EDA Lesson 5

Tom Gocken

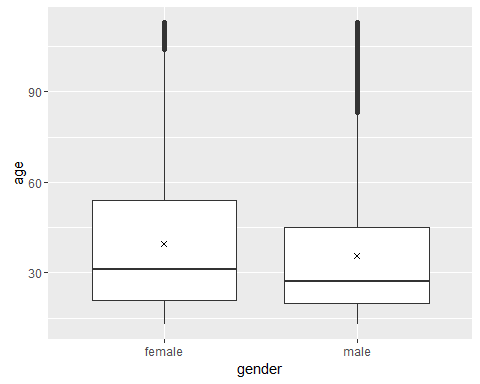
Friday, January 29, 2016

### Third Qualitative Variable

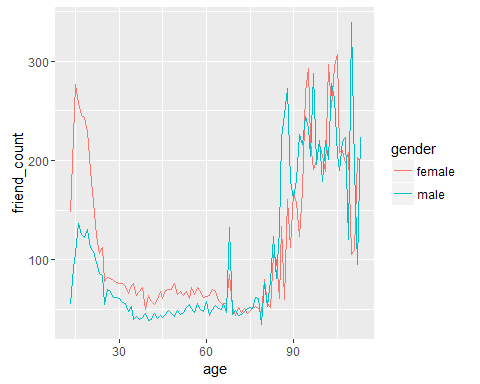
# https://s3.amazonaws.com/udacity-hosted-downloads/ud651/pseudo\_facebook.tsv  
# Set directory, load data and ggplot2 package  
setwd("C:/Projects/UD651")  
list.files()

## [1] "EDA\_lesson3.docx" "EDA\_lesson3.html" "EDA\_lesson3.Rmd"   
## [4] "EDA\_lesson4.docx" "EDA\_lesson4.html" "EDA\_lesson4.Rmd"   
## [7] "EDA\_lesson5.docx" "EDA\_lesson5.html" "EDA\_lesson5.Rmd"   
## [10] "nci.tsv" "pseudo\_facebook.tsv" "UD651.Rproj"   
## [13] "yogurt.csv"

pf <- read.csv('pseudo\_facebook.tsv', sep = '\t')  
library(ggplot2)  
  
# Boxplot with mean  
ggplot(aes(x = gender, y = age),  
 data = subset(pf, !is.na(gender))) + geom\_boxplot() +  
 stat\_summary(fun.y = mean, geom = 'point', shape = 4)



# Friend count by age, gender  
ggplot(aes(x = age, y = friend\_count),  
 data = subset(pf, !is.na(gender))) +  
 geom\_line(aes(color = gender), stat = 'summary', fun.y = median)



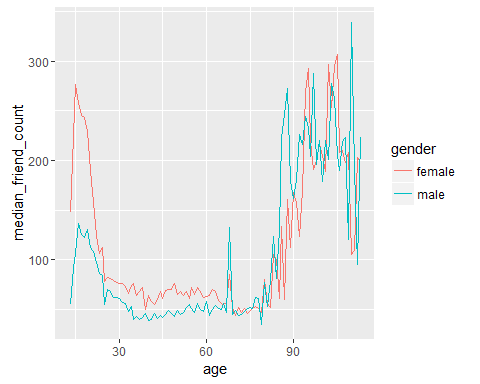
# Alt method by creating dataset in dplyr, chaining functions with %>%  
library(dplyr)

##   
## Attaching package: 'dplyr'  
##   
## The following objects are masked from 'package:stats':  
##   
## filter, lag  
##   
## The following objects are masked from 'package:base':  
##   
## intersect, setdiff, setequal, union

pf.fc\_by\_age\_gender <- pf %>%  
 filter(!is.na(gender)) %>%  
 group\_by(age, gender) %>%  
 summarise(mean\_friend\_count = mean(friend\_count),  
 median\_friend\_count = median(friend\_count),  
 n = n()) %>%  
 ungroup() %>% # summarise ungroups gender layer, need to further ungroup age layer  
 arrange(age)

### Plotting Conditional Summaries

ggplot(aes(x = age, y = median\_friend\_count),  
 data = pf.fc\_by\_age\_gender) +  
 geom\_line(aes(color = gender))



