DSA 8020 R Session 11: Principle Components Analysis

Whitney

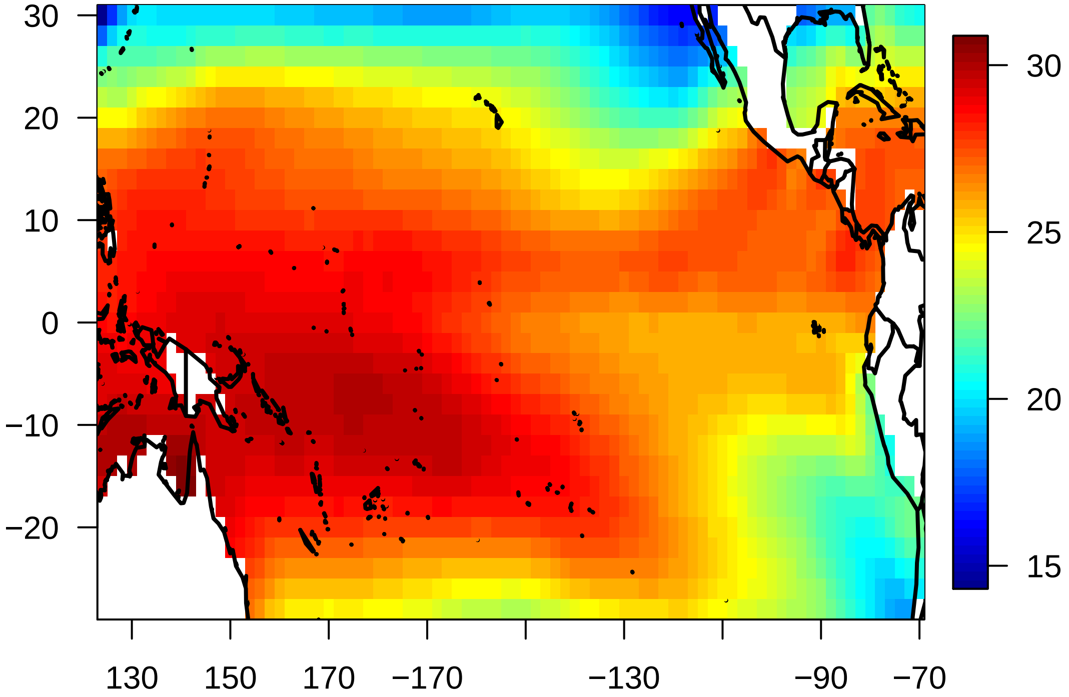
March 28, 2021

Contents

PCA: SST Example

Load and visualize the data

```r
load("SST1.rda")
library(fields)
library(maps)
par(las = 1, mar = c(3, 3, 1, 1))
image.plot(lon1, lat1, SST1[,, 1], xaxt = "n", xlab = "", ylab = "")
lon <- ifelse(lon1 <= 180, lon1, lon1 - 360)
axis(1, at = lon1[seq(4, 84, 10)], lon[seq(4, 84, 10)])
map("world2", add = TRUE, lwd = 2)
```



Compute the SST anomalies by subtracting means

```
t <- array(SST1, dim = c(84, 30, 12, 46))
SST_temp <- apply(t, 1:3, function(x) x - mean(x, na.rm = T))
# Change the data into lon-lat-month format
SST_anomalies <- array(dim = c(84, 30, 552))
for (i in 1:84){
  for (j in 1:30){
    SST_anomalies[i, j,] <- c(t(SST_temp[, i, j,]))
  }
}
```

