FIT5197 2018 S1 Assignment2

KEWEN DENG, 29330440

18 May 2018

Task A: Linear Regression

```
# initialization
rawdata = read.csv('auto_mpg_train.csv')
testdata = read.csv('auto_mpg_test.csv')
modifieddata = read.csv('auto_mpg_train.modified.csv')
```

A.1

After examine the given dataset, I find "horsepower" is the only variable contains missing values listed as "?". The observations with "?" as listed:

```
# show lines with '?' in horsepower
replace_rawdata = rawdata[rawdata$horsepower == '?',]
print(replace_rawdata)
```

```
##      mpg cylinders displacement horsepower weight acceleration model.year
## 33  25.0         4           98          ?   2046         19.0         71
## 127 21.0         6          200          ?   2875         17.0         74
## 281 40.9         4           85          ?   1835         17.3         80
## 287 23.6         4          140          ?   2905         14.3         80
## 305 34.5         4          100          ?   2320         15.8         81
## 325 23.0         4          151          ?   3035         20.5         82
##     origin             car.name
## 33       1           ford pinto
## 127      1        ford maverick
## 281      2 renault lecar deluxe
## 287      1   ford mustang cobra
## 305      2         renault 18i
## 325      1       amc concord dl
```

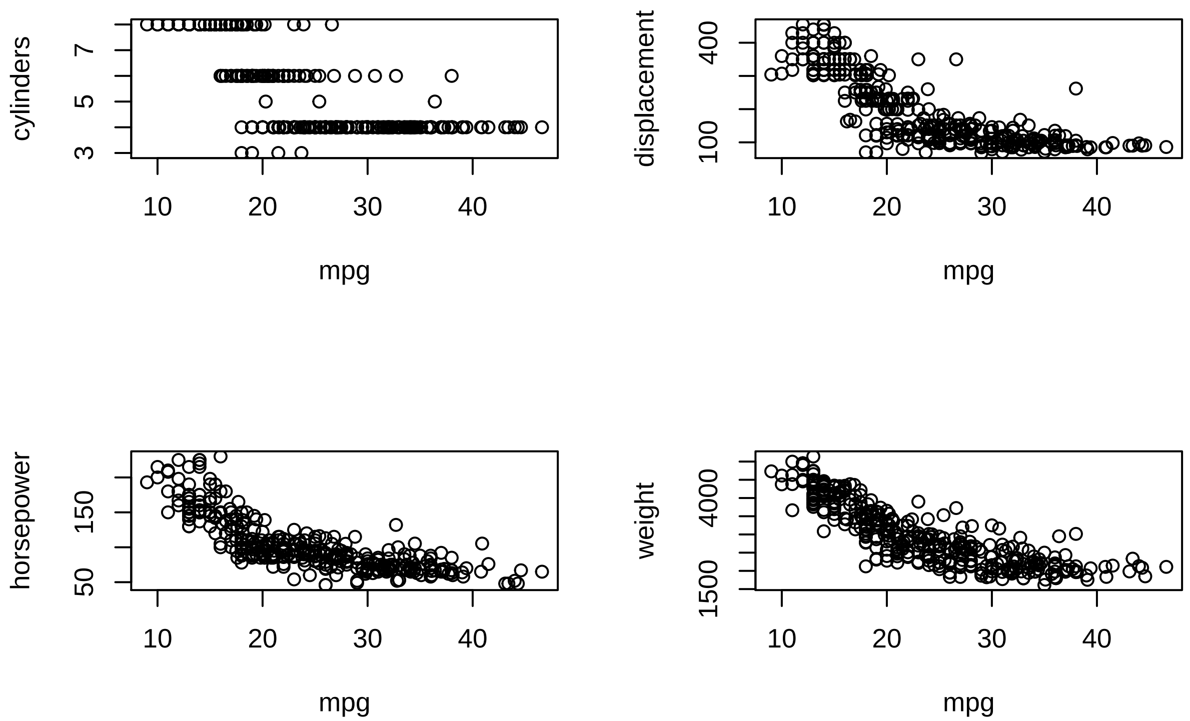
According to the explaination and Data Set Information, "horsepower" is a continuous variable. Thus, I replaced all the "?" with average value calculated, which is 105.3040936.

A.2

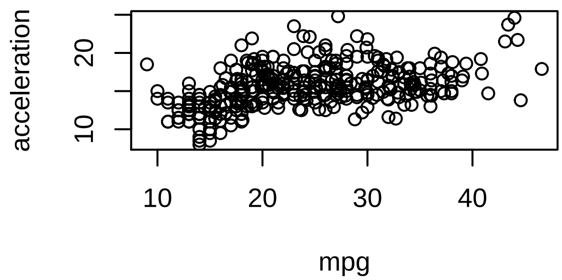
Pair plots mpg vs. the other bariables are as following:

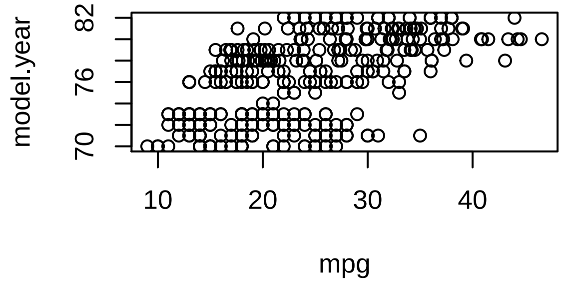
```
# initialization
rawdata = modifieddata
# pair plots
par(mfrow=c(2,2))
plot(rawdata$mpg, rawdata$cylinders,
     xlab = 'mpg',
     ylab = 'cylinders')
plot(rawdata$mpg, rawdata$displacement,
```

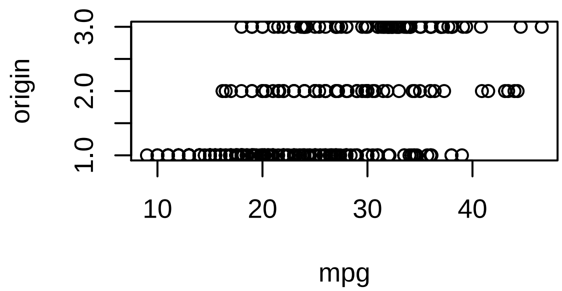
```
     xlab = 'mpg',
     ylab = 'displacement')
plot(rawdata$mpg, rawdata$horsepower,
     xlab = 'mpg',
     ylab = 'horsepower')
plot(rawdata$mpg, rawdata$weight,
     xlab = 'mpg',
     ylab = 'weight')
```

