ch13output

```
##################################chapter 13, PCA############################
#### Example: Temperature of cities
## The Temperature data file is in "fixed width format", an older data file format.
## Each field is specified by column ranges.
## Below I've provided numbers to help identify the column numbers
##   as well as the first three observations in the dataset.
## 12345678901234567890123456
## [  14 char   ][ 5 ][ 5 ]
#  mobile         51.2 81.6
#  phoenix        51.2 91.2
#  little rock    39.5 81.4

fn.data <- "http://statacumen.com/teach/ADA2/ADA2_notes_Ch13_temperature.dat"
temp <- read.fwf(fn.data, widths = c(14, 5, 5))
# the city names have trailing white space (we fix this below)
str(temp)
```

```
## 'data.frame':    64 obs. of  3 variables:
##  $ V1: Factor w/ 64 levels "albany        ",..: 39 48 33 56 21 27 64 62 31 36 ...
##  $ V2: num  51.2 51.2 39.5 45.1 29.9 24.8 32 35.6 54.6 67.2 ...
##  $ V3: num  81.6 91.2 81.4 75.2 73 72.7 75.8 78.7 81 82.3 ...
```

```
head(temp)
```

```
##               V1   V2   V3
## 1 mobile         51.2 81.6
## 2 phoenix        51.2 91.2
## 3 little rock    39.5 81.4
## 4 sacramento     45.1 75.2
## 5 denver         29.9 73.0
## 6 hartford       24.8 72.7
```

```
# remove that white space with strip.white=TRUE
temp <- read.fwf(fn.data, widths = c(14, 5, 5), strip.white = TRUE)
# name columns
colnames(temp) <- c("city", "january", "july")
temp$id <- 1:nrow(temp)
str(temp)
```

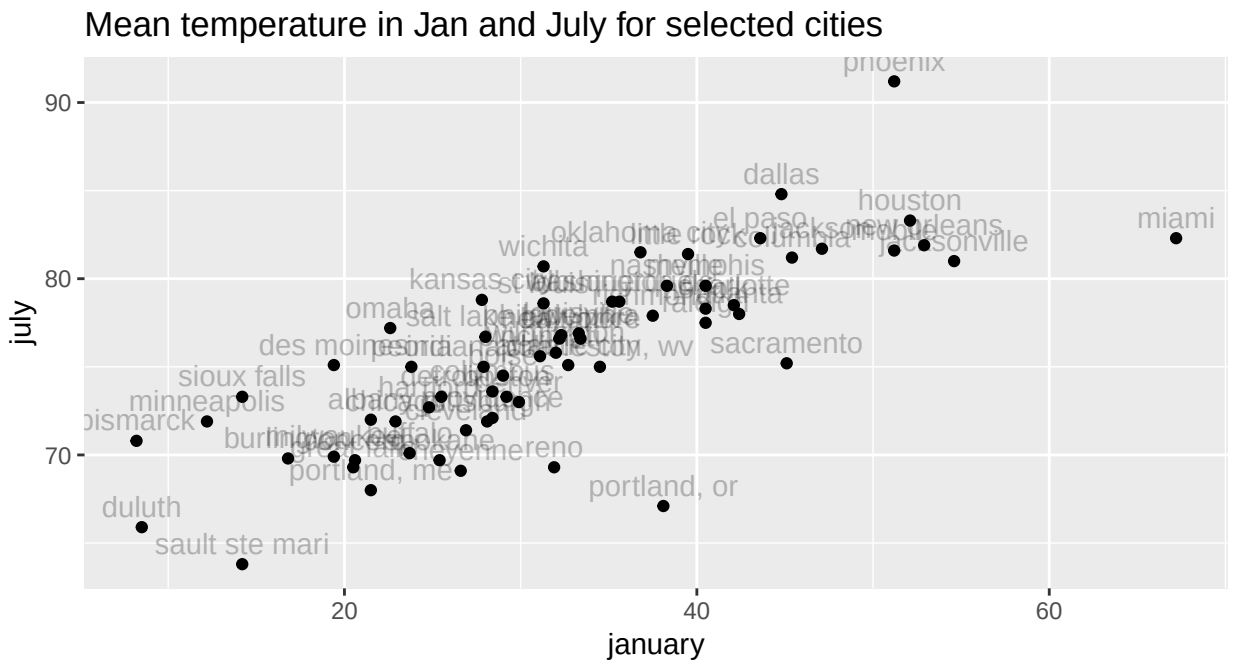
```
## 'data.frame':    64 obs. of  4 variables:
##  $ city   : Factor w/ 64 levels "albany","albuquerque",..: 39 48 33 56 21 27 64 62 31 36 ...
##  $ january: num  51.2 51.2 39.5 45.1 29.9 24.8 32 35.6 54.6 67.2 ...
##  $ july   : num  81.6 91.2 81.4 75.2 73 72.7 75.8 78.7 81 82.3 ...
##  $ id     : int  1 2 3 4 5 6 7 8 9 10 ...
```

```
head(temp)
```

```
##          city january july id
## 1      mobile    51.2 81.6  1
## 2     phoenix    51.2 91.2  2
## 3 little rock    39.5 81.4  3
## 4  sacramento    45.1 75.2  4
## 5      denver    29.9 73.0  5
```

```
## 6    hartford    24.8 72.7  6
```

```
# plot original data
library(ggplot2)
p1 <- ggplot(temp, aes(x = january, y = july))
p1 <- p1 + geom_point()           # points
p1 <- p1 + coord_fixed(ratio = 1) # makes 1 unit equal length on x- and y-axis
                                  # good idea since both are in the same units
p1 <- p1 + geom_text(aes(label = city), vjust = -0.5, alpha = 0.25) # city labels
p1 <- p1 + labs(title = "Mean temperature in Jan and July for selected cities")
print(p1)
```




```
dev.copy(jpeg,filename="~/Desktop/jenn/teaching/ADA2/lecture notes/plots/ch13plot1.jpg")
```

```
## jpeg
##    3
```

```
dev.off()
```

```
## pdf
##   2
```

```
# perform PCA on covariance matrix
temp.pca <- princomp( ~ january + july, data = temp)
# standard deviation and proportion of variation for each component
summary(temp.pca)
```

```
## Importance of components:
##                            Comp.1    Comp.2
## Standard deviation     12.3217642 3.0004557
## Proportion of Variance  0.9440228 0.0559772
## Cumulative Proportion   0.9440228 1.0000000
```

