PCA

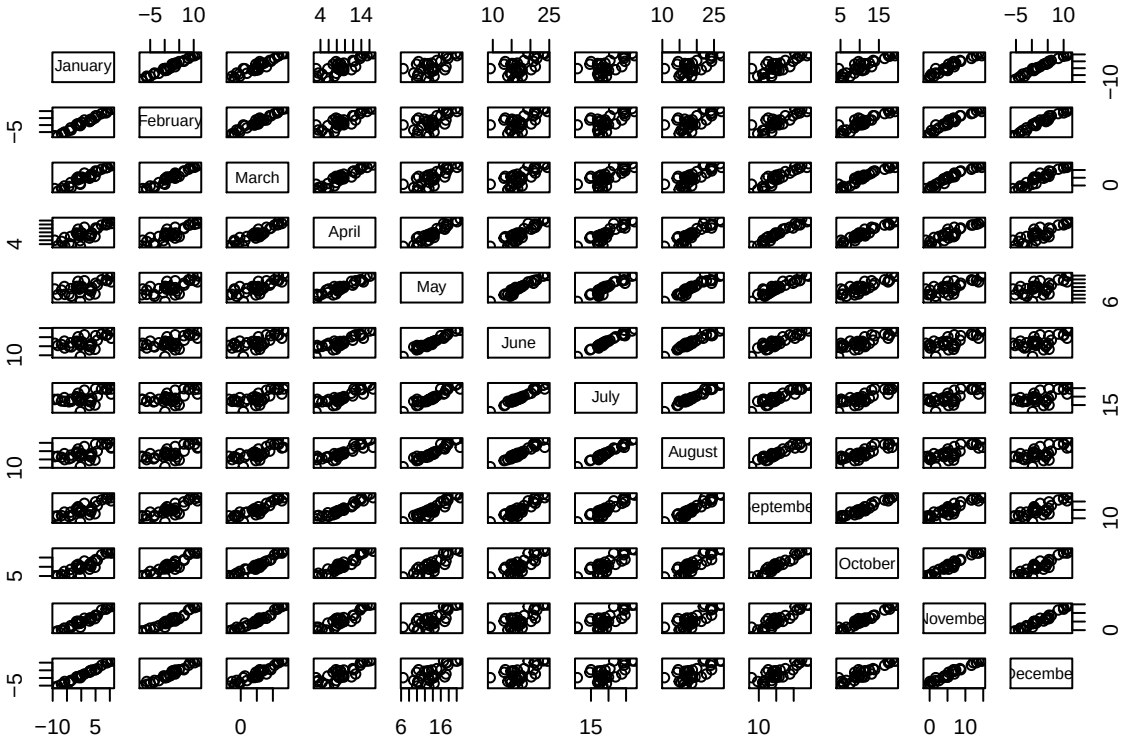
Chase Enzweiler

9/14/2017

Exploratory Phase

From looking at various scatter plots of the variables of the monthly temperatures we notice a strong positive correlation between neighboring months and weaker positve correlations between months of different seasons.

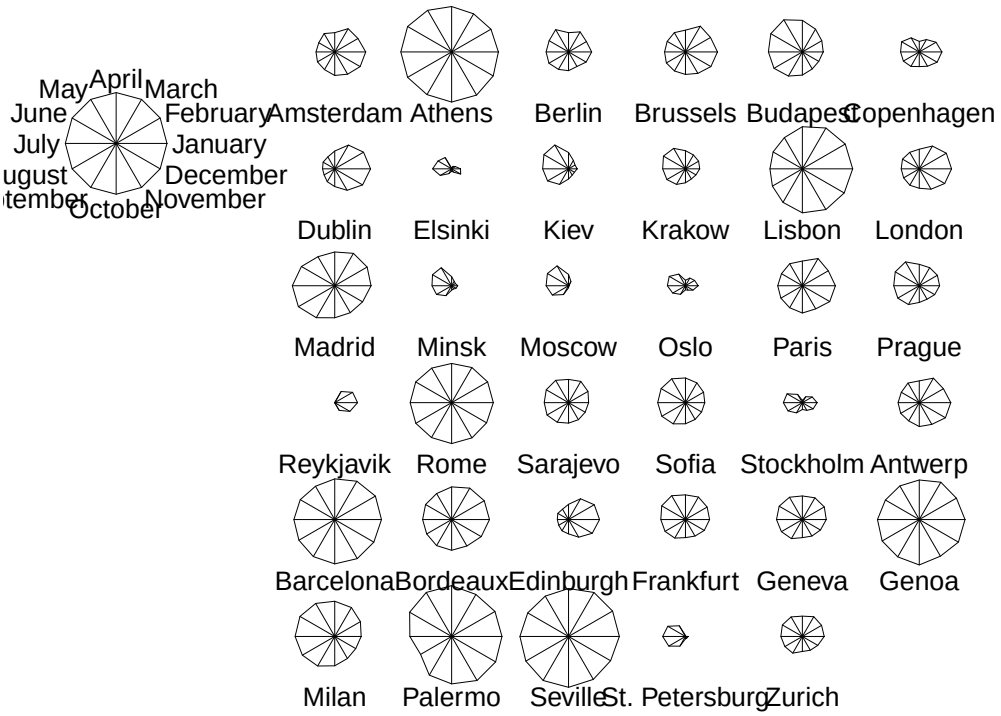
```
# pairs plot
temperature <- read.csv("~/Desktop/stat 154/temperature.csv")
pairs(temperature[,2:13])
```



By looking at the pairs plot of the montly temperatures we confirm that neighboring months have very strong positive correlations while we also notice that all of our variables have positive correlation with all the other monthly temperatures.