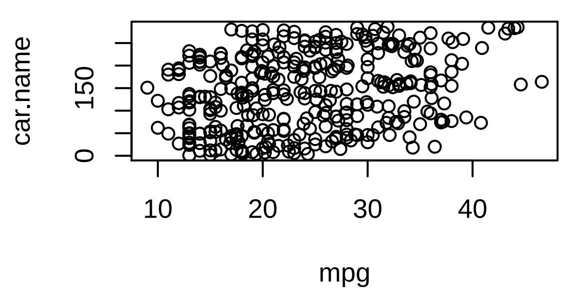



```
par(mfrow=c(2,2))
plot(rawdata$mpg, rawdata$acceleration,
     xlab = 'mpg',
     ylab = 'acceleration')
plot(rawdata$mpg, rawdata$model.year,
     xlab = 'mpg',
     ylab = 'model.year')
plot(rawdata$mpg, rawdata$origin,
     xlab = 'mpg',
     ylab = 'origin')
plot(rawdata$mpg, rawdata$car.name,
     xlab = 'mpg',
     ylab = 'car.name')
```









A.3

Based on the pair plots, it is clear that displacement, weight, horsepower and acceleration are more likely to be used in a linear regression model to predict mpg.

$$mpg = \beta_0 + \beta_1 displacement + \beta_2 weight + \beta_3 horsepower + \beta_4 acceleration$$

A.4

Then I used lm() routine in R and print the summary of the model to get the R diagnostics.

```
model <- lm(formula = mpg ~ displacement + weight + horsepower + acceleration, data = rawdata)
summary(model)
```

```
##
## Call:
## lm(formula = mpg ~ displacement + weight + horsepower + acceleration,
##     data = rawdata)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -11.4891  -2.8606  -0.4506   2.5857  15.7218
##
## Coefficients:
```

```
##                Estimate Std. Error t value Pr(>|t|)
## (Intercept)  46.1669125  2.5983227  17.768  < 2e-16 ***
## displacement -0.0009364  0.0072054  -0.130  0.89667
## weight       -0.0057107  0.0008810  -6.482 3.15e-10 ***
## horsepower   -0.0512626  0.0171113  -2.996  0.00294 **
## acceleration  0.0096859  0.1313547   0.074  0.94126
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 4.357 on 343 degrees of freedom
## Multiple R-squared:  0.7069, Adjusted R-squared:  0.7034
## F-statistic: 206.8 on 4 and 343 DF,  p-value: < 2.2e-16
```

Based on our multiple variables linear regression model, we need to use Adjusted R-squared to represent R^2 value according to the number of modeled variables because as the number of variable increases the R^2 also grows regardless of the model being improved or not. Generally speaking, R^2 evaluated the goodness of the fit which the higher is the better. The value shows the amount of variability in the estimated response variable that is explained by the model. In our result, almost 70.34% of the cause for a mpg can be explained by displacement, weight, horsepower and acceleration, then the model is appropriate.

The t-value measures the size of the difference relative to the variation in the data. These values are not informative per se, unless we use them to calculate other statistics. In our result, the t-value of intercept, displacement, weight, horsepower and acceleration are 17.768, -0.130, -6.482, -2.996 and 0.074.

The standard error comes from the estimated coefficients which measures their variablility. Obviously, the lower the error, the better the fit. In our result, the standard error of intercept, displacement, weight, horsepower and acceleration are 2.5983227, 0.0072054, 0.0008810, 0.0171113 and 0.1313547.

The p-value is calculated by standard error and t_value. As a rule of thumb, the lesser the p-value, the more descriptive the predictor variable is. Also, we can interpret the p-values as the probability the variable is irrelevant. In our result, the p-value of intercept, displacement, weight, horsepower and acceleration are less than $2e^{-16}$, 0.89667, $3.15e^{-10}$, 0.00294 and 0.94126.

Consequently, weight and horsepower are significant in predictors. Also, the standard error and t-value of weight are lowest, so weight should be the most influential predictor.

A.5

The MSE of the test data set is:

```r
# initialization
library(Metrics)

# calculate MSE of test data set
testmodel <- predict(model, newdata = testdata)
mse.result <- mse(testdata$mpg, testmodel)
print(mse.result)
```

```
## [1] 14.49873
```

A.6

In order to select the best linear regression model for this question, Backwards selection with step() routine has been chosen. First, start with the full model.

```r
model.plus <- lm(formula = mpg ~ cylinders + displacement + horsepower + weight + acceleration + model.y
summary(model.plus)
```

```
##
## Call:
## lm(formula = mpg ~ cylinders + displacement + horsepower + weight +
##     acceleration + model.year + origin, data = rawdata)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -9.5312 -2.1877 -0.1035  1.8899 12.8438
##
## Coefficients:
##                Estimate Std. Error t value Pr(>|t|)
## (Intercept)  -1.580e+01  4.884e+00  -3.235  0.00134 **
## cylinders    -5.443e-01  3.477e-01  -1.565  0.11840
## displacement  2.690e-02  8.108e-03   3.318  0.00100 **
## horsepower   -1.953e-02  1.413e-02  -1.382  0.16784
## weight       -7.175e-03  7.073e-04 -10.144  < 2e-16 ***
## acceleration  1.292e-01  1.032e-01   1.251  0.21161
## model.year    7.403e-01  5.329e-02  13.891  < 2e-16 ***
## origin        1.431e+00  2.945e-01   4.859  1.8e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3.41 on 340 degrees of freedom
## Multiple R-squared:  0.8219, Adjusted R-squared:  0.8183
## F-statistic: 224.2 on 7 and 340 DF,  p-value: < 2.2e-16
```

Second, find the predictor that reduces info criterion by most and end with no predictor improves model.

```r
step <- step(model.plus)
```

```
## Start:  AIC=861.8
## mpg ~ cylinders + displacement + horsepower + weight + acceleration +
##     model.year + origin
##
##                Df Sum of Sq    RSS     AIC
## - acceleration  1     18.22 3972.9  861.40
## - horsepower    1     22.22 3976.9  861.75
## <none>                      3954.7  861.80
## - cylinders     1     28.51 3983.2  862.30
## - displacement  1    128.06 4082.7  870.89
## - origin        1    274.61 4229.3  883.16
## - weight        1   1196.95 5151.6  951.81
## - model.year    1   2244.51 6199.2 1016.23
##
## Step:  AIC=861.4
## mpg ~ cylinders + displacement + horsepower + weight + model.year +
##     origin
##
##                Df Sum of Sq    RSS     AIC
## <none>                      3972.9  861.40
## - cylinders     1     29.57 4002.4  861.98
## - horsepower    1     76.75 4049.6  866.05
```

```
## - displacement  1    116.75 4089.6  869.47
## - origin        1    269.52 4242.4  882.24
## - weight        1   1335.42 5308.3  960.24
## - model.year    1   2227.05 6199.9 1014.27
```

```r
summary(step)
```

```
##
## Call:
## lm(formula = mpg ~ cylinders + displacement + horsepower + weight +
##     model.year + origin, data = rawdata)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -9.8173 -2.2004 -0.1449  1.8325 12.9447
##
## Coefficients:
##                Estimate Std. Error t value Pr(>|t|)
## (Intercept)  -1.317e+01  4.412e+00  -2.985  0.00304 **
## cylinders    -5.542e-01  3.479e-01  -1.593  0.11206
## displacement  2.541e-02  8.025e-03   3.166  0.00169 **
## horsepower   -2.969e-02  1.157e-02  -2.567  0.01069 *
## weight       -6.779e-03  6.332e-04 -10.706  < 2e-16 ***
## model.year    7.355e-01  5.320e-02  13.826  < 2e-16 ***
## origin        1.417e+00  2.946e-01   4.810 2.27e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3.413 on 341 degrees of freedom
## Multiple R-squared:  0.8211, Adjusted R-squared:  0.818
## F-statistic: 260.9 on 6 and 341 DF,  p-value: < 2.2e-16
```

