154HW2_3032235220_JiyoonJeong

Jiyoon Clover Jeong

9/17/2017

```r
data <- read.csv("/Users/cloverjiyoon/2017Fall/Stat 154/data/temperature.csv")

head(data)
```

```
##            X January February March April  May June July August September
## 1  Amsterdam     2.9      2.5   5.7   8.2 12.5 14.8 17.1   17.1      14.5
## 2     Athens     9.1      9.7  11.7  15.4 20.1 24.5 27.4   27.2      23.8
## 3     Berlin    -0.2      0.1   4.4   8.2 13.8 16.0 18.3   18.0      14.4
## 4   Brussels     3.3      3.3   6.7   8.9 12.8 15.6 17.8   17.8      15.0
## 5   Budapest    -1.1      0.8   5.5  11.6 17.0 20.2 22.0   21.3      16.9
## 6 Copenhagen    -0.4     -0.4   1.3   5.8 11.1 15.4 17.1   16.6      13.3
##   October November December Annual Amplitude Latitude Longitude  Area
## 1    11.4      7.0      4.4    9.9      14.6     52.2       4.5  West
## 2    19.2     14.6     11.0   17.8      18.3     37.6      23.5 South
## 3    10.0      4.2      1.2    9.1      18.5     52.3      13.2  West
## 4    11.1      6.7      4.4   10.3      14.4     50.5       4.2  West
## 5    11.3      5.1      0.7   10.9      23.1     47.3      19.0  East
## 6     8.8      4.1      1.3    7.8      17.5     55.4      12.3 North
```

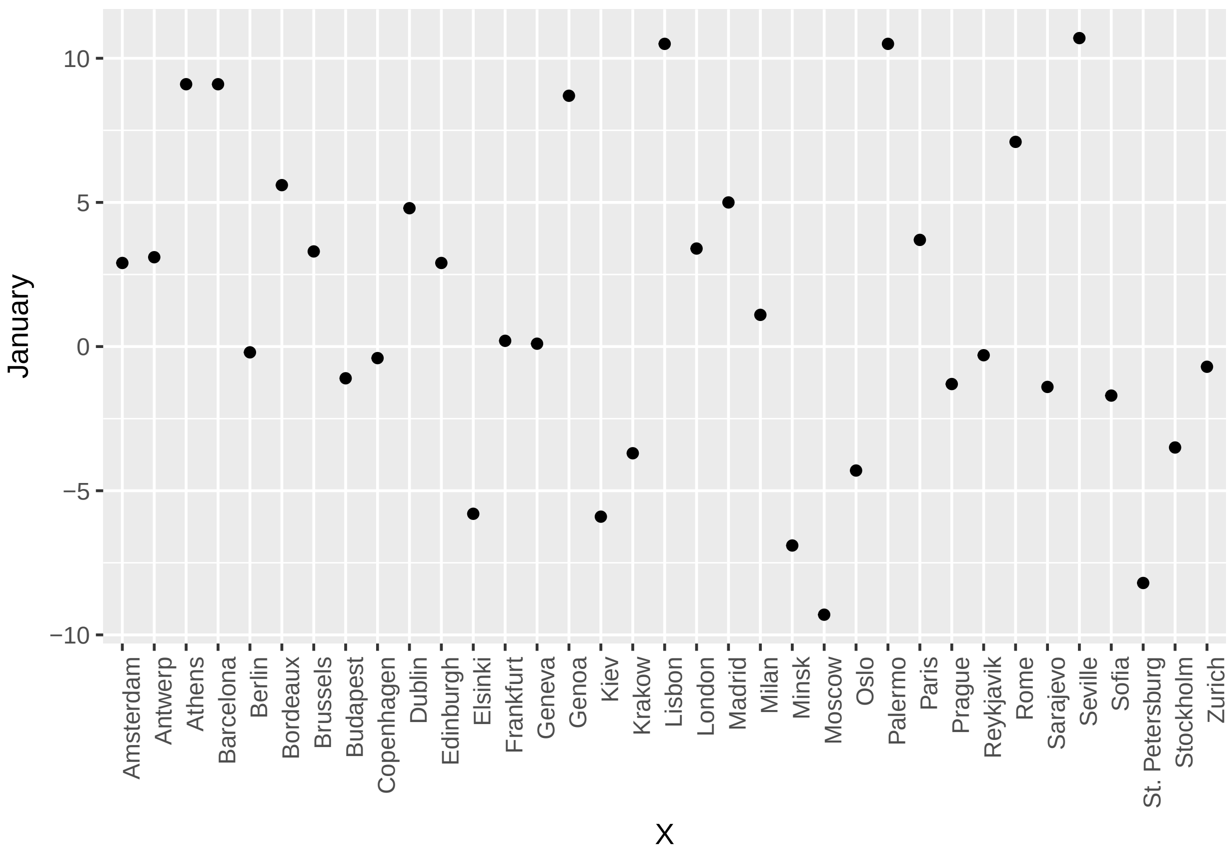
Explanatory Phase

```r
summary(data)
```

