Heart Attack Risk Prediction. Final Project

Giselle Rahimi, Pilar Gore, WonJune Lee, Jiseon Kim, Jiwon Jung, Yoon-hye Cho, Sukyoung Yoon

2024-06-03

#Tidying (Yoonhye) #Removed unwanted columns

```
untidied_dataset <- untidied_dataset %>%
select(Age,
       Sex,
       Cholesterol,
       `Blood Pressure`,
       `Heart Rate`,
       `Family History`,
       `Exercise Hours Per Week`,
       `Stress Level`,
       `Sleep Hours Per Day`,
       `Heart Attack Risk`,
       `Triglycerides`,
       `Physical Activity Days Per Week`,
       `Sedentary Hours Per Day`,
       `Diet`)
```

#Renamed columns:

```
untidied_dataset <- untidied_dataset %>%
  rename(
    heart_attack_risk = "Heart Attack Risk",
    blood_pressure = "Blood Pressure",
    heart_rate = "Heart Rate",
    family_history = "Family History",
    exercise_hrs_week = "Exercise Hours Per Week",
    stress_level = "Stress Level",
    sedentary_hrs_day = "Sedentary Hours Per Day",
    physical_days_week = "Physical Activity Days Per Week",
    sleep_hrs = "Sleep Hours Per Day",
    age = "Age",
    sex = "Sex",
    cholesterol = "Cholesterol",
    diet = "Diet",
    triglycerides = "Triglycerides"
  )
```

#Split original dataset into “no family history” and “family history”:

```
no_family_history_dataset <-
  subset(untidied_dataset,
         family_history == 0)
```

```
family_history_dataset <-
  subset(
    untidied_dataset,
    family_history == 1)
```

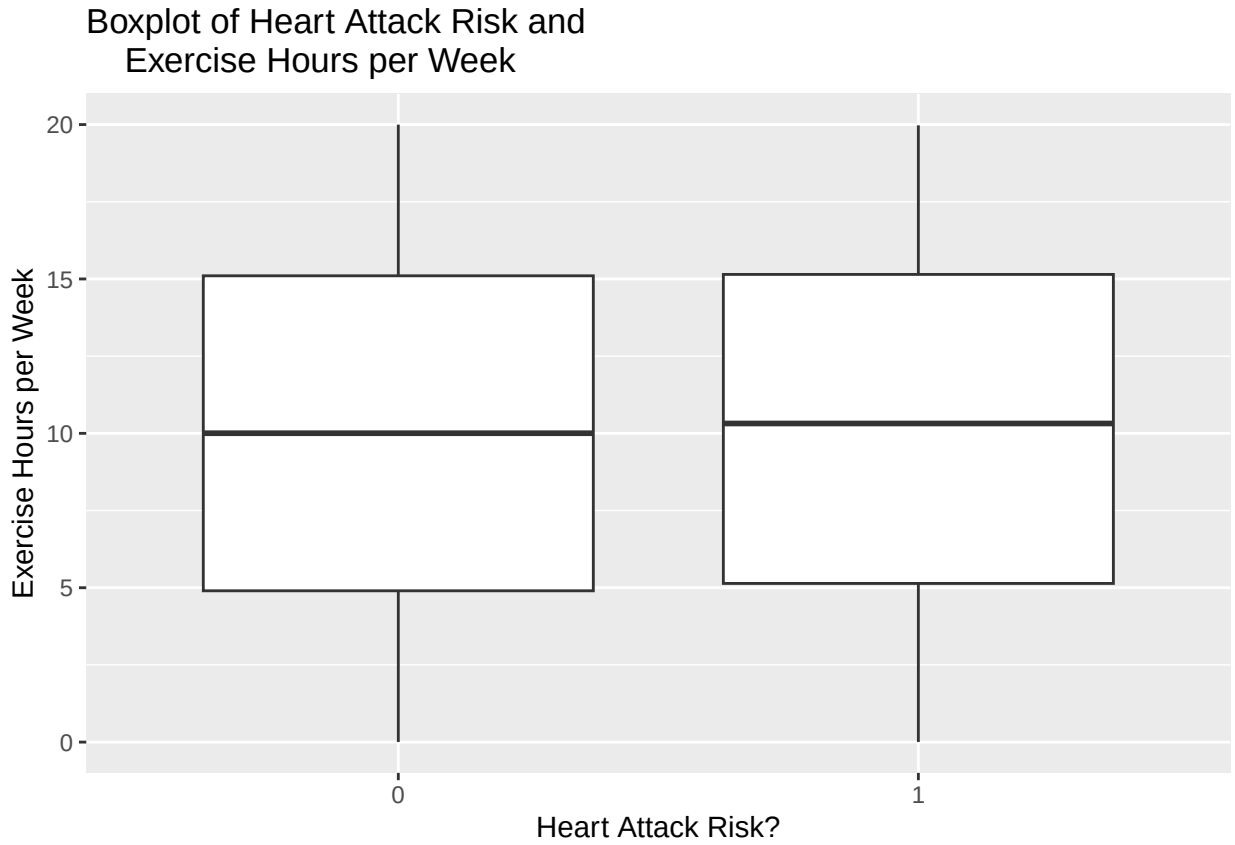
#Separated the blood_pressure column into systolic and diastolic:

```
no_family_history_dataset <- no_family_history_dataset %>%
  separate(
    col = blood_pressure,
    into = combine("systolic",
                   "diastolic"),
    sep = "/",
    convert = FALSE
  )
```

```
## Warning: `combine()` was deprecated in dplyr 1.0.0.
## i Please use `vctrs::vec_c()` instead.
## Call `lifecycle::last_lifecycle_warnings()` to see where this warning was
## generated.
```

#EDA (Jiwon Jung) #boxplot for the exercise hours explanatory variable:

```
no_family_history_dataset %>%
ggplot() +
  geom_boxplot(
    mapping = aes(
      x =
        factor(heart_attack_risk),
      y = exercise_hrs_week))+
  labs(
    title = "Boxplot of Heart Attack Risk and
    Exercise Hours per Week",
    x = "Heart Attack Risk?",
    y = "Exercise Hours per Week")
```



#summary statistics for exercise hours:

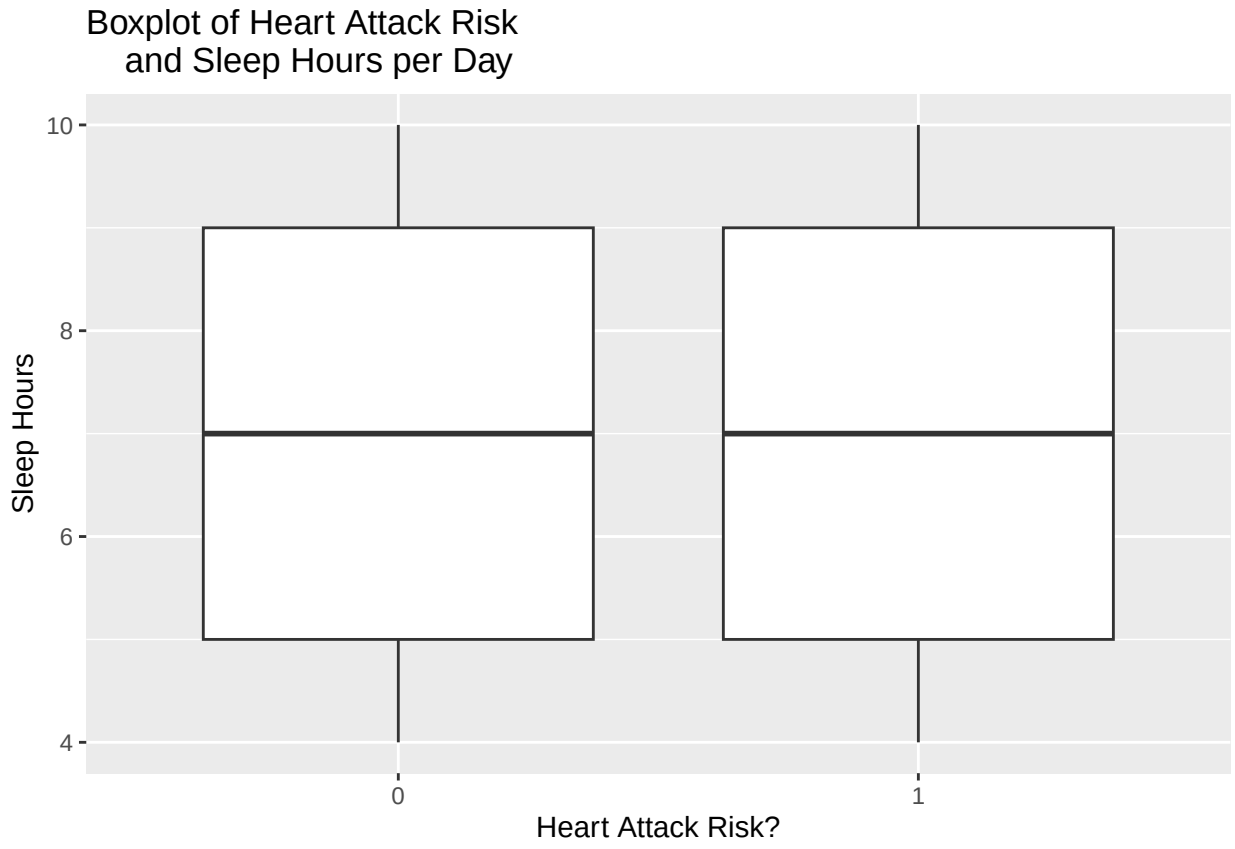
```
no_family_history_dataset %>%
  group_by(
    factor(heart_attack_risk)) %>%
    summarize(
    median = median(exercise_hrs_week),
    mean = mean(exercise_hrs_week),
    max = max(exercise_hrs_week),
    min = min(exercise_hrs_week)
  )
```

```
## # A tibble: 2 x 5
##   `factor(heart_attack_risk)` median  mean   max     min
##   <fct>                        <dbl> <dbl> <dbl>   <dbl>
## 1 0                             10.0  9.99  20.0 0.00244
## 2 1                             10.3 10.2   20.0 0.00511
```

#Boxplot for sleep hours explanatory variable:

```
no_family_history_dataset %>%
  ggplot() +
  geom_boxplot(
    mapping = aes(
      x = factor(heart_attack_risk),
      y = sleep_hrs)) +
```

```
  labs(
    title = "Boxplot of Heart Attack Risk
    and Sleep Hours per Day",
    x = "Heart Attack Risk?",
    y = "Sleep Hours")
```



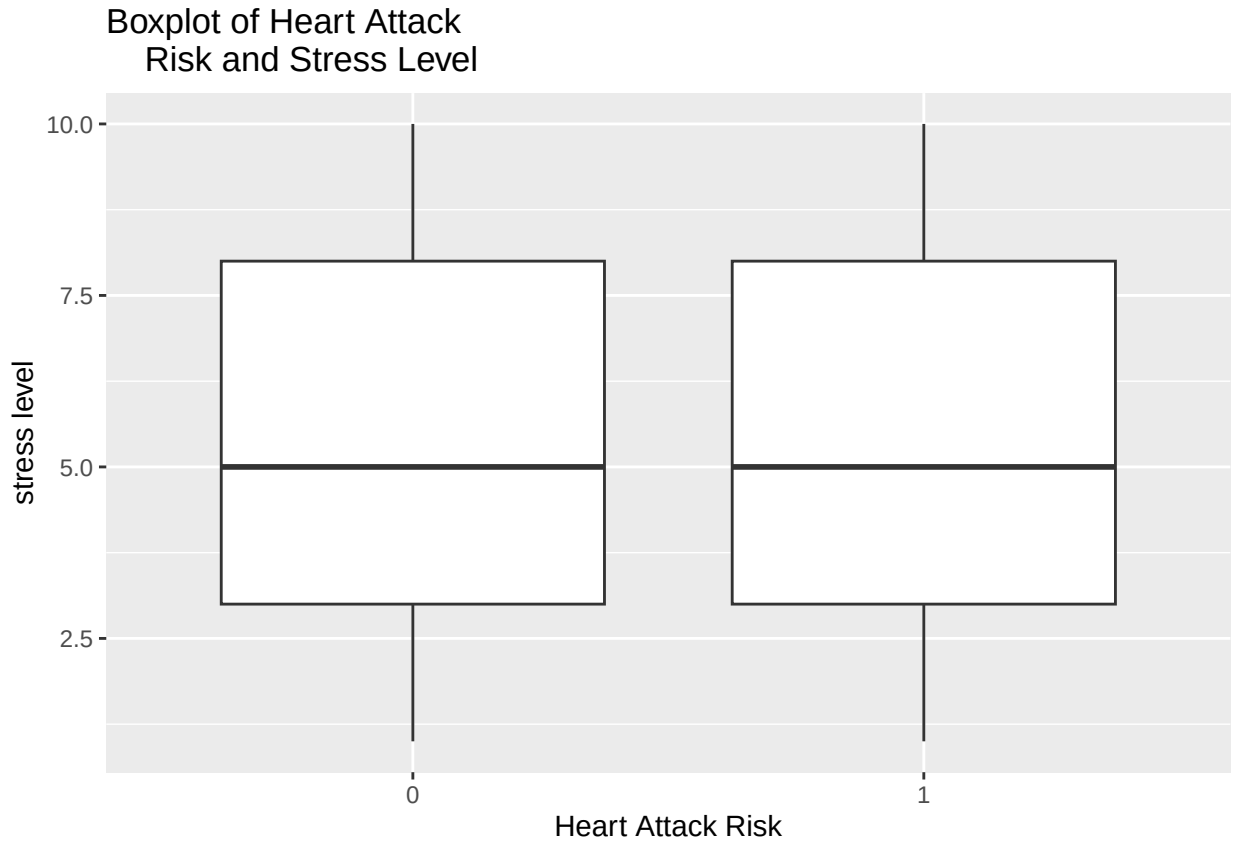
#Summary statistics of sleep hours:

```
no_family_history_dataset %>%
  group_by(
    factor(heart_attack_risk)) %>%
    summarize(
    median = median(sleep_hrs),
    mean = mean(sleep_hrs),
    max = max(sleep_hrs),
    min = min(sleep_hrs)
  )
```

```
## # A tibble: 2 x 5
##   `factor(heart_attack_risk)` median  mean   max   min
##   <fct>                        <dbl> <dbl> <dbl> <dbl>
## 1 0                                7  7.07    10     4
## 2 1                                7  7.01    10     4
```

#Boxplot of stress level explanatory variable:

```r
no_family_history_dataset %>%
  ggplot() +
  geom_boxplot(
    mapping = aes(
      x = factor(heart_attack_risk),
      y = stress_level)) +
  labs(
    title = "Boxplot of Heart Attack
    Risk and Stress Level",
    x = "Heart Attack Risk",
    y = "stress level")
```



#Summary statistics of stress level explanatory variable:

```r
no_family_history_dataset %>%
  group_by(
    factor(heart_attack_risk)) %>%
    summarize(
    median = median(stress_level),
    mean = mean(stress_level),
    max = max(stress_level),
    min = min(stress_level)
  )
```

```
## # A tibble: 2 x 5
##   `factor(heart_attack_risk)` median  mean   max   min
```

```
##   <fct>                          <dbl> <dbl> <dbl> <dbl>
## 1 0                                  5  5.43    10     1
## 2 1                                  5  5.42    10     1
```

