Homework 3
Intro Math Modeling
Math 336
Stephen Giang RedID: 823184070

Problem 1-2 Exercise (5.8): Make an SVD space-time data decomposition for the 5° × 5° latitude-longitude gridded annual (July-June) mean sea surface temperature field anomalies from 1951-2000 over the tropical Pacific: $(20°S - 20°N, 160°E - 100°W)$. The dataset is posted on the course blackboard.

(a) Perform the SVD analysis for the data. Print out the first 10 eigenvalues.

Notice the first 10 eigenvalues can be found by $\lambda_i = \frac{d_i^2}{t}$

27.7879120	4.3773517	2.5610076	0.9341387	0.6615021
0.4678065	0.3243314	0.2490007	0.1680932	0.1443271

```
# 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
```

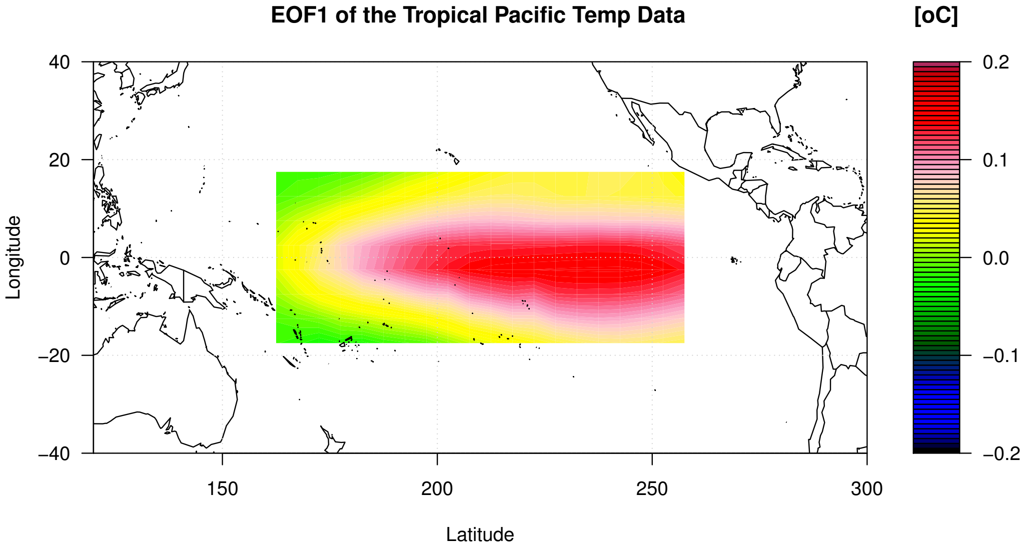
```
##  [1] 27.7879120  4.3773517  2.5610076  0.9341387  0.6615021
##  [6]  0.4678065  0.3243314  0.2490007  0.1680932  0.1443271
```

(b) Plot the map of the first three U column vectors, which are the spatial patterns of the data field, and are also known as Empirical Orthogonal Functions (EOFs).

