Correlplot: a collection of functions for the graphical representation of correlation matrices

version 1.0.4

Jan Graffelman[†,‡]

[†]Department of Statistics and Operations Research
Universitat Politècnica de Catalunya
Avinguda Diagonal 647, 08028 Barcelona, Spain.
jan.graffelman@upc.edu

[‡]Department of Biostatistics
University of Washington
University Tower, 15th Floor
4333 Brooklyn Avenue
Seattle, WA 98105-9461

September 2022

1 Introduction

This documents gives some instructions on how to create graphical representations of correlation matrices in the statistical environment *R* [R Core Team, 2022] using package **Correlplot**, using a variety of different statistical methods. We use principal component analysis (PCA), multidimensional scaling (MDS), principal factor analysis (PFA), weighted alternating least squares (WALS), correlograms (CRG) and corrgrams to produces displays of correlation structure. The outline of this vignette is as follows. Section 2 explains how to install the **Correlplot** package. Section 3 shows how to use the functions of the package for creating all graphical representaions (biplots, correlograms, MDS maps, etc.) for a given correlation matrix. The computation of goodness-of-fit statistics is also addressed. All methods are illustrated on a single data set, the wheat kernel data introduced below.

2 Installation

The package **Correlplot** can be installed in *R* by typing:

```
install.packages("Correlplot")
library("Correlplot")
```

This will download **Correlplot** from the CRAN server. This instruction will make, among others, the functions `correlogram`, `pfa`, `ipSymLS` and `rmse` available. Some data sets and correlation matrices are included in the package, and can be accessed with the `data` instruction. By typing the instruction `data(package="Correlplot")` a list of all correlation and data matrices available in the package will appear. We will also make use of the packages **calibrate**, **corrgram** and **xtable**, and first connect these:

```
> #install.packages("calibrate")
> #install.packages("corrplot")
> #install.packages("xtable")
> library(calibrate)
> library(corrplot)
> library(xtable)
```

3 Graphical representations of a correlation matrix

In this section we indicate how to create different plots of a correlation matrix, and how to obtain the goodness-of-fit of the displays. We will subsequently treat the corrgram, the correlogram, the PCA-based correlation biplot, the PFA-based correlation biplot, an MDS-based map of correlation structure and WALS-based correlation biplots. Throughout this vignette, we will use a wheat kernel data set taken from the UCI Machine Learning Repository (`https://archive.ics.uci.edu/ml/datasets/seeds`) in order to illustrate the different plots.
The wheat kernel data [Charytanowicz et al., 2010] consists of 210 wheat kernels, of which the variables *area* (A), *perimeter* (P), *compactness* ($C = 4 * \pi * A/P^2$), *length*, *width*, *asymmetry coefficient* and *groove* (length of kernel groove) were registered. There are 70 kernels of each of the varieties *Kama*, *Rosa* and *Canadian*; here we will only use the kernels of variety *Kama*. The data is made available with:

```
> library(Correlplot)
> data("Kernels")
> X <- Kernels[Kernels$variety==1,]
> X <- X[,-8]
> head(X)

   area perimeter compactness length width asymmetry groove
1 15.26     14.84      0.8710  5.763 3.312     2.221  5.220
2 14.88     14.57      0.8811  5.554 3.333     1.018  4.956
3 14.29     14.09      0.9050  5.291 3.337     2.699  4.825
4 13.84     13.94      0.8955  5.324 3.379     2.259  4.805
5 16.14     14.99      0.9034  5.658 3.562     1.355  5.175
6 14.38     14.21      0.8951  5.386 3.312     2.462  4.956
```

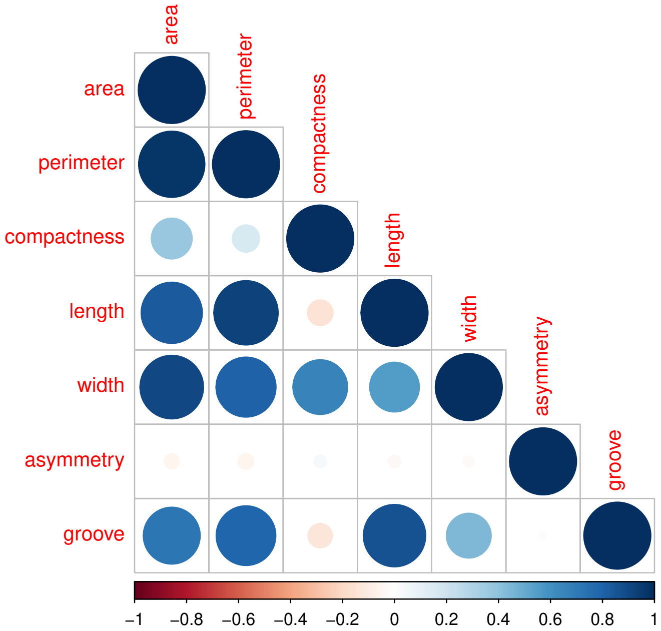
The correlation matrix of the variables is given by:

```
> R <- cor(X)
> xtable(R,digits=3)
```

