Practice problems for PCA with partial answers

1. The purpose of this problem is to examine the effect that different correlations have on the outcome of the PCA. To make this easier, suppose **x** has a bivariate normal distribution with **μ** = (0, 0)′, σ11 = 1, and σ22 = 1. For σ12 = -0.99, -0.9, -0.5, 0, 0.5, 0.9, and 0.99 (remember that σ12 = ρ12 because the variances are equal to 1), complete the following:
   1. Simulate 1,000 observations from the bivariate normal where a seed number of 8128 is set right before each data simulation.

I will examine part of this problem by using ρ12 = 0.99 only.

> library(mvtnorm)

> mu <- c(0, 0)

> rho12 <- 0.99

> sigma <- matrix(data = c(1, rho12, rho12, 1), nrow = 2, ncol = 2, byrow = TRUE)

> P <- cov2cor(V = sigma)

> P

[,1] [,2]

[1,] 1.00 0.99

[2,] 0.99 1.00

> N <- 1000

> set.seed(8128)

> X <- rmvnorm(n = N, mean = mu, sigma = sigma)

> head(X)

[,1] [,2]

[1,] 0.08789969 0.1239771

[2,] 0.64654881 0.5685544

[3,] -1.51649613 -1.6042760

[4,] 0.13794791 0.2297071

[5,] -1.14282093 -1.0670453

[6,] 2.41497659 2.2771057

* 1. Use princomp() with cor = TRUE to find the estimated eigenvalues and eigenvectors from the correlation matrix.

> pca.save <- princomp(x = X, cor = TRUE, scores = FALSE)

> summary(pca.save, loadings = TRUE, cutoff = 0.0)

Importance of components:

Comp.1 Comp.2

Standard deviation 1.4105423 0.101834613

Proportion of Variance 0.9948149 0.005185144

Cumulative Proportion 0.9948149 1.000000000

Loadings:

Comp.1 Comp.2

[1,] 0.707 -0.707

[2,] 0.707 0.707

* 1. Interpret the PCs.
  2. How many PCs are necessary?

Only one – make sure you understand why this would make sense in the context of PCA!

* 1. Create separate scatter plots of the data and the PC scores, but use one overall x-axis and y-axis set of limits. Describe the relationship between these plots for each ρ12.

> pca.save$scale <- apply(X = X, MARGIN = 2, FUN = sd)

> score.save <- predict(pca.save, newdata = X)

> head(score.save)

Comp.1 Comp.2

[1,] 0.1846096 0.02391515

[2,] 0.9007901 -0.05776389

[3,] -2.1943678 -0.06367319

[4,] 0.2957979 0.06362483

[5,] -1.5441590 0.05285472

[6,] 3.3828429 -0.10136960

> dev.new(width = 12)

> par(mfrow = c(1,2)) #One row and two columns of plots

> par(pty = "s")

> common.limits <- c(min(score.save, X), max(score.save, X))

> plot(x = X[,1], y = X[,2], xlab = expression(x[1]), ylab = expression(x[2]), main

= "Original data", xlim = common.limits, ylim = common.limits, panel.first =

grid(col = "lightgray", lty = "dotted"))

> abline(h = 0)

> abline(v = 0)

> plot(x = score.save[,1], y = score.save[,2], xlab = "PC #1", ylab = "PC #2", main

= "Principal components", xlim = common.limits, ylim = common.limits,

panel.first = grid(col = "lightgray", lty = "dotted"))

> abline(h = 0)

> abline(v = 0)



It looks like the x and y-axes have been rotated so that all of the variability in the data in represented in only one dimension!

* 1. Relate your answers in c) – e) to the value of σ12.

1. Just as a reminder, the suggested readings for this class contain additional examples that may help you better understand PCA.