Third, remove this predictor from the model. In this case, the original model are the best fitting model. Thus, no predictor has been removed.

```r
drop1(step)
```

```
## Single term deletions
##
## Model:
## mpg ~ cylinders + displacement + horsepower + weight + model.year +
##     origin
##              Df Sum of Sq    RSS     AIC
## <none>                    3972.9  861.40
## cylinders     1     29.57 4002.4  861.98
## displacement  1    116.75 4089.6  869.47
## horsepower    1     76.75 4049.6  866.05
## weight        1   1335.42 5308.3  960.24
## model.year    1   2227.05 6199.9 1014.27
## origin        1    269.52 4242.4  882.24
```

```r
summary(step)
```

```
##
## Call:
## lm(formula = mpg ~ cylinders + displacement + horsepower + weight +
##     model.year + origin, data = rawdata)
```

```
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -9.8173 -2.2004 -0.1449  1.8325 12.9447
##
## Coefficients:
##                Estimate Std. Error t value Pr(>|t|)
## (Intercept)  -1.317e+01  4.412e+00  -2.985  0.00304 **
## cylinders    -5.542e-01  3.479e-01  -1.593  0.11206
## displacement  2.541e-02  8.025e-03   3.166  0.00169 **
## horsepower   -2.969e-02  1.157e-02  -2.567  0.01069 *
## weight       -6.779e-03  6.332e-04 -10.706  < 2e-16 ***
## model.year    7.355e-01  5.320e-02  13.826  < 2e-16 ***
## origin        1.417e+00  2.946e-01   4.810 2.27e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3.413 on 341 degrees of freedom
## Multiple R-squared:  0.8211, Adjusted R-squared:  0.818
## F-statistic: 260.9 on 6 and 341 DF,  p-value: < 2.2e-16
```

Also, the AIC and BIC of the model can be calculated:

```
cat(' AIC = ', AIC(step), '\n',
    'BIC = ', BIC(step), '\n',
    'The differences is: ', abs(AIC(step) - BIC(step)), '\n')
```

```
##  AIC =  1850.976
##  BIC =  1881.794
##  The differences is:  30.81762
```

Thus, the differences is greater than 3, which means considered significant. Consequently, except 'displacement', 'weight', 'horsepower' and 'acceleration' 4 predictors, 'cylinders', 'model.year' and 'origin' have been added to the model, which comes out 7 predictors in total. The new MSE of the test data set is :

```
# initialization
library(Metrics)

# calculate new MSE of test data set
testmodel <- predict(step, newdata = testdata)
mse(testdata$mpg, testmodel)
```

```
## [1] 8.189157
```

Task B: Logistic Regression

```
# initialization
b.rawdata <- read.csv('adult_income_train.csv')
```

B.1

After examine the given dataset, I find "workclass", "occupation" and "native_country" are the variables which contain "?". According to the explaination and Data Set Information, all the 3 variables are multi-valued discrete, which means we can not simply insert a value into those missing places. Also, it is acceptable for us to delete lines contains missing value when the size are small. Thus, it is necessary to have a look at the percentage of the missing values:

```
replace_brawdata = b.rawdata[b.rawdata$workclass == '?'|b.rawdata$occupation == '?'|b.rawdata$native_co
cat('The percentage of missing values : ', (nrow(replace_brawdata)/nrow(b.rawdata))*100, '%')
```

```
## The percentage of missing values :  8.256923 %
```

The percentage of missing values are 8.256923%, more than 5%, which means the size are too large and we can not delete those lines. According to the "missing informative" method, I decided not to replace the missing values and leave these as separate categorical values.

