University of Michigan

STATS 415

Data Mining and Statistical Learning

Final Project Report

Does Your Muscle Speak Your Heart?

Data Analysis Based On NHANES

Jialin Kou	gracekou
Qingyang Liu	liuqingy
Ziwei Tian	ziweit
Kaggle account:	**qingyangliu0605**

Contents

1 Introduction and Question Description

Cardiovascular disease (CVD) is the leading cause of death worldwide, accounting for around 17.5 million deaths per year. While total CVD mortality has fallen dramatically in recent years, scientists are still interested in various physical predictors that could be used to indicate any potential cardiovascular risks.

Browsing through related literature work, what surprises us the most are some controversial studies aiming to examine the association between muscle strength and cardiovascular disease (CVD) risk. According to the 2017 Swiss CoLaus study, there is no association between absolute handgrip strength and cardiovascular disease. However, a study in 2018 based on the Korean NHANES suggests a significant inverse correlation between these two variables stratified by sex. Therefore, we are enlightened to explore this puzzle based on the US NHANES.

In our report, we are going to explore **(1) whether there exists a gender-based correlation between the handgrip muscle strength and a bunch of suggested cardiovascular risk factors** such as blood pressure, cholesterol, blood glucose, hypertension, waist circumference, etc.in our regression session. Then in the classification session, we are interested in examine **(2) whether above-mentioned suggested factors show strong correlations with the existence of coronary disease,** which is one the representative among all cardiovascular diseases.

2 Data Preprocessing

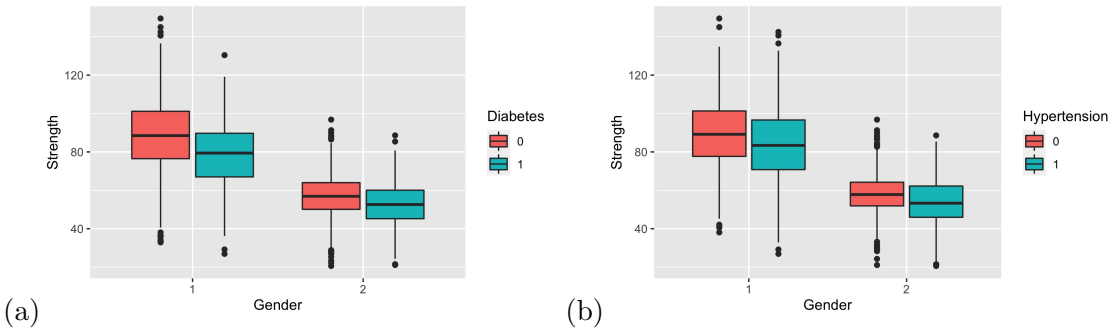
All the data used are obtained from The National Health and Nutrition Examination Survey (NHANES) from the year 2011 to 2014. For the regression analysis, we selected **handgrip muscle strength** from the database 'Muscle Strength - Grip Test' as our numerical response. For the classification problem, **whether the respondent has coronary heart disease** is filtered out from the cardiovascular health questionnaire and coded as a binary response. To study our research question, we search all the risk factors for cardiovascular disease and retrieve the data from different laboratories, examinations, and questionnaires. Specifically, we obtained age, race, gender, and poverty ratio (a ratio of family income to poverty guidelines) from demographic data; systolic and diastolic pressure, blood glucose level, LDL, triglyceride, HDL from different laboratory datasets; waist circumference and days of drinking alcohol form questionnaire; smoke, has hypertension and has diabetes or not from questionnaires and then transformed into factor levels of 2. All of the NAs are omitted due to the relatively large size of the data. Then all these predictors and responses are joined by each respondent's unique ID "SEQN", merged into a whole data frame of 17 predictors, 3108 observations for regression and 17 predictors, 3168 observations for classification. In both part of classification section and regression section, we bucketize the continuous variable **age** into equal length of chunks for the sake of a more convenient visualization in the exploratory data analysis part.

3 Regression Problem

3.1 Data Exploratory Analysis

Since we are interested about the relationship between the hand grip muscle strength and a series of variables that are believed to be indicators for cardiovascular risk, how specifically how the relationship may differs across gender groups, we firstly make some

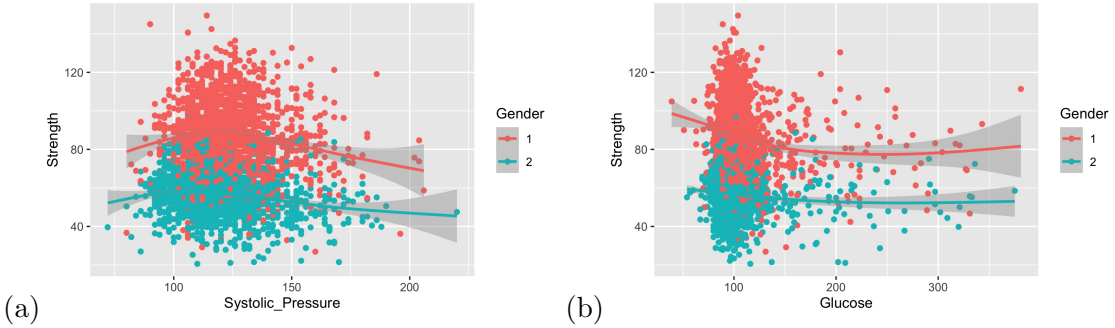
boxplots adjusted for gender. From Fig.1(a), we can tell that the muscle strength of male(1) and female(2) is pretty disparate, as we should expect. Inside each gender group, the muscle strength of observations without diabetes is higher than the strength of those who has been diagnosed diabetes. The mean difference is slightly larger in male group. In Fig.1(b), we witness a similar trend that within each gender group, the muscle strength of those who has been diagnosed with hypertension on average has less muscle strength than those who have no hypertension.



(a) (b)

Figure 1: muscle strength interacted with diseases, adjusted for gender

In Fig.2, we use scatterplot with smoothing lines to show a general trend between the muscle strength and two cardiovascular health indicators: systolic blood pressure and glucose. According to WHO standards, systolic pressure over 120 mmHg is defined as an abnormally high level. In 2(a), we could identify a decreasing trend of muscle strength as the value of systolic blood pressure increases from roughly 120 mmHg in both gender groups. However, we could not identify any obvious trend between the glucose level and muscle strength as displayed in 2(b).



(a) (b)

Figure 2: muscle strength interacted with blood pressure and glucose level, adjusted for gender

However, we explore some potential collinearity from Fig.3, where the muscle strength is manifested to be highly correlated with the person's age, which is also a natural common sense.Despite, this, we still notice from Fib.3(b) that as the age increases, the ratio of observations with very high waist circumference decreases. Also within each age group, as the waist circumference increases, the variation of muscle strength increases, and the smoothing lines of 15-30 and 30-45 age group even manifest a decreasing trends as the waist circumference becomes extremely high. This serves as a interesting hint for us to assume that people with abnormally large waist circumference is associated with weaker muscle strength adjusted for age and gender, and this is exactly a indicator of a high cardiovascular risk.

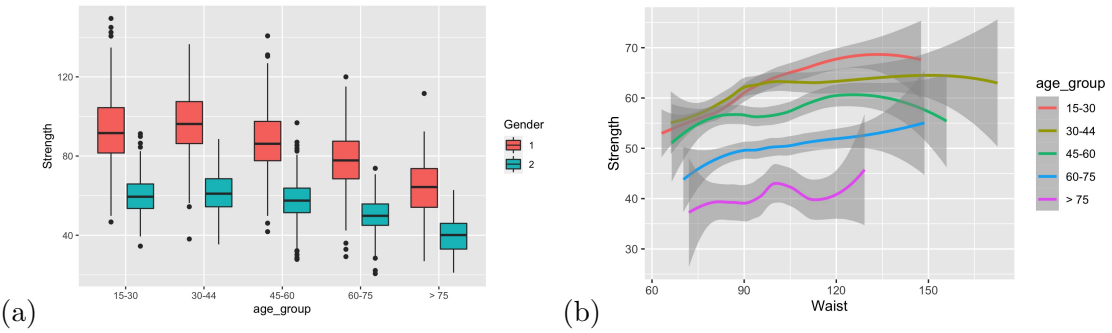


Figure 3: muscle strength interacted with cholesterol level, adjusted for gender

3.2 Fitting Models

3.2.1 Benchmark linear regression model

In order to examine all possible correlation that are potential missing from eyeballing the plots, we would firstly fit a the most commonly used simple linear model using the full dataset as a baseline to see the preliminary relationships between cardiovascular disease risk factors and muscle strength, so that we have a pretty decent benchmark model to compare with as we develop other fancier and more precise models later. From the linear regression output, we see that age is statistically important and negatively correlated with muscle strength, meaning an increase in age decreases the ability to grip while holding other variables constant. Also, the poverty ratio is positively correlated with muscle strength, which aligns with our intuition as people who suffer from poverty may have food security problems and are less healthy than people who don't. Other statistically significant variables such as diastolic pressure and waist are positively correlated with muscle strength, suggesting that an increase in these values may increase the strength. One notable difference between the relationship between men's and women's muscle strength and cardiovascular disease risk factors is that the LDL level and whether the respondent is overweight are important to men's muscle strength but not for women, and whether or not the respondent plays a key role in determining women's muscle strength but not men's. Furthermore, the test MSEs and 10-fold cross-validation error are also significantly different for men and women at a level of approximately 200 versus 90. This shows that at least for a linear model, muscle strength seems to be more negatively correlated with cardiovascular risk, which was converse to previous studies.

3.2.2 Variable selection using Best Subset Selection

However, despite its good performance and low bias, the large number of covariates may cause high variation when fitting on a different dataset. Therefore, we wish to choose a parsimonious model that simultaneously preserves high performance. To achieve this, we adopt the best subset selection method to choose predictors that are not only statistically significant but also preserve a globally minimum BIC score, which indicates a good fitting without the overfitting problem. Despite the best subsetting subset selection method being usually computationally expensive, we only have 16 predictors, and hence can easily overcome this problem. From Table 1, we observe that the models with selected predictors for both men and women have relatively lower test MSEs (266.8 compared to 267.7 for women, 89.6 compared to 90.5 for women) and 10-fold cross-validation error (216.8 compared to 218.5 for men, both 84.4 for women), indicating that our models have a valid performance in explaining muscle strength. We find

	Gender	Linear Model	Linear Model (Features Selected By Best Subsetting)	GAM Model	Ridge based on GAM Model	Generalized Boosted Regression Model
Training MSE	Male	211.776	213.554	208.479	209.045	**191.247**
	Female	81.090	82.728	80.288	80.557	**71.871**
Test MSE	Male	267.651	266.777	264.020	294.631	**258.442**
	Female	90.515	89.571	87.313	99.189	**83.676**
10-Fold CV Error	Male	218.398	**216.782**	217.114	217.363	\
	Female	84.418	84.477	84.203	**84.099**	\
BIC	Male	11056.490	**11010.070**	11057.19	\	\
	Female	8495.463	**8448.028**	8505.134	\	\

Table 1: Summary of Model Performances

that both age, race, poverty rate, diastolic pressure, and waist circumference are statistically significant while modeling the muscle strength, while LDL level and whether the respondents have diabetes are only selected for men. The differently selected variables suggest that there could be a difference in the relationships between muscle strength and cardiovascular disease risk factors based on sex.

3.2.3 Non-linear Relationship Exploration by GAM Model

Although linear regression provides a good approximation of muscle strength, it ignores possible non-linear relationships between the response and covariates. During the exploratory data analysis, we observe that waist circumference and other body measurements such as LDL and glucose level have strong non-linear relationships with muscle strength. Besides the relationship between muscle strength and age as well as blood pressure (both systolic and diastolic) may vary in the different ranges: for example, the grip-ability would increase for people under 20 since they are still developing, and would decrease for elder people as senescence occurring. Therefore, we employ the Generative Additive Model, which has the ability to model very complicated nonlinear relationships and is hence suitable for our purpose. As shown in Table 1, the resulted model has a relatively lower test MSEs (264.0 and 87.3) and a 10-fold cross-validation error (217.1 and 84.2), showing a good fit. It also shows that muscle strength is indeed non-linearly correlated with age, diastolic pressure, and waist circumference. However, these quadratic and splined terms greatly increase the variability of this model and cause it to overfit. To address this issue, we want to further improve this model by reducing the variability. We hence utilize ridge regression to shrink the coefficients in the previous model, and the resulting model has a lower 10-fold cross-validation error for both men and women. We observe that except for triglyceride, HDL level, and alcohol, all of the predictors have a coefficient whose absolute value is larger than 0.1, showing that they have a relatively important effect on muscle strength.

3.2.4 Accuracy improvement by Boosting

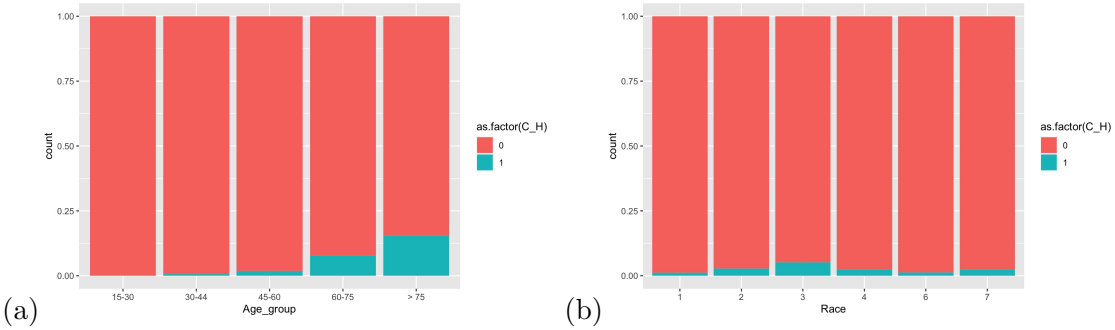
To improve based on the previous two models, we also propose a boosting model that combines different simple models into a single regression model that has both low variance and low bias. Despite the relatively slower training time, the ensemble boosted regression tree achieves the lowest test MSE (258.4 and 84.1), shown in Table 1. The

implicit feature selection method deduces that age, waist circumference, race, poverty ratio, and LDL level are the most important explanatory variables associated with muscle strength for women, while age, waist circumference, race, poverty ratio, and diastolic pressure are the most important variables for men. This aligns with our previous results showing that different cardiovascular disease risk factors are affecting muscle strength for different sex.

4 Classification Problem

4.1 Data Exploratory Analysis

First and foremost, we would like to see how the binary response **CH** (0 represents no disease, 1 represents having been diagnosed with the disease) is distributed across different age groups and racial groups. Intuitively, coronary heart disease is strongly gene-related and age-related, and our assumption is further proved by the ratio barplot in Fig.5(a). We barely find no cases of diseases among the all observations under 45 years old. As the age increases in the last three age groups, there is an obvious increasing trend between the ratio of W/disease cases and Wo/disease cases. This trend of disease rate with the person's age is further confirmed by the statistical significance of the predictor **age** in our fitted models later on. Then looking at Fig.5(b), we cannot identify any significant difference of the ratio between different racial groups. This brings us a intuition that the incidence of CH disease may not be strongly correlated with some genetic factors affected by the person's **race** as we have assumed earlier.