### Thinking in Ratios

# Creating dataset allows inspection of variables or further operations

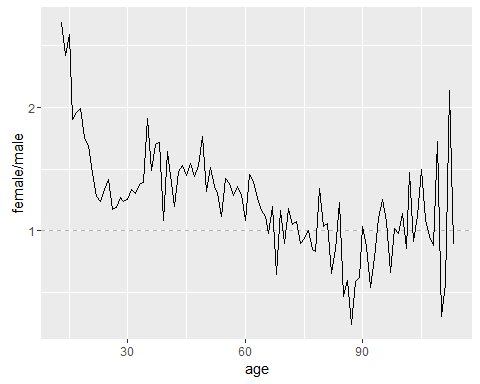
### Reshaping Data

# install.packages('reshape2')  
library(reshape2)  
  
pf.fc\_by\_age\_gender.wide <- dcast(pf.fc\_by\_age\_gender,  
 age ~ gender,  
 value.var = 'median\_friend\_count')  
  
head(pf.fc\_by\_age\_gender.wide)

## age female male  
## 1 13 148.0 55.0  
## 2 14 224.0 92.5  
## 3 15 276.0 106.5  
## 4 16 258.5 136.0  
## 5 17 245.5 125.0  
## 6 18 243.0 122.0

### Ratio Plot

# Plot females to males  
ggplot(aes(x = age, y = female / male),  
 data = pf.fc\_by\_age\_gender.wide) +  
 geom\_line() +  
 geom\_hline(yintercept = 1, alpha = 0.3, linetype = 2)



### Third Quantitative Variable

pf$year\_joined <- floor(2014 - pf$tenure / 365)

### Cut a Variable

summary(pf$year\_joined)

## Min. 1st Qu. Median Mean 3rd Qu. Max. NA's   
## 2005 2012 2012 2012 2013 2014 2

table(pf$year\_joined)

##   
## 2005 2006 2007 2008 2009 2010 2011 2012 2013 2014   
## 9 15 581 1507 4557 5448 9860 33366 43588 70

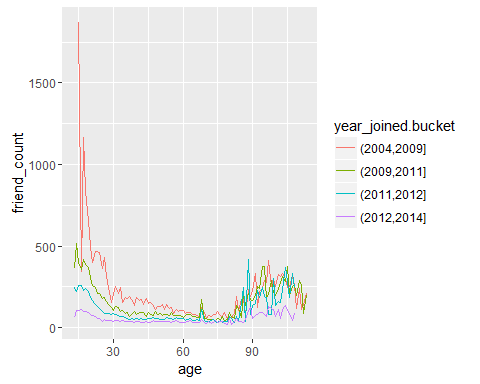
pf$year\_joined.bucket <- cut(pf$year\_joined,  
 c(2004, 2009, 2011, 2012, 2014))

### Plotting It All Together

table(pf$year\_joined.bucket, useNA = 'ifany')

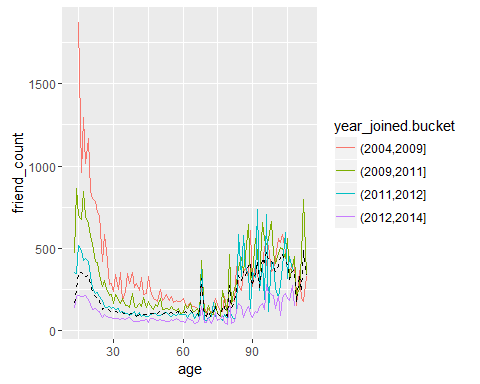
##   
## (2004,2009] (2009,2011] (2011,2012] (2012,2014] <NA>   
## 6669 15308 33366 43658 2

ggplot(aes(x = age, y = friend\_count),  
 data = subset(pf, !is.na(year\_joined.bucket))) +  
 geom\_line(aes(color = year\_joined.bucket),  
 stat = 'summary', fun.y = median)



### Plot the Grand Mean

ggplot(aes(x = age, y = friend\_count),  
 data = subset(pf, !is.na(year\_joined.bucket))) +  
 geom\_line(aes(color = year\_joined.bucket),  
 stat = 'summary', fun.y = mean) +  
 geom\_line(stat = 'summary', fun.y = mean, linetype = 2)



