Assignment3

Stephen Giang

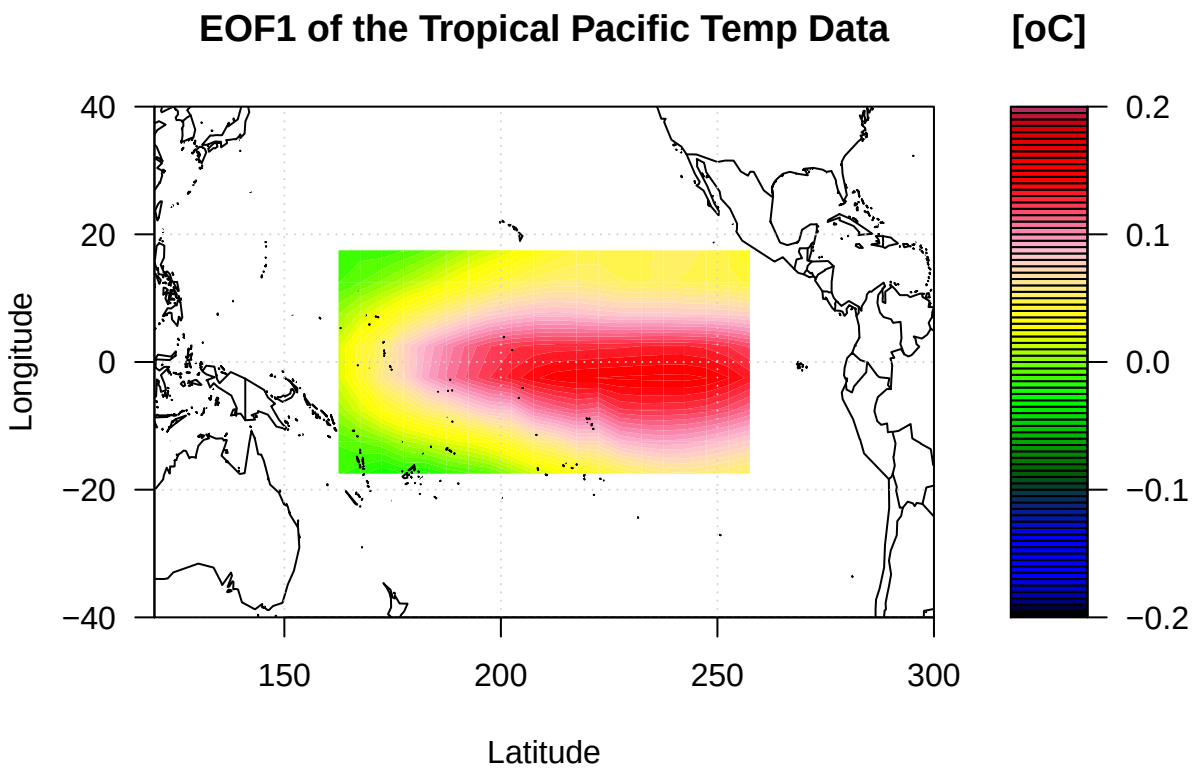
11/20/2020

```
# Problem 1-2

setwd('C:/Users/Stephen Giang/Documents/Math336Files/data')
readData = read.csv('NOAAGlobalT.csv')

pacific1 = subset(readData, LAT >= -20 & LAT <= 20)   #20S  - 20N
pacific1 = subset(pacific1, LON >= 160 & LON <= 260)  #160E - 100W
pacific1 = pacific1[, 856:1455]   # 01/1951 - 12/2000

# -999.9 means missing data so set to 0
for ( i in 1:dim(pacific1)[1] ) {
  for ( j in 1:dim(pacific1)[2] ) {
    if (pacific1[i,j] < -800) {
      pacific1[i,j] = 0
    }
  }
}

yearDiff = 1999 - 1951 + 1
pacific = matrix(0,nrow = dim(pacific1)[1], ncol = yearDiff)
# Annual (July - June) Mean Sea Temp
for (k in 1:yearDiff) {
  pacific[, k] = rowMeans(pacific1[, (12*k - 5) : (12*k + 6) ])
}

svdPacific = svd(pacific)
D = diag(svdPacific$d)
U = svdPacific$u
V = svdPacific$v

eigVals = (svdPacific$d[1:10])^2 / yearDiff
eigVals
```

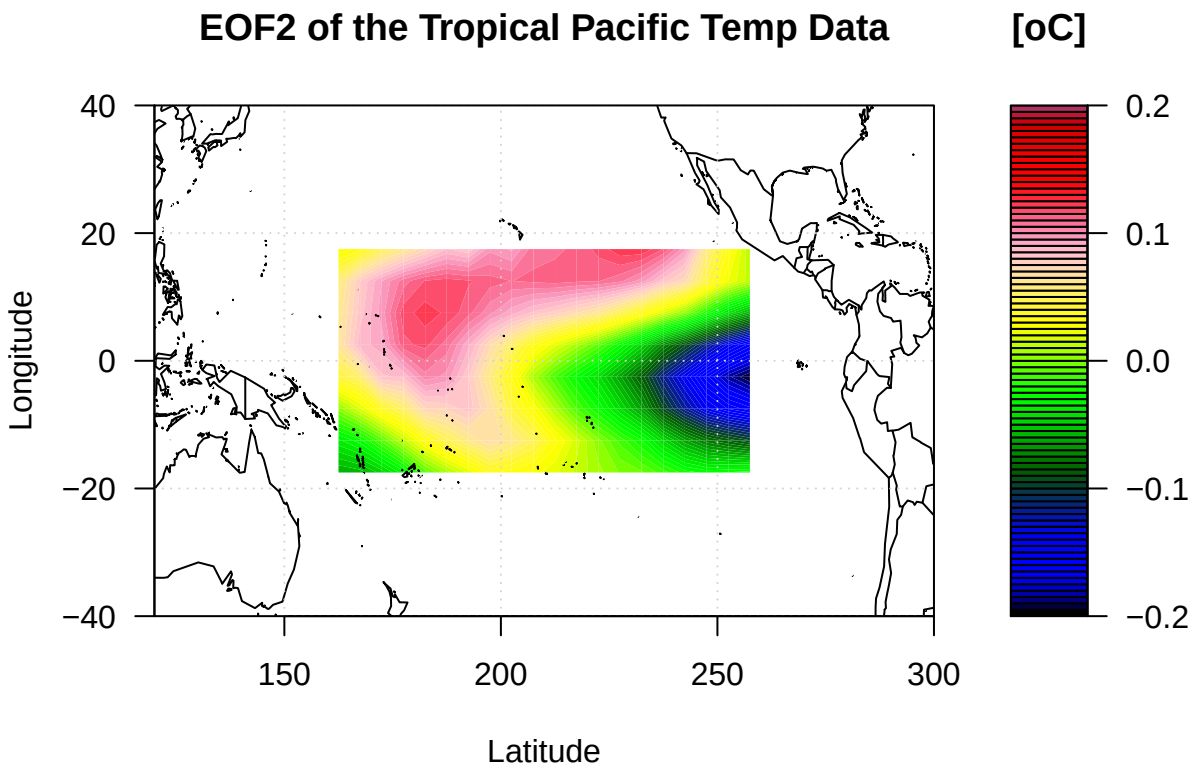
