = lm(log_psa~capspen)
abline(fit2)
```



Relation between log(psa) and gleason

```
plot(gleason, log_psa)
fit3 = lm(log_psa~gleason)
abline(fit3)
```



From the plot above, a significant positive trend seems to exist between cancervol and log(psa), between log(psa) and capspen and also between log(psa) and gleason which correlates with our observation from correlation table.

Lets fit the model with all our significant predictors

```
fit_4 = lm(log_psa~cancervol+capspen+gleason+vesinv)
fit_4
```

```
##
## Call:
## lm(formula = log_psa ~ cancervol + capspen + gleason + vesinv)
##
## Coefficients:
## (Intercept)    cancervol      capspen      gleason      vesinv1
##    -0.79386      0.06452     -0.02348      0.39566      0.70675
```

Summary of the above fit (fit_4)

log_psa~cancervol+capspen+gleason+vesinv

```
summary(fit_4)
```

```
##
## Call:
## lm(formula = log_psa ~ cancervol + capspen + gleason + vesinv)
##
```

```
## 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 ***
## capspen     -0.02348    0.03455  -0.680  0.49852
## gleason      0.39566    0.13100   3.020  0.00327 **
## vesinv1      0.70675    0.28024   2.522  0.01339 *
## ---
## 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
```

Reduce reduce our model by removing capspen variable (vesinv)

```
fit_5 = lm(log_psa~cancervol+gleason+vesinv)
fit_5
```

```
##
## Call:
## lm(formula = log_psa ~ cancervol + gleason + vesinv)
##
## Coefficients:
## (Intercept)    cancervol      gleason      vesinv1
##    -0.72120      0.05981      0.38491      0.62117
```

Summary of the above model (fit_5)

```
summary(fit_5)
```

```
##
## Call:
## lm(formula = log_psa ~ cancervol + gleason + vesinv)
##
## 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 ***
## gleason      0.38491    0.12966   2.969   0.0038 **
## vesinv1      0.62117    0.24962   2.488   0.0146 *
```

```
## ---
## 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
```

Lets understand the significance of capspen variable by doing anova test

(Since variables in fit_5 is a subset of variables in fit_4)

```
anova(fit_4, fit_5)
```

```
## Analysis of Variance Table
##
## Model 1: log_psa ~ cancervol + capspen + gleason + vesinv
## Model 2: log_psa ~ cancervol + gleason + vesinv
##   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
```

```
From the anova test, capspen variable is not needed since P value is higher
which indicates fit_4 is not significantly different from fit_5
```

Lets reduce the model parameters by removing gleason variable too

```
fit_6 = lm(log_psa~cancervol+vesinv)
fit_6
```

```
##
## Call:
## lm(formula = log_psa ~ cancervol + vesinv)
##
## Coefficients:
## (Intercept)    cancervol      vesinv1
##     1.80346      0.07249      0.77552
```

```
summary(fit_6)
```

```
##
## Call:
## lm(formula = log_psa ~ cancervol + 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 ***
## vesinv1      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
```

Lets check whether gleason is statistically significant or not by

performing anova test

```
anova(fit_5, fit_6)
```

```
## Analysis of Variance Table
##
## Model 1: log_psa ~ cancervol + gleason + vesinv
## Model 2: log_psa ~ cancervol + vesinv
##   Res.Df    RSS Df Sum of Sq      F   Pr(>F)
## 1     93 60.340
## 2     94 66.058 -1   -5.7179 8.8127 0.003804 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

From the anova summary, it is clear that gleason is a significant variable since p-value is very small. Hence, it can't be removed when building model.

