FIT5197_2018_S1_Assignment_2_solutions

Kelvin Li and Wray Buntine

20 May 2018

A.1

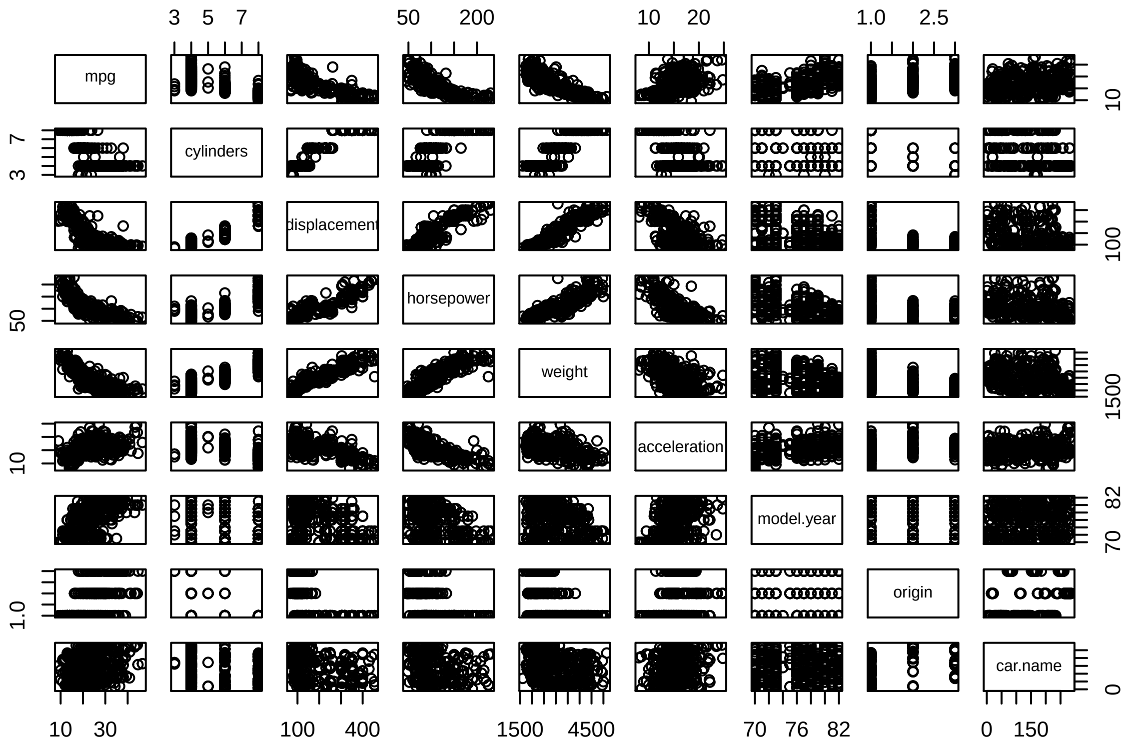
There are some missing values listed as "?". Describe your strategy for treating missing values and update (edit by hand) the file accordingly.

```
auto_mpg_train = read.csv("auto_mpg_train.csv", stringsAsFactors = F)
auto_mpg_test = read.csv("auto_mpg_test.csv")

# only 6 missing values appear in horsepower
# so googled their horsepower
auto_mpg_train$horsepower[33] = "75"
auto_mpg_train$horsepower[127] = "84"
auto_mpg_train$horsepower[281] = "51"
auto_mpg_train$horsepower[287] = "132"
auto_mpg_train$horsepower[305] = "78"
auto_mpg_train$horsepower[325] = "82"

# convert horsepower to numeric format
auto_mpg_train$horsepower = as.numeric(auto_mpg_train$horsepower)
```

A.2

Pair plot mpg vs. the other variables to visualize the relationships and discuss what you see.

```
auto_mpg_train$car.name = as.factor(auto_mpg_train$car.name)
pairs(auto_mpg_train)
```



Negative correlations are observed between mpg and cylinders, displacement, horsepower, weight.

Positive correlations are observed between mpg and acceleration, model year, origin.

mpg and car name don't have a clear linear correlation.

A.3

Based on your pair plots, propose an initial set of variables to use for a multiple linear regression model to predict mpg.

The initial set of predictors based on the pair plots are all variables except car name.

A.4

With variables of your choice build the model using the lm() routine in R, and then print the summary of the model to get the R diagnostics. Briefly explain the statistics in the summary, e.g. R 2 value, t-value, standard error, p-value (ignoring the F-statistics line). What does this imply about the predictors for your model?

```
model1 = lm(mpg ~ cylinders + displacement + horsepower + weight + acceleration + model.year + origin, 
summary(model1)
```

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

```
## Residuals:
##    Min     1Q Median     3Q    Max
## -9.554 -2.154 -0.104  1.836 12.854
##
## Coefficients:
##                Estimate Std. Error t value Pr(>|t|)
## (Intercept)  -1.454e+01  4.913e+00  -2.959 0.003300 **
## cylinders    -5.770e-01  3.476e-01  -1.660 0.097826 .
## displacement  2.811e-02  8.099e-03   3.471 0.000586 ***
## horsepower   -2.771e-02  1.467e-02  -1.889 0.059741 .
## weight       -6.962e-03  7.193e-04  -9.679  < 2e-16 ***
## acceleration  9.056e-02  1.057e-01   0.857 0.392182
## model.year    7.330e-01  5.328e-02  13.756  < 2e-16 ***
## origin        1.469e+00  2.949e-01   4.981 1.01e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3.402 on 340 degrees of freedom
## Multiple R-squared:  0.8228, Adjusted R-squared:  0.8192
## F-statistic: 225.5 on 7 and 340 DF,  p-value: < 2.2e-16
```

The built model can be assessed from a number of metrics as the following.

