Census Income

Contents

Introduction

For this project, I develop classification models using several approaches and compare their performance on dataset "Census Income" from UCI ML. I will try to predict whether a household income is greater than or less than $50,000.

Data Summary, Univariate Analysis and Unsupervised Analysis

Before performing multivariate supervised analysis, I explore and clean the data.

Read in Data

The original data has been split up into training and test subsets. I combine the two subsets and rename columns and factors as necessary as well as read in the table of results found in the dataset description.

```
Tmp1Data=read.table("Datasets/adult.data",sep=",")
Tmp2Data=read.table("Datasets/adult.test",sep=",") #manually removed header in data file
DescPfrm=read.table("Datasets/adultresults.csv",sep=",",header = TRUE) #results from dataset descriptio

colnames(Tmp1Data)=c("age","workclass","fnlwgt","education","education-num","marital-status","occupatio
Tmp1Data=Tmp1Data[,-3]
dim(Tmp1Data)
```

```
## [1] 32561    14
```

```
summary(Tmp1Data)
```

```
##       age                     workclass             education
##  Min.   :17.00    Private         :22696    HS-grad     :10501
##  1st Qu.:28.00    Self-emp-not-inc: 2541    Some-college: 7291
##  Median :37.00    Local-gov       : 2093    Bachelors   : 5355
##  Mean   :38.58    ?               : 1836    Masters     : 1723
##  3rd Qu.:48.00    State-gov       : 1298    Assoc-voc   : 1382
##  Max.   :90.00    Self-emp-inc    : 1116    11th        : 1175
##                  (Other)          :  981   (Other)      : 5134
##  education-num                  marital-status             occupation
##  Min.   : 1.00    Divorced             : 4443    Prof-specialty :4140
##  1st Qu.: 9.00    Married-AF-spouse    :   23    Craft-repair   :4099
##  Median :10.00    Married-civ-spouse   :14976    Exec-managerial:4066
##  Mean   :10.08    Married-spouse-absent:  418    Adm-clerical   :3770
##  3rd Qu.:12.00    Never-married        :10683    Sales          :3650
##  Max.   :16.00    Separated            : 1025    Other-service  :3295
##                   Widowed              :  993   (Other)         :9541
##           relationship                    race           sex
##   Husband       :13193    Amer-Indian-Eskimo:  311    Female:10771
##   Not-in-family : 8305    Asian-Pac-Islander: 1039    Male  :21790
##   Other-relative:  981    Black             : 3124
##   Own-child     : 5068    Other             :  271
##   Unmarried     : 3446    White             :27816
##   Wife          : 1568
##
##   capital-gain    capital-loss    hours-per-week         native-country
##  Min.   :    0   Min.   :   0.0   Min.   : 1.00    United-States:29170
##  1st Qu.:    0   1st Qu.:   0.0   1st Qu.:40.00    Mexico       :  643
##  Median :    0   Median :   0.0   Median :40.00    ?            :  583
##  Mean   : 1078   Mean   :  87.3   Mean   :40.44    Philippines  :  198
##  3rd Qu.:    0   3rd Qu.:   0.0   3rd Qu.:45.00    Germany      :  137
##  Max.   :99999   Max.   :4356.0   Max.   :99.00    Canada       :  121
##                                                   (Other)      : 1709
##    outcome
##   <=50K:24720
##   >50K : 7841
##
##
##
##
##
```

```
colnames(Tmp2Data)=c("age","workclass","fnlwgt","education","education-num","marital-status","occupatio
Tmp2Data=Tmp2Data[,-3]
dim(Tmp2Data)
```

```
## [1] 16281    14
```

```
summary(Tmp2Data)
```

```
##       age                     workclass             education    education-num
##  Min.   :17.00    Private         :11210    HS-grad     :5283   Min.   : 1.00
##  1st Qu.:28.00    Self-emp-not-inc: 1321    Some-college:3587   1st Qu.: 9.00
```

```
##  Median :37.00    Local-gov       : 1043    Bachelors   :2670   Median :10.00
##  Mean   :38.77    ?               :  963    Masters     : 934   Mean   :10.07
##  3rd Qu.:48.00    State-gov       :  683    Assoc-voc   : 679   3rd Qu.:12.00
##  Max.   :90.00    Self-emp-inc    :  579    11th        : 637   Max.   :16.00
##                   (Other)         :  482   (Other)      :2491
##                 marital-status           occupation           relationship
##   Divorced             :2190    Prof-specialty :2032    Husband       :6523
##   Married-AF-spouse    :  14    Exec-managerial:2020    Not-in-family :4278
##   Married-civ-spouse   :7403    Craft-repair   :2013    Other-relative: 525
##   Married-spouse-absent: 210    Sales          :1854    Own-child     :2513
##   Never-married        :5434    Adm-clerical   :1841    Unmarried     :1679
##   Separated            : 505    Other-service  :1628    Wife          : 763
##   Widowed              : 525   (Other)         :4893
##                   race            sex         capital-gain    capital-loss
##   Amer-Indian-Eskimo:  159    Female: 5421   Min.   :    0   Min.   :   0.0
##   Asian-Pac-Islander:  480    Male  :10860   1st Qu.:    0   1st Qu.:   0.0
##   Black             : 1561                   Median :    0   Median :   0.0
##   Other             :  135                   Mean   : 1082   Mean   :  87.9
##   White             :13946                   3rd Qu.:    0   3rd Qu.:   0.0
##                                              Max.   :99999   Max.   :3770.0
##
##  hours-per-week         native-country     outcome
##  Min.   : 1.00    United-States:14662    <=50K.:12435
##  1st Qu.:40.00    Mexico        :  308    >50K. : 3846
##  Median :40.00    ?             :  274
##  Mean   :40.39    Philippines   :   97
##  3rd Qu.:45.00    Puerto-Rico   :   70
##  Max.   :99.00    Germany       :   69
##                   (Other)       :  801
```

```
levels(Tmp2Data$outcome)
```

```
## [1] " <=50K." " >50K."
```

```
levels(Tmp2Data$outcome)=c(" <=50K"," >50K")
levels(Tmp2Data$outcome)
```

```
## [1] " <=50K" " >50K"
```

```
censusDat=rbind(Tmp1Data,Tmp2Data)
###########Subset of data for working only
censusDat=censusDat[sample(c(FALSE,FALSE,FALSE,FALSE,FALSE,FALSE,FALSE,FALSE,FALSE,FALSE,TRUE),nrow(cens
################
dim(censusDat)
```

```
## [1] 4406   14
```

```
summary(censusDat)
```

```
##       age                    workclass            education    education-num
##  Min.   :17.00    Private         :3102    HS-grad     :1427   Min.   : 1.00
##  1st Qu.:28.00    Self-emp-not-inc: 348    Some-college:1002   1st Qu.: 9.00
##  Median :37.00    Local-gov       : 276    Bachelors   : 754   Median :10.00
##  Mean   :38.69    ?               : 242    Masters     : 230   Mean   :10.06
##  3rd Qu.:47.00    State-gov       : 181    Assoc-voc   : 164   3rd Qu.:12.00
##  Max.   :90.00    Self-emp-inc    : 137    11th        : 156   Max.   :16.00
##                   (Other)         : 120   (Other)      : 673
```

```
##                 marital-status           occupation           relationship
##   Divorced             : 596    Craft-repair   : 567    Husband       :1804
##   Married-AF-spouse    :   4    Prof-specialty : 526    Not-in-family :1150
##   Married-civ-spouse   :2048    Exec-managerial: 522    Other-relative: 138
##   Married-spouse-absent:  59    Sales          : 507    Own-child     : 659
##   Never-married        :1425    Adm-clerical   : 497    Unmarried     : 443
##   Separated            : 140    Other-service  : 448    Wife          : 212
##   Widowed              : 134   (Other)         :1339
##                   race           sex         capital-gain    capital-loss
##   Amer-Indian-Eskimo:  42    Female:1447   Min.   :    0   Min.   :   0.00
##   Asian-Pac-Islander: 132    Male  :2959   1st Qu.:    0   1st Qu.:   0.00
##   Black             : 411                  Median :    0   Median :   0.00
##   Other             :  47                  Mean   : 1142   Mean   :  79.92
##   White             :3774                  3rd Qu.:    0   3rd Qu.:   0.00
##                                            Max.   :99999   Max.   :3900.00
##
##  hours-per-week         native-country   outcome
##  Min.   : 1.00    United-States:3938    <=50K:3354
##  1st Qu.:40.00    Mexico       :  89    >50K :1052
##  Median :40.00    ?            :  77
##  Mean   :40.32    Philippines  :  26
##  3rd Qu.:45.00    Puerto-Rico  :  22
##  Max.   :99.00    Germany      :  20
##                  (Other)       : 234
```

```
table(edulvl=censusDat$education,educode=censusDat$`education-num`)
```

```
##                educode
## edulvl             1    2    3    4    5    6    7    8    9   10   11   12
##    10th            0    0    0    0    0  124    0    0    0    0    0    0
##    11th            0    0    0    0    0    0  156    0    0    0    0    0
##    12th            0    0    0    0    0    0    0   58    0    0    0    0
##    1st-4th         0   23    0    0    0    0    0    0    0    0    0    0
##    5th-6th         0    0   45    0    0    0    0    0    0    0    0    0
##    7th-8th         0    0    0   92    0    0    0    0    0    0    0    0
##    9th             0    0    0    0   70    0    0    0    0    0    0    0
##    Assoc-acdm      0    0    0    0    0    0    0    0    0    0    0  147
##    Assoc-voc       0    0    0    0    0    0    0    0    0    0  164    0
##    Bachelors       0    0    0    0    0    0    0    0    0    0    0    0
##    Doctorate       0    0    0    0    0    0    0    0    0    0    0    0
##    HS-grad         0    0    0    0    0    0    0    0 1427    0    0    0
##    Masters         0    0    0    0    0    0    0    0    0    0    0    0
##    Preschool       6    0    0    0    0    0    0    0    0    0    0    0
##    Prof-school     0    0    0    0    0    0    0    0    0    0    0    0
##    Some-college    0    0    0    0    0    0    0    0    0 1002    0    0
##                educode
## edulvl            13   14   15   16
##    10th            0    0    0    0
##    11th            0    0    0    0
##    12th            0    0    0    0
##    1st-4th         0    0    0    0
##    5th-6th         0    0    0    0
##    7th-8th         0    0    0    0
##    9th             0    0    0    0
##    Assoc-acdm      0    0    0    0
```

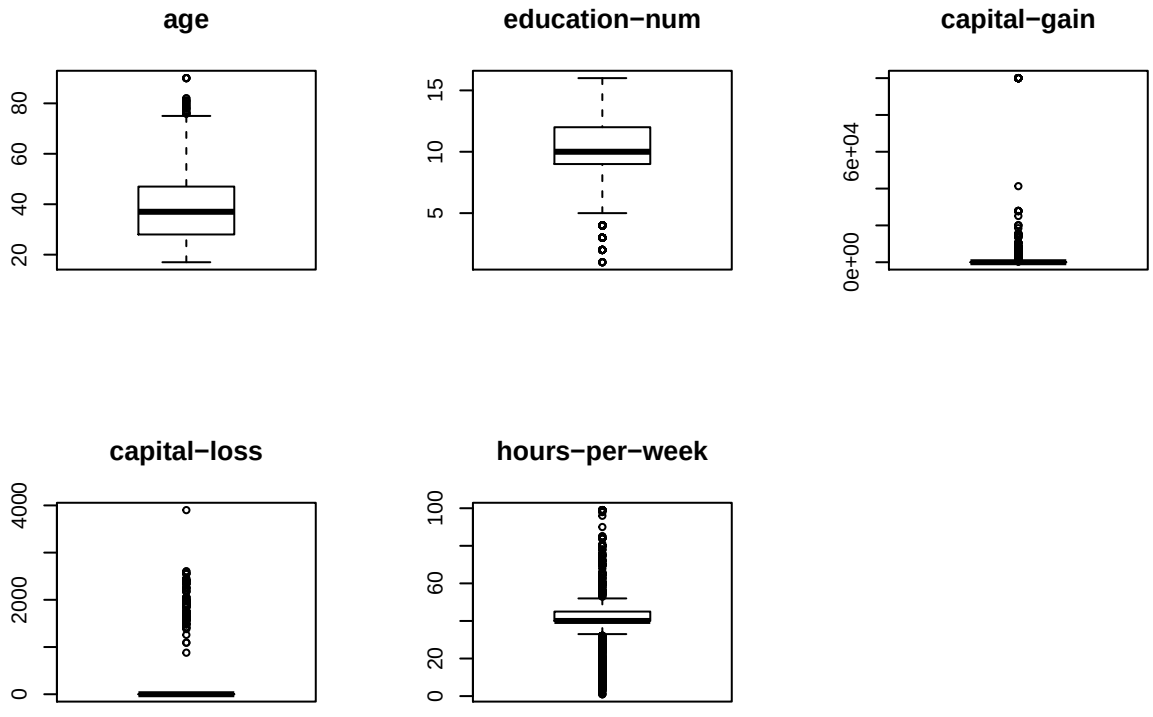
```
##     Assoc-voc       0    0    0    0
##     Bachelors     754    0    0    0
##     Doctorate       0    0    0   52
##     HS-grad         0    0    0    0
##     Masters         0  230    0    0
##     Preschool       0    0    0    0
##     Prof-school     0    0   56    0
##     Some-college    0    0    0    0
```

```
#education-num is discrete and ordinal - will treat as continuous variable
```

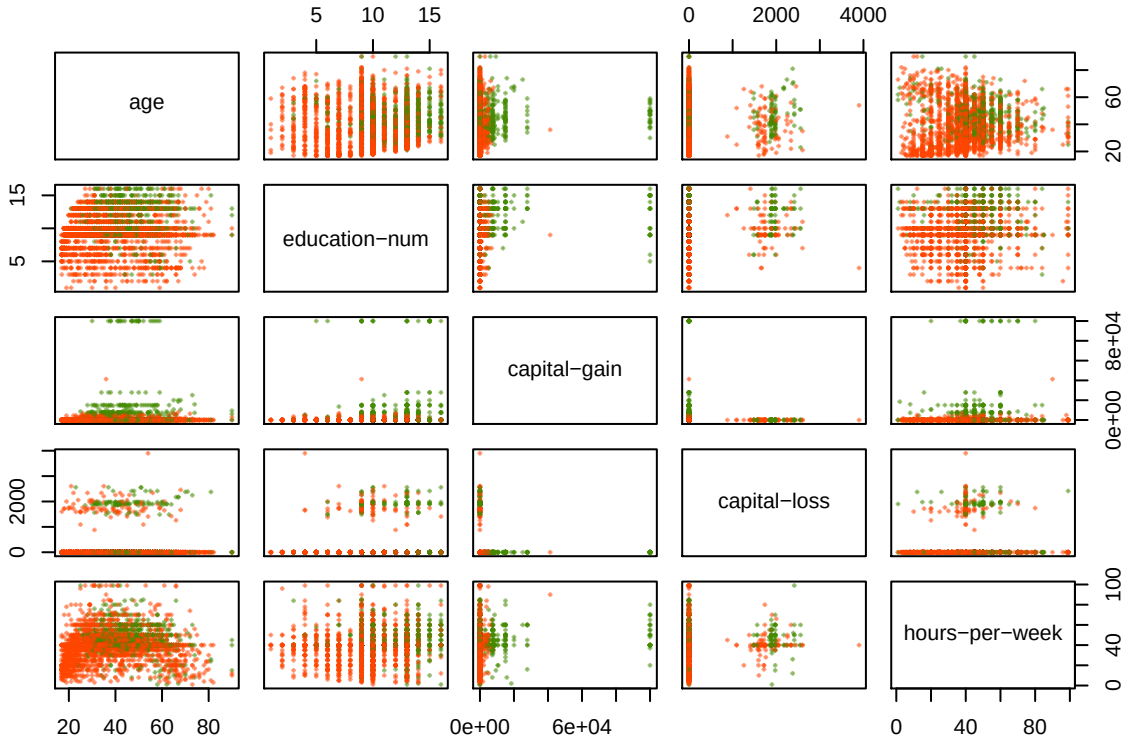
I combined and formatted the split datasets and performed a basic summary of the data. Of note is that there are many more vectors with income below 50k than above. *capital-gain* and *capital-loss* are mostly zero with a few large quantities. There are also some categories which are very sparsely populated and *education-num* and *education* represent the same attribute. I disregard the "final weight" attribute and, because the dataset is large and some of the techniques I will apply are computationally intense, I have taken a sample of the combined dataset.

Summarize Continuous Variables

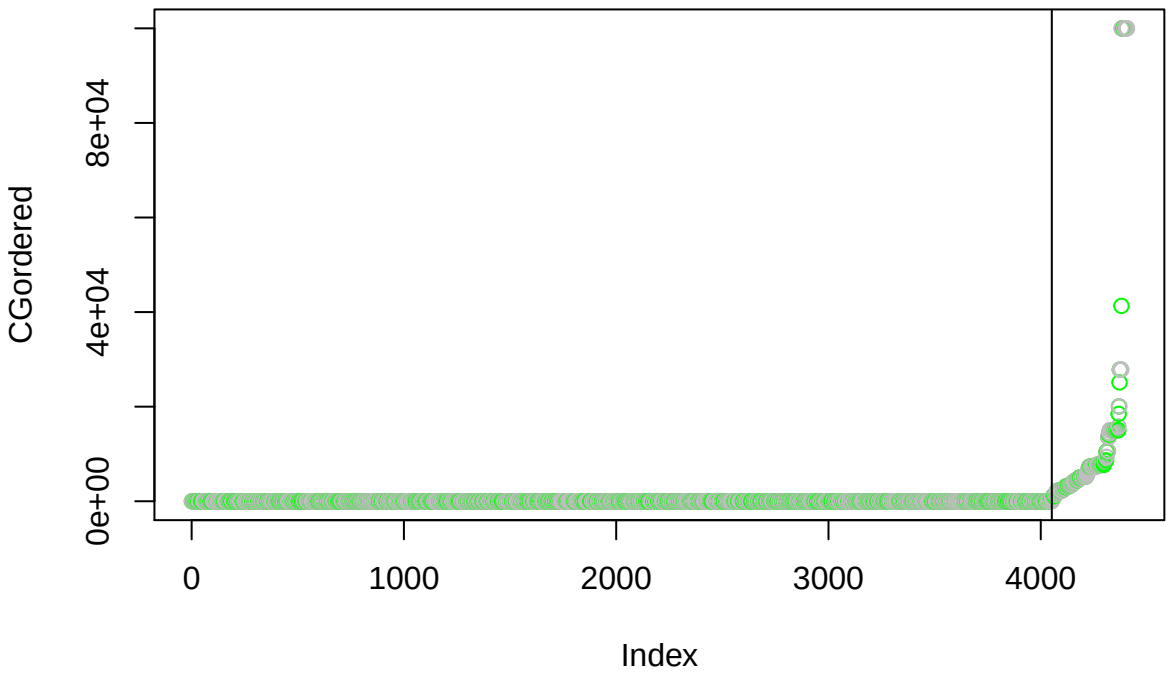
```
#boxplot of continuous variables
old.par=par(mfrow=c(2,3))
for ( clmns in c(1,4,10,11,12)) {
boxplot(censusDat[[clmns]],
main=names(censusDat)[clmns])
}
par(old.par)
```



```
pairs(censusDat[,c(1,4,10,11,12)],pch=10,cex=0.3,col=alpha(c("orangered","chartreuse4")[censusDat$outcor
```




```
orderCG=order(censusDat$`capital-gain`)
CGordered=censusDat$`capital-gain`[orderCG]
MinNon0=which(CGordered != 0)[1]
plot(CGordered,col=c("gray","green")[censusDat$outcome])
abline(v=MinNon0)
```




```
NonOs=which(censusDat$`capital-gain` != 0)
t=censusDat$outcome[NonOs]
summary(t)
```

```
##  <=50K   >50K
##    138    217
```

capital-gain,*capital-loss* and *hours-per-week* all have narrow interquartile ranges and a significant number of outliers. I expect that for all the continuous variables, higher values will be more predictive of higher income - throughout the range of *age* and *education-num* and at the more extreme values for the others. This is further supported by the pairs scatterplots where the largest spreads are with combinations of *age*,*education-num* and *hours-per-week*. It does appear that the data will be classifiable but with a lot of overlap.

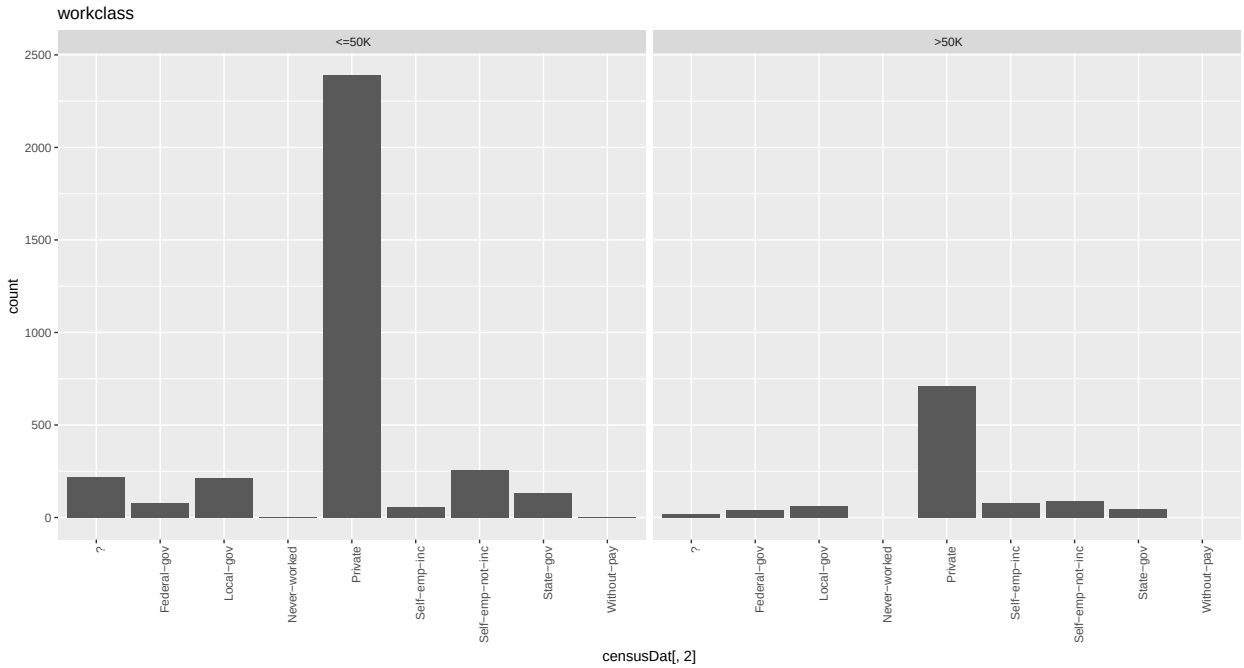
It may be worth discretizing *capital-gain* and *capital-loss*. For the majority of the observations, these are 0, but at least for *capital-gain* there is a much higher percentage over 50k when *capital-gain* is nonzero than in the rest of the population. It seems like most of the information we will get from these two attributes will be contained in whether they are 0 or not.

