Exercises 1

Samruddhi Somani

August 7, 2015

Question 1: Data Visualization

Install the following libraries.

```
library (ggplot2)
library(cowplot)
library(gridExtra)
```

Read in Georgia2000 data with the first row as the variables names.

```
ga2000 <-
read.csv('https://raw.githubusercontent.com/jgscott/STA380/master/data/georgi
a2000.csv', header=TRUE)
```

Calculate additional variables to facilitate analysis of vote undercount.

Diff = Ballots - Votes This is the vote undercount--the number of ballots that were not counted.

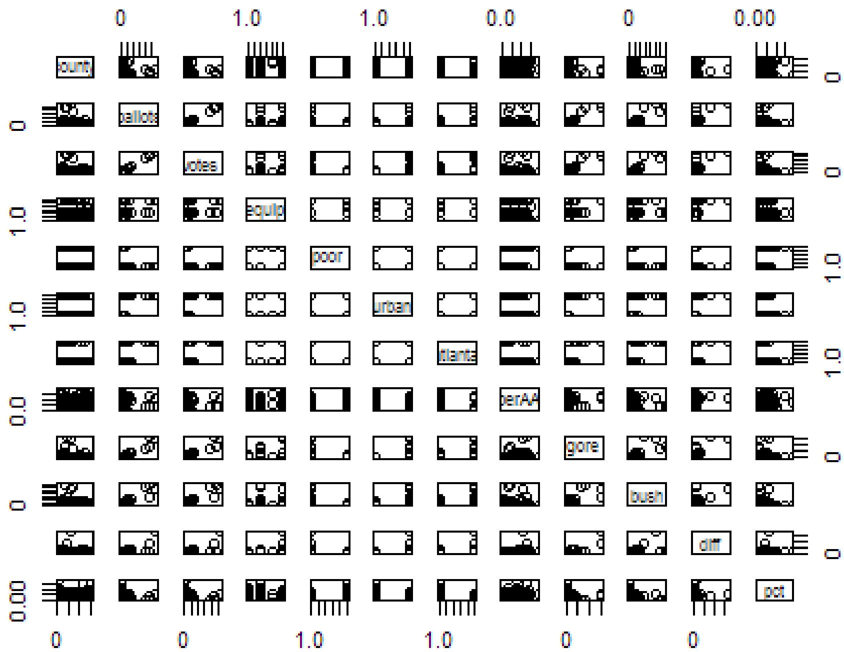
Pct = Diff/Ballots This the the undercount scaled by the number of ballots in each county.

Change categorical variables `poor`, `urban`, and `atlanta` from type *int* to type *factor* so that these variables are interpreted as discrete categorical variables rather than continuous variables. Thus, we can color code our graphics.

```
ga2000$diff<-ga2000$ballots-ga2000$votes
ga2000$poor<-as.factor(ga2000$poor)
ga2000$pct<-(ga2000$diff)/(ga2000$ballots)
ga2000$poor<-factor(ga2000$poor)
ga2000$urban<-factor(ga2000$urban)
ga2000$atlanta<-factor(ga2000$atlanta)
```

Use pairs to create a set of scatterplots relating the correlations of all the variables with each other.

```
pairs(ga2000)
```



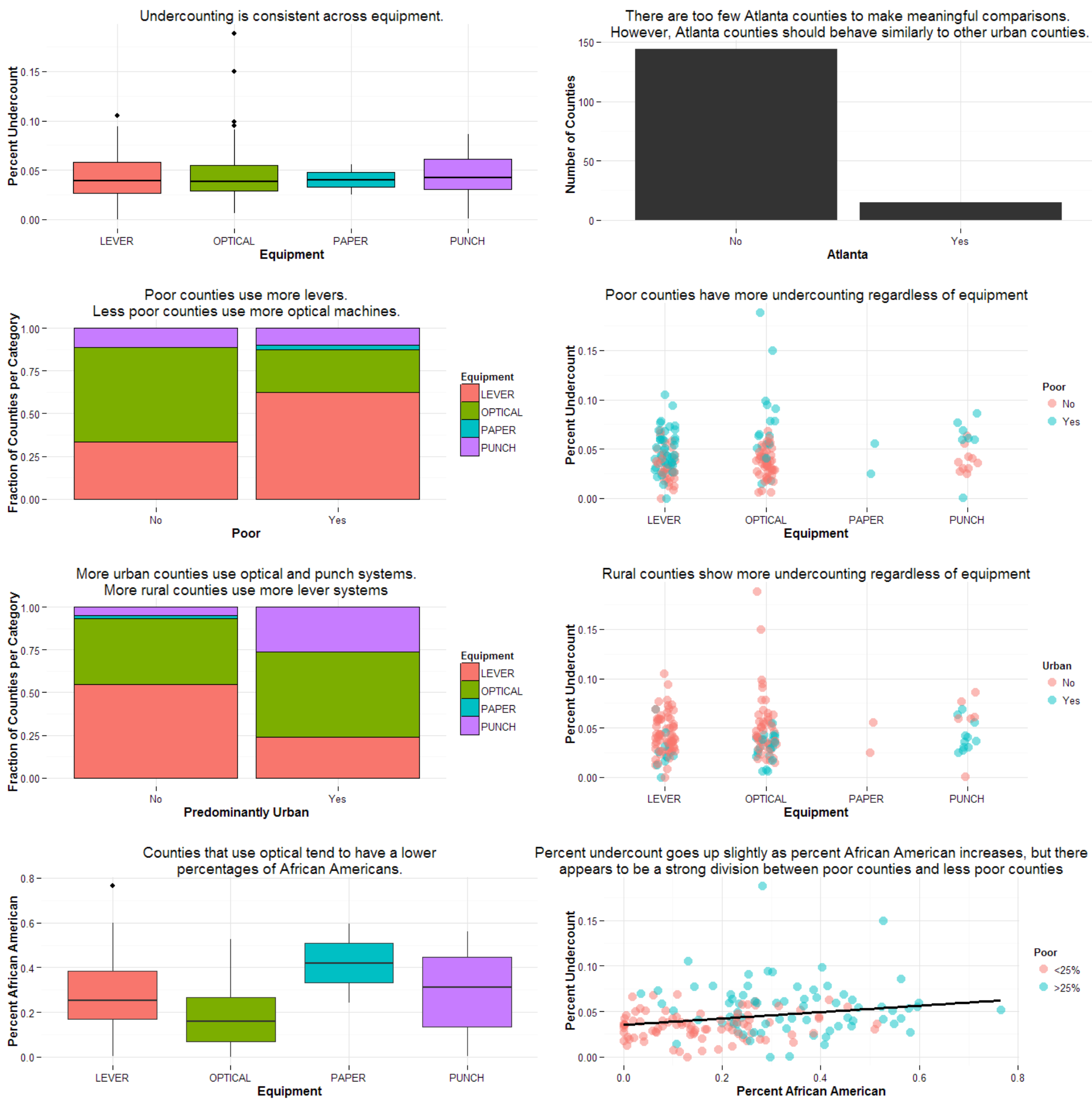
We see that `diff` is strongly correlated with `votes` and `ballots`. Thus, further analysis should focus on `pct`, the percent difference between `ballots` and `votes`. This will ensure that larger counties with more voters do not overshadow smaller counties with fewer voters in our analysis.

Upon further examination, we see that `pct` appears to be correlated with `atlanta`, `urban`, and `poor`. Different equipment types (`equip`) do not appear to have much of an effect on `pct`. However, we will examine this relationship more closely in a bivariate plot.

We create bivariate plots to better visualize particular aspects of the data. The titles below represent the key takeaways from each plot.

```
g1<-ggplot (aes(x=equip, y=pct, fill=equip),
data=ga2000)+geom_boxplot(colour='black')+theme_minimal()
+xlab("Equipment")+ylab("Percent Undercount")+ggtitle ('Undercounting is
consistent across equipment.') + guides (fill=FALSE)

g2<-ggplot (aes(x=poor, fill=equip), data=ga2000)+geom_bar(position="fill",
aes(colour="black")) + theme_minimal() + ggtitle ('Poor counties use more
levers.\nLess poor counties use more optical machines.') +
scale_fill_discrete ("Equipment") + scale_x_discrete(labels=c('No', 'Yes'))+
xlab("Poor")+ylab("Fraction of Counties per Category") +
scale_color_identity() +theme(legend.key = element_rect(colour = "black",
size = 1))

g3<-ggplot (aes(x=equip, y=pct, col=poor), data=ga2000)+geom_point(size=4,
```

```r
position = position_jitter(width=.10),
alpha=.5)+theme_minimal()+xlab("Equipment")+ylab("Percent
Undercount")+ggtitle ('Poor counties have more undercounting regardless of
equipment') + scale_color_discrete("Poor", labels=c('No', 'Yes'))

g4<-ggplot (aes(x=urban, fill=equip), data=ga2000)+geom_bar(position="fill",
aes(color='black')) + theme_minimal() + ggtitle ('More urban counties use
optical and punch systems.\nMore rural counties use more lever systems') +
scale_fill_discrete ("Equipment") + scale_x_discrete(labels=c('No', 'Yes')) +
xlab("Predominantly Urban")+ylab("Fraction of Counties per Category") +
scale_color_identity() +theme(legend.key = element_rect(colour = "black",
size = 1))

g5<-ggplot (aes(x=equip, y=pct, col=urban), data=ga2000)+geom_point(size=4,
position = position_jitter(width=.10),
alpha=.5)+theme_minimal()+xlab("Equipment")+ylab("Percent
Undercount")+ggtitle ('Rural counties show more undercounting regardless of
equipment') + scale_color_discrete("Urban", labels=c('No', 'Yes'))

g6<-ggplot (aes(x=atlanta),
data=ga2000)+geom_bar(size=4)+theme_minimal()+xlab("Equipment")+ylab("Number
of Counties")+ggtitle ('There are too few Atlanta counties to make meaningful
comparisons.\n However, Atlanta counties should behave similarly to other
urban counties.') + scale_x_discrete("Atlanta", labels=c('No', 'Yes'))

g7<-ggplot (aes(x=equip,
y=perAA,fill=equip),data=ga2000)+geom_boxplot()+theme_minimal()+xlab("Equipme
nt")+ylab("Percent African American")+ ggtitle ('Counties that use optical
tend to have a lower\npercentages of African Americans.') +
scale_fill_discrete(guide=FALSE)

g8<-ggplot (aes(x=perAA,
y=pct),data=ga2000)+geom_point(aes(color=poor),size=4,
alpha=.5)+theme_minimal()+xlab("Percent African American")+ylab("Percent
Undercount")+ggtitle ('Percent undercount goes up slightly as percent African
American increases, but there\nappears to be a strong division between poor
counties and less poor counties') +geom_smooth(se=FALSE, method='lm',
colour='black', size=1.1)+ scale_colour_discrete
("Poor",labels=c('<25%','>25%'))

grid.arrange(g1,g6,g2,g3,g4,g5,g7,g8,ncol=2)
```



Conclusion

We see that poor and rural counties are more likely to use different kinds of voting equipment than rich counties. However, certain kinds of voting equipment are not associated with higher undercount percentages. Moreover, poor and rural areas appear to suffer more undercounting *regardless* of equipment. Percentage of African Americans does not explain anything after accounting for poverty.

