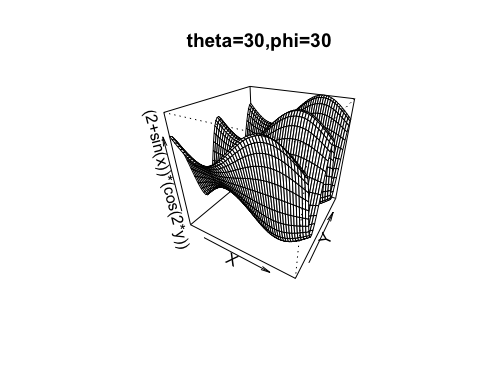
Assignment 3

Wenhao Li 250959325

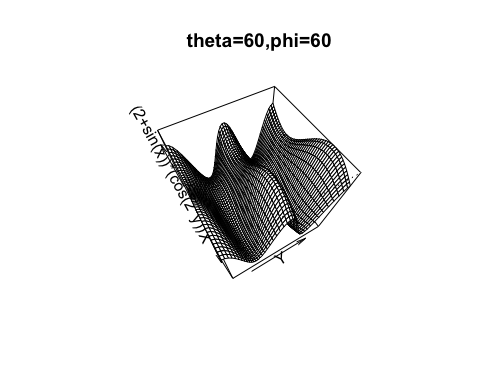
01/03/2020

#Question 1

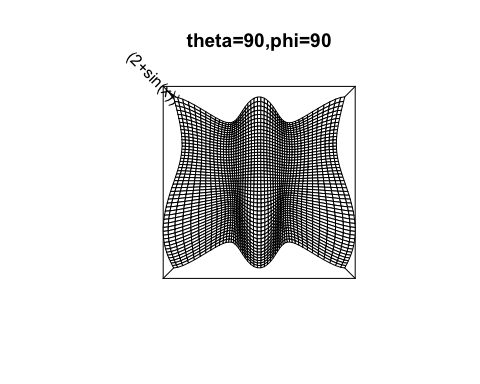
#x,y,z  
x<-seq(-pi,pi,length=50)  
y<-seq(-pi,pi,length=50)  
fun\_xy<-function(x,y){  
 (2+sin(x))\*(cos(2\*y))  
}  
z<-outer(x,y,fun\_xy)  
  
#Plot plots by three different values  
persp(x,y,z,xlab="X",ylab="Y",zlab="(2+sin(x))\*(cos(2\*y))",main="theta=30,phi=30",theta=30,phi=30)



persp(x,y,z,xlab="X",ylab="Y",zlab="(2+sin(x))\*(cos(2\*y))",main="theta=60,phi=60",theta=60,phi=60)



persp(x,y,z,xlab="X",ylab="Y",zlab="(2+sin(x))\*(cos(2\*y))",main="theta=90,phi=90",theta=90,phi=90)



#Question 2

a\_fun<-function(n){  
 #Error checking   
 if(!is.numeric(n)){  
 stop("n should be numeric")  
 }  
 if(is.na(n)){  
 stop("n should not indlude missing values")  
 }  
 #Function Body  
 minimum<-sum(pmin(2^(1:n),(1:n)^3))  
 maximum<-sum(pmax(2^(1:n),(1:n)^3))  
 return(c(minimum,maximum))  
}

#Test for a few bad/wrong n's  
#Test for numeric  
a\_fun("my")

## Error in a\_fun("my"): n should be numeric

#Test for missing valus  
a\_fun(c(NA,30))

## Warning in if (is.na(n)) {: the condition has length > 1 and only the first  
## element will be used

## Error in a\_fun(c(NA, 30)): n should not indlude missing values

#Execute function  
sapply(seq(200,5000,by=600),a\_fun)

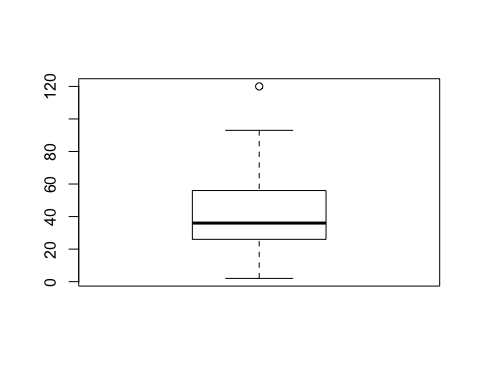
## [,1] [,2] [,3] [,4] [,5]  
## [1,] 4.040090e+08 1.026562e+11 961772488996 4.004001e+12 1.143319e+13  
## [2,] 3.213876e+60 1.333603e+241 Inf Inf Inf  
## [,6] [,7] [,8] [,9]  
## [1,] 2.623079e+13 5.215584e+13 9.3745e+13 1.563125e+14  
## [2,] Inf Inf Inf Inf

#Question 3

IQR.outliers<-function(x){  
 #(a) Error Checking  
 if(is.na(x)){  
 stop("x should not include missing values")  
 }  
 if(!is.numeric(x)){  
 stop("x should be numeric")  
 }  
   
 #(b) Compute IQR as Q3-Q1  
 Q1<-quantile(x,probs = 0.25)  
 Q3<-quantile(x,probs = 0.75)  
 IQR<-Q3-Q1  
   
 #(c) Detect suspected outlier(s)  
 lowerboundary<-Q1-1.5\*IQR  
 upperboundary<-Q3+1.5\*IQR  
 outliers\_lowerboundary<-x[x<lowerboundary]  
 outliers\_upperboundary<-x[x>upperboundary]  
   