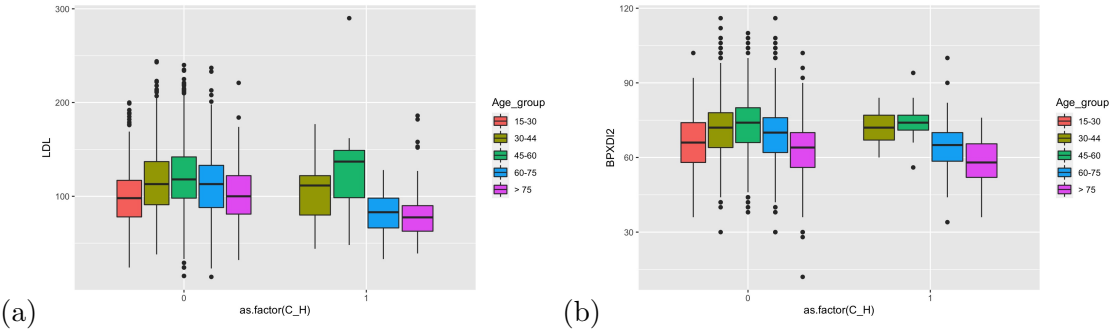
(a) (b)

Figure 4: muscle strength interacted with diseases, adjusted for gender

Fig.6 displays other interesting trends between the some measured physical-related variables and the indicator of having CH disease adjusted for age. In fig.6(a), we observe a decrease of average LDL level in the disease cases among age group of '60-75' and 'above 75' compared with observations without CH disease. In fig.6(b), we also discover a decrease of the mean of **diastolic blood pressure** among observations in '60-75' and 'above 75' age groups. Since these two groups are the most risky groups of having CH disease and therefore is the most representative groups among all, We can initially speculate on a negative correlation between diastolic blood pressure/LDL level and the risk of having coronary artery disease.

4.2 Fitting Models

To investigate the decision boundary of the binary response, we randomly split the data into the training data (approximately 0.7 of the whole data) and the test data (approximately 0.3 of the whole data). Using the training data, we apply various classification methods to explore the shape of decision boundaries of the binary response



(a) (b)

Figure 5: CH disease interacted with LDL and blood pressure, adjusted for age groups

variable **CH**, with a value of 1 or 0, indicating whether this observation ever had coronary heart disease or not.

Before fitting models, we employ **best subset selection** to reduce potentially insignificant predictors, which allows us to try independent combinations of different numbers of predictors and find the best one with the lowest BIC. BIC is the metrics we prefer because it reflects a model's goodness of fit and simplicity. Based on the selection result, including both continuous and categorical predictors, we conduct logistic regression modeling, linear discriminant analysis (LDA), and quadratic discriminant analysis (QDA). Except for these methods, we also use the random forest classifier to explore the decision boundary further.

4.2.1 Logistic regression

Assuming the binary decision boundary is linear, we start our exploration with logistic regression and used the nine predictors selected by the best subset selection. The model's output presents that, **age**, **gender**, **hypertension**, and **diabetes** predictors are statistically significant with at least 95% confidence level. **Race** and **diastolic blood pressure** are statistically significant with approximately 90% confidence level. Since both training error (0.033) and test error (0.037) are small, the model results are relatively reliable.

According to predictors' coefficients, age is positively associated with the probability of ever having coronary heart disease for any individual. Compared with Mexican Americans (the baseline category of race), people of other races are more likely to have coronary heart disease. Comparing males, females have a smaller chance of having the disease. Moreover, individuals with diabetes and hypertension have a higher probability of getting coronary heart disease.

4.2.2 LDA and QDA classification

Under the linear boundary assumption, we apply Linear Discriminant Analysis(LDA) to classify the response **CH** by the identical predictors in the logistic model. The model results present similar associations between the response variable and predictors. However, the training error (0.0370) and test error (0.0505) of the LDA are slightly larger than the measurements in the logistic regression model. Thus, compared with LDA, the logistic regression model would classify the response variable better.

Although the logistic model may classify the response variable well, we want to explore the possible non-linear boundary to check whether a linear boundary is more suitable for our sample. Thus, we also utilize the Quadratic Discriminant Analysis(QDA) with the same predictors. However, both training(0.0780) and test(0.0873) errors are

greater than the values in the previous two classification models. Therefore, we conclude that a linear decision boundary is more suitable for the response variable **CH**.

4.2.3 Random Forest Classifier

Since we make analyses based on predictors obtained from the best subset selection, we try to classify the response variable by other subsets of predictors via the random forest classifier. After comparing the test error of random forest models with various predictor sizes, the model with eight predictors has the lowest test error (0.0358). However, the model outputs show that the six most significant predictors are: **systolic blood pressure**, **age**, **Waist circumference**, **triglyceride**, and **High-density lipoprotein**, which are different from the results in the logistic model.

The training error is 0, implying an overfitted model, but the the test error is slightly smaller than the values of any previous classification method. Therefore, the result could be also reliable.

4.3 Model Comparison

Although all classification models discussed above have a relatively low test error, we need to evaluate the model performances based on their true positive rate(TPR), true positive rate(TNR), and ROC curves. Fig.7(a) shows the value of TPR and TNR for each model. The second column of the table presents the probability threshold for classifying the category of the response variable. We usually classify the response as having coronary heart disease if the probability is greater than 0.5, otherwise classifying as no disease. However, except for the QDA, with a 0.5 probability threshold, the TPRs of other models are close to zero. Our purpose is to correctly classify an individual who has coronary heart disease into the proper category. Hence, a high TPR is more valuable.

As a result, we deal with the trade-off of the TNR and TPR by adjusting the probability threshold. When the threshold decreases to 0.04, the improved TPR varies between 0.7 and 0.8. The TPR and TNR after the adjustment are presented in Fig.7(b). We can observe that all TPRs become much greater. The logistic regression and LDA have the highest TPR, consistent with our previous statement that a linear decision boundary is more suitable for the response variable.

(a)

Classification method	Threshold	TPR	TNR
Logistic regression	0.500	0.0571	0.997
LDA	0.500	0.0857	0.982
QDA	0.500	0.400	0.932
Random forest	0.500	0.0286	0.998

(b)

Classification method	Threshold	TPR	TNR
Logistic regression	0.04	0.800	0.814
LDA	0.04	0.800	0.814
QDA	0.04	0.743	0.778
Random forest	0.04	0.743	0.753

Figure 6: CH disease interacted with LDL and blood pressure, adjusted for age groups

The adjusted TPR results also accord with what is shown in Fig.8 below, which displays the ROC curves of 4 models. Despite that the curves tangled together, we could still tell that the curves of logistic regression model and LDA classifier are slightly more convex towards the upper-left corner, promising both a good TPR and a decent TNR.

In general, before adjusting the probability threshold, the low TPRs of all models indicates that they perform poorly in classifying people with coronary disease to the

correct category. After the adjustment, all models have higher TPRs. However, the significant predictors in different approaches vary a lot, which implies different associations between the response variable and predictors in different models.

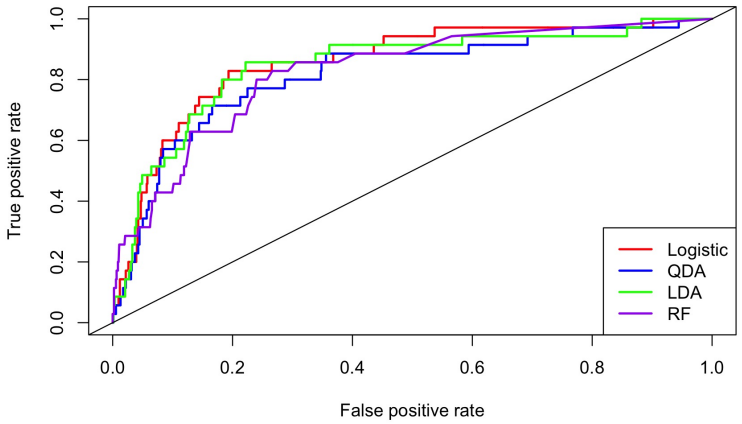


Figure 7: ROC curves of 4 models

5 Discussions and Conclusions

5.1 Regression section

conclusion: In conclusion, we observe that the huge difference in test MSE and cross-validation error of all our models for men and women shows that muscle strength is more strongly correlated with cardiovascular risk factors for women, compared to that for men. Specifically, age, race, waist circumference, poverty ratio, and diastolic pressure are the most statistically significant variables in modeling muscle strength for women, while age, race, waist circumference, poverty ratio, and LDL are the most statistically significant variables in modeling muscle strength.

limitations: One possible limitation for both our regression and classification models is the high multicollinearity among our predictor variables. We observe that age, as an important predictor of muscle strength and coronary heart disease, is also highly correlated with other predictors such as hypertension. Also, variables such as blood glucose level and diabetes or not, systolic pressure and hypertension or not are also highly correlated with each other that may undermine our models.

5.2 Classification section

conclusion: Based on the test error rate of each model, it is hard to conclude whether a linear or a more complex decision boundary is more suitable to classify the binary response. However, after reducing the probability threshold to 0.04, it seems like that a linear boundary is more preferable. Moreover, the priorities of the logistic regression and LDA are also shown in the ROC curve. In conclusion, both logistic regression and LDA can classify the binary response better. **Age**, **gender**, **hypertension**,**diabetes**,**Race** and **diastolic blood pressure** are highly associated with the probability of having coronary heart disease.

limitations: Although our classification analyses are aimed to find the association between having coronary heart disease and other factors for any individual, our conclusion varies from what we expected before. This may caused by the unreliability of the binary response data. The data are collected through questionnaires, which asked whether you have ever been told to have a coronary heart disease. Since it could be chronic or acute, it is possible that the observation had already recovered, despite the response value is still 1. Therefore, to improve this study, we can use a better data set which contains the current medical and health conditions for each observation.

Moreover, one possible reason of the huge improvement of TPR after reducing the threshold to 0.04 is that the overall probability of getting coronary heart disease is extremely low in our sampling data. Therefore, in future studies, the quality of results may be improve by dropping younger observations, who have a extremely low possible to have the disease.

6 Kaggle Prediction Write-up

Preparing Data

Following the instruction, we download corresponding data sets from NHANES website and combine each year's data together. After appropriate data preprocessing, e.g. dropping columns and rows with NA values, we successfully obtain the 8921*145 training set. The very next thing to do is data scaling. Since the dataset we are working on contains different categories of data that varies highly in variables' ranges and units. Scaling data promises us a faster training process as well as a higher prediction accuracy.

Feature Selection with LASSO, PLS and Random Forest

As the first step, we try LASSO regression and pick the 1-SE lambda, which is the maximum value of lambda within 1 standard deviation of the best lambda that minimized CV error. This selected lambda gives us a group of 57 predictors.

Then we try a further selection on the updated training set using PCR methods, but the optimal number of components was still 57. Nevertheless, we still fit two models with the selected predictors, and generate some histogram plots of the prediction in comparison with the response in the training set as a naive evaluation method. The Kaggle accuracy of submissions based on these two methods are roughly around 0.35.

We then try a further feature selection on the current 57 predictors using random forest models. Random forest model with 57 predictors reached around a 0.55 test R-Squared. Another advantage is to visualize the importance of each fitted predictors, and we further select the top 15, 20, 25 and 30 important predictors to refit the random forest model. We find that a model using the top-20 important predictors with parameter **ntree = 1000** and **mtry = 15** generally gives us the best r-squared on Kaggle, which fluctuates around 0.6.

Tuning a better neural network model

The last and the best model we tries is neural networks using a similar frame of code from our lab materials. We reuse the 15, 20, 25 and 30 predictors selected by the previous random forest models, and split the dataset into a 8:2 ratio of training set and validation set. After trying each trick individually, e.g. weight decay, dropout, learning rate scheduling and batch normalization, we combine them pair-wisely to see the performance of the model on the hold-out set, and those trick-combined models

generally perform better especially when we involve the l2-norm regularizer and batch normalization.

We do a permutation work of different values in 5 parameters (the number of layer between 1 to 3, the number of units inside each hidden layers between 32 to 256, the learning rate between 0.01 and 0.001, the regularizer coefficient of each layer in the range of 0.01 and 0.00001). After adaptive trials and adjustments, we settle on a model architecture with 256 units in the first layer and 128 units in the second layer. The l2-norm regularizer rate is fixed at $10^{-}6$. This combination of parameters generally spit out a high accuracy. The last tuning challenge is to deal with the batch size and learning rate, which are two important parameters suggested to be tuned in the final step according to many deep learning studies. Until the due date of the Kaggle competition, we still have not figure out the absolute ideal values for parameters mentioned above. Nevertheless, the model trained under a learning rate between 0.001 and 0.005 and a batch size around 120 gives a pretty decent performance on Kaggle.

As a wrap-up, our model performance based on the public Kaggle data could be stablized around 0.65-0.66 with tiny parameter changes.

7 Contributions of Team Members

Overall, we three members supported each other and made a great collaboration job in every part of this project. Qingyang is mainly responsible for the data processing, model tuning, and modeling for the Kaggle prediction part, and is also involved in the classification part. As for the report writing, she takes charge of the introduction, data exploratory analysis, Kaggle write-up and the Appendix part. Jialin is also one of the main contributors to our Kaggle competition part, doing loads of model tuning work with the NN model. She is mainly responsible for the Classification part, including data processing, model fitting, plot making and the write-up part for the classification part in the report. Ziwei makes great efforts in browsing related literature work and searching useful variables on the NHANES dataset, and she is mainly charged for the regression part, including data processing, model fitting, plot making and the write-up of the regression section in this report. Jialin and Ziwei makes collective efforts in writing the discussion and analysis part.

8 Appendix

8.1 Instruction for Reproducibility

8.1.1 Opened Ended Questions

All of raw data, processed data and Rmd files for two opended ended questions are saved in our project Github Repo. Please note that the all dependencies for the open-ended questions are in the **master** branch instead of the main branch.

If you want to replicate the process of raw data processing, all the raw data we used are placed in the fold named raw data. For example, the .XPT file named 'DEMO1112.XPT' means the demographic data collected during 2011-12 period. Or you can directly access to the processed data files in the fold named 'procssed data' to have a taste of what variables we used, and also detailed information about the dataset we work on. The data-processing code can be found in Data-Cleaning.Rmd.

If you want to replicate all the plots we creates and the model results, please refer to the two Rmd.file saved in the fold named 'code'. For the regression section, run through Regression.Rmd. For the classification section, run through Stats415Project-Classification.Rmd.

We also attach the knitted PDF version of these two .Rmd files at the very end of this report.

8.1.2 Kaggle Competition

Again, all of the raw data, processed data and Rmd files containing all codes are placed at our project Github Repo, or you can go the the next url: `https://github.com/Qingyang-Liu47/STATS415-FINAL-PROJECT.git`. Remember to switch to the **main branch** to find all relevent data and codes of the Kaggle part.

You can find all raw data downloaded directly from the websites of NHANES, and properly grouped in the files named by the number of the years that the data was collected, e.g. [2009-2010].

If you want to replicate the data-preprocessing part and arrive at the processed trainning set, please run kaggle-data-processing.Rmd in the fold named 'Kaggle Code'. You can also directly get the processed data kaggle-train.csv that we saved in the 'kaggle data' fold.

To replicate the kaggle prediction using the Lasso model and PLS model, please run the Rmd file named kaggle-prediction.Rmd in the 'kaggle code' fold. To replicate the NN network part, the code is placed in the same directory, just click here to download the code for NN models and run through the code.

We also upload all our submission history in the fold named kaggle-submission-history. The files are properly named according to the date it was created, the methods we used and a local test R-squared we predicted.