```
# coefficients for PCs
loadings(temp.pca)
```

```
##
## Loadings:
##         Comp.1 Comp.2
## january -0.939  0.343
## july    -0.343 -0.939
##
##                Comp.1 Comp.2
## SS loadings       1.0    1.0
## Proportion Var    0.5    0.5
## Cumulative Var    0.5    1.0
```

```
# scores are coordinates of each observation on PC scale
head(temp.pca$scores)
```

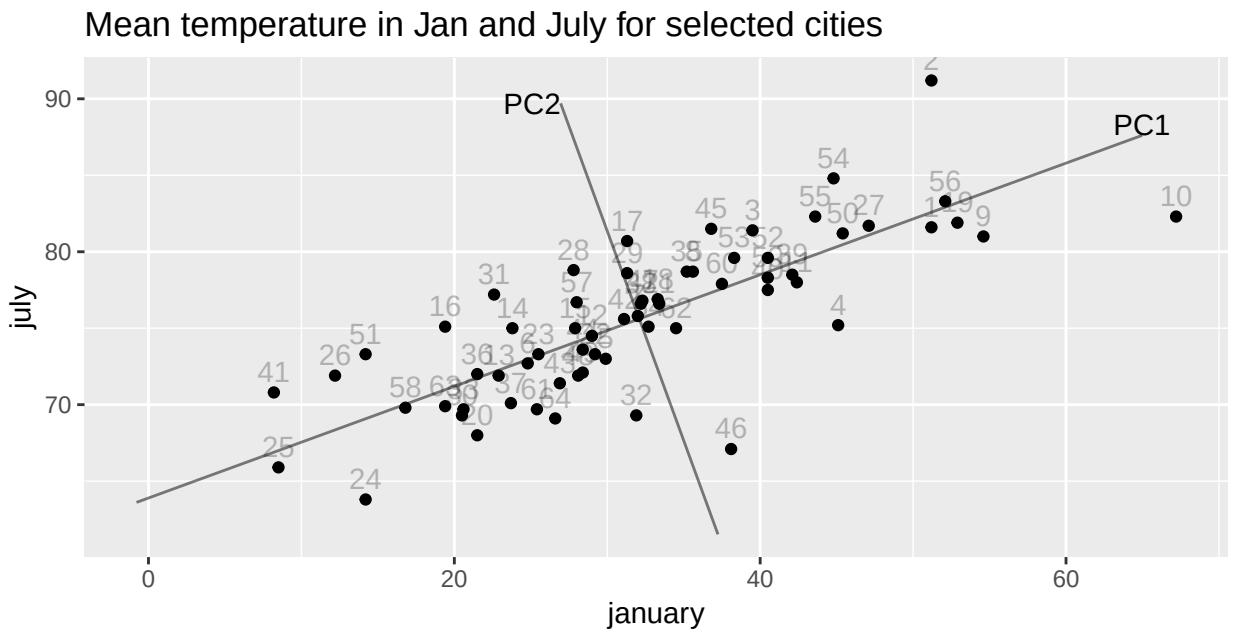
```
##       Comp.1     Comp.2
## 1 -20.000106  0.9239612
## 2 -23.291460 -8.0941867
## 3  -8.940669 -2.8994977
## 4 -12.075589  4.8446790
## 5   2.957414  1.7000283
## 6   7.851160  0.2333138
```

```
# create small data.frame with endpoints of PC lines through data
line.scale <- c(35, 15) # length of PCA lines to draw
# endpoints of lines to draw
temp.pca.line.endpoints <-
  data.frame(PC = c(rep("PC1", 2), rep("PC2", 2))
             ,  x = c(temp.pca$center[1] - line.scale[1] * temp.pca$loadings[1, 1]
                    , temp.pca$center[1] + line.scale[1] * temp.pca$loadings[1, 1]
                    , temp.pca$center[1] - line.scale[2] * temp.pca$loadings[1, 2]
                    , temp.pca$center[1] + line.scale[2] * temp.pca$loadings[1, 2])
             ,  y = c(temp.pca$center[2] - line.scale[1] * temp.pca$loadings[2, 1]
                    , temp.pca$center[2] + line.scale[1] * temp.pca$loadings[2, 1]
                    , temp.pca$center[2] - line.scale[2] * temp.pca$loadings[2, 2]
                    , temp.pca$center[2] + line.scale[2] * temp.pca$loadings[2, 2])
              )
temp.pca.line.endpoints
```

```
##    PC          x        y
## 1 PC1 64.9739769 87.61066
## 2 PC1 -0.7833519 63.61121
## 3 PC2 26.9525727 89.70179
## 4 PC2 37.2380523 61.52008
```

```
# plot original data with PCA vectors overlayed
library(ggplot2)
p1 <- ggplot(temp, aes(x = january, y = july))
p1 <- p1 + geom_point()           # points
p1 <- p1 + coord_fixed(ratio = 1) # makes 1 unit equal length on x- and y-axis
                                  # good idea since both are in the same units
p1 <- p1 + geom_text(aes(label = id), vjust = -0.5, alpha = 0.25) # city labels
# plot PC lines
p1 <- p1 + geom_path(data = subset(temp.pca.line.endpoints, PC=="PC1"), aes(x=x, y=y)
                     , alpha=0.5)
p1 <- p1 + geom_path(data = subset(temp.pca.line.endpoints, PC=="PC2"), aes(x=x, y=y)
                     , alpha=0.5)
```

```
# label lines
p1 <- p1 + annotate("text"
                 , x     = temp.pca.line.endpoints$x[1]
                 , y     = temp.pca.line.endpoints$y[1]
                 , label = as.character(temp.pca.line.endpoints$PC[1])
                 , vjust = 0) #, size  = 10)
p1 <- p1 + annotate("text"
                 , x     = temp.pca.line.endpoints$x[3]
                 , y     = temp.pca.line.endpoints$y[3]
                 , label = as.character(temp.pca.line.endpoints$PC[3])
                 , hjust = 1) #, size  = 10)
p1 <- p1 + labs(title = "Mean temperature in Jan and July for selected cities")
print(p1)
```




```
dev.copy(jpeg,filename="~/Desktop/jenn/teaching/ADA2/lecture notes/plots/ch13plot2.jpg")
```

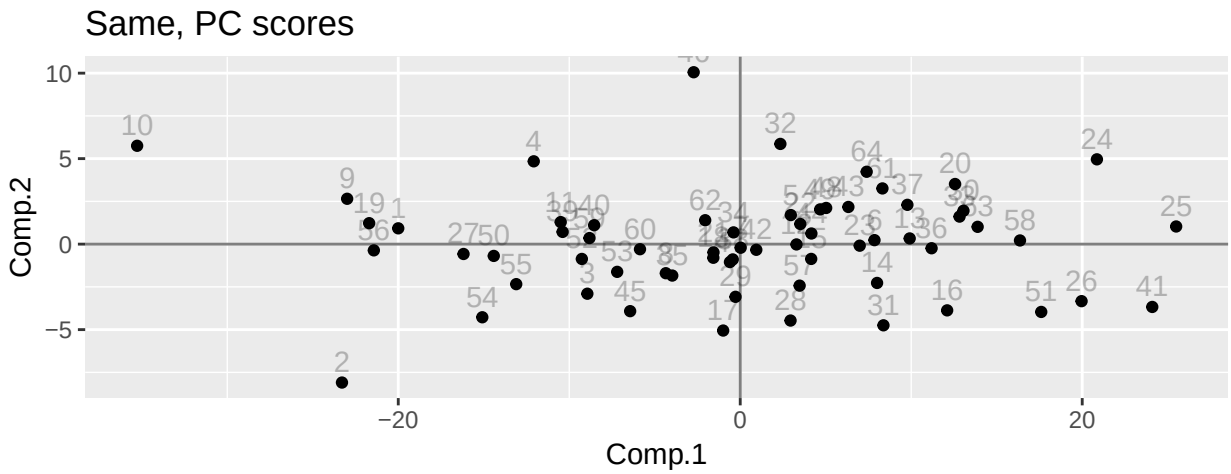
```