```
##  [1] 27.7879120  4.3773517  2.5610076  0.9341387  0.6615021  0.4678065
##  [7]  0.3243314  0.2490007  0.1680932  0.1443271
```

```
x = seq(-17.5, 17.5, by=5)  # lat
y = seq(162.5, 257.5, by=5) # lon
numLatVals = length(x)
numLonVals = length(y)
time = 1951 : 1999

int=seq(-0.2,0.2,length.out=81)
```

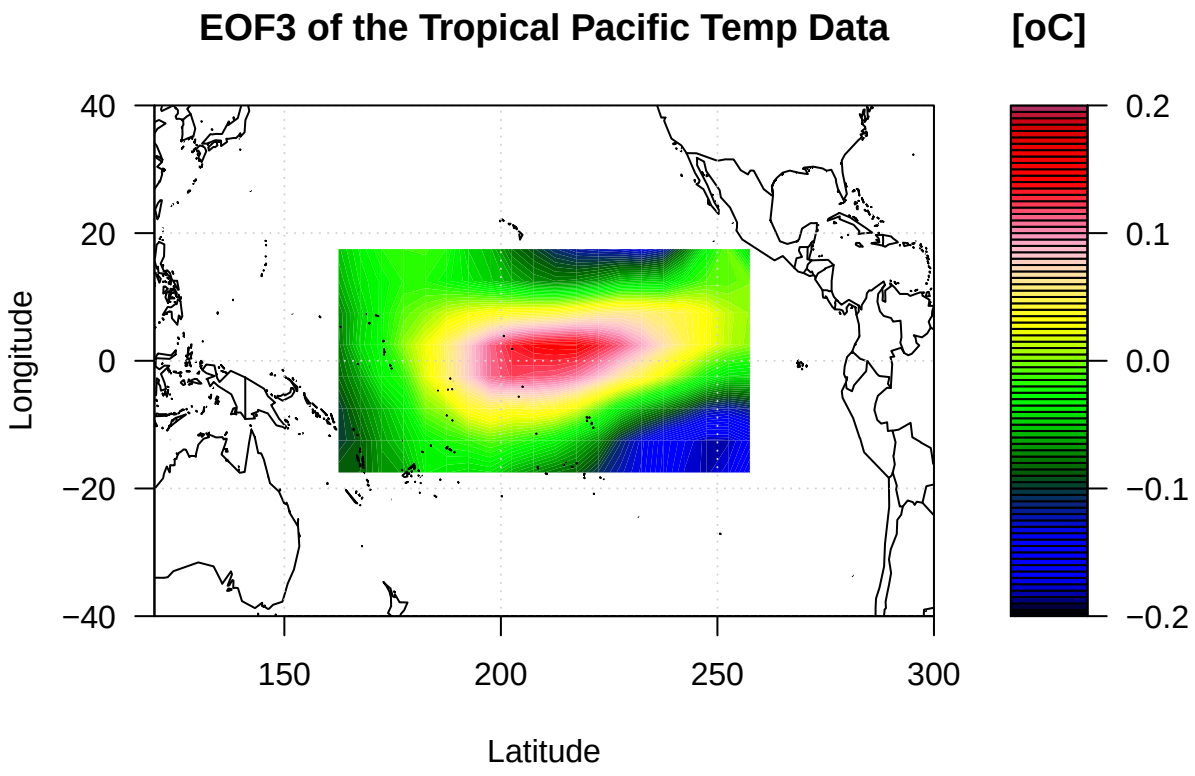
```
rgb.palette=colorRampPalette(c('black','blue', 'darkgreen',
                               'green', 'yellow','pink','red','maroon'),
                             interpolate='spline')
suppressWarnings(library(maps))

umat = matrix(U[,1], nrow = numLonVals)
filled.contour(y, x, umat, color.palette=rgb.palette, levels=int,
               xlim=c(120,300),ylim=c(-40,40),
               plot.title=title(main="EOF1 of the Tropical Pacific Temp Data",
                                xlab="Latitude",ylab="Longitude"),
               plot.axes={axis(1); axis(2); map('world2', add=TRUE); grid()},