8.2 PDF files for open-ended questions

(see next few pages)

Regression

Ziwei Tian (ZIWEIT)

4/16/2022

```
library(haven)
library(tidyverse)
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.1 --
```

```
## v ggplot2 3.3.5     v purrr   0.3.4
## v tibble  3.1.6     v dplyr   1.0.8
## v tidyr   1.2.0     v stringr 1.4.0
## v readr   2.1.2     v forcats 0.5.1
```

```
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

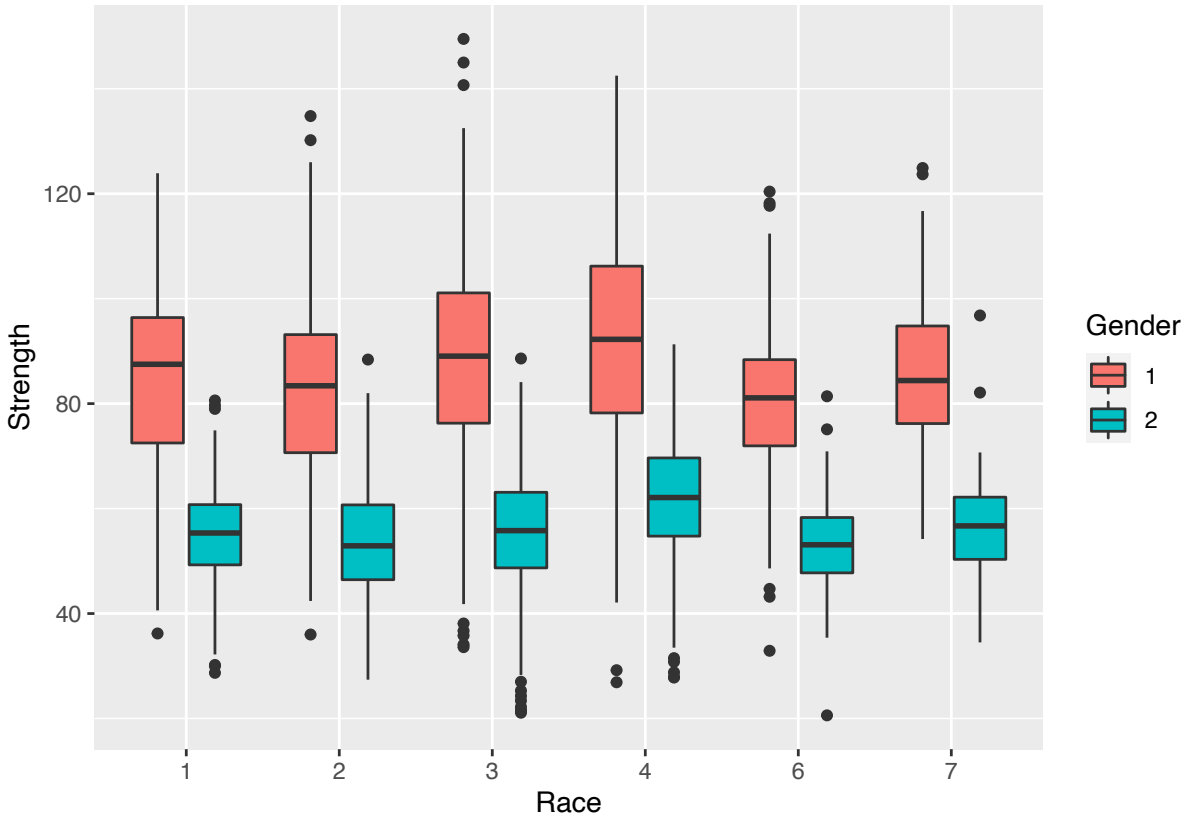
```
setwd("~/Desktop/Stats 415/Project")
```

```
total <- read.csv("Regression.csv")
# Transform categorical data as factors
factor_predictors <- c("Race", "Gender", "OW", "Smoking", "Hypertension", "Diabetes")
factor_predictors_id <- match(factor_predictors, names(total))
for (i in 1:length(factor_predictors_id)) {
  predictor_id <- factor_predictors_id[i]
  total[,predictor_id] <- as.factor(total[,predictor_id])
  colnames(total)[predictor_id] <- factor_predictors[i]
}
```

EDA

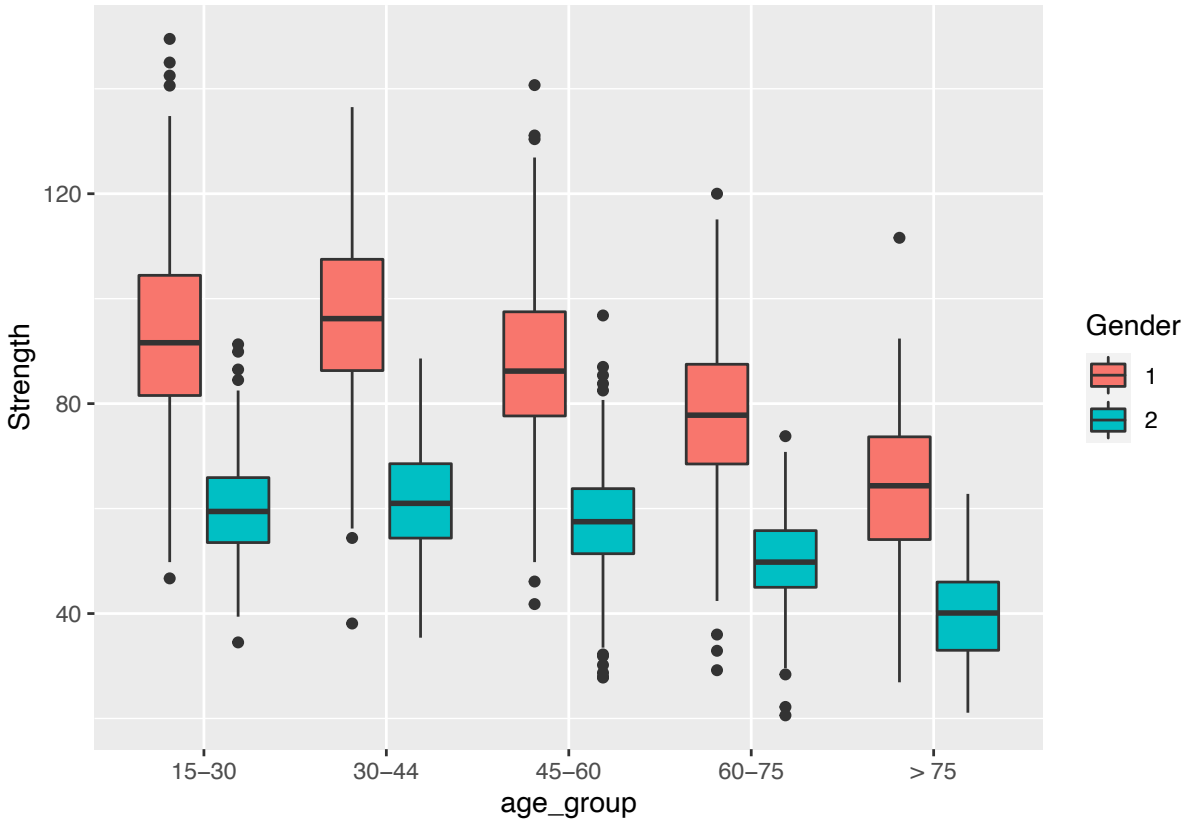
Strength v.s. Demographic data

```
#Strength vs. Race and Gender
ggplot(total, aes(x=Race, y=Strength)) + geom_boxplot(aes(fill=Gender))
```



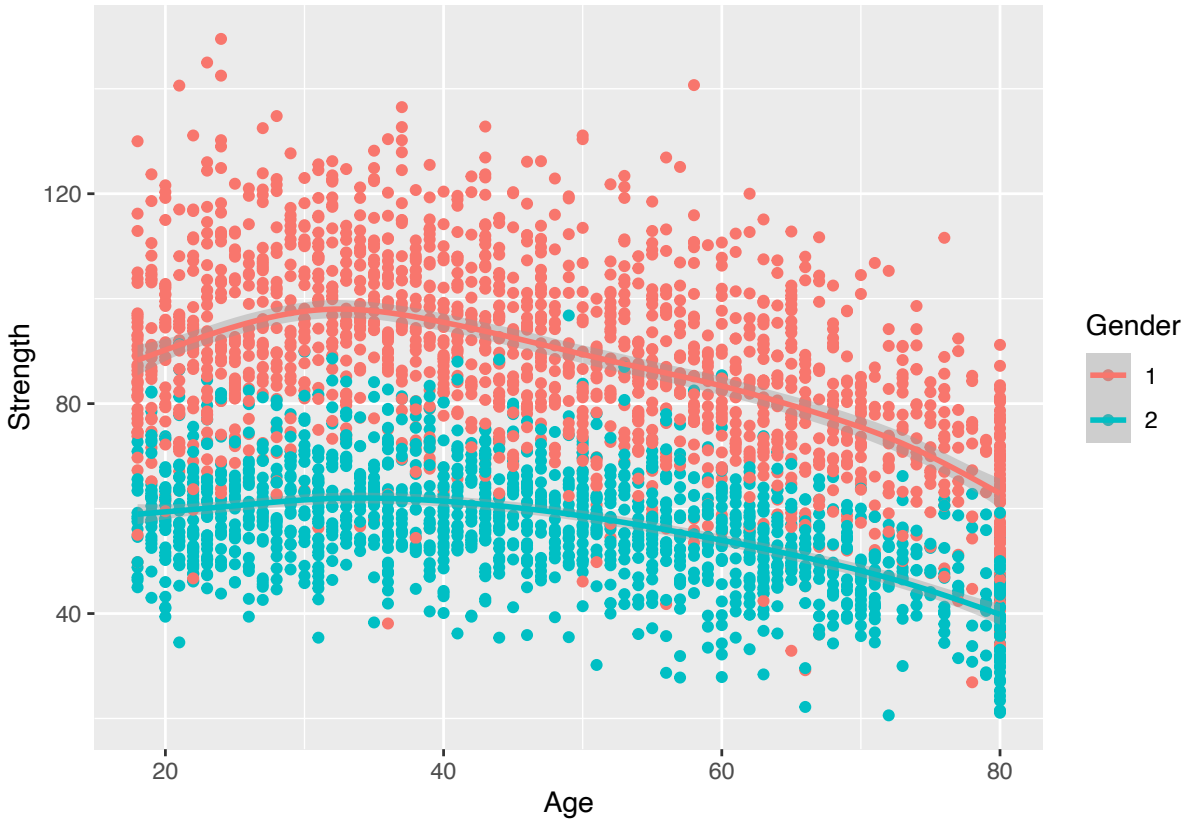

```
#Transform age into 5 different age groups
total <- total %>%
  mutate(
    # Create categories
    age_group = dplyr::case_when(
      Age <= 14             ~ "0-14",
      Age > 14 & Age <= 30 ~ "15-30",
      Age > 30 & Age <= 44 ~ "30-44",
      Age > 44 & Age <= 60 ~ "45-60",
      Age > 60 & Age <= 75 ~ "60-75",
      Age > 75              ~ "> 75",
    ),
    # Convert to factor
    age_group = factor(
      age_group,
      level = c("0-14", "15-30","30-44","45-60", "60-75", "> 75")
    )
  )
```

```
ggplot(data = total, aes(x = age_group, y = Strength)) +
    geom_boxplot(aes(fill=Gender))
```




```
ggplot(data = total, aes(x = Age, y = Strength)) + geom_point(aes(col=Gender)) + geom_smooth(aes(x = Ag
```

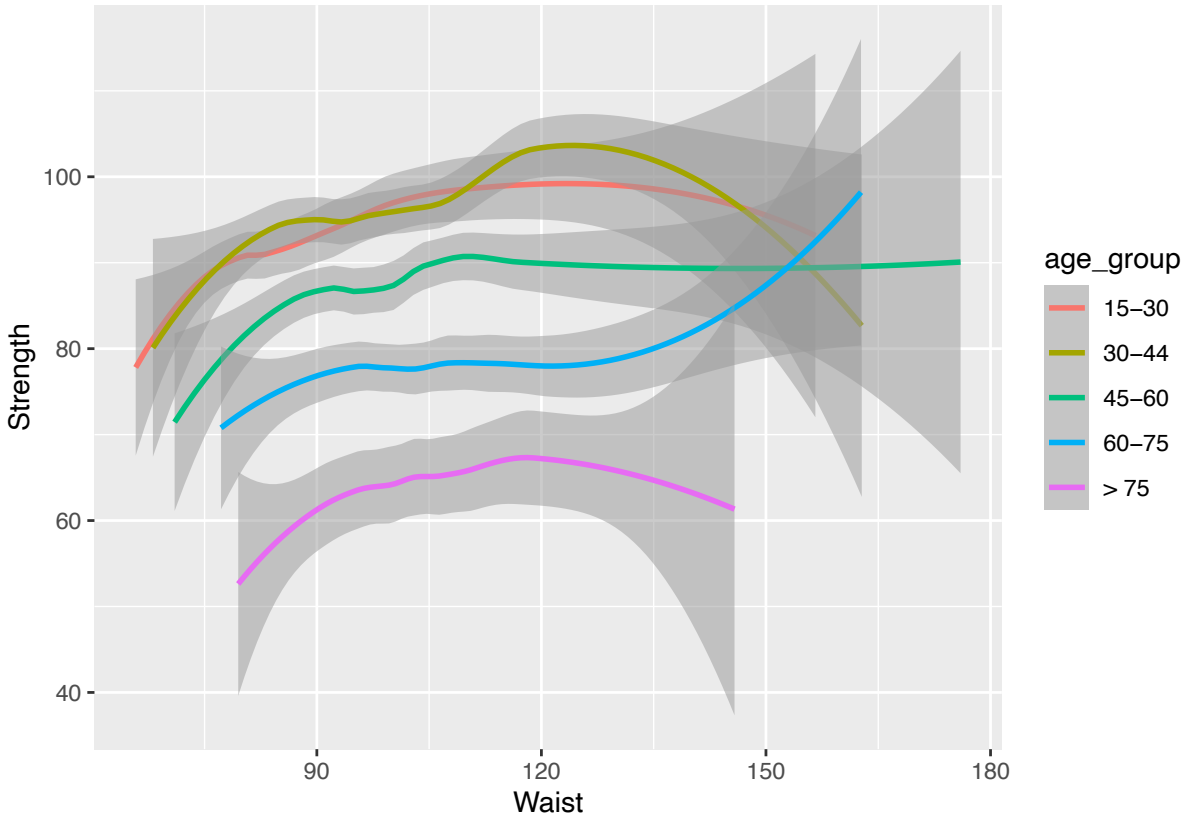
```
## `geom_smooth()` using method = 'gam' and formula 'y ~ s(x, bs = "cs")'
```



Strength vs. Body Measurements

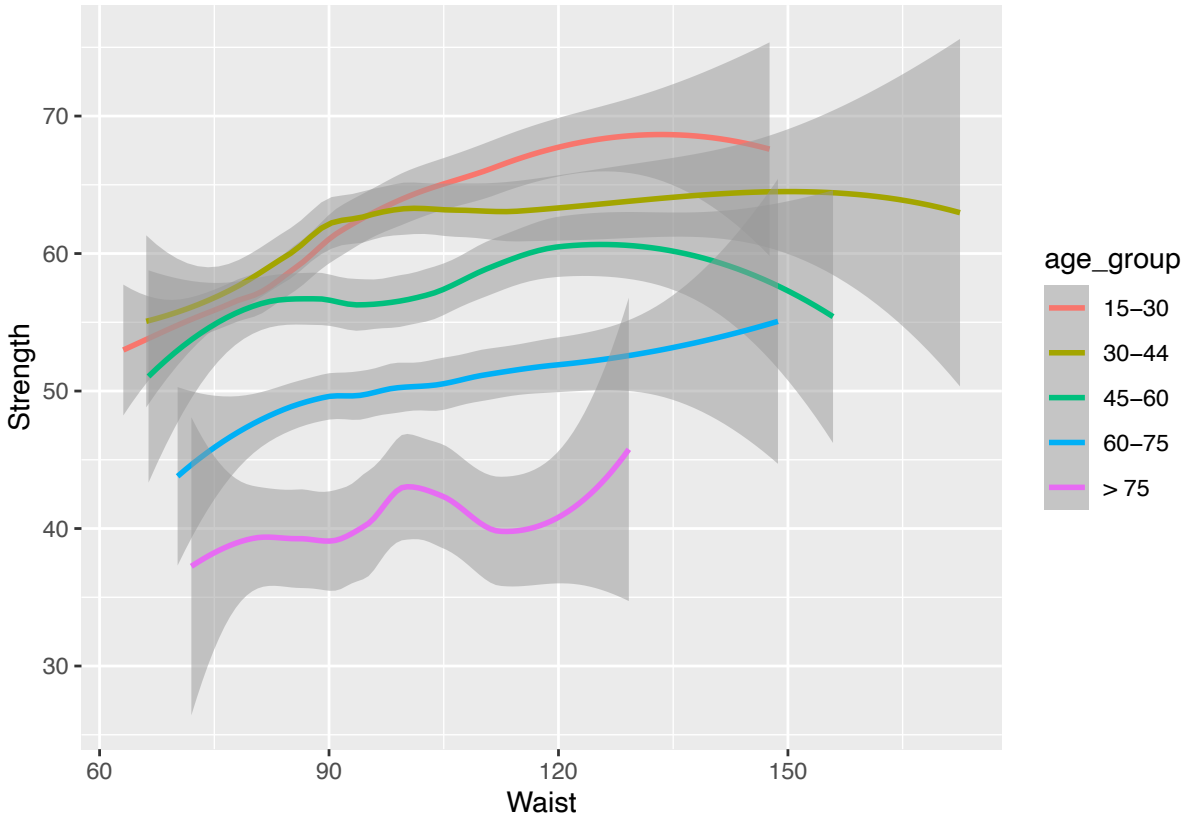
```
male <- total[total$Gender == 1, -4]
female <- total[total$Gender == 2, -4]

ggplot(data = male, aes(x = Waist, y = Strength)) +
    geom_smooth(alpha = 0.5, aes(color=age_group))
```

