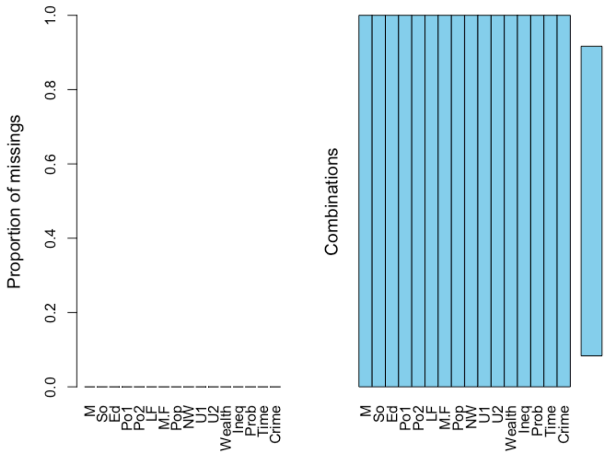
Georgia Tech - SU18: Introduction to Analytics Modeling

Homework Assignment: Week 4

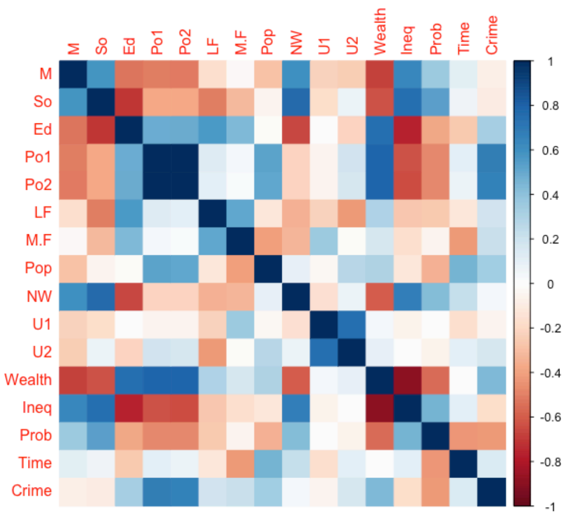
Question 9.1
Using the same crime data set uscrime.txt as in Question 8.2, apply Principal Component Analysis and then create a regression model using the first few principal components. Specify your new model in terms of the original variables (not the principal components), and compare its quality to that of your solution to Question 8.2. You can use the R function prcomp for PCA. (Note that to first scale the data, you can include scale. = TRUE to scale as part of the PCA function. Don't forget that, to make a prediction for the new city, you'll need to unscale the coefficients (i.e., do the scaling calculation in reverse)!)

a. PCA
   I imported the data into r and prepared and tested the data for integrity. I then ran the function prcomp on scaled data and got the following results:

   Data integrity – all columns have the same number of data points:



   I also ran a correlation matrix to visualize any strong correlations in the data set:

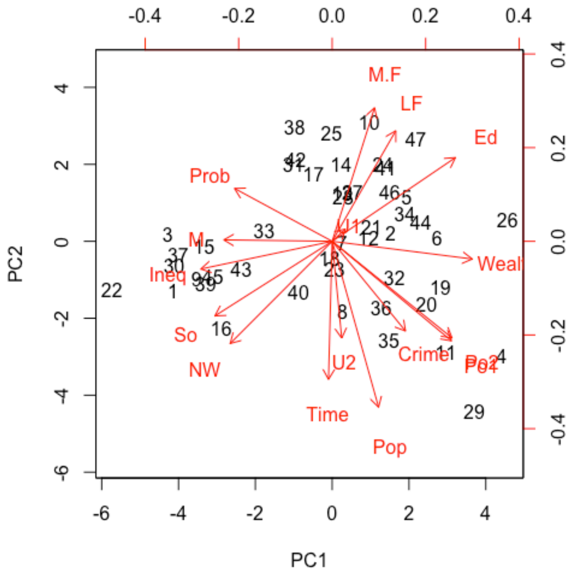


   Finally, I ran the prcomp function and the results were as follow:

Importance of components:

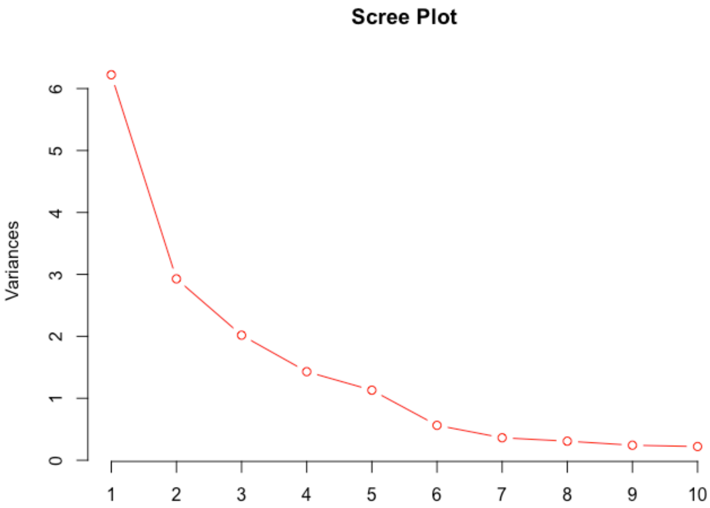
	PC1	PC2	PC3	PC4	PC5	PC6	PC7	PC8	PC9	PC10	PC11	PC12	PC13	PC14	PC15	PC16
Standard deviation	2.4944	1.7111	1.4208	1.19585	1.06341	0.75087	0.60237	0.55503	0.49244	0.47036	0.43856	0.41777	0.29147	0.26063	0.21813	0.06584
Proportion of Variance	0.3889	0.183	0.1262	0.08938	0.07068	0.03524	0.02268	0.01925	0.01516	0.01383	0.01202	0.01091	0.00531	0.00425	0.00297	0.00027
Cumulative Proportion	0.3889	0.5719	0.6981	0.78744	0.85812	0.89336	0.91603	0.93529	0.95044	0.96427	0.97629	0.9872	0.99251	0.99676	0.99973	1

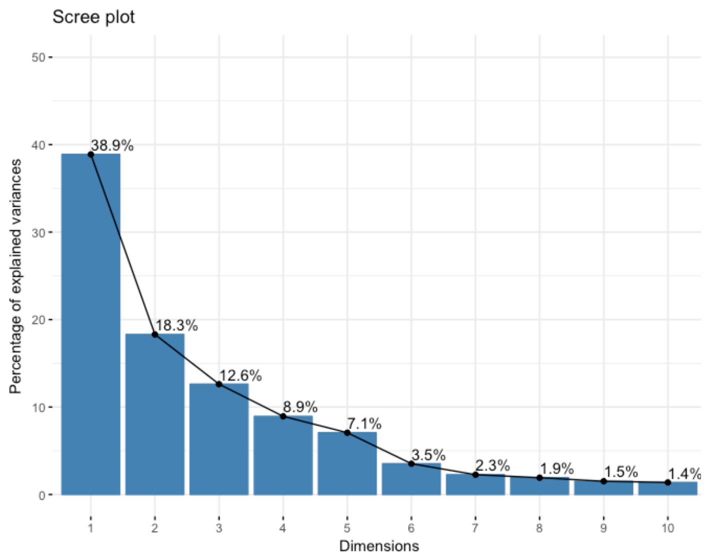
I then extracted means (center in the results) and standard deviations and plot the resultant principle components.



I then calculated the variances of the data to get the proportion variance explained. As seen below and confirmed by the scree plot, the first 5 components explain 98% of the data:

ProportionVariance
[1] 0.70 0.15 0.07 0.04 0.02 0.01 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00





Now to get the regression, I created a new data set using the first 5 components and the "Crime" column of the original crime data as this is the dataset we want to test for.

The resulting regression is as follow:
Call:

lm(formula = Crime ~ ., data = crimedatanew)

Residuals:
```
     Min       1Q   Median       3Q      Max 
-305.496  -89.435    6.064   73.323  281.078 
```

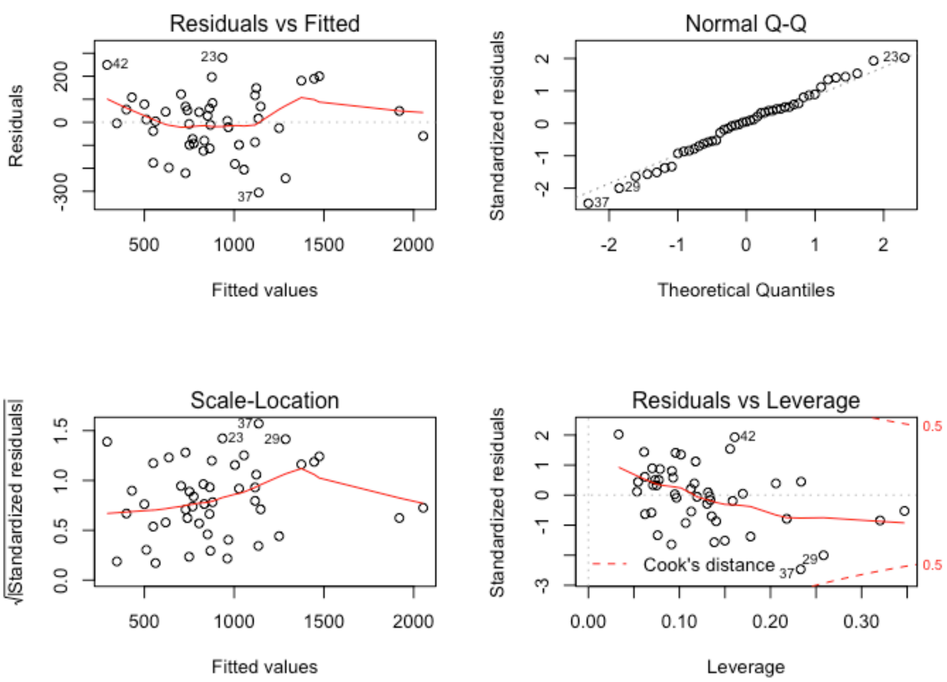
Coefficients:

	Estimate	Std. Error	t value	Pr(>|t|)
(Intercept)	905.085	20.610	43.916	< 2e-16 ***
PC1	75.891	8.352	9.087	2.25e-11 ***
PC2	-92.650	12.175	-7.610	2.30e-09 ***
PC3	40.535	14.662	2.765	0.0085 **
PC4	-212.374	17.420	-12.191	3.22e-15 ***
PC5	51.545	19.590	2.631	0.0119 *

Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 141.3 on 41 degrees of freedom
Multiple R-squared: 0.881,   Adjusted R-squared: 0.8665
F-statistic: 60.74 on 5 and 41 DF, p-value: < 2.2e-16

All five coefficient are significant and the resulting plot (residual vs fitted, etc) support this.



I then reverted back to the original equation as indicated in the code.

Code:
set.seed(42)
rm(list=ls())
options(scipen=4)

```
par(mfrow=c(1,1))

library(ggplot2) #basic plotting package
library(GGally) #advanced plotting tools
library(corrplot)
library(VIM)
library(factoextra)

crimedata <- read.table("/Users/marcthurig/Desktop/uscrimeSummer2018.txt", sep =
"",stringsAsFactors = FALSE, header = TRUE)
crimedata <- data.matrix(crimedata, rownames.force = NA )
crimedata <- data.frame(crimedata)

#prepare data
View(head(crimedata, 10))
aggr(crimedata)
str(crimedata)
summary(crimedata)
crimecor<-cor(crimedata)
corrplot(crimecor, method = "color")

#pca
pca = prcomp(crimedata, scale. = TRUE)
summary(pca)
names(pca)
means <-pca$center #means of variables
stdev <- pca$sdev #standard deviation of variables

biplot(pca, scale = 0) #plot the resultant principal components
VarianceExpl <- pca$sdev^2 #calculate variance
ProportionVariance <- round(((VarianceExpl^2) / sum(VarianceExpl^2)),2) #Proportion of
variance explained
ProportionVariance
plot(pca, type ="lines", col ="red", main = "Scree Plot") #scree plot
fviz_eig(pca, addlabels = TRUE, ylim = c(0, 50))
fviz_contrib(pca, choice = "var", axes = 1:2, top = 10)

#use first 5 PC's
finalpc <- pca$x[,1:5]
finalpc
Crime <- crimedata$Crime
Crime

crimedatanew <- data.frame(Crime, finalpc)
View(crimedatanew)
str(crimedatanew)

#regression
crimepca.lm <- lm(Crime ~ ., data = crimedatanew)
summary(crimepca.lm)

par(mfrow=c(2,2))
plot(crimepca.lm)

#unscale coefficients
```

