Introduction to Data Science
Homework Assignment 3

Michael Discenza

Columbia University, New York, NY 10027, USA
mad2200@columbia.edu

1 FlowingData.com Visualization Tutorial

I went through the following tutorials on FlowingData:

- More on Making Heat Maps in R
- How to Visualize and Compare Distributions
- How to Make Bubble Charts

For my graph, I tried to use Nathan's tutorial on bubble charts to visualize data about the gender pay gap from the Bureau of Labor Statistics.

The graphs that I wanted to combine and the data from which the graphs were made are found here: http://www.bls.gov/opub/ted/2009/ted_20090807.htm

	Occupational Group	Women	W Earnings	Men	M earnings
1	Management	4535000	979	6687000	1384
2	Business and financial operations	2928000	885	2159000	1167
3	Computer and mathematical	828000	1088	2516000	1320
4	Architecture and engineering	334000	1001	2319000	1286
5	Life, physical, and social science	477000	931	603000	1156
6	Community and social services	1117000	753	791000	860
7	Legal	693000	962	506000	1696
8	Education, training, library	4883000	818	1794000	1020
9	Arts, design, entertainment, sports	689000	777	882000	951
10	Healthcare practitioner	4052000	909	1362000	1210

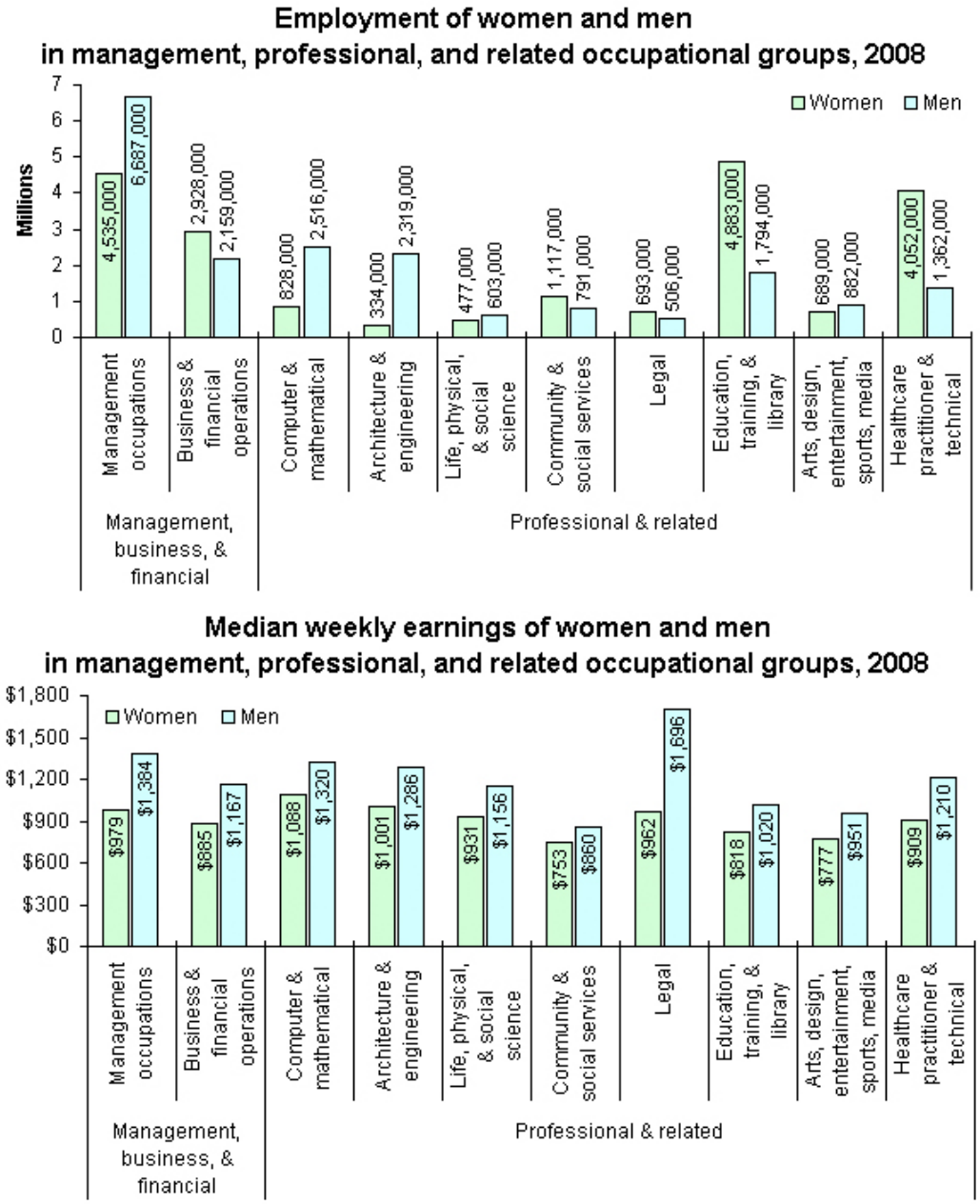

Figure 1: These are the two graphs that the BLS uses to illustrate the gender pay gap by occupation.