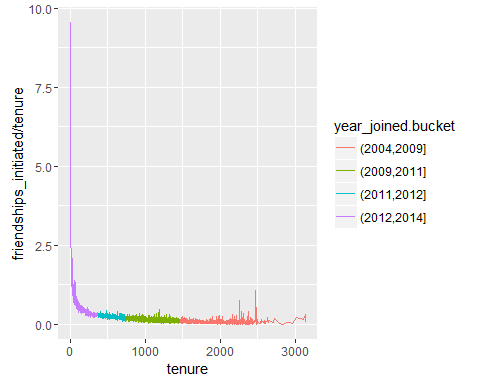
### Friending Rate

with(subset(pf, tenure >= 1), summary(friend\_count / tenure))

## Min. 1st Qu. Median Mean 3rd Qu. Max.   
## 0.0000 0.0775 0.2205 0.6096 0.5658 417.0000

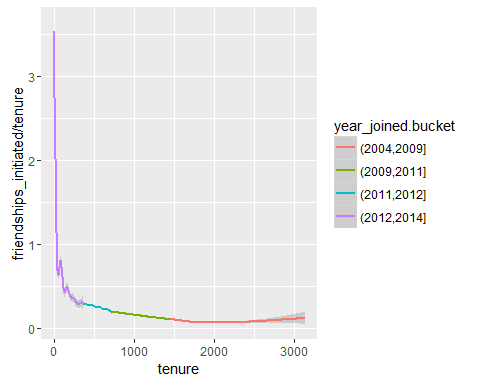
### Friendships Initiated

ggplot(aes(x = tenure, y = friendships\_initiated / tenure),  
 data = subset(pf, tenure >= 1)) +  
 geom\_line(aes(color = year\_joined.bucket),  
 stat = 'summary',  
 fun.y = mean)

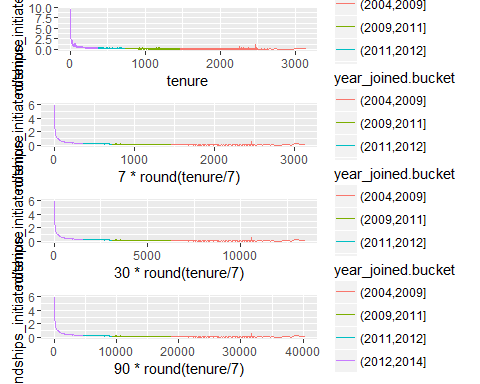


### Bias Variance Trade off Revisited

library(gridExtra)  
  
p1 <- ggplot(aes(x = tenure, y = friendships\_initiated / tenure),  
 data = subset(pf, tenure >= 1)) +  
 geom\_line(aes(color = year\_joined.bucket),  
 stat = 'summary', fun.y = mean)  
  
p2 <- ggplot(aes(x = 7 \* round(tenure/7), y = friendships\_initiated / tenure),  
 data = subset(pf, tenure >= 1)) +  
 geom\_line(aes(color = year\_joined.bucket),  
 stat = 'summary', fun.y = mean)  
  
p3 <- ggplot(aes(x = 30 \* round(tenure/7), y = friendships\_initiated / tenure),  
 data = subset(pf, tenure >= 1)) +  
 geom\_line(aes(color = year\_joined.bucket),  
 stat = 'summary', fun.y = mean)  
  
p4 <- ggplot(aes(x = 90 \* round(tenure/7), y = friendships\_initiated / tenure),  
 data = subset(pf, tenure >= 1)) +  
 geom\_line(aes(color = year\_joined.bucket),  
 stat = 'summary', fun.y = mean)  
  
# Using geom\_smooth  
ggplot(aes(x = tenure, y = friendships\_initiated / tenure),  
 data = subset(pf, tenure >= 1)) +  
 geom\_smooth(aes(color = year\_joined.bucket))



grid.arrange(p1, p2, p3, p4, ncol = 1)



### Introducing the Yogurt Dataset

# https://s3.amazonaws.com/udacity-hosted-downloads/ud651/yogurt.csv

### Histograms Revisited

setwd("C:/Projects/UD651")  
yo <- read.csv('yogurt.csv')  
str(yo)

