R Zone 2: Chapter 3, 4 and 7

Brandon Hufstetler, Garrett Alarcon, Jinan Andrews, Anson Cheng, Nick Forrest, and Nestor Herandez

8 July 2019

Chapter 3

READ IN THE CHURN DATA SET

Read in the curn variable in the Churn data set, look at the first ten records, see the number of true and false churns and the proportion of churners. Notice that the format of the Churn variable contains a period.

```
setwd("~/IMGT680")
churn <- read.csv(file = "churn.txt", stringsAsFactors=TRUE)

# Show the first ten records
churn[1:10,]
```

	State	Account.Length	Area.Code	Phone	Int.l.Plan	VMail.Plan	VMail.Message	Day.Mins
	<fctr>	<int>	<int>	<fctr>	<fctr>	<fctr>	<int>	<dbl>
1	KS	128	415	382-4657	no	yes	25	265.1
2	OH	107	415	371-7191	no	yes	26	161.6
3	NJ	137	415	358-1921	no	no	0	243.4
4	OH	84	408	375-9999	yes	no	0	299.4
5	OK	75	415	330-6626	yes	no	0	166.7
6	AL	118	510	391-8027	yes	no	0	223.4
7	MA	121	510	355-9993	no	yes	24	218.2
8	MO	147	415	329-9001	yes	no	0	157.0
9	LA	117	408	335-4719	no	no	0	184.5
10	WV	141	415	330-8173	yes	yes	37	258.6

1-10 of 10 rows | 1-10 of 22 columns

```
# Summarize the Churn variable
sum.churn <- summary(churn$Churn)
sum.churn
```

```
## False.  True.
##   2850    483
```

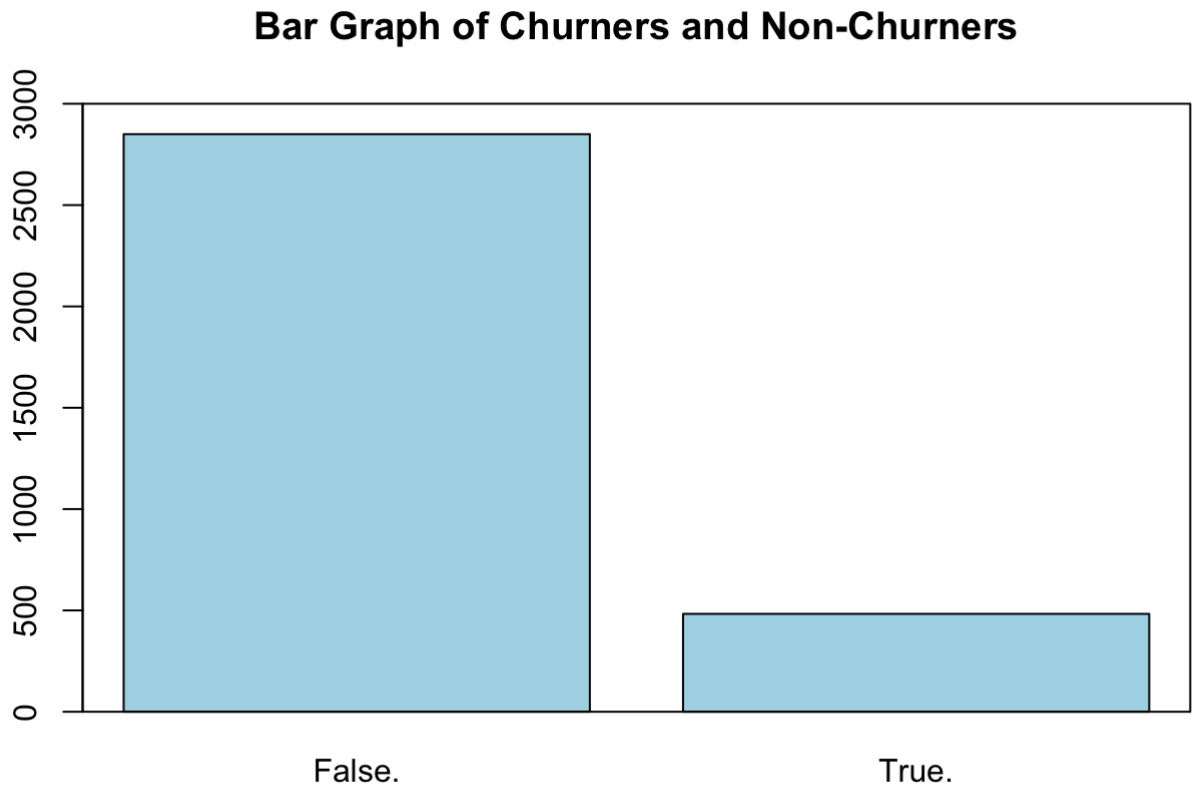
```
  # Calculate proportion of churners
  prop.churn <- sum(churn$Churn == "True.") / length(churn$Churn)
  prop.churn
```

```
## [1] 0.1449145
```

BAR CHART OF VARIABLE CHURN

Bar graph of churners and non-churners show that close to 3000 people did not churn and around 500 people did.

```
  barplot(sum.churn, ylim = c(0, 3000),
          main = "Bar Graph of Churners and Non-Churners",
          col = "lightblue")
  box(which = "plot", lty = "solid", col="black")
```



MAKE A TABLE FOR COUNTS OF CHURN AND INTERNATIONAL PLAN

Now a table is created to look at the number of people that are and are not on an international plan and whether or not they churned. We see 2664 people are not on an international plan and do not churn but 346 do churn. For those with an international plan, 186 do not churn and 137 do.

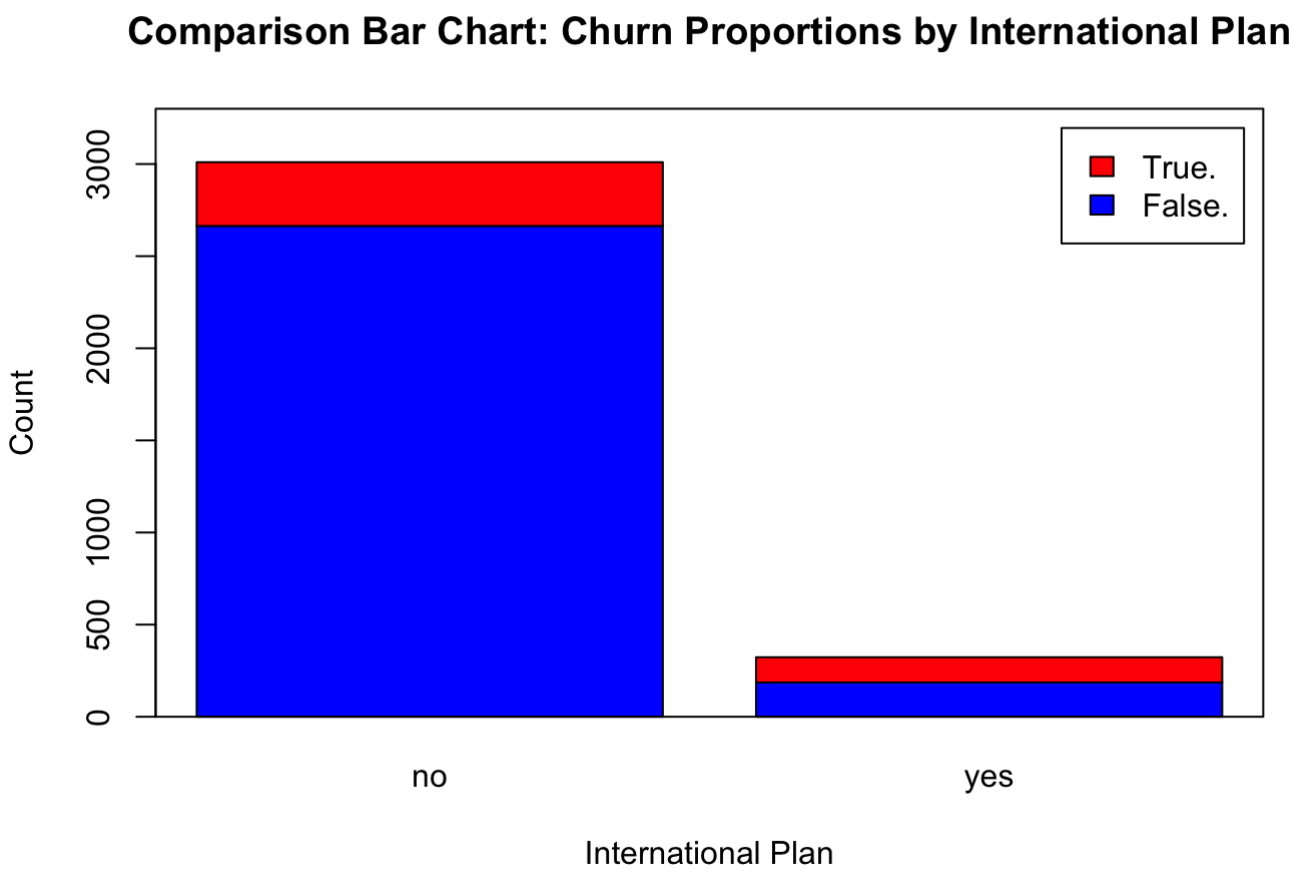
```
counts <- table(churn$Churn, churn$Int.l.Plan, dnn=c("Churn", "International Plan"))
counts
```

```
##         International Plan
## Churn      no  yes
##   False. 2664  186
##   True.   346  137
```