```
## `geom_smooth()` using method = 'loess' and formula 'y ~ x'
```

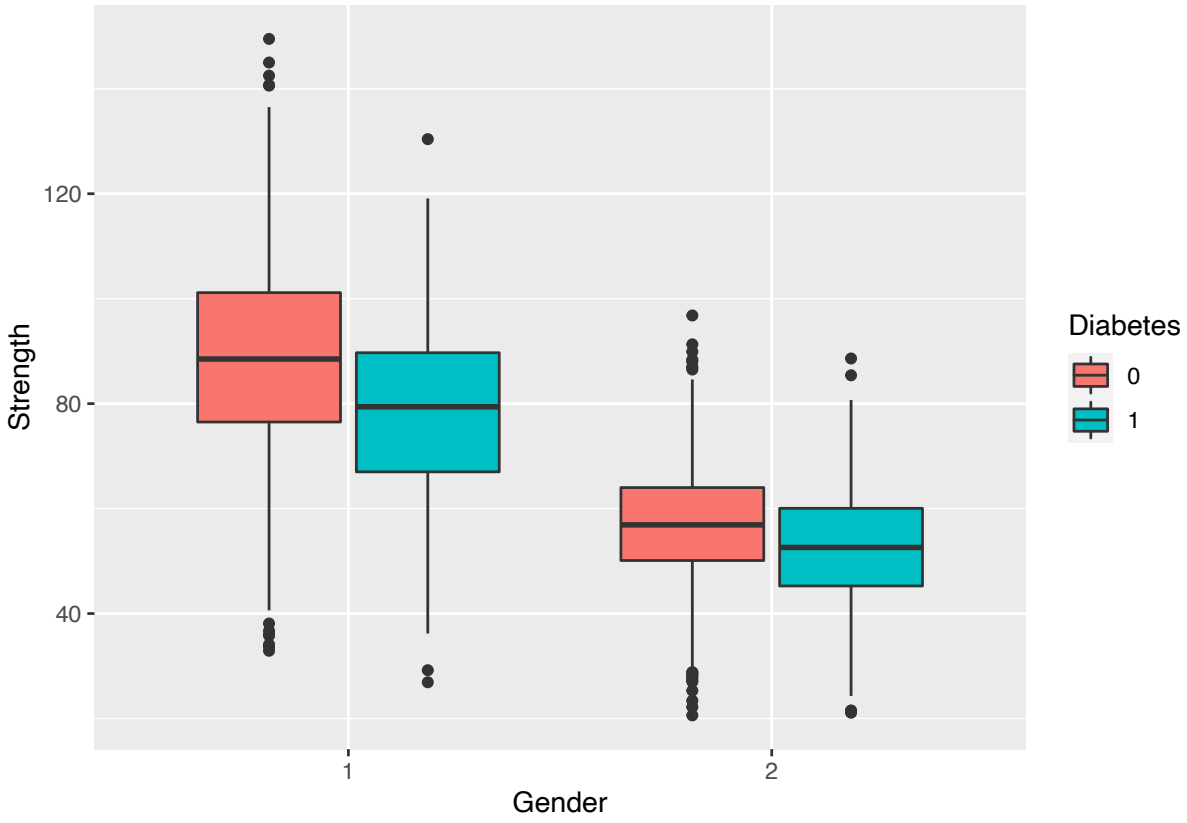



```
ggplot(data = female, aes(x = Waist, y = Strength)) +
    geom_smooth(alpha = 0.5, aes(color=age_group))
```

```
## `geom_smooth()` using method = 'loess' and formula 'y ~ x'
```

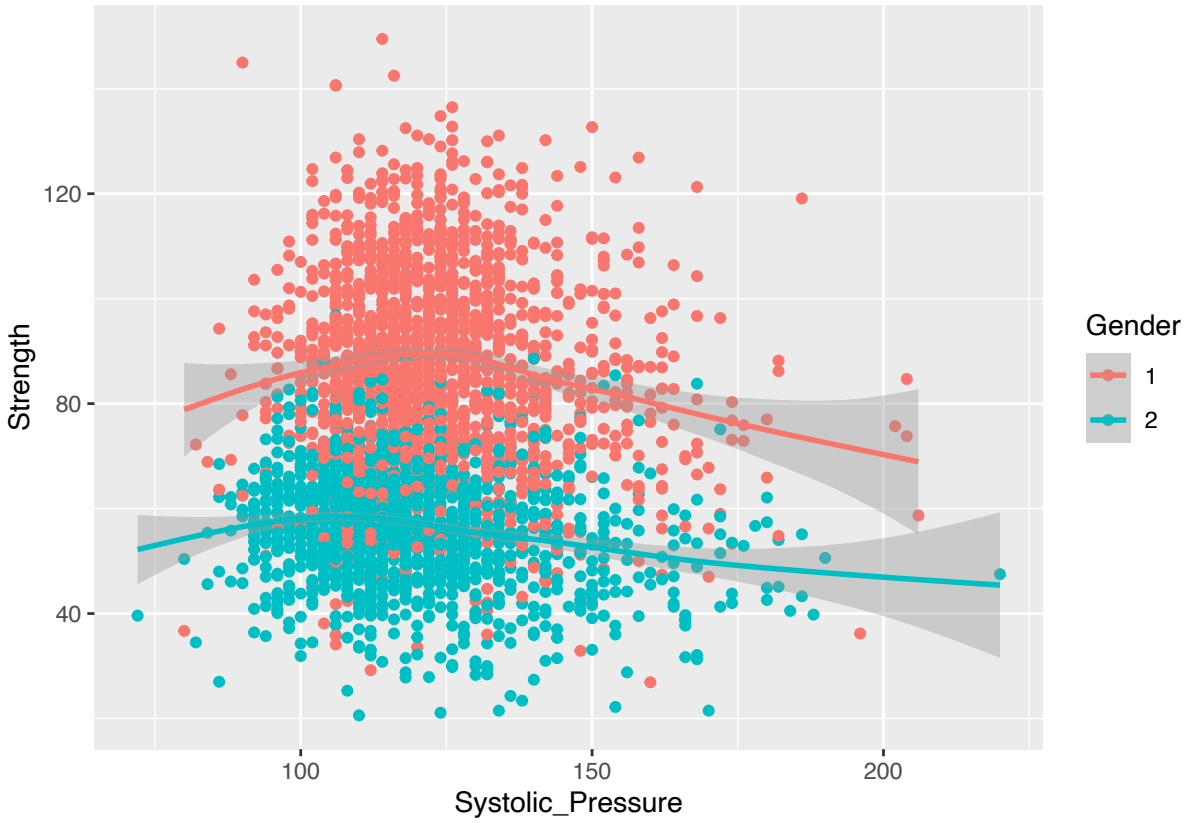



```
ggplot(data = total, aes(x = Gender, y = Strength)) +
    geom_boxplot(aes(fill=Diabetes))
```



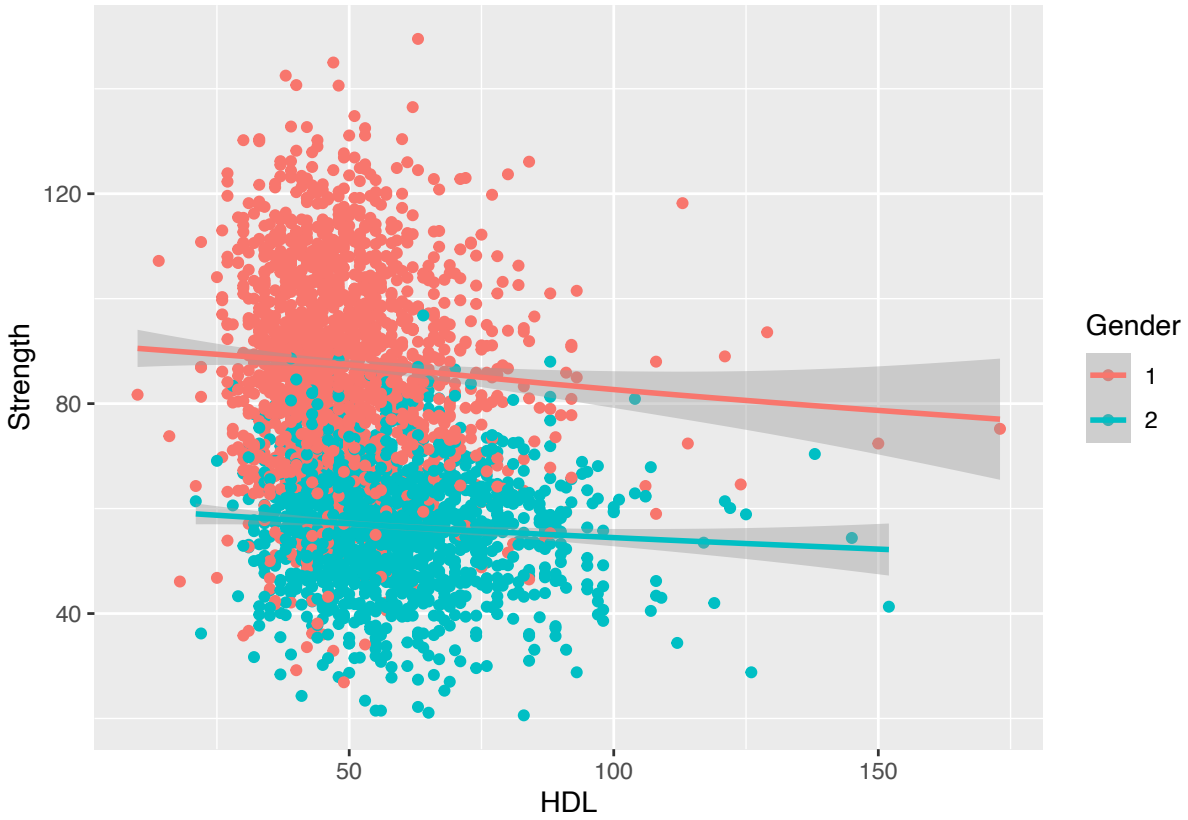

```
ggplot(data = total, aes(x = Systolic_Pressure, y = Strength)) +
    geom_point(aes(col=Gender)) + geom_smooth(aes(x = Systolic_Pressure, y = Strength, col=Gender))
```

```
## `geom_smooth()` using method = 'gam' and formula 'y ~ s(x, bs = "cs")'
```



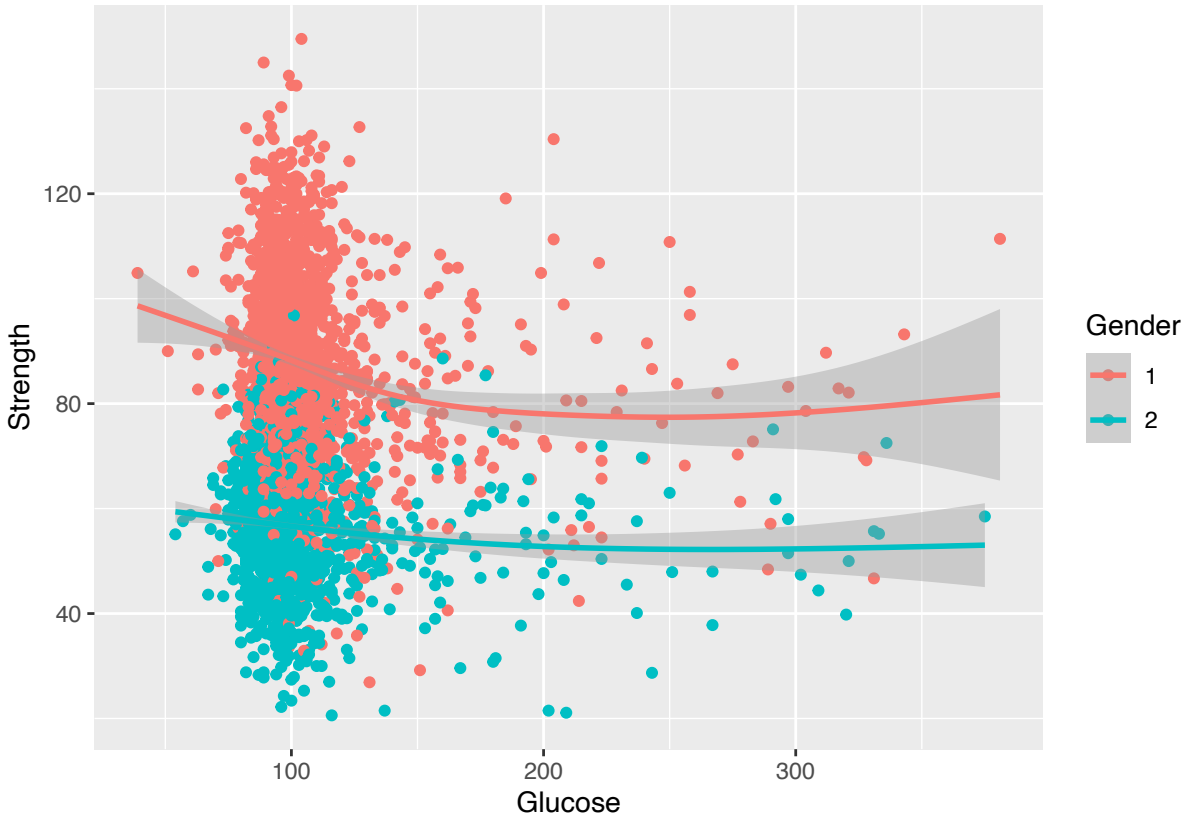

```
ggplot(data = total, aes(x = HDL, y = Strength)) +
    geom_point(aes(col=Gender)) + geom_smooth(aes(x = HDL, y = Strength, col=Gender))
```

