Assignment-2\_Solution\_Bhowmick.R

Ashok Bhowmick

Sun Jul 23 11:06:11 2017

## CMTH 642  
## Assignment 2 Ashok Bhowmick  
  
getwd()

## [1] "C:/Data Science/Ryerson Courese CMTH 642/Assignment"

setwd("C:/Data Science/Ryerson Courese CMTH 642/Assignment")  
## Writing the final file from Assignment 1   
## Fiename USDB\_Final (Please see end of Q.no. 13 in Assignment-1)  
  
USDB <- read.csv("USDA-Bhowmick.csv", header = TRUE)  
View(USDB)  
nrow(USDB)

## [1] 6020

##  
library(dplyr)

##   
## Attaching package: 'dplyr'

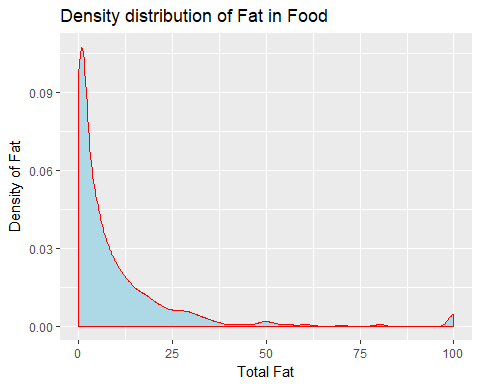
## The following objects are masked from 'package:stats':  
##   
## filter, lag

## The following objects are masked from 'package:base':  
##   
## intersect, setdiff, setequal, union

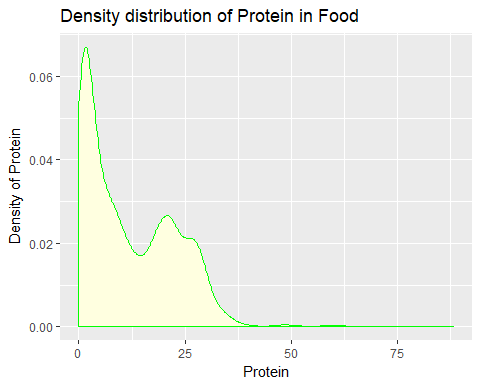
USDB\_New <- mutate(USDB, HighCalories = Calories, HighSodium = Sodium, HighSugar = Sugar, HighFat = TotalFat, HID = ID)  
View(USDB\_New)  
##  
mean.Calories <- mean(USDB$Calories)  
mean.Sodium <- mean(USDB\_New$Sodium)  
mean.Sugar <- mean(USDB\_New$Sugar)  
mean.Fat <- mean(USDB\_New$TotalFat)  
c(mean.Calories, mean.Sodium, mean.Sugar, mean.Fat)

## [1] 221.574086 330.985548 8.192844 10.321616

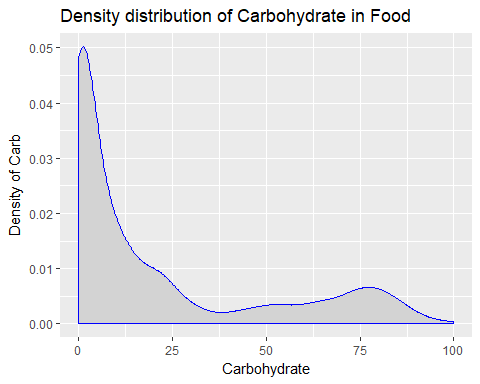
##  
x1 <- USDB\_New[, c(17, 21)]  
x2 <- USDB\_New[, c(18, 21)]  
x3 <- USDB\_New[, c(19, 21)]  
x4 <- USDB\_New[, c(20, 21)]  
##  
x1[x1$HighCalories <= mean.Calories, ] <- 0  
x1[x1$HighCalories > mean.Calories, ] <- 1  
##  
x2[x2$HighSodium <= mean.Sodium, ] <- 0  
x2[x2$HighSodium > mean.Sodium, ] <- 1  
##  
x3[x3$HighSugar <= mean.Sugar, ] <- 0  
x3[x3$HighSugar > mean.Sugar, ] <- 1  
##  
x4[x4$HighFat <= mean.Fat, ] <- 0  
x4[x4$HighFat > mean.Fat, ] <- 1  
##  
x <- cbind(x1, x2, x3, x4)  
View(x)  
x\_edit <- drop(x[, c(1, 3, 5, 7)])  
View(x\_edit)  
##  
## Final working file for Assignment 2  
USDF <- cbind(USDB, x\_edit)  
View(USDF)  
rm(USDB\_New, x1, x2, x3, x4, x, x\_edit)  
##  
## Assignment 2 Questions  
  