#Splitting the stress levels into "not stressed" and "stressed":

```
no_family_history_dataset <- no_family_history_dataset %>%
  mutate(
    count =
      ifelse(
        stress_level >= 1
        & !(stress_level >=6)
        & heart_attack_risk == 1,
        "no stress & risk",
      ifelse(
        stress_level >= 1
        & !(stress_level >=6)
        & heart_attack_risk == 0,
        "no stress & no risk",
      ifelse(
        stress_level >= 6
        & heart_attack_risk == 1,
        "stress & risk",
      ifelse(
        stress_level >= 6
        & heart_attack_risk == 0,
        "stress and no risk", NA))))
  )
```

#Counted how many people were stressed and at a risk of heart attacks:

```
no_family_history_dataset %>%
  group_by(factor(count)) %>%
  summarize(n = n())
```

```
## # A tibble: 4 x 2
##   `factor(count)`         n
##   <fct>               <int>
## 1 no stress & no risk  1465
## 2 no stress & risk      812
## 3 stress & risk         783
## 4 stress and no risk   1383
```

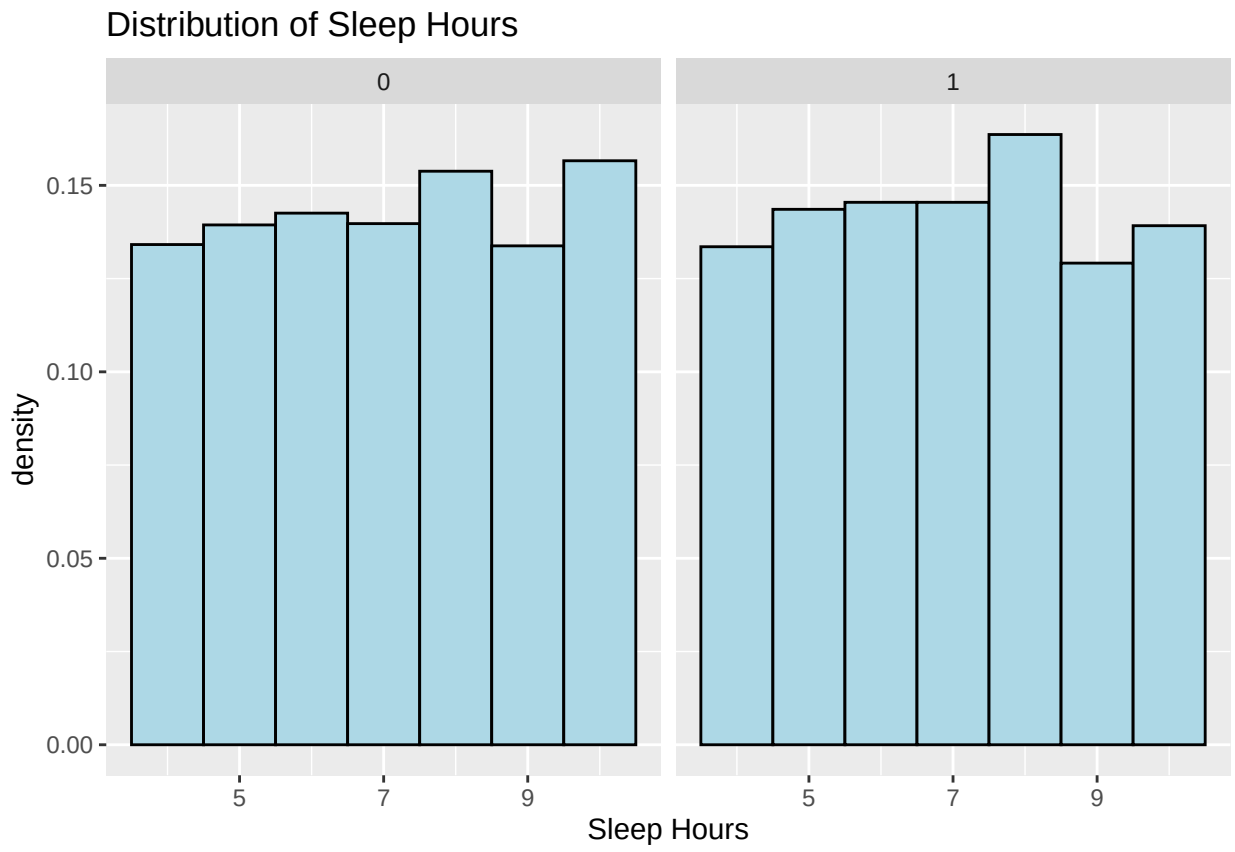
#Bar blot showing distribution of sleep Hours:

```
no_family_history_dataset$sleep_hrs <- as.numeric(no_family_history_dataset$sleep_hrs)
```

```
no_family_history_dataset %>%
  ggplot()+
  geom_histogram(
    mapping = aes(x = sleep_hrs,
                  y = ..density..),
```

```
    bins = 7,
    fill = "light blue",
    color = "black")+
  labs(
    title = "Distribution of Sleep Hours",
    x = "Sleep Hours")+
  facet_wrap(
    ~heart_attack_risk,
    scales = "free_x")
```

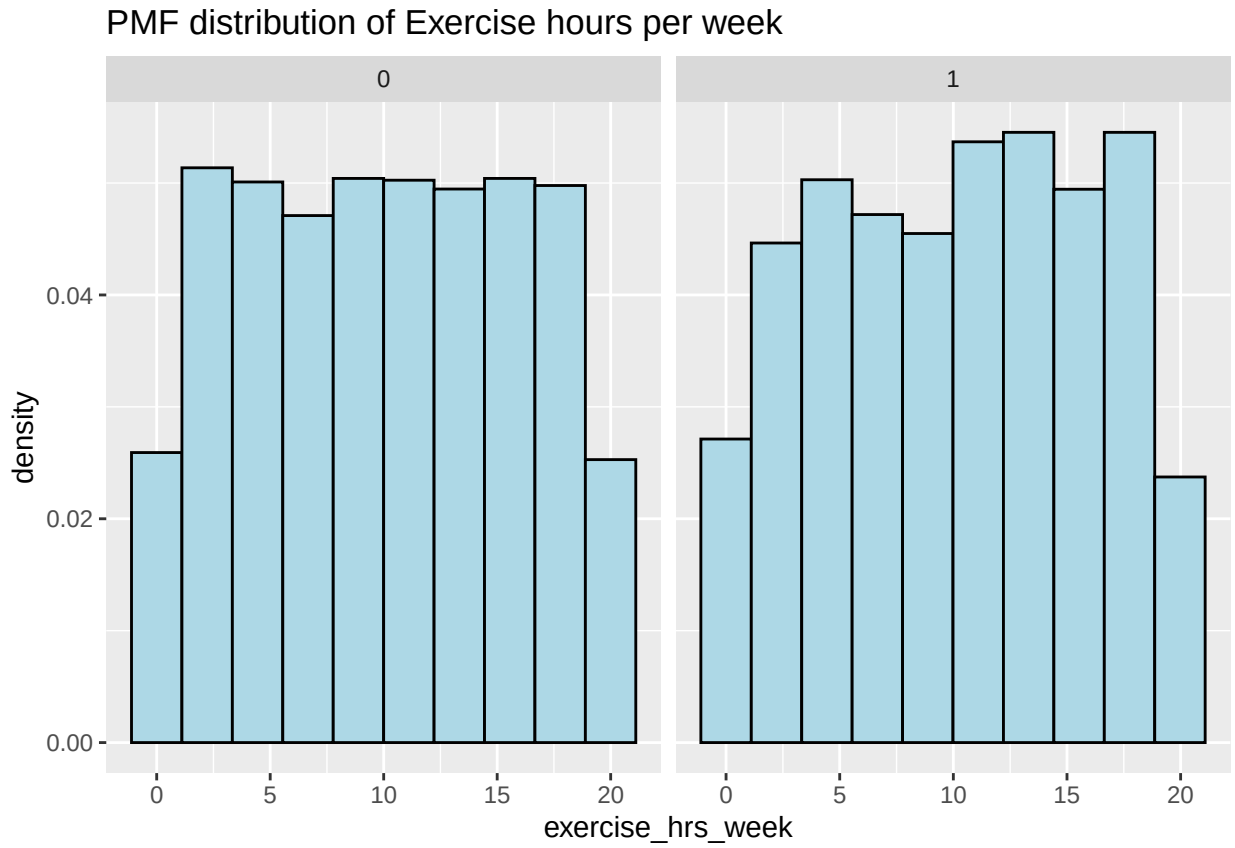
```
## Warning: The dot-dot notation (‘..density..‘) was deprecated in ggplot2 3.4.0.
## i Please use ‘after_stat(density)‘ instead.
## This warning is displayed once every 8 hours.
## Call ‘lifecycle::last_lifecycle_warnings()‘ to see where this warning was
## generated.
```



