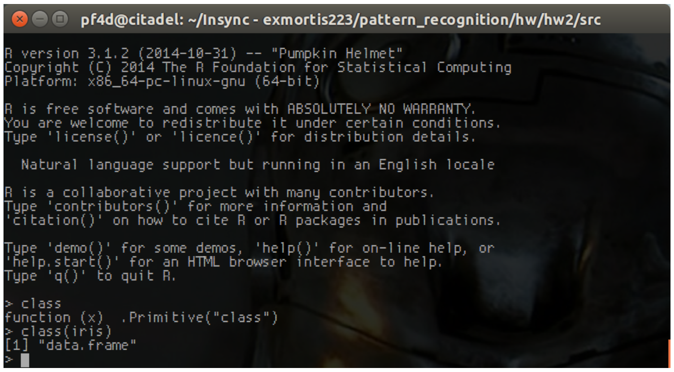
Principle component analysis

Evan Cummings
CSCI 548 – Douglas W. Raiford – Pattern Recognition

November 16, 2016

1 class command :

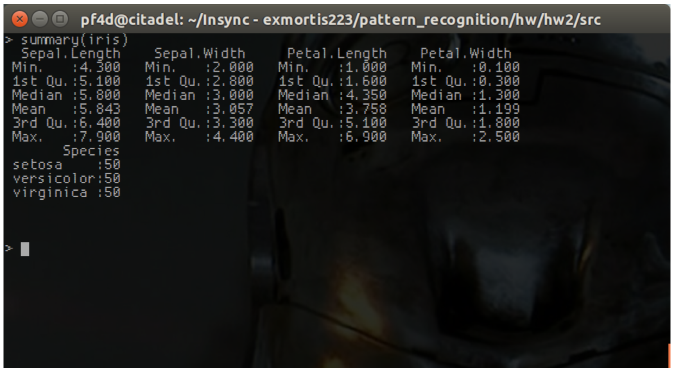
The class of the iris dataset is a data.frame, a “tightly coupled collections of variables which share many of the properties of matrices and of lists, used as the fundamental data structure by most of R’s modeling software.”


pf4d@citadel: ~/Insync - exmortis223/pattern_recognition/hw/hw2/src

```
R version 3.1.2 (2014-10-31) -- "Pumpkin Helmet"
Copyright (C) 2014 The R Foundation for Statistical Computing
Platform: x86_64-pc-linux-gnu (64-bit)

R is free software and comes with ABSOLUTELY NO WARRANTY.
You are welcome to redistribute it under certain conditions.
Type 'license()' or 'licence()' for distribution details.

  Natural language support but running in an English locale

R is a collaborative project with many contributors.
Type 'contributors()' for more information and
'citation()' on how to cite R or R packages in publications.

Type 'demo()' for some demos, 'help()' for on-line help, or
'help.start()' for an HTML browser interface to help.
Type 'q()' to quit R.

> class
function (x)  .Primitive("class")
> class(iris)
[1] "data.frame"
> 
```

2 summary command :

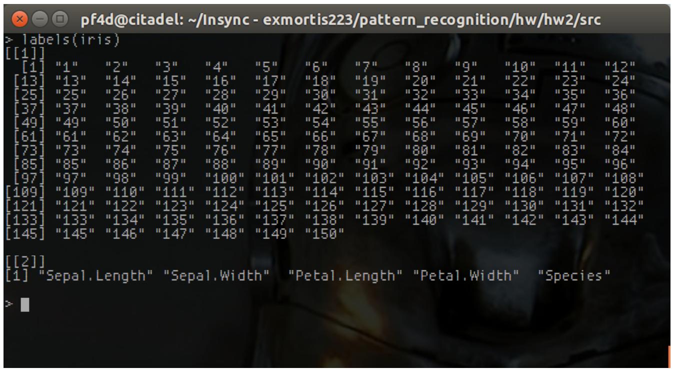
The summary command applied to the iris dataset, is a “generic function used to produce result summaries of the results of various model fitting functions. The function invokes particular methods which depend on the ‘class’ of the first argument.”