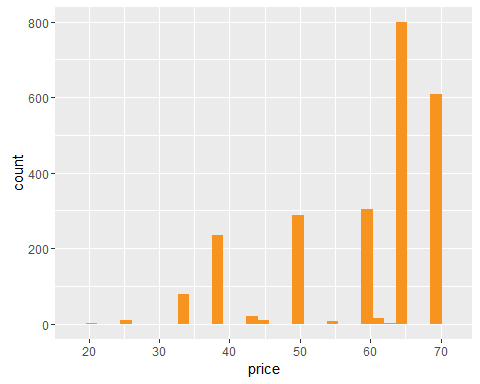
## 'data.frame': 2380 obs. of 9 variables:  
## $ obs : int 1 2 3 4 5 6 7 8 9 10 ...  
## $ id : int 2100081 2100081 2100081 2100081 2100081 2100081 2100081 2100081 2100081 2100081 ...  
## $ time : int 9678 9697 9825 9999 10015 10029 10036 10042 10083 10091 ...  
## $ strawberry : int 0 0 0 0 1 1 0 0 0 0 ...  
## $ blueberry : int 0 0 0 0 0 0 0 0 0 0 ...  
## $ pina.colada: int 0 0 0 0 1 2 0 0 0 0 ...  
## $ plain : int 0 0 0 0 0 0 0 0 0 0 ...  
## $ mixed.berry: int 1 1 1 1 1 1 1 1 1 1 ...  
## $ price : num 59 59 65 65 49 ...

# Change id from int ot factor  
yo$id <- factor(yo$id)  
str(yo)

## 'data.frame': 2380 obs. of 9 variables:  
## $ obs : int 1 2 3 4 5 6 7 8 9 10 ...  
## $ id : Factor w/ 332 levels "2100081","2100370",..: 1 1 1 1 1 1 1 1 1 1 ...  
## $ time : int 9678 9697 9825 9999 10015 10029 10036 10042 10083 10091 ...  
## $ strawberry : int 0 0 0 0 1 1 0 0 0 0 ...  
## $ blueberry : int 0 0 0 0 0 0 0 0 0 0 ...  
## $ pina.colada: int 0 0 0 0 1 2 0 0 0 0 ...  
## $ plain : int 0 0 0 0 0 0 0 0 0 0 ...  
## $ mixed.berry: int 1 1 1 1 1 1 1 1 1 1 ...  
## $ price : num 59 59 65 65 49 ...

qplot(data = yo, x = price, fill = I('#F79420'))

## `stat\_bin()` using `bins = 30`. Pick better value with `binwidth`.



### Number of Purchases

summary(yo)

## obs id time strawberry   
## Min. : 1.0 2132290: 74 Min. : 9662 Min. : 0.0000   
## 1st Qu.: 696.5 2130583: 59 1st Qu.: 9843 1st Qu.: 0.0000   
## Median :1369.5 2124073: 50 Median :10045 Median : 0.0000   
## Mean :1367.8 2149500: 50 Mean :10050 Mean : 0.6492   
## 3rd Qu.:2044.2 2101790: 47 3rd Qu.:10255 3rd Qu.: 1.0000   
## Max. :2743.0 2129528: 39 Max. :10459 Max. :11.0000   
## (Other):2061   
## blueberry pina.colada plain mixed.berry   
## Min. : 0.0000 Min. : 0.0000 Min. :0.0000 Min. :0.0000   
## 1st Qu.: 0.0000 1st Qu.: 0.0000 1st Qu.:0.0000 1st Qu.:0.0000   
## Median : 0.0000 Median : 0.0000 Median :0.0000 Median :0.0000   
## Mean : 0.3571 Mean : 0.3584 Mean :0.2176 Mean :0.3887   
## 3rd Qu.: 0.0000 3rd Qu.: 0.0000 3rd Qu.:0.0000 3rd Qu.:0.0000   
## Max. :12.0000 Max. :10.0000 Max. :6.0000 Max. :8.0000   
##   
## price   
## Min. :20.00   
## 1st Qu.:50.00   
## Median :65.04   
## Mean :59.25   
## 3rd Qu.:68.96   
## Max. :68.96   
##

length(unique(yo$price))

## [1] 20

table(yo$price)