We can also use a stars plot to observe how the individual city monthly temperatures relate to other cities

```
# stars plot
row_character_vect <- as.matrix(temperature[,1])
row_character_vect <- as.vector(row_character_vect)

stars(temperature[, 2:13], labels = row_character_vect, key.loc = c(-2,12), flip.labels = FALSE)
```



Notice that cities such as Athens, Seville, and Barcelona have some of the warmest temperatures year round, while cities like Elsinki, Oslo, and St. Petersburg ahve some of the lowest. Also notice interesting symmetry between St. Petersburg and Rekjavic where Rekjavic has the coldest summer months but not the coldest winter months, while St.Petersburg is the opposite.
Also looking at our monthly temperature variables, some variables have data that is skewed.

```
# summary of month temperatures
summary(temperature$January)
```

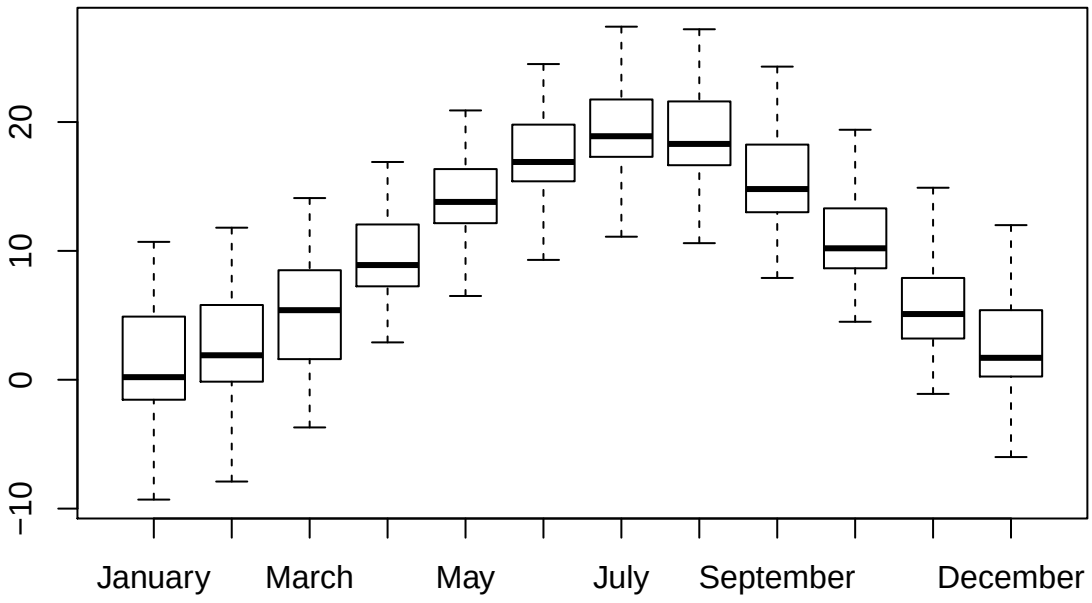
```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##  -9.300  -1.550   0.200   1.346   4.900  10.700
```

```
summary(temperature$March)
```

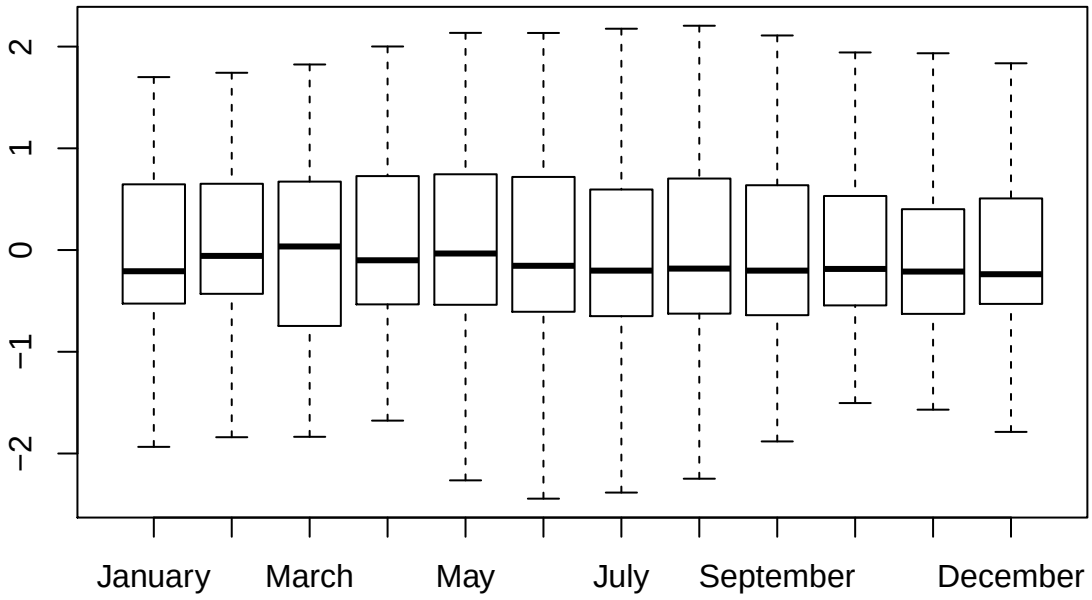
```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##  -3.700   1.600   5.400   5.229   8.500  14.100
```

From these statistics the month of january has data that is right skewed. Also March is the month that has data closest to being normally distributed.

```
# boxplot
boxplot(temperature[,2:13])
```




```
# boxplot of standardized data

boxplot(scale(as.matrix(temperature[2:13])))
```



From these boxplots we see that many of the variables hae data that is slightly skewed right. We also notice that March is not normally distributed and is the only month with a slight right skew.