## jpeg
##    3
```

```
dev.off()
```

```
## pdf
##   2
```

```
# plot PCA scores (data on PC-scale centered at 0)
library(ggplot2)
p2 <- ggplot(as.data.frame(temp.pca$scores), aes(x = Comp.1, y = Comp.2))
p2 <- p2 + geom_point()           # points
p2 <- p2 + coord_fixed(ratio = 1) # makes 1 unit equal length on x- and y-axis
                                  # good idea since both are in the same units
p2 <- p2 + geom_text(aes(label = rownames(temp.pca$scores)), vjust = -0.5, alpha = 0.25) # city labels
# plot PC lines
p2 <- p2 + geom_vline(xintercept = 0, alpha=0.5)
p2 <- p2 + geom_hline(yintercept = 0, alpha=0.5)
p2 <- p2 + labs(title = "Same, PC scores")
```

```
print(p2)
```




```
dev.copy(jpeg,filename="~/Desktop/jenn/teaching/ADA2/lecture notes/plots/ch13plot3.jpg")
```

```
## jpeg
##    3
```

```
dev.off()
```

```
## pdf
##   2
```

```
# plot PCA scores (data on (negative) PC-scale centered at 0)
library(ggplot2)          # negative temp.pca$scores
p3 <- ggplot(as.data.frame(-temp.pca$scores), aes(x = Comp.1, y = Comp.2))
p3 <- p3 + geom_point()            # points
p3 <- p3 + coord_fixed(ratio = 1) # makes 1 unit equal length on x- and y-axis
                                   # good idea since both are in the same units
p3 <- p3 + geom_text(aes(label = rownames(temp.pca$scores)), vjust = -0.5, alpha = 0.25) # city labels
# plot PC lines
p3 <- p3 + geom_vline(xintercept = 0, alpha=0.5)
p3 <- p3 + geom_hline(yintercept = 0, alpha=0.5)
p3 <- p3 + labs(title = "Same, but negative PC scores match orientation of original data")
#print(p3)




# variance of data (on diagonals, covariance of off-diags)
var(temp[,c("january","july")])
```

```
##           january     july
## january 137.1811 46.72910
## july     46.7291 26.20035
```

```
# sum of variance
sum(diag(var(temp[,c("january","july")])))
```

```
## [1] 163.3814
```

```
# variance of PC scores
var(temp.pca$scores)
```

```
##              Comp.1       Comp.2
## Comp.1 1.542358e+02 1.831125e-15
## Comp.2 1.831125e-15 9.145635e+00
```

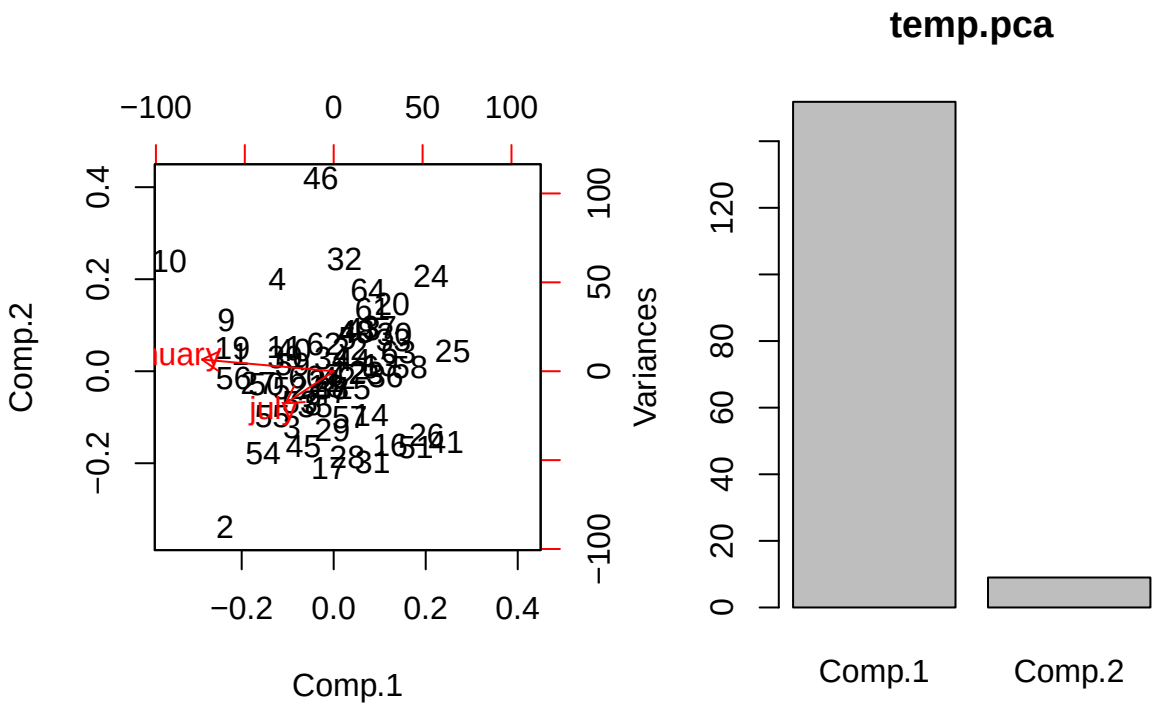
```
# sum is same as original data
sum(diag(var(temp.pca$scores)))
```

```
## [1] 163.3814
```

```
# eigenvalues and eigenvectors of covariance matrix give PC variance and loadings
eigen(var(temp[,c("january","july")]))
```

```
## eigen() decomposition
## $values
## [1] 154.235808   9.145635
##
## $vectors
##            [,1]       [,2]
## [1,] -0.9393904  0.3428493
## [2,] -0.3428493 -0.9393904
```

```
# a couple built-in plots
par(mfrow=c(1,2))
biplot(temp.pca)
screeplot(temp.pca)
```




```
dev.copy(jpeg,filename="~/Desktop/jenn/teaching/ADA2/lecture notes/plots/ch13plot4.jpg")
```

```
## jpeg
##    3
```

```
dev.off()
```

```
## pdf
```

```
##   2
```

```
temp.z <- temp
# manual z-score
temp.z$january <- (temp.z$january - mean(temp.z$january)) / sd(temp.z$january)
# z-score using R function scale()
temp.z$july <- scale(temp.z$july)

# the manual z-score and scale() match
all.equal(temp.z$january, as.vector(scale(temp.z$january)))
```

```
## [1] TRUE
```

```
# scale() includes attributes for the mean() and sd() used for z-scoring
str(temp.z)
```