1. To check if the residuals are standard normally distributet. The median is close to 0, and distribution is not too different to symmetric, so the residuals are fine though perhaps could be improved. This could be caused by the dependencies between the predictors.
2. The p-value (or t value) of each predictor suggests its importance for predicting the target. Acceleration shows insignificance to mpg. Cylinders and horsepower could be considered insignificant at 5% level.
3. The standard errors measure the variance of each sample from the mean. In our case, none of the values are too large.
4. According to the R-square value, The built model fits about 82% of the training data, which can be considered as a good fit.

A.5

Test the fitted model using the "auto mpg test.csv", and calculate the MSE on the test set, reporting it. Note the test set has no missing values.

```r
# define a mse function
# students may use existing mse functions from other packages
mse = function(x, y) mean((x - y) ^ 2)

# predict mpg using model1 and test data
mpg_predicted = predict(model1, newdata = auto_mpg_test)
cat("Test MSE = ",mse(auto_mpg_test$mpg, mpg_predicted),"\n")
```

```
## Test MSE =  8.462446
```

A.6

Try out some new constructed variables. Just give one here, but others may have been found. Note these are quite tricky, in that some give a mild improvement to adjusted R-squared on the training set, but don't improve MSE on the test set. The one used below gave a good

```r
#  build the new model with a new feature
auto_mpg_train$newfeature =   auto_mpg_train$acceleration/auto_mpg_train$horsepower
model2 = lm(mpg ~ cylinders + displacement + horsepower + weight + acceleration + model.year + origin +

# check quality
summary(model2)
```

```
##
## Call:
## lm(formula = mpg ~ cylinders + displacement + horsepower + weight +
##     acceleration + model.year + origin + newfeature, data = auto_mpg_train)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -9.7947 -1.8168 -0.0982  1.5797 12.2045
##
## Coefficients:
##                Estimate Std. Error t value Pr(>|t|)
## (Intercept)  -1.667e+01  4.421e+00  -3.770 0.000192 ***
## cylinders    -3.098e-01  3.137e-01  -0.987 0.324142
## displacement  2.819e-03  7.795e-03   0.362 0.717886
## horsepower    7.898e-03  1.376e-02   0.574 0.566239
## weight       -3.966e-03  7.261e-04  -5.462 9.11e-08 ***
## acceleration -9.609e-01  1.500e-01  -6.406 4.99e-10 ***
## model.year    7.403e-01  4.789e-02  15.458  < 2e-16 ***
## origin        9.167e-01  2.719e-01   3.371 0.000836 ***
## newfeature    5.499e+01  6.072e+00   9.057  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3.057 on 339 degrees of freedom
## Multiple R-squared:  0.8573, Adjusted R-squared:  0.854
## F-statistic: 254.6 on 8 and 339 DF,  p-value: < 2.2e-16
```

```r
#now test
auto_mpg_test$newfeature = auto_mpg_test$acceleration/auto_mpg_test$horsepower
mpg_predicted = predict(model2, newdata = auto_mpg_test)
cat("Test MSE with new feature = ",mse(auto_mpg_test$mpg, mpg_predicted),"\n")
```

```
## Test MSE with new feature =  6.388582
```

B.1

There are some missing values listed as "?". Describe your strategy for treating missing values, but note sometimes it is OK to leave missing value as a separate categorical value (we call this "missing informative").

Most missings appear at workclass (6.4%) and occupation (6.4%), and the missings in these two columns appear almost simultaneously, so we treat the missings in these two columns as new states "not given". The missings in native_country are less than 2%, so we replaced missings with the mode, which is US.

```r
adult_income_train = read.csv("adult_income_train.csv", stringsAsFactors = F)
adult_income_test = read.csv("adult_income_test.csv", stringsAsFactors = F)

adult_income_train$workclass[which(adult_income_train$workclass == "?")] = "Not_given"
```

```r
adult_income_train$occupation[which(adult_income_train$occupation == "?")] = "Not_given"
adult_income_train$native_country[which(adult_income_train$native_country == "?")] = "United-State"
```

B.2

With all variables, build a model using the routine in R, and then print the summary of the model to get the R diagnostics. Briefly explain the statistics in the summary, e.g. Z-value, standard error, p-value. What does this imply about the predictors for your model? Notice many of the variables are multi-valued categorical, and in most cases only some of the values are significant.

```r
adult_income_train$income = as.factor(adult_income_train$income)
adult_income_test$income = as.factor(adult_income_test$income)
model1 = glm(income ~ ., data = adult_income_train, family = binomial(link = "logit"))
```