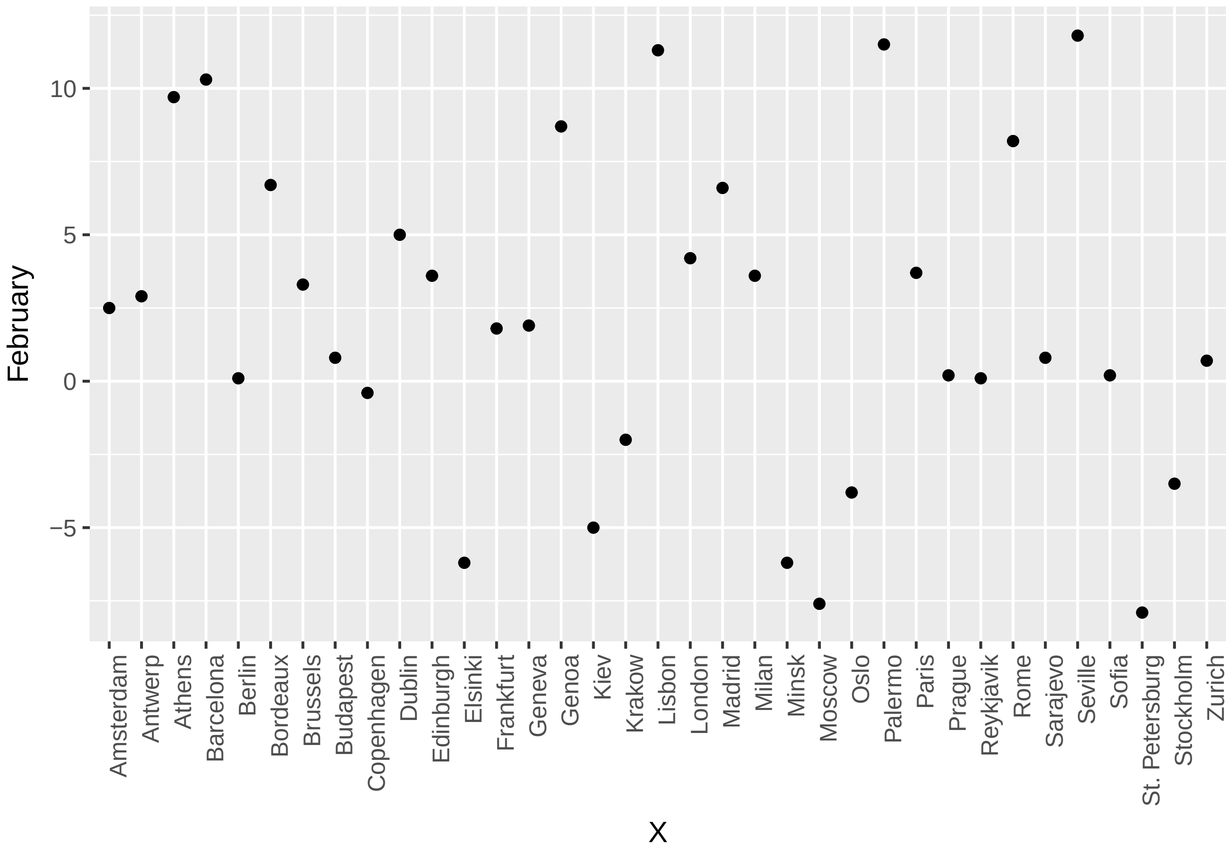
```
##           X         January          February           March       
##  Amsterdam: 1   Min.   :-9.300   Min.   :-7.900   Min.   :-3.700  
##  Antwerp  : 1   1st Qu.:-1.550   1st Qu.:-0.150   1st Qu.: 1.600  
##  Athens   : 1   Median : 0.200   Median : 1.900   Median : 5.400  
##  Barcelona: 1   Mean   : 1.346   Mean   : 2.217   Mean   : 5.229  
##  Berlin   : 1   3rd Qu.: 4.900   3rd Qu.: 5.800   3rd Qu.: 8.500  
##  Bordeaux : 1   Max.   :10.700   Max.   :11.800   Max.   :14.100  
##  (Other)  :29                                                     
##      April             May             June            July      
##  Min.   : 2.900   Min.   : 6.50   Min.   : 9.30   Min.   :11.10  
##  1st Qu.: 7.250   1st Qu.:12.15   1st Qu.:15.40   1st Qu.:17.30  
##  Median : 8.900   Median :13.80   Median :16.90   Median :18.90  
##  Mean   : 9.283   Mean   :13.91   Mean   :17.41   Mean   :19.62  
##  3rd Qu.:12.050   3rd Qu.:16.35   3rd Qu.:19.80   3rd Qu.:21.75  
##  Max.   :16.900   Max.   :20.90   Max.   :24.50   Max.   :27.40  
##                                                                  
##      August        September        October         November     
##  Min.   :10.60   Min.   : 7.90   Min.   : 4.50   Min.   :-1.100  
##  1st Qu.:16.65   1st Qu.:13.00   1st Qu.: 8.65   1st Qu.: 3.200  
##  Median :18.30   Median :14.80   Median :10.20   Median : 5.100  
##  Mean   :18.98   Mean   :15.63   Mean   :11.00   Mean   : 6.066  
##  3rd Qu.:21.60   3rd Qu.:18.25   3rd Qu.:13.30   3rd Qu.: 7.900  
##  Max.   :27.20   Max.   :24.30   Max.   :19.40   Max.   :14.900  
```

```
##
##     December          Annual         Amplitude         Latitude
##  Min.   :-6.00   Min.   : 4.50   Min.   :10.20   Min.   :37.20
##  1st Qu.: 0.25   1st Qu.: 7.75   1st Qu.:14.90   1st Qu.:43.90
##  Median : 1.70   Median : 9.70   Median :18.50   Median :50.00
##  Mean   : 2.88   Mean   :10.27   Mean   :18.32   Mean   :49.04
##  3rd Qu.: 5.40   3rd Qu.:12.65   3rd Qu.:21.45   3rd Qu.:53.35
##  Max.   :12.00   Max.   :18.20   Max.   :27.60   Max.   :64.10
##
##    Longitude         Area
##  Min.   : 0.00   East : 8
##  1st Qu.: 5.05   North: 8
##  Median :10.50   South:10
##  Mean   :13.01   West : 9
##  3rd Qu.:19.30
##  Max.   :37.60
##
```