B.2

With all variables and the given model, the summary is :

```
b.model <- glm(income~., family = binomial, data = b.rawdata)
```

```
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
```

```
summary(b.model)
```

```
##
## Call:
## glm(formula = income ~ ., family = binomial, data = b.rawdata)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -5.1131  -0.5027  -0.1823  -0.0336   3.8667
##
## Coefficients: (2 not defined because of singularities)
##                                           Estimate Std. Error z value
## (Intercept)                             -8.983e+00  3.817e-01 -23.534
## age                                      2.493e-02  1.421e-03  17.547
## workclassFederal-gov                     1.199e+00  1.312e-01   9.142
## workclassLocal-gov                       5.072e-01  1.191e-01   4.258
## workclassNever-worked                   -8.058e+00  8.524e+01  -0.095
## workclassPrivate                         6.753e-01  1.056e-01   6.396
## workclassSelf-emp-inc                    8.304e-01  1.272e-01   6.528
## workclassSelf-emp-not-inc                1.356e-01  1.160e-01   1.169
## workclassState-gov                       3.109e-01  1.295e-01   2.401
## workclassWithout-pay                    -2.029e-01  7.916e-01  -0.256
## fnlwgt                                   7.803e-07  1.473e-07   5.298
## education11th                            4.803e-02  1.834e-01   0.262
## education12th                            4.517e-01  2.274e-01   1.987
## education1st-4th                        -6.381e-01  4.437e-01  -1.438
## education5th-6th                        -3.744e-01  2.842e-01  -1.317
## education7th-8th                        -4.565e-01  2.001e-01  -2.281
## education9th                            -1.979e-01  2.244e-01  -0.882
## educationAssoc-acdm                      1.370e+00  1.525e-01   8.985
## educationAssoc-voc                       1.297e+00  1.473e-01   8.808
## educationBachelors                       1.930e+00  1.368e-01  14.113
## educationDoctorate                       2.871e+00  1.849e-01  15.524
## educationHS-grad                         8.032e-01  1.333e-01   6.027
## educationMasters                         2.265e+00  1.454e-01  15.578
## educationPreschool                      -5.110e+00  3.713e+00  -1.376
```

```
## educationProf-school                  2.782e+00  1.739e-01  16.001
## educationSome-college                 1.168e+00  1.352e-01   8.642
## educational_num                              NA         NA      NA
## marital_statusMarried-AF-spouse       2.484e+00  4.762e-01   5.216
## marital_statusMarried-civ-spouse      2.324e+00  2.318e-01  10.028
## marital_statusMarried-spouse-absent   1.221e-01  1.898e-01   0.643
## marital_statusNever-married          -4.299e-01  7.584e-02  -5.668
## marital_statusSeparated              -1.449e-01  1.446e-01  -1.002
## marital_statusWidowed                 8.315e-02  1.355e-01   0.613
## occupationAdm-clerical                9.467e-02  8.477e-02   1.117
## occupationArmed-Forces                3.553e-01  9.076e-01   0.391
## occupationCraft-repair                1.303e-01  7.292e-02   1.787
## occupationExec-managerial             8.604e-01  7.496e-02  11.477
## occupationFarming-fishing            -8.822e-01  1.231e-01  -7.166
## occupationHandlers-cleaners          -6.438e-01  1.247e-01  -5.161
## occupationMachine-op-inspct          -1.953e-01  9.134e-02  -2.138
## occupationOther-service              -7.800e-01  1.071e-01  -7.280
## occupationPriv-house-serv            -2.508e+00  1.007e+00  -2.490
## occupationProf-specialty              6.173e-01  8.039e-02   7.679
## occupationProtective-serv             5.778e-01  1.130e-01   5.115
## occupationSales                       3.531e-01  7.737e-02   4.564
## occupationTech-support                6.917e-01  1.018e-01   6.797
## occupationTransport-moving                   NA         NA      NA
## relationshipNot-in-family             5.902e-01  2.294e-01   2.573
## relationshipOther-relative           -4.475e-01  2.163e-01  -2.069
## relationshipOwn-child                -5.141e-01  2.249e-01  -2.286
## relationshipUnmarried                 4.186e-01  2.437e-01   1.718
## relationshipWife                      1.207e+00  8.796e-02  13.727
## raceAsian-Pac-Islander                8.455e-01  2.338e-01   3.616
## raceBlack                             4.001e-01  2.033e-01   1.968
## raceOther                             4.883e-01  2.904e-01   1.681
## raceWhite                             6.173e-01  1.934e-01   3.192
## genderMale                            7.743e-01  6.793e-02  11.398
## capital_gain                          3.231e-04  9.041e-06  35.736
## capital_loss                          6.397e-04  3.199e-05  19.999
## hours_per_week                        2.867e-02  1.382e-03  20.745
## native_countryCambodia                9.898e-01  5.506e-01   1.798
## native_countryCanada                  6.812e-01  2.420e-01   2.815
## native_countryChina                  -7.025e-01  3.243e-01  -2.167
## native_countryColumbia               -2.253e+00  7.956e-01  -2.832
## native_countryCuba                    3.318e-01  2.955e-01   1.123
## native_countryDominican-Republic     -1.551e+00  7.610e-01  -2.038
## native_countryEcuador                -4.960e-01  6.298e-01  -0.788
## native_countryEl-Salvador            -6.264e-01  4.477e-01  -1.399
## native_countryEngland                 5.152e-01  2.980e-01   1.729
## native_countryFrance                  8.185e-01  4.584e-01   1.786
## native_countryGermany                 2.507e-01  2.520e-01   0.995
## native_countryGreece                 -2.283e-01  4.052e-01  -0.563
## native_countryGuatemala              -3.188e-01  7.477e-01  -0.426
## native_countryHaiti                   1.927e-01  5.078e-01   0.379
## native_countryHoland-Netherlands     -8.348e+00  3.247e+02  -0.026
## native_countryHonduras               -1.413e+00  2.105e+00  -0.671
## native_countryHong                   -4.326e-01  5.976e-01  -0.724
## native_countryHungary                 4.144e-01  6.326e-01   0.655
```

```
## native_countryIndia                       -2.766e-01  2.836e-01  -0.975
## native_countryIran                         2.927e-01  4.009e-01   0.730
## native_countryIreland                      1.276e+00  5.018e-01   2.542
## native_countryItaly                        7.790e-01  2.991e-01   2.605
## native_countryJamaica                      2.015e-01  4.142e-01   0.486
## native_countryJapan                       -6.937e-02  3.474e-01  -0.200
## native_countryLaos                        -1.304e+00  8.638e-01  -1.510
## native_countryMexico                      -5.924e-01  2.234e-01  -2.651
## native_countryNicaragua                   -9.403e-01  7.831e-01  -1.201
## native_countryOutlying-US(Guam-USVI-etc) -7.466e-01  1.080e+00  -0.692
## native_countryPeru                        -6.493e-01  6.353e-01  -1.022
## native_countryPhilippines                  2.360e-01  2.417e-01   0.976
## native_countryPoland                      -2.304e-02  3.639e-01  -0.063
## native_countryPortugal                     6.013e-01  4.451e-01   1.351
## native_countryPuerto-Rico                 -1.232e-01  3.323e-01  -0.371
## native_countryScotland                    -1.595e-01  7.555e-01  -0.211
## native_countrySouth                       -1.147e+00  3.835e-01  -2.991
## native_countryTaiwan                      -2.788e-02  4.148e-01  -0.067
## native_countryThailand                    -7.829e-01  6.973e-01  -1.123
## native_countryTrinadad&Tobago             -1.156e+00  8.340e-01  -1.386
## native_countryUnited-States                2.445e-01  1.135e-01   2.155
## native_countryVietnam                     -9.150e-01  5.077e-01  -1.802
## native_countryYugoslavia                   7.909e-01  6.123e-01   1.292
##                                           Pr(>|z|)
## (Intercept)                                < 2e-16 ***
## age                                        < 2e-16 ***
## workclassFederal-gov                       < 2e-16 ***
## workclassLocal-gov                        2.06e-05 ***
## workclassNever-worked                     0.924684
## workclassPrivate                          1.60e-10 ***
## workclassSelf-emp-inc                     6.68e-11 ***
## workclassSelf-emp-not-inc                 0.242303
## workclassState-gov                        0.016340 *
## workclassWithout-pay                      0.797719
## fnlwgt                                    1.17e-07 ***
## education11th                             0.793465
## education12th                             0.046958 *
## education1st-4th                          0.150368
## education5th-6th                          0.187776
## education7th-8th                          0.022567 *
## education9th                              0.377763
## educationAssoc-acdm                        < 2e-16 ***
## educationAssoc-voc                         < 2e-16 ***
## educationBachelors                         < 2e-16 ***
## educationDoctorate                         < 2e-16 ***
## educationHS-grad                          1.67e-09 ***
## educationMasters                           < 2e-16 ***
## educationPreschool                        0.168785
## educationProf-school                       < 2e-16 ***
## educationSome-college                      < 2e-16 ***
## educational_num                                 NA
## marital_statusMarried-AF-spouse           1.83e-07 ***
## marital_statusMarried-civ-spouse           < 2e-16 ***
## marital_statusMarried-spouse-absent       0.520216
```

```
## marital_statusNever-married              1.44e-08 ***
## marital_statusSeparated                  0.316373
## marital_statusWidowed                    0.539548
## occupationAdm-clerical                   0.264077
## occupationArmed-Forces                   0.695457
## occupationCraft-repair                   0.074013 .
## occupationExec-managerial                 < 2e-16 ***
## occupationFarming-fishing                7.74e-13 ***
## occupationHandlers-cleaners              2.46e-07 ***
## occupationMachine-op-inspct              0.032506 *
## occupationOther-service                  3.34e-13 ***
## occupationPriv-house-serv                0.012767 *
## occupationProf-specialty                 1.61e-14 ***
## occupationProtective-serv                3.14e-07 ***
## occupationSales                          5.03e-06 ***
## occupationTech-support                   1.07e-11 ***
## occupationTransport-moving                     NA
## relationshipNot-in-family                0.010078 *
## relationshipOther-relative               0.038592 *
## relationshipOwn-child                    0.022251 *
## relationshipUnmarried                    0.085822 .
## relationshipWife                          < 2e-16 ***
## raceAsian-Pac-Islander                   0.000299 ***
## raceBlack                                0.049111 *
## raceOther                                0.092677 .
## raceWhite                                0.001414 **
## genderMale                                < 2e-16 ***
## capital_gain                              < 2e-16 ***
## capital_loss                              < 2e-16 ***
## hours_per_week                            < 2e-16 ***
## native_countryCambodia                   0.072240 .
## native_countryCanada                     0.004880 **
## native_countryChina                      0.030269 *
## native_countryColumbia                   0.004622 **
## native_countryCuba                       0.261368
## native_countryDominican-Republic         0.041571 *
## native_countryEcuador                    0.430972
## native_countryEl-Salvador                0.161737
## native_countryEngland                    0.083770 .
## native_countryFrance                     0.074147 .
## native_countryGermany                    0.319802
## native_countryGreece                     0.573207
## native_countryGuatemala                  0.669817
## native_countryHaiti                      0.704402
## native_countryHoland-Netherlands         0.979492
## native_countryHonduras                   0.501972
## native_countryHong                       0.469151
## native_countryHungary                    0.512433
## native_countryIndia                      0.329360
## native_countryIran                       0.465257
## native_countryIreland                    0.011025 *
## native_countryItaly                      0.009196 **
## native_countryJamaica                    0.626736
## native_countryJapan                      0.841702
```

```
## native_countryLaos                        0.131047
## native_countryMexico                      0.008020 **
## native_countryNicaragua                   0.229859
## native_countryOutlying-US(Guam-USVI-etc) 0.489229
## native_countryPeru                        0.306739
## native_countryPhilippines                 0.328968
## native_countryPoland                      0.949509
## native_countryPortugal                    0.176764
## native_countryPuerto-Rico                 0.710764
## native_countryScotland                    0.832756
## native_countrySouth                       0.002779 **
## native_countryTaiwan                      0.946418
## native_countryThailand                    0.261537
## native_countryTrinadad&Tobago             0.165765
## native_countryUnited-States               0.031189 *
## native_countryVietnam                     0.071501 .
## native_countryYugoslavia                  0.196480
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 48173  on 43841  degrees of freedom
## Residual deviance: 27615  on 43743  degrees of freedom
## AIC: 27813
##
## Number of Fisher Scoring iterations: 11
```

