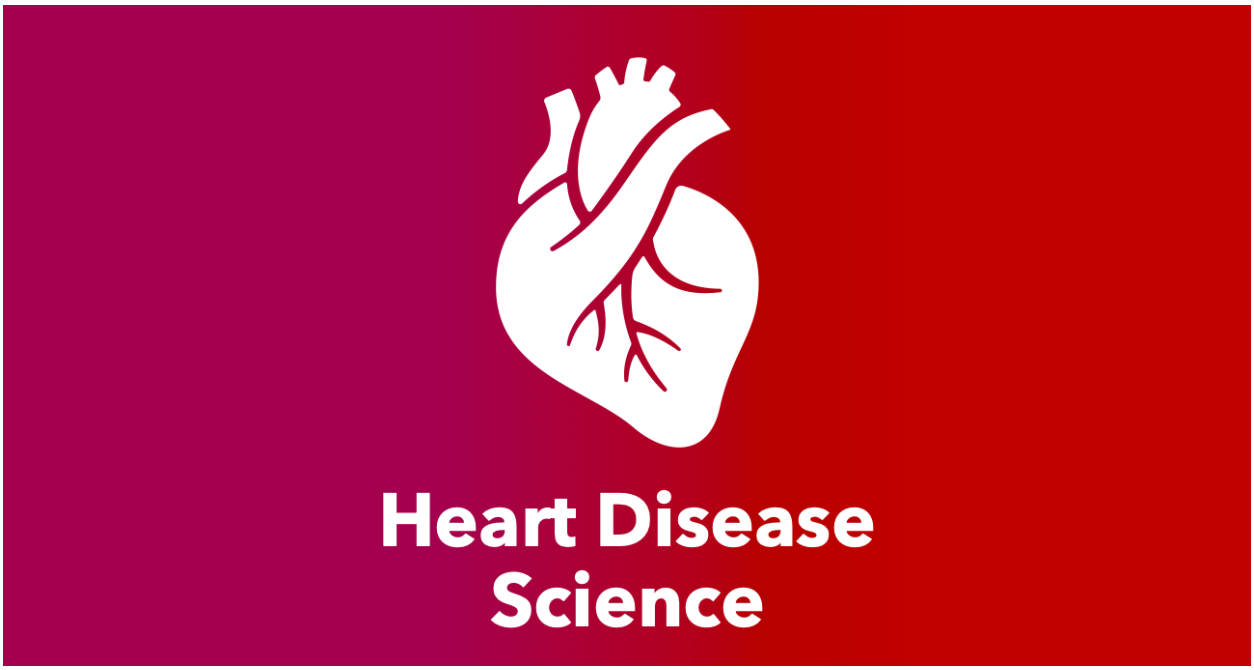
Heart Disease Science

Finn B

2/9/2021



Introduction

The purpose of this report is the analysis and methodology of several health data of patients from 1988. The data shows if a patient has heart disease. It describes a range of conditions that affect the heart. The data set used is a data set provided by **Donald Bren School of Information and Computer Sciences** from the **University of California, Irvine** originally. This project will concentrate on a database from the **V.A. Medical Center, Long Beach and Cleveland Clinic Foundation** provided created by **Robert Detrano, M.D., Ph.D.**. The data was sourced from Kaggle, where the data was initially processed.
Origin of this database: Archive.ics.uci
First step is exploration and cleaning of the dataset. After that, each attribute is examined for its relation to the target variable. In the section of modeling several classification methods are run through to predict whether a patient has heart disease or not. The methods used are **decision tree**, **random forest**, **support vector machines** and **k-nearest neighbors**.

Data exploration and cleaning

Data exploration

This report excludes 62 attributes from the original database to work with a subset of 14 attributes, containing **13 features** and **one outcome variable** to consider if a patient has heart disease. The database contains health data of **303 patients**.
On a first view you can see what features will accompany the final outcome variable in this project. Before heading into the analysis we need to understand what the different attributes tell us:

age	sex	cp	trestbps	chol	fbs	restecg	thalach	exang	oldpeak	slope	ca	thal	target
63	1	3	145	233	1	0	150	0	2.3	0	0	1	1
37	1	2	130	250	0	1	187	0	3.5	0	0	2	1
41	0	1	130	204	0	0	172	0	1.4	2	0	2	1
56	1	1	120	236	0	1	178	0	0.8	2	0	2	1
57	0	0	120	354	0	1	163	1	0.6	2	0	2	1
57	1	0	140	192	0	1	148	0	0.4	1	0	1	1

Data cleaning

For data cleaning some data from the original data base was changed. The levels of 'sex' were changed to 'female' and 'male'. The levels of 'target' were changed to 'disease' and 'no disease' to have a quiet better overview. Furthermore some of the attributes were encoded as factors to enable a better work: **sex, cp, fbs, restecg, exang, slope, ca, thal, disease(target)**.

```
HeartData <- HeartData %>%
  mutate(sex=ifelse(sex==0, 'female', 'male'))
HeartData$sex <- as.factor(HeartData$sex)
```

```
HeartData$cp <- as.factor(HeartData$cp)
HeartData$cp <- revalue(HeartData$cp, c('0'='asymptomatic',
                                        '1'='atypical angina',
                                        '2'='non anginal pain',
                                        '3'='typical angina'))
```

```
HeartData <- HeartData %>%
  mutate(fbs=ifelse(fbs==1,
                    '>120',
                    '<=120'))
HeartData$fbs <- as.factor(HeartData$fbs)
```

```
HeartData$restecg <- as.factor(HeartData$restecg)
HeartData$restecg <- revalue(HeartData$restecg, c('0'='left vetricular hypertrophy',
                                                  '1'='normal',
                                                  '2'='st-t abnormality'))
```

```
HeartData <- HeartData %>%
  mutate(exang=ifelse(exang==0,
                      'no',
                      'yes'))
HeartData$exang <- as.factor(HeartData$exang)
```

```
HeartData$slope <- as.factor(HeartData$slope)
HeartData$slope <- revalue(HeartData$slope, c('0'='downsloping',
                                              '1'='flat',
                                              '2'='upsloping'))
```

```
HeartData$ca <- as.factor(HeartData$ca)
HeartData$ca <- revalue(HeartData$ca, c('4'=NA))
```

```
HeartData$thal <- as.factor(HeartData$thal)
HeartData$thal <- revalue(HeartData$thal, c('0'=NA,
                                            '1'='fixed defect',
                                            '2'='normal',
                                            '3'='reversible defect'))
```

```
HeartData <- HeartData %>%
  mutate(target=ifelse(target==0,
                       "disease",
                       "no disease"))
HeartData$target <- as.factor(HeartData$target)
HeartData$disease <- HeartData$target
HeartData$target <- NULL
attr(HeartData, 'spec') <- NULL
```

Attribute	Meaning
age	Patients age (29-77 years)
sex	Female (0) and Male (1)
cp - **chest pain type**	asymptomatic (0); atypical angina (1); non-anginal pain (2); typical angina (3)
trestbps - **resting blood pressure**	in mm/Hg on admission to the hospital[1]
chol - **serum cholesterol**	in mg/dl
fbs - **fasting blood sugar**	$>$ 120 mg/dl; no(0) yes(1)
restecg - **resting electrocardiographic results**	probable or definite left ventricular hypertrophy by Estes' criteria(0); normal(1); having ST-T wave abnormality(2)
thalach	maximum heart rate achieved
exang - **exercise induced angina**	no(0); yes(1)
oldpeak	ST depression induced by exercise relative to rest
slope - **slope of peak exercise ST segment**	downsloping(0); flat(1); upsloping(2)
ca - **number of major vessels colored by flouroscopy**	vessels(0-3); NA(4)
thal - **Thalium Stress Test Result**	NA(0); fixed defect(1); normal(2); reversible defect(3)
disease - **angiographic disease status**	$>$ 50% diameter narrowing (0); $<$ 50% diameter narrowing (1)

[1] Judging from the values, the systolic pressure (the pressure when the heart pushes blood out) is given here.

age	sex	cp	trestbps	chol	fbs	restecg	thalach	exang	oldpeak	slope	ca	thal	disease
63	male	typical angina	145	233	>120	left vetricular hypertrophy	150	no	2.3	downsloping	0	fixed defect	no disease
37	male	non anginal pain	130	250	<=120	normal	187	no	3.5	downsloping	0	normal	no disease
41	female	atypical angina	130	204	<=120	left vetricular hypertrophy	172	no	1.4	upsloping	0	normal	no disease
56	male	atypical angina	120	236	<=120	normal	178	no	0.8	upsloping	0	normal	no disease
57	female	asymptomatic	120	354	<=120	normal	163	yes	0.6	upsloping	0	normal	no disease
57	male	asymptomatic	140	192	<=120	normal	148	no	0.4	flat	0	fixed defect	no disease

Data analysis

In this part of the project we will dig deeper into the attributes and potential effects on the disease. But first we will have a look on the categorization of disease and on the most obvious and superficial indicators: Age and Sex.

Disease

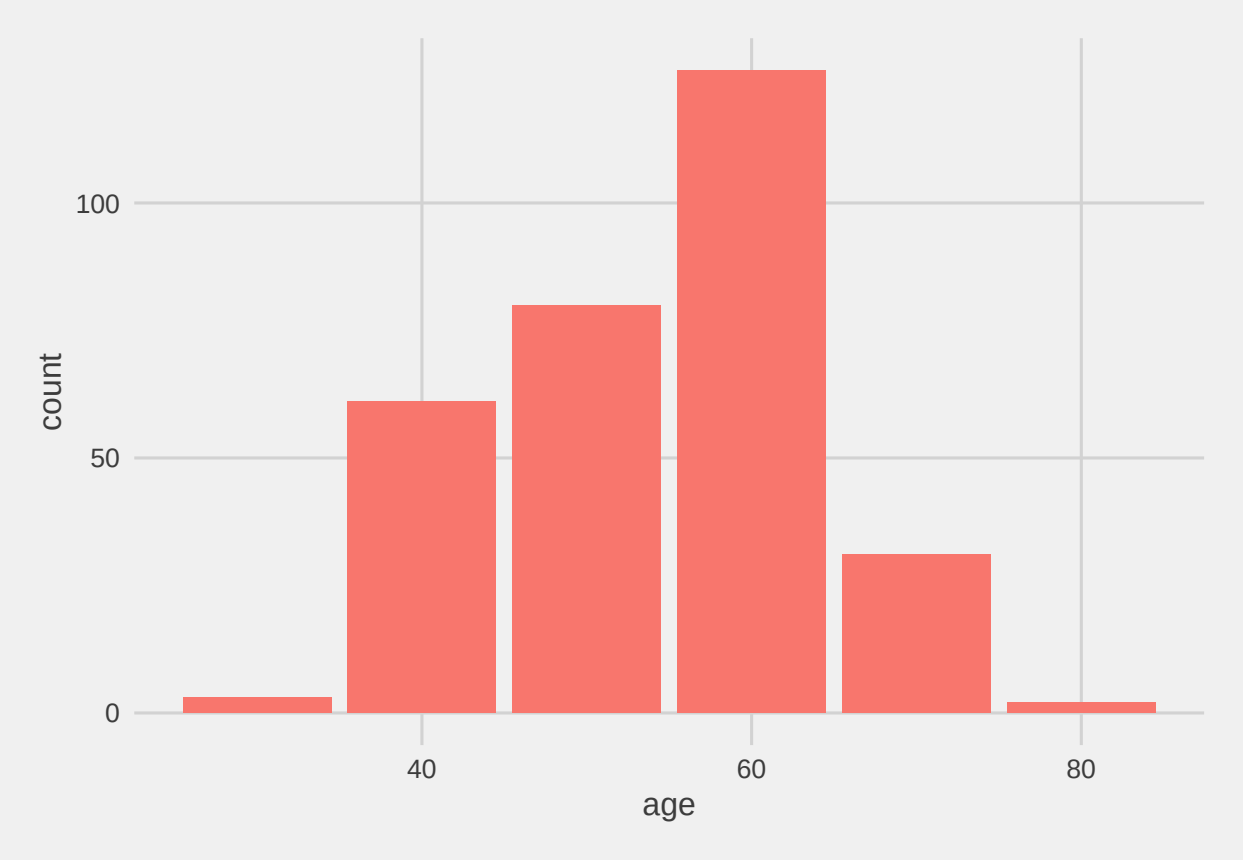
To determine the diagnosis of heart disease status the angiographic disease status was used. This was differentiated into two conditions that differ in the percentage diameter narrowing of <50% and >50% of coronary arteries.