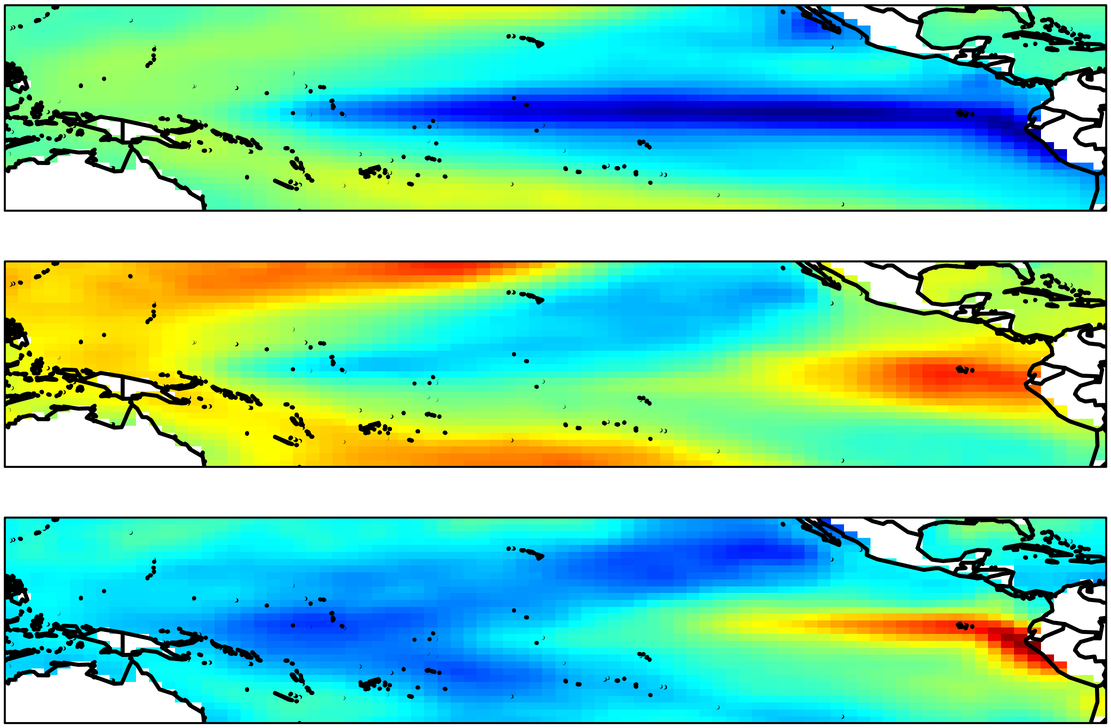
EOFs

```
# Extracting first three EOFs via singular value decomposition
temp <- array(SST_anomalies, c(84 * 30, 552))
ind <- is.na(temp[, 1])
temp <- temp[!ind, ]
temp2 <- svd(temp)
U1 <- matrix(NA, 84 * 30)
U1[!ind] <- temp2$u[, 1]; U1 <- matrix(U1, 84, 30)
U2 <- matrix(NA, 84 * 30)
U2[!ind] <- temp2$u[, 2]; U2 <- matrix(U2, 84, 30)
U3 <- matrix(NA, 84 * 30)
U3[!ind] <- temp2$u[, 3]; U3 <- matrix(U3, 84, 30)
zr <- range(c(U1, U2, U3), na.rm = TRUE)

set.panel(3, 1)
```