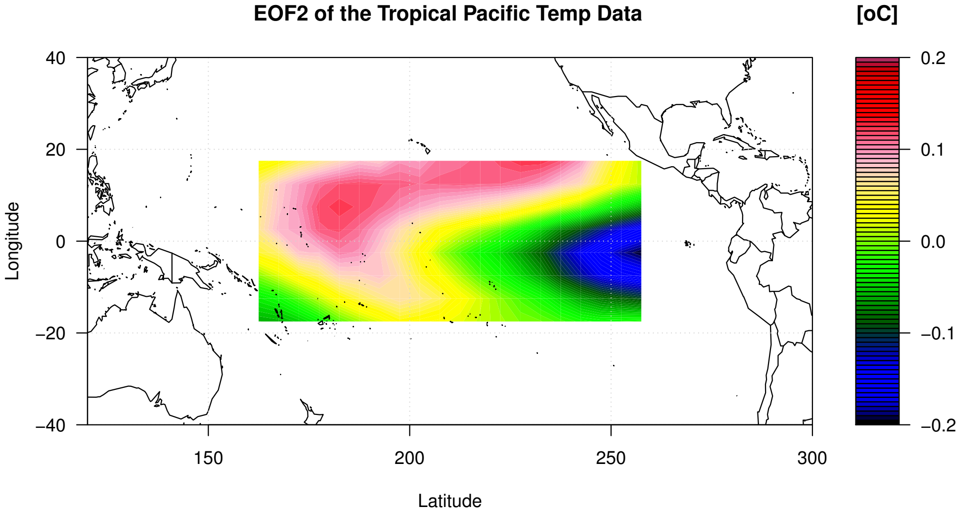
```
x = seq(-17.5, 17.5, by=5)  # lat
y = seq(162.5, 257.5, by=5) # lon
numLatVals = length(x)
numLonVals = length(y)
time = 1951 : 1999

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))
```

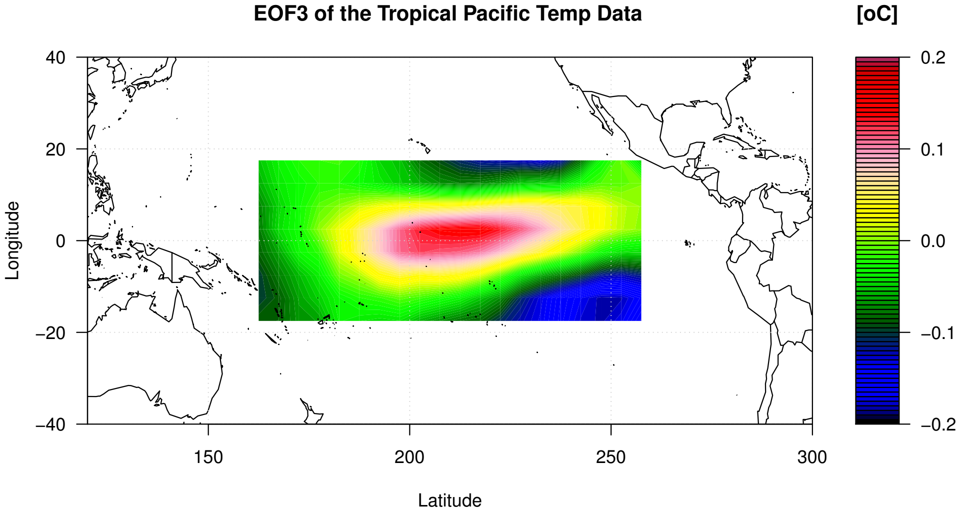
```
umat = matrix(U[,1], nrow = numLonVals)
filled.contour(y, x, umat, color.palette=rgb.palette, levels=int,
               xlim=c(120,300),ylim=c(-40,40),
               plot.title=title(main="EOF1 of the Tropical Pacific Temp Data",
                                xlab="Latitude",ylab="Longitude"),
               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]"))
```