               key.title=title(main="[oC]"))
```



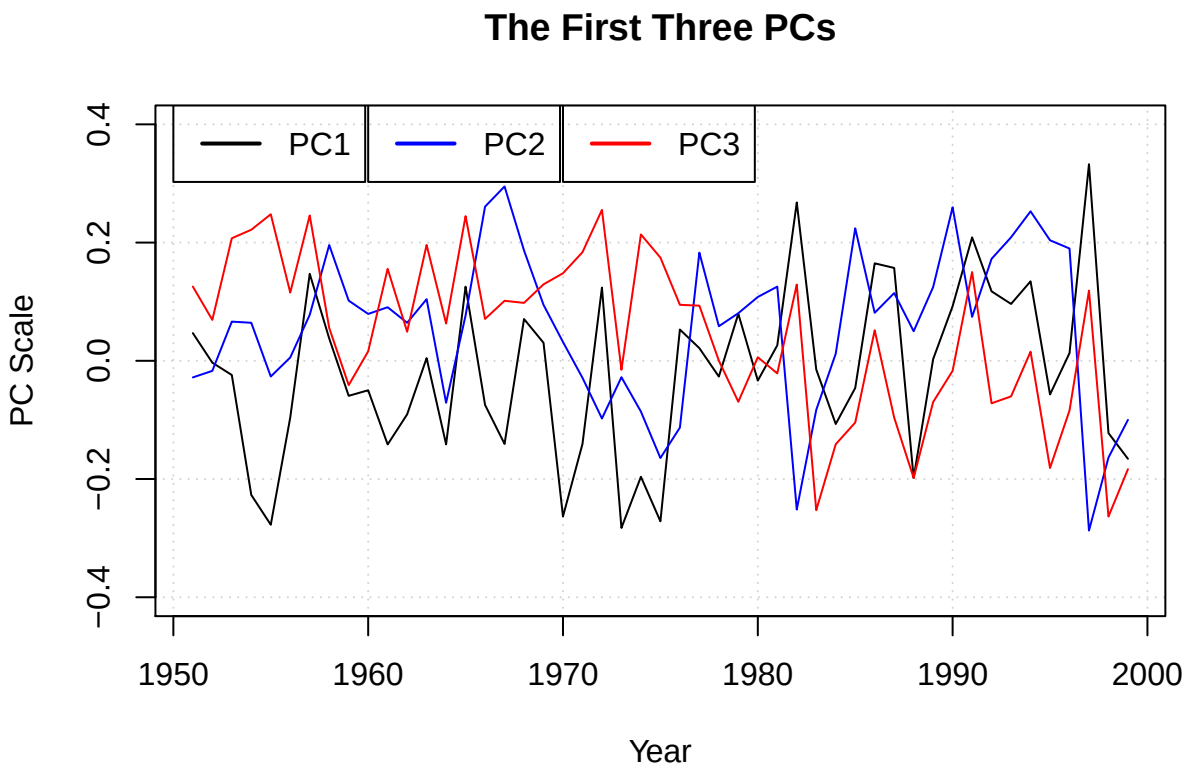

```
umat = matrix(U[,2], nrow = numLonVals)
filled.contour(y, x, umat, color.palette=rgb.palette, levels=int,
               xlim=c(120,300),ylim=c(-40,40),
               plot.title=title(main="EOF2 of the Tropical Pacific Temp Data",
                                xlab="Latitude",ylab="Longitude"),
               plot.axes={axis(1); axis(2); map('world2', add=TRUE); grid()},
               key.title=title(main="[oC]"))
```




```
umat = matrix(U[,3], nrow = numLonVals)
filled.contour(y, x, umat, color.palette=rgb.palette, levels=int,
               xlim=c(120,300),ylim=c(-40,40),
               plot.title=title(main="EOF3 of the Tropical Pacific Temp Data",
                                xlab="Latitude",ylab="Longitude"),
               plot.axes={axis(1); axis(2); map('world2', add=TRUE); grid()},
               key.title=title(main="[oC]"))