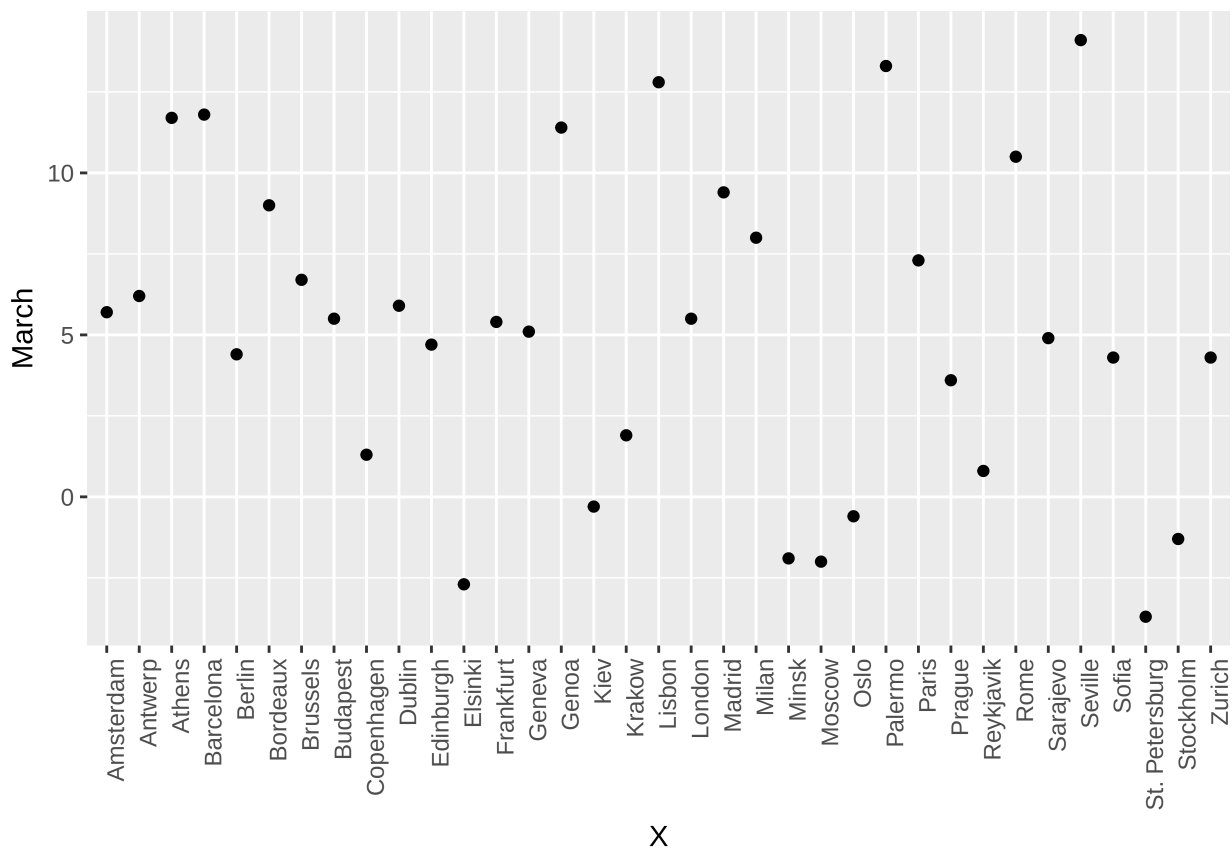
```
ggplot(data,aes(x=X,y=January)) + geom_point()+ theme(axis.text.x = element_text(angle = 90, hjust = 1))
```



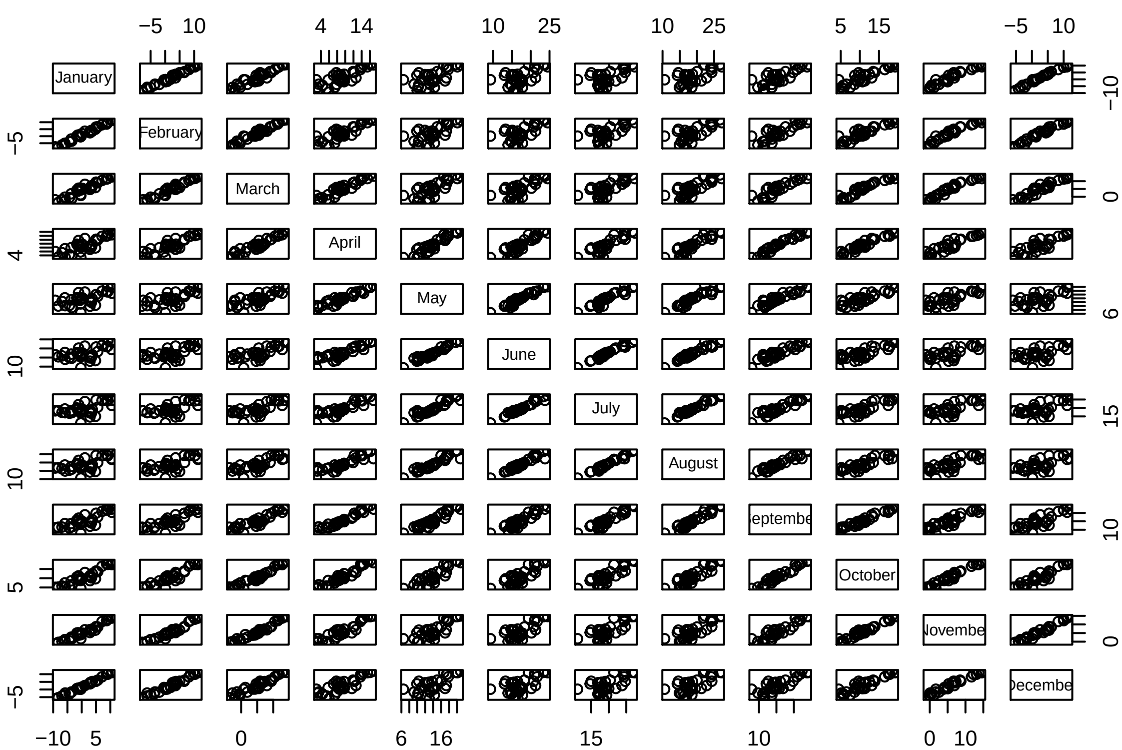

```
ggplot(data,aes(x=X,y=February)) + geom_point()+ theme(axis.text.x = element_text(angle = 90, hjust = 1
```




```
ggplot(data,aes(x=X,y=March)) + geom_point()+ theme(axis.text.x = element_text(angle = 90, hjust = 1))
```




```
pairs(data[,2:13])
```



