prVis --
A Novel Method for Visual Dimension Reduction

Wenxuan Zhao
Norman Matloff
Tiffany Jiang
Robert Tucker
University of California, Davis

Overview

- Motivation
- The Big Picture
- Features
    - Data Preprocessing
    - Processing
    - Result Processing
    - Producing Output
- Helper Functions

Motivation

Goals:

- Discover unknown patterns (Swiss Roll)
- Separation between known components (Spam Dataset)

Partial list of methods:
PCA, t-sne, UMAP, etc

Nice overview paper:
Ten quick tips for effective dimensionality reduction. PLoS Comput Biol 15(6): e1006907.

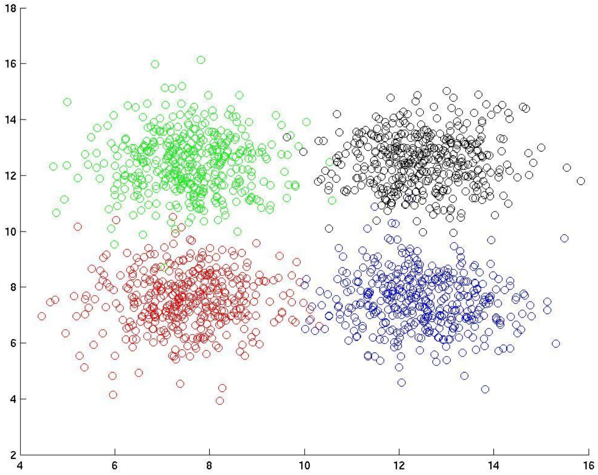
Authors: Lan Huong Nguyen, Susan Holmes
https://doi.org/ 10.1371/journal.pcbi.1006907

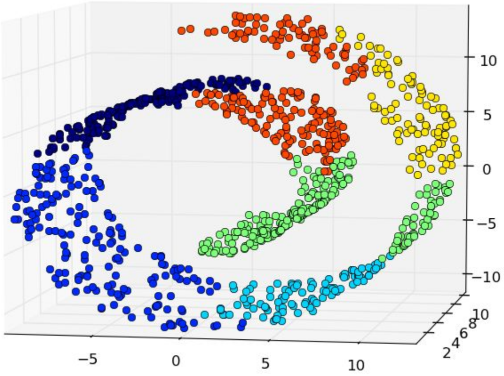
Motivation

Swiss roll:

The dataset was created to test various dimensionality reduction algorithms.

The idea was to create several points in **2d**, and then map them to **3d** with some smooth function, and then to see what the algorithm would find when it mapped the points back to **2d**.



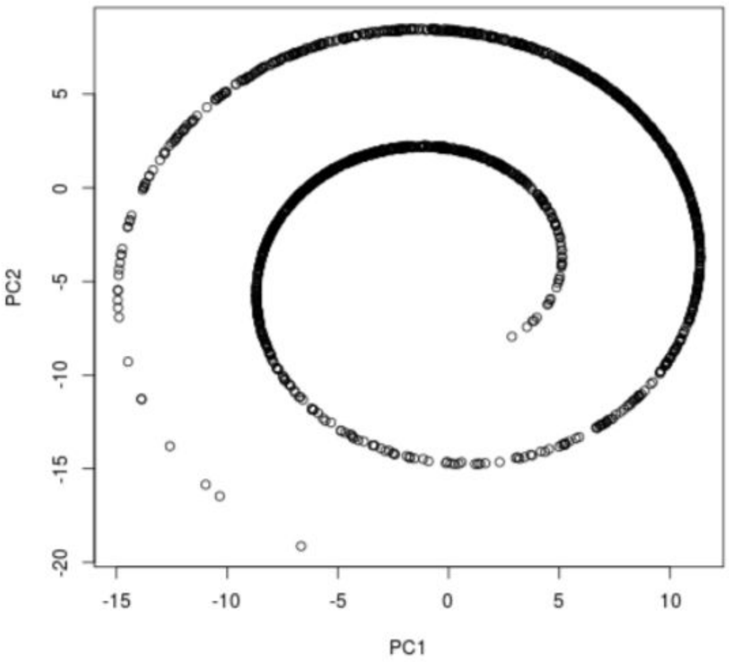


Citation: Dinoj Surendran, 16 May 2004

Motivation

How many components?

Plain PCA:

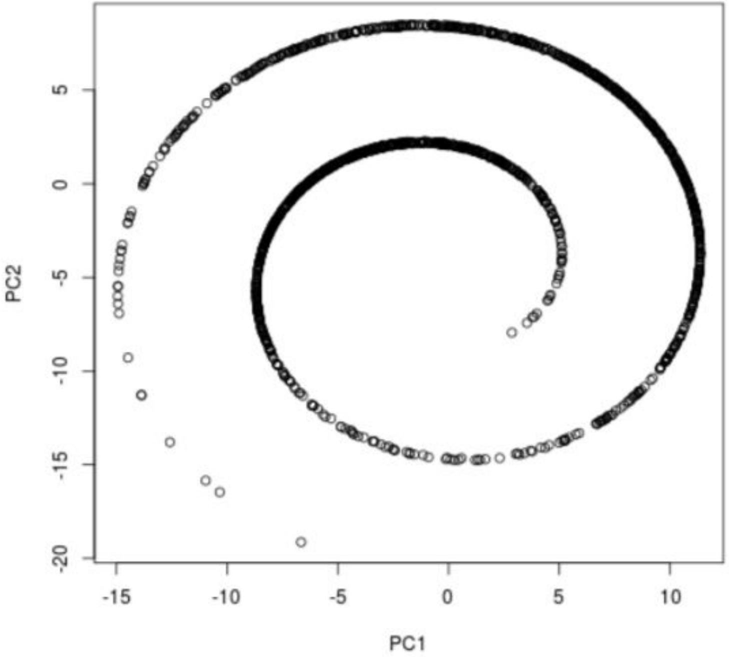


Motivation

How many components?

1?

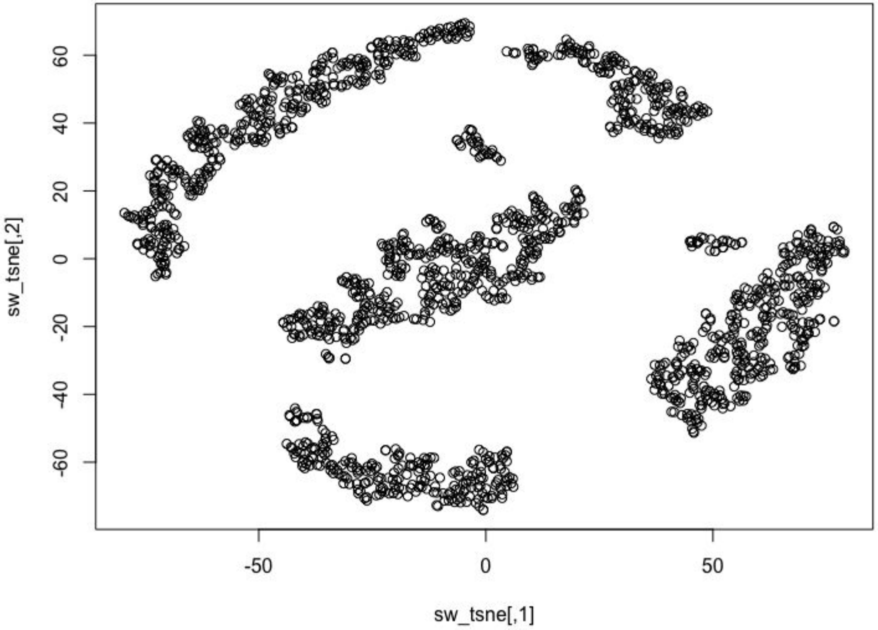
Plain PCA:



Motivation

How many components?

t-sne:

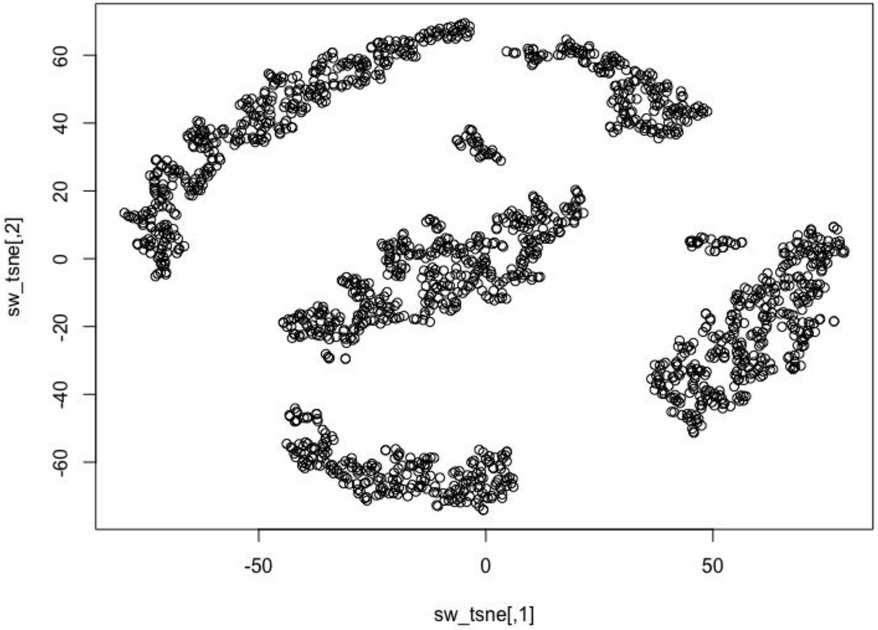


Motivation

How many components?

5?
7?

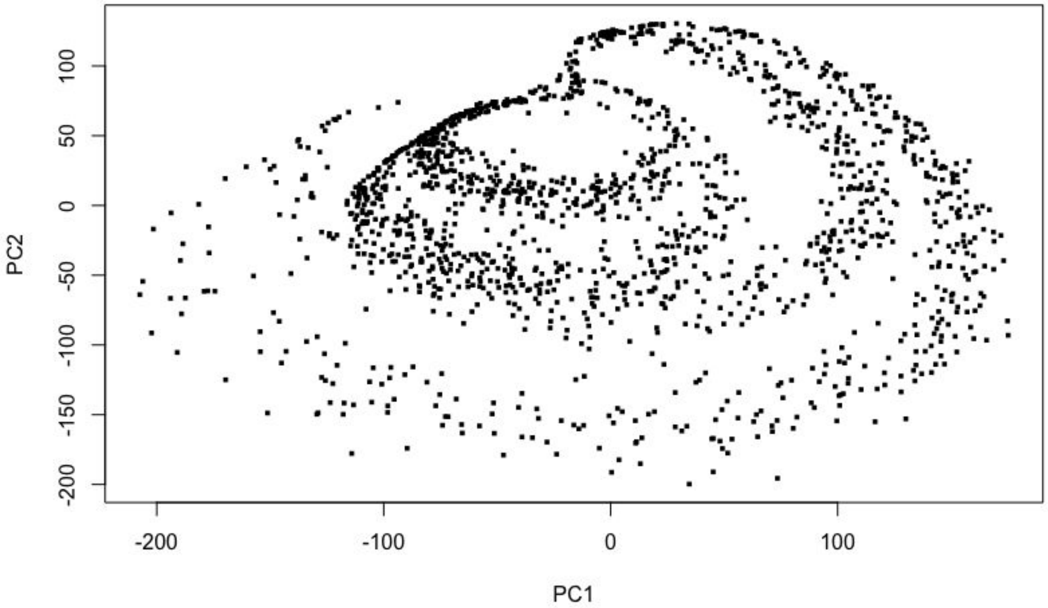
t-sne:



Motivation

prVis:

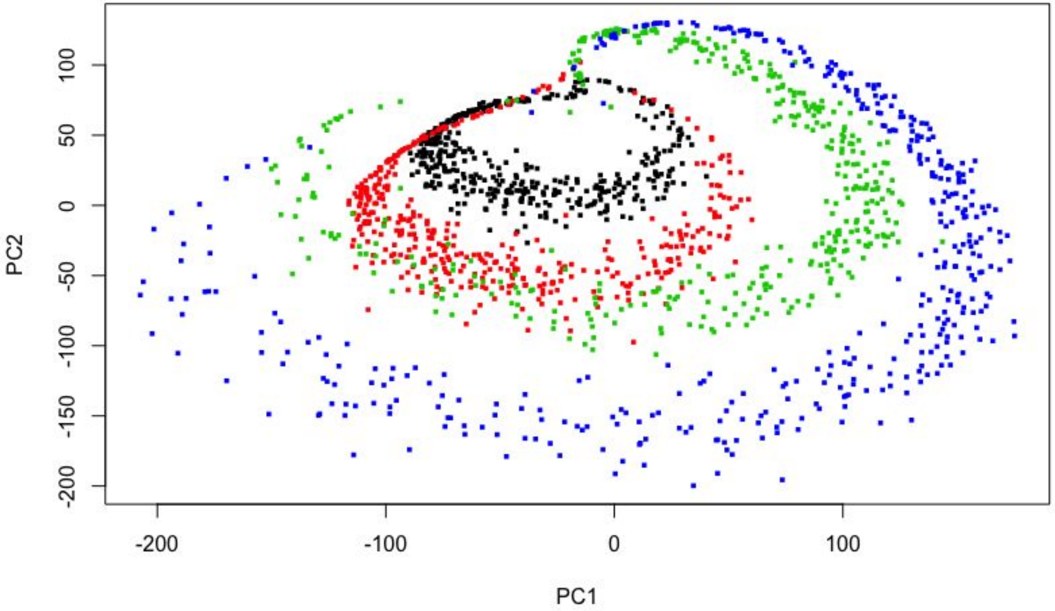
How many components?



