Inference_hw4

Warren Geither

10/16/2020

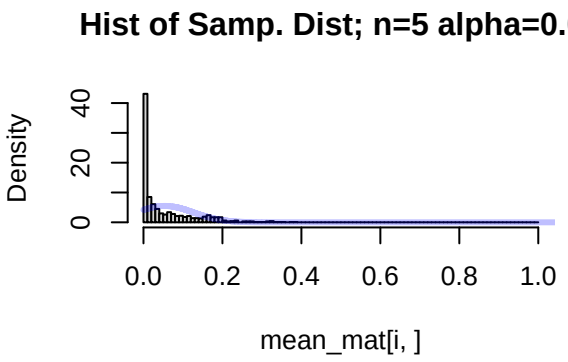
Problem 1

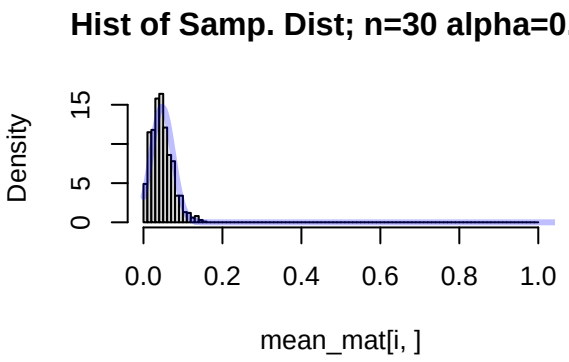
a.) The Central Limit Theoreom states, as n increases the sampling distribution of the sample mean should get closer to the normal distribution.

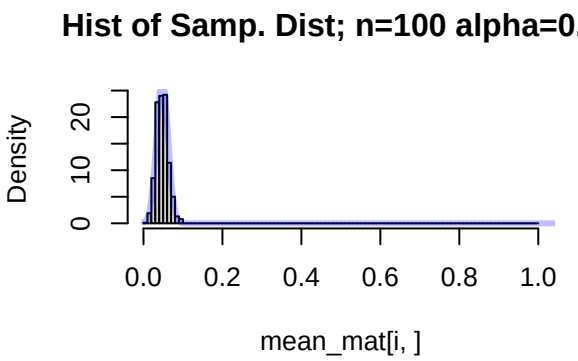
b.)

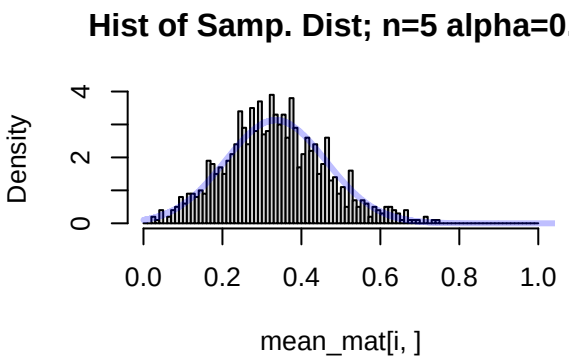
```
# intialize n values
n1 <- 5
n2 <- 30
n3 <- 100

# set alphas
alpha1 <- 0.05
alpha2 <- 0.5
alpha3 <- 1
alpha4 <- 5

# set beta
beta <- 1

# set number of simulations
M <- 1000

# set seed
set.seed(82349)

# create a matrix to store mean vectors
mean_mat <- matrix(NA, nrow = 12, ncol = M)

# create an alpha vector to loop through
alpha_vec <- c(alpha1, alpha2, alpha3, alpha4)

# create a n vector to loop through
n_vec <- c(n1, n2, n3)

# initialize variable
j=1

# for loop to generate mean vector for each combo of alpha and n
for (alpha in alpha_vec){
  for (n in n_vec){
```