Question 2

Import libraries and source returns function.

```
library(mosaic)
library(foreach)
```

```
library(fImport)


YahooPricesToReturns = function(series) {
    mycols = grep('Adj.Close', colnames(series))
    closingprice = series[,mycols]
    N = nrow(closingprice)
    percentreturn = as.data.frame(closingprice[2:N,]) /
as.data.frame(closingprice[1:(N-1),]) - 1
    mynames = strsplit(colnames(percentreturn), '.', fixed=TRUE)
    mynames = lapply(mynames, function(x) return(paste0(x[1], ".PctReturn")))
    colnames(percentreturn) = mynames
    as.matrix(na.omit(percentreturn))
}
```

Import prices and calculate returns.

```
assets=c("SPY", "TLT", "LQD", "EEM", "VNQ")
prices = yahooSeries(assets, from='2010-01-01', to='2015-07-30')
returns = YahooPricesToReturns (prices)
```

Calculate standard variance to determine how much returns vary across assets. Calculate betas to determine how risky an investment is compared to the market (taken to be SPY).

```
sd<-apply(returns,2,sd)
beta<-apply(returns,2, function (x) coef(summary(lm(x~returns[,1])))[2])
rbind(sd,beta)

##      SPY.PctReturn TLT.PctReturn LQD.PctReturn EEM.PctReturn VNQ.PctReturn
## sd      0.00977304   0.009694163   0.003548176    0.01427438    0.01263622
## beta    1.00000000  -0.547628662  -0.038198270    1.24343172    1.02949742
```

Larger betas and larger standard deviations both indicate higher risk (but also the possibility for higher returns). We see that betas and standard deviations by and large express the same information about these assets' riskiness. EEM is by far the riskiest: It has the highest beta and the highest standard deviation. LQD is the least risky: It has the lowest beta and the lowest standard deviation. The TLT betas and standard deviations do not line up: Although TLT returns are about as volatile as SPY returns, TLT moves in the opposite direction of SPY (which we have taken to represent the market).

Equal Weight Portfolio

Set the seed to ensure reproducibility.

```
set.seed(1234)
```

Run a simulation of 5000 trading months.

```
sim1 = foreach(i=1:5000, .combine='rbind') %do% {
    totalwealth = 100000 #Total wealth is $100,000
    weights = c(0.2, 0.2, 0.2, 0.2, 0.2) #Weight each asset equally.
```

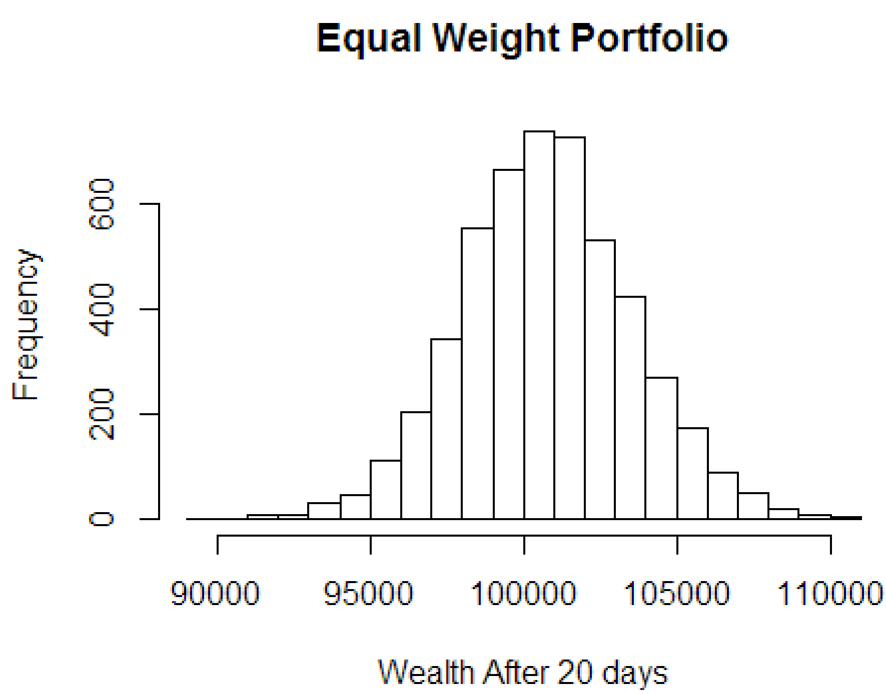
```
    holdings = weights * totalwealth #Create a vector that tracks wealth in
each asset. Reset for each 'month'
    wealthtracker = rep(0, 20) # Set up a placeholder to track total wealth
for each day.
    for(today in 1:20) {
        return.today = resample(returns, 1, orig.ids=FALSE) #Choose a random
day of returns for each asset
        holdings = holdings + holdings*return.today #Calculate new holdings
        totalwealth = sum(holdings) #Sum holdings
        wealthtracker[today] = totalwealth #Add new holdings to monthly
wealth tracker.
        holdings = weights * totalwealth #Rebalance holdings each night to
reflect weights.
    }
    wealthtracker #return wealthtracker to sim1
}
```

