1. Read Data and calculate basic parameters

```
d<-read.csv("C:/Data-L-Reg-Gradient-Descent.csv") #Depending on the path to
your dataset file
meanX1<-mean(d$x1)
meanX2<-mean(d$x2)
sdX1<-sd(d$x1)
sdX2<-sd(d$x2)
```

2. Plots to Explore Data for Logistic Regression

```
library(ggplot2)
library("ggthemes")
library("scales")

#Scattered plot
ggplot(d, aes(x=x1, y=x2, col=as.factor(y))) + geom_point(size=6,alpha=0.75)+
  ggtitle("Data for Logistic Regression")+ theme_calc() +
scale_color_calc("Event (Y)")+
  theme(axis.text=element_text(size=18),
axis.title=element_text(size=26,face="bold"),legend.text=element_text(size=20
))
```

3. Density plot

```
ggplot(d,aes(x=X1plusX2))+ labs(x = "x1 + x2")+
  geom_density(data=subset(d,y == '0'),fill = "blue", alpha = 0.3) +
  geom_density(data=subset(d,y == '1'),fill = "red", alpha = 0.3)+
theme_calc() +
  geom_vline(xintercept = 140, linetype="dotted",color = "black", size=1.5)+
  theme(axis.text=element_text(size=16),
axis.title=element_text(size=26,face="bold"))
```

4. Logit plot

```
l1<-glm(y~x1+x2,family = 'binomial',data=d)
x<-as.matrix(cbind(rep(1,100),seq(min(d$x1),max(d$x1),length.out = 100),
seq(min(d$x2),max(d$x2),length.out = 100)))
p<-(exp(x%*%l1$coefficients)/(1+exp(x%*%l1$coefficients)))
plot(d$X1plusX2,d$y,xlab = "x1 +
x2",ylab="y",col="Blue",cex.lab=2,cex=1.5,cex.axis=2)
par(new=TRUE)
plot(x%*%as.matrix(c(0,1,1)),p,type = "l",lwd=4,col="Orange",axes =
FALSE,xlab="",ylab="")
```

5. Data prep and functions for Gradient Descent

```
#Normalization of data for gradient descent
d[,1:2]<-scale(d[,1:2])

#Predictor variables matrix
X <- as.matrix(d[,c(1,2)])

#Add ones to Predictor variables matrix
X <- cbind(rep(1,nrow(X)),X)
```

6. Response variable matrix

```
Y <- as.matrix(d$y)
```

7. Sigmoid function for logistic regression

```
sigmoid <- function(z)
{
  g <- 1/(1+exp(-z))
  return(g)
}
```

8. Loss Function for gradient descent (logistic regression)

```
cost <- function(beta)
{
  m <- nrow(X)
  g <- sigmoid(X%*%beta)
  J <- (1/m)*sum((-Y*log(g)) - ((1-Y)*log(1-g)))
  return(J)
}

#Intial betas (0,0,0)
beta <- rep(0,ncol(X))

#Cost at inital beta
cost(beta)
```

9. Gradient Descent to solve logistic regression model

```
# Define learning rate and iteration limit
alpha <- 0.1
num_iters <- 50000
```

10. initialize coefficients

```
beta <- matrix(c(0,0,0), nrow=3)

# gradient descent
for (i in 1:num_iters) {
  error <- (sigmoid(X%*%beta) - Y)
  delta <- (t(X) %*% error) / length(Y)
  beta <- beta - (alpha*delta)
}
```

11. Logistic Regression w/o gradient descent to compare results

```
l<-glm(y~x1+x2,family = 'binomial',data=d)
beta
l$coefficients
```

12. The assumption that Beta1=Beta2 to use X1+X2 instead of X1 and X2 independently

```
#Predictor variables matrix
```

```
X1 <- as.matrix(scale(d[,4]))
X1 <- cbind(rep(1,nrow(X1)),X1)
```

13. Loss/cost Function for gradient descent (logistic regression)

```
cost <- function(beta0,beta1)
{
  m <- nrow(X1)
  g <- sigmoid(X1%*%rbind(beta0,beta1))
  J <- (1/m)*colSums((-rep(Y,length(beta1))*log(g)) - ((1-
rep(Y,length(beta1)))*log(1-g)))
  return(J)
}

d[,4]<-scale(d[,4])
l2<-glm(y~X1plusX2,family = 'binomial',data=d)
```

14. 3D Plot (Loss Function, m, c)

```
beta0<-seq(-2, 2, length.out=100)
beta1<-seq(0.2, 7.2, length.out=100)
z <- outer(beta0,beta1,cost)
```

15. 3D Plot (Loss Function, beta0, beta1)

```
persp(beta0, beta1, z, phi = 30, theta = 30,col = "orange",xlab = "Beta0
(constant)",ylab = "Beta1 (coefficient of x1+x2)",zlab = "Loss Function")
```

16. 2D Heat map (Loss Function, beta0, beta1)

```
image(beta0, beta1, z,xlab = "Beta0 (constant)",ylab = "Beta1 (coefficient of
x1+x2)",main="Loss Function Vs (Beta0 &
Beta1)",cex.lab=1.8,cex.axis=1.5,cex.main=2)
par(new=TRUE)
contour(beta0, beta1, z, xaxt='n', yaxt='n',lwd = 2)
abline(h=2.07252,col="blue",lwd=2)
abline(v=-0.03145,col="blue",lwd=2)
```

17. Data simulation for Logistic Regression

```
# set.seed(47)
# x1<-rnorm(400,70,10)
# x2<-rnorm(400,70,15)
# y<-ifelse(x1+x2+rnorm(400,0,15)>140,1,0)
# d<-data.frame(x1,x2,y,X1plusX2=x1+x2)#
```

Good job!