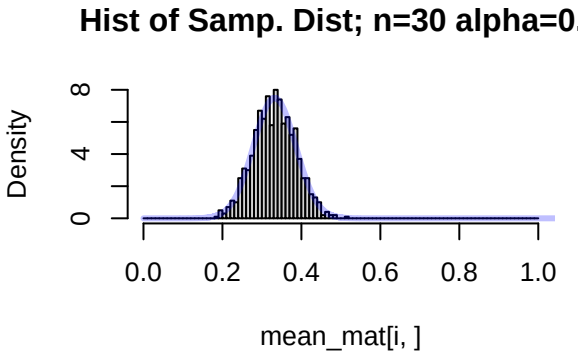
```
    # generate data and obtain mean
    random_beta_matrix <- matrix(rbeta((n*M), shape1 = alpha, shape2 = beta), nrow = M, ncol = n)

    # apply mean to every row
    mean_vector <- apply(random_beta_matrix, 1, mean)

    # store mean vector
    mean_mat[j,] <- mean_vector

    # increment j
    j = j+1
  }
}

# initalize empty vector
title_vec <- c(rep(NA,12))

# intialize variable
j = 1

# loop to create title names for histograms
for(alpha in alpha_vec){
  for(n in n_vec){
  title_vec[j] <- paste0(paste0(paste0("Hist of Samp. Dist; n=", n) , " alpha="), alpha)
  j = j+1
  }
}

# create histograms
par(mfrow=c(2,2))

# sequence for x values
xs<-seq(0,5,.0001)

# loop to plot graphs
for (i in 1:12) {
  hist(mean_mat[i,],freq=FALSE, main = title_vec[i], breaks = c(seq(0,1,0.01)))
  lines(xs,dnorm(xs,mean=mean(mean_mat[i,]),sd=sd(mean_mat[i,])),col=rgb(0,0,1,1/4),lwd=3)
}
```

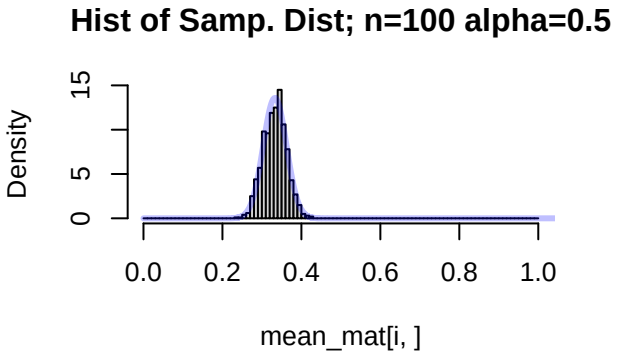

Hist of Samp. Dist; n=5 alpha=0.05
Density
0
20
40
0.0
0.2
0.4
0.6
0.8
1.0
mean_mat[i,]

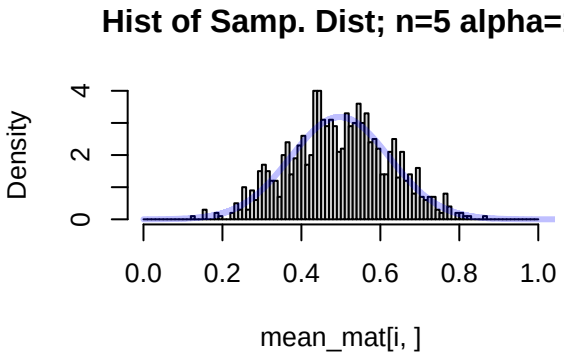

Hist of Samp. Dist; n=30 alpha=0.05
Density
0
5
15
0.0
0.2
0.4
0.6
0.8
1.0
mean_mat[i,]

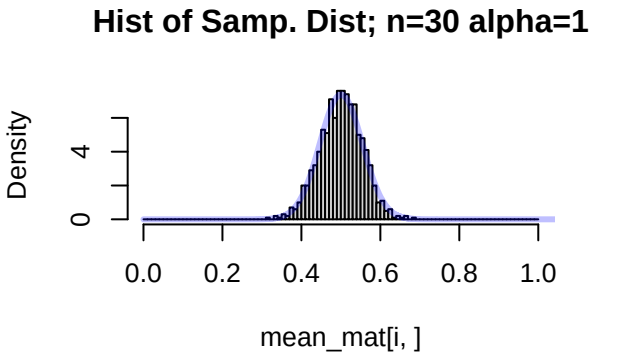

Hist of Samp. Dist; n=100 alpha=0.05
Density
0
10
20
0.0
0.2
0.4
0.6
0.8
1.0
mean_mat[i,]