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

```r
summary(model1)
```

```
##
## Call:
## glm(formula = income ~ ., family = binomial(link = "logit"),
##     data = adult_income_train)
##
## 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)                              -6.700e+00  6.587e-01 -10.171
## age                                       2.493e-02  1.421e-03  17.547
## workclassLocal-gov                       -6.918e-01  9.684e-02  -7.144
## workclassNever-worked                    -9.352e+00  8.524e+01  -0.110
## workclassNot_given                       -1.294e+00  1.191e-01 -10.858
## workclassPrivate                         -5.237e-01  8.067e-02  -6.491
## workclassSelf-emp-inc                    -3.686e-01  1.061e-01  -3.474
## workclassSelf-emp-not-inc                -1.063e+00  9.446e-02 -11.257
## workclassState-gov                       -8.881e-01  1.081e-01  -8.213
## workclassWithout-pay                     -1.402e+00  7.893e-01  -1.776
## 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
## occupationArmed-Forces                   2.606e-01  9.062e-01   0.288
## occupationCraft-repair                   3.560e-02  6.866e-02   0.518
## occupationExec-managerial                7.657e-01  6.629e-02  11.552
## occupationFarming-fishing               -9.769e-01  1.216e-01  -8.031
## occupationHandlers-cleaners             -7.385e-01  1.219e-01  -6.056
## occupationMachine-op-inspct             -2.900e-01  8.773e-02  -3.305
## occupationNot_given                             NA         NA      NA
## occupationOther-service                 -8.747e-01  1.014e-01  -8.627
## occupationPriv-house-serv               -2.603e+00  1.006e+00  -2.586
## occupationProf-specialty                 5.226e-01  6.971e-02   7.496
## occupationProtective-serv                4.832e-01  1.087e-01   4.445
## occupationSales                          2.584e-01  7.065e-02   3.657
## occupationTech-support                   5.970e-01  9.442e-02   6.323
## occupationTransport-moving              -9.467e-02  8.477e-02  -1.117
## 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_countryCanada                    -3.086e-01  5.845e-01  -0.528
## native_countryChina                     -1.692e+00  6.027e-01  -2.808
## native_countryColumbia                  -3.243e+00  9.571e-01  -3.388
## native_countryCuba                      -6.580e-01  6.086e-01  -1.081
## native_countryDominican-Republic        -2.540e+00  9.281e-01  -2.737
## native_countryEcuador                   -1.486e+00  8.241e-01  -1.803
## native_countryEl-Salvador               -1.616e+00  6.952e-01  -2.325
## native_countryEngland                   -4.746e-01  6.100e-01  -0.778
## native_countryFrance                    -1.713e-01  7.021e-01  -0.244
## native_countryGermany                   -7.391e-01  5.886e-01  -1.256
## native_countryGreece                    -1.218e+00  6.687e-01  -1.822
## native_countryGuatemala                 -1.309e+00  9.171e-01  -1.427
## native_countryHaiti                     -7.971e-01  7.355e-01  -1.084
## native_countryHoland-Netherlands        -9.338e+00  3.247e+02  -0.029
## native_countryHonduras                  -2.403e+00  2.172e+00  -1.107
## native_countryHong                      -1.422e+00  7.842e-01  -1.814
## native_countryHungary                   -5.754e-01  8.268e-01  -0.696
## native_countryIndia                     -1.266e+00  5.843e-01  -2.167
## native_countryIran                      -6.971e-01  6.623e-01  -1.052
```

```
## native_countryIreland                    2.857e-01  7.299e-01   0.391
## native_countryItaly                     -2.108e-01  6.102e-01  -0.346
## native_countryJamaica                   -7.883e-01  6.746e-01  -1.169
## native_countryJapan                     -1.059e+00  6.214e-01  -1.704
## native_countryLaos                      -2.294e+00  1.001e+00  -2.291
## native_countryMexico                    -1.582e+00  5.759e-01  -2.747
## native_countryNicaragua                 -1.930e+00  9.460e-01  -2.040
## native_countryOutlying-US(Guam-USVI-etc) -1.736e+00  1.202e+00  -1.445
## native_countryPeru                      -1.639e+00  8.286e-01  -1.978
## native_countryPhilippines               -7.538e-01  5.631e-01  -1.339
## native_countryPoland                    -1.013e+00  6.437e-01  -1.573
## native_countryPortugal                  -3.885e-01  6.927e-01  -0.561
## native_countryPuerto-Rico               -1.113e+00  6.272e-01  -1.775
## native_countryScotland                  -1.149e+00  9.241e-01  -1.244
## native_countrySouth                     -2.137e+00  6.358e-01  -3.361
## native_countryTaiwan                    -1.018e+00  6.567e-01  -1.550
## native_countryThailand                  -1.773e+00  8.637e-01  -2.052
## native_countryTrinadad&Tobago           -2.146e+00  9.875e-01  -2.173
## native_countryUnited-State              -9.898e-01  5.506e-01  -1.798
## native_countryUnited-States             -7.453e-01  5.439e-01  -1.370
## native_countryVietnam                   -1.905e+00  7.172e-01  -2.656
## native_countryYugoslavia                -1.989e-01  8.113e-01  -0.245
##                                         Pr(>|z|)
## (Intercept)                              < 2e-16 ***
## age                                      < 2e-16 ***
## workclassLocal-gov                      9.06e-13 ***
## workclassNever-worked                   0.912637
## workclassNot_given                       < 2e-16 ***
## workclassPrivate                        8.50e-11 ***
## workclassSelf-emp-inc                   0.000513 ***
## workclassSelf-emp-not-inc                < 2e-16 ***
## workclassState-gov                       < 2e-16 ***
## workclassWithout-pay                    0.075699 .
## 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
## occupationArmed-Forces                    0.773664
## occupationCraft-repair                    0.604111
## occupationExec-managerial                  < 2e-16 ***
## occupationFarming-fishing                 9.68e-16 ***
## occupationHandlers-cleaners               1.39e-09 ***
## occupationMachine-op-inspct               0.000949 ***
## occupationNot_given                             NA
## occupationOther-service                    < 2e-16 ***
## occupationPriv-house-serv                 0.009701 **
## occupationProf-specialty                  6.56e-14 ***
## occupationProtective-serv                 8.81e-06 ***
## occupationSales                           0.000255 ***
## occupationTech-support                    2.56e-10 ***
## occupationTransport-moving                0.264077
## 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_countryCanada                      0.597558
## native_countryChina                       0.004984 **
## native_countryColumbia                    0.000703 ***
## native_countryCuba                        0.279693
## native_countryDominican-Republic          0.006196 **
## native_countryEcuador                     0.071399 .
## native_countryEl-Salvador                 0.020086 *
## native_countryEngland                     0.436612
## native_countryFrance                      0.807261
## native_countryGermany                     0.209173
## native_countryGreece                      0.068508 .
## native_countryGuatemala                   0.153624
## native_countryHaiti                       0.278475
## native_countryHoland-Netherlands          0.977061
## native_countryHonduras                    0.268439
## native_countryHong                        0.069713 .
## native_countryHungary                     0.486441
## native_countryIndia                       0.030208 *
## native_countryIran                        0.292596
## native_countryIreland                     0.695492
## native_countryItaly                       0.729712
## native_countryJamaica                     0.242560
## native_countryJapan                       0.088305 .
## native_countryLaos                        0.021982 *
## native_countryMexico                      0.006010 **
```

```
## native_countryNicaragua                    0.041311 *
## native_countryOutlying-US(Guam-USVI-etc)  0.148439
## native_countryPeru                         0.047901 *
## native_countryPhilippines                  0.180656
## native_countryPoland                       0.115616
## native_countryPortugal                     0.574856
## native_countryPuerto-Rico                  0.075946 .
## native_countryScotland                     0.213620
## native_countrySouth                        0.000777 ***
## native_countryTaiwan                       0.121235
## native_countryThailand                     0.040130 *
## native_countryTrinadad&Tobago              0.029784 *
## native_countryUnited-State                 0.072240 .
## native_countryUnited-States                0.170587
## native_countryVietnam                      0.007915 **
## native_countryYugoslavia                   0.806358
## ---
## 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
```

