Contents

1 01-pca-iris.R

```
### include
library(pls)
```

```
## Attaching package: 'pls'
```

```
## The following object(s) are masked from 'package:stats':
##
## loadings
```

```
library(reshape)
```

```
## Loading required package: plyr
```

```
## Attaching package: 'reshape'
```

```
## The following object(s) are masked from 'package:plyr':
##
## rename, round_any
```

```
library(ggplot2)
```

```
### data
data(iris)
```

```
str(iris)
```

```
## 'data.frame':    150 obs. of  5 variables:
##  $ Sepal.Length: num  5.1 4.9 4.7 4.6 5 5.4 4.6 5 4.4 4.9 ...
##  $ Sepal.Width : num  3.5 3 3.2 3.1 3.6 3.9 3.4 3.4 2.9 3.1 ...
##  $ Petal.Length: num  1.4 1.4 1.3 1.5 1.4 1.7 1.4 1.5 1.4 1.5 ...
##  $ Petal.Width : num  0.2 0.2 0.2 0.2 0.2 0.4 0.3 0.2 0.2 0.1 ...
##  $ Species     : Factor w/ 3 levels "setosa","versicolor",..: 1 1 1 1 1 1 1 1 1 1 ...
```

```r
table(iris$Species)
```

```
##
##     setosa versicolor  virginica
##         50         50         50
```

```r
### plot iris data
# credits: http://blog.echen.me/2012/01/17/quick-introduction-to-ggplot2/
qplot(Sepal.Length, Petal.Length, data = iris)
```

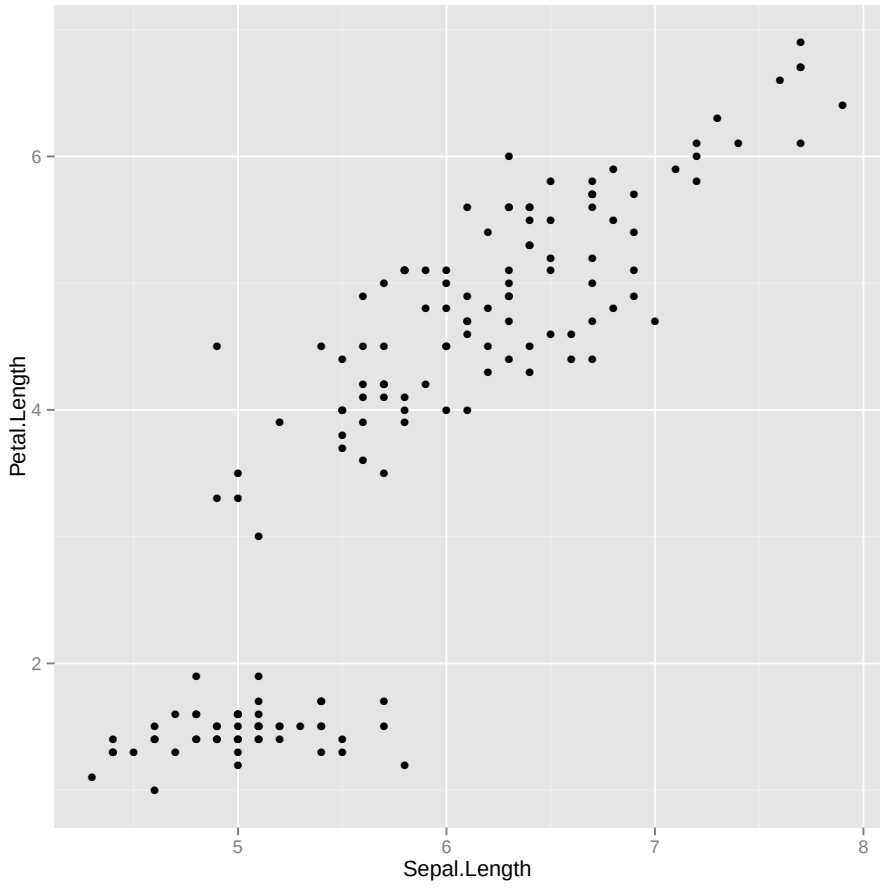

Figure 1: plot of chunk 1-pca-iris.R

```r
# add class labels with colors
qplot(Sepal.Length, Petal.Length, data = iris, color = Species)
```