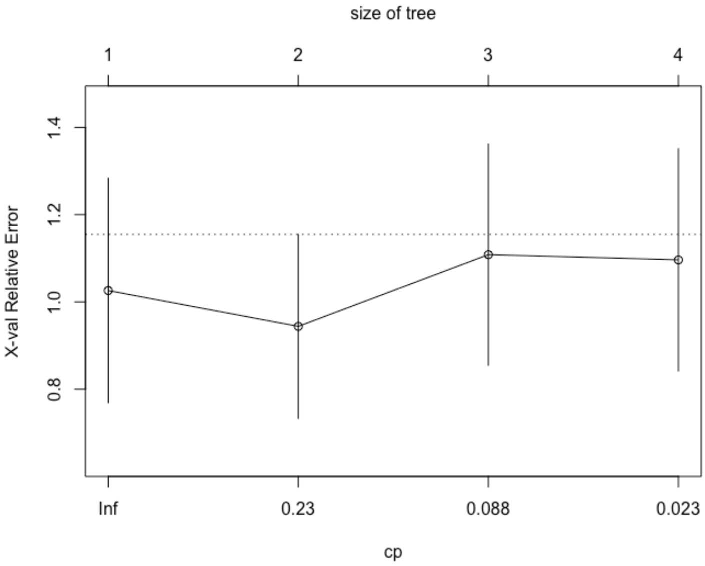
```
center<-pca$center
scaling<- pca$scale
rotation<- pca$rotation
intercept<- crimepca.lm$coefficients[1]
alphas<- crimepca.lm$coefficients[-1]

unscaledalphas <- alphas/sapply(crimedata[,1:15], sd)
uncaledintercept <- intercept - sum((alphas - sapply(
crimedata[,1:15],mean))/sapply(crimedata[,1:15],sd))
unscaledalphas
unscaledintercept

#predict new city
CrimePred <- 5703.843 +10*(36.23362633)+12.0*(-
71.46091294)+15.5*(18.43430614)+3200*(-0.09601941)+20.1*(10.16003882)
CrimePred
```

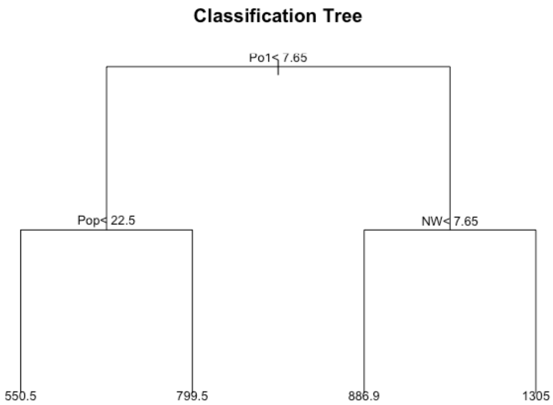
Question 10.1
Using the same crime data set uscrime.txt as in Questions 8.2 and 9.1, find the best model you can using
(a) a regression tree model, and
(b) a random forest model.
In R, you can use the tree package or the rpart package, and the randomForest package. For each model, describe one or two qualitative takeaways you get from analyzing the results (i.e., don't just stop when you have a good model, but interpret it too).

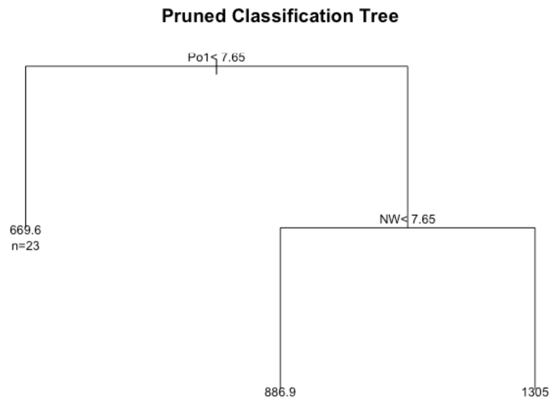
a. Based on the results, the x-val relative error is minimized when the size of tree value is 4 (CP value).

```
Regression tree:
rpart(formula = Crime ~ ., data = crimedata)

Variables actually used in tree construction:
[1] NW  Po1 Pop

Root node error: 6880928/47 = 146403

n= 47

        CP  nsplit rel error   xerror    xstd
1 0.362963      0   1.00000   1.02611   0.25727
2 0.148143      1   0.63704   0.94398   0.21078
3 0.051732      2   0.48889   1.10832   0.25348
```