```
## `geom_smooth()` using method = 'gam' and formula 'y ~ s(x, bs = "cs")'
```



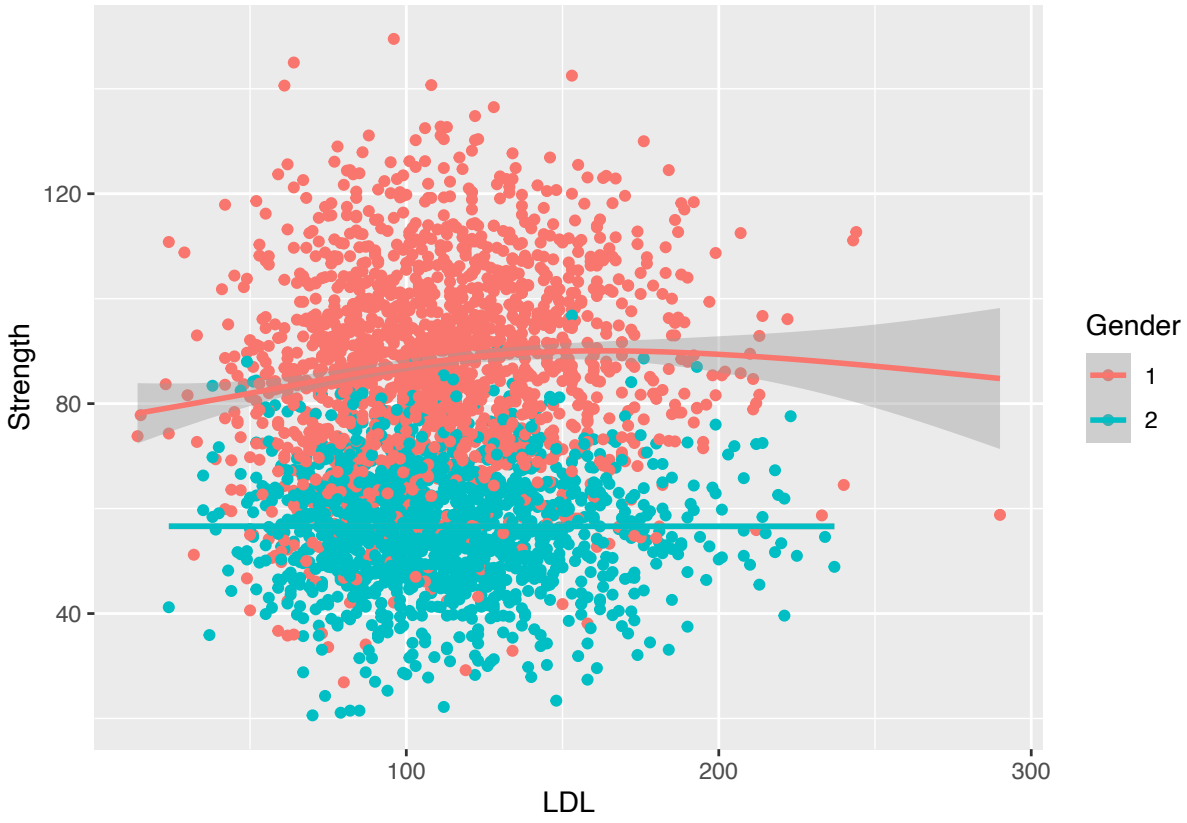

```
ggplot(data = total, aes(x = Glucose, y = Strength)) + geom_point(aes(col=Gender)) + geom_smooth(aes(x =
```

```
## `geom_smooth()` using method = 'gam' and formula 'y ~ s(x, bs = "cs")'
```



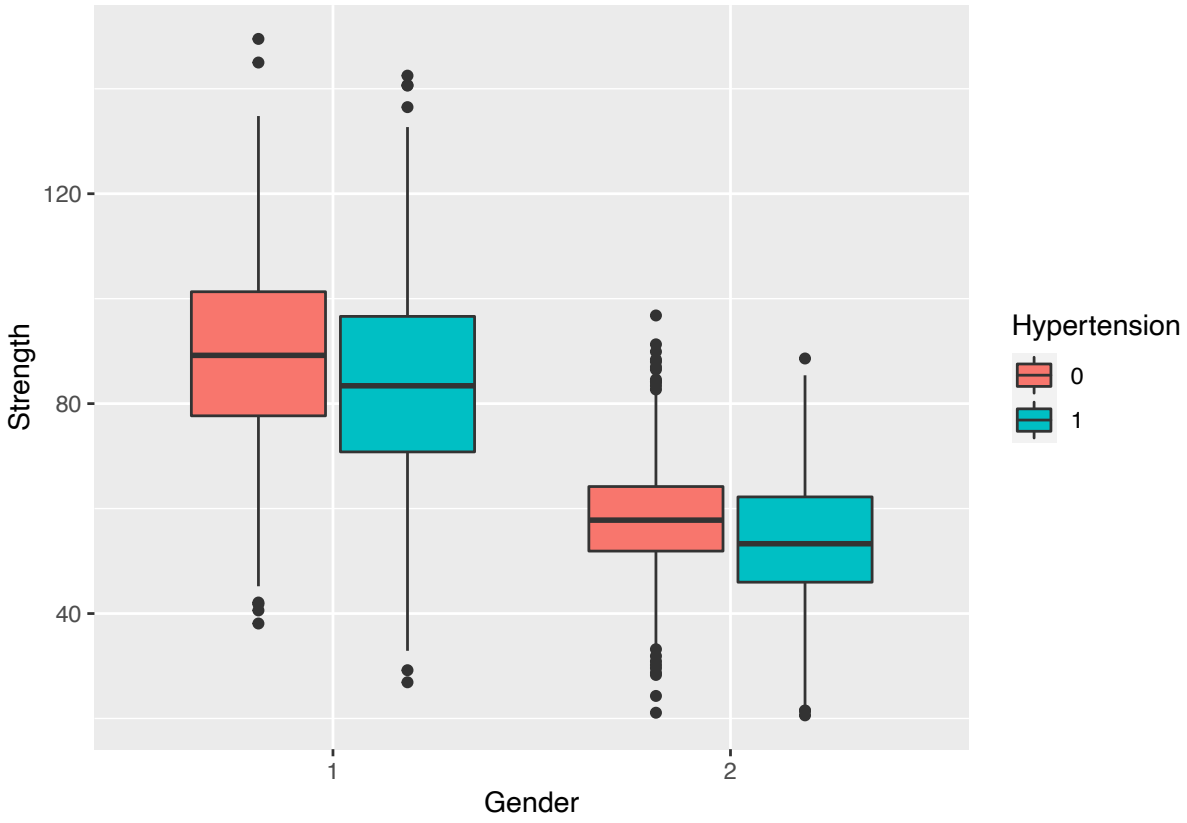

```
ggplot(data = total, aes(x = LDL, y = Strength)) + geom_point(aes(col=Gender)) + geom_smooth(aes(x = LDL
```

```
## `geom_smooth()` using method = 'gam' and formula 'y ~ s(x, bs = "cs")'
```

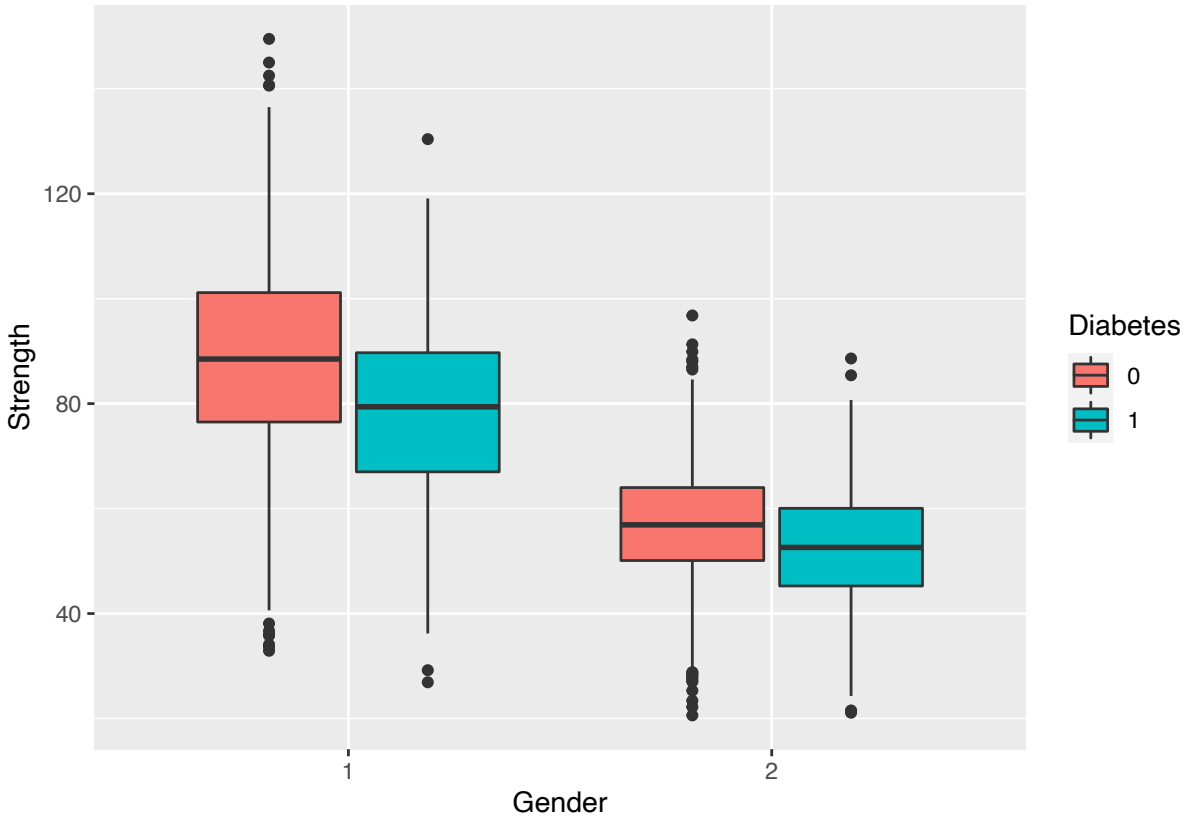


Strength vs. Diseases

```
ggplot(data = total, aes(x = Gender, y = Strength)) +
    geom_boxplot(aes(fill=Hypertension))
```




```
ggplot(data = total, aes(x = Gender, y = Strength)) +
    geom_boxplot(aes(fill=Diabetes))
```



Model 1: Linear Model

```
set.seed(415)

# Dividing age into two groups: Old and Young.
# This is for fit 2 which is not included in the paper due to its poor performance
male$age_group <- as.factor(ifelse(male$Age < 65, 0, 1))
colnames(male)[ncol(male)] <- "Old"

female$age_group <- as.factor(ifelse(female$Age < 65, 0, 1))
colnames(female)[ncol(female)] <- "Old"

# Train test split
id_male <- sample(1:nrow(male), 0.8*nrow(male), replace=FALSE)
train_male <- male[id_male, ]
test_male <- male[-id_male, ]

id_female <- sample(1:nrow(female), 0.8*nrow(female), replace=FALSE)
train_female <- female[id_female, ]
test_female <- female[-id_female, ]
```

```
# Baseline Linear Model
fit1_m <- glm(Strength ~ .-Old, data=train_male)
summary(fit1_m)
```

```
##
```

```
## Call:
## glm(formula = Strength ~ . - Old, data = train_male)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -59.627  -10.061    0.237    9.539   57.990
##
## Coefficients:
##                    Estimate Std. Error t value Pr(>|t|)
## (Intercept)        74.337542   4.983769  14.916  < 2e-16 ***
## Age                -0.522390   0.027659 -18.887  < 2e-16 ***
## Race2              -1.175787   1.827444  -0.643 0.520074
## Race3               4.823920   1.344742   3.587 0.000346 ***
## Race4              10.724493   1.555147   6.896 8.29e-12 ***
## Race6              -4.670971   1.811052  -2.579 0.010013 *
## Race7              -0.650545   2.702847  -0.241 0.809834
## Poverty             1.243602   0.256280   4.853 1.37e-06 ***
## Systolic_Pressure  -0.050483   0.029779  -1.695 0.090268 .
## Diastolic_Pressure  0.184354   0.037836   4.872 1.24e-06 ***
## Triglyceride        0.004162   0.006876   0.605 0.545150
## LDL                 0.041944   0.012142   3.454 0.000569 ***
## HDL                -0.027699   0.032549  -0.851 0.394928
## Glucose            -0.011049   0.016378  -0.675 0.500036
## OW1                -2.691991   1.105606  -2.435 0.015030 *
## Waist               0.223028   0.035296   6.319 3.60e-10 ***
## Smoking1           -0.069942   0.932779  -0.075 0.940240
## Alcohol            -0.004838   0.022785  -0.212 0.831885
## Hypertension1       0.549589   1.016777   0.541 0.588930
## Diabetes1          -3.116924   1.615003  -1.930 0.053825 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for gaussian family taken to be 215.0065)
##
##     Null deviance: 435946  on 1330  degrees of freedom
## Residual deviance: 281874  on 1311  degrees of freedom
## AIC: 10947
##
## Number of Fisher Scoring iterations: 2
```

```
fit1_f <- glm(Strength ~ .-Old, data=train_female)
summary(fit1_f)
```

```
##
## Call:
## glm(formula = Strength ~ . - Old, data = train_female)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -29.617   -5.719    0.072    5.716   37.374
##
## Coefficients:
##                    Estimate Std. Error t value Pr(>|t|)
## (Intercept)        40.909461   3.374345  12.124  < 2e-16 ***
## Age                -0.328802   0.021169 -15.532  < 2e-16 ***
```

```
## Race2               -0.588262   1.262300  -0.466  0.64129
## Race3                2.385857   0.991937   2.405  0.01632 *
## Race4                7.206025   1.081931   6.660 4.25e-11 ***
## Race6               -1.919198   1.313328  -1.461  0.14420
## Race7                3.957805   2.086475   1.897  0.05810 .
## Poverty              1.077912   0.177977   6.056 1.89e-09 ***
## Systolic_Pressure    0.009332   0.019936   0.468  0.63981
## Diastolic_Pressure   0.080635   0.027507   2.931  0.00344 **
## Triglyceride         0.005797   0.005340   1.086  0.27791
## LDL                  0.013600   0.008313   1.636  0.10211
## HDL                  0.029006   0.020892   1.388  0.16531
## Glucose             -0.003389   0.011867  -0.286  0.77523
## OW1                 -0.255329   0.697022  -0.366  0.71420
## Waist                0.168968   0.021646   7.806 1.34e-14 ***
## Smoking1             1.212439   0.695830   1.742  0.08170 .
## Alcohol             -0.105225   0.061807  -1.702  0.08894 .
## Hypertension1       -1.512919   0.697136  -2.170  0.03020 *
## Diabetes1           -1.800055   1.129945  -1.593  0.11143
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for gaussian family taken to be 82.52057)
##
##     Null deviance: 148333  on 1153  degrees of freedom
## Residual deviance:  93578  on 1134  degrees of freedom
## AIC: 8389.4
##
## Number of Fisher Scoring iterations: 2
```

```
# Fit 2: Adding interactions of Age with all other predictors
fit2_m <- glm(Strength ~ .-Age-Old+ Old:(.-Age-Old), data=train_male)
summary(fit2_m)
```

```
##
## Call:
## glm(formula = Strength ~ . - Age - Old + Old:(. - Age - Old),
##     data = train_male)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -57.917  -10.143   -0.229   10.123   50.158
##
## Coefficients:
##                           Estimate Std. Error t value Pr(>|t|)
## (Intercept)              83.837808   5.841424  14.352  < 2e-16 ***
## Race2                    -1.233612   2.088915  -0.591 0.554924
## Race3                     4.937823   1.502368   3.287 0.001041 **
## Race4                     9.859544   1.761661   5.597 2.66e-08 ***
## Race6                    -5.055715   1.981958  -2.551 0.010860 *
## Race7                     1.495588   2.958904   0.505 0.613327
## Poverty                   0.382778   0.289807   1.321 0.186800
## Systolic_Pressure        -0.101066   0.038759  -2.608 0.009225 **
## Diastolic_Pressure        0.094716   0.047488   1.995 0.046303 *
## Triglyceride             -0.005955   0.007654  -0.778 0.436720
## LDL                       0.001858   0.013973   0.133 0.894218
```