Examine results. After 20 days, we average $100,776 total wealth, with a minimum of $89,222 and a maximum of $110,483. Similarly, our profits range from -$10,778 to $10,482, averaging $775. We lose no more than $4689 95% of the time (the 5% value at risk). On the other hand, we will make more than $5496 5% of the time.

```
summary(sim1)

##        V1               V2               V3               V4
##  Min.   : 95734   Min.   : 94337   Min.   : 93622   Min.   : 93461
##  1st Qu.: 99738   1st Qu.: 99595   1st Qu.: 99486   1st Qu.: 99396
##  Median :100063   Median :100097   Median :100157   Median :100201
##  Mean   :100047   Mean   :100098   Mean   :100146   Mean   :100190
##  3rd Qu.:100379   3rd Qu.:100623   3rd Qu.:100803   3rd Qu.:100977
##  Max.   :104164   Max.   :104980   Max.   :106159   Max.   :106188
##        V5               V6               V7               V8
##  Min.   : 93407   Min.   : 92826   Min.   : 93604   Min.   : 92693
##  1st Qu.: 99328   1st Qu.: 99299   1st Qu.: 99266   1st Qu.: 99222
##  Median :100220   Median :100279   Median :100320   Median :100356
##  Mean   :100226   Mean   :100270   Mean   :100318   Mean   :100360
##  3rd Qu.:101130   3rd Qu.:101227   3rd Qu.:101369   3rd Qu.:101497
##  Max.   :106967   Max.   :107181   Max.   :108205   Max.   :107842
##        V9              V10              V11              V12
##  Min.   : 92757   Min.   : 93003   Min.   : 92393   Min.   : 91204
##  1st Qu.: 99204   1st Qu.: 99189   1st Qu.: 99144   1st Qu.: 99106
##  Median :100385   Median :100423   Median :100432   Median :100459
##  Mean   :100402   Mean   :100435   Mean   :100458   Mean   :100495
##  3rd Qu.:101601   3rd Qu.:101710   3rd Qu.:101778   3rd Qu.:101881
##  Max.   :107877   Max.   :107940   Max.   :108270   Max.   :108425
##       V13              V14              V15              V16
##  Min.   : 91555   Min.   : 91523   Min.   : 91177   Min.   : 89761
##  1st Qu.: 99115   1st Qu.: 99042   1st Qu.: 99035   1st Qu.: 99006
##  Median :100517   Median :100601   Median :100603   Median :100624
##  Mean   :100529   Mean   :100560   Mean   :100598   Mean   :100622
```

```
##  3rd Qu.:101979   3rd Qu.:102015   3rd Qu.:102119   3rd Qu.:102220  
##  Max.   :109741   Max.   :109776   Max.   :110629   Max.   :110883  
##       V17              V18              V19              V20        
##  Min.   : 90069   Min.   : 88270   Min.   : 87934   Min.   : 89222  
##  1st Qu.: 99001   1st Qu.: 98945   1st Qu.: 98958   1st Qu.: 98915  
##  Median :100615   Median :100633   Median :100710   Median :100738  
##  Mean   :100645   Mean   :100677   Mean   :100734   Mean   :100776  
##  3rd Qu.:102298   3rd Qu.:102379   3rd Qu.:102475   3rd Qu.:102536  
##  Max.   :111667   Max.   :110556   Max.   :110470   Max.   :110483
```

```
hist(sim1[,20], 25, main="Equal Weight Portfolio", xlab="Wealth After 20 days")
```




```
# Profit/loss
summary(sim1-100000)
```

```
##        V1                 V2                 V3                 V4         
##  Min.   :-4265.98   Min.   :-5662.71   Min.   :-6377.7   Min.   :-6539.0  
##  1st Qu.: -262.47   1st Qu.: -404.86   1st Qu.: -513.6   1st Qu.: -603.6  
##  Median :   63.02   Median :   97.45   Median :  156.7   Median :  200.9  
##  Mean   :   47.43   Mean   :   98.13   Mean   :  146.2   Mean   :  190.1  
##  3rd Qu.:  378.99   3rd Qu.:  622.54   3rd Qu.:  802.5   3rd Qu.:  977.2  
##  Max.   : 4164.20   Max.   : 4979.97   Max.   : 6158.9   Max.   : 6187.8  
##        V5                V6                V7                V8         
##  Min.   :-6593.4   Min.   :-7173.6   Min.   :-6396.1   Min.   :-7306.9  
##  1st Qu.: -671.7   1st Qu.: -700.7   1st Qu.: -733.9   1st Qu.: -778.4  
##  Median :  220.2   Median :  278.9   Median :  320.4   Median :  355.6
```

```
##  Mean   :  226.3   Mean   :  270.4   Mean   :  317.7   Mean   :  360.4  
##  3rd Qu.: 1130.0   3rd Qu.: 1226.6   3rd Qu.: 1369.4   3rd Qu.: 1497.0  
##  Max.   : 6967.1   Max.   : 7181.2   Max.   : 8204.7   Max.   : 7842.4  
##        V9                V10               V11               V12        
##  Min.   :-7242.6   Min.   :-6996.9   Min.   :-7607.3   Min.   :-8796.2  
##  1st Qu.: -795.7   1st Qu.: -811.5   1st Qu.: -855.6   1st Qu.: -894.3  
##  Median :  385.1   Median :  423.0   Median :  431.6   Median :  458.7  
##  Mean   :  401.9   Mean   :  434.8   Mean   :  458.3   Mean   :  495.2  
##  3rd Qu.: 1601.3   3rd Qu.: 1710.3   3rd Qu.: 1778.1   3rd Qu.: 1880.5  
##  Max.   : 7877.0   Max.   : 7940.4   Max.   : 8270.1   Max.   : 8425.3  
##       V13               V14               V15               V16         
##  Min.   :-8445.1   Min.   :-8477.1   Min.   :-8823.0   Min.   :-10238.9  
##  1st Qu.: -884.6   1st Qu.: -958.0   1st Qu.: -964.9   1st Qu.:  -994.3  
##  Median :  517.1   Median :  600.7   Median :  603.4   Median :   624.1  
##  Mean   :  529.0   Mean   :  560.2   Mean   :  598.4   Mean   :   621.5  
##  3rd Qu.: 1978.7   3rd Qu.: 2015.1   3rd Qu.: 2118.6   3rd Qu.:  2220.0  
##  Max.   : 9740.5   Max.   : 9775.5   Max.   :10629.1   Max.   : 10883.4  
##       V17               V18                V19          
##  Min.   :-9930.7   Min.   :-11729.8   Min.   :-12066.2  
##  1st Qu.: -999.4   1st Qu.: -1054.5   1st Qu.: -1041.6  
##  Median :  614.6   Median :   633.3   Median :   710.3  
##  Mean   :  645.0   Mean   :   676.6   Mean   :   734.3  
##  3rd Qu.: 2297.6   3rd Qu.:  2378.6   3rd Qu.:  2475.1  
##  Max.   :11667.4   Max.   : 10556.1   Max.   : 10470.2  
##       V20          
##  Min.   :-10778.2  
##  1st Qu.: -1085.3  
##  Median :   738.2  
##  Mean   :   775.5  
##  3rd Qu.:  2535.7  
##  Max.   : 10482.8
```

```
hist(sim1[,20]- 100000, main="Equal Weight Portfolio", xlab="Profits After 20 days")
```


```
# Calculate 5% value at risk
quantile(sim1[,20], 0.05) - 100000

##        5%
## -3689.445

quantile(sim1[,20], 0.95) - 100000

##      95%
## 5496.288
```

Safe Portfolio

Portfolio beta is a weighted average of the component asset betas. Thus, to create a safe portfolio, we will choose low-risk assets (assets with low betas and low standard deviations) and use higher weights on the safer assets.

We choose 85% of LQD, the asset with the lowest beta and the lowest standard deviation (both almost zero). The returns of this asset do not vary much at all, and they vary with little with the market. This is a very safe asset.

For the remaining 15%, we choose 10% TLT and 5% SPY. These two assets have similar standard deviations, meaning their returns vary about the same amount. However, TLT has a negative beta of about half the magnitude of SPY's positive beta. Thus, having twice as much TLT as SPY should create an effective hedge.

We first create a second dataset with just the returns from the three assets we will use for this portfolio. We also set the seed to ensure reproducibility.

```
safe<-returns[,c(1:3)]
set.seed(1234)
```

As we did for the equal weights portflio, run a 5000 month bootstrap using this risky portfolio.

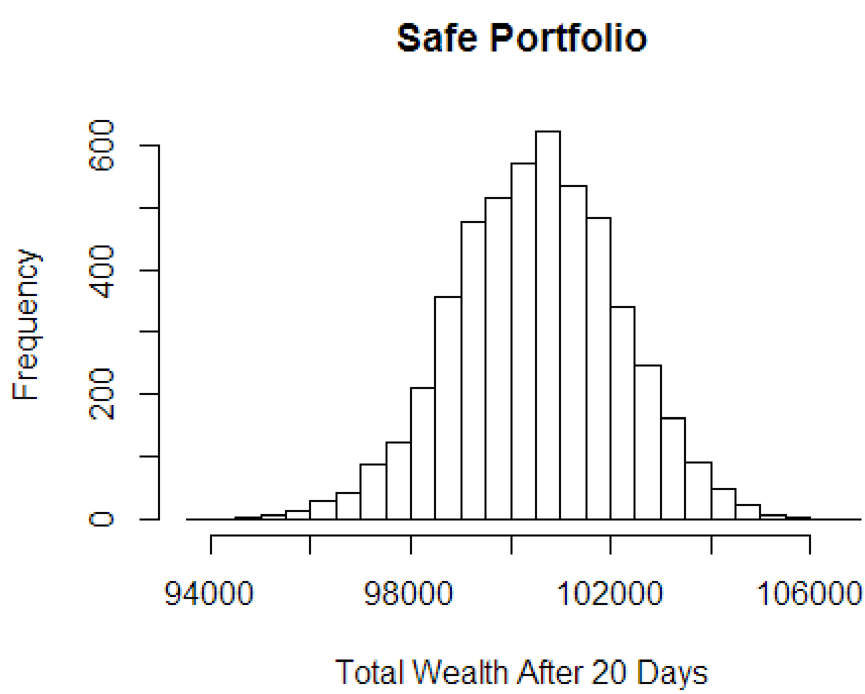
```
sim2 = foreach(i=1:5000, .combine='rbind') %do% {
    totalwealth = 100000
    weights = c(.05, .10, .85)
    holdings = weights * totalwealth
    wealthtracker = rep(0, 20)
    for(today in 1:20) {
        return.today = resample(safe, 1, orig.ids=FALSE)
        holdings = holdings + holdings*return.today
        totalwealth = sum(holdings)
        wealthtracker[today] = totalwealth
        holdings = weights * totalwealth #rebalance
    }
    wealthtracker
}
```

For this safe portfolio, our total wealth ranges from $93,929 to $106,560, averaging $100,522. Our profits similarly range from -$6,071 to $6,560 averaging $522. 95% of the time, we will lose no more than $2,226 (5% value at risk), and 5% of the time, we gain at least $3,194. The likely range of values of this portfolio is much smaller than our equally weighted portfolio: Our risk is lower, but so are our potential returns.

```
summary(sim2)

##        V1               V2               V3               V4        
##  Min.   : 97975   Min.   : 97565   Min.   : 96996   Min.   : 96622  
##  1st Qu.: 99819   1st Qu.: 99724   1st Qu.: 99658   1st Qu.: 99650  
##  Median :100059   Median :100077   Median :100101   Median :100128  
##  Mean   :100023   Mean   :100046   Mean   :100073   Mean   :100104  
##  3rd Qu.:100245   3rd Qu.:100389   3rd Qu.:100498   3rd Qu.:100594  
##  Max.   :101428   Max.   :102280   Max.   :102738   Max.   :102974  
##        V5               V6               V7               V8        
##  Min.   : 96558   Min.   : 96064   Min.   : 96116   Min.   : 95799  
##  1st Qu.: 99607   1st Qu.: 99578   1st Qu.: 99540   1st Qu.: 99527  
##  Median :100160   Median :100184   Median :100204   Median :100238  
##  Mean   :100126   Mean   :100151   Mean   :100178   Mean   :100208  
##  3rd Qu.:100668   3rd Qu.:100760   3rd Qu.:100835   3rd Qu.:100909  
##  Max.   :102768   Max.   :103012   Max.   :103788   Max.   :103810  
##        V9              V10              V11              V12        
##  Min.   : 95576   Min.   : 95157   Min.   : 95147   Min.   : 94785  
##  1st Qu.: 99517   1st Qu.: 99511   1st Qu.: 99499   1st Qu.: 99466  
##  Median :100264   Median :100296   Median :100292   Median :100339  
##  Mean   :100243   Mean   :100264   Mean   :100283   Mean   :100310  
```

```
##   3rd Qu.:100991   3rd Qu.:101051   3rd Qu.:101089   3rd Qu.:101157  
##   Max.   :103787   Max.   :103932   Max.   :104240   Max.   :104202  
##        V13              V14              V15              V16        
##   Min.   : 94985   Min.   : 94778   Min.   : 94733   Min.   : 94523  
##   1st Qu.: 99431   1st Qu.: 99429   1st Qu.: 99443   1st Qu.: 99427  
##   Median :100375   Median :100381   Median :100416   Median :100430  
##   Mean   :100343   Mean   :100357   Mean   :100387   Mean   :100411  
##   3rd Qu.:101217   3rd Qu.:101261   3rd Qu.:101341   3rd Qu.:101405  
##   Max.   :104835   Max.   :104567   Max.   :105031   Max.   :105478  
##        V17              V18              V19              V20        
##   Min.   : 94134   Min.   : 94119   Min.   : 94407   Min.   : 93929  
##   1st Qu.: 99437   1st Qu.: 99445   1st Qu.: 99426   1st Qu.: 99409  
##   Median :100469   Median :100479   Median :100492   Median :100539  
##   Mean   :100441   Mean   :100472   Mean   :100497   Mean   :100522  
##   3rd Qu.:101470   3rd Qu.:101518   3rd Qu.:101567   3rd Qu.:101651  
##   Max.   :106848   Max.   :106861   Max.   :107046   Max.   :106560  
```