Lets compare our model with stepwise model selection process both in

forward, backward and two-way selection direction

```
forward_step_selection = step(lm(formula=log_psa~1), scope=list(upper=~cancervol
                                                                +weight
                                                                +age
                                                                +benpros
                                                                +vesinv
                                                                +capspen
                                                                +gleason),
                              direction="forward")
```

```
## Start:  AIC=28.72
## log_psa ~ 1
##
##             Df Sum of Sq     RSS      AIC
## + cancervol  1    55.164  72.605 -24.0986
## + vesinv     1    40.984  86.785  -6.7944
```

```
## + gleason    1    37.122  90.647  -2.5707
## + capspen    1    34.286  93.482   0.4169
## + age        1     3.688 124.080  27.8831
## + benpros    1     3.166 124.603  28.2911
## <none>                   127.769  28.7246
## + weight     1     1.893 125.876  29.2767
##
## Step:  AIC=-24.1
## log_psa ~ cancervol
##
##           Df Sum of Sq    RSS     AIC
## + gleason  1    8.2468 64.358 -33.794
## + benpros  1    7.8034 64.802 -33.128
## + vesinv   1    6.5468 66.058 -31.265
## + age      1    2.6615 69.944 -25.721
## + weight   1    1.7901 70.815 -24.520
## <none>                 72.605 -24.099
## + capspen  1    0.9673 71.638 -23.400
##
## Step:  AIC=-33.79
## log_psa ~ cancervol + gleason
##
##           Df Sum of Sq    RSS     AIC
## + benpros  1    6.2827 58.075 -41.758
## + vesinv   1    4.0178 60.340 -38.047
## + weight   1    2.0334 62.325 -34.908
## <none>                 64.358 -33.794
## + age      1    0.9611 63.397 -33.253
## + capspen  1    0.1685 64.190 -32.048
##
## Step:  AIC=-41.76
## log_psa ~ cancervol + gleason + benpros
##
##           Df Sum of Sq    RSS     AIC
## + vesinv   1    4.8466 53.229 -48.211
## <none>                 58.075 -41.758
## + weight   1    0.4006 57.675 -40.429
## + capspen  1    0.1863 57.889 -40.069
## + age      1    0.0059 58.070 -39.768
##
## Step:  AIC=-48.21
## log_psa ~ cancervol + gleason + benpros + 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
```

```
backward_step_selection = step(lm(formula=log_psa~cancervol+weight+age+benpros+
                                    vesinv+capspen+gleason),
                               scope=list(lower=~1),
                              direction="backward")
```

```
## Start:  AIC=-43.59
## log_psa ~ cancervol + weight + age + benpros + vesinv + capspen +
##     gleason
##
##             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
## - 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
## log_psa ~ cancervol + weight + benpros + vesinv + capspen + gleason
##
##             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
## - 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
## log_psa ~ cancervol + benpros + vesinv + capspen + gleason
##
##             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
## - 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
## log_psa ~ cancervol + benpros + vesinv + gleason
##
##             Df Sum of Sq    RSS     AIC
## <none>                   53.229 -48.211
## - gleason    1    4.2389 57.468 -42.778
## - vesinv     1    4.8466 58.075 -41.758
## - benpros    1    7.1115 60.340 -38.047
## - cancervol  1   14.7580 67.987 -26.473
```

```
two_way_selection = step(lm(formula=log_psa~1), scope=list(upper=~cancervol
                                                           +weight
                                                           +age
                                                           +benpros
                                                           +vesinv
                                                           +capspen
                                                           +gleason),
```

```
                        direction="both")
```

```
## Start:  AIC=28.72
## log_psa ~ 1
##
##             Df Sum of Sq     RSS      AIC
## + cancervol  1    55.164  72.605 -24.0986
## + vesinv     1    40.984  86.785  -6.7944
## + gleason    1    37.122  90.647  -2.5707
## + capspen    1    34.286  93.482   0.4169
## + age        1     3.688 124.080  27.8831
## + benpros    1     3.166 124.603  28.2911
## <none>                   127.769  28.7246
## + weight     1     1.893 125.876  29.2767
##
## Step:  AIC=-24.1
## log_psa ~ cancervol
##
##             Df Sum of Sq     RSS     AIC
## + gleason    1     8.247  64.358 -33.794
## + benpros    1     7.803  64.802 -33.128
## + vesinv     1     6.547  66.058 -31.265
## + age        1     2.662  69.944 -25.721
## + weight     1     1.790  70.815 -24.520
## <none>                    72.605 -24.099
## + capspen    1     0.967  71.638 -23.400
## - cancervol  1    55.164 127.769  28.725
##
## Step:  AIC=-33.79
## log_psa ~ cancervol + gleason
##
##             Df Sum of Sq    RSS     AIC
## + benpros    1    6.2827 58.075 -41.758
## + vesinv     1    4.0178 60.340 -38.047
## + weight     1    2.0334 62.325 -34.908
## <none>                   64.358 -33.794
## + age        1    0.9611 63.397 -33.253
## + capspen    1    0.1685 64.190 -32.048
## - gleason    1    8.2468 72.605 -24.099
## - cancervol  1   26.2887 90.647  -2.571
##
## Step:  AIC=-41.76
## log_psa ~ cancervol + gleason + benpros
##
##             Df Sum of Sq    RSS     AIC
## + vesinv     1    4.8466 53.229 -48.211
## <none>                   58.075 -41.758
## + weight     1    0.4006 57.675 -40.429
## + capspen    1    0.1863 57.889 -40.069
## + age        1    0.0059 58.070 -39.768
## - benpros    1    6.2827 64.358 -33.794
## - gleason    1    6.7262 64.802 -33.128
## - cancervol  1   29.9589 88.034  -3.407
```

```
##
## Step:  AIC=-48.21
## log_psa ~ cancervol + gleason + benpros + vesinv
##
##             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
## - vesinv     1    4.8466 58.075 -41.758
## - benpros    1    7.1115 60.340 -38.047
## - cancervol  1   14.7580 67.987 -26.473
```

Inference

======

From automated step wise model selection (performed in forward, backward and both directions),
we could infer that the features of interest are cancervol, gleason, benpros, vesinv

Fitting a model with features from step wise model selection process

```
fit_7 = lm(log_psa~cancervol + gleason + benpros + vesinv)
fit_7
```

```
##
## Call:
## lm(formula = log_psa ~ cancervol + gleason + benpros + vesinv)
##
## Coefficients:
## (Intercept)    cancervol      gleason      benpros      vesinv1
##    -0.65013      0.06488      0.33376      0.09136      0.68421
```

