R-bloggers

R news and tutorials contributed by (573) R bloggers

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

Welcome!

Follow @rbloggers 27.8K

Here you will find daily **news and tutorials about R**, contributed by over 573 bloggers.
There are many ways to **follow us -**
By e-mail:

Your e-mail here
Subscribe
24349 readers
BY FEEDBURNER

On Facebook:

R blogg...
30k 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

- Junior Data Scientist @ Farnham, England
- Vehicle Valuations Manager @ Farnham, England
- Manager – Quantitative Analytics @ London, United Kingdom
- Pharmacometrics Bootcamp @ Wellesley, Massachusetts, US
- Data Scientist for

Search & Hit Enter

Popular Searches

- web scraping
- heatmap
- maps
- shiny
- twitter
- alt=
- boxplot
- hadoop
- time series
- animation
- ggplot2
- trading
- ggplot
- latex
- PCA
- finance
- excel
- quantmod
- googlevis
- RSTUDIO
- how to import image file to R
- rattle
- eclipse
- market research
- knitr
- rcmdr
- tutorial
- coplot
- Map
- SQL

Recent Posts

- functions exercises
- Build your own neural network classifier in R
- There's a party at Alexa's place
- Hadley Wickham's Advanced R in Amsterdam
- Mastering R plot – Part 2: Axis
- What has Kaggle learned from 2 million machine learning models?
- Pitfall of XML package: to know the cause
- Speaking at DataPhilly February 2016
- Introducing Microsoft R Open: Replay and slides
- Shiny Developer Conference 2016 Recap
- Cricket analytics

with cricketr in paperback and Kindle versions
- New Version of “Wrangling F1 Data With R” Just Released…
- Data from the World Health Organization API
- Alternate R Markdown Templates
- Death Comes to Us All

Other sites

- Statistics of Israel
- SAS blogs
- Jobs for R-users

Computing and visualizing PCA in R

November 28, 2013
By thiagogm

(This article was first published on **Thiago G. Martins » R**, and kindly contributed to R-bloggers)

Following my introduction to PCA, I will demonstrate how to apply and visualize PCA in R. There are many packages and functions that can apply PCA in R. In this post I will use the function `prcomp` from the `stats` package. I will also show how to visualize PCA in R using Base R graphics. However, my favorite visualization function for PCA is `ggbiplot`, which is implemented by Vince Q. Vu and available on 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 Petal.
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.

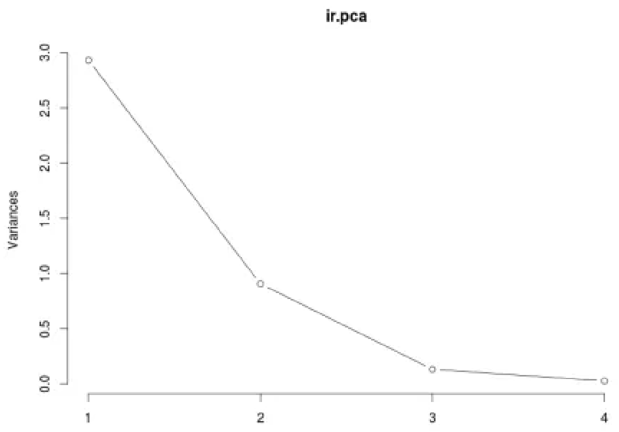
Analyzing the results

The `prcomp` function returns an object of class `prcomp`, which have some methods available. The `print` method returns the standard 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 0.1656840