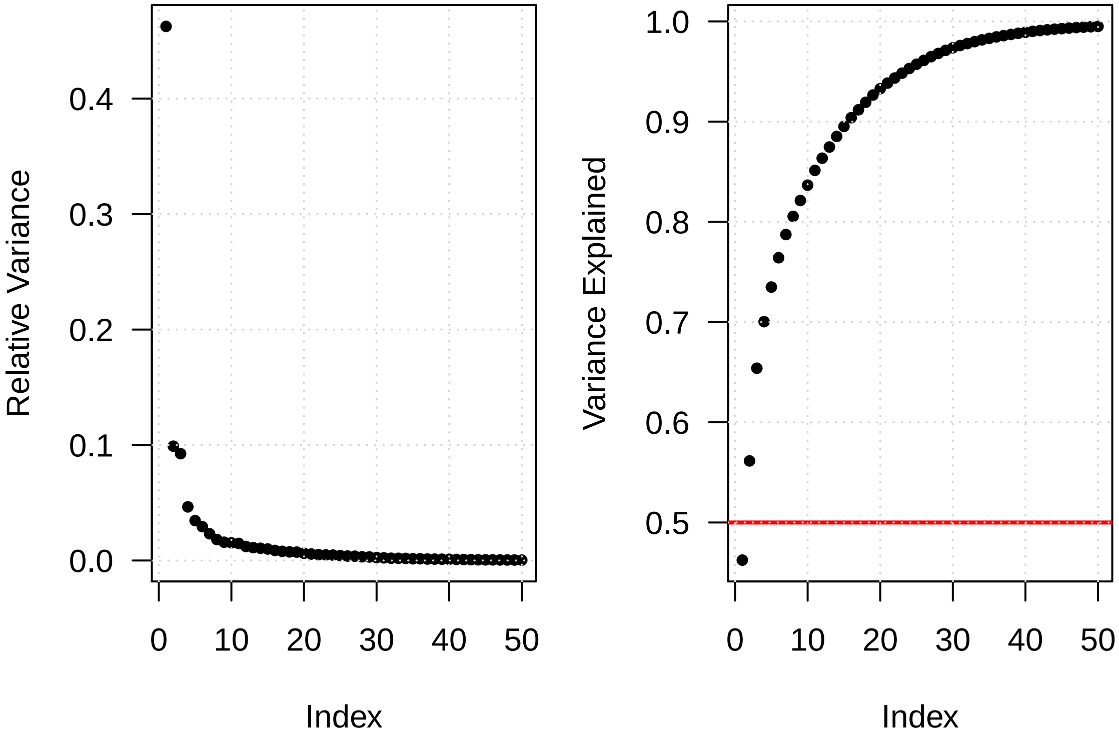
```
## plot window will lay out plots in a 3 by 1 matrix
```

```r
par(oma = c(0, 0, 0, 0))
ct <- tim.colors(256)
par(mar = c(1, 1, 1, 1))
image(lon1, lat1, U1, axes = FALSE, xlab = "", ylab = "", zlim = zr, col = ct)
map("world2", add = TRUE, lwd = 2)
box()
image(lon1, lat1, U2, axes = FALSE, xlab = "", ylab = "", zlim = zr, col = ct)
map("world2", add = TRUE, lwd = 2)
box()
image(lon1, lat1, U3, axes = FALSE, xlab = "", ylab = "", zlim = zr, col = ct)
map("world2", add = TRUE, lwd = 2)
box()
```



Screen plot

```r
par(mar = c(4, 4, 1, 1), mfrow = c(1, 2), las = 1)
dt <- ((temp2$d^2)/sum(temp2$d^2))
plot(1:50, dt[1:50], xlab = "Index", ylab = "Relative Variance",
     pch = 16, cex = 0.8)
grid()
dt <- (cumsum(temp2$d^2)/sum(temp2$d^2))
plot(1:50, dt[1:50], xlab = "Index", ylab = "Variance Explained",
     pch = 16, cex = 0.8)
yline(0.5, col = "red", lwd = 2)
grid()
```



