What Your Heart Rate is Telling You

Introduction

Millions of people develop some sort of heart disease every year, and heart disease is the biggest killer of both men and women in the United States and around the world. More than 17 million people died of heart disease in 2017. Statistical analysis has identified many risk factors associated with heart disease such as age, blood pressure, total cholesterol, diabetes, hypertension, family history of heart disease, obesity, lack of physical exercise, etc. We will run statistical tests and regression models using the Cleveland heart disease dataset to assess one particular factor -- the maximum heart rate one can achieve during exercise and associated with a higher likelihood of getting heart disease.

The dataset

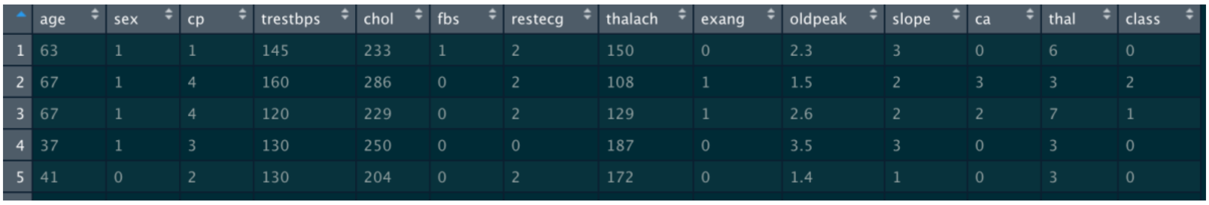
We are using a dataset from the UCI Machine Learning repository(Source: https://archive.ics.uci.edu/ml/datasets/Heart+Disease). It has the following column:

Name	Type	Description
Age	Continuous	Patient's age in years
Sex	Discrete	0 = female, 1 = male
Cp	Discrete	Chest pain type: 1 = typical angina, 2 = atypical angina, 3 = non-anginal pain, 4 = asymtopm.
Trestbps	Continuous	Resting blood pressure (mm Hg)
Chol	Continuous	Serum cholesterol in mg/dl
Fbs	Discrete	Fasting blood sugar>120 mg/dl? 1 = Yes, 0 = no
Exang Continuous Maximum heart rate achieved	Discrete	Exercise-induced angina: 1 = Yes, 0 = No
Thalach	Continuous	Maximum heart rate achieved
Old peak ST	Continuous	Depression induced by exercise relative to rest
Slope	Discrete	The slope of the peak exercise segment: 1 = up sloping, 2 = flat
Ca	Continuous	A number of major vessels were colored by fluoroscopy that ranged between 0 and 3.
Thal	Discrete	3 = normal, 6 = fixed defect, 7 = reversible defect.
Class	Discrete	Diagnosis classes: 0 = No Presence, 1 – 4 (The higher

		the number, the more likely have heart disease)

First, we load the dataset and packages. Then we take a look at the first few rows.

~~~
"`{r}

# Read datasets Cleveland_hd.csv into hd_data

hd_data <- read.csv("Cleveland_hd.csv")

# take a look at the first five rows of hd_data

head(hd_data,5)

```
~~~



	age	sex	cp	trestbps	chol	fbs	restecg	thalach	exang	oldpeak	slope	ca	thal	class
1	63	1	1	145	233	1	2	150	0	2.3	3	0	6	0
2	67	1	4	160	286	0	2	108	1	1.5	2	3	3	2
3	67	1	4	120	229	0	2	129	1	2.6	2	2	7	1
4	37	1	3	130	250	0	0	187	0	3.5	3	0	3	0
5	41	0	2	130	204	0	2	172	0	1.4	1	0	3	0

From the output above, we notice that the outcome variable *class* has more than two categories. Since we only want two categories for the output, with any non-zero values coded as an "event", we will create a new variable called *hd* to represent a binary 1 or 0 outcome. And also, there's variables that we can change the type of data into appropriate data type, such as sex.