## Question No. 1  
## Data Visualization and illustration of distribution  
library(ggplot2)  
a <- ggplot(USDF, aes(TotalFat))  
b <- ggplot(USDF, aes(Protein))  
c <- ggplot(USDF, aes(Carbohydrate))  
## Density distribution  
a + geom\_density(kernel = "gaussian", color = "red", fill = "light blue") + labs(x = "Total Fat", y = "Density of Fat", title = "Density distribution of Fat in Food")



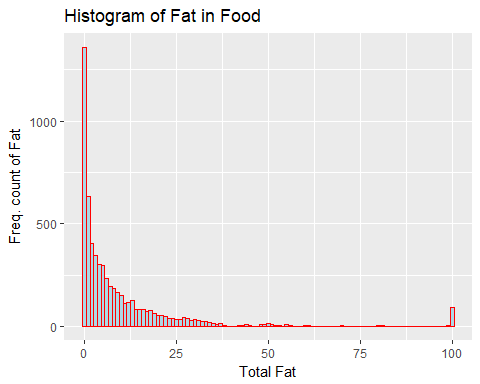
b + geom\_density(kernel = "gaussian", color = "green", fill = "light yellow") + labs(x = "Protein", y = "Density of Protein", title = "Density distribution of Protein in Food")



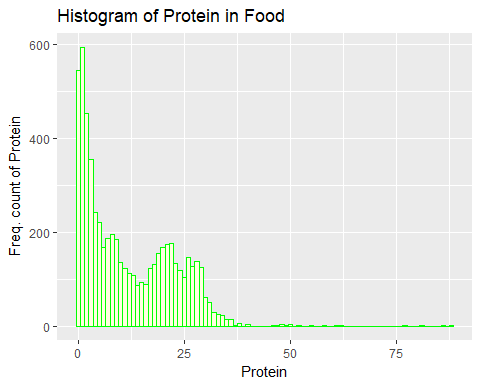
c + geom\_density(kernel = "gaussian", color = "blue", fill = "light grey") + labs(x = "Carbohydrate", y = "Density of Carb", title = "Density distribution of Carbohydrate in Food")



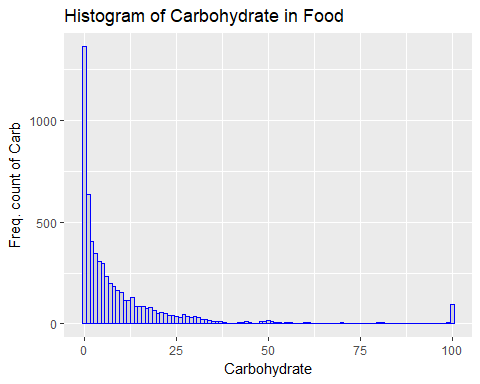
## Frequency Histogram  
a + geom\_histogram(binwidth = 1, color = "red", fill = "light blue") + labs(x = "Total Fat", y = "Freq. count of Fat", title = "Histogram of Fat in Food")



