Problem Set 2: PCA

Stat 154, Fall 2017, Prof. Sanchez

Due date: Wed Sep-20 (before midnight)

The purpose of this assignment is to perform an exhaustive Principal Component Analysis (PCA) from scratch in R. You will investigate the climates of different European countries using the data set `temperature.csv`, available in the `data/` folder of the github repository.

Preliminary Steps

```
# (optional) packages
library(reshape2)
library(ggplot2)

# import data
dat <- read.csv("temperature.csv", row.names = 1)
```

Identifying active and supplementary elements for both individuals and variables, as well as forming the main data matrix:

```
# individuals
active_indivs <- 1:23
supplementary_indivs <- 24:nrow(dat)

# variables
active_vars <- (names(dat))[1:12]
supplementary_vars <- c('Annual', 'Amplitude', 'Latitude', 'Longitude', 'Area')

# 'n' and 'p'
n <- length(active_indivs)
p <- length(active_vars)

# standardized data
Active <- dat[active_indivs,active_vars]
X <- scale(Active)
```

1) Calculation of basic PCA outputs (30 pts)

a. Obtain the loadings and store them in a matrix, include row and column names. Display the first four loadings (10 pts).

One way to get the primary outputs for PCA is via the EVD of the correlation matrix:

$$\text{EVD}(\mathbf{R}) = \mathbf{V}\mathbf{\Lambda}\mathbf{V}^\top$$

where $\mathbf{V}$ is the matrix of eigenvectors, and $\mathbf{\Lambda}$ is the diagonal matrix of eigenvalues. The matrix of principal components $\mathbf{Z}$ is thus obtained as:

$$\mathbf{Z} = \mathbf{X}\mathbf{V}$$

```
# correlation matrix
R <- (1/(n-1)) * t(X) %*% X

# EVD of R
EVD <- eigen(R)

# matrix of loadings or eigenvectors (with names)
# (displaying the first 4 eigenvectors)
V <- EVD$vectors
rownames(V) <- colnames(X)
colnames(V) <- paste0('V', 1:ncol(V))
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
```

b. Obtain the principal components and store them in a matrix, include row and column names. Display the first four PCs (10 pts).

```
# matrix of PCs (with names)
# (displaying first 4 PCs)
Z <- X %*% V
rownames(Z) <- rownames(X)
colnames(Z) <- paste0('PC', 1:ncol(Z))
Z[ ,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. Obtain the eigenvalues and store them in a vector. Display the entire vector, and compute their sum. (10 pts)