~~~
```{r}

# Use the mutate() from dplyr to recode our data

hd_data %>% mutate(hd <- ifelse(class>0, 1, 0)) -> hd_data

# recode sex using mutate function and save as hd_data
hd_data %>% mutate(sex <- factor(sex, levels = 0:1, labels = c("Female", "Male"))) ->
hd_data
```
~~~

Empirical Analysis

Now, let's use statistical tests to see which predictors are related to heart disease. We can explore the associations for each variable in the dataset. Depending on the data type (i.e., continuous or categorical), we use a t-test or chi-squared test to calculate the p-values. Recall, the t-test is used to determine whether there is a significant difference between the means of two groups (e.g., is the mean age from group A different from the mean age from group B?). A chi-squared test for independence compares the equivalence of two proportions.

~~~~
```{r}
# Does sex have an effect? Sex is a binary variable in this dataset, so the appropriate test is chi-squared test
hd_sex <- chisq.test(hd_data$sex, hd_data$hd)

# Does age have an effect? Age is continuous, so we use a t-test
hd_age <- t.test(hd_data$age ~hd_data$hd)

# What about thalach? Thalach is continuous, so we use a t-test
hd_heartrate <- t.test(hd_data$thalach ~hd_data$hd)
```
```{r}
hd_sex
```
~~~~

```
        Pearson's Chi-squared test with Yates' continuity correction

data:  hd_data$sex and hd_data$hd
X-squared = 22.043, df = 1, p-value = 2.667e-06
```

From the output above, we have a p-value of less than 0.05 significance level, So we conclude that sex and disease status have a significant association between the two variables.

~~~~
```{r}
hd_age
```
~~~~

```
Welch Two Sample t-test

data:  hd_data$age by hd_data$hd
t = -4.0303, df = 300.93, p-value = 7.061e-05
alternative hypothesis: true difference in means is not equal to 0
95 percent confidence interval:
 -6.013385 -2.067682
sample estimates:
mean in group 0 mean in group 1
    52.58537        56.62590
```

From the output above, we have a p-value of less than 0.05 significance level, So we conclude that the average age from who owns a disease different from the average age who doesn't have heart disease.

~~~~
```{r}
hd_heartrate
```
~~~~

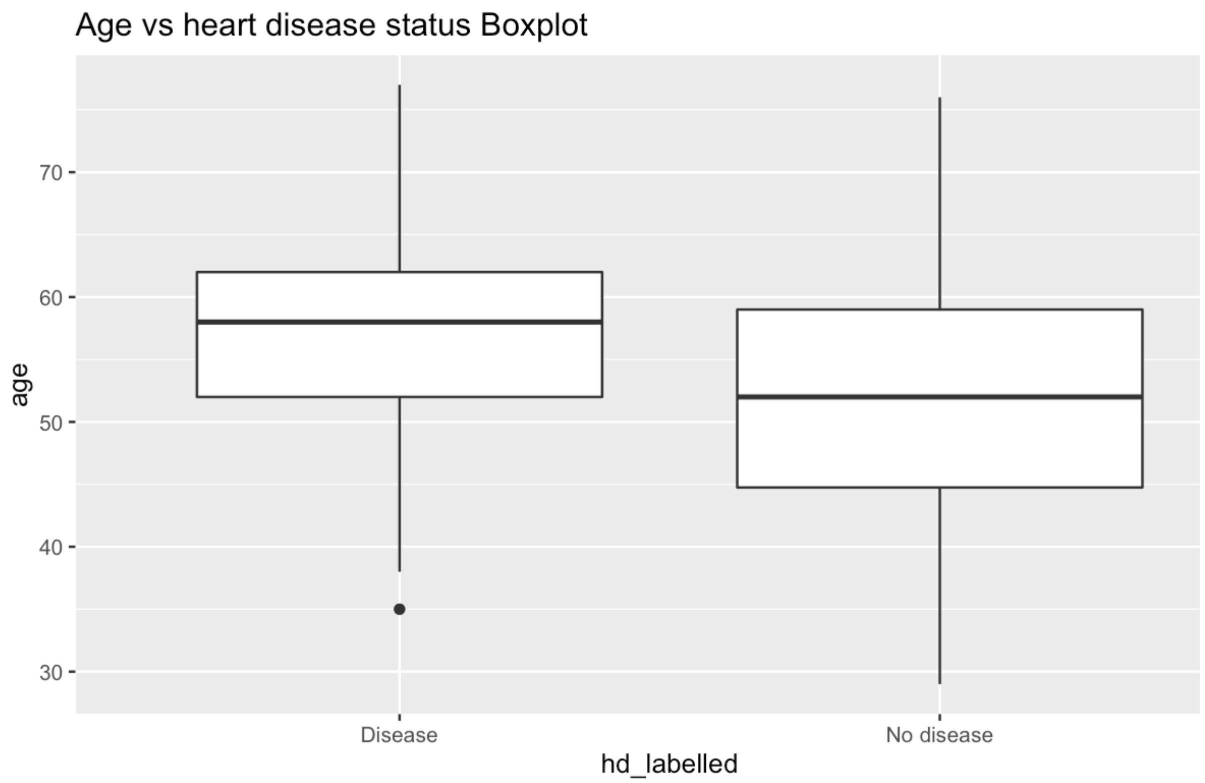
```
Welch Two Sample t-test