OVERLAYED BAR CHART

Overlayed bar chart allows us to visually see proportion of churners and non-churners by international plan

```
barplot(counts, legend = rownames(counts), col = c("blue", "red"),
        ylim = c(0, 3300), ylab = "Count", xlab = "International Plan",
        main = "Comparison Bar Chart: Churn Proportions by International Plan")
box(which = "plot", lty = "solid", col="black")
```



CREATE A TABLE WITH SUMS FOR BOTH VARIABLES

Sumtable tells us that 3010 are not on the international plan, 323 are on the international plan, 2850 people do not churn overall and 483 people churn overall

```
  sumtable <- addmargins(counts, FUN = sum)
```

```
## Margins computed over dimensions
## in the following order:
## 1: Churn
## 2: International Plan
```

```
  sumtable
```

```
##         International Plan
## Churn      no  yes  sum
##   False. 2664  186 2850
##   True.   346  137  483
##   sum    3010  323 3333
```

CREATE A TABLE OF PROPORTIONS OVER ROWS

Creating proportions over rows, we see that 93.47% of non-churners are not on the international plan compared to the 6.53% of non-churners on the international plan. For churners, 71.64% were not on the international plan compared to the 28.36% on the international plan

```
  row.margin <- round(prop.table(counts, margin = 1), 4)*100
  row.margin
```

```
##         International Plan
## Churn       no   yes
##   False. 93.47  6.53
##   True.  71.64 28.36
```

CREATE A TABLE OF PROPORTIONS OVER COLUMNS

For proportions by columns, we see that 88.50% of those not on the international plan do not churn compared to the 11.50% not on the international plan that churn. 57.59% of those on the international plan do not churn and 42.41% on the international plan churn.

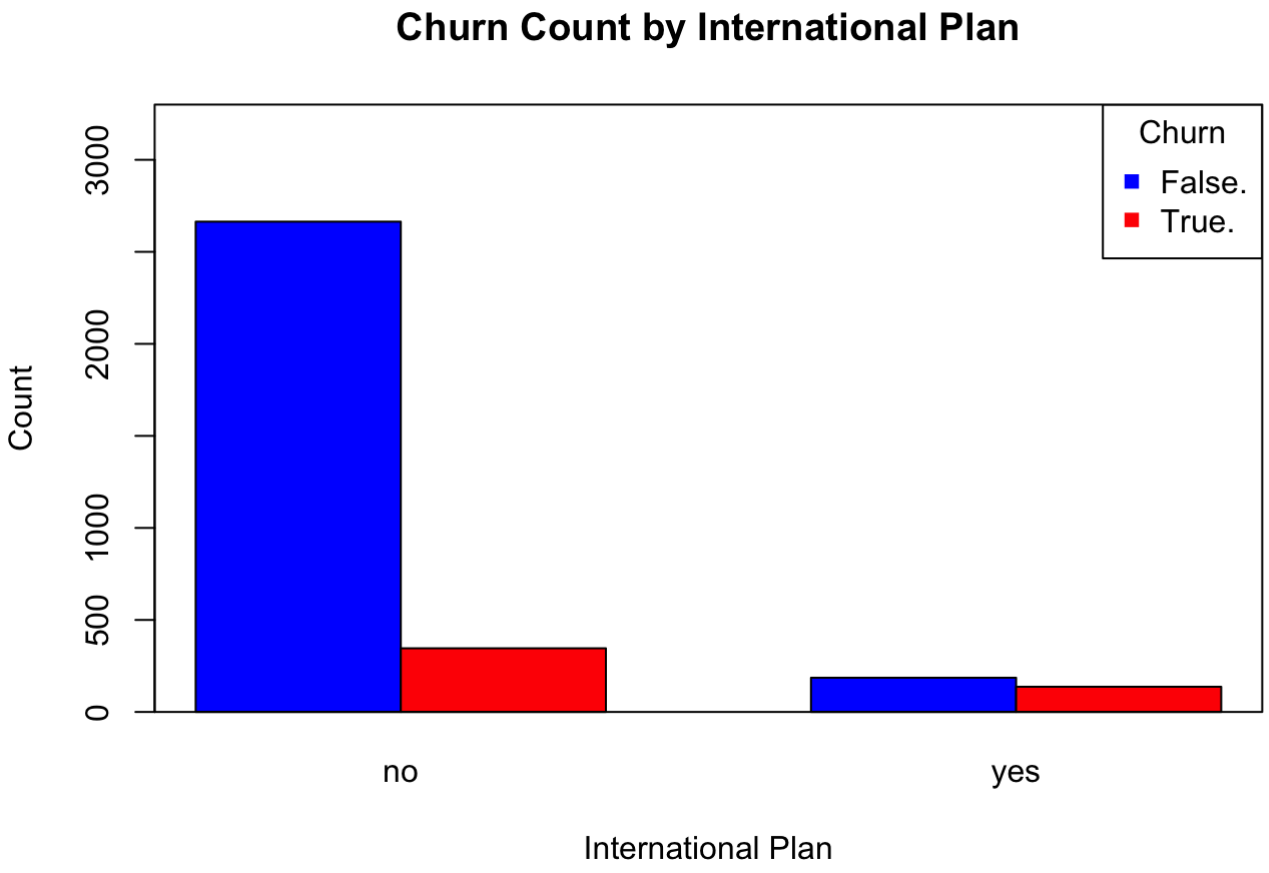
```
  col.margin <- round(prop.table(counts, margin = 2), 4)*100
  col.margin
```

```
##         International Plan
## Churn       no   yes
##   False. 88.50 57.59
##   True.  11.50 42.41
```

CLUSTERED BAR CHART, WITH LEGEND

Instead of an overlayed barchart, we break up the proportions of those who do and those who don't churn by international plan

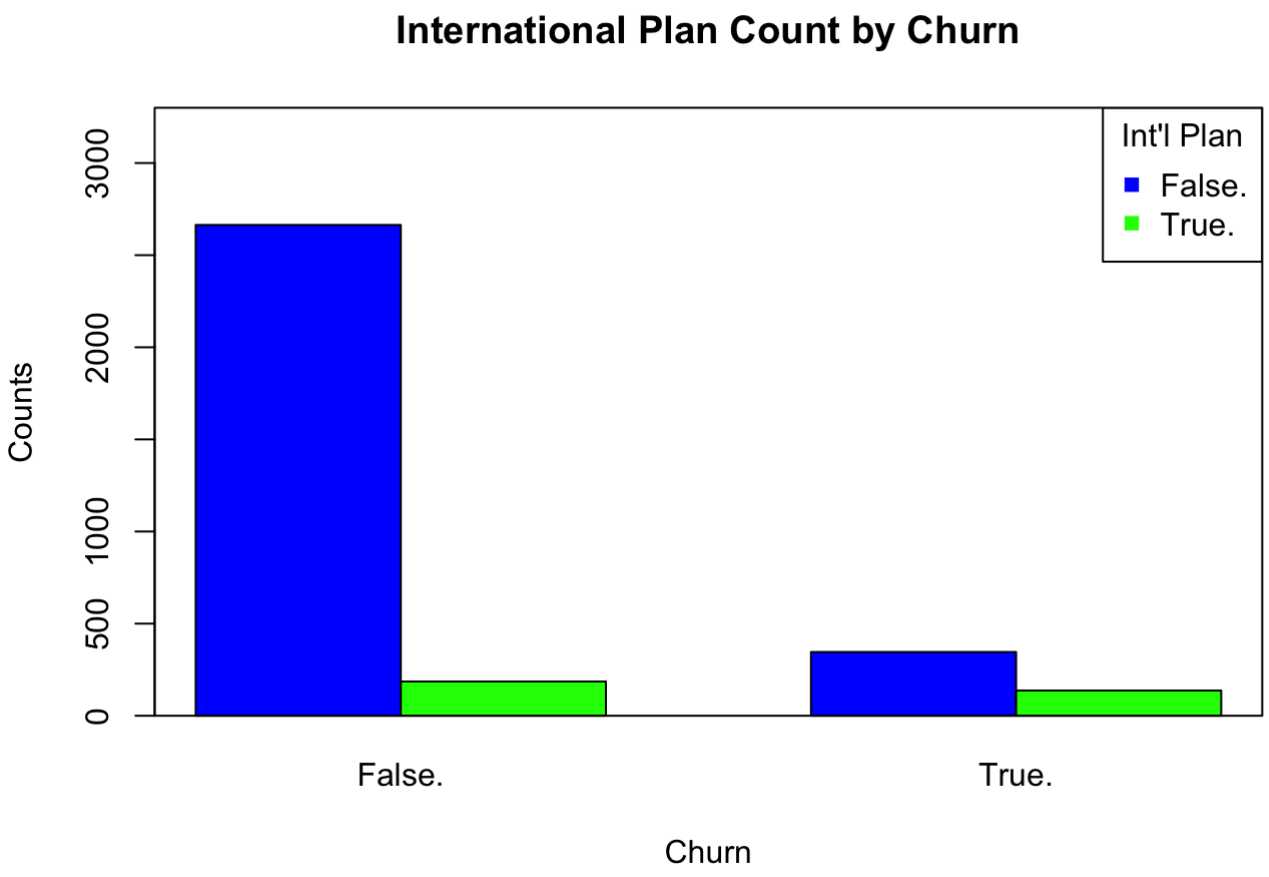
```
barplot(counts, col = c("blue", "red"), ylim = c(0, 3300),
        ylab = "Count", xlab = "International Plan",
        main = "Churn Count by International Plan", beside = TRUE)
legend("topright", c(rownames(counts)), col = c("blue", "red"),
       pch = 15, title = "Churn")
box(which = "plot", lty = "solid", col="black")
```