```
# vector of eigenvalues
eigenvalues <- EVD$values
eigenvalues
```

```
##  [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(eigenvalues)
```

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

With standardized data, the sum of eigenvalues is equal to the number of (active) variables:

$$\sum_{k=1}^{p} \lambda_k = p$$

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

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. (10 pts)

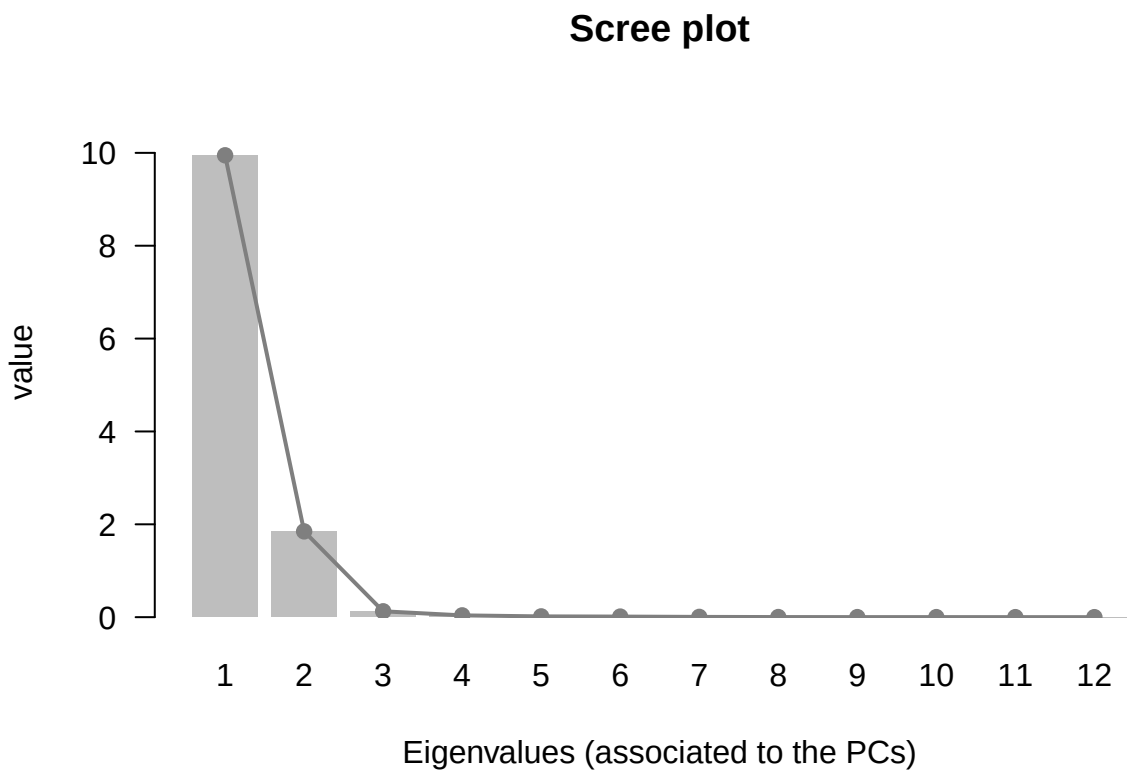
```
# summary table of eigenvalues
eigs_prop <- 100 * eigenvalues / sum(eigenvalues)
eigs_cum <- cumsum(eigs_prop)

eigen_table <- data.frame(
  eigenvalues = eigenvalues,
  proportion = eigs_prop,
  cumulative = eigs_cum
)

print(eigen_table, digits = 4)
```

```
##    eigenvalues proportion cumulative
## 1    9.9477504  82.897920      82.90
## 2    1.8476485  15.397071      98.29
## 3    0.1262558   1.052132      99.35
## 4    0.0382934   0.319112      99.67
## 5    0.0167094   0.139245      99.81
## 6    0.0128330   0.106942      99.91
## 7    0.0058303   0.048586      99.96
## 8    0.0020319   0.016932      99.98
## 9    0.0010235   0.008529      99.99
## 10   0.0009528   0.007940      99.99
## 11   0.0005368   0.004473     100.00
## 12   0.0001342   0.001118     100.00
```

b. Create a barchart (with axis labels) of the eigenvalues. What do you see? How do you read/interpret this chart? (10 pts)

Here's one possible barplot. Yours will likely be different than the one below.

