REVOLUTION ANALYTICS

Introduction to R: High Level R Demo

Revolution Analytics

Module Objectives

Briefly Explore R's abilities to

- Import data
- Process data
- Visualize data

Demonstration: Where we want to go…

- Let's say that we have a csv data file containing information on an ad click campaign.
- How do we load that data so that we can interact with it?
- How might we explore it?
- What else might we want to do?

Demonstration - Overview

In this demonstration we run a short demo of data set loading, manipulation, and exploration. Let's say that we have a .csv data file containing information on an ad click campaign.

- How do we load that data so that we can interact with it?
- How might we explore it?
- What else might we want to do?

Change the working directory

```
getwd()
`?`(setwd)
```

Load the data into the workspace

```
data.path <- "../data"
datafile <- file.path(data.path, "performance.csv")
performance <- read.csv(datafile, stringsAsFactors = FALSE)
```

Inspect the data structure

```
str(performance)  ## look at the data
```

```
## 'data.frame':    90 obs. of  18 variables:
## $ Origin.Code  : chr  "4800 - Expansion- Health 2013" "4800 - Expansion- Health 2013" "4800 - Expan
##  $ Notes        : chr  "Intraday LD" "Intraday LD" "Intraday LD" "Intraday LD" ...
##  $ Domain       : chr  "ABC" "ABC" "ABC" "ABC" ...
##  $ Tier         : chr  "other" "other" "other" "other" ...
##  $ Search.Engine: chr  "XYZ" "XYZ" "XYZ" "XYZ" ...
##  $ Click.Date   : chr  "-" "3/13/13" "3/14/13" "3/15/13" ...
## $ Impressions  : int  14881374 574194 537764 457362 366241 407772 580147 577391 572859 543243 ...
##  $ Clicks       : int  309356 11573 10645 9412 9998 8693 11491 11901 11398 10734 ...
##  $ Engine.CTR   : chr  "2.08%" "2.02%" "1.98%" "2.06%" ...
##  $ Picks        : int  156498 5890 5115 4659 5381 4470 5816 6070 5608 5366 ...
##  $ LP.CTR       : chr  "50.59%" "50.89%" "48.05%" "49.50%" ...
##  $ CPC          : num  0.001 0.002 0.002 0.002 0.002 0.002 0.001 0.002 0.001 0.002 ...
##  $ RPC          : num  0.002 0.002 0.002 0.002 0.002 0.002 0.002 0.002 0.002 0.002 ...
...
```

Reformat Data

```
## Date Not character!
performance$Click.Date <- as.Date(performance$Click.Date, format = "%m/%d/%y")

## Numeric, not chararacter!
numeric.cols <- c("Engine.CTR", "LP.CTR")
removePercent <- function(x) as.numeric(sub("%", "", x))/100
performance[numeric.cols] <- lapply(performance[numeric.cols], removePercent)

## Numeric, not character!
dollar.cols <- c("Cost", "Rev", "Margin")
removeDollar <- function(x) as.numeric(gsub("[\\$ )]", "", sub("\\(",
  "-", x)))

performance[dollar.cols] <- lapply(performance[dollar.cols], removeDollar)
## write.csv(performance,
## file='../data/performance.refmt.csv',row.names=FALSE)
```

Inspect the data structure (again)

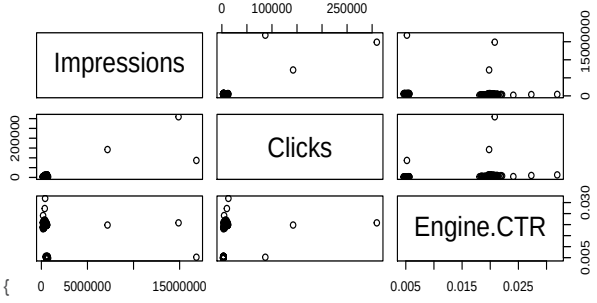
```
str(performance)  ## look at the data
```

```
## 'data.frame':    90 obs. of  18 variables:
## $ Origin.Code  : chr  "4800 - Expansion- Health 2013" "4800 - Expansion- Health 2013" "4800 - Expan
##  $ Notes        : chr  "Intraday LD" "Intraday LD" "Intraday LD" "Intraday LD" ...
##  $ Domain       : chr  "ABC" "ABC" "ABC" "ABC" ...
##  $ Tier         : chr  "other" "other" "other" "other" ...
##  $ Search.Engine: chr  "XYZ" "XYZ" "XYZ" "XYZ" ...
##  $ Click.Date   : Date, format: NA "2013-03-13" ...
## $ Impressions  : int  14881374 574194 537764 457362 366241 407772 580147 577391 572859 543243 ...
##  $ Clicks       : int  309356 11573 10645 9412 9998 8693 11491 11901 11398 10734 ...
## $ Engine.CTR   : num  0.0208 0.0202 0.0198 0.0206 0.0273 0.0213 0.0198 0.0206 0.0199 0.0198 ...
##  $ Picks        : int  156498 5890 5115 4659 5381 4470 5816 6070 5608 5366 ...
##  $ LP.CTR       : num  0.506 0.509 0.48 0.495 0.538 ...
##  $ CPC          : num  0.001 0.002 0.002 0.002 0.002 0.002 0.001 0.002 0.001 0.002 ...
##  $ RPC          : num  0.002 0.002 0.002 0.002 0.002 0.002 0.002 0.002 0.002 0.002 ...
...
```