1) Calculation of primary PCA outputs (30 pts)

a)

```r
X <- as.matrix(data[1:23,2:13])
rownames(X) <- data[1:23,1]
X <- scale(X, center = T, scale = T)
n <- 23
corr <- 1/(n-1) * t(X) %*% X
loadings <- eigen(corr)$vectors
rownames(loadings) <- colnames(X)
colnames(loadings) <- paste0("PC",1:12)
sqrt(sum(loadings[,1]^2))  # unit norm
```

```
## [1] 1
```

```r
loadings[,1:4]
```

```
##                  PC1         PC2           PC3          PC4
## January   -0.2671050 -0.39091041  0.1907187341 -0.059731884
## February  -0.2803688 -0.33534791 -0.0097552190 -0.427798846
## March     -0.2996355 -0.21137095 -0.3399569587 -0.397667051
## April     -0.3087780  0.07324821 -0.5579573828 -0.127078736
## May       -0.2757927  0.33680390 -0.4392770157  0.392591602
## June      -0.2642082  0.40118372  0.1394431457 -0.000489339
## July      -0.2676478  0.37421361  0.4325313064 -0.222824851
## August    -0.2882824  0.29568869  0.2462557102 -0.226852869
## September -0.3124996  0.11221817  0.0636774480 -0.026537477
## October   -0.3144017 -0.06235990 -0.0001874864  0.366581807
## November  -0.3019515 -0.21291689  0.1244515912  0.356372148
## December  -0.2768287 -0.34787886  0.2386777766  0.349002937
```

```r
# Check PCA with function prcomp
pca = prcomp(X, scale. = T)

# loading
pca$rotation[,1:4]
```

```
##                  PC1         PC2           PC3          PC4
## January   -0.2671050 -0.39091041  0.1907187341  0.059731884
## February  -0.2803688 -0.33534791 -0.0097552190  0.427798846
## March     -0.2996355 -0.21137095 -0.3399569587  0.397667051
## April     -0.3087780  0.07324821 -0.5579573828  0.127078736
## May       -0.2757927  0.33680390 -0.4392770157 -0.392591602
## June      -0.2642082  0.40118372  0.1394431457  0.000489339
## July      -0.2676478  0.37421361  0.4325313064  0.222824851
## August    -0.2882824  0.29568869  0.2462557102  0.226852869
## September -0.3124996  0.11221817  0.0636774480  0.026537477
## October   -0.3144017 -0.06235990 -0.0001874864 -0.366581807
## November  -0.3019515 -0.21291689  0.1244515912 -0.356372148
## December  -0.2768287 -0.34787886  0.2386777766 -0.349002937
```

b) Principal Components

```r
pc <- X %*% loadings[,1:12,drop=F]

pc[,1:4]
```

```
##                   PC1          PC2         PC3         PC4
## Amsterdam  -0.22195025 -1.341234829 -0.10209889  0.27657677
## Athens     -7.43360390  0.909925426  0.54908835  0.28025851
## Berlin      0.28153099  0.016092403 -0.28422057  0.05437108
## Brussels   -0.61729994 -1.151341565 -0.14870076 -0.01669466
## Budapest   -1.63136395  1.675051425 -0.48801530 -0.10996512
## Copenhagen  1.43025066 -0.481240562  0.43068897  0.17283180
## Dublin      0.49413580 -2.614731574 -0.17458563 -0.02925371
## Elsinki     3.94757646  0.451883416  0.58015037  0.23907168
## Kiev        1.67458427  1.963469194 -0.16691889  0.11032784
## Krakow      1.23099109  0.855756199 -0.26794138 -0.03573418
## Lisbon     -5.47621202 -1.520180219 -0.26440940  0.13422375
## London     -0.05637309 -1.539174219 -0.08281278 -0.05087152
## Madrid     -3.97473636  0.682329696  0.45164881 -0.64836153
## Minsk       3.16672621  1.360708200 -0.07068160  0.17931195
## Moscow      3.38650106  2.134053560 -0.29467958  0.00526448
## Oslo        3.23331905  0.303237840  0.28881834 -0.18641912
## Paris      -1.38850720 -0.877868695 -0.10790241  0.07732927
## Prague      0.10660691  0.682697725 -0.23723947 -0.09816888
## Reykjavik   4.60066569 -2.892196405 -0.05662577 -0.19107214
## Rome       -5.26370105  0.287243017  0.18510843  0.01231239
## Sarajevo   -0.15985914  0.312466849 -0.35657228 -0.07199691
## Sofia      -0.40862719  0.777598162 -0.23556939 -0.04675281
## Stockholm   3.07934588  0.005454959  0.85347084 -0.05658895
```

c) eigenvalues and their sum (equal to the number of variables)

```r
eval <- eigen(corr)$values
eval
```

```
##  [1] 9.9477504204 1.8476485015 0.1262558038 0.0382934463 0.0167094089
##  [6] 0.0128330357 0.0058302931 0.0020318929 0.0010234516 0.0009527707
## [11] 0.0005367834 0.0001341917
```

```r
# sqrt of eigenvalues
pca$sdev^2            # sd(pc[,1,drop=F])
```

```
##  [1] 9.9477504204 1.8476485015 0.1262558038 0.0382934463 0.0167094089
##  [6] 0.0128330357 0.0058302931 0.0020318929 0.0010234516 0.0009527707
## [11] 0.0005367834 0.0001341917
```

```r
sum(eval)
```

```
## [1] 12
```

```r
#check with the build in function prcomp
sum(pca$sdev^2)
```

```
## [1] 12
```