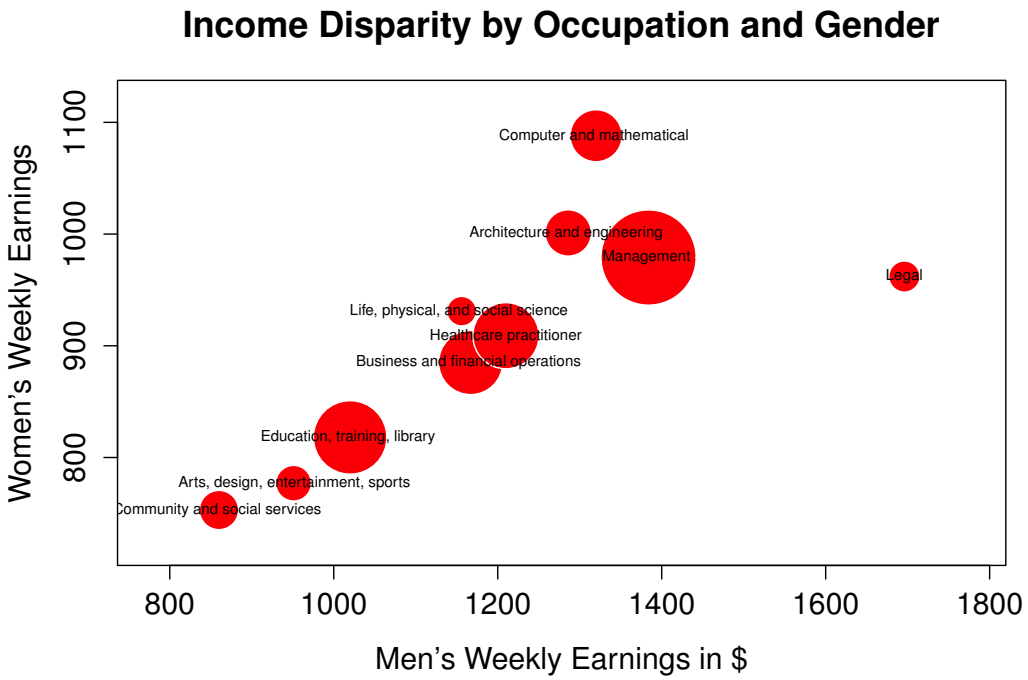

Figure 2: This is the graph that I made using the methods that Nathan used in his tutorial. I was not able to to make sure the coordinates were square to accurately show the slope of the line and indicate the pay inequality between men and women nor show the additional dimension of percentage women in each occupation group with a color scale.

