All Countries Environmental Data

Method: PCA

Principal component analysis (PCA), part of descriptive analytics, is used to analyze one table of quantitative data, specifically useful for *high dimensional data* and comparitively lesser data rows. PCA mixes the input variables to give new variables, called principal components. The first principal component is the line of best fit. It is the line that maximizes the inertia (similar to variance) of the cloud of data points. Subsequent components are defined as orthogonal to previous components, and maximize the remaining inertia.

PCA gives one map for the rows (called factor scores), and one map for the columns (called loadings). These 2 maps are related, because they both are described by the same components. However, these 2 maps project different kinds of information onto the components, and so they are *interpreted differently*. Factor scores are the coordinates of the row observations and Loadings describe the column variables. Both can be interpreted through their distance from origin. However, Factor scores are also interpreted by the distances between them and Loadings interpreted by the angle between them.

The distance from the origin is important in both maps, because squared distance from the mean is inertia (variance, information; see sum of squares as in ANOVA/regression). Because of the Pythagorean Theorem, the total information contributed by a data point (its squared distance to the origin) is also equal to the sum of its squared factor scores.

With both Factor and Loadings maps combined we can interpret which grouping criteria of rows of data is most impacted by which columns. This can interpreted visually by observing which a factors and loadings on a particular component and the distance on this component.

PCA also helps in *dimensionality reduction*. Using SVD, we get eigen values arranged in descending order in the diagonal matrix. We can simply ignore the lower eigen values to reduce dimensions. We can also take help of SCREE plot to visually analyze importance of eigen values.

Dataset

- Data: Measurements of environment conditions in Countries
- Rows: There are 137 observations, 1 for each country.
- Columns: Total 29 variables
- Qualitative: Country (nominal), Happiness (Ordinal).
- Quantitative: Aspect, Slope Crop Land, Tree Canopy Wind Cloud & Multiple variables for Temp & Rain
- Structure of Data

```
str(country_env_df)
```

```
## 'data.frame':    137 obs. of  29 variables:
##  $ Country                : Factor w/ 137 levels "Afghanistan",..: 1 2 3 4 5 6 7 8 9 10 ...
##  $ Happiness_Rank         : Ord.factor w/ 3 levels "VH"<"H"<"U": 3 2 2 3 1 3 1 1 2 2 ...
##  $ accessibility_to_cities: num  317.7 73.8 1212.8 378.2 209.2 ...
##  $ elevation              : num  1832 652 557 1061 683 ...
##  $ aspect                 : num  201 192 185 174 145 ...
##  $ slope                  : num  1.516 1.89 0.171 0.193 0.624 ...
##  $ cropland_cover         : num  9.51 23.35 3.69 2.79 21.96 ...
##  $ tree_canopy_cover      : num  0.375 12.805 0.177 19.87 8.834 ...
```

```
##  $ isothermality          : num  35.9 33.2 40.3 64.3 49.9 ...
##  $ rain_coldestQuart      : num  128.72 392.51 25.29 8.05 79.09 ...
##  $ rain_driestMonth       : num  1.722 40.088 0.935 0.26 17.183 ...
##  $ rain_driestQuart       : num  8.3 138.15 6.09 4.43 60.49 ...
##  $ rain_mean_annual       : num  311.3 1151.1 79.5 1023.4 539.9 ...
##  $ rain_seasonailty       : num  91.6 38.5 67.1 91.5 48.3 ...
##  $ rain_warmestQuart      : num  12.69 138.33 9.51 318.54 183.14 ...
##  $ rain_wettestMonth      : num  67.8 159 13.4 202.2 79.2 ...
##  $ rain_wettestQuart      : num  175.8 435.9 33.3 524.3 211.7 ...
##  $ temp_annual_range      : num  40.3 27.1 36.5 21.5 26.8 ...
##  $ temp_coldestQuart      : num  -0.261 3.58 13.152 18.794 8.024 ...
##  $ temp_diurnal_range     : num  14.72 9.11 14.87 13.85 13.46 ...
##  $ temp_driestQuart       : num  21.1 19.6 26.9 18.9 11.1 ...
##  $ temp_max_warmestMonth  : num  32 26.3 41.5 31 28.2 ...
##  $ temp_mean_annual       : num  11.5 11.5 23 21.6 14.2 ...
##  $ temp_min_coldestMonth  : num  -8.312 -0.806 5.058 9.549 1.443 ...
##  $ temp_seasonality       : num  88.2 62.7 75.1 18.5 47.6 ...
##  $ temp_warmestQuart      : num  22.7 19.6 32.5 23.3 20.2 ...
##  $ temp_wettestQuart      : num  3.95 5.27 20.81 22.76 16.48 ...
##  $ wind                   : num  3.43 2.47 4.03 2.16 4.27 ...
##  $ cloudiness             : num  114.2 181.1 90.7 187.5 159 ...
```

- Research Question

How do the 137 countries differ on these variables?

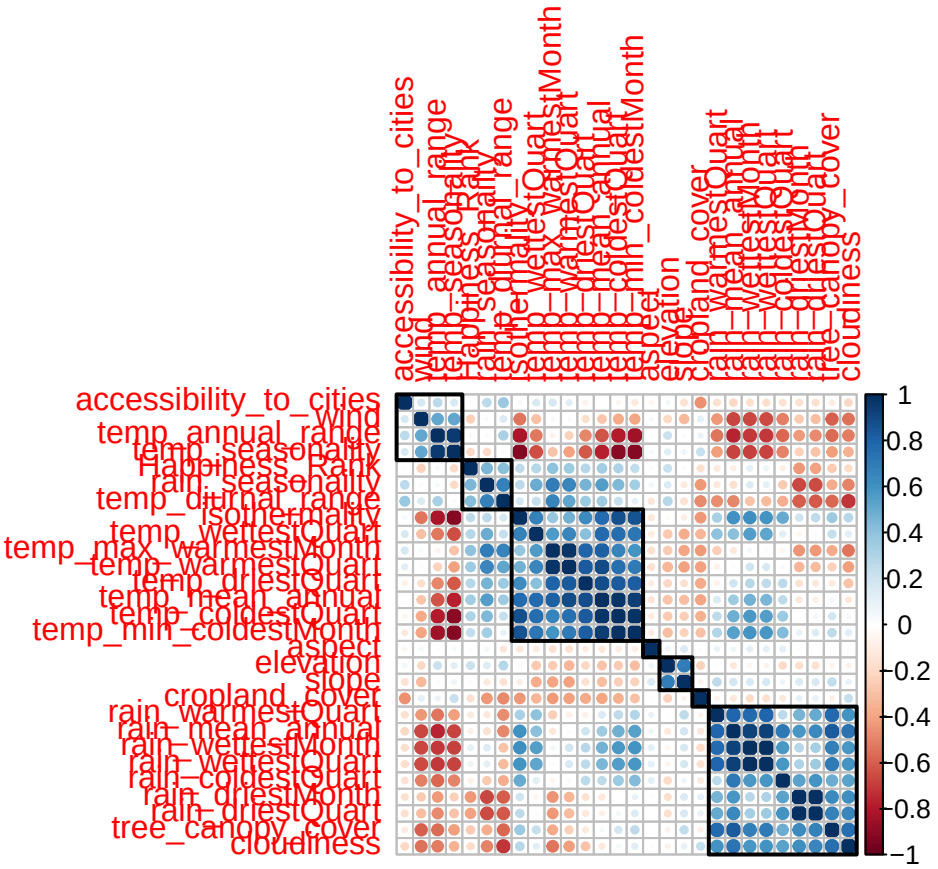
Analysis

There are multiple variables representing rain and Temp. Hence, for analysis purposes, lets choose annual mean for Rain and Temp.