The corrgram The corrgram ([Friendly, 2002]) is a tabular display of the entries of a correlation matrix that uses colour and shading to represent correlations. Corrgrams can be made with the fuction `corrplot`.
This shows most correlations are positive, and correlations with *asymmetry* are weak.

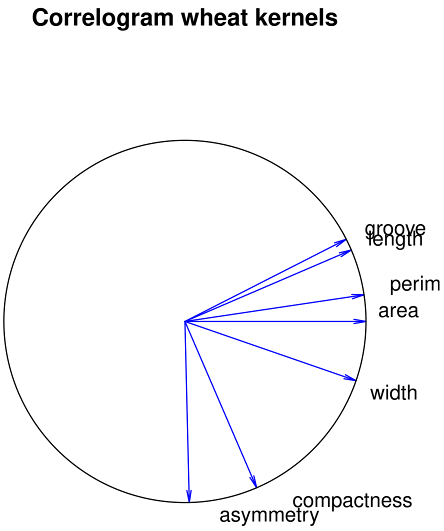
	area	perimeter	compactness	length	width	asymmetry	groove
area	1.000	0.976	0.371	0.835	0.900	-0.050	0.721
perimeter	0.976	1.000	0.165	0.921	0.802	-0.054	0.794
compactness	0.371	0.165	1.000	-0.146	0.667	0.037	-0.131
length	0.835	0.921	-0.146	1.000	0.551	-0.037	0.866
width	0.900	0.802	0.667	0.551	1.000	-0.027	0.447
asymmetry	-0.050	-0.054	0.037	-0.037	-0.027	1.000	-0.011
groove	0.721	0.794	-0.131	0.866	0.447	-0.011	1.000

```
> #install.packages("corrplot")
> library(corrplot)
> R <- cor(X)
> corrplot(R, method="circle",type="lower")
```



The correlogram The correlogram ([Trosset, 2005]) represents correlations by the cosines between vectors.

```
> theta.cos <- correlogram(R,main="Correlogram wheat kernels",
+              xlim=c(-1.3,1.3),ylim=c(-1.3,1.3))
```



The approximation this gives to the correlation matrix calculated by

```
> Rhat.cor <- angleToR(theta.cos)
```

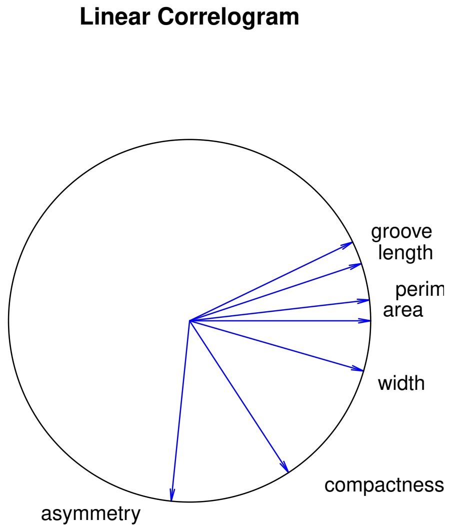
and the root mean squared error (RMSE) of the approximation, is calculated as

```
> rmse.crg <- rmse(R,Rhat.cor,verbose=TRUE)
```

```
7 variables
rmse (off-diagonal) =  0.2632838
```

which shows this representation has a large amount of error. The correlogram can be modified by using a linear interpretation rule, rendering correlations linear in the angle [Graffelman, 2013]. This representation is obtained by:

```
> theta.lin <- correlogram(R,ifun="lincos",labs=colnames(R),
+                          main="Linear Correlogram",
+                          xlim=c(-1.3,1.3),ylim=c(-1.3,1.3))
```



The approximation to the correlation matrix by using this linear interpretation function is calculated by