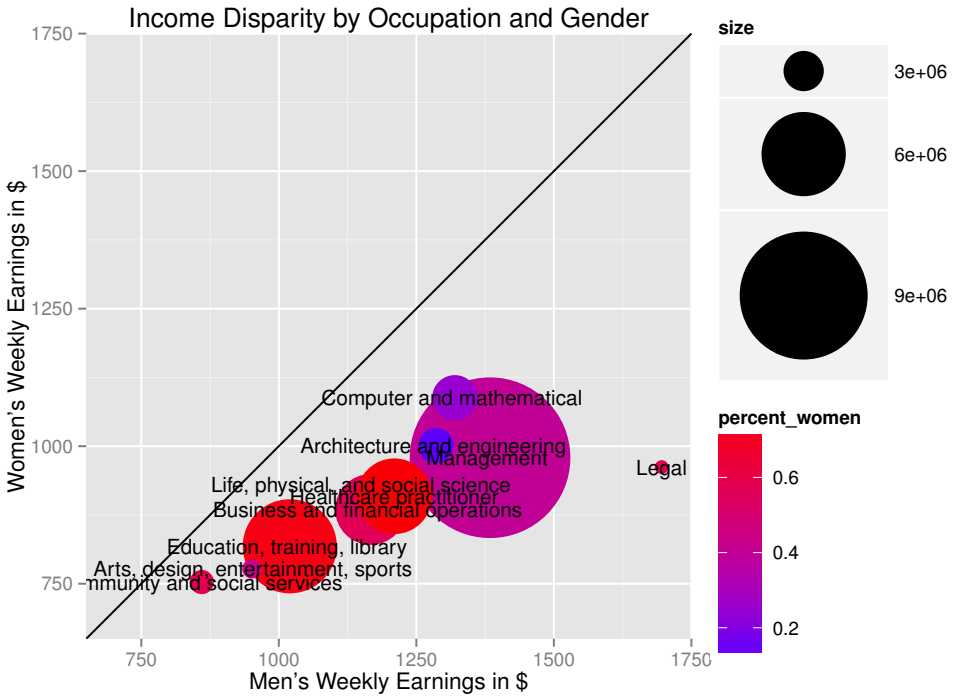

Figure 3: As an alternative to the symbols() graphing function that Nathan used, I recreated the graph using Hadley Wickham's ggplot2 package. The graph shows all of the information that I wanted it to show, but I think the end product might not be the most ideal because it contains perhaps too much information for a viewer to easily interpret and the labels associated with the data points are not aesthetically pleasing. This is something that I might have been able to change by editing them in Inkscape and drawing arrows from the labels to their respective data points in order to reduce the clutter on the plot.

It was very easy to follow Nathan's tutorials and helpful that he had explained what certain options do. In general, I spend hours, like multiple hours trying to get graphs to look the way I want becasue there aren't good cheat sheets and there isn't good graphical documentation for packages like ggplot. Next I would like to find or make a cheat sheet so that I don't spend as much time on visualizing data as I have been.

2 The Data Science of Art

I think that the points that everyone brings up about using social network analysis with data from service like Foursquare and Instagram are quite valid. And certainly I would say that the list of metrics to measure that Anderson created is quite exhaustive. I think though that Andersons list and mentality is honestly more like the U.S. News & World Report's rankings of Universities and Hospitals. Though its unclear whether museums would pay any attention to their rankings or if those rankings would even be published or circulated, but if they did start paying attention to their rank or composite index based on this data, we have to ask if that would be actually a good thing for the museums in terms of satisfying their more intrinsic goals or if it would lead to manipulation of metrics for higher rank like what happens in some colleges. I also think that the index fails to account for the various different missions of museums. For instance a Museum like the Brooklyn Museum, my favorite art Museum in the city, and one that gives me great joy has an altogether different character and atmosphere than the MoMA or the Met.

Moreover the rankings would probably create a consolidation of donations and it would make the process by which benefactors might figure out how to best spend their money overly market-based.

Though I am not in a position to endow and collections or donate much more money than a suggested entry fee to a museum, I imagine that there is an importance that a donor have some kind of personal connection the museum, some more intimate reason for donating as to make the donation somehow more legitimate.

Having articulated my discomfort with this idea of an index, I would like to focus now on some of the ways that I think methods in data science and machine learning might be useful for providing specific insights about user experience to help museum management enhance the enjoyment of their guests.

First I think it is important that internally as opposed to externally museum management and boards devise a set of metrics and indicators that they find to be important for their particular priorities. Once they do this they should maintain some kind of dashboard that is updated with at least some consistent frequency, as has become standard organizational practice so going forward, they can have sense of how various interventions they make in guest experiences effect the metrics that they deem important.

Second, I think that art museums (or maybe some new tech consulting firm that could specialize in art museums) should begin to leverage technology that has been used by retailers for a good amount of time now including motion tracking with web cams to better track guest interest in certain piece of certain kinds based on the among of time of the inferred position of units (the guests) within a map of the museum. This would allow the museums to make a number of useful interventions including feeding information and feedback to curators, better planning operations and queuing strategies to avoid lines and waits (basically what Disney Operations Research people do).

Finally, museums should try to experiment with new service like Art.sy and develop (or use a template for some kind of mobile app) to create mobile apps to scan QR codes associated with pieces of art to read descriptions and facts about the works they are viewing on their smart phones. These would be services presented under the auspices of providing enhanced information about works they are viewing, but that just as importantly for the museum management serve to create a rich click-stream-like dataset that can be cross referenced with the visual tracking of users and stored from one visit to the next to better understand how guests interaction with a museums collection (or multiple museums collections) varies over time.

3 Finance; Time Series and Regression

[Kaushik Reddy and I spent a few hours working together on the code and then split up for the remainder of the project due to schedule constraints]