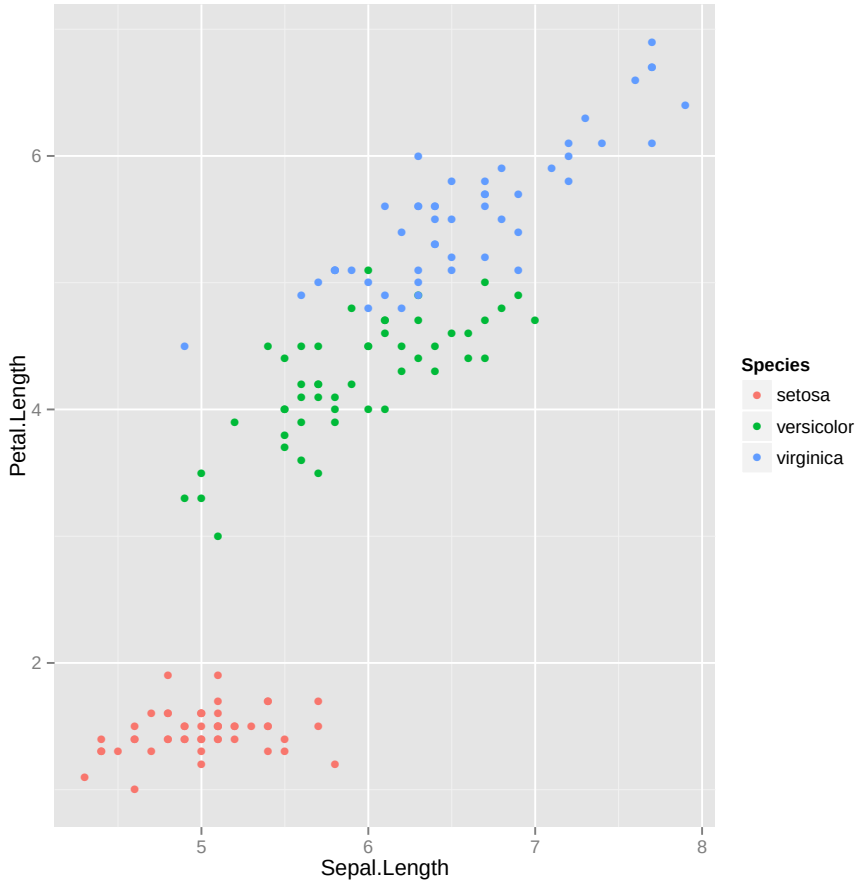

Figure 2: plot of chunk 1-pca-iris.R

```r
# add 3rd dimension with points' size
qplot(Sepal.Length, Petal.Length, data = iris, color = Species, size = Petal.Width)


# -> that makes sense

# improve the last plot with alpha (try to cope with overplotting)
qplot(Sepal.Length, Petal.Length, data = iris, color = Species, size = Petal.Width, alpha =
```