disease	cases
disease	138
no disease	165

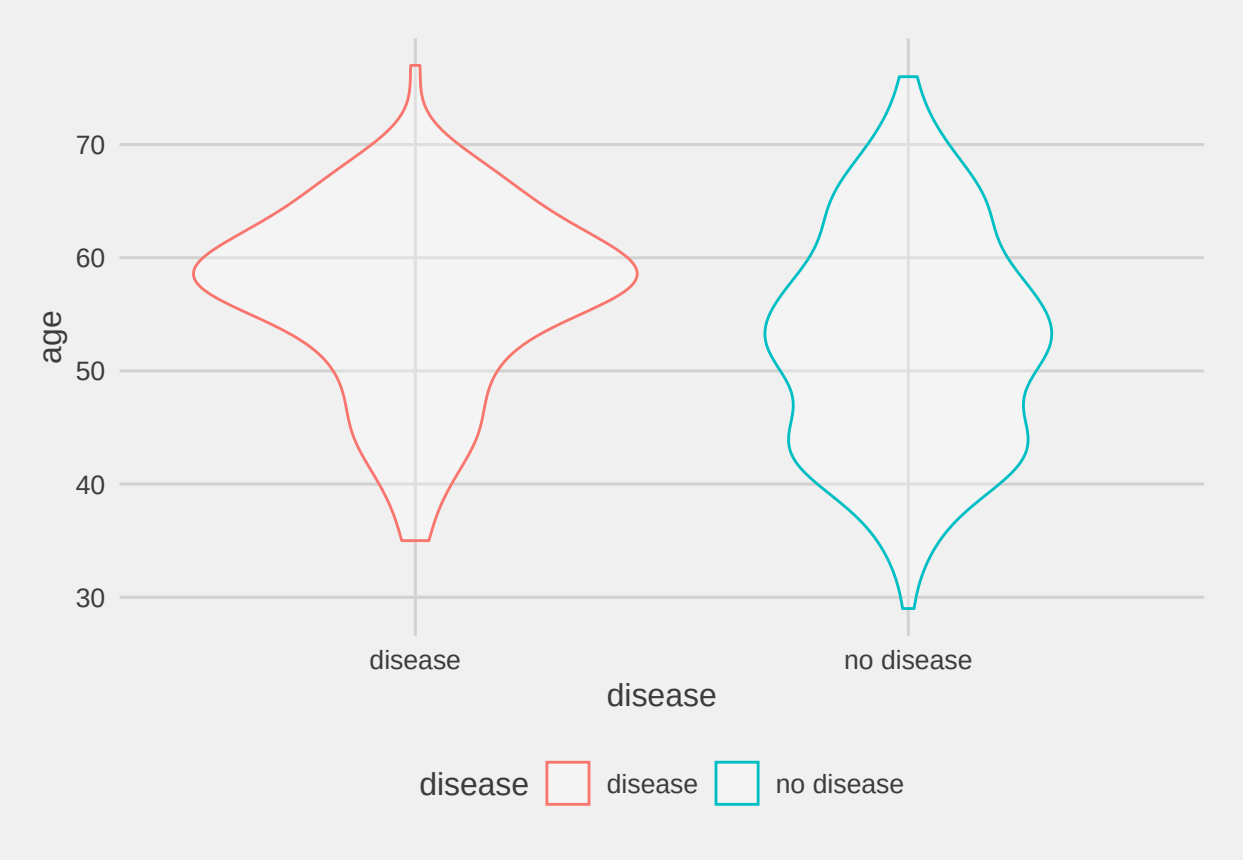
As we can see the proportion of patients with disease was at 45.5% in the database.

Age

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##    29.0    47.5    55.0    54.4    61.0    77.0
```



The age range goes from 29 years to 77 year. The median age is at 55 years, while we can see that the most patients are between 55 and 65 years.



Sex

sex	count
female	96
male	207

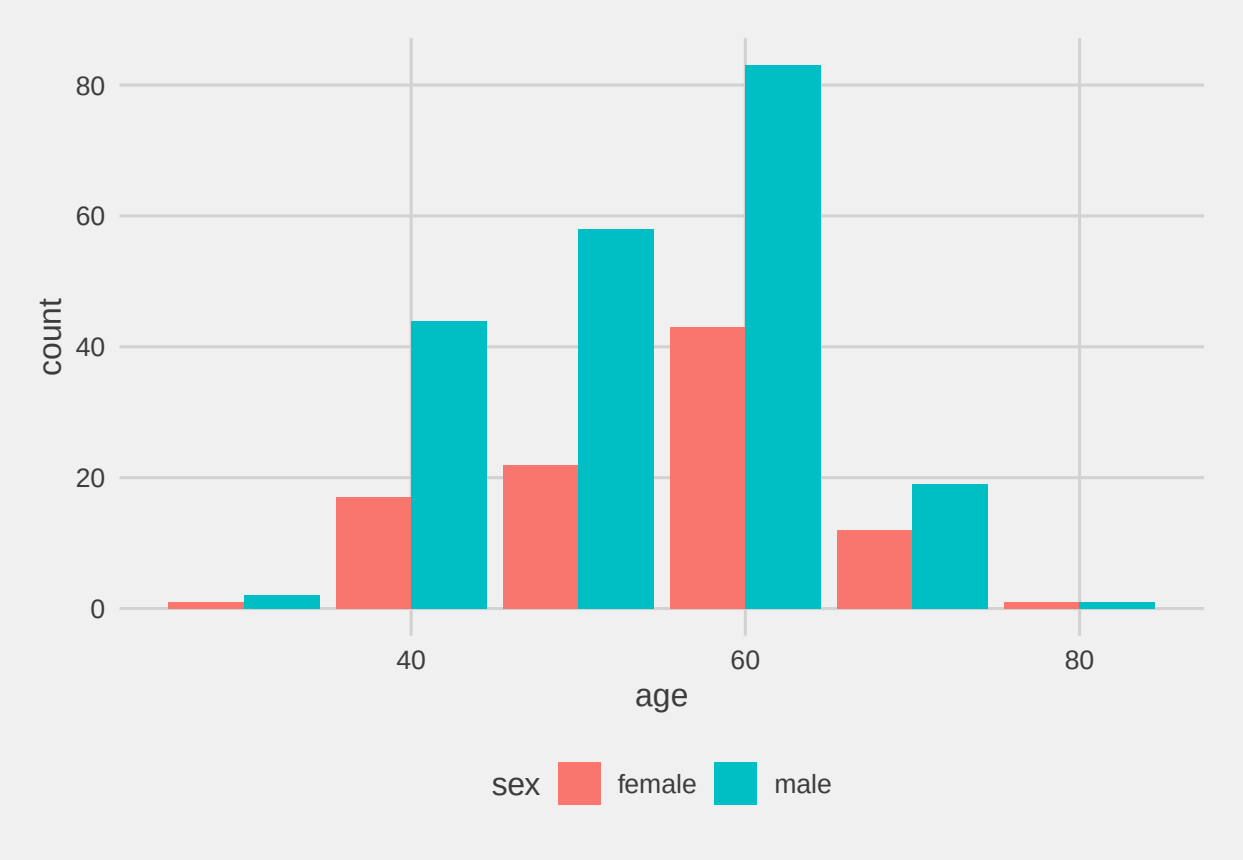
The distribution by sex is dominated by male patients, around 68% of the patients are male. The mean age of female patients is quite lower than the mean age of male patients.

This can be seen in the mean of patients that have heart diseases as well:
female:

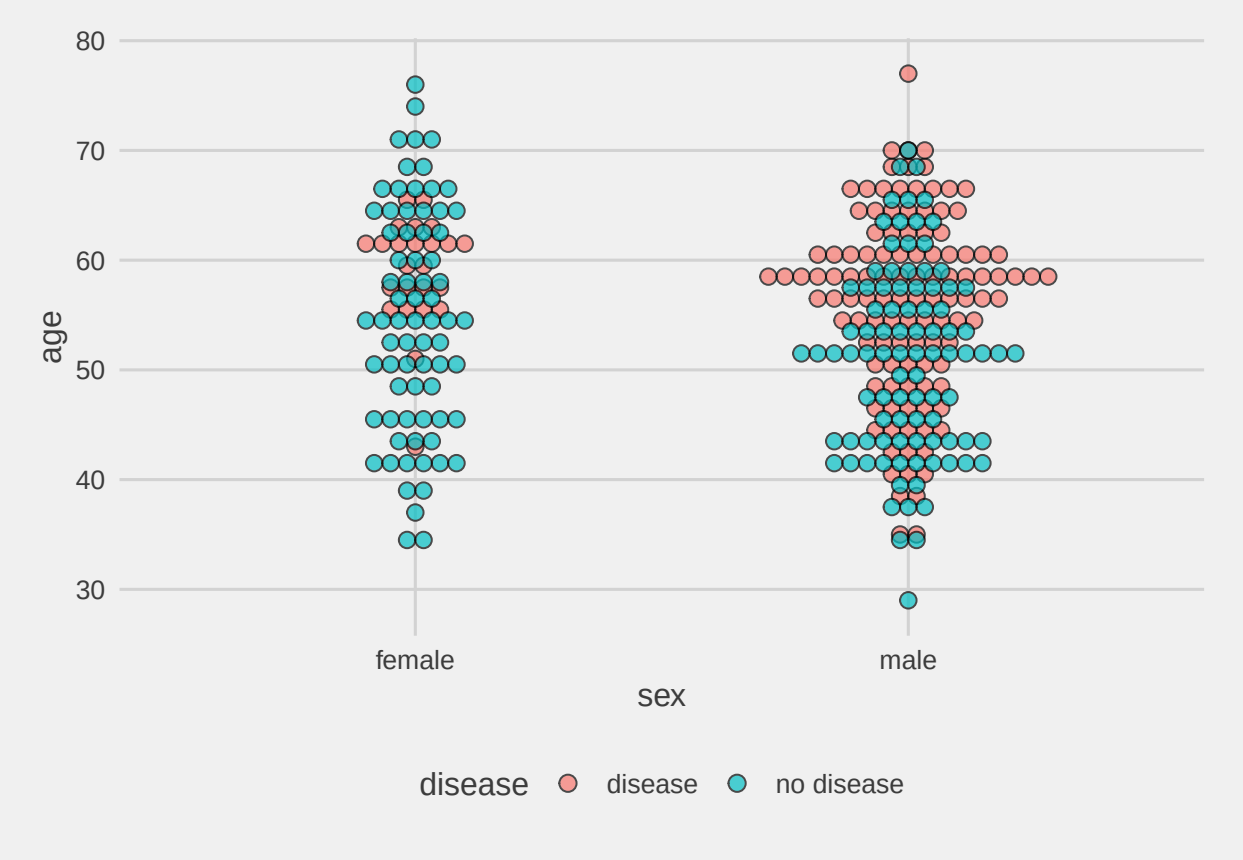
```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##    43.0    56.8    60.5    59.0    62.0    66.0
```

male:

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##    35.0    51.0    57.5    56.1    61.0    77.0
```

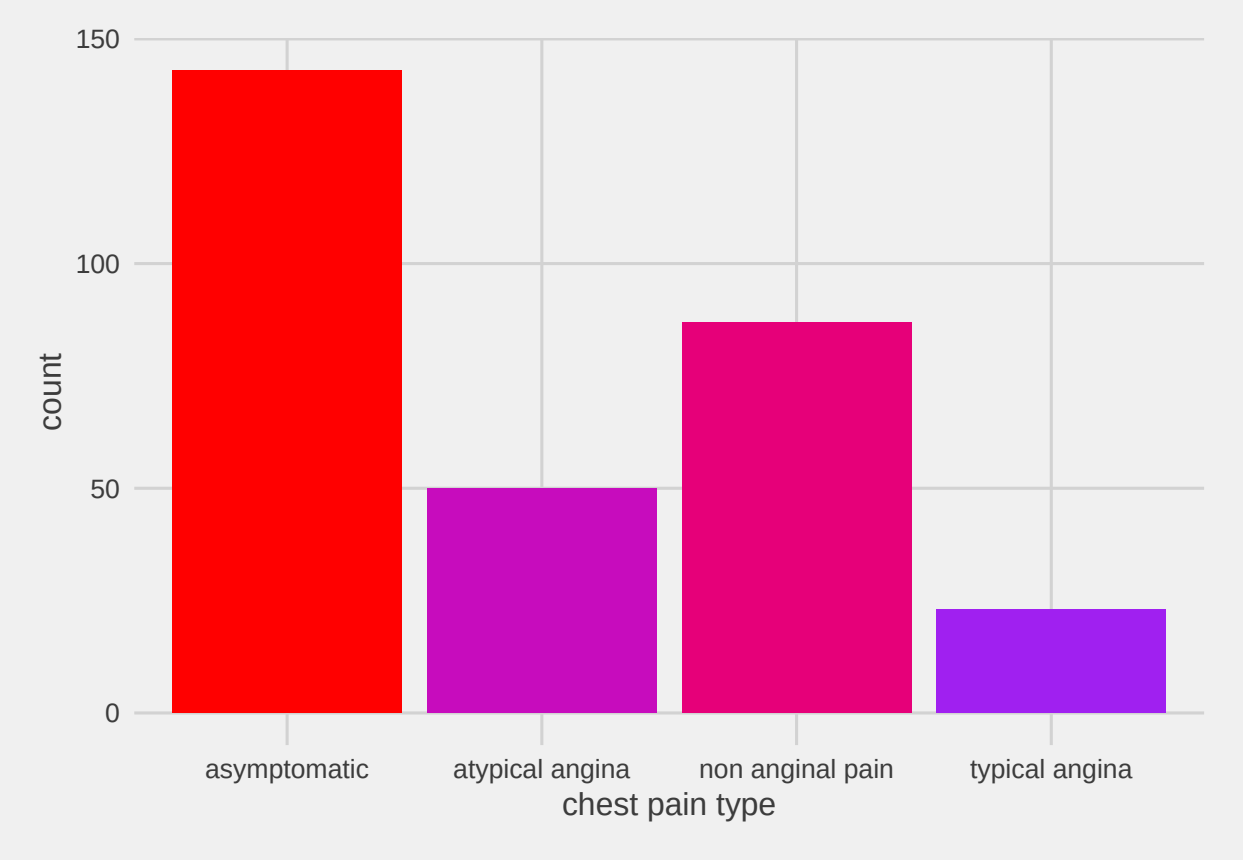


As we can see in the following plot there are much more female patients without heart disease than with heart disease. Furthermore the number of male patients with and without diseases seem to be similar while there are more male patients around 60 with heart disease.