```




```
plot(time, V[,1], 'l', col = 'black',
     xlab = 'Year', ylab = 'PC Scale', ylim=c(-0.4,0.4),
     main = 'The First Three PCs', panel.first=grid())
lines(time, V[,2], type="l", col="blue")
lines(time, V[,3], type="l", col="red")


legend(1950,0.4325, 'PC1',lwd=2 )
legend(1960,0.4325, 'PC2',lwd=2, col="blue")
legend(1970,0.4325, 'PC3',lwd=2, col="red")
```

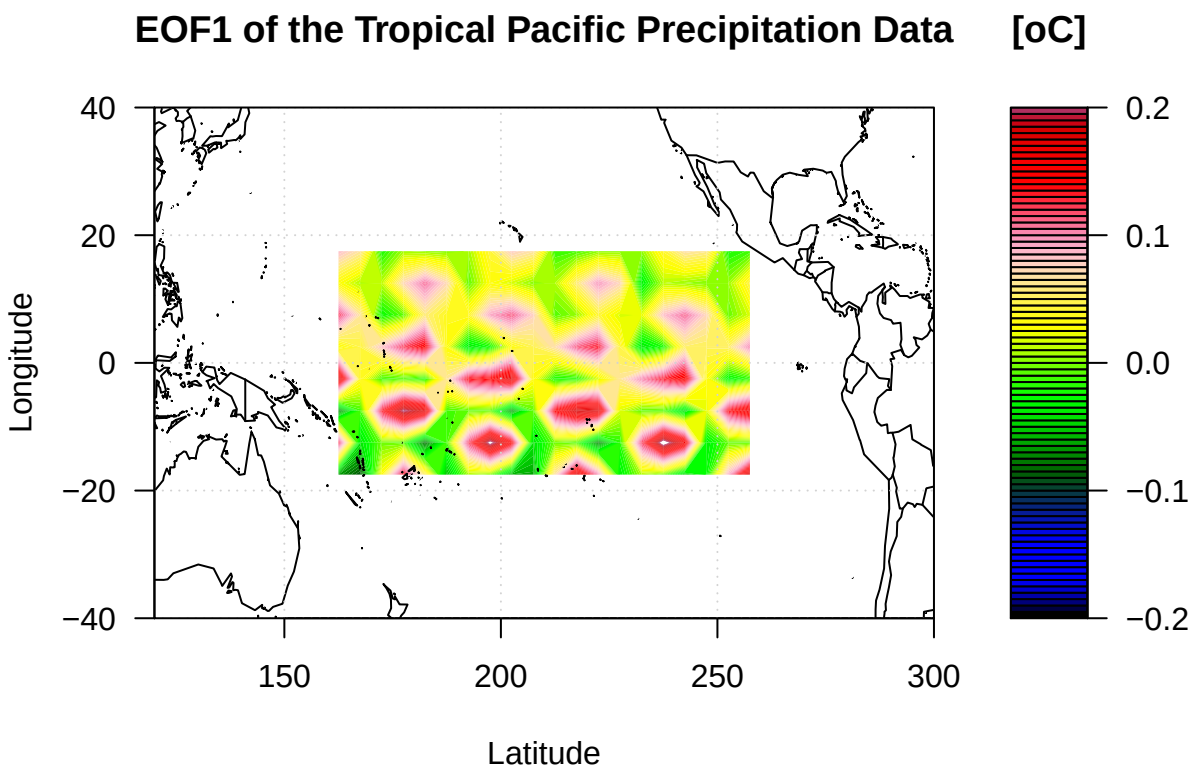



```
# Problem 3-4

setwd('C:/Users/Stephen Giang/Documents/Math336Files')
readData = read.csv('PrcpRecon5degAnn.csv')

pacific = subset(readData, Lat >= -20 & Lat <= 20)   #20S  - 20N
pacific = subset(pacific, Lon >= 160 & Lon <= 260)  #160E - 100W
pacific = pacific[, 54:102] #1951 - 1999

yearDiff = 1999 - 1951 + 1
svdPacific = svd(pacific)
D = diag(svdPacific$d)
U = svdPacific$u
V = svdPacific$v

eigVals = (svdPacific$d[1:10])^2 / yearDiff
eigVals
```

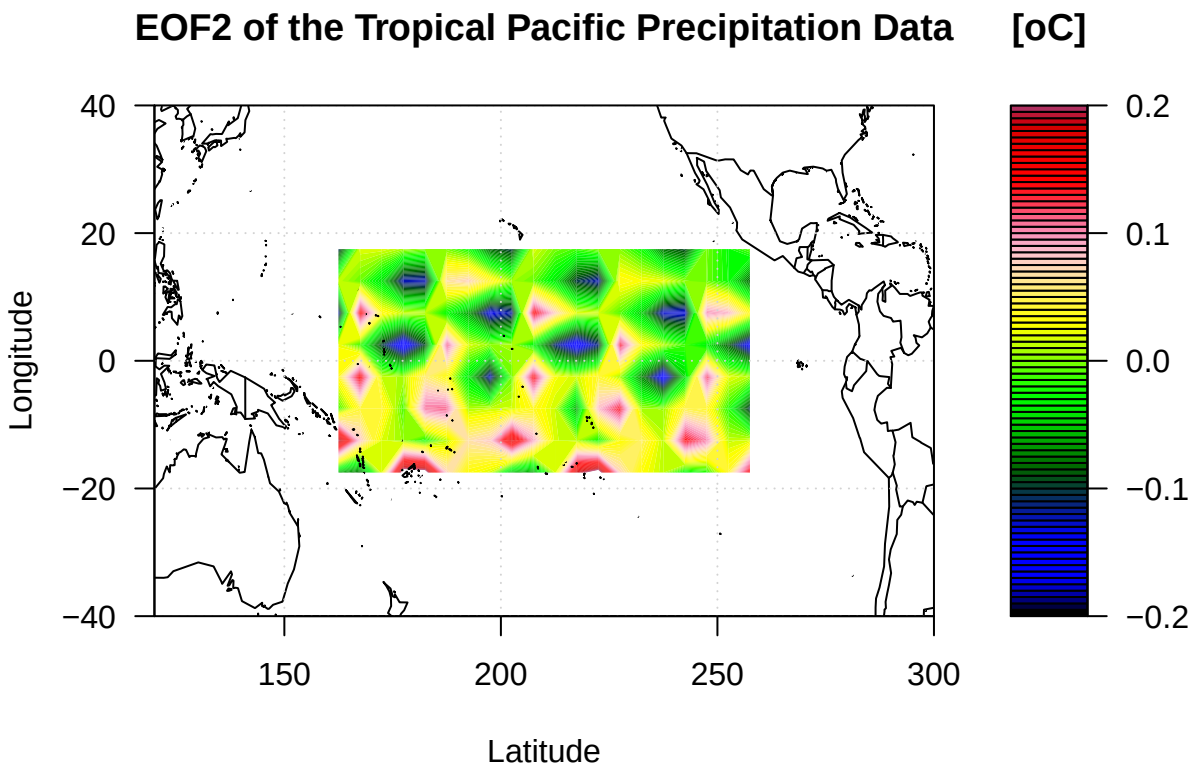
```
