Homework 1

Jordan Giebas

Due Wednesday, November 1 at 5:30 PM

Part 1:

Reconsider the data set that was presented in the final exam for Mini 1. Create a new data frame that gives the day-to-day changes in all of the rates. Run a principal components analysis on these vectors. Is PCA effective in reducing the dimensionality of the rate change vectors? Try it with and without scaling the variables first and describe how the results change.

```
library(quantmod)
```

```
## Warning: package 'quantmod' was built under R version 3.4.2

## Loading required package: xts

## Loading required package: zoo

##
## Attaching package: 'zoo'

## The following objects are masked from 'package:base':
##
##     as.Date, as.Date.numeric

## Loading required package: TTR

## Version 0.4-0 included new data defaults. See ?getSymbols.
```

```
library(ggplot2)
library(plyr)

## Get the data from the final exam
######################################################################
fileheader = read.table("commercial_paper_rates.csv",sep=",",nrows=6,stringsAsFactors=FA
cprfullnames = as.character(c("Time.Period",fileheader[1,-1]))
cprdata = read.table("commercial_paper_rates.csv",sep=",", header=T,
                      skip=5,stringsAsFactors = FALSE,
                      na.strings=c("ND","NA"))

cprdata$Time.Period = as.Date(cprdata$Time.Period, format="%Y-%m-%d")
fulldata = cprdata
#fulldata = cprdata[complete.cases(cprdata),]
######################################################################

dates = fulldata$Time.Period
dates = dates[2:length(dates)]

data = fulldata[,c(2:ncol(fulldata))]
data = colwise(diff)(data)

df = cbind(dates, data)
df_dates = df[,1]
df_data = df[,c(2:ncol(df))]

df_data = df_data[complete.cases(df_data),]

df_pca_noscale = princomp(df_data)
summary(df_pca_noscale)
```

```
## Importance of components:
##                           Comp.1    Comp.2     Comp.3     Comp.4
## Standard deviation     0.2125105 0.1152238 0.10521763 0.08534759
## Proportion of Variance 0.3988022 0.1172415 0.09776283 0.06432492
## Cumulative Proportion  0.3988022 0.5160437 0.61380655 0.67813147
##                            Comp.5     Comp.6     Comp.7     Comp.8
## Standard deviation     0.06999202 0.06349741 0.06111537 0.06007489
## Proportion of Variance 0.04326072 0.03560482 0.03298357 0.03187005
## Cumulative Proportion  0.72139219 0.75699701 0.78998058 0.82185063
##                            Comp.9    Comp.10    Comp.11    Comp.12
## Standard deviation     0.05346768 0.05150504 0.04665964 0.04277101
## Proportion of Variance 0.02524523 0.02342590 0.01922558 0.01615459
## Cumulative Proportion  0.84709586 0.87052176 0.88974734 0.90590193
##                           Comp.13    Comp.14    Comp.15     Comp.16
## Standard deviation     0.03797872 0.03783245 0.03495439 0.033545953
## Proportion of Variance 0.01273730 0.01263938 0.01078947 0.009937497
## Cumulative Proportion  0.91863923 0.93127861 0.94206808 0.952005579
##                            Comp.17     Comp.18     Comp.19     Comp.20
## Standard deviation     0.032606195 0.029661307 0.028966165 0.027525308
## Proportion of Variance 0.009388517 0.007769218 0.007409327 0.006690539
## Cumulative Proportion  0.961394096 0.969163314 0.976572640 0.983263180
##                            Comp.21    Comp.22     Comp.23     Comp.24
## Standard deviation     0.025793501 0.02484151 0.020239544 0.014256505
## Proportion of Variance 0.005875128 0.00544945 0.003617414 0.001794828
## Cumulative Proportion  0.989138307 0.99458776 0.998205172 1.000000000
```

```
df_pca_scale = princomp(df_data, cor=TRUE)
summary(df_pca_scale)
```

```
## Importance of components:
##                           Comp.1    Comp.2     Comp.3     Comp.4
## Standard deviation     2.9536502 1.6378165 1.28926926 1.18723955
## Proportion of Variance 0.3635021 0.1117685 0.06925897 0.05873074
## Cumulative Proportion  0.3635021 0.4752705 0.54452949 0.60326023
##                            Comp.5     Comp.6     Comp.7     Comp.8
## Standard deviation     1.04363855 0.92928704 0.89737000 0.88305403
## Proportion of Variance 0.04538256 0.03598227 0.03355304 0.03249102
## Cumulative Proportion  0.64864278 0.68462505 0.71817809 0.75066911
##                            Comp.9    Comp.10    Comp.11    Comp.12
## Standard deviation     0.83641909 0.80561087 0.77785778 0.73805502
## Proportion of Variance 0.02914987 0.02704204 0.02521095 0.02269688
## Cumulative Proportion  0.77981898 0.80686101 0.83207196 0.85476884
##                           Comp.13    Comp.14    Comp.15    Comp.16
## Standard deviation     0.70608927 0.65751753 0.65074845 0.62492918
## Proportion of Variance 0.02077342 0.01801372 0.01764473 0.01627235
## Cumulative Proportion  0.87554226 0.89355598 0.91120072 0.92747307
##                          Comp.17    Comp.18    Comp.19    Comp.20
## Standard deviation     0.6052827 0.58880096 0.56802434 0.53127221
## Proportion of Variance 0.0152653 0.01444527 0.01344382 0.01176042
## Cumulative Proportion  0.9427384 0.95718364 0.97062746 0.98238788
##                           Comp.21     Comp.22     Comp.23     Comp.24
## Standard deviation     0.49626409 0.315356253 0.227632765 0.158576562
## Proportion of Variance 0.01026159 0.004143732 0.002159028 0.001047772
## Cumulative Proportion  0.99264947 0.996793200 0.998952228 1.000000000
```