```
## HDL                       -0.087711   0.038358  -2.287 0.022379 *
## Glucose                   -0.028755   0.019617  -1.466 0.142929
## OW1                       -1.926849   1.298607  -1.484 0.138111
## Waist                      0.183161   0.039868   4.594 4.77e-06 ***
## Smoking1                   0.137078   1.036684   0.132 0.894825
## Alcohol                   -0.018063   0.028372  -0.637 0.524479
## Hypertension1             -0.621609   1.217473  -0.511 0.609737
## Diabetes1                 -7.918625   2.208597  -3.585 0.000349 ***
## Race1:Old1               -52.419525  13.777746  -3.805 0.000149 ***
## Race2:Old1               -49.984823  13.829114  -3.614 0.000313 ***
## Race3:Old1               -53.238190  13.527924  -3.935 8.75e-05 ***
## Race4:Old1               -48.637387  13.810921  -3.522 0.000444 ***
## Race6:Old1               -52.481036  14.131279  -3.714 0.000213 ***
## Race7:Old1               -53.644352  15.574300  -3.444 0.000591 ***
## Poverty:Old1               1.530917   0.709005   2.159 0.031014 *
## Systolic_Pressure:Old1     0.068588   0.065099   1.054 0.292271
## Diastolic_Pressure:Old1    0.118625   0.092686   1.280 0.200823
## Triglyceride:Old1          0.035237   0.020990   1.679 0.093445 .
## LDL:Old1                   0.058351   0.033131   1.761 0.078435 .
## HDL:Old1                   0.065191   0.080051   0.814 0.415583
## Glucose:Old1               0.014761   0.041687   0.354 0.723329
## OW1:Old1                   0.194789   2.754302   0.071 0.943630
## Waist:Old1                -0.034776   0.097189  -0.358 0.720537
## Smoking1:Old1             -1.893243   3.071823  -0.616 0.537788
## Alcohol:Old1               0.034540   0.051268   0.674 0.500615
## Hypertension1:Old1        -2.083018   2.376111  -0.877 0.380840
## Diabetes1:Old1             8.880335   3.457865   2.568 0.010336 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for gaussian family taken to be 228.7012)
##
##     Null deviance: 435946  on 1330  degrees of freedom
## Residual deviance: 295711  on 1293  degrees of freedom
## AIC: 11047
##
## Number of Fisher Scoring iterations: 2
```

```
fit2_f <- glm(Strength ~ .-Age-Old+ Old:(.-Age-Old), data=train_female)
summary(fit2_f)
```

```
##
## Call:
## glm(formula = Strength ~ . - Age - Old + Old:(. - Age - Old),
##     data = train_female)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -35.346   -5.763    0.106    5.831   33.889
##
## Coefficients:
##                           Estimate Std. Error t value Pr(>|t|)
## (Intercept)              41.575298   3.868892  10.746  < 2e-16 ***
## Race2                    -0.650557   1.407718  -0.462  0.64407
## Race3                     3.485483   1.100015   3.169  0.00157 **
```

```
## Race4                      8.321486   1.186283   7.015 3.99e-12 ***
## Race6                     -1.181405   1.433433  -0.824  0.41001
## Race7                      6.434643   2.258884   2.849  0.00447 **
## Poverty                    0.513493   0.200205   2.565  0.01045 *
## Systolic_Pressure         -0.034983   0.025169  -1.390  0.16482
## Diastolic_Pressure         0.067378   0.032815   2.053  0.04028 *
## Triglyceride               0.001175   0.006216   0.189  0.85012
## LDL                       -0.017064   0.009274  -1.840  0.06602 .
## HDL                        0.015639   0.024344   0.642  0.52072
## Glucose                   -0.011705   0.015208  -0.770  0.44164
## OW1                       -0.926862   0.794992  -1.166  0.24391
## Waist                      0.164826   0.024376   6.762 2.19e-11 ***
## Smoking1                   0.355125   0.773055   0.459  0.64605
## Alcohol                   -0.106810   0.066660  -1.602  0.10937
## Hypertension1             -3.235139   0.795226  -4.068 5.07e-05 ***
## Diabetes1                 -3.544205   1.532149  -2.313  0.02089 *
## Race1:Old1                -6.393399  10.882124  -0.588  0.55698
## Race2:Old1                -7.217657  10.161459  -0.710  0.47767
## Race3:Old1               -12.414011  10.179190  -1.220  0.22289
## Race4:Old1               -10.917767  10.544562  -1.035  0.30071
## Race6:Old1                -4.757083  10.196748  -0.467  0.64093
## Race7:Old1               -17.091523  12.793443  -1.336  0.18183
## Poverty:Old1               0.688100   0.518048   1.328  0.18437
## Systolic_Pressure:Old1     0.043163   0.044408   0.972  0.33127
## Diastolic_Pressure:Old1   -0.011863   0.073705  -0.161  0.87216
## Triglyceride:Old1          0.002234   0.014072   0.159  0.87391
## LDL:Old1                   0.022820   0.023419   0.974  0.33005
## HDL:Old1                  -0.083041   0.052260  -1.589  0.11235
## Glucose:Old1               0.015075   0.026487   0.569  0.56938
## OW1:Old1                   2.965498   1.943475   1.526  0.12732
## Waist:Old1                -0.076874   0.069074  -1.113  0.26598
## Smoking1:Old1              3.684702   2.389943   1.542  0.12342
## Alcohol:Old1               0.009835   0.243276   0.040  0.96776
## Hypertension1:Old1        -0.130374   1.729483  -0.075  0.93992
## Diabetes1:Old1             1.506545   2.495817   0.604  0.54621
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for gaussian family taken to be 87.83769)
##
##     Null deviance: 148333  on 1153  degrees of freedom
## Residual deviance:  98027  on 1116  degrees of freedom
## AIC: 8479
##
## Number of Fisher Scoring iterations: 2
```

```r
# Using best subsetting selection method to select significant predictors for male
library(leaps)
X_male <- model.matrix(fit1_m)[,-1]
regfit.full.m <- regsubsets(x=X_male, y=train_male$Strength, nvmax = 16, data=train_male)
reg.summary.m <- summary(regfit.full.m)

best_id.m <- which.min(reg.summary.m$bic)
reg.summary.m$which[best_id.m, ]
```

```
##        (Intercept)                 Age               Race2               Race3
##               TRUE                TRUE               FALSE                TRUE
##              Race4               Race6               Race7            Poverty
##               TRUE               FALSE               FALSE               TRUE
##  Systolic_Pressure Diastolic_Pressure       Triglyceride                 LDL
##              FALSE                TRUE               FALSE                TRUE
##                HDL             Glucose                 OW1               Waist
##              FALSE               FALSE               FALSE                TRUE
##           Smoking1             Alcohol      Hypertension1          Diabetes1
##              FALSE               FALSE               FALSE                TRUE
```

```
# Using best subsetting selection method to select significant predictors for female
X_female <- model.matrix(fit1_f)[,-1]
regfit.full.f <- regsubsets(x=X_female, y=train_female$Strength, nvmax = 20, data=train_female)
reg.summary.f <- summary(regfit.full.f)

best_id.f <- which.min(reg.summary.f$bic)
reg.summary.f$which[best_id.f, ]
```

```
##        (Intercept)                 Age               Race2               Race3
##               TRUE                TRUE               FALSE                TRUE
##              Race4               Race6               Race7            Poverty
##               TRUE               FALSE               FALSE               TRUE
##  Systolic_Pressure Diastolic_Pressure       Triglyceride                 LDL
##              FALSE                TRUE               FALSE               FALSE
##                HDL             Glucose                 OW1               Waist
##              FALSE               FALSE               FALSE                TRUE
##           Smoking1             Alcohol      Hypertension1          Diabetes1
##              FALSE               FALSE               FALSE               FALSE
```

```
# Observe the difference in predictors selected for male and female
coef <- as.data.frame(reg.summary.m$which[best_id.m, ])
coef <- cbind(coef, reg.summary.f$which[best_id.f, ])
colnames(coef) <- c("Male_Coefficients", "Female_Coefficients")

rownames(coef[coef$Male_Coefficients != coef$Female_Coefficients, ])
```

```
## [1] "LDL"       "Diabetes1"
```

```
# Fit the linear model using predictors selected by best subsetting selection method
fit3_m <- glm(Strength ~ Age + Race + Poverty + Diastolic_Pressure + LDL + Waist + Diabetes, data=train
summary(fit3_m)
```

```
##
## Call:
## glm(formula = Strength ~ Age + Race + Poverty + Diastolic_Pressure +
##     LDL + Waist + Diabetes, data = train_male)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -57.399  -10.145    0.308    9.832   57.794
##
## Coefficients:
##                    Estimate Std. Error t value Pr(>|t|)
## (Intercept)        70.62197    3.57864  19.734  < 2e-16 ***
## Age                -0.52994    0.02421 -21.886  < 2e-16 ***
```

```
## Race2              -1.17253    1.82561  -0.642 0.520813
## Race3               4.92287    1.33447   3.689 0.000234 ***
## Race4              10.36822    1.49909   6.916 7.20e-12 ***
## Race6              -4.56370    1.81229  -2.518 0.011914 *
## Race7              -0.70962    2.69608  -0.263 0.792433
## Poverty             1.24888    0.24883   5.019 5.90e-07 ***
## Diastolic_Pressure  0.16230    0.03489   4.651 3.63e-06 ***
## LDL                 0.04274    0.01195   3.577 0.000360 ***
## Waist               0.19002    0.02839   6.694 3.21e-11 ***
## Diabetes1          -4.06555    1.36656  -2.975 0.002983 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for gaussian family taken to be 215.4967)
##
##     Null deviance: 435946  on 1330  degrees of freedom
## Residual deviance: 284240  on 1319  degrees of freedom
## AIC: 10943
##
## Number of Fisher Scoring iterations: 2
```

```
fit3_f <- glm(Strength ~ Age + Race + Poverty + Diastolic_Pressure + Waist, data=train_female)
summary(fit3_f)
```

```
##
## Call:
## glm(formula = Strength ~ Age + Race + Poverty + Diastolic_Pressure +
##     Waist, data = train_female)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -29.754   -5.827    0.079    5.642   39.880
##
## Coefficients:
##                    Estimate Std. Error t value Pr(>|t|)
## (Intercept)        46.74342    2.38488  19.600  < 2e-16 ***
## Age                -0.34011    0.01592 -21.367  < 2e-16 ***
## Race2              -0.63118    1.25951  -0.501 0.616375
## Race3               2.47803    0.97658   2.537 0.011298 *
## Race4               6.91102    1.03926   6.650 4.53e-11 ***
## Race6              -2.11106    1.31503  -1.605 0.108696
## Race7               3.58953    2.06917   1.735 0.083051 .
## Poverty             1.07409    0.16867   6.368 2.77e-10 ***
## Diastolic_Pressure  0.08767    0.02478   3.539 0.000418 ***
## Waist               0.14864    0.01653   8.991  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for gaussian family taken to be 83.45139)
##
##     Null deviance: 148333  on 1153  degrees of freedom
## Residual deviance:  95468  on 1144  degrees of freedom
## AIC: 8392.5
##
## Number of Fisher Scoring iterations: 2
```

Measurement

```
library(boot)
# Fit 1 (baseline)

# Training Error
mean((predict(fit1_m, train_male) - train_male$Strength)^2)
```

```
## [1] 211.7758
```

```
mean((predict(fit1_f, train_female) - train_female$Strength)^2)
```

```
## [1] 81.09041
```

```
#Test Error
mean((predict(fit1_m, test_male) - test_male$Strength)^2)
```

```
## [1] 267.6512
```

```
mean((predict(fit1_f, test_female) - test_female$Strength)^2)
```

```
## [1] 90.5151
```

```
#Cross-Validation Error
set.seed(415)
cv.glm(train_male, fit1_m, K=10)$delta[1]
```

```
## [1] 218.3982
```

```
cv.glm(train_female, fit1_f, K=10)$delta[1]
```

```
## [1] 84.41793
```

```
#Fit 2 (with age as interaction terms)

# Training Error
mean((predict(fit2_m, train_male) - train_male$Strength)^2)
```

```
## [1] 222.1718
```

```
mean((predict(fit2_f, train_female) - train_female$Strength)^2)
```

```
## [1] 84.94529
```

```
# Test Error
mean((predict(fit2_m, test_male) - test_male$Strength)^2)
```

```
## [1] 282.2454
```

```
mean((predict(fit2_f, test_female) - test_female$Strength)^2)
```

```
## [1] 93.34603
```

```
# CV Error
set.seed(415)
cv.glm(train_male, fit2_m, K=10)$delta[1]
```

```
## [1] 236.5893
```

```
cv.glm(train_female, fit2_f, K=10)$delta[1]
```

```
## [1] 91.24769
```

```r
# Fit 3 (model of best subset)

# Training Error
mean((predict(fit3_m, train_male) - train_male$Strength)^2)
```

```
## [1] 213.5539
```

```r
mean((predict(fit3_f, train_female) - train_female$Strength)^2)
```

```
## [1] 82.72824
```

```r
# Test Error
mean((predict(fit3_m, test_male) - test_male$Strength)^2)
```

```
## [1] 266.7771
```

```r
mean((predict(fit3_f, test_female) - test_female$Strength)^2)
```

```
## [1] 89.57061
```

```r
# CV Error
set.seed(415)
cv.glm(train_male, fit3_m, K=10)$delta[1]
```

```
## [1] 216.782
```

```r
cv.glm(train_female, fit3_f, K=10)$delta[1]
```

```
## [1] 84.47692
```

```r
# BIC score for all three models
BIC(fit1_m, fit2_m, fit3_m)
```

```
##        df      BIC
## fit1_m 21 11056.49
## fit2_m 39 11249.76
## fit3_m 13 11010.07
```

```r
BIC(fit1_f, fit2_f, fit3_f)
```

```
##        df      BIC
## fit1_f 21 8495.463
## fit2_f 39 8675.975
## fit3_f 11 8448.028
```

Model 2

```r
# Model 2: GAM
library(splines)

mod2_m <- glm(Strength ~ ns(Age) + ns(Systolic_Pressure) + ns(Diastolic_Pressure) + Triglyceride + poly

summary(mod2_m)
```

```
##
## Call:
## glm(formula = Strength ~ ns(Age) + ns(Systolic_Pressure) + ns(Diastolic_Pressure) +
##     Triglyceride + poly(LDL, 2) + HDL + poly(Glucose, 2) + log(Alcohol +
```