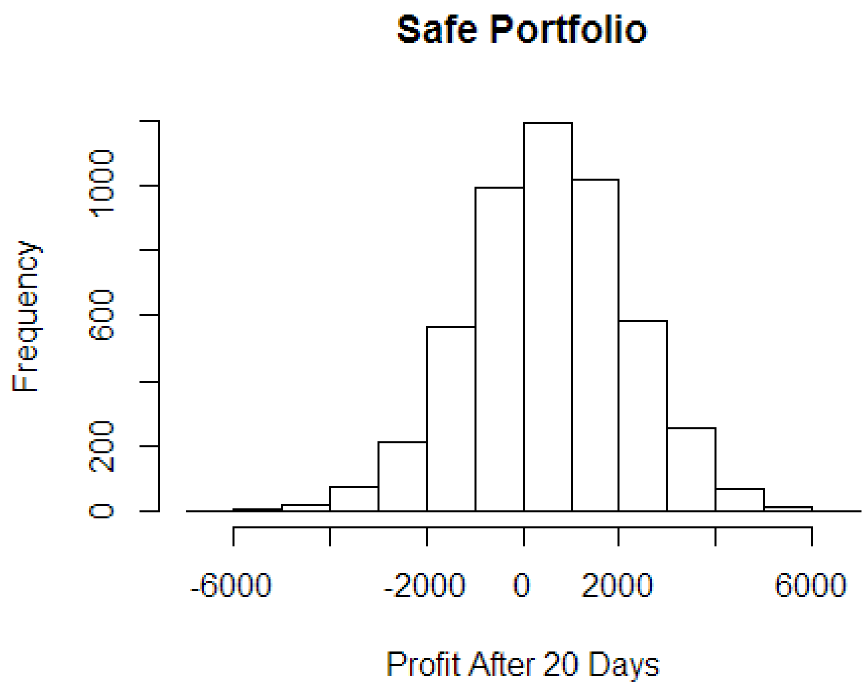
```
hist(sim2[,20], 25, main="Safe Portfolio", xlab="Total Wealth After 20 Days")
```




```
# Profit/loss
summary(sim2-100000)
```

```
##         V1                  V2                  V3                 V4         
##   Min.   :-2024.55   Min.   :-2434.51   Min.   :-3004.1   Min.   :-3378.3  
##   1st Qu.: -180.60   1st Qu.: -276.02   1st Qu.: -341.7   1st Qu.: -349.6  
##   Median :   59.32   Median :   76.78   Median :  101.3   Median :  128.0  
##   Mean   :   23.00   Mean   :   45.82   Mean   :   72.8   Mean   :  103.6  
```

```
##  3rd Qu.:  244.79   3rd Qu.:  389.19   3rd Qu.:  497.7   3rd Qu.:  594.5
##  Max.   : 1428.05   Max.   : 2279.66   Max.   : 2737.8   Max.   : 2974.4
##        V5                 V6                 V7                 V8
##  Min.   :-3441.8   Min.   :-3935.9   Min.   :-3883.8   Min.   :-4201.2
##  1st Qu.: -393.5   1st Qu.: -422.5   1st Qu.: -460.0   1st Qu.: -472.9
##  Median :  160.2   Median :  184.1   Median :  203.9   Median :  238.5
##  Mean   :  126.2   Mean   :  150.5   Mean   :  178.3   Mean   :  207.8
##  3rd Qu.:  668.3   3rd Qu.:  760.5   3rd Qu.:  835.1   3rd Qu.:  908.8
##  Max.   : 2768.2   Max.   : 3011.7   Max.   : 3787.5   Max.   : 3810.2
##        V9               V10                V11                V12
##  Min.   :-4423.7   Min.   :-4843.3   Min.   :-4853.1   Min.   :-5215.2
##  1st Qu.: -482.8   1st Qu.: -489.0   1st Qu.: -500.7   1st Qu.: -533.8
##  Median :  264.2   Median :  296.3   Median :  292.4   Median :  338.8
##  Mean   :  242.7   Mean   :  264.1   Mean   :  283.1   Mean   :  309.7
##  3rd Qu.:  991.1   3rd Qu.: 1050.8   3rd Qu.: 1089.4   3rd Qu.: 1156.6
##  Max.   : 3786.8   Max.   : 3932.5   Max.   : 4239.9   Max.   : 4202.1
##       V13               V14                V15                V16
##  Min.   :-5014.9   Min.   :-5221.5   Min.   :-5266.6   Min.   :-5477.2
##  1st Qu.: -569.4   1st Qu.: -571.0   1st Qu.: -556.7   1st Qu.: -573.3
##  Median :  374.5   Median :  381.3   Median :  416.0   Median :  430.3
##  Mean   :  343.3   Mean   :  357.3   Mean   :  386.8   Mean   :  410.9
##  3rd Qu.: 1217.4   3rd Qu.: 1261.4   3rd Qu.: 1340.7   3rd Qu.: 1404.8
##  Max.   : 4834.6   Max.   : 4567.2   Max.   : 5031.2   Max.   : 5478.4
##       V17               V18                V19                V20
##  Min.   :-5865.5   Min.   :-5880.6   Min.   :-5593.1   Min.   :-6071.3
##  1st Qu.: -563.1   1st Qu.: -554.8   1st Qu.: -574.1   1st Qu.: -590.6
##  Median :  469.3   Median :  479.2   Median :  492.0   Median :  538.6
##  Mean   :  441.2   Mean   :  472.4   Mean   :  497.4   Mean   :  522.5
##  3rd Qu.: 1469.9   3rd Qu.: 1518.1   3rd Qu.: 1566.5   3rd Qu.: 1650.9
##  Max.   : 6848.5   Max.   : 6861.3   Max.   : 7045.6   Max.   : 6560.3

hist(sim2[,20]- 100000, main="Safe Portfolio", xlab="Profit After 20 Days")
```




```
# Calculate 5% value at risk
quantile(sim2[,20], 0.05) - 100000

##        5%
## -2225.985

#Calculate 5% upside
quantile(sim2[,20],0.95) - 100000

##      95%
## 3194.347
```

Risky portfolio

Because portfolio beta is a weighted average, we will use high beta/high standard deviation assets (risky assets) to create a risky portfolio.

We weight EEM (emerging markets) by 85% as it is by far the riskiest asset. The remainder of our portfolio will be in SPY (US equities) to ensure some diversification in our portfolio.

```
risky<-returns[,c(1,4)]

set.seed(1234)

sim3 = foreach(i=1:5000, .combine='rbind') %do% {
    totalwealth = 100000
    weights = c(.15,.85)
    holdings = weights * totalwealth
```

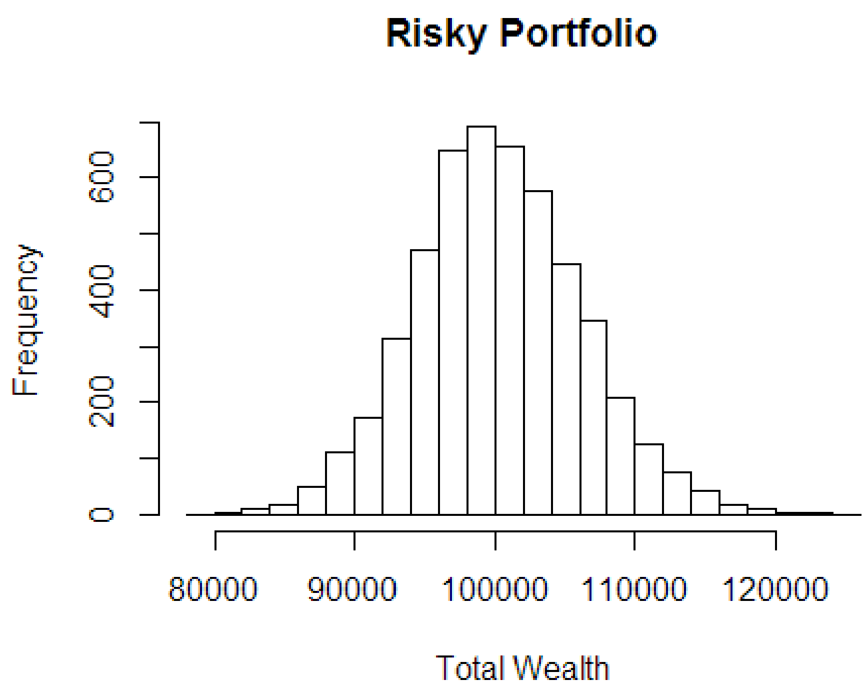
```
    wealthtracker = rep(0, 20) # Set up a placeholder to track total wealth
    for(today in 1:20) {
        return.today = resample(risky, 1, orig.ids=FALSE)
        holdings = holdings + holdings*return.today
        totalwealth = sum(holdings)
        wealthtracker[today] = totalwealth
        holdings = weights * totalwealth #rebalance
    }
    wealthtracker
}
```