pf4d@citadel: ~/Insync - exmortis223/pattern_recognition/hw/hw2/src

```
> summary(iris)
  Sepal.Length    Sepal.Width     Petal.Length    Petal.Width   
 Min.   :4.300   Min.   :2.000   Min.   :1.000   Min.   :0.100  
 1st Qu.:5.100   1st Qu.:2.800   1st Qu.:1.600   1st Qu.:0.300  
 Median :5.800   Median :3.000   Median :4.350   Median :1.300  
 Mean   :5.843   Mean   :3.057   Mean   :3.758   Mean   :1.199  
 3rd Qu.:6.400   3rd Qu.:3.300   3rd Qu.:5.100   3rd Qu.:1.800  
 Max.   :7.900   Max.   :4.400   Max.   :6.900   Max.   :2.500  
       Species  
 setosa    :50  
 versicolor:50  
 virginica :50  



> 
```

3 labels command :

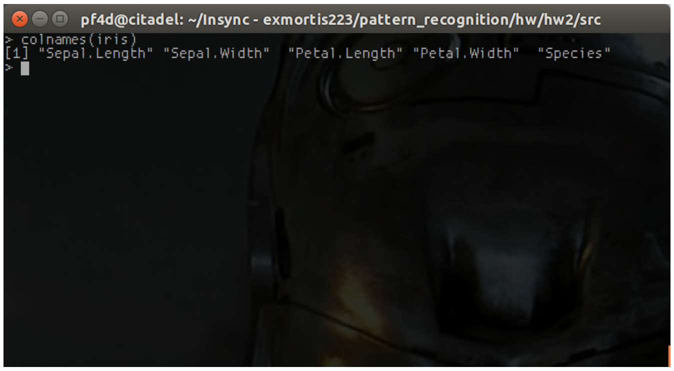
The labels command: “Find[s] a suitable set of labels from an object for use in printing or plotting. For example, a generic function.”


pf4d@citadel: ~/Insync - exmortis223/pattern_recognition/hw/hw2/src

```
> labels(iris)
[[1]]
  [1] "1"   "2"   "3"   "4"   "5"   "6"   "7"   "8"   "9"   "10"  "11"  "12" 
 [13] "13"  "14"  "15"  "16"  "17"  "18"  "19"  "20"  "21"  "22"  "23"  "24" 
 [25] "25"  "26"  "27"  "28"  "29"  "30"  "31"  "32"  "33"  "34"  "35"  "36" 
 [37] "37"  "38"  "39"  "40"  "41"  "42"  "43"  "44"  "45"  "46"  "47"  "48" 
 [49] "49"  "50"  "51"  "52"  "53"  "54"  "55"  "56"  "57"  "58"  "59"  "60" 
 [61] "61"  "62"  "63"  "64"  "65"  "66"  "67"  "68"  "69"  "70"  "71"  "72" 
 [73] "73"  "74"  "75"  "76"  "77"  "78"  "79"  "80"  "81"  "82"  "83"  "84" 
 [85] "85"  "86"  "87"  "88"  "89"  "90"  "91"  "92"  "93"  "94"  "95"  "96" 
 [97] "97"  "98"  "99"  "100" "101" "102" "103" "104" "105" "106" "107" "108"
[109] "109" "110" "111" "112" "113" "114" "115" "116" "117" "118" "119" "120"
[121] "121" "122" "123" "124" "125" "126" "127" "128" "129" "130" "131" "132"
[133] "133" "134" "135" "136" "137" "138" "139" "140" "141" "142" "143" "144"
[145] "145" "146" "147" "148" "149" "150"

[[2]]
[1] "Sepal.Length" "Sepal.Width"  "Petal.Length" "Petal.Width"  "Species"     

> 
```

4 colnames command :

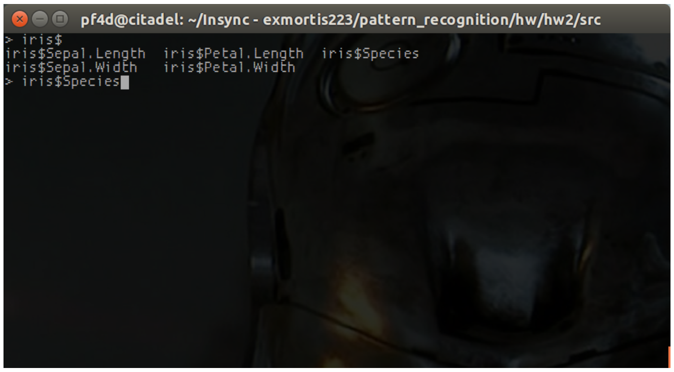
The colnames command: “Retrieve[s] or set[s] the row or column names of a matrix-like object.”