```
##     1) + poly(Waist, 2) + Race + Poverty + OW + Smoking + Hypertension +
##     Diabetes, data = train_male)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -60.722   -9.664    0.363    9.874   56.771
##
## Coefficients:
##                           Estimate Std. Error t value Pr(>|t|)
## (Intercept)             89.365708   3.030169  29.492  < 2e-16 ***
## ns(Age)                -42.348285   2.239186 -18.912  < 2e-16 ***
## ns(Systolic_Pressure)   -7.693349   4.655325  -1.653 0.098654 .
## ns(Diastolic_Pressure)  22.414871   4.885996   4.588 4.91e-06 ***
## Triglyceride             0.004018   0.006881   0.584 0.559353
## poly(LDL, 2)1           48.798356  15.668562   3.114 0.001883 **
## poly(LDL, 2)2          -41.801371  14.842646  -2.816 0.004931 **
## HDL                     -0.017389   0.032860  -0.529 0.596776
## poly(Glucose, 2)1       -8.578412  18.781282  -0.457 0.647924
## poly(Glucose, 2)2      -19.964220  15.842723  -1.260 0.207840
## log(Alcohol + 1)         0.238980   0.527061   0.453 0.650322
## poly(Waist, 2)1        133.353420  20.724472   6.435 1.73e-10 ***
## poly(Waist, 2)2        -52.032898  15.755366  -3.303 0.000984 ***
## Race2                   -1.350551   1.818530  -0.743 0.457820
## Race3                    4.986660   1.337566   3.728 0.000201 ***
## Race4                   11.184826   1.550965   7.212 9.33e-13 ***
## Race6                   -4.278936   1.811856  -2.362 0.018341 *
## Race7                   -0.219666   2.690978  -0.082 0.934953
## Poverty                  1.115074   0.257887   4.324 1.65e-05 ***
## OW1                     -2.607810   1.103458  -2.363 0.018258 *
## Smoking1                 0.012515   0.928031   0.013 0.989242
## Hypertension1            0.398246   1.013121   0.393 0.694318
## Diabetes1               -2.666915   1.622873  -1.643 0.100555
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for gaussian family taken to be 212.1452)
##
##     Null deviance: 435946  on 1330  degrees of freedom
## Residual deviance: 277486  on 1308  degrees of freedom
## AIC: 10933
##
## Number of Fisher Scoring iterations: 2
```

```
mod2_f <- glm(Strength ~ ns(Age) + ns(Systolic_Pressure) + ns(Diastolic_Pressure) + Triglyceride + poly

summary(mod2_f)
```

```
##
## Call:
## glm(formula = Strength ~ ns(Age) + ns(Systolic_Pressure) + ns(Diastolic_Pressure) +
##     Triglyceride + poly(LDL, 2) + HDL + poly(Glucose, 2) + log(Alcohol +
##     1) + poly(Waist, 2) + Race + Poverty + OW + Smoking + Hypertension +
##     Diabetes, data = train_female)
##
## Deviance Residuals:
```

```
##     Min       1Q   Median       3Q      Max
## -29.772   -5.535    0.133    5.693   37.125
##
## Coefficients:
##                           Estimate Std. Error t value Pr(>|t|)
## (Intercept)             55.802275   1.907839  29.249  < 2e-16 ***
## ns(Age)                -26.877221   1.708527 -15.731  < 2e-16 ***
## ns(Systolic_Pressure)    1.413239   3.671993   0.385  0.70041
## ns(Diastolic_Pressure)   7.961194   2.670075   2.982  0.00293 **
## Triglyceride             0.004629   0.005342   0.866  0.38644
## poly(LDL, 2)1           11.513486   9.752326   1.181  0.23801
## poly(LDL, 2)2           11.465648   9.162622   1.251  0.21107
## HDL                      0.039246   0.021090   1.861  0.06302 .
## poly(Glucose, 2)1        1.751552  12.145444   0.144  0.88536
## poly(Glucose, 2)2       -7.767056   9.863685  -0.787  0.43119
## log(Alcohol + 1)        -0.507588   0.387712  -1.309  0.19074
## poly(Waist, 2)1        107.692948  13.220912   8.146 9.90e-16 ***
## poly(Waist, 2)2        -32.336351   9.857738  -3.280  0.00107 **
## Race2                   -0.398502   1.259366  -0.316  0.75173
## Race3                    2.739693   0.995972   2.751  0.00604 **
## Race4                    7.314746   1.080273   6.771 2.05e-11 ***
## Race6                   -1.046535   1.339835  -0.781  0.43491
## Race7                    4.242770   2.088992   2.031  0.04249 *
## Poverty                  1.085083   0.177696   6.106 1.40e-09 ***
## OW1                     -0.570146   0.699364  -0.815  0.41511
## Smoking1                 1.136542   0.694101   1.637  0.10182
## Hypertension1           -1.360606   0.695735  -1.956  0.05075 .
## Diabetes1               -2.097573   1.143115  -1.835  0.06677 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for gaussian family taken to be 81.92036)
##
##     Null deviance: 148333  on 1153  degrees of freedom
## Residual deviance:  92652  on 1131  degrees of freedom
## AIC: 8383.9
##
## Number of Fisher Scoring iterations: 2
```

```
# Training Errors
mean((predict(mod2_m, train_male) - train_male$Strength)^2)
```

```
## [1] 208.4793
```

```
mean((predict(mod2_f, train_female) - train_female$Strength)^2)
```

```
## [1] 80.28763
```

```
# Test Errors
mean((predict(mod2_m, test_male) - test_male$Strength)^2)
```

```
## [1] 264.0197
```

```
mean((predict(mod2_f, test_female) - test_female$Strength)^2)
```

```
## [1] 87.31337
```

```r
# CV Errors
set.seed(415)
cv.glm(train_male, mod2_m, K=10)$delta[1]
```

```
## [1] 217.1135
```

```r
cv.glm(train_female, mod2_f, K=10)$delta[1]
```

```
## [1] 84.20277
```

```r
# Reducing variance by using ridge for men
library(glmnet)
```

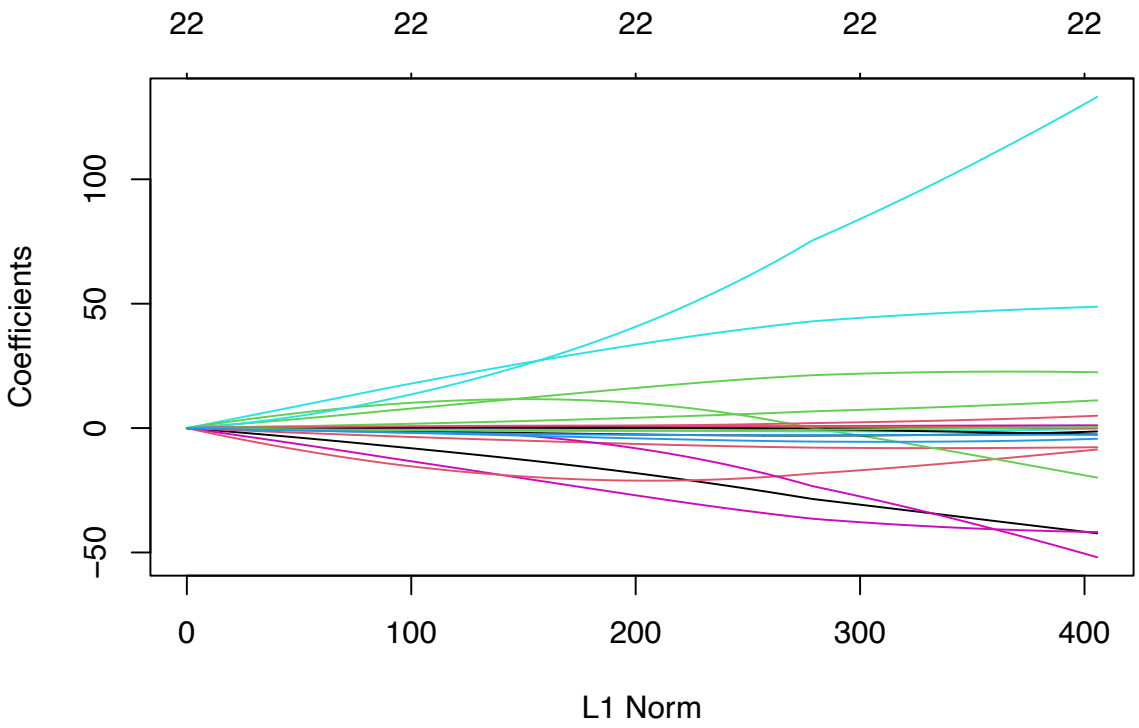
```
## Loading required package: Matrix
```

```
##
## Attaching package: 'Matrix'
```

```
## The following objects are masked from 'package:tidyr':
##
##     expand, pack, unpack
```

```
## Loaded glmnet 4.1-3
```

```r
set.seed(415)
grid <- 10^seq(10, -2, length = 1000)
x <- model.matrix(mod2_m)[, -1]
y <- train_male$Strength

ridge_m <- glmnet(x, y, alpha = 0, lambda = grid)

# Using CV to find the best lambda (hyperparameter)
cv.out <- cv.glmnet(x, y, alpha = 0)
plot(ridge_m)
```




```
bestlam_m <- cv.out$lambda.min

test_mat_m <- model.matrix(lm(Strength ~ ns(Age) + ns(Systolic_Pressure) + ns(Diastolic_Pressure) + Trig

# Training  Error
pred_ridge_m_train <- predict(ridge_m, s = bestlam_m, newx = x)
mean((pred_ridge_m_train  - train_male$Strength)^2)
```

```
## [1] 209.0445
```

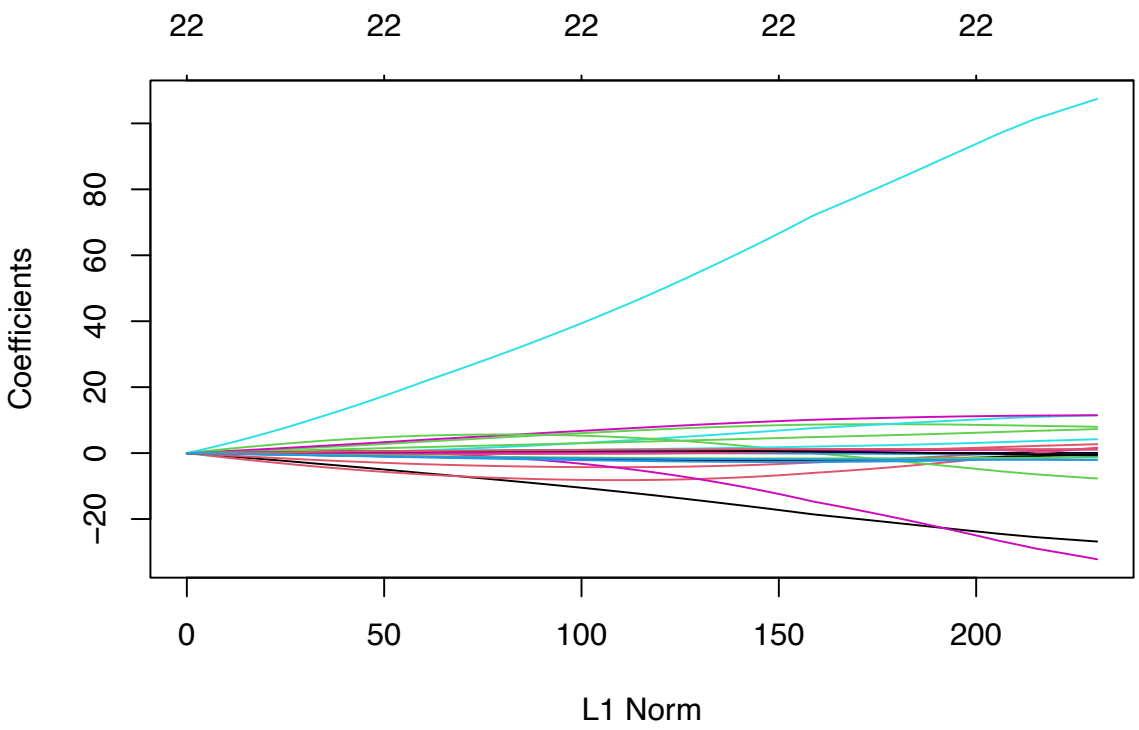
```
# Test Error
pred_ridge_m <- predict(ridge_m, s = bestlam_m, newx = test_mat_m)
mean((pred_ridge_m - test_male$Strength)^2) # Test MSE
```

```
## [1] 294.6308
```

```
# CV error
min(cv.out$cvm)
```

```
## [1] 217.3631
```

```
# Reducing variance by using ridge for men
set.seed(415)
grid <- 10^seq(10, -2, length = 1000)
x <- model.matrix(mod2_f)[, -1]
y <- train_female$Strength

ridge_f <- glmnet(x, y, alpha = 0, lambda = grid)

cv.out <- cv.glmnet(x, y, alpha = 0)
plot(ridge_f)
```




```
bestlam_f <- cv.out$lambda.min
test_mat_f <- model.matrix(lm(Strength ~ ns(Age) + ns(Systolic_Pressure) + ns(Diastolic_Pressure) + Trig

# Training  Error
pred_ridge_f_train <- predict(ridge_f, s = bestlam_f, newx = x)
mean((pred_ridge_f_train  - train_female$Strength)^2)
```

```
## [1] 80.55717
```

```
# Test  Error
pred_ridge_f_test <- predict(ridge_f, s = bestlam_f, newx = test_mat_f)
mean((pred_ridge_f_test - test_female$Strength)^2)
```

```
## [1] 99.18864
```

```
# CV error
min(cv.out$cvm)
```

```
## [1] 84.09853
```

```
# BIC scores for GAM
BIC(mod2_m)
```

```
## [1] 11057.19
```

```
BIC(mod2_f)
```

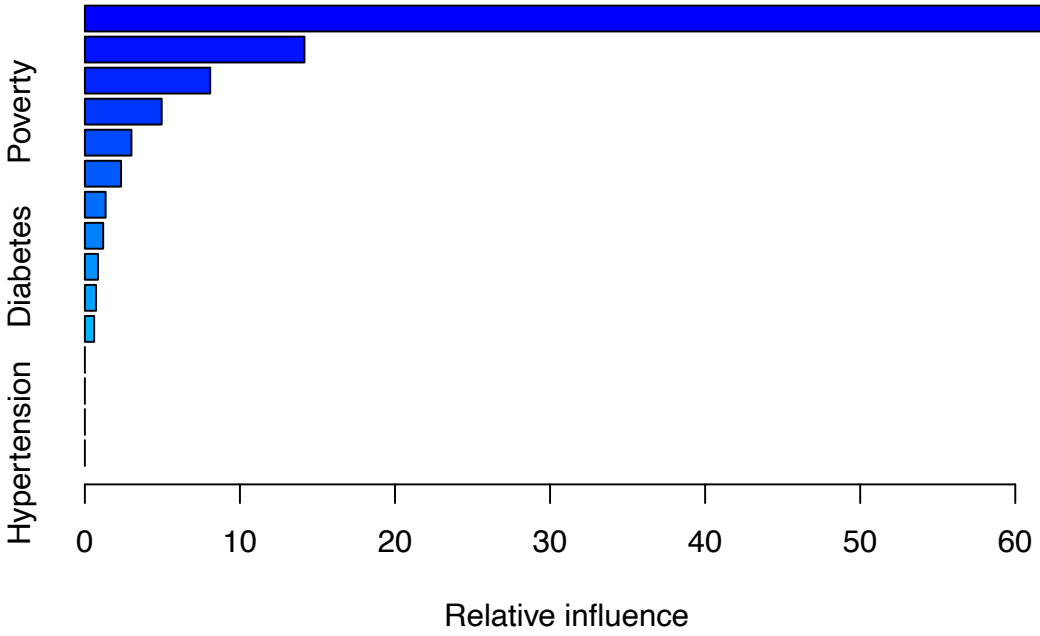
```
## [1] 8505.134
```

Model 3