2) Choosing the number of dimensions to retain/examine (30 pts)

a) Make a summary table of the eigenvalues

```
proportion <- eval/ncol(X) * 100
cum_prop <- cumsum(proportion)
table <- cbind(eval, proportion, cum_prop)
rownames(table) <- paste0("PC",1:12)
table
```

```
##               eval   proportion  cum_prop
## PC1  9.9477504204 82.897920170  82.89792
## PC2  1.8476485015 15.397070846  98.29499
## PC3  0.1262558038  1.052131698  99.34712
## PC4  0.0382934463  0.319112052  99.66623
## PC5  0.0167094089  0.139245074  99.80548
## PC6  0.0128330357  0.106941964  99.91242
## PC7  0.0058302931  0.048585776  99.96101
## PC8  0.0020318929  0.016932441  99.97794
## PC9  0.0010234516  0.008528764  99.98647
## PC10 0.0009527707  0.007939756  99.99441
## PC11 0.0005367834  0.004473195  99.99888
## PC12 0.0001341917  0.001118264 100.00000
```

eval(first column) : Eigenvalues. Each eigenvalue represents the variance captured by each principal component

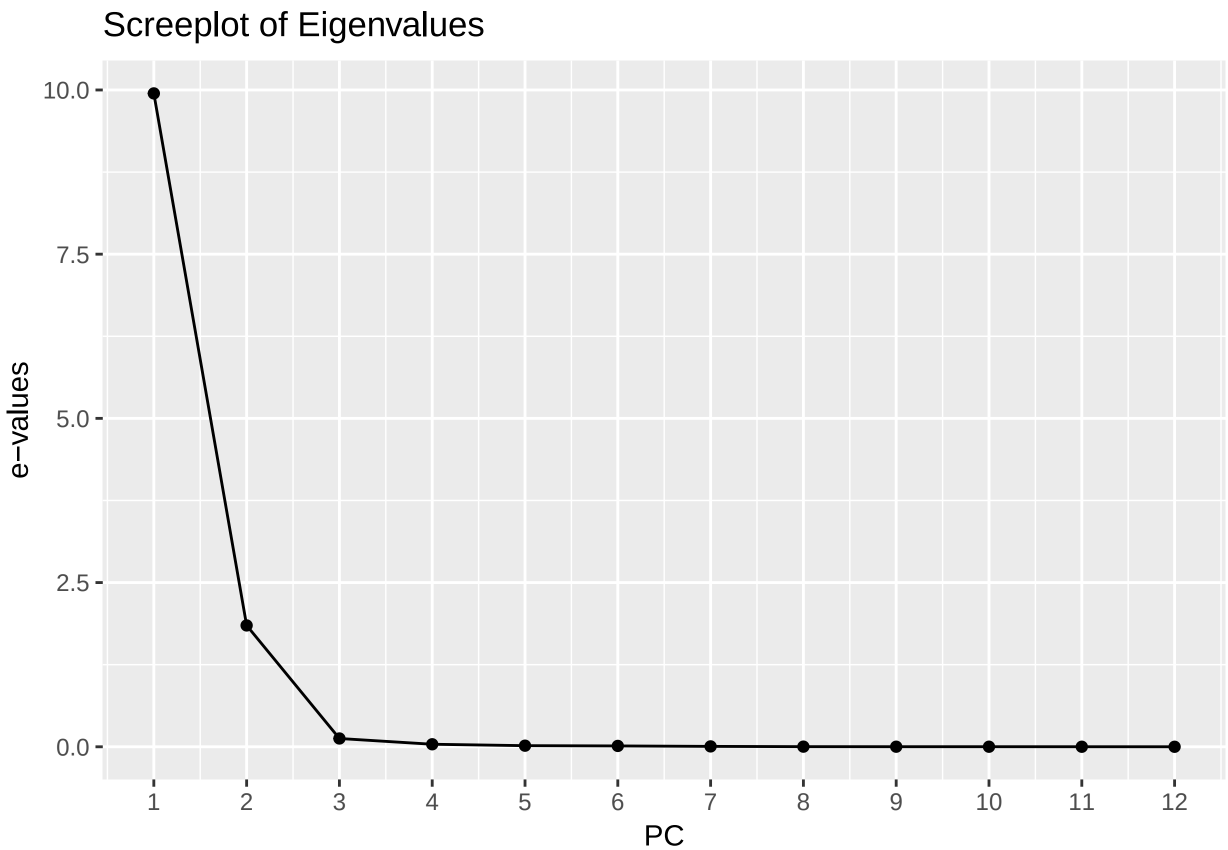
proportion(second column): The percentage of variance = λ_i/p. P is the number of variable = sum of eigenvalues.

cum_prop(Third column): Cmulative percentage of variance

Comment : Since PC1, PC2, and PC3 captures almost all the variances(proportion is greater than 1), we can say these three components explains the data properly.

b) Create a scree-plot (with axis labels) of the eigenvalues

```
ggplot(data = as.data.frame(table[,1]),aes(x = 1:12,y = eval)) + geom_point() + geom_line() +
ggtitle("Screeplot of Eigenvalues") + labs(x = "PC", y = "e-values") + scale_x_continuous(breaks=seq(1,
```



We can see that the 'elbow' of screeplot appears at PC2(2 at axis). Therefore we can keep PC1 and PC2 to compress original dimension of X and get the similar variation as the original data.

c) If you had to choose a number of dimensions (i.e. a number of PCs), how many would you choose and why?