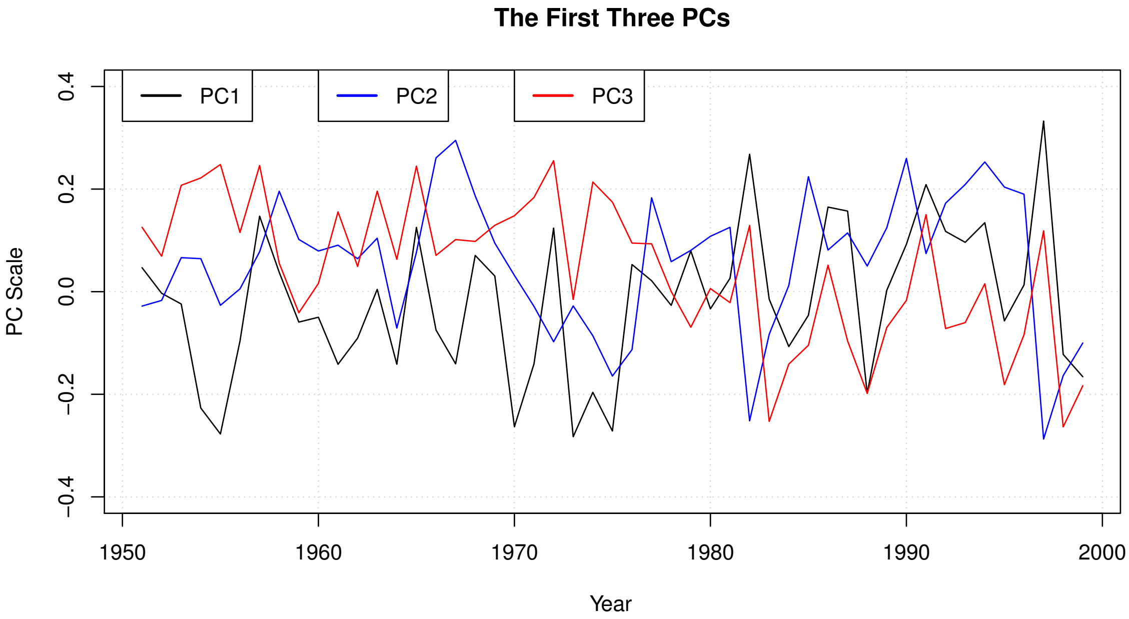
```
umat = matrix(U[,3], nrow = numLonVals)
filled.contour(y, x, umat, color.palette=rgb.palette, levels=int,
               xlim=c(120,300),ylim=c(-40,40),
               plot.title=title(main="EOF3 of the Tropical Pacific Temp Data",
                                xlab="Latitude",ylab="Longitude"),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



(c) Plot the time series of the first three V column vectors, which are the temporal patterns of the data field, and are also known as Principal Components (PCs).

```
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")
```



(d) El Nino signals should show in the figures of Steps (b) and (c). Make a brief description of the El Ninos.

You can see from the EOF's and PC's that the temperature spikes in the same location and the same time in the pacific ocean consistently. These spikes are consistent with the data that we have on the El Ninos. The El Ninos spike in temperature in the pacific ocean at 200° Latitude on the equator and around the summer time. During these El Ninos as well, we can see that the wind reverses to become western winds instead of eastern winds.

Problem 3-4 Exercise (5.9): Make an SVD data representation analysis for the $5^\circ \times 5^\circ$ latitude-longitude gridded annual (July-June) precipitation field from 1951-2000 over the tropical Pacific: $(20^\circ S - 20^\circ N, 160^\circ E - 100^\circ W)$.

(a) Perform the SVD analysis for the data. Print out the first 10 eigenvalues.

Notice the first 10 eigenvalues can be found by $\lambda_i = \frac{d_i^2}{t}$

105.4814490	22.2559501	5.4832410	3.3866133	2.8810880
2.1684325	1.4404727	1.2810827	0.9563928	0.7468040