Summarize Categorical Variables

```
#tables of categorical variables and outcome
CatList=list()
lst=1
for (clmns in c(2,3,5,6,7,8,9,13)) {
  CatList[[lst]]=table(censusDat[,clmns],censusDat$outcome)
  lst=lst+2
}
```

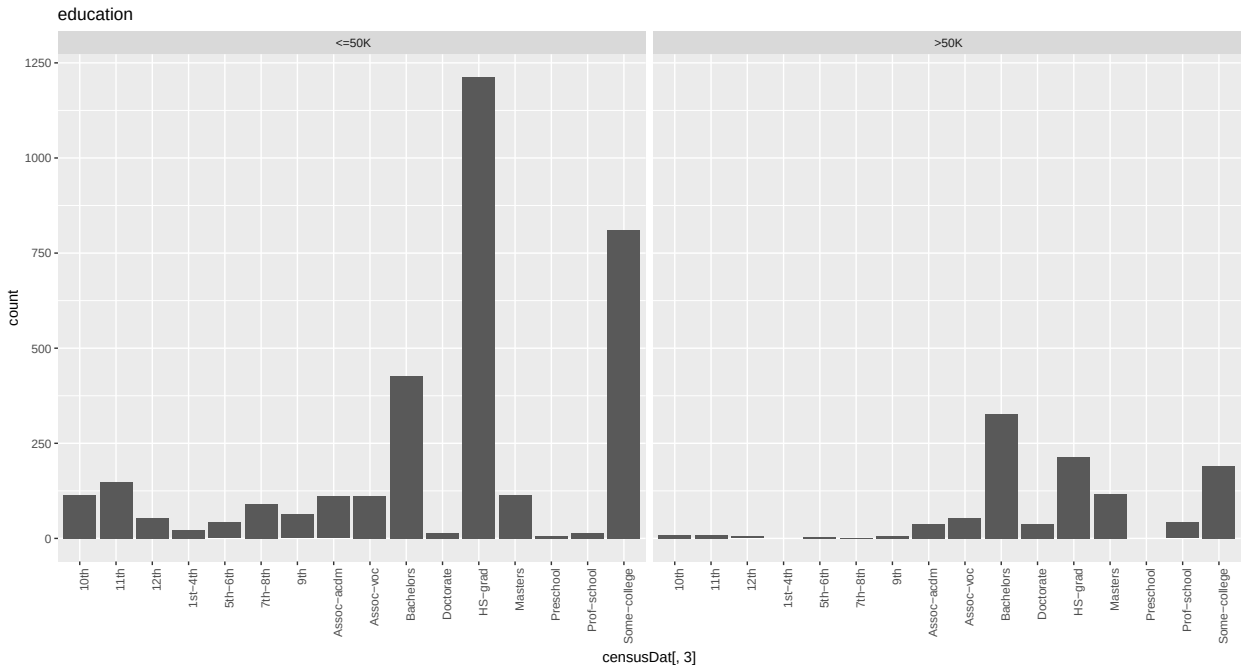
```
#barcharts of categorical variables split by outcome
CatList[[2]]=ggplot(censusDat,aes(censusDat[,2])) + geom_bar() + facet_wrap('outcome') + labs(title =nam
CatList[[4]]=ggplot(censusDat,aes(censusDat[,3])) + geom_bar() + facet_wrap('outcome') + labs(title =nam
CatList[[6]]=ggplot(censusDat,aes(censusDat[,5])) + geom_bar() + facet_wrap('outcome') + labs(title =nam
CatList[[8]]=ggplot(censusDat,aes(censusDat[,6])) + geom_bar() + facet_wrap('outcome') + labs(title =nam
CatList[[10]]=ggplot(censusDat,aes(censusDat[,7])) + geom_bar() + facet_wrap('outcome') + labs(title =na
CatList[[12]]=ggplot(censusDat,aes(censusDat[,8])) + geom_bar() + facet_wrap('outcome') + labs(title =na
CatList[[14]]=ggplot(censusDat,aes(censusDat[,9])) + geom_bar() + facet_wrap('outcome') + labs(title =na
CatList[[16]]=ggplot(censusDat,aes(censusDat[,13])) + geom_bar() + facet_wrap('outcome') + labs(title =n

CatList
```

```
## [[1]]
##
##                       <=50K  >50K
##    ?                    221    21
##    Federal-gov           76    39
##    Local-gov            212    64
##    Never-worked           1     0
##    Private             2390   712
##    Self-emp-inc          58    79
##    Self-emp-not-inc     258    90
##    State-gov            134    47
##    Without-pay            4     0
##
## [[2]]
```

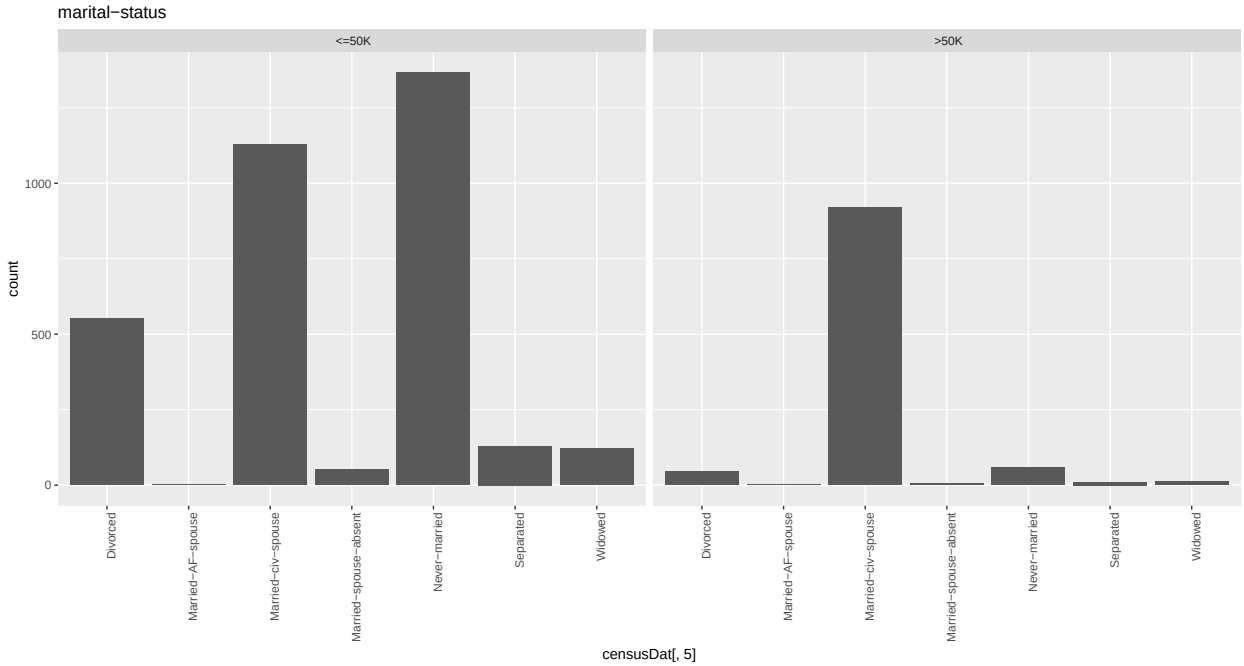



```
##
## [[3]]
##
##                   <=50K  >50K
##    10th             115     9
```

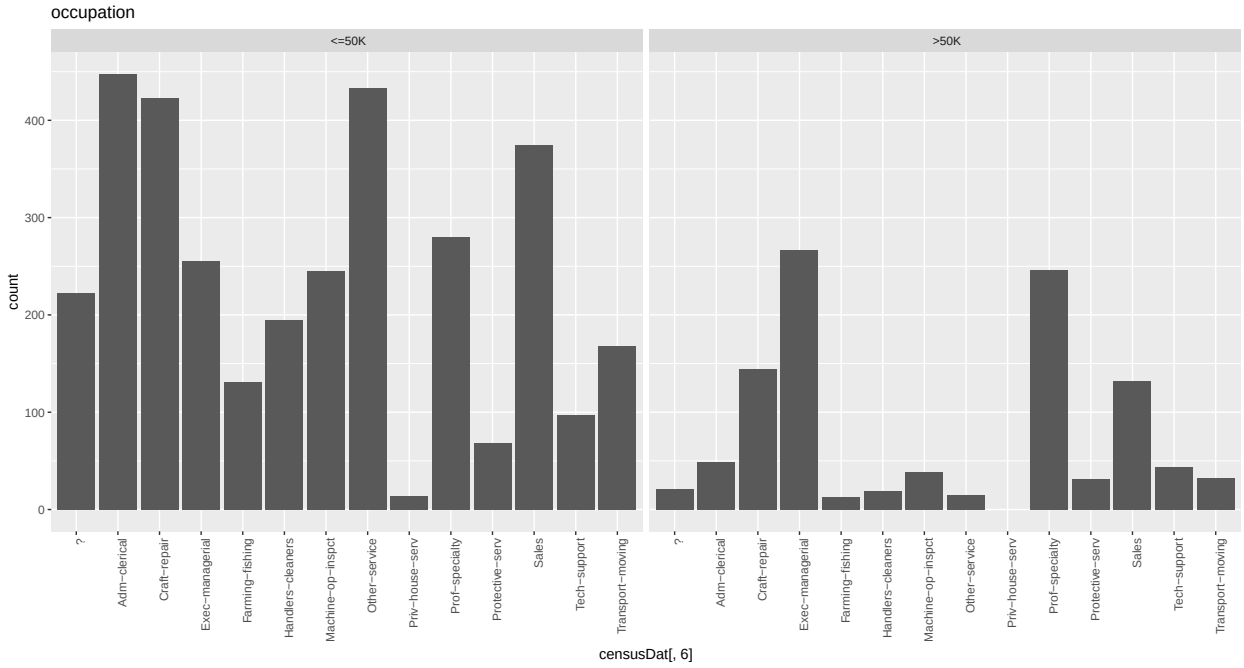
```
##    11th             148     8
##    12th              53     5
##    1st-4th           23     0
##    5th-6th           42     3
##    7th-8th           90     2
##    9th               63     7
##    Assoc-acdm       110    37
##    Assoc-voc        111    53
##    Bachelors        427   327
##    Doctorate         15    37
##    HS-grad         1212   215
##    Masters          114   116
##    Preschool          6     0
##    Prof-school       14    42
##    Some-college     811   191
##
## [[4]]
```



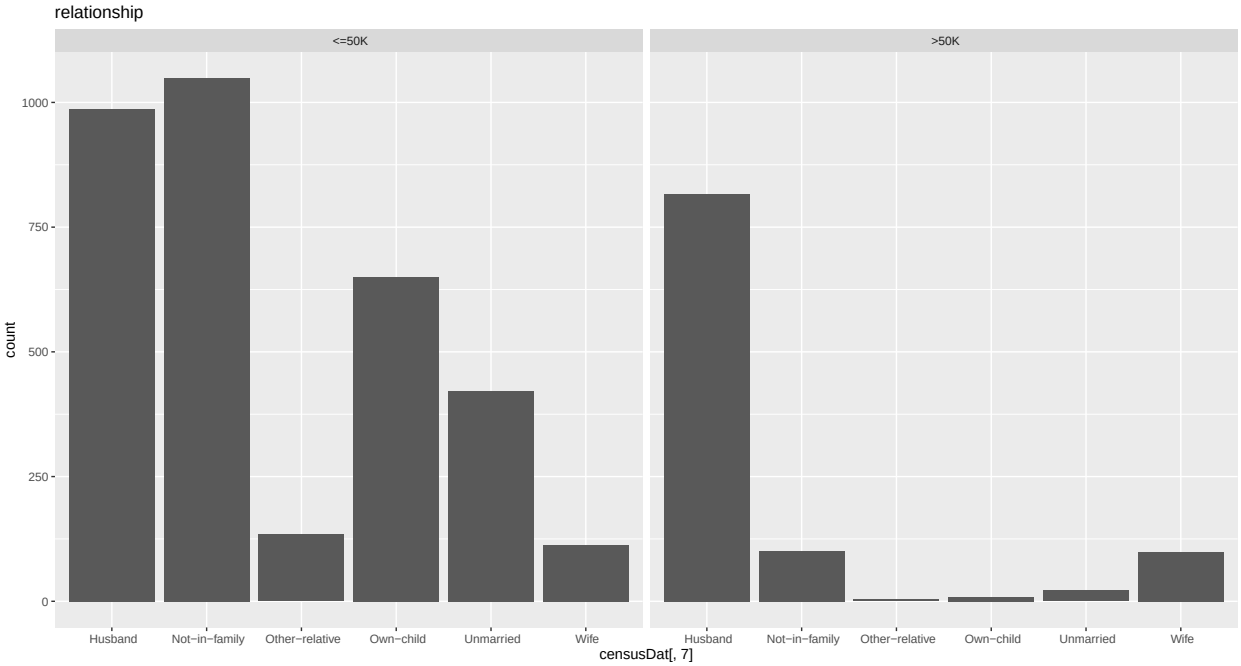

```
##
## [[5]]
##
##                          <=50K  >50K
##    Divorced                551    45
##    Married-AF-spouse         2     2
##    Married-civ-spouse     1129   919
##    Married-spouse-absent    53     6
##    Never-married          1367    58
##    Separated               130    10
##    Widowed                 122    12
##
## [[6]]
```



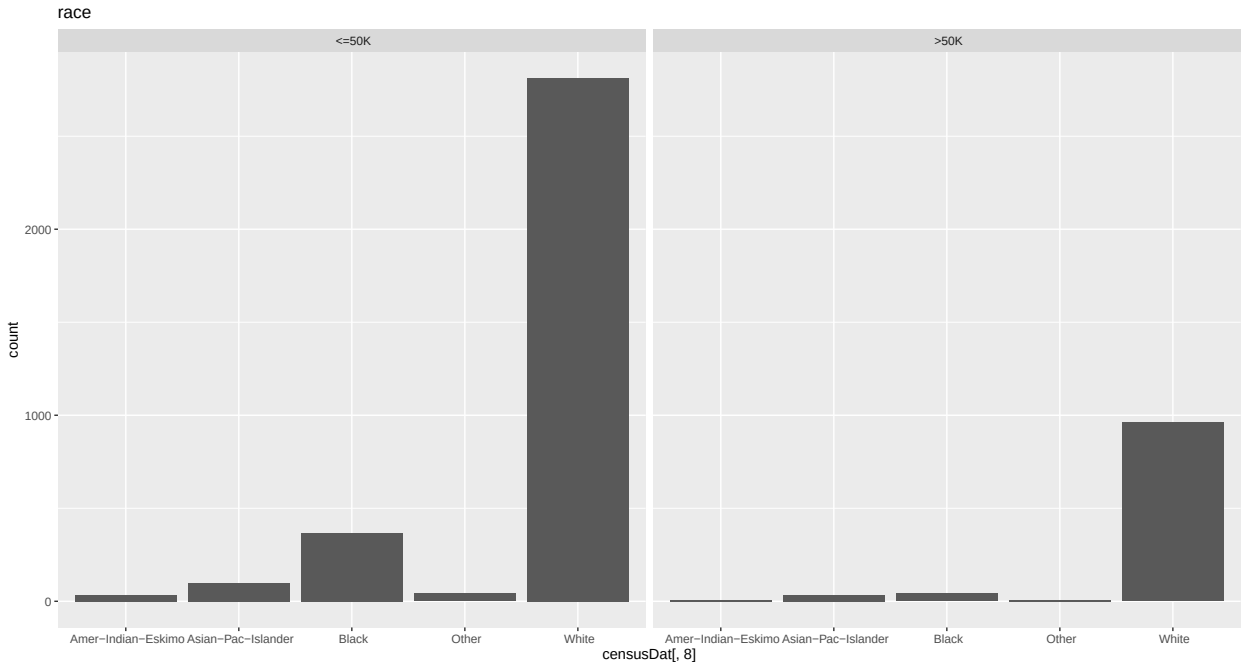

```
##
## [[7]]
##
##                      <=50K  >50K
##    ?                   222    21
##    Adm-clerical        448    49
##    Armed-Forces          0     0
##    Craft-repair        423   144
##    Exec-managerial     255   267
##    Farming-fishing     131    13
##    Handlers-cleaners   195    19
##    Machine-op-inspct   245    39
##    Other-service       433    15
##    Priv-house-serv      14     0
##    Prof-specialty      280   246
##    Protective-serv      68    31
##    Sales               375   132
##    Tech-support         97    44
##    Transport-moving    168    32
##
## [[8]]
```



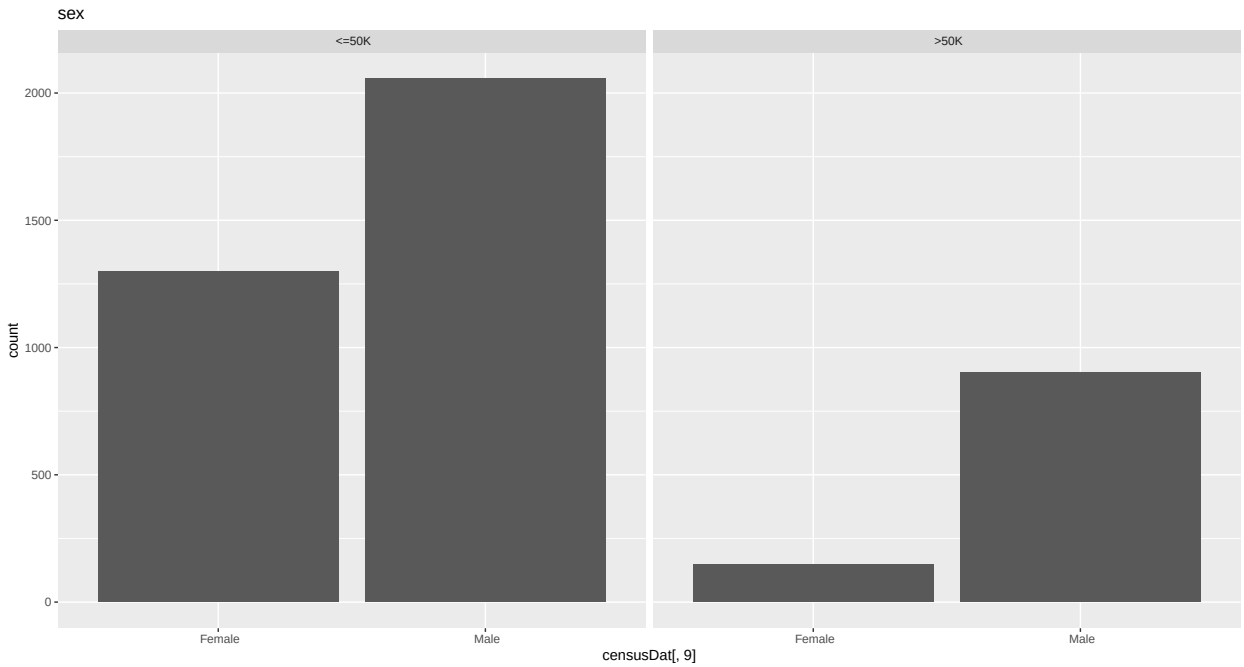

```
##
## [[9]]
##
##                   <=50K  >50K
##    Husband          987   817
##    Not-in-family   1049   101
##    Other-relative   134     4
##    Own-child        650     9
##    Unmarried        421    22
##    Wife             113    99
##
## [[10]]
```




```
##
## [[11]]
##
##                         <=50K  >50K
##    Amer-Indian-Eskimo      34     8
##    Asian-Pac-Islander      98    34
##    Black                  368    43
##    Other                   41     6
##    White                 2813   961
##
## [[12]]
```

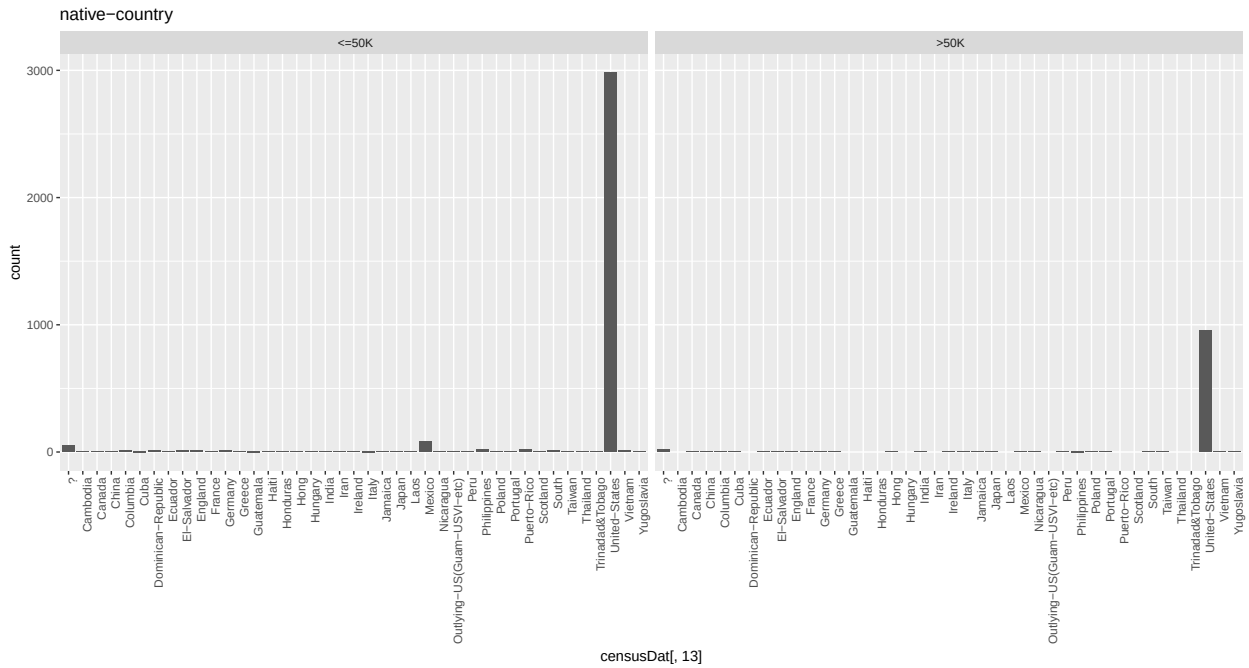



```
##
## [[13]]
##
##            <=50K  >50K
##    Female   1298   149
##    Male     2056   903
##
## [[14]]
```




```
##
## [[15]]
```

```
##
##                               <=50K  >50K
##    ?                             54    23
##    Cambodia                       2     0
##    Canada                         7     6
##    China                          6     3
##    Columbia                       9     1
##    Cuba                           8     4
##    Dominican-Republic             9     0
##    Ecuador                        6     1
##    El-Salvador                   12     2
##    England                       10     3
##    France                         1     1
##    Germany                       14     6
##    Greece                         2     2
##    Guatemala                      8     0
##    Haiti                          5     0
##    Holand-Netherlands             0     0
##    Honduras                       2     0
##    Hong                           3     1
##    Hungary                        1     0
##    India                          5     5
##    Iran                           4     0
##    Ireland                        5     2
##    Italy                          8     3
##    Jamaica                        5     3
##    Japan                          4     1
##    Laos                           3     0
##    Mexico                        83     6
##    Nicaragua                      5     1
##    Outlying-US(Guam-USVI-etc)     3     0
##    Peru                           4     1
##    Philippines                   18     8
##    Poland                         6     5
##    Portugal                       4     1
##    Puerto-Rico                   22     0
##    Scotland                       1     0
##    South                         11     1
##    Taiwan                         2     2
##    Thailand                       6     0
##    Trinadad&Tobago                2     0
##    United-States               2981   957
##    Vietnam                        9     1
##    Yugoslavia                     4     2
##
## [[16]]
```



I plotted tables and bar charts of each categorical variable. The structure of the categories is similar between the under 50k class and the over 50k class. *workclass* has very few observations with "never-worked" and "without-pay". Most of the data is in "Pvt". Also, in *marital-status*, "Married-AF-spouse" is a very small group which seems like would not help with our predictive models. I may regroup these categories. In *occupation*, "Armed-Forces" is a very small group. Because there are so many occupations, it may be worth combing them by job type. For *race*, "white" is by far the most numerous category and seems to have a higher proportion in the over 50k group. *native-country* contains many countries, mostly with low numbers of observations. These should be combined - possibly by region, economic development or US/non US. Only Mexico and the Phillipines have an appreciable number of observations and are still dwarfed the the number from the United States - recategorizing as US/non US may be best.

Clean Data

```
#remove rows with "?"
for(clmns in c("workclass","occupation","native-country")) {
if (dim(censusDat[which(censusDat[,clmns] == " ?"),])[1]!=0)
{ censusDat = censusDat[-which(censusDat[,clmns] == " ?"),] }
}
dim(censusDat)
```

```
## [1] 4088   14
```

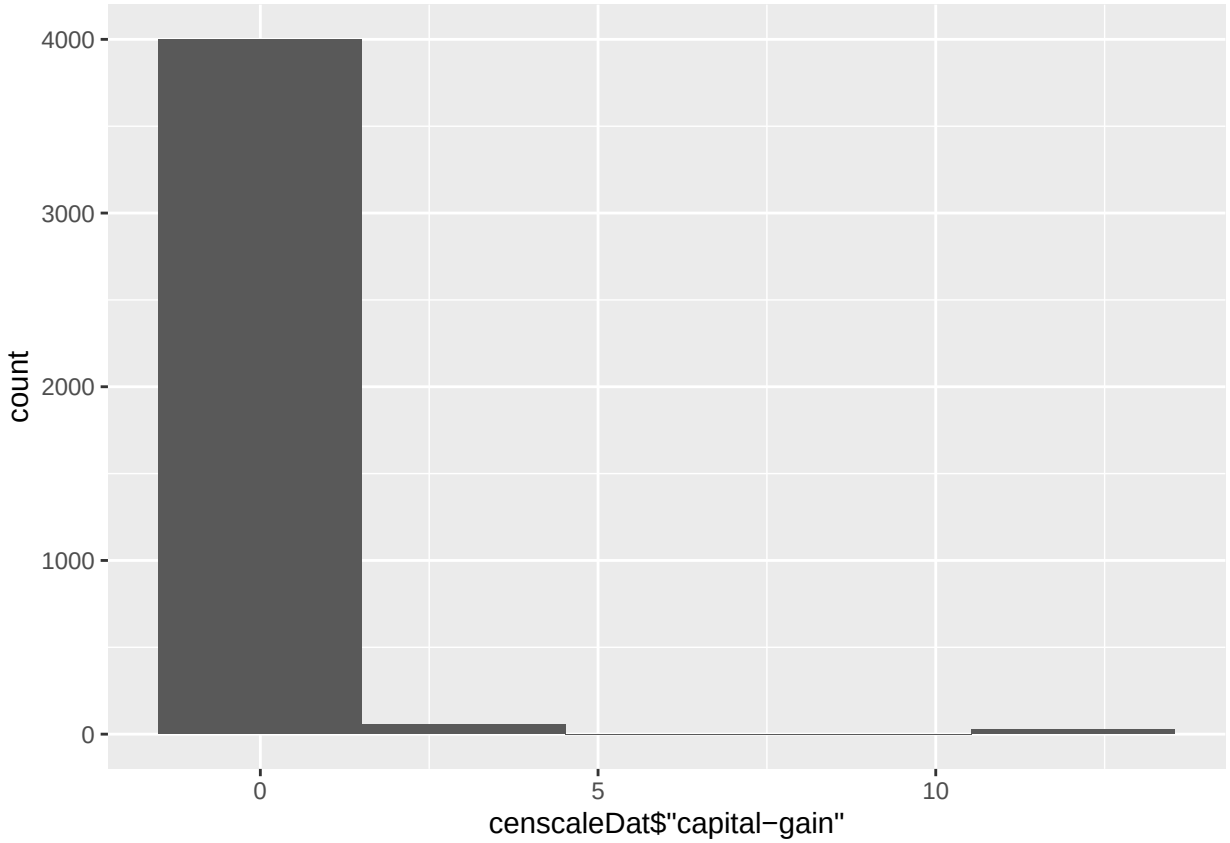
```
#rmv never worked and without pay from workclass
if (dim(censusDat[which(censusDat$workclass == " Without-pay"),])[1]!=0)
{ censusDat = censusDat[-which(censusDat$workclass == " Without-pay"),] }
if (dim(censusDat[which(censusDat$workclass == " Never-worked"),])[1]!=0)
{ censusDat[-which(censusDat$workclass == " Never-worked"),] }

#rmv armed forces from occupation
if (dim(censusDat[which(censusDat$occupation == " Armed-Forces"),])[1]!=0)
{ censusDat = censusDat[-which(censusDat$occupation == " Armed-Forces"),] }

#For Marital Status, combine married-civ-spouse and married-af-spouse as together. Divorced, separated,
```

```
levels(censusDat$`marital-status`)[6]="Separated"
levels(censusDat$`marital-status`)[4]="Separated"
levels(censusDat$`marital-status`)[3]="Together"
levels(censusDat$`marital-status`)[2]="Together"
levels(censusDat$`marital-status`)[1]="Separated"

#Group United States and not United States from Native Country
for (indx in (length(levels(censusDat$`native-country`)):1)) {
  if (levels(censusDat$`native-country`)[indx]==" United-States" || levels(censusDat$`native-country`)[
    levels(censusDat$`native-country`)[indx]="Domestic"
  } else {
    levels(censusDat$`native-country`)[indx]="Foreign"
  }
}
table(censusDat$`native-country`,censusDat$outcome)
```

```
##
##             <=50K  >50K
##   Foreign     299    70
##   Domestic   2777   938
```

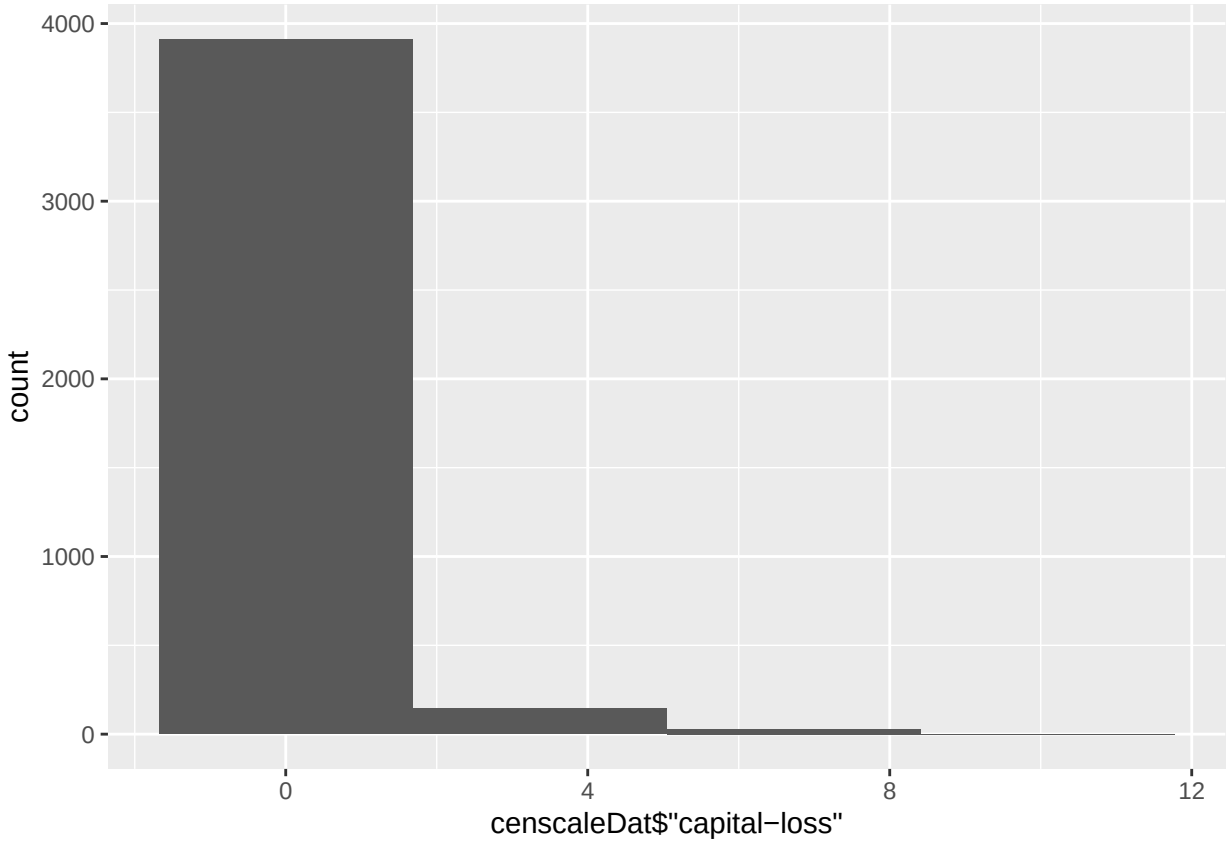
```
#scale all continuous vars
censcaleDat = censusDat
censcaleDat[,1]=scale(censusDat[,1])
censcaleDat[,4]=scale(censusDat[,4])
censcaleDat[,10]=scale(censusDat[,10])
censcaleDat[,11]=scale(censusDat[,11])
censcaleDat[,12]=scale(censusDat[,12])

#outliers in capital gains/Loss?
ggplot(censcaleDat,aes(censcaleDat$'capital-gain')) +geom_histogram(bins=5)
```

```
## Warning: Use of `censcaleDat$"capital-gain"` is discouraged. Use `capital-gain`
## instead.
```




```
ggplot(censcaleDat,aes(censcaleDat$'capital-loss')) +geom_histogram(bins=4)
```

```
## Warning: Use of `censcaleDat$"capital-loss"` is discouraged. Use `capital-loss`
## instead.
```



I removed any rows with missing values (coded "?") and removed the "never-worked" and "without-pay" categories from *workclass* and "Armed-Forces" from *occupation*. In *marital-status* I re-grouped "mareied-civ-spouse" and "married-af-spouse" as "Together" and "Divorced". I grouped "separated" and "married-spouse-absent" as "Separated". Finally, I grouped *native-country* as "Domestic" and "Foreign".