1998 Jan El Ni~no Event

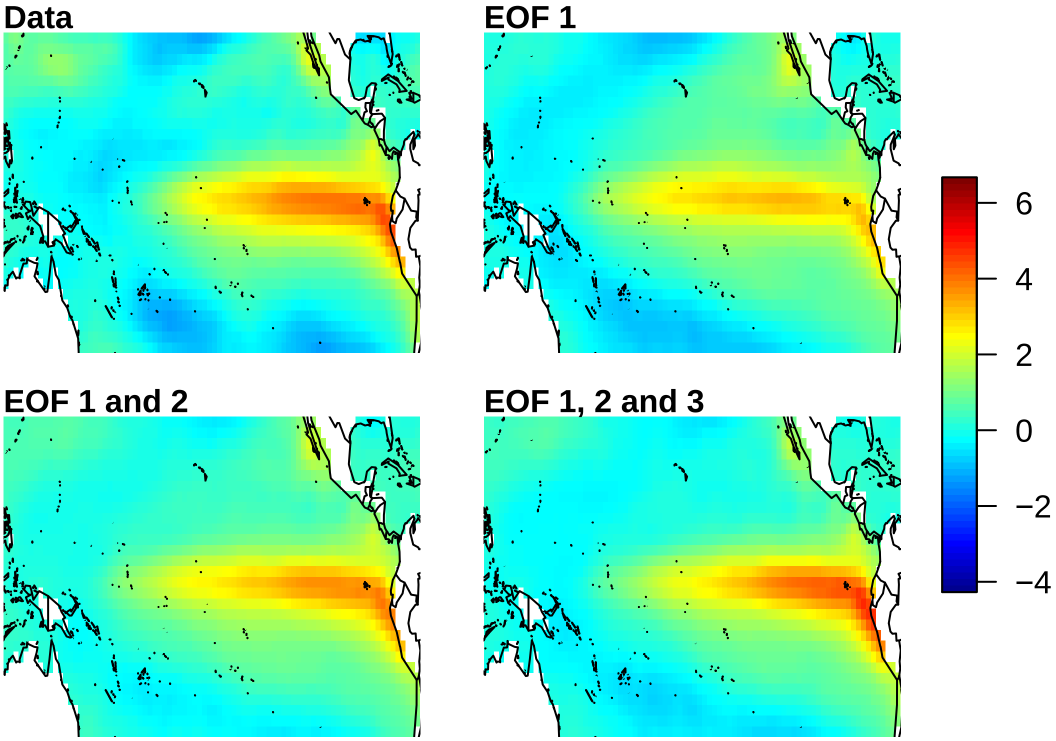
```
V <- temp2$v %*% diag(temp2$d)
J <- 337 # the index for 1998 Jan.
zr <- range(SST_anomalies, na.rm = TRUE)
set.panel(2, 2)
```

```
## plot window will lay out plots in a 2 by 2 matrix
```

```
par(mar = c(1, 1, 1, 1), oma = c(0, 0, 0, 6))
image(lon1, lat1, SST_anomalies[, , J], axes = FALSE, xlab = "", ylab = "",
      col = tim.colors(256), zlim = zr)
map("world2", add = TRUE)
title("Data", adj = 0)
image(lon1, lat1, V[J, 1] * U1, axes = FALSE, xlab = "", ylab = "",
      col = tim.colors(256), zlim = zr)
map("world2", add = TRUE)
title("EOF 1", adj = 0)
image(lon1, lat1, V[J, 1] * U1 + V[J, 2] * U2, axes = FALSE,
      xlab = "", ylab = "", col = tim.colors(256), zlim = zr)
map("world2", add = TRUE)
title("EOF 1 and 2", adj = 0)
image(lon1, lat1, V[J, 1] * U1 + V[J, 2] * U2 + V[J, 3] * U3,
      axes = FALSE, xlab = "", ylab = "", col = tim.colors(256),
      zlim = zr)
map("world2", add = TRUE)
title("EOF 1, 2 and 3", adj = 0)
set.panel()
```