##   
## 20 24.96 33.04 33.2 33.28 33.36 33.52 39.04 44 45.04 48.96 49.52   
## 2 11 54 1 1 22 1 234 21 11 81 1   
## 49.6 50 55.04 58.96 62 63.04 65.04 68.96   
## 1 205 6 303 15 2 799 609

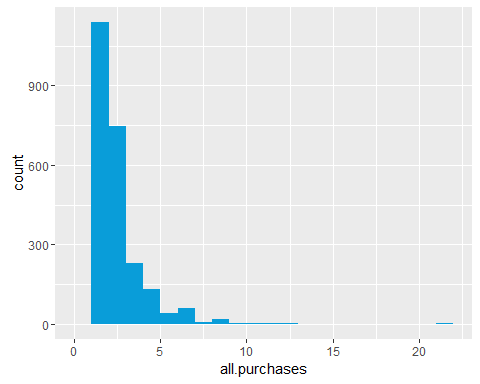
str(yo)

## 'data.frame': 2380 obs. of 9 variables:  
## $ obs : int 1 2 3 4 5 6 7 8 9 10 ...  
## $ id : Factor w/ 332 levels "2100081","2100370",..: 1 1 1 1 1 1 1 1 1 1 ...  
## $ time : int 9678 9697 9825 9999 10015 10029 10036 10042 10083 10091 ...  
## $ strawberry : int 0 0 0 0 1 1 0 0 0 0 ...  
## $ blueberry : int 0 0 0 0 0 0 0 0 0 0 ...  
## $ pina.colada: int 0 0 0 0 1 2 0 0 0 0 ...  
## $ plain : int 0 0 0 0 0 0 0 0 0 0 ...  
## $ mixed.berry: int 1 1 1 1 1 1 1 1 1 1 ...  
## $ price : num 59 59 65 65 49 ...

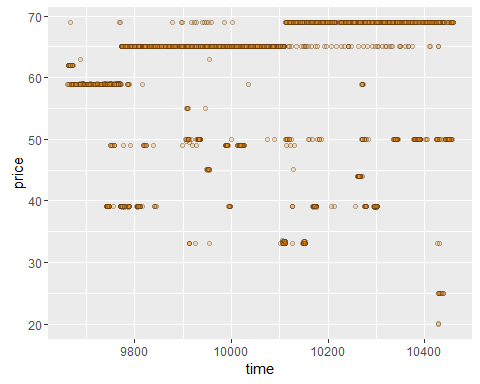
yo <- transform(yo, all.purchases = strawberry + blueberry + pina.colada + plain + mixed.berry)

### Prices Over Time

qplot(x = all.purchases, data = yo, binwidth = 1, fill = I('#099DD9'))

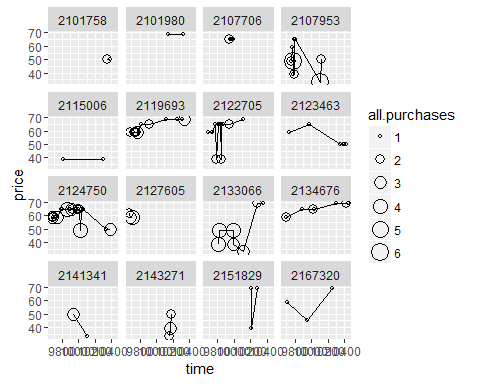


ggplot(aes(x = time, y = price), data = yo) +  
 geom\_jitter(alpha = 1/4, shape = 21, fill = I('#F79420'))



### Looking at Samples of Households

# Set seed for reproducible results  
set.seed(4230)  
# Sample 16 households  
sample.ids <- sample(levels(yo$id), 16)  
  
# Plot sample household purchases, facet by household  
# %in% loops over ids  
ggplot(aes(x = time, y = price),  
 data = subset(yo, id %in% sample.ids)) +  
 facet\_wrap( ~ id) +   
 geom\_line() +   
 geom\_point(aes(size = all.purchases), pch = 1)



