Principal Component 3d

Manu

5/3/2020

Contents


```
ccaa %>% head(9)
```

```
##                 WomanPopulationPtge  Izda_Pct Dcha_Pct Otros_Pct
## Andalucía                  49.33785 55.177999 41.40927  1.355818
## Aragón                     45.85021 41.598175 54.71111  1.783967
## Asturias                   49.84190 49.701974 46.29538  1.860462
## Baleares                   49.46037 44.388761 48.68112  4.588448
## Canarias                   49.45169 39.926080 45.63513 12.472602
## Cantabria                  47.71637 38.197441 58.29959  1.602990
## CastillaLeón               45.69278 31.514937 64.86095  1.457867
## CastillaMancha             46.49989 42.159800 54.73302  1.301682
## Cataluña                   48.40921  9.709147 15.48950 72.753068
##                 AbstentionPtge SameComAutonPtge ForeignersPtge
## Andalucía             28.70203         87.46541       6.171985
## Aragón                25.03356         79.41704       7.397142
## Asturias              33.76299         86.72031       2.669716
## Baleares              33.57470         64.38828      16.215110
## Canarias              34.84340         76.89585      11.118610
## Cantabria             26.88024         81.73901       3.450859
## CastillaLeón          23.82293         86.50172       3.859272
## CastillaMancha        22.69899         73.34752       7.224714
## Cataluña              34.28672         78.89126       9.123566
##                 Unemploy25_40_Ptge Age_under19_Ptge
## Andalucía                 43.23133        18.304943
## Aragón                    35.52834        11.171264
## Asturias                  43.44077        11.563526
## Baleares                  39.98404        19.773448
## Canarias                  42.53759        17.961705
## Cantabria                 42.20102        14.450471
## CastillaLeón              31.67205         8.723355
## CastillaMancha            35.25856        12.399628
## Cataluña                  38.00286        18.674294
```

1. Understanding Linear Algebra

Principal components are extracted from the covariance matrix of the data. They try to explain the variability of the dataset.

In the covariance matrix, the diagonal is the variance of the column X_i, this is easily explained with the covariance and variance formula:

Recall the covariance formula:

$$cov(x,y) = \sum_{1}^{n}(x_i - \overline{x})(y_i - \overline{y})/n - 1$$

Now the variance:

$$\sigma^2 = \sum_{1}^{n}(x_i + \overline{x})^2/n - 1$$

The data is going to be simplified to a (4x3) matrix.

```
cov_matrix_math <- as.matrix(ccaa[1:4, 1:3])
cov_matrix_math

##           WomanPopulationPtge Izda_Pct Dcha_Pct
## Andalucía            49.33785 55.17800 41.40927
## Aragón               45.85021 41.59818 54.71111
## Asturias             49.84190 49.70197 46.29538
## Baleares             49.46037 44.38876 48.68112
```

Considering the formula, you want to substract each observation the mean of its vairable: $\boldsymbol{x_i} - \boldsymbol{\overline{x}_i}$

$$cov(x,y) = \sum_{1}^{n}(x_i - \overline{x})(y_i - \overline{y})$$

```
cov_matrix_math[,1] <- cov_matrix_math[,1] - mean(cov_matrix_math[,1])
cov_matrix_math[,2] <- cov_matrix_math[,2] - mean(cov_matrix_math[,2])
cov_matrix_math[,3] <- cov_matrix_math[,3] - mean(cov_matrix_math[,3])
cov_matrix_math

##           WomanPopulationPtge  Izda_Pct   Dcha_Pct
## Andalucía           0.7152665  7.461271 -6.3649520
## Aragón             -2.7723729 -6.118552  6.9368926
## Asturias            1.2193154  1.985247 -1.4788377
## Baleares            0.8377911 -3.327966  0.9068971

# If you want to try it in a bigger dataframe and not waste time writing down co
de:
  # A <- ccaa
  # for (i in 1:length(A)) A[,i] = A[,i]-mean(A[,i])
  # A
```