 #(d) Boxplot  
 boxplot(x)  
   
 #(e)Proper output  
 return(list(IQR=IQR,outliers\_lowerboundary=outliers\_lowerboundary,outliers\_upperboundary=outliers\_upperboundary))  
}

# Test with dist and speed in the data.frame cars  
IQR.outliers(cars$dist)

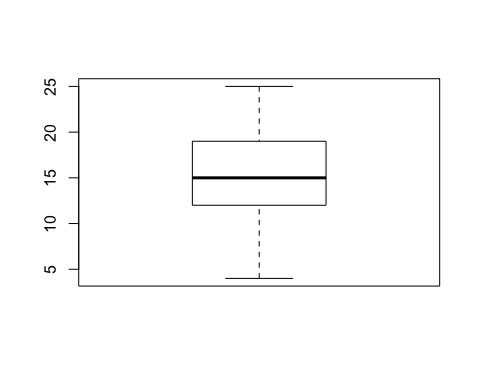
## Warning in if (is.na(x)) {: the condition has length > 1 and only the first  
## element will be used



## $IQR  
## 75%   
## 30   
##   
## $outliers\_lowerboundary  
## numeric(0)  
##   
## $outliers\_upperboundary  
## [1] 120

IQR.outliers(cars$speed)

## Warning in if (is.na(x)) {: the condition has length > 1 and only the first  
## element will be used



## $IQR  
## 75%   
## 7   
##   
## $outliers\_lowerboundary  
## numeric(0)  
##   
## $outliers\_upperboundary  
## numeric(0)

# Test with wrong inputs  
IQR.outliers(c(NA,30))

## Warning in if (is.na(x)) {: the condition has length > 1 and only the first  
## element will be used

## Error in IQR.outliers(c(NA, 30)): x should not include missing values

#Question 4

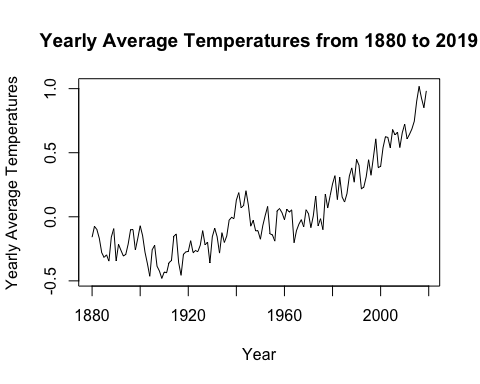
# (a)

# Import dataset  
dataset1<-read.table("~/Desktop/SS2864/Assignment 3/GLB.Ts\_dSST.csv",header = TRUE,sep=",")  
# Keep only 13 columns  
dataset2<-dataset1[,1:13]

# (b)

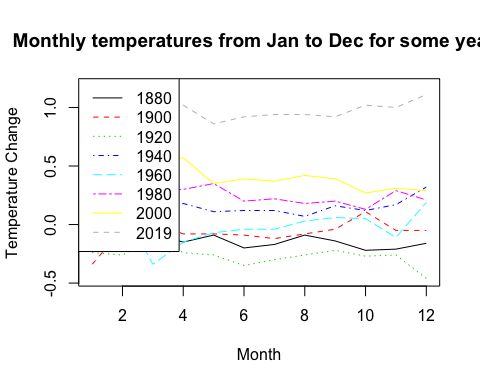
my.newfunction<-function(x){  
 # Error checking  
 if(is.na(x)){  
 stop("x should not include missing values")  
 }  
 # Function Body  
 new\_x<-x[-1]  
 return(mean(new\_x))  
}  
average\_temperatures<-apply(dataset2,1,my.newfunction)  
## Warning in if (is.na(x)) {: the condition has length > 1 and only the first  
## element will be used

plot(dataset2$Year,average\_temperatures,xlab="Year",ylab="Yearly Average Temperatures",main="Yearly Average Temperatures from 1880 to 2019",type="l")

 Comment: The overall trend of yearly average temperature from 1880 to 2019 is upward. The yearly average temperature between 1880 and 1940 is under 0 and is stable. The yearly average temperature increase rapidly between 1960 and 2019.

# (c)

Years<-c(seq(1880,2000,20),2019)  
n<-length(Years)  
dataset3<-dataset2[Years-1880+1,2:13]  
dataset4<-t(as.matrix(dataset3))  
matplot(dataset4,xlab="Month",ylab="Temperature Change",type="l",col=1:n,lty=1:n)  
legend("topleft",legend=Years,col=1:n,lty=1:n)  
title("Monthly temperatures from Jan to Dec for some years")

 Comment: Each year’s monthly temperatures is stable. Monthly temperature of 2019 is much higher than other years’ monthly temperature.

#Question 5

my.ecdf<-function(x,y){  
 # Error checking  
 if(!is.numeric(y)){  
 stop("y should be a number")  
 }  
 if(length(y)!=1){  
 stop("y should be a vector")  
 }  
 # Function Body  
 n<-length(x)  
 a\_sum<-sum(as.integer(x<=y))  
 b<-a\_sum/n  
 return(b)  
}  
  
# Test  
my.ecdf(x=rnorm(20),y=-2)

## [1] 0

my.ecdf(x=rnorm(20),y=2)

## [1] 0.95

my.ecdf(x=rnorm(20),y=median(x))

## [1] 0.55