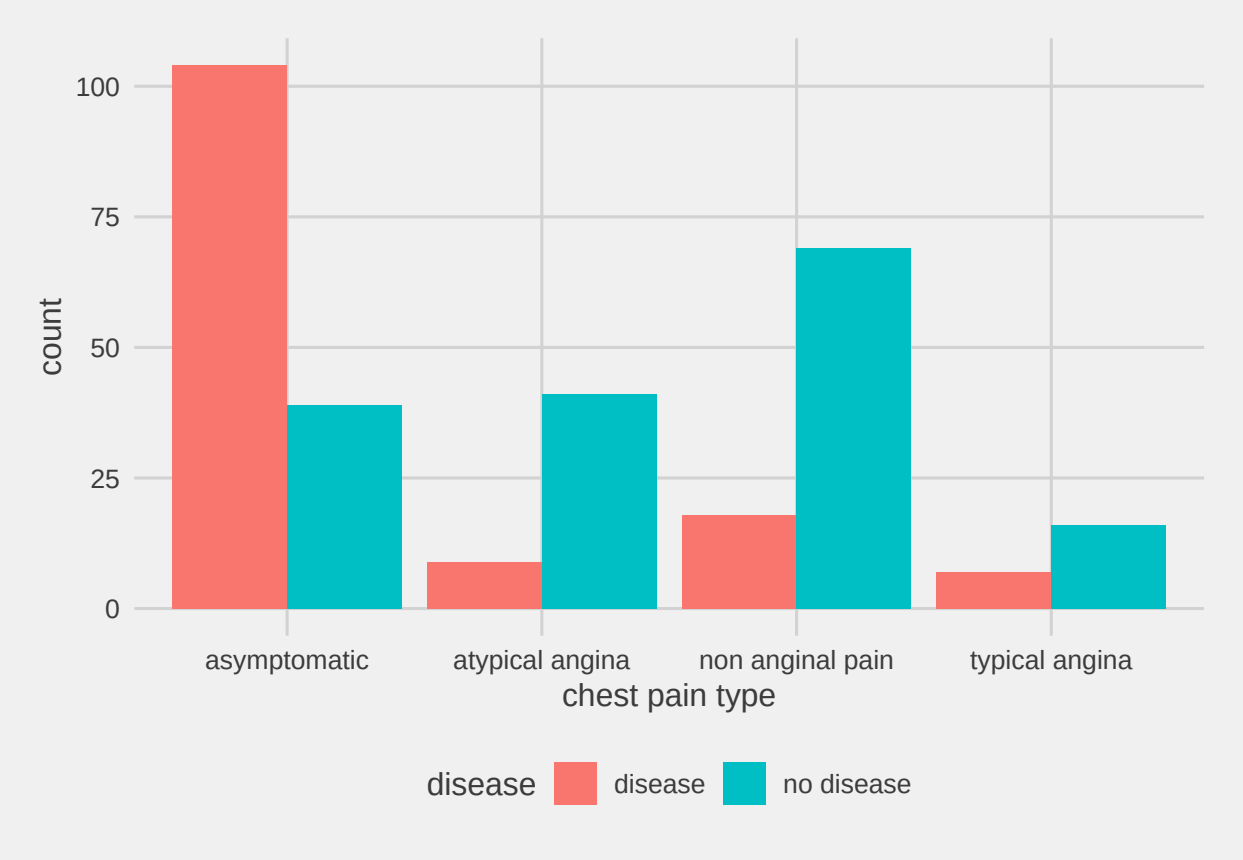
Chest pain type

As previously researched, there are four types of chest pain. **Asymptomatic** pain means that a patient has no symptoms/pain. Angina is a pain that the patient has near the heart, often described as chest tightness. Angina pain is categorized into **atypical** angina and **typical** angina.
Most patients data shows asymptomatic and non anginal pain. Only a small amount of patients had typical angina:



Something that may not have been expected by many is the following result. Except of asymptomatic pain the proportion of patients with disease is far below 50% for each category of pain.

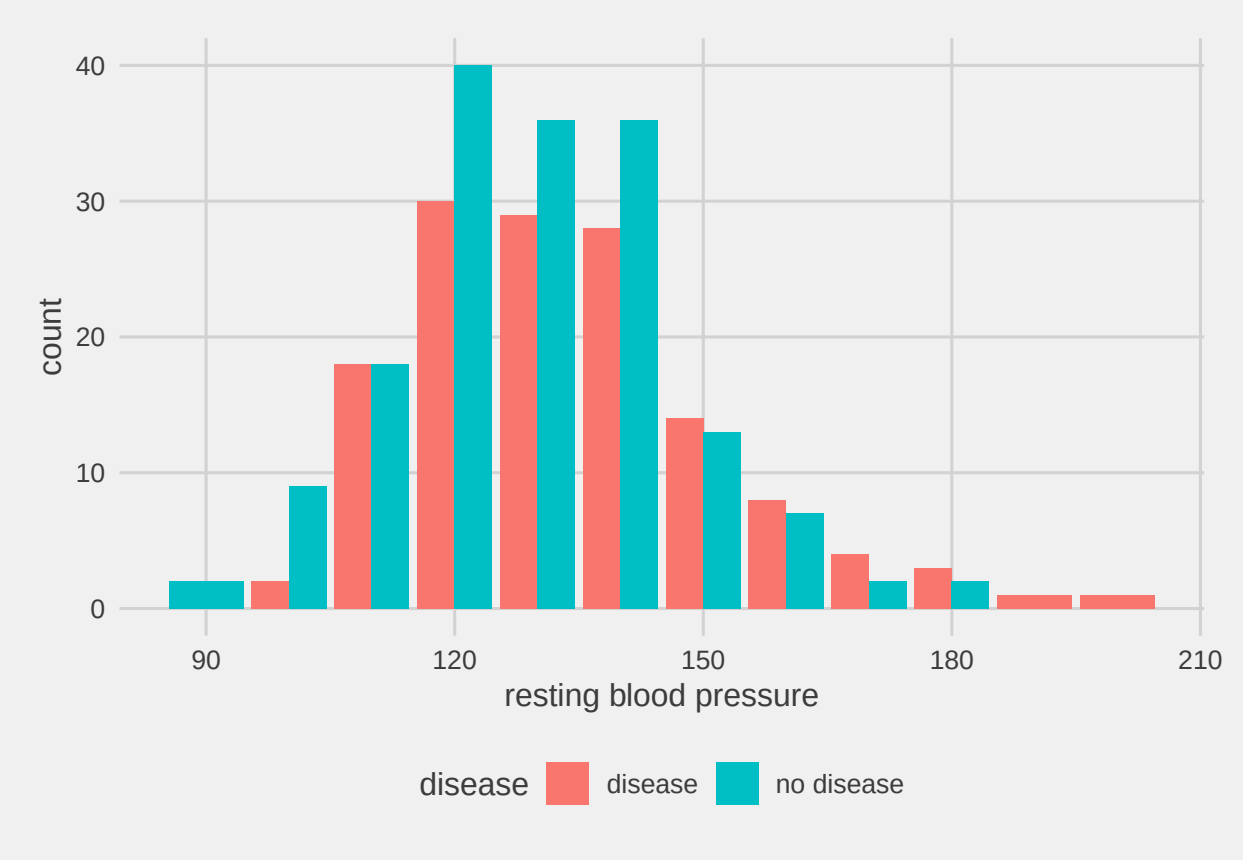
chest pain	disease (prop)
asymptomatic	0.727
atypical angina	0.180
non anginal pain	0.207
typical angina	0.304

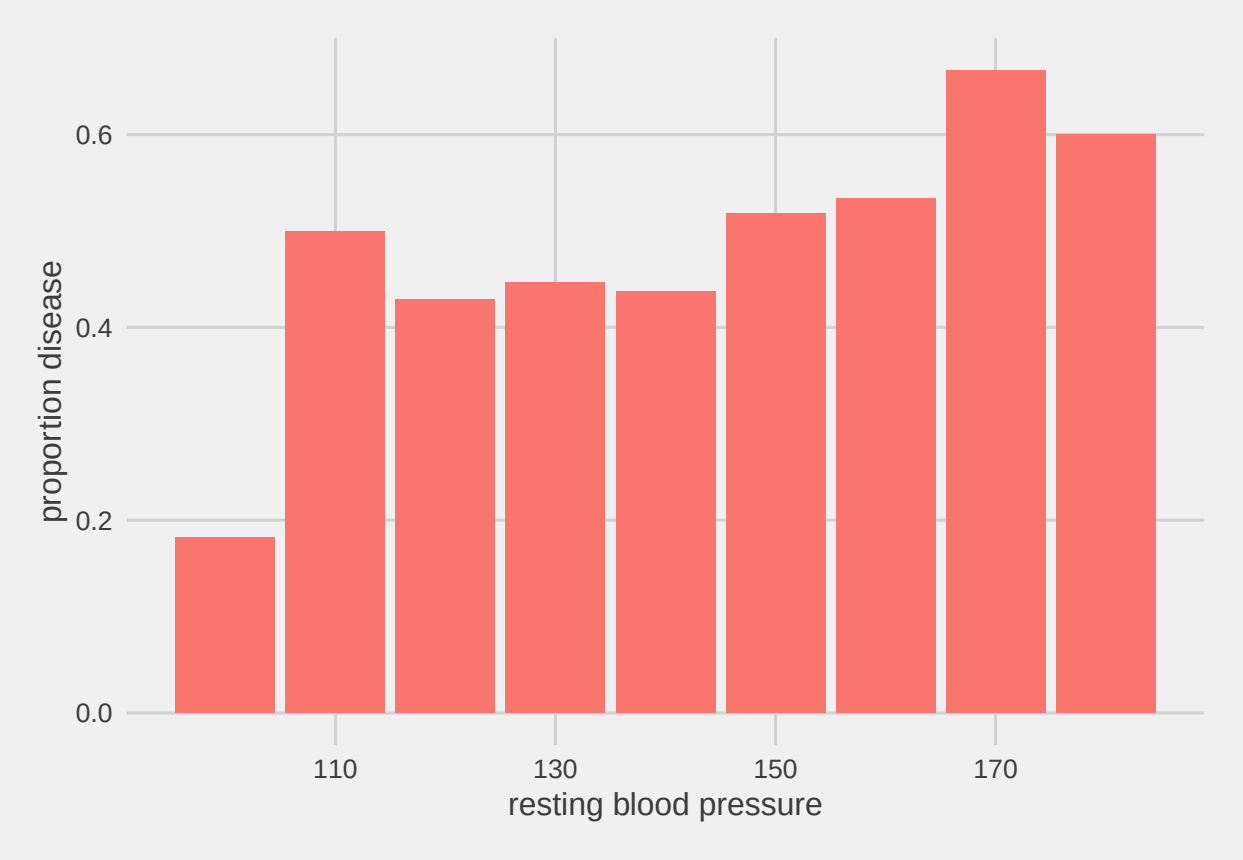


Resting blood pressure

For most patients (68%) the blood pressure is higher than the ideal systolic blood pressure of between 90 and 120mm/Hg. In the second plot we can observe an increasing proportion of disease with a higher systolic resting blood pressure.

