Stat_projectReadtxt_1115

Group 5

11/15/2019

The procedures for analyzing data include (1) data cleaning and manipulation(2) first model (3)test hypothesis (4) multicollinearity (5)second model (6) model evaluation and (7)conclusions.

```
##(1) data cleaning and manipulation
## store data file with the variable name data
## data cleaning
## import library
library(stringr)
library(ggplot2)
library(dplyr)

##
## Attaching package: 'dplyr'

## The following objects are masked from 'package:stats':
##
##     filter, lag

## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union

library(ROCR)

## Loading required package: gplots

##
## Attaching package: 'gplots'

## The following object is masked from 'package:stats':
##
##     lowess

library(boot)
library(extrafont)

## Registering fonts with R

library(ggthemes)
library(ROCR)
library(caret)
```

```
## Loading required package: lattice

##
## Attaching package: 'lattice'

## The following object is masked from 'package:boot':
##
##     melanoma

library(plyr)

## -------------------------------------------------------------------------

## You have loaded plyr after dplyr - this is likely to cause problems.
## If you need functions from both plyr and dplyr, please load plyr first, th
en dplyr:
## library(plyr); library(dplyr)

## -------------------------------------------------------------------------

##
## Attaching package: 'plyr'

## The following objects are masked from 'package:dplyr':
##
##     arrange, count, desc, failwith, id, mutate, rename, summarise,
##     summarize
```

read data in

```
##(1) data cleaning and manipulation
##read in data
adult <- read.table("adult.data", sep = ",", header = FALSE)
```

check data dimension

```
##(1) data cleaning and manipulation
###check dimension of adult
dim(adult)[1]

## [1] 32561

dim(adult)[2]

## [1] 15
```

(1) data cleaning and manipulation

handle missing data and add header in

```
##(1) data cleaning and manipulation
## handle missing data and add header in
adult <- read.table("adult.data",
                       sep = ",",
```

```
                       header = FALSE,
                       na.strings = " ?")


colnames(adult) <- c("age", "wc", "wgt",
                        "edu", "edu_num",
                        "marital", "occup",
                        "rp", "race", "sex",
                        "c_gain", "c_loss",
                        "hours_w", "nc", "income")
```

(1) data cleaning and manipulation

check data set

```
##(1) data cleaning and manipulation
## check hander and check data
# adult
```

(1) data cleaning and manipulation

omit NA data

```
##(1) data cleaning and manipulation
#Remove all na value
adult <- na.omit(adult)
```

(1) data cleaning and manipulation

check data dimension after removing NA data

```
##(1) data cleaning and manipulation
##check dimension after remove na
dim(adult)[1]

## [1] 30162

dim(adult)[2]

## [1] 15

row.names(adult) <- 1:nrow(adult)
```

(1) data cleaning and manipulation

```
data<-adult
```

(1) data cleaning and manipulation

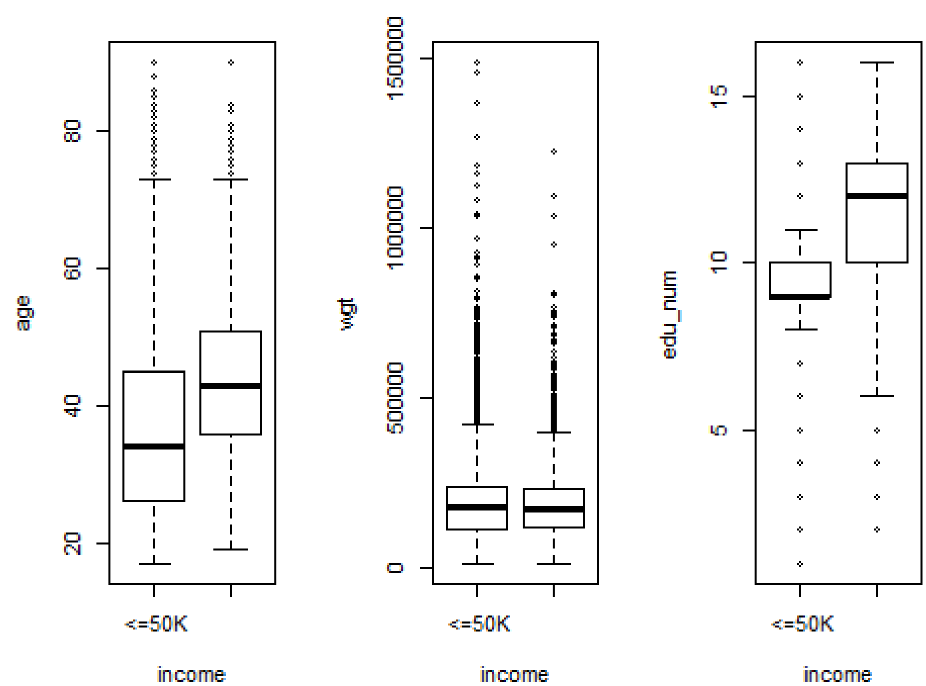
Attach data

```
##(1) data cleaning and manipulation
attach(data)
#data
```

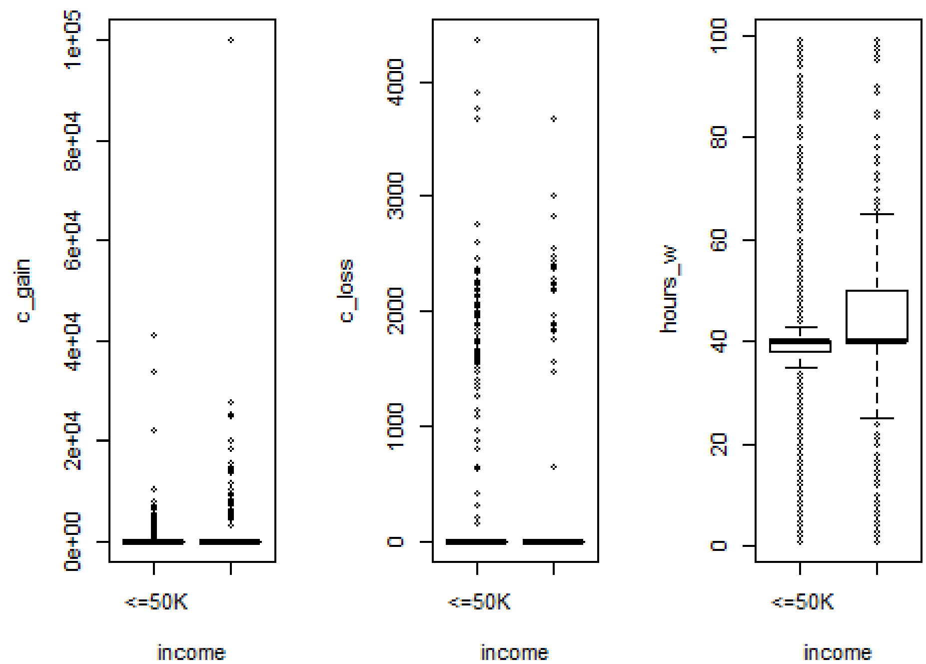
(1) data cleaning and manipulation

start to check six numerical variables and eight categorical variables

```
##(1) data cleaning and manipulation
## check six numerical variable
#Use box plot to see each numerical predictors vs. income (Figures 1 and 2)
##############
par(mfrow=c(1,3))
is.numeric(age)

## [1] TRUE

boxplot(age~income)
boxplot(wgt~income)

###############
boxplot(edu_num~income)
```



```r
###############
boxplot(c_gain~income)
boxplot(c_loss~income)
###############
boxplot(hours_w~income)
```

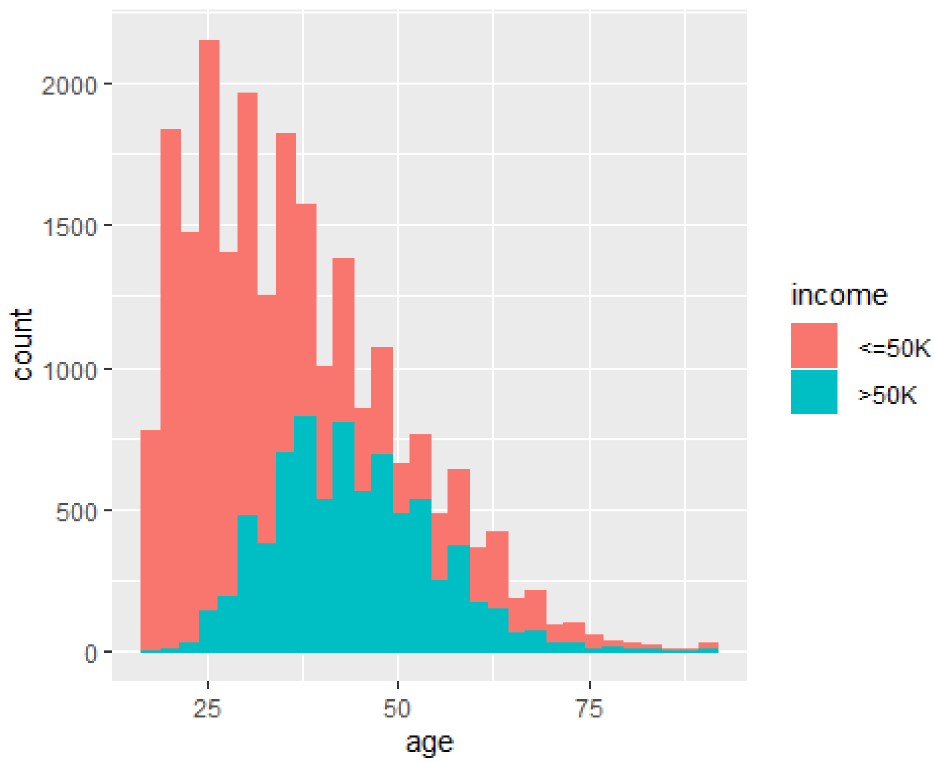



```r
###############
```

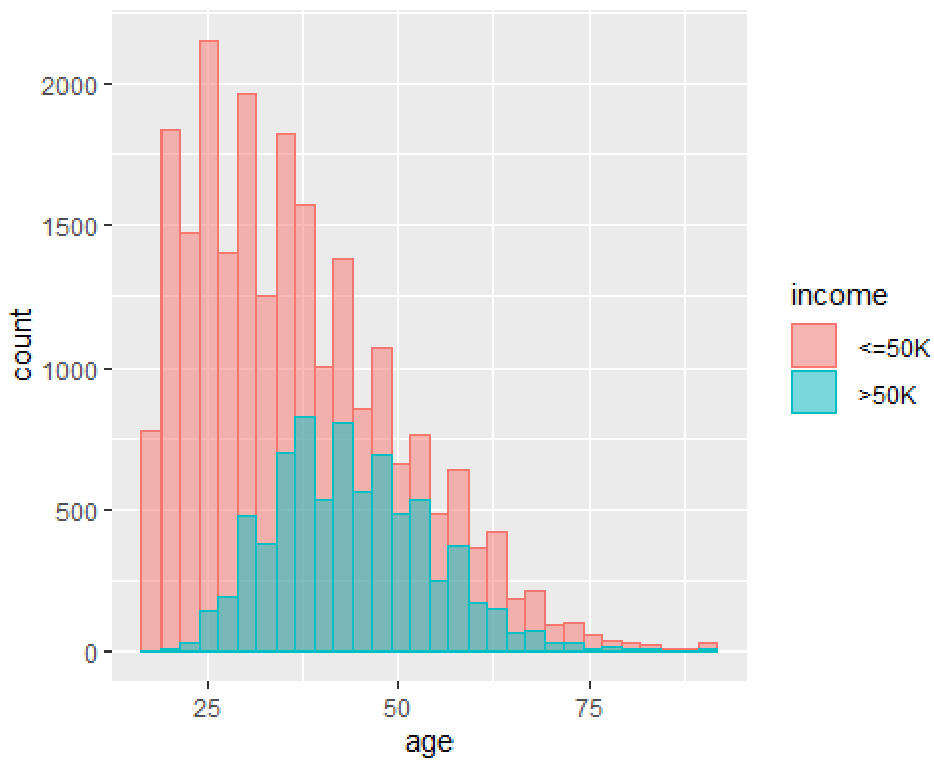
(1) data cleaning and manipulation

age effect understanding

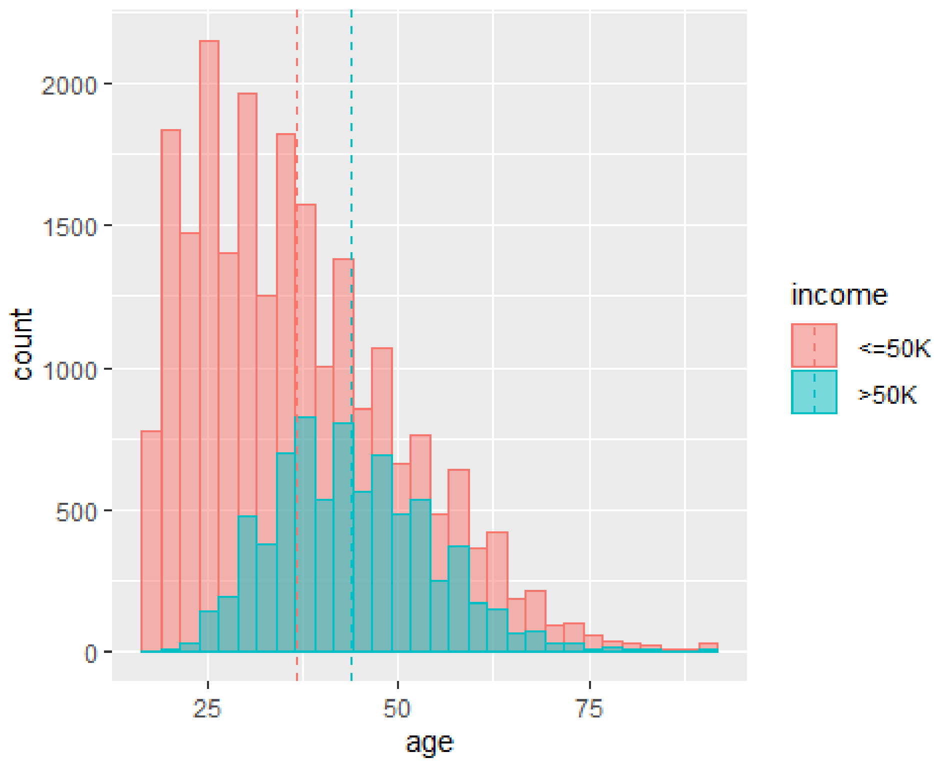
```r
##(1) data cleaning and manipulation
##Check age histogram colored by
library(plyr)
mu <- ddply(data, "income", summarise, grp.mean=mean(age))
head(mu)

##   income grp.mean
## 1  <=50K 36.60806
## 2   >50K 43.95911

# Change histogram plot fill colors by groups
ggplot(data, aes(x=age, fill=income, color=income)) +
  geom_histogram(position="identity")

## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



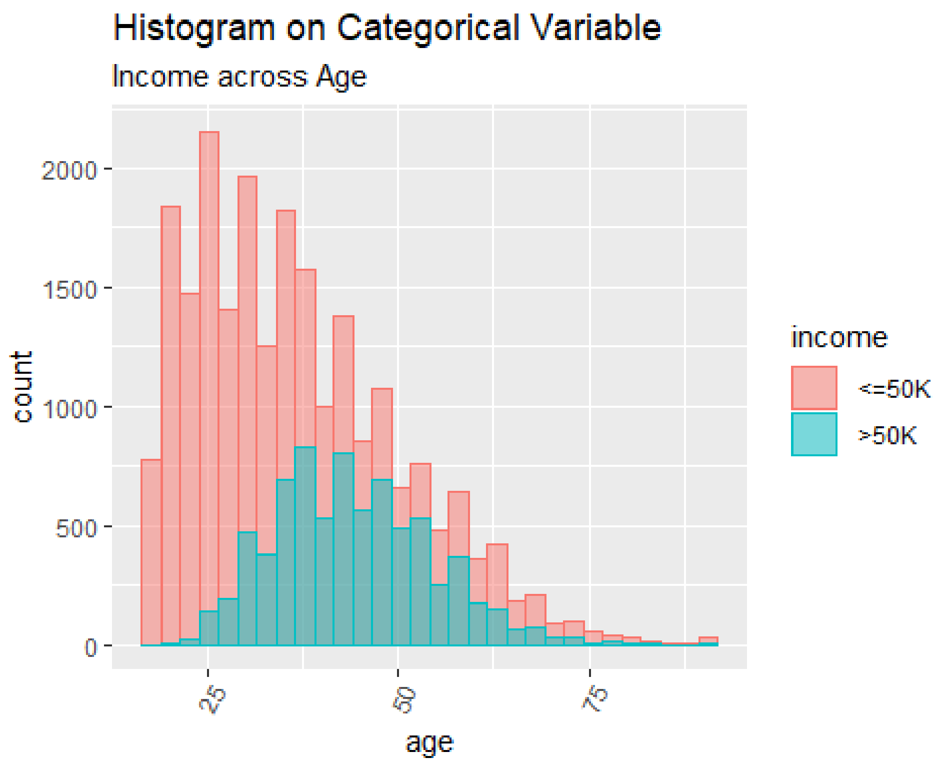

```r
# Use semi-transparent fill
p<-ggplot(data, aes(x=age, fill=income, color=income)) +
  geom_histogram(position="identity", alpha=0.5)
p

## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```




```
# Add mean lines
p+geom_vline(data=mu, aes(xintercept=grp.mean, color=income),
             linetype="dashed")

## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```




```
p+theme(axis.text.x = element_text(angle=65, vjust=0.6)) +
  labs(title="Histogram on Categorical Variable",
       subtitle="Income across Age")

## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



(1) data cleaning and manipulation ## capital-gain and capital-loss quantile checking

```
##(1) data cleaning and manipulation
###many zero in capital_gain and capital_loss
summary(data$c_gain)

##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##       0       0       0    1092       0   99999

summary(data$c_loss)

##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##    0.00    0.00    0.00   88.37    0.00 4356.00
```

(1) data cleaning and manipulation

check zero counts of capital-gain

```
##(1) data cleaning and manipulation
sum(data$c_gain == 0)/length(data$c_gain)

## [1] 0.9158544
```

(1) data cleaning and manipulation

check zero counts of capital-loss

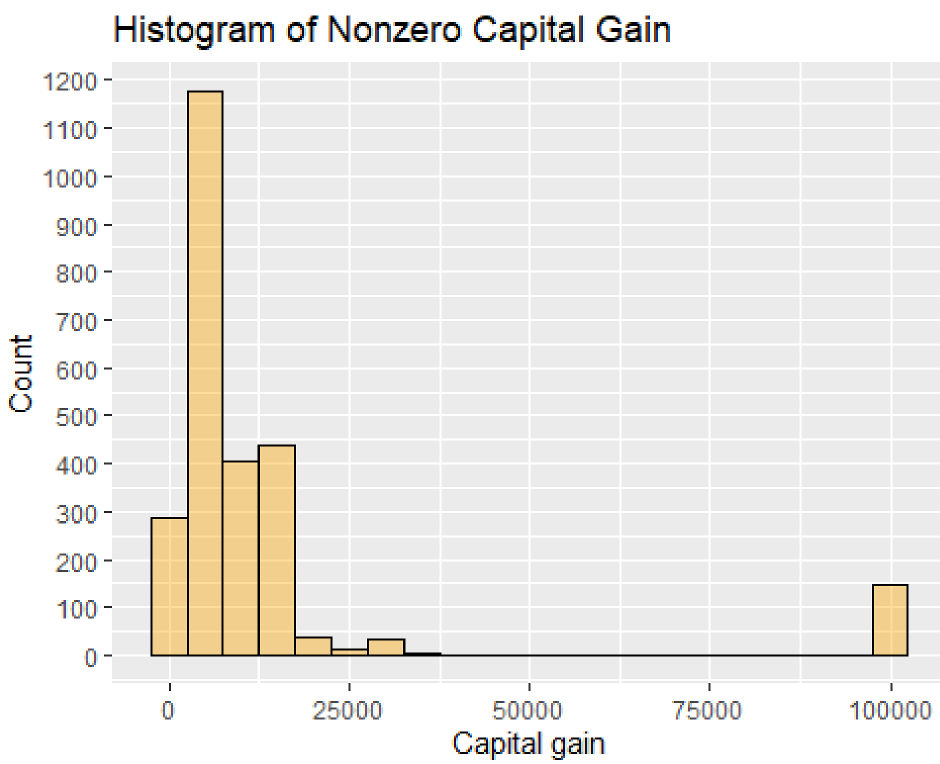
```
##(1) data cleaning and manipulation
sum(data$c_loss == 0)/length(data$c_loss)
```

```
## [1] 0.9526888
```