Also looking at the variance of the temperatures of each month, we notice that January and February have the highest variances. The variances decrease month by month until its lowest in May then increases again.

```
# variances
var(temperature$January)
```

```
## [1] 30.27373
```

```
var(temperature$February)
```

```
## [1] 30.23852
```

```
var(temperature$May)
```

```
## [1] 10.71634
```

"Explain to the reader why they need to know these things"

1.) Calculation of PCA Outputs

Our active individuals are the Capitals of each of their respective countries and are the first 23 columns of our data and our active variables are the 12 monthly temperatures. We also want to standardize our data to have mean zero and standard deviation one.

```
# data of active individuals and variables

active_temp <- as.matrix(temperature[1:23, 2:13])
row.names(active_temp) <- row_character_vect[1:23]

# standardized active data

s_active_temp <- scale(active_temp, center = TRUE, scale = TRUE)
```

a.) Obtain the loadings and store them in a matrix, include row and column names. Display the first four loadings

```
n <- dim(s_active_temp)[1]
p <- dim(s_active_temp)[2]

# calculate the sample correlation matrix

R <- (1/(n-1)) * t(s_active_temp) %*% s_active_temp

decomp_R <- eigen(R)

# matrix of eigenvectors(loadings)
V <- decomp_R$vectors

colnames(V) <- paste("v", 1:12, sep = "")
rownames(V) <- colnames(active_temp)
V[,1:4]
```

```
##                   v1          v2            v3           v4
## 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
```

```
# has dimnsion p x r = 12 x 12

# note need to name rows dont know what to name them
```

b. Obtain the principal components and store them in a matrix, include row and column names. Display the first four PCs

```
# calculate principal components (scores) matrix
# has dim nxr = ind x rank
Z <- s_active_temp %*% V

colnames(Z) <- paste("PC", 1:12)

Z[,1:4]
```

```
##                   PC 1         PC 2        PC 3        PC 4
## 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. Obtain the eigenvalues and store them in a vector. Display the entire vector, and compute their sum.

```
e_value_vect <- decomp_R$values

e_value_vect
```

```
##  [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
```

```
sum(e_value_vect)
```

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

2.) Choosing the number of dimensions to retain/examine

a.) Make a summary table of the eigenvalues: eigenvalue in the first column (each eigenvalue represents the variance captured by each component); percentage of variance in the second column; and cumulative percentage in the third column. Comment on the table.

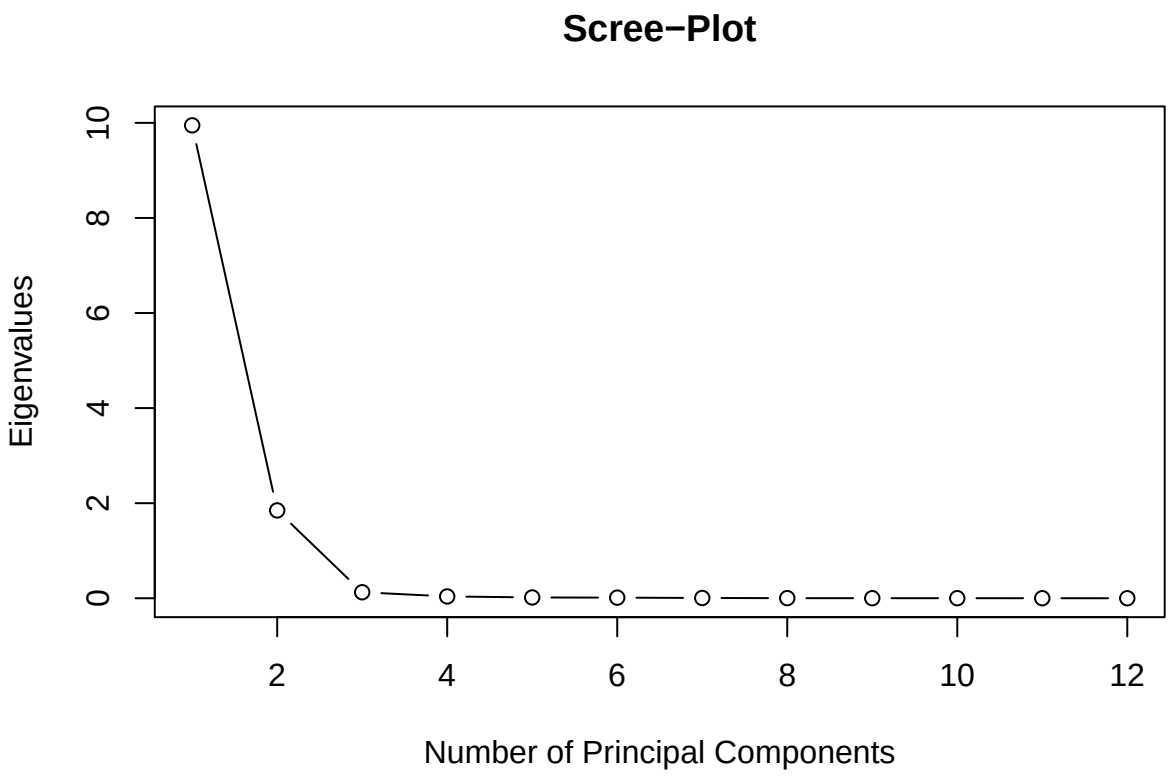
```
# proportion of explained inertia (percentage)
proportion <- (e_value_vect / p) * 100

#cumulative proportion of explained inertia
cum_prop <- c(proportion[1])

for(i in 2:12){
  cum_prop <- append(cum_prop, sum(proportion[1:i]))
}

# summary table of eigen values
eigen_sum_tab <- cbind(e_value_vect, proportion, cum_prop)

rownames(eigen_sum_tab) <- paste("comp", 1:12)

eigen_sum_tab <- as.table(eigen_sum_tab)

eigen_sum_tab
```

```
##         e_value_vect   proportion     cum_prop
## comp 1  9.947750e+00 8.289792e+01 8.289792e+01
## comp 2  1.847649e+00 1.539707e+01 9.829499e+01
## comp 3  1.262558e-01 1.052132e+00 9.934712e+01
## comp 4  3.829345e-02 3.191121e-01 9.966623e+01
## comp 5  1.670941e-02 1.392451e-01 9.980548e+01
## comp 6  1.283304e-02 1.069420e-01 9.991242e+01
## comp 7  5.830293e-03 4.858578e-02 9.996101e+01
## comp 8  2.031893e-03 1.693244e-02 9.997794e+01
## comp 9  1.023452e-03 8.528764e-03 9.998647e+01
## comp 10 9.527707e-04 7.939756e-03 9.999441e+01
## comp 11 5.367834e-04 4.473195e-03 9.999888e+01
## comp 12 1.341917e-04 1.118264e-03 1.000000e+02
```

we see from this table that the first principal component explains about 82 percent of the variance, or inertia, and the second principal component explains about 15 percent. Cumulatively with our first two principal components about 98 percent of the total variance is explained. the proportion of variance explained for the next principal components are much smaller percentages and this is a good indicator that we should use our first two principal components.

b.) Create a scree-plot (with axis labels) of the eigenvalues. What do you see? How do you read/interpret this chart?