With the combination of standard error and z-value, if the standard error is low enough and the z-value is extreme enough, the predictor is effective. Based on the summary, total effective predictors are age, fnlwgt, gender, capital gain, capital loss and hours per week. Also the effective predictors of workclass, education, occupation, relationship, marital_status, race and native_country are 0.625, 0.666, 0.714, 0.8, 0.5, 0.75 and 0.220. The effective predictors of native_country is relatively low, which means the native_country may not be an appropriate predictor. Unfortunately, all the stastical summary of educational_num are NA, which means educational_num is not an appropriate predictor. According to the p-value in the summary, we could safely draw the conclusion that age, workclass, fnlwgt, educaion, marital_status, occupation, relationship, race, capital_gain, capital_loss, hour_per_week are significant in preditors.

B.3

The confusion matrix on the test set is :

```r
b.testdata <- read.csv('adult_income_test.csv')
b.testdata$predict <- predict(b.model, b.testdata, type = 'response')
```

```
## Warning in predict.lm(object, newdata, se.fit, scale = 1, type =
## ifelse(type == : prediction from a rank-deficient fit may be misleading
```

```r
b.testdata$Y_predict[b.testdata$predict>=0.5] <- 1
b.testdata$Y_predict[b.testdata$predict<0.5] <- 0

b.testdata$Y[b.testdata$income == '>50K'] <- 1
b.testdata$Y[b.testdata$income == '<=50K'] <- 0

confusion.matrix <- as.matrix(table('Actual'=b.testdata$Y, 'Prediction'=b.testdata$Y_predict))
```

```
print(confusion.matrix)
```

```
##       Prediction
## Actual    0    1
##      0 3507  264
##      1  493  736
```

As the confusion matrix shows, 3507 of income less than 50K(i.e. type=0) and 736 of income greater than 50K(i.e. type=1) are predicted correctly. Here, 3507 and 736 represent True Negative and True Positive values, respectively. The confusion matrix also indicates that 264 of income less than 50K are predicted as the greater than 50K, and 493 greater than 50K are classified as less than 50K. Thus, the accuracy has been calculated as :

```
N <- nrow(b.testdata)               # number of observations
diag <- diag(confusion.matrix)  # TN and TP
Accuracy <- sum(diag)/N         # accuracy = (TP + TN)/N
round(Accuracy*100,2)
```

```
## [1] 84.86
```

Also, the precision and recall have been calculated as :

```
rowsums = apply(confusion.matrix, 1, sum)   # number of observations per class
colsums = apply(confusion.matrix, 2, sum)   # number of predictions per class
Precision = diag / colsums
Recall = diag / rowsums
round(data.frame(Precision, Recall)*100,5)
```

```
##   Precision   Recall
## 0    87.675 92.99920
## 1    73.600 59.88609
```