Correlation Plot

Visually analyze multicollinearity in the system. Analyze the

```
corr_result = cor(country_env_df_for_corr)
corrplot(corr_result,order = 'hclust', addrect = 7)
```



Identify Latent Components

PCA (with Inference)

```
country_env_pca <- epPCA(DATA = country_env_df_for_pca, center = TRUE, scale = 'SS1', DESIGN = country_
country_env_pca_inf <- InPosition::epPCA.inference.battery(DATA = country_env_df_for_pca, center = TRUE
```

```
## [1] "It is estimated that your iterations will take 0.02 minutes."
## [1] "R is not in interactive() mode. Resample-based tests will be conducted. Please take note of the
## ================================================================================
```

Now we have Factor scores and Loadings. * Factor Scores are the new Data points w.r.t. new Components achieved with help of SVD. * Loadings represent correlation between variables w.r.t the choosen Components. Can be interpreted in 3 ways + As slices of inertia of the contribution data table w.r.t. the choosen Components + As correlation between columns (features) of Original Data and Factor scores of each Components (latent features). + As coefficients of optimal linear combination i.e. Right Sigular Vectors (Q matrix of SVD) #### Scree Plot

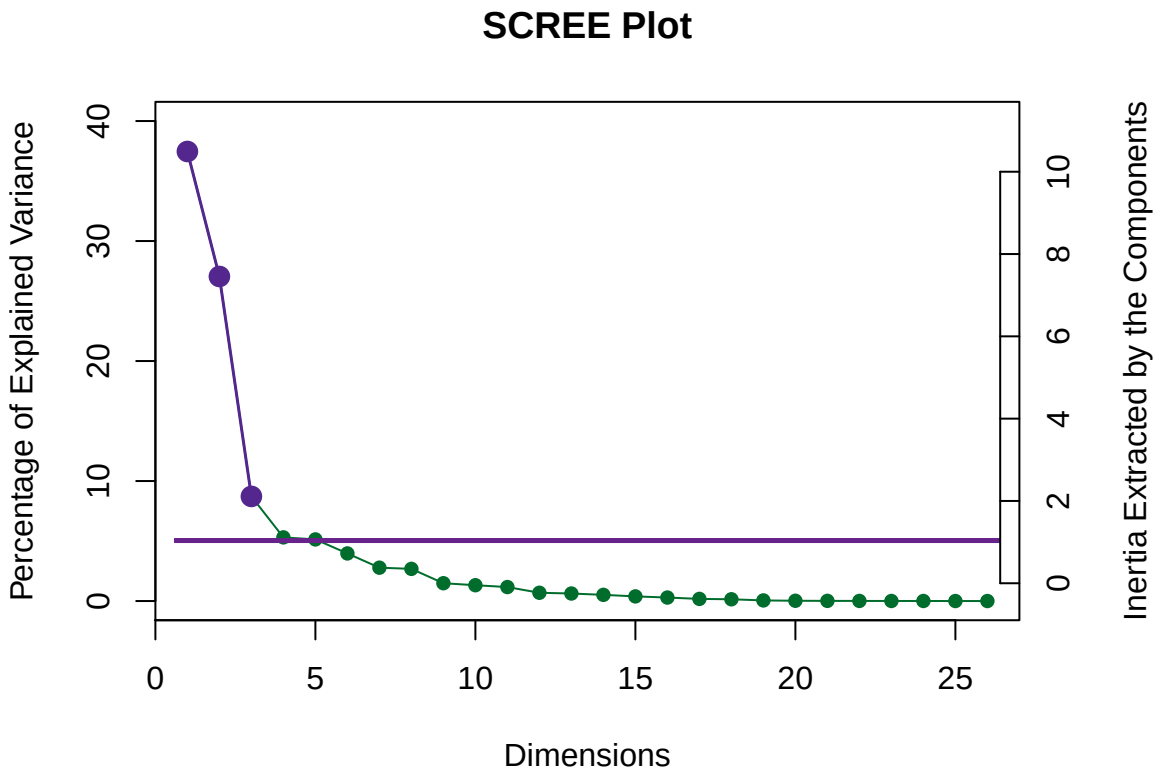
Scree Plot

Gives amount of information explained by corresponding component. Gives an intuition to decide which components best represent data in order to answer the research question.

P.S. The most contribution component may not always be most useful for a given research question.

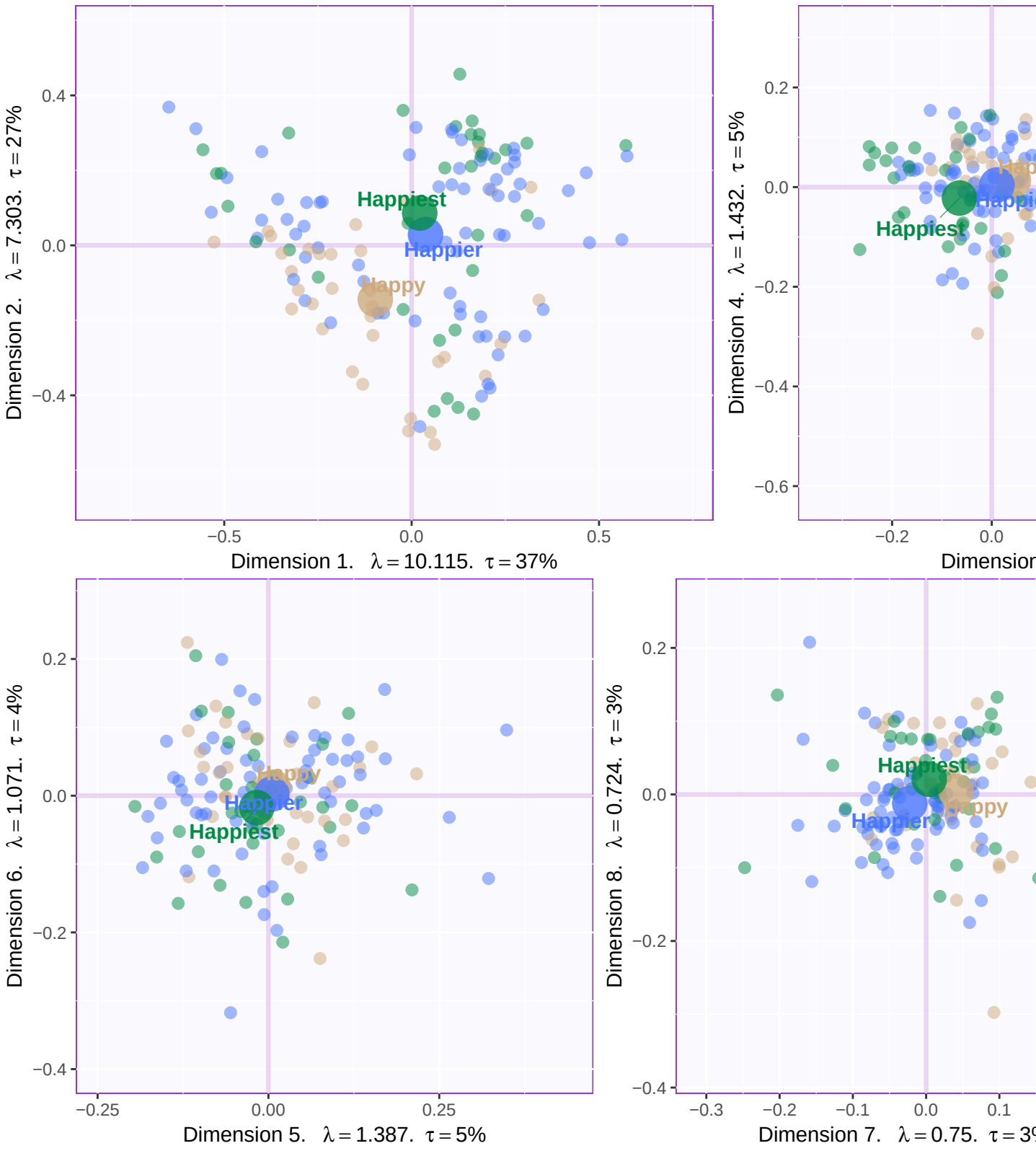
```
PTCA4CATA::PlotScree(ev = country_env_pca$ExPosition.Data$eigs,
                     p.ev =  country_env_pca_inf$Inference.Data$components$p.vals,
```