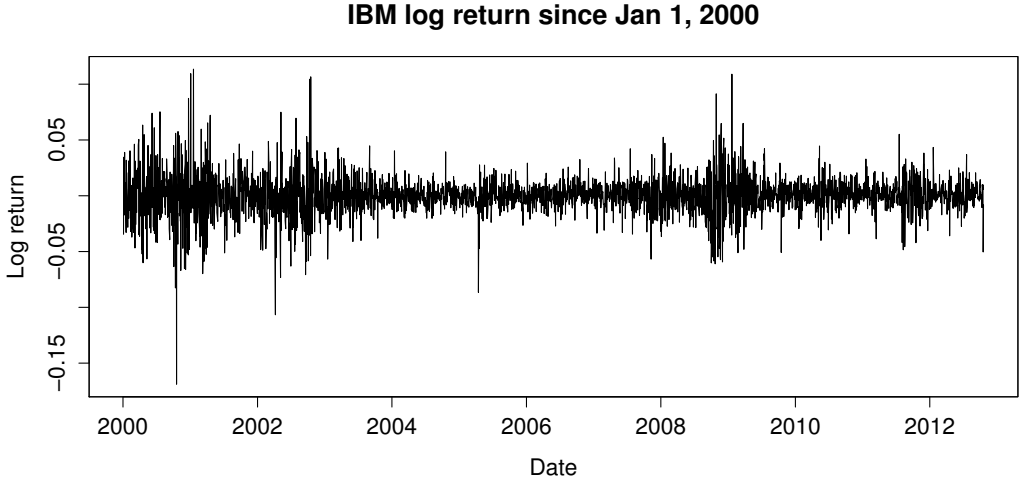

Figure 4:

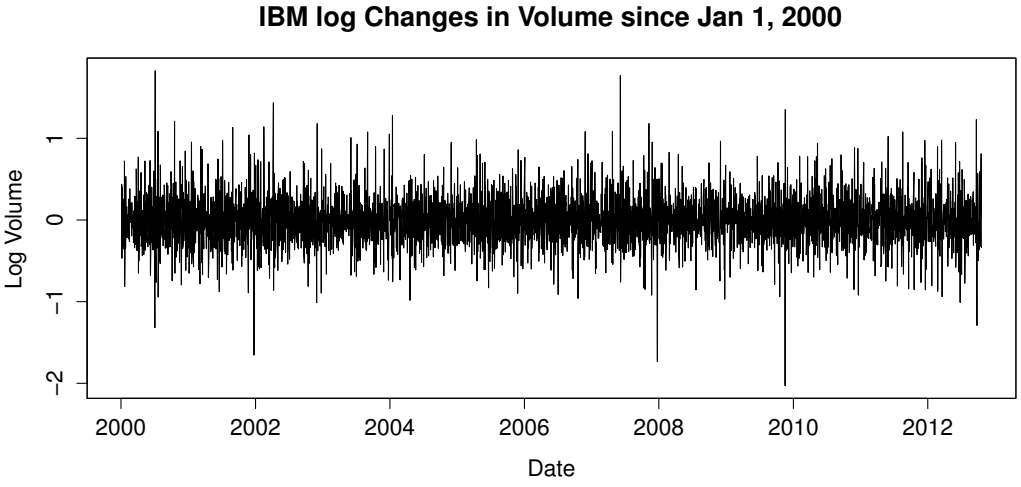

Figure 5:

I fit a model by regressing log returns on the previous two days' returns (an autoregressive second order model). This procedure was conducted by manually lagging the data and using the two previous days' log returns as perdictor vectors, not using the AR or ARIMA functions. I did fit an AR function that found the optimal number of predictors to be 36 using AIC as the selection criterion for the model, but was unable to use the model with R's predict function and such a model would likely have been tremendously overfit.

In the second order autoregressive model, the log returns of the previous day proved to be significant.

	Estimate	Std. Error	t value	Pr(>|t|)
(Intercept)	0.0001	0.0004	0.26	0.7919
Predictor1	-0.0502	0.0190	-2.64	0.0083
Predictor2	-0.0149	0.0190	-0.79	0.4318

Despite the significance of the coefficients fit by the model, the signal was week and often times not in the right direction of the log return. Run over the course of two years (on totally new test data after the model was fit to the previous 10 years of training data), the model earned a log return of -0.2447558, meaning that with this model, one would actually lose money.

This resulting log return is however based on a number of simplifying assumptions: 1) there are zero transaction fees, 2)when the model predicts either a positive or a negative return correctly, it makes the full log return associated with that day regardless of whether it was on the positive or negative side [i.e., the options market for purchasing shorts is perfectly efficient], 3) the model was not updated over time- it was only fit to the training data and was not recalibrated to either forget previous data points over time or incorporate new data as time progressed.