Motivation

4!

prVis:



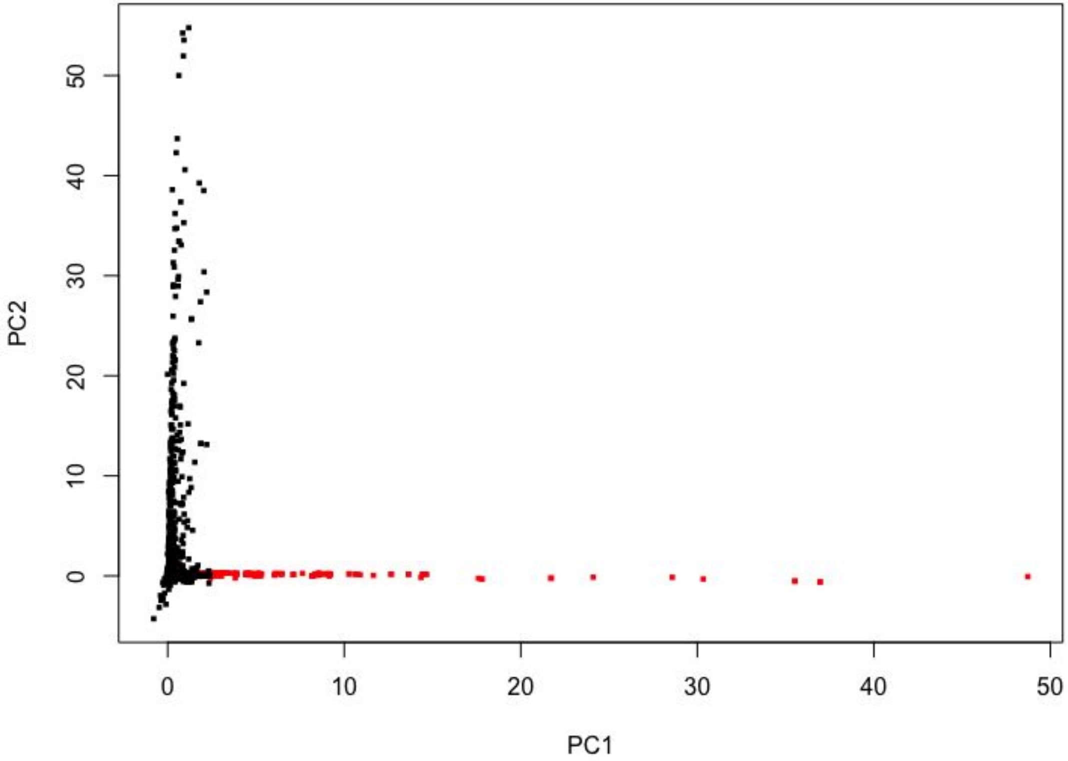
Motivation

Spam: A Kernlab built-in data set which has 57 predictors that predict whether a E-mail is spam

Motivation

Spam: A Kernlab built-in data set which has 57 predictors that predict whether a E-mail is spam

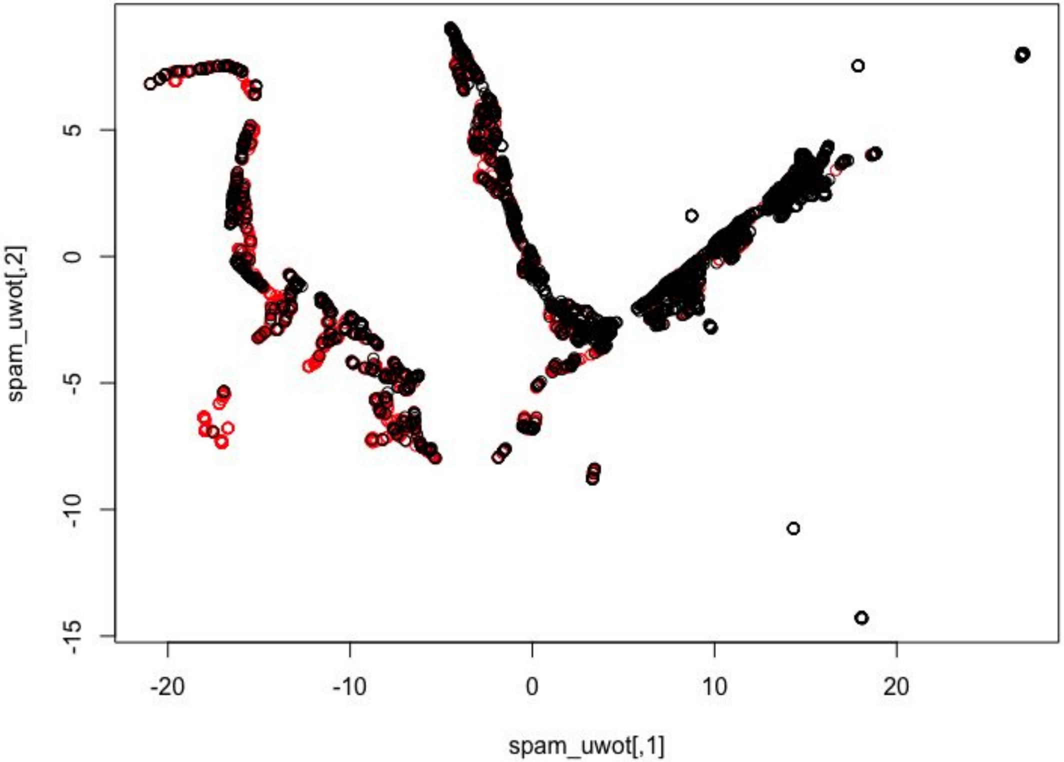
prVis:



Motivation

Spam: A Kernlab built-in data set which has 57 predictors that predict whether a E-mail is spam

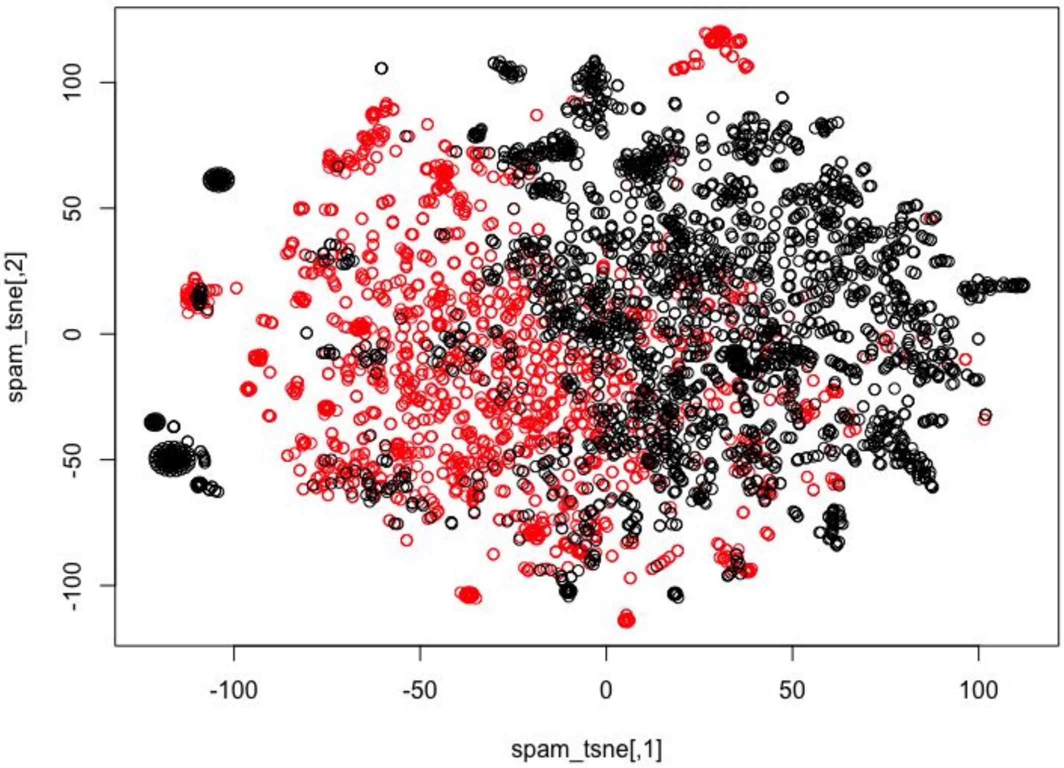
UMAP:



Motivation

Spam: A Kernlab built-in data set which has 57 predictors that predict whether a E-mail is spam

t-sne



Many More Examples

Our github page:

https://github.com/matloff/prVis

Please refer to our gallery in the link below for dozen more examples:

https://github.com/matloff/prVis/tree/master/inst/gallery

The Big Picture

- **P**olynomial **E**xpansion + **PCA**
- **G**nanadesikan and **W**ilk, **1969**
- Captures the **non-linearity**
- **Simple** but **Powerful**

Features

- Grouped features based on their functionality
- Based on needs we have in real life

Data Preprocessing Stage

Data Preprocessing

Processing Stage

Result Processing

Producing Output

Features: sub-sampling

Issue with **large** data sets:

- Dense Plot
- Time Consuming
- Space Consuming

nSubSam: option to subsample the data by specifying the number of rows we want.

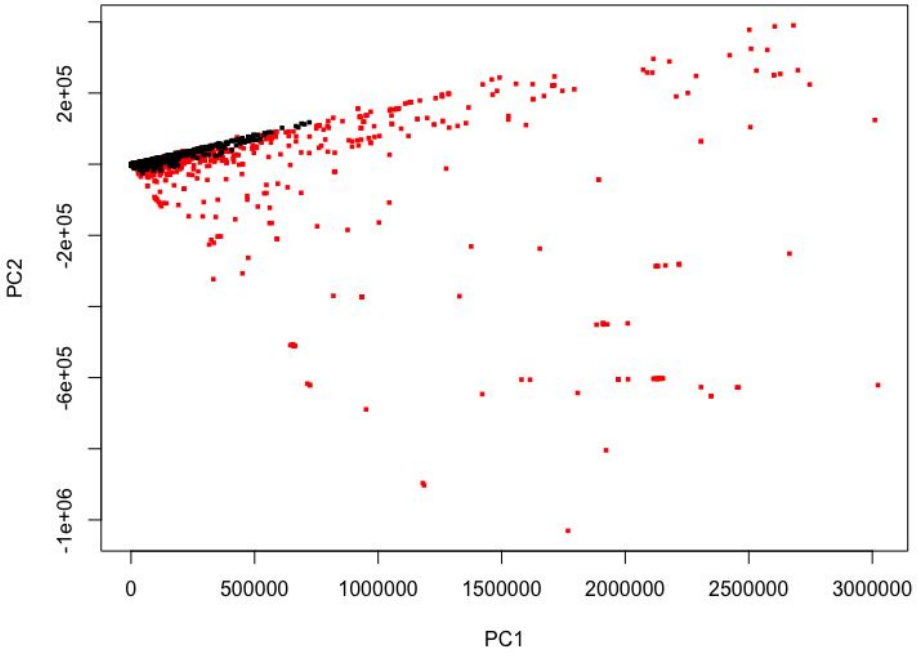
Features: nInterval

nInterval: partitions one of the continuous variables into n intervals, each of them corresponds to one label (one color).

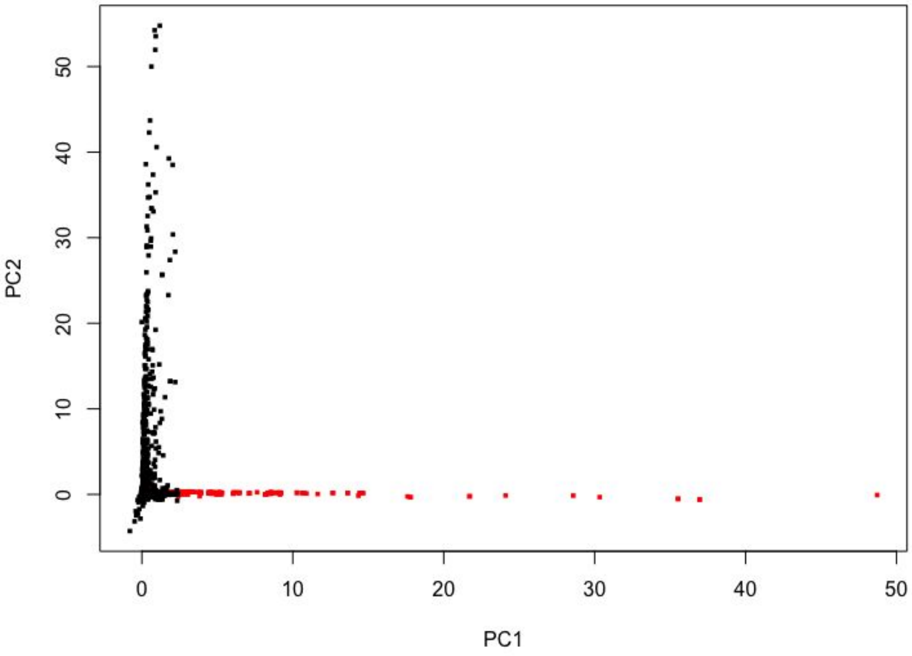
Features: scale

scale: Transforming the data to comparable scales

No Scale



Scaled



Processing Stage

Data Preprocessing

Processing Stage

Result Processing

Producing Output

Features: degree & maxInteractDeg

- Options for the subroutine **getPoly** powered by **polyreg**
- **degree**: specifies the highest degree for polynomial terms
- **maxInteractDeg**: specifies the highest degree for interaction terms

Features: handling large dataset

- Powered by package **bigstatsr**
  - Uses memory-mapping
  - Provides PCA for large matrices

- Enable users to handle “big” data set:
  - Data set with many columns
  - User specifies high **degree** & **maxInteractDeg**
  - Or both

Features: pca methods

- prcomp
- **RSpectra**
- By benchmarking the two implementations of PCA on various data sets, we have gained about 4-5 times speedup on average when using **RSpectra.**