data:  hd_data$thalach by hd_data$hd
t = 7.8579, df = 272.27, p-value = 9.106e-14
alternative hypothesis: true difference in means is not equal to 0
```

```
95 percent confidence interval:
 14.32900 23.90912
sample estimates:
mean in group 0 mean in group 1
       158.378        139.259
```

From the output above, we have a p-value of less than 0.05 significance level, So we conclude that the average heart rate from who owns a disease different from the average heart rate who doesn't have heart disease.

Not only from many tests, but we can also see the associations graphically using a plot. In addition to p-values from statistical tests, we can plot the age, sex, and maximum heart rate distributions for our outcome variable. This will give us a sense of both the direction and magnitude of the relationship. First, let's plot age using a boxplot since it is a continuous variable.

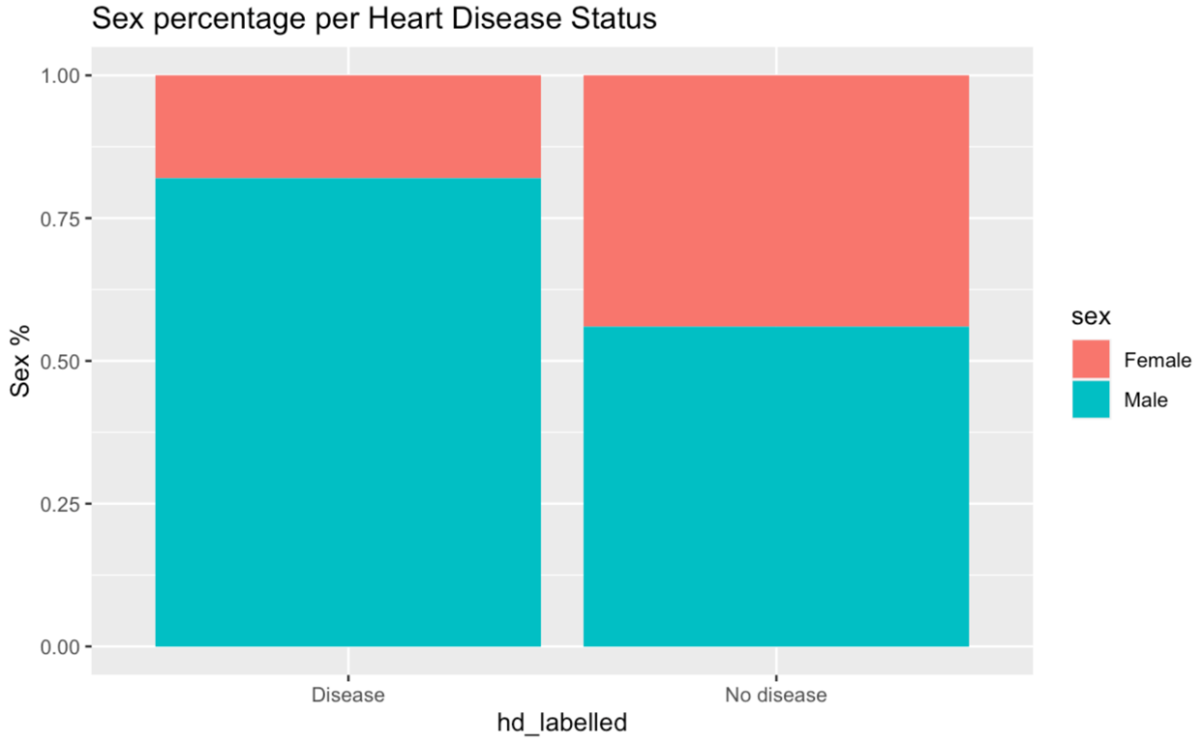
~~~
```{r}
# Recode hd to be labelled
hd_data %>% mutate(hd_labelled = ifelse(hd ==0, "No disease", "Disease")) -> hd_data

# Age vs hd
ggplot(data = hd_data, aes(x=hd_labelled, y = age)) + geom_boxplot() + labs(title="Age vs heart disease status Boxplot")
```
~~~



From this boxplot, we can conclude that the median age of patients with heart disease is greater than that of those without heart disease, Seems weird, right?

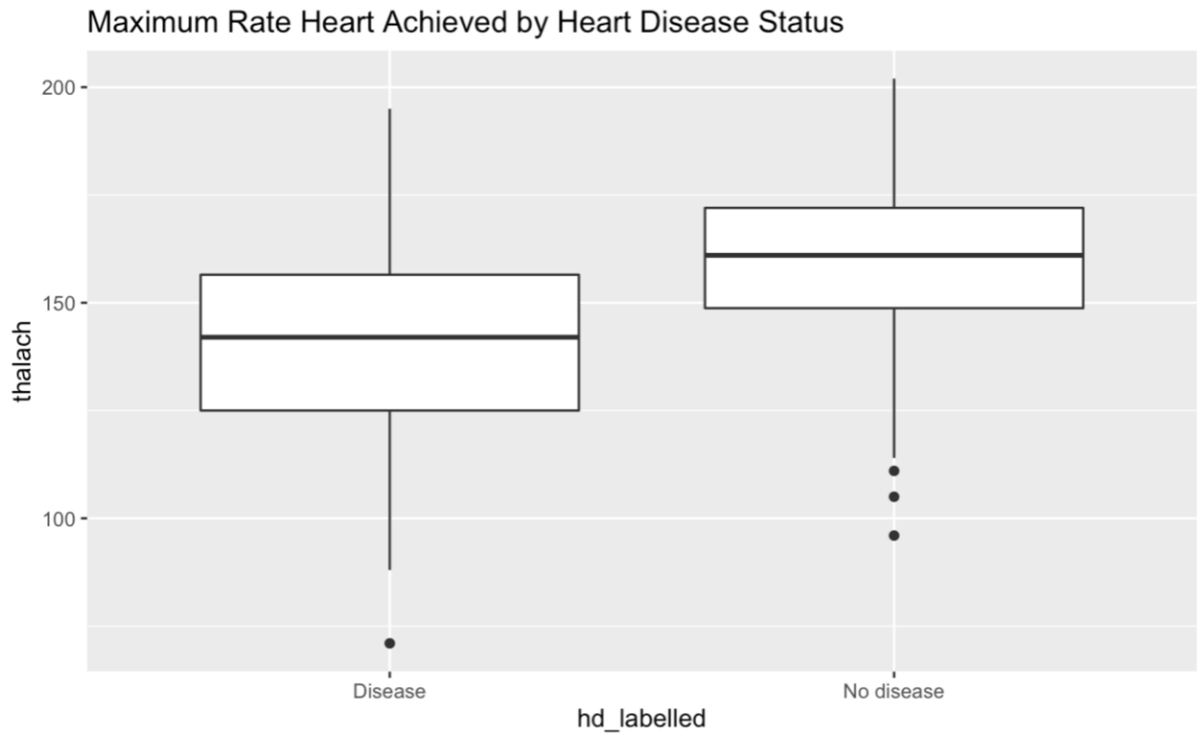
Next, let's plot sex using a bar plot since it is a binary variable in this dataset.