#PMF plot showing distribution of Exercise Hours per week:

```
no_family_history_dataset %>%
  ggplot() +
  geom_histogram(
    mapping = aes(
      x = exercise_hrs_week,
      y = ..density..),
    bins = 10,
```

```
    fill = "light blue",
    color = "black")+
  labs(
    title =
      "PMF distribution of Exercise hours per week")+
  facet_wrap(
    ~heart_attack_risk,
    scales = "free_x")
```

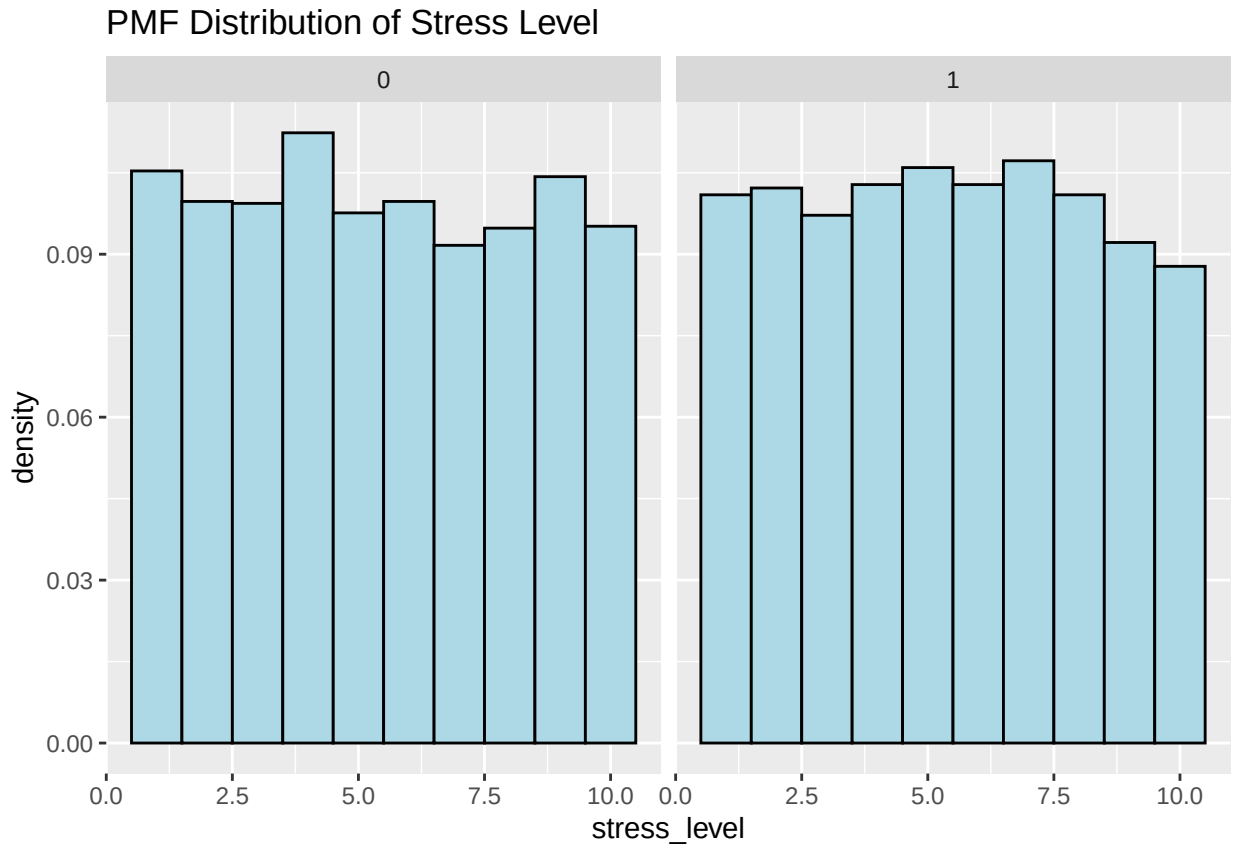


#Histogram showing distribution of stress level:

```
no_family_history_dataset$stress_level <- as.numeric(no_family_history_dataset$stress_level)
```

```
no_family_history_dataset %>%
  ggplot() +
  geom_histogram(
    mapping = aes(
      x = stress_level,
      y = ..density..),
    bins = 10,
    fill = "light blue",
    color = "black")+
  labs(
    title =
      "PMF Distribution of Stress Level")+
  facet_wrap(
```

```
    ~heart_attack_risk,
    scales = "free_x")
```



#count of people with heart attack risk among those with no family history.

```
no_family_history_dataset %>%
  group_by(factor(heart_attack_risk)) %>%
  summarize(n = n())
```

```
## # A tibble: 2 x 2
##   `factor(heart_attack_risk)`     n
##   <fct>                       <int>
## 1 0                            2848
## 2 1                            1595
```

#count of people with heart attack risk among those with family history.

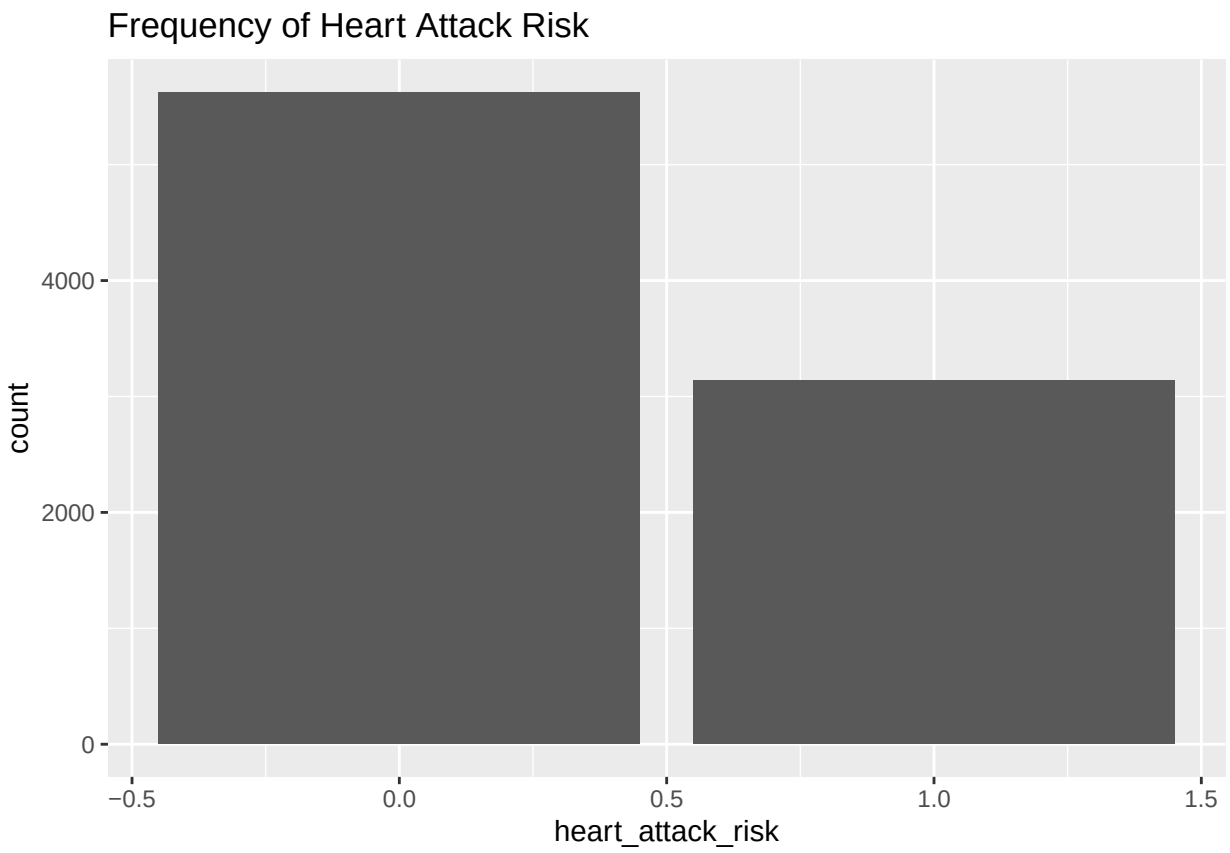
```
family_history_dataset %>%
  group_by(
    factor(heart_attack_risk)) %>%
  summarize(n = n())
```

```
## # A tibble: 2 x 2
##   `factor(heart_attack_risk)`     n
```