```
## 'data.frame':    64 obs. of  4 variables:
##  $ city   : Factor w/ 64 levels "albany","albuquerque",..: 39 48 33 56 21 27 64 62 31 36 ...
##  $ january: num  1.631 1.631 0.632 1.11 -0.187 ...
##  $ july   : num [1:64, 1] 1.1701 3.0456 1.131 -0.0803 -0.5101 ...
##   ..- attr(*, "scaled:center")= num 75.6
##   ..- attr(*, "scaled:scale")= num 5.12
##  $ id     : int  1 2 3 4 5 6 7 8 9 10 ...
```

```
head(temp.z)
```

```
##          city    january        july id
## 1      mobile  1.6311459  1.17005226  1
## 2     phoenix  1.6311459  3.04555476  2
## 3 little rock  0.6322075  1.13097929  3
## 4  sacramento  1.1103319 -0.08028274  4
## 5      denver -0.1874344 -0.51008540  5
## 6    hartford -0.6228691 -0.56869485  6
```

```
# z-scored data has mean 0 and variance 1
colMeans(temp.z[,c("january","july")])
```

```
##       january          july
##  1.228943e-16 -1.214842e-15
```

```
var(temp.z[,c("january","july")])
```

```
##           january      july
## january 1.0000000 0.7794472
## july    0.7794472 1.0000000
```

```
# the correlation is used to contruct the PCs
#   (same as covariance for z-scored data)
cor(temp.z[,c("january","july")])
```

```
##           january      july
## january 1.0000000 0.7794472
## july    0.7794472 1.0000000
```

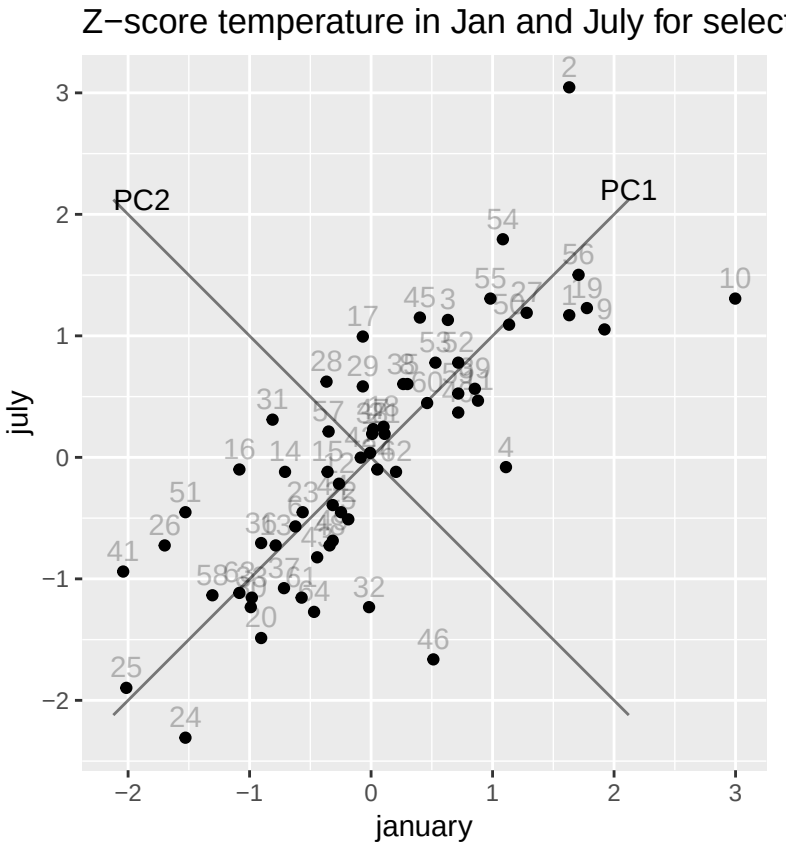
```
## Plot z-scored data
temp.z.pca <- princomp( ~ january + july, data = temp.z)

# create small data.frame with endpoints of PC lines through data
line.scale <- c(3, 3) # length of PCA lines to draw
```

```
# endpoints of lines to draw
temp.z.pca.line.endpoints <-
  data.frame(PC = c(rep("PC1", 2), rep("PC2", 2))
           ,  x = c(temp.z.pca$center[1] - line.scale[1] * temp.z.pca$loadings[1, 1]
                  , temp.z.pca$center[1] + line.scale[1] * temp.z.pca$loadings[1, 1]
                  , temp.z.pca$center[1] - line.scale[2] * temp.z.pca$loadings[1, 2]
                  , temp.z.pca$center[1] + line.scale[2] * temp.z.pca$loadings[1, 2])
           ,  y = c(temp.z.pca$center[2] - line.scale[1] * temp.z.pca$loadings[2, 1]
                  , temp.z.pca$center[2] + line.scale[1] * temp.z.pca$loadings[2, 1]
                  , temp.z.pca$center[2] - line.scale[2] * temp.z.pca$loadings[2, 2]
                  , temp.z.pca$center[2] + line.scale[2] * temp.z.pca$loadings[2, 2])
            )
temp.z.pca.line.endpoints
```

```
##    PC        x        y
## 1 PC1  2.12132  2.12132
## 2 PC1 -2.12132 -2.12132
## 3 PC2 -2.12132  2.12132
## 4 PC2  2.12132 -2.12132
```

```
# plot original data with PCA vectors overlayed
library(ggplot2)
p1 <- ggplot(temp.z, aes(x = january, y = july))
p1 <- p1 + geom_point()           # points
p1 <- p1 + coord_fixed(ratio = 1) # makes 1 unit equal length on x- and y-axis
                                  # good idea since both are in the same units
p1 <- p1 + geom_text(aes(label = id), vjust = -0.5, alpha = 0.25) # city labels
# plot PC lines
p1 <- p1 + geom_path(data = subset(temp.z.pca.line.endpoints, PC=="PC1"), aes(x=x, y=y), alpha=0.5)
p1 <- p1 + geom_path(data = subset(temp.z.pca.line.endpoints, PC=="PC2"), aes(x=x, y=y), alpha=0.5)
# label lines
p1 <- p1 + annotate("text"
                  , x     = temp.z.pca.line.endpoints$x[1]
                  , y     = temp.z.pca.line.endpoints$y[1]
                  , label = as.character(temp.z.pca.line.endpoints$PC[1])
                  , vjust = 0) #, size  = 10)
p1 <- p1 + annotate("text"
                  , x     = temp.z.pca.line.endpoints$x[3]
                  , y     = temp.z.pca.line.endpoints$y[3]
                  , label = as.character(temp.z.pca.line.endpoints$PC[3])
                  , hjust = 0) #, size  = 10)
p1 <- p1 + labs(title = "Z-score temperature in Jan and July for selected cities")
print(p1)
```




```
# perform PCA on correlation matrix
temp.pca2 <- princomp( ~ january + july, data = temp, cor = TRUE)
# standard deviation and proportion of variation for each component
summary(temp.pca2)
```

```
## Importance of components:
##                           Comp.1    Comp.2
## Standard deviation     1.3339592 0.4696305
## Proportion of Variance 0.8897236 0.1102764
## Cumulative Proportion  0.8897236 1.0000000
```

```
# coefficients for PCs
loadings(temp.pca2)
```

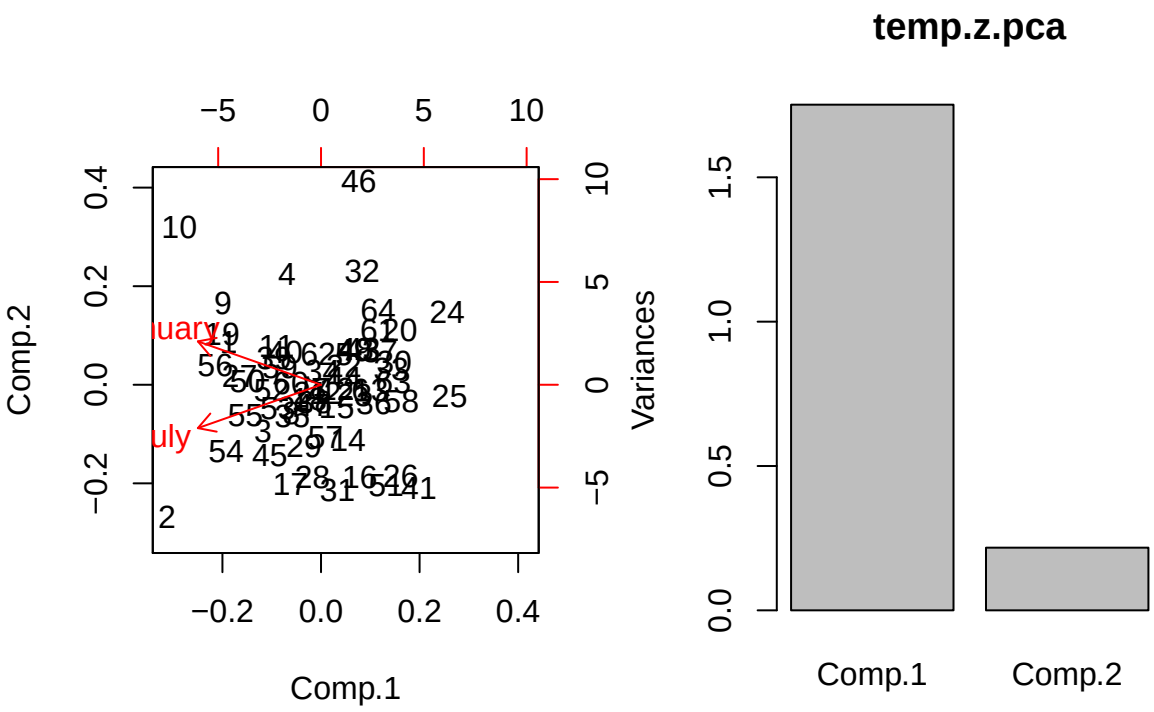
```
##
## Loadings:
##         Comp.1 Comp.2
## january  0.707 -0.707
## july     0.707  0.707
##
##                Comp.1 Comp.2
## SS loadings       1.0    1.0
## Proportion Var    0.5    0.5
## Cumulative Var    0.5    1.0
```

```
# scores are coordinates of each observation on PC scale
head(temp.pca2$scores)
```

```
##       Comp.1     Comp.2
```

```
## 1  1.9964045 -0.3286199
## 2  3.3330689  1.0080444
## 3  1.2566173  0.3554730
## 4  0.7341125 -0.8485470
## 5 -0.4971200 -0.2299523
## 6 -0.8492236  0.0386098
```

```
# a couple built-in plots
par(mfrow=c(1,2))
biplot(temp.z.pca)
screeplot(temp.z.pca)
```




```
#### Example: Painted turtle shells
fn.data <- "http://statacumen.com/teach/ADA2/ADA2_notes_Ch13_shells.dat"
shells <- read.table(fn.data, header = TRUE)
str(shells)
```