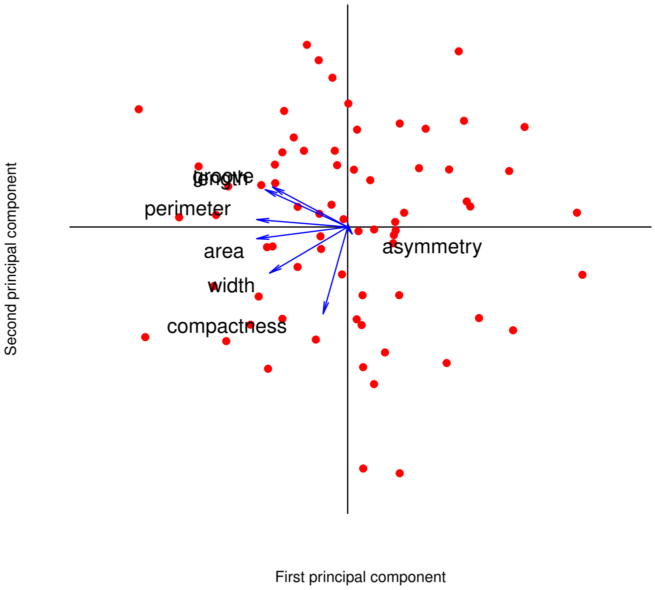
```
> Rhat.corlin <- angleToR(theta.lin,ifun="lincos")
> rmse.lin <- rmse(R,Rhat.corlin,verbose=TRUE)

7 variables
rmse (off-diagonal) =  0.1801166
```

The linear representation is seen to improve the approximation.

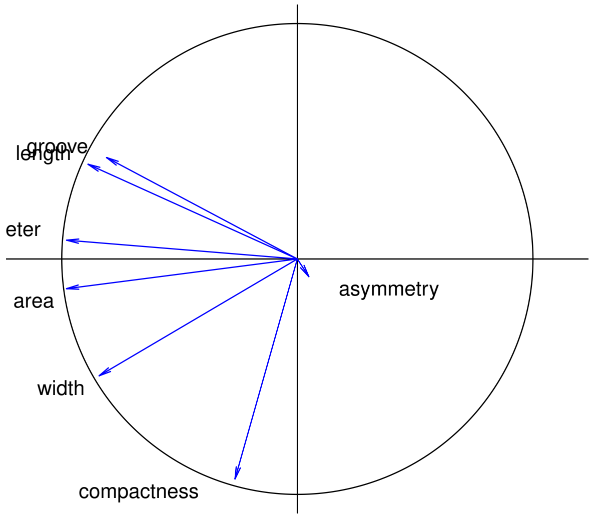
The PCA biplot of the correlation matrix We create a PCA biplot of the correlation matrix, doing the calcutions for a PCA by hand, using the singular value decomposition of the (scaled) standardized data. Alternatively, standard R function `princomp` may be used to obtain the coordinates needed for the correlation biplot. We use function `bplot` from package **calibrate** to make the biplot:

```
> n <- nrow(X)
> Xt <- scale(X)/sqrt(n)
> res.svd <- svd(Xt)
> Fs <- sqrt(n)*res.svd$u
> Gp <- res.svd$v%*%diag(res.svd$d)
> bplot(Fs,Gp,colch=NA,collab=colnames(X),
+       xlab = "First principal component",
+       ylab="Second principal component")
```



The joint representation of kernels and variables emphasizes this is a biplot of the (standardized) data matrix. However, this plot is a *double biplot* because scalar products between variable vectors approximate the correlation matrix. We stress this by plotting the variable vectors only, and adding a unit circle:

```
> bplot(Gp,Gp,colch=NA,rowch=NA,collab=colnames(X),
+       xl=c(-1,1),yl=c(-1,1))
> circle()
```



The PCA biplot of the correlation matrix can be obtained from a correlation-based PCA or also directly from the spectral decomposition of the correlation matrix. The rank two approximation, by scalar products between vectors, and the RMSE are calculated by:

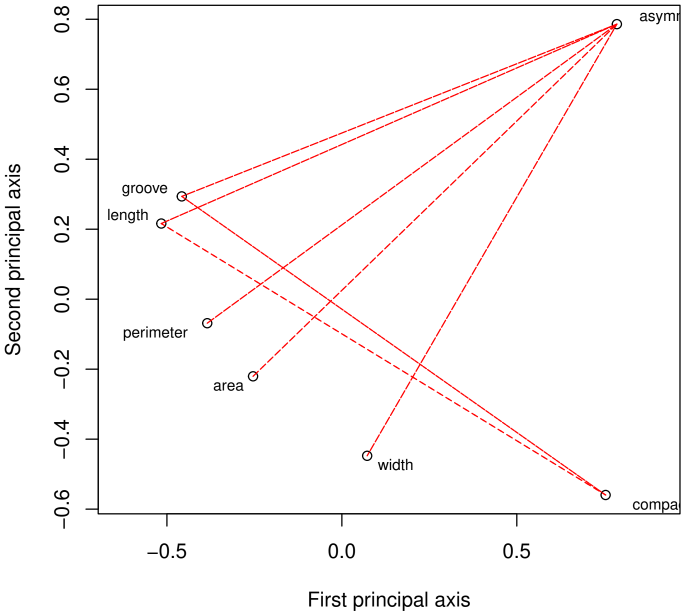
```
> Rhat.pca <- Gp[,1:2]%*%t(Gp[,1:2])
> rmse.pca <- rmse(R,Rhat.pca,verbose=TRUE)

7 variables
rmse (off-diagonal) =  0.02642067
```

which is a considerable improvement over the previous correlograms.

The MDS map of a correlation matrix We transform correlations to distances with the $\sqrt{2(1-r)}$ transformation, and use the `cmdscale` function from the **stats** package to perform metric multidimensional scaling. We mark negative correlations with a dashed red line.

```
> Di <- sqrt(2*(1-R))
> out.mds <- cmdscale(Di,eig = TRUE)
> Fp <- out.mds$points
> plot(Fp[,1],Fp[,2],asp=1,xlab="First principal axis",
+      ylab="Second principal axis")
> textxy(Fp[,1],Fp[,2],colnames(R),cex=0.75)
> ii <- which(R < 0,arr.ind = TRUE)
> for(i in 1:nrow(ii)) {
+   segments(Fp[ii[i,1],1],Fp[ii[i,1],2],
+            Fp[ii[i,2],1],Fp[ii[i,2],2],col="red",lty="dashed")
+ }
```



We calculate distances in the map, convert back to correlations, and compute the RMSE.

```
> Dest <- as.matrix(dist(Fp[,1:2]))
> Rhat.mds <- 1-0.5*Dest*Dest
> rmse.mds <- rmse(R,Rhat.mds,verbose=TRUE)

7 variables
rmse (off-diagonal) =  0.07385311
```

The approximation by distance worsens with respect to the representation by scalar products in PCA.

The PFA biplot of a correlation matrix Principal factor analysis can be performed by the function pfa of package **Correlplot**.

```
> out.pfa <- pfa(X)

Initial communalities
[1] 0.99889901 0.99874292 0.97714079 0.95523330 0.95143278 0.07280894 0.77171779
Final communalities
[1] 1.000000000 0.988505336 0.841902892 0.992114945 0.977490455 0.002200171
[7] 0.741231621
27  iterations till convergence
Specific variances:
[1] 0.000000000 0.011494664 0.158097108 0.007885055 0.022509545 0.997799829
[7] 0.258768379
Variance explained by each factor
[1] 4.154912 1.390012
Loadings:
           [,1]        [,2]
[1,] -0.9932963  0.12182699
[2,] -0.9897642 -0.09419243
[3,] -0.2611277  0.87961082
[4,] -0.8973292 -0.43233698
[5,] -0.8499806  0.50499849
[6,]  0.0380343  0.02745110
[7,] -0.7689868 -0.38715752

> L <- out.pfa$La
```