resting blood pressure >120mm/Hg
0.68

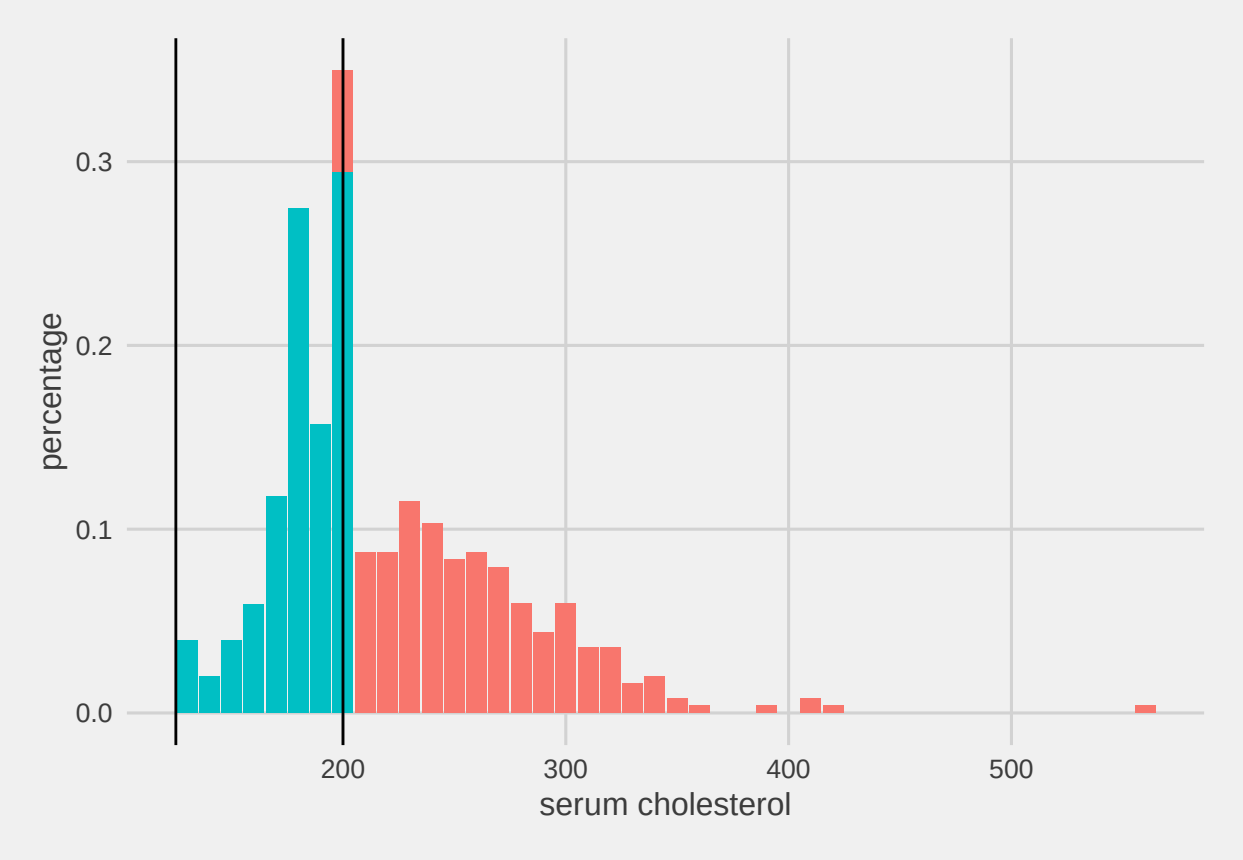

40
30
20
10
0
count
90
120
150
180
210
resting blood pressure
disease
disease
no disease



Serum cholesterol

Since there is too little information about what type of cholesterol level is given we assume total cholesterol. Healthy cholesterol level for adults is between 125mg/dL and 200mg/dL.

Most patients serum cholesterol is higher than the healthy range:

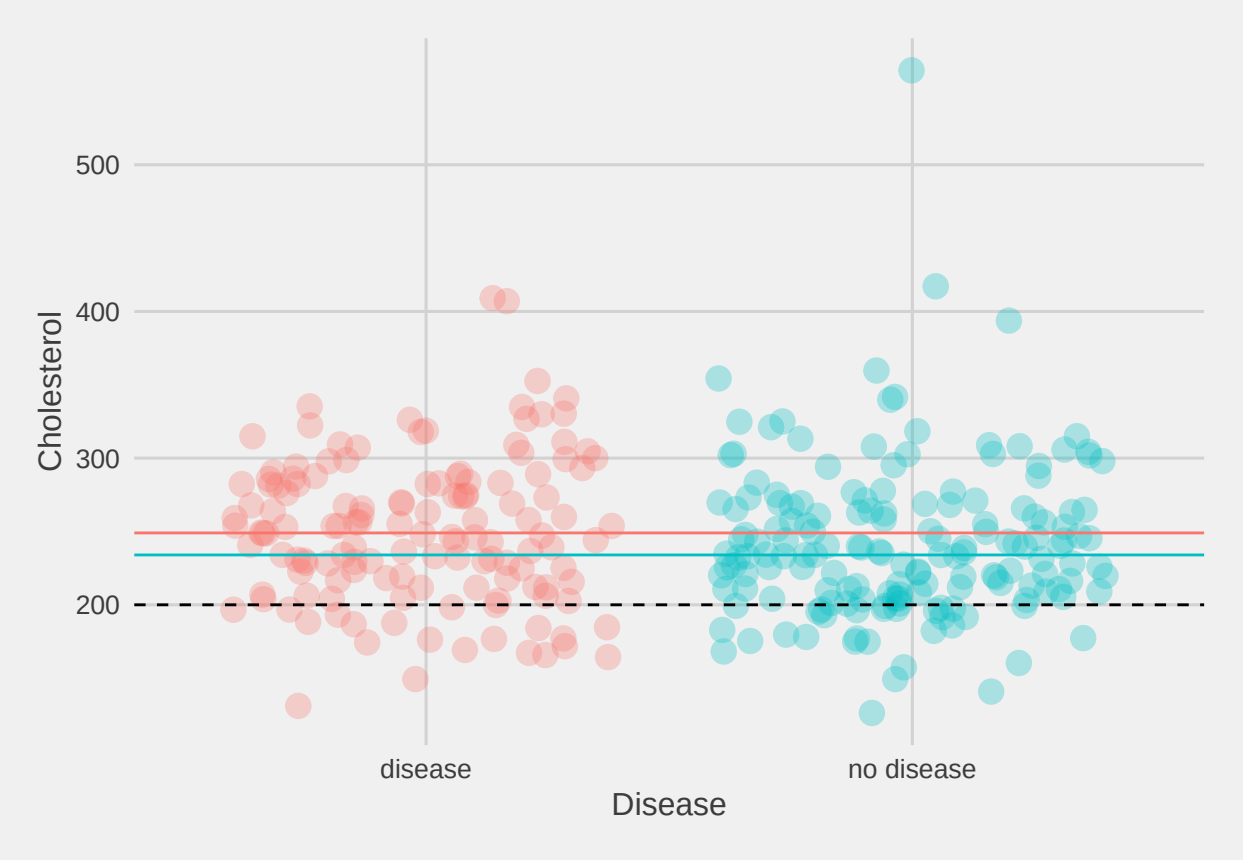


This can also be seen in comparison between diseased and healthy patients. The median of cholesterol of both groups is higher than 200.

```
HD.chol.median.mean <- HeartData %>%
  group_by(disease) %>%
  summarize(me=mean(chol), med=median(chol))
```

```
## `summarise()` ungrouping output (override with `.groups` argument)
```

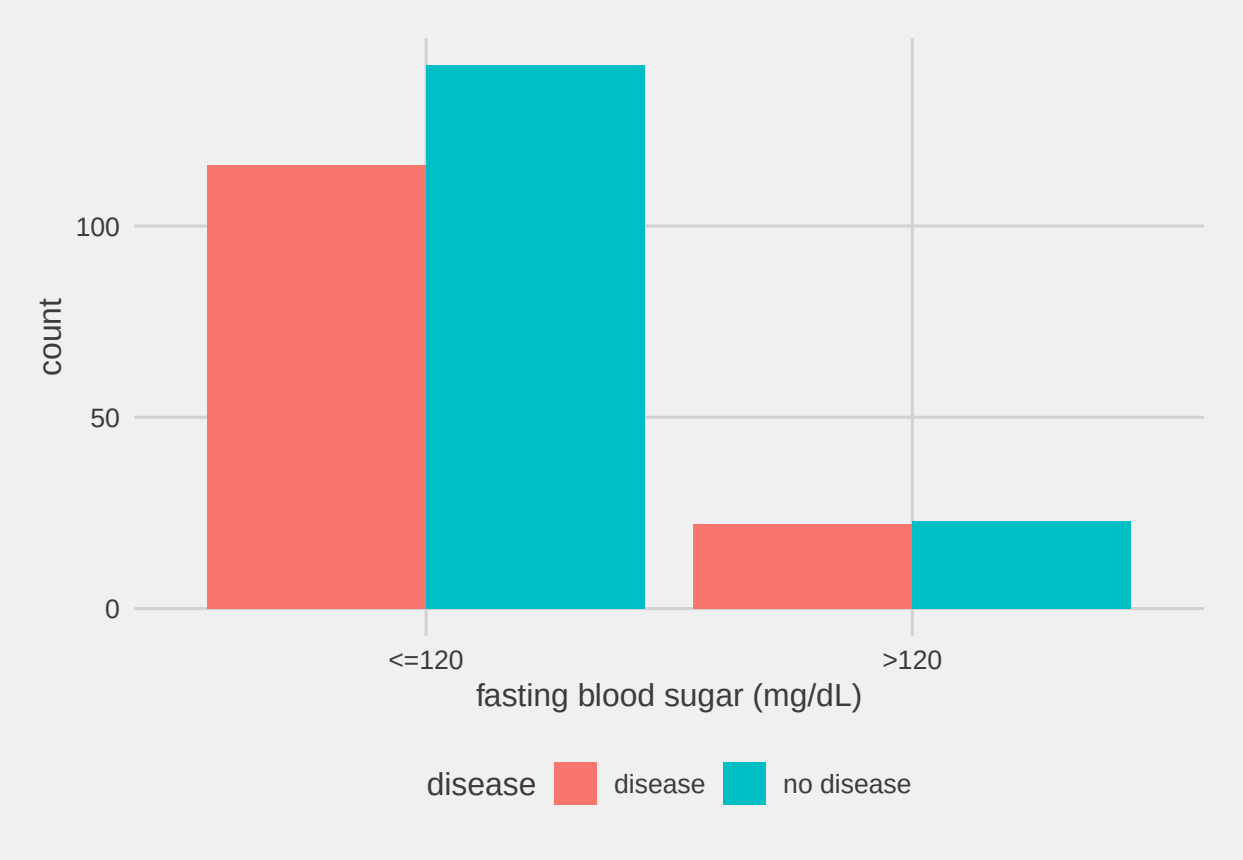
```
HD.chol <- HeartData %>%
  group_by(disease) %>%
  select(disease, chol)

  ggplot(data=HD.chol, aes(disease, chol, color=disease)) +
  geom_jitter(width = 0.4, alpha = 0.3, size=4) +
  stat_smooth(method="lm", formula=disease~1, se=FALSE) +
  geom_hline(data=HD.chol.median.mean, aes(yintercept = med, color=disease)) +
  geom_hline(yintercept=200, linetype = "dashed") +
  xlab("Disease") +
  ylab("Cholesterol") +
  theme_fivethirtyeight() +
  theme(axis.title = element_text(), legend.position = "none")
```



Fasting blood sugar

Normal blood sugar levels of **non-diabetic** people are between **72mg/dL** and **99mg/dL** when fasting. Fasting blood sugar levels of **100mg/dL** up to **125mg/dL** are already described as **prediabetic**, while **fbs > 125mg/dL** are diagnosed as **diabetic**.
The study shows two possible outcomes of **fbs**: **<= 120mg/dL** and **>120mg/dL**.It must be considered, that people with a **fbs >120mg/dL** are at greater risk of developing heart disease or cardiovascular disease, however the symptoms of the patient may be caused by diabetes and secondary diseases.

Only a few patients have blood sugar levels in the range where diabetes would be diagnosed. The number of patients with and without disease are similar. The most patients have fasting blood sugar levels of 120 and lower.