p-values show that all continuous variables age, fnlwgt, capital_gain, capital_loss, hours_per_week, and the binary variable gender are important for predicting income. The other categorical variables more or less contain insignificant values. This suggests that we could group the insignificant values into one to reduce the model complexity.

B.3

Test the fitted model using the "adult income test.csv", and calculate the confusion matrix on the test set, reporting it. Also, give the precision, accuracy and recall (Lecture 3). Note the test set has no missing values.

```r
predicted_probs = predict(model1, newdata = adult_income_test, 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
predicted_income = rep(0, nrow(adult_income_test))
for (i in 1:length(predicted_income)) {
  if (predicted_probs[i] > 0.5) {
    predicted_income[i] = ">50K"
  } else {
    predicted_income[i] = "<=50K"
  }
}

# the default setting of glm() predicts >50K, so >50K is  positive and <=50K is negative.
tp = tn = fp = fn = 0
for (i in 1:length(predicted_income)) {
  if ((adult_income_test$income[i] == ">50K") && (predicted_income[i] == ">50K")) {
```

```r
    tp = tp + 1
  } else if ((adult_income_test$income[i] == "<=50K") && (predicted_income[i] == "<=50K")) {
    tn = tn + 1
  } else if ((adult_income_test$income[i] == ">50K") && (predicted_income[i] == "<=50K")) {
    fn = fn + 1
  } else {
    fp = fp + 1
  }
}

(m = matrix(c(tp, fp, fn, tn), 2, 2, byrow = T))
```

```
##      [,1] [,2]
## [1,]  736  264
## [2,]  493 3507
```

```r
(precision = tp / (tp + fp))
```

```
## [1] 0.736
```

```r
(recall = tp / (tp + fn))
```

```
## [1] 0.5988609
```

```r
(accuracy = (tp + tn) / sum(m))
```

```
## [1] 0.8486
```