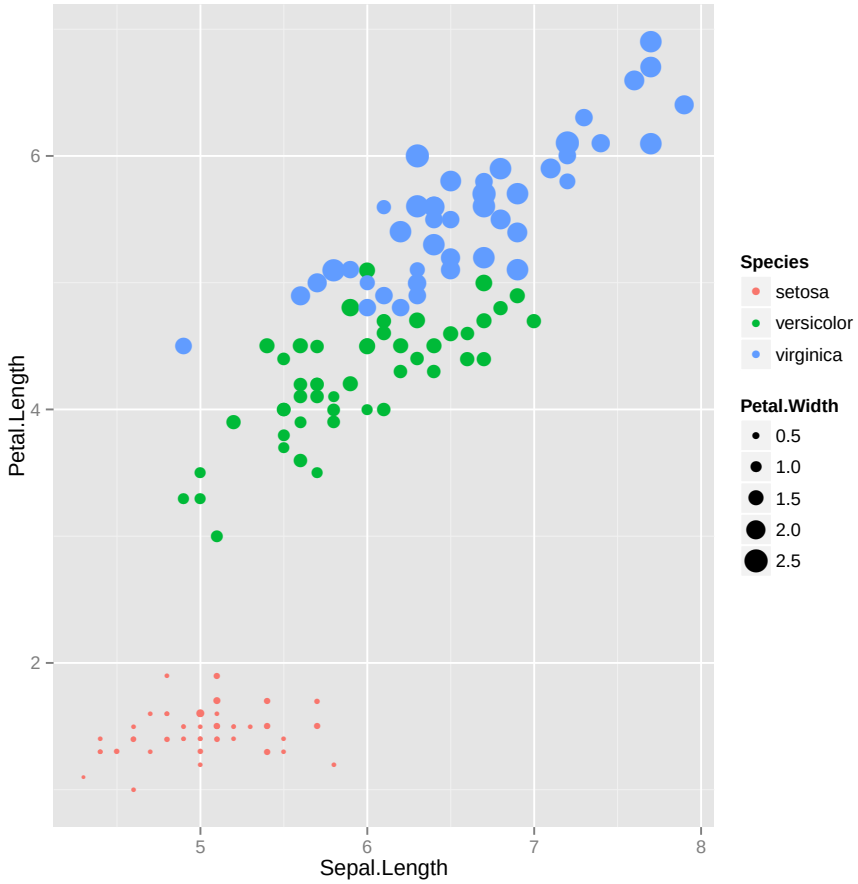

Figure 3: plot of chunk 1-pca-iris.R

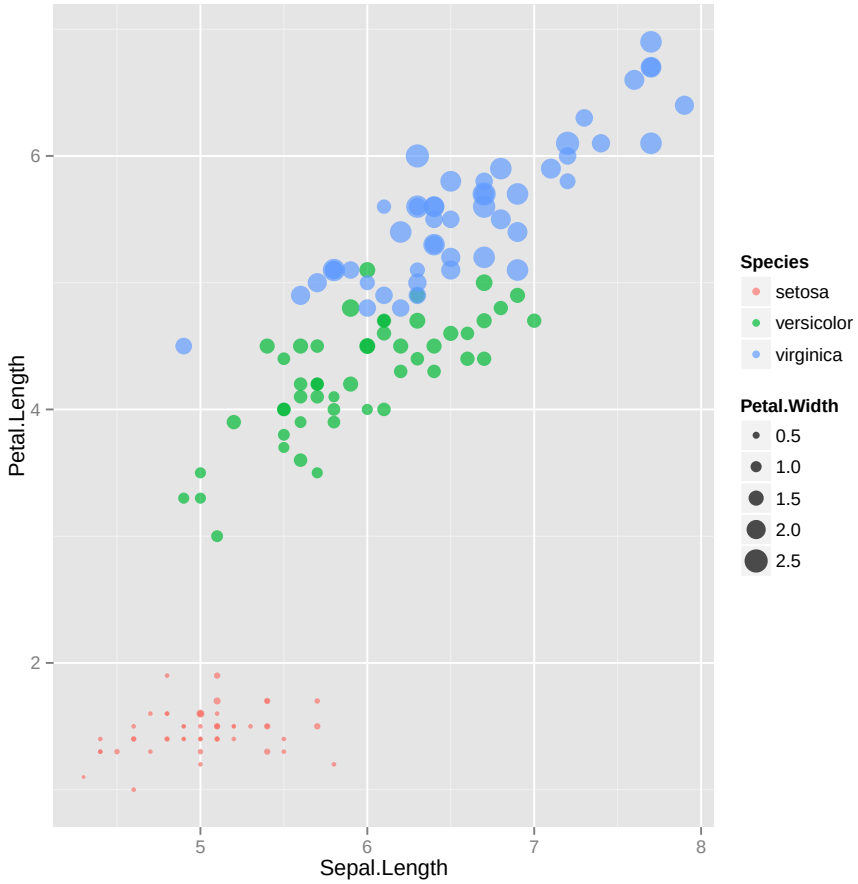

Figure 4: plot of chunk 1-pca-iris.R

```r
### PCA model (see `?prcomp` for more details)
mod <- prcomp(iris[, 1:4], center = TRUE, scale = TRUE)

# captured variance by PCs
mod$sdev
```

```
## [1] 1.7084 0.9560 0.3831 0.1439
```

```r
qplot(paste("PC", 1:4), mod$sdev / sum(mod$sdev), geom = "bar") + ggtitle("PCA: Captured Va
```