```
# barchart of eigenvalues
barchart <- barplot(
  eigenvalues, las = 1, border = NA, names.arg = 1:length(eigenvalues),
  ylim = c(0, 1.1 * ceiling(max(eigenvalues))), ylab = "value",
  xlab = "Eigenvalues (associated to the PCs)", main = "Scree plot")
points(barchart, eigenvalues, pch = 19, col = "gray50")
lines(barchart, eigenvalues, lwd = 2, col = "gray50")
```



c. If you had to choose a number of dimensions (i.e. a number of PCs), how many would you choose and why? (10 pts)

To choose the number of dimensions (i.e. number of PCs) to be retained, you can use any of the four approaches discussed in class:

- Elbow in screeplot: first two PCs
- Kaiser's rule $\lambda_k > 1$: first two PCs
- Jollife's rule $\lambda_k > 0.7$: first two PCs
- A predetermined amount of captured variation: say you are interested in using as many PCs as possible in order to caputre 90% of variation; then use the first two PCs.

3) Studying the cloud of individuals (30 pts)

a. Create a scatter plot of the cities on the 1st and 2nd PCs (10 pts).
   - 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.

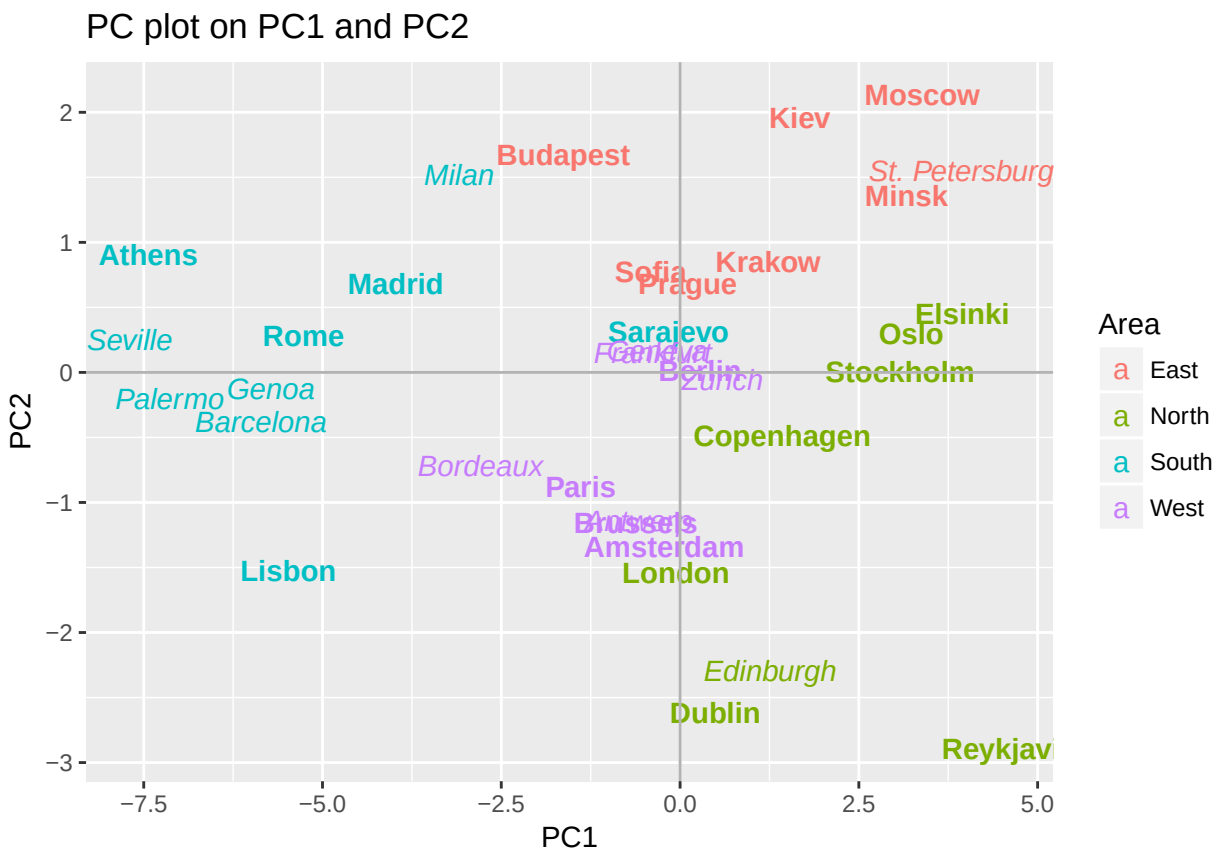
```
# centroid and std-devs of active individuals
centroid <- colMeans(Active)
stdevs <- apply(Active, 2, sd)

# data of supplementary individuals
Data_sup <- as.matrix(dat[supplementary_indivs, active_vars])
Xsup <- sweep(Data_sup, 2, centroid)
Xsup <- sweep(Xsup, 2, stdevs, FUN = "/")

# projection of supplementary individuals
PCsup <- Xsup %*% V

# binding PC coordinates of active and supplementary individuals
PC_all <- rbind(Z, PCsup)

# auxiliary vector that indicates active vs supplementary individuals
status <- rep(c('active', 'supplementary'),
              times = c(length(active_indivs), length(supplementary_indivs)))