```{r}
# Sex vs hd
ggplot(data = hd_data, aes(x = hd_labelled, fill = sex)) + geom_bar(position = "fill") +
ylab("Sex %") + labs(title = "Sex percentage per Heart Disease Status")
```



And finally, let's plot thalach using a boxplot since it is a continuous variable.

```{r}
# max heart rate vs hd
ggplot(data = hd_data, aes(x=hd_labelled, y = thalach)) + geom_boxplot() + labs(title =
"Maximum Rate Heart Achieved by Heart Disease Status")
```



The plots and the statistical tests confirmed that all the three variables are highly significantly associated with our outcome (p<0.001 for all tests).

In general, we want to use multiple logistic regression when we have one binary outcome variable and two or more predicting variables. The binary variable is the dependent (Y) variable; we are studying the independent (X) variables' effect on the probability of obtaining a particular value of the dependent variable. For example, we might want to know the impact of maximum heart rate, age, and sex on the probability that a person will have heart disease in the next year. The model will also tell us the remaining effect of maximum heart rate after we control or adjust for the impact of the other two effectors.

The glm() command is designed to perform generalized linear models (regressions) on binary outcome data, count data, probability data, proportion data, and many other data types. In our case, the outcome is binary following a binomial distribution.

~~~
"`{r}
# Use glm() from base R and specify the family argument as binomial
model <- glm(data = hd_data, hd~age+sex+thalach, family = "binomial")

# Extract the model summary
summary(model)
```
~~~

~~~
Call:
glm(formula = hd ~ age + sex + thalach, family = "binomial",
    data = hd_data)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-2.2250  -0.8486  -0.4570   0.9043   2.1156
~~~

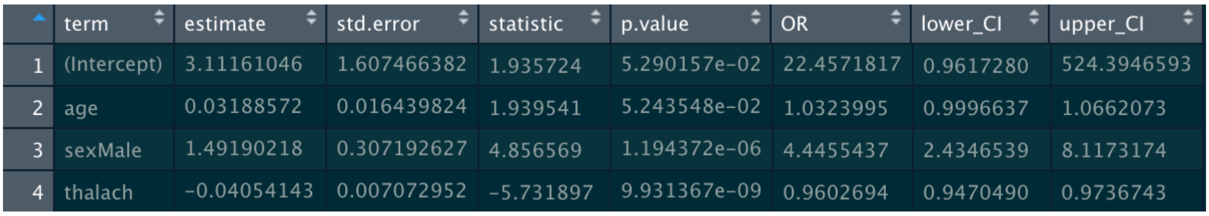
```
Coefficients:
            Estimate Std. Error z value Pr(>|z|)
(Intercept)  3.111610   1.607466   1.936   0.0529 .
age          0.031886   0.016440   1.940   0.0524 .
sexMale      1.491902   0.307193   4.857 1.19e-06 ***
thalach     -0.040541   0.007073  -5.732 9.93e-09 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 417.98  on 302  degrees of freedom
Residual deviance: 332.85  on 299  degrees of freedom
AIC: 340.85

Number of Fisher Scoring iterations: 4
```

The output above shows that variable age is insignificant to the model, while the remaining variables have a significant effect on the model.

It's common practice in medical research to report Odds Ratio (OR) to quantify how strongly the presence or absence of property A is associated with the presence or absence of the outcome. When the OR is greater than 1, we say A is positively associated with outcome B (increases the Odds of having B). Otherwise, we say A is negatively associated with B (decreases the Odds of having B).

The raw glm coefficient table (the 'estimate' column in the printed output) in R represents the outcome's log(Odds Ratios). Therefore, we need to convert the values to the original OR scale and calculate the corresponding 95% Confidence Interval (CI) of the estimated Odds Ratios when reporting results from logistic regression.