pf4d@citadel: ~/Insync - exmortis223/pattern_recognition/hw/hw2/src

```
> colnames(iris)
[1] "Sepal.Length" "Sepal.Width"  "Petal.Length" "Petal.Width"  "Species"     
> 
```

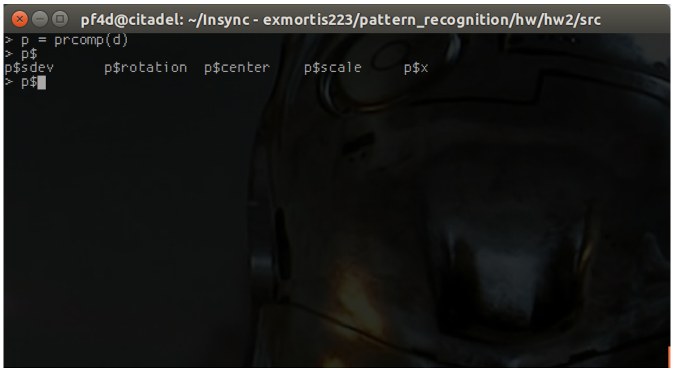
5 tab-completion :

Pressing tab after a data.frame object with accessor-operator $ will provide variable names.


pf4d@citadel: ~/Insync - exmortis223/pattern_recognition/hw/hw2/src

```
> iris$
iris$Sepal.Length  iris$Petal.Length  iris$Species
iris$Sepal.Width   iris$Petal.Width
> iris$Species
```

6 Iris data PCA plotting :


pf4d@citadel: ~/Insync - exmortis223/pattern_recognition/hw/hw2/src

```
> p = prcomp(d)
> p$
p$sdev       p$rotation  p$center    p$scale     p$x
> p$
```

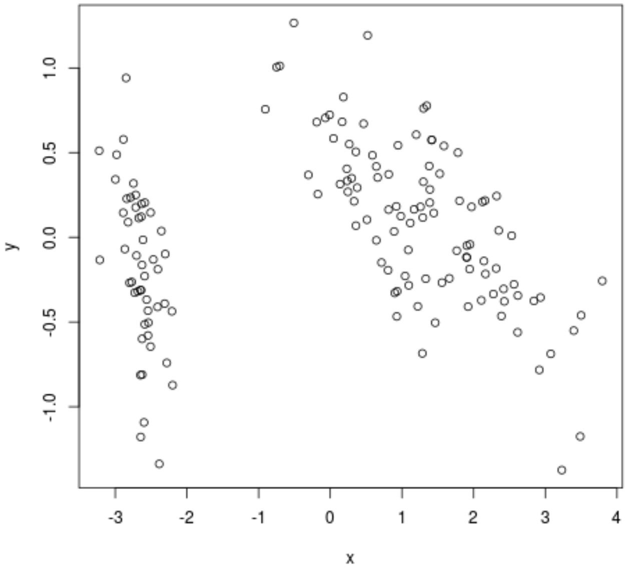
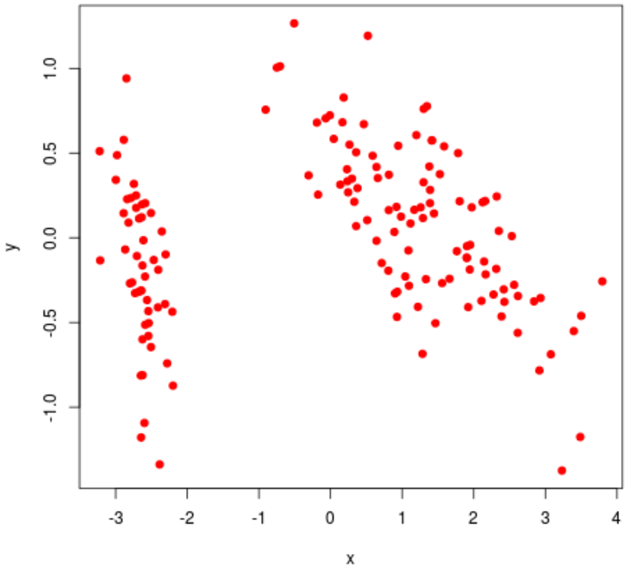
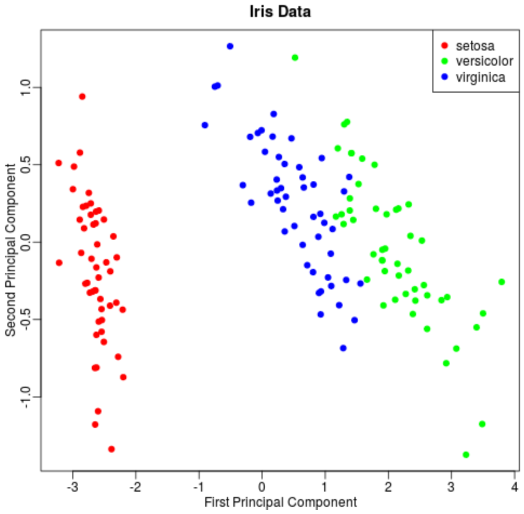