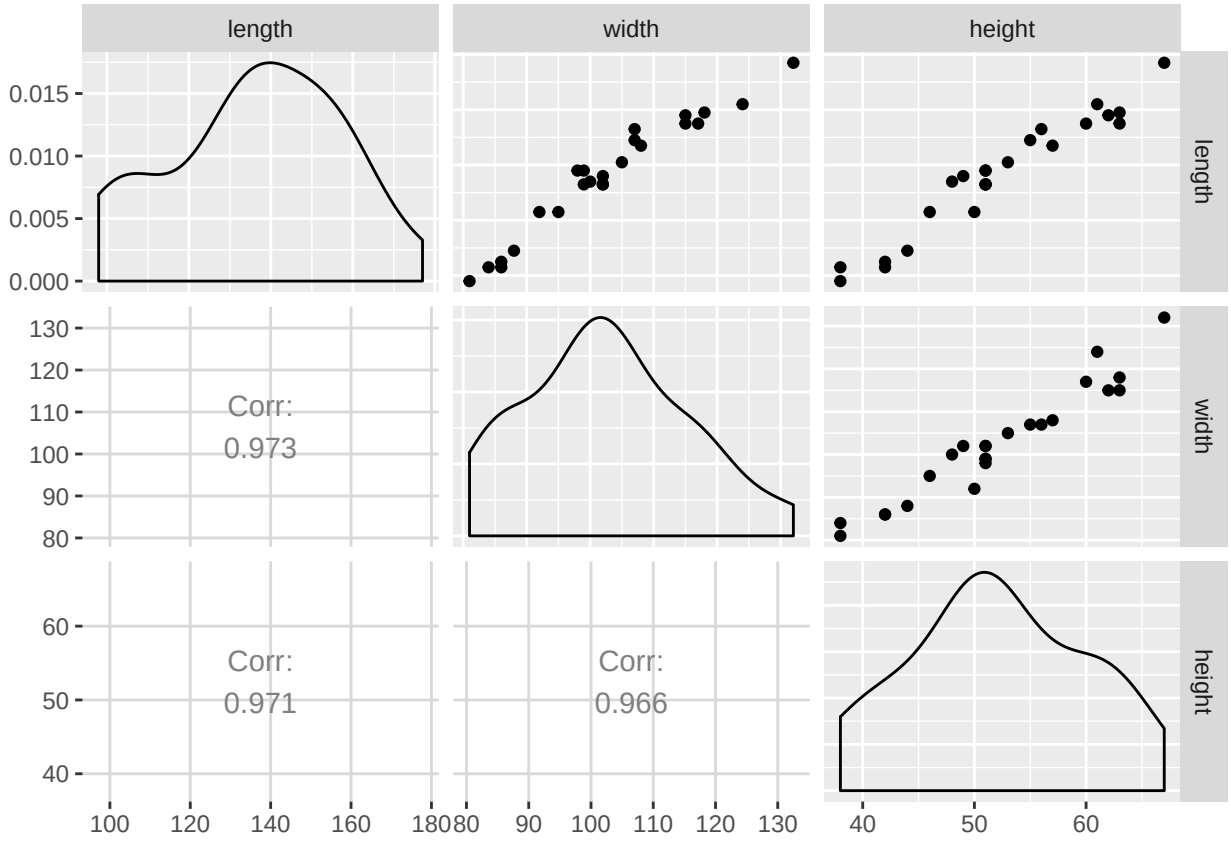
```
## 'data.frame':    24 obs. of  3 variables:
##  $ length: int  98 103 103 105 109 123 123 133 133 133 ...
##  $ width : int  81 84 86 86 88 92 95 99 102 102 ...
##  $ height: int  38 38 42 42 44 50 46 51 51 51 ...
```

```
head(shells)
```

```
##   length width height
## 1     98    81     38
## 2    103    84     38
## 3    103    86     42
## 4    105    86     42
## 5    109    88     44
## 6    123    92     50
```

```
## Scatterplot matrix
library(ggplot2)
```

```
suppressMessages(suppressWarnings(library(GGally)))
# put scatterplots on top so y axis is vertical
p <- ggpairs(shells, upper = list(continuous = "points")
                   , lower = list(continuous = "cor")
             )
print(p)
```




```
dev.copy(jpeg,filename="~/Desktop/jenn/teaching/ADA2/lecture notes/plots/ch13plot5.jpg")
```

```
## jpeg
##   3
```

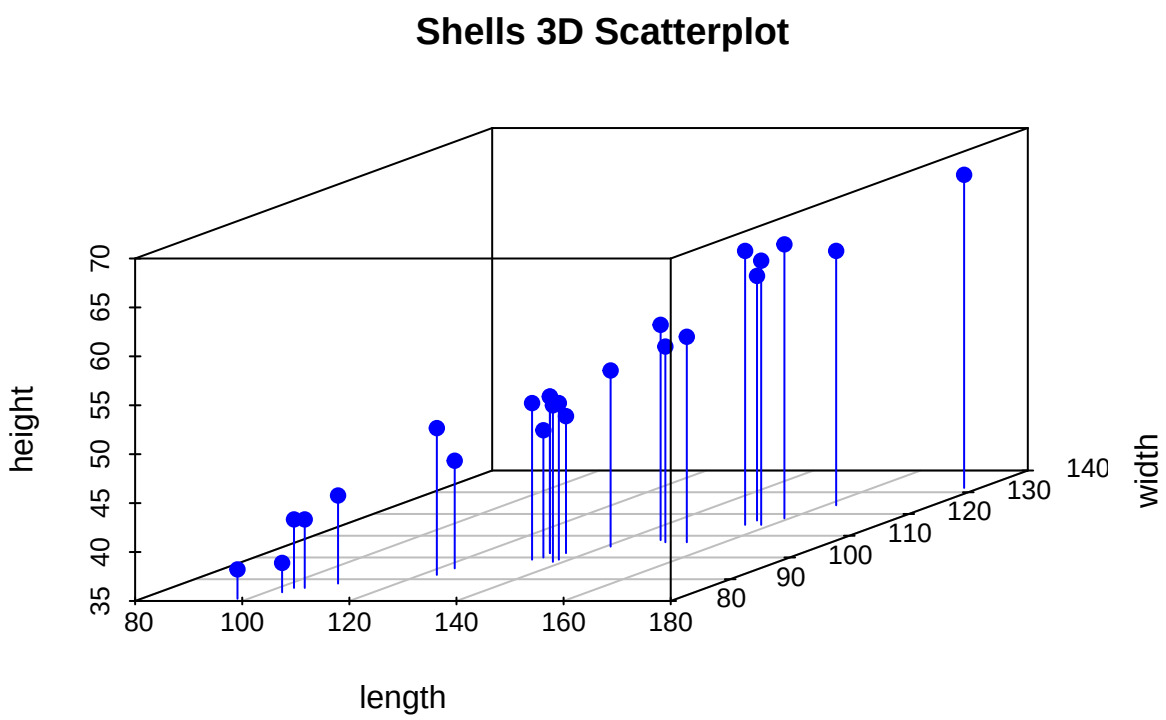
```
dev.off()
```

```
## pdf
##   2
```

```
  # detach package after use so reshape2 works (old reshape (v.1) conflicts)
  #detach("package:GGally", unload=TRUE)
  #detach("package:reshape", unload=TRUE)

## 3D scatterplot
library(scatterplot3d)
par(mfrow=c(1,1))
with(shells, {
  scatterplot3d(x=length
              , y=width
              , z=height
```

```
              , main="Shells 3D Scatterplot"
              , type = "h" # lines to the horizontal xy-plane
              , color="blue", pch=19, # filled blue circles