For this riskier portfolio, our total wealth ranges from $79,158 to $125,348 with a mean of $100,270. Our profits range from -$20,842 to $25,348 with a mean of $269.64. 5% of the time, we lose at least $9280 (5% value at risk), and 5% of the time we gain at least $10,401. Thus this portfolio has a possibility of higher returns, but it is also far riskier than our safe or equal weights portfolios.

```
summary(sim3)

##        V1                V2                V3                V4
##  Min.   : 91936   Min.   : 88027   Min.   : 87161   Min.   : 87557
##  1st Qu.: 99341   1st Qu.: 98968   1st Qu.: 98695   1st Qu.: 98474
##  Median :100069   Median :100142   Median :100112   Median :100148
##  Mean   :100041   Mean   :100086   Mean   :100121   Mean   :100128
##  3rd Qu.:100763   3rd Qu.:101221   3rd Qu.:101544   3rd Qu.:101793
##  Max.   :106781   Max.   :108952   Max.   :110071   Max.   :111071
##        V5                V6                V7                V8
##  Min.   : 86520   Min.   : 84161   Min.   : 85818   Min.   : 84205
##  1st Qu.: 98204   1st Qu.: 98024   1st Qu.: 97915   1st Qu.: 97804
##  Median :100161   Median :100136   Median :100230   Median :100160
##  Mean   :100134   Mean   :100155   Mean   :100184   Mean   :100199
##  3rd Qu.:101920   3rd Qu.:102214   3rd Qu.:102375   3rd Qu.:102640
##  Max.   :112556   Max.   :112738   Max.   :113102   Max.   :115367
##        V9               V10               V11               V12
##  Min.   : 84401   Min.   : 83391   Min.   : 81946   Min.   : 81308
##  1st Qu.: 97600   1st Qu.: 97460   1st Qu.: 97378   1st Qu.: 97293
##  Median :100223   Median :100181   Median :100131   Median :100138
##  Mean   :100216   Mean   :100221   Mean   :100213   Mean   :100228
##  3rd Qu.:102799   3rd Qu.:103064   3rd Qu.:103165   3rd Qu.:103254
##  Max.   :116351   Max.   :116845   Max.   :118319   Max.   :118429
##       V13               V14               V15               V16
##  Min.   : 81573   Min.   : 80612   Min.   : 82140   Min.   : 79184
##  1st Qu.: 97114   1st Qu.: 96921   1st Qu.: 96851   1st Qu.: 96721
##  Median :100222   Median :100208   Median :100195   Median :100121
##  Mean   :100219   Mean   :100237   Mean   :100243   Mean   :100230
##  3rd Qu.:103447   3rd Qu.:103477   3rd Qu.:103580   3rd Qu.:103682
##  Max.   :117740   Max.   :120360   Max.   :120400   Max.   :119878
##       V17               V18               V19               V20
##  Min.   : 77808   Min.   : 77166   Min.   : 78026   Min.   : 79158
##  1st Qu.: 96667   1st Qu.: 96435   1st Qu.: 96442   1st Qu.: 96332
```

```
##   Median :100089    Median :100014    Median :100091    Median :100020
##   Mean   :100207    Mean   :100199    Mean   :100253    Mean   :100270
##   3rd Qu.:103763    3rd Qu.:103805    3rd Qu.:103987    3rd Qu.:104126
##   Max.   :119746    Max.   :123977    Max.   :124642    Max.   :125348
```

```
hist(sim3[,20], 25, main="Risky Portfolio",xlab="Total Wealth")
```

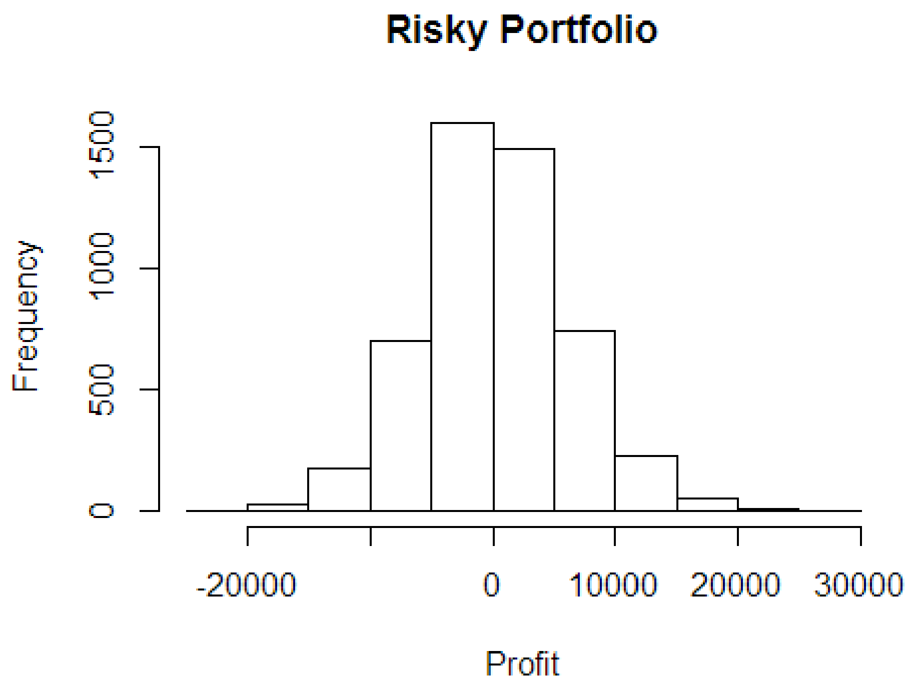



```
# Profit/loss
summary(sim3-100000)
```

```
##         V1                  V2                   V3
##   Min.   :-8063.51   Min.   :-11973.49   Min.   :-12839.3
##   1st Qu.: -658.89   1st Qu.: -1032.36   1st Qu.: -1305.5
##   Median :   68.90   Median :   141.93   Median :   111.9
##   Mean   :   41.22   Mean   :    85.62   Mean   :   121.0
##   3rd Qu.:  762.64   3rd Qu.:  1221.07   3rd Qu.:  1544.1
##   Max.   : 6781.33   Max.   :  8952.08   Max.   : 10070.8
##         V4                  V5                  V6
##   Min.   :-12442.6   Min.   :-13480.0   Min.   :-15838.6
##   1st Qu.: -1526.1   1st Qu.: -1796.5   1st Qu.: -1975.7
##   Median :   148.1   Median :   160.9   Median :   135.9
##   Mean   :   128.5   Mean   :   133.7   Mean   :   154.5
##   3rd Qu.:  1793.3   3rd Qu.:  1920.1   3rd Qu.:  2214.0
##   Max.   : 11071.1   Max.   : 12555.8   Max.   : 12738.0
##         V7                  V8                  V9
##   Min.   :-14181.6   Min.   :-15795.3   Min.   :-15599.4
##   1st Qu.: -2085.2   1st Qu.: -2196.1   1st Qu.: -2400.5
```

```
##  Median :   229.7   Median :   160.1   Median :   223.1  
##  Mean   :   183.6   Mean   :   199.2   Mean   :   216.1  
##  3rd Qu.:  2374.9   3rd Qu.:  2639.7   3rd Qu.:  2799.3  
##  Max.   : 13102.5   Max.   : 15366.6   Max.   : 16351.3  
##       V10                V11                V12          
##  Min.   :-16609.1   Min.   :-18054.3   Min.   :-18691.7  
##  1st Qu.: -2540.1   1st Qu.: -2621.6   1st Qu.: -2707.5  
##  Median :   181.1   Median :   130.6   Median :   137.8  
##  Mean   :   221.2   Mean   :   212.5   Mean   :   227.8  
##  3rd Qu.:  3064.4   3rd Qu.:  3164.6   3rd Qu.:  3254.5  
##  Max.   : 16845.3   Max.   : 18319.1   Max.   : 18429.1  
##       V13                V14                V15          
##  Min.   :-18427.3   Min.   :-19387.7   Min.   :-17859.5  
##  1st Qu.: -2885.7   1st Qu.: -3078.9   1st Qu.: -3148.7  
##  Median :   222.0   Median :   207.9   Median :   194.8  
##  Mean   :   218.8   Mean   :   237.1   Mean   :   243.4  
##  3rd Qu.:  3446.8   3rd Qu.:  3477.4   3rd Qu.:  3580.2  
##  Max.   : 17740.3   Max.   : 20359.9   Max.   : 20400.0  
##       V16                V17                V18           
##  Min.   :-20815.7   Min.   :-22191.8   Min.   :-22834.47  
##  1st Qu.: -3279.1   1st Qu.: -3333.1   1st Qu.: -3565.18  
##  Median :   121.2   Median :    88.8   Median :    14.44  
##  Mean   :   229.5   Mean   :   207.1   Mean   :   199.45  
##  3rd Qu.:  3681.8   3rd Qu.:  3762.6   3rd Qu.:  3804.55  
##  Max.   : 19878.4   Max.   : 19746.3   Max.   : 23977.27  
##       V19                 V20           
##  Min.   :-21973.60   Min.   :-20842.34  
##  1st Qu.: -3558.01   1st Qu.: -3668.44  
##  Median :    91.25   Median :    19.97  
##  Mean   :   252.64   Mean   :   269.64  
##  3rd Qu.:  3987.22   3rd Qu.:  4126.31  
##  Max.   : 24642.32   Max.   : 25348.14
```

```
hist(sim3[,20]- 100000, main="Risky Portfolio", xlab="Profit")
```




```r
# Calculate 5% value at risk and 95% upside.
quantile(sim3[,20], 0.05) - 100000

##        5%
## -9280.723

quantile(sim3[,20],0.95) - 100000

##      95%
## 10400.98
```

Summary:

Although all of these methods have similar averages, the potential risk and return differ greatly. The safe portfolio has by far the least potential risk and return, the risky by far the greatest. Equal weighted sits in the middle.