Result Processing

Data Preprocessing

Processing Stage

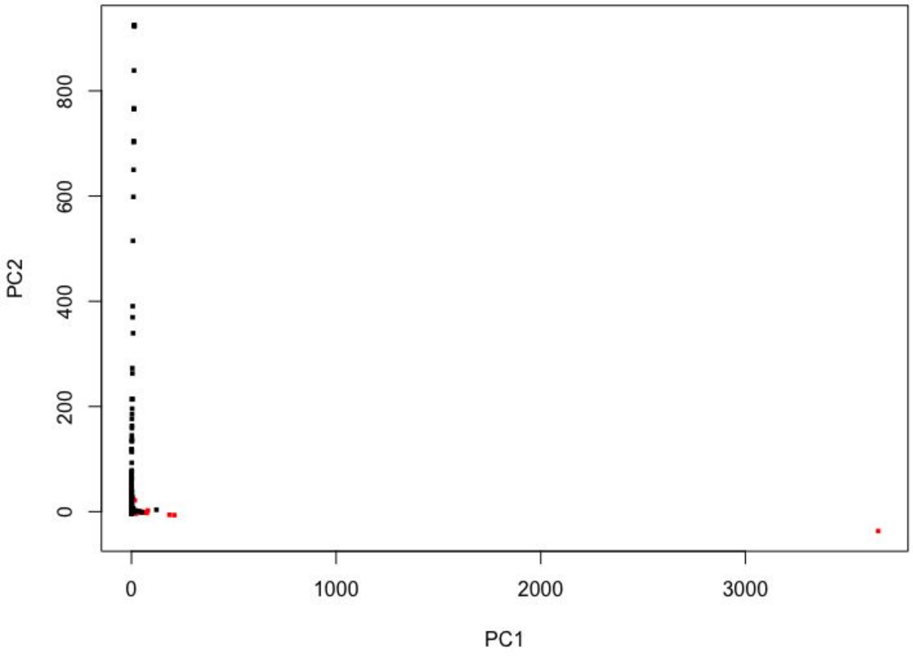
Result Processing

Producing Output

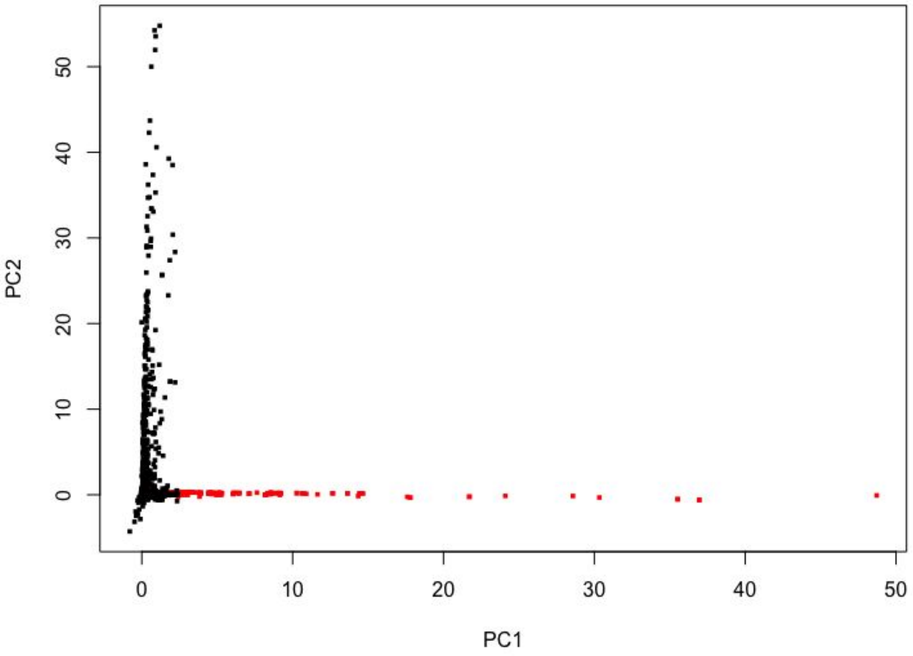
Features: outlier removal

- Outlier removal **by class** if any
- Uses **Mahalanobis Distance**

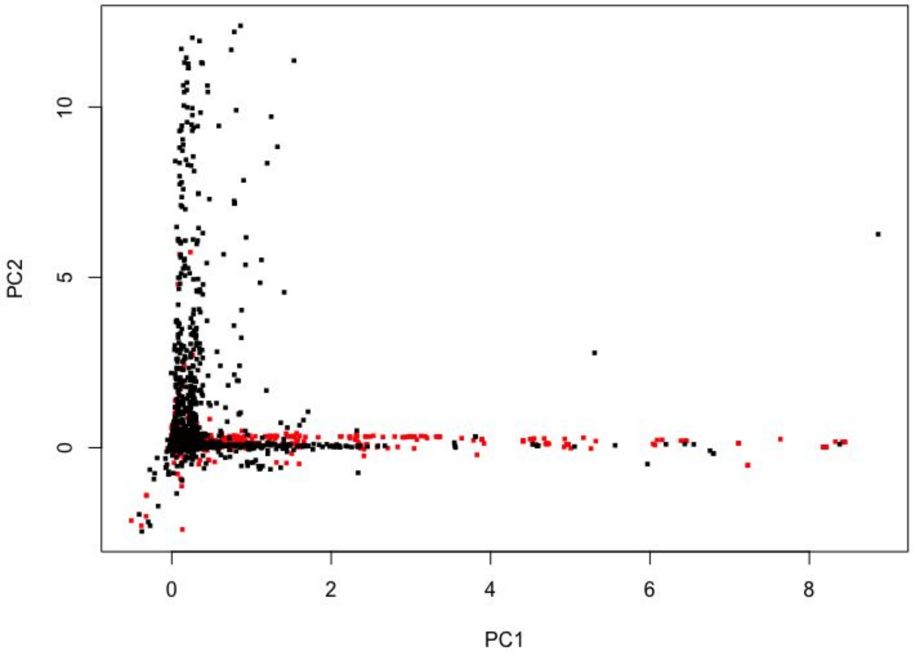
Removes no outlier



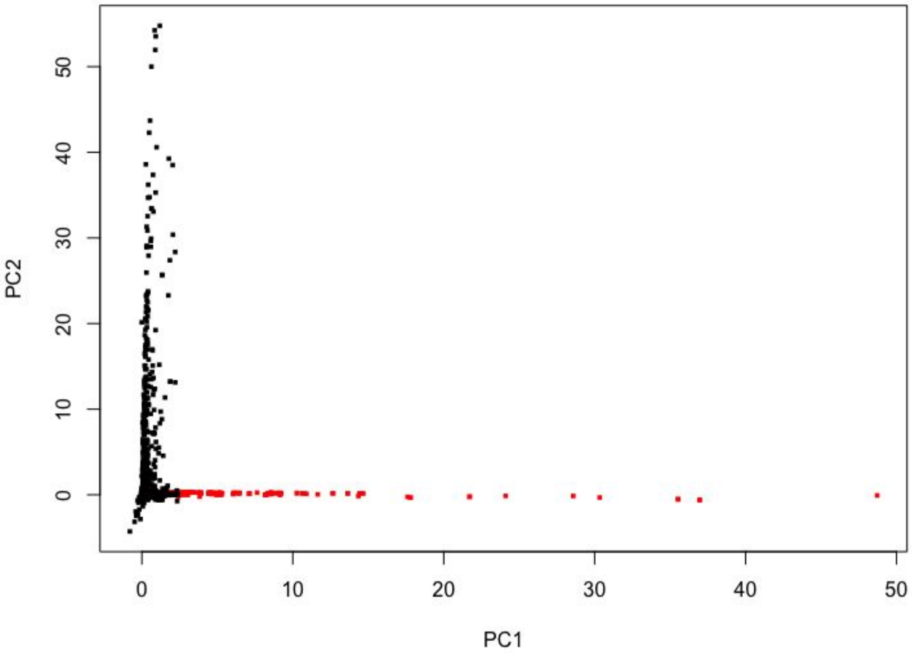
Removes 5% outliers



Removes outliers not by class



Removes outliers by class



Producing Output

Data Preprocessing

Processing Stage

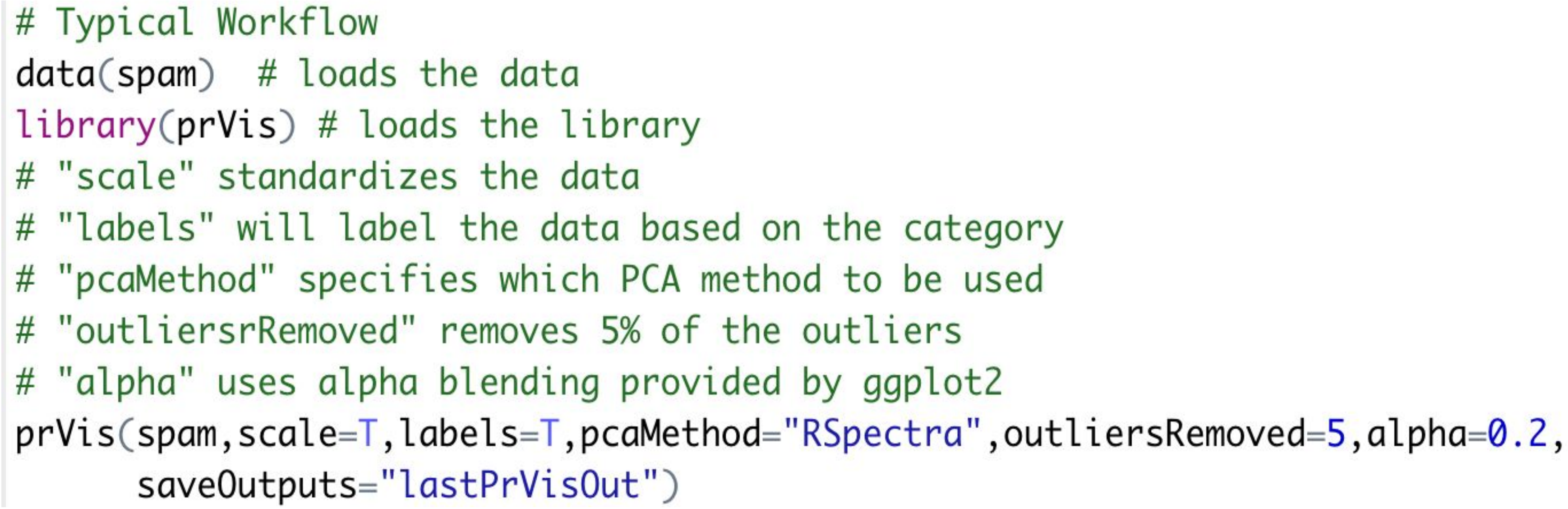
Result Processing

Producing Output

Features: saving outputs

- For future replication
- Can be passed as argument to helper functions
- By default automatically saves the latest prVis output

A typical screen

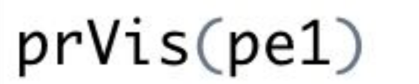


```
# Typical Workflow
data(spam)  # loads the data