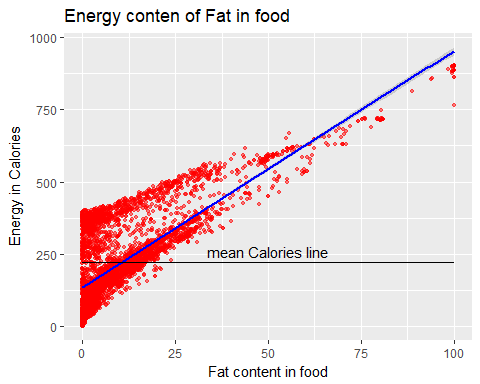
b + geom\_histogram(binwidth = 1, color = "green", fill = "light yellow") + labs(x = "Protein", y = "Freq. count of Protein", title = "Histogram of Protein in Food")



a + geom\_histogram(binwidth = 1, color = "blue", fill = "light grey") + labs(x = "Carbohydrate", y = "Freq. count of Carb", title = "Histogram of Carbohydrate in Food")



##  
  
## Question No. 2  
d <- ggplot(USDF, aes(TotalFat, Calories))  
d + geom\_point(size=1, alpha=0.5, color="red") + labs(x="Fat content in food", y="Energy in Calories", title="Energy conten of Fat in food") + geom\_smooth(method="lm", se=TRUE, size=1, color="blue") + geom\_line(y=mean(USDB$Calories)) + annotate("text", x=50, y= 260, label="mean Calories line")



##  
  
## Question No. 3  
## Examining linearity  
## The question says, High Calories. It is the added nominal variable in Assignment 1 which  
## is created by assigning 1 to high Sodium and 0 to low Sodium. So it has only 0 and 1 as values  
## and ideally, a Multiple Logistic Regression has to be done in such case.  
##   
## formula : ln(p/(1-p))= a + b1x1 + b2x2 + b3x3 + ... and the result is given  
## by Maximum Likelihood Method.   
##   
  
## Regression   
summary(glm(HighCalories ~ Carbohydrate + Protein + TotalFat + Sodium, data = USDF, family = binomial))

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

##   
## Call:  
## glm(formula = HighCalories ~ Carbohydrate + Protein + TotalFat +   
## Sodium, family = binomial, data = USDF)  
##   
## Deviance Residuals:   
## Min 1Q Median 3Q Max   
## -4.9312 -0.0494 -0.0039 0.0117 5.2880   
##   
## Coefficients:  
## Estimate Std. Error z value Pr(>|z|)   
## (Intercept) -1.398e+01 6.374e-01 -21.936 <2e-16 \*\*\*  
## Carbohydrate 2.360e-01 1.094e-02 21.581 <2e-16 \*\*\*  
## Protein 2.208e-01 1.445e-02 15.283 <2e-16 \*\*\*  
## TotalFat 6.400e-01 2.794e-02 22.909 <2e-16 \*\*\*  
## Sodium -6.984e-05 6.896e-05 -1.013 0.311   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## (Dispersion parameter for binomial family taken to be 1)  
##   
## Null deviance: 8157.2 on 6019 degrees of freedom  
## Residual deviance: 1138.2 on 6015 degrees of freedom  
## AIC: 1148.2  
##   
## Number of Fisher Scoring iterations: 9

##  
P1 <- (1-pchisq(8157.2, 6019))  
P1

## [1] 0

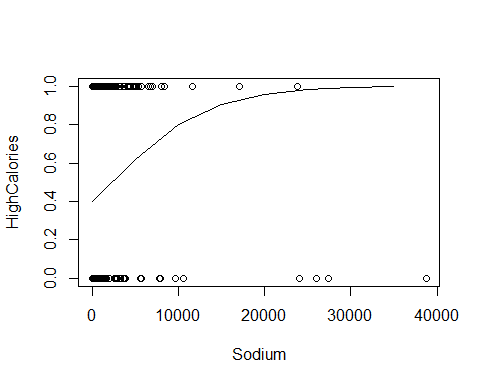
## Question No. 4  
## From the answer of question No. 3  
## Significance level of Sodium is found to be the lowest  
## To visualize  
model1 <- glm(HighCalories ~ Sodium, data = USDF, family = binomial("logit"))  
summary(model1)