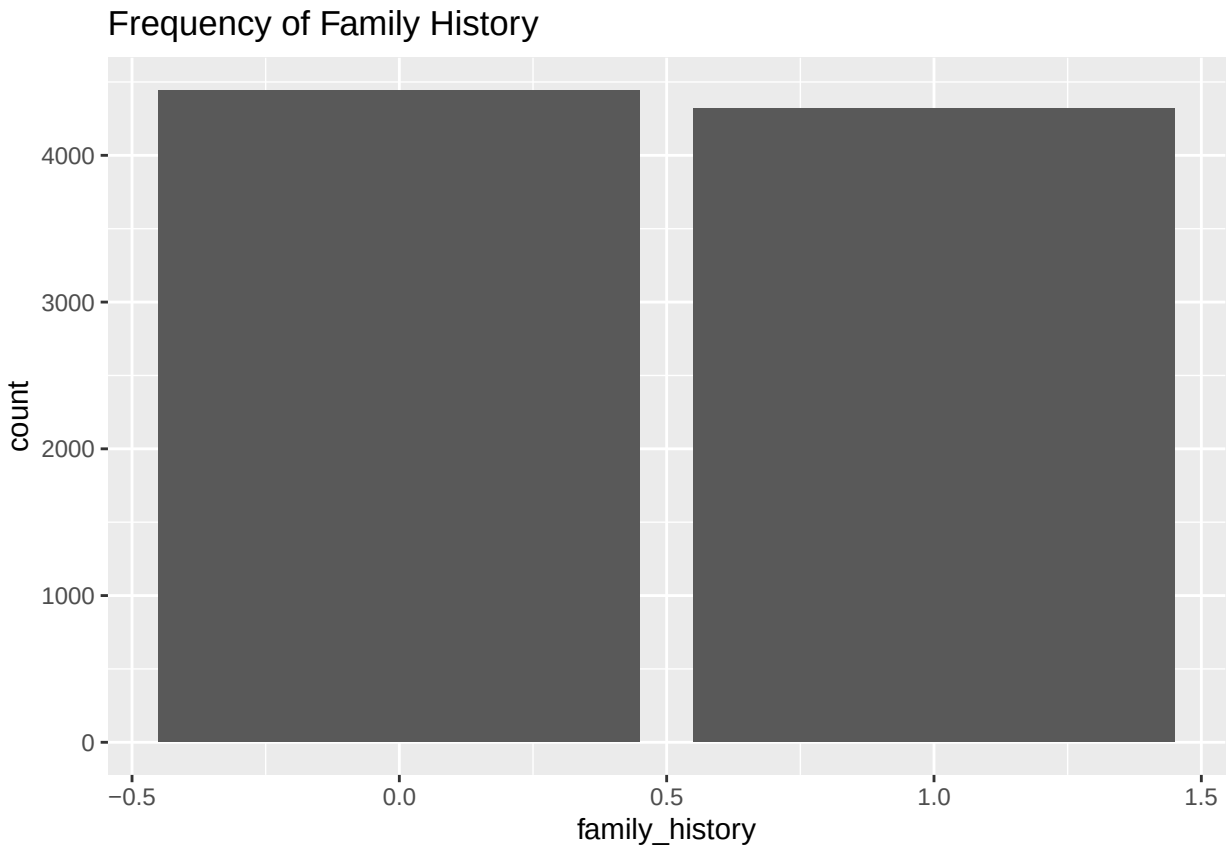
```
##   <fct>                       <int>
## 1 0                            2776
## 2 1                            1544
```

#Barplot showing the distribution of people with different combinations of heart attack risk and family history #Why is the count messed up!

```
untidied_dataset %>%
  ggplot()+
  geom_bar(
    mapping = aes(
      x = heart_attack_risk)
  )+
  labs(title = "Frequency of Heart Attack Risk")
```




```
untidied_dataset %>%
  ggplot()+
  geom_bar(
    mapping = aes(
      x = family_history)
  )+
  labs(title = "Frequency of Family History")
```



#Modeling (Giselle Rahimi) #Modeling (exercise_hrs_week)

```
split <- initial_split(no_family_history_dataset, prop = 0.7)
trainData <- training(split)
testData <-testing(split)
```

```
trainData %>%
  summarize(
    total = n(),
    missing = sum(is.na(heart_attack_risk)),
    fraction_missing = sum(is.na(heart_attack_risk)/total)
  )
```

```
## # A tibble: 1 x 3
##   total missing fraction_missing
##   <int>   <int>            <dbl>
## 1  3110       0                0
```

```
model_1 <-
  glm(
  heart_attack_risk ~ exercise_hrs_week,
  data = trainData,
  family = binomial()
)
```

```
model_1_preds <-
  trainData %>%
  add_predictions(
    model_1,
    type = "response"
  ) %>%
  mutate(
    outcome =
      if_else(
        pred >= 0.36,
                        true = 1,
                        false = 0))
```

```
model_1_preds %>%
  mutate(
    correct = if_else(
      outcome == heart_attack_risk,
      true = 1,
      false = 0
    )
  ) %>%
  summarize(
    total_correct = sum(correct),
    accuracy = total_correct/n()
  )
```

```
## # A tibble: 1 x 2
##   total_correct accuracy
##           <dbl>    <dbl>
## 1          1887    0.607
```

```
logistic_cv1 <-
  cv.glm(
    trainData,
    model_1, K =5)
```

```
logistic_cv1$delta
```

```
## [1] 0.2304086 0.2303714
```

#Modeling (sleep_hrs)

```
model_2 <-
  glm(
  heart_attack_risk ~ sleep_hrs,
  data = trainData,
  family = binomial()
)
```

```
model_2_preds <-
  trainData %>%
```

```
  add_predictions(
    model_2,
    type = "response"
  ) %>%
  mutate(
    outcome =
      if_else(
        pred >0.36,
                        true = 1,
                        false = 0))
```

```
model_2_preds %>%
  mutate(
    correct =
      if_else(
      outcome == heart_attack_risk,
      true = 1,
      false = 0
    )
  ) %>%
  summarize(
    total_correct = sum(correct),
    accuracy = total_correct/n()
  )
```

```
## # A tibble: 1 x 2
##   total_correct accuracy
##           <dbl>    <dbl>
## 1          1636    0.526
```

```
logistic_cv2 <-
  cv.glm(
    trainData,
    model_2,
    K = 5)
```

```
logistic_cv2$delta
```

```
## [1] 0.2301777 0.2301645
```