```
# create a scree-plot from scratch

plot(1:12, e_value_vect, ylab = "Eigenvalues", type = "b", xlab = "Number of Principal Components", mai
```



This scree-plot plots the values of the eigenvalues against the number of principal components that we have. Our eigenvalues represent the variance of each of our 12 principle components. In order to reduce our amount of dimensions while maintaing a large portion of our variance we look for an elbow in the graph where the curve levels off or follow kaiser's rule. looking for an elbow we see the cuve leveling off after our 2nd principal component therefore we determine our elbow is at the 2nd component and we should keep the first two components.

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

I would choose 2 dimensions which would be the first two principal components. Doing this reduces our dimensions from 12 which is what we are striving for in PCA and still has a large proportion of variance explained at a total of around 98 percent. We noticed an elbow at the second component from our scree-plot and adding one more dimension is not worth only the 1 percent more of the variance that component will give us.

3.) Studying the Cloud of Individuals

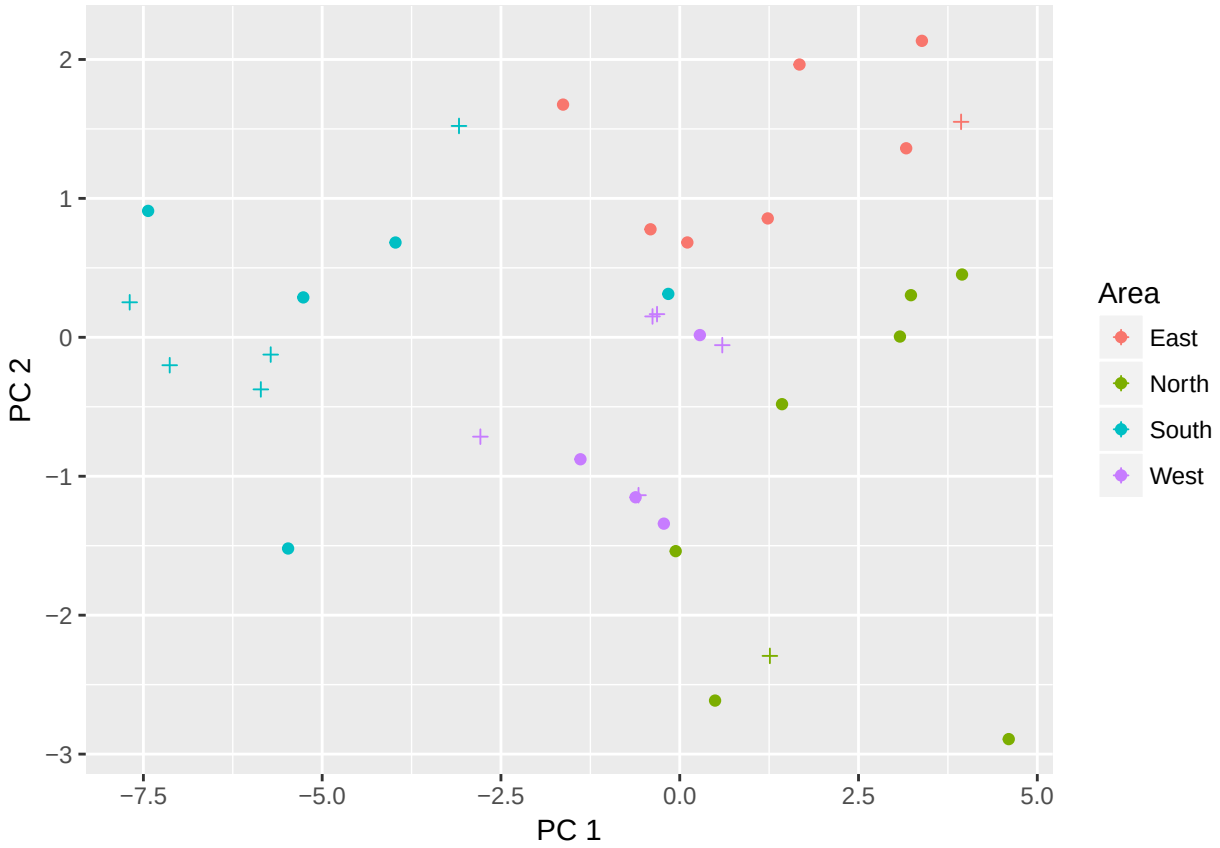
A.) Create a scatter plot of the cities on the 1st and 2nd PCs. * In this plot, you should also project the supplementary cities. * Make sure to add a visual cue (e.g. size, font, shape) to differentiate between active and supplementary cities. * Color the cities according to the variable Area. * Comment on general patterns, as well as on particular patterns.