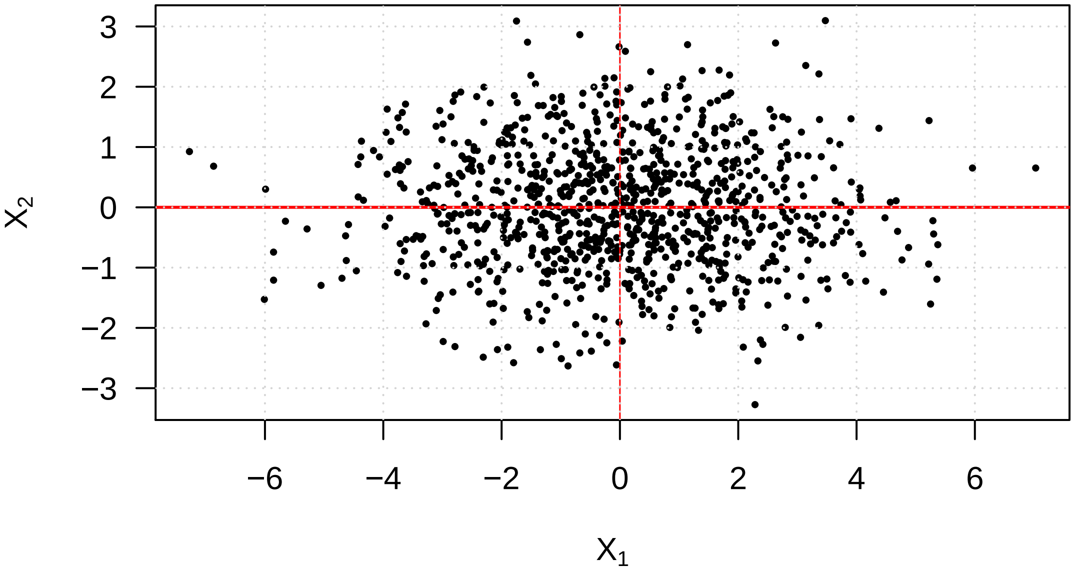
```
## plot window will lay out plots in a 1 by 1 matrix
```

```
par(oma = c(0, 0, 0, 0))
image.plot(legend.only = TRUE, zlim = zr, horizontal = FALSE, legend.shrink = 0.6)
```