The next step is to multiply: $(\boldsymbol{x_i} - \overline{\boldsymbol{x}})(\boldsymbol{y_i} - \overline{\boldsymbol{y}})$**.** But you need a squared matrix (Same rows, same columns) to get the same variables in rows and columns. This can be achivied with the transpose of the matrix, that gives a (3x4) matrix, so you get: (3x4)*(4x3) = (3x3) matrix.

```r
t(cov_matrix_math)
```

	Andalucía	Aragón	Asturias	Baleares
WomanPopulationPtge	0.7152665	-2.772373	1.219315	0.8377911
Izda_Pct	7.4612714	-6.118552	1.985247	-3.3279661
Dcha_Pct	-6.3649520	6.936893	-1.478838	0.9068971

Transposed matrix

	WomanPopulationPtge	Izda_Pct	Dcha_Pct
Andalucía	0.7152665	7.461271	-6.3649520
Aragón	-2.7723729	-6.118552	6.9368926
Asturias	1.2193154	1.985247	-1.4788377
Baleares	0.8377911	-3.327966	0.9068971

Matrix – mean

Now just multiply $\boldsymbol{A^t} * \boldsymbol{A}$**.** The multiplication of 2 matrix is: **row*column**, which is shown below manually. Notice that is the first number of the row of the first matrix times the first second matrix's fist column number plus the number of the first matrix's row times plus the second number of the second matrix, so at the end you are computing a $\sum_i^n$, part of the variance and covariance formula!

To compute A[1,1] = (0.7152665 * 0.7152665) + (-2.772373 * -2.7723732) + (1.2193154 * 1.2193154) + (0.8377911*0.8377911) = 10.39

- This is equal to $\sum_1^{\mathbf{n}}(\mathbf{x_i} - \overline{\mathbf{x}})^2$, so in the diagonal you are computing ***the variance!***

To compute A[2,1] = (7.4612714 * 0.7152665) + (-6.118552 * -2.7723732) + (1.985247 * 1.2193154) + (-3.3279661 * 0.8377911) = 21.93

- Also equals to $\sum_1^{\mathbf{n}}(\mathbf{x_i} - \overline{\mathbf{x}})(\mathbf{y_i} - \overline{\mathbf{y}})$, ***the covariance!***

```r
(0.7152665 * 0.7152665) + (-2.772373 * -2.7723732) + (1.2193154 * 1.2193154) + (
0.8377911*0.8377911)

## [1] 10.38628

(7.4612714 * 0.7152665) + (-6.118552 * -2.7723732) + (1.985247 * 1.2193154) + (-
3.3279661 * 0.8377911)

## [1] 21.93221
```

Let's check and divide by n, (numbers of A rows-1), so that you can finished the formula: $cov(x, y) = \sum_1^n(x_i - \overline{x})(y_i - \overline{y})/n - 1$

```r
t(cov_matrix_math)%*%cov_matrix_math

##                     WomanPopulationPtge  Izda_Pct  Dcha_Pct
## WomanPopulationPtge            10.38628  21.93221 -24.82767
## Izda_Pct                       21.93221 108.12382 -95.88835
## Dcha_Pct                      -24.82767 -95.88835  91.64252

t(cov_matrix_math)%*%cov_matrix_math/(nrow(cov_matrix_math)-1) # covariance + va
riance
```

```
##                        WomanPopulationPtge   Izda_Pct  Dcha_Pct
## WomanPopulationPtge              3.462094   7.310736  -8.27589
## Izda_Pct                         7.310736  36.041272 -31.96278
## Dcha_Pct                        -8.275890 -31.962785  30.54751
```

Now you know how to calculate the covariance using linear algebra and the reason why the variance of x_i is in the diagonal.

```
cov(cov_matrix_math)

##                        WomanPopulationPtge   Izda_Pct  Dcha_Pct
## WomanPopulationPtge              3.462094   7.310736  -8.27589
## Izda_Pct                         7.310736  36.041272 -31.96278
## Dcha_Pct                        -8.275890 -31.962785  30.54751

var(cov_matrix_math[,1]) # variance of first column

## [1] 3.462094
```

2. Eigenvalues & Eigenvectors.

So why all of this theory?