```
#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
```

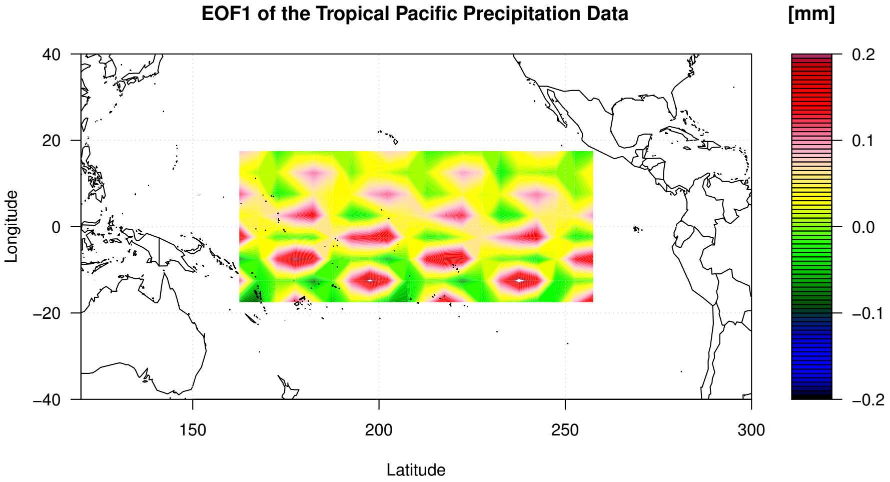
```
##  [1] 105.4814490  22.2559501   5.4832410   3.3866133   2.8810880
##  [6]   2.1684325   1.4404727   1.2810827   0.9563928   0.7468040
```

(b) Plot the map of the first three U column vectors, which are the spatial patterns of the data field, and are also known as Empirical Orthogonal Functions (EOFs).

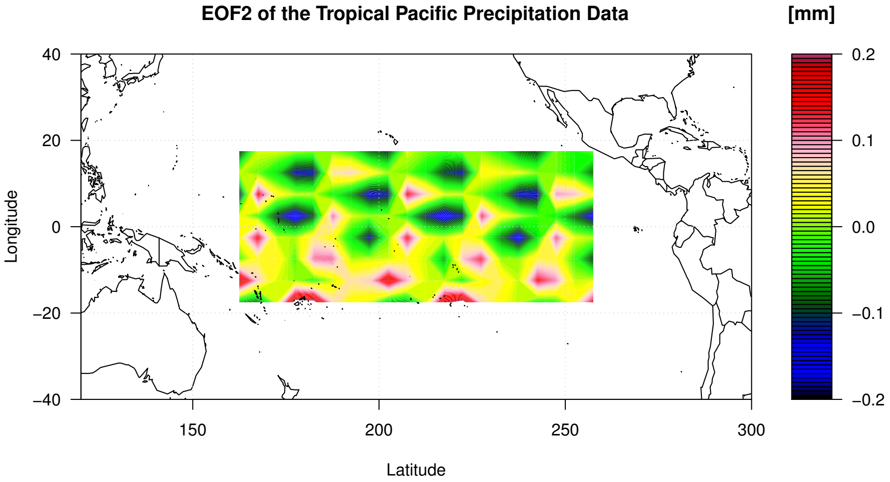
```
x = seq(-17.5, 17.5, by=5)  # lat
y = seq(162.5, 257.5, by=5) # lon
numLatVals = length(x)
numLonVals = length(y)
time = 1951 : 1999

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))
```

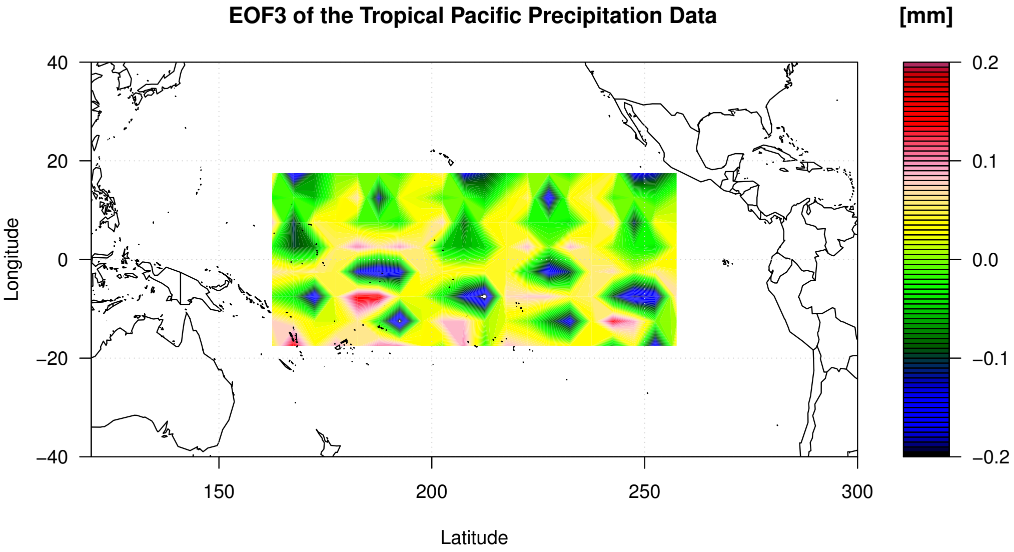
```
umat = matrix(U[,1], nrow = numLonVals)
filled.contour(y, x, umat, color.palette=rgb.palette, levels=int,
               xlim=c(120,300),ylim=c(-40,40),
               plot.title=title(main="EOF1 of the Tropical Pacific
                                      Precipitation Data",
                 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



```
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="[mm]"))
```