(1) data cleaning and manipulation

check non-zero of capital-gain

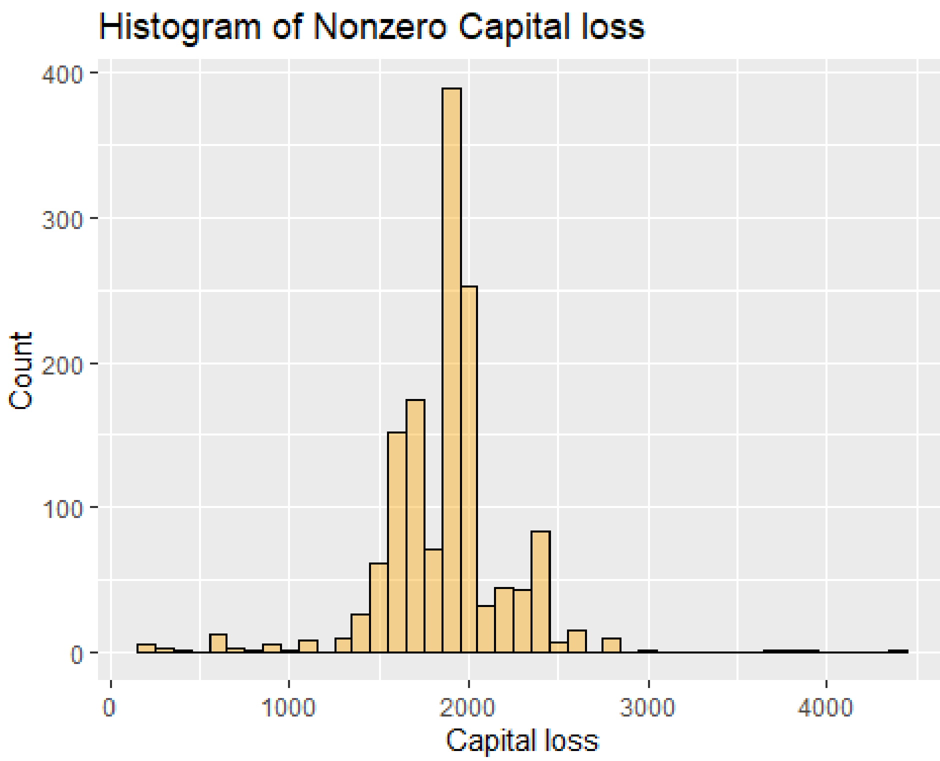
```r
##(1) data cleaning and manipulation
###check capital_gain > 0 histogram distribution
df <- data[data$c_gain > 0, ]

ggplot(data = df,
       aes(x = df$c_gain)) +
  geom_histogram(binwidth = 5000,
                 colour = "black",
                 fill = "orange",
                 alpha = 0.4) +
  scale_y_continuous(breaks = seq(0, 4000, 100)) +
  labs(x = "Capital gain", y = "Count") +
  ggtitle("Histogram of Nonzero Capital Gain")
```



(1) data cleaning and manipulation ## check non-zero of capital-loss

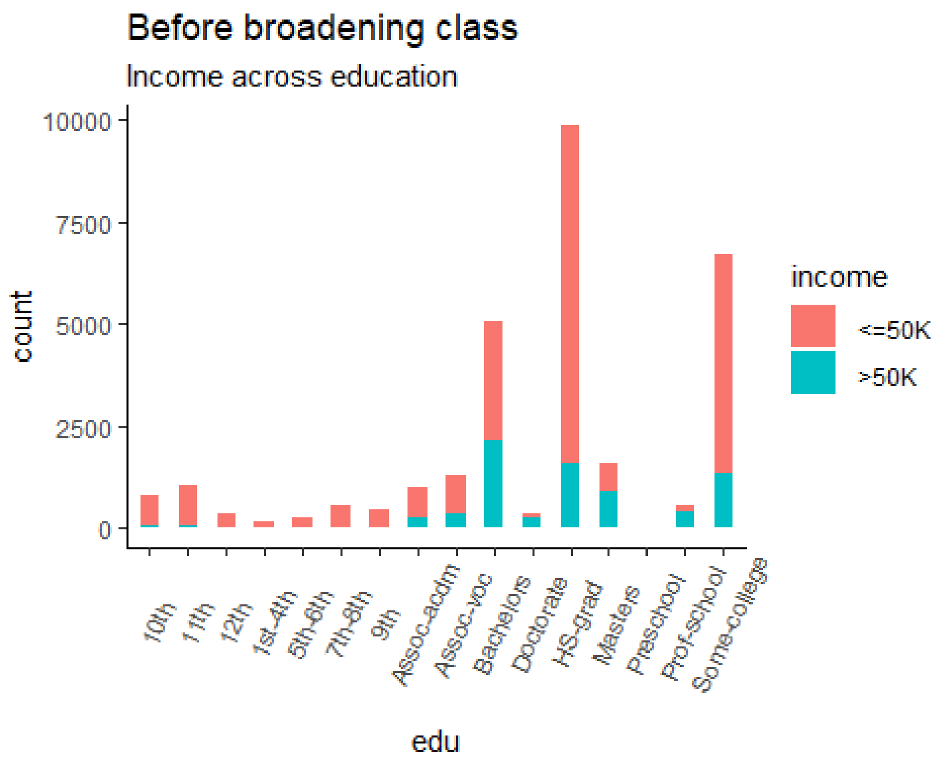
```r
##(1) data cleaning and manipulation
###check capital_loss >0 histogram distribution
df <- data[data$c_loss > 0, ]

ggplot(data = df,
       aes(x = df$c_loss)) +
```

```r
  geom_histogram(binwidth = 100,
                 colour = "black",
                 fill = "orange",
                 alpha = 0.4) +
  scale_y_continuous(breaks = seq(0, 4000, 100)) +
  labs(x = "Capital loss", y = "Count") +
  ggtitle("Histogram of Nonzero Capital loss")
```



(1) data cleaning and manipulation ## broaden classess

```r
##(1) data cleaning and manipulation
#Before broaden education
theme_set(theme_classic())

# Histogram on a Categorical variable
g <- ggplot(data, aes(edu))
g + geom_bar(aes(fill=income), width = 0.5) +
  theme(axis.text.x = element_text(angle=65, vjust=0.6)) +
  labs(title="Before broadening class",
       subtitle="Income across education")
```



(1) data cleaning and manipulation ## check classes of education before broadening

```
##(1) data cleaning and manipulation
##summarize the classes of education
summary(data$edu)
```

```
##          10th          11th          12th       1st-4th       5th-6th
##           820          1048           377           151           288
##       7th-8th           9th    Assoc-acdm     Assoc-voc     Bachelors
##           557           455          1008          1307          5044
##     Doctorate       HS-grad       Masters     Preschool   Prof-school
##           375          9840          1627            45           542
##  Some-college
##          6678
```

(1) data cleaning and manipulation

trim space

```
##(1) data cleaning and manipulation
###trim space
data$edu <- trimws(data$edu)
```

(1) data cleaning and manipulation

use gsub() to group class

```
##(1) data cleaning and manipulation
###combine high school below or 12th together
data$edu <-gsub('^12th', '<HS', data$edu)
data$edu <-gsub('^10th', '<HS', data$edu)
data$edu <-gsub('^11th', '<HS', data$edu)
data$edu <-gsub('^1st-4th', '<HS', data$edu)
data$edu <-gsub('^5th-6th', '<HS', data$edu)
data$edu <-gsub('^7th-8th', '<HS', data$edu)
data$edu <-gsub('^9th', '<HS', data$edu)
data$edu <-gsub('^Preschool', '<HS', data$edu)
data$edu <-gsub('^Assoc-acdm', 'Assoc', data$edu)
data$edu <-gsub('^Assoc-voc', 'Assoc', data$edu)
data$edu <-as.factor(data$edu)
```

(1) data cleaning and manipulation

check classes after broadening

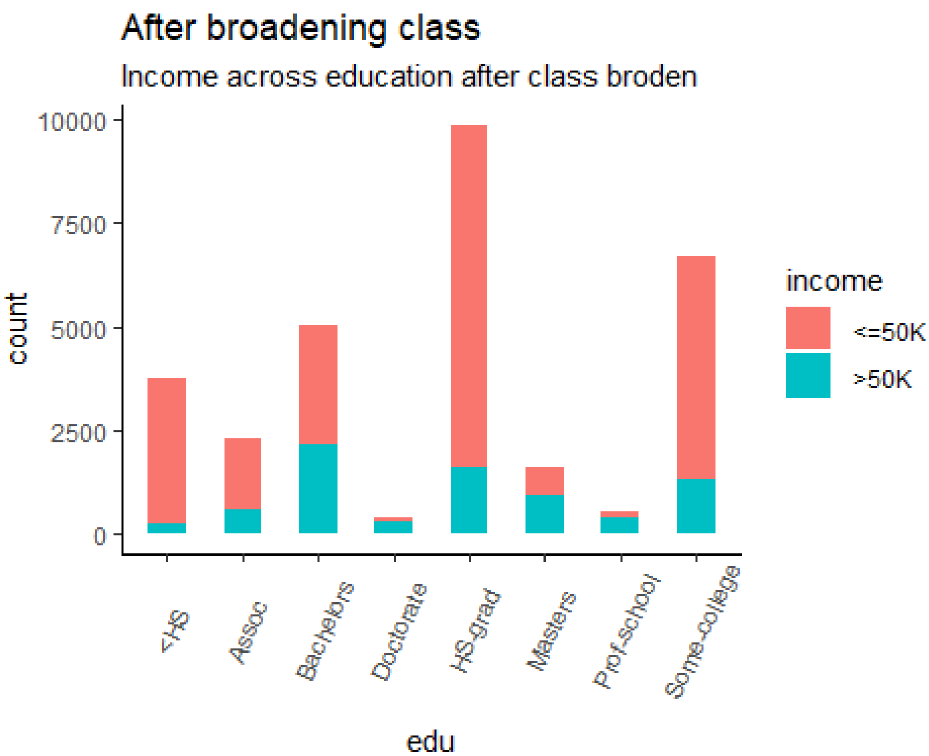
```
##(1) data cleaning and manipulation
summary(data$edu)
```

```
##          <HS        Assoc    Bachelors    Doctorate      HS-grad
##         3741         2315         5044          375         9840
##      Masters  Prof-school Some-college
##         1627          542         6678
```

(1) data cleaning and manipulation

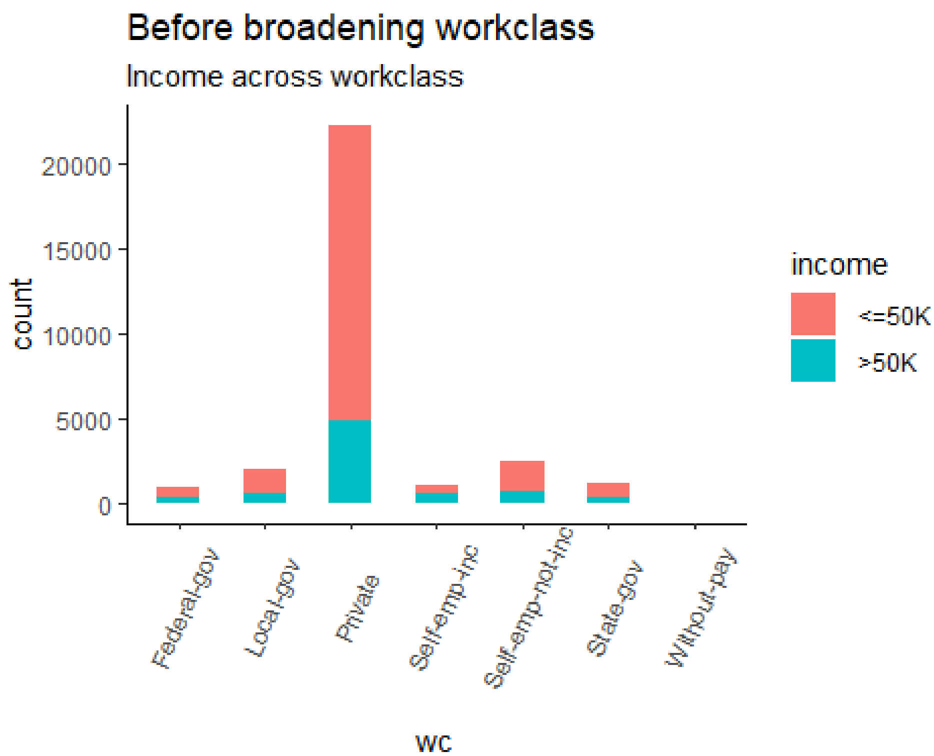
check the plot after broadening

```
##(1) data cleaning and manipulation
#after broaden education
theme_set(theme_classic())

# Histogram on a Categorical variable
g <- ggplot(data, aes(edu))
g + geom_bar(aes(fill=income), width = 0.5) +
  theme(axis.text.x = element_text(angle=65, vjust=0.6)) +
  labs(title="After broadening class",
       subtitle="Income across education after class broden")
```



(1) data cleaning and manipulation ## check work class before broadening

```
##(1) data cleaning and manipulation
###before broaden work class
theme_set(theme_classic())

# Histogram on a Categorical variable
g <- ggplot(data, aes(wc))
g + geom_bar(aes(fill=income), width = 0.5) +
  theme(axis.text.x = element_text(angle=65, vjust=0.6)) +
  labs(title="Before broadening workclass",
       subtitle="Income across workclass")
```



summary of workclass before broadening

```
##(1) data cleaning and manipulation
summary(data$wc)
```

```
##       Federal-gov          Local-gov       Never-worked            Private
##               943               2067                  0              22286
##      Self-emp-inc   Self-emp-not-inc          State-gov        Without-pay
##              1074               2499               1279                 14
```

trim space

```
##(1) data cleaning and manipulation
data$wc <- trimws(data$wc)
```

broadening work class based on government, other and self-employed

```
##(1) data cleaning and manipulation
levels(data$wc)[1] <- 'Unknown'

# combine into Sele-Employed job
data$wc <- gsub('^Self-emp-inc', 'Self-Employed', data$wc)
data$wc <- gsub('^Self-emp-not-inc', 'Self-Employed', data$wc)

# combine into Other/Unknown
data$wc <- gsub('^Never-worked', 'Other', data$wc)
data$wc <- gsub('^Without-pay', 'Other', data$wc)
```

```r
data$wc <- gsub('^Other', 'Others', data$wc)
data$wc <- gsub('^Unknown', 'Other', data$wc)

# combine into Government job
data$wc <- gsub('^Federal-gov', 'Government', data$wc)
data$wc <- gsub('^Local-gov', 'Government', data$wc)
data$wc <- gsub('^State-gov', 'Government', data$wc)
```

factor workclass

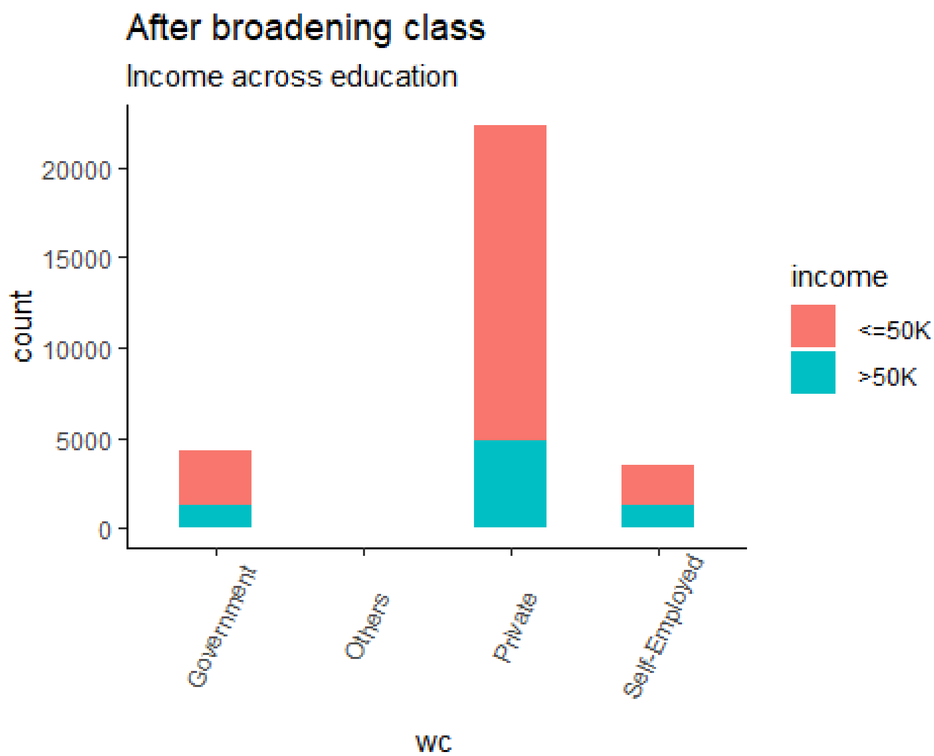
```r
##(1) data cleaning and manipulation
data$wc <- as.factor(data$wc)
```

check classes after broadening

```r
##(1) data cleaning and manipulation
summary(data$wc)

##    Government        Others       Private Self-Employed
##          4289            14         22286          3573
```

check bar plot after broadening

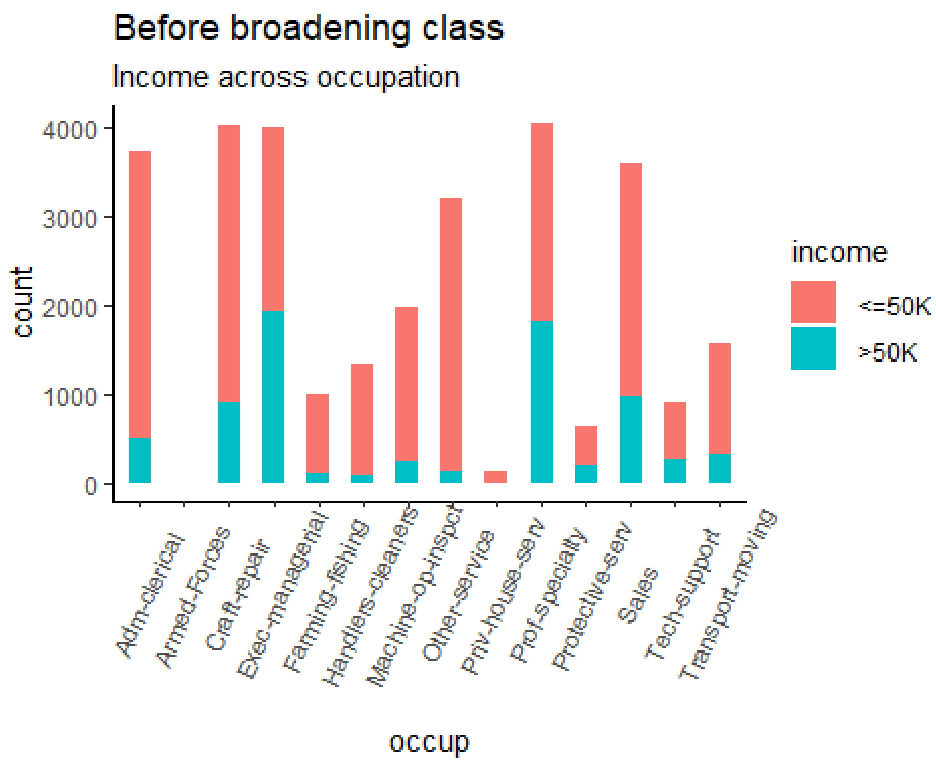
```r
##(1) data cleaning and manipulation
theme_set(theme_classic())

# Histogram on a Categorical variable
g <- ggplot(data, aes(wc))
g + geom_bar(aes(fill=income), width = 0.5) +
  theme(axis.text.x = element_text(angle=65, vjust=0.6)) +
  labs(title="After broadening class",
       subtitle="Income across education")
```



occupation

before broadening

```
##(1) data cleaning and manipulation
#before broadening class
theme_set(theme_classic())

# Histogram on a Categorical variable
g <- ggplot(data, aes(occup))
g + geom_bar(aes(fill=income), width = 0.5) +
  theme(axis.text.x = element_text(angle=65, vjust=0.6)) +
  labs(title="Before broadening class",
       subtitle="Income across occupation")
```



check summary of occupation before broadening

```
##(1) data cleaning and manipulation
summary(data$occup)
```

```
##       Adm-clerical       Armed-Forces       Craft-repair
##               3721                  9               4030
##    Exec-managerial    Farming-fishing  Handlers-cleaners
##               3992                989               1350
##  Machine-op-inspct      Other-service    Priv-house-serv
##               1966               3212                143
##     Prof-specialty    Protective-serv              Sales
##               4038                644               3584
##       Tech-support   Transport-moving
##                912               1572
```

group the occupation such as blue-collar or white-collar, etc

```
##(1) data cleaning and manipulation
data$occup <- trimws(data$occup)
data$occup <- gsub('^Adm-clerical','Administrator',data$occup)
data$occup <- gsub('^Armed-Forces','Military',data$occup)
data$occup <- gsub('^Craft-repair','Blue-Collar',data$occup)
data$occup <- gsub('^Exec-managerial','White-Collar',data$occup)
data$occup <- gsub('^Farming-fishing','Blue-Collar',data$occup)
data$occup <- gsub('^Handlers-cleaners','Blue-Collar',data$occup)
data$occup <- gsub('^Machine-op-inspct','Blue-Collar',data$occup)
data$occup <- gsub('^Other-service','Service',data$occup)
```