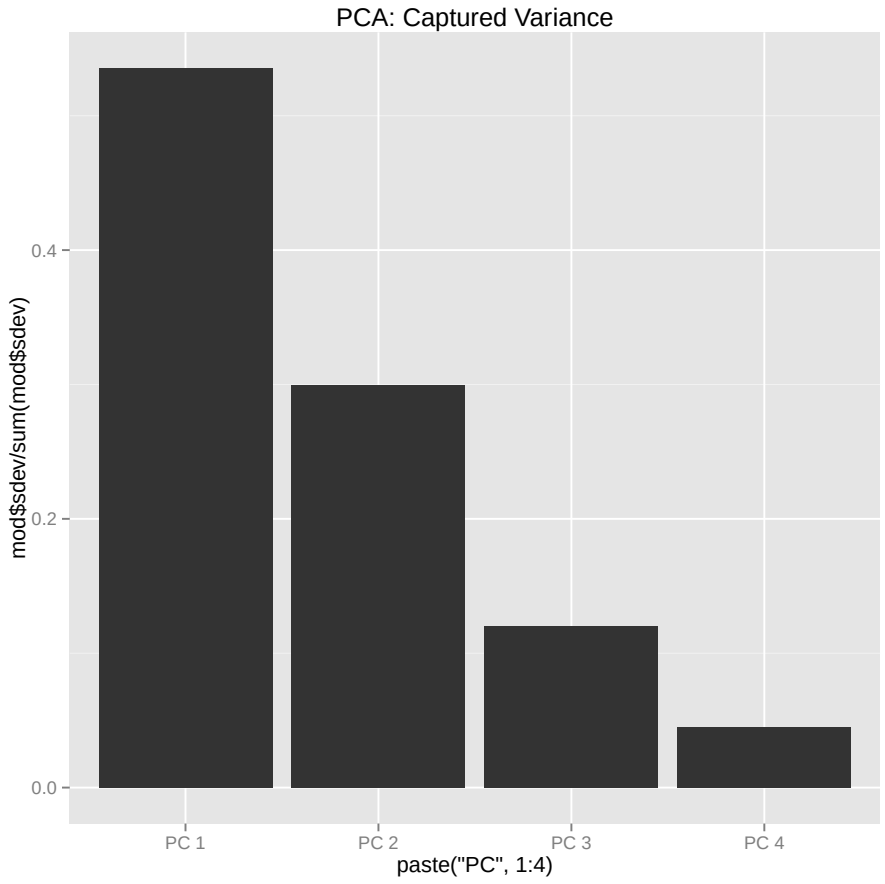

Figure 5: plot of chunk 1-pca-iris.R

```r
# loadings
loadings <- data.frame(x = rownames(mod$rotation), mod$rotation)
mf <- melt(loadings)

## Using x as id variables


qplot(x, value, data = mf, group = variable, color = variable, geom = "line") + ggtitle("PCA
```

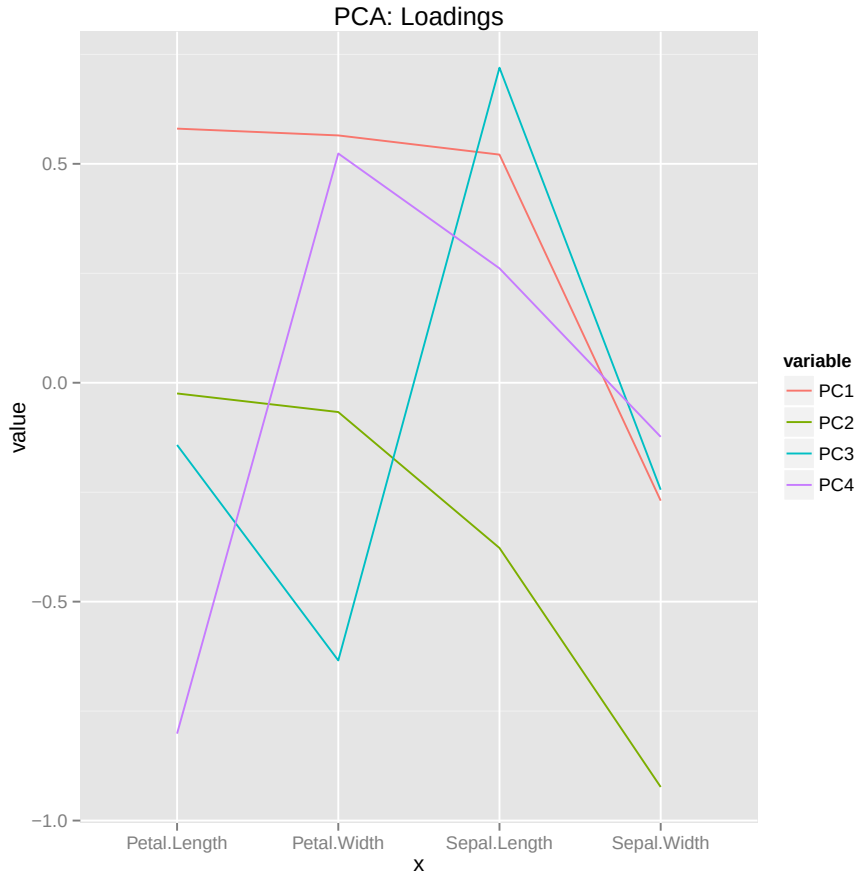

Figure 6: plot of chunk 1-pca-iris.R

```r
# scores
```

```r
scores <- as.data.frame(mod$x)

qplot(PC1, PC2, data = scores, color = iris$Species) + ggtitle("PCA: Scores")
```