CLUSTERED BAR CHART OF CHURN AND INTERNATIONAL PLAN WITH LEGEND

Now we look at international plan count by whether customers churned or not

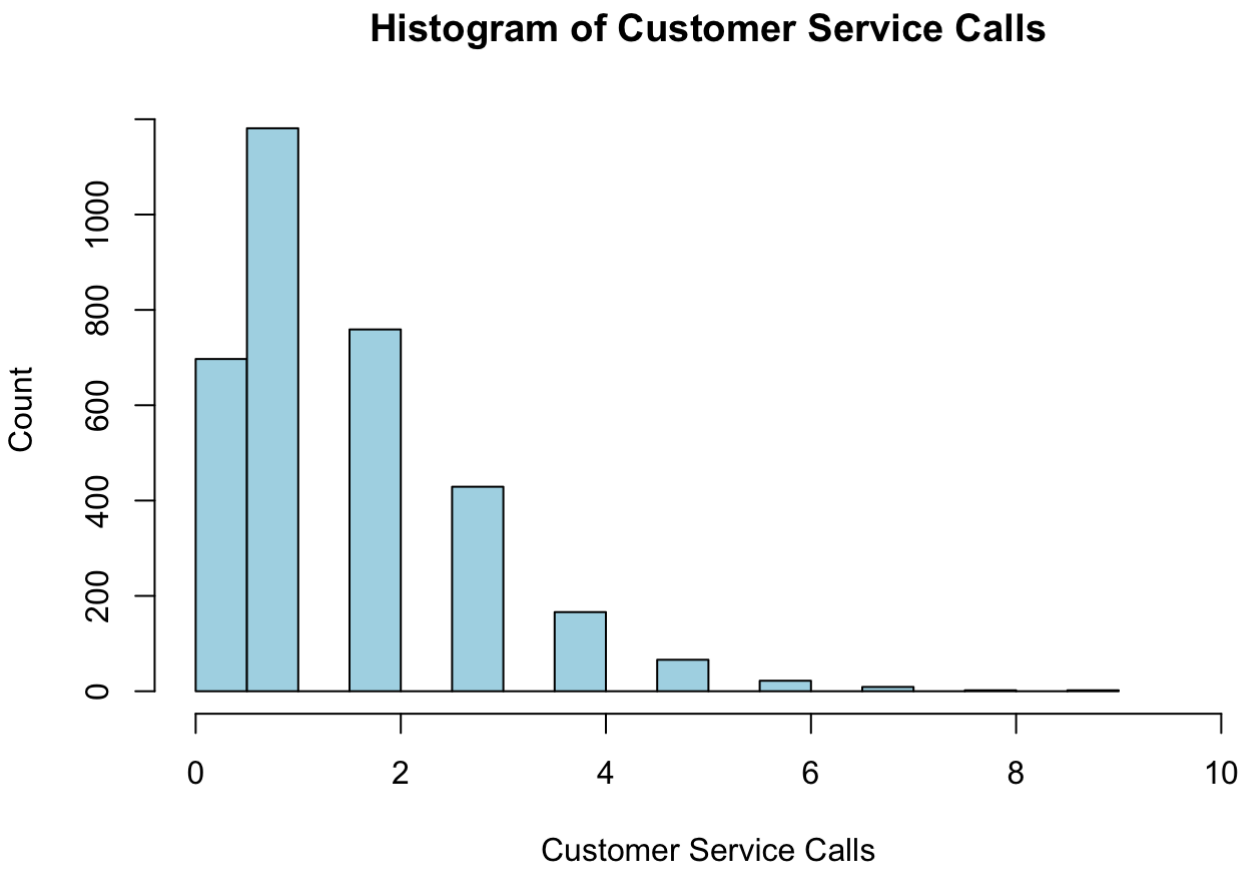
```
barplot(t(counts), col = c("blue", "green"), ylim = c(0, 3300),
        ylab = "Counts", xlab = "Churn",
        main = "International Plan Count by Churn", beside = TRUE)
legend("topright", c(rownames(counts)), col = c("blue", "green"),
       pch = 15, title = "Int'l Plan")
box(which = "plot", lty = "solid", col="black")
```



HISTOGRAM OF NON-OVERLAYED CUSTOMER SERVICE CALLS

This output creates a histogram of customer service calls, counting number of occurences, up to 10 customer service calls

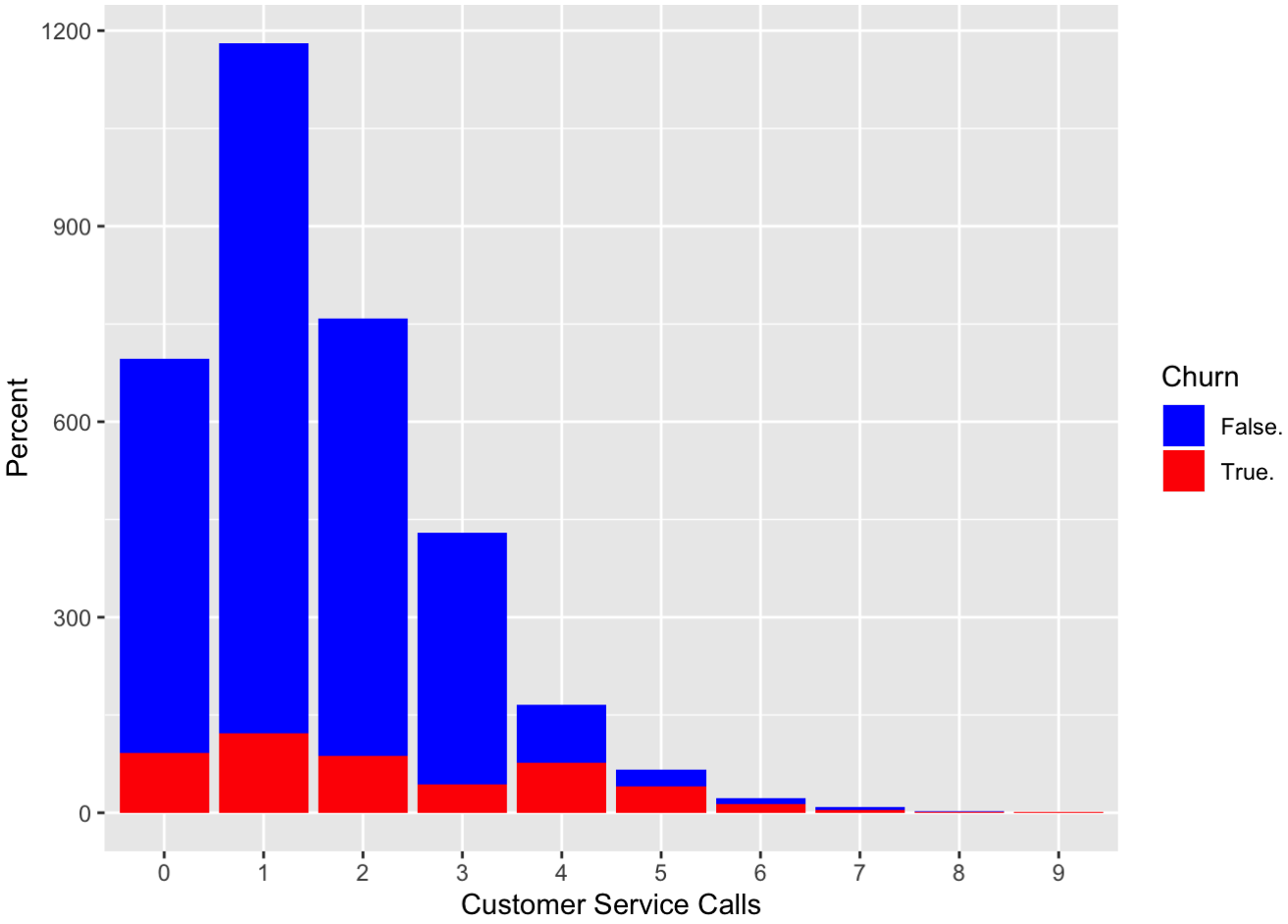
```
hist(churn$CustServ.Calls, xlim = c(0,10),
     col = "lightblue", ylab = "Count", xlab = "Customer Service Calls",
     main = "Histogram of Customer Service Calls")
```