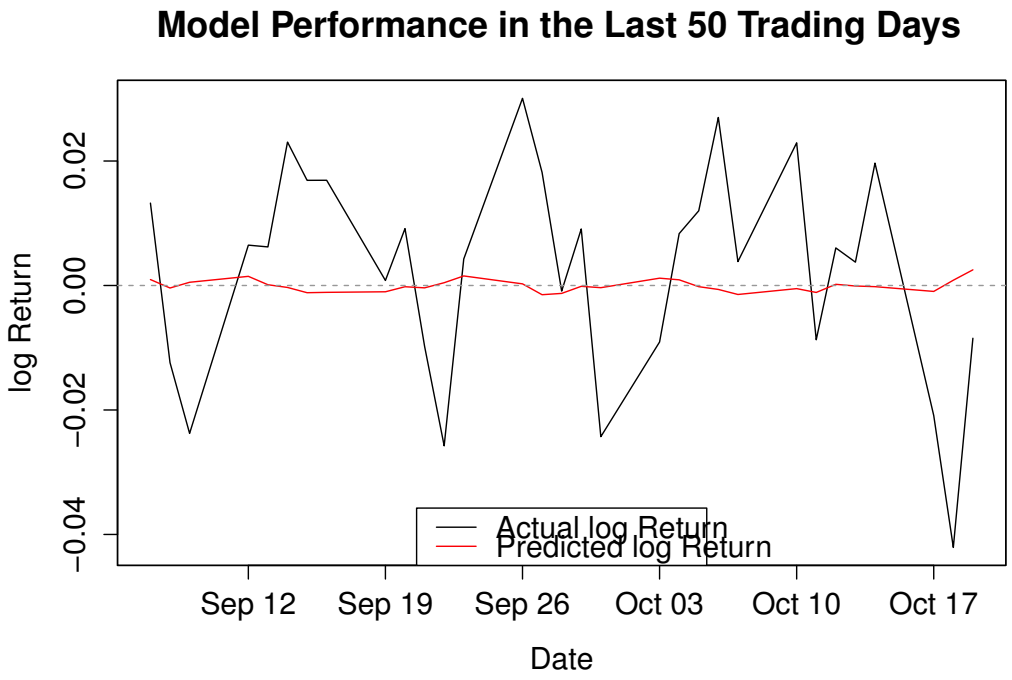

Figure 6: This graph shows the performance of the model over the last 50 days of tranding. When the red prediction line and the black line for actual log return have the same sign, the trading strategy makes money, and when they have opposite sides, it loses money

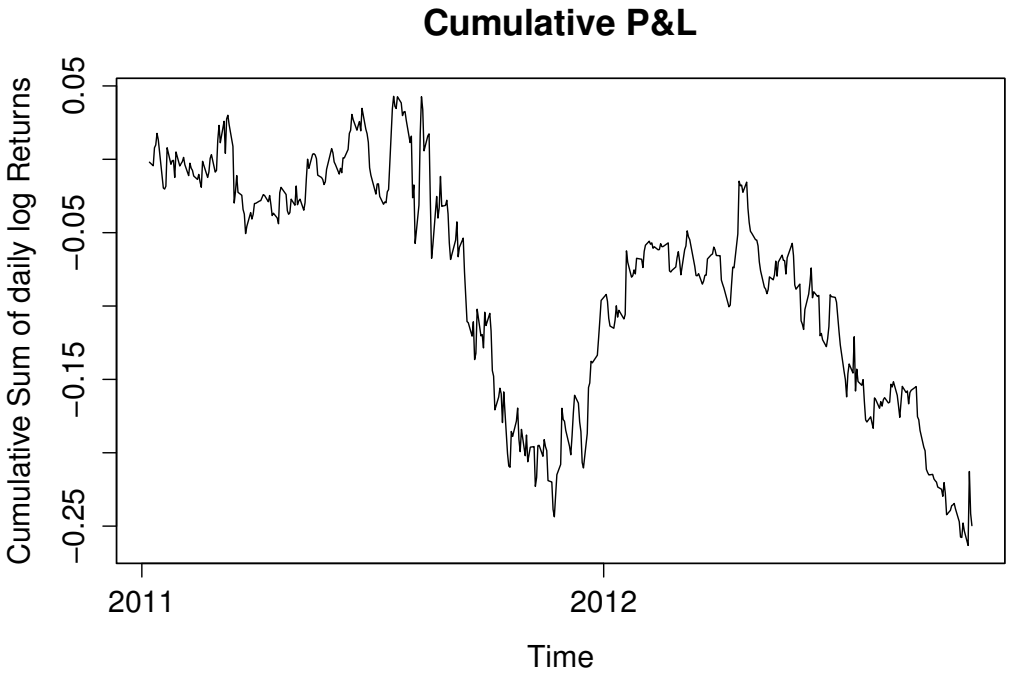

Figure 7: An ideal cumulative P&L graph would show a steady growth in profits. This model failed to produce a steady trajecotry or profitability

4 Get Glue Case Study

Note that to do this problem, I spent a ton of time figuring out how to use AWS's EC2 and run RStudio through the browser, which was pretty cool, but when I went to actually do the processing, the code that ran on my own machine did not run correctly on the server so I just ended up using the first 10,000 records of the dataset for the exercise and did the processing locally.