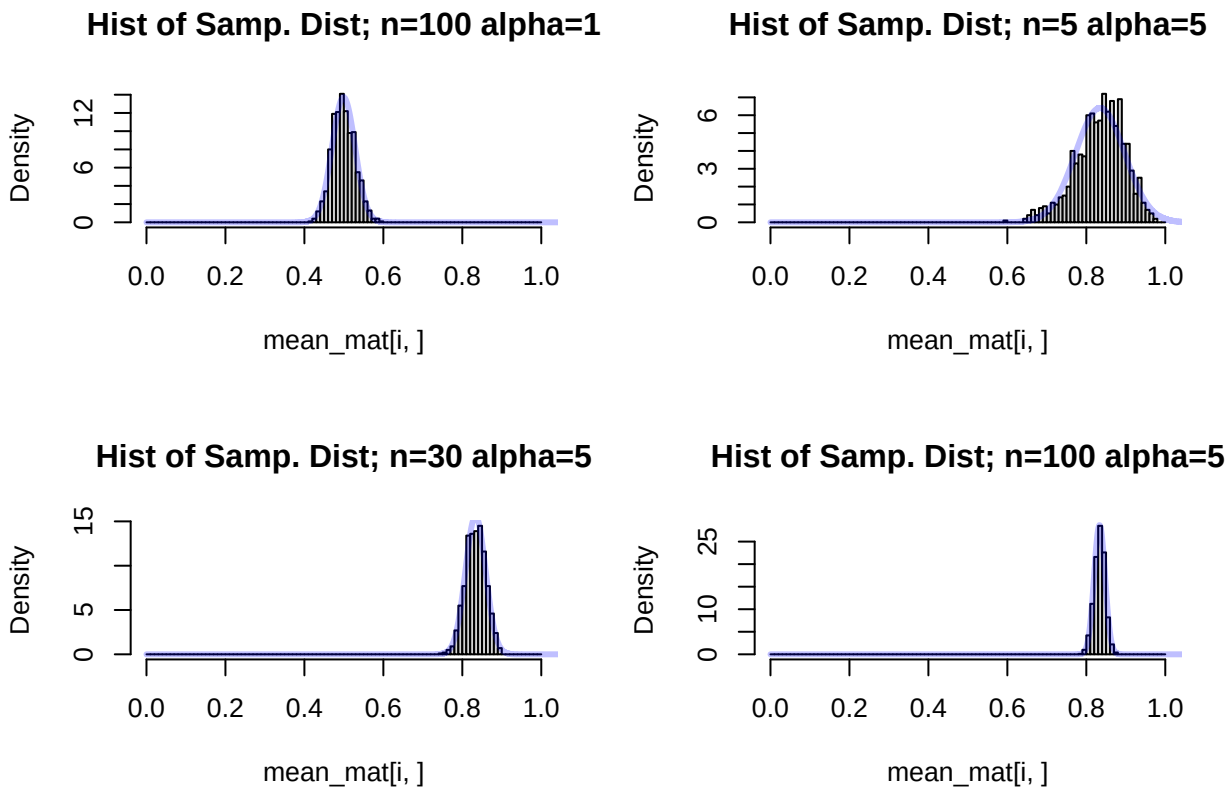

Hist of Samp. Dist; n=5 alpha=0.5
Density
0
2
4
0.0
0.2
0.4
0.6
0.8
1.0
mean_mat[i,]


Hist of Samp. Dist; n=30 alpha=0.5
Density
0
4
8
0.0
0.2
0.4
0.6
0.8
1.0
mean_mat[i,]