##  [1] 105.4814490  22.2559501   5.4832410   3.3866133   2.8810880   2.1684325
##  [7]   1.4404727   1.2810827   0.9563928   0.7468040
```

```
x = seq(-17.5, 17.5, by=5)  # lat
y = seq(162.5, 257.5, by=5) # lon
numLatVals = length(x)
numLonVals = length(y)
time = 1951 : 1999
```

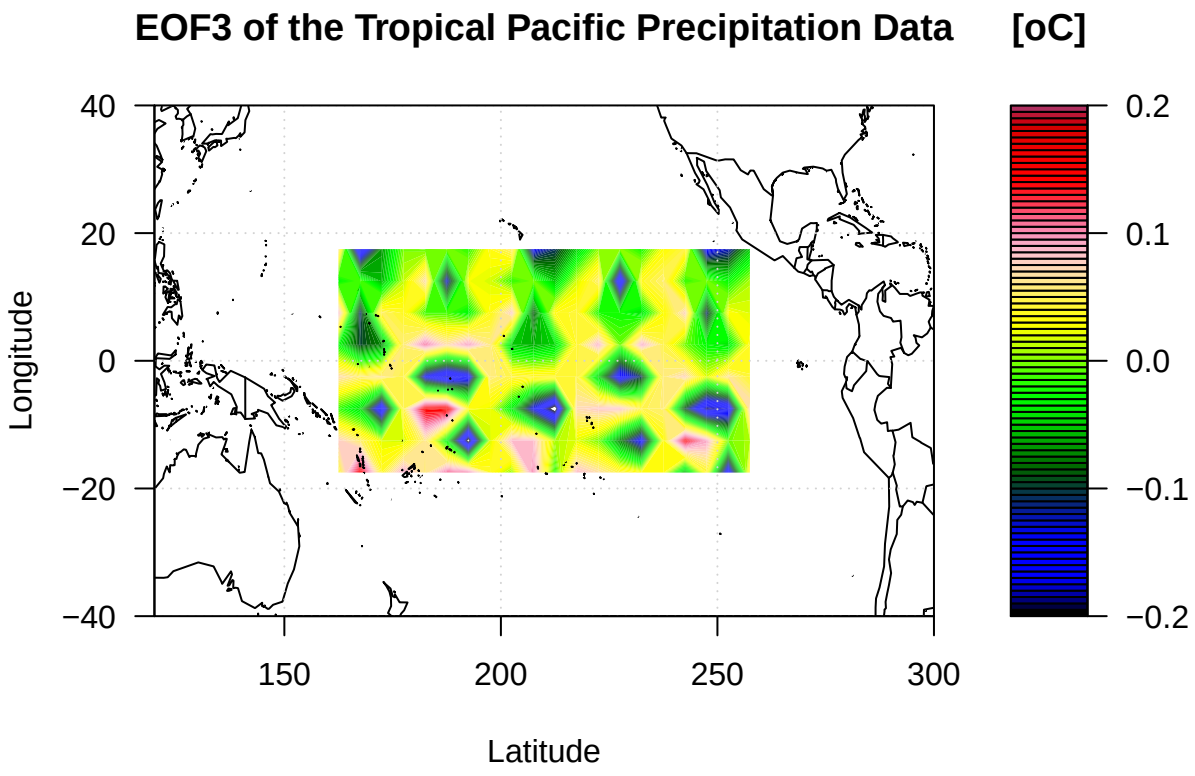
```
int=seq(-0.2,0.2,length.out=81)
rgb.palette=colorRampPalette(c('black','blue', 'darkgreen',
                               'green', 'yellow','pink','red','maroon'),
                             interpolate='spline')
suppressWarnings(library(maps))

umat = matrix(U[,1], nrow = numLonVals)
filled.contour(y, x, umat, color.palette=rgb.palette, levels=int,
               xlim=c(120,300),ylim=c(-40,40),
               plot.title=title(main="EOF1 of the Tropical Pacific Precipitation Data",
                                xlab="Latitude",ylab="Longitude"),
               plot.axes={axis(1); axis(2); map('world2', add=TRUE); grid()},