4	0.010000	3	0.43716	1.09626	0.25466





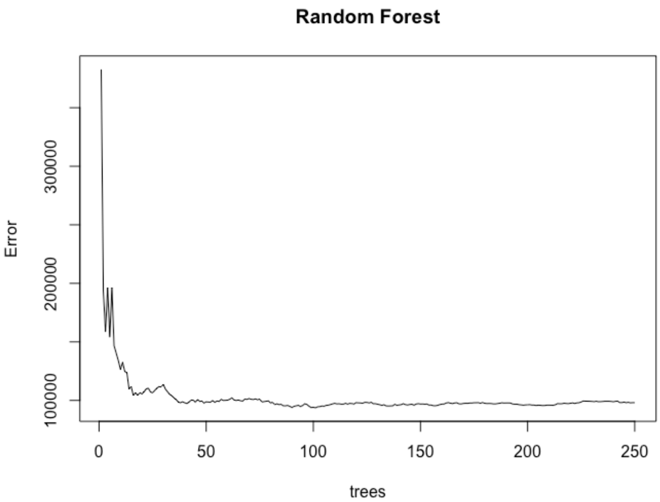


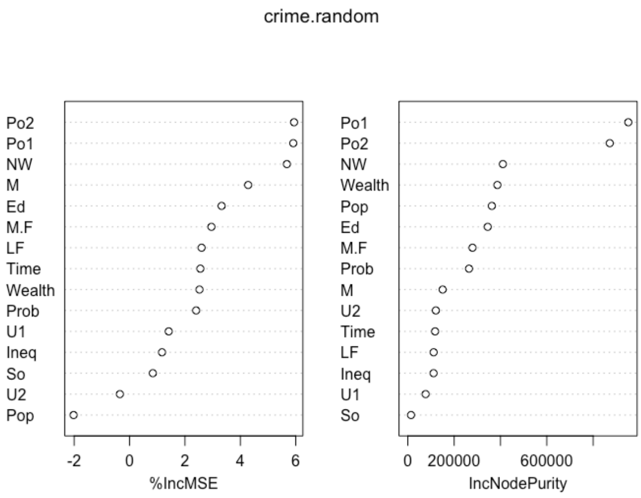

```
Code:
#regression tree
set.seed(42)
rm(list=ls())
options(scipen=4)
par(mfrow=c(1,1))

library(rpart)
```

```
crimedata <- read.table("/Users/marcthurig/Desktop/uscrimeSummer2018.txt", sep =
"\t",stringsAsFactors = FALSE, header = TRUE)
crime.tree = rpart(Crime ~ ., data=crimedata)
plotcp(crime.tree)
printcp(crime.tree)
summary(crime.tree)
plot(crime.tree, uniform = TRUE,main="Classification Tree")
text(crime.tree, use.n = TRUE, cex = 0.8)

crime.tree2 = prune(crime.tree, cp = 0.1)
summary(crime.tree2)
plot(crime.tree2, uniform = TRUE, main="Pruned Classification Tree")
text(crime.tree2, use.n = TRUE, cex = 0.8)
```

b. The test clearly shows that the first three elements (Po2, Po1 and NW) are relevant in creating the tree.

```
Call:
 randomForest(formula = Crime ~ ., data = crimedata, ntree = 250,      mtry = 5, importance
= TRUE, subset = train)
               Type of random forest: regression
                     Number of trees: 250
No. of variables tried at each split: 5

          Mean of squared residuals: 98073.42
                    % Var explained: 37.54
```






```
Code:
#random forest
set.seed(42)
rm(list=ls())
options(scipen=4)
par(mfrow=c(1,1))

library(randomForest)
```

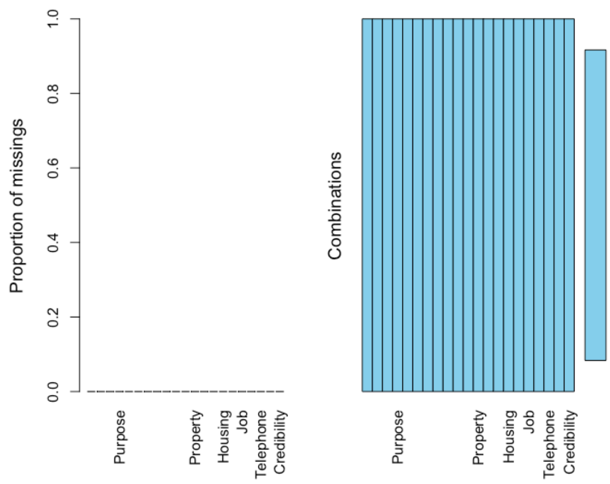
```
crimedata <- read.table("/Users/marcthurig/Desktop/uscrimeSummer2018.txt", sep =
"\t",stringsAsFactors = FALSE, header = TRUE)
dim(crimedata)
train = sample(1:nrow(crimedata), 0.7*nrow(crimedata))
test = crimedata[-train, "Crime"]

View(train)
crime.random = randomForest(Crime ~., data = crimedata, subset = train, ntree = 250,
mtry=5, importance = TRUE)
crime.random
plot(crime.random, main = "Random Forest")

importance(crime.random)
varImpPlot(crime.random)
```