Answers to questions (albiet on only a subset of the data):

What actions can a user take?
Liking, disliking, or commenting

Number of Unique Users?
740 (based on userID and not display name, because the latter is not unique)

Top 10 most popular movies?
"The Dark Knight", "Fight Club", "Pulp Fiction", "The Hangover", "Slumdog Millionaire", "Iron Man", "WALL-E", "Star Wars: Episode V: The Empire Strikes Back", "Indiana Jones and the Last Crusade", "Up", "Star Wars: Episode VI: Return of the Jedi"(based on the total number of likes each recieved)

Number of events that occured in 2011?
I have no way of knowing only using this subset of the data

Other interesting questions that I proposed:

Which user generates the most useful comments?
In the 10,000 record sample, there were not that many comments and even fewer comments marked as useful, only 5 actually. But they were from the users with user IDs dandhroberts, kenjamd, PrinceAL, DelayedReality, and aka_sulley (We could potentially design this piece

of information back into the system and highlight his or her comments so that the other users could benefit the most)

Do more people rate movies or TV shows?

646 of the 740 users rated movies, but only 249 of those 740 users rated TV shows. Another way of interpeting the question is that 469 users have rated only movies and no TV shows and only 72 people have rated TV shows and no movies.

Are the majority of the comments negative or positive?

To solve this, we could read the comments, but if we had a much larger data set that had more than 84 comments, that would be unfeasible, so we can assume that if the same user likes and makes a comment. [I actually struggled a lot getting this from the data-I think that I was trying to do this in a way that relied too much on primative operations and not plyr or reshape–I'll come to office hrs and maybe you can help me figure this out]

Genres?

Another interesting way to look at the data is to use the genres associated with the movies - very possible to get by scraping IMDB or using one of the various APIs that provides movie genres and figure out the distribution of likes of our users. I did not have enough time to do this, but this is certainly something that would be useful to do.

Is there a correlation between the number of movies and number of TV shows a user rates/comments one? See graph b

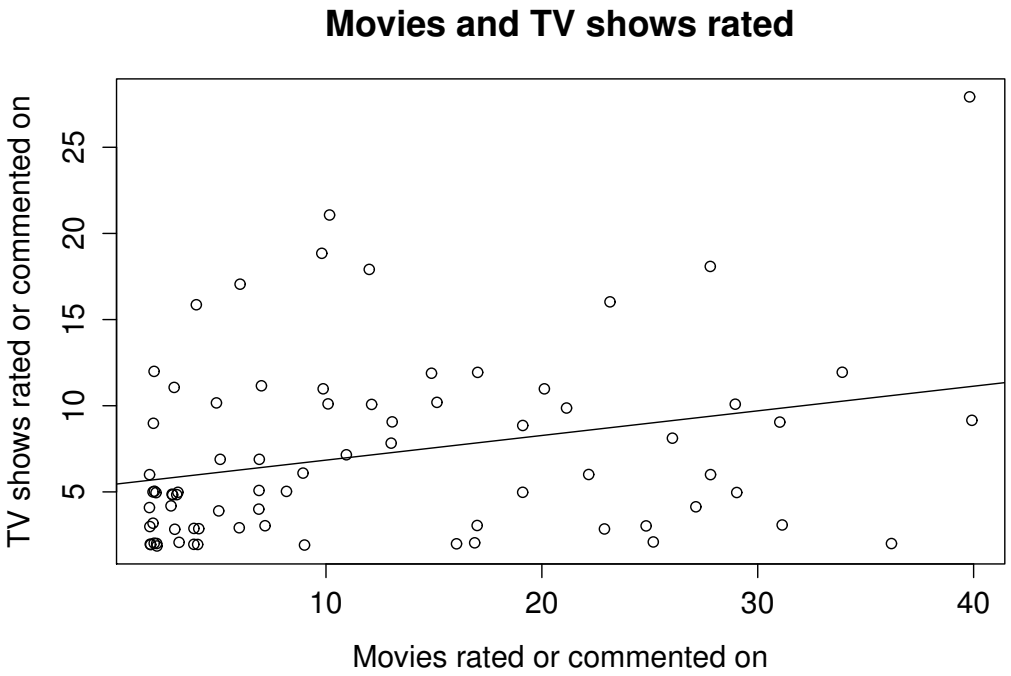

Figure 8: This plot shows the non-outlying counts of ratings and comments from users who have rated more than one title in each category. There seems to be a postive correlation bewteen the number of movies and the number of shows that users rate. Though this is supposed to be exploratory data analysis, I displayed the regression line that was fit using a first order linear model (which was found to have an R-squared value of 0.5949) for reference. With this concentration of data near the origin and and only a few throughout the range, this is certainly a data set that does not meet the constant variance assumption for applying linear regression. In order to apply a model to this kind of data, we need more users who might meet a minimum threshold of ratings, i.e. power users whose rate of Movie to TV show ratings we might actually be able to model