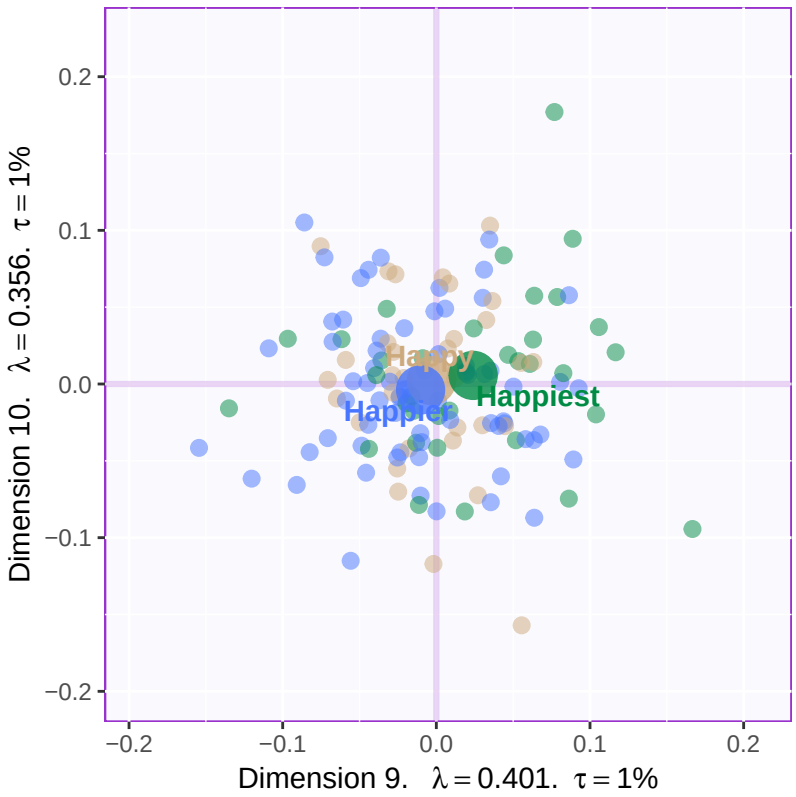
```
                        title = 'SCREE Plot',
                        plotKaiser = TRUE
)
```



Factor Scores

Lets visualize happiness categories for components 1-10, to make a decision (visually) on the most important components.


Happiest
Happier
Happy
Dimension 1. λ = 10.115. τ = 37%
Dimension 2. λ = 7.303. τ = 27%
Dimension 4. λ = 1.432. τ = 5%
Dimension 5. λ = 1.387. τ = 5%
Dimension 6. λ = 1.071. τ = 4%
Dimension 7. λ = 0.75. τ = 3%
Dimension 8. λ = 0.724. τ = 3%



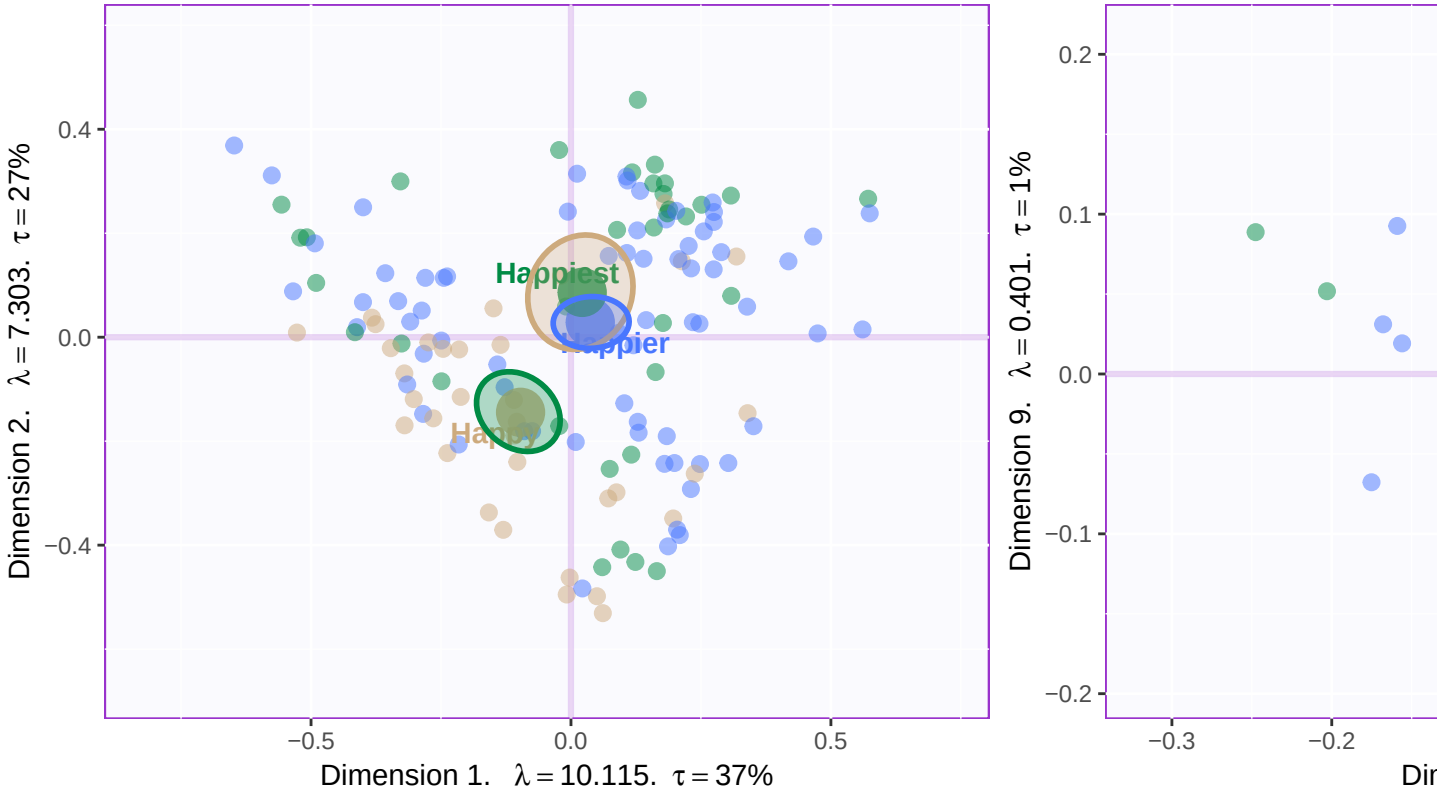
Since, it's not very straightforward to decide which components may be best suited for the research question at hand, let's represent, in a tabular format, which component helps to differentiate between which design variable values (Happy, Happier, Happiest)

P.S. here -1 represents -ve quadrant of the component and +1 represent +ve quadrant. 0 represents that component was not decisive enough to clearly seperate happiness levels.

