R-bloggers
R news and tutorials contributed by hundreds of R bloggers

- [Home]
- [About]
- [RSS]
- [add your blog!]
- [Learn R]
- [R jobs���]
- [Contact us]

Never miss an update!
Subscribe to R-bloggers to receive e-mails with the latest R posts.
(You will not see this message again.)

Your E-mail...

Click here to close (This popup will not appear again)

Welcome!

Follow @rbloggers 84.5K

Here you will find daily **news and tutorials about R**, contributed by hundreds of bloggers. There are many ways to **follow us** -

By e-mail:

Your e-mail here

Subscribe

52665 readers
BY FEEDBURNER

On Facebook:

R blogg…
79K likes

Like Page

Be the first of your friends to like this

If you are an R blogger yourself you are invited to add your own R content feed to this site (**Non-English** R bloggers should add themselves- here)

Jobs for R-users

- Data Analytics Auditor, Future of Audit Lead
- Data Analytics Auditor, Future of Audit Lead @ London or Newcastle
- Senior Scientist, Translational Informatics @ Vancouver, BC

- Senior Principal Data Scientist @ Mountain View, California, United States
- Technical Research Analyst – New York, U.S.

Recent Posts

- How to Add a Vignette to a Package in RStudio
- Why R? Webinar – Humanitarian Data Analysis with R
- ¿Cómo Subir una Base de Datos a un Hosting?
- #FunDataFriday – See your TwitterBot
- Online R trainings: Learning data science – live and interactive
- How to Set Up TensorFlow 2 in R in 5 Minutes (BONUS Image Recognition Tutorial)
- Implied risk premia
- R doesn't need to throttle AWS Athena anymore
- glmnet v4.0: generalizing the family parameter
- Data Privacy in the Age of COVID-19
- Using {drake} for Machine Learning
- RStudio Team Admin Training – Remotely
- Use existing functions and data through packages
- Checking your R package on Solaris
- Intro to {polite} Web Scraping of Soccer Data with R!

Other sites

- Jobs for R-users
- SAS blogs

Computing and visualizing PCA in R

November 28, 2013
By thiagogm

Following my introduction to PCA, I will demonstrate how to
visualize PCA in R. There are many packages and functio
apply PCA in R. In this post I will use the function prcom
stats package. I will also show how to visualize PCA in R
R graphics. However, my favorite visualization function f
ggbiplot, which is implemented by Vince Q. Vu and a
github. Please, let me know if you have better ways to visualize PCA in R.

Computing the Principal Components (PC)

I will use the classical `iris` dataset for the demonstration. The data contain four continuous variables which corresponds to physical measures of flowers and a categorical variable describing the flowers' species.

```
# Load data
data(iris)
head(iris, 3)

  Sepal.Length Sepal.Width Petal.Length
1          5.1         3.5          1.4
2          4.9         3.0          1.4
3          4.7         3.2          1.3
```

We will apply PCA to the four continuous variables and use the categorical variable to visualize the PCs later. Notice that in the following code we apply a log transformation to the continuous variables as suggested by [1] and set `center` and `scale.` equal to `TRUE` in the call to `prcomp` to standardize the variables prior to the application of PCA:

```
# log transform
log.ir <- log(iris[, 1:4])
ir.species <- iris[, 5]

# apply PCA - scale. = TRUE is highly
# advisable, but default is FALSE.
ir.pca <- prcomp(log.ir,
                 center = TRUE,
                 scale. = TRUE)
```

Since skewness and the magnitude of the variables influence the resulting PCs, it is good practice to apply skewness transformation, center and scale the variables prior to the application of PCA. In the example above, we applied a log transformation to the variables but we could have been more general and applied a Box and Cox transformation [2]. See at the end of this post how to perform all those transformations and then apply PCA with only one call to the `preProcess` function of the `caret` package.

Analyzing the results

The `prcomp` function returns an object of class `prcomp`, which have deviation of each of the four PCs, and their rotation (or loadings), which are the coefficients of the linear combinations of the continuous variables.

```
# print method
print(ir.pca)

Standard deviations:
[1] 1.7124583 0.9523797 0.3647029 0.165

Rotation:
                    PC1         PC2
Sepal.Length  0.5038236 -0.45499872  0.
Sepal.Width  -0.3023682 -0.88914419 -0.
Petal.Length  0.5767881 -0.03378802 -0.
Petal.Width   0.5674952 -0.03545
```

The plot method returns a plot of the variances (y-axis) asso

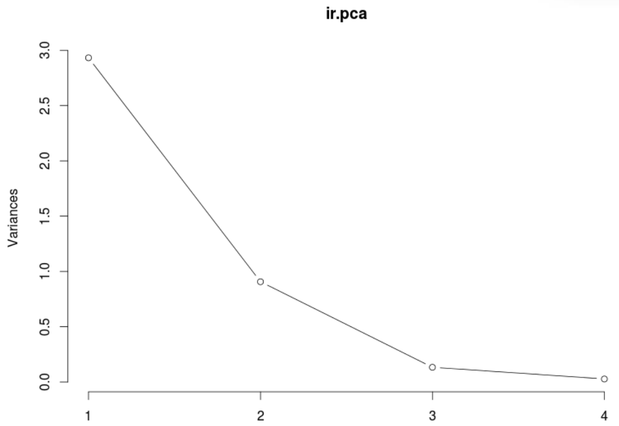
the PCs (x-axis). The Figure below is useful to decide how m