              #, highlight.3d = TRUE  # makes color change with z-axis value
              )
  })
```




```
dev.copy(jpeg,filename="~/Desktop/jenn/teaching/ADA2/lecture notes/plots/ch13plot6.jpg")
```

```
## jpeg
##    3
```

```
dev.off()
```

```
## pdf
##   2
```

```
#### For a rotatable 3D plot, use plot3d() from the rgl library
# ##   This uses the R version of the OpenGL (Open Graphics Library)
# library(rgl)
# with(shells, { plot3d(x = length, y = width, z = height) })



# perform PCA on covariance matrix
shells.pca <- princomp( ~ length + width + height, data = shells)
# standard deviation and proportion of variation for each component
summary(shells.pca)
```

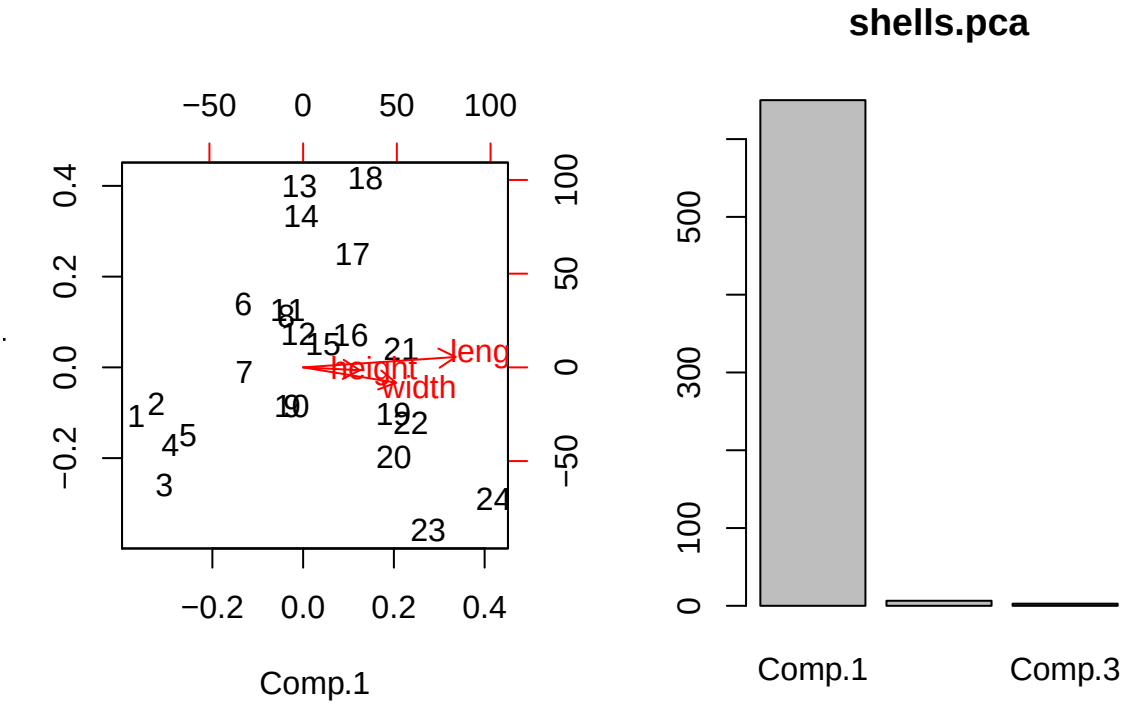
```
## Importance of components:
##                            Comp.1      Comp.2      Comp.3
## Standard deviation     25.4970668 2.547081962 1.653745717
## Proportion of Variance  0.9860122 0.009839832 0.004148005
```

```
## Cumulative Proportion   0.9860122 0.995851995 1.000000000
```

```
# coefficients for PCs
loadings(shells.pca)
```

```
##
## Loadings:
##        Comp.1 Comp.2 Comp.3
## length  0.814  0.555 -0.172
## width   0.496 -0.818 -0.291
## height  0.302 -0.151  0.941
##
##                Comp.1 Comp.2 Comp.3
## SS loadings     1.000  1.000  1.000
## Proportion Var  0.333  0.333  0.333
## Cumulative Var  0.333  0.667  1.000
```

```
# a couple built-in plots
par(mfrow=c(1,2))
biplot(shells.pca)