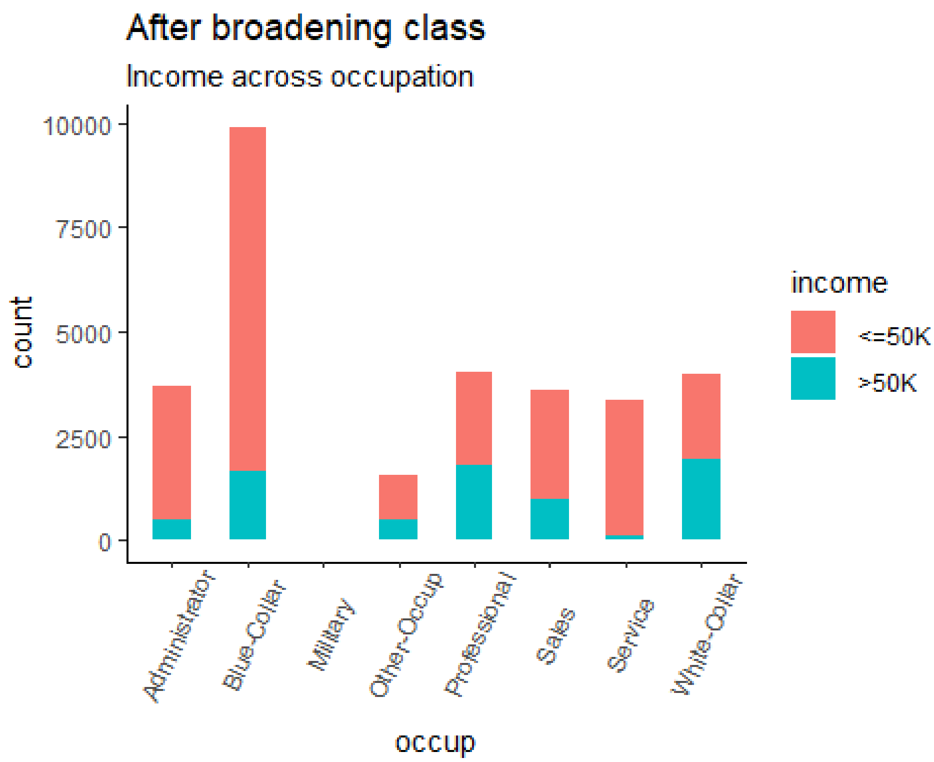
```
data$occup <- gsub('^Priv-house-serv','Service',data$occup)
data$occup <- gsub('^Prof-specialty','Professional',data$occup)
data$occup <- gsub('^Protective-serv','Other-Occup',data$occup)
data$occup <- gsub('^Sales','Sales',data$occup)
data$occup <- gsub('^Tech-support','Other-Occup',data$occup)
data$occup <- gsub('^Transport-moving','Blue-Collar',data$occup)
data$occup <-as.factor(data$occup)
```

check the classes after broadening

```
##(1) data cleaning and manipulation
summary(data$occup)

## Administrator   Blue-Collar      Military   Other-Occup  Professional
##          3721          9907             9          1556          4038
##         Sales       Service  White-Collar
##          3584          3355          3992
```

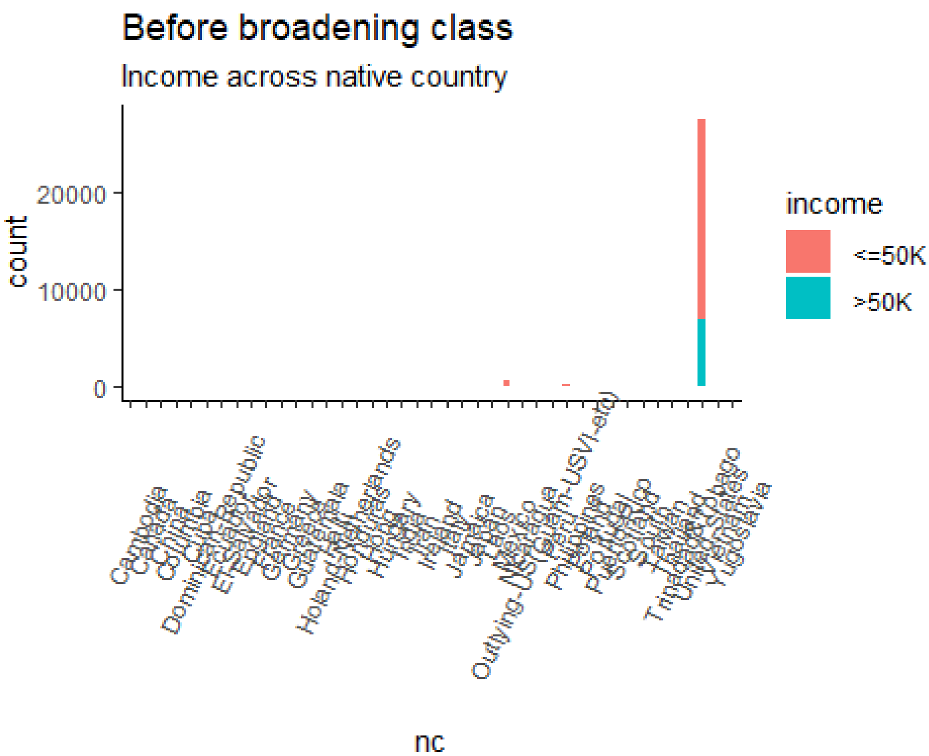
check bar plot after broadening

```
##(1) data cleaning and manipulation
theme_set(theme_classic())

# Histogram on a Categorical variable
g <- ggplot(data, aes(occup))
g + geom_bar(aes(fill=income), width = 0.5) +
  theme(axis.text.x = element_text(angle=65, vjust=0.6)) +
  labs(title="After broadening class",
       subtitle="Income across occupation")
```



check the class before broadening

```r
##(1) data cleaning and manipulation
###before broadening class
theme_set(theme_classic())

# Histogram on a Categorical variable
g <- ggplot(data, aes(nc))
g + geom_bar(aes(fill=income), width = 0.5) +
  theme(axis.text.x = element_text(angle=65, vjust=0.6)) +
  labs(title="Before broadening class",
       subtitle="Income across native country")
```



group countries based on geo location

```r
##(1) data cleaning and manipulation
Asia_East <- c(" Cambodia", " China", " Hong", " Laos", " Thailand",
               " Japan", " Taiwan", " Vietnam")

Asia_Central <- c(" India", " Iran")

Central_America <- c(" Cuba", " Guatemala", " Jamaica", " Nicaragua",
                     " Puerto-Rico",  " Dominican-Republic", " El-Salvador",
                     " Haiti", " Honduras", " Mexico", " Trinadad&Tobago")

South_America <- c(" Ecuador", " Peru", " Columbia")


Europe_West <- c(" England", " Germany", " Holand-Netherlands", " Ireland",
                 " France", " Greece", " Italy", " Portugal", " Scotland")

Europe_East <- c(" Poland", " Yugoslavia", " Hungary")
```

mutate to the column as nc

```r
##(1) data cleaning and manipulation
data <- mutate(data,
       nc = ifelse(nc %in% Asia_East, " East-Asia",
                ifelse(nc %in% Asia_Central, " Central-Asia",
                ifelse(nc %in% Central_America, " Central-America",
```

```
                ifelse(nc %in% South_America, " South-America",
                ifelse(nc %in% Europe_West, " Europe-West",
                ifelse(nc %in% Europe_East, " Europe-East",
                ifelse(nc == " United-States", " United-States",
                       " Outlying-US" ))))))))
data$nc <-as.factor(data$nc)
```

factor native country

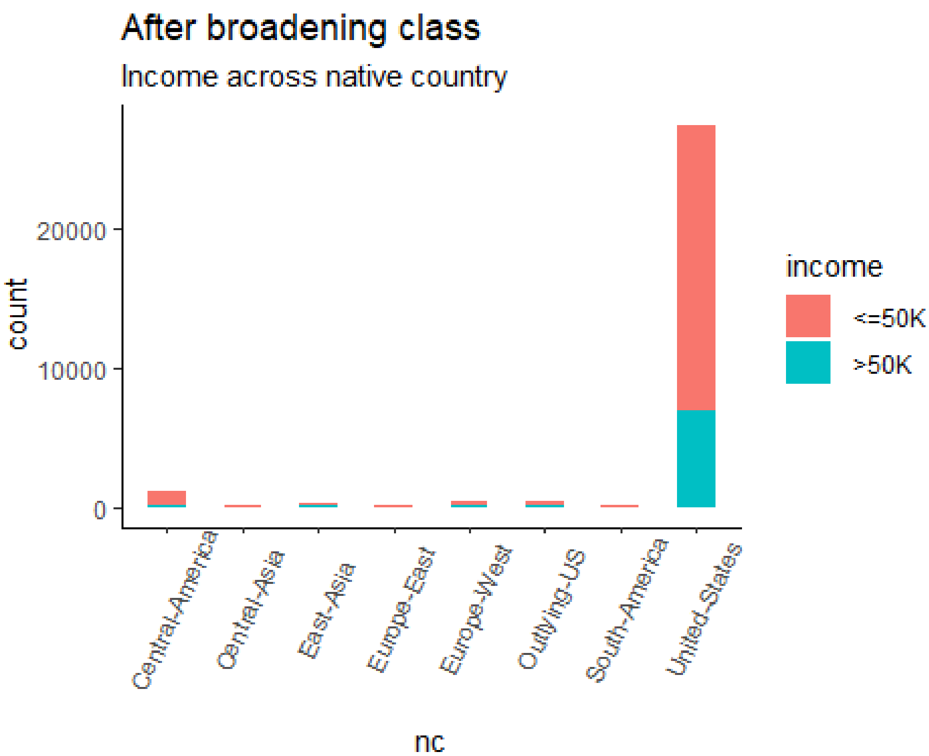
```
##(1) data cleaning and manipulation
data$nc <- factor(data$nc, ordered = FALSE)
```

summary of data after broadening

```
##(1) data cleaning and manipulation
summary(data$nc)

##  Central-America    Central-Asia       East-Asia     Europe-East
##             1226             142             304              85
##      Europe-West     Outlying-US   South-America   United-States
##              408             380             113           27504
```

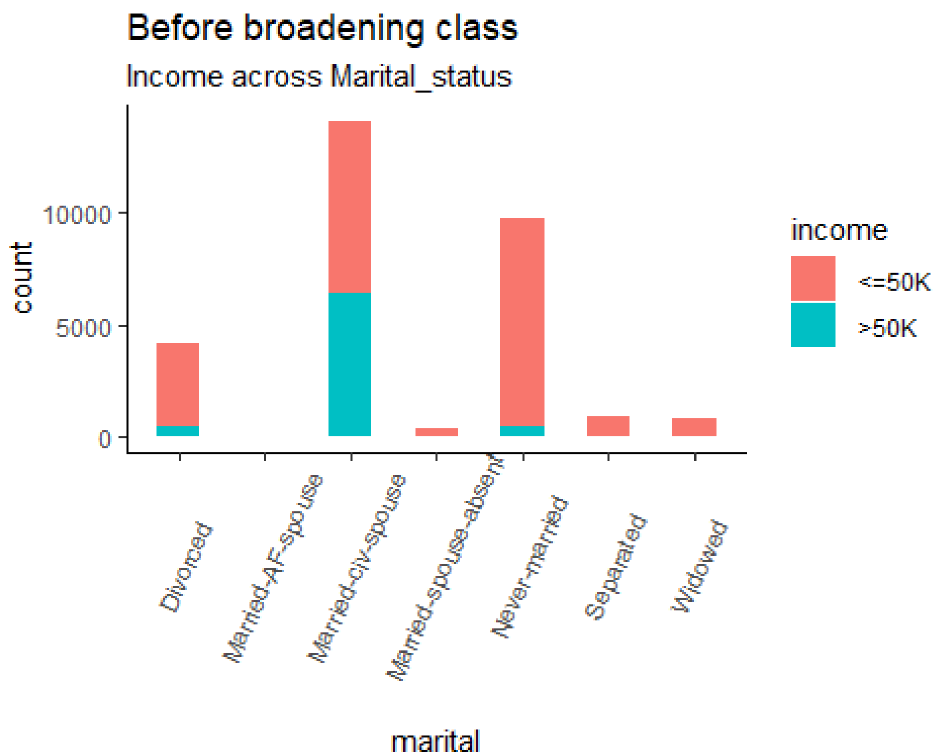
check the plot after broadening

```
##(1) data cleaning and manipulation
theme_set(theme_classic())

# Histogram on a Categorical variable
g <- ggplot(data, aes(nc))
g + geom_bar(aes(fill=income), width = 0.5) +
  theme(axis.text.x = element_text(angle=65, vjust=0.6)) +
  labs(title="After broadening class",
       subtitle="Income across native country")
```



broaden the class of marital

```r
##(1) data cleaning and manipulation
###before brodening class: martial_status
theme_set(theme_classic())

# Histogram on a Categorical variable
g <- ggplot(data, aes(marital))
g + geom_bar(aes(fill=income), width = 0.5) +
  theme(axis.text.x = element_text(angle=65, vjust=0.6)) +
  labs(title="Before broadening class",
       subtitle="Income across Marital_status")
```



check summary of marital

```
##(1) data cleaning and manipulation
summary(data$marital)
```

```
##              Divorced     Married-AF-spouse    Married-civ-spouse
##                  4214                    21                 14065
##  Married-spouse-absent        Never-married             Separated
##                   370                  9726                   939
##               Widowed
##                   827
```

trim space

```
##(1) data cleaning and manipulation
data$marital <- trimws(data$marital)
```

broadening the classes

```
##(1) data cleaning and manipulation
# combine same group into martial status -group married
data$marital <- gsub('^Married-AF-spouse', 'Married', data$marital)
data$marital <- gsub('^Married-civ-spouse', 'Married', data$marital)
data$marital <- gsub('^Married-spouse-absent', 'Married', data$marital)
###change to a short name
data$marital <- gsub('^Never-married', 'single', data$marital)
data$marital <-as.factor(data$marital)
```

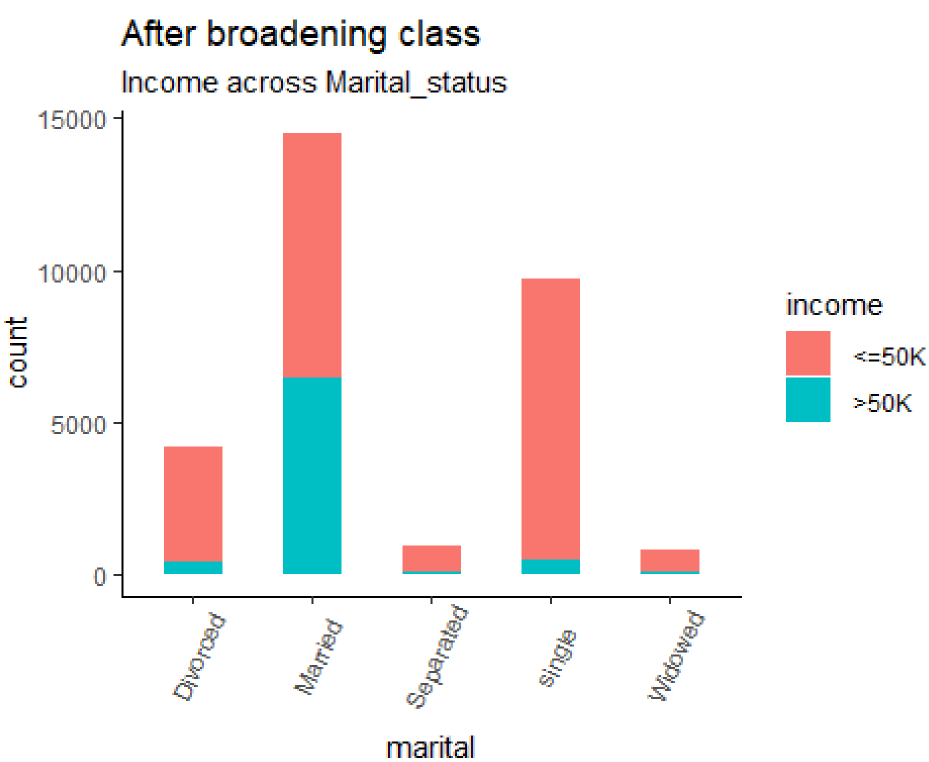
check summary after broadening

```r
##(1) data cleaning and manipulation
summary(data$marital)
```

```
##   Divorced    Married Separated     single    Widowed 
##       4214      14456        939       9726        827
```

check the bar plot after broadening

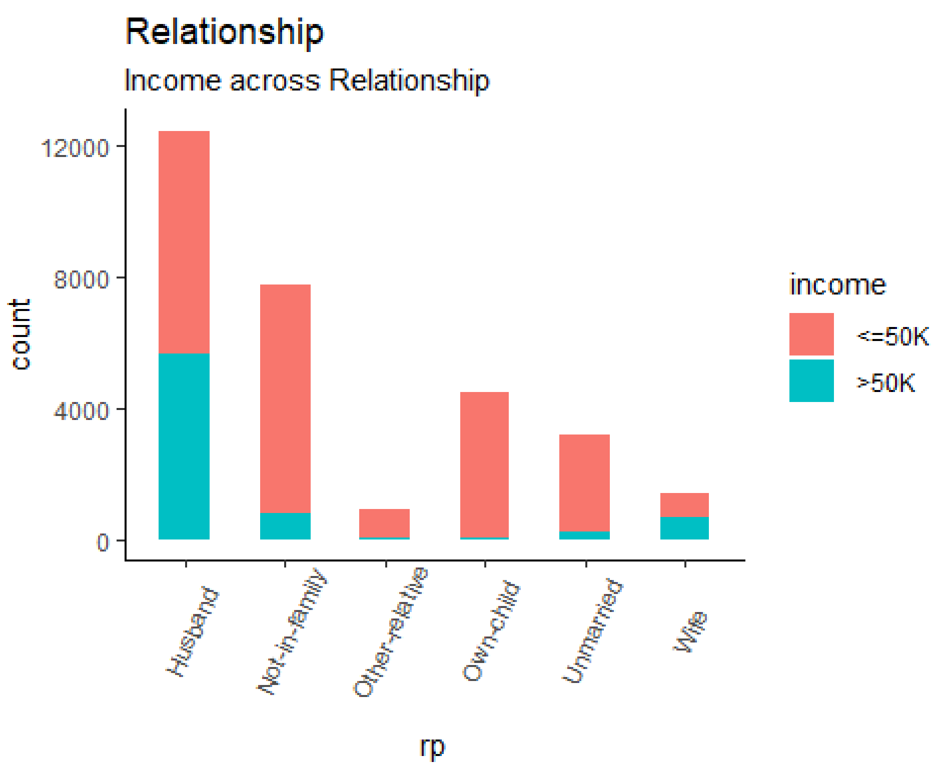
```r
##(1) data cleaning and manipulation
theme_set(theme_classic())

# Histogram on a Categorical variable
g <- ggplot(data, aes(marital))
g + geom_bar(aes(fill=income), width = 0.5) +
  theme(axis.text.x = element_text(angle=65, vjust=0.6)) +
  labs(title="After broadening class",
       subtitle="Income across Marital_status")
```



check relationship - no change. Undestand the Husband/Wife people have more income > 50K

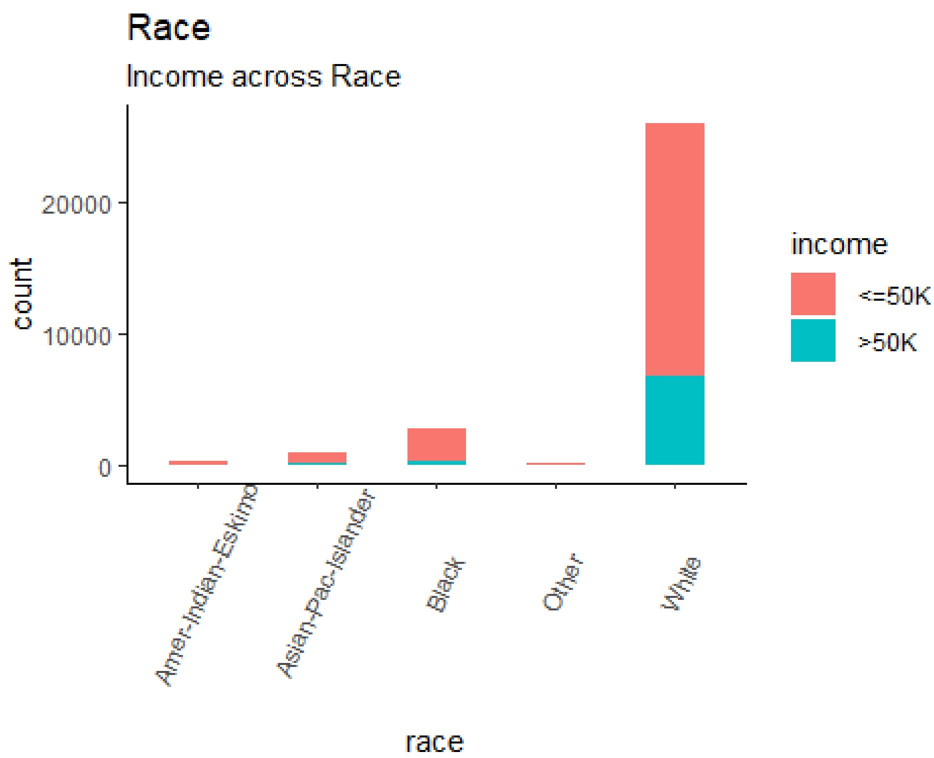
```r
##(1) data cleaning and manipulation
theme_set(theme_classic())

# Histogram on a Categorical variable
g <- ggplot(data, aes(rp))
g + geom_bar(aes(fill=income), width = 0.5) +
  theme(axis.text.x = element_text(angle=65, vjust=0.6)) +
```

```
  labs(title="Relationship",
       subtitle="Income across Relationship")
```