Unscaled case: The PCA is definitely reasonable; however, it's not great. We see approximately 90% of the variability captured within 12 of the Principal Components, i.e. half the predictors that were original. Further, 95% of the variability is captured in the first 16 pcs, i.e. 2/3 of the predictors. In this sense, the unscaled PCA is rather useful depending on how willing the user is to sacrifice capturing some of the variability.

Scaled case: The scaled case is quite similar to the unscaled case, perhaps it is a little worse in the sense that the scaled case requires 15 principal components to capture 90% of the variability rather than 12. It's sensible that the scaled case is highly similar to the unscaled since the variables are rates and are on the same scale.

Part 2:

As described in lecture, one approach to the time series dimension reduction situation we were facing would be to first smooth the time series, and then use these smoothed time series as the input to a dimension reduction algorithm. We will try this here.

In particular, smooth each time series using `loess()`. I will leave the choice of the smoothing parameter up to you. After smoothing, you should use the `predict()` function to evaluate the fitted model on a regular grid of x values. These smoothed time series are what should be utilized in the dimension reduction. Use Isomap. Explore the first two-dimensions to see if there is meaningful low-dimensional structure in the plot.

Comment: If you watched lecture, this will make a lot more sense. Be sure to watch the lecture prior to attempting this.

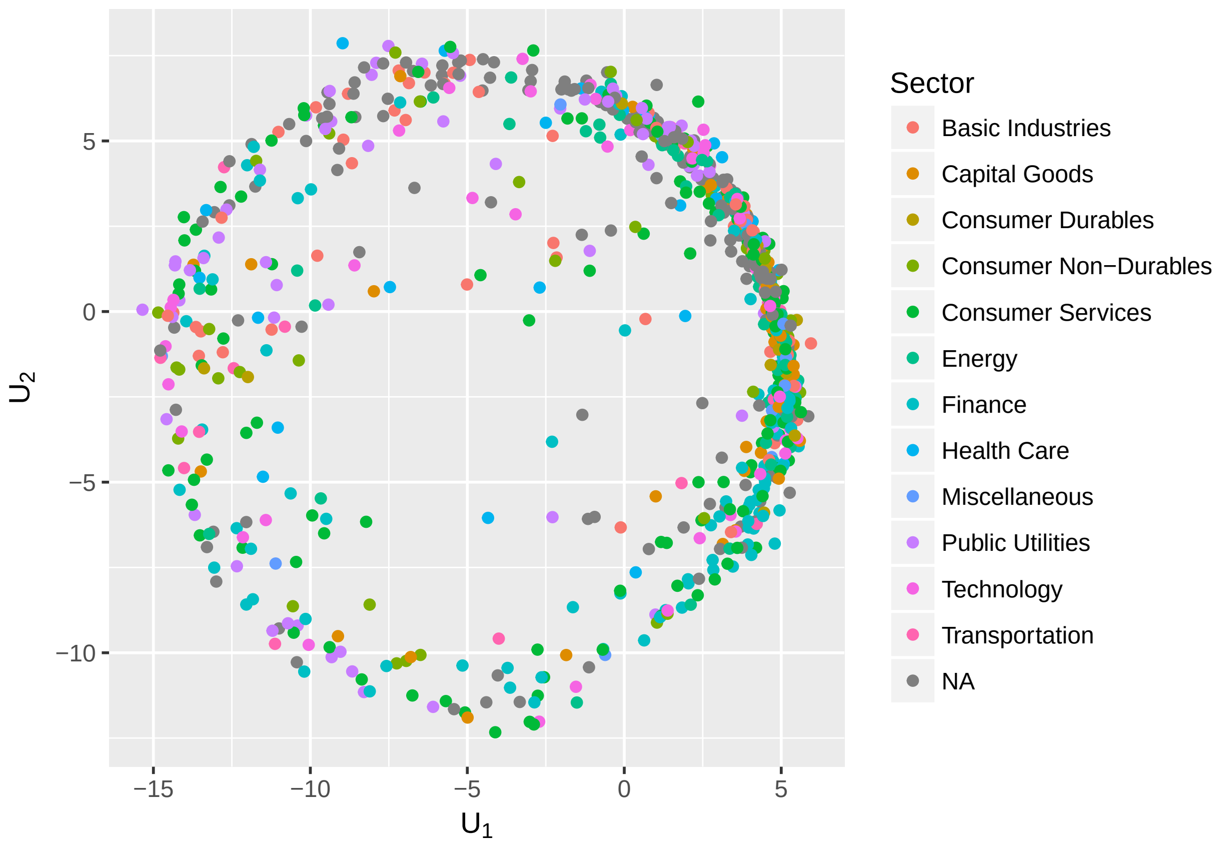
```r
library(vegan)
```

```
## Loading required package: permute
```

```
## Loading required package: lattice
```

```
## This is vegan 2.4-4
```

```r
library(ggplot2)

## Data from lecture, smoothing the data using loess
stocksample = read.table("stocksample.txt", header=T, sep="\t", comment.char="")
smooth_df = apply(t(stocksample)[5:34,], 2, function(x) { return (predict(loess(x~c(1:30

## Scaling the data
ss_scaled = apply(smooth_df, 2, scale)

## Distance/Isomap
iso_out = isomap(dist(t(ss_scaled)), k=5)
stocksample = cbind(stocksample, iso_out$points[,1:5])

plt = ggplot(stocksample,aes(x=stocksample[,35],y=stocksample[,36], color=sector)) +
  geom_point() +
  labs(x=expression(U[1]), y=expression(U[2]), color="Sector")

plt
```



After running isomap and looking at the first two dimensions, it's clear that there is a centric pattern across the variety of sectors. Further, there is some apparent clustering when $U1 \approx 5$. The dimension reduction is rather efficient, for there is a clear pattern represented in this two dimensional space. Whether this is because of similarities across sectors versus across time is a different question: one could test this by sampling a proportion of companies falling in some neighborhood of $U1 = 5$ and looking at comparing data of different sectors over time. If there is a similar trend over the time horizon, this clustering effect is likely due in part because of the time than the sector.