5 Code Appendix

All code (in easier to read form) as well as additional graphics files can be found on this assignment's git repository: https://github.com/mdiscenza/HW3

```
crime <- read.csv("http://datasets.flowingdata.com/crimeRatesByState2005.tsv", heade
#Flowing Data Tutorial Distruction
radius <- sqrt( oc$size/ pi)
symbols(oc$M_w_earnings, oc$W_w_earnings, circles=radius, inches=0.35, fg="white", bg
text(oc$M_w_earnings, oc$W_w_earnings, oc$Occupational.Group, cex=0.5)


#Can't use xy, ehhhh?
radius <- sqrt( oc$size/ pi)
xysymbols(oc$M_w_earnings, oc$W_w_earnings, circles=radius, inches=0.35, fg="white",
text(oc$M_w_earnings, oc$W_w_earnings, oc$Occupational.Group, cex=0.5)




library(ggplot2)
library(directlabels)
library(stringr)
??trim
oc$Occupational.Group<-trim(as.character(oc$Occupational.Group))
#my data set:
oc <- read.csv("~/Dropbox/Fall_2012/Intro_to_Data_Science/Assignments/HW3/Data_Scienc
oc$percent_women <- oc$W_employment/(oc$M_employment+oc$W_employment)
oc$size <- oc$W_employment+oc$M_employment
p <- ggplot(oc, aes(M_w_earnings, W_w_earnings, label = as.character(Occupational.Gr
p + geom_point(aes(size = size, colour=percent_women)) + scale_size_continuous(range=
    #geom_point(aes(colour = oc$percent_women)) +
    coord_equal() +
    scale_colour_gradient(high = "red", low="blue")+
    ylim(700, 1700) +
    xlim(700, 1700) +
    geom_abline(slope=1) +
    labs(title = "Income Disparity by Occupation and Gender") +
    ylab("Women's Weekly Earnings in $") +
    xlab("Men's Weekly Earnings in $") +
    geom_text(aes(label=Occupational.Group), size=4)


library(xtable)
oc_2 <- oc[1:6]
xtable(oc)

# Helper function that was that takes three arguments- the dates of the data, actual
test_returns <-function(dates, actual, predicted){
    returns <- 0
    earnings <- NULL
    for (day in 3:length(predicted)){
        temp_return <- NULL
        if (predicted[day]*actual[day]>0){ #multiply together, if same sign, positive
            returns <- returns + abs(actual[day])
            temp_return <- abs(actual[day])
        }
        else{
```

```
            returns <- returns - abs(actual[day])
            temp_return <- abs(actual[day])*-1
        }
        earnings[day-3] <- temp_return
    }
    new.dates <- dates[3:length(dates)]
    PL <- cumsum(earnings)
    ts.PL <- zoo(PL,new.dates)
    plot(ts.PL[1:(length(ts.PL)-1)], main="Cumulative P&L", xlab="Time", ylab= "Cumu
    return (returns)
# zoo object with earnings to see where the model works best... if it gets worse ove
}

setwd("~/Dropbox/Fall_2012/Intro_to_Data_Science/Assignments/HW3/Data_Science_HW3/LaT
library(lattice, xtable)
#getting the data from Yahoo Finance and making it into a time series object
IBM <- read.csv("http://ichart.finance.yahoo.com/table.csv?s=IBM&a=00&b=1&c=2000&d=09
IBM$Date<-as.Date(IBM$Date)
IBM<-IBM[order(IBM$Date),]
library(zoo)
dim(IBM)
ts.IBM <- zoo(IBM$Close, IBM$Date)

#calculating and plotting log return and storing it into a data frame
# this new df will be one record shorter because log returns is a change between two
log_return <- diff(log(IBM$Close))
IBM2 <- cbind(IBM[-1,], log_return)
ts.IBM.return <- zoo(IBM2$log_return, IBM2$Date)
plot(ts.IBM.return, main = "IBM log return since Jan 1, 2000", xlab="Date", ylab="Log
# calculating and plotting log volume changes:
IBM2$log_volume_change <- diff(log(IBM$Volume))
ts.IBM.volume <- as.ts(IBM2$log_volume_change, IBM2$Date)
plot(ts.IBM.volume, main = "IBM log Changes in Volume since Jan 1, 2000", xlab="Date"


#now fit regression to it with AR - this failed completely
model.ar.1 <- ar(log_return,aic=FALSE,2) # fit ar(4)
model.ar.2 <- ar(log_return)# fit ar(4)
summary(model.ar.1)
summary(model.ar.2)
model.ar.2$partialacf
plot(model1)
pred.ar2 <- predict(model1, newdata=IBM2$log_return[3:23])
model1
plot(pred.ar2)
n <-

#not using ar, but doing just an lm:
# need to order the data correctly:
train <-IBM2$log_return[which(IBM2$Date<="2011-01-03")]
test <- IBM2$log_return[which(IBM2$Date>="2011-01-03")]
test_df <- subset(IBM2,(IBM2$Date>="2011-01-05"))
#test <- subset(IBM2,IBM2$Date>"2012-01-03")
# build a lagged training data set
lag <- data.frame(Response=train[3:length(train)],
                  Predictor1 = train[2:(length(train)-1)], #lagged by 1 day