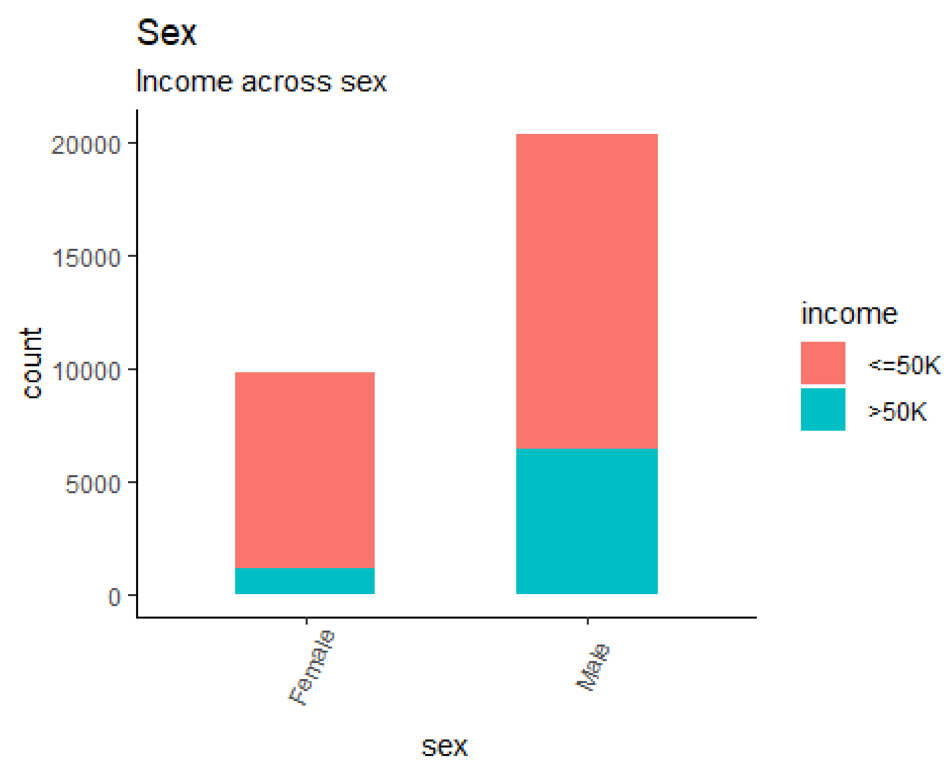
Check race- no change. Understand White has greater $50K income

```
##(1) data cleaning and manipulation
theme_set(theme_classic())

# Histogram on a Categorical variable
g <- ggplot(data, aes(race))
g + geom_bar(aes(fill=income), width = 0.5) +
  theme(axis.text.x = element_text(angle=65, vjust=0.6)) +
  labs(title="Race",
       subtitle="Income across Race")
```



chekc Sex and no change. Understand male has greater $50K than female

```
##(1) data cleaning and manipulation
##sex
theme_set(theme_classic())

# Histogram on a Categorical variable
g <- ggplot(data, aes(sex))
g + geom_bar(aes(fill=income), width = 0.5) +
  theme(axis.text.x = element_text(angle=65, vjust=0.6)) +
  labs(title="Sex",
       subtitle="Income across sex")
```



##finish checking all numerical and categorical variables

```
##(1) data cleaning and manipulation
attach(data)

## The following objects are masked from data (pos = 3):
##
##     age, c_gain, c_loss, edu, edu_num, hours_w, income, marital,
##     nc, occup, race, rp, sex, wc, wgt
```

check dimension again

```
##(1) data cleaning and manipulation
dim(data)[1]

## [1] 30162
```

null capital-gain, capital-loss due to skewed data

```
##(1) data cleaning and manipulation
data$c_gain <- NULL
data$c_loss <- NULL
attach(data)

## The following objects are masked from data (pos = 3):
##
##     age, edu, edu_num, hours_w, income, marital, nc, occup, race,
##     rp, sex, wc, wgt
```

```
## The following objects are masked from data (pos = 4):
##
##     age, edu, edu_num, hours_w, income, marital, nc, occup, race,
##     rp, sex, wc, wgt
```

check the dimension of data again

```
##(1) data cleaning and manipulation
dim(data)[2]

## [1] 13
```

check level

```
##(1) data cleaning and manipulation
##check coding scheme
contrasts(income)

##        >50K
##  <=50K    0
##  >50K     1
```

drop capital-gain, capital-loss, native-country and fnlwgt. The predictors(14) reduced to predictors (10)

First Model

```
## (2) first model
###drop c_gain, c_loss, nc, and wgt for regression - first model
m2 <- glm(income ~age+wc+edu_num+occup+sex+hours_w+ edu +rp + marital + race,
family = "binomial", data = data)
summary(m2)

##
## Call:
## glm(formula = income ~ age + wc + edu_num + occup + sex + hours_w +
##     edu + rp + marital + race, family = "binomial", data = data)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.7209  -0.5889  -0.2297  -0.0229   3.3771
##
## Coefficients:
##                            Estimate Std. Error z value Pr(>|z|)
## (Intercept)               -7.471918   0.405648 -18.420  < 2e-16 ***
## age                        0.028390   0.001591  17.844  < 2e-16 ***
## wcOthers                 -12.065173 119.636904  -0.101 0.919671
## wcPrivate                  0.096513   0.050212   1.922 0.054592 .
## wcSelf-Employed           -0.214622   0.064835  -3.310 0.000932 ***
## edu_num                    0.213156   0.043430   4.908 9.20e-07 ***
## occupBlue-Collar          -0.247904   0.069482  -3.568 0.000360 ***
```

```
## occupMilitary           -0.265374   1.296768  -0.205 0.837852    
## occupOther-Occup         0.498761   0.088834   5.615 1.97e-08 ***
## occupProfessional        0.461820   0.076333   6.050 1.45e-09 ***
## occupSales               0.281975   0.077174   3.654 0.000258 ***
## occupService            -1.013563   0.112069  -9.044  < 2e-16 ***
## occupWhite-Collar        0.806173   0.072285  11.153  < 2e-16 ***
## sex Male                 0.896780   0.073168  12.256  < 2e-16 ***
## hours_w                  0.029123   0.001567  18.581  < 2e-16 ***
## eduAssoc                 0.290400   0.264660   1.097 0.272531    
## eduBachelors             0.577348   0.326916   1.766 0.077388 .  
## eduDoctorate             0.970276   0.476384   2.037 0.041675 *  
## eduHS-grad               0.269412   0.163245   1.650 0.098871 .  
## eduMasters               0.735076   0.373338   1.969 0.048961 *  
## eduProf-school           1.253929   0.428663   2.925 0.003442 ** 
## eduSome-college          0.408761   0.203328   2.010 0.044394 *  
## rp Not-in-family        -0.943345   0.159611  -5.910 3.42e-09 ***
## rp Other-relative       -1.337961   0.214903  -6.226 4.79e-10 ***
## rp Own-child            -2.062235   0.198591 -10.384  < 2e-16 ***
## rp Unmarried            -1.168674   0.176157  -6.634 3.26e-11 ***
## rp Wife                  1.325758   0.097640  13.578  < 2e-16 ***
## maritalMarried           0.572832   0.163933   3.494 0.000475 ***
## maritalSeparated        -0.067375   0.150089  -0.449 0.653506    
## maritalsingle           -0.494008   0.080825  -6.112 9.83e-10 ***
## maritalWidowed           0.166163   0.142698   1.164 0.244247    
## race Asian-Pac-Islander  0.408697   0.235270   1.737 0.082362 .  
## race Black               0.430887   0.223627   1.927 0.054003 .  
## race Other              -0.223874   0.347034  -0.645 0.518857    
## race White               0.527018   0.213638   2.467 0.013630 *  
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## 
## (Dispersion parameter for binomial family taken to be 1)
## 
##     Null deviance: 33851  on 30161  degrees of freedom
## Residual deviance: 21806  on 30127  degrees of freedom
## AIC: 21876
## 
## Number of Fisher Scoring iterations: 12
```

First model

A few interpretations:

```r
## (2) first model
# consider age effect
## logodds to 0.028390* age
## estimated odds
exp(0.028390)
```

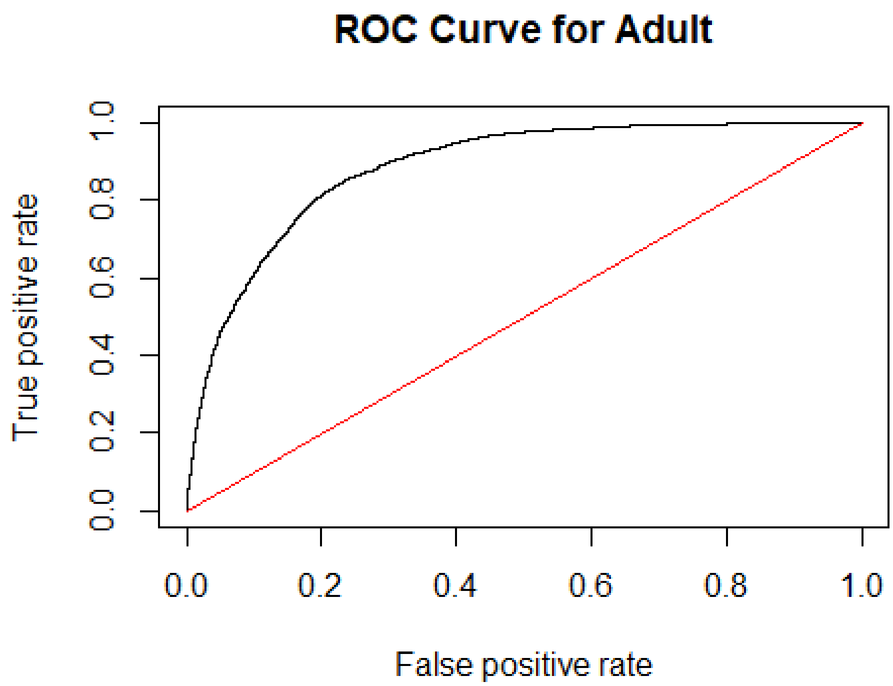
```
## [1] 1.028797
```

Considering a male at age 40 years old with workclass = government, education number= 16 y, occupation is White-Collar, hours-per-week is 40 hrs, education is Doctoral, relationship is Wife, maritial status is married, race is White

```
## (2) first model
#calculate estimate odds for age while holding all other predictors constant

logodds = -7.14719+0.02839* 40 + 0 + 0.2132*16 + 0.8062 + 0.8968 + 0.02912*40
+ 0.9703 + 1.3258 + 0.5728 + 0.5270
logodds

## [1] 3.66331

estimatedodds = exp(logodds)
prob = estimatedodds/(1+estimatedodds)
prob

## [1] 0.9749939
```

First model - machine learning

```
###apply ML train/test for first model
##set the random number generator so same results can be reproduced
set.seed(2019)
##choose the observations to be in the training. I am splitting the dataset i
nto halves
sample<-sample.int(nrow(data), floor(.50*nrow(data)), replace = F)
train<-data[sample, ]
test<-data[-sample, ]
##use training data to fit logistic regression model with 10 predictors
result<-glm(income ~age+wc+edu_num+occup+sex+hours_w+ edu +rp + marital + rac
e, family = "binomial", data = train)
library(ROCR)
##predicted survival rate for testing data based on training data
preds<-predict(result,newdata=test, type="response")

##produce the numbers associated with classification table
rates<-prediction(preds, test$income)

##store the true positive and false postive rates
roc_result<-performance(rates,measure="tpr", x.measure="fpr")

##plot ROC curve and overlay the diagonal line for random guessing
plot(roc_result, main="ROC Curve for Adult")
lines(x = c(0,1), y = c(0,1), col="red")
```




```r
##compute the AUC
auc<-performance(rates, measure = "auc")
auc
```

```
## An object of class "performance"
## Slot "x.name":
## [1] "None"
##
## Slot "y.name":
## [1] "Area under the ROC curve"
##
## Slot "alpha.name":
## [1] "none"
##
## Slot "x.values":
## list()
##
## Slot "y.values":
## [[1]]
## [1] 0.8842191
##
##
## Slot "alpha.values":
## list()
```

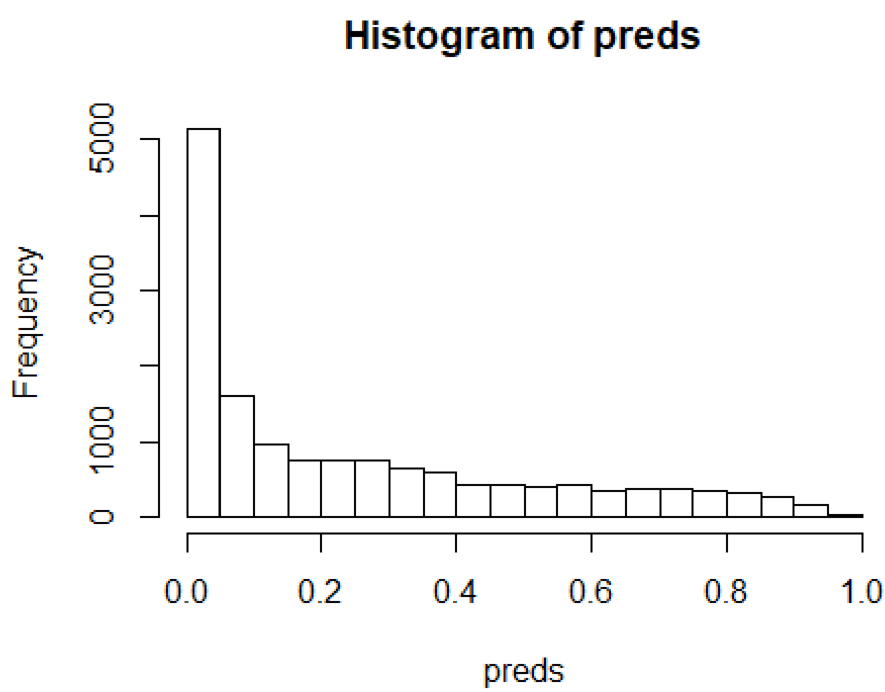
```
##confusion matrix. Actual values in the rows, predicted classification in co
ls
table(test$income, preds>0.5)

##
##           FALSE  TRUE
##    <=50K 10357   911
##    >50K   1696  2117
```

First model - machine learning

```
## first model accuracy from confusion matrix
(10357+2117)/(10357+911+1696+2117)

## [1] 0.8271335
```

First model - machine learning

```
hist(preds)
```



Test hypothesis

consider to drop race

```
## Test hypothesis
##consider to drop race and will adopt test hypothesis
## check residual deviance to compared with first model
droprace <- glm(income ~age+wc+edu_num+occup+sex+hours_w+ edu +rp + marital,
family = "binomial", data = data)
summary(droprace)
```

```
##
## Call:
## glm(formula = income ~ age + wc + edu_num + occup + sex + hours_w +
##     edu + rp + marital, family = "binomial", data = data)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.7612  -0.5889  -0.2306  -0.0225   3.3923
##
## Coefficients:
##                      Estimate Std. Error z value Pr(>|z|)
## (Intercept)         -6.970043   0.345408 -20.179  < 2e-16 ***
## age                  0.028630   0.001590  18.008  < 2e-16 ***
## wcOthers           -12.049759 119.503380  -0.101 0.919684
## wcPrivate            0.102136   0.050056   2.040 0.041307 *
## wcSelf-Employed     -0.206782   0.064616  -3.200 0.001374 **
## edu_num              0.214655   0.043307   4.957 7.18e-07 ***
## occupBlue-Collar    -0.248263   0.069424  -3.576 0.000349 ***
## occupMilitary       -0.290128   1.283847  -0.226 0.821215
## occupOther-Occup     0.501929   0.088773   5.654 1.57e-08 ***
## occupProfessional    0.462510   0.076225   6.068 1.30e-09 ***
## occupSales           0.285632   0.077097   3.705 0.000212 ***
## occupService        -1.026949   0.111958  -9.173  < 2e-16 ***
## occupWhite-Collar    0.810031   0.072224  11.216  < 2e-16 ***
## sex Male             0.898467   0.073111  12.289  < 2e-16 ***
## hours_w              0.029226   0.001567  18.654  < 2e-16 ***
## eduAssoc             0.290642   0.264085   1.101 0.271086
## eduBachelors         0.575141   0.326107   1.764 0.077790 .
## eduDoctorate         0.963722   0.475230   2.028 0.042570 *
## eduHS-grad           0.271812   0.162958   1.668 0.095319 .
## eduMasters           0.732132   0.372402   1.966 0.049301 *
## eduProf-school       1.244394   0.427611   2.910 0.003613 **
## eduSome-college      0.407814   0.202924   2.010 0.044464 *
## rp Not-in-family    -0.969909   0.159293  -6.089 1.14e-09 ***
## rp Other-relative   -1.377141   0.214238  -6.428 1.29e-10 ***
## rp Own-child        -2.088491   0.198291 -10.532  < 2e-16 ***
## rp Unmarried        -1.204266   0.175632  -6.857 7.04e-12 ***
## rp Wife              1.321093   0.097565  13.541  < 2e-16 ***
## maritalMarried       0.544962   0.163515   3.333 0.000860 ***
## maritalSeparated    -0.079193   0.149680  -0.529 0.596749
## maritalsingle       -0.496743   0.080723  -6.154 7.57e-10 ***
## maritalWidowed       0.164877   0.142603   1.156 0.247601
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 33851  on 30161  degrees of freedom
## Residual deviance: 21824  on 30131  degrees of freedom
## AIC: 21886
```

```
##
## Number of Fisher Scoring iterations: 12
```

Test hypothesis - consider to drop race

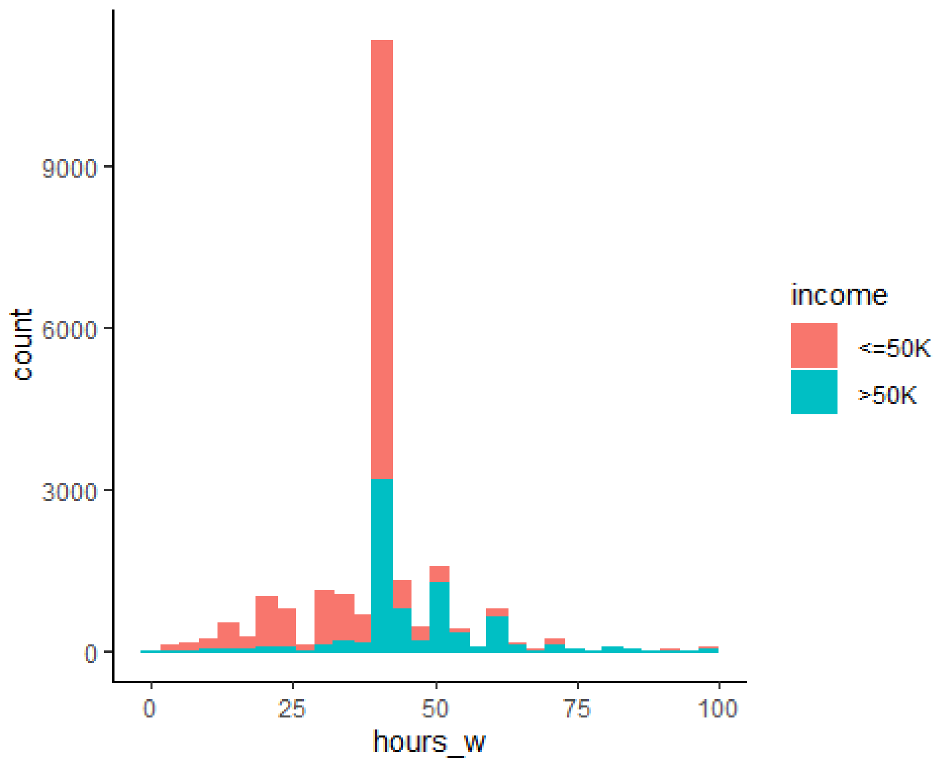
```
## Test hypothesis
## p value for dropping race
1-pchisq(16, 4)

## [1] 0.003019164
```