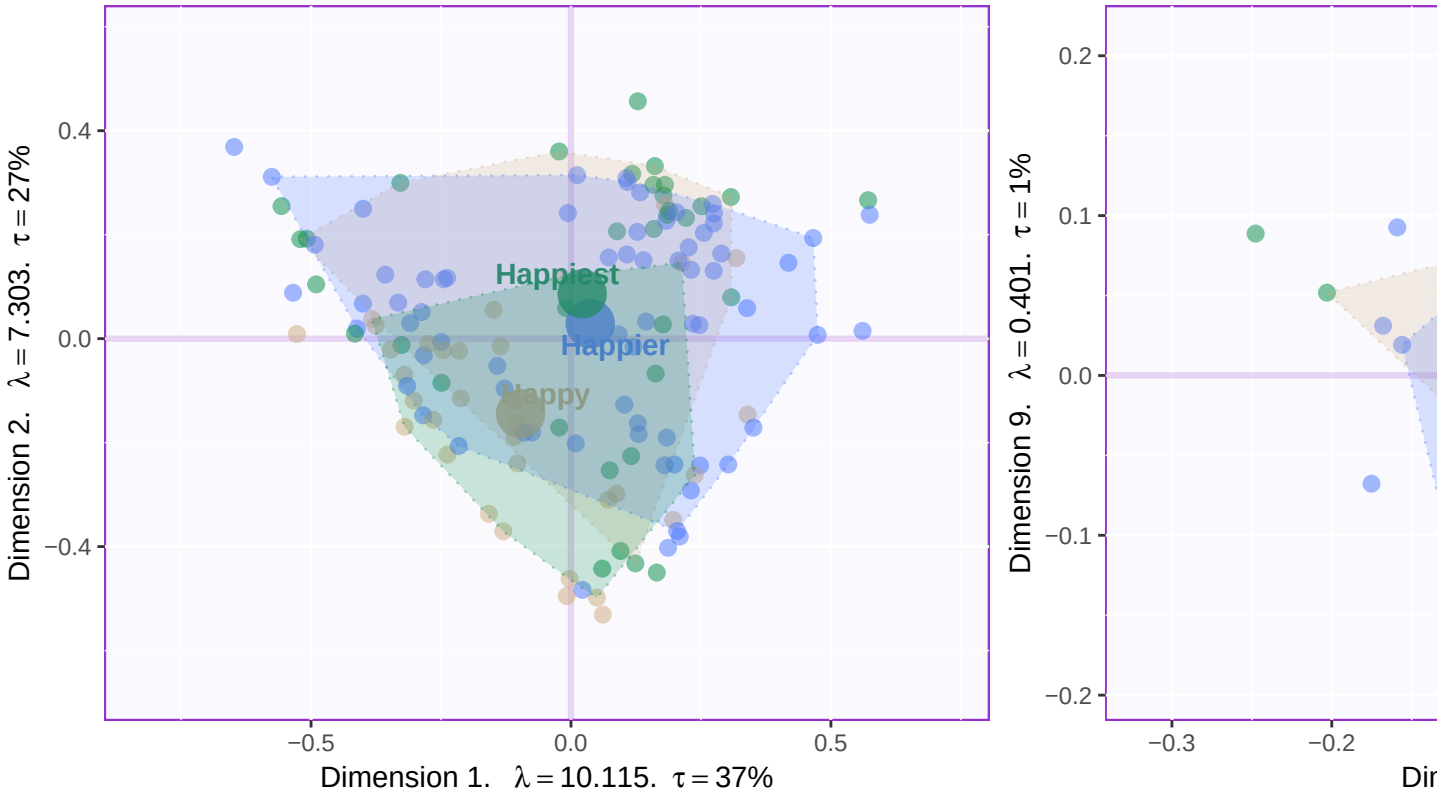
```
##              happy happier happiest
## Component 1     -1       1        0
## Component 2     -1       0        1
## Component 3      1       0       -1
## Component 4      0       0        0
## Component 5      0       0        0
## Component 6      0       0        0
## Component 7      1      -1        0
## Component 8      0       0        0
## Component 9      0      -1        1
## Component 10     0       0        0
```

Looking at the table, it seems component 1, 2, 7, 9 may be able to best represent all 3 happiness levels. Although, SCREE Plot suggests that 3^{rd} and 4^{th} components might be useful, from our above analysis we know otherwise. Also, SCREE plot suggests that component 6^{th} and onwards might not be useful which is contradicting our findings above. Hence, let's plot components 1 vs 2 and 7 vs 9. Similarily, we will also plot Loading plots for these componenets.

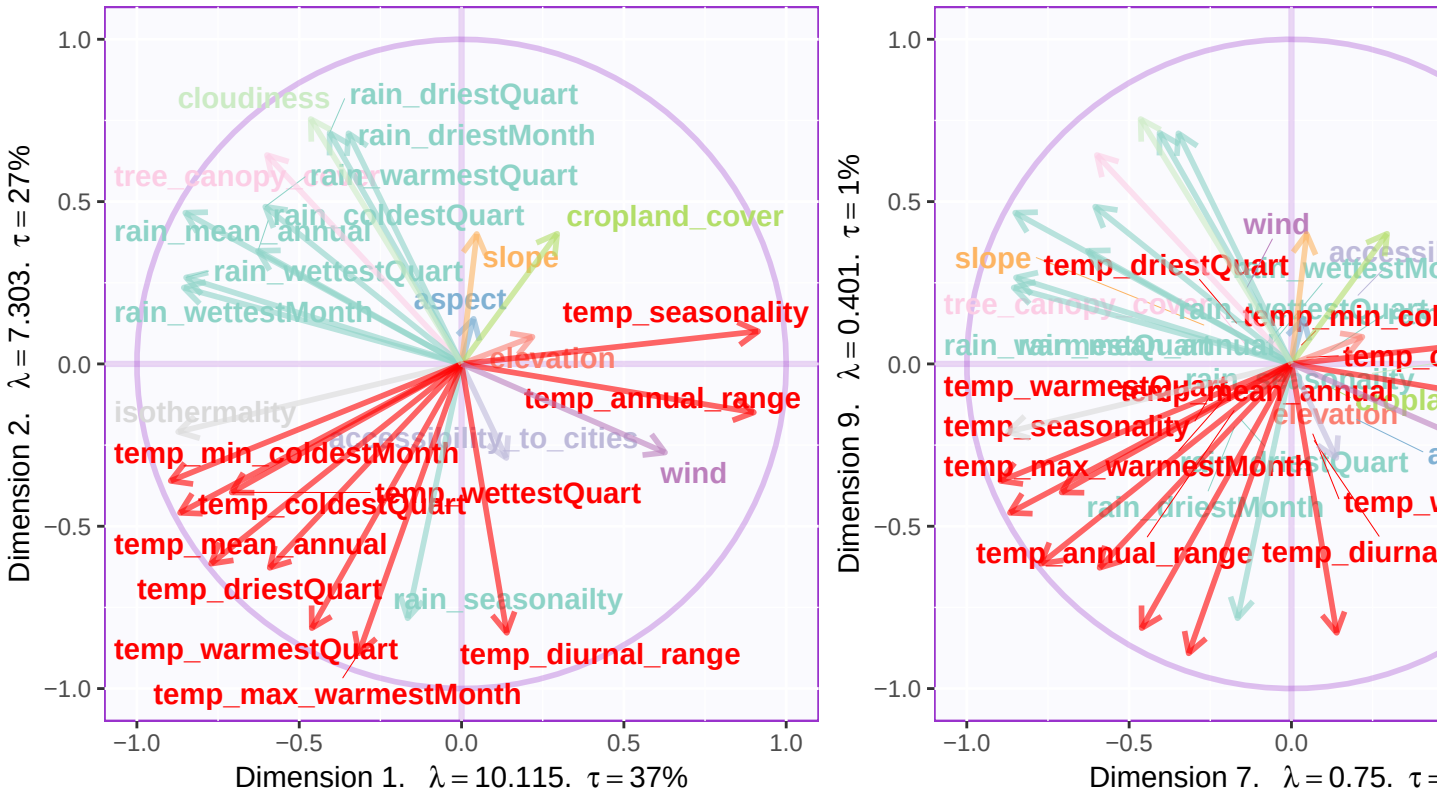
- With Confidence Interval



- With Tolerance Interval



Loadings



- Component 1:
  - Rows: Normal & Happy
  - Columns: Cloudiness & Rain vs Cropland, Aspect, Elevation
  - Interpret: People in countries with more Cloudiness, Trees and Rain tends to be happier.
- Component 7:
  - Rows: Happy & Unhappy
  - Columns: Temp and Rain vs Accessibility and Cropland
  - Interpret: Rain and Temp seems to be main reason for unhappiness and Cropland is important for Happiness.
- Component 9:
  - Rows: Happy & Very Happy
  - Columns: Temp vs Rain
  - Interpret: Rain and Temp seems to be main reason for Happiness. *This contradicts with Component 7 and 1.*