               key.title=title(main="[oC]"))
```



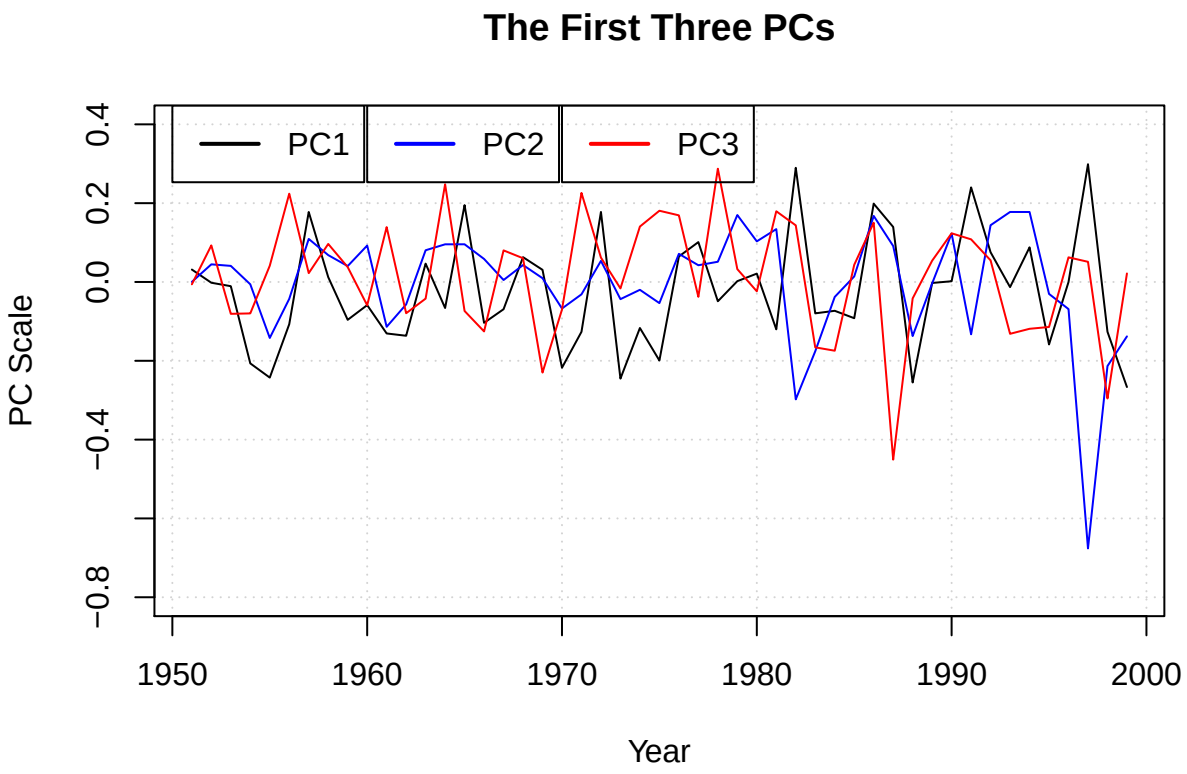

```
umat = matrix(U[,2], nrow = numLonVals)
filled.contour(y, x, umat, color.palette=rgb.palette, levels=int,
               xlim=c(120,300),ylim=c(-40,40),
               plot.title=title(main="EOF2 of the Tropical Pacific Precipitation Data",
                                xlab="Latitude",ylab="Longitude"),
               plot.axes={axis(1); axis(2); map('world2', add=TRUE); grid()},
               key.title=title(main="[oC]"))
```




```
umat = matrix(U[,3], nrow = numLonVals)
filled.contour(y, x, umat, color.palette=rgb.palette, levels=int,
               xlim=c(120,300),ylim=c(-40,40),
               plot.title=title(main="EOF3 of the Tropical Pacific Precipitation Data",
                                xlab="Latitude",ylab="Longitude"),
               plot.axes={axis(1); axis(2); map('world2', add=TRUE); grid()},
               key.title=title(main="[oC]"))