1. A PC is a "new variable" which is made with a set of correlated independent variables. So they are just lineal combinations of those variables!

   - $\boldsymbol{CP_1 = v_{11} * x_1 + v_{12} * x_2 + \ldots + a_{1m} * x_m}$; m=nº of variables; v=eigenvector position.
2. What are thevij? The ij value of an **eigenvector**, for instance, first eigenvalue, first position. The eigenvectors are the Principal Component that describe a portion of the variability of the dataset. The firsrt PC1 tries to explain the maximum variability, the PC2, ties to explain the **remaining** variability that the PC1 couldn't explain, and it is not correlated to the PC1. So the vectors are *orthogonal* (each one pointing in distinct directions).

3. Eigenvalue: Each eigenvector is associated to an eigenvalue. The eigenvalues (λ_i) display the portion of variation retained by the PC that are associated to.

4. The covariance matrix is essenctial to compute all of this. R can compute eigenvalues and eigenvectos, just keep in mind that it finds the values and vectors that follow this rule:

$$\boldsymbol{A * v_{(matricialproduct)} = v * \lambda_{(escalarproduct)}};$$

A = CovMatrix, v=eigenvector, lambda = eigenvalue.

For example:

The Eigen function computes the v, λ for a given matrix.

```
eigen(cov(cov_matrix_math))
```

```
## eigen() decomposition
## $values
## [1] 67.2658402  2.5439566  0.2410742
##
## $vectors
##            [,1]       [,2]      [,3]
## [1,] -0.1696792 -0.7488229 0.6406819
## [2,] -0.7240665  0.5357413 0.4344065
## [3,]  0.6685332  0.3901866 0.6331017
```

It can also be shown that: $A * v_{(matricialproduct)} = v * \lambda_{(escalarproduct)}$

```
c(-0.1696792 ,-0.7240665 , 0.6685332)*67.265 # PC1 eigenvector

## [1] -11.41347 -48.70433  44.96889

cov(cov_matrix_math)%*%c(-0.1696792 ,-0.7240665 , 0.6685332)

##                          [,1]
## WomanPopulationPtge -11.41361
## Izda_Pct            -48.70494
## Dcha_Pct             44.96945
```

Now compute PCA. You can tell that PC1 is explained by the relations between Izda_Pct (left_wing) and Dcha_Pct (right_wing). You can see it as “the correlation” between the variables and the PC. So that, WomanPoplation would be in the PC2.

```
pc <- prcomp(cov_matrix_math)
pc$rotation # eigenvalues (Principal Component)

##                            PC1        PC2        PC3
## WomanPopulationPtge -0.1696792  0.7488229 -0.6406819
## Izda_Pct            -0.7240665 -0.5357413 -0.4344065
## Dcha_Pct             0.6685332 -0.3901866 -0.6331017

pc$sdev**2 # eigen vectors (the function shows sd(lambda))

## [1] 67.2658402  2.5439566  0.2410742
```

So now as you would do in a linear regression, apply the first PC1, to the first observation, in this case “Andalucía”.

$$\boldsymbol{CP_1 = v_{11} * x_1 + v_{12} * x_2 + \ldots + a_{1m} * x_m}$$

$C_1 = (-0.1696792 * 0.7152665) + (-0.7488229 * 7.461271) + (0.6406819 * -6.3649520)$

```
cov_matrix_math

##            WomanPopulationPtge  Izda_Pct   Dcha_Pct
## Andalucía            0.7152665  7.461271 -6.3649520
## Aragón              -2.7723729 -6.118552  6.9368926
## Asturias             1.2193154  1.985247 -1.4788377
## Baleares             0.8377911 -3.327966  0.9068971
```