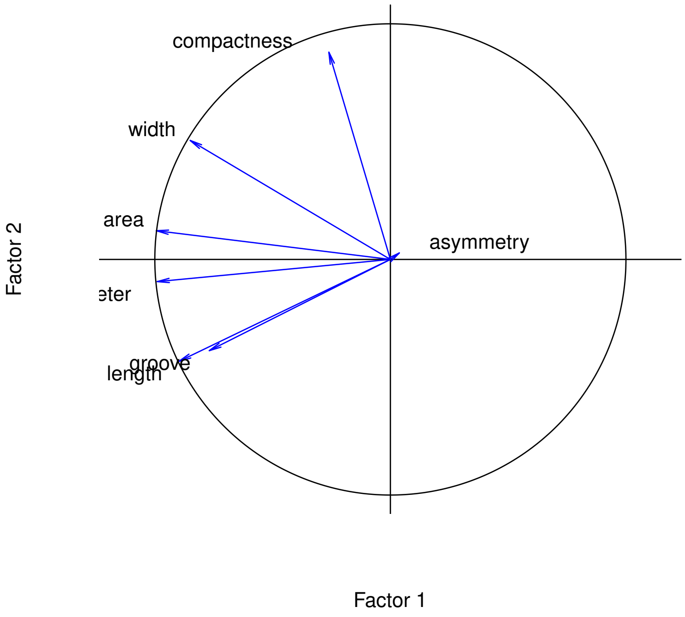
The biplot of the correlation matrix obtained by PFA is in fact the same as what is known as a factor loading plot in factor analysis, to which a unit circle can be added. The approximation to the correlation matrix and its RMSE are calculated as:

```
> Rhat.pfa <- L[,1:2]%*%t(L[,1:2])
> rmse.pfa <- rmse(R,Rhat.pfa,verbose=TRUE)

7 variables
rmse (off-diagonal) =  0.01119688
```

To make the factor loading plot, aka PFA biplot of the correlation matrix:

```
> opar <- par(bty="n",xaxt="n",yaxt="n")
> plot(L[,1],L[,2],pch=NA,asp=1,xlim=c(-1,1),ylim=c(-1,1),
+       xl="Factor 1",yl="Factor 2")
> origin()
> arrows(0,0,L[,1],L[,2],angle=10,length=0.1,col="blue")
> textxy(L[,1],L[,2],colnames(X),cex=1)
> circle()
> par(opar)
```

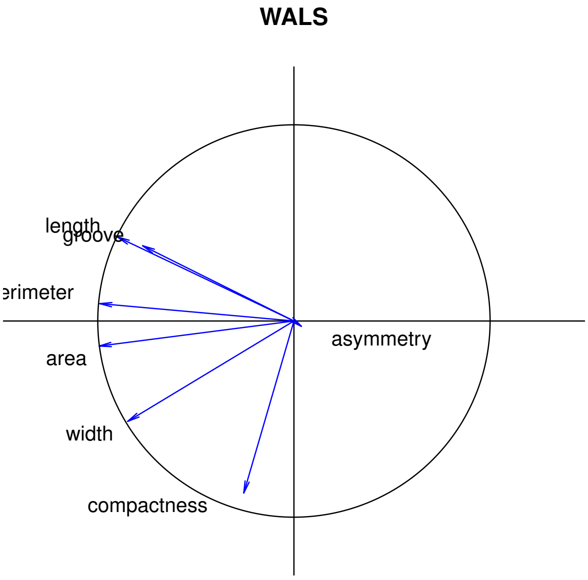


The RMSE of the plot obtained by PFA is lower than the RMSE obtained by PCA. Note that variable *Area* reaches the unit circle for having a communality of 1.

The WALS biplot of a correlation matrix The correlation matrix can also be factored using weighted alternating least squares, avoiding the fit of the ones on the diagonal of the correlation matrix by assigning them weight 0, using function ipSymLS [De Leeuw, 2006].

```
> W <- matrix(1,nrow(R),nrow(R))
> diag(W) <- 0
> Fp.als <- ipSymLS(R,w=W,eps=1e-15)
```

```
> bplot(Fp.als,Fp.als,rowch=NA,colch=NA,collab=colnames(R),
+       xl=c(-1.2,1.2),yl=c(-1.2,1.2),main="WALS")
> circle()
```



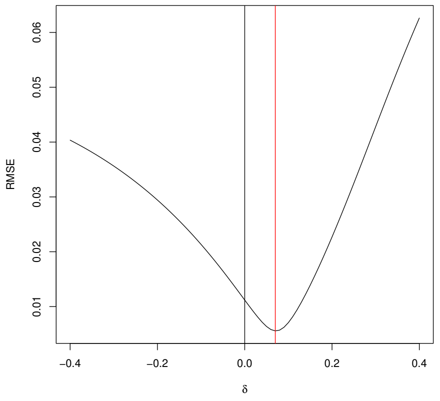
Weighted alternating least squares has, in contrast to PFA, no restriction on the vector length, and variable *Top* is seen to move out of the unit circle.

```
> Rhat.wals <- Fp.als%*%t(Fp.als)
> rmse.als <- rmse(R,Rhat.wals,verbose=TRUE)
```

```
7 variables
rmse (off-diagonal) =  0.01118619
```

This is only slightly below the RMSE of PFA.