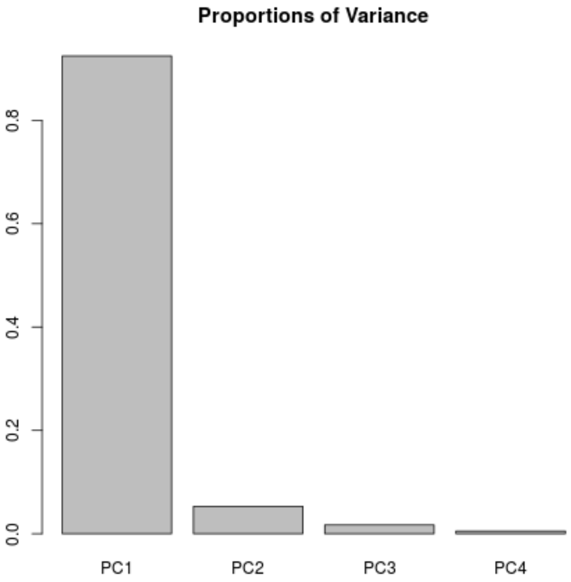
Figure 1: PCA-plot showing what appears to be two clusters, despite the fact that there are three distinct species of iris.

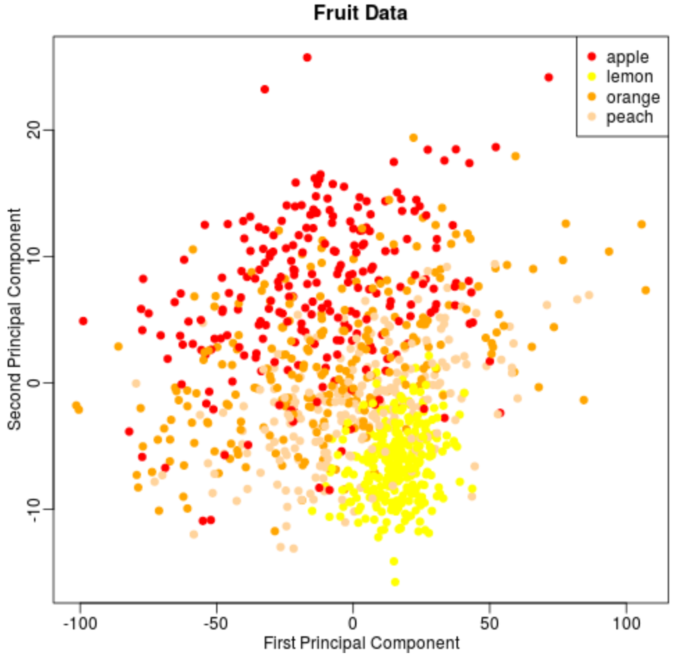
The plot command, a “Generic function for plotting of R objects. For more details about the graphical parameter arguments, see ‘par’.

For simple scatter plots, plot.default will be used. However, there are plot methods for many R objects, including function’s, data.frame’s, density objects, etc. Use methods(plot) and the documentation for these.”