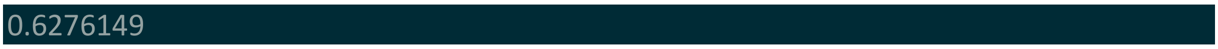
```
"`{r}
# Tidy up the coeffictient table
tidy_m <-tidy(model)

# Calculate OR
tidy_m$OR <- exp(tidy_m$estimate)

# Calculate 95% CI and save a slower CI and upper CI
tidy_m$lower_CI <- exp(tidy_m$estimate - 1.96 * tidy_m$std.error)
tidy_m$upper_CI <- exp(tidy_m$estimate + 1.96 * tidy_m$std.error)

# display the updated coefficient table
tidy_m
```



	term	estimate	std.error	statistic	p.value	OR	lower_CI	upper_CI
1	(Intercept)	3.11161046	1.607466382	1.935724	5.290157e-02	22.4571817	0.9617280	524.3946593
2	age	0.03188572	0.016439824	1.939541	5.243548e-02	1.0323995	0.9996637	1.0662073
3	sexMale	1.49190218	0.307192627	4.856569	1.194372e-06	4.4455437	2.4346539	8.1173174
4	thalach	-0.04054143	0.007072952	-5.731897	9.931367e-09	0.9602694	0.9470490	0.9736743

So far, we have built a logistic regression model and examined the model coefficients/ORs. We may wonder how we can use this model to predict a person's likelihood of having heart disease given their age, sex, and maximum heart rate. Furthermore, we'd like to translate the predicted probability into a decision rule for clinical use by defining a cutoff value on the probability scale. In practice, when an individual comes in for a health check-up, the doctor would like to know the predicted probability of heart disease for specific values of the predictors: a 50-year-old male with a max heart rate of 140. To do that, we create a data frame called newdata, in which we include the desired values for our prediction.

~~~
```{r}
# Get the predicted probability in our dataset
pred_prob <- predict(model, hd_data, type = "response")

# Create a decision rule using probability 0.5 as cutoff and save the predicted decision into the main data frame
hd_data$pred_hd <- ifelse(pred_prob >= 0.5, 1, 0)

# Create a new dataframe to save a new case information
newdata <- data.frame(age = 50, sex= "Male", thalach = 140)

# predict probability for this new case and print out the predicted value
p_new <- predict(model, newdata, type = "response")
p_new
```
~~~



```
0.6276149
```

Conclusion

After these metrics are calculated, we'll see (from the logistic regression OR table) that older age, being male, and having a lower max heart rate are all risk factors for heart disease. We can also apply our model to predict the probability of having heart disease. For a 45 years old male with a max heart rate of 140, our model generated a heart disease probability of 0.6276149, indicating a high risk of heart disease. Although our model has an overall accuracy of 0.71, some cases were misclassified, as shown in the confusion matrix. One way to improve our current model is to include other relevant predictors from the dataset into our model; try another model such as Neural Network or decision tree, etc.

~~~
```{r}
# Calculate auc, accuracy, classification error
auc <- auc(hd_data$hd, hd_data$pred_hd)
accuracy<- accuracy(hd_data$hd, hd_data$pred_hd)
classification_error <- ce(hd_data$hd, hd_data$pred_hd)

# Print out the mertics on to screen
~~~

~~~
print(paste("AUC=", auc))
print(paste("Accuracy=", accuracy))
print(paste("Classification Error", classification_error))

# Confusion matrix
table(hd_data$hd, hd_data$pred_hd, dnn=c("Actual Status", "Predicted Status"))
```
~~~

```
"AUC= 0.706483593612915"
"Accuracy= 0.70957095709571"
"Classification Error 0.29042904290429"
           Predicted Status
Actual Status   0   1
          0 122  42
          1  46  93
```

Source

- https://archive.ics.uci.edu/ml/datasets/Heart+Disease
- https://world-heart-federation.org/wp-content/uploads/2017/05/WCC2016_CVDs_infographic.pdf
- https://www.cdc.gov/heartdisease/facts.htm
- https://www.statology.org/chi-square-test-vs-t-test/
- https://psychscenehub.com/psychpedia/odds-ratio-2/