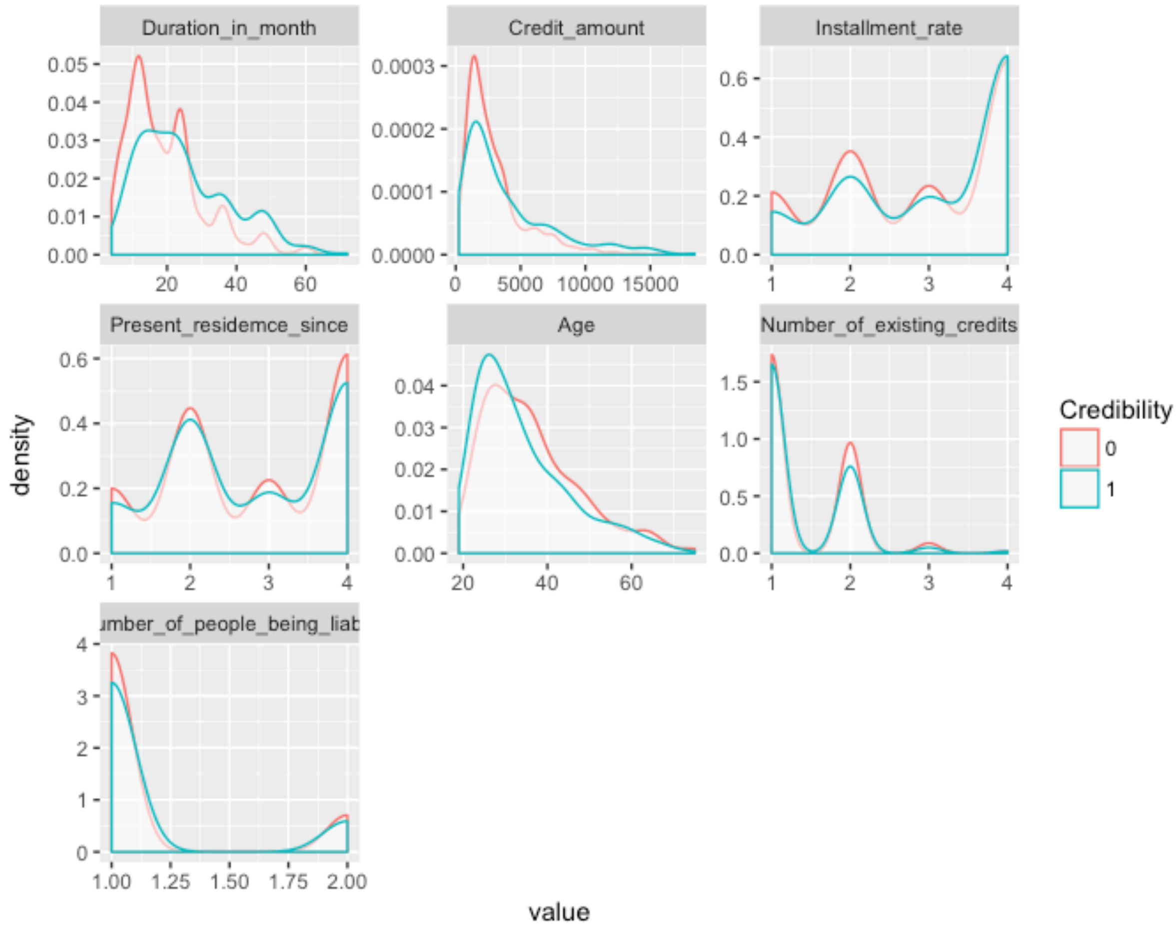
Question 10.2
Describe a situation or problem from your job, everyday life, current events, etc., for which a logistic regression model would be appropriate. List some (up to 5) predictors that you might use.

Election prediction: whether voters will vote for one or another party based on factors such as income, ethnicity, gender, age group, geographic location.

Question 10.3
1. Using the GermanCredit data set germancredit.txt from http://archive.ics.uci.edu/ml/machine-learning-databases/statlog/german / (description at http://archive.ics.uci.edu/ml/datasets/Statlog+%28German+Credit+Data%29), use logistic regression to find a good predictive model for whether credit applicants are good credit risks or not. Show your model (factors used and their coefficients), the software output, and the quality of fit. You can use the glm function in R. To get a logistic regression (logit) model on data where the response is either zero or one, use family=binomial(link="logit") in your glm function call.
2. Because the model gives a result between 0 and 1, it requires setting a threshold probability to separate between "good" and "bad" answers. In this data set, they estimate that incorrectly identifying a bad customer as good, is 5 times worse than incorrectly classifying a good customer as bad. Determine a good threshold probability based on your model.

After importing the data, I added headers to data and tested the data for completeness.





I then created a training and a testing data set using a 70% sample for the test data and 30% sample for the train data.