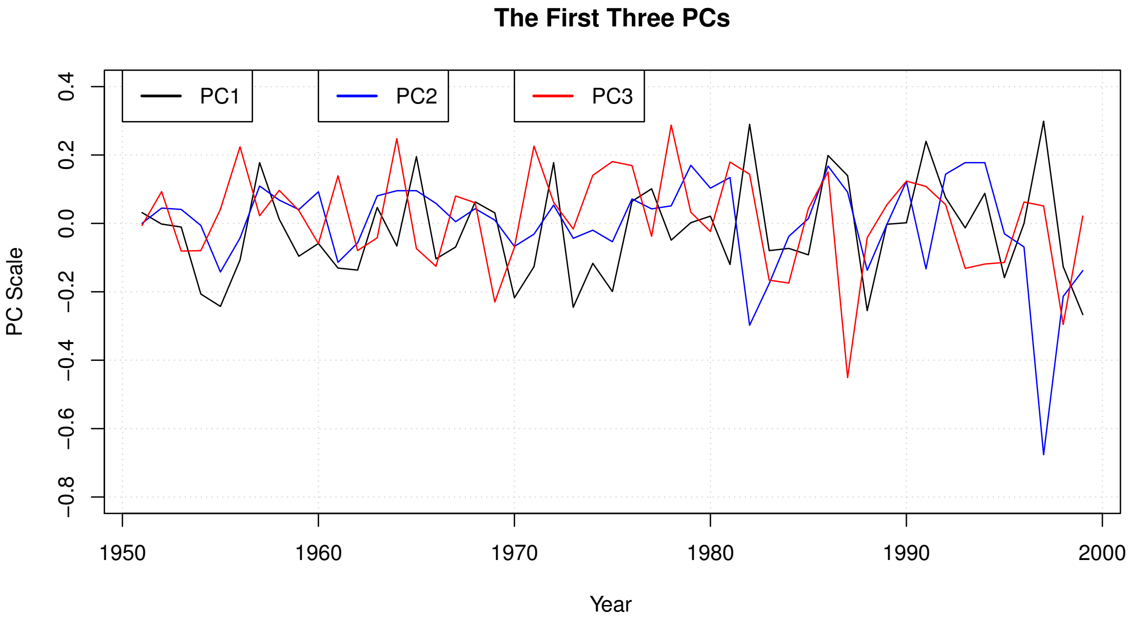
```
umat = matrix(U[,3], nrow = numLonVals)
filled.contour(y, x, umat, color.palette=rgb.palette, levels=int,
               xlim=c(120,300),ylim=c(-40,40),
               plot.title=title(main="EOF3 of the Tropical Pacific
                                      Precipitation Data",
                 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



(c) Plot the time series of the first three V column vectors, which are the temporal patterns of the data field, and are also known as Principal Components (PCs).