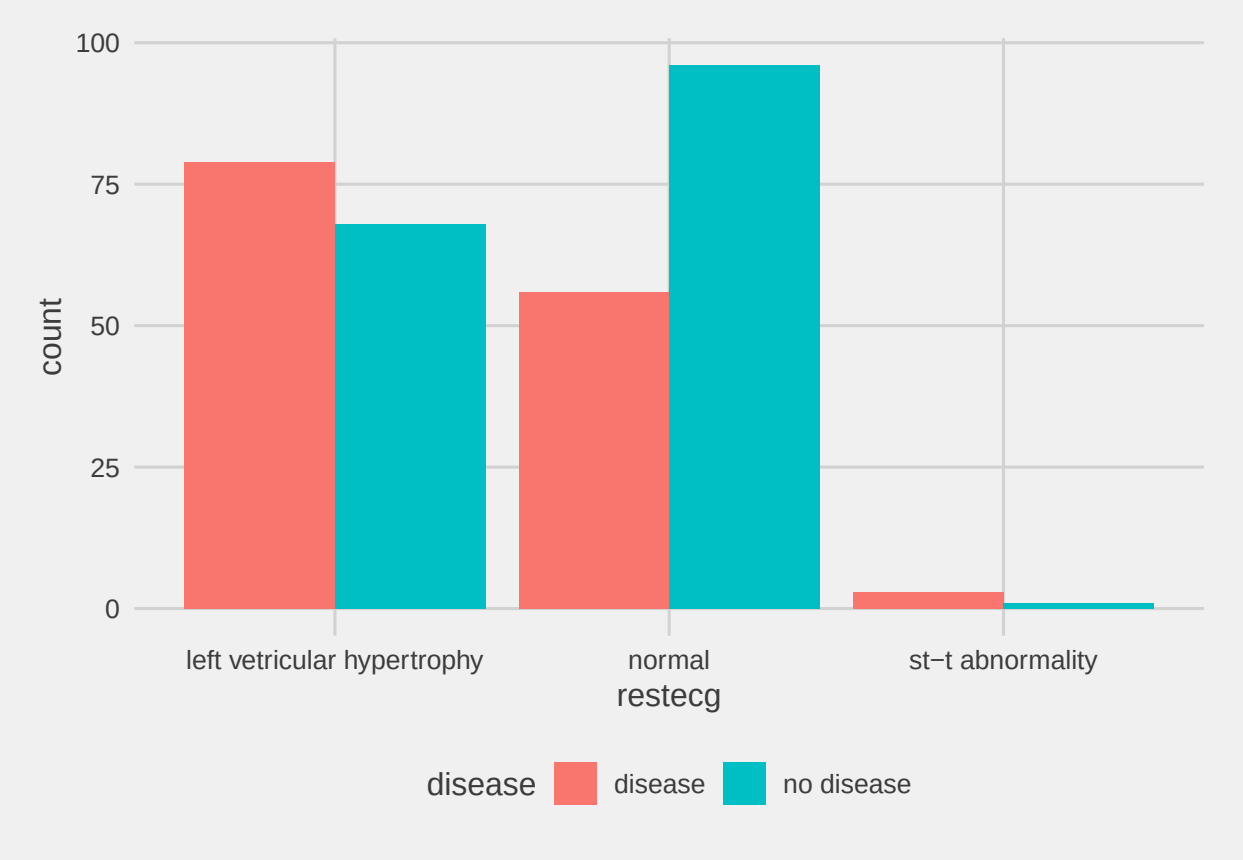
There were 14.90% of patients with a critical value of fasting blood sugar level in the database, while the prevalence of diabetes in the US was 4.90%[2] in the year 1990. So we can observe a much higher prevalence in the study from 1988.

[2]Diabetes trends in the U.S.: 1990-1998

Resting electrocardiographic results

As results of the restecg there are three potential outcomes:

- **left ventricular hypertrophy:**
  Left ventricular hypertrophy is enlargement and thickening (hypertrophy) of the walls of the heart's main pumping chamber.
- **Normal:**
  No abnormalities or hypertrophies.
- **Having ST-T wave abnormality:**
  Abnormalities of **ST-** and/or **T wave** in the imaging procedures of the electrocardiogram.

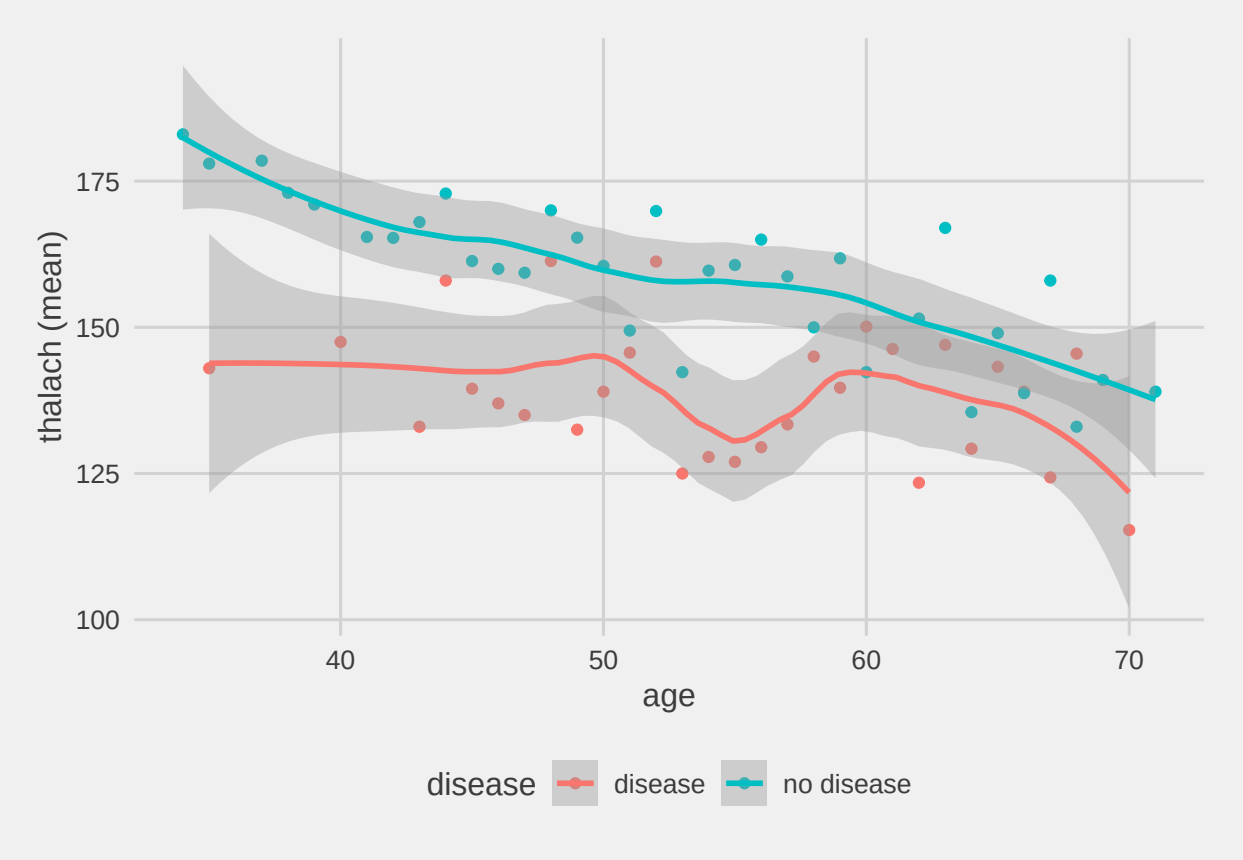


THALACH {thalach}

THALACH is the **maximum heart rate** that has been achieved of each patient.
We observe a lower maximum heart rate for patients with disease than without disease:

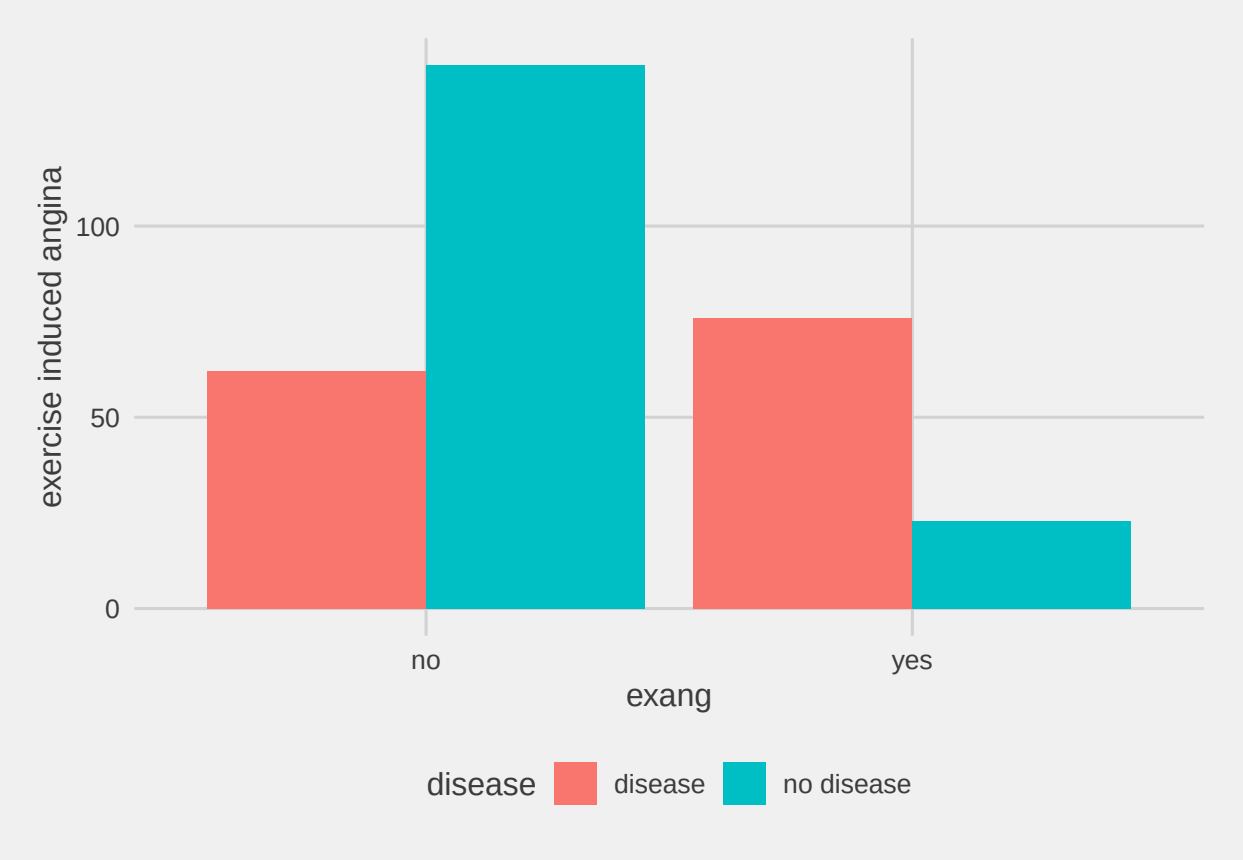
disease	mean	median
disease	139	142
no disease	158	161

As you can see in the next chart the average maximum heart rate decreases with age. An interesting abnormality is that patients with heart disease show a lower maximum heart rate at almost any age.



Exercise induced angina

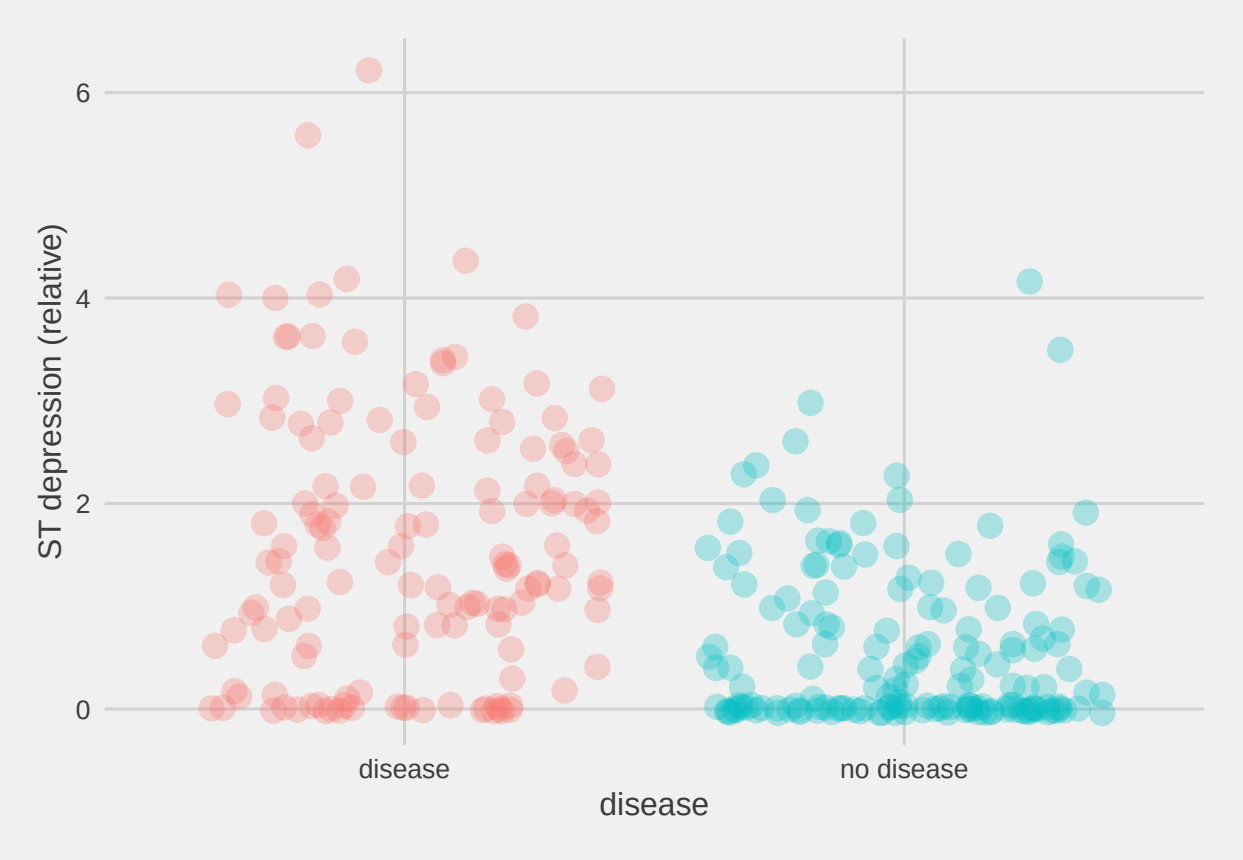
We can assume that angina indicates heart disease at exercise more often than without. We can tell by the fact that most patients with exercise induced angina had heart disease but only a minority of patients with heart disease had angina outside the exercises, most were asymptomatic[3].