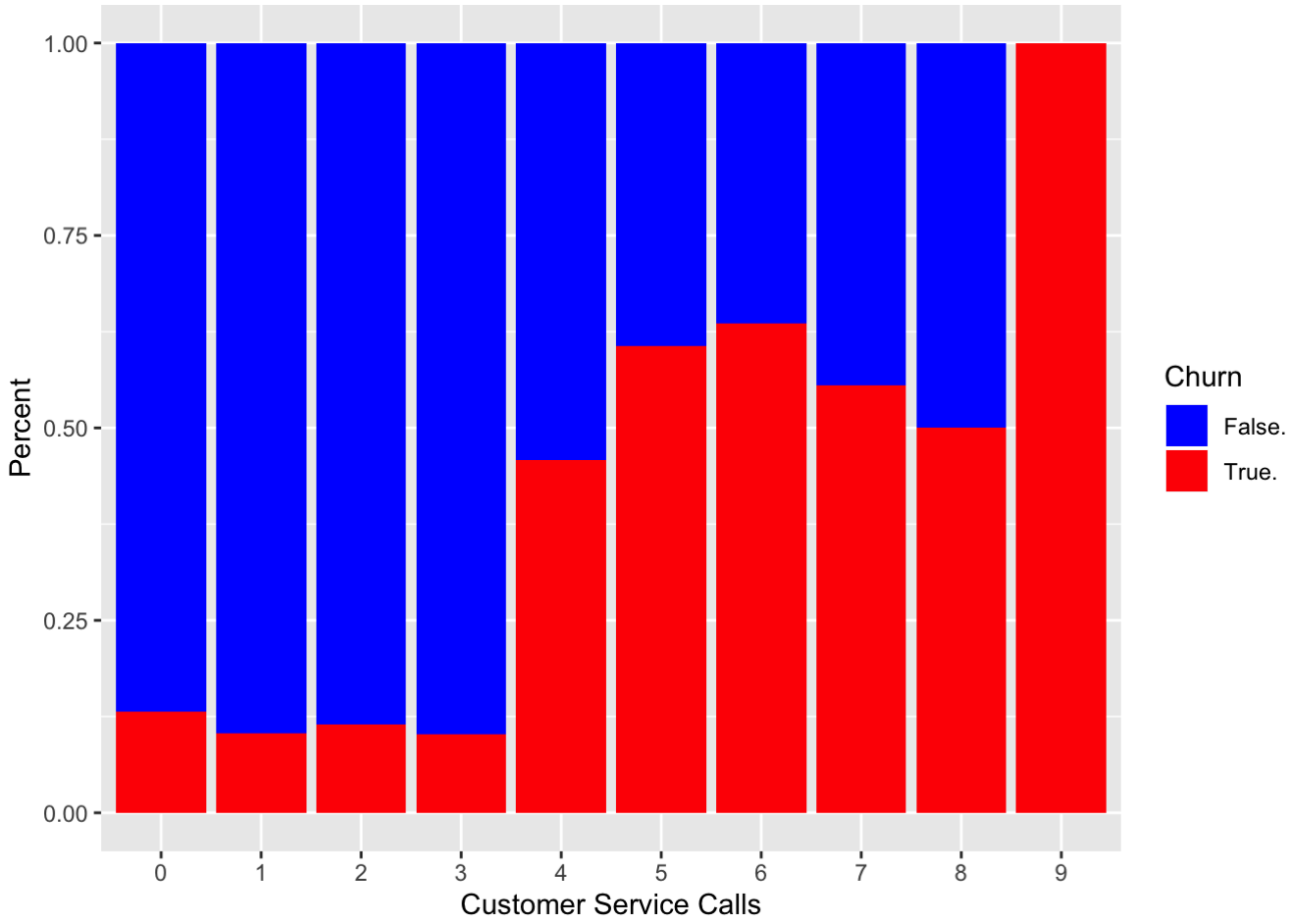
DOWNLOAD AND INSTALL THE R PACKAGE GGPLOT2

Two overlayed bar graphs. Seems like a type in the first graph as the y axis isn't percent but rather number of occurences of each number of customer service calls. The bar is split to visually see proportion of churners and non churners. The second graph changes the y axis to a percentage so instead of graphing number of occurences, the percentage of churners and non churners in each number of customer service calls can be seen.

```
#install.packages("ggplot2")
# Pick any CRAN mirror
# (see example image)
# Open the new package
library(ggplot2)

# OVERLAYED BAR CHARTS
ggplot() +
  geom_bar(data = churn,
           aes(x = factor(churn$CustServ.Calls),
               fill = factor(churn$Churn)),
           position = "stack") +
  scale_x_discrete("Customer Service Calls") +
  scale_y_continuous("Percent") +
  guides(fill=guide_legend(title="Churn")) +
  scale_fill_manual(values=c("blue", "red"))
```




```
ggplot() +
  geom_bar(data=churn,
           aes(x = factor(churn$CustServ.Calls),
               fill = factor(churn$Churn)),
           position = "fill") +
  scale_x_discrete("Customer Service Calls") +
  scale_y_continuous("Percent") +
  guides(fill=guide_legend(title="Churn")) +
  scale_fill_manual(values=c("blue", "red"))
```



TWO-SAMPLE T-TEST FOR INT'L CALLS

Two sample t-test performed for the difference in mean number of international calls for churners and non-churners is seen to be statistically significant.The p-value of 0.003186 tells us so. The confidence interval does not contain 0 either. That means the variable, international calls, is useful for predicting churn.

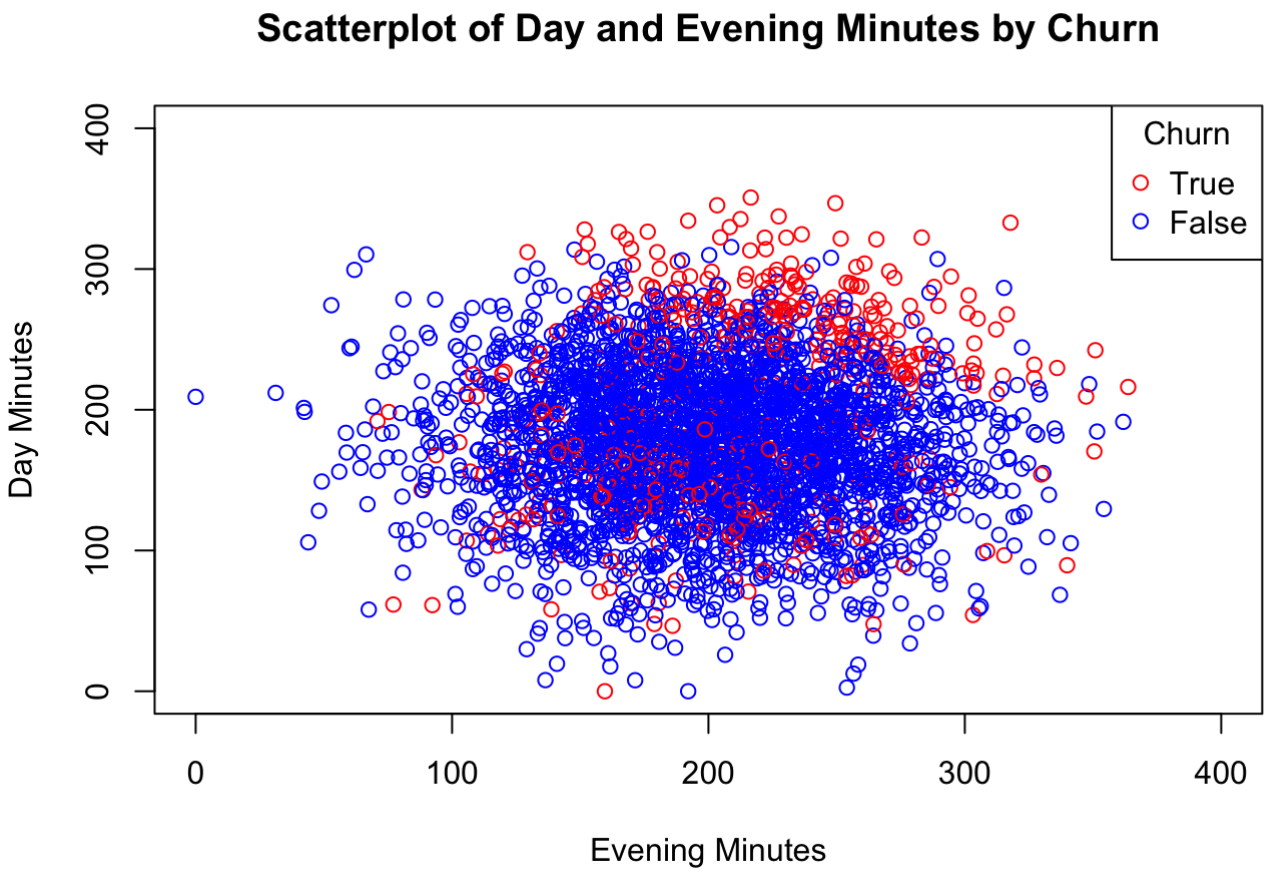
```
# Partition data
churn.false <- subset(churn, churn$Churn == "False.")
churn.true <- subset(churn, churn$Churn == "True.")

# Run the test
t.test(churn.false$Intl.Calls, churn.true$Intl.Calls)
```

```
##
##  Welch Two Sample t-test
##
## data:  churn.false$Intl.Calls and churn.true$Intl.Calls
## t = 2.9604, df = 640.64, p-value = 0.003186
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  0.1243807 0.6144620
## sample estimates:
## mean of x mean of y
##  4.532982  4.163561
```