B.4

Modify some of the categoricals in place. You might have selected different ones, but main thing is to reduce some values that appear to have no significance.

```r
#  mark some values of education as Lower

adult_income_train$education[which(adult_income_train$education == "11th")] = "Lower"
adult_income_test$education[which(adult_income_test$education == "11th")] = "Lower"
adult_income_train$education[which(adult_income_train$education == "1st-4th")] = "Lower"
adult_income_test$education[which(adult_income_test$education == "1st-4th")] = "Lower"
adult_income_train$education[which(adult_income_train$education == "5th-6th")] = "Lower"
adult_income_test$education[which(adult_income_test$education == "5th-6th")] = "Lower"
adult_income_train$education[which(adult_income_train$education == "7th-8th")] = "Lower"
adult_income_test$education[which(adult_income_test$education == "7th-8th")] = "Lower"
adult_income_train$education[which(adult_income_train$education == "12th")] = "Lower"
adult_income_test$education[which(adult_income_test$education == "12th")] = "Lower"
adult_income_train$education[which(adult_income_train$education == "9th")] = "Lower"
adult_income_test$education[which(adult_income_test$education == "9th")] = "Lower"
adult_income_train$education[which(adult_income_train$education == "Preschool")] = "Lower"
adult_income_test$education[which(adult_income_test$education == "Preschool")] = "Lower"
adult_income_train$education[which(adult_income_train$education == "HS-grad")] = "Lower"
adult_income_test$education[which(adult_income_test$education == "HS-grad")] = "Lower"

#  mark some values of marital_status as Other

adult_income_train$marital_status[which(adult_income_train$marital_status == "Married-spouse-absent")] =
adult_income_test$marital_status[which(adult_income_test$marital_status == "Married-spouse-absent")] =
```

```r
adult_income_train$marital_status[which(adult_income_train$marital_status == "Separated")] = "Other"
adult_income_test$marital_status[which(adult_income_test$marital_status == "Separated")] = "Other"
adult_income_train$marital_status[which(adult_income_train$marital_status == "Widowed")] = "Other"
adult_income_test$marital_status[which(adult_income_test$marital_status == "Widowed")] = "Other"

#   only keep these countries, mark the rest as Other

countries <- c("Vietnam", "Trinadad&Tobago", "Thailand", "South", "Peru", "Nicaragua", "Mexico", "Laos"

for (i in 1:length(adult_income_test$native_country)) {
  if ( is.element(adult_income_test$native_country[i], countries) ) {
    #
  } else {
    adult_income_test$native_country[i] = "Other"
  }
}
for (i in 1:length(adult_income_train$native_country)) {
  if ( is.element(adult_income_train$native_country[i], countries) ) {
    #
  } else {
    adult_income_train$native_country[i] = "Other"
  }
}

model2 = glm(income ~ ., data = adult_income_train, family = binomial(link = "logit"))
```

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

```r
summary(model2)
```

```
##
## Call:
## glm(formula = income ~ ., family = binomial(link = "logit"),
##     data = adult_income_train)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -5.1066  -0.5053  -0.1835  -0.0352   3.8121
##
## Coefficients: (1 not defined because of singularities)
##                                      Estimate Std. Error z value Pr(>|z|)
## (Intercept)                        -9.843e+00  4.910e-01 -20.046  < 2e-16
## age                                 2.530e-02  1.406e-03  18.003  < 2e-16
## workclassLocal-gov                 -6.908e-01  9.672e-02  -7.143 9.16e-13
## workclassNever-worked              -9.337e+00  8.580e+01  -0.109 0.913338
## workclassNot_given                 -1.294e+00  1.190e-01 -10.880  < 2e-16
## workclassPrivate                   -5.251e-01  8.057e-02  -6.518 7.13e-11
## workclassSelf-emp-inc              -3.719e-01  1.059e-01  -3.512 0.000445
## workclassSelf-emp-not-inc          -1.063e+00  9.433e-02 -11.269  < 2e-16
## workclassState-gov                 -8.911e-01  1.080e-01  -8.249  < 2e-16
## workclassWithout-pay               -1.400e+00  7.895e-01  -1.774 0.076123
## fnlwgt                              7.693e-07  1.469e-07   5.236 1.64e-07
## educationAssoc-acdm                -1.358e-01  1.960e-01  -0.693 0.488350
## educationAssoc-voc                  3.823e-02  1.796e-01   0.213 0.831407
```

```
## educationBachelors                1.655e-01  1.988e-01   0.832 0.405160
## educationDoctorate                3.409e-01  2.782e-01   1.225 0.220475
## educationLower                    4.145e-02  1.431e-01   0.290 0.772123
## educationMasters                  2.402e-01  2.204e-01   1.090 0.275715
## educationProf-school              5.018e-01  2.553e-01   1.966 0.049337
## educationSome-college             1.600e-01  1.580e-01   1.013 0.311174
## educational_num                   2.523e-01  2.098e-02  12.029  < 2e-16
## marital_statusMarried-AF-spouse   2.502e+00  4.755e-01   5.262 1.42e-07
## marital_statusMarried-civ-spouse  2.331e+00  2.306e-01  10.109  < 2e-16
## marital_statusNever-married      -4.259e-01  7.562e-02  -5.633 1.77e-08
## marital_statusOther              -4.453e-03  9.647e-02  -0.046 0.963185
## occupationArmed-Forces            2.002e-01  9.072e-01   0.221 0.825372
## occupationCraft-repair            3.296e-02  6.852e-02   0.481 0.630555
## occupationExec-managerial         7.615e-01  6.614e-02  11.514  < 2e-16
## occupationFarming-fishing        -9.739e-01  1.216e-01  -8.012 1.13e-15
## occupationHandlers-cleaners      -7.450e-01  1.218e-01  -6.118 9.45e-10
## occupationMachine-op-inspct      -2.910e-01  8.748e-02  -3.327 0.000878
## occupationNot_given                      NA         NA      NA       NA
## occupationOther-service          -8.842e-01  1.011e-01  -8.745  < 2e-16
## occupationPriv-house-serv        -2.610e+00  1.002e+00  -2.605 0.009178
## occupationProf-specialty          5.212e-01  6.957e-02   7.491 6.82e-14
## occupationProtective-serv         4.794e-01  1.085e-01   4.419 9.91e-06
## occupationSales                   2.541e-01  7.050e-02   3.604 0.000313
## occupationTech-support            5.954e-01  9.425e-02   6.317 2.66e-10
## occupationTransport-moving       -1.035e-01  8.461e-02  -1.223 0.221311
## relationshipNot-in-family         5.974e-01  2.282e-01   2.618 0.008855
## relationshipOther-relative       -4.460e-01  2.154e-01  -2.071 0.038395
## relationshipOwn-child            -5.137e-01  2.238e-01  -2.295 0.021735
## relationshipUnmarried             4.198e-01  2.425e-01   1.731 0.083483
## relationshipWife                  1.195e+00  8.744e-02  13.662  < 2e-16
## raceAsian-Pac-Islander            7.759e-01  2.177e-01   3.565 0.000364
## raceBlack                         3.803e-01  2.029e-01   1.874 0.060914
## raceOther                         4.154e-01  2.872e-01   1.446 0.148080
## raceWhite                         6.163e-01  1.934e-01   3.186 0.001442
## genderMale                        7.633e-01  6.731e-02  11.341  < 2e-16
## capital_gain                      3.221e-04  9.006e-06  35.768  < 2e-16
## capital_loss                      6.389e-04  3.193e-05  20.009  < 2e-16
## hours_per_week                    2.872e-02  1.380e-03  20.813  < 2e-16
## native_countryColumbia           -1.623e+00  8.429e-01  -1.925 0.054245
## native_countryDominican-Republic -9.165e-01  8.108e-01  -1.130 0.258357
## native_countryEl-Salvador         1.522e-02  5.284e-01   0.029 0.977029
## native_countryIndia               4.260e-01  3.750e-01   1.136 0.255842
## native_countryLaos               -5.892e-01  8.987e-01  -0.656 0.512115
## native_countryMexico              8.929e-02  3.581e-01   0.249 0.803081
## native_countryNicaragua          -3.066e-01  8.305e-01  -0.369 0.712033
## native_countryOther               8.752e-01  3.018e-01   2.899 0.003739
## native_countryPeru               -7.662e-03  6.933e-01  -0.011 0.991182
## native_countrySouth              -4.447e-01  4.516e-01  -0.985 0.324735
## native_countryThailand           -8.410e-02  7.400e-01  -0.114 0.909508
## native_countryTrinadad&Tobago    -4.844e-01  8.799e-01  -0.550 0.581993
## native_countryVietnam            -2.240e-01  5.609e-01  -0.399 0.689671
##
## (Intercept)                      ***
## age                              ***
```

```
## workclassLocal-gov              ***
## workclassNever-worked
## workclassNot_given              ***
## workclassPrivate                ***
## workclassSelf-emp-inc           ***
## workclassSelf-emp-not-inc       ***
## workclassState-gov              ***
## workclassWithout-pay            .
## fnlwgt                          ***
## educationAssoc-acdm
## educationAssoc-voc
## educationBachelors
## educationDoctorate
## educationLower
## educationMasters
## educationProf-school            *
## educationSome-college
## educational_num                 ***
## marital_statusMarried-AF-spouse  ***
## marital_statusMarried-civ-spouse ***
## marital_statusNever-married     ***
## marital_statusOther
## occupationArmed-Forces
## occupationCraft-repair
## occupationExec-managerial       ***
## occupationFarming-fishing       ***
## occupationHandlers-cleaners     ***
## occupationMachine-op-inspct     ***
## occupationNot_given
## occupationOther-service         ***
## occupationPriv-house-serv       **
## occupationProf-specialty        ***
## occupationProtective-serv       ***
## occupationSales                 ***
## occupationTech-support          ***
## occupationTransport-moving
## relationshipNot-in-family       **
## relationshipOther-relative      *
## relationshipOwn-child           *
## relationshipUnmarried           .
## relationshipWife                ***
## raceAsian-Pac-Islander          ***
## raceBlack                       .
## raceOther
## raceWhite                       **
## genderMale                      ***
## capital_gain                    ***
## capital_loss                    ***
## hours_per_week                  ***
## native_countryColumbia          .
## native_countryDominican-Republic
## native_countryEl-Salvador
## native_countryIndia
## native_countryLaos
```

```
## native_countryMexico
## native_countryNicaragua
## native_countryOther              **
## native_countryPeru
## native_countrySouth
## native_countryThailand
## native_countryTrinadad&Tobago
## native_countryVietnam
## ---
## 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: 27660  on 43779  degrees of freedom
## AIC: 27786
##
## Number of Fisher Scoring iterations: 11
```