6.1 Source code :

```
# store the iris ’classes’ :
c = iris[,5]

# store the iris ’data’ :
d = iris[,seq(1,4)]

# perform PCA on d :
p = prcomp(d)

# set the 1st principle component :
x = p$x[,1]

# set the 2nd principle component :
y = p$x[,2]

# plot them :
png(’../doc/images/pca_plot.png’)
plot(x,y)
dev.off()

# plot them in red :
png(’../doc/images/pca_plot_red.png’)
plot(x,y, col=’red’, bg=’red’, type=’p’, pch=21)
dev.off()

# store indexes of classes :
s  = which(c == ’setosa’)
vc = which(c == ’versicolor’)
vg = which(c == ’virginica’)

# color vector :
cc     = as.vector(c)
cc[s]  = ’red’
cc[vc] = ’blue’
cc[vg] = ’green’

# plot them colored by class :
png(’../doc/images/pca_plot_class.png’)
X11(width=4, height=4)
par(mar=c(2.5,2.5,2.5,.1),mgp = c(1.5, .5, 0))
plot(x,y, col=cc, bg=cc, type=’p’, pch=21, xlab=’First Principal Component’,
     ylab=’Second Principal Component’, main=’Iris Data’)
legend(’topright’, levels(c),
        col   = c(’red’, ’green’, ’blue’),
        pt.bg = c(’red’, ’green’, ’blue’), pch = 21)
dev.off()

# do a barplot of the proportions of variance :
sig  = p$sdev**2
pvar =  sig / sum(sig)
png(’../doc/images/pca_pvar_barplot.png’)
barplot(pvar, names.arg=c(’PC1’,’PC2’,’PC3’,’PC4’),
         main=’Proportions of Variance’)
dev.off()
```

7 Fruit data :

While the iris data appears to be highly grouped, the fruit data appears much less so. However, the lemons seem to be clustered distintly from the other fruit. Therefore, I believe that these data *do* show enough structure for machine learning techniques; with apples, oranges, and peaches potentially presenting the highest challenge.

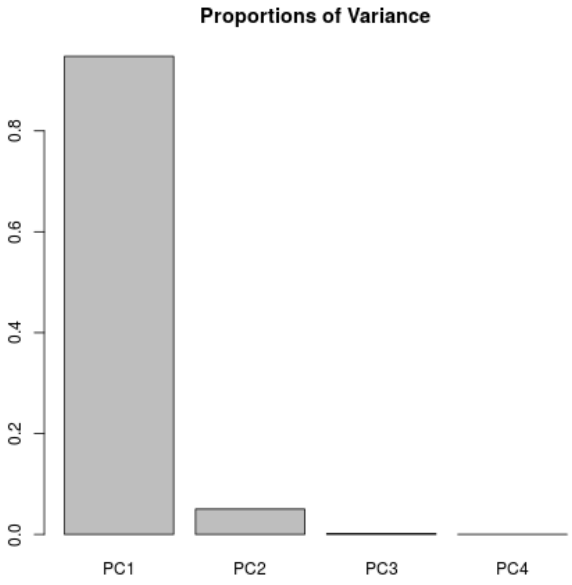


Figure 2: The proportions of the variance in the fruit data. The principle component captures ≈ 94% of the variation.

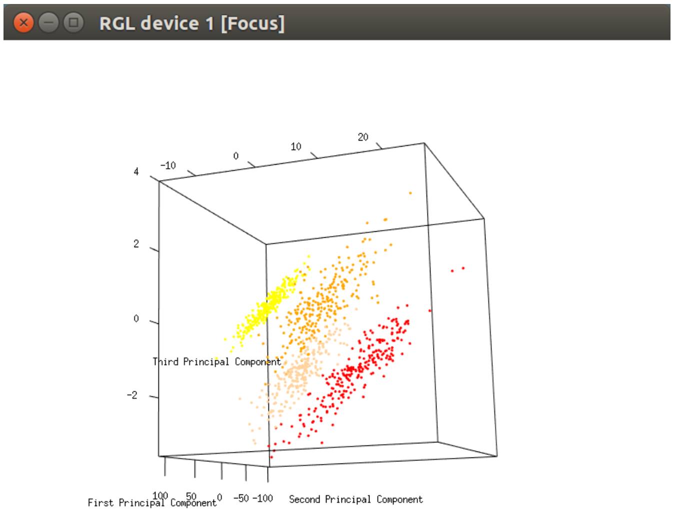

RGL device 1 [Focus]
-10
0
10
20
4
2
0
-2
Third Principal Component
100
50
0
-50
-100
First Principal Component
Second Principal Component

Figure 3: In 3D, we can see much more distinct clustering.