Figure 7: plot of chunk 1-pca-iris.R

```r
qplot(PC1, PC2, data = scores, size = PC3, color = iris$Species) + ggtitle("PCA: Scores (3D)
```

2 02-captured-var.R

```r
### include
library(pls)
```

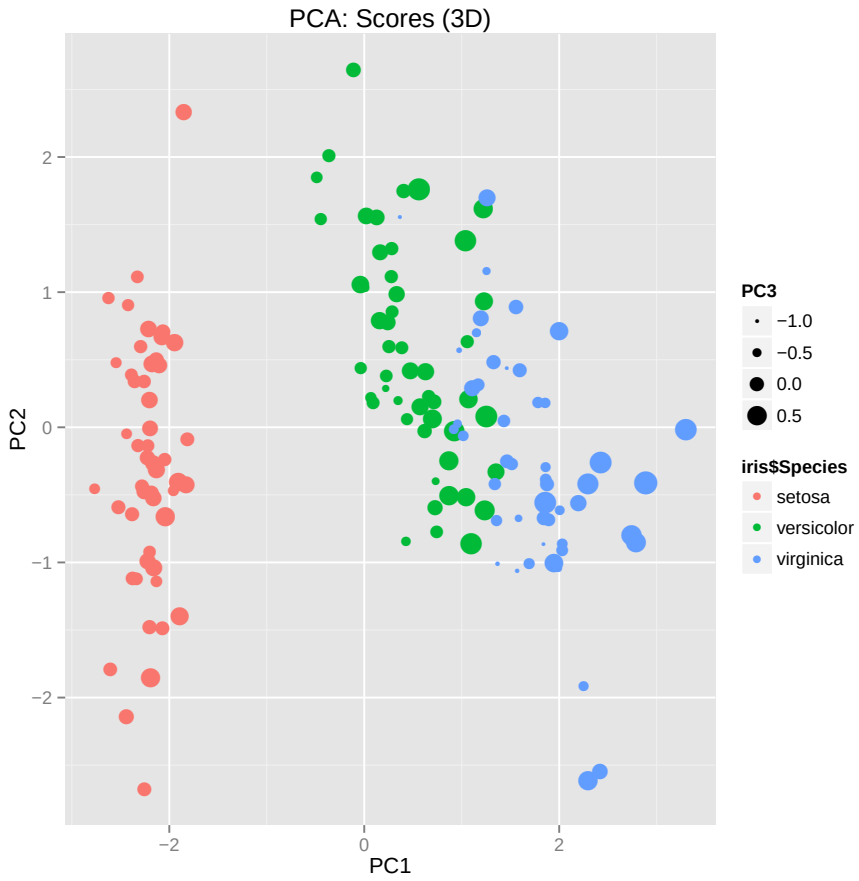

Figure 8: plot of chunk 1-pca-iris.R

```r
### parameters
pc <- 1:2 # let's compute variance for just first 2 PCs

### data
data(iris)
X <- iris[, 1:4] # data matrix

### PCA model
mod <- prcomp(X, center = TRUE, scale = FALSE)

### Option 1: captured variance via method `summary` of package `pls`
smod <- summary(mod)
var.pls <- smod$importance["Proportion of Variance", pc]

### PCA matricies
X <- as.matrix(X) # matrix of scores (needed to be a matrix)
E <- as.matrix(mod$rotation[, pc]) # `E` is a sub-space defined by PCs `pc`

# scale 'X' according to the model `mod`
X.scaled <- X
if(mod$center[1]) X.scaled <- as.matrix(sweep(X.scaled, 2, mod$center))
if(mod$scale[1])  X.scaled <- as.matrix(sweep(X.scaled, 2, mod$scale, "/"))

var.projected <- apply(E, 2, function(e) sum((X.scaled %*% e)^2))
var.total <- sum(apply(X.scaled, 2, function(x) sum((x)^2)))

var.pc <- var.projected / var.total

### compare numbers of proportion of projected variance
var.pls
```

```
##     PC1     PC2
## 0.92462 0.05307
```

```r
var.pc
```

```
##     PC1     PC2
## 0.92462 0.05307
```