Most Contributing Variables

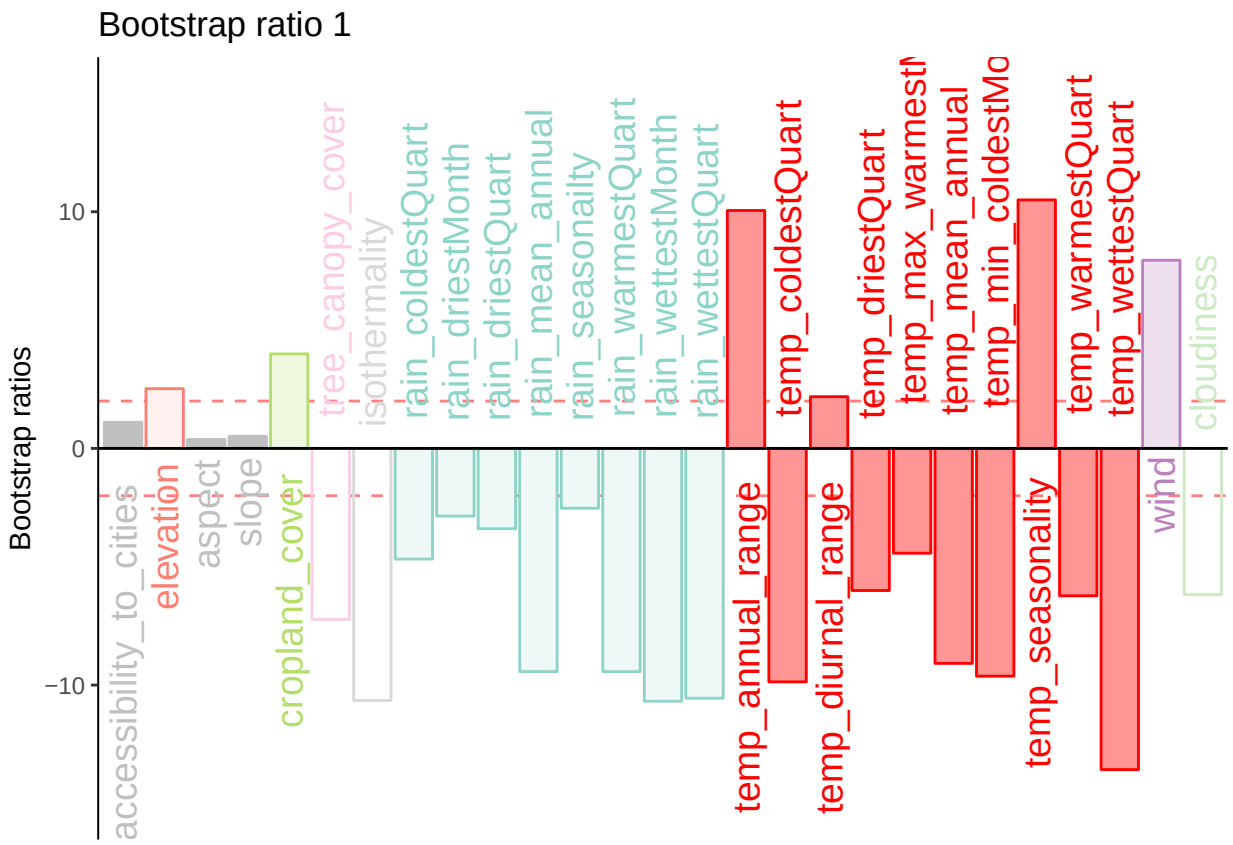
Let's plot variable contributions against each chosen components i.e. 1, 7, 9.