screeplot(shells.pca)
```




```
dev.copy(jpeg,filename="~/Desktop/jenn/teaching/ADA2/lecture notes/plots/ch13plot7.jpg")
```

```
## jpeg
##    3
```

```
dev.off()
```

```
## pdf
##   2
```

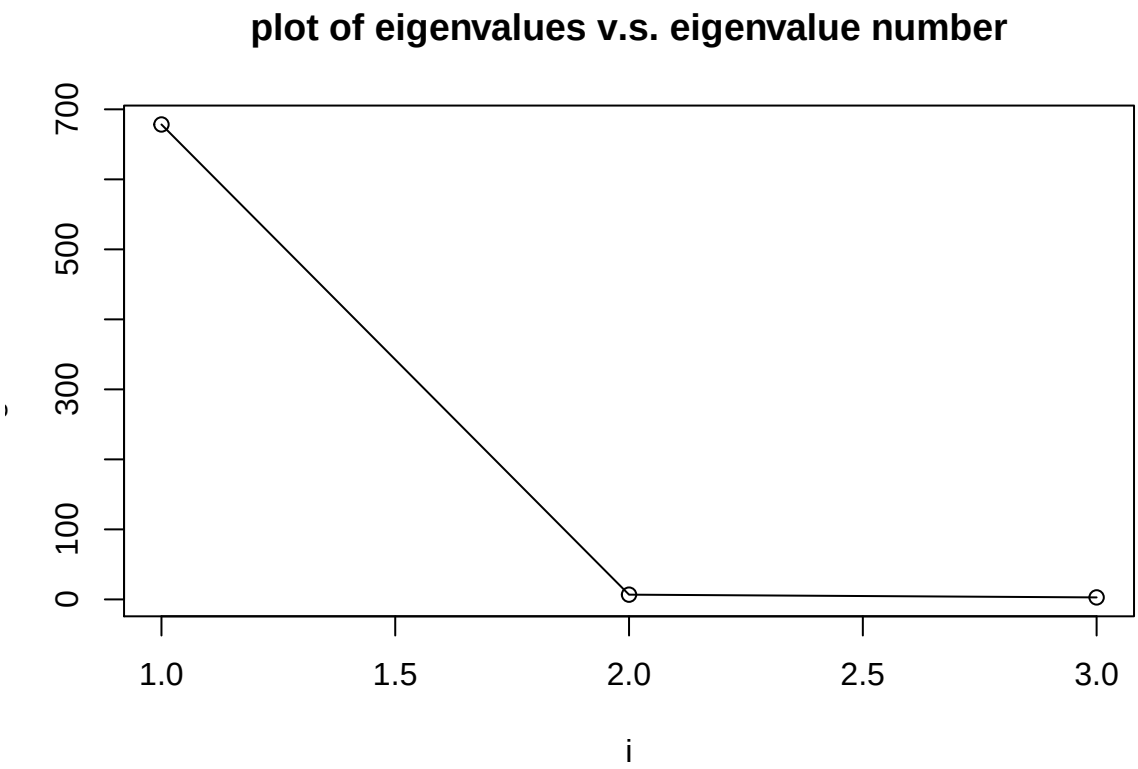
```
# eigenvalues and eigenvectors of covariance matrix give PC variance and loadings
par(mfrow=c(1,1))
eigen(var(shells[,c("length","width","height")]))
```

```
## eigen() decomposition
## $values
## [1] 678.365651   6.769697   2.853783
##
## $vectors
##           [,1]       [,2]       [,3]
## [1,] 0.8138808  0.5548963 -0.1723025
## [2,] 0.4961059 -0.8180268 -0.2910518
## [3,] 0.3024516 -0.1514012  0.9410636
```

```
eigenvalues<-eigen(var(shells[,c("length","width","height")]))
eigenvalues
```

```
## eigen() decomposition
## $values
## [1] 678.365651   6.769697   2.853783
##
## $vectors
##           [,1]       [,2]       [,3]
## [1,] 0.8138808  0.5548963 -0.1723025
## [2,] 0.4961059 -0.8180268 -0.2910518
## [3,] 0.3024516 -0.1514012  0.9410636
```

```
plot(seq(1:3),eigenvalues$values,main="plot of eigenvalues v.s. eigenvalue number",xlab="i",ylab="Eigen
lines(seq(1:3),eigenvalues$values)
```




```
dev.copy(jpeg,filename="~/Desktop/jenn/teaching/ADA2/lecture notes/plots/ch13plot8.jpg")
```

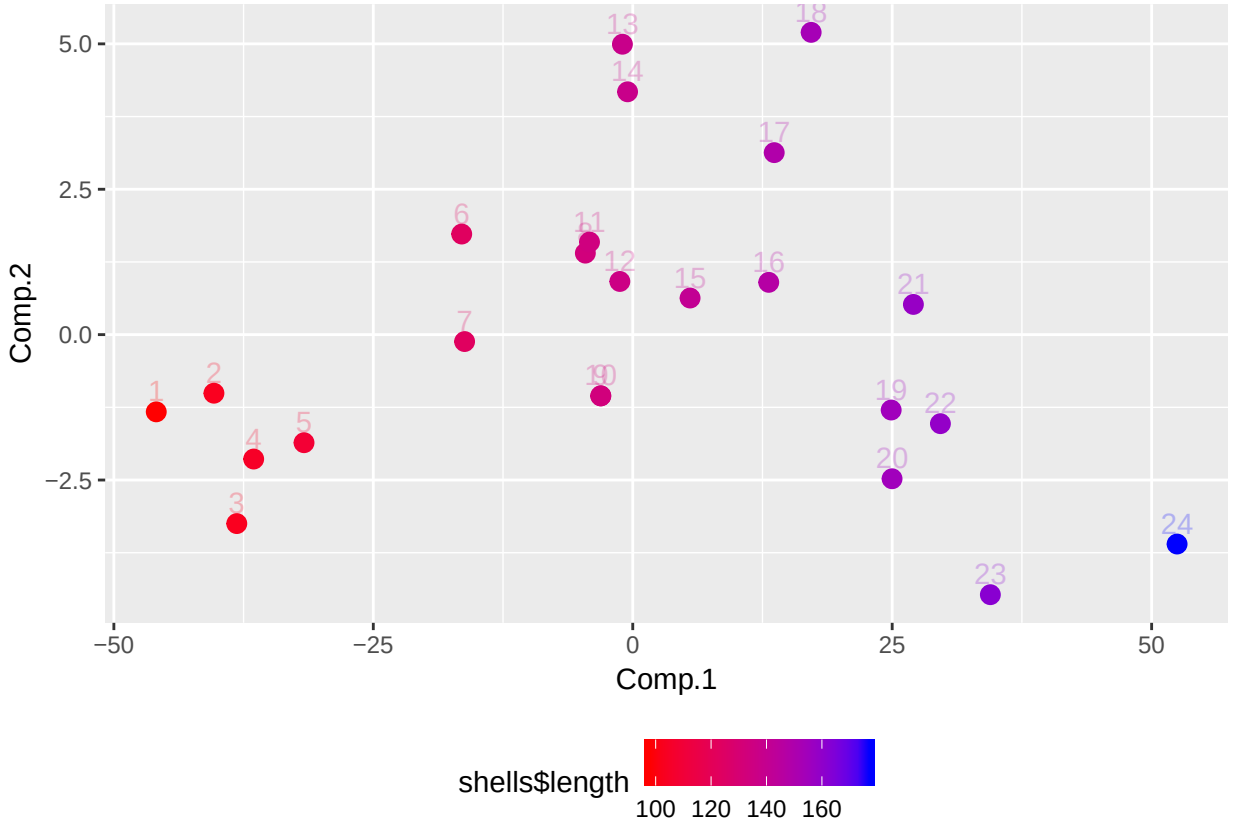
```
## jpeg
##    3
```

```
dev.off()
```

```
## pdf
##   2
```

```
##two-dimensional plots of PC1 against PC2, PC1 against PC3, and PC2 against PC3.
