Tutorial 4

Exploratory Data Analysis

Note: Lines beginning with "#" are comment lines explaining what we are doing. The R compiler skips these lines. Lines indented (e.g., Create a Histogram, below) are meant to be on the same line as the one above. A semicolon tells R to separate one line into two lines, one before and one after the semicolon.

 # Input data set Churn into Data Frame "Churn"

```
churn <- read.csv(file = "C:/ . . . /churn.txt",
   stringsAsFactors=TRUE)
# Show the first ten records
churn[1:10,]
```

```
> churn[1:10,]
   State Account.Length Area.Code    Phone Int.l.Plan
1     KS            128       415 382-4657         no
2     OH            107       415 371-7191         no
3     NJ            137       415 358-1921         no
4     OH             84       408 375-9999        yes
5     OK             75       415 330-6626        yes
6     AL            118       510 391-8027        yes
7     MA            121       510 355-9993         no
8     MO            147       415 329-9001        yes
9     LA            117       408 335-4719         no
10    WV            141       415 330-8173        yes
```

 # Summarize the Churn variable

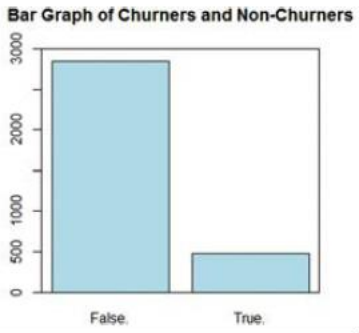
```
sum.churn <- summary(churn$Churn)
sum.churn
```

 # Calculate proportion of churners

```
prop.churn <- sum(churn$Churn == "True") /
   length(churn$Churn)
prop.churn
```

 # Bar chart of variable Churn

```
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

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

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

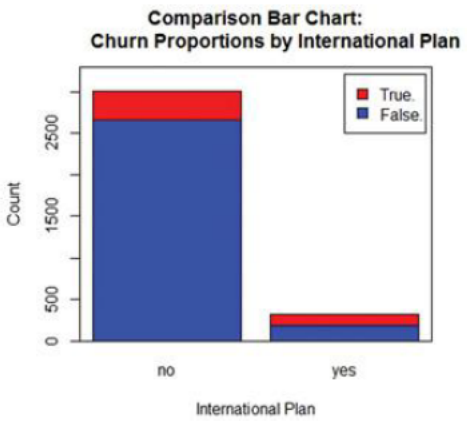
 # Create a table with sums for both variables

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

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

 # Overlayed bar chart

```
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 of proportions over rows

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

```
> row.margin
        International Plan
Churn       no   yes
  False. 93.47  6.53
  True.  71.64 28.36
```

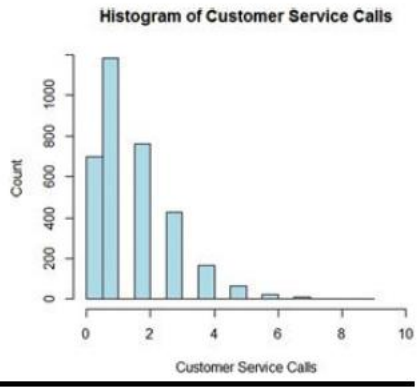
 # Create a table of proportions over columns

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

```
> col.margin
        International Plan
Churn       no   yes
  False. 88.50 57.59
  True.  11.50 42.41
```

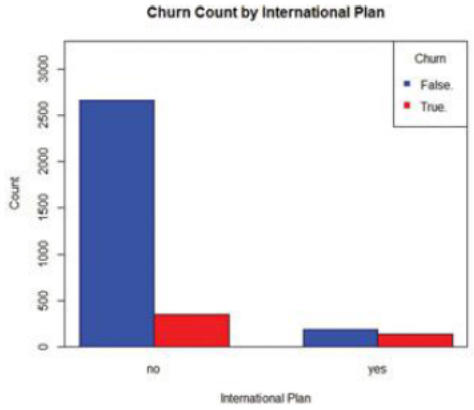
 # Histogram of non-overlayed Customer Service Calls

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



 # Clustered Bar Chart, with legend

```
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")
```



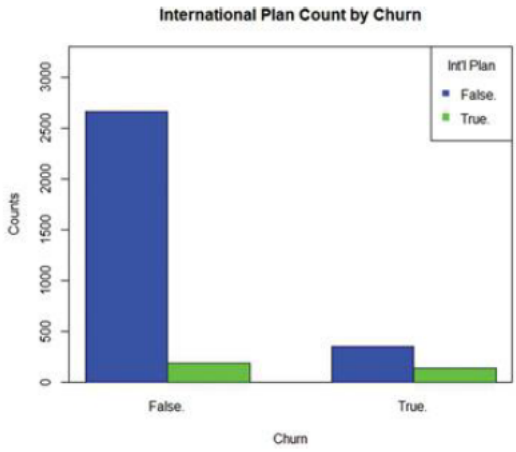
 # Download and install the R Package *ggplot2*

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

```
70: Turkey                71: UK (Bristol)          72: UK (London)
73: UK (St Andrews)       74: USA (CA 1)            75: USA (CA 2)
76: USA (IA)              77: USA (IN)              78: USA (KS)
79: USA (MD)              80: USA (MI)              81: USA (MO)
82: USA (OH)              83: USA (OR)              84: USA (PA 1)
85: USA (PA 2)            86: USA (TN)              87: USA (TX 1)
88: USA (WA 1)            89: USA (WA 2)            90: Venezuela
91: Vietnam

Selection: 74
```