# auxiliary factor to be used in ggplot() to distinguish
# between active (in bold) and supplementary individuals (in italics)
face <- factor(rep(c('bold', 'italic'),
              times = c(length(active_indivs), length(supplementary_indivs))))

# convenient data frame to be used for ggplot
pc_df <- data.frame(
  PC1 = PC_all[ ,1],
  PC2 = PC_all[ ,2],
  Area = dat$Area,
  Status = status,
  Face = face
)
```

To get the scatterplot of cities, you can follow a different approach than the one used in this rubric. Also, depending on the numeric results from `eigen()` (or `svd()`), your plot may show different coordinates. The important part is NOT the absolute position, but the relative position of the cities with respect to the origin.

I've chosen to display the names of active cities in bold type, and the names of the supplmentary cities in italics. Also, I've used colors to differentiate the geographical areas. Your graph will likely be different from mine. What matters is getting a visual display that makes sense.

```
ggplot(data = pc_df,
       aes(x = PC1, y = PC2, color = Area, label = rownames(pc_df))) +
  geom_text(aes(fontface = Face)) +
```

```
  geom_vline(xintercept = 0, color = "gray70") +
  geom_hline(yintercept = 0, color = "gray70") +
  ggtitle("PC plot on PC1 and PC2")
```



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

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

where:

- z_{ik} is the square value of the i-th individual on PC k
- $\mathbf{x_i}$ represents the row-vector of the i-th individual
- $\mathbf{g}$ is the centroid (i.e. average individual)

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? (10 pts).

```
# quality of representation
dist2centroid <- apply(X, 1, function(x) sum(x*x))
cos2 <- sweep(Z^2, 1, dist2centroid, FUN = "/")
```

```r
# displaying first 4 columns
cos2[ ,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
```

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

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

where:

- m_i is the mass or weight of individual i, in our case: $(\frac{1}{n-1})$
- z_{ik} is the value of k-th PC for individual i
- λ_k is the eigenvalue associated to k-th PC

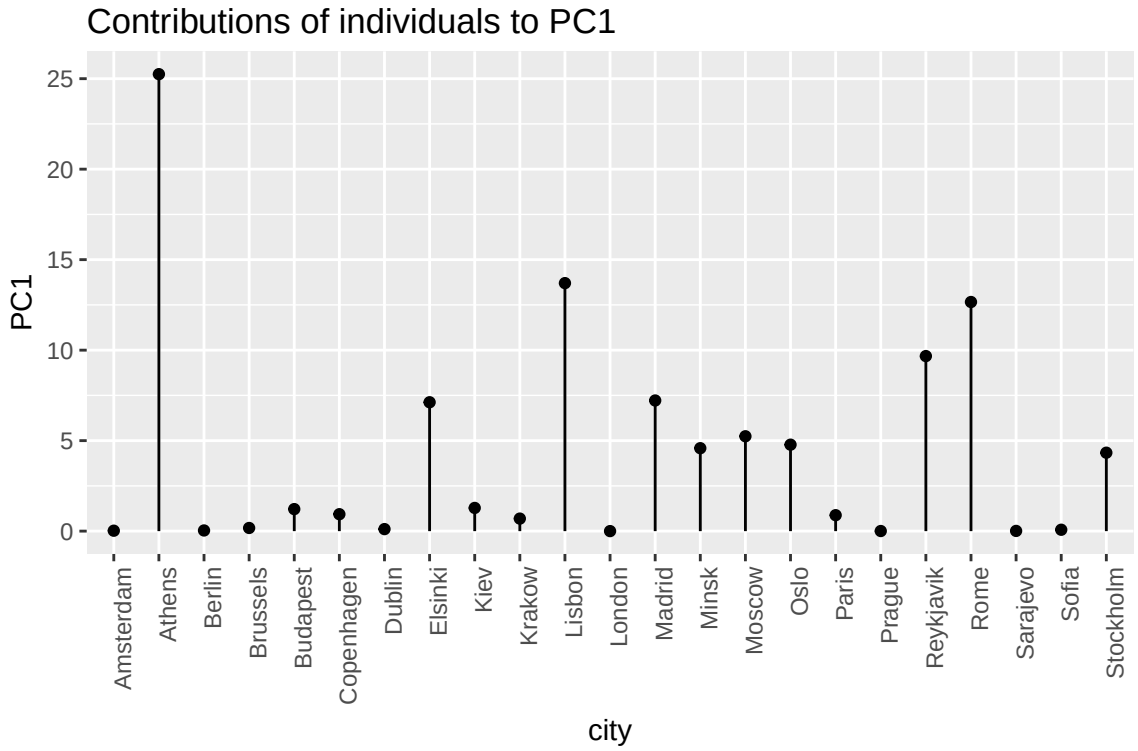
Store the individuals contributions in a matrix or data frmae, include row and column names. Display the first four columns. Are there any influential cities on the first two PCs? (10 pts).