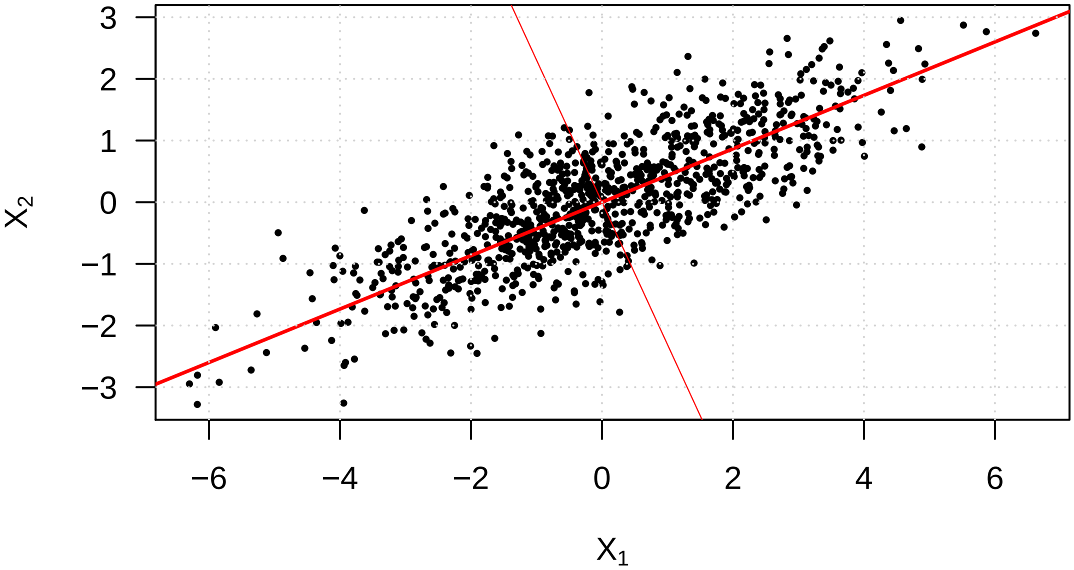


Toy Examples

```
library(MASS)
sim1 <- mvrnorm(n = 1000, mu = c(0, 0), Sigma = matrix(c(4, 0, 0, 1), 2, 2))
plot(sim1, pch = 16, cex = 0.5, las = 1,
     xlab = expression(X[1]),
     ylab = expression(X[2]))
abline(h = 0, col = "red", lwd = 1.5)
abline(v = 0, col = "red", lwd = 0.75)
grid()
```



```r
sim2 <- mvrnorm(n = 1000, mu = c(0, 0), Sigma = matrix(c(4, 1.6, 1.6, 1), 2, 2))
plot(sim2, pch = 16, cex = 0.5, las = 1,
     xlab = expression(X[1]),
     ylab = expression(X[2]))
abline(0, .4332, col = "red", lwd = 1.8)
abline(0, -2.308, col = "red", lwd = 0.6)
grid()
```



Principal Component Regression

Longley's Economic Regression Data

Longley's Economic data set provides a well-known example for a highly collinear regression.

Performing a linear regression

```
data(longley)
head(longley)
```

```
##      GNP.deflator     GNP Unemployed Armed.Forces Population Year Employed
## 1947         83.0 234.289      235.6        159.0    107.608 1947   60.323
## 1948         88.5 259.426      232.5        145.6    108.632 1948   61.122
## 1949         88.2 258.054      368.2        161.6    109.773 1949   60.171
## 1950         89.5 284.599      335.1        165.0    110.929 1950   61.187
## 1951         96.2 328.975      209.9        309.9    112.075 1951   63.221
## 1952         98.1 346.999      193.2        359.4    113.270 1952   63.639
```

```
round(cor(longley[, -7]), 3)
```

```
##              GNP.deflator   GNP Unemployed Armed.Forces Population  Year
## GNP.deflator        1.000 0.992      0.621        0.465      0.979 0.991
## GNP                 0.992 1.000      0.604        0.446      0.991 0.995
## Unemployed          0.621 0.604      1.000       -0.177      0.687 0.668
## Armed.Forces        0.465 0.446     -0.177        1.000      0.364 0.417
## Population          0.979 0.991      0.687        0.364      1.000 0.994
## Year                0.991 0.995      0.668        0.417      0.994 1.000
```

```
library(faraway)
```

```
##
## Attaching package: 'faraway'

## The following object is masked from 'package:maps':
##
##     ozone
```

```
vif(longley[, -7])
```