A bit ugly, but just copy the previous evaluation code

```r
predicted_probs = predict(model2, newdata = adult_income_test, 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
predicted_income = rep(0, nrow(adult_income_test))
for (i in 1:length(predicted_income)) {
  if (predicted_probs[i] > 0.5) {
    predicted_income[i] = ">50K"
  } else {
    predicted_income[i] = "<=50K"
  }
}

# the default setting of glm() predicts >50K, so >50K is  positive and <=50K is negative.
tp = tn = fp = fn = 0
for (i in 1:length(predicted_income)) {
  if ((adult_income_test$income[i] == ">50K") && (predicted_income[i] == ">50K")) {
    tp = tp + 1
  } else if ((adult_income_test$income[i] == "<=50K") && (predicted_income[i] == "<=50K")) {
    tn = tn + 1
  } else if ((adult_income_test$income[i] == ">50K") && (predicted_income[i] == "<=50K")) {
    fn = fn + 1
  } else {
    fp = fp + 1
  }
}

(m = matrix(c(tp, fp, fn, tn), 2, 2, byrow = T))
```

```
##      [,1] [,2]
## [1,]  734  263
## [2,]  495 3508
```

```
(precision = tp / (tp + fp))
```

```
## [1] 0.7362086
```

```
(recall = tp / (tp + fn))
```

```
## [1] 0.5972335
```

```
(accuracy = (tp + tn) / sum(m))
```

```
## [1] 0.8484
```

So the result is slightly worse than previous. But the model is quite a lot simpler. Well, in learning you win some, you loose some.