Also, I created a new data set, *censcaleDat*, where the continuous variables are normalized.

I considered removing outliers from *capital-gain* and *capital-loss* but I don't think this is appropriate because the high values should be very predictive of high income. Also, I decided not to discretize these attributes. I do think it's likely that higher values are more predictive although I expect this will be a small impact.

Finally, preschool in *education* has a very low number of datapoints. As a category it's small but it's meaningfull as an ordinal category. I plan on using *education-num* in place of *education* so I will not remove or re-group these.

Re-Summarize

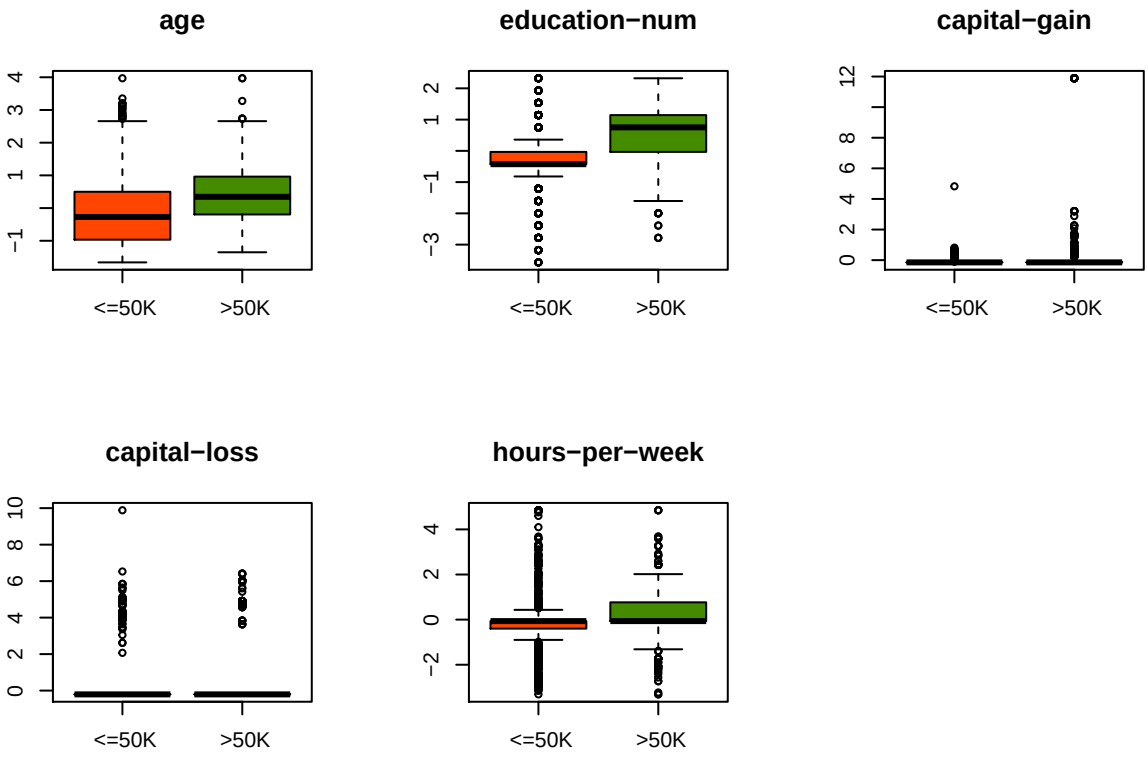
Now that some changes have been made I re-summarize the data in more detail.

```
summary(censcaleDat)
```

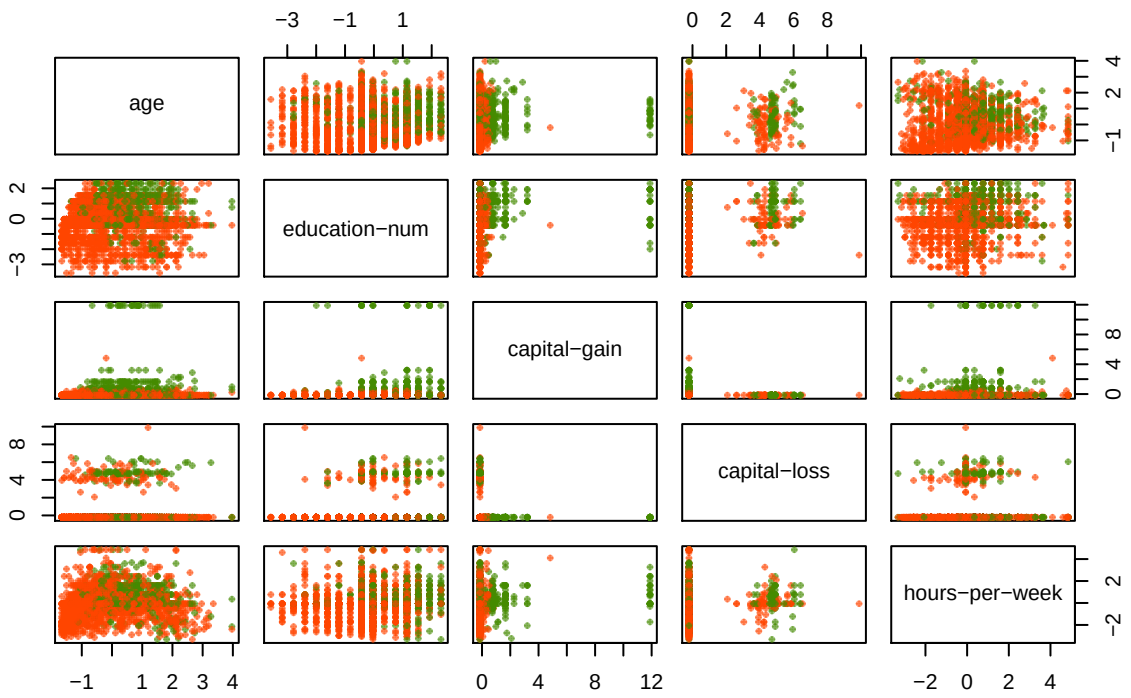
```
##        age.V1                    workclass            education
##  Min.   :-1.660335    Private         :3041    HS-grad     :1330
##  1st Qu.:-0.812001    Self-emp-not-inc: 343    Some-college: 921
##  Median :-0.117910    Local-gov       : 275    Bachelors   : 710
##  Mean   : 0.000000    State-gov       : 178    Masters     : 212
##  3rd Qu.: 0.653302    Self-emp-inc    : 132    Assoc-voc   : 160
##  Max.   : 3.969515    Federal-gov     : 115    11th        : 137
##                      (Other)          :   0   (Other)      : 614
```

```
##   education-num.V1          marital-status            occupation
##  Min.   :-3.567922   Separated     : 748    Craft-repair   : 554
##  1st Qu.:-0.426520   Together      :1921    Prof-specialty : 513
##  Median :-0.033845    Never-married:1301    Exec-managerial: 511
##  Mean   : 0.000000    Widowed      : 114    Sales          : 497
##  3rd Qu.: 1.144181                          Adm-clerical   : 491
##  Max.   : 2.322207                          Other-service  : 439
##                                            (Other)         :1079
##           relationship                    race           sex
##   Husband       :1702    Amer-Indian-Eskimo:  40    Female:1323
##   Not-in-family :1067    Asian-Pac-Islander: 110    Male  :2761
##   Other-relative: 123    Black             : 374
##   Own-child     : 582    Other             :  44
##   Unmarried     : 423    White             :3516
##   Wife          : 187
##
##    capital-gain.V1     capital-loss.V1    hours-per-week.V1   native-country
##  Min.   :-0.145581   Min.   :-0.207183   Min.   :-3.314871   Foreign : 369
##  1st Qu.:-0.145581   1st Qu.:-0.207183   1st Qu.:-0.066407   Domestic:3715
##  Median :-0.145581   Median :-0.207183   Median :-0.066407
##  Mean   : 0.000000   Mean   : 0.000000   Mean   : 0.000000
##  3rd Qu.:-0.145581   3rd Qu.:-0.207183   3rd Qu.: 0.350063
##  Max.   :11.886634   Max.   : 9.884089   Max.   : 4.847936
##
##    outcome
##   <=50K:3076
##   >50K :1008
##
##
##
##
##
```

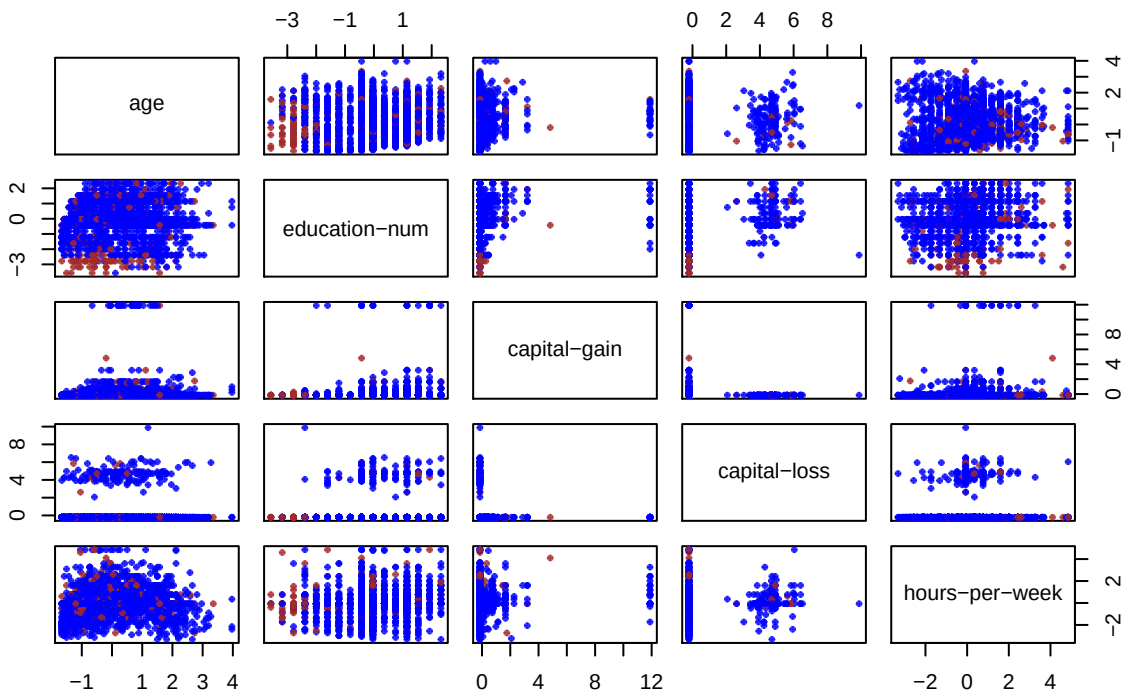
```
#boxplots for continuos variables
old.par=par(mfrow=c(2,3))
for ( clmns in c(1,4,10,11,12)) {
boxplot(censcaleDat[[clmns]]~censcaleDat$outcome,
main=names(censcaleDat)[clmns],col=c("orangered","chartreuse4"))
}
par(old.par)
```



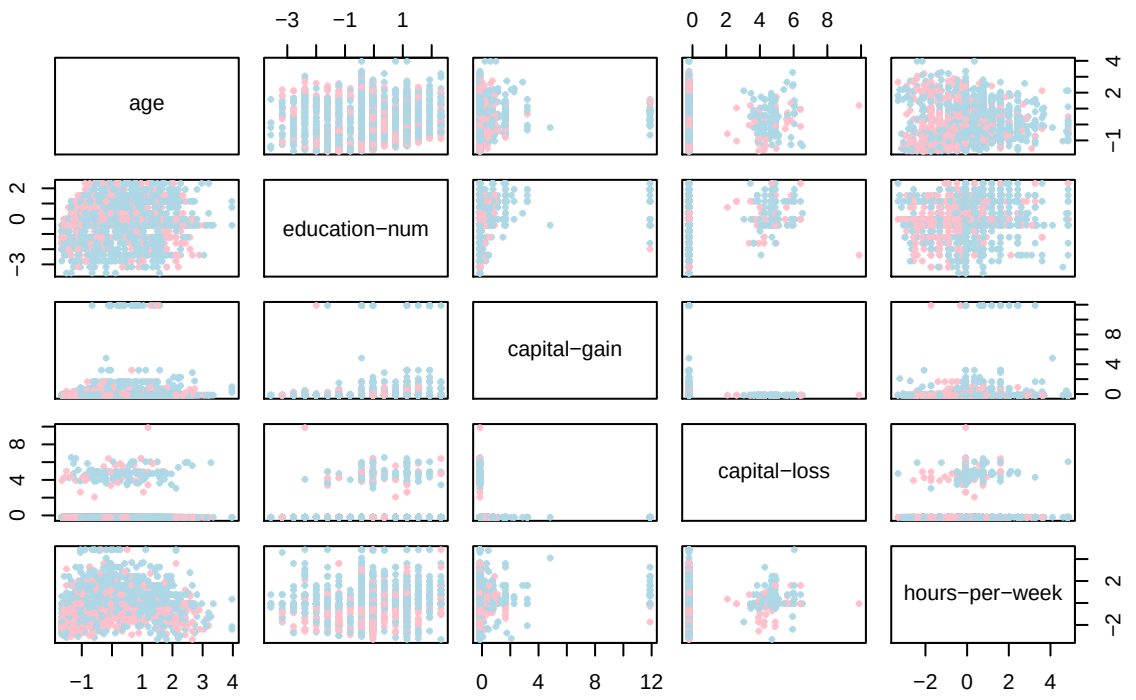

```
#pairs plot for continuous vars
pairs(censcaleDat[,c(1,4,10,11,12)],pch=10,cex=0.5,col=alpha(c("orangered","chartreuse4")[censcaleDat$ou
```



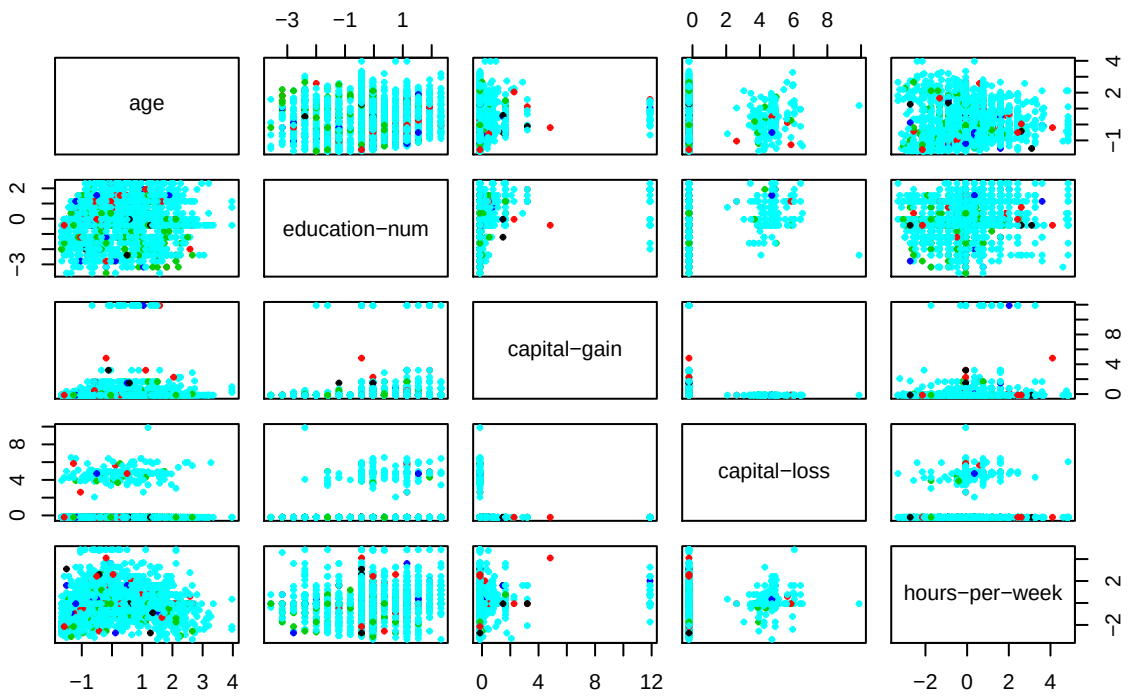

```
#country
pairs(censcaleDat[,c(1,4,10,11,12)],pch=10,cex=0.5,col=alpha(c("brown","blue")[censcaleDat$`native-coun
```



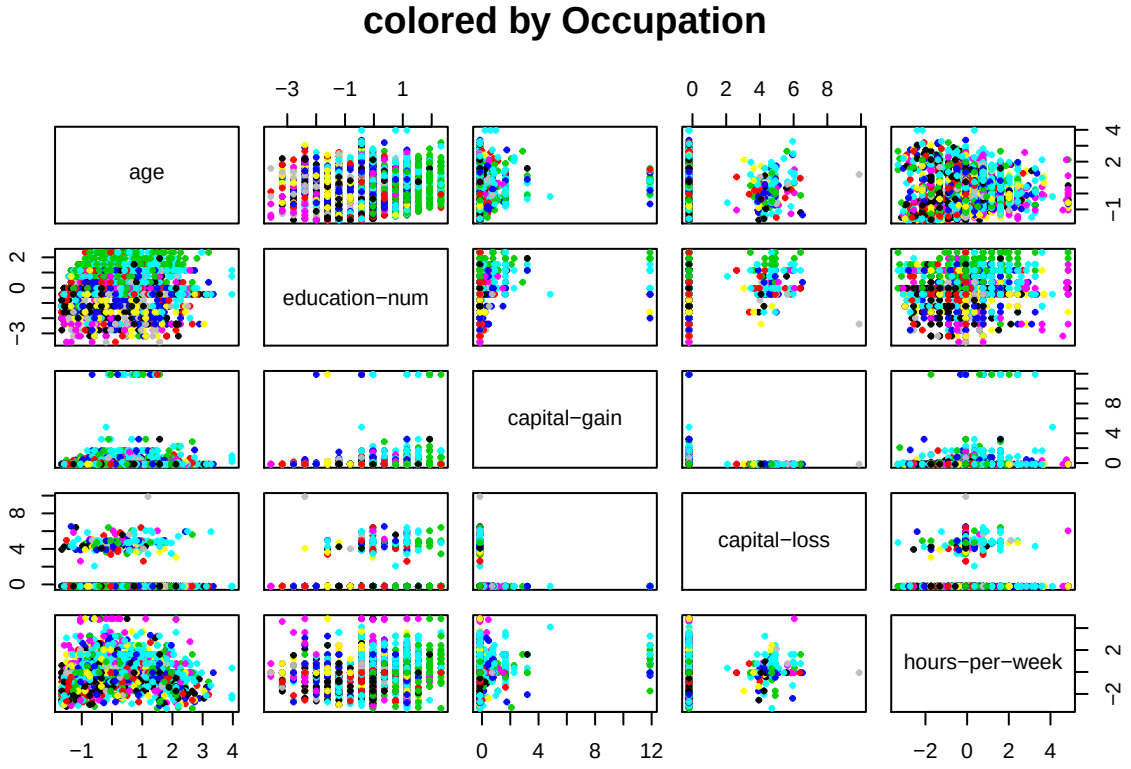

```
#sex
pairs(censcaleDat[,c(1,4,10,11,12)],pch=10,cex=0.5,col=alpha(c("pink","lightblue")[censcaleDat$sex],0.9
```



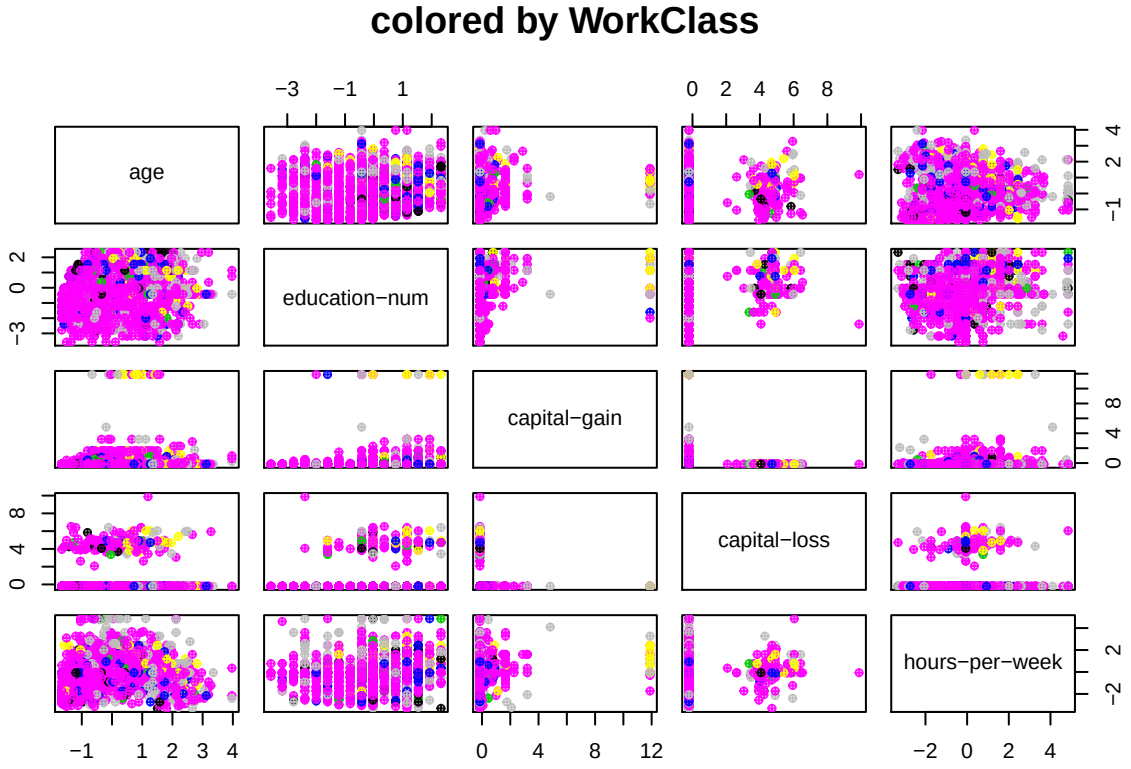

```
#race
pairs(censcaleDat[,c(1,4,10,11,12)],pch=10,cex=0.5,col=alpha(c(1:5)[censcaleDat$race],0.9),main="colore
```



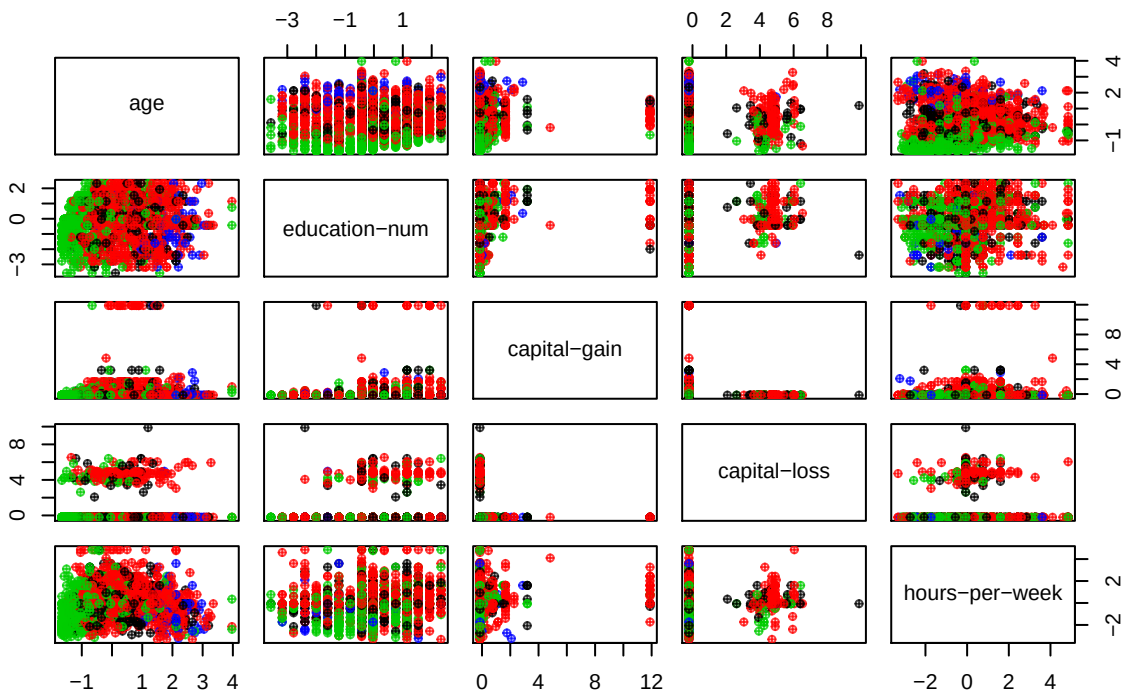

```
#occupation
pairs(censcaleDat[,c(1,4,10,11,12)],pch=10,cex=0.5,col=alpha(c(1:15)[censcaleDat$occupation],0.9),main=
```