Test hopothes- consider to drop hours-per-week

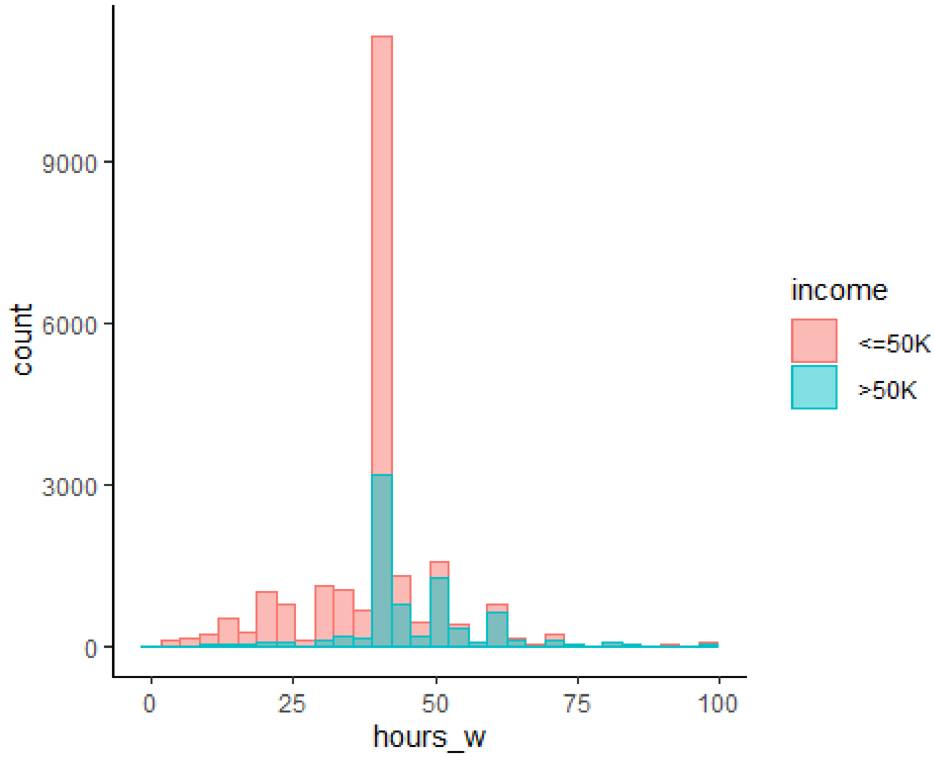
use Wald test

Consider to drop hours-per-week due to wide distribution from histogram diagram

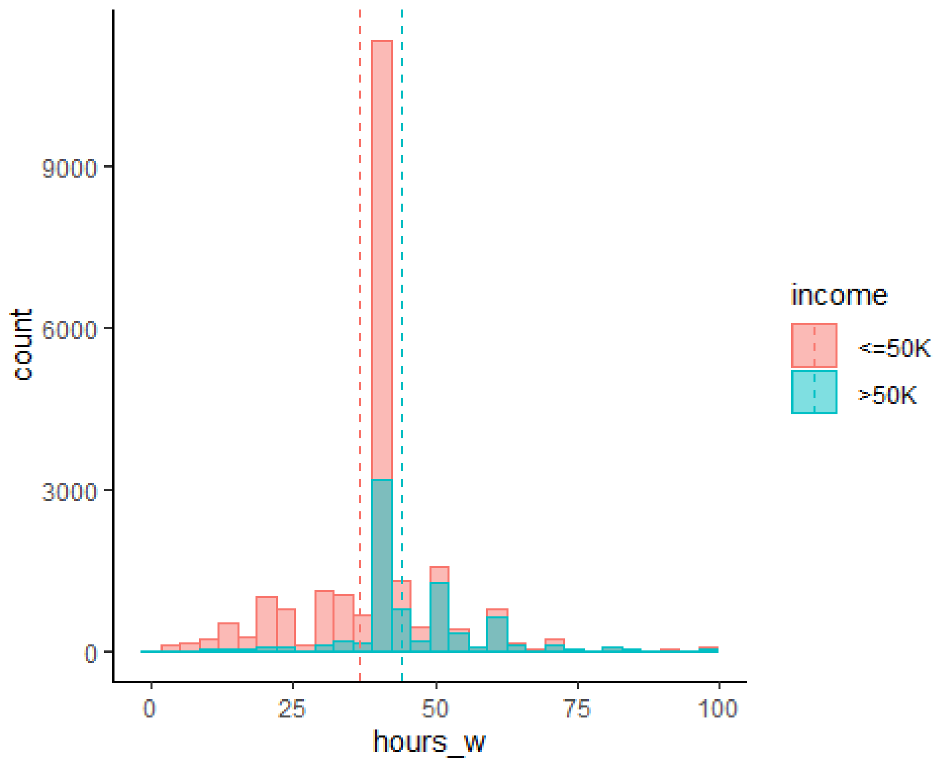
```
##(1) data cleaning and manipulation
##Check age histogram colored by
library(plyr)
muHour <- ddply(data, "income", summarise, grp.mean=mean(age))
head(mu)

##   income grp.mean
## 1  <=50K 36.60806
## 2   >50K 43.95911

# Change histogram plot fill colors by groups
ggplot(data, aes(x=hours_w, fill=income, color=income)) +
  geom_histogram(position="identity")

## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



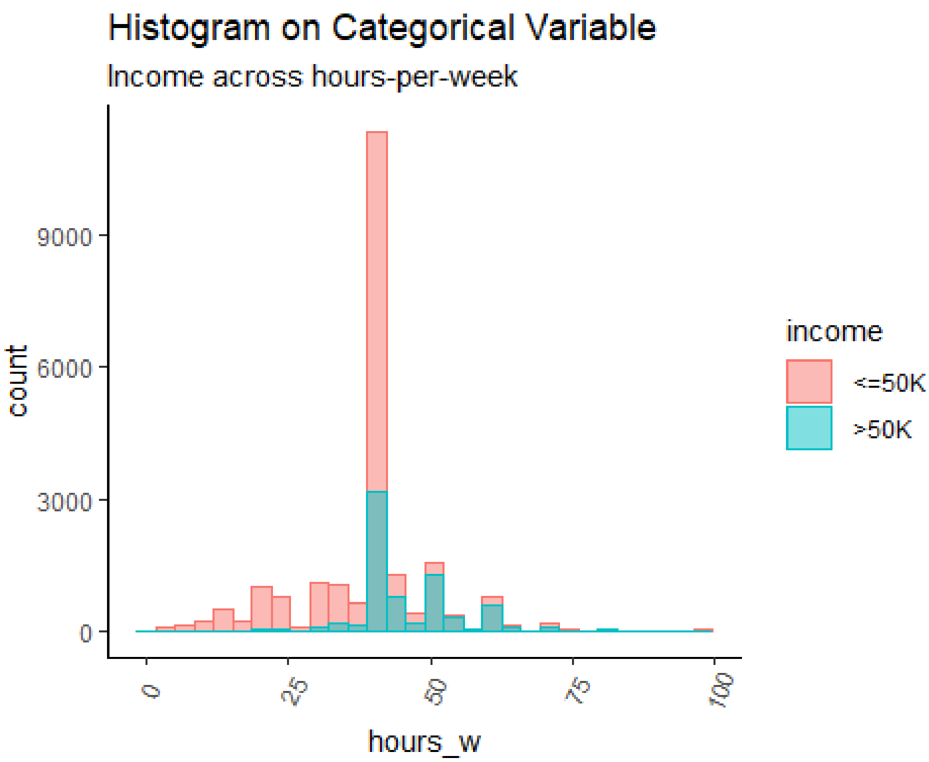

```r
# Use semi-transparent fill
p1<-ggplot(data, aes(x=hours_w, fill=income, color=income)) +
  geom_histogram(position="identity", alpha=0.5)
p1

## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```




```
# Add mean lines
p1+geom_vline(data=mu, aes(xintercept=grp.mean, color=income),
             linetype="dashed")

## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```




```
p1+theme(axis.text.x = element_text(angle=65, vjust=0.6)) +
  labs(title="Histogram on Categorical Variable",
       subtitle="Income across hours-per-week")

## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



considering to drop hours-per-week

```
## Test hypothesis
#consider to drop hours-per-week
drophour_w <- glm(income ~age+wc+edu_num+occup+sex+hours_w + marital + race,
family = "binomial", data = data)
summary(drophour_w)

##
## Call:
## glm(formula = income ~ age + wc + edu_num + occup + sex + hours_w +
##     marital + race, family = "binomial", data = data)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.6947  -0.5936  -0.2526  -0.0380   3.4782
##
## Coefficients:
##                             Estimate Std. Error z value Pr(>|z|)
## (Intercept)                -8.823594   0.262487 -33.615  < 2e-16 ***
## age                         0.029720   0.001543  19.261  < 2e-16 ***
## wcOthers                  -12.161906 119.171734  -0.102  0.91871
## wcPrivate                   0.105419   0.049306   2.138  0.03251 *
## wcSelf-Employed            -0.169846   0.063807  -2.662  0.00777 **
## edu_num                     0.298867   0.008887  33.631  < 2e-16 ***
## occupBlue-Collar           -0.300743   0.067679  -4.444 8.84e-06 ***
## occupMilitary              -0.376567   1.279174  -0.294  0.76847
```

```
## occupOther-Occup          0.466906   0.087178   5.356 8.52e-08 ***
## occupProfessional         0.505090   0.072322   6.984 2.87e-12 ***
## occupSales                0.237541   0.075274   3.156  0.00160 **
## occupService             -1.026116   0.110663  -9.272  < 2e-16 ***
## occupWhite-Collar         0.787345   0.070027  11.243  < 2e-16 ***
## sex Male                  0.298893   0.048944   6.107 1.02e-09 ***
## hours_w                   0.029100   0.001543  18.856  < 2e-16 ***
## maritalMarried            1.986955   0.061327  32.399  < 2e-16 ***
## maritalSeparated         -0.061873   0.146342  -0.423  0.67244
## maritalsingle            -0.490623   0.075780  -6.474 9.52e-11 ***
## maritalWidowed            0.026656   0.139630   0.191  0.84860
## race Asian-Pac-Islander   0.352767   0.231220   1.526  0.12709
## race Black                0.435840   0.220187   1.979  0.04777 *
## race Other               -0.280052   0.343516  -0.815  0.41493
## race White                0.555098   0.210463   2.638  0.00835 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 33851  on 30161  degrees of freedom
## Residual deviance: 22270  on 30139  degrees of freedom
## AIC: 22316
##
## Number of Fisher Scoring iterations: 12
```

##adopt Wald test

```
##Test hypothesis calculate test statistic to compare 95% confidence level
18.856/0.001543

## [1] 12220.35

##the p-value
(2 * pnorm(12220.35, lower.tail=FALSE))

## [1] 0
```

multicollinearity to drop relationship and eductaion, total predictors for second model is 8

Second model

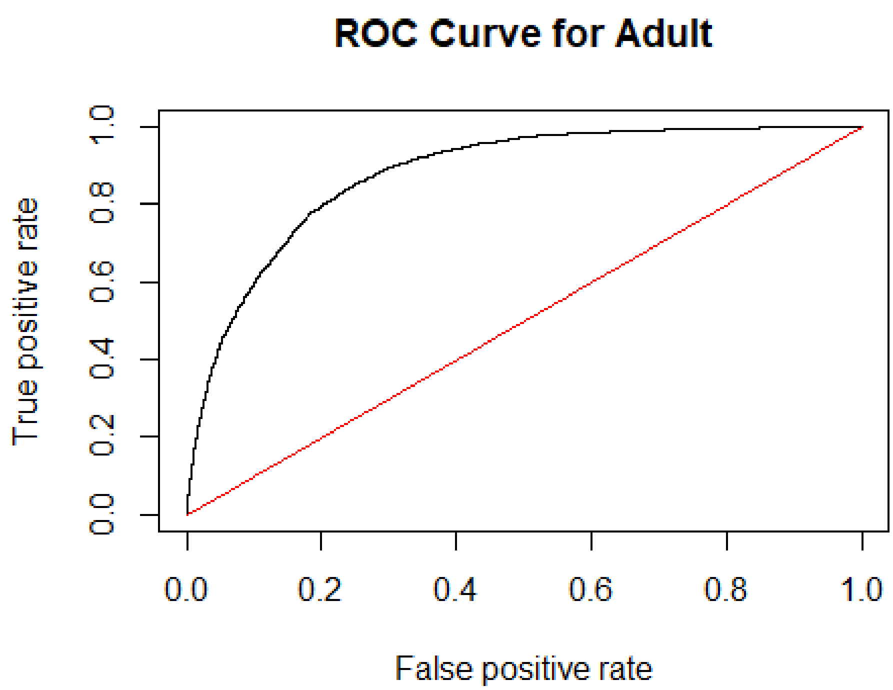
```
## second model
##considering muliconllinearity, drop relationship and education and native c
ountry
m4 <- glm(income ~age+wc+edu_num+occup+sex+hours_w+ marital + race, family =
"binomial", data = data)
summary(m4)
```

```
##
## Call:
## glm(formula = income ~ age + wc + edu_num + occup + sex + hours_w +
##     marital + race, family = "binomial", data = data)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.6947  -0.5936  -0.2526  -0.0380   3.4782
##
## Coefficients:
##                           Estimate Std. Error z value Pr(>|z|)
## (Intercept)              -8.823594   0.262487 -33.615  < 2e-16 ***
## age                       0.029720   0.001543  19.261  < 2e-16 ***
## wcOthers                -12.161906 119.171734  -0.102  0.91871
## wcPrivate                 0.105419   0.049306   2.138  0.03251 *
## wcSelf-Employed          -0.169846   0.063807  -2.662  0.00777 **
## edu_num                   0.298867   0.008887  33.631  < 2e-16 ***
## occupBlue-Collar         -0.300743   0.067679  -4.444 8.84e-06 ***
## occupMilitary            -0.376567   1.279174  -0.294  0.76847
## occupOther-Occup          0.466906   0.087178   5.356 8.52e-08 ***
## occupProfessional         0.505090   0.072322   6.984 2.87e-12 ***
## occupSales                0.237541   0.075274   3.156  0.00160 **
## occupService             -1.026116   0.110663  -9.272  < 2e-16 ***
## occupWhite-Collar         0.787345   0.070027  11.243  < 2e-16 ***
## sex Male                  0.298893   0.048944   6.107 1.02e-09 ***
## hours_w                   0.029100   0.001543  18.856  < 2e-16 ***
## maritalMarried            1.986955   0.061327  32.399  < 2e-16 ***
## maritalSeparated         -0.061873   0.146342  -0.423  0.67244
## maritalsingle            -0.490623   0.075780  -6.474 9.52e-11 ***
## maritalWidowed            0.026656   0.139630   0.191  0.84860
## race Asian-Pac-Islander   0.352767   0.231220   1.526  0.12709
## race Black                0.435840   0.220187   1.979  0.04777 *
## race Other               -0.280052   0.343516  -0.815  0.41493
## race White                0.555098   0.210463   2.638  0.00835 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 33851  on 30161  degrees of freedom
## Residual deviance: 22270  on 30139  degrees of freedom
## AIC: 22316
##
## Number of Fisher Scoring iterations: 12
```

Second model-Machine learning

```
##apply ML train/test for simple model (50/50)
##set the random number generator so same results can be reproduced
set.seed(2019)
##choose the observations to be in the training. I am splitting the dataset i
```

```
nto halves
sample<-sample.int(nrow(data), floor(.50*nrow(data)), replace = F)
train<-data[sample, ]
test<-data[-sample, ]
##use training data to fit logistic regression model with fare and gender as
predictors
result<-glm(income ~age+wc+edu_num+occup+sex+hours_w+ marital + race, family
= "binomial", data = train)
library(ROCR)
##predicted survival rate for testing data based on training data
preds<-predict(result,newdata=test, type="response")

##produce the numbers associated with classification table
rates<-prediction(preds, test$income)

##store the true positive and false postive rates
roc_result<-performance(rates,measure="tpr", x.measure="fpr")

##plot ROC curve and overlay the diagonal line for random guessing
plot(roc_result, main="ROC Curve for Adult")
lines(x = c(0,1), y = c(0,1), col="red")
```




```
##compute the AUC
auc<-performance(rates, measure = "auc")
auc
```

```
## An object of class "performance"
## Slot "x.name":
## [1] "None"
##
## Slot "y.name":
## [1] "Area under the ROC curve"
##
## Slot "alpha.name":
## [1] "none"
##
## Slot "x.values":
## list()
##
## Slot "y.values":
## [[1]]
## [1] 0.8784091
##
##
## Slot "alpha.values":
## list()
```

```
##confusion matrix. Actual values in the rows, predicted classification in co
ls
table(test$income, preds>0.5)
```

```
##
##          FALSE  TRUE
##    <=50K 10319   949
##    >50K   1721  2092
```

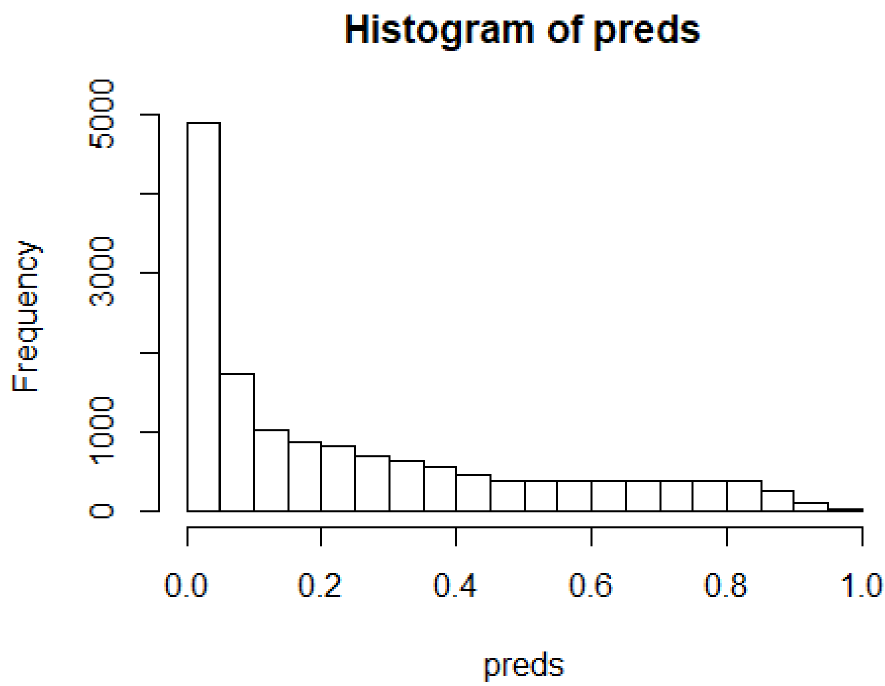
Second model-Machine learning

```
## Second model
#accuracy
(10319+2092)/(10319+949+1721+2092)
```

```
## [1] 0.822956
```

Second model histogram-prediction plot

```
## Second model
hist(preds)
```



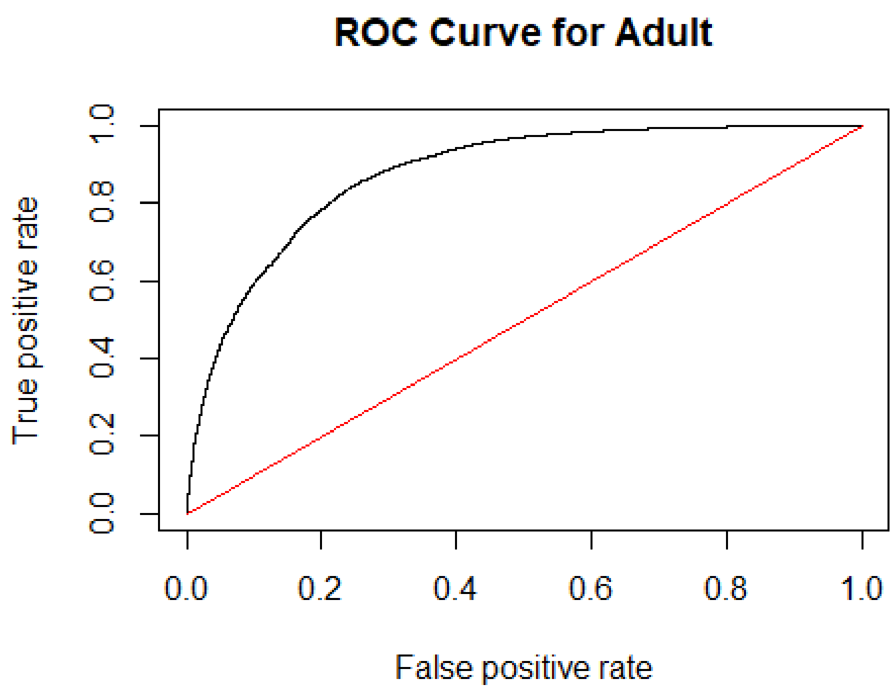
Second

```
## Second model
##calculate false positive rate and false negative rate of second model
949/(949+10319)

## [1] 0.0842208

1721/(1721+2092)

## [1] 0.4513506
```

##Model evaluation - to see train/test split effects on model accuracy/false positive/false negative rates

```
## Model evaluation
#Use train/test split 20/80 for simple model
###apply ML train/test for simple model (20/80)
##set the random number generator so same results can be reproduced
set.seed(2019)
##choose the observations to be in the training. I am splitting the dataset i
nto halves
sample<-sample.int(nrow(data), floor(.20*nrow(data)), replace = F)
train<-data[sample, ]
test<-data[-sample, ]
##use training data to fit logistic regression model with fare and gender as
predictors
result<-glm(income ~age+wc+edu_num+occup+sex+hours_w+ marital + race, family
= "binomial", data = train)
```

```r
library(ROCR)
##predicted survival rate for testing data based on training data
preds<-predict(result,newdata=test, type="response")

##produce the numbers associated with classification table
rates<-prediction(preds, test$income)

##store the true positive and false postive rates
roc_result<-performance(rates,measure="tpr", x.measure="fpr")

##plot ROC curve and overlay the diagonal line for random guessing
plot(roc_result, main="ROC Curve for Adult")
lines(x = c(0,1), y = c(0,1), col="red")
```




```r
##compute the AUC
auc<-performance(rates, measure = "auc")
auc

## An object of class "performance"
## Slot "x.name":
## [1] "None"
##
## Slot "y.name":
## [1] "Area under the ROC curve"
##
## Slot "alpha.name":
## [1] "none"
```

```
##
## Slot "x.values":
## list()
##
## Slot "y.values":
## [[1]]
## [1] 0.8762996
##
##
## Slot "alpha.values":
## list()
```

```
##confusion matrix. Actual values in the rows, predicted classification in co
ls
table(test$income, preds>0.5)
```

```
##
##          FALSE  TRUE
##    <=50K 16639  1383
##    >50K   2866  3242
```

```
###accuracy
```

```
###accuracy
(16639+3242)/(16639+1383+2866+3242)
```

```
## [1] 0.8239121
```

```
##false positive
1383/(1383+16639)
```

```
## [1] 0.07673954
```

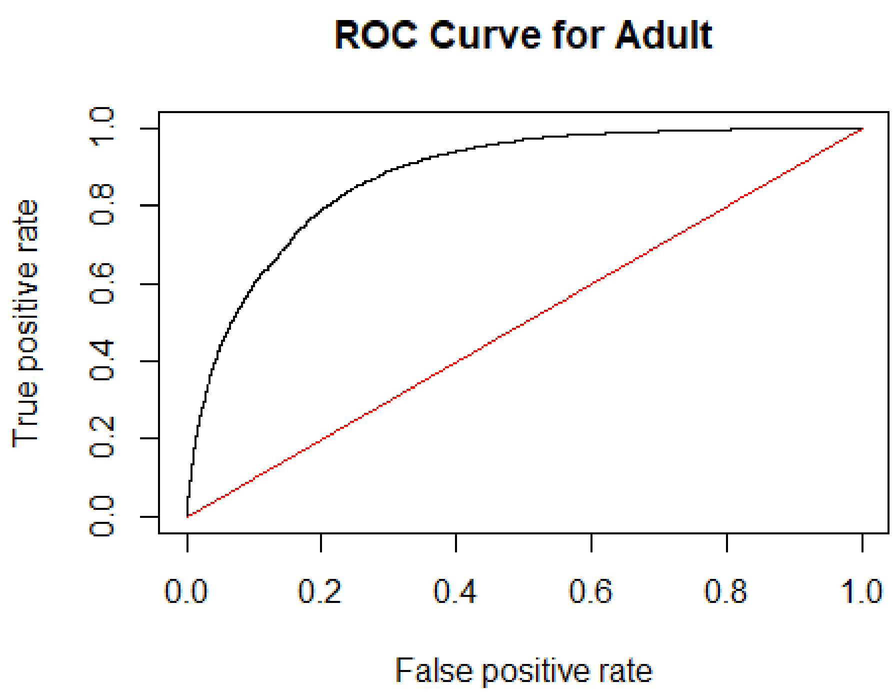
```
##false negative
2866/(2866+3242)
```

```
## [1] 0.4692207
```