B.4

Based on the conclusioni on B.2, I decided to delete education_num and native_country from the predictors. Thus :

```
b.rawdata <- read.csv('adult_income_train.exact1.csv')
b.testdata <- read.csv('adult_income_test.exact1.csv')

b.rawdata <- subset(b.rawdata, select=c("age", "workclass", "fnlwgt", "education","marital_status","occu
                                       "relationship", "race", "gender", "capital_gain",
                                       "capital_loss", "hours_per_week", "income"))
b.testdata <- subset(b.testdata, select=c("age", "workclass", "fnlwgt", "education","marital_status", "
                                     "relationship", "race", "gender", "capital_gain",
                                     "capital_loss", "hours_per_week", "income"))

b.model <- glm(income~., family = binomial, data = b.rawdata)
```

```
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
```

```
summary(b.model)
```

```
##
## Call:
## glm(formula = income ~ ., family = binomial, data = b.rawdata)
##
```

```
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -5.0989  -0.5060  -0.1850  -0.0348   3.6948
##
## Coefficients:
##                                    Estimate Std. Error z value Pr(>|z|)
## (Intercept)                      -7.332e+00  3.594e-01 -20.399  < 2e-16
## age                               2.528e-02  1.398e-03  18.089  < 2e-16
## workclassLocal-gov               -7.088e-01  9.592e-02  -7.389 1.48e-13
## workclassother                   -1.138e+00  8.842e-02 -12.871  < 2e-16
## workclassPrivate                 -5.476e-01  7.958e-02  -6.880 5.97e-12
## workclassSelf-emp-inc            -4.091e-01  1.050e-01  -3.896 9.78e-05
## workclassState-gov               -9.083e-01  1.076e-01  -8.443  < 2e-16
## fnlwgt                            6.543e-07  1.455e-07   4.496 6.92e-06
## education12th                     4.278e-01  2.263e-01   1.890 0.058749
## education7th-8th                 -4.857e-01  1.992e-01  -2.438 0.014755
## educationAssoc-acdm               1.391e+00  1.517e-01   9.170  < 2e-16
## educationAssoc-voc                1.310e+00  1.467e-01   8.931  < 2e-16
## educationBachelors                1.942e+00  1.361e-01  14.268  < 2e-16
## educationDoctorate                2.852e+00  1.839e-01  15.508  < 2e-16
## educationHS-grad                  8.130e-01  1.329e-01   6.117 9.53e-10
## educationMasters                  2.252e+00  1.447e-01  15.570  < 2e-16
## educationother                   -2.467e-01  1.600e-01  -1.541 0.123240
## educationProf-school              2.777e+00  1.729e-01  16.061  < 2e-16
## educationSome-college             1.181e+00  1.347e-01   8.771  < 2e-16
## marital_statusMarried-AF-spouse   2.453e+00  4.734e-01   5.182 2.19e-07
## marital_statusMarried-civ-spouse  2.278e+00  2.292e-01   9.941  < 2e-16
## marital_statusNever-married      -4.283e-01  7.551e-02  -5.672 1.41e-08
## marital_statusother              -2.043e-02  9.649e-02  -0.212 0.832277
## occupationExec-managerial         7.293e-01  5.559e-02  13.118  < 2e-16
## occupationFarming-fishing        -9.957e-01  1.117e-01  -8.911  < 2e-16
## occupationHandlers-cleaners      -7.901e-01  1.142e-01  -6.917 4.63e-12
## occupationMachine-op-inspct      -3.498e-01  7.675e-02  -4.557 5.19e-06
## occupationother                  -1.076e-01  5.388e-02  -1.998 0.045725
## occupationOther-service          -9.390e-01  9.452e-02  -9.935  < 2e-16
## occupationPriv-house-serv        -2.667e+00  9.947e-01  -2.681 0.007343
## occupationProf-specialty          4.850e-01  6.228e-02   7.787 6.84e-15
## occupationProtective-serv         4.467e-01  1.017e-01   4.393 1.12e-05
## occupationSales                   2.226e-01  5.879e-02   3.787 0.000153
## occupationTech-support            5.536e-01  8.774e-02   6.309 2.81e-10
## relationshipNot-in-family         5.563e-01  2.268e-01   2.453 0.014177
## relationshipOther-relative       -5.069e-01  2.136e-01  -2.373 0.017632
## relationshipOwn-child            -5.432e-01  2.233e-01  -2.433 0.014992
## relationshipUnmarried             3.695e-01  2.411e-01   1.532 0.125420
## relationshipWife                  1.190e+00  8.719e-02  13.650  < 2e-16
## raceAsian-Pac-Islander            4.952e-01  2.104e-01   2.354 0.018585
## raceBlack                         3.892e-01  2.025e-01   1.922 0.054650
## raceOther                         2.211e-01  2.840e-01   0.778 0.436313
## raceWhite                         6.199e-01  1.931e-01   3.211 0.001324
## genderMale                        7.498e-01  6.693e-02  11.203  < 2e-16
## capital_gain                      3.229e-04  9.000e-06  35.878  < 2e-16
## capital_loss                      6.370e-04  3.187e-05  19.989  < 2e-16
## hours_per_week                    2.886e-02  1.362e-03  21.193  < 2e-16
##
```

```
## (Intercept)                       ***
## age                               ***
## workclassLocal-gov                ***
## workclassother                    ***
## workclassPrivate                  ***
## workclassSelf-emp-inc             ***
## workclassState-gov                ***
## fnlwgt                            ***
## education12th                     .
## education7th-8th                  *
## educationAssoc-acdm               ***
## educationAssoc-voc                ***
## educationBachelors                ***
## educationDoctorate                ***
## educationHS-grad                  ***
## educationMasters                  ***
## educationother
## educationProf-school              ***
## educationSome-college             ***
## marital_statusMarried-AF-spouse   ***
## marital_statusMarried-civ-spouse ***
## marital_statusNever-married       ***
## marital_statusother
## occupationExec-managerial         ***
## occupationFarming-fishing         ***
## occupationHandlers-cleaners       ***
## occupationMachine-op-inspct       ***
## occupationother                   *
## occupationOther-service           ***
## occupationPriv-house-serv         **
## occupationProf-specialty          ***
## occupationProtective-serv         ***
## occupationSales                   ***
## occupationTech-support            ***
## relationshipNot-in-family         *
## relationshipOther-relative        *
## relationshipOwn-child             *
## relationshipUnmarried
## relationshipWife                  ***
## raceAsian-Pac-Islander            *
## raceBlack                         .
## raceOther
## raceWhite                         **
## genderMale                        ***
## capital_gain                      ***
## capital_loss                      ***
## hours_per_week                    ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 48173  on 43841  degrees of freedom
## Residual deviance: 27761  on 43795  degrees of freedom
```

```
## AIC: 27855
##
## Number of Fisher Scoring iterations: 7
```

Acoording to the summary above, almost all the predictors are significant execpt some other types. Report the confusion matrix on the test set is :

```
b.testdata$predict <- predict(b.model, b.testdata, type = 'response')
b.testdata$Y_predict[b.testdata$predict>=0.5] <- 1
b.testdata$Y_predict[b.testdata$predict<0.5] <- 0

b.testdata$Y[b.testdata$income == '>50K'] <- 1
b.testdata$Y[b.testdata$income == '<=50K'] <- 0

confusion.matrix <- as.matrix(table('Actual'=b.testdata$Y, 'Prediction'=b.testdata$Y_predict))
print(confusion.matrix)
```

```
##       Prediction
## Actual    0    1
##      0 3507  264
##      1  493  736
```

Also, the accuracy :

```
N <- nrow(b.testdata)               # number of observations
diag <- diag(confusion.matrix)  # TN and TP
Accuracy <- sum(diag)/N         # accuracy = (TP + TN)/N
round(Accuracy*100,2)
```

```
## [1] 84.86
```

Moreover, the precision and recall :

```
rowsums = apply(confusion.matrix, 1, sum)   # number of observations per class
colsums = apply(confusion.matrix, 2, sum)   # number of predictions per class
Precision = diag / colsums
Recall = diag / rowsums
round(data.frame(Precision, Recall)*100,5)
```

```
##   Precision   Recall
## 0    87.675 92.99920
## 1    73.600 59.88609
```

Task C: Sampling

C.1

According to the rejection method, the sampling algorithm can be defined as :

$$AcceptanceProbability = (\frac{Cq(X)}{P_{prop}(X)})$$

In which, $q(X)$ means the PDF proportion, $P_{prop}(X)$ means distribution, and C means constant. Also, it rejects $1 - (\frac{Cq(X)}{P_{prop}(X)})$ of its samples.