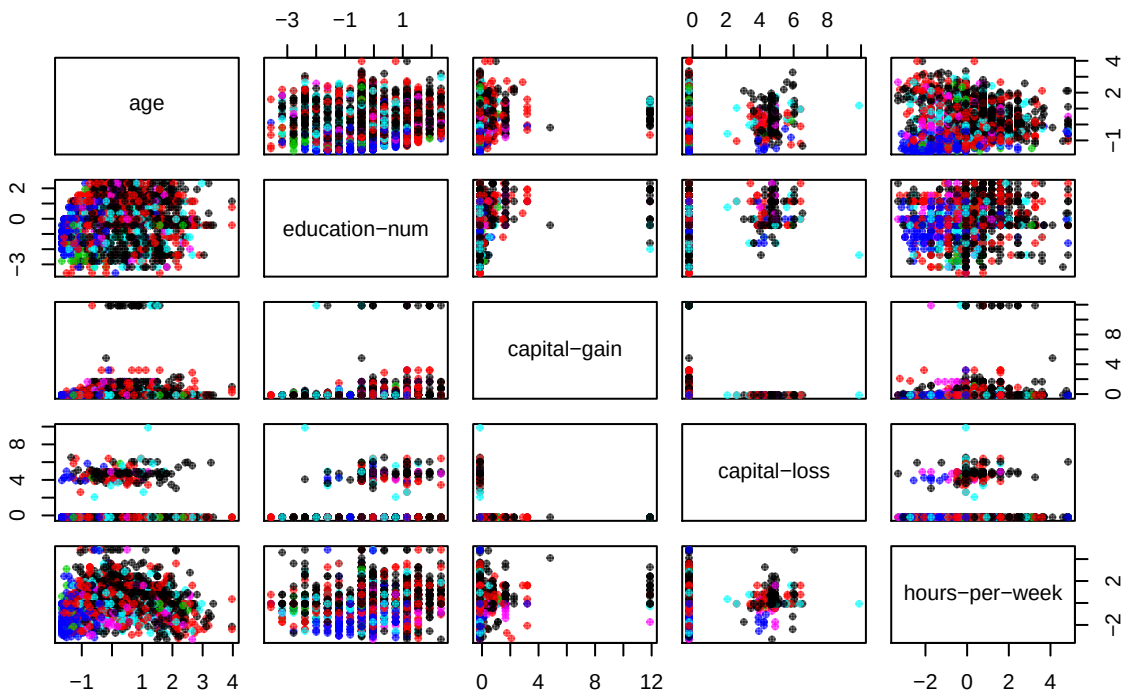

```
#workclass
pairs(censcaleDat[,c(1,4,10,11,12)],pch=10,cex=.75,col=alpha(c(2:10)[censcaleDat$workclass],0.9),main="
```




```
#Marital status
pairs(censcaleDat[,c(1,4,10,11,12)],pch=10,cex=.75,col=alpha(c(1:4)[censcaleDat$`marital-status`],0.75)
```

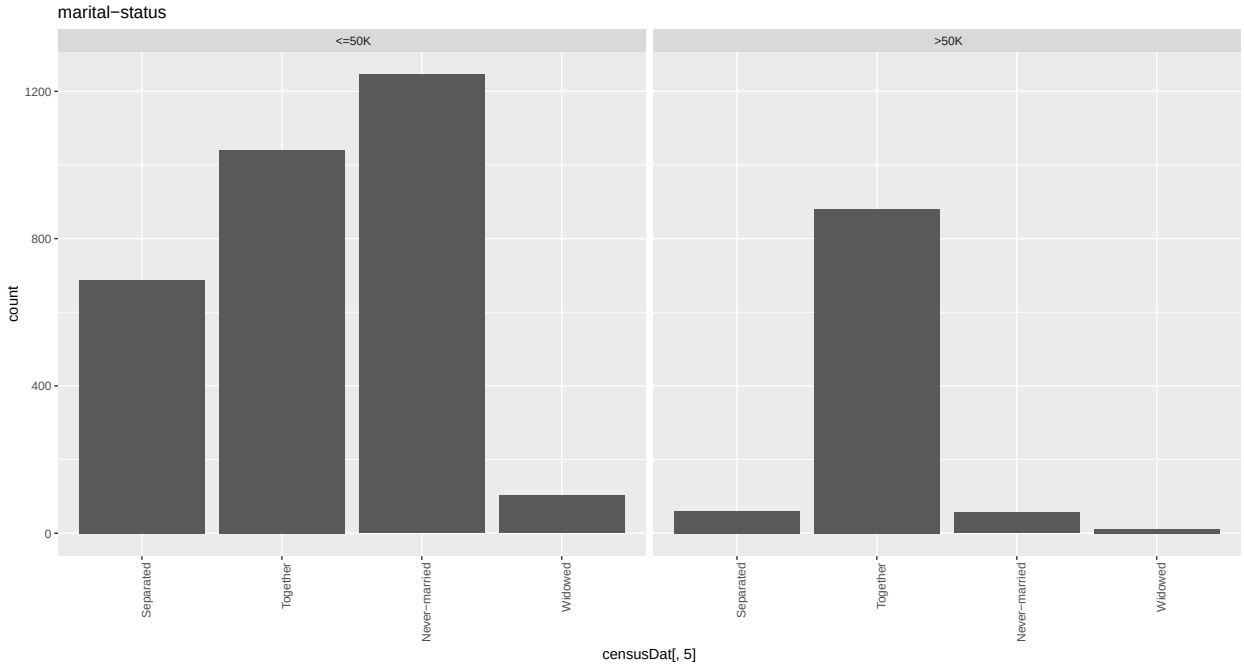



```
#relationship
pairs(censcaleDat[,c(1,4,10,11,12)],pch=10,cex=.6,col=alpha(c(1:6)[censcaleDat$relationship],0.6),main=
```

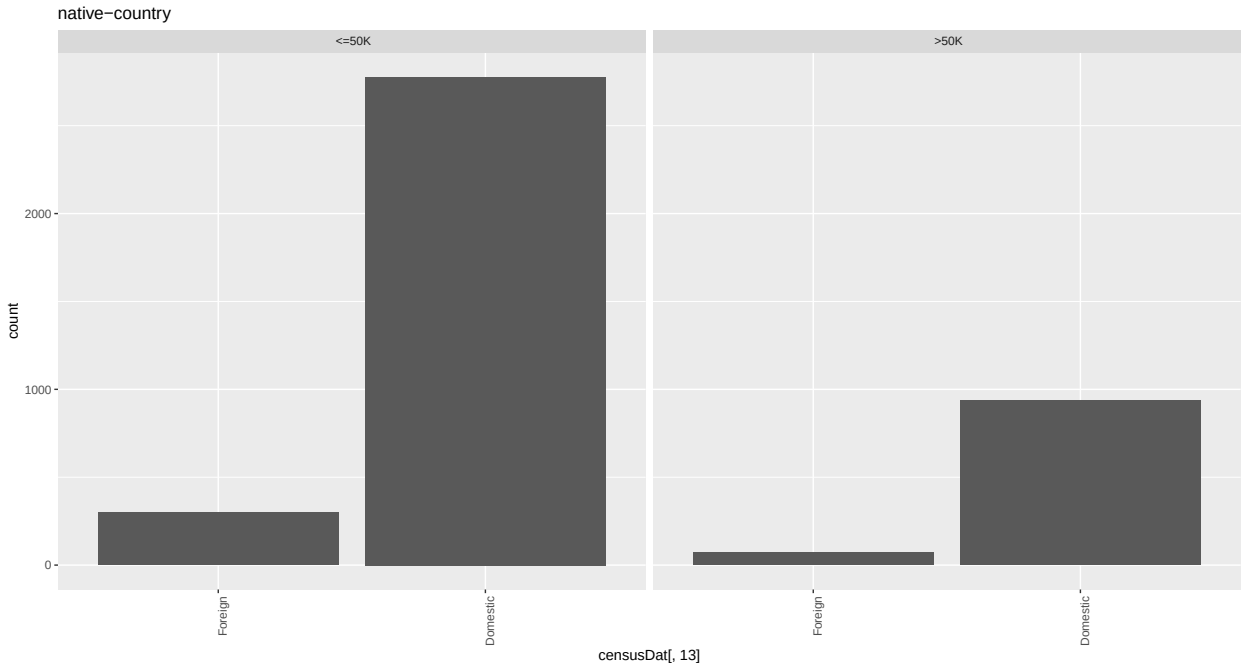



```
#redo table and bar chart for marital status and native country
CatList=list()
lst=1
for (clmns in c(5,13)) {
  CatList[[lst]]=table(censusDat[,clmns],censusDat$outcome)
  lst=lst+2
}

#barcharts of categorical variables split by outcome
CatList[[2]]=ggplot(censusDat,aes(censusDat[,5])) + geom_bar() + facet_wrap('outcome') + labs(title =nam
CatList[[4]]=ggplot(censusDat,aes(censusDat[,13])) + geom_bar() + facet_wrap('outcome') + labs(title =na
CatList
```

```
## [[1]]
##
##                     <=50K  >50K
##   Separated           688    60
##   Together           1041   880
##    Never-married     1245    56
##    Widowed            102    12
##
## [[2]]
```




```
##
## [[3]]
##
##              <=50K  >50K
##   Foreign      299    70
##   Domestic    2777   938
##
## [[4]]
```



After cleaning and scaling the data, I see, as expected, that *age education-num* and *hours-per-week* tend to have higher values for the higher income category. *education-num* has the clearest separation.

I also combined scatterplots of the continuous variables colored by each categorical variable. As I mentioned earlier, the biggest spread of data is seen in plots with *age*, *education* and *hours-per-week*. I also see the best separation of clusters of the outcome within these predictors. Possibly this is because *capital-gain* and *capital-loss* are so compact. Also, for these continuous variables, I see more cluster separation for *native-country*, *occupation*, *workclass* and *marital status*.

For *native-country* we see many more US datapoints than non-US. "Foreign" seems to concentrate more at lower and higher education levels than in the middle. For *sex*, women seem to be more likely to have higher education and work less hours per week. *occupation* seems to have a relationship with *education-num*, *capital-gain* and *hours-per-week*. *workclass* has discernable relatioships to *age* where the oldest people are less likely to be "Pvt". Also, non "Pvt" tends to complete more education and there seems to be some relationship to *capital-gains* and *hours-per-week*. *marital-status* has a clear relationship with *age*, as does *relationship*.

PCA Analysis

I visualize the scaled data using the top principal components to see along which attributes the most variation occurs and whether any useful structures emerge with these projections.

```
#confirm continuous variables are centered and scaled
signif(apply(censcaleDat[,c(1,4,10,11,12)],2,mean),2)
```

```
##            age  education-num   capital-gain   capital-loss hours-per-week
##       -2.3e-16        3.3e-16        6.2e-18       -1.5e-17       -1.8e-17
```

```
signif(apply(censcaleDat[,c(1,4,10,11,12)],2,var),2)
```

```
##            age  education-num   capital-gain   capital-loss hours-per-week
##              1              1              1              1              1
```

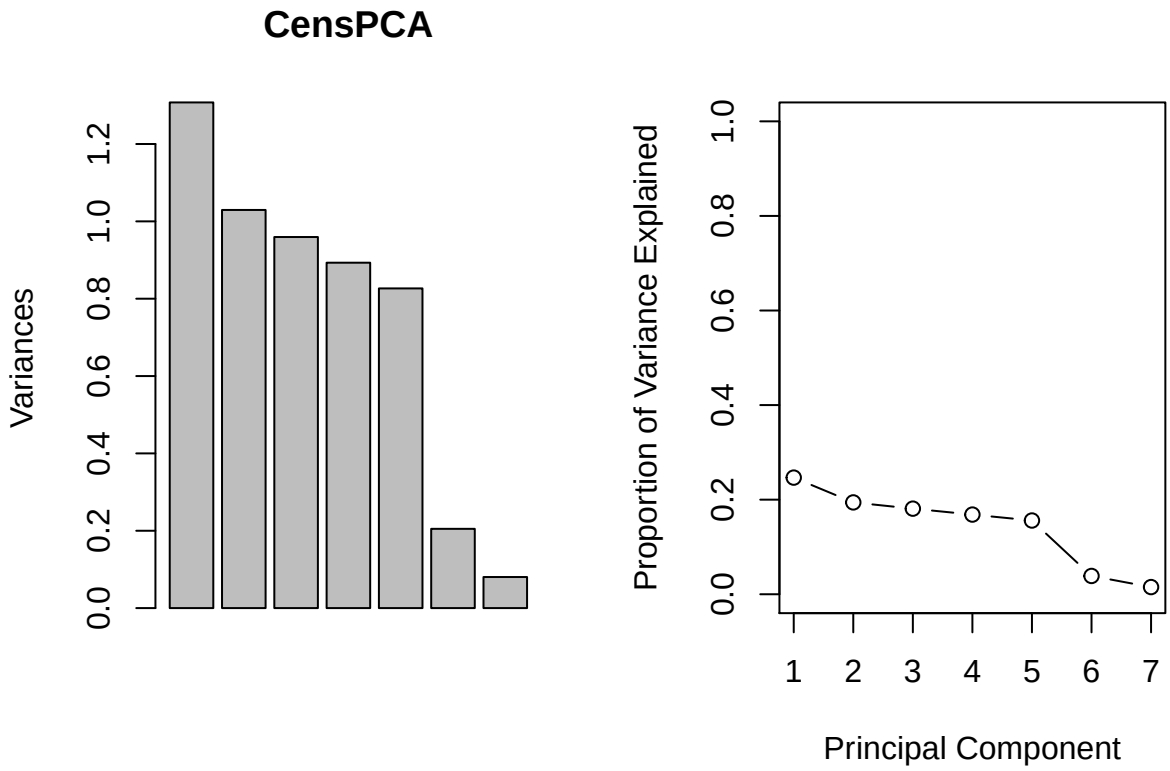
```
#check correlations
signif(cor(censcaleDat[,c(1,4,10,11,12)],method="pearson"),3) #linear
```

```
##                   age education-num capital-gain capital-loss hours-per-week
## age            1.0000        0.0414       0.0847       0.0668         0.1090
## education-num  0.0414        1.0000       0.1020       0.0693         0.1300
## capital-gain   0.0847        0.1020       1.0000      -0.0302         0.0873
## capital-loss   0.0668        0.0693      -0.0302       1.0000         0.0452
## hours-per-week 0.1090        0.1300       0.0873       0.0452         1.0000
```

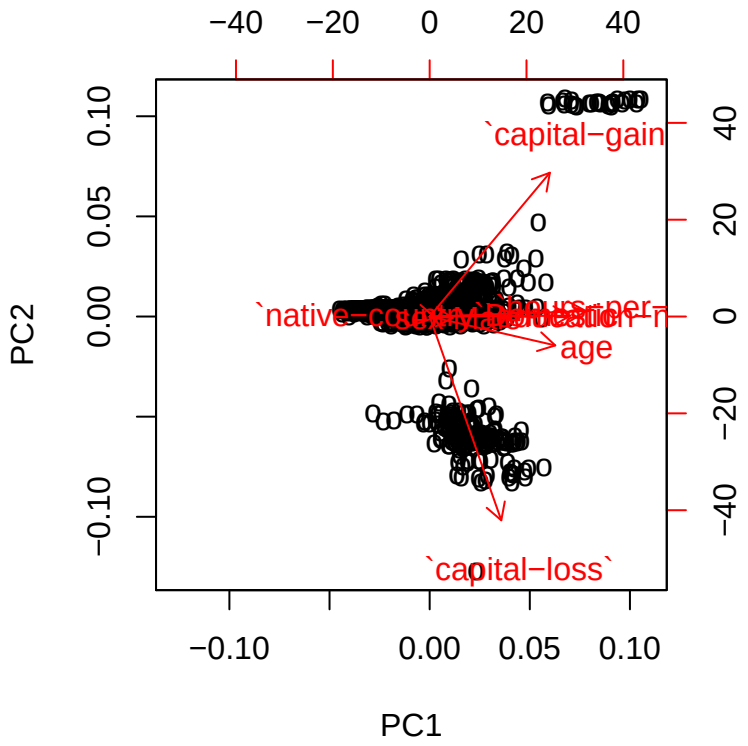
```
signif(cor(censcaleDat[,c(1,4,10,11,12)],method="spearman"),3) #monotonic
```

```
##                   age education-num capital-gain capital-loss hours-per-week
## age            1.0000        0.0647       0.1220       0.0679         0.1670
## education-num  0.0647        1.0000       0.1350       0.0649         0.1630
## capital-gain   0.1220        0.1350       1.0000      -0.0633         0.0897
## capital-loss   0.0679        0.0649      -0.0633       1.0000         0.0560
## hours-per-week 0.1670        0.1630       0.0897       0.0560         1.0000
```

```
censcaleDatpca=model.matrix(~age+`education-num`+sex+`capital-gain`+`capital-loss`+`hours-per-week`+`na

CensPCA=prcomp(censcaleDatpca)

old.par=par(mfrow=c(1,2))
plot(CensPCA)

Censpve=(CensPCA$sdev^2)/sum(CensPCA$sdev^2)
plot(Censpve,xlab="Principal Component",ylab="Proportion of Variance Explained",ylim = c(0,1),type='b')
```




```
par(old.par)


biplot(CensPCA,xlabs=rep("o",nrow(CensPCA$x)))
```




```
#biplot(CensPCA$x[,c(3,4)],CensPCA$rotation[,c(3,4)],xlabs=rep("o",nrow(CensPCA$x)))


CensPCA$rotation[,1:5]
```

```
##                                PC1          PC2         PC3         PC4
## age                     0.44387209 -0.115021080  0.75568946 -0.07620745
## `education-num`         0.49625287  0.003331456 -0.64720833 -0.11616716
## sex Male                0.08153381 -0.005372147  0.02138876  0.08071191
## `capital-gain`          0.42471303  0.572659316  0.04718440 -0.54800636
## `capital-loss`          0.25304194 -0.811358025 -0.07728212 -0.35711400
## `hours-per-week`        0.55254269  0.021733632 -0.02902709  0.73896677
## `native-country`Domestic 0.01865728 -0.004159853 -0.02357703 -0.01642983
##                                PC5
## age                     -0.46086305
## `education-num`         -0.56496362
## sex Male                 0.05664163
## `capital-gain`           0.43470926
## `capital-loss`           0.37932706
## `hours-per-week`         0.36246828
## `native-country`Domestic -0.03112359
```

```
sort(CensPCA$rotation[,1],decreasing=TRUE)
```

```
##         `hours-per-week`          `education-num`                      age
##               0.55254269               0.49625287               0.44387209
##           `capital-gain`           `capital-loss`                 sex Male
```

```
##               0.42471303               0.25304194               0.08153381
## `native-country`Domestic
##               0.01865728
```

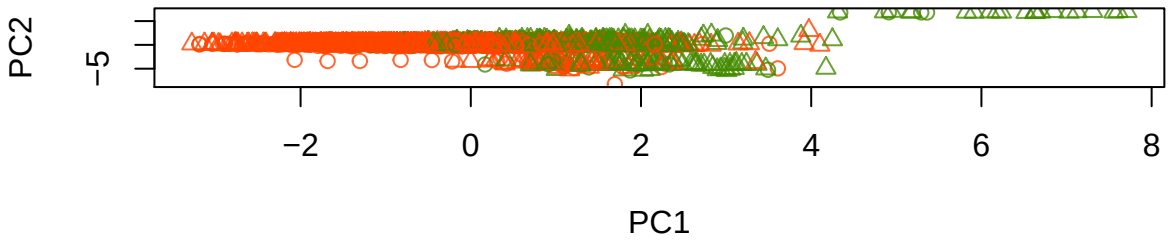
```
sort(CensPCA$rotation[,2],decreasing=TRUE)
```

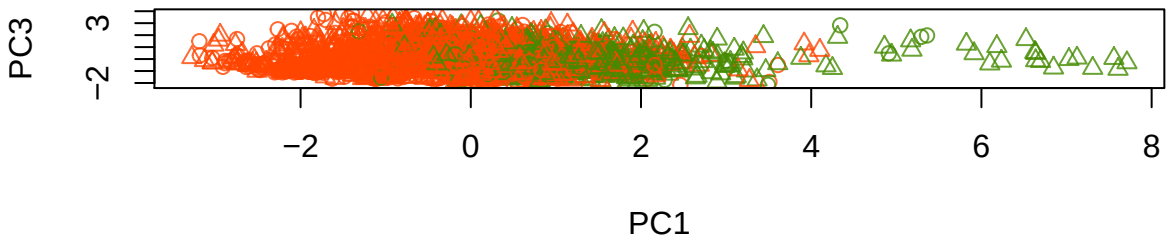
```
##           `capital-gain`         `hours-per-week`          `education-num`
##               0.572659316              0.021733632              0.003331456
## `native-country`Domestic                 sex Male                      age
##              -0.004159853             -0.005372147             -0.115021080
##           `capital-loss`
##              -0.811358025
```

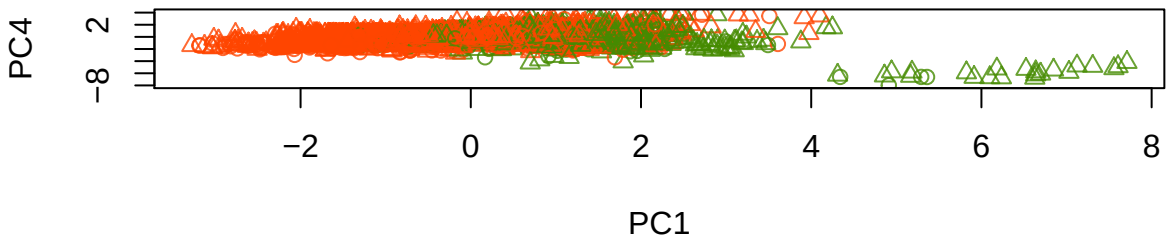
```
sort(CensPCA$rotation[,3],decreasing=TRUE)
```

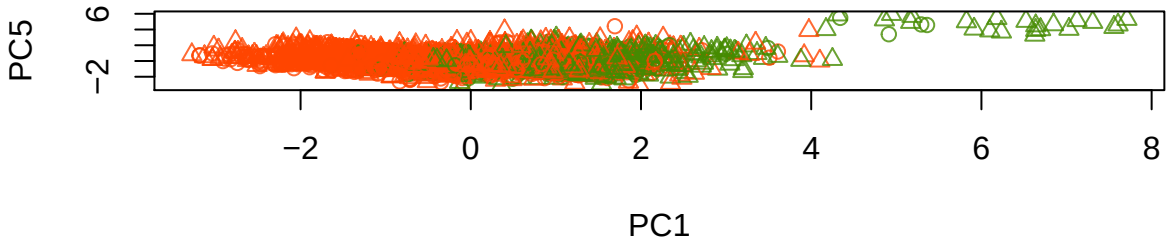
```
##                       age           `capital-gain`                 sex Male
##                0.75568946               0.04718440               0.02138876
## `native-country`Domestic         `hours-per-week`           `capital-loss`
##               -0.02357703              -0.02902709              -0.07728212
##          `education-num`
##               -0.64720833
```

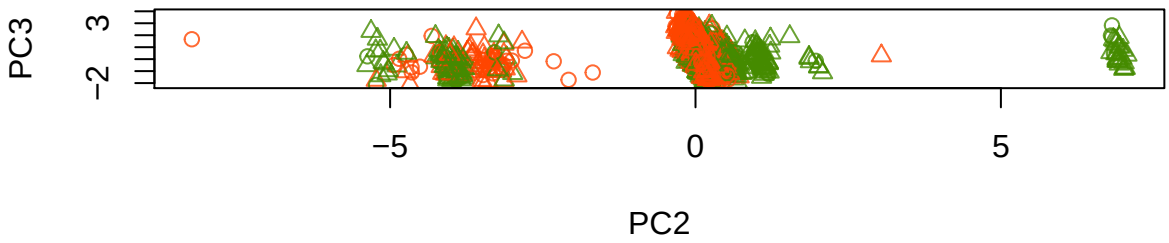
```
old.par=par(mfrow=c(2,1))
for (PCn in 1:4) {
  for (PCm in ((PCn+1):5)) {
plot(CensPCA$x[,c(PCn,PCm)],col=alpha(c("orangered","chartreuse4")[censcaleDat$outcome],.8),pch=c(1,2)[
  }}
```