```r
# data frame of contribution of individuals
CTR <- 100 * (1/(n-1)) * Z^2
CTR <- sweep(CTR, 2, eigenvalues, FUN = "/")
```

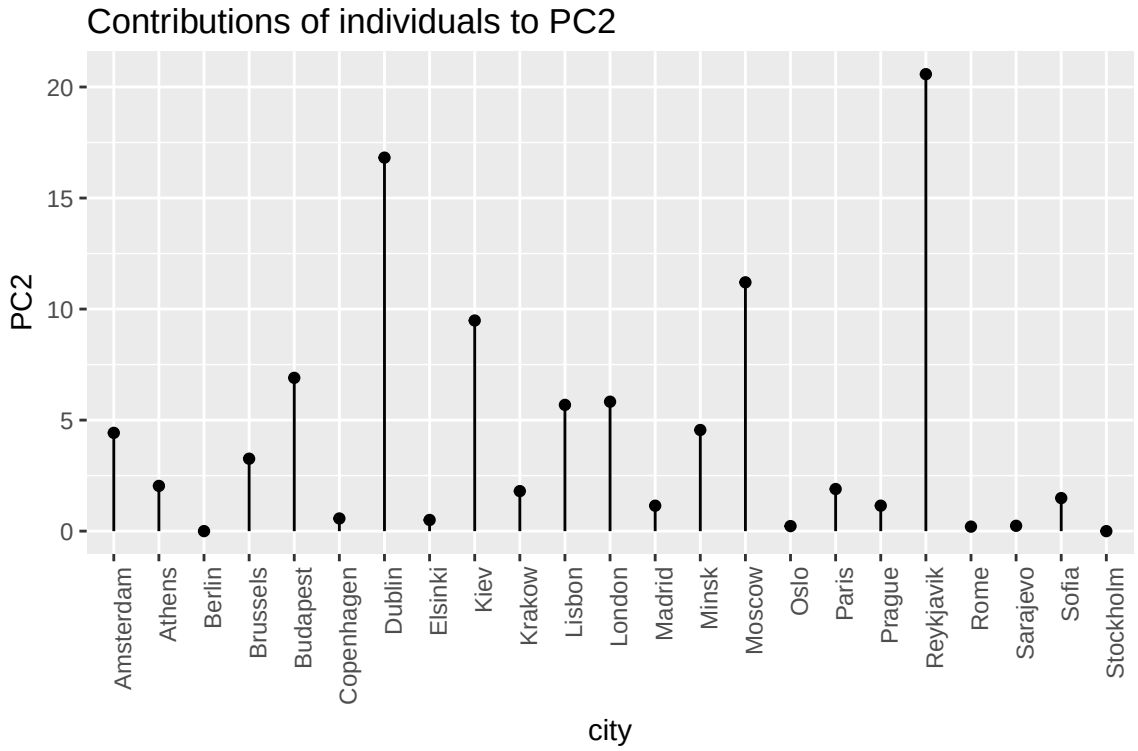
```r
# display four first columns
CTR[ ,1:4]
```