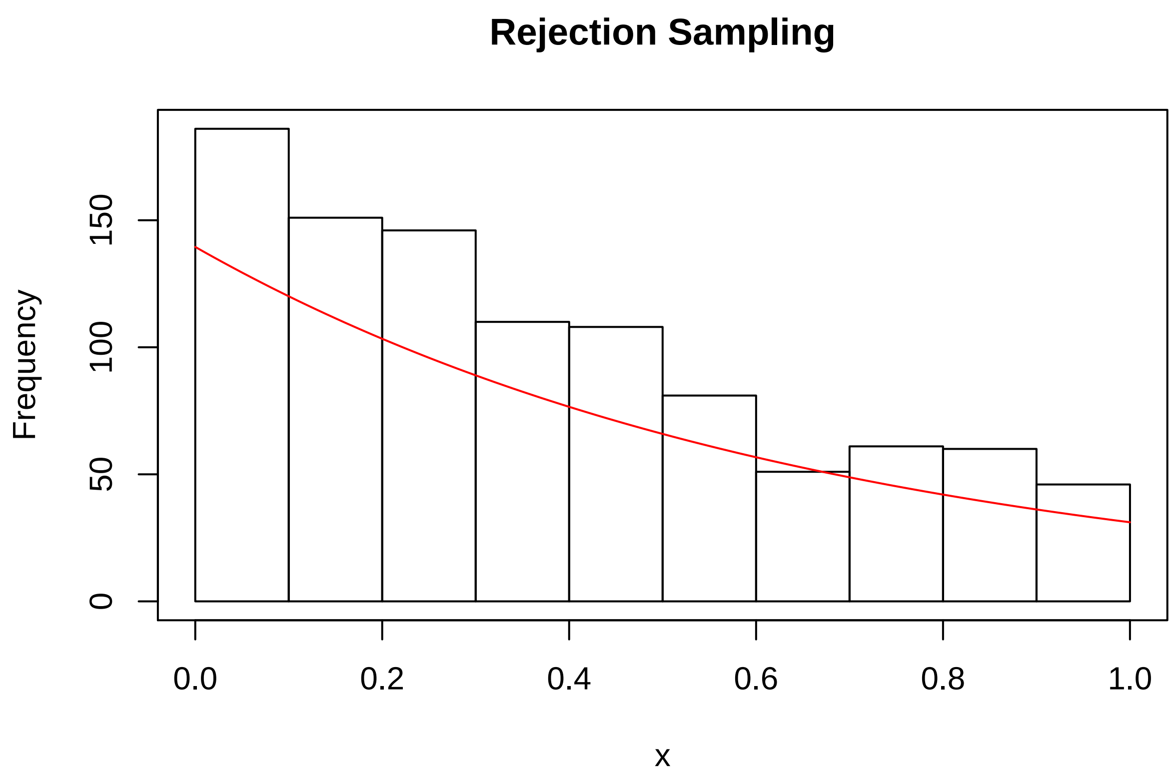
```
pdf <- function(x) {
  lambda <- 1.5
  if (x > 0)
    lambda * (1/exp(lambda*x))
}

atest <- function(x) {
  if (x > 0 && x < 1)
    1
  else 0
}

C <- 1/2
size <- 1000
count <- 0
output <- c()
#x.grid <- seq(0, 1, by = 0.01)

while (count < size) {
  sample <- runif(1, 0, 1)
  aprob <- (C * pdf(sample))/atest(sample)
  u <- runif(1, 0, 1)
  if (aprob >= u) {
    output <- c(output, sample)
    count <- count + 1
  }
}

hist(output, main="Rejection Sampling", xlab='x')
par(new=TRUE)
curve(1.5 * (1/exp(1.5*x)), from = 0, to = 1, xlim=c(0,1), ylim=c(0,2), col = "red", xlab = "", ylab=""
```