shells$id<-seq(1:24)
library(ggplot2)
p1 <- ggplot(as.data.frame(shells.pca$scores), aes(x = Comp.1, y = Comp.2, colour = shells$length))
p1 <- p1 + scale_colour_gradientn(colours=c("red", "blue"))
p1 <- p1 + geom_text(aes(label = shells$id), vjust = -0.5, alpha = 0.25)
p1 <- p1 + geom_point(size = 3)
p1 <- p1 + theme(legend.position="bottom")
p2 <- ggplot(as.data.frame(shells.pca$scores), aes(x = Comp.1, y = Comp.3, colour = shells$length))
p2 <- p2 + scale_colour_gradientn(colours=c("red", "blue"))
p2 <- p2 + geom_text(aes(label = shells$id), vjust = -0.5, alpha = 0.25)
p2 <- p2 + geom_point(size = 3)
p2 <- p2 + theme(legend.position="none")
p3 <- ggplot(as.data.frame(shells.pca$scores), aes(x = Comp.2, y = Comp.3, colour = shells$length))
p3 <- p3 + scale_colour_gradientn(colours=c("red", "blue"))
p3 <- p3 + geom_text(aes(label = shells$id), vjust = -0.5, alpha = 0.25)
p3 <- p3 + geom_point(size = 3)
p3 <- p3 + theme(legend.position="none")

print(p1)
```




```
dev.copy(jpeg,filename="~/Desktop/jenn/teaching/ADA2/lecture notes/plots/ch13plot9.jpg")
```

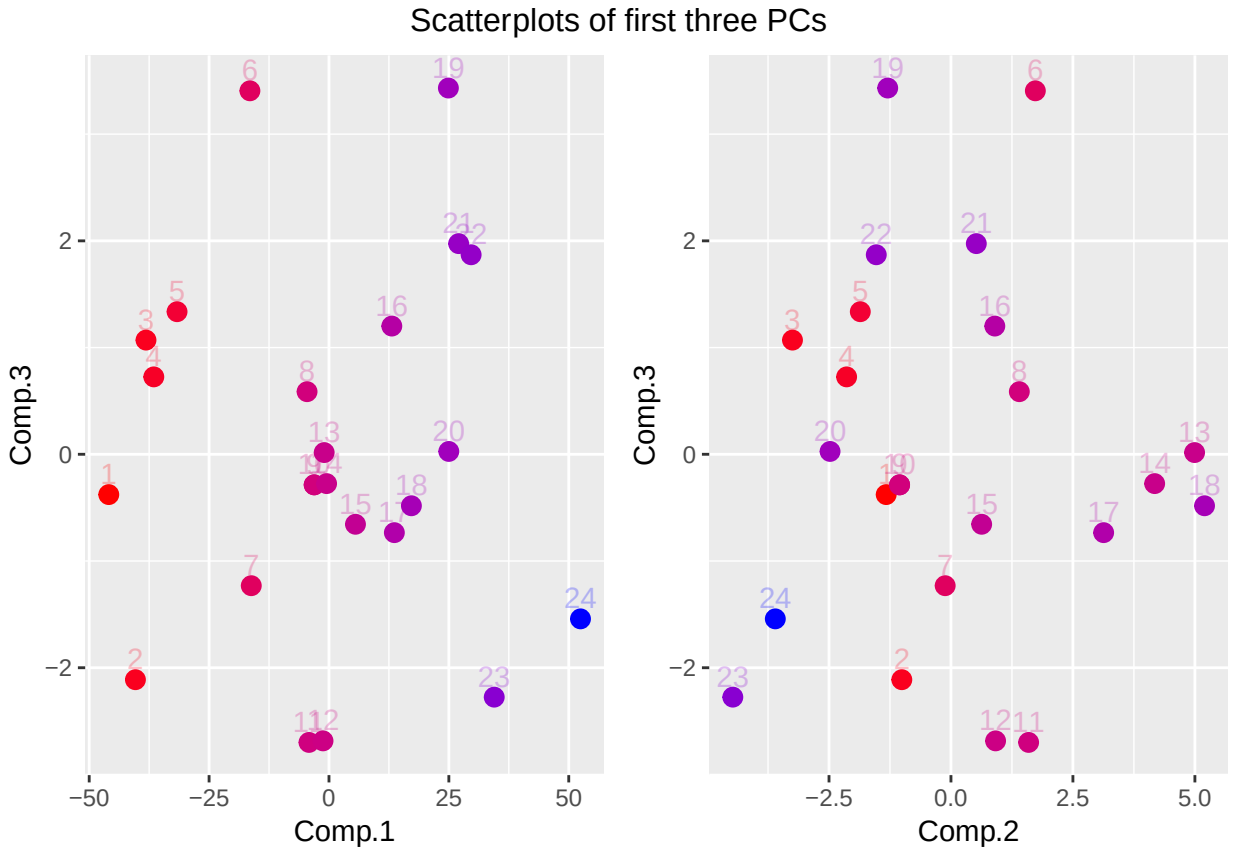
```
## jpeg
```

```
##     3
```

```
dev.off()
```

```
## pdf
##   2
```

```
library(gridExtra)
grid.arrange(grobs = list(p2, p3), nrow=1, top = "Scatterplots of first three PCs")
```




```
dev.copy(jpeg,filename="~/Desktop/jenn/teaching/ADA2/lecture notes/plots/ch13plot10.jpg")
```

```
## jpeg
##     3
```

```
dev.off()
```

```
## pdf
##   2
```

```
# perform PCA on correlation matrix
shells.pca <- princomp( ~ length + width + height, data = shells, cor = TRUE)
# standard deviation and proportion of variation for each component
summary(shells.pca)
```

```
## Importance of components:
##                           Comp.1    Comp.2      Comp.3
## Standard deviation     1.714584 0.1853043 0.160820482
## Proportion of Variance 0.979933 0.0114459 0.008621076
## Cumulative Proportion  0.979933 0.9913789 1.000000000
```

```r
# coefficients for PCs
loadings(shells.pca)
```

```
##
## Loadings:
##        Comp.1 Comp.2 Comp.3
## length -0.578 -0.137  0.804
## width  -0.577 -0.628 -0.522
## height -0.577  0.766 -0.284
##
##                Comp.1 Comp.2 Comp.3
## SS loadings     1.000  1.000  1.000
## Proportion Var  0.333  0.333  0.333
## Cumulative Var  0.333  0.667  1.000
```