Question 3: Wine

Load libraries

```r
library(ggplot2)
library(cowplot)
```

Read in data. Create a separate data frame containing only the chemical properties. Scale the dataset.

```r
wine<-
read.csv('https://raw.githubusercontent.com/jgscott/STA380/master/data/wine.c
sv',row.names<-1)
wine_adj<-wine[,c(1:11)]
wine_adj_s<-scale(wine_adj,center=TRUE,scale=TRUE)
```

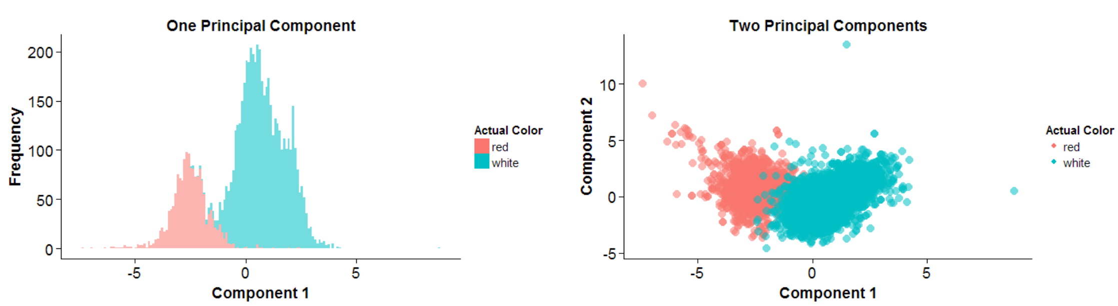
Using PCA to determine color

Run PCA and review the summary statistics. We see that it takes 7 principal components to explain 90% of the variance in the data.

```r
pca<-prcomp(wine_adj_s)
summary(pca)

## Importance of components:
##                           PC1    PC2    PC3     PC4     PC5     PC6
## Standard deviation     1.7407 1.5792 1.2475 0.98517 0.84845 0.77930
## Proportion of Variance 0.2754 0.2267 0.1415 0.08823 0.06544 0.05521
## Cumulative Proportion  0.2754 0.5021 0.6436 0.73187 0.79732 0.85253
##                            PC7     PC8     PC9   PC10    PC11
## Standard deviation     0.72330 0.70817 0.58054 0.4772 0.18119
## Proportion of Variance 0.04756 0.04559 0.03064 0.0207 0.00298
## Cumulative Proportion  0.90009 0.94568 0.97632 0.9970 1.00000
```

Plot PC1 and PC1 versus PC2. It appears that both the first principal component model and the first and second component models have similar errors distinguishing at the red/white boundary.

```r
scores = pca$x

q1<-qplot(scores[,1], fill=wine$color, xlab='Component 1', ylab='Frequency',
main="One Principal Component", binwidth=.1, alpha=.1) + scale_fill_discrete
("Actual Color")+ scale_alpha(guide=FALSE)

q2<-qplot(scores[,1],scores[,2], color=wine$color, xlab='Component 1',
ylab='Component 2', main="Two Principal Components", size=1, alpha=.001)+
scale_color_discrete ("Actual Color") + scale_size(guide=FALSE) +
scale_alpha(guide=FALSE)

plot_grid(q1,q2)
```



Using clustering to find color

Set seed.

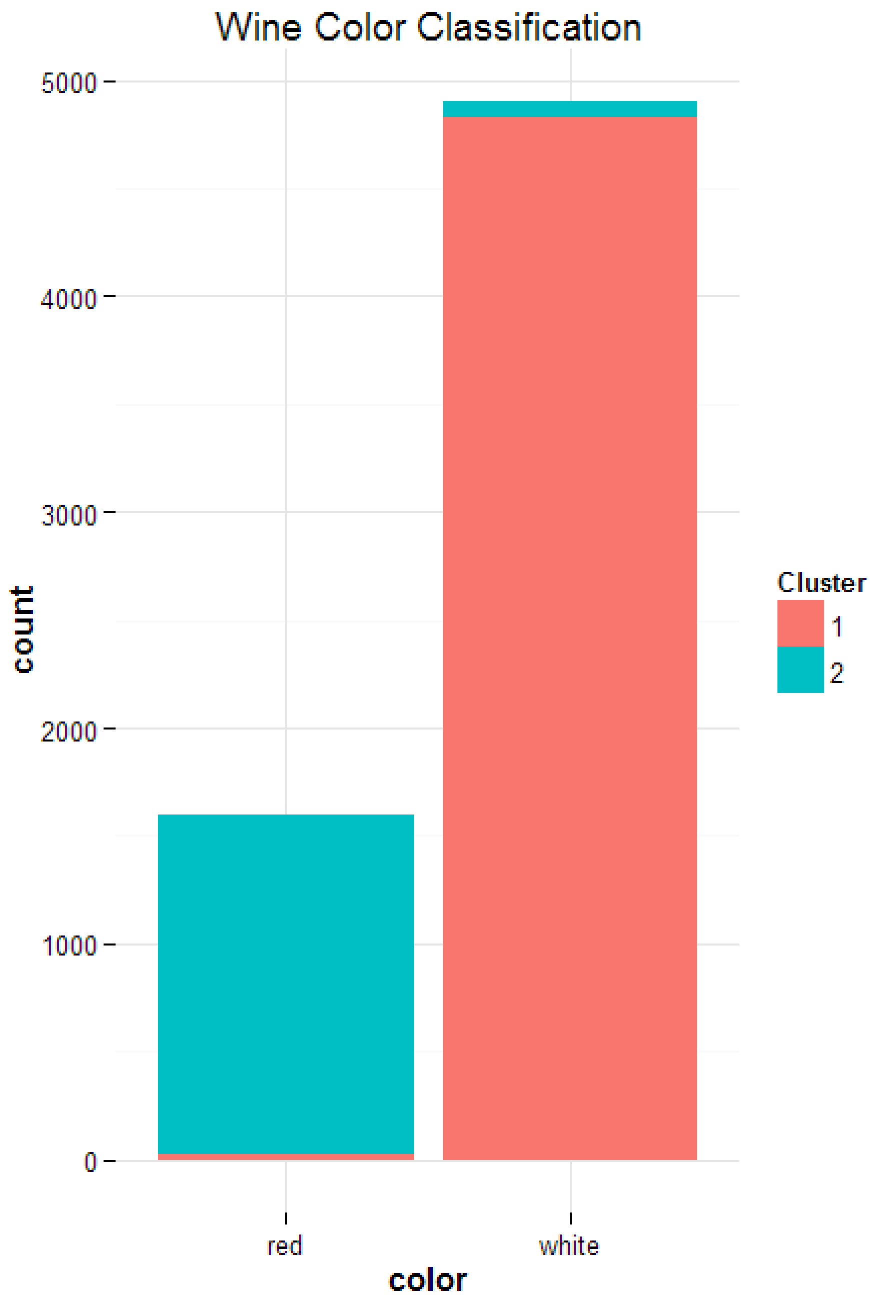
```
set.seed(78705)
```

Run k-means. Use two centers because we expect two clusters: a red cluster and a white cluster.

```
wcl<- kmeans(wine_adj_s, centers=2, nstart=50)
```

Create a plot to examine the accuracy of the k-means model. We see that cluster 2 largely maps to red wines and that cluster 1 largely maps to white wines. K-means accuracy appears to be far better than PCA.

```
ggplot (aes(x=color, fill=factor(wcl$cluster)),
data=wine)+geom_bar(position="stack") + theme_minimal() +ggtitle ('Wine Color
Classification') + scale_fill_discrete ("Cluster")
```



Create a confusion matrix and a proportion table. This model accurately predicts color 98% of the time.

```
t1 = xtabs(~wine$color + wcl$cluster)
t1

##           wcl$cluster
## wine$color    1    2
##      red     24 1575
##      white 4830   68
```

```
prop.table(t1,margin=1)

##            wcl$cluster
## wine$color          1          2
##       red   0.01500938 0.98499062
##       white 0.98611678 0.01388322
```

Final Model Choice:

K-means clustering appears to be a far superior method for determining whether a wine is red or white from its chemical properties.

Clustering to find quality

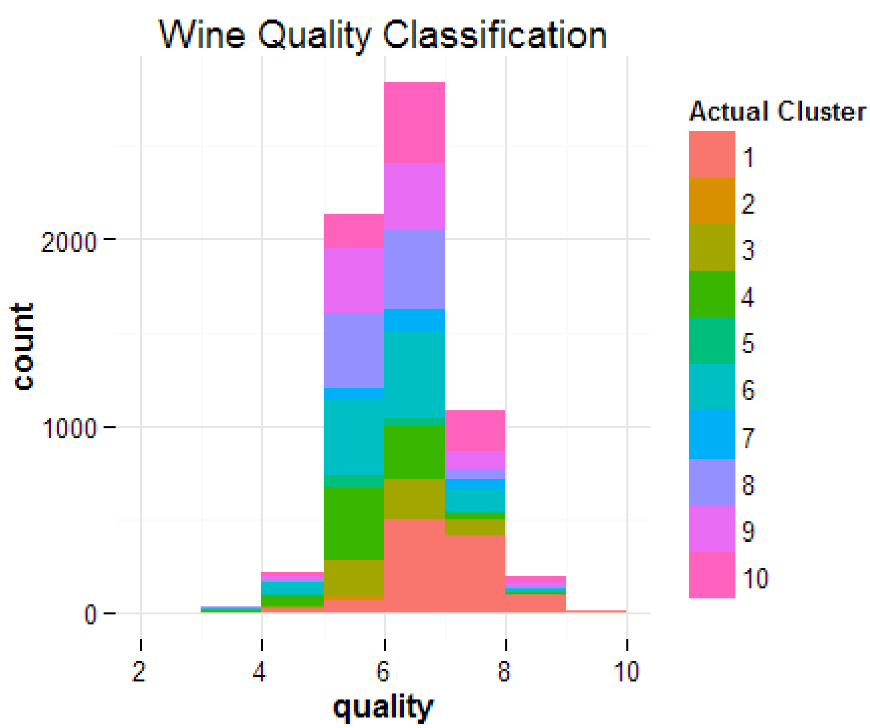
Set seed.

```
set.seed(78705)
```

Run k-means.

Create a plot to examine the accuracy of the k-means model. We see that k-means is not capable of accurately predicting quality: It predicts a variety of qualities for each actual quality.