```
# incorporate the supplementary cities
# we have to mean center and scale the supplementary cities according to our standardization of active
# we standardize using mean and sd of active cities

active_means <- colMeans(active_temp)
active_sd <- apply(active_temp, 2, sd)
supp_temp <- as.matrix(temperature[24:35 , 2:13])
row.names(supp_temp) <- row_character_vect[24:35]

# standardized with respect to active cities
```

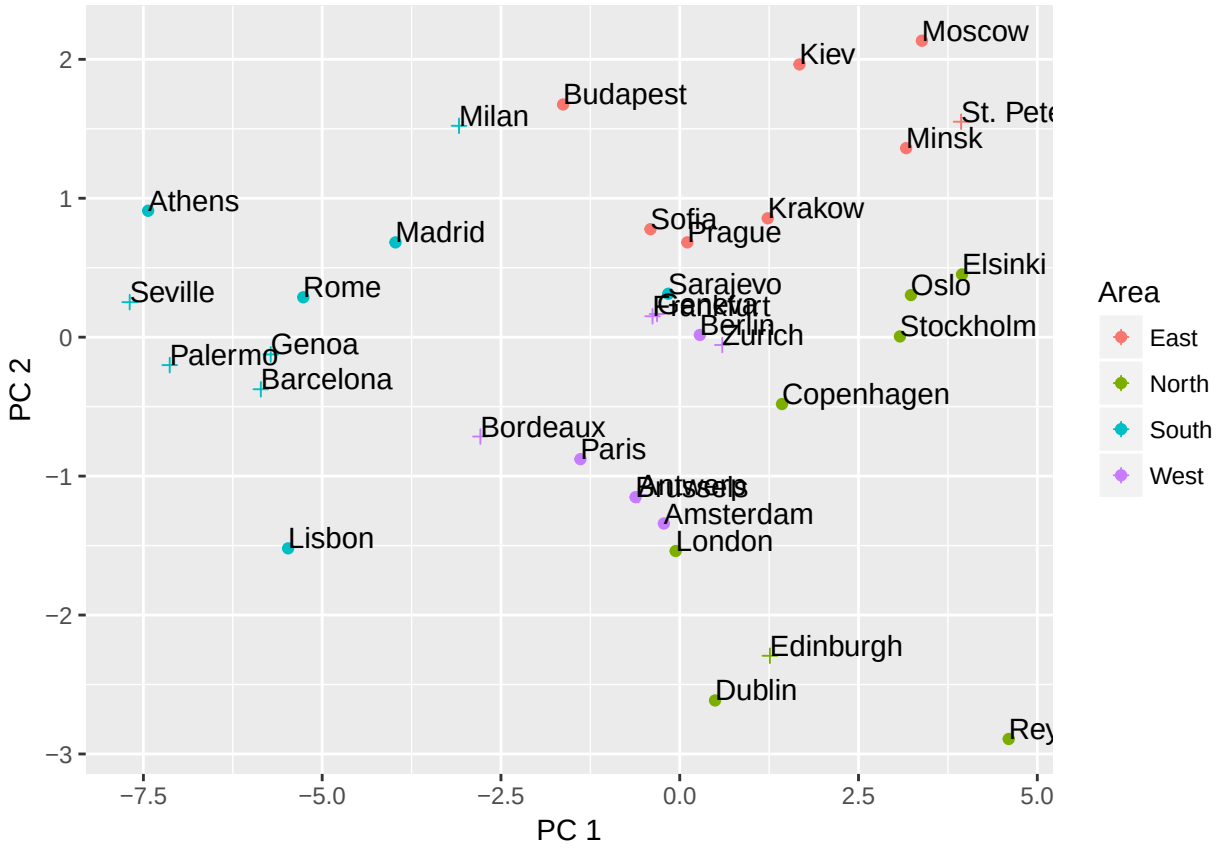
```
s_supp_temp <- t(t(supp_temp) - active_means)

for(i in 1:12){
  s_supp_temp[,i] <- s_supp_temp[,i] / active_sd[i]
}

library(ggplot2)
```

```
## Warning: package 'ggplot2' was built under R version 3.3.2
```

```
# supplemental projections
Z_s <- s_supp_temp %*% V

Z_df <- as.data.frame(Z)
Z_df$Area <- temperature[1:23, 18]



Z_s_df <- as.data.frame(Z_s)
Z_s_df$Area <- temperature[24:35, 18]

#added
colnames(Z_s_df) <- colnames(Z_df)

ggplot(data = Z_df, aes(x = `PC 1`, y = `PC 2`, group = Area)) + geom_point(aes(color = Area)) + geom_p
```



```
ggplot(data = rbind(Z_df, Z_s_df), aes(x = `PC 1`, y = `PC 2`, group = Area)) + geom_point(data = Z_df,
```



In this plot our active cities are represented as circular dots and our supplementary cities are represented as crosses(+). We see the general grouping by the Area of the cities with the Eastern cities grouping in the top right corner while the Northern cities are grouped far from the southern cities. We see one blue southern city Sarajevo closer to the groupings of the western and eastern cities than the southern cities.

B.) Compute the quality of individuals representation, that is, the squared cosines. Store the squared cosines in a matrix or data frame, include row and column names. Display the first four columns. What cities are best represented on the first two PCs? What cities have the worst representation on the first two PCs?

```
# we want matrix of Zik^2 / d^2(xi,g)
# the centroid of centered data is the origin
#since the centroid is zero the euclidean squared distance we just need to square the observations

dist2centroid <- apply(s_active_temp, 1, function(x) sum(x*x))

cos2_matrix <- sweep(Z^2, 1, dist2centroid, FUN = "/")