```
##                    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
# data frame of contributions (for ggplot)
ctr_df <- data.frame(CTR)
ctr_df$city <- 1:nrow(ctr_df)
ctr_df$zeros <- rep(0, nrow(ctr_df))

# plot of contributions of individuals to PC1
ggplot(data = ctr_df, aes(x = city, y = PC1)) +
  geom_point() +
  geom_segment(aes(x = city, xend = city, y = zeros, yend = PC1)) +
  scale_x_discrete(limit = rownames(ctr_df)) +
  theme(axis.text.x = element_text(angle = 90, hjust = 1)) +
  ggtitle("Contributions of individuals to PC1")
```




```
# plot of contributions of individuals to PC2
ggplot(data = ctr_df, aes(x = city, y = PC2)) +
  geom_point() +
  geom_segment(aes(x = city, xend = city, y = zeros, yend = PC2)) +
  scale_x_discrete(limit = rownames(ctr_df)) +
  theme(axis.text.x = element_text(angle = 90, hjust = 1)) +
  ggtitle("Contributions of individuals to PC2")
```



4) Studying the cloud of variables (30 pts)

a. Calculate the correlation of all quantitative variables (active and supplementary) with the principal components. (10 pts)
   - How are the supplementary variables related to the components?

```
# active and supplementary variables
supplementary_quant <- c('Annual', 'Amplitude', 'Latitude', 'Longitude')
corrs <- data.frame(cor(dat[active_indivs,c(active_vars, supplementary_quant)], Z))

corrs$type <- c(
  rep('active', length(active_vars)),
  rep('supp', length(supplementary_quant)))