```




```
plot(time, V[,1], 'l', col = 'black',
     xlab = 'Year', ylab = 'PC Scale', ylim=c(-0.8,0.4),
     main = 'The First Three PCs', panel.first=grid())
lines(time, V[,2], type="l", col="blue")
lines(time, V[,3], type="l", col="red")


legend(1950,0.448, 'PC1',lwd=2 )
legend(1960,0.448, 'PC2',lwd=2, col="blue")
legend(1970,0.448, 'PC3',lwd=2, col="red")
```




```
# Problem 8

BuffonLongSim = function(d, l, n = 10000) {
  k = 0
  for (i in 1 : n) {
    y = runif(1, min = 1, max = d)
    theta = runif(1, min = -pi /2, max = pi/2)
    if (y + l*cos(theta) >= d) {
      k = k + 1
    }
  }
  return(k / n)
}

BuffonLongExact = function(d, l) {
  dl = d/ l
  ld = l / d
  twopi = 2 / pi
  return( twopi * ( acos(dl) + ld - sqrt( (ld)^2 - 1) ) )
}


d = floor(runif(1, min = 1, max = 100))
l = floor(runif(1, min = 1, max = 100)) + d   # l > d

BuffonLongSim(d, l, 10000)
```

```
## [1] 0.9405
```

```
BuffonLongExact(d, l)