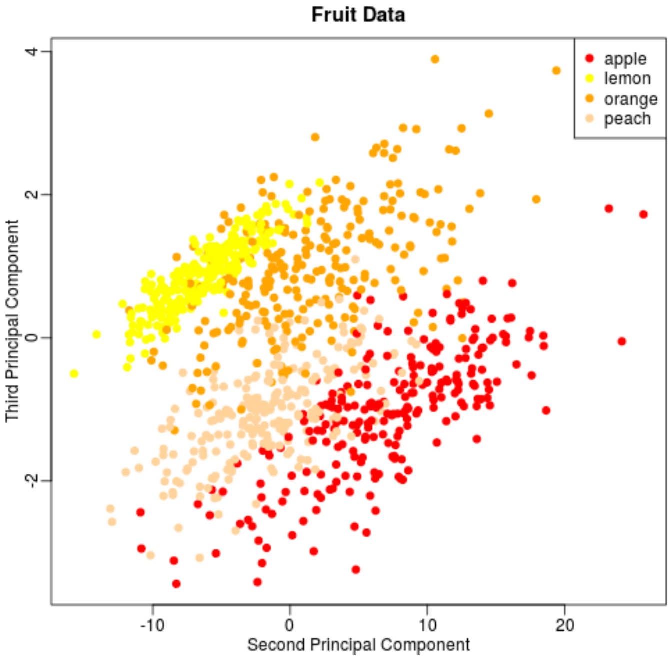


Figure 4: The second and third principle components, highlighting the clustering nature of the fruit data.

7.1 Source code :

```
# read the variable back in :
f = read.csv("../data/fruit.csv")

# store the fruit 'classes' :
c = f[,5]

# store the fruit 'data' :
d = f[,seq(1,4)]

# perform PCA on d :
p = prcomp(d)

# set the 1st principle component :
x = p$x[,1]

# set the 2nd principle component :
y = p$x[,2]

# set the 3nd principle component :
z = p$x[,3]

# store indexes of classes :
a = which(c == 'apple')
l = which(c == 'lemon')
o = which(c == 'orange')
z = which(c == 'peach')

# color vector :
cols  = c('red', 'yellow', 'orange', 'burlywood1')
cc    = as.vector(c)
cc[a] = cols[1]
cc[l] = cols[2]
cc[o] = cols[3]
cc[z] = cols[4]

# plot them colored by class :
png('../doc/images/pca_plot_fruit.png')
#X11(width=4, height=4)
par(mar=c(2.5,2.5,2.5,.1),mgp = c(1.5, .5, 0))
plot(x,y, col=cc, bg=cc, type='p', pch=21, xlab='First Principal Component',
     ylab='Second Principal Component', main='Fruit Data')
legend('topright', levels(c), col = cols, pt.bg = cols, pch = 21)
dev.off()

# do a barplot of the proportions of variance :
sig  = p$sdev**2
pvar =  sig / sum(sig)
png('../doc/images/pca_pvar_fruit_barplot.png')
barplot(pvar, names.arg=c('PC1','PC2','PC3','PC4'),
        main='Proportions of Variance')
dev.off()

# import the rgl library for 3D stuff:
library(rgl)

plot3d(x,y,z, col=cc, bg=cc, type='p', pch=21,
       xlab='First Principal Component',
       ylab='Second Principal Component',
       zlab='Third Principal Component')

# plot them colored by class :
png('../doc/images/pca_plot_fruit_23.png')
#X11(width=4, height=4)
par(mar=c(2.5,2.5,2.5,.1),mgp = c(1.5, .5, 0))
plot(y,z, col=cc, bg=cc, type='p', pch=21, xlab='Second Principal Component',
     ylab='Third Principal Component', main='Fruit Data')
legend('topright', levels(c), col = cols, pt.bg = cols, pch = 21)
dev.off()
```