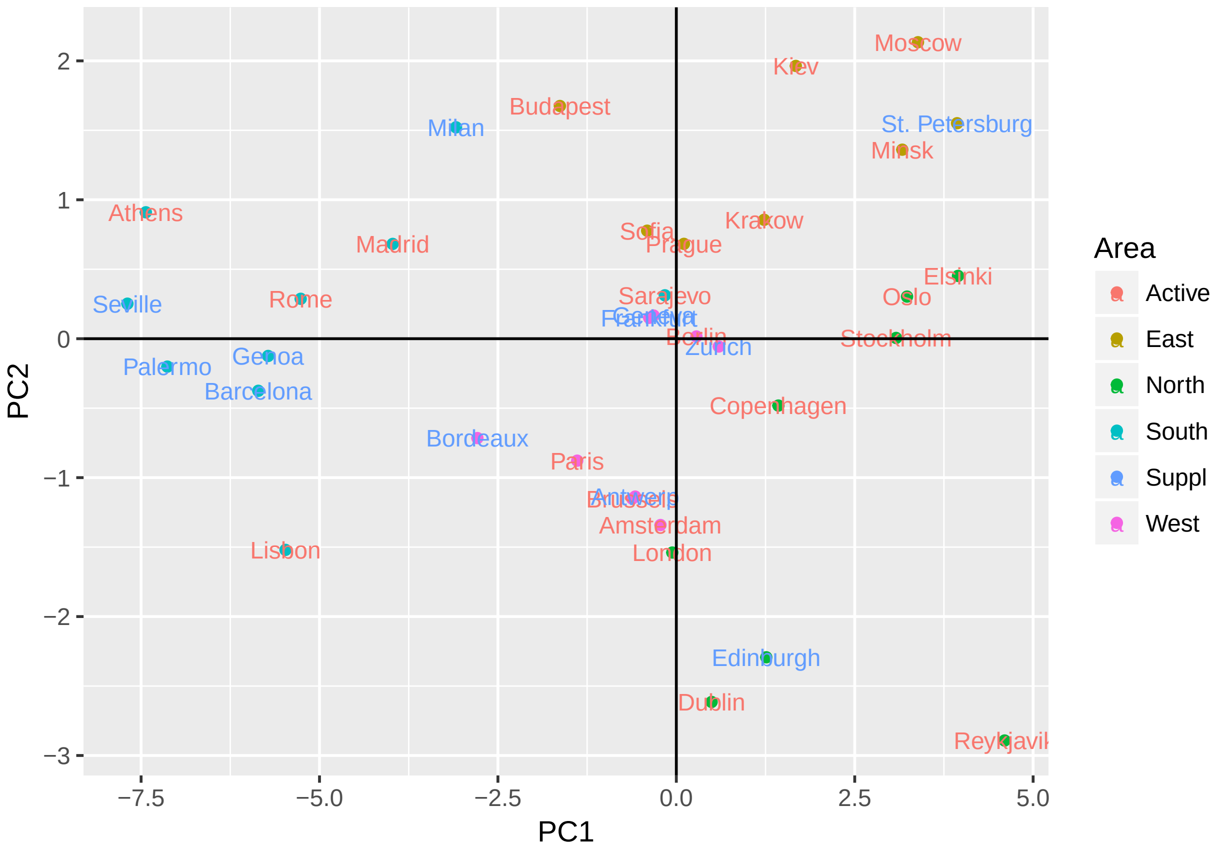
According to the information from screeplot and the Kaiser's rule, I would choose 2 dimensions(PC1 and PC2) since the eigenvalues of PC1 and PC2 are greater than 1.

3) Studying the cloud of individuals

a) Create a scatter plot of the cities on the 1st and 2nd PCs

```
X <- as.matrix(data[1:23,2:13])
rownames(X) <- data[1:23,1]

# Get Supplementary
Y <- as.matrix(data[24:35,2:13])
rownames(Y) <- data[24:35,1]
Y <- scale(Y, center = colMeans(X), scale = apply(X,2,sd))

supplPC1 <- Y %*% loadings[, 1, drop=F]
supplPC2 <- Y %*% loadings[, 2, drop=F]

supplPC1 <- rbind(pc[,1, drop = F],supplPC1)
```

```
supplPC2 <- rbind(pc[,2, drop = F],supplPC2)
suppl <- c(rep("Active",23),rep("Suppl",35-23))
totalPC <- cbind(supplPC1, supplPC2, data[,"Area",drop = F], suppl)

ggplot(data = totalPC, aes(x = PC1, y = PC2, color = Area)) + geom_point() + geom_text(aes(color = fact
```



The graph shows that the areas located in south side tend to have lower pc1, the areas located in East side tend to have higher pc2, the areas located in west side tend to have medium pc1 and pc2, and areas located in north side tend to have high pc1. In conclusion, cities seems to share some common information according to their location.

b) Compute the quality of individuals representation, that is, the squared cosines given by:

$\cos^2(i,k) = \frac{z_{ik}^2}{d^2(x_i,g)}$

```
dist <- function(x1,x2){
  sum((x1-x2)^2)
}

X <- as.matrix(data[1:23,2:13])

X <- scale(X, center = T , scale = T)
```

```r
g <- colMeans(X)

cos <- data.frame()
for(i in 1:23){
  for(k in 1:12){
    cos[i,k] = pc[i,k]^2 / dist(X[i,],g)
  }
}

rownames(cos) <- data[1:23,1]
colnames(cos) <- paste0("PC",1:12)

print("Adding the squared cosines over all principal axes for a given
individual should be 1")
```

```
## [1] "Adding the squared cosines over all principal axes for a given\nindividual should be 1"
```

```r
sum(cos[3,])    # should sum up to 1
```

```
## [1] 1
```

```r
# What cities are best represented on the first two PCs?
rownames(cos)[which.max(cos[,1])]
```

```
## [1] "Rome"
```

```r
rownames(cos)[which.max(cos[,2])]
```

```
## [1] "London"
```

```r
# What cities have the worst representation on the first two PCs?
rownames(cos)[which.min(cos[,1])]
```

```
## [1] "London"
```

```r
rownames(cos)[which.min(cos[,2])]
```

```
## [1] "Stockholm"
```

```r
cos[,1:4]
```

```
##                   PC1          PC2         PC3          PC4
## Amsterdam  0.02474831 9.037408e-01 0.005236924 3.842958e-02
## Athens     0.97830645 1.465844e-02 0.005337778 1.390573e-03
## Berlin     0.32789958 1.071347e-03 0.334194626 1.222994e-02
## Brussels   0.21639751 7.527801e-01 0.012557007 1.582759e-04
## Budapest   0.46337591 4.885264e-01 0.041466618 2.105434e-03
## Copenhagen 0.80588015 9.123692e-02 0.073075813 1.176775e-02
## Dublin     0.03411320 9.551775e-01 0.004258404 1.195616e-04
## Elsinki    0.95654320 1.253419e-02 0.020659730 3.508325e-03
## Kiev       0.41732984 5.737380e-01 0.004146449 1.811489e-03
## Krakow     0.64509341 3.117546e-01 0.030562745 5.436012e-04
## Lisbon     0.92554429 7.132255e-02 0.002157697 5.560264e-04
## London     0.00131785 9.824220e-01 0.002843919 1.073178e-03
## Madrid     0.93424771 2.753176e-02 0.012062772 2.485878e-02
## Minsk      0.84071389 1.552234e-01 0.000418832 2.695539e-03
## Moscow     0.71081284 2.822692e-01 0.005382114 1.717765e-06
## Oslo       0.97838231 8.605543e-03 0.007806583 3.252313e-03
```

```
## Paris      0.69481859 2.777373e-01 0.004196019 2.155078e-03
## Prague     0.01987080 8.148950e-01 0.098405324 1.684971e-02
## Reykjavik  0.71527677 2.826756e-01 0.000108358 1.233751e-03
## Rome       0.99549987 2.964543e-03 0.001231151 5.446829e-06
## Sarajevo   0.07819664 2.987590e-01 0.389052585 1.586138e-02
## Sofia      0.18627345 6.745387e-01 0.061906201 2.438439e-03
## Stockholm  0.92423563 2.900338e-06 0.070997512 3.121254e-04
```