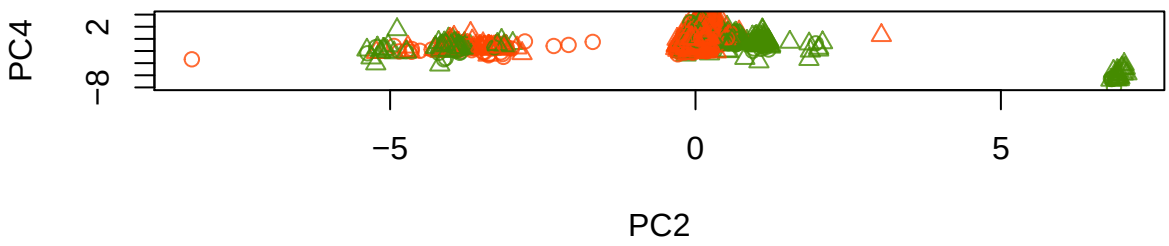


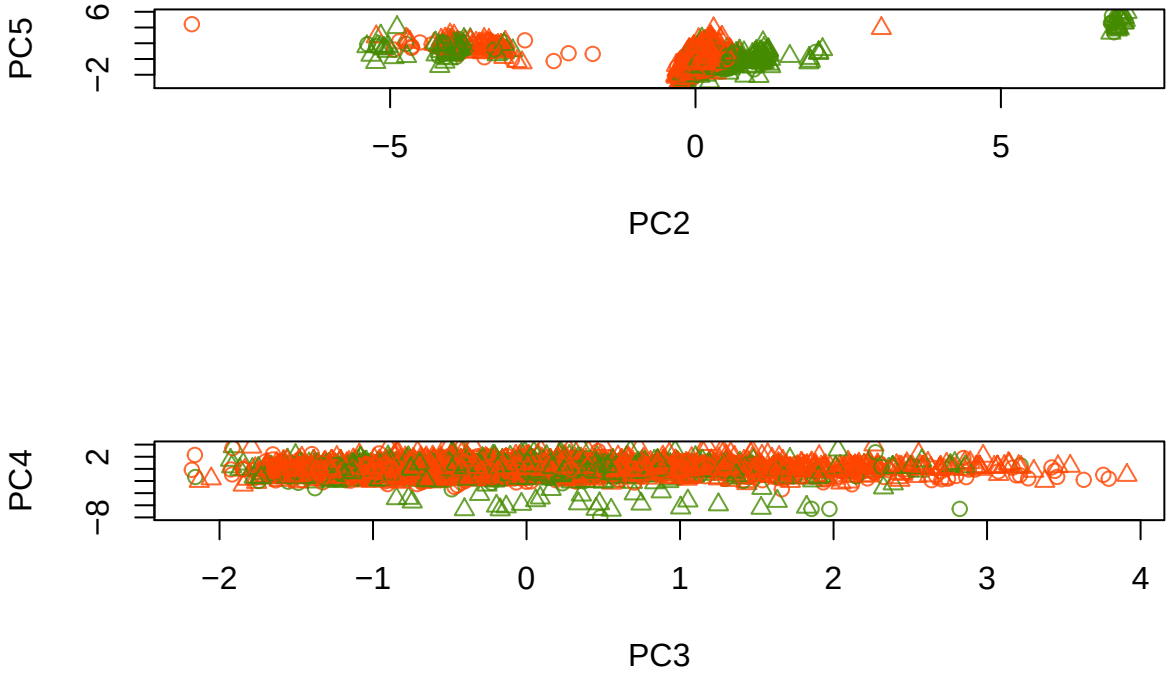


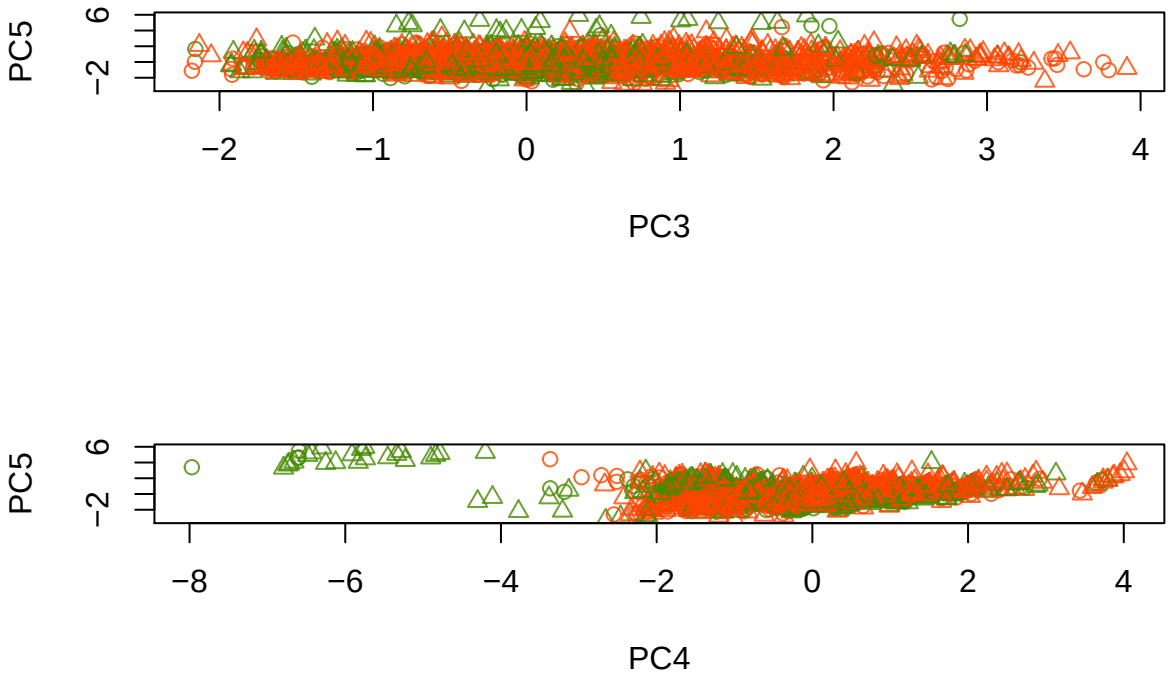

PC4
2
−8
−2
0
2
4
6
8
PC1


PC5
6
−2
−2
0
2
4
6
8
PC1


PC3
3
−2
−5
0
5
PC2


PC4
2
−8
−5
0
5
PC2


PC5
6
−2
−5
0
5
PC2
PC4
2
−8
−2
−1
0
1
2
3
4
PC3




```
par(old.par)
```

I checked the correlations between continuous predictors and see that all are positively correlated with eachother except *capital-gain* and *capital-loss*. None of the correlations are very strong. The highest pearson result is *education-num* with *hours-per-week* at about .153. Other higher coorelations are *hours-per-week* with *age*, and *education-num* with *capital-gain*. Spearman is probably more appropriate here as it's unlikely these relationships are linear. *education-num* and *hours-per-week* are still the most correlated at .1770. *age* and *capital-gain* have higher correlation than with Pearson, and *hours-per-week* and *capital-gain* also show strong correlation under Spearman.

I performed a principal component analysis, including all the continuous predictors and binary discrete predictors. I do not include categorical variables with more than two levels - I could create dummies but I think this would complicate the results. The PCA was performed on the normalized continous variables so that variance wouldn't be a function of the units of measurement. Notice that the categorical variables I did include, *sex* and *native-country*, do not have a large impact. This is partly related to the attributes having lower variance. I could have scaled the categorical variables as well but judging from the preliminary analysis I doubt that this will have much of an impact. The first 5 principal components contain the majority of the variance, PC1 explains around 25%. The separation present with PC2 is interesting. Looking at the biplot, it's driven by *capital-loss* and *capital-gain* where *education-num* and *hours-per-week* drives PC1 more. PC3 is driven primarily by *age*, PC4 by *hours-per-week* and PC5 by *education-num*

I graphed the top 5 principal components against eachother and colored the points by outcome. I see some separation in PC1, PC2, PC4 and PC5. It is interesting that the plots with PC4 and PC5 look very similar but these components are not that similar based on rotations.

As a result of this analysis, I think it was good not to discretize *capital-gain* and *capital-loss* and I think they will be strong predictors. *education-num* and *hours-per-week* will also be useful.

Univariate Assessment

Before creating multivariate models, I would like to check each attribute on its own. I create logistic models to see each attirbute's impact and significance.

$

```
censcaleDatdum=model.matrix(~age+workclass+`education-num`+`marital-status`+occupation+relationship+sex
censcaleDatdum=data.frame(censcaleDatdum)

#continuous
#age
summary(glm(outcome..50K~age,data=censcaleDatdum,family=binomial))
```

```
##
## Call:
## glm(formula = outcome..50K ~ age, family = binomial, data = censcaleDatdum)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.6245  -0.7780  -0.6078  -0.4790   2.0353
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept) -1.18832    0.03854  -30.83   <2e-16 ***
## age          0.55345    0.03740   14.80   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 4564.4  on 4083  degrees of freedom
## Residual deviance: 4333.5  on 4082  degrees of freedom
## AIC: 4337.5
##
## Number of Fisher Scoring iterations: 4
```

```
#education-num
summary(glm(outcome..50K~X.education.num.,data=censcaleDatdum,family=binomial))
```

```
##
## Call:
## glm(formula = outcome..50K ~ X.education.num., family = binomial,
##     data = censcaleDatdum)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.5445  -0.6836  -0.5804  -0.1706   2.7867
##
## Coefficients:
##                  Estimate Std. Error z value Pr(>|z|)
## (Intercept)      -1.30381    0.04211  -30.96   <2e-16 ***
## X.education.num.  0.91935    0.04527   20.31   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
```

```
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 4564.4  on 4083  degrees of freedom
## Residual deviance: 4069.5  on 4082  degrees of freedom
## AIC: 4073.5
##
## Number of Fisher Scoring iterations: 4
```

```
#capital-gain
summary(glm(outcome..50K~X.capital.gain.,data=censcaleDatdum,family=binomial))
```

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

```
##
## Call:
## glm(formula = outcome..50K ~ X.capital.gain., family = binomial,
##     data = censcaleDatdum)
##
## Deviance Residuals:
##    Min      1Q  Median      3Q     Max
## -4.916  -0.683  -0.683  -0.683   1.772
##
## Coefficients:
##                 Estimate Std. Error z value Pr(>|z|)
## (Intercept)     -0.94365    0.04299  -21.95   <2e-16 ***
## X.capital.gain.  2.70003    0.19990   13.51   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 4564.4  on 4083  degrees of freedom
## Residual deviance: 4154.9  on 4082  degrees of freedom
## AIC: 4158.9
##
## Number of Fisher Scoring iterations: 6
```

```
#capital-loss
summary(glm(outcome..50K~X.capital.loss.,data=censcaleDatdum,family=binomial))
```

```
##
## Call:
## glm(formula = outcome..50K ~ X.capital.loss., family = binomial,
##     data = censcaleDatdum)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.9097  -0.7295  -0.7295  -0.7295   1.7054
##
## Coefficients:
##                 Estimate Std. Error z value Pr(>|z|)
## (Intercept)     -1.12986    0.03677 -30.731   <2e-16 ***
## X.capital.loss.  0.28097    0.03209   8.756   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
```

```
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 4564.4  on 4083  degrees of freedom
## Residual deviance: 4488.2  on 4082  degrees of freedom
## AIC: 4492.2
##
## Number of Fisher Scoring iterations: 4
```

```
#hours-per-week
summary(glm(outcome..50K~X.hours.per.week.,data=censcaleDatdum,family=binomial))
```

```
##
## Call:
## glm(formula = outcome..50K ~ X.hours.per.week., family = binomial,
##     data = censcaleDatdum)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.7412  -0.7243  -0.7243  -0.3733   2.4064
##
## Coefficients:
##                   Estimate Std. Error z value Pr(>|z|)
## (Intercept)       -1.17095    0.03794  -30.87   <2e-16 ***
## X.hours.per.week.  0.50306    0.03934   12.79   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 4564.4  on 4083  degrees of freedom
## Residual deviance: 4382.9  on 4082  degrees of freedom
## AIC: 4386.9
##
## Number of Fisher Scoring iterations: 4
```

```
#categorical
#workclass
Log.workclass=glm(outcome..50K~workclass.Local.gov+workclass.Private+workclass.Self.emp.not.inc+workclas
  summary(Log.workclass)
```

```
##
## Call:
## glm(formula = outcome..50K ~ workclass.Local.gov + workclass.Private +
##     workclass.Self.emp.not.inc + workclass.Self.emp.inc + workclass.State.gov,
##     family = binomial, data = censcaleDatdum)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.3232  -0.7186  -0.7186  -0.7186   1.7207
##
## Coefficients:
##                             Estimate Std. Error z value Pr(>|z|)
## (Intercept)                  -0.6672     0.1970  -3.387 0.000706 ***
## workclass.Local.gov          -0.5258     0.2432  -2.162 0.030641 *
## workclass.Private            -0.5550     0.2017  -2.752 0.005923 **
## workclass.Self.emp.not.inc   -0.3664     0.2321  -1.579 0.114387
```

```
## workclass.Self.emp.inc       1.0036     0.2645   3.794 0.000148 ***
## workclass.State.gov         -0.3870     0.2610  -1.483 0.138129
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 4564.4  on 4083  degrees of freedom
## Residual deviance: 4485.1  on 4078  degrees of freedom
## AIC: 4497.1
##
## Number of Fisher Scoring iterations: 4
```

```
  #####Fisher Exact
#predict Prob
Log.workclass.probs=predict(Log.workclass,type="response")
#set assignment threshold of 0.5
Log.workclass.pred=rep(0,length(Log.workclass.probs))
Log.workclass.pred[Log.workclass.probs>.5]=1
Log.workclass.pred=factor(Log.workclass.pred)
#confusion table. rows are pred, cols actual
Log.workclass.table=table(pred=Log.workclass.pred,resp=censcaleDatdum$outcome..50K)
fisher.test(Log.workclass.table)
```

```
##
##  Fisher's Exact Test for Count Data
##
## data:  Log.workclass.table
## p-value < 2.2e-16
## alternative hypothesis: true odds ratio is not equal to 1
## 95 percent confidence interval:
##  3.144423 6.593799
## sample estimates:
## odds ratio
##   4.541075
```

```
#marital-status
Log.marital.status=glm(outcome..50K~X.marital.status.Together+X.marital.status..Never.married+X.marital
summary(Log.marital.status)
```

```
##
## Call:
## glm(formula = outcome..50K ~ X.marital.status.Together + X.marital.status..Never.married +
##     X.marital.status..Widowed, family = binomial, data = censcaleDatdum)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.1070  -1.1070  -0.2966  -0.2966   2.5082
##
## Coefficients:
##                                Estimate Std. Error z value Pr(>|z|)
## (Intercept)                     -2.4394     0.1346 -18.122  < 2e-16 ***
## X.marital.status.Together        2.2714     0.1422  15.975  < 2e-16 ***
## X.marital.status..Never.married -0.6621     0.1918  -3.453 0.000555 ***
## X.marital.status..Widowed        0.2994     0.3336   0.898 0.369429
## ---
```

```
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 4564.4  on 4083  degrees of freedom
## Residual deviance: 3606.0  on 4080  degrees of freedom
## AIC: 3614
##
## Number of Fisher Scoring iterations: 5
```

```
  #####Fisher Exact
#predict Prob
Log.marital.status.probs=predict(Log.marital.status,type="response")
#set assignment threshold of 0.5
Log.marital.status.pred=rep(0,length(Log.marital.status.probs))
Log.marital.status.pred[Log.marital.status.probs>.5]=1
Log.marital.status.pred=factor(Log.marital.status.pred)
levels(Log.marital.status.pred)[2]="1" #on some runs, never get over p=0.5
#confusion table. rows are pred, cols actual
Log.marital.status.table=table(pred=Log.marital.status.pred,resp=censcaleDatdum$outcome..50K)
fisher.test(Log.marital.status.table)
```

```
##
##  Fisher's Exact Test for Count Data
##
## data:  Log.marital.status.table
## p-value = 1
## alternative hypothesis: true odds ratio is not equal to 1
## 95 percent confidence interval:
##    0 Inf
## sample estimates:
## odds ratio
##          0
```

```
#occupation
Log.occupation=glm(outcome..50K~occupation.Adm.clerical + occupation.Craft.repair + occupation.Exec.mana
summary(Log.occupation)
```

```
##
## Call:
## glm(formula = outcome..50K ~ occupation.Adm.clerical + occupation.Craft.repair +
##     occupation.Exec.managerial + occupation.Farming.fishing +
##     occupation.Handlers.cleaners + occupation.Machine.op.inspct +
##     occupation.Other.service + occupation.Priv.house.serv + occupation.Prof.specialty +
##     occupation.Protective.serv + occupation.Sales + occupation.Tech.support +
##     occupation.Transport.moving, family = binomial, data = censcaleDatdum)
##
## Deviance Residuals:
##      Min        1Q    Median        3Q       Max
## -1.19910  -0.78225  -0.45360  -0.00097   2.65313
##
## Coefficients: (1 not defined because of singularities)
##                              Estimate Std. Error z value Pr(>|z|)
## (Intercept)                   -1.6463     0.1931  -8.527  < 2e-16 ***
## occupation.Adm.clerical       -0.5761     0.2457  -2.345 0.019034 *
## occupation.Craft.repair        0.5905     0.2161   2.733 0.006285 **
```

```
## occupation.Exec.managerial     1.6971     0.2124   7.991 1.34e-15 ***
## occupation.Farming.fishing    -0.6330     0.3494  -1.812 0.070032 .
## occupation.Handlers.cleaners  -0.7364     0.3130  -2.353 0.018634 *
## occupation.Machine.op.inspct  -0.1968     0.2603  -0.756 0.449618
## occupation.Other.service      -1.8432     0.3414  -5.399 6.69e-08 ***
## occupation.Priv.house.serv   -12.9198   235.9232  -0.055 0.956327
## occupation.Prof.specialty      1.4861     0.2124   6.996 2.63e-12 ***
## occupation.Protective.serv     0.8279     0.2921   2.835 0.004588 **
## occupation.Sales               0.6188     0.2183   2.835 0.004580 **
## occupation.Tech.support        0.8871     0.2658   3.338 0.000844 ***
## occupation.Transport.moving        NA         NA      NA       NA
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 4564.4  on 4083  degrees of freedom
## Residual deviance: 3953.9  on 4071  degrees of freedom
## AIC: 3979.9
##
## Number of Fisher Scoring iterations: 13
```

```
#####Fisher Exact
#predict Prob
Log.occupation.probs=predict(Log.occupation,type="response")
#set assignment threshold of 0.5
Log.occupation.pred=rep(0,length(Log.occupation.probs))
Log.occupation.pred[Log.occupation.probs>.5]=1
Log.occupation.pred=factor(Log.occupation.pred)
levels(Log.occupation.pred)[2]="1" #on some runs, never get over p=0.5
#confusion table. rows are pred, cols actual
Log.occupation.table=table(pred=Log.occupation.pred,resp=censcaleDatdum$outcome..50K)
fisher.test(Log.occupation.table)
```

```
##
##  Fisher's Exact Test for Count Data
##
## data:  Log.occupation.table
## p-value < 2.2e-16
## alternative hypothesis: true odds ratio is not equal to 1
## 95 percent confidence interval:
##  3.276802 4.849076
## sample estimates:
## odds ratio
##   3.985705
```

```
#relationship
Log.relationship=glm(outcome..50K~relationship.Not.in.family + relationship.Other.relative + relationshi
summary(Log.relationship)
```

```
##
## Call:
## glm(formula = outcome..50K ~ relationship.Not.in.family + relationship.Other.relative +
##     relationship.Own.child + relationship.Unmarried + relationship.Wife,
##     family = binomial, data = censcaleDatdum)
##
```

```
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.1458  -0.4390  -0.3268  -0.1766   2.8876
##
## Coefficients:
##                             Estimate Std. Error z value Pr(>|z|)
## (Intercept)                 -0.15542    0.04863  -3.196  0.00139 **
## relationship.Not.in.family  -2.13587    0.11662 -18.315  < 2e-16 ***
## relationship.Other.relative -3.23741    0.51065  -6.340  2.3e-10 ***
## relationship.Own.child      -3.99824    0.33933 -11.783  < 2e-16 ***
## relationship.Unmarried      -2.74750    0.22430 -12.249  < 2e-16 ***
## relationship.Wife            0.08052    0.15422   0.522  0.60159
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 4564.4  on 4083  degrees of freedom
## Residual deviance: 3564.0  on 4078  degrees of freedom
## AIC: 3576
##
## Number of Fisher Scoring iterations: 6
```

```
#####Fisher Exact
#predict Prob
Log.relationship.probs=predict(Log.relationship,type="response")
#set assignment threshold of 0.5
Log.relationship.pred=rep(0,length(Log.relationship.probs))
Log.relationship.pred[Log.relationship.probs>.5]=1
Log.relationship.pred=factor(Log.relationship.pred)
levels(Log.relationship.pred)[2]="1" #on some runs, never get over p=0.5
#confusion table. rows are pred, cols actual
Log.relationship.table=table(pred=Log.relationship.pred,resp=censcaleDatdum$outcome..50K)
fisher.test(Log.relationship.table)
```

```
##
##  Fisher's Exact Test for Count Data
##
## data:  Log.relationship.table
## p-value = 1
## alternative hypothesis: true odds ratio is not equal to 1
## 95 percent confidence interval:
##    0 Inf
## sample estimates:
## odds ratio
##          0
```

```
#race
Log.race=glm(outcome..50K~race.Asian.Pac.Islander+race.Black+race.Other+race.White,data=censcaleDatdum,
summary(Log.race)
```

```
##
## Call:
## glm(formula = outcome..50K ~ race.Asian.Pac.Islander + race.Black +
##     race.Other + race.White, family = binomial, data = censcaleDatdum)
##
```

```
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -0.7824  -0.7824  -0.7824  -0.4693   2.1264
##
## Coefficients:
##                          Estimate Std. Error z value Pr(>|z|)
## (Intercept)               -1.3863     0.3953  -3.507 0.000453 ***
## race.Asian.Pac.Islander    0.3591     0.4506   0.797 0.425478
## race.Black                -0.7643     0.4300  -1.778 0.075484 .
## race.Other                -0.6678     0.6180  -1.081 0.279836
## race.White                 0.3592     0.3971   0.905 0.365712
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 4564.4  on 4083  degrees of freedom
## Residual deviance: 4504.6  on 4079  degrees of freedom
## AIC: 4514.6
##
## Number of Fisher Scoring iterations: 4
```

```
#####Fisher Exact
#predict Prob
Log.race.probs=predict(Log.race,type="response")
#set assignment threshold of 0.5
Log.race.pred=rep(0,length(Log.race.probs))
Log.race.pred[Log.race.probs>.5]=1
Log.race.pred=factor(Log.race.pred)
levels(Log.race.pred)[2]="1" #on some runs, never get over p=0.5
#confusion table. rows are pred, cols actual
Log.race.table=table(pred=Log.race.pred,resp=censcaleDatdum$outcome..50K)
fisher.test(Log.race.table)
```

```
##
##  Fisher's Exact Test for Count Data
##
## data:  Log.race.table
## p-value = 1
## alternative hypothesis: true odds ratio is not equal to 1
## 95 percent confidence interval:
##    0 Inf
## sample estimates:
## odds ratio
##          0
```

```
#sex
Log.sex=glm(outcome..50K~sex.Male,data=censcaleDatdum,family=binomial)
summary(Log.sex)
```

```
##
## Call:
## glm(formula = outcome..50K ~ sex.Male, family = binomial, data = censcaleDatdum)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
```

```
## -0.8688  -0.8688  -0.4730  -0.4730   2.1194
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept) -2.13417    0.08937  -23.88   <2e-16 ***
## sex.Male     1.35444    0.09832   13.78   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 4564.4  on 4083  degrees of freedom
## Residual deviance: 4331.3  on 4082  degrees of freedom
## AIC: 4335.3
##
## Number of Fisher Scoring iterations: 4
```

```
#####Fisher Exact
#predict Prob
Log.sex.probs=predict(Log.sex,type="response")
#set assignment threshold of 0.5
Log.sex.pred=rep(0,length(Log.sex.probs))
Log.sex.pred[Log.sex.probs>.5]=1
Log.sex.pred=factor(Log.sex.pred)
levels(Log.sex.pred)[2]="1" #on some runs, never get over p=0.5
#confusion table. rows are pred, cols actual
Log.sex.table=table(pred=Log.sex.pred,resp=censcaleDatdum$outcome..50K)
fisher.test(Log.sex.table)
```

```
##
##  Fisher's Exact Test for Count Data
##
## data:  Log.sex.table
## p-value = 1
## alternative hypothesis: true odds ratio is not equal to 1
## 95 percent confidence interval:
##    0 Inf
## sample estimates:
## odds ratio
##          0
```

```
#native-country
Log.native=glm(outcome..50K~X.native.country.Domestic,data=censcaleDatdum,family=binomial)
summary(Log.native)
```

```
##
## Call:
## glm(formula = outcome..50K ~ X.native.country.Domestic, family = binomial,
##     data = censcaleDatdum)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -0.7629  -0.7629  -0.7629  -0.6486   1.8234
##
## Coefficients:
##                           Estimate Std. Error z value Pr(>|z|)
```

```
## (Intercept)                -1.4519     0.1328 -10.935  < 2e-16 ***
## X.native.country.Domestic   0.3666     0.1380   2.655  0.00792 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 4564.4  on 4083  degrees of freedom
## Residual deviance: 4556.9  on 4082  degrees of freedom
## AIC: 4560.9
##
## Number of Fisher Scoring iterations: 4
```

```
#####Fisher Exact
#predict Prob
Log.native.probs=predict(Log.native,type="response")
#set assignment threshold of 0.5
Log.native.pred=rep(0,length(Log.native.probs))
Log.native.pred[Log.native.probs>.5]=1
Log.native.pred=factor(Log.native.pred)
levels(Log.native.pred)[2]="1" #on some runs, never get over p=0.5
#confusion table. rows are pred, cols actual
Log.native.table=table(pred=Log.native.pred,resp=censcaleDatdum$outcome..50K)
fisher.test(Log.native.table)
```

```
##
##  Fisher's Exact Test for Count Data
##
## data:  Log.native.table
## p-value = 1
## alternative hypothesis: true odds ratio is not equal to 1
## 95 percent confidence interval:
##    0 Inf
## sample estimates:
## odds ratio
##          0
```

$

I created a new data frame, *censcaleDatdum*, which includes dummy variables for the categorical data. I left out *education* but included *education-num*. For the continuous variables, all have small p values. *capital-gain* has the largest coefficient followed by *education-num*. *capital-loss* has the smallest.

For the categorical variables, "self employed" has a relatively large positive effect on income and "privately employed" a relatively large negative effect. marital status "together" also has a strong positive effect while "separated" and "never married" have negative effects. In *occupation*, "adm-clerical", "farming-fishing", "handlers-cleaners", "machine-op-inspct", and "other-service" all have a relatively large negative effect while "exec-managerial" and "Prof-specialty"" have high positive effects. For *marital-status*, "not-in-family", "other-relative", "own-child", and "unmarried" all have strong negative coefficients. "Amer-Indian-Eskimo" has a strong negative coefficient while "Asian-pac-Islander" and "white" have fairly strong positive coefficients. "Female" has a strong negative and "male" has a strong positive coeffienct. Finally, "Foreign" from *native-country* has a strong negative coefficient.

However, when making contingency tables using threshold of 0.5, all datapoints are sorted into the ">50k" category for all categorical attributes except *workclass*. the model built with *workclass* is significant according to the Fisher exact test, but the others are not. The prior probability is heavily weighted toward the under 50k category but it's possible that some of these attributes will be useful when combined with others.