cos2_matrix[,1:4]
```

```
##                 PC 1         PC 2        PC 3         PC 4
## 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
```

The cities Rome, Athens, and Helsinki are best represented on the first principal component, while Dublin, london, and Amsterdam are best represented on the second principal component.

C.) Compute the contributions of the individuals to each extracted PC. Store the individuals contributions in a matrix or data frmae, include row and column names. Display the first four columns. Are there any influential cities on the first two PCs?

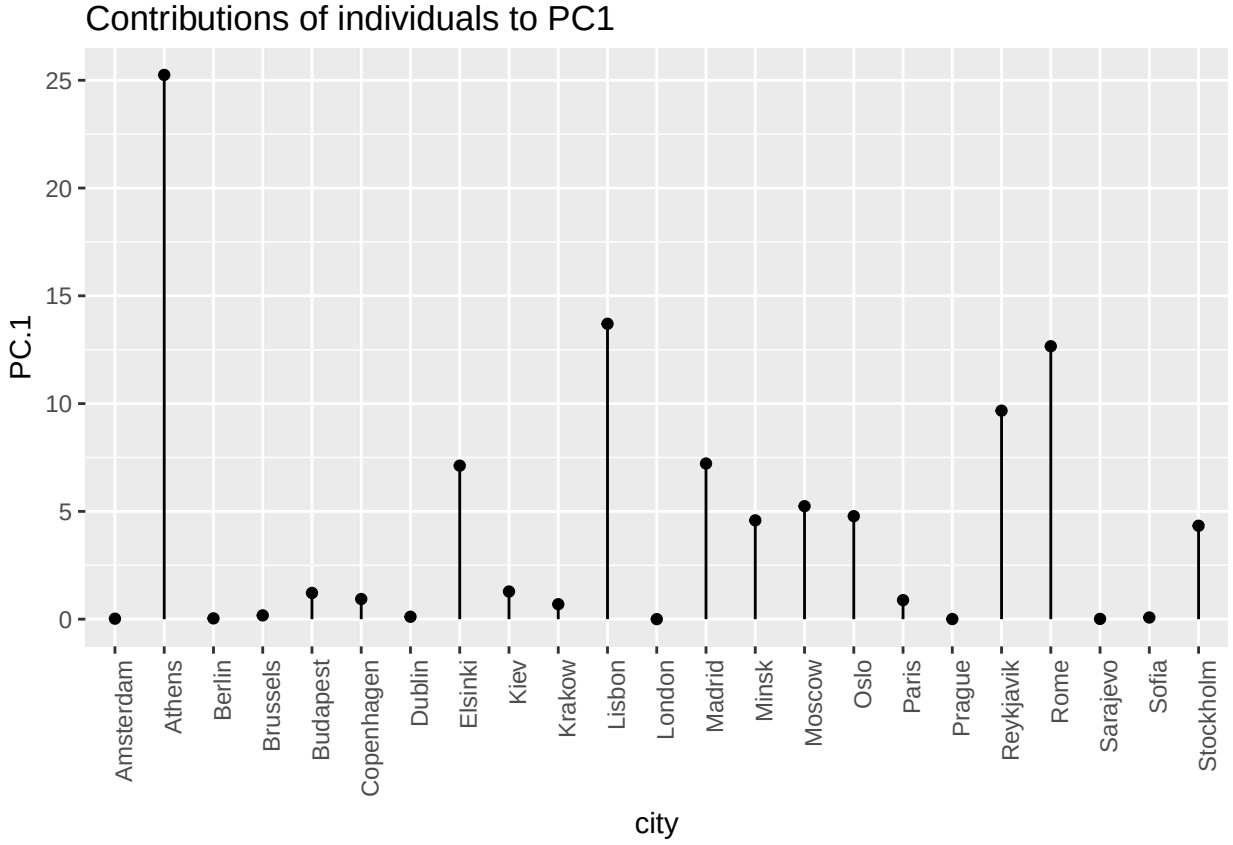
```
# ctr(i,k) = (mi * Zik^2)/lamda k
# percentage of inertia explained by individual i on component k

ctr <- (1 / (n - 1)) * Z^2

ctr <- t(t(ctr) / e_value_vect) * 100

ctr[,1:4]
```

```
##                   PC 1         PC 2       PC 3         PC 4
## 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
```

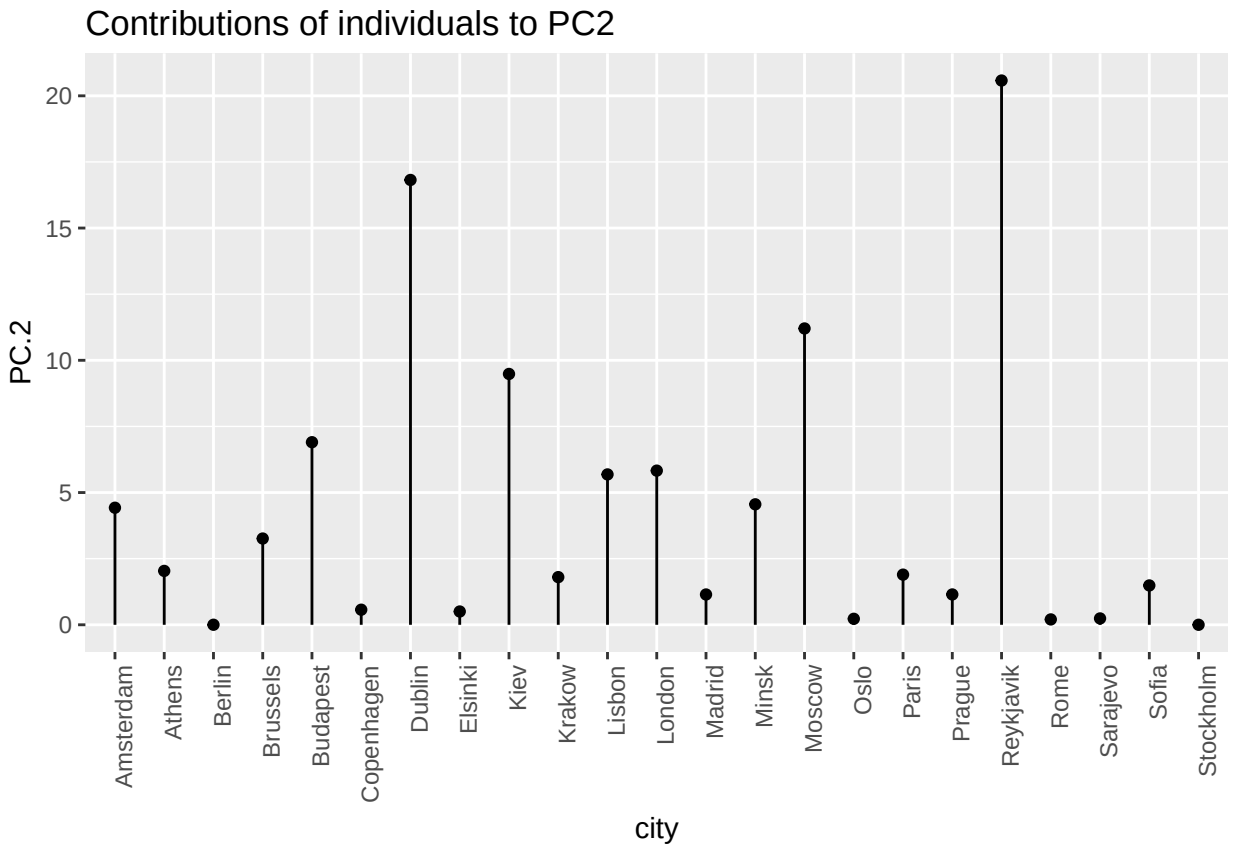
```
## Sarajevo    0.011676896 2.401960e-01  4.5774239  0.615291054
## Sofia       0.076296912 1.487539e+00  1.9978537  0.259458701
## Stockholm   4.332807405 7.320502e-05 26.2242659  0.380116049
```

influential cities to the first PC are Athens, Lisbon, and Rome which account for about 25%, 13%, and 12% of the explained intertia or variance respectively. Influential cities to the second PC are Kiev with 9% explained variance and Budapest with about 7% of the explained variance of the principal component.