[3]3.3 chest pain type and disease

ST depression induced by exercise relative to rest

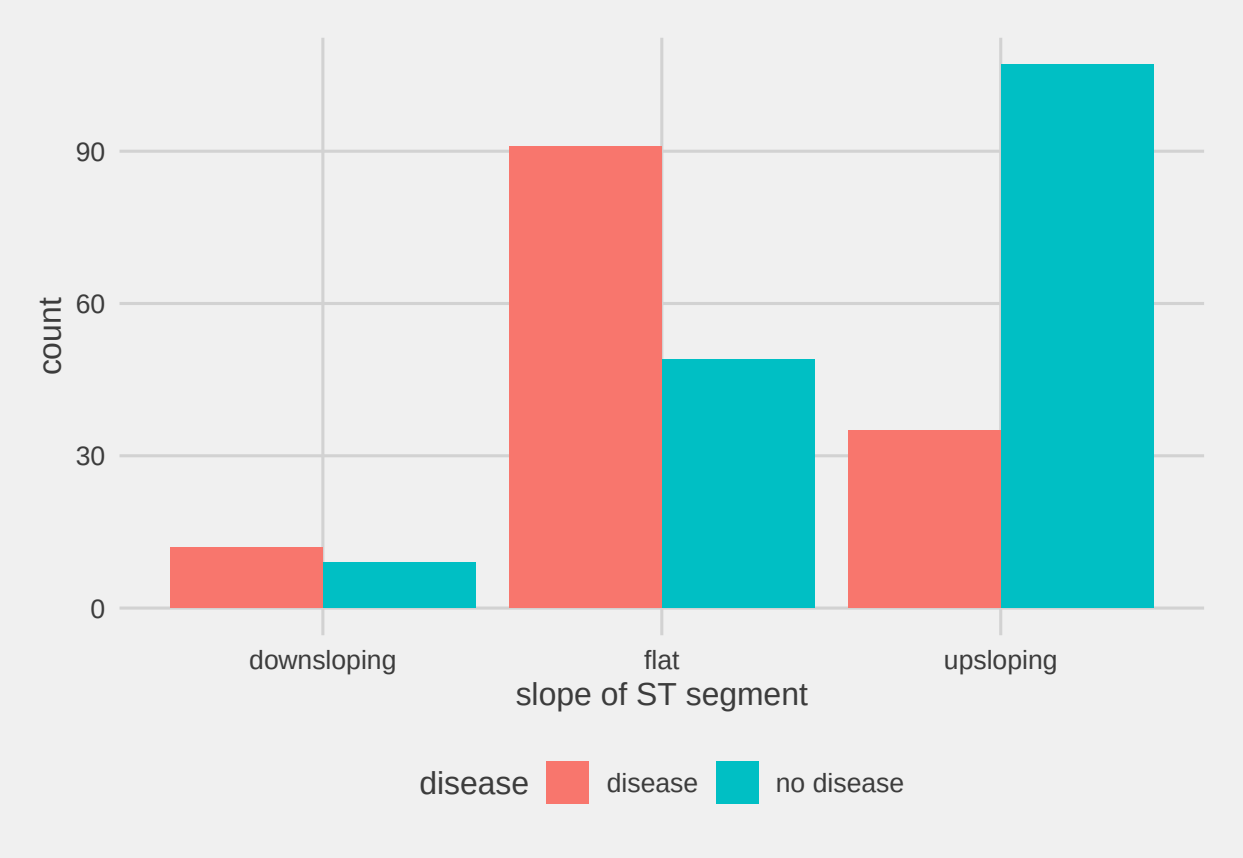
We can say that a greater ST depression is a sign of an increased probability of heart disease. The following findings from the database show, that the ST depression increase at exercise for patients with heart disease is greather than for patients without heart disease:



disease	mean	median
disease	1.586	1.4
no disease	0.583	0.2

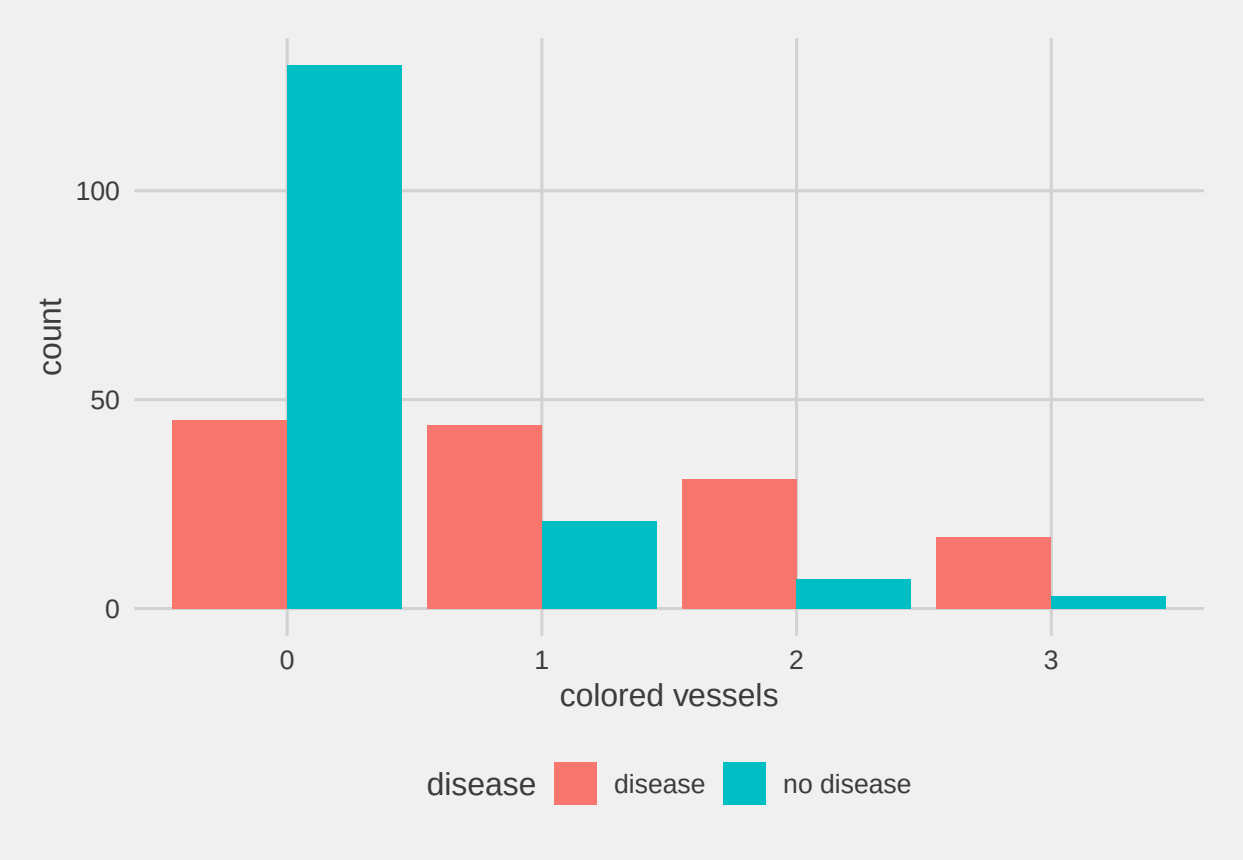
Mean and median show increased values of ST depression relation in people with heart disease than in people without heart disease.

Slope of peak exercise ST segment

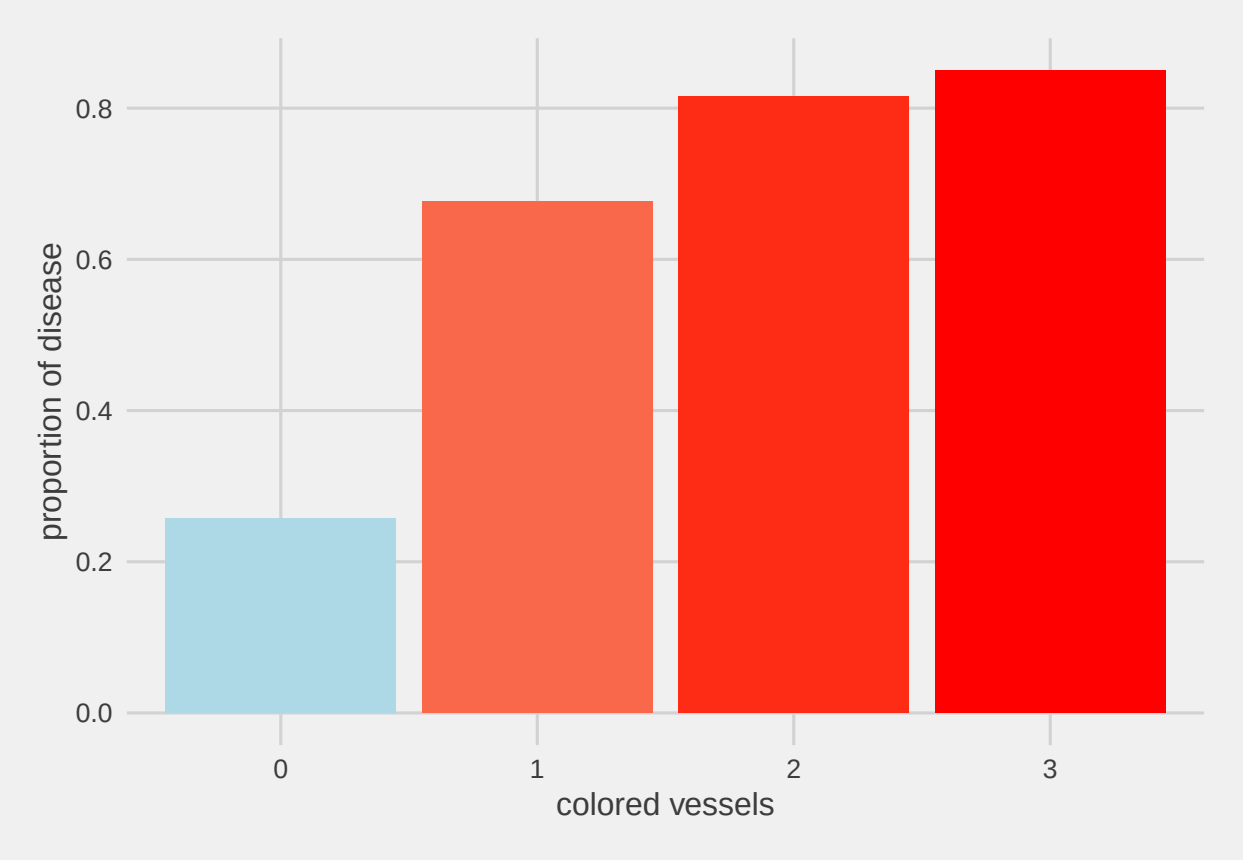


Major vessels colored by flouroscopy

Flouroscopy is an imaging tool which is made for looking on several body systems. In this case the flouroscopy was used to observe the flow of blood through three major vessels in order to evaluate the presence of arterial blockages.