Using the train dataset, I ran the the glm function and got the following results. Based on their respective p-value, the factor Status of existing checking account (especially “no checking account” but as well “>= 200DM”), Duration, Credit history (paid fully), Purpose (radio/television), Status of saving account (especially “no account” but as well as “>=1000DM”), Installment rate, Other debtor (co-applicant), and number of existing credits.

```
 Call:
glm(formula = Credibility ~ ., family = binomial, data = germancredit[train,
   ])

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-2.3059  -0.6777  -0.2929   0.6530   2.5274

Coefficients:
                                      Estimate   Std. Error z value   Pr(>|z|)
(Intercept)                          -0.96185013   2.48559524  -0.387    0.69878
Status_of_existing_checking_accountA12   -0.84879134   0.45665906  -1.859    0.06307 .
Status_of_existing_checking_accountA13   -0.53196147   0.75245040  -0.707    0.47958
Status_of_existing_checking_accountA14   -2.08948882   0.46882164  -4.457 0.00000832
***
Duration_in_month                     0.01010532   0.01885826   0.536    0.59206
Credit_historyA31                     0.55035674   1.09276404   0.504    0.61452
```

Credit_historyA32	-0.41704061	0.85414892	-0.488	0.62537	
Credit_historyA33	-1.19279867	0.95782578	-1.245	0.21301	
Credit_historyA34	-2.10485432	0.92530747	-2.275	0.02292	*
PurposeA41	-1.81635662	0.75273230	-2.413	0.01582	*
PurposeA410	-2.78429674	1.71297850	-1.625	0.10407	
PurposeA42	-0.58644426	0.56402112	-1.040	0.29845	
PurposeA43	-0.75806088	0.48963755	-1.548	0.12157	
PurposeA44	-0.59773167	1.48014320	-0.404	0.68634	
PurposeA45	1.75469230	1.04586610	1.678	0.09340	.
PurposeA46	0.91619339	0.78167805	1.172	0.24116	
PurposeA48	-14.62496113	882.74387681	-0.017	0.98678	
PurposeA49	-0.32190386	0.74076964	-0.435	0.66389	
Credit_amount	0.00022198	0.00009076	2.446	0.01446	*
Savings_accountA62	-0.19604312	0.58111447	-0.337	0.73585	
Savings_accountA63	0.60501970	0.75390541	0.803	0.42226	
Savings_accountA64	-0.46477083	0.81340916	-0.571	0.56774	
Savings_accountA65	-0.18387823	0.51808497	-0.355	0.72265	
Present_employment_sinceA72	0.48145326	0.84276681	0.571	0.56781	
Present_employment_sinceA73	0.33463164	0.79893943	0.419	0.67533	
Present_employment_sinceA74	-0.51358315	0.85010537	-0.604	0.54575	
Present_employment_sinceA75	0.85178747	0.80659004	1.056	0.29095	
Installment_rate	0.32883385	0.18796101	1.749	0.08021	.
Status_sexA92	1.99419957	0.97192006	2.052	0.04019	*
Status_sexA93	0.83128468	0.96179574	0.864	0.38742	
Status_sexA94	1.39336250	1.06683424	1.306	0.19153	
Other_debtorA102	0.21428667	0.78787784	0.272	0.78564	
Other_debtorA103	-0.84533418	0.82990685	-1.019	0.30840	
Present_residemce_since	-0.06675410	0.18100493	-0.369	0.71228	
PropertyA122	0.02774259	0.52567376	0.053	0.95791	
PropertyA123	0.67055178	0.48294280	1.388	0.16499	
PropertyA124	-0.07426684	0.96364943	-0.077	0.93857	
Age	-0.05245565	0.01867086	-2.809	0.00496	**
Other_installmentsA142	0.56152385	0.97028852	0.579	0.56278	
Other_installmentsA143	-1.22149374	0.47118138	-2.592	0.00953	**
HousingA152	0.08714212	0.52500114	0.166	0.86817	
HousingA153	0.57341260	1.01889747	0.563	0.57359	
Number_of_existing_credits	0.70441685	0.37362210	1.885	0.05938	.
JobA172	0.37749057	1.56354500	0.241	0.80922	
JobA173	0.57778456	1.53027128	0.378	0.70575	
JobA174	0.98617846	1.53420582	0.643	0.52036	
Number_of_people_being_liable	0.18064076	0.44347212	0.407	0.68376	
TelephoneA192	-0.51051745	0.40772859	-1.252	0.21053	
Foreign_workerA202	-1.96701284	1.47564613	-1.333	0.18254	