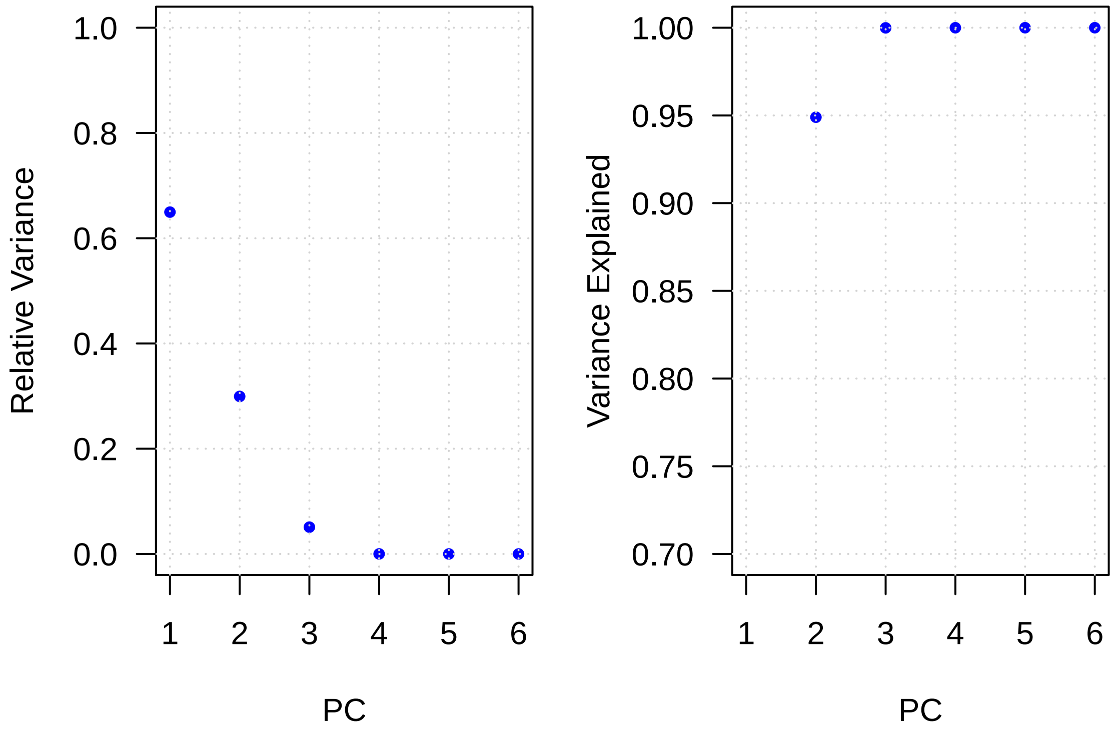
```
## GNP.deflator          GNP   Unemployed Armed.Forces   Population         Year
##    135.53244   1788.51348     33.61889      3.58893    399.15102    758.98060
```

```
lm <- lm(Employed ~ ., data = longley)
summary(lm)
```

```
##
## Call:
## lm(formula = Employed ~ ., data = longley)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.41011 -0.15767 -0.02816  0.10155  0.45539
##
## Coefficients:
##                Estimate Std. Error t value Pr(>|t|)
## (Intercept)  -3.482e+03  8.904e+02  -3.911 0.003560 **
## GNP.deflator  1.506e-02  8.492e-02   0.177 0.863141
## GNP          -3.582e-02  3.349e-02  -1.070 0.312681
## Unemployed   -2.020e-02  4.884e-03  -4.136 0.002535 **
## Armed.Forces -1.033e-02  2.143e-03  -4.822 0.000944 ***
## Population   -5.110e-02  2.261e-01  -0.226 0.826212
## Year          1.829e+00  4.555e-01   4.016 0.003037 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.3049 on 9 degrees of freedom
## Multiple R-squared:  0.9955, Adjusted R-squared:  0.9925
## F-statistic: 330.3 on 6 and 9 DF,  p-value: 4.984e-10
```

Performing Principal Component Regression

```
longley.pca <- prcomp(longley[, -7], center = TRUE)
vars <- longley.pca$sdev^2
# Scrren plot
par(mfrow = c(1, 2), mar = c(4.1, 4.1, 1.1, 1.1))
plot(1:6, vars/sum(vars),
     xlab = "PC", ylim = c(0, 1),
     ylab = "Relative Variance",
     pch = 16, cex = 0.8, las = 1,
     col = "blue")
grid()
plot(1:6, cumsum(vars)/sum(vars),
     xlab = "PC", ylim = c(0.7, 1),
     ylab = "Variance Explained",
     pch = 16, cex = 0.8, las = 1,
     col = "blue")
grid()
```



```
# Performing PCR
library(pls)
pcrFit <- pcr(Employed ~ ., data = longley, valdiation = "cv")
summary(pcrFit)
```

```
## Data:    X dimension: 16 6
##  Y dimension: 16 1
## Fit method: svdpc
## Number of components considered: 6
## TRAINING: % variance explained
##           1 comps  2 comps  3 comps  4 comps  5 comps  6 comps
## X           64.96    94.90    99.99   100.00   100.00   100.00
## Employed    78.42    89.73    98.51    98.56    98.83    99.55
```