#Third model:

```
model_3 <-
  glm(
  heart_attack_risk ~ stress_level,
  data = trainData,
  family = binomial()
)
```

```
model_3_preds <-
  trainData %>%
```

```
  add_predictions(
    model_3,
    type = "response"
  ) %>%
  mutate(
    outcome =
      if_else(
        pred >0.36,
                        true = 1,
                        false = 0))
```

```
model_3_preds %>%
  mutate(
    correct =
      if_else(
      outcome == heart_attack_risk,
      true = 1,
      false = 0
    )
  ) %>%
  summarize(
    total_correct = sum(correct),
    accuracy = total_correct/n()
  )
```

```
## # A tibble: 1 x 2
##   total_correct accuracy
##           <dbl>    <dbl>
## 1          1673    0.538
```

```
logistic_cv3 <-
  cv.glm(
    trainData,
    model_3,
    K = 5)
```

```
logistic_cv3$delta
```

```
## [1] 0.2303995 0.2303624
```

#Fourth model:

```
model_4 <-
  glm(
  heart_attack_risk ~ exercise_hrs_week + sleep_hrs + stress_level,
  data = trainData,
  family = binomial()
)
```

```
model_4_preds <-
  trainData %>%
```

```
  add_predictions(
    model_4,
    type = "response"
  ) %>%
  mutate(
    outcome =
      if_else(pred >= 0.36,
                       true = 1,
                       false = 0))
```

```
model_4_preds %>%
summarize(
  min = min(outcome),
  max = max(outcome)
)
```

```
## # A tibble: 1 x 2
##     min   max
##   <dbl> <dbl>
## 1     0     1
```

```
model_4_preds %>%
summarize(
  min = min(pred)
)
```

```
## # A tibble: 1 x 1
##     min
##   <dbl>
## 1 0.347
```

```
model_4_preds %>%
  arrange(desc(pred))
```

```
## # A tibble: 3,110 x 18
##      age sex    cholesterol systolic diastolic heart_rate family_history
##    <dbl> <chr>        <dbl> <chr>    <chr>          <dbl>          <dbl>
##  1    75 Female         334 157      103               92              0
##  2    48 Male           212 180      78               108              0
##  3    90 Female         244 100      60                85              0
##  4    37 Male           272 132      88                44              0
##  5    27 Male           327 100      105               89              0
##  6    19 Female         296 98       93                97              0
##  7    67 Male           251 147      107               69              0
##  8    20 Female         165 119      79                96              0
##  9    60 Female         335 111      74                62              0
## 10    90 Female         160 170      64                56              0
## # i 3,100 more rows
## # i 11 more variables: exercise_hrs_week <dbl>, stress_level <dbl>,
## #   sleep_hrs <dbl>, heart_attack_risk <dbl>, triglycerides <dbl>,
## #   physical_days_week <dbl>, sedentary_hrs_day <dbl>, diet <chr>, count <chr>,
## #   pred <dbl>, outcome <dbl>
```

```r
model_4_preds %>%
  mutate(
    correct =
      if_else(
      outcome == heart_attack_risk,
      true = 1,
      false = 0
    )
  ) %>%
  summarize(
    total_correct = sum(correct),
    accuracy = total_correct/n()
  )
```

```
## # A tibble: 1 x 2
##   total_correct accuracy
##           <dbl>    <dbl>
## 1          1648    0.530
```

```r
logistic_cv4 <-
  cv.glm(
    trainData,
         model_4,
    K = 5)
```

```r
logistic_cv4$delta
```

```
## [1] 0.2309549 0.2308533
```

##Model 5:

```r
model_5 <-
  glm(
  heart_attack_risk ~ stress_level + exercise_hrs_week,
  data = trainData,
  family = binomial()
)
```

```r
model_5_preds <-
  trainData %>%
  add_predictions(
    model_5,
    type = "response"
  ) %>%
  mutate(
    outcome =
      if_else(
        condition = pred >0.36,
                      true = 1,
                      false = 0))
```

```
model_5_preds %>%
  mutate(
    correct =
      if_else(
      condition = heart_attack_risk == outcome,
      true = 1,
      false = 0
    )
  ) %>%
  summarize(
    total_correct = sum(correct),
    accuracy = total_correct/n()
  )
```

```
## # A tibble: 1 x 2
##   total_correct accuracy
##           <dbl>    <dbl>
## 1          1676    0.539
```

```
logistic_cv5 <-
  cv.glm(
    trainData,
         model_5,
    K = 5)
```

```
logistic_cv5$delta
```

```
## [1] 0.2303419 0.2303112
```

Model 1 has the highest accuracy!

Accuracy and cross validation error on test dataset:

```
model_1_preds_test <-
  testData %>%
  add_predictions(
    model_1,
    type = "response"
  ) %>%
  mutate(
    outcome =
      if_else(
        pred >= 0.36,
                        true = 1,
                        false = 0))
```

```
model_1_preds_test %>%
  mutate(
    correct = if_else(
      outcome == heart_attack_risk,
```

```
      true = 1,
      false = 0
    )
  ) %>%
  summarize(
    total_correct = sum(correct),
    accuracy = total_correct/n()
  )
```

```
## # A tibble: 1 x 2
##   total_correct accuracy
##           <dbl>    <dbl>
## 1           823    0.617
```

```
logistic_cv1_test <-
  cv.glm(
    testData,
    model_1, K =5)
```

```
## Warning in y - yhat: longer object length is not a multiple of shorter object
## length

## Warning in y - yhat: longer object length is not a multiple of shorter object
## length

## Warning in y - yhat: longer object length is not a multiple of shorter object
## length

## Warning in y - yhat: longer object length is not a multiple of shorter object
## length

## Warning in y - yhat: longer object length is not a multiple of shorter object
## length
```

```
logistic_cv1_test$delta
```

```
## [1] 0.2285452 0.2288119
```

#Hypothesis testing: (Sukyoung Yoon)

#Change the heart_attack_risk variable to a character object.

```
  no_family_history_dataset$heart_attack_risk <-
  as.character(
    no_family_history_dataset$heart_attack_risk)
```

#Separated the stress_level variable into two categories (True indicates stress and False indicates no stress)

```
no_family_history_dataset <-
  no_family_history_dataset %>%
  mutate(
```

```
    stress =
      ifelse(
        stress_level >= 6,
        "True",
      ifelse(
        stress_level >= 1,
        "False", NA
  )))
```