library(prVis) # loads the library
# "scale" standardizes the data
# "labels" will label the data based on the category
# "pcaMethod" specifies which PCA method to be used
# "outliersrRemoved" removes 5% of the outliers
# "alpha" uses alpha blending provided by ggplot2
prVis(spam,scale=T,labels=T,pcaMethod="RSpectra",outliersRemoved=5,alpha=0.2,
      saveOutputs="lastPrVisOut")
```

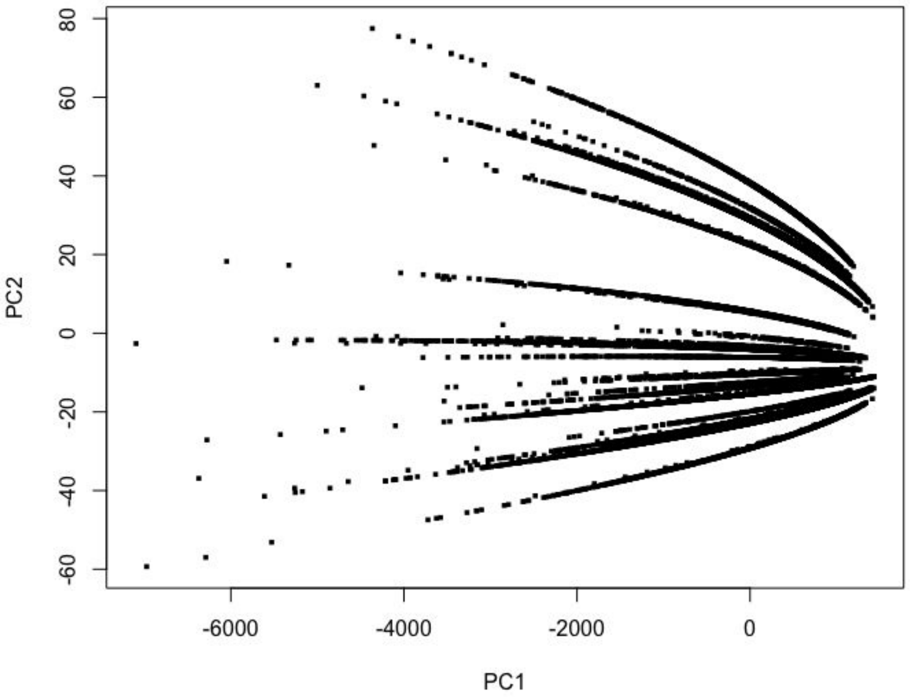
Helpers: colorCode

Programmer and Engineer dataset:

Records age, gener, occupation, education level, and salary information of the programmers and engineers in the bay area.

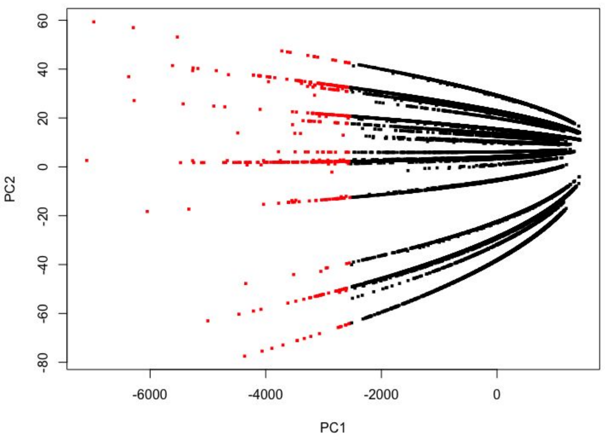


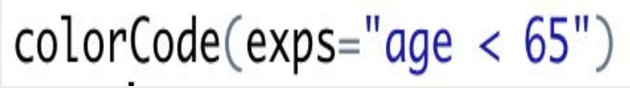
```
prVis(pe1)
```



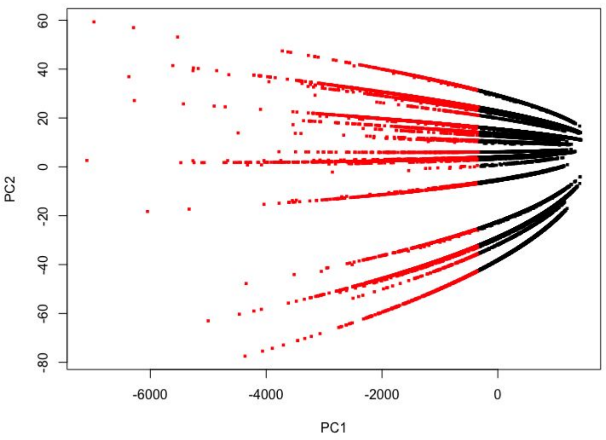
Helpers: colorCode

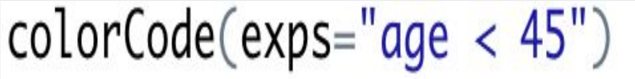
- Display color coding for user-specified expressions