```
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")
```



(d) El Nino signals should show in the figures of Steps (b) and (c). Compare the El Nino signals from the precipitation data with those from the temperature data. Use 100-200 words to describe your comparison.

We can see from this that where the temperature is lower, we can see higher amounts of precipitation. This is because of the El Nino signals show a change in winds that will lead to a change in temperature and precipitation. From the PC's we can see that the precipitation dropped a fairly large amount within the decade of 1990 - 2000.

Problem 5 Exercise (6.1): Suppose that the greenhouse effect results in a net energy gain of the Earth's surface by 1.0 $[Wm^{-2}]$. If the gained heat is all used to heat the Earth's atmosphere, how many years will be needed to warm the Earth's entire atmosphere by 1.0°C? If the heat is all used to heat the Earth's surface water, including the water in the oceans, lakes, and rivers, how many years will it take to warm the water by 1.0°C? Make comments about this study's implications for global warming.

Hint: You can find the relevant information about the Earth's atmosphere and water in this book or from the Internet

Source: https://atmos.washington.edu/~dennis/321/321_Lecture_12.pdf

Notice the following for heating the Earth's Atmosphere by 1° C or by 1 K.

We can use the following equation to determine the mass of the Earth's Atmosphere and Water per square meter:

$$F = ma \qquad \rightarrow \qquad P_s = mg$$

where P_s is the pressure at sea level and g is the acceleration of gravity.

From here notice the following:

$$P_s = 101,325Pa = 101,325\frac{kg}{ms^2} = m\left(9.81\frac{m}{s^2}\right) \qquad \rightarrow \qquad m = 10,328.7462\frac{kg}{m^2}$$

Now, we can determine the time to heat up the Earth's atmosphere or the Earth's water by 1°C using the following equation:

$$q = 1\,\frac{W}{m^2} = 1\,\frac{J}{s\,m^2} = mc\Delta T = \left(10,328.7462\,\frac{kg}{m^2}\right)c\Delta T$$

where c is the specific heat capacity of the desired item.

We get the following for the **Earth's Atmosphere** with $c = 1003.5\frac{J}{kg\,K}$:

$$\Delta T = \frac{q}{mc} = \frac{1\,\frac{J}{s\,m^2}}{\left(10,328.7462\,\frac{kg}{m^2}\right)\left(1003.5\frac{J}{kg\,K}\right)}\frac{K}{s}\left(\frac{31,536,000\,s}{yr}\right) = 3.042577\frac{K}{yr}$$

With this, we can see that:

$$3.042577\,\frac{K}{yr} = 3.042577\,\frac{°C}{yr} = 0.3286687\,\frac{yr}{°C} = \mathbf{120}\,\frac{\boldsymbol{days}}{\mathbf{1°}\boldsymbol{C}}$$

We get the following for the **Earth's Water** with $c = 4184\frac{J}{kg\,K}$:

$$\Delta T = \frac{q}{mc} = \frac{1\,\frac{J}{s\,m^2}}{\left(10,328.7462\,\frac{kg}{m^2}\right)\left(4184\frac{J}{kg\,K}\right)}\frac{K}{s}\left(\frac{31,536,000\,s}{yr}\right) = 0.729739\frac{K}{yr}$$

With this, we can see that:

$$0.729739\,\frac{K}{yr} = 0.729739\,\frac{°C}{yr} = \mathbf{1.370353}\,\frac{\boldsymbol{yr}}{\boldsymbol{°C}}$$