Changing the stress variable to a character

```
  no_family_history_dataset$stress <-
  as.character(
    no_family_history_dataset$stress)
```

```
str(no_family_history_dataset)
```

```
## tibble [4,443 x 17] (S3: tbl_df/tbl/data.frame)
##  $ age               : num [1:4443] 67 21 90 84 20 38 50 60 66 45 ...
##  $ sex               : chr [1:4443] "Male" "Female" "Male" "Male" ...
##  $ cholesterol       : num [1:4443] 208 324 358 220 145 166 303 145 340 294 ...
##  $ systolic          : chr [1:4443] "158" "174" "102" "131" ...
##  $ diastolic         : chr [1:4443] "88" "99" "73" "68" ...
##  $ heart_rate        : num [1:4443] 72 72 84 107 68 56 104 71 69 66 ...
##  $ family_history    : num [1:4443] 0 0 0 0 0 0 0 0 0 0 ...
##  $ exercise_hrs_week : num [1:4443] 4.17 2.08 4.1 3.43 16.87 ...
##  $ stress_level      : num [1:4443] 9 9 7 4 5 9 1 8 1 9 ...
##  $ sleep_hrs         : num [1:4443] 6 4 10 7 4 6 5 7 10 6 ...
##  $ heart_attack_risk : chr [1:4443] "0" "0" "1" "1" ...
##  $ triglycerides     : num [1:4443] 286 587 284 370 790 402 517 247 747 360 ...
##  $ physical_days_week: num [1:4443] 0 4 4 6 7 0 1 7 1 4 ...
##  $ sedentary_hrs_day : num [1:4443] 6.615 9.463 0.627 10.544 11.349 ...
##  $ diet              : chr [1:4443] "Average" "Healthy" "Healthy" "Average" ...
##  $ count             : chr [1:4443] "stress and no risk" "stress and no risk" "stress & risk" "no st
##  $ stress            : chr [1:4443] "True" "True" "True" "False" ...
```

#Test for a difference in proportions (True - False) with stress

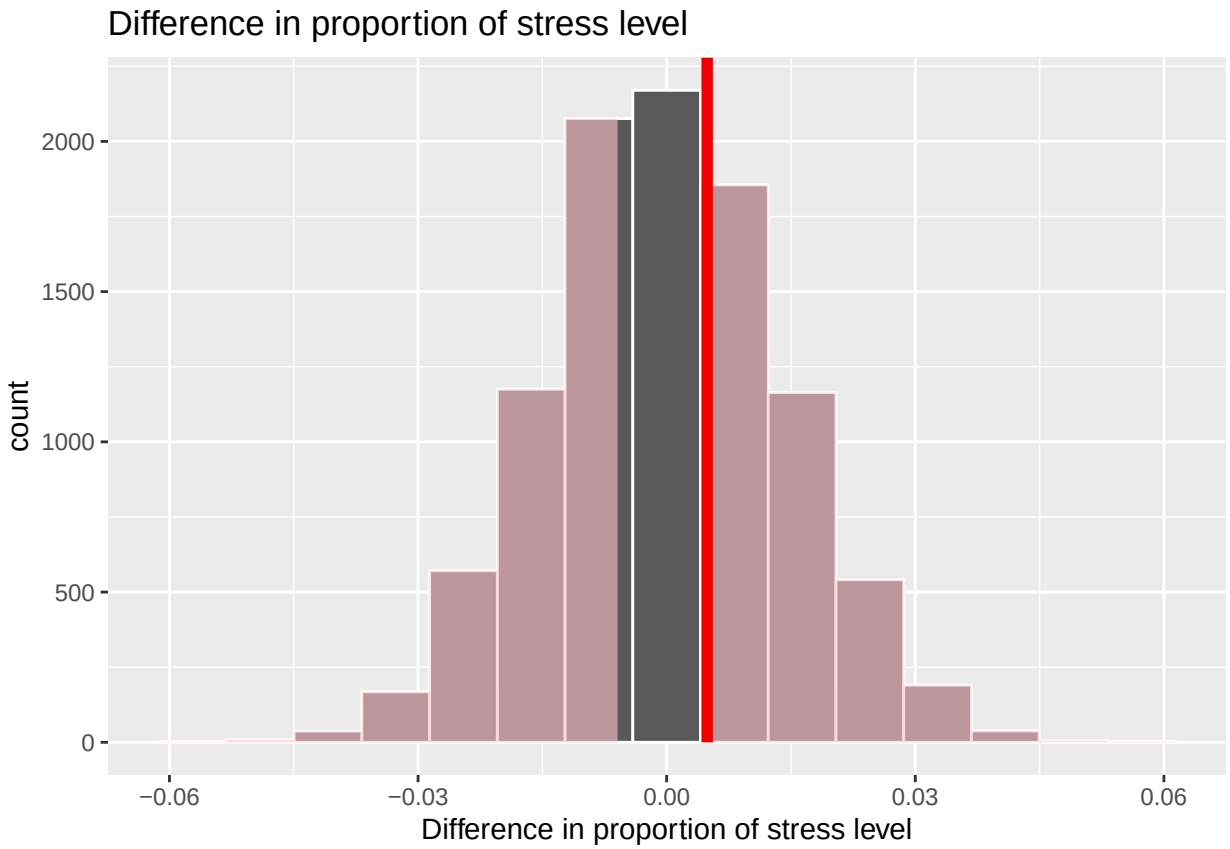
```
heart_null <- no_family_history_dataset %>%
  specify(
    heart_attack_risk ~ stress,
    success = "1") %>%
  hypothesize(
    null ="independence") %>%
  generate(
    reps = 10000,
    type = "permute") %>%
  calculate(
    stat = "diff in props",
    order = c("True", "False"))
```

```
heart_obs_stat <-
  no_family_history_dataset %>%
  specify(
    heart_attack_risk ~ stress,
    success = "1") %>%
  calculate(
    stat = "diff in props",
    order = c("True", "False"))
```

```
heart_null %>%
  get_p_value(
    obs_stat = heart_obs_stat,
    direction = "both")
```

```
## # A tibble: 1 x 1
##   p_value
##     <dbl>
## 1   0.759
```

```
heart_null %>%
  visualize() +
  shade_p_value(
    obs_stat = heart_obs_stat,
    direction = "both") +
  labs(
    title = "Difference in proportion of stress level",
    x = "Difference in proportion of stress level",
    y = "count")
```




```
  no_family_history_dataset <-
  no_family_history_dataset %>%
  mutate(
    yes_no =
      ifelse(
        heart_attack_risk == 0,
        "yes",
      ifelse(
        heart_attack_risk == 1,
        "no", NA
  )))
```

```
unique(
  no_family_history_dataset$sleep_hrs)
```

```
## [1]  6  4 10  7  5  8  9
```

#separating sleep into high and low

```r
no_family_history_dataset <-
  no_family_history_dataset %>%
  mutate(
    sleep =
      ifelse(
        sleep_hrs >= 7,
        "High",
      ifelse(
        sleep_hrs < 7,
        "Low", NA
  )))
```

```r
heart_null2 <- no_family_history_dataset %>%
  specify(
    yes_no ~ sleep,
    success = "yes") %>%
  hypothesize(
    null = "independence") %>%
  generate(
    reps = 10000,
    type = "permute") %>%
  calculate(
    stat = "diff in props",
    order=c("High", "Low"))
```