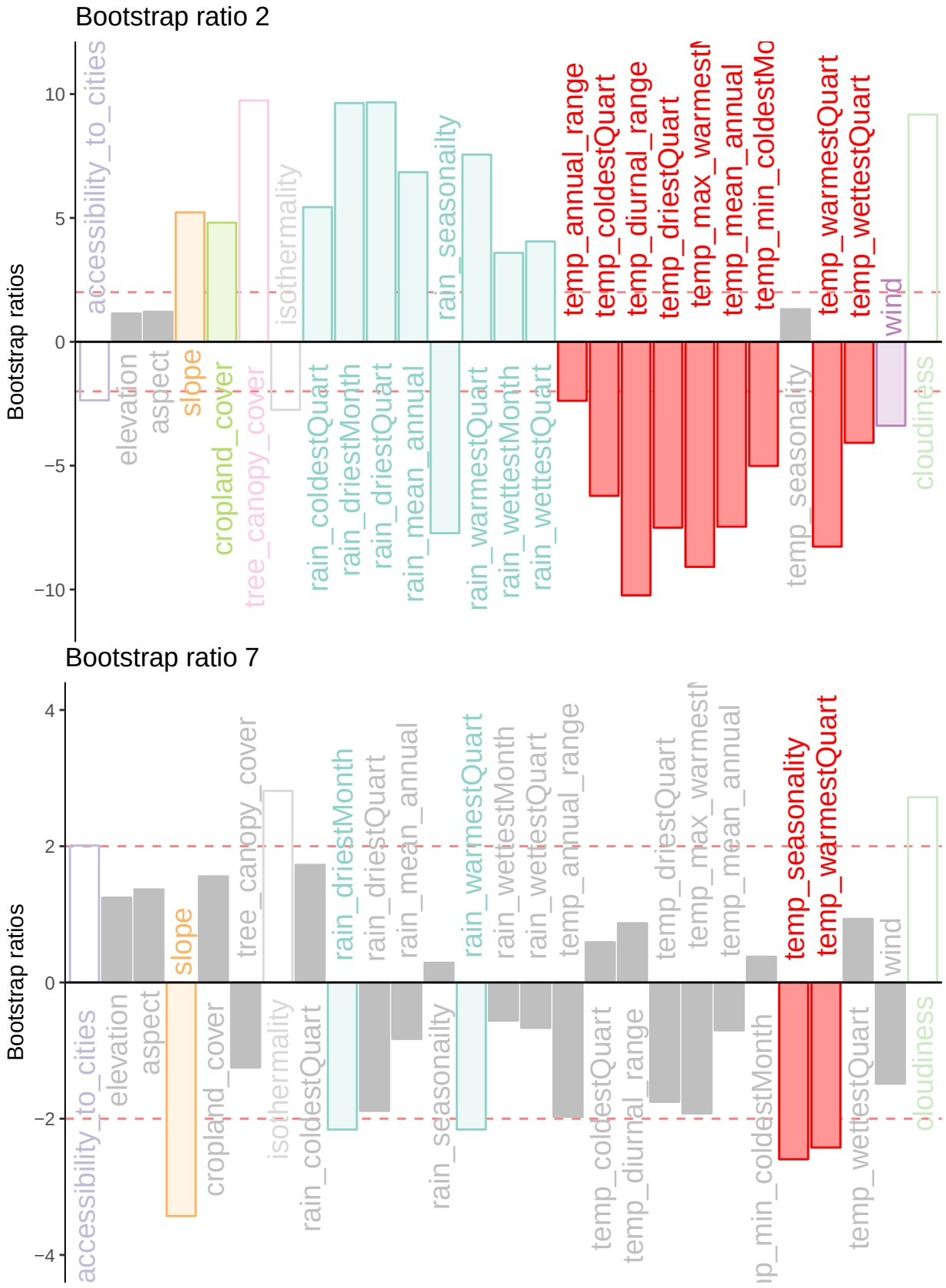
- With Bootstrap Ratio

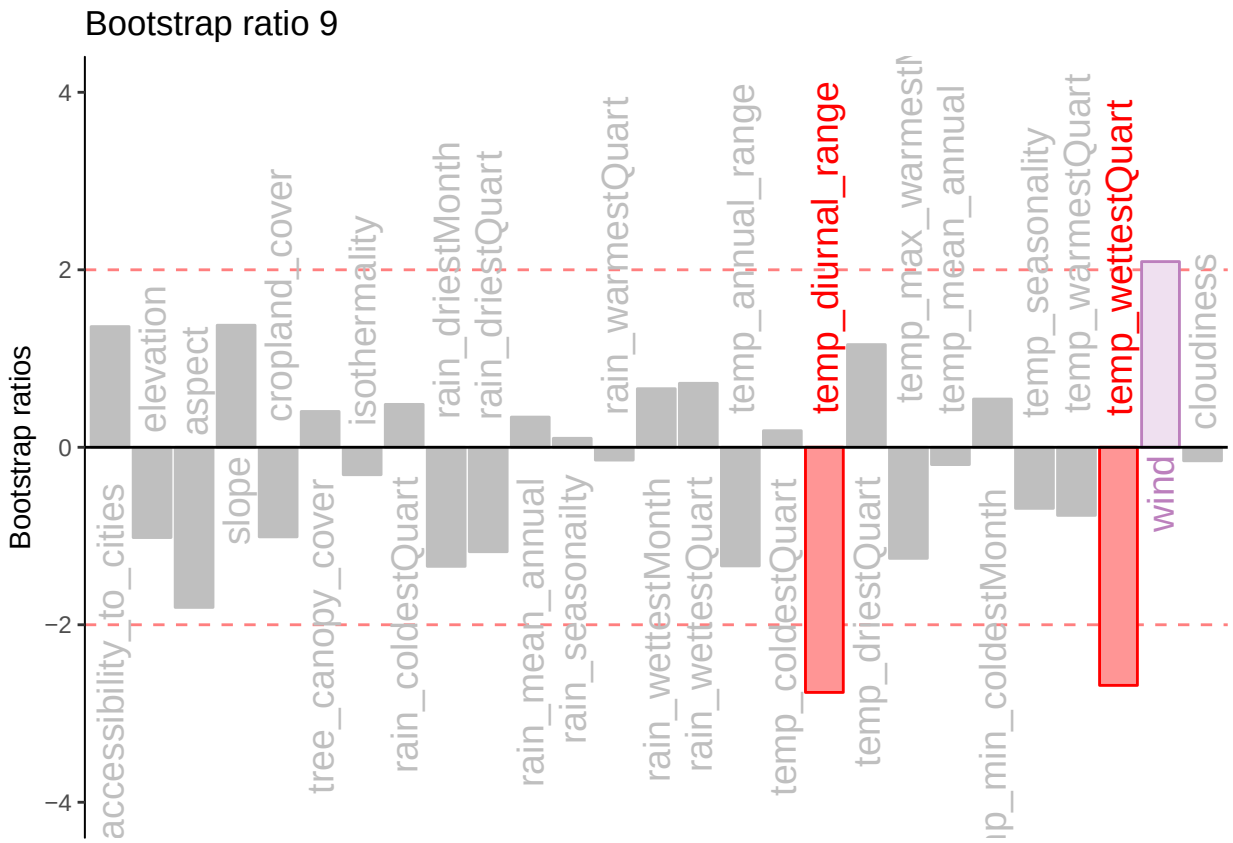
```
BR <- country_env_pca_inf$Inference.Data$fj.boots$tests$boot.ratios

for (i in c(1, 2, 7, 9)) {
  laDim = i
  ba001.BR1 <- PrettyBarPlot2(BR[,laDim],
                              threshold = 2,
                              font.size = 5,
```

```
                                color4bar = gplots::col2hex(col4J), # we need hex code
                                main = paste0('Bootstrap ratio ',laDim),
                                ylab = 'Bootstrap ratios'
                                #ylim = c(1.2*min(BR[,laDim]), 1.2*max(BR[,laDim]))
  )
  print(ba001.BR1)
}
```

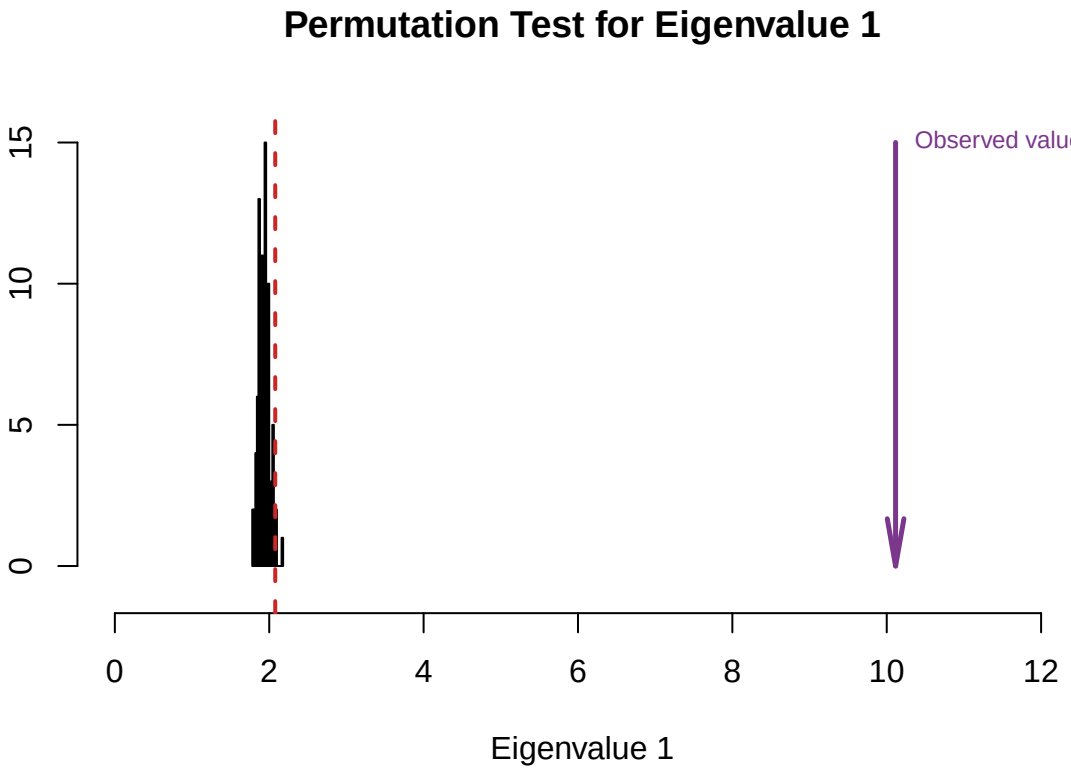


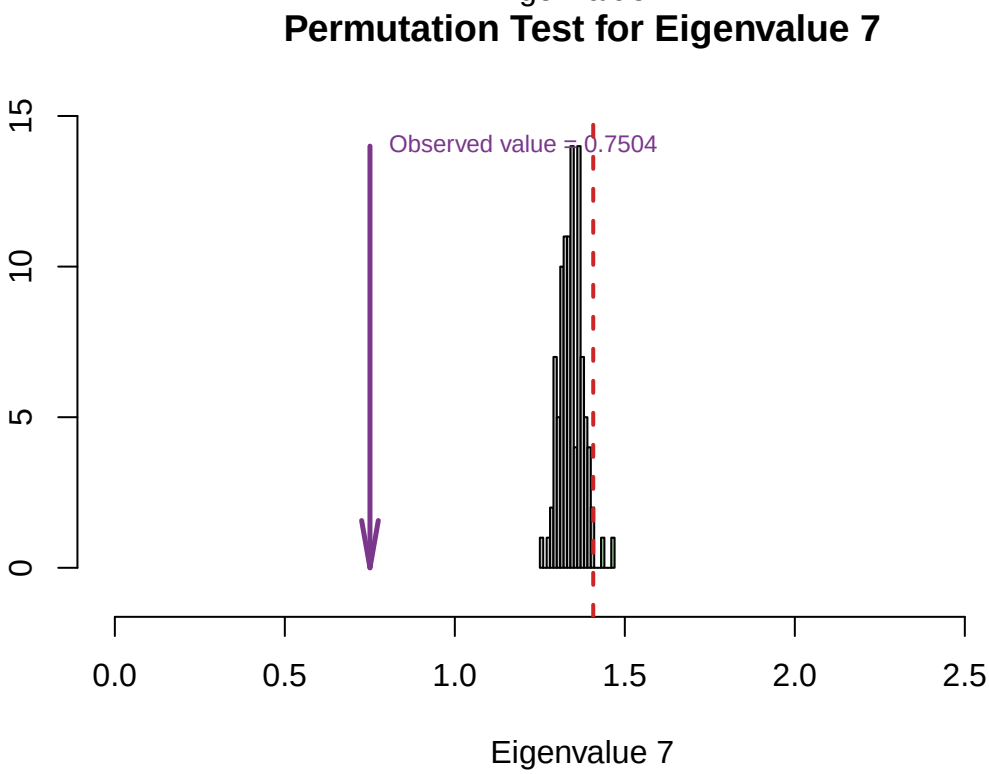

Bootstrap ratio 2
Bootstrap ratios
10
5
0
−5
−10
accessibility_to_cities
elevation
aspect
slope
cropland_cover
tree_canopy_cover
isothermality
rain_coldestQuart
rain_driestMonth
rain_driestQuart
rain_mean_annual
rain_seasonality
rain_warmestQuart
rain_wettestMonth
rain_wettestQuart
temp_annual_range
temp_coldestQuart
temp_diurnal_range
temp_driestQuart
temp_max_warmestM
temp_mean_annual
temp_min_coldestMo
temp_seasonality
temp_warmestQuart
temp_wettestQuart
wind
cloudiness
Bootstrap ratio 7
Bootstrap ratios
4
2
0
−2
−4
accessibility_to_cities
elevation
aspect
slope
cropland_cover
tree_canopy_cover
isothermality
rain_coldestQuart
rain_driestMonth
rain_driestQuart
rain_mean_annual
rain_seasonality
rain_warmestQuart
rain_wettestMonth
rain_wettestQuart
temp_annual_range
temp_coldestQuart
temp_diurnal_range
temp_driestQuart
temp_max_warmestM
temp_mean_annual
p_min_coldestMonth
temp_seasonality
temp_warmestQuart
temp_wettestQuart
wind
cloudiness