Logistic regression

Now I develop a logistic regression model of household income as a function of multiple predictors and determine which variables are significantly associated with the outcome. I test model performance and compare it to the performance of other methods reported in the dataset description.

```
Logist.all=glm(as.factor(outcome..50K)~.,data=censcaleDatdum,family=binomial)
```

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

```
summary(Logist.all)
```

```
##
## Call:
## glm(formula = as.factor(outcome..50K) ~ ., family = binomial,
##     data = censcaleDatdum)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -5.0197  -0.5034  -0.1758  -0.0018   3.1368
##
## Coefficients: (5 not defined because of singularities)
##                                  Estimate Std. Error z value Pr(>|z|)
## (Intercept)                     -4.789136   1.042046  -4.596 4.31e-06 ***
## age                              0.370698   0.061301   6.047 1.47e-09 ***
## workclass.Federal.gov            0.708855   0.358687   1.976 0.048126 *
## workclass.Local.gov             -0.328084   0.302706  -1.084 0.278438
## workclass.Never.worked                NA         NA      NA       NA
## workclass.Private                0.154464   0.250739   0.616 0.537872
## workclass.Self.emp.inc           0.348597   0.342921   1.017 0.309367
## workclass.Self.emp.not.inc      -0.615604   0.294738  -2.089 0.036739 *
## workclass.State.gov                   NA         NA      NA       NA
## workclass.Without.pay                 NA         NA      NA       NA
## X.education.num.                 0.667006   0.066396  10.046  < 2e-16 ***
## X.marital.status.Together        2.734407   0.740508   3.693 0.000222 ***
## X.marital.status..Never.married -0.370972   0.242875  -1.527 0.126658
## X.marital.status..Widowed        0.088620   0.495023   0.179 0.857920
## occupation.Adm.clerical         -0.001096   0.300199  -0.004 0.997087
## occupation.Armed.Forces               NA         NA      NA       NA
## occupation.Craft.repair          0.462812   0.245475   1.885 0.059380 .
## occupation.Exec.managerial       1.258928   0.256846   4.901 9.51e-07 ***
## occupation.Farming.fishing      -0.566869   0.411244  -1.378 0.168072
## occupation.Handlers.cleaners    -0.038102   0.362341  -0.105 0.916252
## occupation.Machine.op.inspct    -0.007475   0.293206  -0.025 0.979661
## occupation.Other.service        -1.080967   0.385774  -2.802 0.005078 **
## occupation.Priv.house.serv     -12.229874 318.320385  -0.038 0.969353
## occupation.Prof.specialty        1.117449   0.272014   4.108 3.99e-05 ***
## occupation.Protective.serv       0.730045   0.366814   1.990 0.046565 *
## occupation.Sales                 0.343309   0.259936   1.321 0.186586
## occupation.Tech.support          1.072839   0.332877   3.223 0.001269 **
## occupation.Transport.moving           NA         NA      NA       NA
## relationship.Not.in.family       0.769242   0.730229   1.053 0.292146
## relationship.Other.relative      0.102480   0.653111   0.157 0.875315
## relationship.Own.child          -0.223528   0.702844  -0.318 0.750460
## relationship.Unmarried           0.631284   0.784087   0.805 0.420751
## relationship.Wife                1.057968   0.294759   3.589 0.000332 ***
```

```
## sex.Male                           0.699219   0.225930   3.095 0.001969 **
## race.Asian.Pac.Islander            0.333897   0.671437   0.497 0.618987
## race.Black                         0.117960   0.611968   0.193 0.847151
## race.Other                        -0.934713   0.924520  -1.011 0.312004
## race.White                         0.310456   0.577325   0.538 0.590750
## X.capital.gain.                    2.518080   0.251406  10.016  < 2e-16 ***
## X.capital.loss.                    0.244618   0.042698   5.729 1.01e-08 ***
## X.hours.per.week.                  0.301232   0.055438   5.434 5.52e-08 ***
## X.native.country.Domestic          0.213387   0.213026   1.002 0.316490
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 4564.4  on 4083  degrees of freedom
## Residual deviance: 2583.4  on 4047  degrees of freedom
## AIC: 2657.4
##
## Number of Fisher Scoring iterations: 14
```

```
model.atts=formula("as.factor(outcome..50K)~age+X.education.num.+X.capital.gain.+X.capital.loss.+X.hours
#excluding race and native-country as don't pass the z test in the model

Logist.some=glm(model.atts,data=censcaleDatdum,family=binomial) #.05 p value
```

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

```
summary(Logist.some)
```

```
##
## Call:
## glm(formula = model.atts, family = binomial, data = censcaleDatdum)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -5.0266  -0.5066  -0.1780  -0.0018   3.1726
##
## Coefficients: (1 not defined because of singularities)
##                                  Estimate Std. Error z value Pr(>|z|)
## (Intercept)                     -3.594880   0.833065  -4.315 1.59e-05 ***
## age                              0.373536   0.061269   6.097 1.08e-09 ***
## X.education.num.                 0.673725   0.065778  10.242  < 2e-16 ***
## X.capital.gain.                  2.493394   0.250451   9.956  < 2e-16 ***
## X.capital.loss.                  0.245371   0.042760   5.738 9.56e-09 ***
## X.hours.per.week.                0.300397   0.055254   5.437 5.43e-08 ***
## workclass.Local.gov             -1.006931   0.332390  -3.029 0.002451 **
## workclass.Private               -0.539029   0.274691  -1.962 0.049726 *
## workclass.Self.emp.not.inc      -1.309200   0.316098  -4.142 3.45e-05 ***
## workclass.Self.emp.inc          -0.340750   0.359722  -0.947 0.343507
## workclass.State.gov             -0.687738   0.357105  -1.926 0.054121 .
## X.marital.status.Together        2.720466   0.735640   3.698 0.000217 ***
## X.marital.status..Never.married -0.383723   0.242387  -1.583 0.113398
## X.marital.status..Widowed        0.086731   0.493502   0.176 0.860494
## occupation.Adm.clerical         -0.001663   0.299339  -0.006 0.995567
## occupation.Craft.repair          0.465543   0.244614   1.903 0.057018 .
```

```
## occupation.Exec.managerial        1.274717   0.255968   4.980 6.36e-07 ***
## occupation.Farming.fishing       -0.540784   0.409973  -1.319 0.187146
## occupation.Handlers.cleaners     -0.046607   0.361563  -0.129 0.897433
## occupation.Machine.op.inspct     -0.010897   0.292389  -0.037 0.970272
## occupation.Other.service         -1.102110   0.384192  -2.869 0.004122 **
## occupation.Priv.house.serv      -12.290550 317.265135  -0.039 0.969098
## occupation.Prof.specialty         1.107207   0.270743   4.090 4.32e-05 ***
## occupation.Protective.serv        0.747924   0.366127   2.043 0.041072 *
## occupation.Sales                  0.362889   0.259145   1.400 0.161415
## occupation.Tech.support           1.083026   0.331553   3.267 0.001089 **
## occupation.Transport.moving             NA         NA      NA       NA
## relationship.Not.in.family        0.758610   0.725616   1.045 0.295805
## relationship.Other.relative       0.039298   0.646398   0.061 0.951522
## relationship.Own.child           -0.219917   0.698078  -0.315 0.752738
## relationship.Unmarried            0.584463   0.778821   0.750 0.452986
## relationship.Wife                 1.048383   0.294719   3.557 0.000375 ***
## sex.Male                          0.691420   0.225653   3.064 0.002183 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 4564.4  on 4083  degrees of freedom
## Residual deviance: 2589.6  on 4052  degrees of freedom
## AIC: 2653.6
##
## Number of Fisher Scoring iterations: 14
```

```
#predicts P(clss=1|Xs)
log.some.probs=predict(Logist.some,type="response")
#set class 1 assignment threshold of 0.5
log.some.pred=rep(0,length(log.some.probs))
log.some.pred[log.some.probs>.5]=1
log.some.pred=factor(log.some.pred)
#confusion table. rows are pred, cols actual
log.some.table=table(pred=log.some.pred,resp=censcaleDatdum$outcome..50K)
log.some.table
```

```
##     resp
## pred    0    1
##    0 2867  365
##    1  209  643
```

```
#if call 0 the "negative" outcome and 1 the "positive" outcome
#training error - missclassified/total
1-(sum(diag(log.some.table))/sum(log.some.table))
```

```
## [1] 0.1405485
```

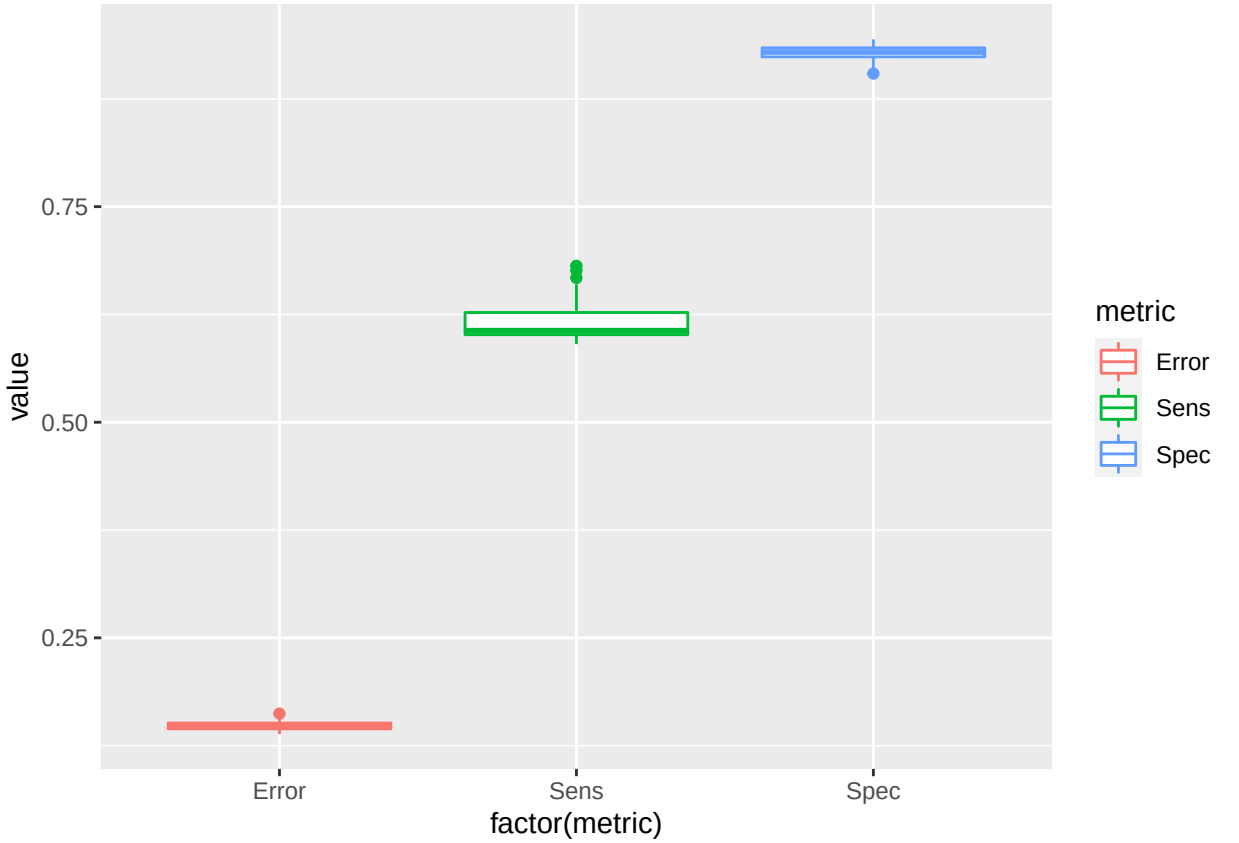
```
#Training sensitivity - TP/P, true pos rate
log.some.table[2,2]/sum(log.some.table[,2])
```

```
## [1] 0.6378968
```

```
#training Specificity - TN/N, false pos rate
log.some.table[1,1]/sum(log.some.table[,1])
```

```
## [1] 0.9320546
```

```
#test performance

oldw <- getOption("warn") #turn off warnings
options(warn = -1)

dfTmp <- NULL
nTries <- 25
for ( iTry in 1:nTries ) {
#split data
s=sample(c(TRUE,FALSE),nrow(censcaleDatdum),replace = TRUE)
censcaleDatdum.test<-NULL
censcaleDatdum.train<-NULL
censcaleDatdum.train=censcaleDatdum[s,]
censcaleDatdum.test=censcaleDatdum[!s,]

log.fit=glm(model.atts,data=censcaleDatdum.train,family=binomial)
#predict logistic
log.probs=predict(log.fit,newdata=censcaleDatdum.test,type="response")
log.pred=rep(0,length(log.probs))
log.pred[log.probs>.5]=1 #set class 1 assignment threshold of 0.5
log.pred=factor(log.pred)
#confusion table
t=table(log.pred,censcaleDatdum.test$outcome)
#Testing error rate
ER=(t[1,2]+t[2,1])/nrow(censcaleDatdum.test)
#Testing sensitivity
TPR=t[2,2]/(t[1,2]+t[2,2])
#testing Specificity
TNR=t[1,1]/(t[1,1]+t[2,1])
dfTmp=rbind(dfTmp,data.frame(sim=iTry,method="Logistic",metric=c("Error","Sens","Spec"),value=c(ER,TPR,T
}

options(warn = oldw)

ggplot(dfTmp,aes(x=factor(metric),y=value,colour=metric)) + geom_boxplot()
```




```
signif(mean(dfTmp[dfTmp$metric=="Error","value"]),4)
```

```
## [1] 0.1481
```

```
signif(mean(dfTmp[dfTmp$metric=="Sens","value"]),4)
```

```
## [1] 0.6184
```

```
signif(mean(dfTmp[dfTmp$metric=="Spec","value"]),4)
```

```
## [1] 0.9285
```

```
DescPfrm
```

```
##                 Algorithm Error
## 1                      C4.5 15.54
## 2                 C4.5-auto 14.46
## 3                C4.5 rules 14.94
## 4          Voted ID3 (0.6) 15.64
## 5          Voted ID3 (0.8) 16.47
## 6                       T2 16.84
## 7                       1R 19.54
## 8                   NBTree 14.10
## 9                      CN2 16.00
## 10                   HOODG 14.82
## 11         FSS Naive Bayes 14.05
## 12 IDTM (Decision table) 14.46
## 13            Naive-Bayes 16.12
## 14  Nearest-neighbor (1) 21.42
```

```
## 15  Nearest-neighbor (3) 20.35
## 16                   OC1 15.04
```

I ran a logistic model for all variables but *education*, and all of them but *race* and *native-country* have at least one of their dummy variables with significant p-values (although it's possible the parts of the dummies bundled into the intercept would have been significant). It is interesting that the largest coefficient is *marital-status's* "Together". The next largest is *capital-gain* then *relationship's* "not-in-family" and "Wife". It's also interesting that the *workclass* dummy variables have high p-values even though it was the only categorical univariate model that showed significance in the Fisher Exact test.

I re-ran the logistic model, excluding *native-country* and *race*. Then, using a 0.5 threshold, It has an error around 16%, predicted about 60% of the high income category correctly and about 90% of low income category correctly. Depending on our purpose it may be useful to lower the threshold to increase the true positive rate. I tested a model by splitting the data into training and test sets 25 times and got similar test results. The dataset description provides their model performances. My Logistic model performs simlar to Naive-Bayes, Voted ID3 and C4.5.

Random Forest Classification

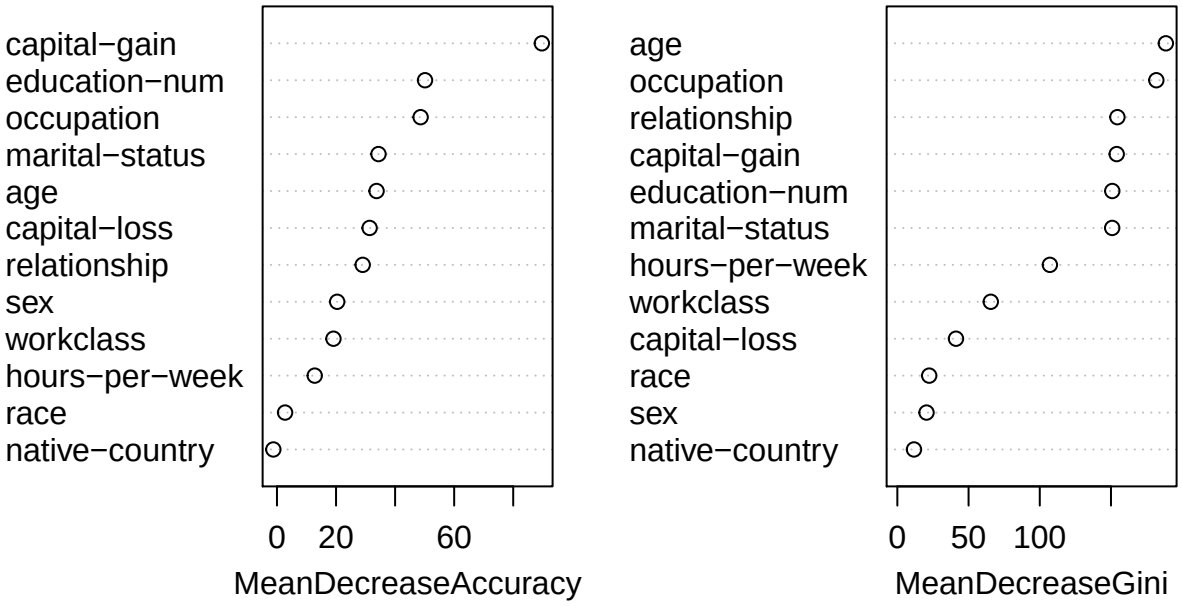
Next I develop a random forest model of the categorized income where I summarize variable importance and model performance and compare it to the results from logistic regression as well as to other methods reported in the dataset description.

```
RF.all=randomForest(censusDat[,-c(3,14)],censusDat$outcome,importance=TRUE)
RF.all
```

```
##
## Call:
##  randomForest(x = censusDat[, -c(3, 14)], y = censusDat$outcome,      importance = TRUE)
##                Type of random forest: classification
##                      Number of trees: 500
## No. of variables tried at each split: 3
##
##         OOB estimate of  error rate: 14.03%
## Confusion matrix:
##         <=50K  >50K class.error
##  <=50K   2872   204   0.0663199
##  >50K     369   639   0.3660714
```

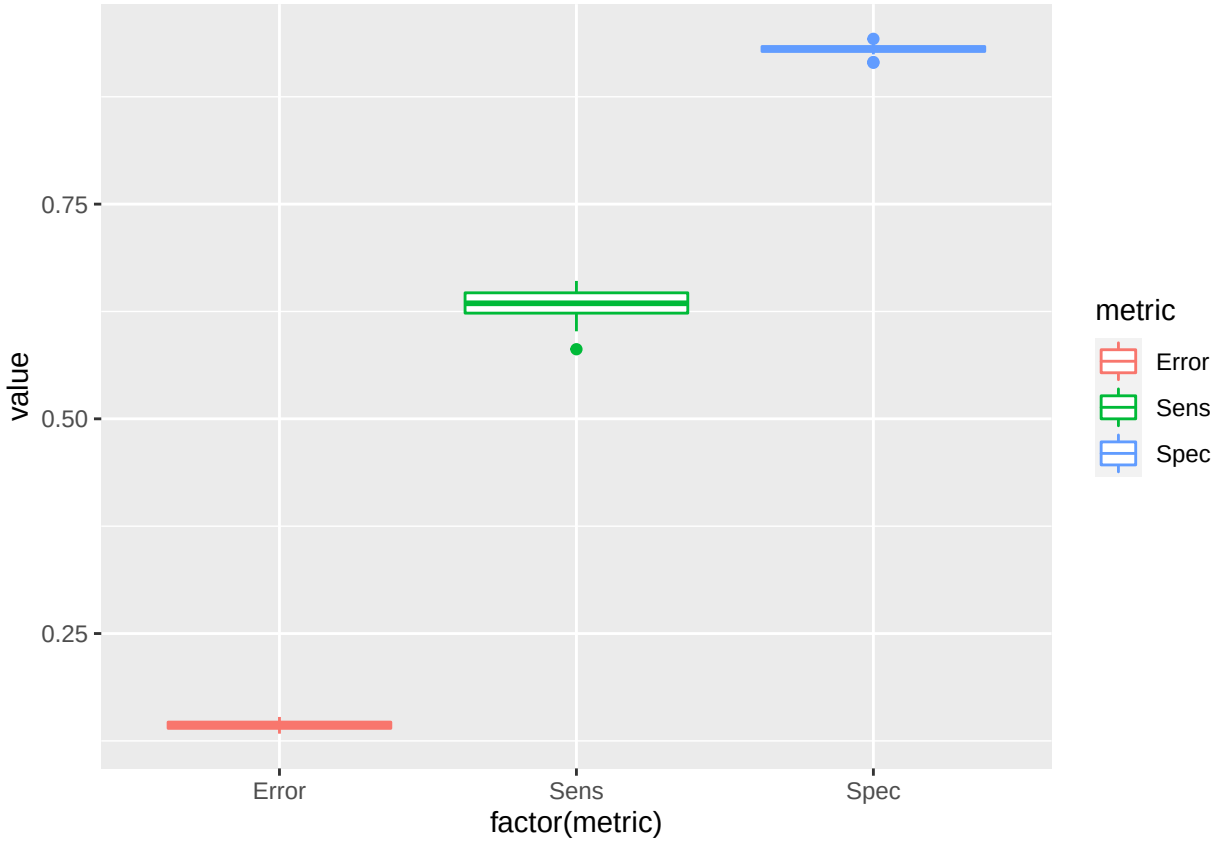
```
varImpPlot(RF.all)
```

RF.all

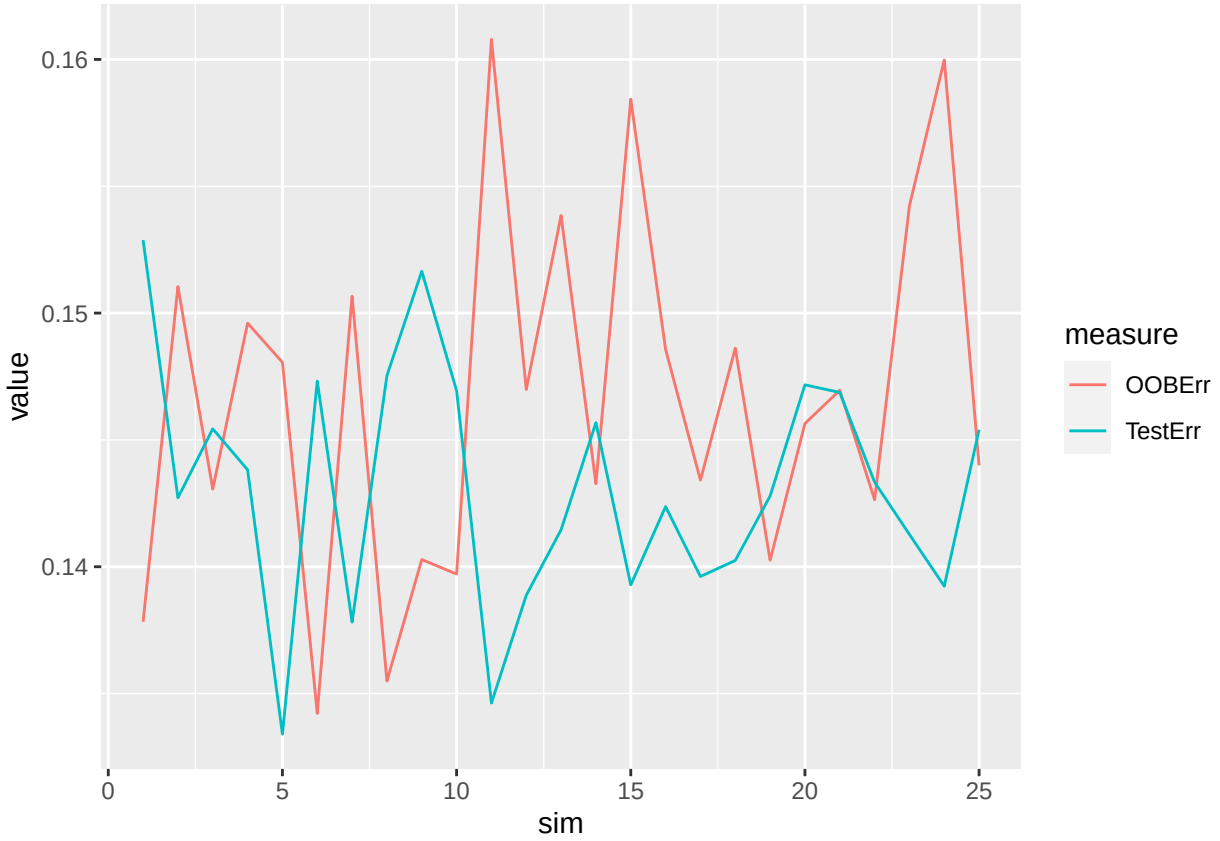



```
#gini is avg of differences between every pair divided by total mean. In this case coefficient of 1 mea


#test performance
nTries <- 25
OOBTest <- NULL
for ( iTry in 1:nTries ) {
#split data
s <- sample(c(FALSE,TRUE),nrow(censusDat),replace=TRUE)
rfRes <- randomForest(censusDat[s,-c(3,14)],censusDat$outcome[s],importance=TRUE)
# store OOB error
OOB=rfRes$err.rate[[nrow(rfRes$err.rate),1]]
#predict RF and table
t <- table(predict(rfRes,newdata=censusDat[!s,-c(3,14)]),censusDat$outcome[!s])
#Testing error rate
ER=(t[1,2]+t[2,1])/sum(t)
#Testing sensitivity
TPR=t[2,2]/(t[1,2]+t[2,2])
#testing Specificity
TNR=t[1,1]/(t[1,1]+t[2,1])
#compare test error with OOB error
OOBTest=rbind(OOBTest,data.frame(sim=iTry,measure=c("OOBErr","TestErr"),value=c(OOB,ER)))
#add to compare to logistic
dfTmp=rbind(dfTmp,data.frame(sim=iTry,method="RF",metric=c("Error","Sens","Spec"),value=c(ER,TPR,TNR)))
}
```

```
ggplot(dfTmp[dfTmp$method=="RF",],aes(x=factor(metric),y=value,colour=metric)) + geom_boxplot()
```




```
ggplot(OOBTest,aes(x=sim,y=value,colour=measure)) +geom_line()
```




```
signif(mean(dfTmp[dfTmp$metric=="Error" & dfTmp$method=="RF","value"]),4)
```

```
## [1] 0.1431
```

```
signif(mean(dfTmp[dfTmp$metric=="Sens" & dfTmp$method=="RF","value"]),4)
```

```
## [1] 0.632
```

```
signif(mean(dfTmp[dfTmp$metric=="Spec" & dfTmp$method=="RF","value"]),4)
```

```
## [1] 0.9303
```

```
DescPfrm
```

```
##                 Algorithm Error
## 1                    C4.5 15.54
## 2              C4.5-auto 14.46
## 3             C4.5 rules 14.94
## 4        Voted ID3 (0.6) 15.64
## 5        Voted ID3 (0.8) 16.47
## 6                      T2 16.84
## 7                      1R 19.54
## 8                  NBTree 14.10
## 9                     CN2 16.00
## 10                 HOODG 14.82
## 11       FSS Naive Bayes 14.05
## 12 IDTM (Decision table) 14.46
## 13           Naive-Bayes 16.12
## 14  Nearest-neighbor (1) 21.42
```

```
## 15  Nearest-neighbor (3) 20.35
## 16                   OC1 15.04
```

I trained a Random Forest model on all attributes but *education*. The result looks fairly good. *capital-gain*,*occupation*, and *education-num* are the three most important attributes measured by mean decrease in accuracy. Using gini index it's *age*,*occupation* and *education-num*. *sex*,*race* and *native-country* are least important under both measures. In the Logistic model, *native-country* and *race* were also not important and while *sex* was significant, it did not have a strong impact on the model. *capital-gain*, *occupation*, *education-num*, and *age* are all significant in the Logistic model. *capital-gain* as well as some of the *occupation* categories have some of the highest coefficients. So for the most part the variable importance is similar between Random Forest and Logistic.