C.1

Rejection sampling. Use the notation in the lecture notes. Assume working on range $[0, B]$. In the sheet, $B = 2$. Only need to know the shape of the target distribution, forget the normaliser. So use

$$q(x) = e^{-\lambda x}$$

and the proposal distribution is

$$p_{prop}(x) = 1/B$$

thus the best C is given by

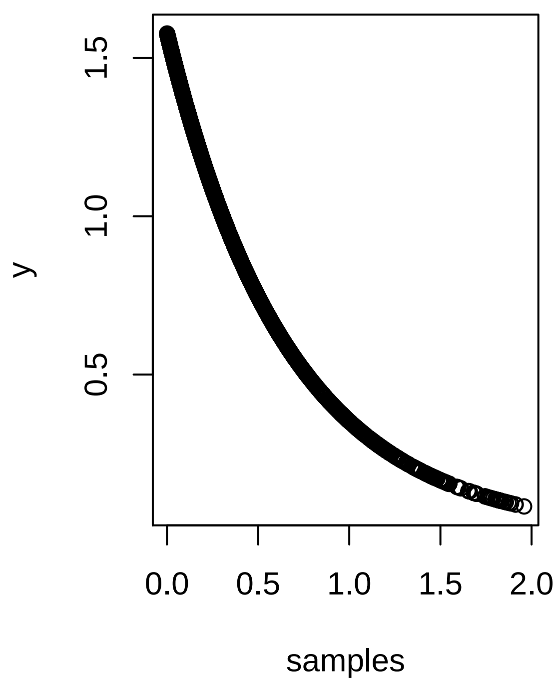
$$C \leq \min_x \frac{p_{prop}(x)}{q(x)} = \min_x \frac{e^{\lambda x}}{B} = \frac{1}{B}$$

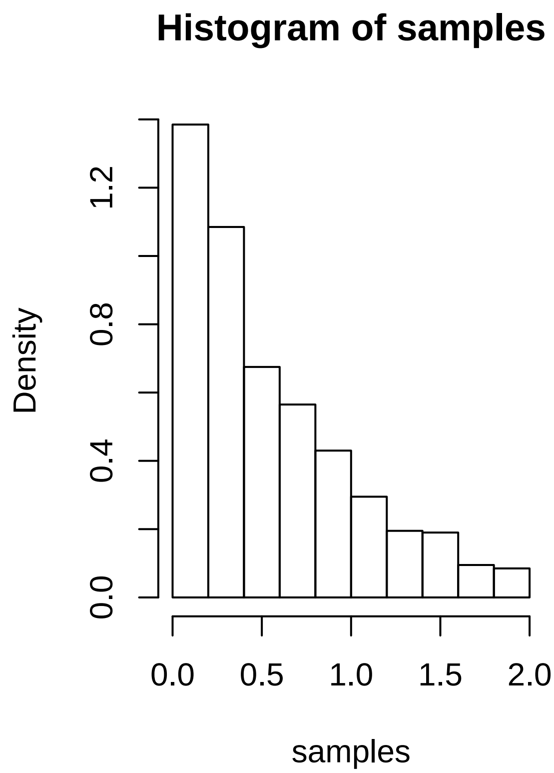
thus the rejection ration to use in the sampler is

$$\frac{Cq(x)}{p_{prop}(x)} = q(x)$$

```
#  upper bound on x
B = 2

# target distribution
pt = function(x) 1.5 * exp(-1.5 * x) / (1-exp(-1.5*B))

# target distribution, ignoring normaliser
q = function(x) exp(-1.5 * x)

# proposal distribution
p = 1.0/B

# start rejection sampling
samples = rep(0, 1000)
i = 1
repeat {
  if (i > 1000) break
  x = B * runif(1)
  ratio = q(x)
  u = runif(1)
  if (u < ratio) {
    samples[i] = x
    i = i + 1
```

```
  }
}

# plot to check
par(mfrow = c(1,2))
y = sapply(samples, pt)
plot(samples, y)
hist(samples, freq = F, breaks=10)
```





C.2

Inverse transform sampling. Assume working on range $[0, B]$. In the sheet, $B = 2$ or ∞. Cumulative distribution is