Permutation Test

```
for (i in c(1, 2, 7, 9)) {
zeDim = i
pH1 <- prettyHist(
  distribution = country_env_pca_inf$Inference.Data$components$eigs.perm[,zeDim],
  observed = country_env_pca_inf$Fixed.Data$ExPosition.Data$eigs[zeDim],
  xlim = c(0, country_env_pca_inf$Fixed.Data$ExPosition.Data$eigs[zeDim]+2), # needs to be set by hand
  breaks = 20,
  border = "black",
  main = paste0("Permutation Test for Eigenvalue ",zeDim),
  xlab = paste0("Eigenvalue ",zeDim),
  ylab = "",
  counts = FALSE,
  cutoffs = c( 0.975))
}
```

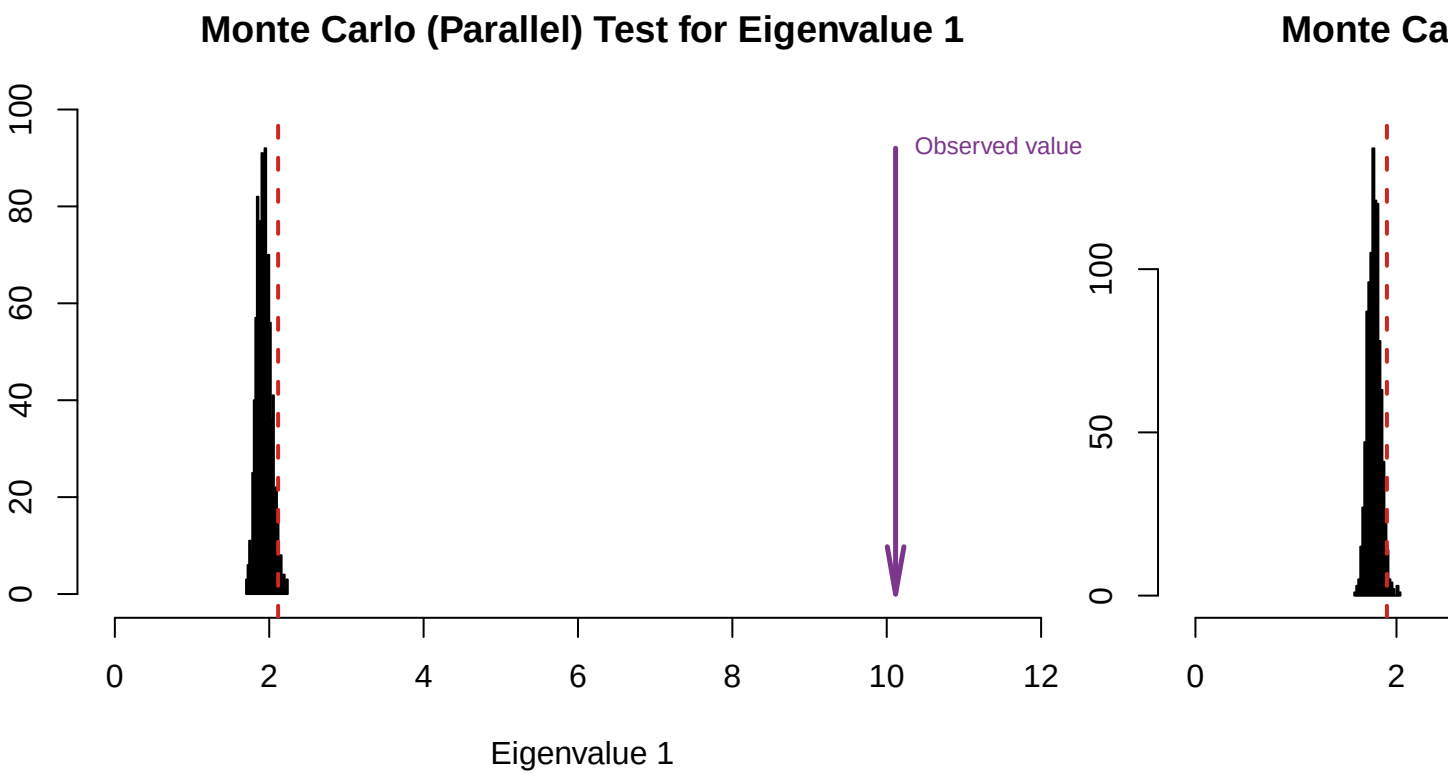


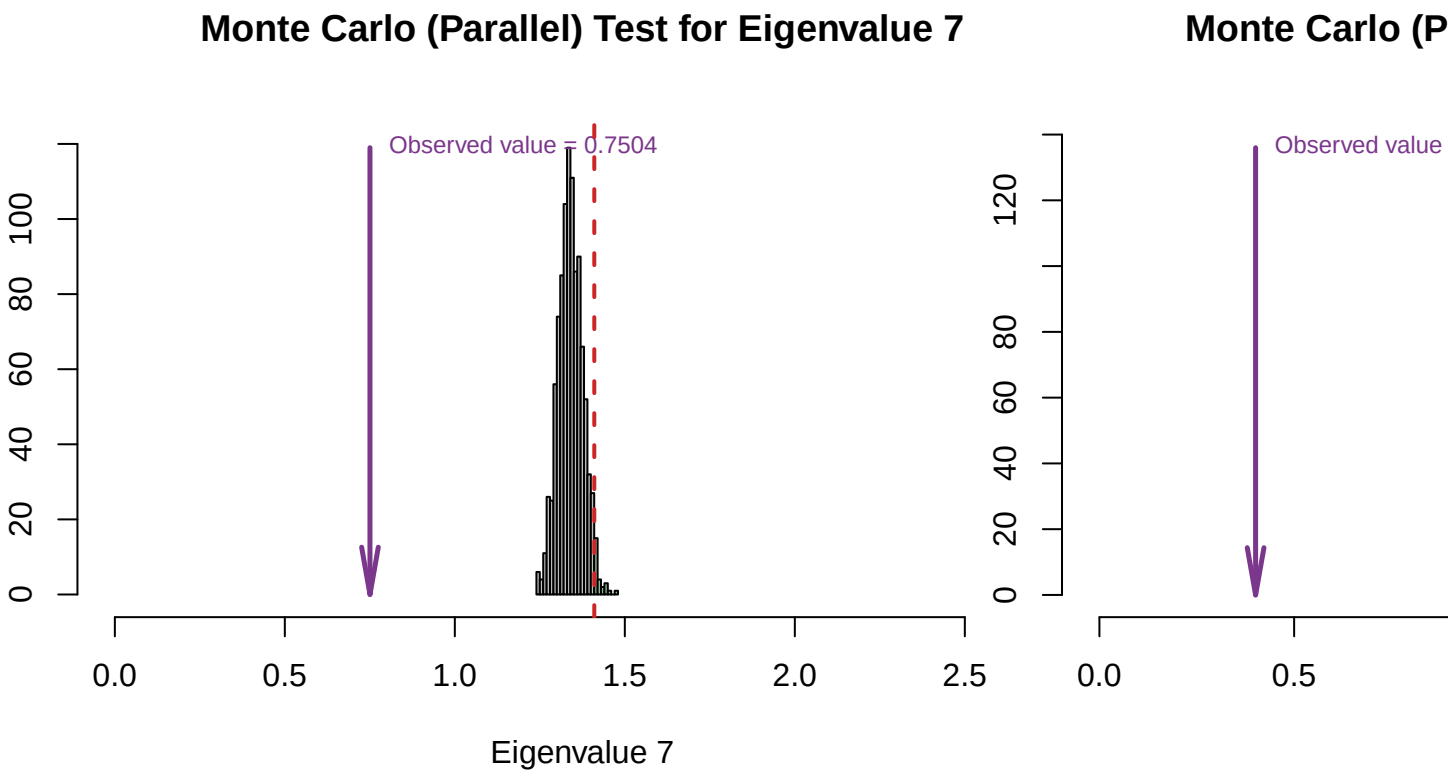


Parallet Test

```
country_env_pca_mc <- data4PCCAR::monteCarlo.eigen(X = country_env_df_for_pca, nIter = 1000)
for (i in c(1, 2, 7, 9)) {
  zeDim = i
```

```
  pH1.p <- prettyHist(country_env_pca_mc$rand.eigs[,zeDim],
                      observed = country_env_pca_mc$fixed.eigs[zeDim],
                      xlim = c(0, country_env_pca_mc$fixed.eigs[zeDim]+2), # needs to set by hand
                      breaks = 20,
                      border = "black",
                      main = paste0("Monte Carlo (Parallel) Test for Eigenvalue ",zeDim),
                      xlab = paste0("Eigenvalue ",zeDim),