                  Predictor2 = train[1:(length(train)-2)]) #blagged by 2 days
mod2 <- lm(Response~Predictor1+Predictor2, data=lag)
xtable(summary(mod2))
```

```
test_lag <- data.frame(Predictor1 = test[2:(length(test)-1)], #lagged by 1 day
                       Predictor2 = test[1:(length(test)-2)]) #blagged by 2 days
allignment <- cbind(test_df,test_lag)
allignment$predicted <- predict(mod2, test_lag)

ts.actual <- zoo(allignment$log_return, allignment$Date)
ts.forecast <- zoo(allignment$predicted,allignment$Date)
ts.compare <- cbind(ts.actual,ts.forecast)
ts.test<- ts.compare[150:200]
plot(ts.test, plot.type="single", col= c("black", "red"),lty=1:1,pch=16,main="Model_
lty=1:1)
abline(h=0, v=0, col = "gray60",lty=2)
library(lattice)
ts.test<- ts.compare[150:200]
xyplot(ts.test, type = "o", lty=0:1, panel ="panel.superpose", screens=1, auto.key=TR



test_returns(dates=allignment$Date,actual=allignment$log_return,predicted=allignment$

#test case to make sure that the model is fit right and all of the
mod2$fitted.values[1:3]
act[1:3]
mod2$coefficients
-0.017391743 *(-0.0116064093) + (-0.0444074158)*(-0.004395611)+0.0001715389

require(rjson)
require(plyr)
#######This part is Jared's code [I had code, it worked, and then for some reason it
# the location of the data
dataPath <- "http://getglue-data.s3.amazonaws.com/getglue_sample.tar.gz"
# build a connection that can decompress the file
theCon <- gzcon(url(dataPath))
# read 10 lines of the data
n.rows <- 10
theLines <- readLines(theCon, n=n.rows)
# check its structure
str(theLines)
# notice the first element is different than the rest
theLines[1]
# use fromJSON on each element of the vector, except the first
theRead <- lapply(theLines[-1], fromJSON)
# turn it all into a data.frame
theData <- ldply(theRead, as.data.frame)
# see how we did
View(theData)

# Now do it with 5000 lines of data
n.rows <- 10001 #one more because this is not a real row
theLines <- readLines(theCon, n=n.rows)
theRead <- lapply(theLines[-1], fromJSON)
# turn it all into a data.frame
data <- ldply(theRead, as.data.frame)
# save disk image so that if we crash we don't need to download everything again
save.image("~/Dropbox/Fall_2012/Intro_to_Data_Science/Intro_to_Data_Science_HW/R/get


# different actuons
unique(data$action)
```

```
#number of users
unique(data$userId)
#to calculate favorite movie - just get # of likes
data$action <- as.character(data$action)
likes.data <- subset(data,data$action=="Liked") #get only likes
likes.movie.data <- subset(likes.data, likes.data$modelName=="movies")#get only movi
likes.tv.data <- subset(likes.data, likes.data$modelName=="tv_shows")#get only movie
colnames(likes.movie.data)
top<-summary(likes.movie.data$title)
#useful comments
useful.data <-subset(data,data$useful=="1")
useful.data$userId
#correlation between number of movie likes and number of tv shows
# will need the aggregate function here

#prepare data
library(reshape)
test <- cbind(data$userId)
movie.likes<-as.data.frame(table(likes.movie.data$userId))
tv.likes<-as.data.frame(table(likes.tv.data$userId))
both <- merge(movie.likes,tv.likes, by="row.names")[,c(2,3,5)]
colnames(both) <- c("userId", "movie", "tv")

#do more users rank movies or tv shows? (minimum of 1)
length(both$tv[which(both$tv>0)])
length(both$movie[which(both$movie>0)])
length(both$tv[which(both$tv>0 & both$movie==0)])
length(both$movie[which(both$movie>0 & both$tv==0)])

# correlation between number of tv likes and movie likes for users with more than on
both <- subset(both, both$movie>1 & both$tv>1 & both$movie<50 & both$tv<50)
    #plot(log(both$movie)~log(both$tv))
mod1<-lm(both$tv~both$movie)
plot(jitter(both$tv)~jitter(both$movie), xlab="Movies rated or commented on", ylab="
plot(both$tv~both$movie, xlab="Movies rated", ylab="TVs rated", main="Movies and TV s

#more positive comments or negative comments? (this is making assumptions that are d
comments.data <- subset(data,data$action=="Comment") #get only likes
likes.data[which(lapply(likes.data$userID, function(x) comments.data$userId))]
likes.data[which(match(likes.data$userID=="rollyp"))]
lapply(likes.data$userID,function(x) match(x, comments.data$userID))
#huge struggle couldn't do this, but should be able to... I'll come to office hrs.
```