Hist of Samp. Dist; n=100 alpha=0.5
Density
0
5
15
0.0
0.2
0.4
0.6
0.8
1.0
mean_mat[i,]


Hist of Samp. Dist; n=5 alpha=1
Density
0
2
4
0.0
0.2
0.4
0.6
0.8
1.0
mean_mat[i,]


Hist of Samp. Dist; n=30 alpha=1
Density
0
4
0.0
0.2
0.4
0.6
0.8
1.0
mean_mat[i,]



c.) As n increases for all levels of alpha, the variance gets smaller and the distribution looks more normal. As alpha increases, the distribution starts to move to the right, since its mean is increasing.

d.) Whenever alpha = 0.5 or 5 and n is low, the tail of the distribution goes outside the support of x. However at higher values of n, the problem seems to remedy itself.

Problem 2

a.)

$$\text{Bernoulli pdf: } f(x) = p^x(1-p)^{1-x} \text{ ; where, } x = 0, 1 \text{ ; } 0 \leq p \leq 1$$

$$\begin{aligned}
\text{Joint Bernoulli pdf: } f(x_1, x_2, ..., x_n) &= \prod_{i=1}^{n} p^{x_i}(1-p)^{1-x_i} \\
&= [p^{x_1}(1-p)^{1-x_1}] * [p^{x_2}(1-p)^{1-x_2}] * ... * [p^{x_n}(1-p)^{1-x_n}] \\
&= p^{x_1} * p^{x_2} * ... * p^{x_n} * (1-p)^{1-x_1} * (1-p)^{1-x_2} * ... * (1-p)^{1-x_n} \\
&= p^{\sum_{i=1}^{n} x_i}(1-p)^{\sum_{i=1}^{n} 1-x_i}
\end{aligned}$$

Prove its a pdf:

$$\begin{aligned}
\sum_{i=1}^{n} p^{\sum_{i=1}^{n} x_i}(1-p)^{\sum_{i=1}^{n} 1-x_i} &= \sum_{i=2}^{n} p^{\sum_{i=2}^{n} x_i}(1-p)^{\sum_{i=2}^{n} 1-x_i} * \sum_{x=0}^{1} p^{x_1}(1-p)^{1-x_1} \text{ , pull a single term out of the sum} \\
&= \sum_{i=3}^{n} p^{\sum_{i=3}^{n} x_i}(1-p)^{\sum_{i=3}^{n} 1-x_i} * \sum_{x=0}^{1} p^{x_1}(1-p)^{1-x_1} * \sum_{x=0}^{1} p^{x_2}(1-p)^{1-x_2} \text{ , again} \\
&= \sum_{i=3}^{n} p^{\sum_{i=3}^{n} x_i}(1-p)^{\sum_{i=3}^{n} 1-x_i} * [p^0(1-p)^{1-0} + p^1(1-p)^{1-1}] * [p^0(1-p)^{1-0} + p^1(1- \\
&= \sum_{i=3}^{n} p^{\sum_{i=3}^{n} x_i}(1-p)^{\sum_{i=3}^{n} 1-x_i} * [p + (1-p)] * [p + (1-p)] \\
&= \sum_{i=3}^{n} p^{\sum_{i=3}^{n} x_i}(1-p)^{\sum_{i=3}^{n} 1-x_i} * 1 * 1 \text{ , continue this n times...} \\
&= 1
\end{aligned}$$

b.)