### Scatterplot Matrices

# install.packages(GGally)  
library(GGally)

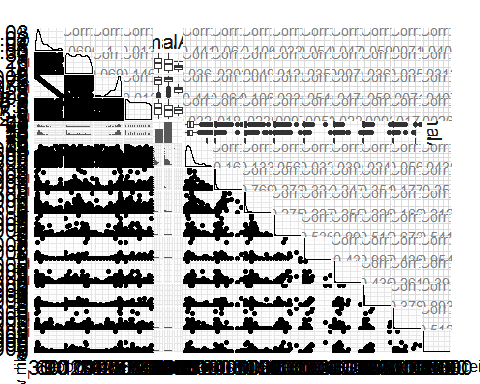
##   
## Attaching package: 'GGally'  
##   
## The following object is masked from 'package:dplyr':  
##   
## nasa

theme\_set(theme\_minimal(20))  
  
# Set seed for reproducible results  
set.seed(1836)  
pf\_subset <- pf[, c(2:15)] # variables of interest  
names(pf\_subset)

## [1] "age" "dob\_day"   
## [3] "dob\_year" "dob\_month"   
## [5] "gender" "tenure"   
## [7] "friend\_count" "friendships\_initiated"  
## [9] "likes" "likes\_received"   
## [11] "mobile\_likes" "mobile\_likes\_received"  
## [13] "www\_likes" "www\_likes\_received"

ggpairs(pf\_subset[sample.int(nrow(pf\_subset), 1000), ])

## `stat\_bin()` using `bins = 30`. Pick better value with `binwidth`.  
## `stat\_bin()` using `bins = 30`. Pick better value with `binwidth`.  
## `stat\_bin()` using `bins = 30`. Pick better value with `binwidth`.  
## `stat\_bin()` using `bins = 30`. Pick better value with `binwidth`.  
## `stat\_bin()` using `bins = 30`. Pick better value with `binwidth`.  
## `stat\_bin()` using `bins = 30`. Pick better value with `binwidth`.  
## `stat\_bin()` using `bins = 30`. Pick better value with `binwidth`.  
## `stat\_bin()` using `bins = 30`. Pick better value with `binwidth`.  
## `stat\_bin()` using `bins = 30`. Pick better value with `binwidth`.  
## `stat\_bin()` using `bins = 30`. Pick better value with `binwidth`.  
## `stat\_bin()` using `bins = 30`. Pick better value with `binwidth`.  
## `stat\_bin()` using `bins = 30`. Pick better value with `binwidth`.  
## `stat\_bin()` using `bins = 30`. Pick better value with `binwidth`.



### Even More Variables

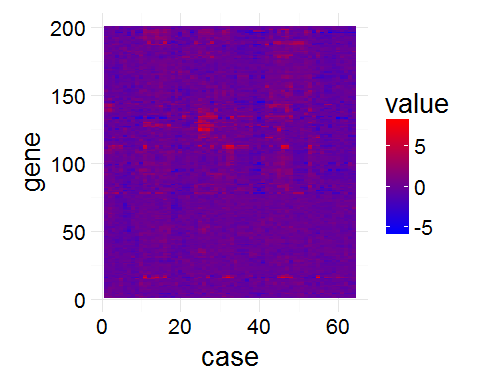
# Micro-array/Gene Expression Data  
# https://s3.amazonaws.com/udacity-hosted-downloads/ud651/nci.tsv  
setwd("C:/Projects/UD651")  
nci <- read.table('nci.tsv')  
# Rename columns 1 to 64 for simpler labeling  
colnames(nci) <- c(1:64)

### Heat Maps

# Melt data to long format  
library(reshape2)  
nci.long.samp <- melt(as.matrix(nci[1:200, ]))  
names(nci.long.samp) <- c('gene', 'case', 'value')  
head(nci.long.samp)

## gene case value  
## 1 1 1 0.300  
## 2 2 1 1.180  
## 3 3 1 0.550  
## 4 4 1 1.140  
## 5 5 1 -0.265  
## 6 6 1 -0.070

# Plot first 200 genes  
ggplot(aes(y = gene, x = case, fill = value),  
 data = nci.long.samp) +  
 geom\_tile() +  
 scale\_fill\_gradientn(colors = colorRampPalette(c('blue', 'red'))(100))



# scale\_fill\_gradientn(colors = colorRampPallet(c('blue', 'red')))  
# scale\_fill\_gradientn(colors = heat.colors(10))#colorRampPallet(c('blue', 'red'))(100))