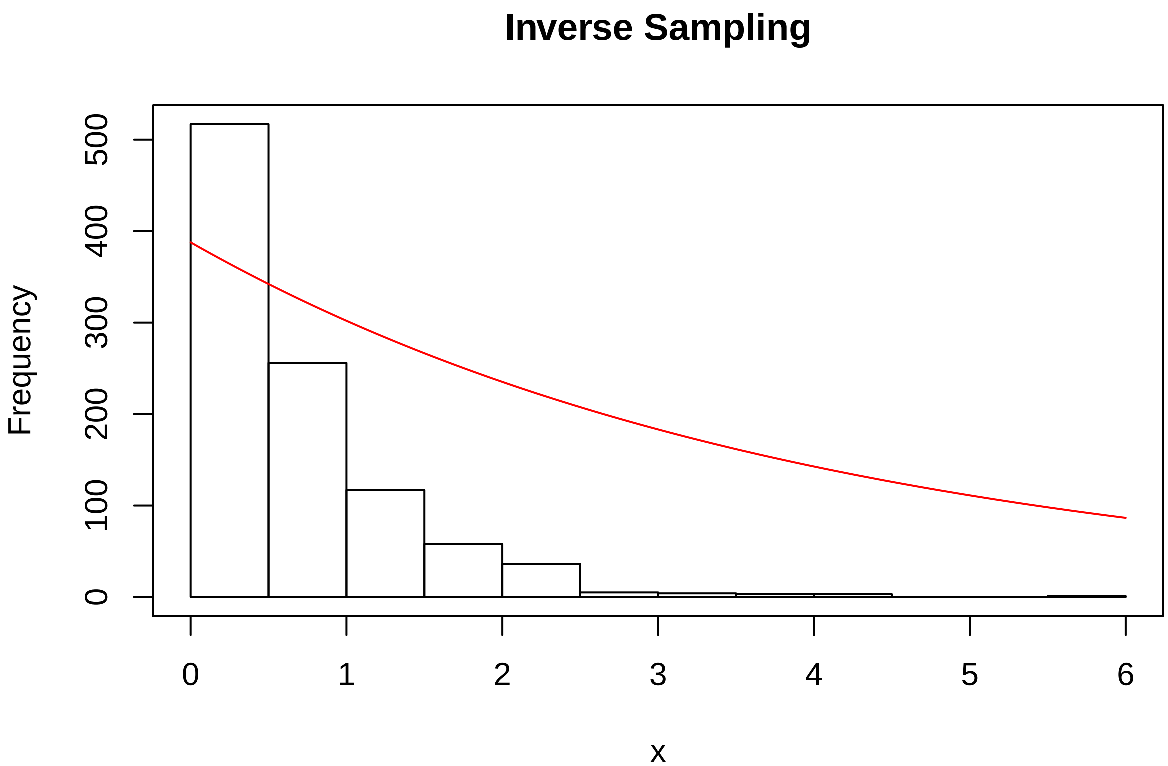


C.2

According to the inverse sampling method, let RV X have the CDF P(X), and its quantile function be Q(p). To sample X, sample Uas a uniform variable in (0,1) and then return Q(U).

```
size <- 1000
u <- runif(size, 0, 1)
quantile <- function(u) {
  lambda = 1.5
  if (u > 0 && u < 1)
    -log(1-u)/lambda
}
x <- unlist(lapply(u, quantile))

hist(x, main="Inverse Sampling")
par(new=TRUE)
curve(1.5 * (1/exp(1.5*x)), from = 0, to = 1, xlim=c(0,1), ylim=c(0,2), col = "red",xlab = "", ylab="",
```



C.3

The Monte Carlo method is the use of randomness to solve probelms, which often done with simulation. In this case, the threshold/possibility of C, S, R, W for each combinations can be calculated as following:

$$
\begin{aligned}
p(\mathrm{c}|\mathrm{r},\ \mathrm{s},\ \mathrm{w}) &= p(\mathrm{c}|\mathrm{s},\ \mathrm{r}) \\
&= \frac{p(\mathrm{c})p(\mathrm{s},\ \mathrm{r}|\mathrm{c})}{p(\mathrm{s},\ \mathrm{r})} \\
&= \frac{p(\mathrm{c})p(\mathrm{s}|\mathrm{c})p(\mathrm{r}|\mathrm{c})}{p(\mathrm{c})p(\mathrm{s}|\mathrm{c})p(\mathrm{r}|\mathrm{c}) + p(\neg c)p(\mathrm{s}|\neg c)p(\mathrm{r}|\neg c)} \\
&= \frac{0.5 * 0.1 * 0.8}{0.5 * 0.1 * 0.8 + 0.5 * 0.5 * 0.2} \\
&= 0.4444
\end{aligned}
$$

$$
\begin{aligned}
p(\text{r}|\text{c, s, w}) &= \frac{p(\text{r}|\text{c, s})p(\text{w}|\text{r, c, s})}{p(\text{w}|\text{c, s})} \\
&= \frac{p(\text{r}|\text{c})p(\text{w}|\text{r, s})}{p(\text{w}|\text{c, s})} \\
&= \frac{p(\text{r}|\text{c})p(\text{w}|\text{r, s})}{p(\text{r}|\text{c})p(\text{w}|\text{r, s}) + p(\neg r|c)p(\text{w}|\neg r, s)} \\
&= \frac{0.8 * 0.99}{0.8 * 0.99 + 0.2 * 0.9} \\
&= 0.8148
\end{aligned}
$$

$$
\begin{aligned}
p(\text{s}|\text{c, r, w}) &= \frac{p(\text{s}|\text{c, r})p(\text{w}|\text{r, c, s})}{p(\text{w}|\text{c, r})} \\
&= \frac{p(\text{s}|\text{c})p(\text{w}|\text{r, s})}{p(\text{w}|\text{c, r})} \\
&= \frac{p(\text{s}|\text{c})p(\text{w}|\text{r, s})}{p(\text{s}|\text{c})p(\text{w}|\text{r, s}) + p(\neg s|c)p(\text{w}|\neg s, r)} \\
&= \frac{0.1 * 0.01}{0.1 * 0.01 + 0.9 * 0.9} \\
&= 0.9878
\end{aligned}
$$

$$
\begin{aligned}
p(\text{w}|\text{c, s, r}) &= \frac{p(\text{w}|\text{r,s})p(\text{s}|\text{c})p(\text{r}|\text{c})}{p(\text{w}|\text{r,s})p(\text{s}|\text{c})p(\text{r}|\text{c} + p(\neg w|r, s)p(\text{s}|\text{c})p(\text{r}|\text{c})} \\
&= \frac{0.99 * 0.1 * 0.8}{0.99 * 0.1 * 0.8 + 0.01 * 0.1 * 0.8} \\
&= 0.99
\end{aligned}
$$

Also, the simulation is :

```r
# set first sample value
c <- 0
s <- 0
r <- 0
w <- 0

RNA <- c()
for(i in 1:1000)
{

  x <- runif(1,min = 0, max = 1) #P(Cloudy)
  if(x <= 0.5)
  {
```

```r
    c <- 0
  }
  else
  {
    c <- 1
  }


  x <- runif(1,min = 0, max = 1) #P(Sprinkler)

  if((c == 0 && x <= 0.5) ||
      (c == 1 && x <= 0.9))
  {
    s <- 0
  }
  else
  {
    s <- 1
  }


  x <- runif(1,min = 0, max = 1) #P(Rain)

  if(c == 0 && x <= 0.8 ||
      c == 1 && x <= 0.2)
  {
    r <- 0
  }
  else
  {
    r <- 1
  }


  x <- runif(1,min = 0, max = 1) #P(Wetgrass)

  if(s==0 && r==0 && x<=1 ||
      s==1 && r==0 && x<=0.1 ||
      s==0 && r==1 && x<=0.1 ||
      s==1 && r==1 && x<=0.01  )
  {
    w <- 0
  }
  else
  {
    w <- 1
  }


  RN <- c(c,s,r,w)
  RNA[[i]] <- RN
}
```

```r
RNA = as.data.frame(matrix(unlist(RNA),nrow=1000))

colnames(RNA)[colnames(RNA)=="V1"] <- "Cloudy"
colnames(RNA)[colnames(RNA)=="V2"] <- "Sprinkler"
colnames(RNA)[colnames(RNA)=="V3"] <- "Rain"
colnames(RNA)[colnames(RNA)=="V4"] <- "Wetgrass"

df = tail(RNA, -100)
```

The table of Wetgrass and Cloudy :

```r
drops <- c("Wetgrass","Cloudy")
df1 = df[ , !(names(df) %in% drops)]

tb1 = table(df1)#Finding number of 0s and 1s
round(tb1/900,2)
```

```
##           Rain
## Sprinkler    0    1
##         0 0.28 0.25
##         1 0.24 0.23
```

The table of Sprinkler and Rain :

```r
drops <- c("Sprinkler","Rain")
df2 = df[ , !(names(df) %in% drops)]

tb2 = table(df2)#Finding number of 0s and 1s
round(tb2/900,2)
```

```
##       Wetgrass
## Cloudy    0    1
##      0 0.28 0.24
##      1 0.22 0.25
```