```
# plot the contributions
ctr_df <- data.frame(ctr)
ctr_df$city <- 1:nrow(ctr_df)
ctr_df$zeros <- rep(0, nrow(ctr_df))

ggplot(data = ctr_df, aes(x = city, y = PC.1)) + geom_point() + geom_segment(aes(x = city, xend = city,
theme(axis.text.x = element_text(angle = 90, hjust = 1)) + ggtitle("Contributions of individuals to PC1
```




```
ggplot(data = ctr_df, aes(x = city, y = PC.2)) + geom_point() + geom_segment(aes(x = city, xend = city,
theme(axis.text.x = element_text(angle = 90, hjust = 1)) + ggtitle("Contributions of individuals to PC2
```



4.) Studying the cloud of Variables

a. Calculate the correlation of all quantitative variables (active and supplementary) with the principal components. Store the correlations in a matrix or data frame, include row and column names. Display the first four columns.

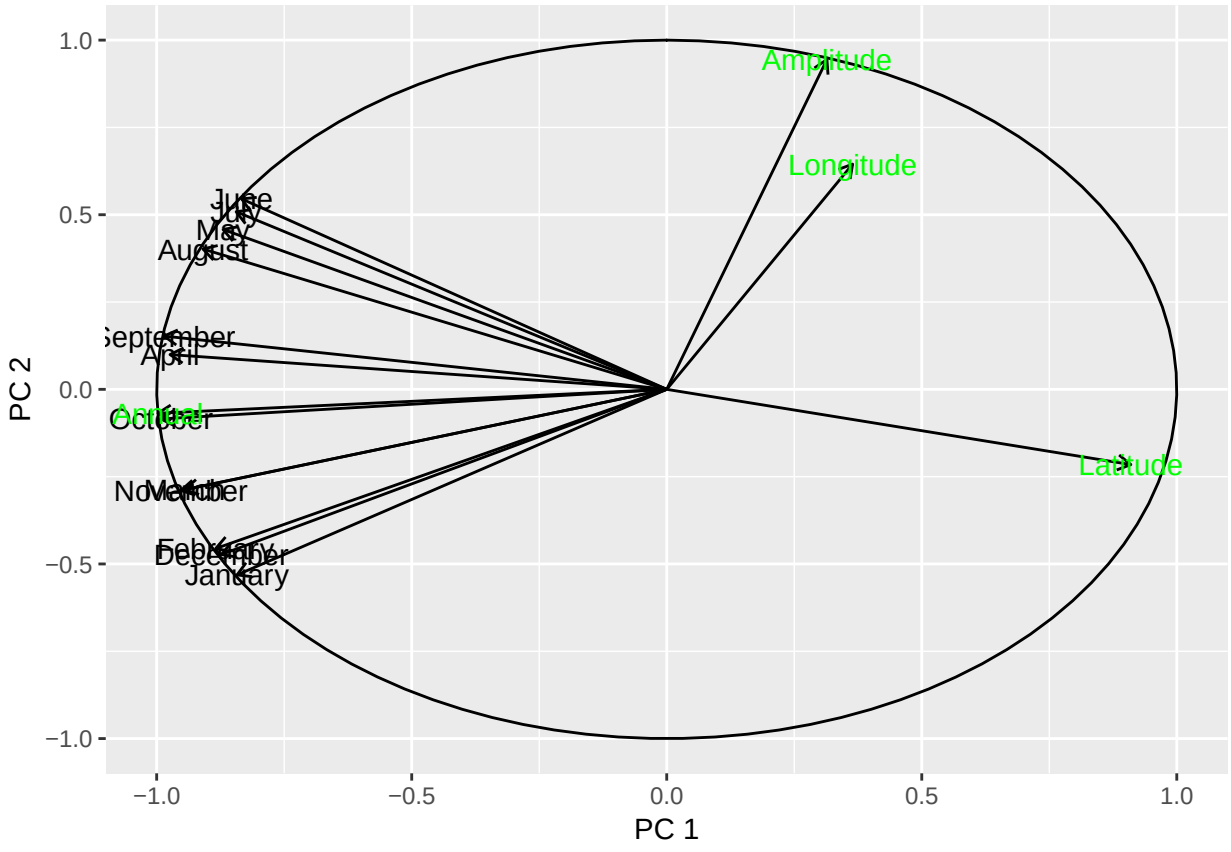
```
# correlation matrix of variables with PC's
#standardized supplementary variables
supp_var <- temperature[1:23,14:17]

stand_supp_var <- scale(supp_var)