Problem 6 Exercise (6.5): Repeat Elisha Mitchell's calculation with the current data you can find online, such as the North Carolina Climate Office NC ECONet, and make your own estimation of the elevation of Mount Mitchell using the hypsometric equation.

Information Taken on: Sunday, November 22, 2020 @ 8:35PM EST.

Station BURN (Base): Burnsville Tower Station: Location (35.91892° N, 82.26035° W), Elevation (2702 ft above Sea Level), Temperature (9.4° C), Pressure (924.9 mb)

Station MITC (Top): Mount Mitchell State Park Station: Location (35.7585° N, 82.2712° W), Elevation (6200 ft above Sea Level), Temperature (4.5° C), Pressure (811.8 mb).

$$\begin{aligned} z_2 &= z_1 + \frac{2RT_1T_2}{g(T_1+T_2)} \ln\left(\frac{p_1}{p_2}\right) \\ &= (2702)(0.3408) + \frac{2(287.055)(273.15+9.4)(273.15+4.5)}{9.80665(273.15+9.4+273.15+4.5)} \ln\left(\frac{924.9}{811.8}\right) \\ &= 1990.160413\, m \\ &= 5839.672572\, ft \end{aligned}$$

This is about a 5% error from the actual 6200ft elevation. This is could be from the high humidity (95%) or the high winds (20 mph).

Problem 7 Exercise (6.6): Calculate the elevation of a high mountain peak near your location, or in another place with which you are familiar, using observed data and the hypsometric equation.

Information Taken on: Saturday, November 28, 2020 @ 3:33 PM & @4:18 PM

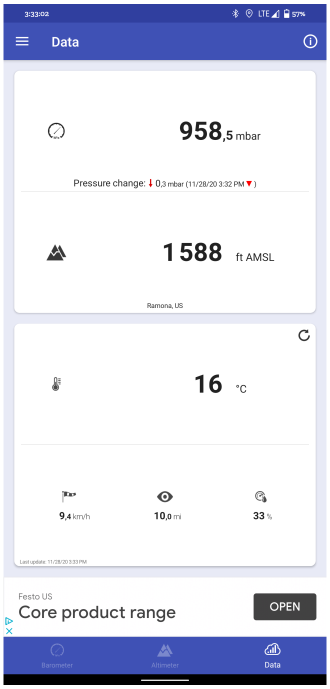
Mount Woodson (Base): Elevation (1588 ft above Sea Level), Temperature (16° C), Pressure (958.5 mb)

Mount Woodson (Top): Elevation (2402 ft above Sea Level), Temperature (16° C), Pressure (919.5 mb).

$$\begin{aligned}
z_2 &= z_1 + \frac{2RT_1T_2}{g(T_1+T_2)} \ln\left(\frac{p_1}{p_2}\right) \\
&= (1588)(0.3408) + \frac{2(287.055)(273.15+16)(273.15+16)}{9.80665(273.15+16+273.15+16)} \ln\left(\frac{958.5}{919.5}\right) \\
&= 892.7743933\, m \\
&= 2619.643173\, ft
\end{aligned}$$

This is about a 9% error from the actual 2402ft elevation. This is most likely to be from an inaccurate barometer reading from a weak GPS signal. When reading online, it says that this mountain has a peak of between 2500 ft - 2800 ft depending on different sources, so our calculation is not too far off.

For this problem, I was unable to find accurate barometer readings of these local mountains online due to the fact they weren't major mountains in the US. So I decided to hike this mountain, and calculate the barometric pressure, temperature, and altitude using my phones GPS. Below, you'll see the data collected.



958,5 mbar

Pressure change: ↓ 0,3 mbar (11/28/20 3:32 PM ▼)

1 588 ft AMSL

Ramona, US

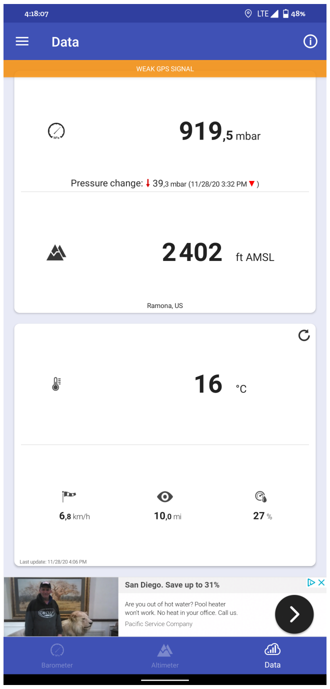
16 °C

9,4 km/h

10,0 mi

33 %

Last update: 11/28/20 3:33 PM



WEAK GPS SIGNAL

919,5 mbar

Pressure change: ↓ 39,3 mbar (11/28/20 3:32 PM ▼)

2 402 ft AMSL

Ramona, US

16 °C

6,8 km/h

10,0 mi

27 %

Last update: 11/28/20 4:06 PM

Problem 8:

(a) Derive the Buffon's needle problem solution when the needle length is longer than the gap of two lines.

Notice the following:

Let y be position of the lower end of the needle, d be the gap between the two lines, and $\ell > d$ be the length of the needle.

We have the probability region be in the space $[-\pi/2, \pi/2] \times [0, d]$. Thus we get the probability region is $A = \pi d$.

We can use the following equation to determine the angles for which the line will cross the top line:

$$y + \ell \cos \alpha \geq d$$

The lower bound, $y = 0$, means that the only way for the needle to cross the top line is if at least $\ell \cos \alpha = d$. From this we can see that this is the case for:

$$\alpha = \arccos \frac{d}{l}$$

Using trig rules, we can see that the line definitely crosses for $\theta \in [-\alpha, \alpha]$. From this we get the following "definite" area

$$A_x = 2\alpha d$$

Now for the indefinite area, we can find this by taking the following integral:

$$A_c = 2\int_{\alpha}^{\pi/2} [d - (d - \ell \cos \theta)]\, d\theta = 2\ell \int_{\alpha}^{\pi/2} \cos \theta\, d\theta = 2\ell - 2\ell \sin \alpha$$

Using algebra, we see the following inequality:

$$\ell \sin \alpha = \sqrt{\ell^2 - d^2}$$

Now from here, we get the probability of a needle crossing the top line being the following:

$$\begin{aligned}
P_L = \frac{A_x + A_c}{A} &= \frac{1}{\pi d}[2\alpha d + 2\ell - 2\ell \sin \alpha] \\
&= \frac{2}{\pi d}\left[\alpha d + \ell - \sqrt{\ell^2 - d^2}\right] \\
&= \frac{2}{\pi}\left[\alpha + \frac{\ell}{d} - \sqrt{\left(\frac{\ell}{d}\right)^2 - 1}\right] \\
&= \frac{2}{\pi}\left[\arccos\left(\frac{d}{\ell}\right) + \frac{\ell}{d} - \sqrt{\left(\frac{\ell}{d}\right)^2 - 1}\right]
\end{aligned}$$

(b) Write an R program to simulate the above Buffon's needle problem in (a).

```
# Problem 8

BuffonLongSim = function(d, l, n = 10000) {
  k = 0
  for (i in 1 : n) {
    y = runif(1, min = 0, 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) # d = 28, l = 81
```

```
## [1] 0.8899
```

```
BuffonLongExact(d, l) # d = 28, l = 81
```

```
## [1] 0.8888297
```

Problem 9: Write an R code for Monte Carlo simulation to calculate the volume of a unit ball in 8-dimensional space.

```
# 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] 4.06016
```

```
MCExact(8)
```

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

Problem 10: When one roll two dice randomly, it is known that the event of two dice's sum equal to 7 has a probability 1/6. Write an R code to simulate this process and verify this result.

We can verify this result through the following:

Each dice has numbers ranging from 1 - 6. With this, each roll each dice has a chance of falling on 6 numbers. This means there will be a total of $6^2 = 36$ combinations that 2 dice can be rolled. Notice the numbers that add up to 7:

$$(1 + 6), (2 + 5), (3 + 4), (4 + 3), (5 + 2), (6 + 1)$$

Thus we get there is a $6/36 = 1/6$ probability of getting two dice whose sum at up to 7.

Notice the code below, that simulates a dice roll:

```
# 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] 0.167
```

```
diceRollExact()
```

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