retain for further analysis. In this simple case with only 4 PC

a hard task and we can see that the first two PCs explain m

variability in the data.

```
# plot method
plot(ir.pca, type = "l")
```



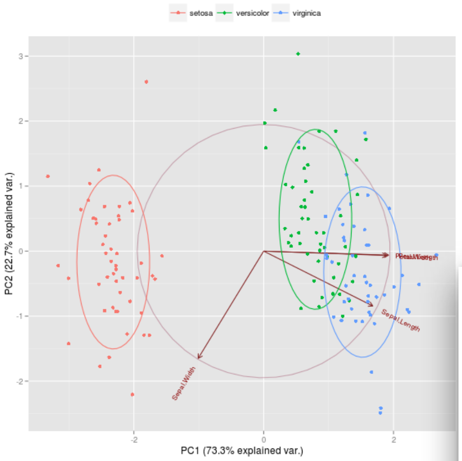
The summary method describe the importance of the PCs. The first row describe again the standard deviation associated with each PC. The second row shows the proportion of the variance in the data explained by each component while the third row describe the cumulative proportion of explained variance. We can see there that the first two PCs accounts for more than 95% of the variance of the data.

```
# summary method
summary(ir.pca)

Importance of components:
                          PC1    PC2
Standard deviation     1.7125 0.9524 0.3
Proportion of Variance 0.7331 0.2268 0.0
Cumulative Proportion  0.7331 0.9599 0.9
```

We can use the predict function if we observe new data and want to predict their PCs values. Just for illustration pretend the last two rows of the iris data has just arrived and we want to see what is their PCs values:

```
# Predict PCs
predict(ir.pca,
        newdata=tail(log.ir, 2))

          PC1         PC2        PC3
149 1.0809930 -1.01155751 -0.7082289 -0.
150 0.9712116 -0.06158655 -0.5008674 -0.
```

The Figure below is a biplot generated by the function `ggbiplot` of the `ggbiplot` package available on github.



The code to generate this Figure is given by

```
library(devtools)
install_github("ggbiplot", "vqv")

library(ggbiplot)
g <- ggbiplot(ir.pca, obs.scale = 1, va
              groups = ir.species, elli
              circle = TRUE)
g <- g + scale_color_discrete(name = ''
g <- g + theme(legend.direction = 'hori
               legend.position = 'top')
print(g)
```

It projects the data on the first two PCs. Other PCs can be chosen through the argument `choices` of the function. It colors each point according to the flowers’ species and draws a Normal contour line with `ellipse.prob` probability (default to) for each group. More info about `ggbiplot` can be obtained by the usual `?ggbiplot`. I think you will agree that the plot produced by `ggbiplot` is much better than the one produced by `biplot(ir.pca)` (Figure below).

I also like to plot each variables coefficients inside a unit circle to get insight on a possible interpretation for PCs. Figure 4 was generated by this code available on gist.

PCA on caret package

As I mentioned before, it is possible to first apply a Box-Cox transformation to correct for skewness, center and scale each variable and then apply PCA in one call to the `preProcess` function of the `caret` package.

```
require(caret)
trans = preProcess(iris[,1:4],
                   method=c("BoxCox", "ce
                            "scale", "pca
PC = predict(trans, iris[,1:4])
```

By default, the function keeps only the PCs that are necessary to explain at least 95% of the variability in the data, but this can be changed through the argument `thresh`.

```
# Retained PCs
head(PC, 3)

        PC1        PC2
1 -2.303540 -0.4748260
2 -2.151310 -0.6482903
```

```
# Loadings
trans$rotation

                    PC1         PC2
Sepal.Length  0.5202351 -0.38632246
Sepal.Width  -0.2720448 -0.92031253
Petal.Length  0.5775402 -0.04885509
Petal.Width   0.5672693 -0.03732262
```

See Unsupervised data pre-processing for predictive modeling for an introduction of the `preProcess` function.

References:

[1] Venables, W. N., Brian D. R. Modern applied statistics with S-PLUS. Springer-verlag. (Section 11.1)
[2] Box, G. and Cox, D. (1964). An analysis of transformatio
of the Royal Statistical Society. Series B (Methodological) 21

Never miss an update!
Subscribe to R-bloggers to receive e-mails with the latest R posts.
(You will not see this message again.)

Your E-mail...

Click here to close (This popup will not appear again)

Share Tweet

To **leave a comment** for the author, please follow the link and comment on their blog: **Thiago G. Martins » R**.

R-bloggers.com offers **daily e-mail updates** about R news and tutorials about learning R and many other topics. Click here if you're looking to post or find an R/data-science job.

Want to share your content on R-bloggers? click here if you have a blog, or here if you don't.

If you got this far, why not **subscribe for updates** from the site? Choose your flavor: e-mail, twitter, RSS, or facebook...

Like 48 Share Tweet Share

Comments are closed.

Search R-bloggers

Search.. Go

Most visited articles of the week

1. Free Springer Books during COVID19
2. 5 Ways to Subset a Data Frame in R
3. How to write the first for loop in R
4. R – Sorting a data frame by the contents of a column
5. Date Formats in R
6. How to schedule R scripts
7. Installing R packages
8. In-depth introduction to machine learning in 15 hours of expert videos
9. Intro to {polite} Web Scraping of Soccer Data with R!

Sponsors