Rotation:
                    PC1         PC2        PC
Sepal.Length  0.5038236 -0.45499872  0.708854
Sepal.Width  -0.3023682 -0.88914419 -0.331162
Petal.Length  0.5767881 -0.03378802 -0.219279
Petal.Width   0.5674952 -0.03545628 -0.582900
```

The `plot` method returns a plot of the variances (y-axis) associated with the PCs (x-axis). The Figure below is useful to decide how many PCs to retain for further analysis. In this simple case with only 4 PCs this is not a hard task and we can see that the first two PCs explain most of the variability in the data.

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



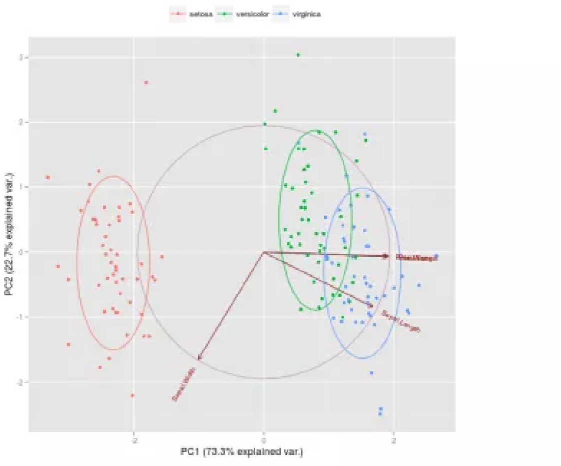
The `summary` method describe the importance of the PCs. The first row describe again the standard deviation associated with each PC. The second row shows the proportion of the variance in the data explained by each component while the third row describe the cumulative proportion of explained variance. 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     PC3
Standard deviation     1.7125 0.9524 0.36470 0
Proportion of Variance 0.7331 0.2268 0.03325 0
Cumulative Proportion  0.7331 0.9599 0.99314 1
```

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         P
149 1.0809930 -1.01155751 -0.7082289 -0.068110
150 0.9712116 -0.06158655 -0.5008674 -0.124115
```

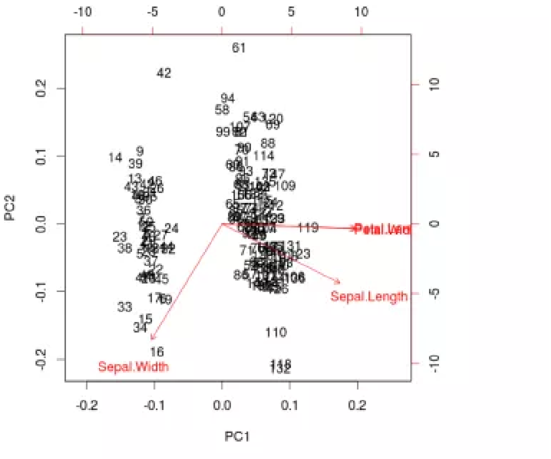
The Figure below is a biplot generated by the function `ggbiplot` of the `ggbiplot` package available on github.



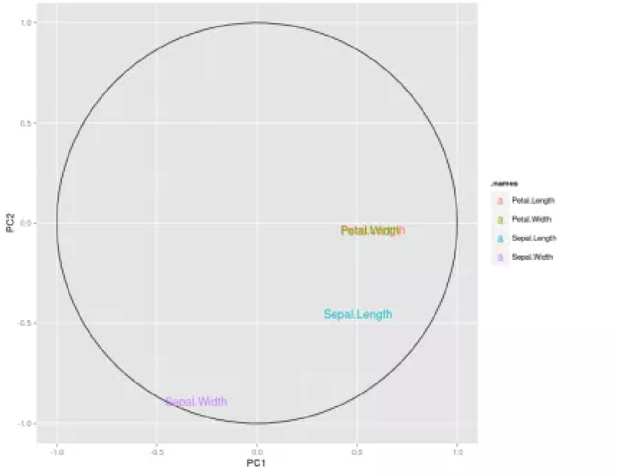
The code to generate this Figure is given by

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

library(ggbiplot)
g <- ggbiplot(ir.pca, obs.scale = 1, var.scal
              groups = ir.species, ellipse =
              circle = TRUE)
g <- g + scale_color_discrete(name = '')
g <- g + theme(legend.direction = 'horizontal
               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 68%) 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](https://gist.github.com/thigm85/7689076).



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", "center"
                            "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
3 -2.461341  0.3463921

# 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
```

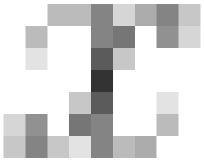
See Unsupervised data pre-processing for predictive modeling for an introduction of the `preProcess` function.