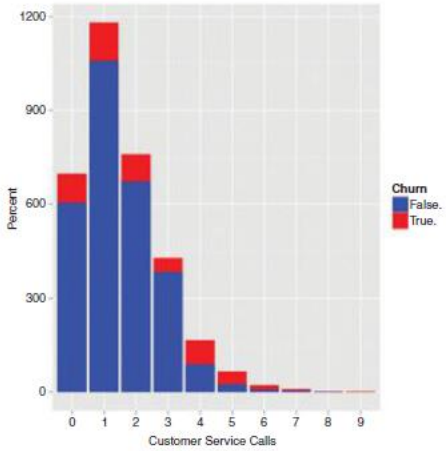
 # Clustered Bar Chart of Churn and Int'l Plan with legend

```
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")
```

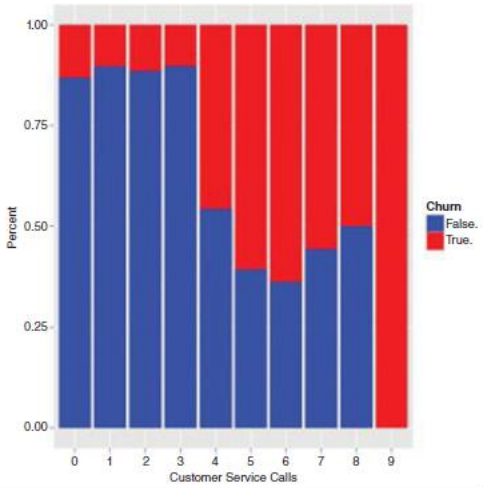


 # Overlayed bar charts

```
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"))
```



Check these commands

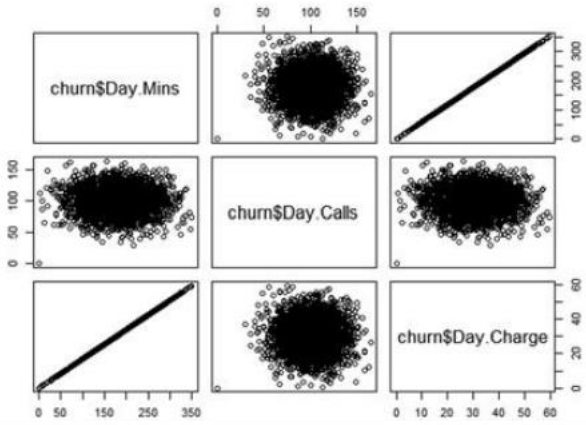
 # Two-sample T-Test for Int'l Calls

```
# 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)
```

```
> 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 = 640.643, 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 matrix

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



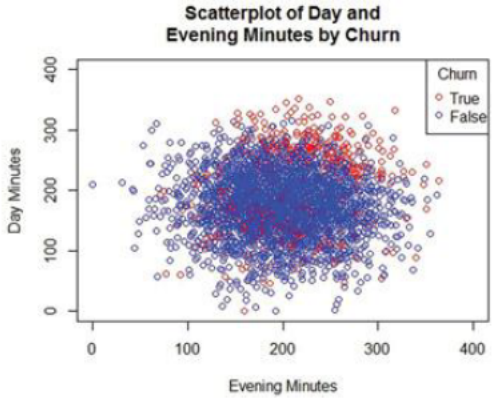
 # Regression of Day Charge vs Day Minutes

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

```
> 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
```

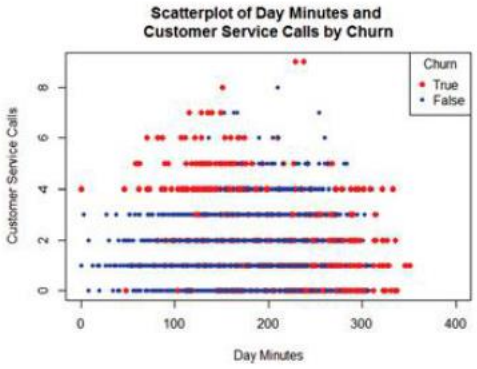
 # Scatterplot of Evening Minutes and Day Minutes, colored by Churn

```
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

```
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")
```



 # Correlation values, with p-values

```
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$Day.Charge)
round(cor(days),
   4)
MinsCallsTest$p.value
MinsChargeTest$p.value
CallsChargeTest$p.value
```

```
> 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

```
# Collect variables of interest
corrdata <- cbind(churn$Account.Length,
   churn$VMail.Message,
   churn$Day.Mins,
   churn$Day.Calls,
   churn$CustServ.Calls)
# Declare the matrix
corrpvalues <- matrix(rep(0, 25),
   ncol = 5)
# Fill the matrix with correlations
for (i in 1:4) {
  for (j in (i+1):5) {
    corrpvalues[i,j] <- corrpvalues[j,i] <-
      round(cor.test(corrdata[,i],
        corrdata[,j])$p.value,
    4)
  }
}
round(cor(corrdata), 4)
corrpvalues
```

```
> 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
```

Reference:

Discovering Knowledge in Data: An Introduction to Data exploration, Second Edition, by Daniel Larose and Chantal Larose, John Wiley and Sons, Inc., 2014.