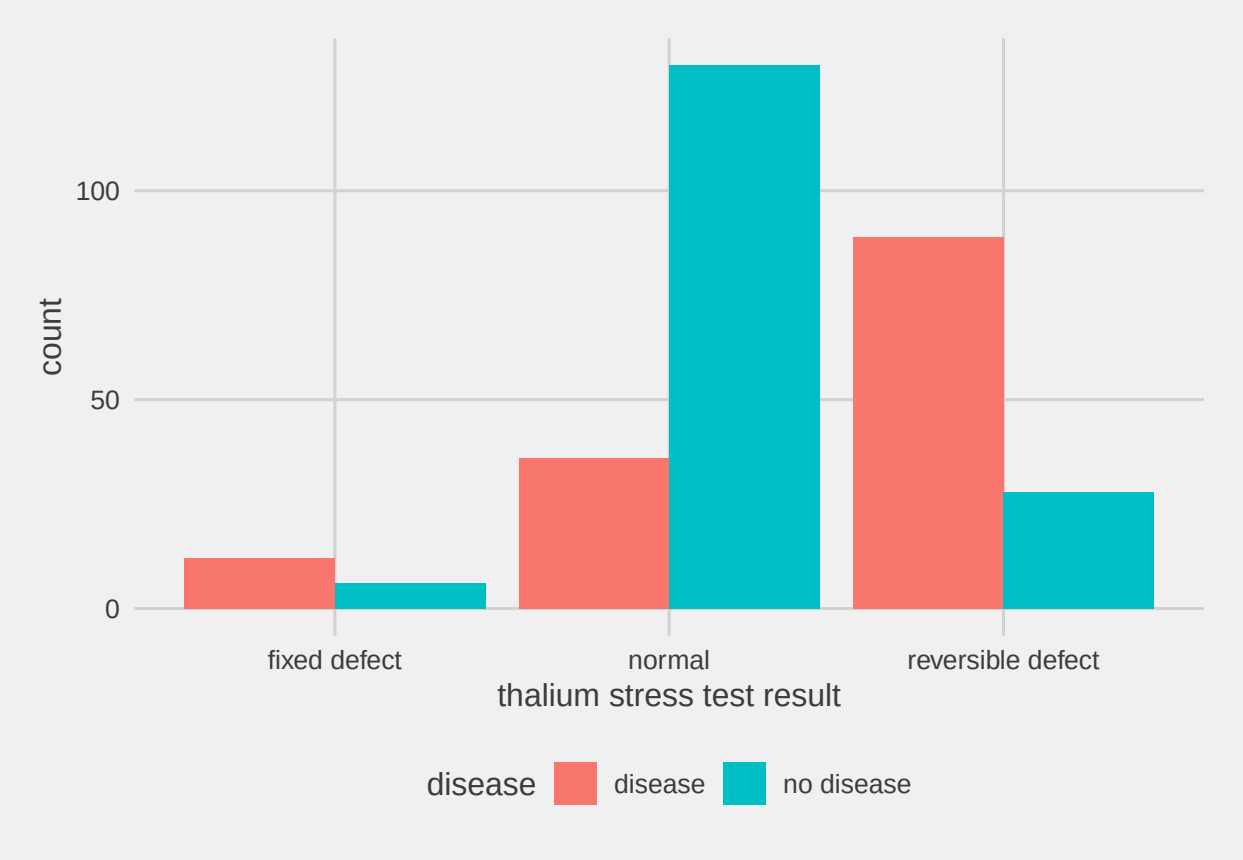


The next plot shows that the more vessels are colored at flouroscopy the higher the proportion of patients with disease.

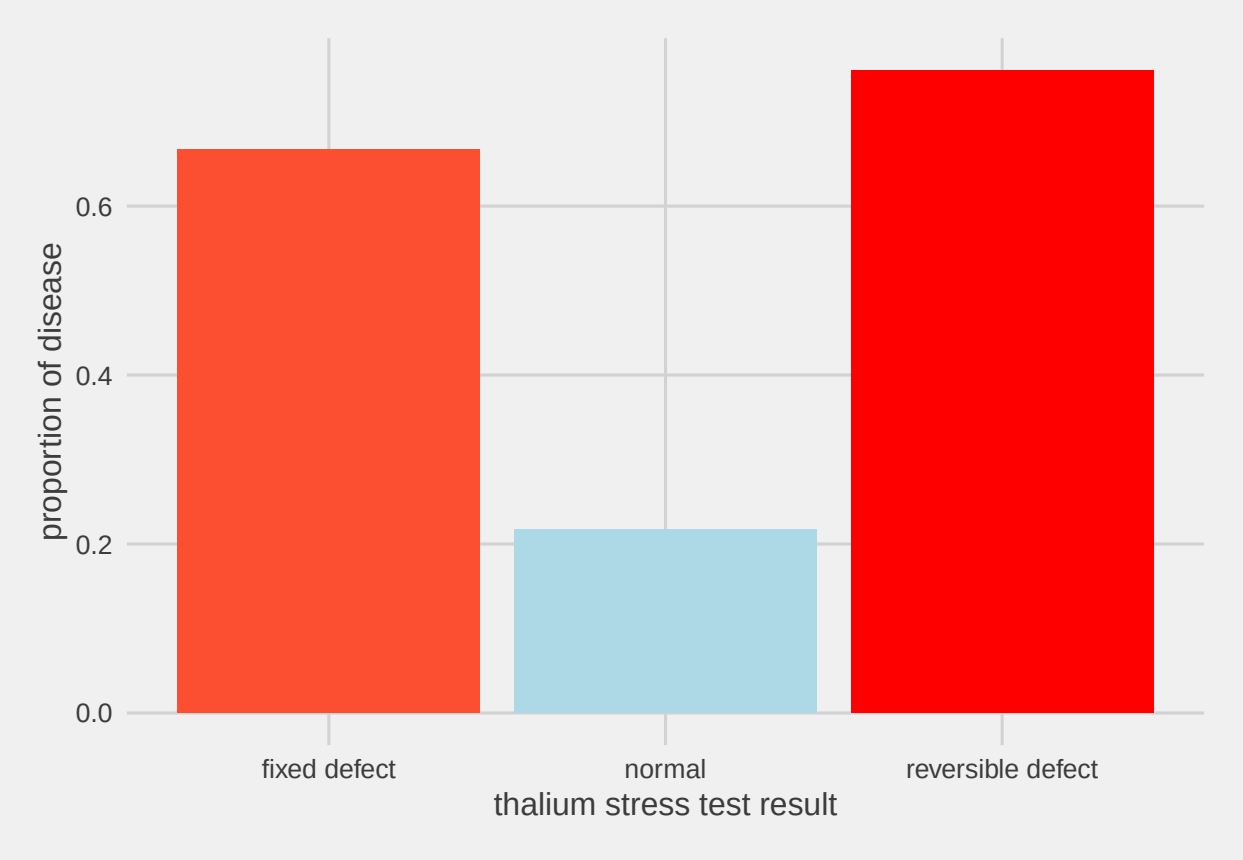


Thalium Stress Test Result

Thalium Stress Test is a test used to measure how the bloodflow works while exercising or resting. The result was divided into normal result, fixed defect and reversible defect.



As we can see, the proportion of disease at normal thalium stress test results is near 20%. While for any type of defect it is higher than 60%.



Methods

Training and testing set

In order to run different machine learning methods on the data we will first check the dataset on NA values.

```
sum(is.na(HeartData))
```

```
## [1] 7
```

```
compl <- as.vector(complete.cases(HeartData))
HeartDataRM <- HeartData[compl, ]
```

To get reproducible results a seed has been set with as.integer(Sys.time()) using the last five characters.

```
set.seed(50866, sample.kind = "default")
```

The data will be partitioned into two sets of 70% of the data for training and 30% of the data for testing.

```
test_index <- createDataPartition(y = HeartDataRM$disease,
                                  times = 1,
                                  p = 0.30,
```

```
                                    list = FALSE)
training <- HeartDataRM[-test_index,]
testing <- HeartDataRM[test_index,]
```

Running functions of the caret package for the next model prediction attempts already brings cross validation as default but we will use repeated cross validation with 3 repeats.

```
control.repeat <- trainControl(method = "repeatedcv",
                               number = 10,
                               repeats = 3)
```

To run the k-nearest neighbors algorithm, the categorical data of the training and testing data will be encoded by using dummy variables. We will set this data to binary variables and get a data frames of 29 columns each. This data will be used for the kNN-Algorithm exclusively.

```
#one hot encoding (Training data)
Training.dummy <- dummyVars(" ~.", data=training)
training.onehot <- data.frame(predict(Training.dummy, newdata = training))
training.onehot$disease.disease <- as.factor(training.onehot$disease.disease)
training.onehot$disease.no.disease <- as.factor(training.onehot$disease.no.disease)
training.onehot$disease.disease <- NULL
#one hot encoding (Testing data)
Testing.dummy <- dummyVars(" ~.", data=testing)
testing.onehot <- data.frame(predict(Testing.dummy, newdata = testing))
testing.onehot$disease.disease <- as.factor(testing.onehot$disease.disease)
testing.onehot$disease.no.disease <- as.factor(testing.onehot$disease.no.disease)
testing.onehot$disease.disease <- NULL
```

Logistic regression

Because the heart disease dataset is a categorical problem we choose the logistic regression as the first modeling apporach. **binomial** as the family parameter indicates that the generalized linear model method we choose is logistic regression.

```
#Logistic regression (generalized linear model)
Train.glm <- train(disease ~ ., data=training,
                   method="glm",
                   trControl=control.repeat,
                   family="binomial")
#apply model on testing
Model.glm <- predict(Train.glm, testing)
Conf.glm <- confusionMatrix(Model.glm, testing$disease)
Conf.glm$table
```

```
##             Reference
## Prediction   disease no disease
##   disease         32          5
##   no disease       9         43
```

```
Sens.glm <- Conf.glm$byClass[c("Sensitivity")]
Spec.glm <- Conf.glm$byClass[c("Specificity")]
Acc.glm <- Conf.glm$overall[["Accuracy"]]
F1.glm <- F_meas(Model.glm, testing$disease)
Prec.glm <- Conf.glm$byClass[c("Precision")]
Prev.glm <- Conf.glm$byClass[c("Prevalence")]
```

Decision Tree

Decision trees are pretty suitable for the purpose of identifying if several indicators implicate diseases or not. We have used the train function from the caret package and set the rpart method. TuneLength is set to 10, which means that the function uses ten different hyperparameters and chooses the best fitting for the training data. The hyperparameter of rpart is the **complexity parameter**.

```
Train.dec.tree <- train(disease ~ ., data=training,
        method="rpart",
        trControl=control.repeat,
        tuneLength=10
)
```

- 82.90% of patients would have been diagnosed correctly to have heart disease (sensitivity)
- 77.10% of patients would have been diagnosed correctly to have no heart disease (specificity)
- The overall accuracy of the decision tree is 79.80%.

```
Model.dec.tree <- predict(Train.dec.tree, testing, type="raw")
Conf.dec.tree <- confusionMatrix(table(Model.dec.tree, testing$disease))
Conf.dec.tree$table %>%
  knitr::kable() %>%
  kableExtra::kable_styling(full_width = FALSE)
```

	disease	no disease
disease	34	11
no disease	7	37

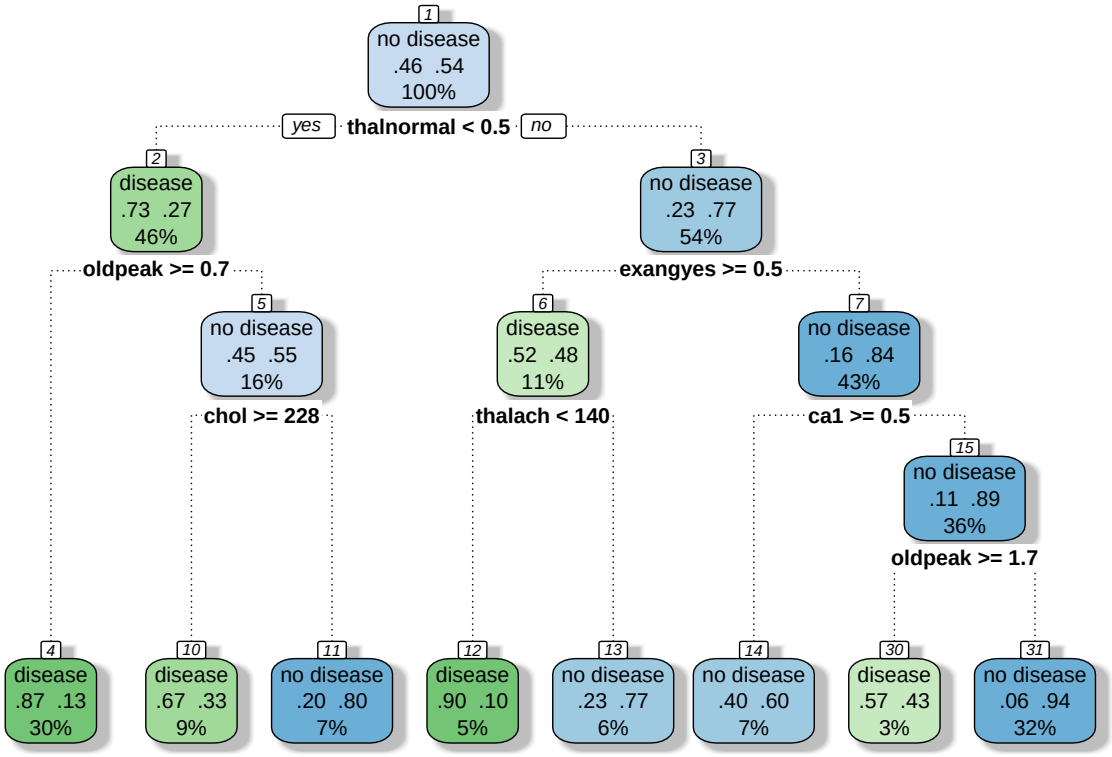
```
Sens.dec.tree <- confusionMatrix(Model.dec.tree, testing$disease)$byClass[c("Sensitivity")]
Spec.dec.tree <- confusionMatrix(Model.dec.tree, testing$disease)$byClass[c("Specificity")]
Acc.dec.tree <- confusionMatrix(Model.dec.tree, testing$disease)$overall[["Accuracy"]]
F1.dec.tree <- F_meas(Model.dec.tree, testing$disease)
Prec.dec.tree <- Conf.dec.tree$byClass[c("Precision")]
Prev.dec.tree <- Conf.dec.tree$byClass[c("Prevalence")]
tibble(
  Method=c("Decision tree"),
  Sensitivity=c(Sens.dec.tree),
  Specificity=c(Spec.dec.tree),
  Accuracy=c(Acc.dec.tree)
  ) %>%
  knitr::kable() %>%
  kableExtra::kable_styling(full_width = FALSE)
```

Method	Sensitivity	Specificity	Accuracy
Decision tree	0.829	0.771	0.798

The classification tree shows that the data was first split at the result of thal. Furthermore split at oldpeak, chol, exang, thalach and ca.