# display firs four columns
corrs[ ,1:4]
```

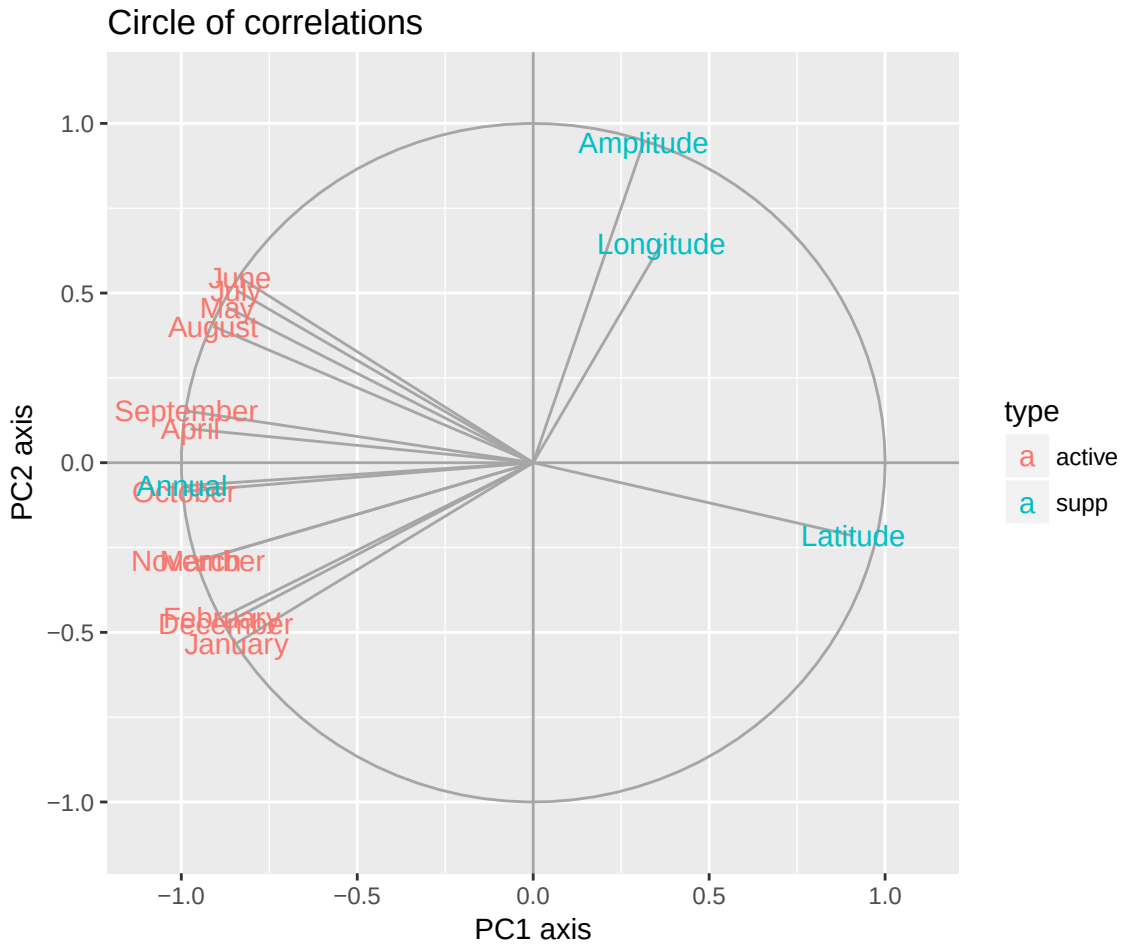
```
##                   PC1         PC2           PC3           PC4
## January   -0.8424506 -0.53135762  6.776712e-02 -1.168876e-02
## February  -0.8842848 -0.45583250 -3.466272e-03 -8.371472e-02
## March     -0.9450521 -0.28731281 -1.207952e-01 -7.781832e-02
## April     -0.9738876  0.09956500 -1.982562e-01 -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 plot of the correlations between the PCs and all the quantitative variables (10 pts).
   - 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.
   - Comment on your graph of variables.

```r
# function to create a circle
circle <- function(center = c(0, 0), npoints = 100) {
  r = 1
  tt = seq(0, 2 * pi, length = npoints)
  xx = center[1] + r * cos(tt)
  yy = center[1] + r * sin(tt)
  return(data.frame(x = xx, y = yy))
}
corcir <- circle(c(0, 0), npoints = 100)


# data frame with arrows coordinates
arrows_df <- data.frame(
  x1 = rep(0, nrow(corrs)),
  y1 = rep(0, nrow(corrs)),
  x2 = corrs$PC1,
  y2 = corrs$PC2,
  type = corrs$type
)


# circle of correlations
ggplot() +
  geom_path(data = corcir, aes(x = x, y = y), color = "gray65") +
  geom_segment(
```

```
  data = arrows_df, aes(x = x1, y = y1, xend = x2, yend = y2),
  color = "gray65") +
geom_text(data = corrs,
          aes(x = PC1, y = PC2, color = type,
              label = rownames(corrs))) +
geom_hline(yintercept = 0, color = "gray65") +
geom_vline(xintercept = 0, colour = "gray65") +
xlim(-1.1, 1.1) +
ylim(-1.1, 1.1) +
labs(x = "PC1 axis", y = "PC2 axis") +
ggtitle("Circle of correlations")
```



c. Based on the above parts (a) and (b), how are the active and supplementary variables related to the components? (10 pts)

The first PC can be summarized by the term "average annual temperature". The months September, October and April are more closely linked than the others to the first component: they "represent" the best annual temperatures in Europe. 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 (10 pts)

Assessing the relationships between temperatures revealed positive correlations between monthly temperatures and, more precisely, two periods: the summer season (May to August) and the winter season (November to March). Temperatures are more closely linked within each period than from one period to the other. Temperatures can be summarized by the first two Principal Components: PC1 can be labeled "average annual temperature", and PC2 "thermal amplitude". From these two variables we can outline the city *typologies*. By bringing together those cities which are close on the map defined by the first two components, and respecting the geographical location, we can propose the following typology:

- Southern European cities are characterized by high temperatures throughout the year.
- Western European cities are characterized by average temperatures throughout the year.
- Northern Eurpoean cities are characterized by cold temperatures, especially during summer.
- Eastern European cities are characterized by cold temperatures, especially during winter.