```

```
## [1] 0.9327365
```

```
# Problem 9

MCSim = function(dim, n = 10000) {
  x = matrix(runif(dim*n, min= -1, max = 1), ncol = dim)
  k = 0
  for (i in 1 : n) {
    if ( (t(x[i,]) %*% x[i,]) < 1) {
      k = k + 1
    }
  }
  return( (k/n) * 2^dim )
}

MCExact = function(n,R=1) {
  numer = pi^(n/2)
  denom = gamma((n/2) + 1)
  return((numer/denom)*(R^n))
}


MCSim(8, 100000)
```

```
## [1] 3.98848
```

```
MCExact(8)
```

```
## [1] 4.058712
```

```
# Problem 10


diceRollSim = function (n) {
  k = 0
  for (i in 1 : n) {
    a = floor(runif(1, min = 1, max = 6))
    b = floor(runif(1, min = 1, max = 6))
    if (a + b == 7) {
      k = k + 1
    }
  }
  return(k / n)
}

diceRollExact = function () {
  k = 0
  for (i in 1 : 6) {
    for (j in 1 : 6) {
      if (i + j == 7) {
        k = k + 1
      }
    }
```

```
  }
  return(k / 36)
}

diceRollSim(1000)
```

```
## [1] 0.155
```

```
diceRollExact()
```

```
## [1] 0.1666667
```