##Model evaluation - to see train/test split effects on model accuracy/false positive/false negative rates

```
## Model evaluation
#Use train/test split 40/60 for simple model
###apply ML train/test for simple model (40/60)
##set the random number generator so same results can be reproduced
set.seed(2019)
##choose the observations to be in the training. I am splitting the dataset i
nto halves
sample<-sample.int(nrow(data), floor(.40*nrow(data)), replace = F)
train<-data[sample, ]
test<-data[-sample, ]
##use training data to fit logistic regression model with fare and gender as
```

```
predictors
result<-glm(income ~age+wc+edu_num+occup+sex+hours_w+ marital + race, family
= "binomial", data = train)
library(ROCR)
##predicted survival rate for testing data based on training data
preds<-predict(result,newdata=test, type="response")

##produce the numbers associated with classification table
rates<-prediction(preds, test$income)

##store the true positive and false postive rates
roc_result<-performance(rates,measure="tpr", x.measure="fpr")

##plot ROC curve and overlay the diagonal line for random guessing
plot(roc_result, main="ROC Curve for Adult")
lines(x = c(0,1), y = c(0,1), col="red")
```




```
##compute the AUC
auc<-performance(rates, measure = "auc")
auc

## An object of class "performance"
## Slot "x.name":
## [1] "None"
##
## Slot "y.name":
## [1] "Area under the ROC curve"
```

```
##
## Slot "alpha.name":
## [1] "none"
##
## Slot "x.values":
## list()
##
## Slot "y.values":
## [[1]]
## [1] 0.8773427
##
##
## Slot "alpha.values":
## list()
```

```
##confusion matrix. Actual values in the rows, predicted classification in co
ls
table(test$income, preds>0.5)
```

```
##
##           FALSE  TRUE
##    <=50K 12388  1118
##    >50K   2055  2537
```

```
#Accuracy
(12388+2537)/(12388+1118+2055+2537)
```

```
## [1] 0.8246768
```

```
##false positive
1118/(1118+12388)
```

```
## [1] 0.08277802
```

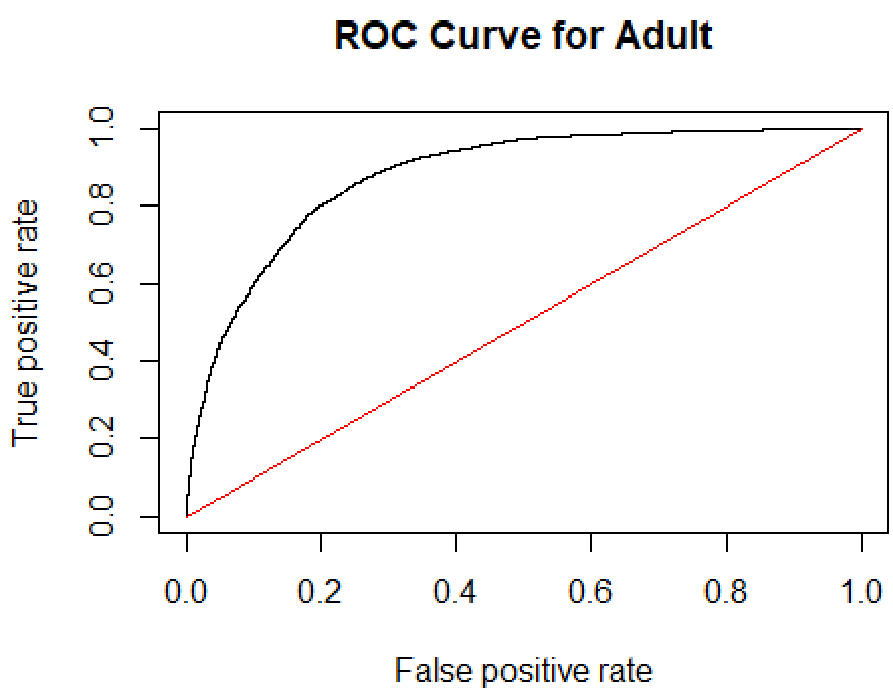
```
##false negative
2055/(2055+2537)
```

```
## [1] 0.4475174
```

##Model evaluation - to see train/test split effects on model accuracy/false positive/false negative rates

```
## Model evaluation
#Use train/test split 60/40 for simple model
###apply ML train/test for simple model (60/40)
##set the random number generator so same results can be reproduced
set.seed(2019)
##choose the observations to be in the training. I am splitting the dataset i
nto halves
sample<-sample.int(nrow(data), floor(.60*nrow(data)), replace = F)
train<-data[sample, ]
test<-data[-sample, ]
```

```r
##use training data to fit logistic regression model with fare and gender as predictors
result<-glm(income ~age+wc+edu_num+occup+sex+hours_w+ marital + race, family = "binomial", data = train)
library(ROCR)
##predicted survival rate for testing data based on training data
preds<-predict(result,newdata=test, type="response")

##produce the numbers associated with classification table
rates<-prediction(preds, test$income)

##store the true positive and false postive rates
roc_result<-performance(rates,measure="tpr", x.measure="fpr")

##plot ROC curve and overlay the diagonal line for random guessing
plot(roc_result, main="ROC Curve for Adult")
lines(x = c(0,1), y = c(0,1), col="red")
```




```r
##compute the AUC
auc<-performance(rates, measure = "auc")
auc

## An object of class "performance"
## Slot "x.name":
## [1] "None"
##
## Slot "y.name":
```

```
## [1] "Area under the ROC curve"
##
## Slot "alpha.name":
## [1] "none"
##
## Slot "x.values":
## list()
##
## Slot "y.values":
## [[1]]
## [1] 0.8798726
##
##
## Slot "alpha.values":
## list()
```

```
##confusion matrix. Actual values in the rows, predicted classification in co
ls
table(test$income, preds>0.5)
```

```
##
##           FALSE TRUE
##    <=50K   8270  736
##    >50K    1387 1672
```

```
#Accuracy
(8270+1672)/(8270+736+1387+1672)
```

```
## [1] 0.8240365
```

```
##false positive
736/(736+8270)
```

```
## [1] 0.0817233
```

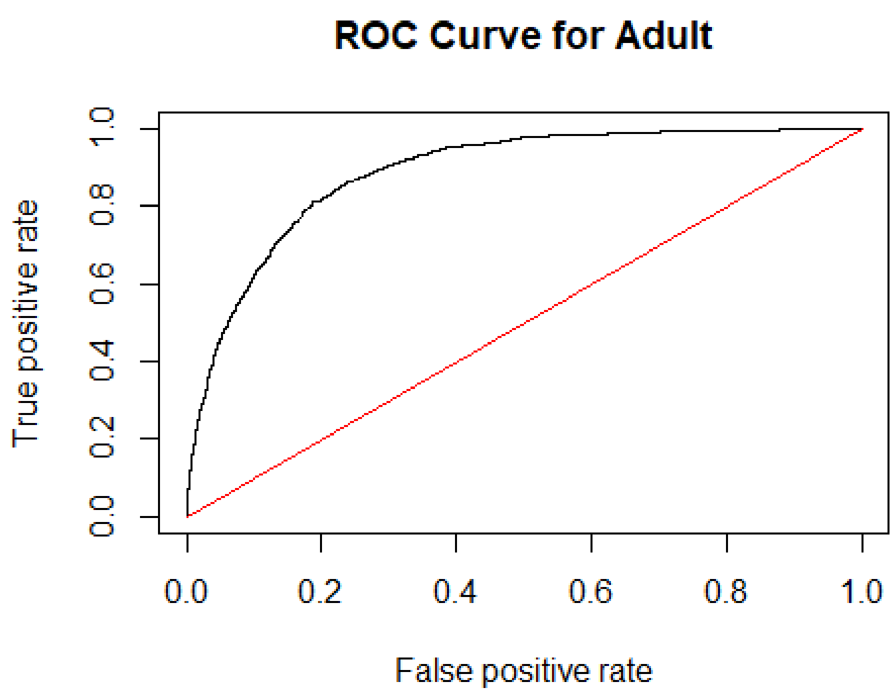
```
##false negative
1387/(1387+1672)
```

```
## [1] 0.4534161
```

##Model Evaluation - to see train/test split effects on model accuracy/false positive/false negative rates

```
## Model evaluation
##Use train/test split 80/20 for simple model
##set the random number generator so same results can be reproduced
set.seed(2019)
##choose the observations to be in the training. I am splitting the dataset i
nto halves
sample<-sample.int(nrow(data), floor(.80*nrow(data)), replace = F)
train<-data[sample, ]
test<-data[-sample, ]
```

```r
##use training data to fit logistic regression model with fare and gender as predictors
result<-glm(income ~age+wc+edu_num+occup+sex+hours_w+ marital + race, family = "binomial", data = train)
library(ROCR)
##predicted survival rate for testing data based on training data
preds<-predict(result,newdata=test, type="response")

##produce the numbers associated with classification table
rates<-prediction(preds, test$income)

##store the true positive and false postive rates
roc_result<-performance(rates,measure="tpr", x.measure="fpr")

##plot ROC curve and overlay the diagonal line for random guessing
plot(roc_result, main="ROC Curve for Adult")
lines(x = c(0,1), y = c(0,1), col="red")
```




```r
##compute the AUC
auc<-performance(rates, measure = "auc")
auc

## An object of class "performance"
## Slot "x.name":
## [1] "None"
##
## Slot "y.name":
```

```
## [1] "Area under the ROC curve"
##
## Slot "alpha.name":
## [1] "none"
##
## Slot "x.values":
## list()
##
## Slot "y.values":
## [[1]]
## [1] 0.8865583
##
##
## Slot "alpha.values":
## list()
```

```
##confusion matrix. Actual values in the rows, predicted classification in co
ls
table(test$income, preds>0.5)
```

```
##
##          FALSE TRUE
##    <=50K  4174  355
##    >50K    670  834
```

```
###accuray
(4174+834)/(4174+355+670+834)
```

```
## [1] 0.8301011
```

```
##false positive
355/(355+4174)
```

```
## [1] 0.07838375
```

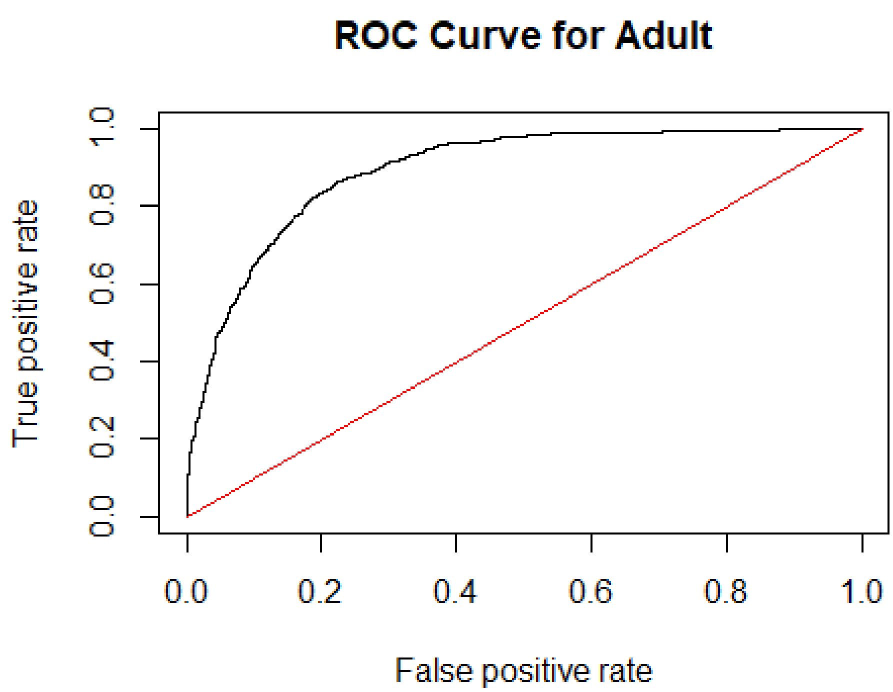
```
##false negative
670/(670+834)
```

```
## [1] 0.4454787
```

##Model evaluation - to see train/test split effects on model accuracy/false positive/false negative rates

```
## Model evaluation
##Use train/test split 90/10 for simple model
##set the random number generator so same results can be reproduced
set.seed(2019)
##choose the observations to be in the training. I am splitting the dataset i
nto halves
sample<-sample.int(nrow(data), floor(.90*nrow(data)), replace = F)
train<-data[sample, ]
test<-data[-sample, ]
```

```
##use training data to fit logistic regression model with fare and gender as predictors
result<-glm(income ~age+wc+edu_num+occup+sex+hours_w+ marital + race, family = "binomial", data = train)
library(ROCR)
##predicted survival rate for testing data based on training data
preds<-predict(result,newdata=test, type="response")

##produce the numbers associated with classification table
rates<-prediction(preds, test$income)

##store the true positive and false postive rates
roc_result<-performance(rates,measure="tpr", x.measure="fpr")

##plot ROC curve and overlay the diagonal line for random guessing
plot(roc_result, main="ROC Curve for Adult")
lines(x = c(0,1), y = c(0,1), col="red")
```




```
##compute the AUC
auc<-performance(rates, measure = "auc")
auc

## An object of class "performance"
## Slot "x.name":
## [1] "None"
##
## Slot "y.name":
```

```
## [1] "Area under the ROC curve"
##
## Slot "alpha.name":
## [1] "none"
##
## Slot "x.values":
## list()
##
## Slot "y.values":
## [[1]]
## [1] 0.8930025
##
##
## Slot "alpha.values":
## list()
```

```
##confusion matrix. Actual values in the rows, predicted classification in co
ls
table(test$income, preds>0.5)
```

```
##
##          FALSE TRUE
##    <=50K  2090  174
##    >50K    322  431
```

```
###accuray
(2090+431)/(2090+431+174+322)
```

```
## [1] 0.8355983
```

```
##false positive
174/(174+2090)
```

```
## [1] 0.07685512
```

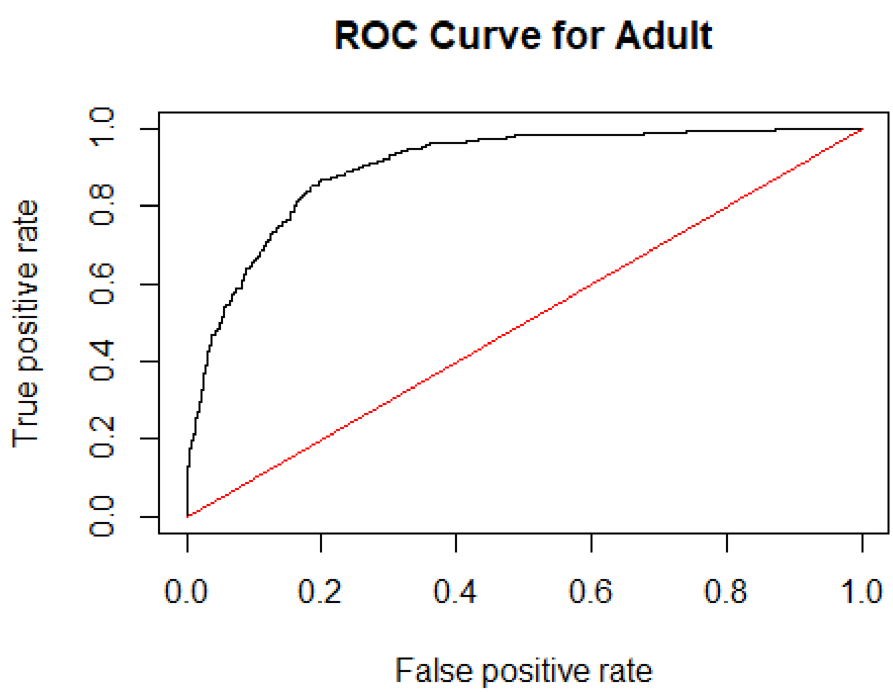
```
##false negative
322/(322+431)
```

```
## [1] 0.4276228
```

##Model evaluation - to see train/test split effects on model accuracy/false positive/false negative rates

```
## Model evaluation
##Use train/test split 95/5 for simple model
##set the random number generator so same results can be reproduced
set.seed(2019)
##choose the observations to be in the training. I am splitting the dataset i
nto halves
sample<-sample.int(nrow(data), floor(.95*nrow(data)), replace = F)
train<-data[sample, ]
test<-data[-sample, ]
```

```
##use training data to fit logistic regression model with fare and gender as predictors
result<-glm(income ~age+wc+edu_num+occup+sex+hours_w+ marital + race, family = "binomial", data = train)
library(ROCR)
##predicted survival rate for testing data based on training data
preds<-predict(result,newdata=test, type="response")

##produce the numbers associated with classification table
rates<-prediction(preds, test$income)

##store the true positive and false postive rates
roc_result<-performance(rates,measure="tpr", x.measure="fpr")

##plot ROC curve and overlay the diagonal line for random guessing
plot(roc_result, main="ROC Curve for Adult")
lines(x = c(0,1), y = c(0,1), col="red")
```




```
##compute the AUC
auc<-performance(rates, measure = "auc")
auc

## An object of class "performance"
## Slot "x.name":
## [1] "None"
##
## Slot "y.name":
```

```
## [1] "Area under the ROC curve"
##
## Slot "alpha.name":
## [1] "none"
##
## Slot "x.values":
## list()
##
## Slot "y.values":
## [[1]]
## [1] 0.9010142
##
##
## Slot "alpha.values":
## list()
```

```
##confusion matrix. Actual values in the rows, predicted classification in co
ls
table(test$income, preds>0.5)
```

```
##
##          FALSE TRUE
##    <=50K  1072   81
##    >50K    150  206
```

```
###accuray
(1072+206)/(1072+81+150+206)
```

```
## [1] 0.8469185
```

```
##false positive
81/(81+1072)
```

```
## [1] 0.07025152
```

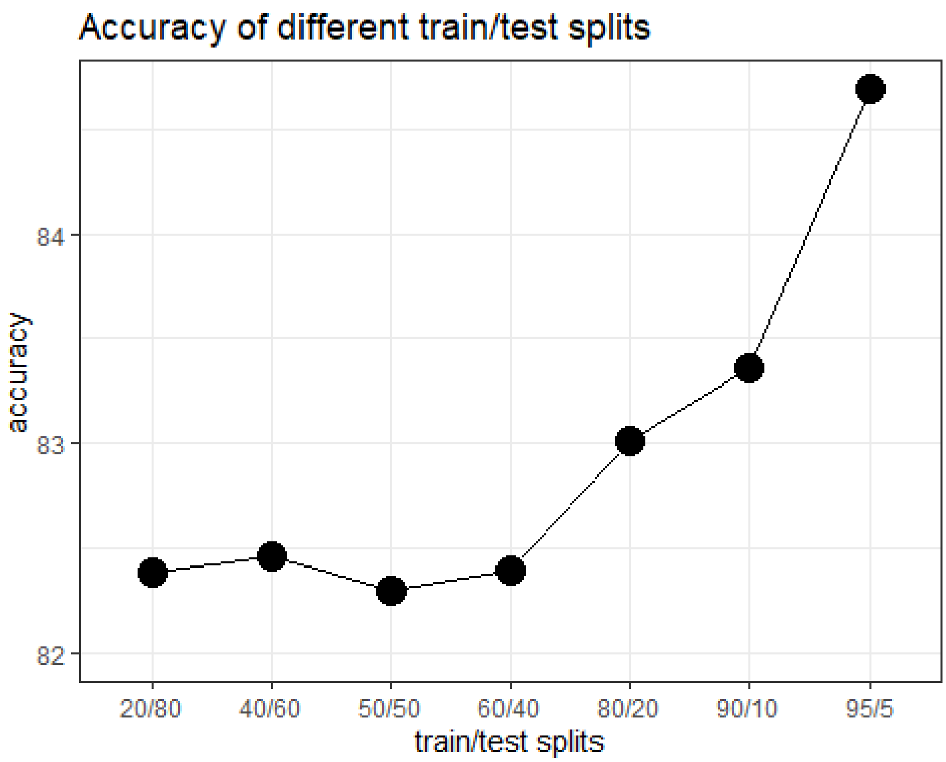
```
##false negative
150/(150+206)
```

```
## [1] 0.4213483
```

##Model evaluation - to see train/test split effects on model accuracy/false positive/false negative rates ###making plot for accuracy vs. tranin/test splits

```
# A line graph
##false positive rate
dat2<-data.frame(
  splits = factor(c("20/80", "40/60", "50/50", "60/40", "80/20", "90/10", "95
/5")),
  accuracy =c(82.39, 82.46, 82.30, 82.40, 83.01, 83.36, 84.69)
)
ggplot(data=dat2, aes(x=splits, y=accuracy, group=1 )) +
    geom_line() +      # Set linetype by accuracy
```

```
    geom_point(size=5, fill="orange") +          # Use larger points, fill wit
h white
    expand_limits(y=c(82, 84)) +                        # Set y range to inclu
de 0
    scale_colour_hue(name="train/test",      # Set legend title
                     l=30)  +                  # Use darker colors (lightness
=30)
    scale_shape_manual(name="train/test",
                       values=c(22,20)) +      # Use points with a fill color
    scale_linetype_discrete(name="train/test") +
    xlab("train/test splits") + ylab("accuracy") + # Set axis labels
    ggtitle("Accuracy of different train/test splits") +     # Set title
    theme_bw() +
    theme(legend.position=c(.7, .4))           # Position legend inside
```