##   
## Call:  
## glm(formula = HighCalories ~ Sodium, family = binomial("logit"),   
## data = USDF)  
##   
## Deviance Residuals:   
## Min 1Q Median 3Q Max   
## -3.619 -1.013 -1.008 1.329 1.357   
##   
## Coefficients:  
## Estimate Std. Error z value Pr(>|z|)   
## (Intercept) -4.133e-01 2.939e-02 -14.060 < 2e-16 \*\*\*  
## Sodium 1.795e-04 4.421e-05 4.061 4.88e-05 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## (Dispersion parameter for binomial family taken to be 1)  
##   
## Null deviance: 8157.2 on 6019 degrees of freedom  
## Residual deviance: 8129.7 on 6018 degrees of freedom  
## AIC: 8133.7  
##   
## Number of Fisher Scoring iterations: 4

range(USDF$Sodium)

## [1] 0 38758

x <- seq(min(USDF$Sodium), max(USDF$Sodium), 5000)  
y <- predict(model1, list(Sodium=x), type = "response")  
plot(HighCalories ~ Sodium, data = USDF)  
lines(x, y)



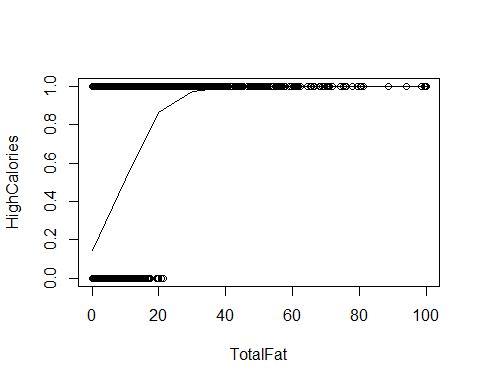
## Question No. 5  
## From the answer of Question 3, all Carbohydrate, Protein and TotalFat have   
## comparable significance where TotalFat has highest z-score  
## Choosing TotalFat as the most significant one  
model2 <- glm(HighCalories ~ TotalFat, data = USDF, family = binomial("logit"))  
summary(model2)

##   
## Call:  
## glm(formula = HighCalories ~ TotalFat, family = binomial("logit"),   
## data = USDF)  
##   
## Deviance Residuals:   
## Min 1Q Median 3Q Max   
## -2.1098 -0.7210 -0.5703 0.6168 1.9587   
##   
## Coefficients:  
## Estimate Std. Error z value Pr(>|z|)   
## (Intercept) -1.759487 0.047258 -37.23 <2e-16 \*\*\*  
## TotalFat 0.181472 0.005435 33.39 <2e-16 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## (Dispersion parameter for binomial family taken to be 1)  
##   
## Null deviance: 8157.2 on 6019 degrees of freedom  
## Residual deviance: 5718.4 on 6018 degrees of freedom  
## AIC: 5722.4  
##   
## Number of Fisher Scoring iterations: 6

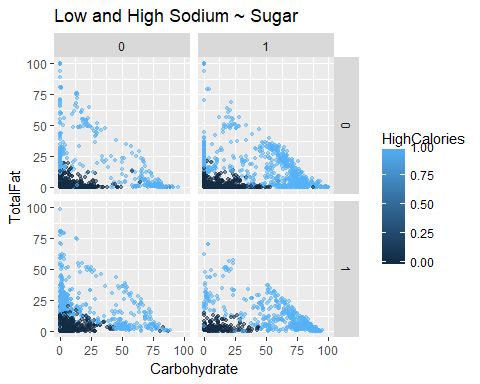
range(USDF$TotalFat)

## [1] 0 100

x <- seq(min(USDF$TotalFat), max(USDF$TotalFat), 10)  
y <- predict(model2, list(TotalFat=x), type = "response")  
plot(HighCalories ~ TotalFat, data = USDF)  
lines(x, y)