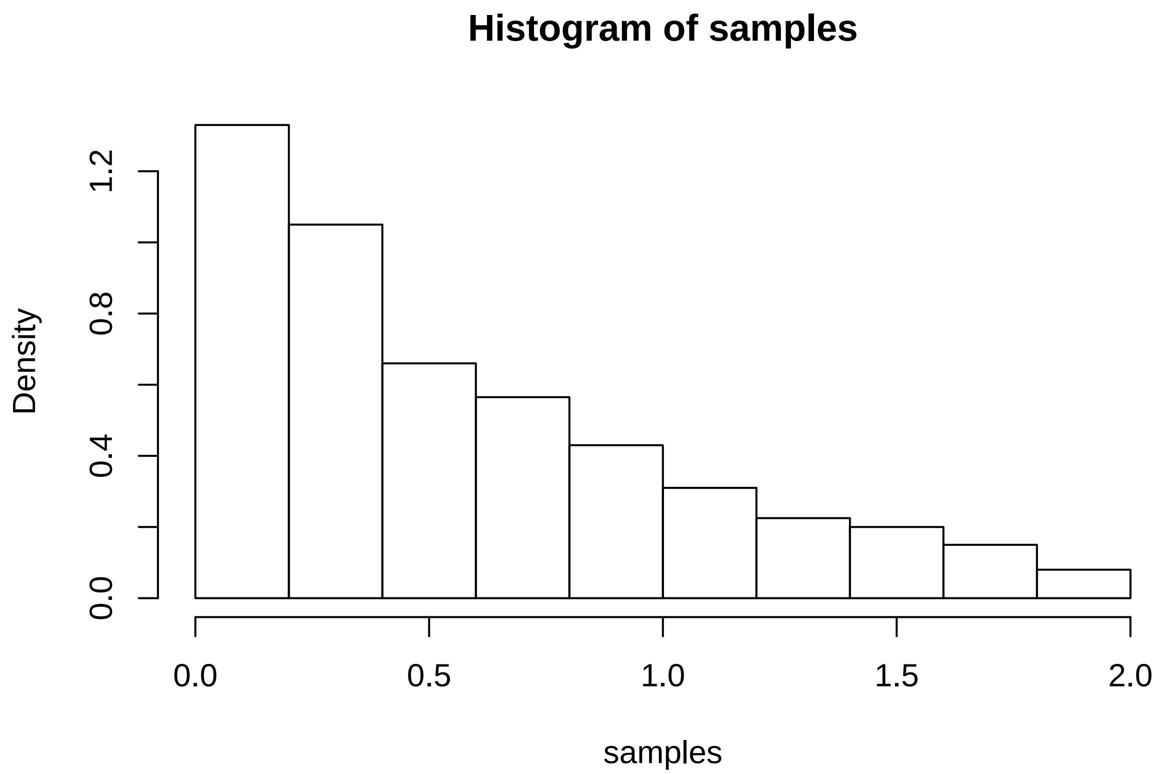
$$P(x) = \int_0^B \frac{1}{1 - e^{-\lambda B}} \lambda e^{-\lambda x} \mathrm{d}x = \frac{1 - e^{-\lambda x}}{1 - e^{-\lambda B}}$$

and therefore the quantile function is

$$Q(p) = \frac{-1}{\lambda} \log \left(1 - p(1 - e^{-\lambda B})\right)$$

```
B=2
q = function(u) log(1 - u*(1-exp(-1.5*B))) / (-1.5)

samples = rep(0, 1000)
for (i in 1:1000) {
```

```r
  u = runif(1)
  samples[i] = q(u)
}

hist(samples, freq = F, breaks=10)
```



C.3

Gibbs sampling

```r
cpt_c = c(0.5, 0.5)
cpt_s_given_c = matrix(c(0.5, 0.5, 0.9, 0.1), 2, 2, byrow = F)
cpt_r_given_c = matrix(c(0.8, 0.2, 0.2, 0.8), 2, 2, byrow = F)
cpt_w_given_sr = matrix(c(1, 0.1, 0.1, 0.01, 0, 0.9, 0.9, 0.99), 2, 4, byrow = T)

p_c_given_sr = function(s, r) {
  p = cpt_c * cpt_s_given_c[s + 1, ] * cpt_r_given_c[r + 1, ]
  return(p / sum(p))
}

p_s_given_crw = function(c, r, w) {
  if (r == 0) {
    ind = c(1, 2)
  } else if (r == 1) {
    ind = c(3, 4)
```

```
  }
  p = cpt_s_given_c[, c + 1] * cpt_w_given_sr[w + 1, ind]
  return(p / sum(p))
}

p_r_given_csw = function(c, s, w) {
  if (s == 0) {
    ind = c(1, 3)
  } else if (s == 1) {
    ind = c(2, 4)
  }
  p = cpt_r_given_c[, c + 1] * cpt_w_given_sr[w + 1, ind]
  return(p / sum(p))
}

p_w_given_sr = function(s, r) {
  if ((s == 0) && (r == 0)) {
    ind = 1
  } else if ((s == 1) && (r == 0)) {
    ind = 2
  } else if ((s == 0) && (r == 1)) {
    ind = 3
  } else {
    ind = 4
  }
  return(cpt_w_given_sr[, ind])
}

samples = matrix(0, 1000, 4)
colnames(samples) = c("C", "S", "R", "W")
samples[1, ] = 1 # initialize random samples
for (i in 2:1000) {

  # sample for C
  p = p_c_given_sr(samples[i - 1, "S"], samples[i - 1, "R"])
  u = runif(1)
  samples[i, "C"] = ifelse(u < p[1], 0, 1)

  # sample for S
  p = p_s_given_crw(samples[i, "C"], samples[i - 1, "R"], samples[i - 1, "W"])
  u = runif(1)
  samples[i, "S"] = ifelse(u < p[1], 0, 1)

  # sample for R
  p = p_r_given_csw(samples[i, "C"], samples[i, "S"], samples[i - 1, "W"])
  u = runif(1)
  samples[i, "R"] = ifelse(u < p[1], 0, 1)

  # sample for W
  p = p_w_given_sr(samples[i, "S"], samples[i, "R"])
  u = runif(1)
  samples[i, "W"] = ifelse(u < p[1], 0, 1)

```

```
}

data = as.data.frame(samples[-c(1:100), ])
(table(data[, c("W", "C")])/900)
```

```
##    C
## W           0         1
##   0 0.2577778 0.1500000
##   1 0.2677778 0.3244444
```

```
(table(data[, c("S", "R")])/900)
```

```
##    R
## S            0          1
##   0 0.33666667 0.37333333
##   1 0.20333333 0.08666667
```