Rome and London are best represented on PC1 and PC2 while London and Stockholm have the worst representation on PC1 and PC2.

c) Compute the contributions of the individuals to each extracted PC.

$ctr(i,k) = \frac{m_i z_{ik}^2}{\lambda_k} * 100$

```
ctr <- data.frame()
for(i in 1:23){
  for(k in 1:12){
    ctr[i,k] =  (pc[i,k]^2 / (n-1)) / eval[k] * 100
  }
}

rownames(ctr) <- data[1:23,1]
colnames(ctr) <- paste0("PC",1:12)

sum(ctr[,1])   # For a given component, the sum of the contributions of all observations is equal to 10
```

```
## [1] 100
```

```
ctr[,1:4]
```

```
##                    PC1          PC2        PC3          PC4
## Amsterdam   0.022509389 4.425554e+00  0.3752909  9.079967206
## Athens     25.249412071 2.036899e+00 10.8545150  9.323317492
## Berlin      0.036216364 6.370885e-04  2.9082851  0.350904333
## Brussels    0.174118494 3.261117e+00  0.7960720  0.033083230
## Budapest    1.216057639 6.902625e+00  8.5741849  1.435366347
## Copenhagen  0.934709699 5.697475e-01  6.6781085  3.545685387
## Dublin      0.111569397 1.681947e+01  1.0973444  0.101581521
## Elsinki     7.120549993 5.023550e-01 12.1173352  6.784364061
## Kiev        1.281346111 9.484319e+00  1.0030831  1.444851066
## Krakow      0.692408290 1.801599e+00  2.5846726  0.151572536
## Lisbon     13.702914482 5.685231e+00  2.5169800  2.138511623
## London      0.001452098 5.828188e+00  0.2468998  0.307186563
## Madrid      7.218867870 1.145372e+00  7.3439161 49.898483504
## Minsk       4.582193987 4.554996e+00  0.1798617  3.816553334
## Moscow      5.240284554 1.120388e+01  3.1262670  0.003289757
## Oslo        4.776937527 2.262167e-01  3.0031395  4.125093151
## Paris       0.880944827 1.895907e+00  0.4191682  0.709807693
## Prague      0.005193057 1.146608e+00  2.0262818  1.143932947
```

```
## Reykjavik   9.671498999 2.057849e+01  0.1154394  4.333588025
## Rome       12.660033938 2.029817e-01  1.2336114  0.017994418
## Sarajevo    0.011676896 2.401960e-01  4.5774239  0.615291054
## Sofia       0.076296912 1.487539e+00  1.9978537  0.259458701
## Stockholm   4.332807405 7.320502e-05 26.2242659  0.380116049
```

```r
# The most influential cities on PC1 and PC2
rownames(ctr)[which.max(ctr[,1])]
```

```
## [1] "Athens"
```

```r
rownames(ctr)[which.max(ctr[,2])]
```

```
## [1] "Reykjavik"
```

Athens is the most influential city on PC1 and Reykjavik is the most influential city on PC2

4) Studying the cloud of variables

a) Calculate the correlation of all quantitative variables (active and supplementary) with the principal components.

```r
X <- as.matrix(data[1:23, -c(1,18)])
X <- scale(X, center = T, scale = T)
rownames(X) <- data[1:23,1]

# Ignore=========================================
#
# cor <- cor(X)
# loadings <- eigen(cor)$vectors
#
# pc <- X %*% loadings[,1:16, drop = F]
#
# # check with prcomp
#
# head(prcomp(X, scale. = T)$x,1)
# head(pc,1)
#
# # why princomp is different?
#
# #head(princomp(X, cor = T)$scores, 1)
#
#
#
# colnames(pc) <- paste0("PC",1:16)
# Ignore=========================================



corr <- cor(X[,1:16],pc[,1:12])   # corr <- cor(X, pc)


corrsq <- corr^2

# sum of the sum of the squared coeffcients of correlation between a variable and all the principal com
```

```r
sum(corrsq[1,])
```