Summary of the above fitted model whose features are obtained from step-wise model selection process

```
summary(fit_7)
```

```
##
## Call:
## lm(formula = log_psa ~ cancervol + gleason + benpros + vesinv)
```

```
##
## 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 ***
## gleason      0.33376    0.12331   2.707 0.008100 **
## benpros      0.09136    0.02606   3.506 0.000705 ***
## vesinv1      0.68421    0.23640   2.894 0.004746 **
## ---
## 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
```

Let's do an anova test between the model we obtained by performing step-wise feature selection process and the model whose features we selected by manually performing Exploratory Data Analysis

```
anova(fit_7, fit_5)
```

```
## Analysis of Variance Table
##
## Model 1: log_psa ~ cancervol + gleason + benpros + vesinv
## Model 2: log_psa ~ cancervol + gleason + vesinv
##   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
```

```
fit_7$coefficients
```

```
## (Intercept)   cancervol     gleason     benpros     vesinv1
## -0.65013037  0.06487865  0.33375914  0.09136387  0.68420905
```

```
forward_step_selection$coefficients
```

```
## (Intercept)   cancervol     gleason     benpros     vesinv1
## -0.65013037  0.06487865  0.33375914  0.09136387  0.68420905
```

```
backward_step_selection$coefficients
```

```
## (Intercept)   cancervol     benpros     vesinv1     gleason
## -0.65013037  0.06487865  0.09136387  0.68420905  0.33375914
```

```
two_way_selection$coefficients
```

```
## (Intercept)   cancervol     gleason     benpros     vesinv1
## -0.65013037  0.06487865  0.33375914  0.09136387  0.68420905
```

Optimum Model Selection Reasoning

```
### The difference between the features we selected for fitting the model by
using stepwise model selection process (in forward, backward and two-way
direction) and the one we obtained through manually selecting the features by
performing various exploratory data analysis manually is that the model fitted
using stepwise modelselection process has an additional variable of benpros
involved.

Therefore, to check whether benpros is significant, we performed anova test.
We could infer from P value of anova test that benpros feature is indeed
significant and hence we select the model fit_7 fitted with cancervol, gleason,
benpros and vesinvpredictor feature variables for predicting log(psa) value as
our optimum model.

The coefficient of the features fitted in our model fit_7, namely,
(Intercept) -0.65013037
cancervol    0.06487865
gleason      0.33375914
benpros      0.09136387
vesinv1      0.68420905

These coefficients mateches with coefficients of the optimum model obtained
through forward stepwise, backward stepwise and two-way stepwise selection
process too.
```

Prediction

=====

Predicting for a patient whose quantitative predictors are sample means of

the quantitative predictors and most frequent value of qualitative predictor

```
The quantitative predictors are cancervol, gleason, benpros
The qualitative predictor is vesinv
```

Calculating Sample Means

```
sample_mean_cancervol = mean(cancervol)
sample_mean_gleason = mean(gleason)
sample_mean_benpros = mean(benpros)

rows_with_vesinv_0 = sum(with(prostate_cancer_log_transformed_df , vesinv==0))
rows_with_vesinv_1 = sum(with(prostate_cancer_log_transformed_df , vesinv==1))
```

```
rows_with_vesinv_0
```

```
## [1] 76
```

```
rows_with_vesinv_1
```

```
## [1] 21
```

```
if(rows_with_vesinv_1 >= rows_with_vesinv_0) {
  max_repeated_vesinv_val = 1
} else {
  max_repeated_vesinv_val = 0
}
```

```
sample_mean_cancervol
```

```
## [1] 6.998682
```

```
sample_mean_gleason
```

```
## [1] 6.876289
```

```
sample_mean_benpros
```

```
## [1] 2.534725
```

```
max_repeated_vesinv_val
```

```
## [1] 0
```

Therefore, the linear regression equation of the model is given by:

```
predicted_value = -0.65013037 + 0.06487865 * cancervol_test_val +
  0.33375914 * gleason_test_val + 0.09136387 * benpros_test_val +
  0.68420905 * vesinv_test_val
```

Assigning sample mean values of quantitative feature values and

most frequent value of qualitative feature column into test variables

```
cancervol_test_val = sample_mean_cancervol
gleason_test_val = sample_mean_gleason
benpros_test_val = sample_mean_benpros
vesinv_test_val = max_repeated_vesinv_val
```

Performing Prediction for a test sample whose quantitative feature values are sample means of the correspoding features in the given test data and qualitative feature value is the most frequent value of the corresponding qualitative feature in the given test dataset

```
log_transformed_psa_predicted_value = -0.65013037 + 0.06487865 *
  sample_mean_cancervol + 0.33375914 * sample_mean_gleason +
  0.09136387 * sample_mean_benpros + 0.68420905 * max_repeated_vesinv_val
log_transformed_psa_predicted_value
```

```
## [1] 2.330541
```

Since we model our prediction on the data on which we performed log transformation we must compute exponential of the predicted value for getting the actual PSA prediction

```
actual_psa_prediction = exp(log_transformed_psa_predicted_value)
actual_psa_prediction
```

```
## [1] 10.2835
```

==========