$$\text{Joint Normal pdf: } f(y|\theta) = (2\pi\sigma^2)^{-n/2} \prod_{i=1}^{n} e^{-\frac{1}{2}\frac{(y_i-\mu)^2}{\sigma}}$$

$$\text{Multivariate Normal: } f(y_1, y_2, ..., y_n) = \frac{e^{-\frac{1}{2}(y-\mu)^T \sum^{-1}(y-\mu)}}{\sqrt{(2\pi)|\sum|}}$$

$$\text{If } y_1 ... y_n \text{ are uncoorelated, then } \sum^{-1} = \begin{pmatrix} \frac{1}{\sigma_1^2} & 0 & 0 \\ 0 & \frac{1}{\sigma_2^2} & 0 \\ \dots & \dots & \frac{1}{\sigma_i^2} \end{pmatrix} = \frac{1}{\sigma_i^2} I$$

$$\implies \frac{e^{-\frac{1}{2}(y-\mu)^T \sum^{-1}(y-\mu)}}{\sqrt{(2\pi)|\sum|}} = \frac{e^{-\sum_{i=1}^{n} \frac{1}{2}(y-\mu)^T \frac{1}{\sigma_i^2}(y-\mu)}}{\sqrt{2\pi\sigma_i^2}}$$

$$= \frac{e^{-\sum_{i=1}^{n} \frac{1}{2}\frac{1}{\sigma_i^2}(y_i-\mu_i)^T(y_i-\mu_i)}}{\sqrt{2\pi\sigma_i^2}}$$

$$= \frac{e^{-\sum_{i=1}^{n} \frac{1}{2}\frac{1}{\sigma_i^2}(y_i-\mu_i)^2}}{\sqrt{2\pi\sigma_i^2}} \text{ , because } (y-\mu) \text{ is symmetric}$$

$$= \prod_{i=1}^{n} \frac{e^{-\frac{1}{2}\frac{1}{\sigma_i^2}(y_i-\mu_i)^2}}{\sqrt{2\pi\sigma_i^2}} \text{ , sum in the exponent turns into the product}$$

$$= (2\pi\sigma^2)^{-n/2} \prod_{i=1}^{n} e^{-\frac{1}{2}\frac{(y_i-\mu)^2}{\sigma}} \text{ , pull out denom}$$

This indicates that if the random variables are uncorrelated they are also jointly independent.

Problem 4

a.)

```
# load data
vegan_data <- read.csv2(file = "Datafiniti_Vegetarian_and_Vegan_Restaurants.csv", header = TRUE, sep =

# clean data, there are alot of very large and small values, these are most likely mistakes
pitt_vegan_df <- vegan_data %>%
                  filter(city == "Pittsburgh"
                         & menus.amountMin < 60
                         & menus.amountMin > 1
                         & !is.na(menus.amountMin)) %>%
                  select(menus.amountMin)

# transform character column into numeric
pitt_vegan_df <- transform(pitt_vegan_df
                           , menus.amountMin = as.numeric(menus.amountMin))

# get portland data
portland_vegan_df <- vegan_data %>%
                  filter(city == "Portland"
                         & menus.amountMin < 60
                         & menus.amountMin > 1
```

```
                        & !is.na(menus.amountMin)) %>%
                  select(menus.amountMin)

portland_vegan_df <- transform(portland_vegan_df
                           , menus.amountMin = as.numeric(menus.amountMin))

# calc means
pitt_mean <- mean(pitt_vegan_df$menus.amountMin)
portland_mean <- mean(portland_vegan_df$menus.amountMin)

# calc n's
n1 <- nrow(pitt_vegan_df)
n2 <- nrow(portland_vegan_df)

# calc standard deviatoins
sd1 <- sd(pitt_vegan_df$menus.amountMin)
sd2 <- sd(portland_vegan_df$menus.amountMin)

# pooled standard deviation
sp <- sqrt(((n1-1)*sd1^2 + (n2-1)*sd2^2)/n1+n2-2)

# cohens d
cohens_d <- (pitt_mean - portland_mean) / sp

# print cohens d
print(paste0("Cohen's d: ", cohens_d))
```

```
## [1] "Cohen's d: 0.565811875385687"
```

Link to data: https://www.kaggle.com/datafiniti/vegetarian-vegan-restaurants Data description: The dataset gives variety of information on vegan restaurants in the US.

Hypothesis addressed by cohens d: What is the effect size on the minimum price for vegan food when moving from Pittsburgh to Portland.