The WALS biplot of an adjusted correlation matrix The adjusted correlation matrix is calculated as $\mathbf{R}_a = \mathbf{R} - \delta\mathbf{J}$. By exploring the RMSE over a grid of values for δ, as shown in the figure below, $\delta = 0.07$ is found to be the optimum value.



Function `ipSymLS` is applied to the adjusted correlation by subtracting δ. When calculating the fitted correlation matrix, δ is added back.

```
> delta <- 0.07
> W <- matrix(1,nrow(R),nrow(R))
> diag(W) <- 0
> Fp.adj <- ipSymLS(R-delta,w=W,verbose=FALSE,eps=1e-10,itmax=1000)
```

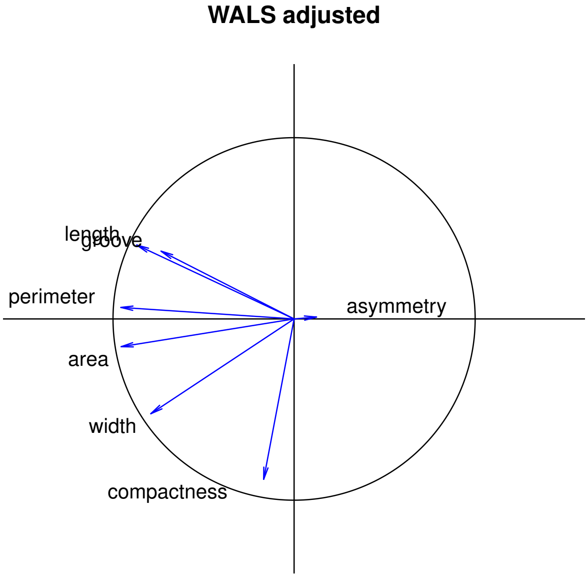
The fitted correlation matrix and its RMSE are now calculated as:

```
> Rhat.adj <- Fp.adj%*%t(Fp.adj) + delta
> rmse.adj <- rmse(R,Rhat.adj,verbose=TRUE)

7 variables
rmse (off-diagonal) =  0.005564538
```

This comes closer to the sample correlation matrix than WALS without adjustment.

```
> bplot(Fp.adj,Fp.adj,rowch=NA,colch=NA,collab=colnames(R),
+       xl=c(-1.3,1.3),yl=c(-1.3,1.3),main="WALS adjusted")
> circle()
```



We summarize the values of the RMSE of all methods in a table below:

```
> rmsevector <- c(rmse.crg,rmse.lin,rmse.pca,rmse.mds,rmse.pfa,rmse.als,rmse.adj)
> methods <- c("Correlogram (cosine)","Correlogram (linear)","PCA","MDS",
+ "PFA","WALS R","WALS Radj")
> xtable(data.frame(methods,rmsevector),digits=c(0,0,4))
```

	methods	rmsevector
1	Correlogram (cosine)	0.2633
2	Correlogram (linear)	0.1801
3	PCA	0.0264
4	MDS	0.0739
5	PFA	0.0112
6	WALS R	0.0112
7	WALS Radj	0.0056

References

M. Charytanowicz, J. Niewczas, P. Kulczycki, P. Kowalski, S. Lukasik, and S. Zak. A complete gradient clustering algorithm for features analysis of x-ray images. In E. Pietka and J. Kawa, editors, *Information Technologies in Biomedicine*, pages 15–24, Berlin-Heidelberg, 2010. Springer-Verlag.

J. De Leeuw. A decomposition method for weighted least squares low-rank approximation of symmetric matrices. *Department of Statistics, UCLA*, 2006. Retrieved from https://escholarship.org/uc/item/1wh197mh.

M. Friendly. Corrgrams: exploratory displays for correlation matrices. *The American Statistician*, 56(4):316–324, 2002.

J. Graffelman. Linear-angle correlation plots: new graphs for revealing correlation structure. *Journal of Computational and Graphical Statistics*, 22(1): 92–106, 2013.

R Core Team. *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria, 2022. URL `http://www.R-project.org/`.

M. W. Trosset. Visualizing correlation. *Journal of Computational and Graphical Statistics*, 14(1):1–19, 2005.

Acknowledgements

This work was supported by the Spanish Ministry of Science, Innovation and Universities and the European Regional Development Fund under Grant RTI2018-095518-B-C22 (MCIU/AEI/FEDER); and the National Institutes of Health under Grant GM075091.