SCATTERPLOT OF EVENING MINUTES AND DAY MINUTES, COLORED BY CHURN

Looking at the scatterplot of day minutes vs evening minutes, the univariate evidence for a high churn rate for high evening minutes is difficult to determine. It appears that customers with high day and evening minutes are more likely to churn.

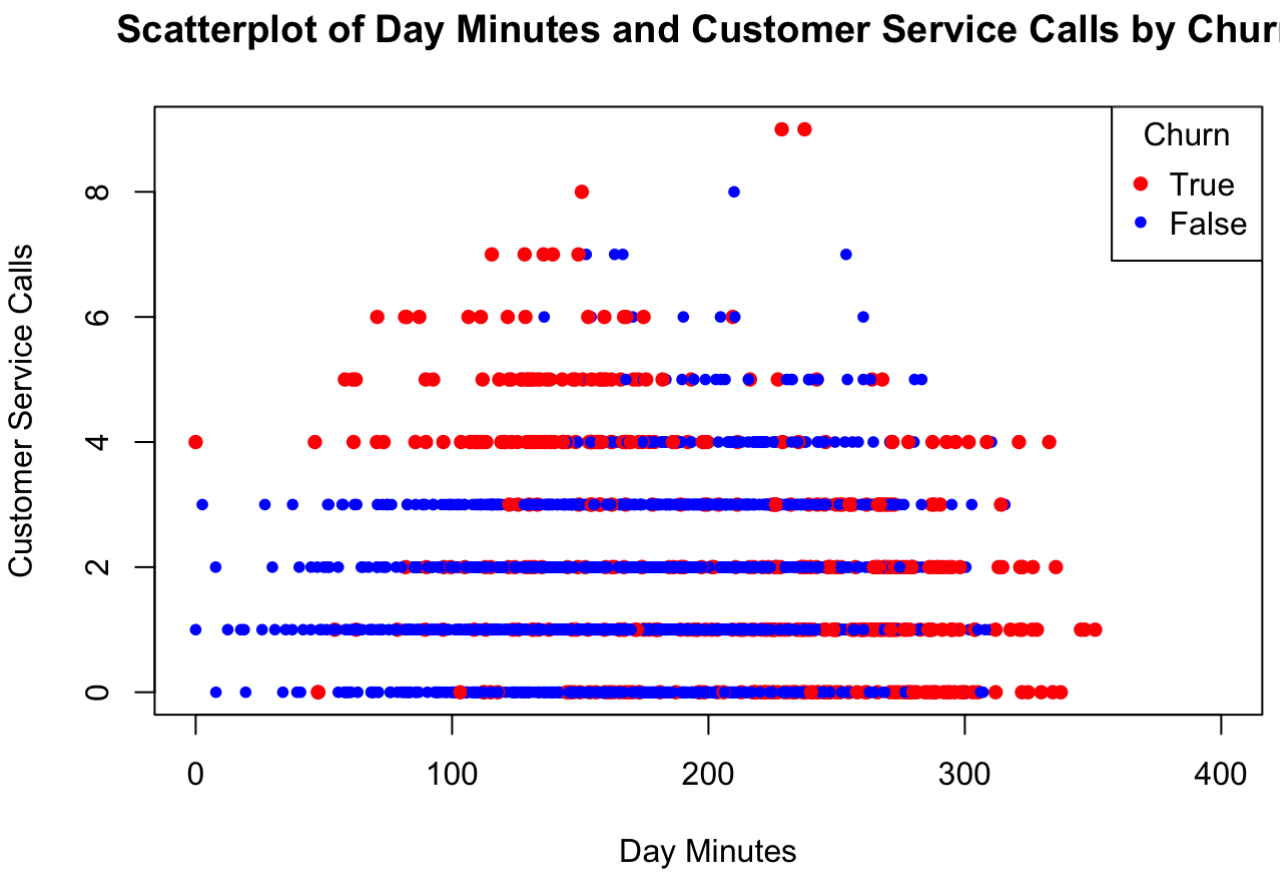
```
plot(churn$Eve.Mins, churn$Day.Mins,
     xlim = c(0, 400), ylim = c(0, 400),
     xlab = "Evening Minutes", ylab = "Day Minutes",
     main = "Scatterplot of Day and Evening Minutes by Churn",
     col = ifelse(churn$Churn== "True.", "red", "blue"))
legend("topright", c("True", "False"),
       col = c("red", "blue"), pch = 1, title = "Churn")
```



SCATTERPLOT OF DAY MINUTES AND CUSTOMER SERVICE CALLS, COLORED BY CHURN

In the graph of customer service calls versus day minutes, there appears to be an interaction between the variables as there is a pattern in the graph.

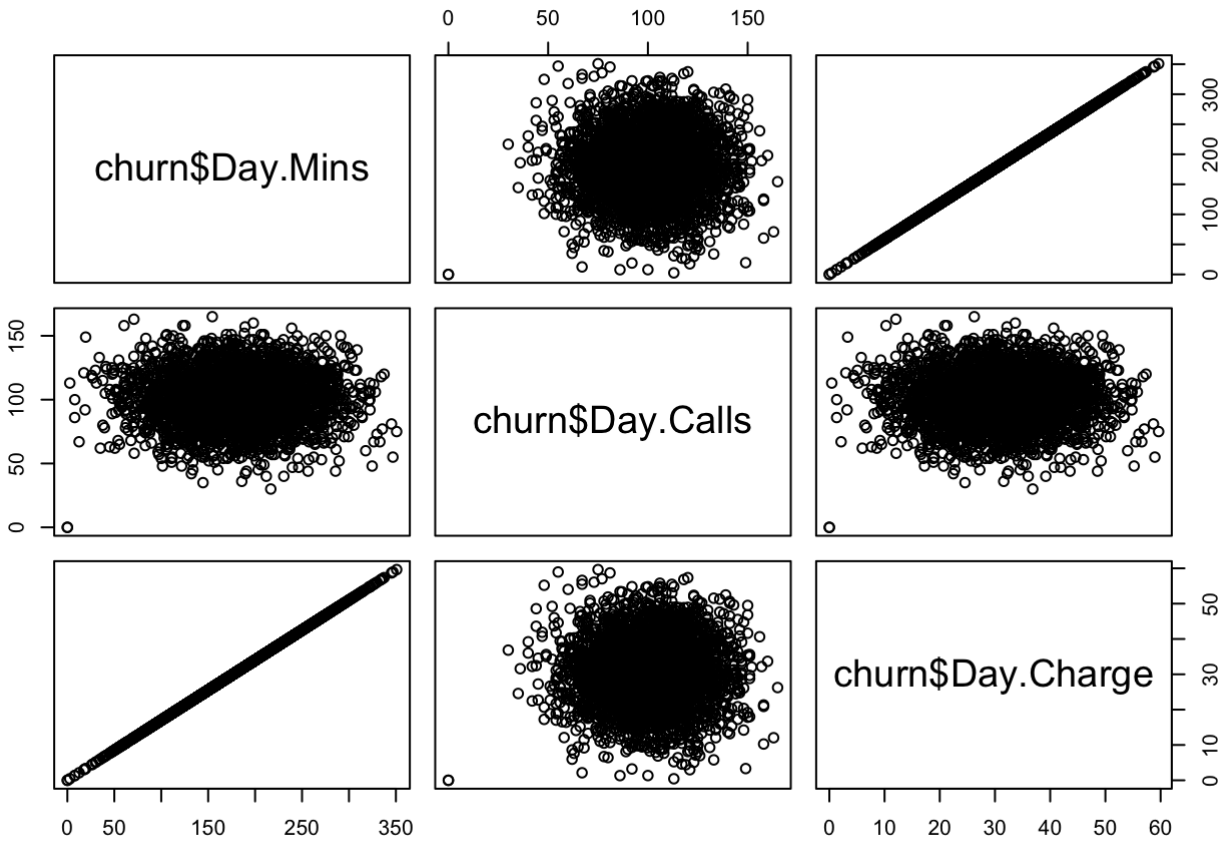
```
plot(churn$Day.Mins, churn$CustServ.Calls, xlim = c(0, 400),
     xlab = "Day Minutes", ylab = "Customer Service Calls",
     main = "Scatterplot of Day Minutes and Customer Service Calls by Churn",
     col = ifelse(churn$Churn=="True.", "red", "blue"),
     pch = ifelse(churn$Churn=="True.", 16, 20))
legend("topright", c("True", "False"),
       col = c("red", "blue"), pch = c(16, 20), title = "Churn")
```