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 transformations. Journal of the Royal Statistical Society. Series B (Methodological) 211-252

Related

Comments: 9
A brief introduction to "apply" in R
In "R bloggers"



Computing and visualizing LDA in R
In "R bloggers"

The reshape function
In "R bloggers"

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 on topics such as: Data science, Big Data, R jobs, visualization (ggplot2, Boxplots, maps, animation), programming (RStudio, Sweave, LaTeX, SQL, Eclipse, git, hadoop, Web Scraping) statistics (regression, PCA, time series, trading) and more...

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

Comments are closed.

Search & Hit Enter

Recent popular posts

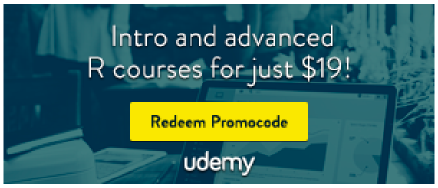
- Build your own neural network classifier in R
- What has Kaggle learned from 2 million machine learning models?
- Mastering R plot – Part 2: Axis
- Hadley Wickham's Advanced R in Amsterdam

Most visited articles of the week

1. Installing R packages
2. Scatterplots
3. Using apply, sapply, lapply in R
4. How to Learn R
5. In-depth introduction to machine learning in 15 hours of expert videos

6. How to Make a Histogram with Basic R
7. 7 Ways to Perplex a Data Scientist
8. Computing and visualizing PCA in R
9. Basics of Histograms

Sponsors


Intro and advanced
R courses for just $19!
Redeem Promocode
udemy

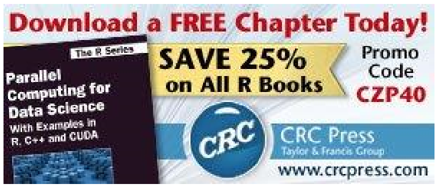
Highland Statistics Ltd
Zero Inflated Models & GLMM
Beginner's Guide to GAM
Beginner's Guide to GLM & GLMM
Beginner's Guide to GAMM

Werden Sie zum Expe[R]ten mit der
R-Akademie von
eoda
daten▸wissen▸nutzen
Beratung I Software
Training I Lösungen

STATISTICS
VIEWS
Bringing Statistics Together

BECOME A DATA SCIENTIST
NYC DATA SCIENCE ACADEMY
12 Weeks to A New Career

#ODSC
BOSTON MAY 20-22
TICKETS ON SALE NOW


Download a FREE Chapter Today!
The R Series
Parallel Computing for Data Science
With Examples in R, C++ and CUDA
SAVE 25%
on All R Books
Promo Code
CZP40
CRC
CRC Press
Taylor & Francis Group
www.crcpress.com

STATWORX
Consulting
Schulung
Data Mining
Mehr erfahren

Try the FASTEST ML
for R
Click for a Free Trial
YOTTAMINE
ANALYTICS

Jobs for R users

- Junior Data Scientist @ Farnham, England
- Vehicle Valuations Manager @ Farnham, England
- Manager – Quantitative Analytics @ London, United Kingdom
- Pharmacometrics Bootcamp @ Wellesley, Massachusetts, US
- Data Scientist for TIBCO (>$100K/year)
- Senior Financial Analyst – Consumer Marketing Analytics @ Seattle, US
- Post-Doctoral Researcher, @ Shanghai, China

Search & Hit Enter

Full list of contributing R-bloggers

R-bloggers was founded by Tal Galili, with gratitude to the R community.
Is powered by WordPress using a bavotasan.com design.