# I am going to include only the active individuals

# matrix of both active and supplementary var
s_both_temp <- cbind(s_active_temp, stand_supp_var)


var_pc_corr <- cor(s_both_temp, Z)

var_pc_corr[,1:4]
```

```
##                 PC 1        PC 2          PC 3          PC 4
## January   -0.8424506 -0.53135762  6.776712e-02 -1.168876e-02
## February  -0.8842848 -0.45583250 -3.466272e-03 -8.371472e-02
```

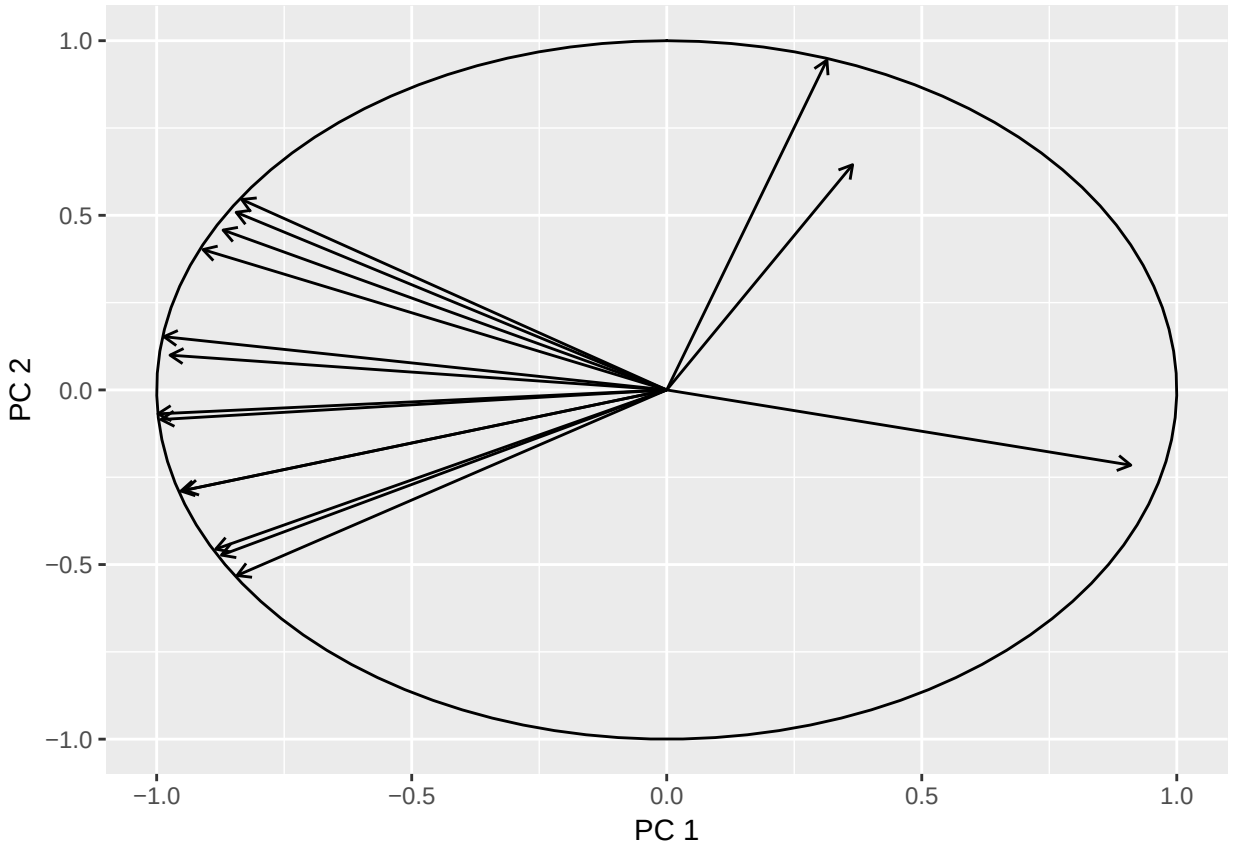
```
## March     -0.9450521 -0.28731281 -1.207952e-01 -7.781832e-02
## April     -0.9738876  0.09956500 -1.982562e-01 -2.486767e-02
## May       -0.8698517  0.45781159 -1.560861e-01  7.682512e-02
## June      -0.8333141  0.54532195  4.954763e-02 -9.575733e-05
## July      -0.8441626  0.50866195  1.536892e-01 -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 * For visualization purposes, include the circumference of a circle of radius one. * Represent each variable in the plot as an arrow. * Use color to distinguish between active and supplementary variables. * Also include names of variables.

```
# circle of correlations

var_pc_corr <- as.data.frame(var_pc_corr)

radians <- seq(0, 2*pi, length = 100)
circle_frame <- data.frame(x = sin(radians), y = cos(radians))

ggplot(data = var_pc_corr, aes(`PC 1`, `PC 2`)) + geom_segment(data = var_pc_corr, aes(x = 0, y = 0, xe
  geom_text(data = var_pc_corr[1:12,], label = rownames(var_pc_corr[1:12,])) + geom_text(data = var_pc_
```



```
ggplot(data = var_pc_corr, aes(`PC 1`, `PC 2`)) + geom_segment(data = var_pc_corr, aes(x = 0, y = 0, xe
```



In our graph the labels in black represent the active variables while the labels in green represent the supplementary variables.

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

We notice that all our active variables are positively correlated with one another and are all well represented on the first principle component. We can summarize the first principal component as "average temperature annually" since we can see our annual arrow close to the first pc. The months September, October and April are more closely linked than the others to the first component and therefore they represent the best annual temperatures.Apart from the average annual temperature, another supplementary quantitative variable is linked to PC1: latitude. The correlation between latitude and the first PC is worth 0.9099106, which means that the cities that are further north (higher latitude) have a higher coordinate on the first component and are therefore the coldest cities.

5.) Conclusions

Temperatures can be summarized by the first two Principal Components: PC1 can be labeled "average annual temperature", and PC2 "thermal amplitude". We conclude that Southern European cities are characterized by high temperatures throughout the year, Western European cities are characterized by average temperatures throughout the year, Northern European cities are characterized by cold temperatures, especially during summer, Eastern European cities are characterized by cold temperatures, especially during winter.