                      ylab = "",
                      counts = FALSE,
                      cutoffs = c( 0.975))
}
```

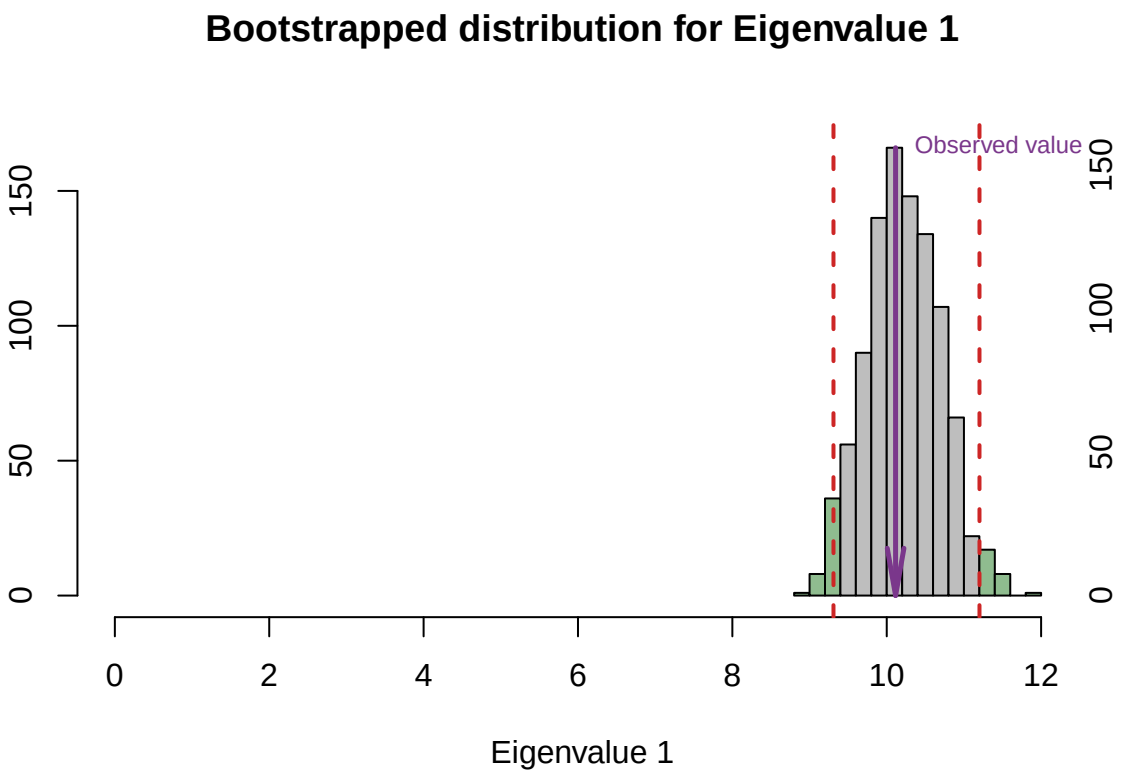


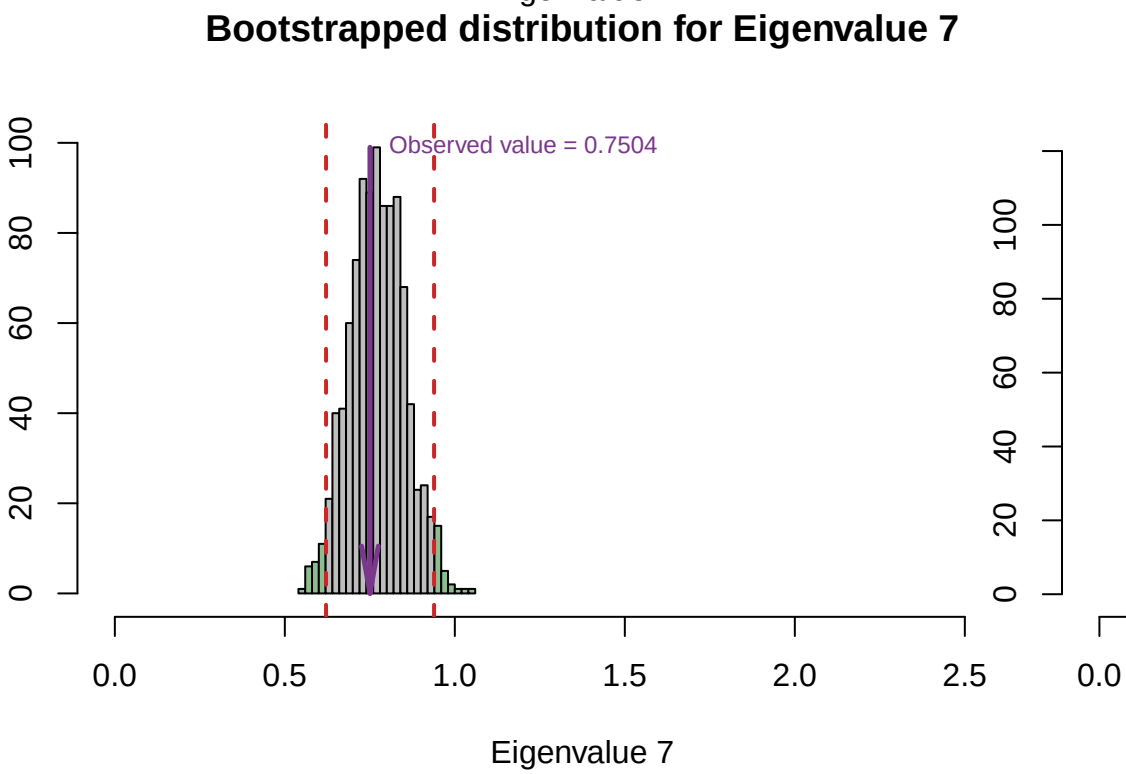


Bootstrap Test

```
#country_env_pca_br <- PTCA4CATA::Boot4Mean(country_env_pca$ExPosition.Data$fi, design = country_env_df
country_env_pca_bs <- data4PCCAR::boot.eigen(X = country_env_df_for_pca, nIter = 1000)

for (i in c(1, 2, 7, 9)) {
  zeDim = i
  prettyHist(country_env_pca_bs$boot.eigs[,zeDim],
                    observed = country_env_pca_bs$fixed.eigs[zeDim],
                    xlim = c(0, country_env_pca_bs$fixed.eigs[zeDim]+2), # needs to set by hand
                    breaks = 20,
                    border = "black",
                    main = paste0("Bootstrapped distribution for Eigenvalue ",zeDim),
                    xlab = paste0("Eigenvalue ",zeDim),
                    ylab = "",
                    counts = FALSE,
                    cutoffs = c(0.025, 0.975))
}
```


Bootstrapped distribution for Eigenvalue 1
Observed value
0
50
100
150
0
2
4
6
8
10
12
Eigenvalue 1


Bootstrapped distribution for Eigenvalue 7
Observed value = 0.7504
0
20
40
60
80
100
0.0
0.5
1.0
1.5
2.0
2.5
Eigenvalue 7