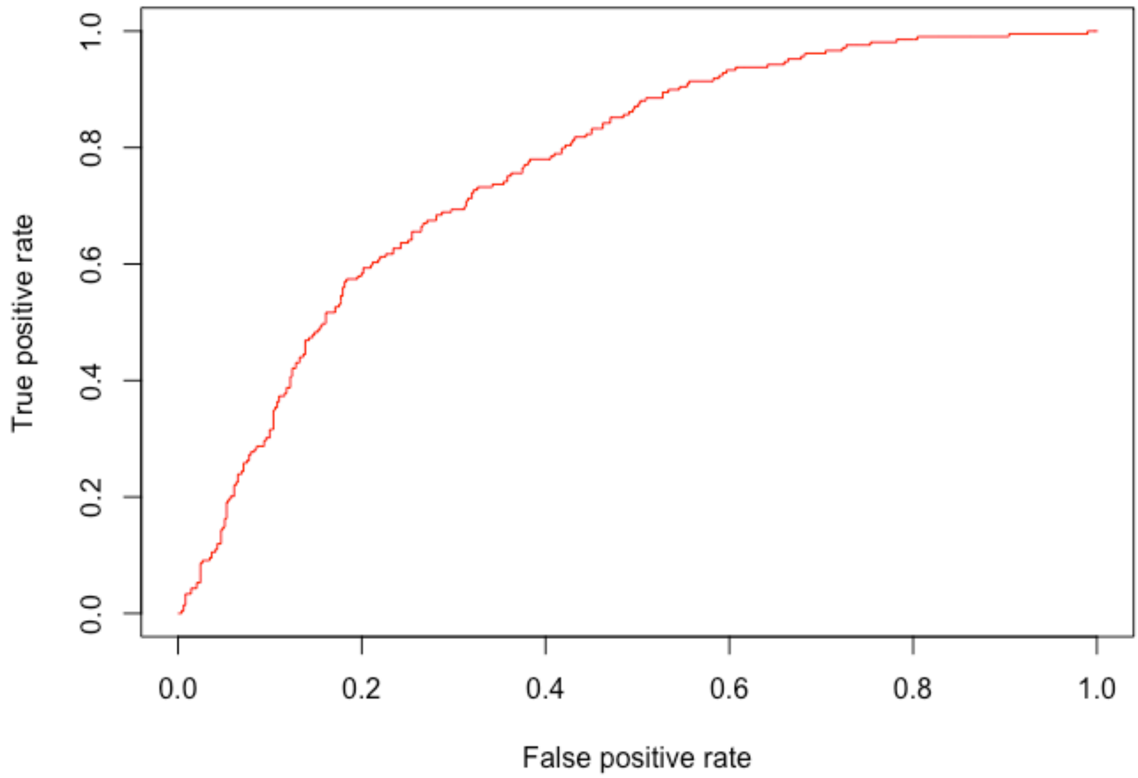
Signif. codes: 0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for binomial family taken to be 1)

Null deviance: 368.20 on 299 degrees of freedom
Residual deviance: 252.67 on 251 degrees of freedom
AIC: 350.67

Number of Fisher Scoring iterations: 13

I them used the train data and ran the regression against as well as calculated the AUC (76.5%).



Code

```
#GermanCredit
set.seed(42)
rm(list=ls())
options(scipen=4)
par(mfrow=c(1,1))

library(VIM)
library(ROCR)

germancredit <- read.table("/Users/marcthurig/Desktop/germandata.txt", sep =
"",stringsAsFactors = FALSE, header = FALSE)
names(germancredit) = c("Status_of_existing_checking_account", "Duration_in_month",
"Credit_history", "Purpose","Credit_amount", "Savings_account",
"Present_employment_since", "Installment_rate", "Status_sex",
"Other_debtor","Present_residemce_since", "Property", "Age", "Other_installments",
"Housing", "Number_of_existing_credits","Job", "Number_of_people_being_liable",
"Telephone", "Foreign_worker", "Credibility")

germancredit$Credibility = germancredit$Credibility - 1
germancredit$Credibility
germancredit$Credibility <- as.factor(germancredit$Credibility)
```

```
dim(germancredit)
str(germancredit)
aggr(germancredit)
summary(germancredit)
View(head(germancredit, 10))
ggplot(data = melt(germancredit), aes(x = value, color= Credibility)) +
geom_density(fill="white",alpha=0.55) + facet_wrap(~variable, scales = "free")

test <- sample(1:nrow(germancredit), 0.7*nrow(germancredit))
train <- (1:nrow(germancredit))[-test]

credit.logit <- glm(Credibility ~., family = binomial, data = germancredit[train, ])
summary(credit.logit)

fitcredit <- predict(credit.logit, type = 'response', newdata = germancredit[test, ])
summary(fitcredit)

pred <- prediction(fitcredit, germancredit$Credibility[test])
perf <- performance(pred, "tpr", "fpr") #True Positive, False Positive
plot(perf, col="red")

AUCCredit <- performance(pred, measure = "auc")@y.values[[1]]
AUCCredit
```