Creating a plot

Use the function `plot()` to explore patterns in the data, after subsetting only numeric columns.

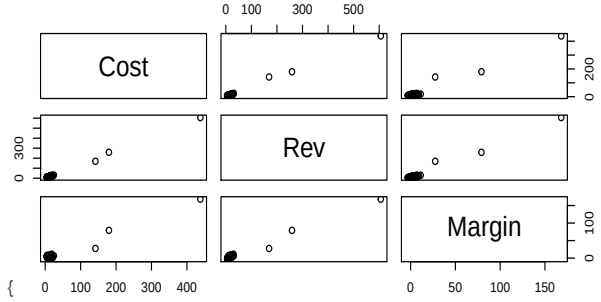
```
numeric.cols <- which(sapply(performance, is.numeric))
plot(performance[numeric.cols[1:3]])
```



Outliers

Zooming in on the last several variables, notice some outliers:

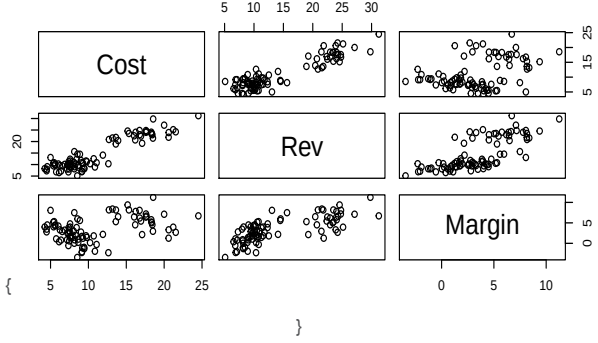
```
plot(performance[dollar.cols])
```



Remove outliers

Let's try removing the outliers associated with cost

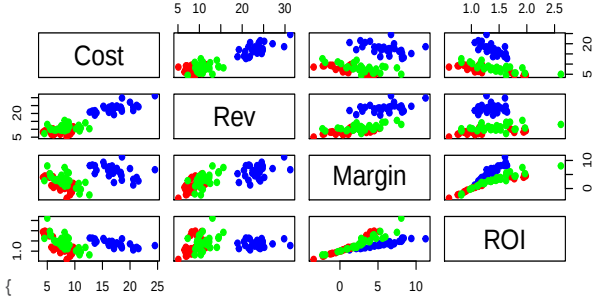
```
perf <- performance[performance$Cost < 100, ]
plot(perf[dollar.cols])
```



Clustering

That looks a lot better, but what explains the clustering pattern?

```
perf$Origin.Code <- factor(perf$Origin.Code)
originCol <- c("blue", "red", "green")[perf$Origin.Code]
plot(perf[, 15:ncol(perf)], col = originCol, pch = 19)
```



}

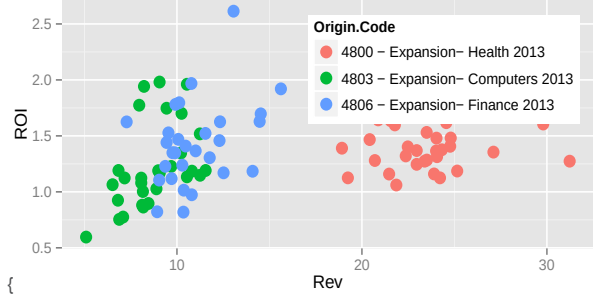
Another plotting package: ggplot2

Is there a better way to plot that allows us to color groupings without writing an akward function?

```
install.packages("ggplot2")
```

Creating a plot with ggplot2

```
library(ggplot2)
ggplot(perf, aes(x=Rev, y=ROI, colour=Origin.Code)) +
  geom_point(na.rm=TRUE, size=4) +
  theme(legend.justification=c(1,1), legend.position=c(1,1))
```



{

}

REVOLUTION ANALYTICS

Thank you

Revolution Analytics is the leading commercial provider of software and support for the popular open source R statistics language.

www.revolutionanalytics.com
1.855.GET.REVO
Twitter: @RevolutionR