SCATTERPLOT MATRIX

We want to uncover possible correlation among the predictors. There does not seem to be any relationship between day mins and day calls, or between day calls and day charge. The scatterplots do not have a pattern to them.

```
pairs(~churn$Day.Mins + churn$Day.Calls + churn$Day.Charge)
```



REGRESSION OF DAY CHARGE VS DAY MINUTES

The graphical results do not support a general sanity check that there should be a correlation.Results tell us the day charge quals 0.0006134 + 0.17 times day minutes. The R squared and adjusted R squared are both 1 so the regression analysis tells us that there is a perfect linear relationship. This also tells us that we should eliminate one of these two variables from our analysis as they are perfectly correlated.

```
fit <- lm(churn$Day.Charge ~ churn$Day.Mins)
summary(fit)
```

```
##
## Call:
## lm(formula = churn$Day.Charge ~ churn$Day.Mins)
##
## Residuals:
##        Min         1Q     Median         3Q        Max
## -0.0045935 -0.0025391  0.0004326  0.0024587  0.0045224
##
## Coefficients:
##                  Estimate Std. Error   t value Pr(>|t|)
## (Intercept)    6.134e-04  1.711e-04 3.585e+00 0.000341 ***
## churn$Day.Mins 1.700e-01  9.108e-07 1.866e+05  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.002864 on 3331 degrees of freedom
## Multiple R-squared:      1,  Adjusted R-squared:      1
## F-statistic: 3.484e+10 on 1 and 3331 DF,  p-value: < 2.2e-16
```

CORRELATION VALUES, WITH P-VALUES

Correlation matrix between day mins, day calls and day charge is made. Of the three p values outputted, Mins ChargeTest has a small p-value of 0 and a correlation coefficient of 1, telling us that day mins and day charge are positively correlated.

```
  days <- cbind(churn$Day.Mins, churn$Day.Calls, churn$Day.Charge)
  MinsCallsTest <- cor.test(churn$Day.Mins, churn$Day.Calls)
  MinsChargeTest <- cor.test(churn$Day.Mins, churn$Day.Charge)
  CallsChargeTest <- cor.test(churn$Day.Calls, churn$Day.Charge)
  round(cor(days), 4)
```

```
##        [,1]   [,2]   [,3]
## [1,] 1.0000 0.0068 1.0000
## [2,] 0.0068 1.0000 0.0068
## [3,] 1.0000 0.0068 1.0000
```

```
  MinsCallsTest$p.value
```

```
## [1] 0.6968515
```

```
  MinsChargeTest$p.value
```

```
## [1] 0
```

```
  CallsChargeTest$p.value
```

```
## [1] 0.6967428
```

CORRELATION VALUES AND P-VALUES IN MATRIX FORM

Correlation matrix done again. Ignoring the diagonals in the p-value matrix, The only low value (below 0.05) is 0.0264, which is the correlation between account length and day calls. The correlation is 0.0385 telling us that account length and day calls are positively correlated.

```
  # Collect variables of interest
  corrdata <-
    cbind(churn$Account.Length, churn$VMail.Message, churn$Day.Mins, churn$Day.Calls, ch
urn$CustServ.Calls)

  # Declare the matrix
  corrpvalues <- matrix(rep(0, 25), ncol = 5)

  # Fill the matrix with correlations
  for (i in 1:4) {
    for (j in (i+1):5) {
      corrpvalues[i,j] <-
        corrpvalues[j,i] <-
        round(cor.test(corrdata[,i],
                       corrdata[,j])$p.value,
              4)
    }
  }
  round(cor(corrdata), 4)
```

```
##         [,1]    [,2]    [,3]    [,4]    [,5]
## [1,]  1.0000 -0.0046  0.0062  0.0385 -0.0038
## [2,] -0.0046  1.0000  0.0008 -0.0095 -0.0133
## [3,]  0.0062  0.0008  1.0000  0.0068 -0.0134
## [4,]  0.0385 -0.0095  0.0068  1.0000 -0.0189
## [5,] -0.0038 -0.0133 -0.0134 -0.0189  1.0000
```

```
  corrpvalues
```

```
##        [,1]   [,2]   [,3]   [,4]   [,5]
## [1,] 0.0000 0.7894 0.7198 0.0264 0.8266
## [2,] 0.7894 0.0000 0.9642 0.5816 0.4440
## [3,] 0.7198 0.9642 0.0000 0.6969 0.4385
## [4,] 0.0264 0.5816 0.6969 0.0000 0.2743
## [5,] 0.8266 0.4440 0.4385 0.2743 0.0000
```

Chapter 4

READ IN THE HOUSES DATASET AND PREPARE THE DATA

```
  setwd("~/IMGT680")
  houses <- read.csv(file="houses.csv", stringsAsFactors = FALSE, header = FALSE)
  names(houses) <- c("MVAL", "MINC", "HAGE", "ROOMS", "BEDRMS", "POPN", "HHLDS", "LAT",
"LONG")

  # Standardize the variables
  houses$MINC_Z <- (houses$MINC - mean(houses$MINC))/(sd(houses$MINC))
  houses$HAGE_Z <- (houses$HAGE - mean(houses$HAGE))/(sd(houses$HAGE))
  houses$ROOMS_Z <- (houses$ROOMS - mean(houses$ROOMS))/(sd(houses$ROOMS))
  houses$BEDRMS_Z <- (houses$BEDRMS - mean(houses$BEDRMS))/(sd(houses$BEDRMS))
  houses$POPN_Z <- (houses$POPN - mean(houses$POPN))/(sd(houses$POPN))
  houses$HHLDS_Z <- (houses$HHLDS - mean(houses$HHLDS))/(sd(houses$HHLDS))
  houses$LAT_Z <- (houses$LAT - mean(houses$LAT))/(sd(houses$LAT))
  houses$LONG_Z <- (houses$LONG - mean(houses$LONG))/(sd(houses$LONG))

  # Randomly select 90% for the Training dataset
  choose <- runif(dim(houses)[1],0, 1)
  test.house <- houses[which(choose < .1),]
  train.house <- houses[which(choose <= .1),]
```

PRINCIPAL COMPONENT ANALYSIS

Performing PCA on part of our training data set. Extracting 8 componenets.

```
  # Requires library "psych"
  #install.packages("psych")
  library(psych)
  pca1 <- principal(train.house[,c(10:17)], nfactors=8, rotate="none", scores=TRUE)
```

PCA RESULTS

Eigenvalues outputted from PCA are outputted. Loadings are also outputted for each principal component. The second table tells us how much of the total variance each component explains. So for example, the first principal componenet explains 49% of the total variance.

```
  # Eigenvalues:
  pca1$values
```

```
## [1] 3.94568147 1.90131010 1.09601995 0.79922247 0.12108387 0.08143062
## [7] 0.04504160 0.01020992
```

```
  # Loadings matrix,
  # variance explained,
  pca1$loadings
```

```
##
## Loadings:
##          PC1    PC2    PC3    PC4    PC5    PC6    PC7    PC8
## MINC_Z                   0.903  0.418
## HAGE_Z   -0.429         -0.480  0.765
## ROOMS_Z   0.959  0.114          0.112          0.167 -0.125
## BEDRMS_Z  0.970         -0.124          0.145
## POPN_Z    0.945         -0.108         -0.296
## HHLDS_Z   0.974         -0.114
## LAT_Z    -0.157  0.967                         0.136  0.108
## LONG_Z    0.170 -0.965                         0.141  0.103
##
##                  PC1   PC2   PC3   PC4   PC5   PC6   PC7   PC8
## SS loadings    3.946 1.901 1.096 0.799 0.121 0.081 0.045 0.010
## Proportion Var 0.493 0.238 0.137 0.100 0.015 0.010 0.006 0.001
## Cumulative Var 0.493 0.731 0.868 0.968 0.983 0.993 0.999 1.000
```