Now just apply it to every observation with $x.

```
-0.1696792 *0.7152665+(-0.7488229*7.461271)+(0.6406819*-6.3649520)

## [1] -9.786446

pc$x

##                 PC1        PC2         PC3
## Andalucía -9.779004 -0.9781846  0.33017913
## Aragón     9.538196 -1.5047374  0.04238926
## Asturias  -2.632995  0.4264950 -0.70734274
## Baleares   2.873804  2.0564270  0.33477435
```

With summary you can see how the PC1 explains the 96% of the data set. You started with 3 variables and now you only need one.

```
pc %>% summary()

## Importance of components:
##                           PC1     PC2     PC3
## Standard deviation     8.2016 1.59498 0.49099
## Proportion of Variance 0.9602 0.03632 0.00344
## Cumulative Proportion  0.9602 0.99656 1.00000

8.2^2/(8.2^2 + 1.59^2 + 0.49^2) # Eigen values.

## [1] 0.9604589
```

3. 3d PCA + Plotly

Now the represention with the whole dataframe.

```
ccaa_location <- data.frame(location = factor(c("South", "North", "North", "Extr
aPeninsular", "ExtraPeninsular", "North", "Center", "Center", "North", "ExtraPen
insular", "Center", "South", "South", "Center", "ExtraPeninsular", "South", "Nor
th", "North", "North" )))

ccaa2 <- data.frame(ccaa)

# PCA
pc <- prcomp(ccaa,retx = T,scale. = T)

# Eigenvectors applied to observations.
res <- pc$x*(-1) # changing the direction.
x <- res[,1]
y <- res[,2]
z <- res[,3]

# Loadings/Eigenvectors
ev <- pc$rotation*-1 # Changing the direction.

# 3D plot
library(plotly)
```