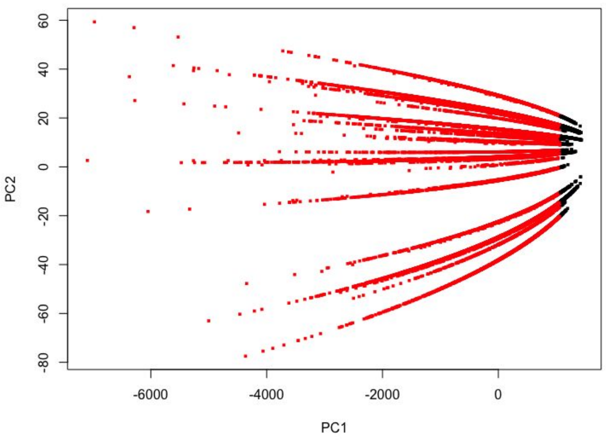


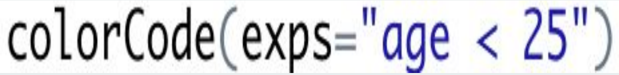
```
colorCode(exps="age < 65")
```





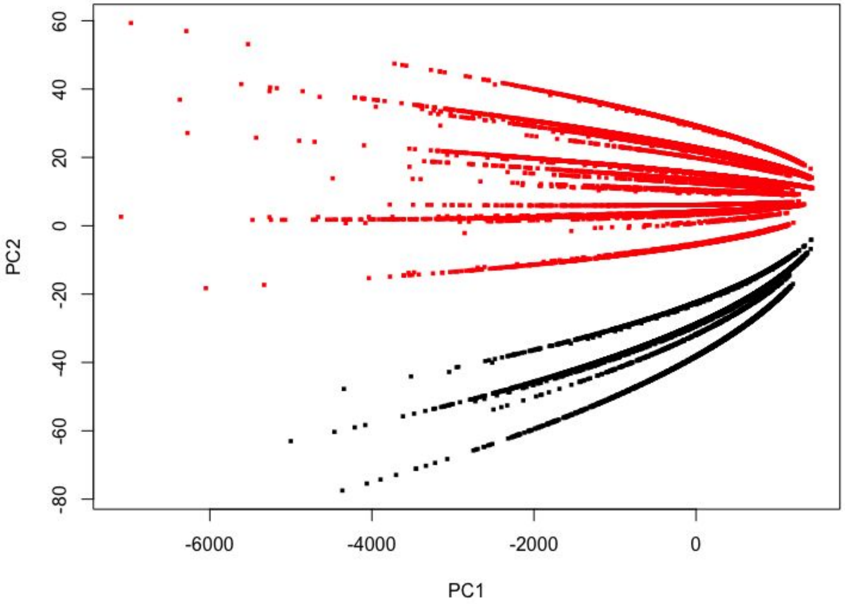
```
colorCode(exps="age < 45")
```

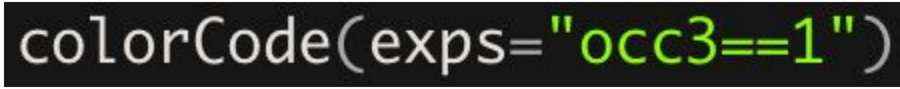




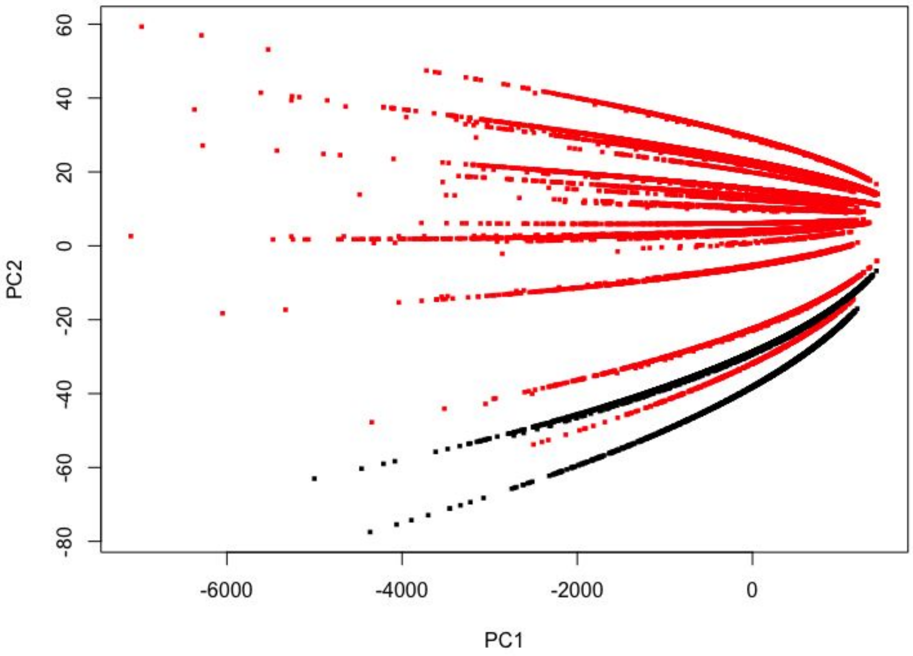
```
colorCode(exps="age < 25")
```

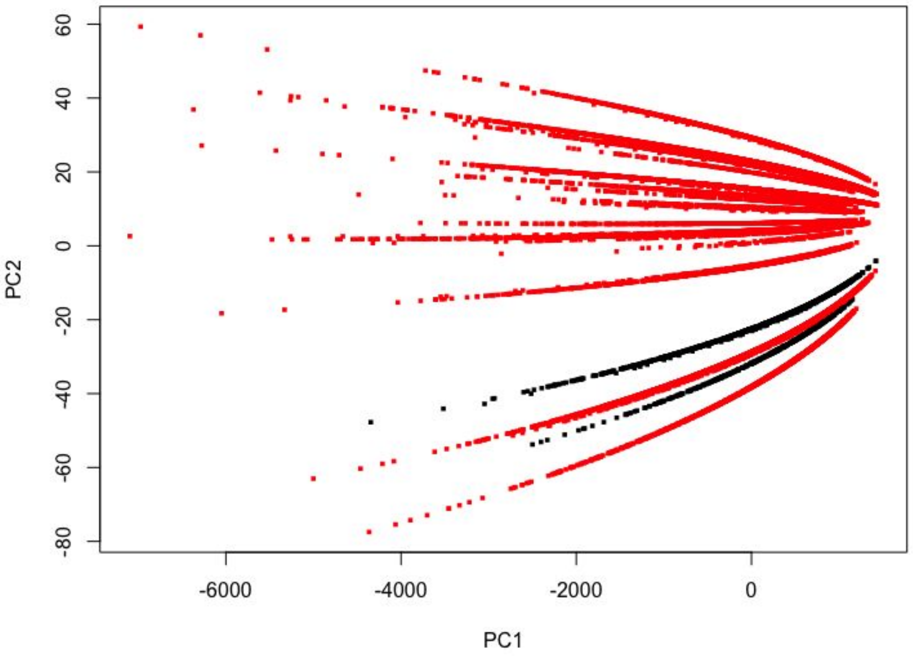
More complex expressions

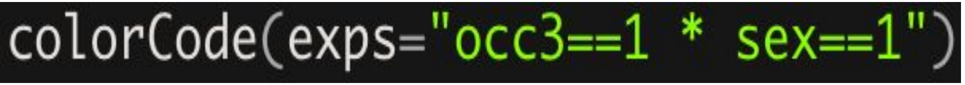




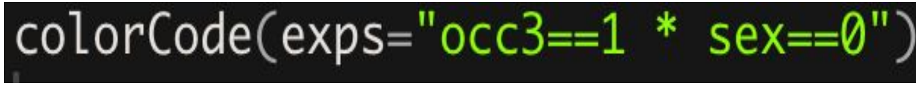
```
colorCode(exps="occ3==1")
```







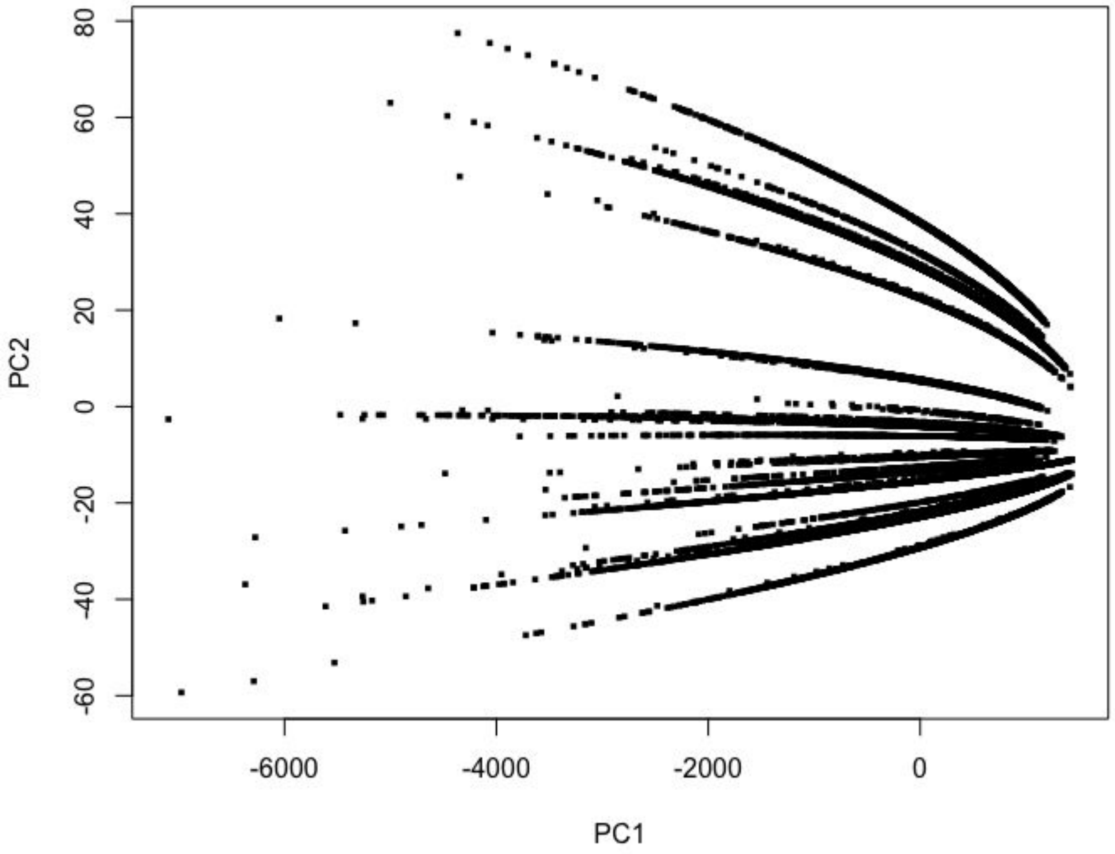
```
colorCode(exps="occ3==1 * sex==1")
```

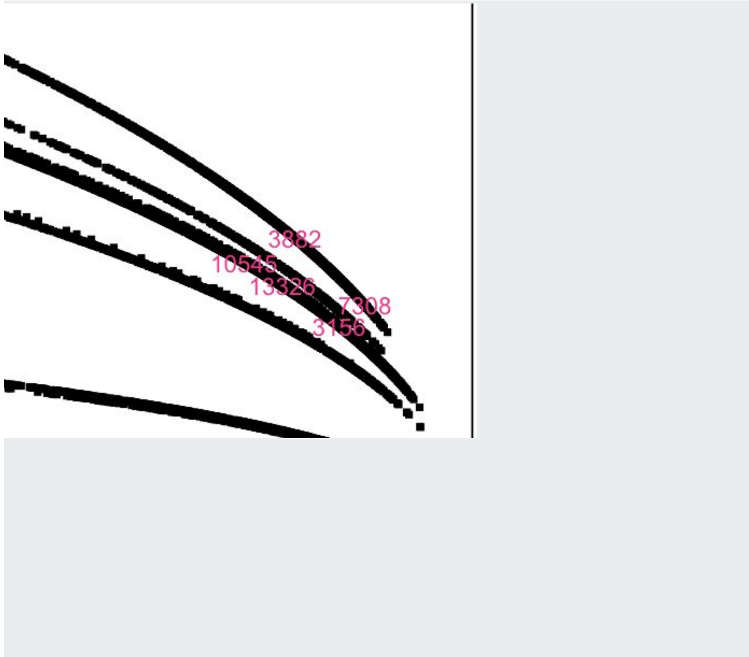