SCREE PLOT

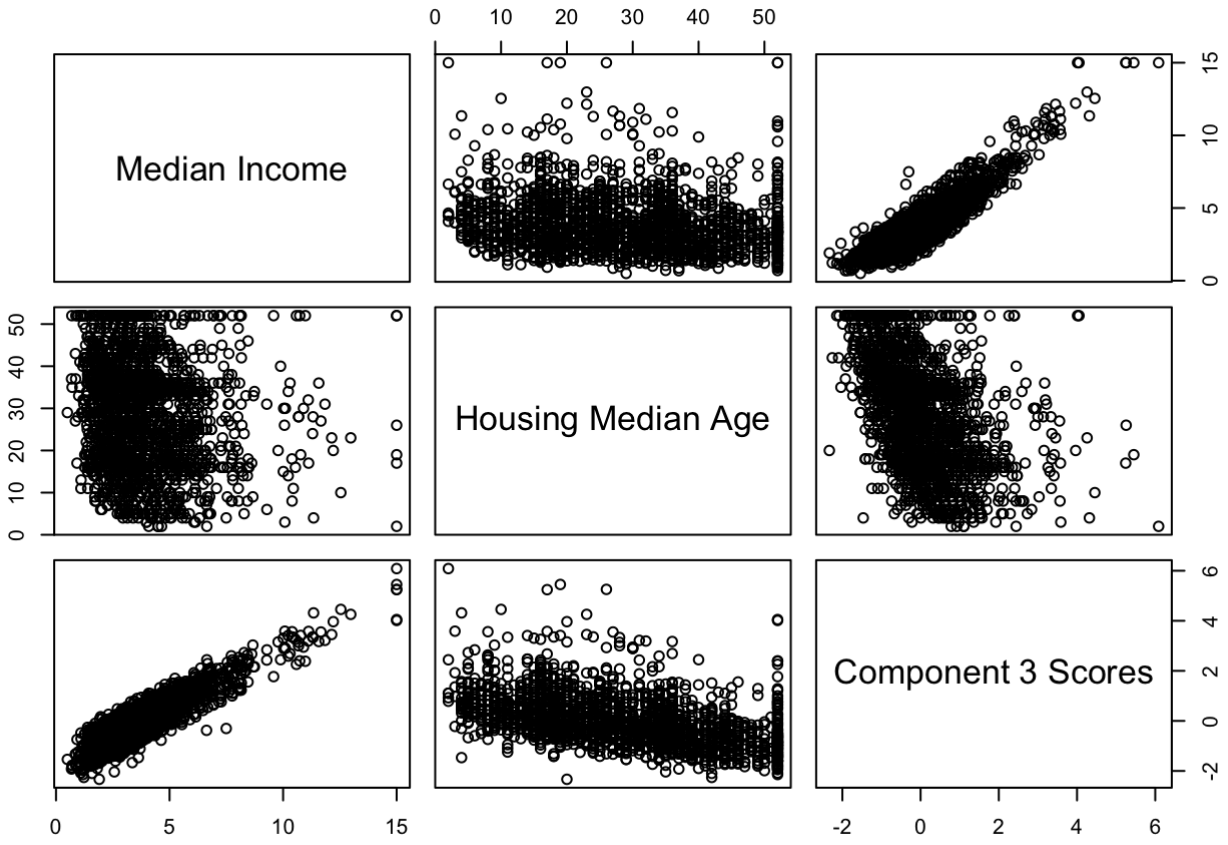
Scree plot of the eigenvalues against the component number. This plot is helpful in looking at the upper bound of componenets to retain. In this case, it appears we should extract a maximum number of four components as after that, the line begins to straighten out.

```
plot(pca1$values, type = "b", main = "Scree Plot for Houses Data")
```

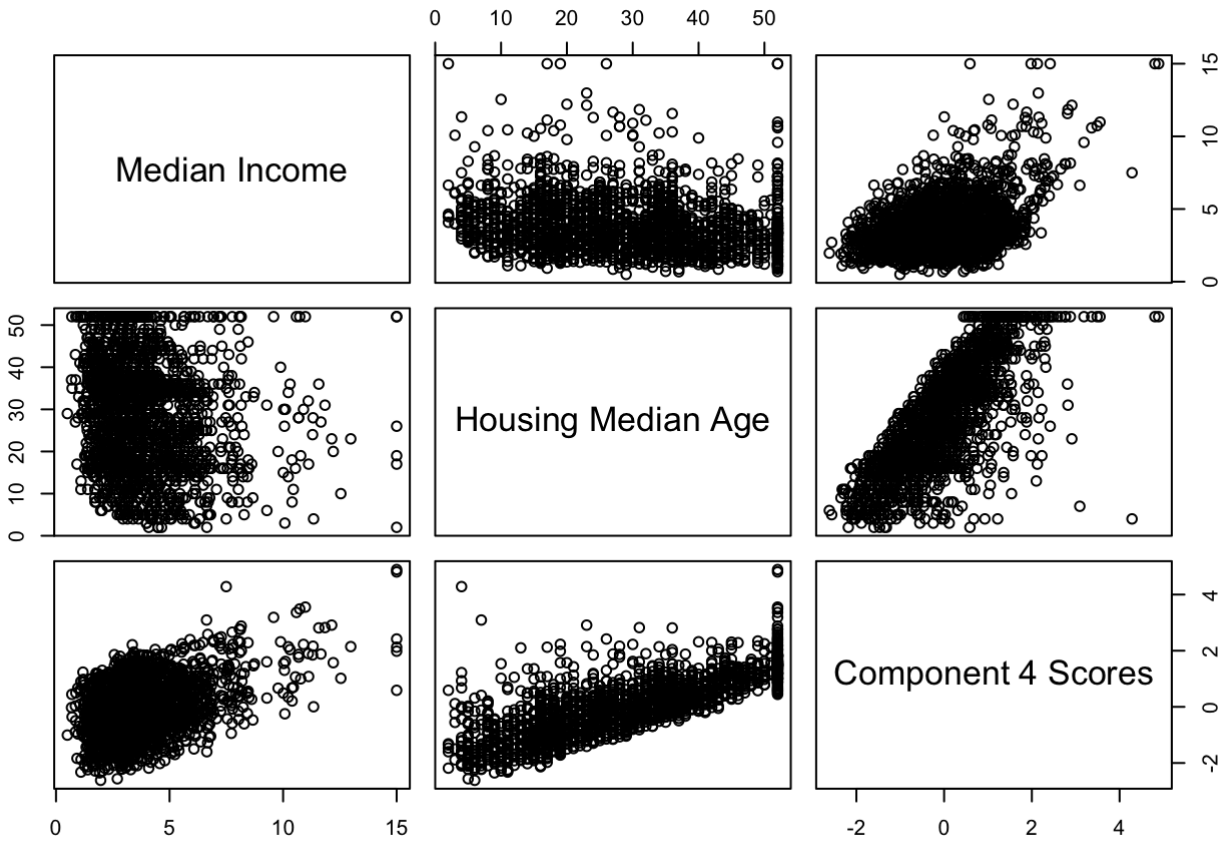
PLOT FACTOR SCORES

Investigate the relationship between principal components 3 and 4 and their constituent variables. We see the relationships among median income, housing median age, and the factor scores for component 3 while the second matrix plot displays the relationships among median income, housing median age, and the factor scores for componenet 4. We see between componenet 3 and median income there is a positive correlation which reflects the positive correlation of 0.922. However, it is harder to estimate the correlation between component 3 and housing median age as seen in the lack of pattern in the graphs. In the second grpah, we see a positive correlation between componenet 4 and housing median age. Therefore, we can tell that the component weight of -0.407 for housing median age in component 3 is not of practical significance and similarly for the component weight for median income in component 4.

```
pairs(~train.house$MINC + train.house$HAGE+pca1$scores[,3],
       labels = c("Median Income", "Housing Median Age", "Component 3 Scores"))
```




```
pairs(~train.house$MINC + train.house$HAGE+pca1$scores[,4],
       labels = c("Median Income", "Housing Median Age", "Component 4 Scores"))
```



CALCULATE COMMUNALITIES

Communality values for housing median age are calculated with one retaining 3 and the other retaining 4 components. Anything less than 0.5 is considered low as the variables share less than half of its variability in common with other variables. We see here that extracting three componenets is not enough as housing median age only shares 35% of its variance with other variables. It would be better the extract the fourth component to get the communality for housing median age over the 50% threshold.

```
  comm3 <- loadings(pca1)[2,1]^2 + loadings(pca1)[2,2]^2 + loadings(pca1)[2,3]^2
  comm4 <- loadings(pca1)[2,1]^2 + loadings(pca1)[2,2]^2+ loadings(pca1)[2,3]^2 + loadin
gs(pca1)[2,4]^2
  comm3; comm4
```

```
## [1] 0.4145037
```

```
## [1] 0.9991174
```

VALIDATION OF THE PRINCIPAL COMPONENTS

Here we are validating our findings from earlier by running PCA with just 4 componenets.

```
  pca2 <- principal(test.house[,c(10:17)], nfactors=4, rotate="none", scores=TRUE)
  pca2$loadings
```

```
##
## Loadings:
##          PC1    PC2    PC3    PC4
## MINC_Z                   0.903  0.418
## HAGE_Z   -0.429         -0.480  0.765
## ROOMS_Z   0.959  0.114          0.112
## BEDRMS_Z  0.970         -0.124
## POPN_Z    0.945         -0.108
## HHLDS_Z   0.974         -0.114
## LAT_Z    -0.157  0.967
## LONG_Z    0.170 -0.965
##
##                  PC1   PC2   PC3   PC4
## SS loadings    3.946 1.901 1.096 0.799
## Proportion Var 0.493 0.238 0.137 0.100
## Cumulative Var 0.493 0.731 0.868 0.968
```