Again, I resampled 25 times to create test results. The average error was around 0.15, Sensitivity around 0.60 and specificity around 0.90

I also compared the out of the bag (OOB) error to test error. OOB error is determined by predicting each observation using all trees that didn't include the observation during traning. This approach is similar to cross-validation so I would expect the errors to be simlar and they are.

These error results are slightly better than the Logistic Regression model and comparable to C4.5 rules, HOODG and OC1 methods from the dataset description.

Support Vector Machines

I develop a Support Vector Machines (SVM) model, summarize its performance and compare it to the performance of other methods reported in the dataset description.

```
 #see which kernel yields best results
tune.lin = tune(svm,model.atts,data=censcaleDatdum,kernel="linear",scale=FALSE,ranges=list(cost=c(.5,1,
  summary(tune.lin)
```

```
##
## Parameter tuning of 'svm':
##
## - sampling method: 10-fold cross validation
##
## - best parameters:
##  cost
##     2
##
## - best performance: 0.1518128
##
## - Detailed performance results:
##   cost     error dispersion
## 1  0.5 0.1520585 0.02405646
## 2  1.0 0.1520585 0.02481529
## 3  2.0 0.1518128 0.02425499
## 4  5.0 0.1527932 0.02305705
```

```
tune.rad = tune(svm,model.atts,data=censcaleDatdum,kernel="radial",scale=FALSE,ranges=list(cost=c(.5,1,
  summary(tune.rad)
```

```
##
## Parameter tuning of 'svm':
##
## - sampling method: 10-fold cross validation
##
```

```
## - best parameters:
##  cost gamma
##     2  0.01
##
## - best performance: 0.1449566
##
## - Detailed performance results:
##    cost gamma     error dispersion
## 1   0.5  0.01 0.1525433 0.02285093
## 2   1.0  0.01 0.1491173 0.02156224
## 3   2.0  0.01 0.1449566 0.02158253
## 4   5.0  0.01 0.1471595 0.02023480
## 5   0.5  0.10 0.1476521 0.02041210
## 6   1.0  0.10 0.1454474 0.02034764
## 7   2.0  0.10 0.1466693 0.02129584
## 8   5.0  0.10 0.1466723 0.02323070
## 9   0.5  1.00 0.1875503 0.02122782
## 10  1.0  1.00 0.1814277 0.02329577
## 11  2.0  1.00 0.1826526 0.02242263
## 12  5.0  1.00 0.1917122 0.02136969
```

```
tune.poly = tune(svm,model.atts,data=censcaleDatdum,kernel="polynomial",scale=FALSE,ranges=list(cost=c(
  summary(tune.poly)
```

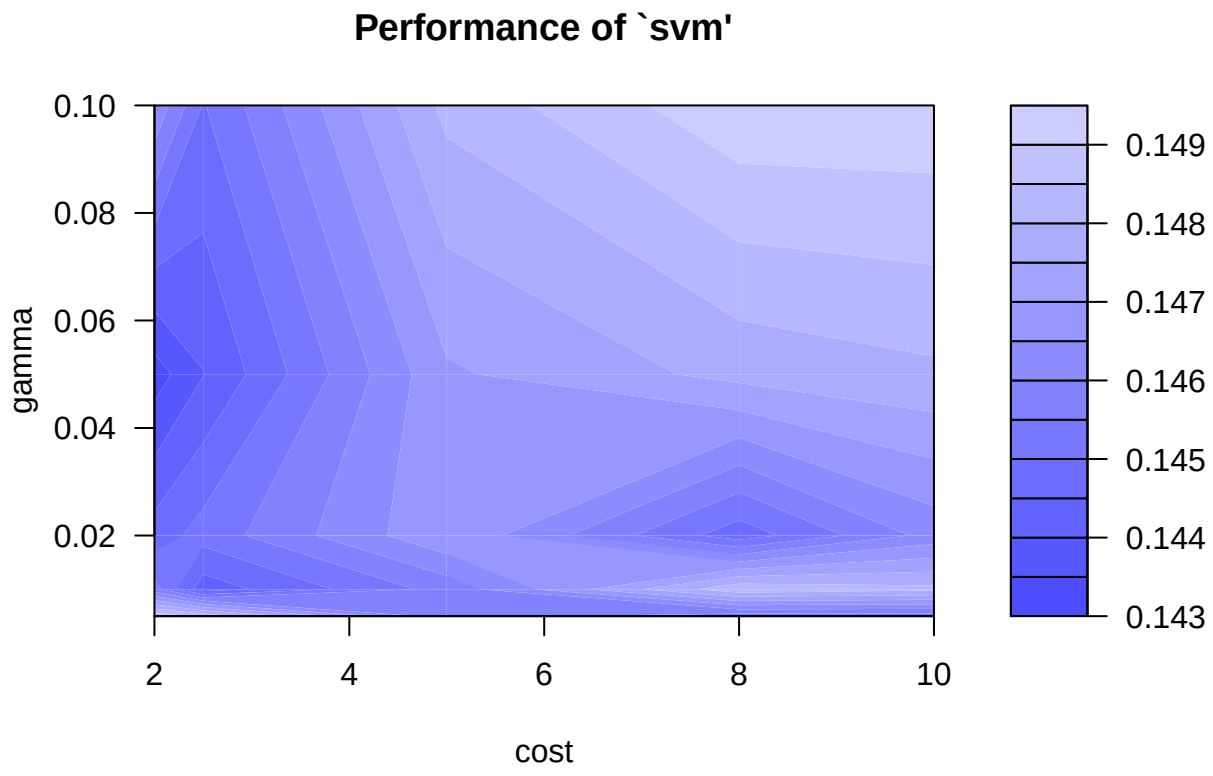
```
##
## Parameter tuning of 'svm':
##
## - sampling method: 10-fold cross validation
##
## - best parameters:
##  cost degree
##     5      2
##
## - best performance: 0.1523006
##
## - Detailed performance results:
##    cost degree     error dispersion
## 1   0.5      2 0.1797282 0.01864271
## 2   1.0      2 0.1665049 0.01744112
## 3   2.0      2 0.1576891 0.01723947
## 4   5.0      2 0.1523006 0.01650801
## 5   0.5      3 0.2250312 0.01741344
## 6   1.0      3 0.2145003 0.01849814
## 7   2.0      3 0.1951556 0.02039895
## 8   5.0      3 0.1753224 0.01830016
## 9   0.5      4 0.2274797 0.01779071
## 10  1.0      4 0.2272346 0.01691900
## 11  2.0      4 0.2247861 0.01772862
## 12  5.0      4 0.2149905 0.01786442
```

```
#Determine which parameters to consider
tune.rad = tune(svm,model.atts,data=censcaleDatdum,kernel="radial",scale=FALSE,ranges=list(cost=c(2,2.5
  summary(tune.rad)
```

```
##
```

```
## Parameter tuning of 'svm':
##
## - sampling method: 10-fold cross validation
##
## - best parameters:
##  cost gamma
##     2  0.05
##
## - best performance: 0.1432487
##
## - Detailed performance results:
##    cost gamma     error dispersion
## 1   2.0 0.005 0.1488776 0.01936047
## 2   2.5 0.005 0.1483880 0.01991973
## 3   5.0 0.005 0.1456931 0.02001036
## 4   8.0 0.005 0.1452011 0.02155599
## 5  10.0 0.005 0.1452011 0.02136939
## 6   2.0 0.010 0.1456949 0.01903902
## 7   2.5 0.010 0.1442261 0.01925681
## 8   5.0 0.010 0.1456907 0.01945175
## 9   8.0 0.010 0.1483862 0.02033087
## 10 10.0 0.010 0.1481405 0.01976471
## 11  2.0 0.020 0.1447145 0.01882623
## 12  2.5 0.020 0.1452035 0.01880398
## 13  5.0 0.020 0.1469174 0.01894026
## 14  8.0 0.020 0.1447145 0.01958522
## 15 10.0 0.020 0.1461845 0.01881616
## 16  2.0 0.050 0.1432487 0.01855495
## 17  2.5 0.050 0.1439828 0.01934700
## 18  5.0 0.050 0.1469234 0.02005305
## 19  8.0 0.050 0.1476569 0.01802137
## 20 10.0 0.050 0.1479026 0.01896963
## 21  2.0 0.100 0.1464338 0.02056700
## 22  2.5 0.100 0.1449650 0.01902386
## 23  5.0 0.100 0.1481459 0.01930562
## 24  8.0 0.100 0.1493714 0.01961576
## 25 10.0 0.100 0.1493714 0.01916795
```

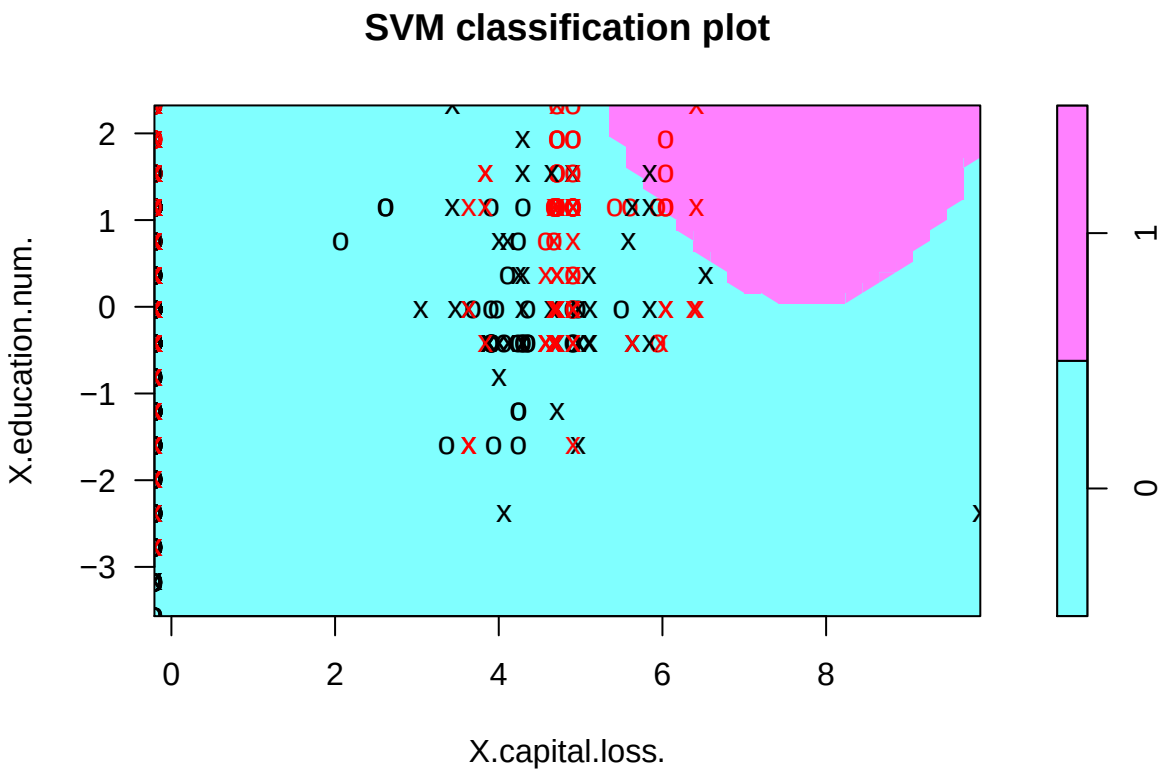
```
plot(tune.rad)
```



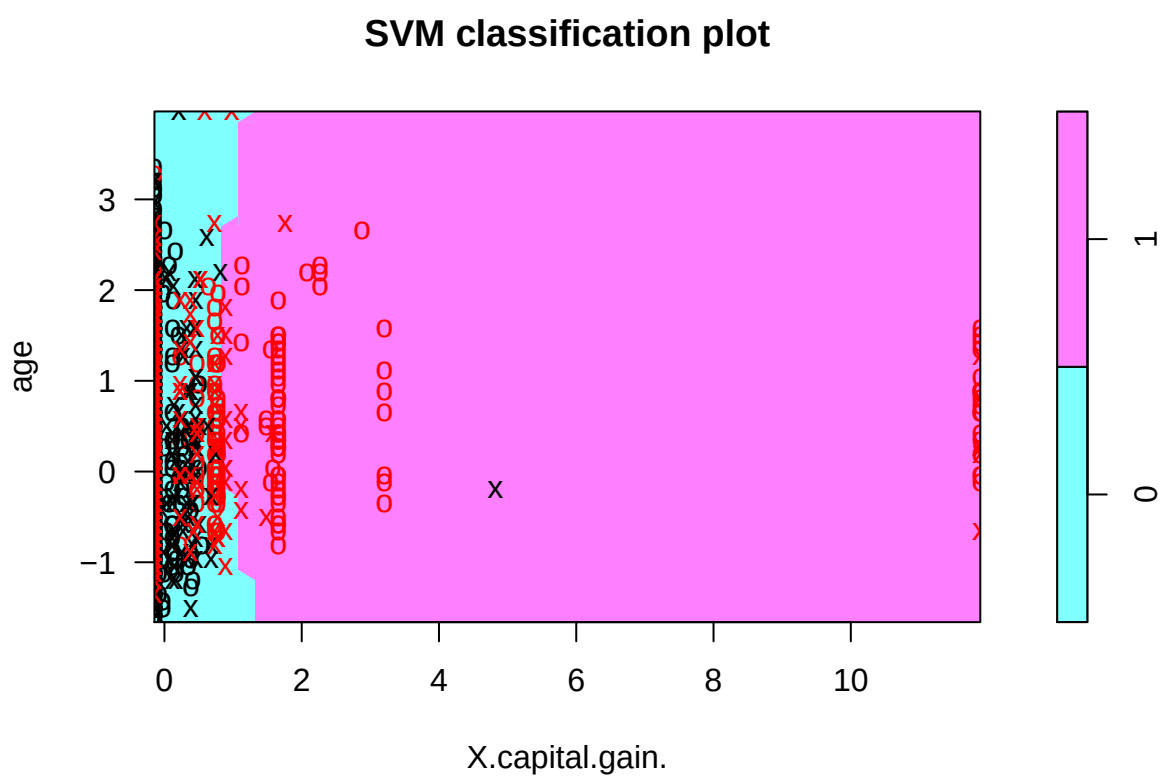

```
summary(tune.rad$best.model)
```

```
##
## Call:
## best.tune(method = svm, train.x = model.atts, data = censcaleDatdum,
##     ranges = list(cost = c(2, 2.5, 5, 8, 10), gamma = c(0.005, 0.01,
##         0.02, 0.05, 0.1)), kernel = "radial", scale = FALSE)
##
##
## Parameters:
##    SVM-Type:  C-classification
##  SVM-Kernel:  radial
##        cost:  2
##
## Number of Support Vectors:  1445
##
##  ( 737 708 )
##
##
## Number of Classes:  2
##
## Levels:
##  0 1
```

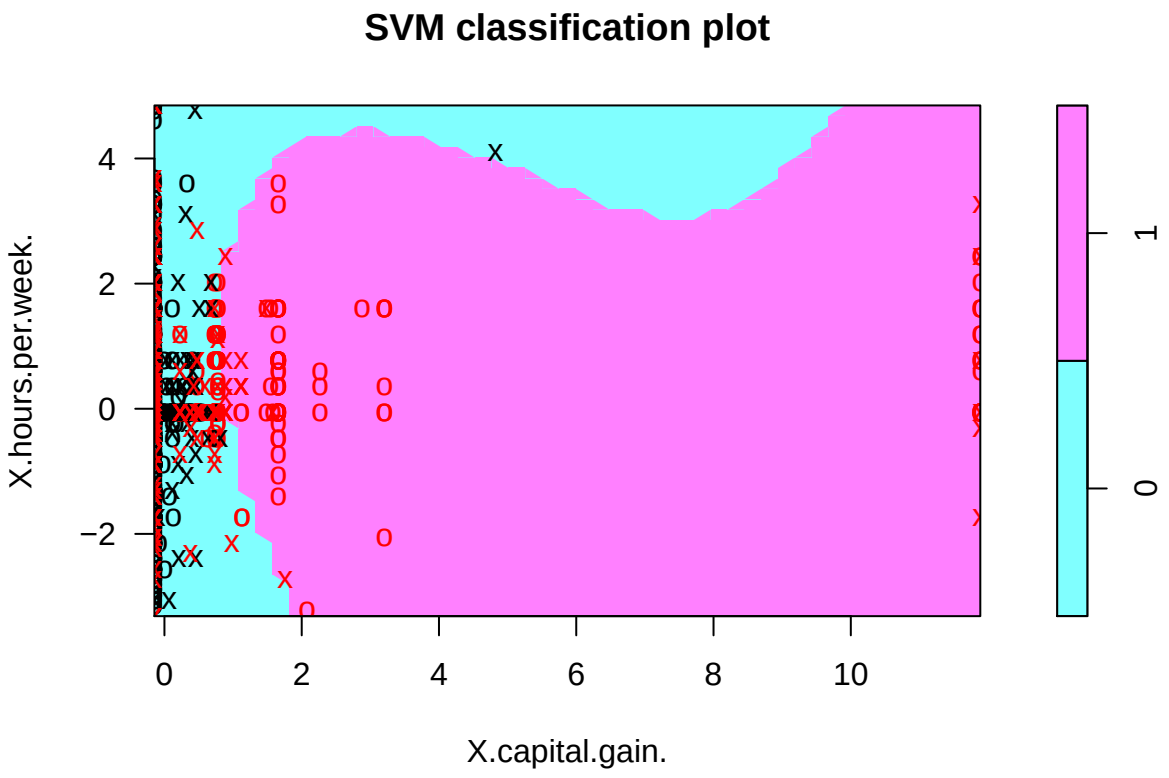
```
#plot 2d
old.par=par(mfrow=c(2,2))
plot(tune.rad$best.model,censcaleDatdum,X.education.num.~X.capital.loss.)
```



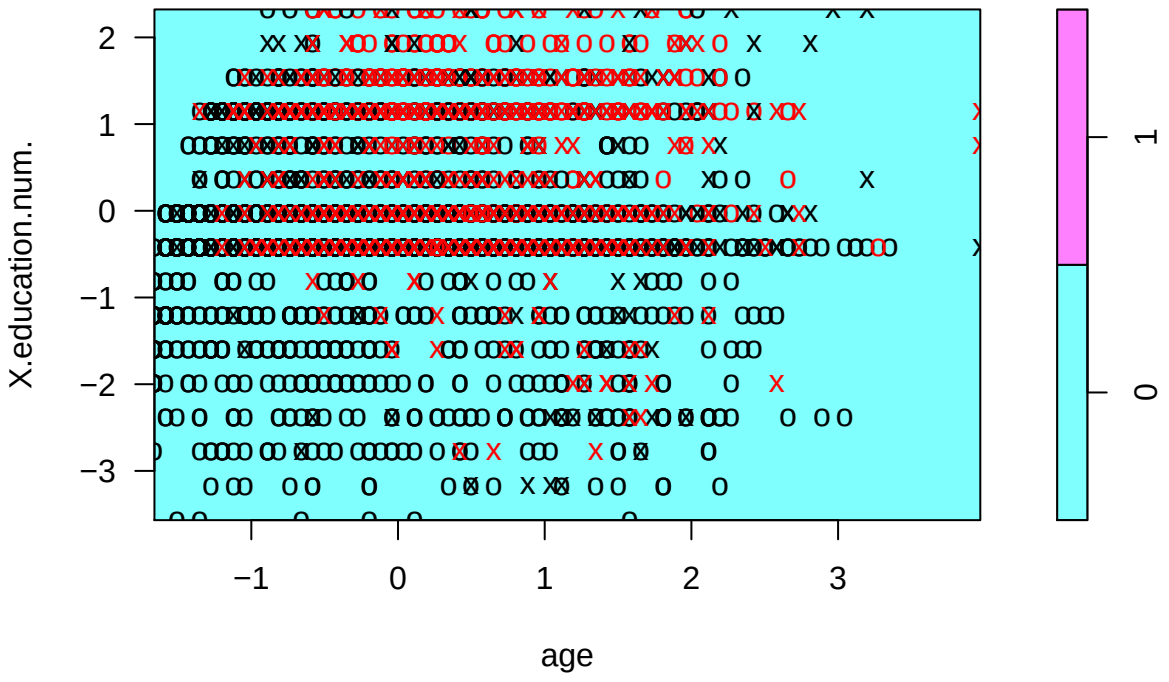

```
plot(tune.rad$best.model,censcaleDatdum,age~X.capital.gain.)
```




```
plot(tune.rad$best.model,censcaleDatdum,X.hours.per.week.~X.capital.gain.)
```




```
plot(tune.rad$best.model,censcaleDatdum,X.education.num.~age)
```

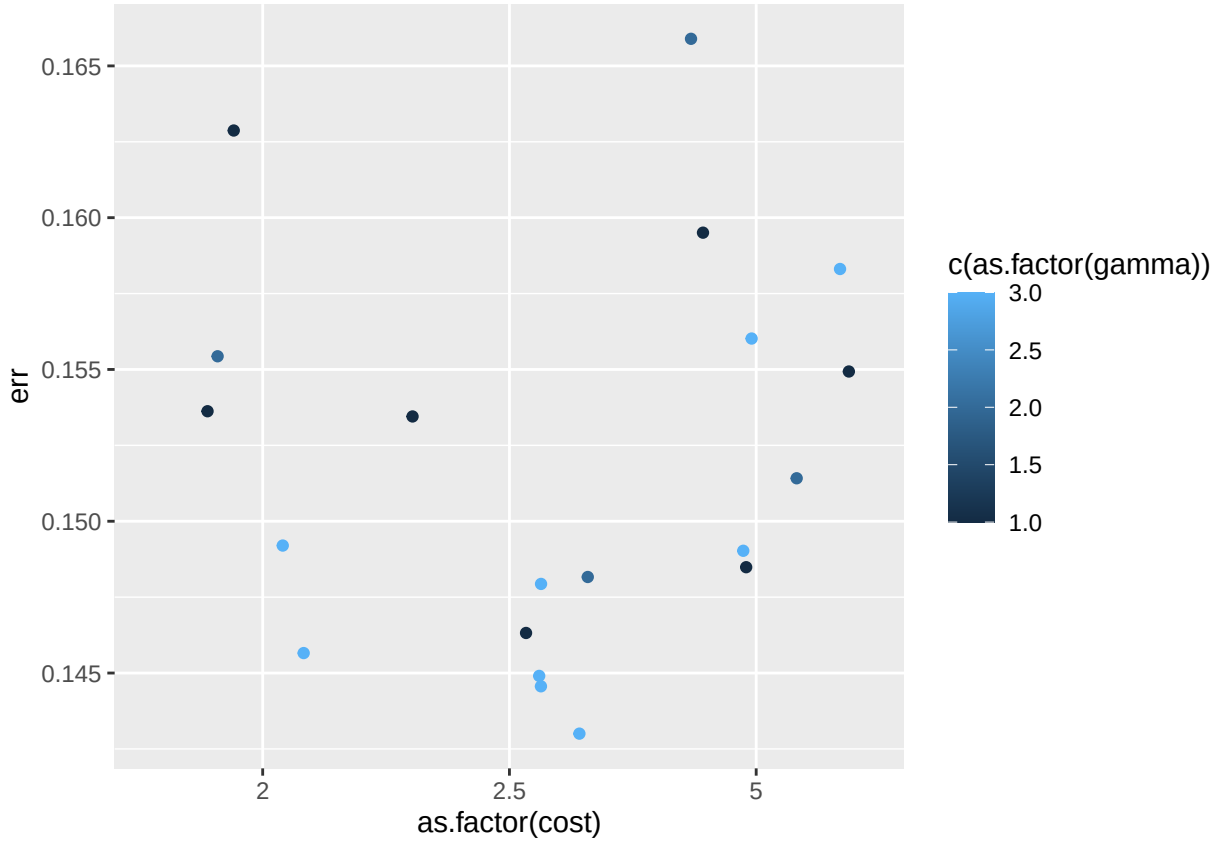



```
par(old.par)
#seems driven by capital gain and loss. even though normalized has the highest range


SVMSamp=data.frame()
#resample
for (iTry in 1:20) {
  #split dataset into test and train
  bTrain <- sample(c(FALSE,FALSE,TRUE),nrow(censcaleDatdum),replace=TRUE)
  # tune svm to train data, obtain test error:
  tmpTune = tune(svm,model.atts,data=censcaleDatdum[bTrain,],kernel="radial",scale=FALSE,ranges=list(cos

  Tmpcost=tmpTune$best.parameters$cost
  Tmpgam=tmpTune$best.parameters$gamma

  tmpTbl <- table(censcaleDatdum$outcome..50K[!bTrain],predict(tmpTune$best.model,newdata=censcaleDatdu
  #Error - testing error rate 1-(TP+TN)/(all obs)
  ER=1-sum(diag(tmpTbl))/sum(tmpTbl)
  #Testing sensitivity
  TPR=tmpTbl[2,2]/(tmpTbl[1,2]+tmpTbl[2,2])
  #testing Specificity
  TNR=tmpTbl[1,1]/(tmpTbl[1,1]+tmpTbl[2,1])

SVMSamp=rbind(SVMSamp,data.frame(cost=Tmpcost,gamma=Tmpgam,err=ER))
#add to compare to logistic and RF
dfTmp=rbind(dfTmp,data.frame(sim=iTry,method="SVM",metric=c("Error","Sens","Spec"),value=c(ER,TPR,TNR)))
```