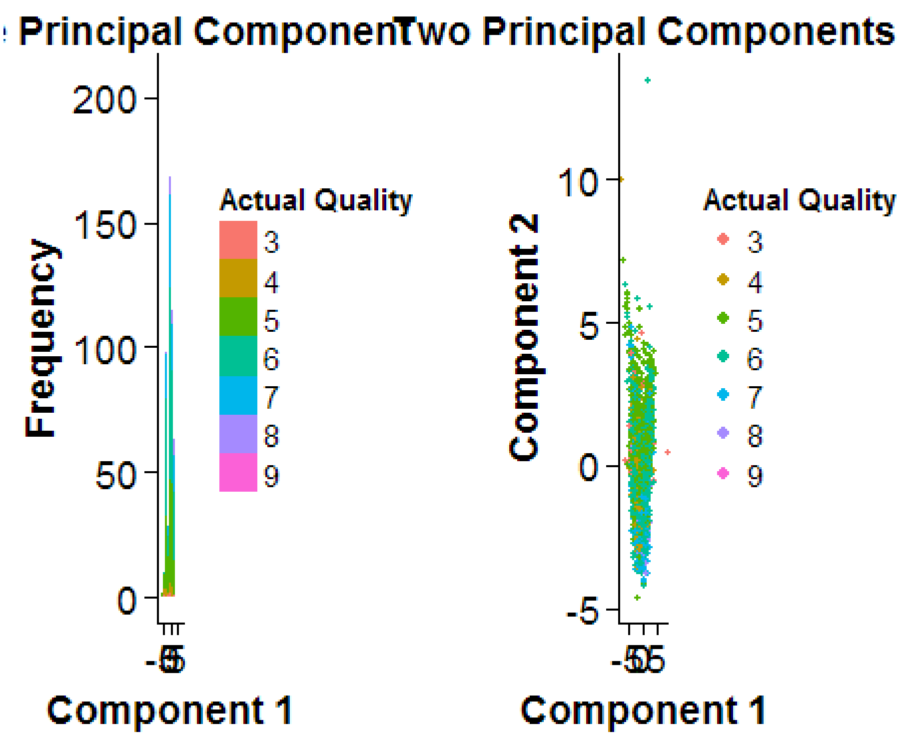
```
ggplot (aes(x=quality, fill=factor(wclq$cluster)),
data=wine)+geom_bar(position="stack", binwidth=1) + theme_minimal() +ggtitle
('Wine Quality Classification') + scale_fill_discrete ("Actual Cluster")
```



PCA to find quality

We ran PCA earlier. We will now examine how well it predicts quality.. We see that it has similar problems to k-means: Every predicted quality level contains a wide range of actual quality levels.

```
q1<-qplot(scores[,1], fill=factor(wine$quality), xlab='Component 1',
ylab='Frequency', main="One Principal Component", binwidth=.1) +
scale_fill_discrete ("Actual Quality")

q2<-qplot(scores[,1], scores[,2], color=factor(wine$quality), xlab='Component
1', ylab='Component 2', main="Two Principal Components", size=1) +
scale_color_discrete ("Actual Quality") + scale_size_identity()

plot_grid (q1,q2)
```



Summary

K-means determined color far better than Principal Component Analysis. However, it failed to accurately determine quality. PCA was also unable to accurately determine quality using just the first two principal components.

Question 4:

Read in data withheader=TRUE to preserve column names. Drop the unique id as it contains no meaningful information. Scale the data and extract the centering and scaling factors (mu and sigma).

```
tweets=read.csv("https://raw.githubusercontent.com/jgscott/STA380/master/data
/social_marketing.csv",header=TRUE)

tweets_s<-tweets[,-1]

tweets_s<-scale(tweets_s)

mu=attr(tweets_s,"scaled:center")

sigma=attr(tweets_s,"scaled:scale")
```

Set the seed to maintain reproducibility.

```
set.seed(1234)
```

Run k-means with 10 centers and 50 nstarts. Although this k and n may not be the optimal in terms of accuracy, they ensure that our code runs in a timely fashion and that our results remain interpretable.

```
tweets_clusters<-kmeans(tweets_s, centers=10, nstart=50)
```

Examine clusters

We will consider how far away each cluster center is from the entire data's means: More standard deviations indicate how much more a particular clusters' members tweet about a particular topic compared to the entire dataset. Standard deviations above one are particularly interesting because they suggest that the members of a particular cluster tweet far more about a particular topic than the other groups.

We then consider the unscaled data, which will tell us how meaningful the standard deviation is. Segments who creates many tweets about a particular topic are likely to be interested in that topic and more likely to respond to targeted social media. We also examine total tweets per week to determine how much a particular segment engages with twitter.

This three-step method minimizes noise from category size and hones in on Twitter engagement. The below function calculates the scaled and unscaled centers as well as the average total tweets per cluster to facilitate our analysis.

```
scaled.unscaled<-function (x) {
rows<-rows<-
rbind(tweets_clusters$center[x,],(tweets_clusters$center[x,]*sigma + mu))
rownames(rows)<-c("Scaled", "Unscaled")
s<-sum(tweets_clusters$center[x,]*sigma + mu)
```

```r
list("Comparison"=rows,"Total Tweets"=s)
}
```

Cluster 1

This cluster appears to be composed of parents. These tweeters write far more than average about `sports_fandom` and `parenting`. They also tweet more often about `school` than other clusters. Of the 58 tweets they average a week, 14 are about these three topics

```r
scaled.unscaled(1)
```

```
## $Comparison
##              chatter current_events     travel photo_sharing uncategorized
## Scaled    -0.1310678     0.09856875 -0.1021084   -0.09702572    -0.1093218
## Unscaled   3.9362018     1.65133531  1.3516320    2.43175074     0.7106825
##               tv_film sports_fandom   politics     food   family
## Scaled    -0.09782764      2.093184 -0.2239573 1.852633 1.519301
## Unscaled   0.90801187      6.117211  1.1097923 4.686944 2.584570
##           home_and_garden      music       news online_gaming    shopping
## Scaled          0.1592284 0.02473611 -0.1105484   -0.07770529 -0.02250247
## Unscaled        0.6379822 0.70474777  0.9732938    1.00000000  1.34866469
##           health_nutrition college_uni sports_playing     cooking       eco
## Scaled          -0.1433213  -0.1312807      0.1021966 -0.09767488 0.1844765
## Unscaled         1.9228487   1.1691395      0.7388724  1.66320475 0.6543027
##            computers  business    outdoors    crafts automotive         art
## Scaled    0.09123101 0.1001457 -0.06687896 0.6998591  0.1180195 -0.02415113
## Unscaled  0.75667656 0.4925816  0.70178042 1.0875371  0.9910979  0.68545994
##           religion    beauty parenting     dating   school personal_fitness
## Scaled    2.297929 0.3214817  2.170670 0.01821377 1.686345      -0.08971009
## Unscaled  5.495549 1.1320475  4.210682 0.74332344 2.771513       1.24629080
##              fashion small_business          spam        adult
## Scaled    0.01242245     0.09195084 -7.768727e-02 -0.004778395
## Unscaled  1.01928783     0.39317507 -2.341877e-17  0.394658754
##
## $`Total Tweets`
## [1] 58.42285
```

Cluster 3:

This cluster appears to be the active cluster. These tweeters discuss `outdoors`, `personal fitness`, and `health_nutrition` far more than the average tweeter. About 20 of their 57 weekly tweets are about these three topics.

```r
scaled.unscaled(3)
```

```
## $Comparison
##              chatter current_events     travel photo_sharing uncategorized
## Scaled    -0.1295931   -0.009409365 -0.1556913    -0.1087449     0.1719999
## Unscaled   3.9414062    1.514322917  1.2291667     2.3997396     0.9739583
##              tv_film sports_fandom   politics      food      family
## Scaled    -0.1483424    -0.1983635 -0.2000389 0.4552042 -0.08904256
```

```
## Unscaled  0.8242187     1.1653646  1.1822917 2.2057292  0.76302083
##          home_and_garden        music        news online_gaming
## Scaled         0.1575134 -0.004650472 -0.07428308    -0.1106515
## Unscaled       0.6367187  0.674479167  1.04947917     0.9114583
##             shopping health_nutrition college_uni sports_playing   cooking
## Scaled   -0.05833223          2.21844  -0.2089876    -0.01853799 0.4162047
## Unscaled  1.28385417         12.54167   0.9440104     0.62109375 3.4257812
##                  eco   computers   business outdoors     crafts automotive
## Scaled   0.5642381 -0.08444139 0.05256166 1.731015 0.06666309 -0.1747389
## Unscaled 0.9466146  0.54947917 0.45963542 2.876302 0.57031250  0.5911458
##                    art    religion     beauty   parenting    dating
## Scaled   -0.07563536 -0.1654254 -0.2015592 -0.08900958 0.1987514
## Unscaled  0.60156250  0.7786458  0.4375000  0.78645833 1.0651042
##              school personal_fitness     fashion small_business
## Scaled   -0.1650178         2.157359 -0.09426523     -0.1164983
## Unscaled  0.5716146         6.651042  0.82421875      0.2643229
##                   spam      adult
## Scaled   -7.768727e-02 0.01812804
## Unscaled -4.076600e-17 0.43619792
## 
## $`Total Tweets`
## [1] 56.69792
```

Cluster 4:

This cluster appears to be the politically-engaged cluster. They discuss news, `travel`, `current events`, `computers`, and `politics` far more than average tweeter. These topics compromise about 30 of their 61 weekly tweets.

```
scaled.unscaled(4)
```

```
## $Comparison
##              chatter current_events   travel photo_sharing uncategorized
## Scaled   -0.07726621      0.1136621 3.265636     -0.110328   -0.08797596
## Unscaled  4.12607450      1.6704871 9.048711      2.395415    0.73065903
##              tv_film sports_fandom politics      food      family
## Scaled   -0.07173772    -0.2085897  3.11929 0.1569816 -0.09231701
## Unscaled  0.95128940     1.1432665 11.24355 1.6762178  0.75931232
##          home_and_garden      music     news online_gaming    shopping
## Scaled        0.05166238 -0.0419082 1.140618    -0.1704632 -0.07586007
## Unscaled      0.55873926  0.6361032 3.601719     0.7507163  1.25214900
##          health_nutrition college_uni sports_playing    cooking       eco
## Scaled         -0.1694973 -0.04922176     0.04384399 -0.1866089 0.1608323
## Unscaled        1.8051576  1.40687679     0.68194842  1.3581662 0.6361032
##          computers  business    outdoors    crafts automotive        art
## Scaled    2.911536 0.5598746 -0.03826403 0.2033299 -0.1313440 -0.1616973
## Unscaled  4.083095 0.8108883  0.73638968 0.6819484  0.6504298  0.4613181
##           religion     beauty  parenting   dating     school
## Scaled   0.1162737 -0.1771492 0.02354578 0.305302 -0.1059236
## Unscaled 1.3180516  0.4699140 0.95702006 1.255014  0.6418338
##          personal_fitness    fashion small_business          spam
```

```
## Scaled          -0.148030 -0.1705090      0.4015086 -7.768727e-02
## Unscaled         1.106017  0.6848138      0.5845272  5.030698e-17
##                adult
## Scaled   -0.1434066
## Unscaled  0.1432665
##
## $`Total Tweets`
## [1] 61.01719
```

Cluster 5:

This cluster appears to be the young male cluster. They discuss `automotive`, `politics`, and news more than other tweeters. 17 of their 50 weekly tweets cover these topics.

```
scaled.unscaled(5)

## $Comparison
##               chatter current_events     travel photo_sharing uncategorized
## Scaled   -0.06873643      0.0720734 -0.1866069    -0.2209537   -0.09408515
## Unscaled  4.15617716      1.6177156  1.1585082     2.0932401    0.72494172
##             tv_film sports_fandom politics       food    family
## Scaled   -0.011457     0.6679035 1.225577 -0.1542867 0.2354565
## Unscaled  1.051282     3.0372960 5.503497  1.1235431 1.1305361
##          home_and_garden       music     news online_gaming   shopping
## Scaled         0.1601955 -0.08917992 2.663931    -0.1219407 -0.1881958
## Unscaled       0.6386946  0.58741259 6.801865     0.8811189  1.0489510
##          health_nutrition college_uni sports_playing    cooking
## Scaled         -0.2428119  -0.1944894    -0.08412803 -0.2346252
## Unscaled        1.4755245   0.9860140     0.55710956  1.1934732
##                 eco   computers   business   outdoors     crafts automotive
## Scaled   -0.09623969 -0.1866707 -0.1231226 0.3107434 -0.1606708   2.590075
## Unscaled  0.43822844  0.4289044  0.3379953 1.1585082  0.3846154   4.368298
##                 art    religion     beauty  parenting      dating
## Scaled   -0.1615621 -0.1788637 -0.1764350 0.04114091 -0.03394992
## Unscaled  0.4615385  0.7529138  0.4708625 0.98368298  0.65034965
##               school personal_fitness    fashion small_business
## Scaled   0.01502133       -0.2299037 -0.2148557     -0.1556956
## Unscaled 0.78554779        0.9090909  0.6037296      0.2400932
##                  spam      adult
## Scaled   -7.768727e-02 -0.1092935
## Unscaled  5.377643e-17  0.2051282
##
## $`Total Tweets`
## [1] 48.94639
```

Cluster 6:

This appears to be the college student cluster. They discuss `online_gaming`, `college_uni`, and `sports_playing` more often than other tweeters. 25 of their weekly 58 tweets compromise of these topics.

```r
scaled.unscaled(6)
```

```
## $Comparison
##              chatter current_events      travel photo_sharing
## Scaled   -0.08870253    -0.09049938 -0.03219177   -0.01451015
## Unscaled  4.08571429     1.41142857  1.51142857    2.65714286
##          uncategorized    tv_film sports_fandom   politics        food
## Scaled     -0.03525299 0.09886703    -0.1347365 -0.1753446 -0.09030691
## Unscaled    0.78000000 1.23428571     1.3028571  1.2571429  1.23714286
##             family home_and_garden       music       news online_gaming
## Scaled   0.2059718      0.07276547 -0.05199434 -0.1875984      3.619885
## Unscaled 1.0971429      0.57428571  0.62571429  0.8114286     10.937143
##            shopping health_nutrition college_uni sports_playing    cooking
## Scaled   -0.1362808       -0.1833537    3.309338       2.147690 -0.1177682
## Unscaled  1.1428571        1.7428571   11.137143       2.734286  1.5942857
##                  eco   computers    business   outdoors     crafts
## Scaled   -0.06795483 -0.08036615 -0.09959447 -0.1392195 0.03305173
## Unscaled  0.46000000  0.55428571  0.35428571  0.6142857 0.54285714
##          automotive       art   religion     beauty  parenting      dating
## Scaled   0.06806834 0.2740668 -0.1930684 -0.2233443 -0.1290952 -0.01090226
## Unscaled 0.92285714 1.1714286  0.7257143  0.4085714  0.7257143  0.69142857
##              school personal_fitness     fashion small_business
## Scaled   -0.2276908       -0.1826045 -0.06531985      0.1261023
## Unscaled  0.4971429        1.0228571  0.87714286      0.4142857
##                  spam       adult
## Scaled   -7.768727e-02 -0.02073959
## Unscaled  5.030698e-17  0.36571429
##
## $`Total Tweets`
## [1] 58.22286
```

Cluster 8:

This appears to be the young female cluster. They discuss `photo_sharing`, `beauty`, `cooking`, and `fashion` more often than other tweeters. 27 of their 62 weekly tweets are about these topics.

```r
scaled.unscaled(8)
```

```
## $Comparison
##              chatter current_events      travel photo_sharing
## Scaled   -0.04319508      0.1775698 -0.05423302      1.241674
## Unscaled  4.24631579      1.7515789  1.46105263      6.088421
##          uncategorized    tv_film sports_fandom   politics       food
## Scaled       0.4990187 -0.1362904    -0.2057172 -0.1275198 -0.2037098
## Unscaled     1.2800000  0.8442105     1.1494737  1.4021053  1.0357895
##              family home_and_garden     music        news online_gaming
## Scaled   0.02911547       0.1419633 0.5525667 -0.07578889   -0.02286982
## Unscaled 0.89684211       0.6252632 1.2484211  1.04631579    1.14736842
##           shopping health_nutrition college_uni sports_playing   cooking
## Scaled   0.2025719      -0.06622745 -0.01816877      0.2015461  2.823952
```

```
## Unscaled 1.7557895       2.26947368  1.49684211      0.8357895 11.684211
##                    eco  computers  business    outdoors     crafts
## Scaled   -0.0009452388 0.05656488 0.2279240 0.007366432 0.08238866
## Unscaled  0.5115789474 0.71578947 0.5810526 0.791578947 0.58315789
##          automotive          art   religion   beauty   parenting
## Scaled   0.01204133 0.0009203335 -0.1212898 2.638198 -0.05784476
## Unscaled 0.84631579 0.7263157895  0.8631579 4.208421  0.83368421
##              dating    school personal_fitness  fashion small_business
## Scaled   0.04883143 0.1724649      -0.04418512 2.728426      0.1642956
## Unscaled 0.79789474 0.9726316       1.35578947 5.985263      0.4378947
##                   spam        adult
## Scaled   -7.768727e-02 0.0004888515
## Unscaled  3.295975e-17 0.4042105263
##
## $`Total Tweets`
## [1] 62.88
```

Cluster 10:

This is the artsy cluster. They discuss `tv_film` and `art` more often than other tweeters. 11 of their 51 weekly tweets are about these topics.

```
scaled.unscaled(10)
```

```
## $Comparison
##             chatter current_events    travel photo_sharing uncategorized
## Scaled   -0.1205556      0.3274398 0.2229927   -0.08181427     0.6900079
## Unscaled  3.9733010      1.9417476 2.0946602    2.47330097     1.4587379
##           tv_film sports_fandom    politics      food     family
## Scaled   2.749474    -0.1153915 -0.09202017 0.1493241 -0.1112548
## Unscaled 5.631068     1.3446602  1.50970874 1.6626214  0.7378641
##          home_and_garden    music        news online_gaming   shopping
## Scaled         0.3343467 1.004183 0.004992348    -0.1680203 0.01956446
## Unscaled       0.7669903 1.713592 1.216019417     0.7572816 1.42475728
##          health_nutrition college_uni sports_playing    cooking        eco
## Scaled         -0.1601716   0.3666255      0.1409726 -0.1424267 0.09753111
## Unscaled        1.8470874   2.6116505      0.7766990  1.5097087 0.58737864
##           computers  business    outdoors   crafts automotive      art
## Scaled   -0.1510870 0.3457334 -0.08922167 0.735322 -0.2272429 2.636900
## Unscaled  0.4708738 0.6626214  0.67475728 1.116505  0.5194175 5.021845
##            religion     beauty  parenting      dating      school
## Scaled   0.01482072 0.01184033 -0.1963584 -0.05974777 -0.04757675
## Unscaled 1.12378641 0.72087379  0.6237864  0.60436893  0.71116505
##          personal_fitness     fashion small_business          spam
## Scaled         -0.1537609 -0.02202118      0.7909234 -7.768727e-02
## Unscaled        1.0922330  0.95631068      0.8252427  6.245005e-17
##               adult
## Scaled   -0.0403804
## Unscaled  0.3300971
##
```

```
## $`Total Tweets`
## [1] 51.49272
```

Conclusion:

For many of the tweet categories above, people who use any particular one are more likely to use a particular subset of the other ones. These association patterns suggest that we have identified distinct segments with particular overlapping interests. In other words, certain interests appear to be correlated with other particular interests.