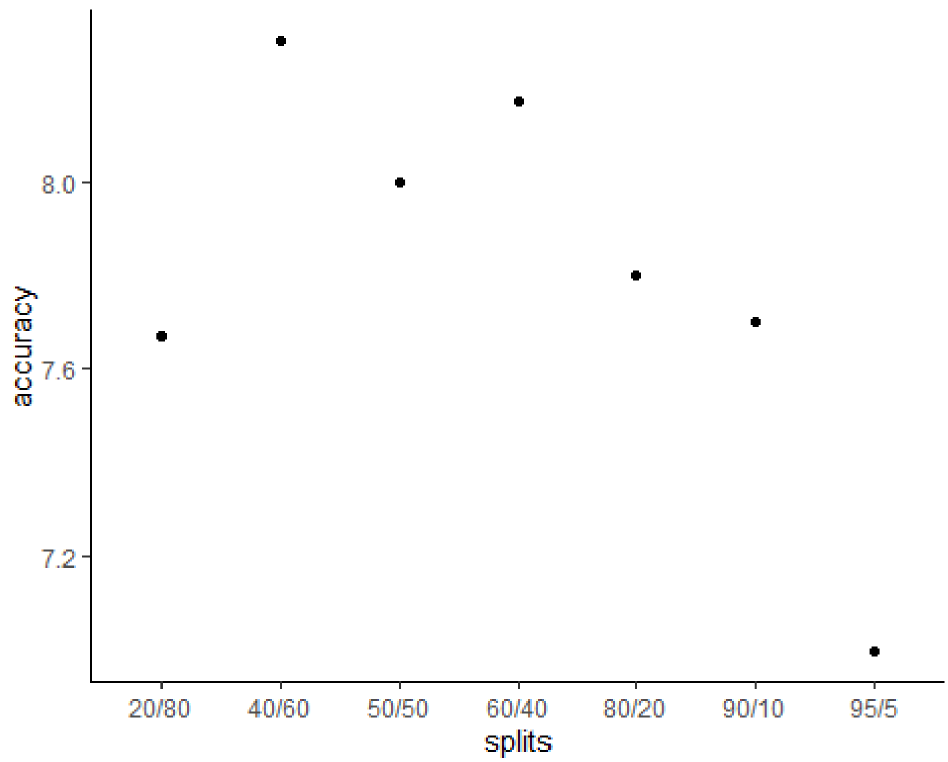



```
                                                # This must go after theme_bw
```

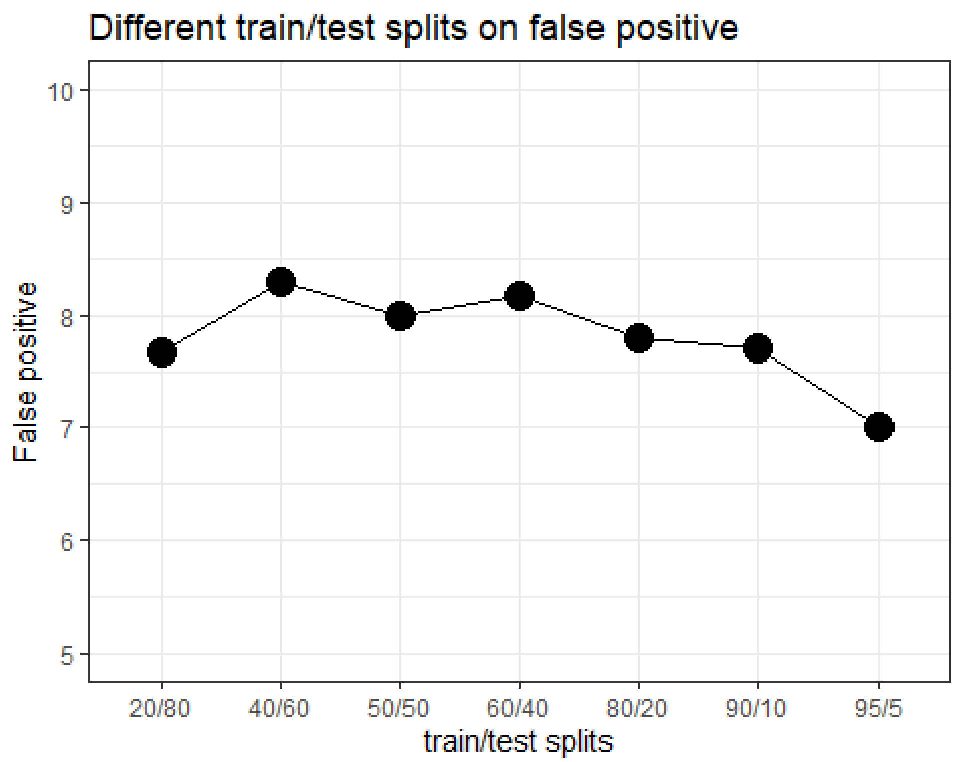
##Model evaluation - to see train/test split effects on model accuracy/false positive/false negative rates ###making plot for false positive rate vs. tranin/test splits

```
##false positive rate
dat2<-data.frame(
  splits = factor(c("20/80", "40/60", "50/50", "60/40", "80/20", "90/10", "95
/5")),
  accuracy =c(7.673, 8.3, 8, 8.172, 7.8, 7.7, 7.0)
)
# Basic line graph with points
```

```r
ggplot(data=dat2, aes(x=splits, y=accuracy)) +
    geom_line() +
    geom_point()

## geom_path: Each group consists of only one observation. Do you need to
## adjust the group aesthetic?
```



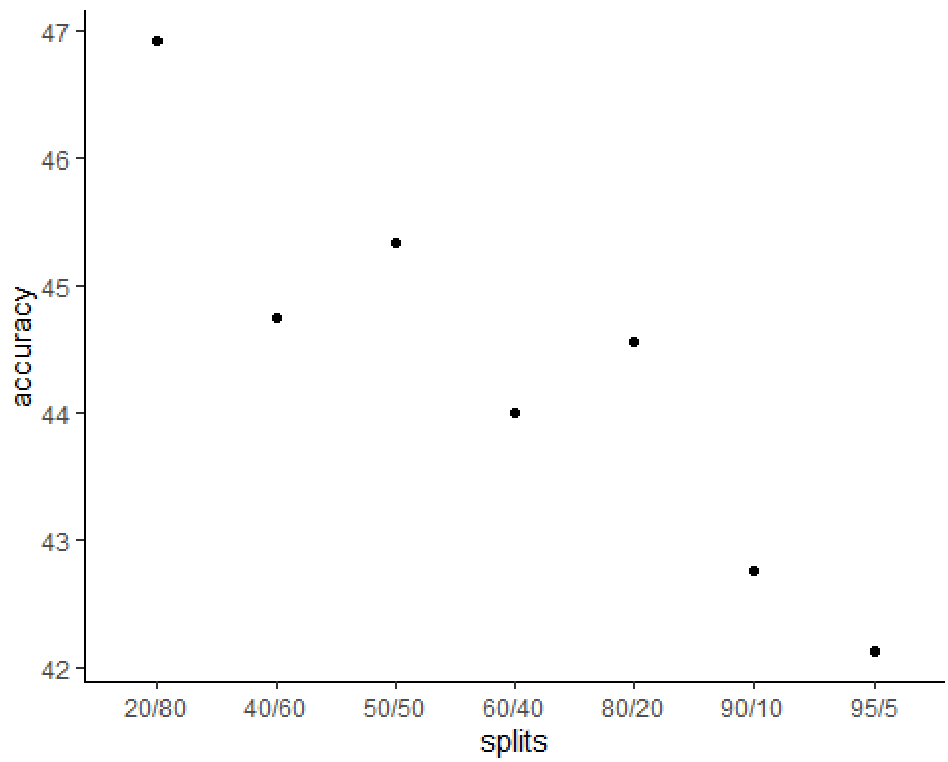

```r
# A line graph
ggplot(data=dat2, aes(x=splits, y=accuracy, group=1 )) +
    geom_line() +     # Set linetype by accuracy
    geom_point(size=5, fill="orange") +         # Use larger points, fill wit
h white
    expand_limits(y=c(5, 10)) +                       # Set y range to includ
e 0
    scale_colour_hue(name="train/test",      # Set legend title
                     l=30)  +                   # Use darker colors (lightness
=30)
    scale_shape_manual(name="train/test",
                       values=c(22,20)) +       # Use points with a fill color
    scale_linetype_discrete(name="train/test") +
    xlab("train/test splits") + ylab("False positive") + # Set axis labels
    ggtitle("Different train/test splits on false positive") +     # Set titl
e
    theme_bw() +
    theme(legend.position=c(.7, .4))            # Position legend inside
```



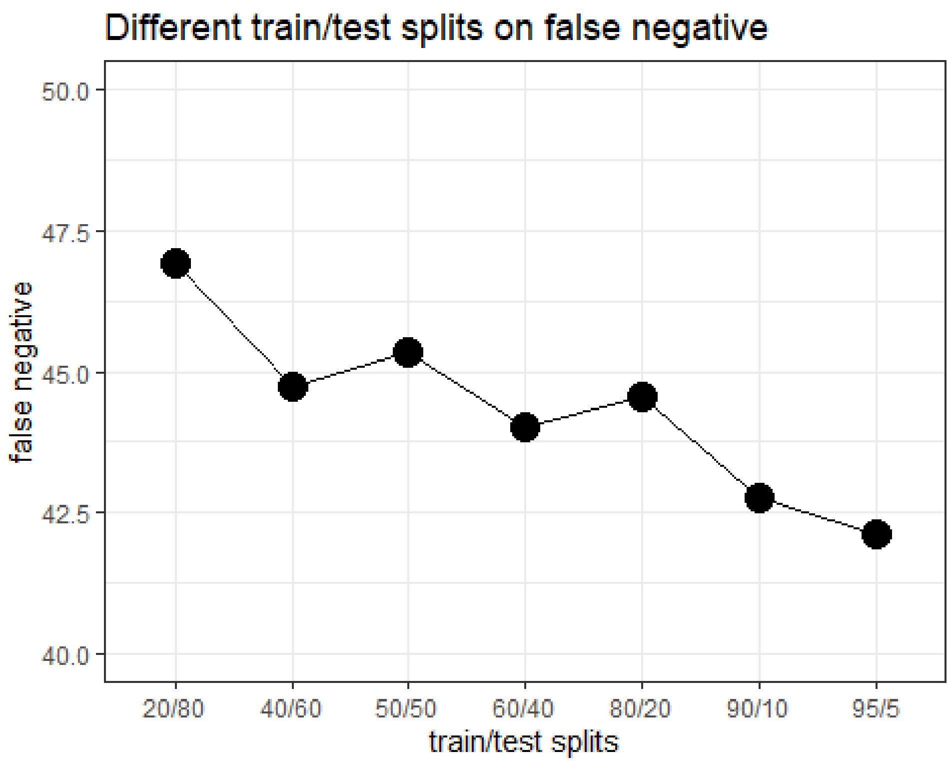

```
                                                           # This must go after theme_bw
```

##Model evaluation - to see train/test split effects on model accuracy/false positive/false negative rates ###making plot for false negative rate vs. tranin/test splits

```
##false negative rate
dat3<-data.frame(
  splits = factor(c("20/80", "40/60", "50/50", "60/40", "80/20", "90/10", "95
/5")),
  accuracy =c(46.92, 44.75, 45.34, 44, 44.55, 42.76, 42.13)
)
# Basic line graph with points
ggplot(data=dat3, aes(x=splits, y=accuracy)) +
    geom_line() +
    geom_point()

## geom_path: Each group consists of only one observation. Do you need to
## adjust the group aesthetic?
```




```
# A line graph
ggplot(data=dat3, aes(x=splits, y=accuracy, group=1 )) +
    geom_line() +      # Set linetype by accuracy
    geom_point(size=5, fill="orange") +         # Use larger points, fill wit
h white
    expand_limits(y=c(40, 50)) +                         # Set y range to inclu
de 0
    scale_colour_hue(name="train/test",      # Set legend title
                     l=30)  +                # Use darker colors (lightness
=30)
    scale_shape_manual(name="train/test",
                       values=c(22,20)) +      # Use points with a fill color
    scale_linetype_discrete(name="train/test") +
    xlab("train/test splits") + ylab("false negative") + # Set axis labels
    ggtitle("Different train/test splits on false negative") +     # Set titl
e
    theme_bw() +
    theme(legend.position=c(.7, .4))           # Position legend inside
```



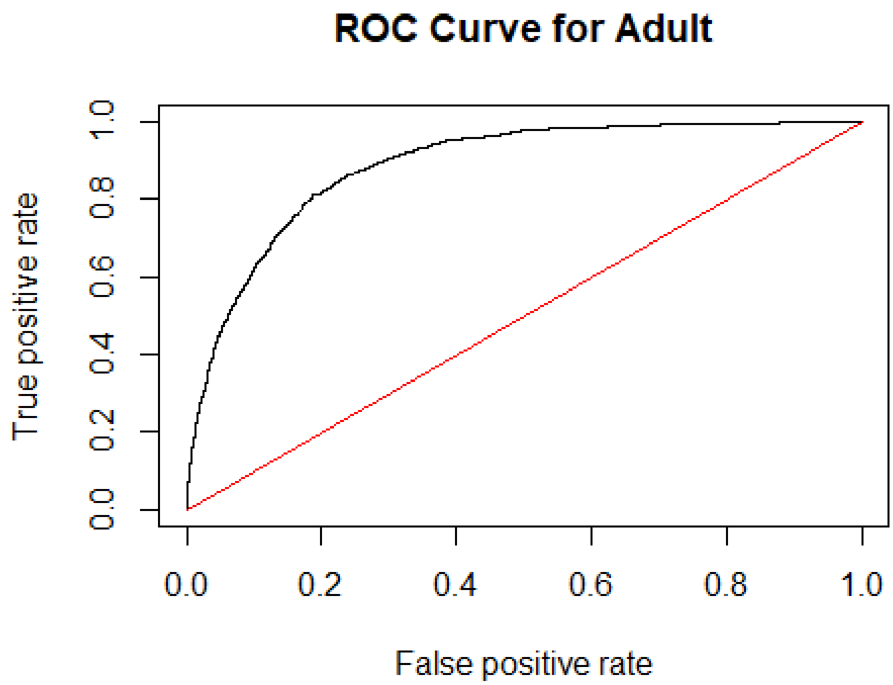

```
                                                    # This must go after theme_bw
```

Model evaluation

Adopt train80%/test20% to see the cutoff at 0.2, 0.5, 0.7 and 0.9

```
###Use train(80%)/test(20%) split, check cutoff 0.2, 0.5, 0.7, 0.9
##set the random number generator so same results can be reproduced
set.seed(2019)
##choose the observations to be in the training. I am splitting the dataset i
nto halves
sample<-sample.int(nrow(data), floor(.80*nrow(data)), replace = F)
train<-data[sample, ]
test<-data[-sample, ]
##use training data to fit logistic regression model with fare and gender as
predictors
result<-glm(income ~age+wc+edu_num+occup+sex+hours_w+ marital + race, family
= "binomial", data = train)
library(ROCR)
##predicted survival rate for testing data based on training data
preds<-predict(result,newdata=test, type="response")

##produce the numbers associated with classification table
rates<-prediction(preds, test$income)

##store the true positive and false postive rates
roc_result<-performance(rates,measure="tpr", x.measure="fpr")
```

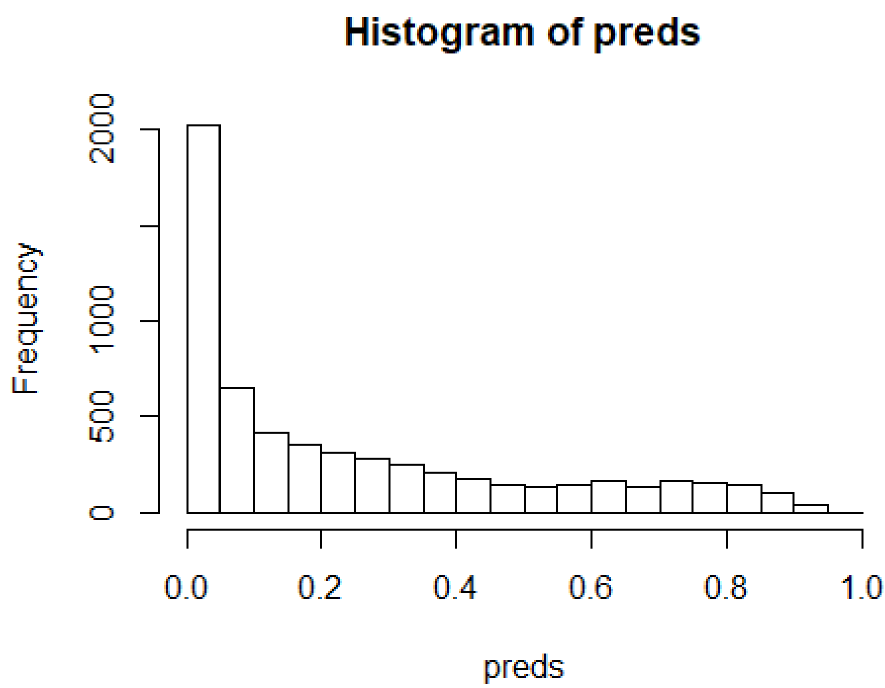
```r
##plot ROC curve and overlay the diagonal line for random guessing
plot(roc_result, main="ROC Curve for Adult")
lines(x = c(0,1), y = c(0,1), col="red")
```




```r
##compute the AUC
auc<-performance(rates, measure = "auc")
auc
```

```
## An object of class "performance"
## Slot "x.name":
## [1] "None"
##
## Slot "y.name":
## [1] "Area under the ROC curve"
##
## Slot "alpha.name":
## [1] "none"
##
## Slot "x.values":
## list()
##
## Slot "y.values":
## [[1]]
## [1] 0.8865583
##
##
```

```
## Slot "alpha.values":
## list()

hist(preds)
```




```
##confusion matrix. Actual values in the rows, predicted classification in co
ls
table(test$income, preds>0.2)

##
##          FALSE TRUE
##    <=50K  3281 1248
##    >50K    175 1329

table(test$income, preds>0.4)

##
##          FALSE TRUE
##    <=50K  4008  521
##    >50K    517  987

table(test$income, preds>0.5)

##
##          FALSE TRUE
##    <=50K  4174  355
##    >50K    670  834
```

```r
table(test$income, preds>0.6)
```

```
##
##          FALSE TRUE
##    <=50K  4302  227
##    >50K    822  682
```

```r
table(test$income, preds>0.7)
```

```
##
##          FALSE TRUE
##    <=50K  4397  132
##    >50K   1027  477
```

Model evaluation

calculate accuacy vs different cutoff

```r
###accuracy for cutoff 0.2
(3281+1329)/(3281+1329+175+1248)
```

```
## [1] 0.7641306
```

```r
#accuracy for cutoff 0.4
(4008+987)/(4008+987+521+517)
```

```
## [1] 0.8279463
```

```r
#accuracy for cutoff 0.5
(4174+834)/(4174+834+355+670)
```

```
## [1] 0.8301011
```

```r
#accuracy for cutoff 0.6
(4302+682)/(4302+682+227+822)
```

```
## [1] 0.826123
```

```r
#accuracy for cutoff 0.7
(4398+477)/(4398+477+132+1027)
```

```
## [1] 0.8079218
```

Model evaluation - calculate false positive at different cutoff

```r
###false positive at varied cutoff
###false positive for cutoff 0.2
(1248)/(1248+3281)
```

```
## [1] 0.2755575
```

```r
#false positivefor cutoff 0.4
(521)/(521+4008)
```

```
## [1] 0.1150364

#false positivefor cutoff 0.5
(355)/(355+4174)

## [1] 0.07838375

#false positive for cutoff 0.6
(227)/(227+4302)

## [1] 0.05012144

#false positive for cutoff 0.7
(132)/(132+4397)

## [1] 0.02914551
```

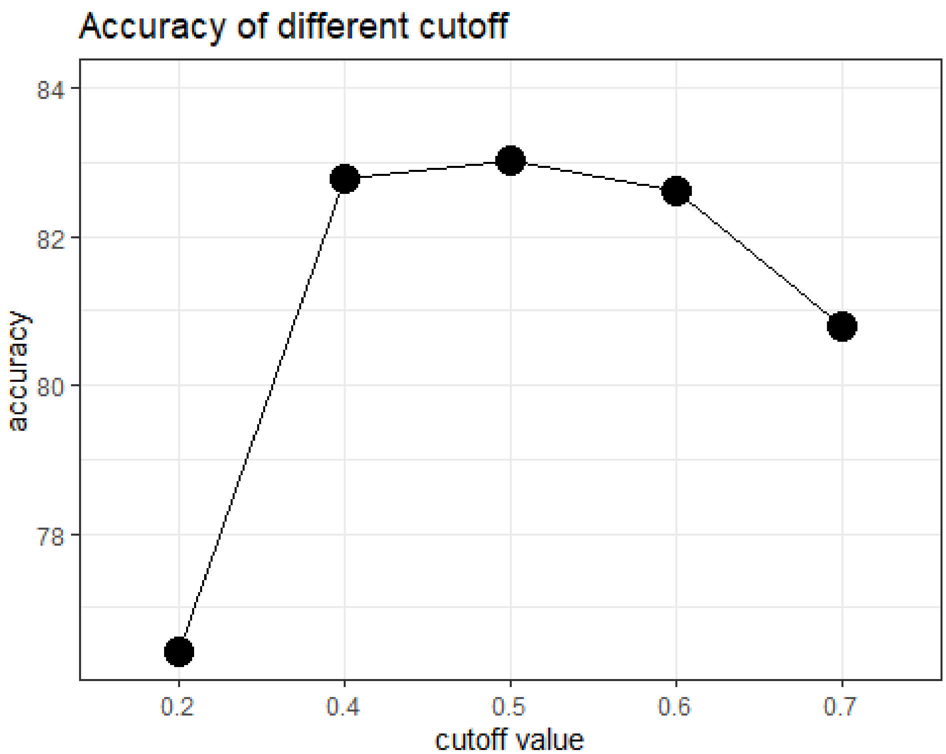
Model evaluation - calculaate false negative at differnt cutoff

```
##false negative at varied cutoff
###false negative for cutoff 0.2
(175)/(175+1329)

## [1] 0.1163564

#false negativefor cutoff 0.4
(517)/(517+987)

## [1] 0.34375

#false negative for cutoff 0.5
(670)/(670+834)

## [1] 0.4454787

#false negative for cutoff 0.6
(822)/(822+682)

## [1] 0.5465426

#false negative for cutoff 0.7
(1027)/(1027+477)

## [1] 0.6828457
```