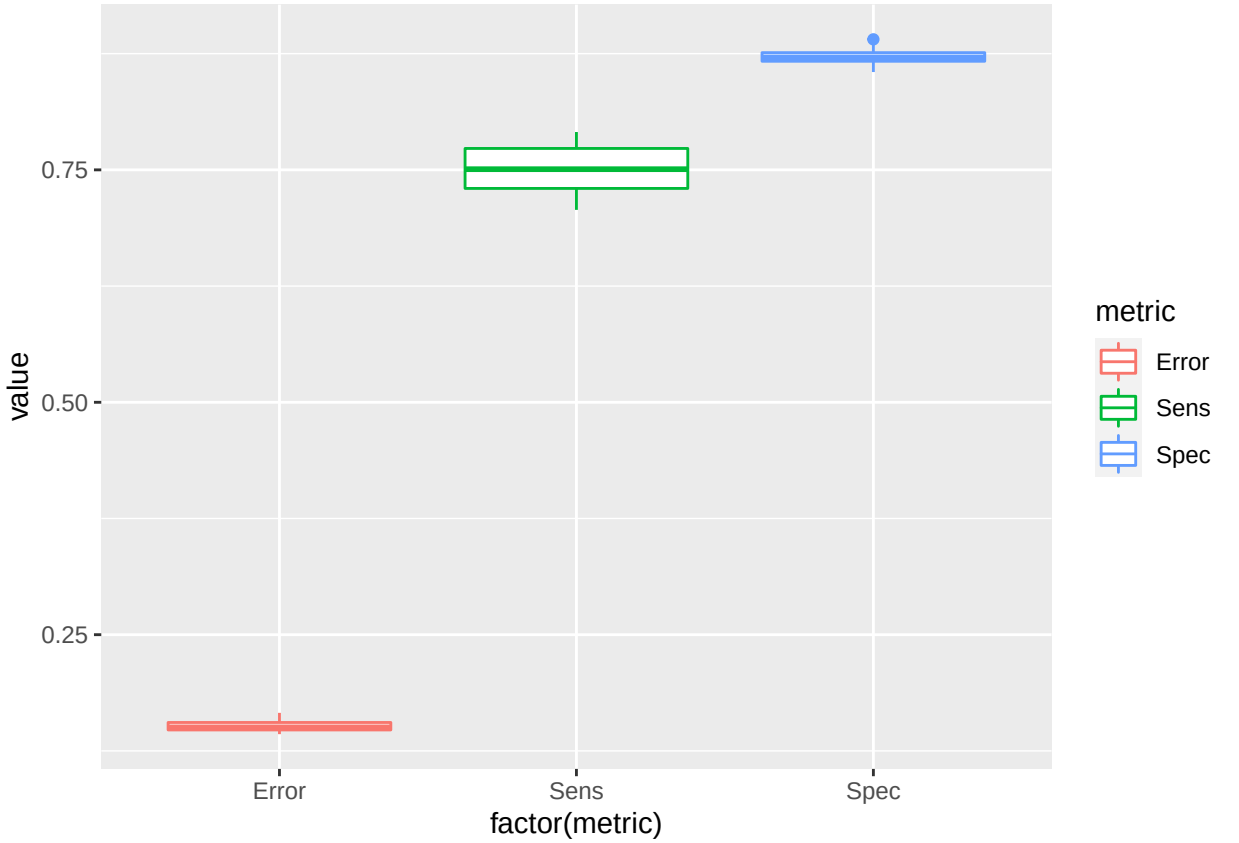
```
}
summary(SVMSamp)
```

```
##       cost           gamma           err        
##  Min.   :2.000   Min.   :0.01   Min.   :0.1430  
##  1st Qu.:2.375   1st Qu.:0.01   1st Qu.:0.1475  
##  Median :2.500   Median :0.02   Median :0.1503  
##  Mean   :3.375   Mean   :0.03   Mean   :0.1519  
##  3rd Qu.:5.000   3rd Qu.:0.05   3rd Qu.:0.1556  
##  Max.   :5.000   Max.   :0.05   Max.   :0.1659
```

```
ggplot(SVMSamp, aes(x=as.factor(cost),y=err,col=c(as.factor(gamma)))) +
  geom_jitter()
```




```
ggplot(dfTmp[dfTmp$method=="SVM",],aes(x=factor(metric),y=value,colour=metric)) + geom_boxplot()
```




```
summary(dfTmp[dfTmp$metric=="Error" & dfTmp$method=="SVM","value"])
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##  0.1430  0.1475  0.1503  0.1519  0.1556  0.1659
```

```
signif(mean(dfTmp[dfTmp$metric=="Error" & dfTmp$method=="SVM","value"]),4)
```

```
## [1] 0.1519
```

```
signif(mean(dfTmp[dfTmp$metric=="Sens" & dfTmp$method=="SVM","value"]),4)
```

```
## [1] 0.7502
```

```
signif(mean(dfTmp[dfTmp$metric=="Spec" & dfTmp$method=="SVM","value"]),4)
```

```
## [1] 0.8714
```

```
DescPfrm
```

```
##                Algorithm Error
## 1                     C4.5 15.54
## 2                C4.5-auto 14.46
## 3               C4.5 rules 14.94
## 4          Voted ID3 (0.6) 15.64
## 5          Voted ID3 (0.8) 16.47
## 6                       T2 16.84
## 7                       1R 19.54
## 8                   NBTree 14.10
## 9                      CN2 16.00
## 10                  HOODG 14.82
```

```
## 11       FSS Naive Bayes 14.05
## 12 IDTM (Decision table) 14.46
## 13           Naive-Bayes 16.12
## 14  Nearest-neighbor (1) 21.42
## 15  Nearest-neighbor (3) 20.35
## 16                   OC1 15.04
```

For creating Support Vector Machine models, I used the normalized dataset with dummy variables. First, I used the tune function with a range of parameters to choose between a radial, polynomial or linear kernel. Radial seemed to provide the best cross-validated results. I then tuned the SVM again with a radial kernel on more parameters to pick the few that would likely give the best results.

I summarized and plotted the best result from this second tune. The decision boundary is visible with *capital-gain* and *capital-loss* but not with other combinations of continuous variables.

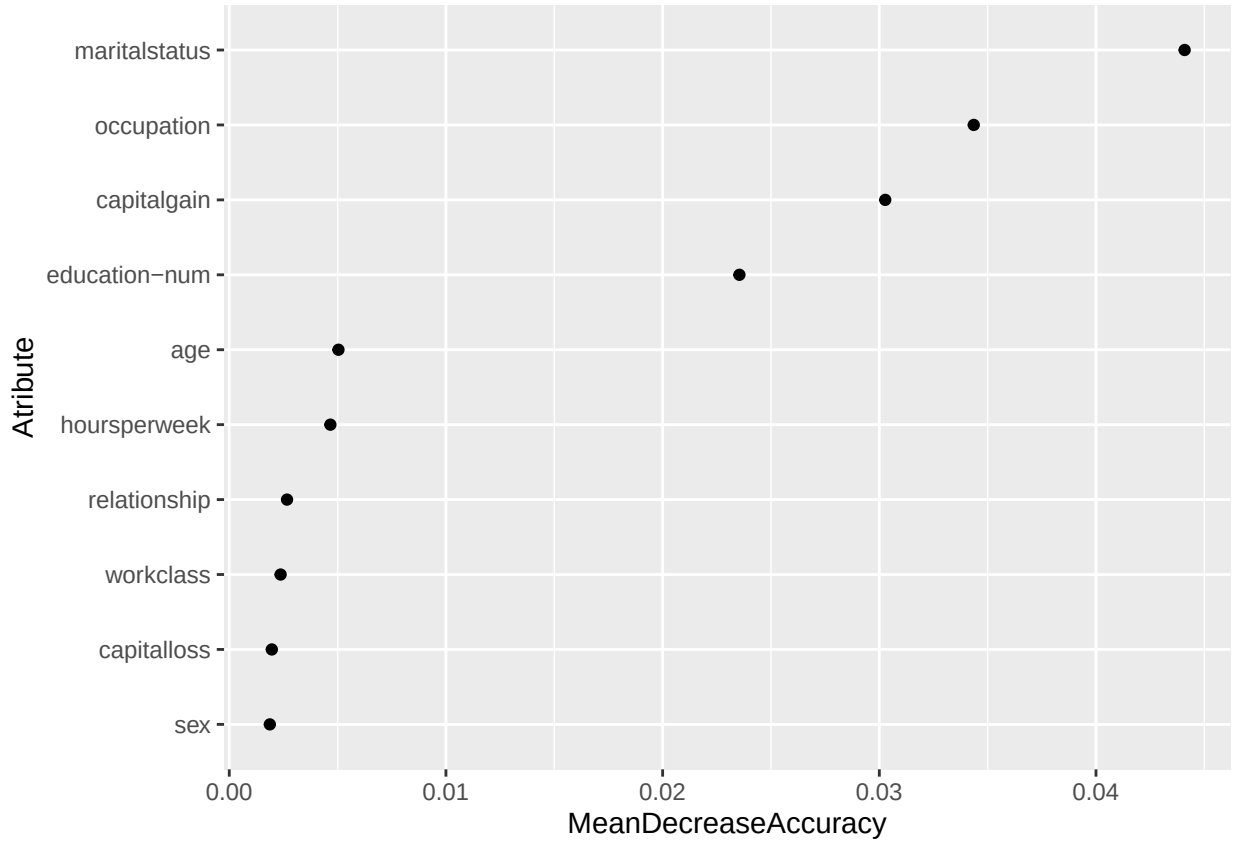
Finally, I used the pared down list of parameters to resample 20 times to train and compute test errors. The error is around 15.25. These numbers are similar to Logistic which makes sense because SVM is a related modeling method. It has comparable error to the dataset description's Oc1,C4.5 rules, and Voted ID3 (0.6). It doesn't perform quite as well as Random Forest. SVM works best for almost separable data which is not what we're working with, Random Forest's use of Bagging can be more flexible in sorting clusters which overlap a lot.

Variable Importance in SVM

The Random Forest function I used has built in measures of variable importance. The mean decrease accuracy is a measure of the decrease in model performance upon randomization of the values of an attribute. I would like to compare the variable importance for each model so I create a similar approach for SVM.

```
#measure decrease in performance when just randomizing that attribute

#create a randomized data frame
rndm.censcaleDat=censcaleDat
for (cols in 1:(ncol(censcaleDat)-1)) {
rndm.censcaleDat[,cols]= sample(censcaleDat[,cols],nrow(censcaleDat),replace=TRUE)
}
#create dummies from randomized data frame
rndm.censcaleDatdum=model.matrix(~age+workclass+`education-num`+`marital-status`+occupation+relationshi
rndm.censcaleDatdum=data.frame(rndm.censcaleDatdum)

#Loop and measure difference in accuracy
AccuDecrease=data.frame()
MeanDecrease=data.frame()
ColList=list("age"=1,"education-num"=10,"capitalgain"=38,"capitalloss"=39,"hoursperweek"=40,"workclass"=

for(Cols in c("age","education-num","capitalgain","capitalloss","hoursperweek","workclass","maritalstatu

for (iTry in 1:12) {

  #split dataset into test and train
  bTrain <- sample(c(FALSE,TRUE),nrow(censcaleDatdum),replace=TRUE)

   # Full data Fit and test
  tmpFit <- svm(model.atts,data=censcaleDatdum[bTrain,],kernel="radial",cost=8,gamma=.02,scale=FALSE)
  tmpTbl <- table(censcaleDatdum$outcome..50K[!bTrain],predict(tmpFit,newdata=censcaleDatdum[!bTrain,]))
```

```
  #with random data Fit and test
  rndmTmpData=cbind(rndm.censcaleDatdum[,ColList[[Cols]]],censcaleDatdum[,-ColList[[Cols]]])
  colnames(rndmTmpData)[1:length(ColList[[Cols]])]=colnames(rndm.censcaleDatdum)[ColList[[Cols]]]

 rndm.tmpFit <-svm(model.atts,data=rndmTmpData[bTrain,],kernel="radial",cost=8,gamma=.02,scale=FALSE)
 rndm.tmpTbl <- table(censcaleDatdum$outcome..50K[!bTrain],predict(tmpFit,newdata=rndmTmpData[!bTrain,])

  Acc=sum(diag(tmpTbl))/sum(tmpTbl)
  Acc.rndm=sum(diag(rndm.tmpTbl))/sum(rndm.tmpTbl)

AccuDecrease=rbind(AccuDecrease,data.frame(sim=iTry,Attribute=Cols,TrueAcc=Acc,RndmAcc=Acc.rndm,Decrease

}
  MeanDecrease=rbind(MeanDecrease,data.frame(Attribute=Cols,MeanDecreaseAccuracy=mean(AccuDecrease[Accu

  }
shuffle=order(MeanDecrease$MeanDecreaseAccuracy)


ggplot(MeanDecrease, aes(x=MeanDecreaseAccuracy,y=reorder(Attribute,MeanDecreaseAccuracy))) +
  geom_point() + ylab("Atribute")
```




```
sum(MeanDecrease$MeanDecreaseAccuracy)
```

```
## [1] 0.1508483
```

```
signif(summary(censcaleDat$outcome)[[2]]/sum(summary(censcaleDat$outcome)),4)
```

```
## [1] 0.2468
```

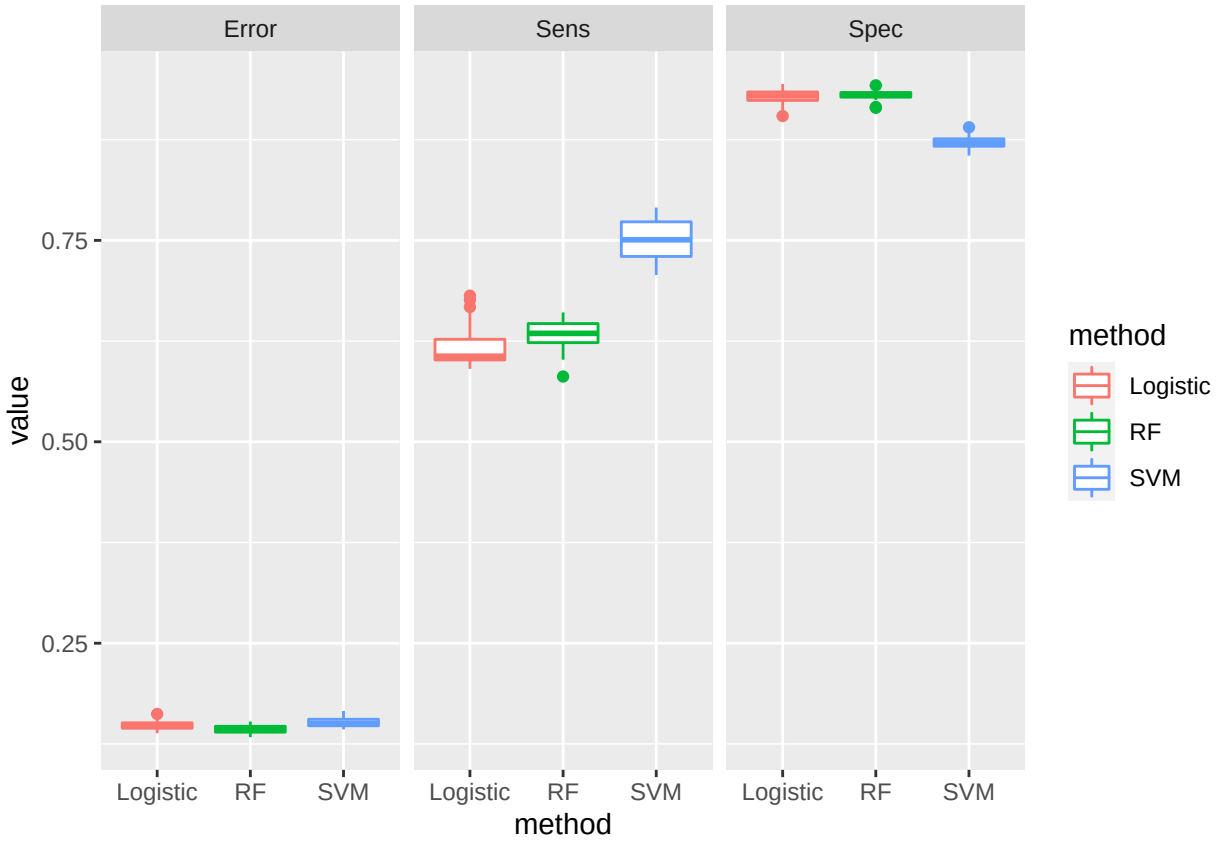
To test variable importance I created a dataframe where each field was individually recreated through bootstrapping. I then ran 12 iterations for each attribute where I measured the difference between test model accuracy using a full model and a model with the attribute randomized across each observation.

marital-status, *capital-gain* and *occupation* are the three attributes which saw the largest decrease in mean accuracy. *sex* and *workclass* had the lowest decrease. *capital-gain* and *occupation* are also important variable in Random Forest and Logistic and *sex* and *workclass* are close to the bottom in all those as well. While *marital-status* is a little above the middle of importance for Random Forest, it is important for Logsitic, where "Together" has a very low p-value and the highest coefficient.

It is surprising to me that mean decrease accuracy is never more than 5 percentage points. However, this makes sense considering that if all datapoints are classified as under 50k, this simple method would already be over 75% accurate. My models only improve on this basic method by around 10 percentage points.

Compare Performance of Logistic Regression, Random Forest and SVM models

```
ggplot(dfTmp,aes(x=method,y=value,colour=method)) + geom_boxplot() + facet_wrap("metric")
```




```
summary(dfTmp[dfTmp$metric=="Error" & dfTmp$method=="Logistic","value"])
```

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

```
##  0.1383  0.1445  0.1470  0.1481  0.1512  0.1622
```

```
summary(dfTmp[dfTmp$metric=="Error" & dfTmp$method=="RF","value"])
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##  0.1334  0.1396  0.1428  0.1431  0.1469  0.1529
```

```
summary(dfTmp[dfTmp$metric=="Error" & dfTmp$method=="SVM","value"])
```

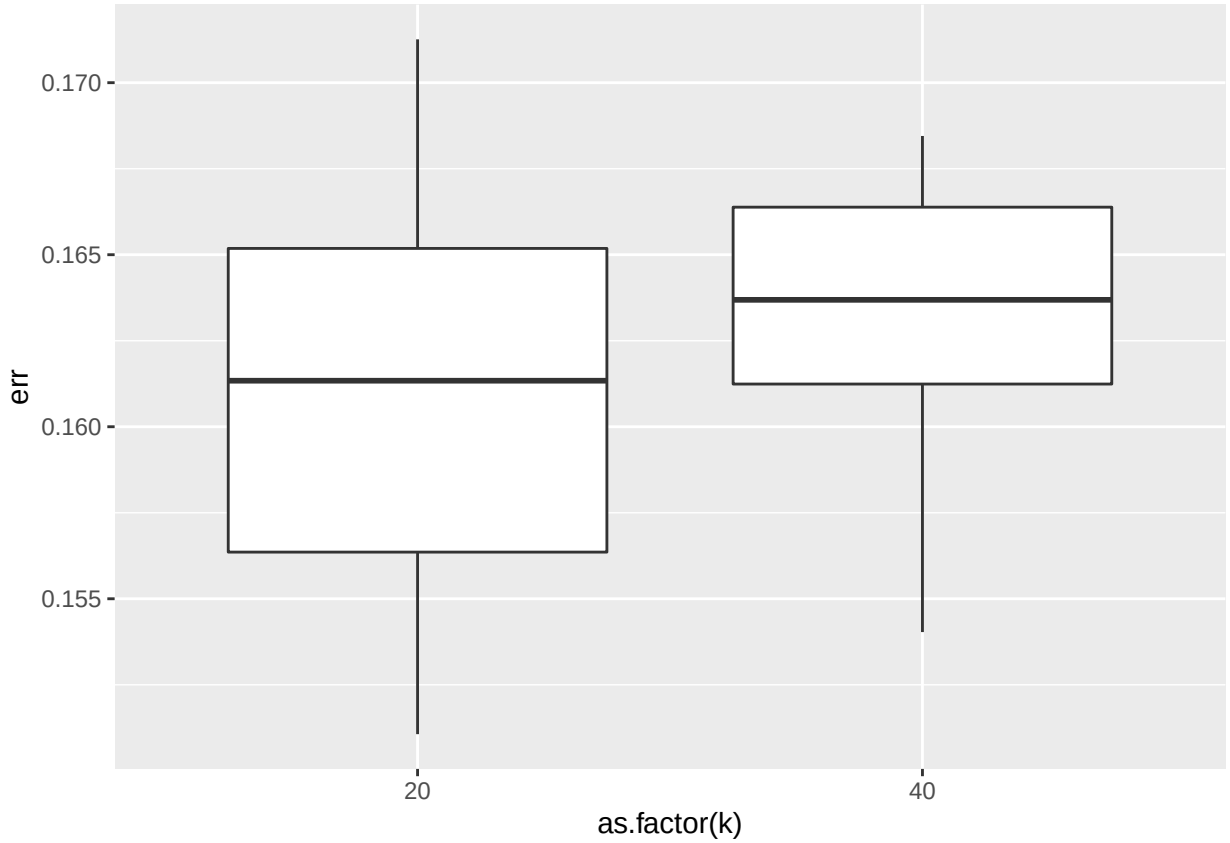
```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##  0.1430  0.1475  0.1503  0.1519  0.1556  0.1659
```

Random Forest has the lowest error followed by SVM but they are all simlar. These errors were very stable over multiple trials. SVM has the highest sensitivity which could be a nice feature - the error is still very low but it's able to sort more of the vectors with income greater than 50k accurately. On the other hand it also has a corresponding lower specificty than the other two models. Sensitivity is less stable over 25 trials than error or specifity which is likely because there is less data in the over 50k category.

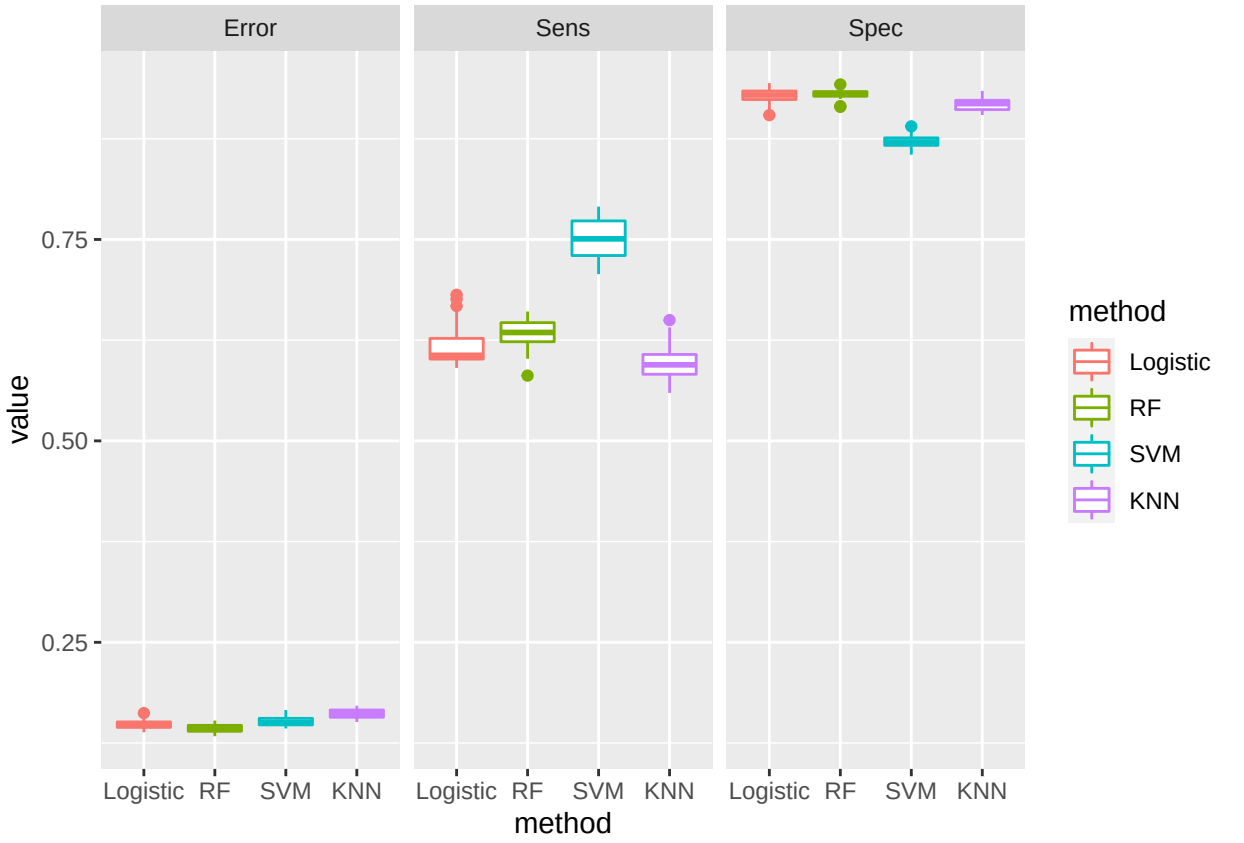
KNN model

Finally, I develop a KNN model for this data and evaluate its performance for different values of k and see how it compares to the other methods attempted.

```
KNNSamp <- data.frame()
for (iTry in 1:25) {
  #split dataset into test and train
  bTrain <- sample(c(FALSE,TRUE),nrow(censcaleDatdum),replace=TRUE)
  # use tune on training data to find best k
 tmptune=tune.knn(censcaleDatdum[,-ncol(censcaleDatdum)],as.factor(censcaleDatdum$outcome..50K),k=c(20,
 ksize=tmptune$best.parameters
 # use optimal k to classify test data
 TmpKNNfit=knn(censcaleDatdum[bTrain,-ncol(censcaleDatdum)],censcaleDatdum[!bTrain,-ncol(censcaleDatdum
 tmpTbl = table(TmpKNNfit,censcaleDatdum$outcome..50K[!bTrain])
 #Error
  ER=1-sum(diag(tmpTbl))/sum(tmpTbl)
  #Testing sensitivity
  TPR=tmpTbl[2,2]/(tmpTbl[1,2]+tmpTbl[2,2])
  #testing Specificity
  TNR=tmpTbl[1,1]/(tmpTbl[1,1]+tmpTbl[2,1])

  #store classification error
  KNNSamp<- rbind(KNNSamp,data.frame(k=ksize,err=ER))

  #add to compare to logistic RF and SVM
dfTmp=rbind(dfTmp,data.frame(sim=iTry,method="KNN",metric=c("Error","Sens","Spec"),value=c(ER,TPR,TNR)))
}

ggplot(KNNSamp, aes(x=as.factor(k),y=err)) +
  geom_boxplot()
```




```
ggplot(dfTmp,aes(x=method,y=value,colour=method)) + geom_boxplot() + facet_wrap("metric")
```




```
DescPfrm
```

```
##                 Algorithm Error
## 1                    C4.5 15.54
## 2               C4.5-auto 14.46
## 3              C4.5 rules 14.94
## 4         Voted ID3 (0.6) 15.64
## 5         Voted ID3 (0.8) 16.47
## 6                      T2 16.84
## 7                      1R 19.54
## 8                  NBTree 14.10
## 9                     CN2 16.00
## 10                  HOODG 14.82
## 11        FSS Naive Bayes 14.05
## 12 IDTM (Decision table) 14.46
## 13           Naive-Bayes 16.12
## 14  Nearest-neighbor (1) 21.42
## 15  Nearest-neighbor (3) 20.35
## 16                   OC1 15.04
```

KNN assigns membership using Euclidean distance so I used normalized continous variables and categorical variables converted to dummies. I resampled 25 times, using the tune function to choose the best cross validated number of neighbors. $k = 20$ and $k = 40$ were most commonly the best models. The error is between 0.170 and 0.185. This is comparable to T2, and 1R in the dataset description errors which is better than the two KNN models listed which have errors around .20. KNN does not perform well on this dataset. It has the highest error and lowest sensitivity.