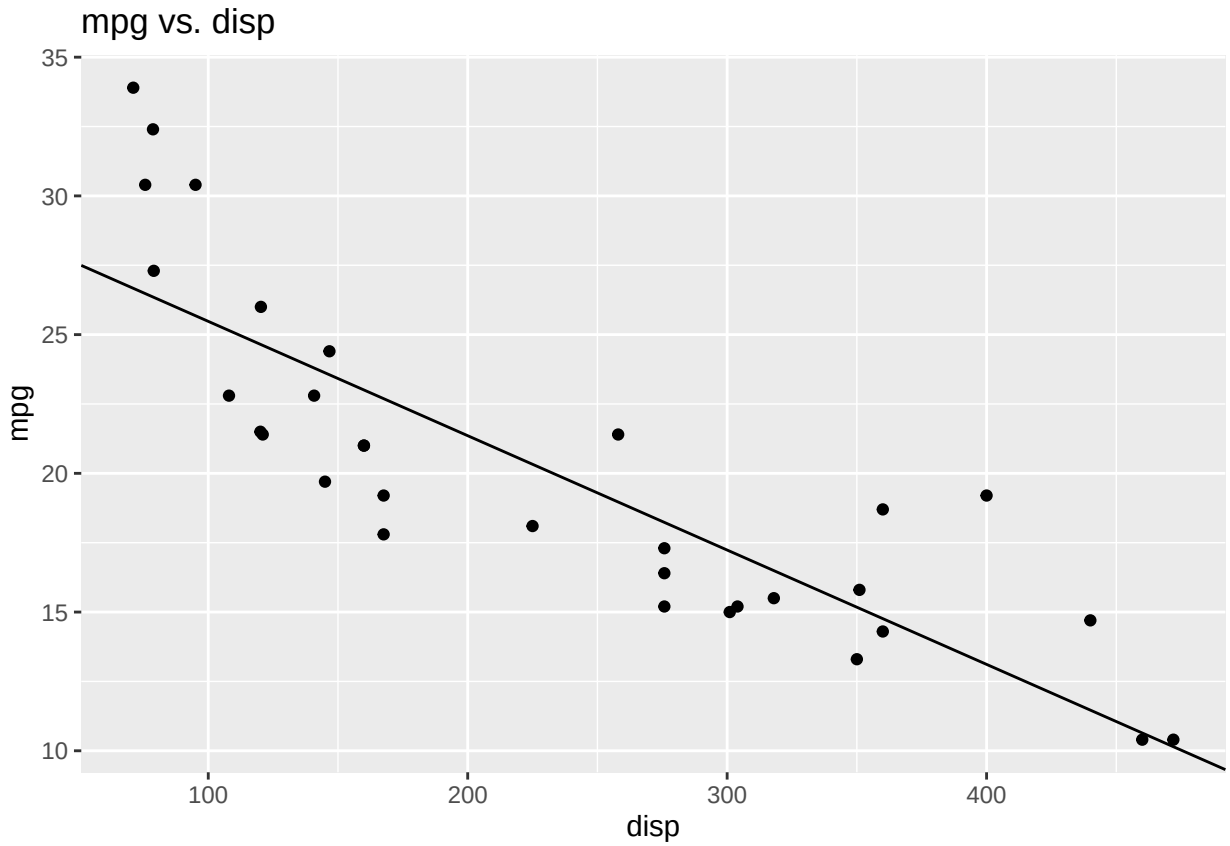
b.)

```
# model of mpg vs displ using mtcars data
lmfit <- lm(mpg ~ disp, mtcars)

# print summary
summary(lmfit)
```

```
##
## Call:
## lm(formula = mpg ~ disp, data = mtcars)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -4.8922 -2.2022 -0.9631  1.6272  7.2305
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
```

```
## (Intercept) 29.599855   1.229720  24.070  < 2e-16 ***
## disp        -0.041215   0.004712  -8.747 9.38e-10 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3.251 on 30 degrees of freedom
## Multiple R-squared:  0.7183, Adjusted R-squared:  0.709
## F-statistic: 76.51 on 1 and 30 DF,  p-value: 9.38e-10
```

```r
# plot function
ggplot(mtcars, aes(x= disp, y=mpg)) +
  geom_point() +
  geom_abline(slope= lmfit$coef[2], intercept=lmfit$coef[1]) +
  ggtitle("mpg vs. disp")
```




```r
# r^2 calc
y <- mtcars$mpg
x <- mtcars$disp

# compute means
y_bar <- mean(y)
x_bar <- mean(x)

# get terms for pearsons coorelation
syy <- sum((y - y_bar)^2)
```

```r
sxx <- sum((x - x_bar)^2)
sxy <- sum((x - x_bar)*(y - y_bar))

# calculate r
r <- sxy / (sqrt(sxx)*sqrt(syy))

# print r
print(paste0("r : ", r))
```

```
## [1] "r : -0.847551379262479"
```