Model evaluation - making plot: cutoff vs. accuracy

```
dat4<-data.frame(
  cutoff = factor(c(0.2, 0.4, 0.5, 0.6, 0.7)),
  accuracy =c(76.41, 82.79, 83.01, 82.61, 80.79)
)
# A line graph
ggplot(data=dat4, aes(x=cutoff, y=accuracy, group=1 )) +
    geom_line() +     # Set linetype by accuracy
    geom_point(size=5, fill="orange") +         # Use larger points, fill wit
```

```
h white
    expand_limits(y=c(82, 84)) +                # Set y range to include 0
    scale_colour_hue(name="train/test",         # Set legend title
                     l=30)  +                   # Use darker colors (lightnes
s=30)
    scale_shape_manual(name="train/test",
                       values=c(22,20)) +       # Use points with a fill colo
r
    scale_linetype_discrete(name="train/test") +
    xlab("cutoff value") + ylab("accuracy") +   # Set axis labels
    ggtitle("Accuracy of different cutoff") +   # Set title
    theme_bw() +
    theme(legend.position=c(.7, .4))            # Position legend inside
```

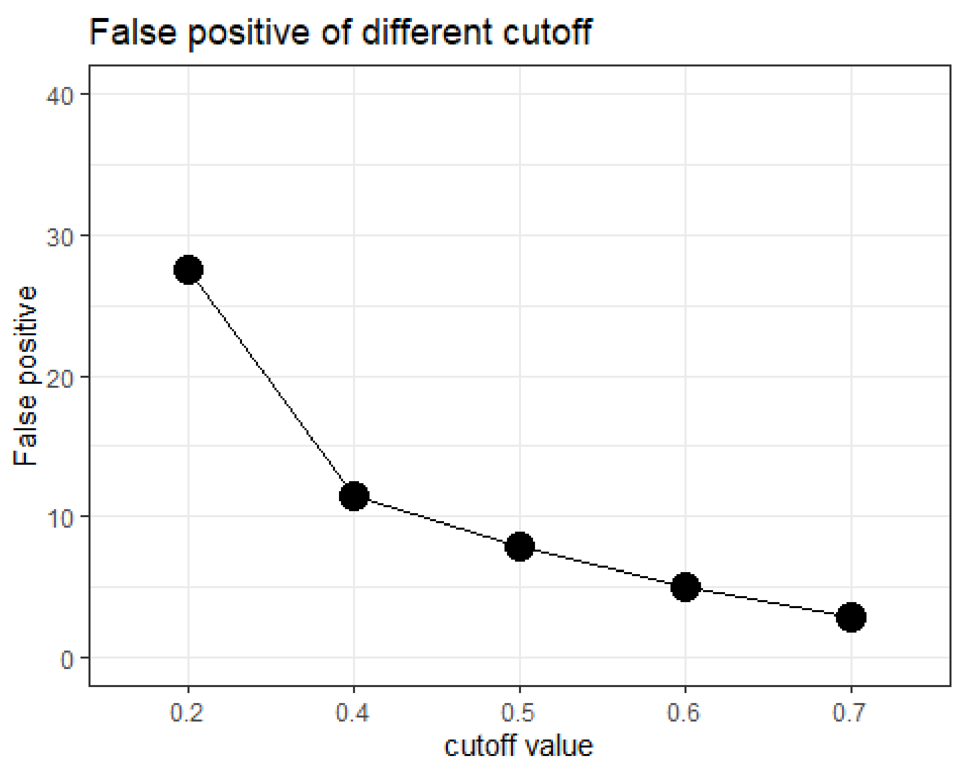



```
                                                # This must go after theme_bw
```

Model evaluation - making plot: cutoff vs. false positive

```
##plot false positive
# A line graph
##plot false positive
dat5<-data.frame(
  cutoff = factor(c(0.2, 0.4, 0.5, 0.6, 0.7)),
  accuracy =c(27.55, 11.50, 7.84, 5.01, 2.91)
)
ggplot(data=dat5, aes(x=cutoff, y=accuracy, group=1 )) +
    geom_line() +      # Set linetype by accuracy
```

```
    geom_point(size=5, fill="orange") +          # Use larger points, fill wit
h white
    expand_limits(y=c(0, 40)) +                 # Set y range to include 0
    scale_colour_hue(name="train/test",          # Set legend title
                     l=30)  +                    # Use darker colors (lightnes
s=30)
    scale_shape_manual(name="train/test",
                       values=c(22,20)) +        # Use points with a fill colo
r
    scale_linetype_discrete(name="train/test") +
    xlab("cutoff value") + ylab("False positive") +    # Set axis labels
    ggtitle("False positive of different cutoff") +    # Set title
    theme_bw() +
    theme(legend.position=c(.7, .4))             # Position legend inside
```

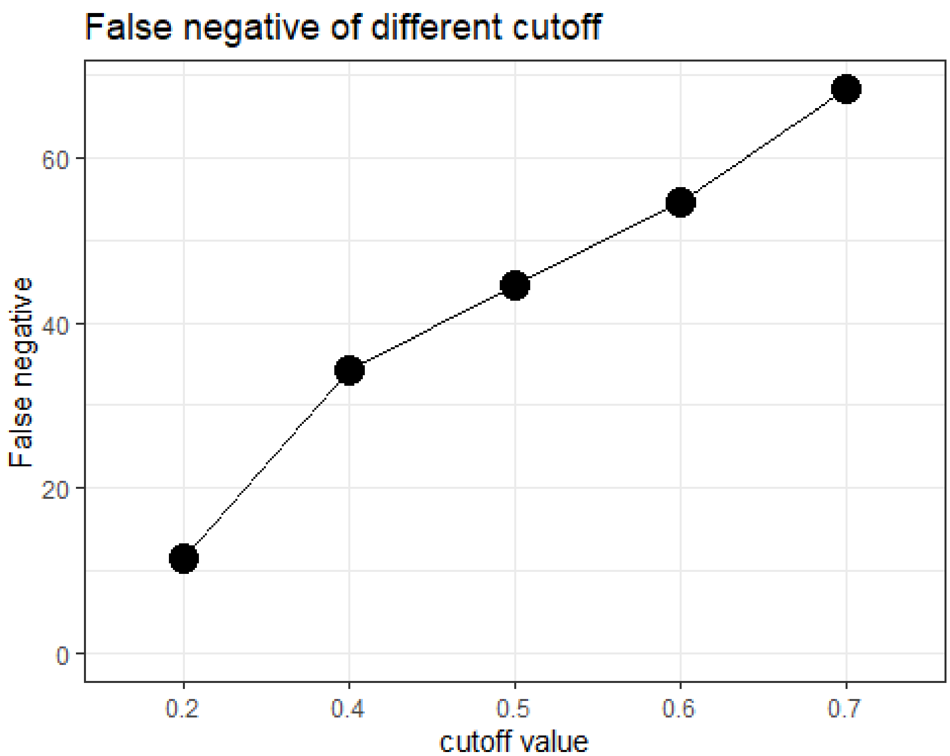



```
                                                 # This must go after theme_bw
```

Model evaluation - making plot: cutoff vs. false negative

```
##plot false negative
dat6<-data.frame(
  cutoff = factor(c(0.2, 0.4, 0.5, 0.6, 0.7)),
  accuracy =c(11.63, 34.38, 44.55, 54.65, 68.28)
)
# A line graph
ggplot(data=dat6, aes(x=cutoff, y=accuracy, group=1 )) +
    geom_line() +      # Set linetype by accuracy
```

```
    geom_point(size=5, fill="orange") +         # Use larger points, fill wit
h white
    expand_limits(y=c(0, 40)) +                # Set y range to include 0
    scale_colour_hue(name="train/test",         # Set legend title
                     l=30)  +                   # Use darker colors (lightnes
s=30)
    scale_shape_manual(name="train/test",
                       values=c(22,20)) +       # Use points with a fill colo
r
    scale_linetype_discrete(name="train/test") +
    xlab("cutoff value") + ylab("False negative") +    # Set axis labels
    ggtitle("False negative of different cutoff") +    # Set title
    theme_bw() +
    theme(legend.position=c(.7, .4))            # Position legend inside
```




```
                                                # This must go after theme_bw
```

K-fold cross validation

```
library(tidyverse)

## -- Attaching packages ---------------------------------------------------
--------------------------- tidyverse 1.2.1 --

## v tibble  2.1.3     v readr   1.3.1
## v tidyr   0.8.3     v purrr   0.3.2
## v tibble  2.1.3     v forcats 0.4.0
```

```
## -- Conflicts -------------------------------------------------------------
---------------------- tidyverse_conflicts() --
## x plyr::arrange()   masks dplyr::arrange()
## x purrr::compact()  masks plyr::compact()
## x plyr::count()     masks dplyr::count()
## x plyr::failwith()  masks dplyr::failwith()
## x dplyr::filter()   masks stats::filter()
## x plyr::id()        masks dplyr::id()
## x dplyr::lag()      masks stats::lag()
## x purrr::lift()     masks caret::lift()
## x plyr::mutate()    masks dplyr::mutate()
## x plyr::rename()    masks dplyr::rename()
## x plyr::summarise() masks dplyr::summarise()
## x plyr::summarize() masks dplyr::summarize()

library(caret)
```

K-fold cross validation

set seed 2019 and k=2 to do cross-validation

```
set.seed(2019)
train.control <- trainControl(method = "cv", number = 2)
# Train the model
model <- train(income ~ age+wc+edu_num+occup+sex+hours_w+ marital + race, dat
a = data, method = "glm",
               trControl = train.control)
# Summarize the results
print(model)

## Generalized Linear Model
##
## 30162 samples
##     8 predictor
##     2 classes: ' <=50K', ' >50K'
##
## No pre-processing
## Resampling: Cross-Validated (2 fold)
## Summary of sample sizes: 15081, 15081
## Resampling results:
##
##   Accuracy   Kappa
##   0.8242159  0.4949525
```

K-fold cross validation

set seed 2019 and k=5 to do cross-validation

```
set.seed(2019)
train.control <- trainControl(method = "cv", number = 5)
# Train the model
```

```r
model <- train(income ~ age+wc+edu_num+occup+sex+hours_w+ marital + race, dat
a = data, method = "glm",
               trControl = train.control)
# Summarize the results
print(model)
```

```
## Generalized Linear Model
##
## 30162 samples
##     8 predictor
##     2 classes: ' <=50K', ' >50K'
##
## No pre-processing
## Resampling: Cross-Validated (5 fold)
## Summary of sample sizes: 24129, 24130, 24130, 24129, 24130
## Resampling results:
##
##   Accuracy   Kappa
##   0.8245475  0.4949273
```

K-fold cross validation

set seed 2019 and k=10 to do cross-validation

```r
set.seed(2019)
train.control <- trainControl(method = "cv", number = 10)
# Train the model
model <- train(income ~ age+wc+edu_num+occup+sex+hours_w+ marital + race, dat
a = data, method = "glm",
               trControl = train.control)
# Summarize the results
print(model)
```

```
## Generalized Linear Model
##
## 30162 samples
##     8 predictor
##     2 classes: ' <=50K', ' >50K'
##
## No pre-processing
## Resampling: Cross-Validated (10 fold)
## Summary of sample sizes: 27146, 27147, 27146, 27146, 27145, 27146, ...
## Resampling results:
##
##   Accuracy   Kappa
##   0.8248459  0.4958799
```

K-fold cross validation

set seed 2019 and k=10 to do cross-validation - repeat three times

```
# Define training control
set.seed(2019)
train.control <- trainControl(method = "repeatedcv",
                              number = 10, repeats = 3)
# Train the model
model <- train(income ~ age+wc+edu_num+occup+sex+hours_w+ marital + race, dat
a = data, method = "glm",
               trControl = train.control)
# Summarize the results
print(model)
```

```
## Generalized Linear Model
##
## 30162 samples
##     8 predictor
##     2 classes: ' <=50K', ' >50K'
##
## No pre-processing
## Resampling: Cross-Validated (10 fold, repeated 3 times)
## Summary of sample sizes: 27146, 27147, 27146, 27146, 27145, 27146, ...
## Resampling results:
##
##   Accuracy   Kappa
##   0.8243376  0.494393
```

K-fold cross validation

set seed 2019 and k=500 to do cross-validation

```
set.seed(2019)
train.control <- trainControl(method = "cv", number = 500)
# Train the model
model <- train(income ~ age+wc+edu_num+occup+sex+hours_w+ marital + race, dat
a = data, method = "glm",
               trControl = train.control)
# Summarize the results
print(model)
```

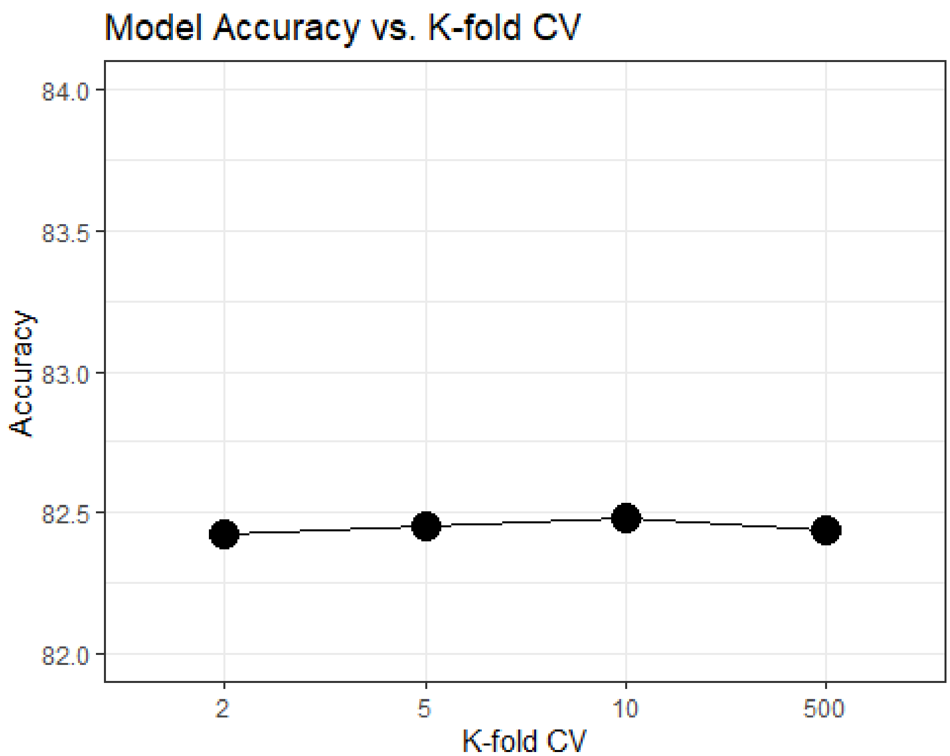
```
## Generalized Linear Model
##
## 30162 samples
##     8 predictor
##     2 classes: ' <=50K', ' >50K'
##
## No pre-processing
## Resampling: Cross-Validated (500 fold)
## Summary of sample sizes: 30101, 30101, 30102, 30102, 30102, 30102, ...
```

```
## Resampling results:
##
##   Accuracy   Kappa
##   0.8244834  0.4922038
```

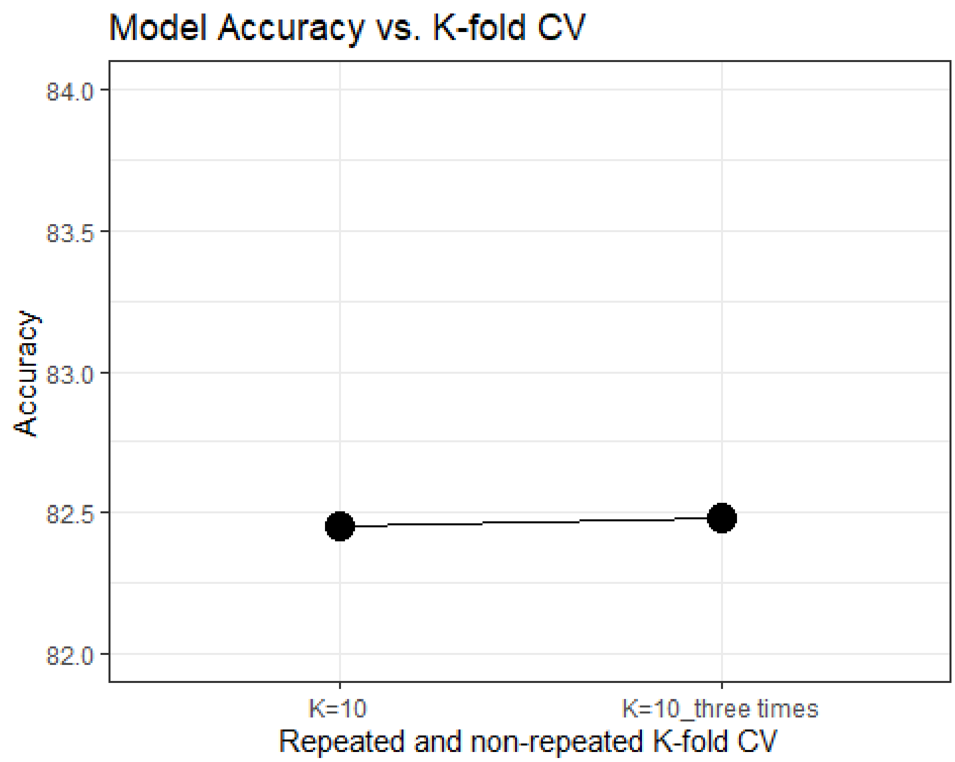
K-fold cross validation

Plot model accuracy vs. k value

```
# A line graph
##false positive rate
dat2<-data.frame(
  splits = factor(c(K=2, K=5, K=10, K=500)),
  accuracy =c(82.42, 82.45, 82.48, 82.44)
)
ggplot(data=dat2, aes(x=splits, y=accuracy, group=1 )) +
    geom_line() +     # Set linetype by accuracy
    geom_point(size=5, fill="orange") +        # Use larger points, fill wit
h white
    expand_limits(y=c(82, 84)) +                      # Set y range to inclu
de 0
    scale_colour_hue(name="cv",      # Set legend title
                     l=30)  +                # Use darker colors (lightness
=30)
    scale_shape_manual(name="cv",
                       values=c(22,20)) +      # Use points with a fill color
    scale_linetype_discrete(name="train/test") +
    xlab("K-fold CV") + ylab("Accuracy") + # Set axis labels
    ggtitle("Model Accuracy vs. K-fold CV") +     # Set title
    theme_bw() +
    theme(legend.position=c(.7, .4))           # Position legend inside
```




```
                                                            # This must go after theme_bw

# A line graph
##false positive rate
dat2<-data.frame(
  splits = factor(c("K=10", "K=10_three times")),
  accuracy =c(82.45, 82.48)
)
ggplot(data=dat2, aes(x=splits, y=accuracy, group=1 )) +
    geom_line() +      # Set linetype by accuracy
    geom_point(size=5, fill="orange") +         # Use larger points, fill wit
h white
    expand_limits(y=c(82, 84)) +                       # Set y range to inclu
de 0
    scale_colour_hue(name="cv",      # Set legend title
                     l=30)  +                  # Use darker colors (lightness
=30)
    scale_shape_manual(name="cv",
                       values=c(22,20)) +      # Use points with a fill color
    scale_linetype_discrete(name="train/test") +
    xlab("Repeated and non-repeated K-fold CV") + ylab("Accuracy") + # Set ax
is labels
    ggtitle("Model Accuracy vs. K-fold CV") +     # Set title
    theme_bw() +
    theme(legend.position=c(.7, .4))           # Position legend inside
```




```
                                                  # This must go after theme_bw
```

case study

```
###second model: 25 years old, workclass: government,educ-num = 16y, White-co
llar, female, hours-per-W: 40, married, White
logodds_2 = -8.8236 + 0.02972*45 + 0 + 0.2989*16 + 0.7873 + 0.02910 *40 -0.49
06 + 0.5551
logodds_2

## [1] -0.688

exp(logodds_2)

## [1] 0.5025802

prob_2 = exp(logodds_2)/(1+exp(logodds_2))
prob_2

## [1] 0.3344781
```

case study

```
## (2) first model
#calculate estimate odds for age while holding all other predictors constant
#Considering a male at age 40 years old with workclass = government, educatio
n number  = 16 years, occupation is White-Collar, hours-per-week is 40 hrs, e
ducation is Doctoral, relationship is Wife, maritial-status is married, race
is White:
```

```r
logodds = -7.14719+0.02839* 40 + 0 + 0.2132*16 + 0.8062 + 0.8968 + 0.02912*40
+ 0.9703 + 1.3258 + 0.5728 + 0.5270
logodds
```

```
## [1] 3.66331
```

```r
estimatedodds = exp(logodds)
prob = estimatedodds/(1+estimatedodds)
prob
```

```
## [1] 0.9749939
```