- 30% of all patients were predicted to have an abnormal thal and an oldpeak of ≥ 0.7
  - 87% of these patients were predicted to have heart disease
- 9% of patients were predicted to have an abnormal thal, an oldpeak of ≥ 0.7 and chol ≥ 228
  - 67% of these patients were predicted to have heart disease
- 5% of all patients were predicted to have a normal thal, exang and thalach < 140
  - 77% of these patients were predicted to have no heart disease
- 32% of the patients were predicted to have a normal thal, exercise induced angina, ca $\neq 1$ and have oldpeak ≥ 1.7
  - 94% of these patients were predicted to have no heart disease

```
fancyRpartPlot(Train.dec.tree$finalModel, sub="")
```



*Split of categorical data is between 1 (true) and 0 (false). e.g. thalnormal < 0.5 means all abnormal thal results.

Random forest

With the random forest as an extension of the decision tree there is a high potential in outperforming a single tree with several hundred trees. The number of trees is set to default (n=500).

```
Train.random.forest <- train(disease ~ ., data=training,
                             method="rf",
                             metric="Accuracy",
                             preProcess=c("center","scale"),
                             tuneLength=10,
                             trControl=control.repeat
                             )
```

```
Model.random.forest <- predict(Train.random.forest, testing, type="raw")
Conf.random.forest <- confusionMatrix(Model.random.forest, testing$disease)
Conf.random.forest$table %>%
  knitr::kable() %>%
  kableExtra::kable_styling(full_width = FALSE)
```

	disease	no disease
disease	34	8
no disease	7	40

```
Sens.random.forest <- confusionMatrix(Model.random.forest, testing$disease)$byClass[c("Sensitivity")]
Spec.random.forest <- confusionMatrix(Model.random.forest, testing$disease)$byClass[c("Specificity")]
Acc.random.forest <- confusionMatrix(Model.random.forest, testing$disease)$overall[["Accuracy"]]
F1.random.forest <- F_meas(Model.random.forest, testing$disease)
Prec.random.forest <- Conf.random.forest$byClass[c("Precision")]
Prev.random.forest <- Conf.random.forest$byClass[c("Prevalence")]
tibble(
  Method=c("Random forest"),
  Sensitivity=c(Sens.random.forest),
  Specificity=c(Spec.random.forest),
  Accuracy=c(Acc.random.forest)
  ) %>%
  knitr::kable() %>%
  kableExtra::kable_styling(full_width = FALSE)
```

Method	Sensitivity	Specificity	Accuracy
Random forest	0.829	0.833	0.831

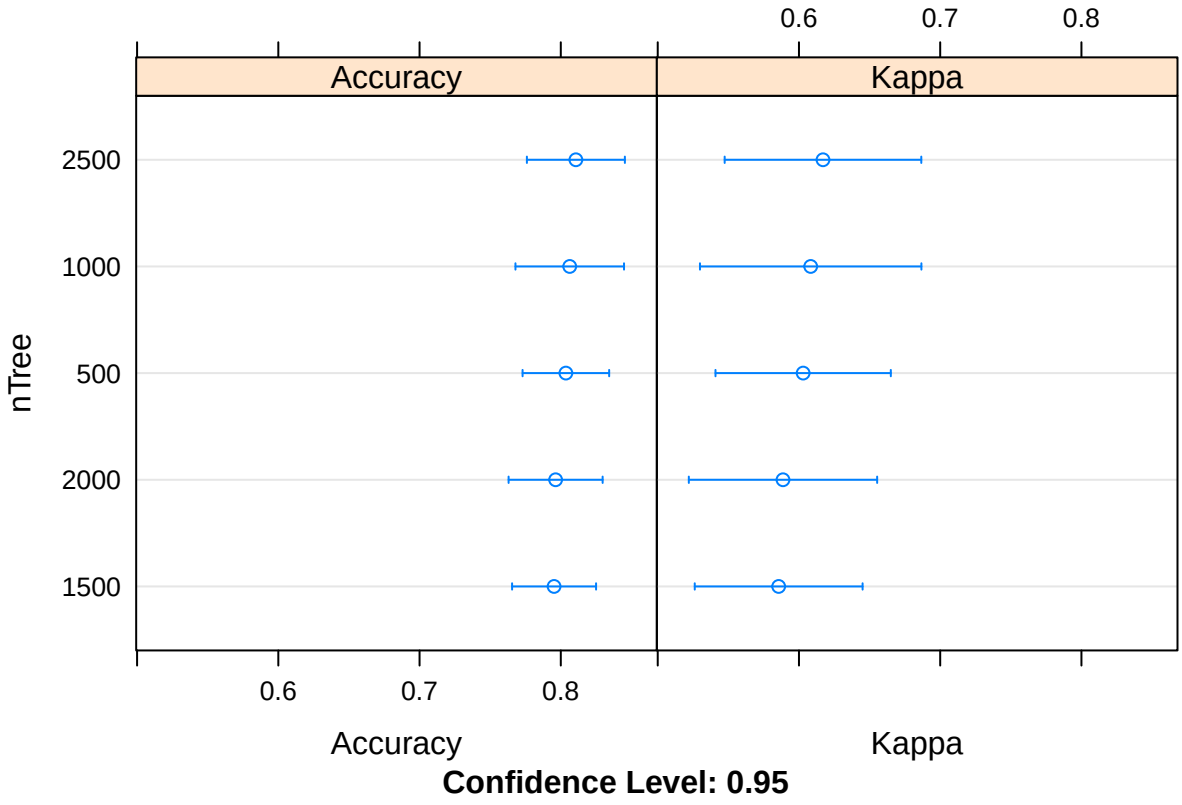
As we can see the random forest outperformes the decision tree in different evaluation metrics.
Now we run this function with different amounts of ntree:

```
modellist <- list()
for (ntree in c(500, 1000, 1500, 2000, 2500)) {
  fit <- train(disease~., data=training,
               method="rf",
               metric="Accuracy",
               preProcess=c("center","scale"),
               tuneGrid=expand.grid(.mtry=c(sqrt(ncol(training)))),
               trControl=control.repeat,
               ntree=ntree)
  key <- toString(ntree)
  modellist[[key]] <- fit
```

```
}
results <- resamples(modellist)
```

In the next plot we can see that the highest accuracy of the random forest is located at an amount of 500 trees.

```
dotplot(results, ylab="nTree")
```



Finally for random forest we have a look on the importance score of each variable in the dataset. We have already seen the three attributes with the highest importance in the decision tree as decision at quite high nodes.

```
## rf variable importance
##
##                          Overall
## oldpeak                   100.00
## thalach                    78.53
## thalreversible defect      62.10
## thalnormal                 58.72
## age                        58.46
## chol                       56.84
## trestbps                   51.26
## exangyes                   37.19
## slopeupsloping             37.11
## ca1                        35.96
## sexmale                    33.56
## cpnon anginal pain         32.68
```

```
## ca2                         28.48
## slopeflat                   25.45
## restecgnormal               19.33
## cptypical angina            12.47
## ca3                         12.23
## cpatypical angina           11.26
## fbs>120                      7.32
## restecgst-t abnormality      0.00
```

Support vector machines

With the support vector machines we choose a linear method. Using a tuneLength of 5 the train function searches for the optimal tuning parameter of cost.

```
train.svmLinear <- train(disease ~ ., data=training,
                         method = "svmLinear",
                         trControl= control.repeat,
                         preProcess=c("center",
                                      "scale"),
                         # tuneGrid=expand.grid(C=1)
                         tuneLength=5)
Model.svmLinear <- predict(train.svmLinear, testing, type="raw")
Conf.svmLinear <- confusionMatrix(Model.svmLinear, testing$disease)
Conf.svmLinear$table %>%
  knitr::kable() %>%
  kableExtra::kable_styling(full_width = FALSE)
```

	disease	no disease
disease	27	7
no disease	14	41

```
Sens.svmLinear <- confusionMatrix(Model.svmLinear, testing$disease)$byClass[c("Sensitivity")]
Spec.svmLinear <- confusionMatrix(Model.svmLinear, testing$disease)$byClass[c("Specificity")]
Acc.svmLinear <- confusionMatrix(Model.svmLinear, testing$disease)$overall[["Accuracy"]]
F1.svmLinear <- F_meas(Model.svmLinear, testing$disease)
Prec.svmLinear <- Conf.svmLinear$byClass[c("Precision")]
Prev.svmLinear <- Conf.svmLinear$byClass[c("Prevalence")]
tibble(
  Method=c("Svm linear"),
  Sensitivity=c(Sens.svmLinear),
  Specificity=c(Spec.svmLinear),
  Accuracy=c(Acc.svmLinear)
  ) %>%
  knitr::kable() %>%
  kableExtra::kable_styling(full_width = FALSE)
```

Method	Sensitivity	Specificity	Accuracy
Svm linear	0.659	0.854	0.764

K-nearest neighbors

Another modeling apporach in this project is the k-nearest neighbors model. Using a tuneLength of 3 the train function searches for the optimal size of neighbors to be based on as a tuning parameter.

```
Train.knn <- train(disease.no.disease~., data=training.onehot,
                method="knn",
                trControl=control.repeat,
                # tuneGrid=expand.grid(k=5)
                tuneLength=3
  )
```

```
Model.knn <- predict(Train.knn, testing.onehot, type = "raw")
Conf.knn <- confusionMatrix(Model.knn, testing.onehot$disease)
Conf.knn$table %>%
  knitr::kable() %>%
  kableExtra::kable_styling(full_width = FALSE)
```

	0	1
0	24	8
1	17	40

- disease (0); no disease (1)

```
Sens.knn <- confusionMatrix(Model.knn, testing.onehot$disease)$byClass[c("Sensitivity")]
Spec.knn <- confusionMatrix(Model.knn, testing.onehot$disease)$byClass[c("Specificity")]
Acc.knn <- confusionMatrix(Model.knn, testing.onehot$disease)$overall[["Accuracy"]]
F1.knn <- F_meas(Model.knn, testing.onehot$disease)
Prec.knn <- Conf.knn$byClass[c("Precision")]
Prev.knn <- Conf.knn$byClass[c("Prevalence")]
tibble(
  Method=c("K-nearest neighbors"),
  Sensitivity=c(Sens.knn),
  Specificity=c(Spec.knn),
  Accuracy=c(Acc.knn)
  ) %>%
  knitr::kable() %>%
  kableExtra::kable_styling(full_width = FALSE)
```

Method	Sensitivity	Specificity	Accuracy
K-nearest neighbors	0.585	0.833	0.719

Results/Evaluation metrics

With different models passed through there is now a bunch of metrics by which we can justify what model is the best to use for the heart disease problem. Besides collecting standard measures for evaluating different models for the problem, we have to take a closer look on the problem itself and its sector of use and consequences of using the model in order to treat patients right.

Method	Sensitivity/Recall	Specificity	Accuracy	F1-score	Precision
Logistic regression	0.780	0.896	0.843	0.821	0.865
Decision tree	0.829	0.771	0.798	0.791	0.756
Random forest	0.829	0.833	0.831	0.819	0.810
Support vector machine (linear)	0.659	0.854	0.764	0.720	0.794
k-nearest neighbors	0.585	0.833	0.719	0.658	0.750

Usually the focus of prediction approaches in the medical field should be on reducing false negative results to prevent overlooking diseases of a patient. Therefore receiving false negative cases is worse than receiving false positive cases. With this approach we will assess the different models.
Sensitivity/Recall also called true positive rate provides information about how precise the model is in order of finding all patients with heart disease. We can see that the decision tree as well as the random forest have the highest sensitivity of 82.9% which means, that eight out of ten patients that have heart disease were diagnosed correctly. K-nearest neighbors estimate shows a poor result for sensitivity, diagnosing six out of ten patients correctly.
In terms of specificity, all methods show a good result.
Logistic regression and random forest are the only methods that show an overall accuracy over 80.0% as well as a F1-score over 80.0%.
With a focus on keeping false negative cases as low as possible the random forest brings solid measurement accuracies over 80.0% in every single metric.

Conclusion

Since some methods like random forest and logistic regression show sufficient results predicting heart disease, the attributes of the dataset seem to be good indicators. However the dataset was released in 1988 around 40 years ago. It can be assumed that there are additional attributes to diagnose patients to have heart disease or not but more precisely.