Data description: The mtcars dataset has a number of different vehicles and specs about them.

Hypothesis addressed by pearson's coorelation coefficient: Is there a linear relationship between mpg and engine size (disp)?

c.)

```r
#Read in data
heart_disease_df <- read.csv(file = "processed.cleveland.data", header = F)

#Prepare column names
names <- c("Age",
           "Sex",
           "Chest_Pain_Type",
           "Resting_Blood_Pressure",
           "Serum_Cholesterol",
           "Fasting_Blood_Sugar",
           "Resting_ECG",
           "Max_Heart_Rate_Achieved",
           "Exercise_Induced_Angina",
           "ST_Depression_Exercise",
           "Peak_Exercise_ST_Segment",
           "Num_Major_Vessels_Flouro",
           "Thalassemia",
           "Diagnosis_Heart_Disease")

#Apply column names to the dataframe
colnames(heart_disease_df) <- names

# create binary column for heart disease, we dont care about specific types
heart_disease_df$Diagnosis_Heart_Disease[heart_disease_df$Diagnosis_Heart_Disease != 0] <- 1

# get numbers for contigency table
male_hd <- sum(heart_disease_df$Sex == 1
               & heart_disease_df$Diagnosis_Heart_Disease == 1)

male_Nhd <- sum(heart_disease_df$Sex == 1
               & heart_disease_df$Diagnosis_Heart_Disease == 0)

female_hd <- sum(heart_disease_df$Sex == 0
               & heart_disease_df$Diagnosis_Heart_Disease == 1)

female_Nhd <- sum(heart_disease_df$Sex == 0
```

```
                & heart_disease_df$Diagnosis_Heart_Disease == 0)

# create table
heart_contigency_table <- data.frame(Heart_Disease = c(male_hd,female_hd)
                           , No_Heart_Disease = c(male_Nhd,female_Nhd))

# label rows
rownames(heart_contigency_table) <- c("Male", "Female")

# calc odds
odds_hd_male <- heart_contigency_table[1,1]/heart_contigency_table[1,2]
odds_hd_female <- heart_contigency_table[2,1]/heart_contigency_table[2,2]

# calculate odds ratio
odds_ratio <- odds_hd_male/odds_hd_female

# print pretty data
knitr::kable(heart_contigency_table)
```

	Heart_Disease	No_Heart_Disease
Male	114	92
Female	25	72

```
# print odds ratio
print(paste0("Odds Ratio: ", odds_ratio))
```

```
## [1] "Odds Ratio: 3.56869565217391"
```

Link to data: https://archive.ics.uci.edu/ml/machine-learning-databases/heart-disease/ How to load: https://www.r-bloggers.com/2019/09/heart-disease-prediction-from-patient-data-in-r/ Data description: The dataset gives individual attributes age, sex, along with heart related issues and measurements as well as if they have a heart disease or not.

Hypothesis odds ratio addresses: Do males have a greater odds of getting heart disease than the odds of females?