##  
library(ggplot2)  
gl <- ggplot(USDF, aes(Carbohydrate, TotalFat))  
gl + geom\_point(size=1, alpha=0.5, aes(color=HighCalories)) + facet\_grid(HighSodium~HighSugar) + labs(title = "Low and High Sodium ~ Sugar")



##  
  
## Question No. 6  
## Comment:  
## As a function call, individual food name or their ID has to be called each time  
## Thereby separating all healthy foods together is an issue and could be a lengthy routine  
## so the following has been done as the question does not specify whether  
## a complete separation or individual merit is to be explored !   
  
USDF\_Check <- USDF[, c(2, 3, 18, 19, 20)]  
View(USDF\_Check)  
nrow(USDF\_Check)

## [1] 6020

##  
attach(USDF\_Check)  
## Separating Safe Foods   
Pass1 <- USDF\_Check[which(USDF\_Check$HighSodium == '0'), ]  
nrow(Pass1)

## [1] 4282

##  
Pass2 <- Pass1[which(Pass1$HighSugar == '0'), ]  
nrow(Pass2)

## [1] 2355

##  
SafeFoods <- Pass2[which(Pass2$HighFat == '0'), ]  
nrow(SafeFoods)

## [1] 1690

View(SafeFoods)  
##  
  
## Unsafe Foods  
Fail1 <- USDF\_Check[which(USDF\_Check$HighSodium == '1'), ]  
nrow(Fail1)

## [1] 1738

##  
Fail2 <- Fail1[which(Fail1$HighSugar == '1'), ]  
nrow(Fail2)

## [1] 706

##  
UnsafeFoods <- Fail2[which(Fail2$HighFat == '1'), ]  
nrow(UnsafeFoods)

## [1] 213

View(UnsafeFoods)  
##  
  
## Note:  
## Notice, Safe + Unsafe does not sum up to total food because these are totally  
## safe and totally unsafe food by criteria. However, there will be intermediates who has one   
## 2 parameters either '0' or '1'. They can also be counted out by cross selecting  
## by doing Pass on Fail or Fail on Pass. Not asked in question though!  
## For example  
## Sodium & Sugar safe  
Int1 <- USDF\_Check[which(USDF\_Check$HighSodium == '0' & USDF\_Check$HighSugar == '0'), ]  
nrow(Int1)

## [1] 2355

##  
## Sodium & Fat safe  
Int2 <- USDF\_Check[which(USDF\_Check$HighSodium == '0' & USDF\_Check$HighFat == '0'), ]  
nrow(Int2)

## [1] 3139

##  
## Sugar and Fat safe  
Int3 <- USDF\_Check[which(USDF\_Check$HighSugar == '0' & USDF\_Check$HighFat == '0'), ]  
nrow(Int3)

## [1] 2317

## etc.  
detach(USDF\_Check)  
##  
source("SFood.R")  
##  
## Question No. 7  
  
View(USDF)  
nrow(USDF)

## [1] 6020

library(dplyr)  
FoodSafety <- mutate(USDF, HealthCheck = ID)  
FoodSafety$HealthCheck[which(FoodSafety$HighSodium == '0' & FoodSafety$HighSugar == '0' & FoodSafety$HighFat == '0')] <- "Safe Food"  
FoodSafety$HealthCheck[which(FoodSafety$HighSodium == '1' | FoodSafety$HighSugar == '1' | FoodSafety$HighFat == '1')] <- "Unsafe Food"  
nrow(FoodSafety)

## [1] 6020

View(FoodSafety)  
##  
write.csv(FoodSafety, "Food-Final\_Bhowmick.csv")  
##  
## Question No. 8  
## Foods that fail HealthCheck which has either one value high  
(nrow(USDF)-nrow(SafeFoods))

## [1] 4330

length(which(FoodSafety$HealthCheck == "Unsafe Food"))

## [1] 4330

##  
rm(USDF\_Check)  
## End Assignment 2