READ IN AND PREPARE DATA FOR FACTOR ANALYSIS

Reading dataset in, standardizing and then spliting it up into testing and training set.

```
  setwd("~/IMGT680")
  adult <- read.csv(file="adult.txt", stringsAsFactors = FALSE)
  adult$"capnet"<- adult$capital.gain-adult$capital.loss
  adult.s <- adult[,c(1,3,5,13,16)]

  # Standardize the data:
  adult.s$AGE_Z <- (adult.s$age - mean(adult.s$age))/(sd(adult.s$age))
  adult.s$DEM_Z <- (adult.s$demogweight - mean(adult.s$demogweight))/(sd(adult.s$demogwe
ight))
  adult.s$EDUC_Z <- (adult.s$education.num - mean(adult.s$education.num))/(sd(adult.s$ed
ucation.num))
  adult.s$CAPNET_Z <- (adult.s$capnet - mean(adult.s$capnet))/(sd(adult.s$capnet))
  adult.s$HOURS_Z <- (adult.s$hours.per.week - mean(adult.s$hours.per.week))/(sd(adult.s
$hours.per.week))

  # Randomly select a Training dataset
  choose <- runif(dim(adult.s)[1],0, 1)
  test.adult <- adult.s[which(choose < .1), c(6:10)]
  train.adult <- adult.s[which(choose >= .1), c(6:10)]
```

BARTLETT'S TEST FOR SPHERICITY

p value rounds to 0 so the null hypothesis that no correlation exists among the variables is rejected.

```r
# Requires package psych
library(psych)
corrmat1 <- cor(train.adult, method = "pearson")
cortest.bartlett(corrmat1, n = dim(train.adult)[1])
```

```
## $chisq
## [1] 1237.758
##
## $p.value
## [1] 1.030877e-259
##
## $df
## [1] 10
```

FACTOR ANALYSIS WITH FIVE COMPONENTS

Table telling us how much variability each factor extracts.

```r
# Requires psych, GPArotation
#install.packages("GPArotation")
library(GPArotation)
fa1 <- fa(train.adult, nfactors=5, fm = "pa", rotate="none", SMC=FALSE)
fa1$values # Eigenvalues
```

```
## [1] 1.2712992 1.0405522 0.9529765 0.9119537 0.8232185
```

```r
fa1$loadings # Loadings, proportion of variance, and cumulative variance
```

```
##
## Loadings:
##          PA1    PA2    PA3    PA4    PA5
## AGE_Z     0.418 -0.541  0.548  0.366  0.313
## DEM_Z    -0.237  0.765  0.412  0.348  0.262
## EDUC_Z    0.630  0.240 -0.389 -0.209  0.592
## CAPNET_Z  0.534  0.253  0.511 -0.531 -0.328
## HOURS_Z   0.598  0.203 -0.267  0.576 -0.445
##
##                  PA1   PA2   PA3   PA4   PA5
## SS loadings    1.271 1.041 0.953 0.912 0.823
## Proportion Var 0.254 0.208 0.191 0.182 0.165
## Cumulative Var 0.254 0.462 0.653 0.835 1.000
```

FACTOR ANALYSIS WITH TWO COMPONENTS

FA is performed with just two factors. We see that the variability extracted is much lower due to factor loadings having much weaker correlations among the standardized variables.

```
  fa2 <- fa(train.adult, nfactors=2, fm = "pa", max.iter = 200, rotate="none")
  fa2$values # Eigenvalues
```

```
## [1]  0.538282457  0.352627230  0.034569556 -0.004095492 -0.031470647
```

```
  fa2$loadings # Loadings
```

```
##
## Loadings:
##          PA1    PA2
## AGE_Z     0.607 -0.314
## DEM_Z    -0.117
## EDUC_Z    0.278  0.438
## CAPNET_Z  0.185  0.148
## HOURS_Z   0.212  0.202
##
##                  PA1   PA2
## SS loadings    0.538 0.353
## Proportion Var 0.108 0.071
## Cumulative Var 0.108 0.178
```

```
  fa2$communality # Communality
```

```
##      AGE_Z      DEM_Z     EDUC_Z   CAPNET_Z    HOURS_Z
## 0.46679843 0.01372287 0.26878565 0.05598299 0.08561973
```

VARIMAX ROTATION

Redistributes the percentage of variance explained. We see percentage of variablity explained is still very low and looking at communality, the low values reflect the fact that there is not much shared correlation among the variables.

```
  fa2v <- fa(train.adult, nfactors = 2, fm = "pa", max.iter = 200, rotate="varimax")
  fa2v$loadings
```

```
##
## Loadings:
##          PA1    PA2
## AGE_Z     0.682
## DEM_Z    -0.106
## EDUC_Z            0.518
## CAPNET_Z          0.221
## HOURS_Z           0.281
##
##                  PA1   PA2
## SS loadings    0.491 0.400
## Proportion Var 0.098 0.080
## Cumulative Var 0.098 0.178
```

```
  fa2v$communality
```

```
##      AGE_Z      DEM_Z     EDUC_Z   CAPNET_Z    HOURS_Z
## 0.46679843 0.01372287 0.26878565 0.05598299 0.08561973
```

USER-DEFINED COMPOSITES

```
  small.houses <- houses[,c(4:7)]
  a <- c(1/4, 1/4, 1/4, 1/4)
  W <- t(a)*small.houses
```

Chapter 7

READ IN THE DATA, PARTITION TRAINING AND TESTING DATA

Reading in the adult data and taking in length of their income. Partitioning if length is less than or equal to 0.75 for training and if greater, put into testing data set.

```
  setwd("~/IMGT680")
  adult <- read.csv(file = "adult.txt", stringsAsFactors=TRUE)
  choose <- runif(length(adult$income), min = 0, max = 1)
  training <- adult[choose <= 0.75,]
  testing <- adult[choose > 0.75,]
  adult[1:5, c(1,2,3)]
```

	age	workclass	demogweight
	<int>	<fctr>	<int>
1	39	State-gov	77516
2	50	Self-emp-not-inc	83311
3	38	Private	215646
4	53	Private	234721
5	28	Private	338409

5 rows

```
  training[1:5, c(1,2,3)]
```

	age	workclass	demogweight
	<int>	<fctr>	<int>
1	39	State-gov	77516
2	50	Self-emp-not-inc	83311
3	38	Private	215646

	age	workclass	demogweight
	<int>	<fctr>	<int>
6	37	Private	284582
8	52	Self-emp-not-inc	209642

5 rows

```
testing[1:5, c(1,2,3)]
```

	age	workclass	demogweight
	<int>	<fctr>	<int>
4	53	Private	234721
5	28	Private	338409
7	49	Private	160187
9	31	Private	45781
10	42	Private	159449

5 rows

REMOVE THE TARGET VARIABLE, INCOME, FROM THE TESTING DATA

Here, the variable income is removed from the testing data set.The -15 is because income is in the 15th column.

```
names(testing)
```

```
##  [1] "age"            "workclass"      "demogweight"    "education"
##  [5] "education.num"  "marital.status" "occupation"     "relationship"
##  [9] "race"           "sex"            "capital.gain"   "capital.loss"
## [13] "hours.per.week" "native.country" "income"
```

```
# Target variable is in Column 15
testing <- testing[,-15]
names(testing)
```

```
##  [1] "age"            "workclass"      "demogweight"    "education"
##  [5] "education.num"  "marital.status" "occupation"     "relationship"
##  [9] "race"           "sex"            "capital.gain"   "capital.loss"
## [13] "hours.per.week" "native.country"
```

```
# Target variable is no longer in the testing data
```

REMOVE THE PARTITIONING VARIABLE, PART, FROM BOTH DATA SETS

Remove the partitioning variable Part from both data sets. -15 indicates that Part is currently the 15th variable and it is being removed in the testing and the 16th in training.

```
  # Part is now the 15th variable
  testing <- testing[,-15]
  names(testing)
```

```
##  [1] "age"            "workclass"      "demogweight"    "education"
##  [5] "education.num"  "marital.status" "occupation"     "relationship"
##  [9] "race"           "sex"            "capital.gain"   "capital.loss"
## [13] "hours.per.week" "native.country"
```

```
  names(training)
```

```
##  [1] "age"            "workclass"      "demogweight"    "education"
##  [5] "education.num"  "marital.status" "occupation"     "relationship"
##  [9] "race"           "sex"            "capital.gain"   "capital.loss"
## [13] "hours.per.week" "native.country" "income"
```

```
  # Part is the 16th variable in the training data set
  training <- training[,-16]
  names(training)
```

```
##  [1] "age"            "workclass"      "demogweight"    "education"
##  [5] "education.num"  "marital.status" "occupation"     "relationship"
##  [9] "race"           "sex"            "capital.gain"   "capital.loss"
## [13] "hours.per.week" "native.country" "income"
```