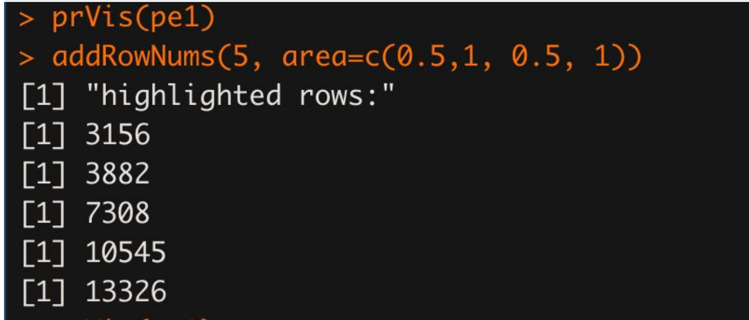
```
colorCode(exps="occ3==1 * sex==0")
```

Helpers: addRowNums

- Chooses **np** points at random from the prVis output, writing their **row numbers** on the plot
- User can specify a vector that has 4 numbers, corresponding to percentages of the graph from left to right and bottom to top.
    - e.g. c(0,1,0,1) specifies the entirety of the graph.
      c(0,0.5,0.5,1) specifies upper-left quadrant.







```
> prVis(pe1)
> addRowNums(5, area=c(0.5,1, 0.5, 1))
[1] "highlighted rows:"
[1] 3156
[1] 3882
[1] 7308
[1] 10545
[1] 13326
```

Thank you!

Merci beaucoup!

Our github page:

https://github.com/matloff/prVis

Questions?