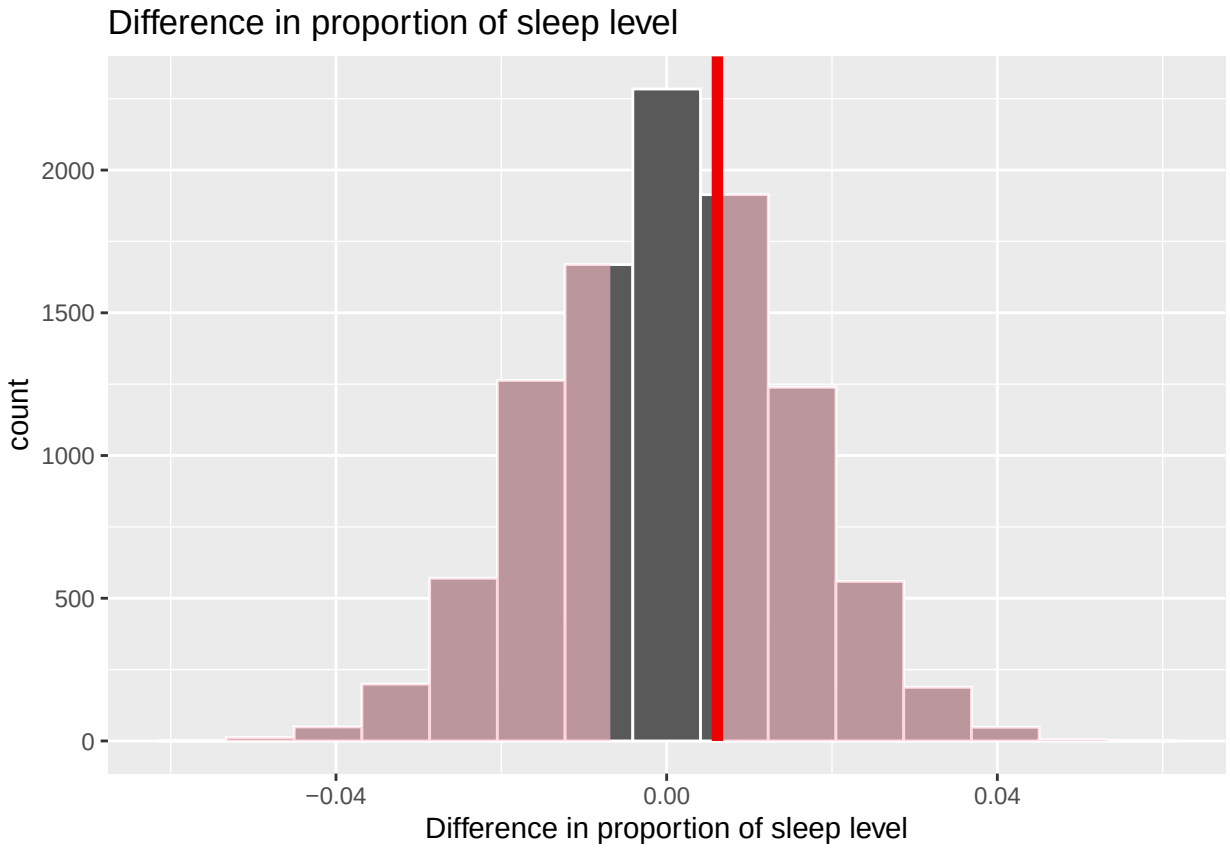
```r
heart_obs_2 <-
  no_family_history_dataset %>%
  specify(
    yes_no ~ sleep,
    success = "yes") %>%
  calculate(
    stat = "diff in props",
    order=c("High", "Low"))
```

```r
heart_null2 %>%
  get_p_value(
    obs_stat = heart_obs_2,
    direction = "both")
```

```
## # A tibble: 1 x 1
##   p_value
##     <dbl>
## 1   0.704
```

```r
heart_null2 %>%
  visualize() +
  shade_p_value(
    obs_stat =
      heart_obs_2,
    direction = "both") +
  labs(
    title = "Difference in proportion of sleep level",
```

```
    x = "Difference in proportion of sleep level",
    y = "count")
```



#Hypothesis test difference in proprotions for exercise_hrs:

#Separating exercise hours into high and low categories.

```
no_family_history_dataset <- no_family_history_dataset %>%
  mutate(
    exercise =
      ifelse(
        exercise_hrs_week >= 10,
        "High",
      ifelse(exercise_hrs_week < 10,
             "Low", NA
  )))
```

```
heart_null3 <-
  no_family_history_dataset %>%
  specify(
    yes_no ~ exercise,
    success = "yes") %>%
  hypothesize(
    null = "independence") %>%
  generate(
    reps = 10000,
```

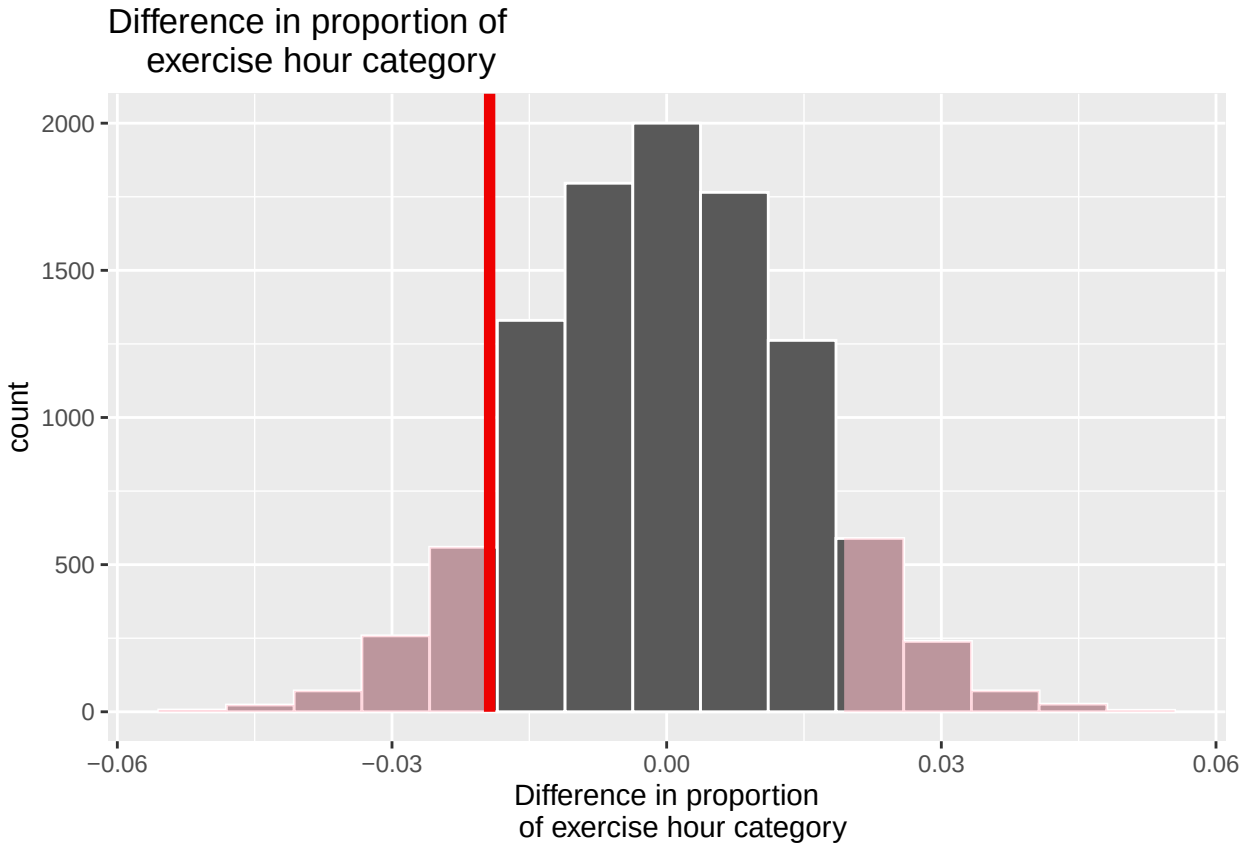
```
    type = "permute") %>%
  calculate(
    stat = "diff in props",
    order=c("High", "Low"))
```

```
heart_obs_3 <-
  no_family_history_dataset %>%
  specify(
    yes_no ~ exercise,
    success = "yes") %>%
  calculate(
    stat = "diff in props",
    order=c("High", "Low"))
```

```
heart_null3 %>%
  get_p_value(
    obs_stat = heart_obs_3,
    direction = "both")
```

```
## # A tibble: 1 x 1
##   p_value
##     <dbl>
## 1   0.183
```

```
heart_null3 %>%
  visualize() +
  shade_p_value(
    obs_stat = heart_obs_3,
    direction = "both") +
  labs(
    title = "Difference in proportion of
    exercise hour category",
    x = "Difference in proportion
    of exercise hour category",
    y = "count")
```


Difference in proportion of
exercise hour category
count
2000
1500
1000
500
0
−0.06
−0.03
0.00
0.03
0.06
Difference in proportion
of exercise hour category