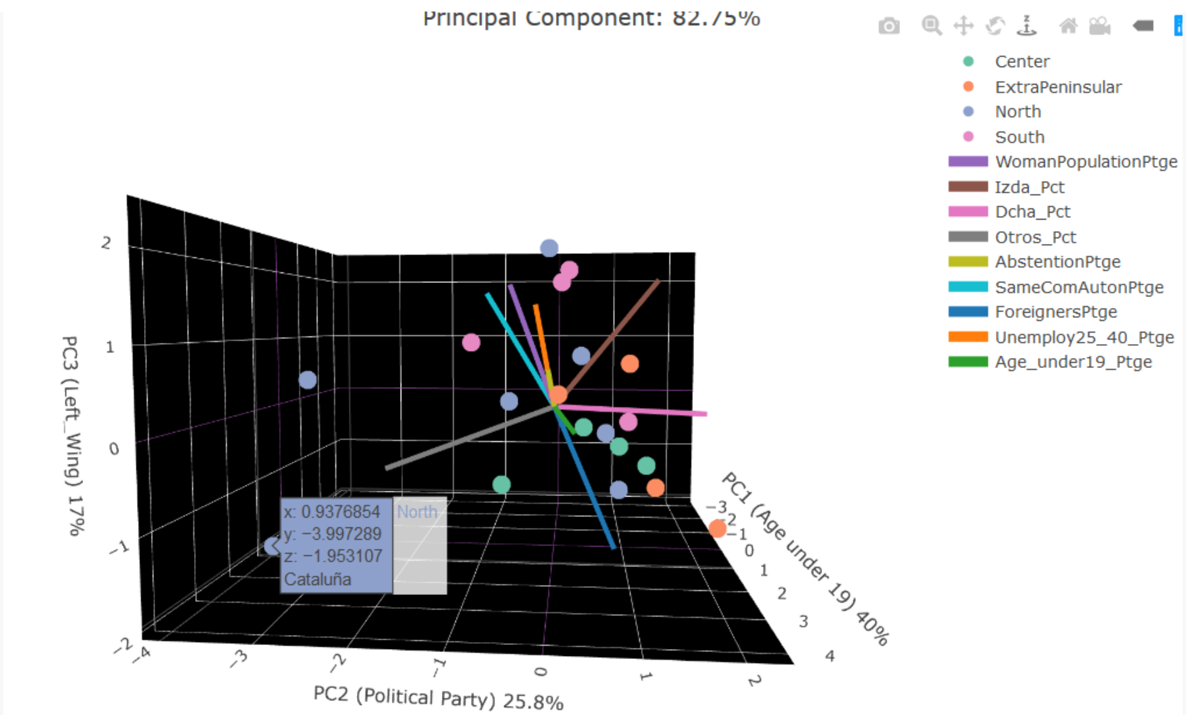
```
ply <- plot_ly() %>%
  add_trace(x=x, y=y, z=z,
            type="scatter3d",
            mode="markers",
            color=ccaa_location$location,
            text = rownames(ccaa)
            )

for (i in 1:nrow(ev)) {
   x <- c(0, ev[i,1])*4 # Creating a vector the origin is 0, and direction vij.
   y <- c(0, ev[i,2])*4 # Multiplied * 4 because of the standarizarion that us P
rComp function.
   z <- c(0, ev[i,3])*4
   ply <- ply %>% add_trace(x=x, y=y, z=z,
            type="scatter3d", mode="lines",
            line = list(width=8),
            opacity = 1, name = names(ccaa)[i])
}

ply <- ply%>%
  layout(
    title = "Principal Component: 82.75%",
    scene = list(
      xaxis = list(title = "PC1 (Age under 19) 40%",
                   backgroundcolor="rgb(0, 0,0)",
                   gridcolor="rgb(255,255,255)",
                    showbackground=TRUE,
                    zerolinecolor="rgb(152, 78, 165)"
      ),

      yaxis = list(title = "PC2 (Political Party) 25.8%",
                   backgroundcolor="rgb(0, 0,0)",
                    gridcolor="rgb(255,255,255)",
                    showbackground=TRUE,
                    zerolinecolor="rgb(152, 78, 165)"
      ),
      zaxis = list(title = "PC3 (Left_Wing) 17%",
                  backgroundcolor="rgb(0, 0,0)",
                  gridcolor="rgb(255,255,255)",
                  showbackground=TRUE,
                  zerolinecolor="rgb(152, 78, 165)"

      )
    ))
ply
```




```
pc %>% summary()
```

```
## Importance of components:
##                          PC1    PC2    PC3     PC4     PC5     PC6     PC7
## Standard deviation     1.897 1.5244 1.2344 0.82493 0.70216 0.49126 0.29438
## Proportion of Variance 0.400 0.2582 0.1693 0.07561 0.05478 0.02682 0.00963
## Cumulative Proportion  0.400 0.6582 0.8275 0.90315 0.95793 0.98474 0.99437
##                            PC8      PC9
## Standard deviation     0.22496 0.005696
## Proportion of Variance 0.00562 0.000000
## Cumulative Proportion  1.00000 1.000000
```

```
pc$rotation[,1:3]
```

```
##                             PC1         PC2         PC3
## WomanPopulationPtge -0.34196353  0.13944709 -0.43313085
## Izda_Pct             0.08772868 -0.38236488 -0.46771386
## Dcha_Pct             0.07151747 -0.56472128  0.02993625
## Otros_Pct           -0.09914024  0.61595065  0.22519556
## AbstentionPtge      -0.43112676  0.01008879 -0.15906279
## SameComAutonPtge     0.35314346  0.28874973 -0.42949733
## ForeignersPtge      -0.32748459 -0.20987545  0.44378276
## Unemploy25_40_Ptge  -0.43547498  0.05120093 -0.36613135
## Age_under19_Ptge    -0.50294719 -0.07680397  0.03560840
```

PC1 is explained between the relations of Age_under_19_Ptge + Unemploy25_40_Ptge.