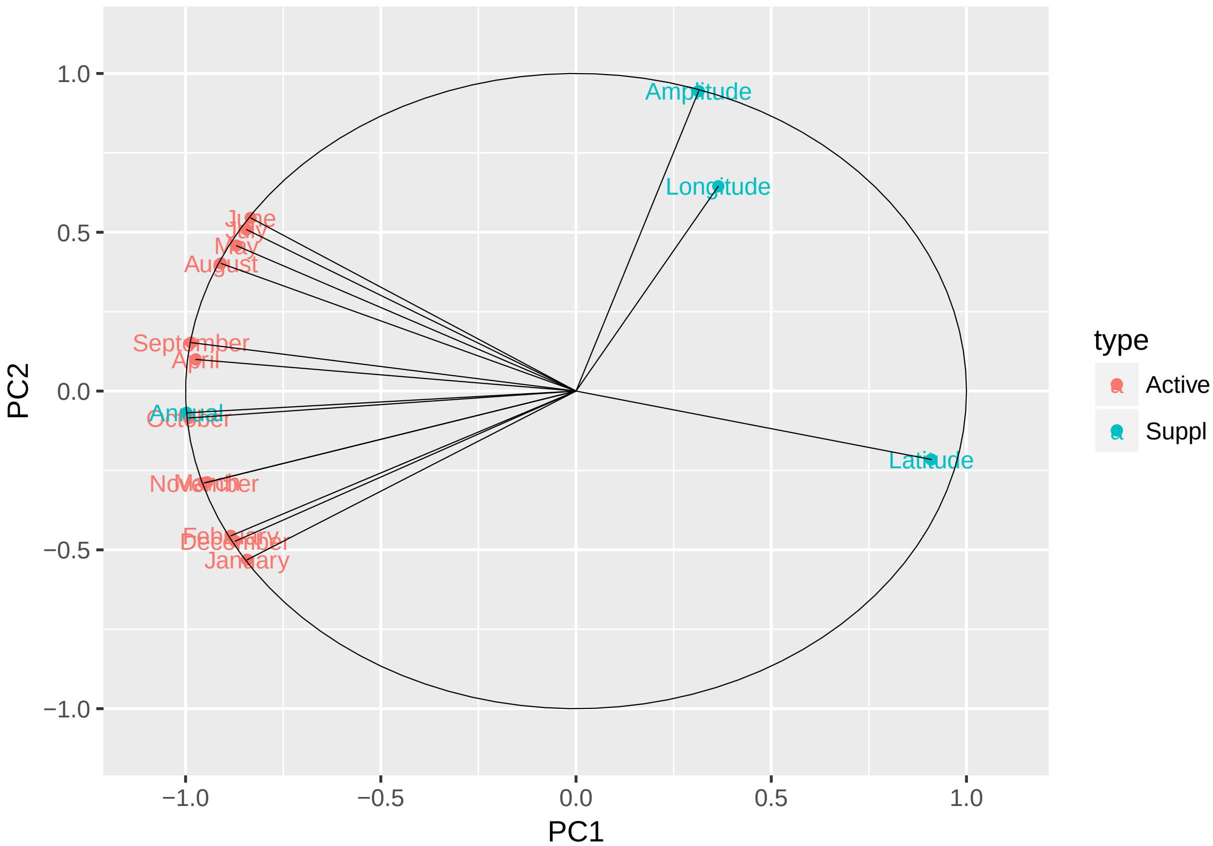
```
## [1] 1
```

```r
corr[,1:4]
```

```
##                  PC1         PC2           PC3           PC4
## January   -0.8424506 -0.53135762  6.776712e-02 -1.168876e-02
## February  -0.8842848 -0.45583250 -3.466272e-03 -8.371472e-02
## March     -0.9450521 -0.28731281 -1.207952e-01 -7.781832e-02
## April     -0.9738876  0.09956500 -1.982562e-01 -2.486767e-02
## May       -0.8698517  0.45781159 -1.560861e-01  7.682512e-02
## June      -0.8333141  0.54532195  4.954763e-02 -9.575733e-05
## July      -0.8441626  0.50866195  1.536892e-01 -4.360395e-02
## August    -0.9092443  0.40192442  8.750079e-02 -4.439218e-02
## September -0.9856254  0.15253617  2.262618e-02 -5.193042e-03
## October   -0.9916246 -0.08476471 -6.661858e-05  7.173534e-02
## November  -0.9523567 -0.28941418  4.422075e-02  6.973744e-02
## December  -0.8731191 -0.47286559  8.480816e-02  6.829538e-02
## Annual    -0.9975483 -0.06845254  4.566805e-03  3.575494e-06
## Amplitude  0.3140756  0.94441398  3.918835e-02 -5.742427e-03
## Latitude   0.9099106 -0.21543731  1.819845e-01  5.929010e-02
## Longitude  0.3644584  0.64497259 -3.643387e-02  2.473234e-01
```

b) Make a Circle of Correlations plot between the PCs and all the quantitative variables

```r
type <- c(rep("Active",12),rep("Suppl",4))
table <- cbind(as.data.frame(corr[,1:2, drop = F]), type)
table <- as.data.frame(table)

dat = data.frame(x=runif(1), y=runif(1)) # for circle

p <- ggplot(data = table, aes(x = PC1, y = PC2, color = type, label = rownames(table))) + geom_point() +
  xlim(c(-1.1,1.1)) + ylim(c(-1.1,1.1)) +
  geom_segment(data=table, aes(x=0, xend=PC1, y=0, yend=PC2), color="black", arrow = arrow(length=unit(


g<-p+annotate("path",
   x=0+1*cos(seq(0,2*pi,length.out=100)),
   y=0+1*sin(seq(0,2*pi,length.out=100)),
   size = 0.2)

g
```



c) Based on the above parts (a) and (b), how are the active and supplementary variables related to the components?

Active variables tend to have lower PC1 scores and supplementary variables tend to have higher PC1 scores. Variable Annual is the only exception for this case since Annual is supplementary variable but has lower PC1 score. This suggests that Months(January~Decemer) variables and Anuual variables share smiliar characteristic while supplementary variables do too. We can make up a story by considering PC1 as temperature and PC2 as precipitation. In spring and winter seasons (January~March and October~December), the amounts of snow differ based on latitude. In summer and fall seasons(April~September), countries in western europe rains more often than the ones on the eastern sides.

5) Conclusions

From the graphs that we draw, we can conclude that variable Annual is highly correlated to other months variable.(January, February, . . .) This is intuitively obvious since the variable Annual is the mean of other months variable. Also, we can conclude that variable Amplitude and variable Longtitude is fairly correlated. April~September and Amplitude and Longtitude has positive PC2 values while January~March and October~December and Latitude has negative PC2 values. In conclusion, summer and fall seasons(April~September) are affected by amplitude and longtitude while spring and winter seasons(January~March and October~December) are affected by latitude.