```
# Generalized boosted regression model for men
library(gbm)
```

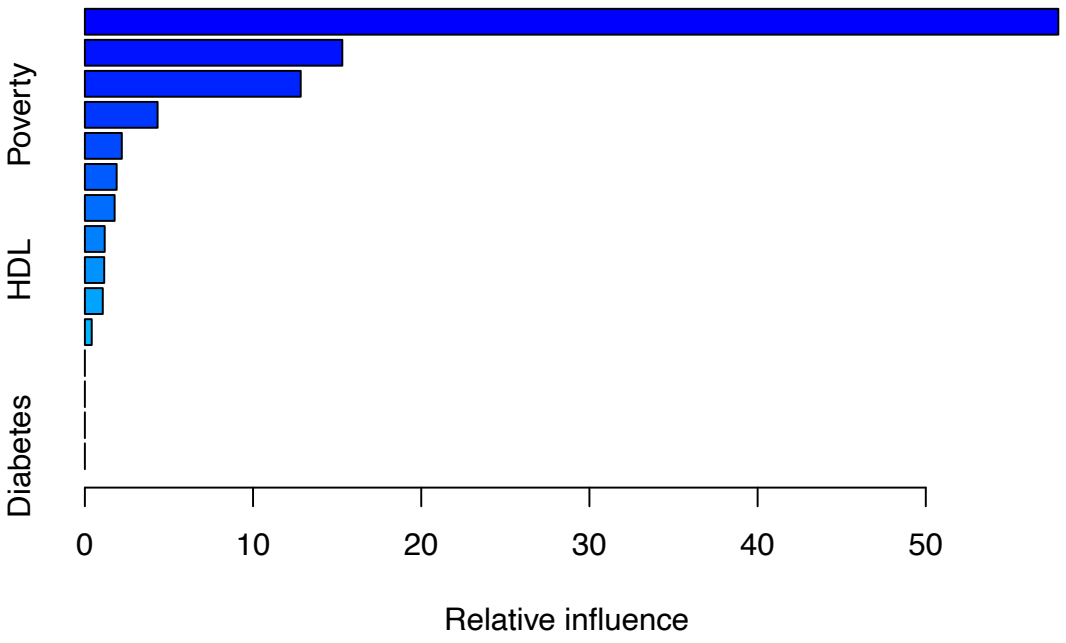
```
## Loaded gbm 2.1.8
```

```
set.seed(415)
boost_male <- gbm(Strength ~ .-Old, data = train_male, distribution = "gaussian", n.trees = 30, interac
summary(boost_male)
```




```
##                                   var    rel.inf
## Age                               Age 62.7765940
## Race                             Race 14.1609818
## Waist                           Waist  8.0835591
## Poverty                       Poverty  4.9545726
## Diastolic_Pressure Diastolic_Pressure  3.0026398
## LDL                               LDL  2.3345397
## HDL                               HDL  1.3398981
## Triglyceride             Triglyceride  1.1811142
## Diabetes                     Diabetes  0.8475736
## Glucose                       Glucose  0.7195217
## Systolic_Pressure   Systolic_Pressure  0.5990054
## OW                                 OW  0.0000000
## Smoking                       Smoking  0.0000000
## Alcohol                       Alcohol  0.0000000
## Hypertension             Hypertension  0.0000000
```

```
# Generalized boosted regression model for women
set.seed(415)
boost_female <- gbm(Strength ~ .-Old, data = train_female, distribution = "gaussian", n.trees = 30, int
summary(boost_female)
```




```
##                                    var    rel.inf
## Age                                Age 57.8796930
## Race                              Race 15.3082669
## Waist                            Waist 12.8325973
## Poverty                        Poverty  4.3239565
## Diastolic_Pressure Diastolic_Pressure  2.1961557
## Glucose                        Glucose  1.8889816
## LDL                                LDL  1.7686365
## Triglyceride              Triglyceride  1.1786894
## HDL                                HDL  1.1499307
## Systolic_Pressure    Systolic_Pressure  1.0652356
## Smoking                        Smoking  0.4078566
## OW                                  OW  0.0000000
## Alcohol                        Alcohol  0.0000000
## Hypertension              Hypertension  0.0000000
## Diabetes                      Diabetes  0.0000000
```

```
#Training Error
pred_boost_male <- predict(boost_male, newdata = train_male, n.trees = 30)
mean((pred_boost_male - train_male$Strength)^2)
```

```
## [1] 191.2473
```

```
pred_boost_female <- predict(boost_female, newdata = train_female, n.trees = 30)
mean((pred_boost_female - train_female$Strength)^2)
```

```
## [1] 71.87133
```

```
#Test Error
pred_boost_male <- predict(boost_male, newdata = test_male, n.trees = 30)
mean((pred_boost_male - test_male$Strength)^2)
```

```
## [1] 258.4416
```

```
pred_boost_female <- predict(boost_female, newdata = test_female, n.trees = 30)
mean((pred_boost_female - test_female$Strength)^2)
```

```
## [1] 83.67582
```

STATS415 Final Project Classification

Qingyang Liu, Ziwei Tian, Jialin Kou

2022-03-11

```
#install.packages("ROCR")
library(haven)
library(tidyverse)
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.1 --

## v ggplot2 3.3.5     v purrr   0.3.4
## v tibble  3.1.4     v dplyr   1.0.7
## v tidyr   1.1.3     v stringr 1.4.0
## v readr   2.0.1     v forcats 0.5.1

## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```
library(ggplot2)
library(MASS)
```

```
## Warning: package 'MASS' was built under R version 4.1.2

##
## Attaching package: 'MASS'

## The following object is masked from 'package:dplyr':
##
##     select
```

```
library(ROCR)
library(leaps)
library(randomForest)
```

```
## Warning: package 'randomForest' was built under R version 4.1.2

## randomForest 4.7-1

## Type rfNews() to see new features/changes/bug fixes.

##
## Attaching package: 'randomForest'
```

```
## The following object is masked from 'package:dplyr':
##
##     combine

## The following object is masked from 'package:ggplot2':
##
##     margin
```

```
library(FNN)
```

Data loading and processing

```
Waist_1112 <- read_xpt(file = "BodyMeasure1112.XPT") %>% dplyr::select(SEQN, BMXWAIST)
Wasit_1314 <- read_xpt(file = "BodyMeasure1314.XPT") %>% dplyr::select(SEQN, BMXWAIST)
Waist_df <- rbind(Waist_1112, Wasit_1314)
```

```
#write.csv(Waist_df, "Waist1114.csv", row.names = F)
```

```
BloodPressure_1314 <- read_xpt("bloodpressure1314.XPT")
BP_1314 <- BloodPressure_1314 %>% dplyr::select(SEQN, BPXSY2, BPXDI2)

BloodPressure_1112 <- read_xpt("BloodPressure1112.XPT")
BP_1112 <- BloodPressure_1112 %>% dplyr::select(SEQN, BPXSY2, BPXDI2)

BloodPressure <- rbind(BP_1314, BP_1112) %>% filter(BPXDI2 != 0)
```

```
#write.csv(BloodPressure, "BloodPressure1114.csv", row.names = F)
```

```
Hypertension <- read.csv("hypertension1114.csv")
```

```
Medical_1112 <- read_xpt("medical1112.XPT")
coronary_heart_1112 <- Medical_1112 %>% dplyr::select(SEQN, MCQ160C) %>%
  filter(MCQ160C %in% c(1,2)) %>% mutate(C_H = ifelse(MCQ160C == 1,1,0)) %>%
  dplyr::select(-MCQ160C)
overweight_1112 <- Medical_1112 %>% dplyr::select(SEQN, MCQ080) %>%
  filter(MCQ080 %in% c(1,2)) %>% mutate(OW = ifelse(MCQ080 == 1,1,0)) %>% dplyr::select(-MCQ080)

Medical_1516 <- read_xpt("medical1516.XPT")
coronary_heart_1516 <- Medical_1516 %>% dplyr::select(SEQN, MCQ160C) %>%
  filter(MCQ160C %in% c(1,2)) %>% mutate(C_H = ifelse(MCQ160C == 1,1,0)) %>%
  dplyr::select(-MCQ160C)
overweight_1516 <- Medical_1516 %>% dplyr::select(SEQN, MCQ080) %>%
  filter(MCQ080 %in% c(1,2)) %>% mutate(OW = ifelse(MCQ080 == 1,1,0)) %>% dplyr::select(-MCQ080)

Medical_1314 <- read_xpt("medical1314.XPT")
coronary_heart_1314 <- Medical_1314 %>% dplyr::select(SEQN, MCQ160C) %>%
  filter(MCQ160C %in% c(1,2)) %>% mutate(C_H = ifelse(MCQ160C == 1,1,0)) %>%
  dplyr::select(-MCQ160C)
overweight_1314 <- Medical_1314 %>% dplyr::select(SEQN, MCQ080) %>%
```

```r
  filter(MCQ080 %in% c(1,2)) %>% mutate(OW = ifelse(MCQ080 == 1,1,0)) %>% dplyr::select(-MCQ080)

coronary_heart <- rbind(coronary_heart_1112,coronary_heart_1314)
overweight <- rbind(overweight_1112, overweight_1314)
```

```r
#write.csv(overweight, "overweight1114.csv", row.names = F)
```

```r
diabetes_data <- read.csv("diabetes_07-18.csv")
demo_data <- read.csv("demo1114.csv")
```

```r
df <- left_join(diabetes_data, demo_data, by = 'SEQN', all.x = TRUE) %>% na.omit()
```

```r
diabetes_df <- df %>%
  mutate(
    # Create categories
    Age_group = dplyr::case_when(
      Age <= 14             ~ "0-14",
      Age > 14 & Age <= 30 ~ "15-30",
      Age > 30 & Age <= 44 ~ "30-44",
      Age > 44 & Age <= 60 ~ "45-60",
      Age > 60 & Age <= 75 ~ "60-75",
      Age > 75              ~ "> 75",
    ),
    # Convert to factor
    Age_group = factor(
      Age_group,
      level = c("0-14", "15-30","30-44","45-60", "60-75", "> 75")
    )
  )
```

```r
diab_df <- diabetes_df %>% mutate(diabetes = ifelse(DIQ010 == 1,1,0)) %>% dplyr::select(-DIQ010)
```

```r
Alcohol <- read.csv("alcohol1114.csv")
```

```r
Chol <- read.csv("cholesterol1114.csv")
```

```r
Glucose <- read.csv("glucose1114.csv")
```

```r
Smoke <- read.csv("smoking1114.csv")
Smoke_df <- Smoke %>% filter(Smoking %in% c(1,2)) %>% mutate(Smoking = ifelse(Smoking == 1, 1, 0))
```

```r
coronary_heart_1 <- left_join(coronary_heart, diab_df, by = 'SEQN', all.x = TRUE)
coronary_heart_2 <- left_join(coronary_heart_1, BloodPressure, by = 'SEQN', all.x = TRUE)
coronary_heart_3 <- left_join(coronary_heart_2, Chol, by = 'SEQN', all.x = TRUE)
coronary_heart_4 <- left_join(coronary_heart_3, Smoke_df, by = 'SEQN', all.x = TRUE)
coronary_heart_5 <- left_join(coronary_heart_4, Alcohol, by = 'SEQN', all.x = TRUE)
coronary_heart_6 <- left_join(coronary_heart_5, overweight, by = 'SEQN', all.x = TRUE)
coronary_heart_7 <- left_join(coronary_heart_6, Waist_df, by = 'SEQN', all.x = TRUE)
coronary_heart_8 <- left_join(coronary_heart_7, Hypertension, by = 'SEQN', all.x = TRUE)
coronary_heart_9 <- left_join(coronary_heart_8, Glucose, by = 'SEQN', all.x = TRUE) %>%
```

```
  mutate(Race = as.factor(Race), Gender = as.factor(Gender), diabetes = as.factor(diabetes),
         Smoking = as.factor(Smoking), OW = as.factor(OW),
         hypertension = as.factor(hypertension)) %>% dplyr::select(-X) %>% na.omit()
```

```
# write.csv(coronary_heart_9, "ClassificationData.csv", row.names = F)
```

Exploratory Data Analysis

```
# ratio of geting CH in each age group
coronary_heart_9 %>% group_by(Age_group) %>% summarize(n = n(), CH = sum(C_H), ratio = CH/n)
```

```
## # A tibble: 5 x 4
##   Age_group     n    CH   ratio
##   <fct>     <int> <dbl>   <dbl>
## 1 15-30       641     0 0
## 2 30-44       805     6 0.00745
## 3 45-60       834    15 0.0180
## 4 60-75       643    50 0.0778
## 5 > 75        245    38 0.155
```

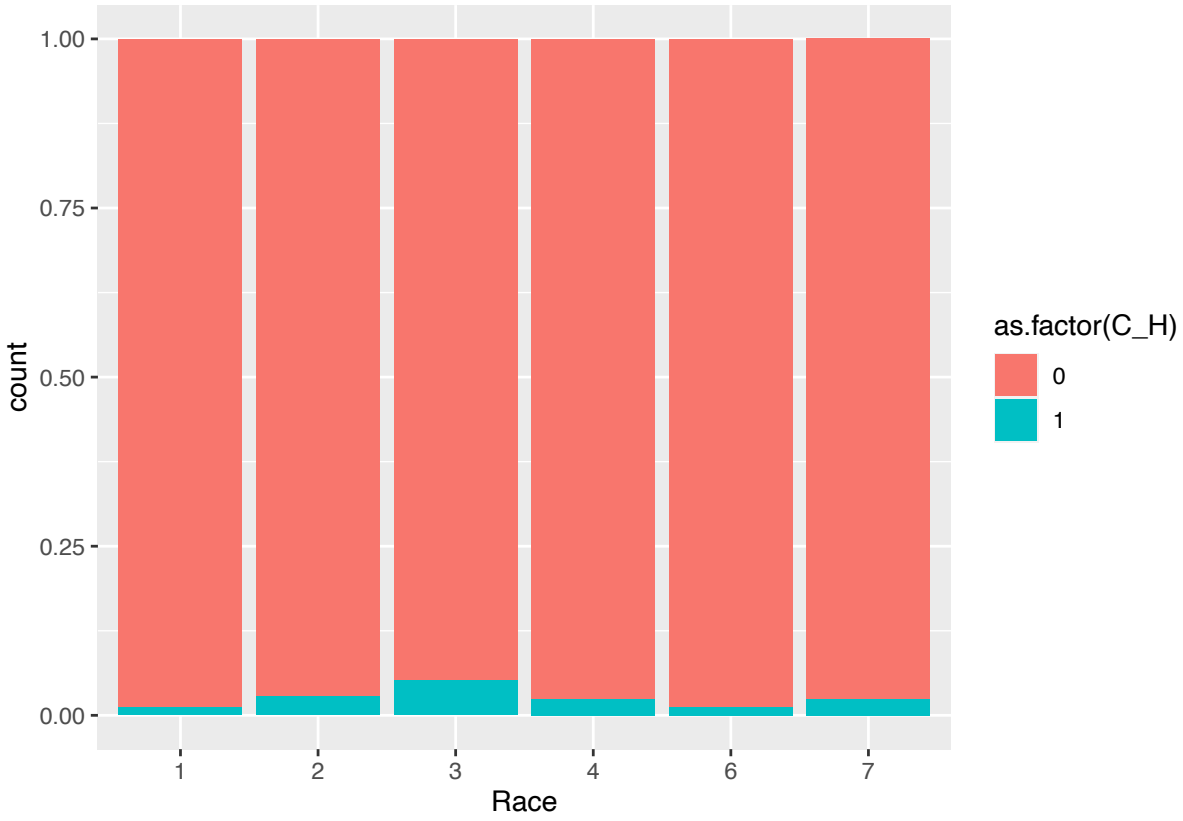
```
# ratio of getting CH conditional on whether the individual also got diabetes
coronary_heart_9 %>% group_by(diabetes) %>% summarize(n=n(), CH = sum(C_H), ratio = CH/n)
```

```
## # A tibble: 2 x 4
##   diabetes     n    CH  ratio
##   <fct>    <int> <dbl>  <dbl>
## 1 0         2791    72 0.0258
## 2 1          377    37 0.0981
```

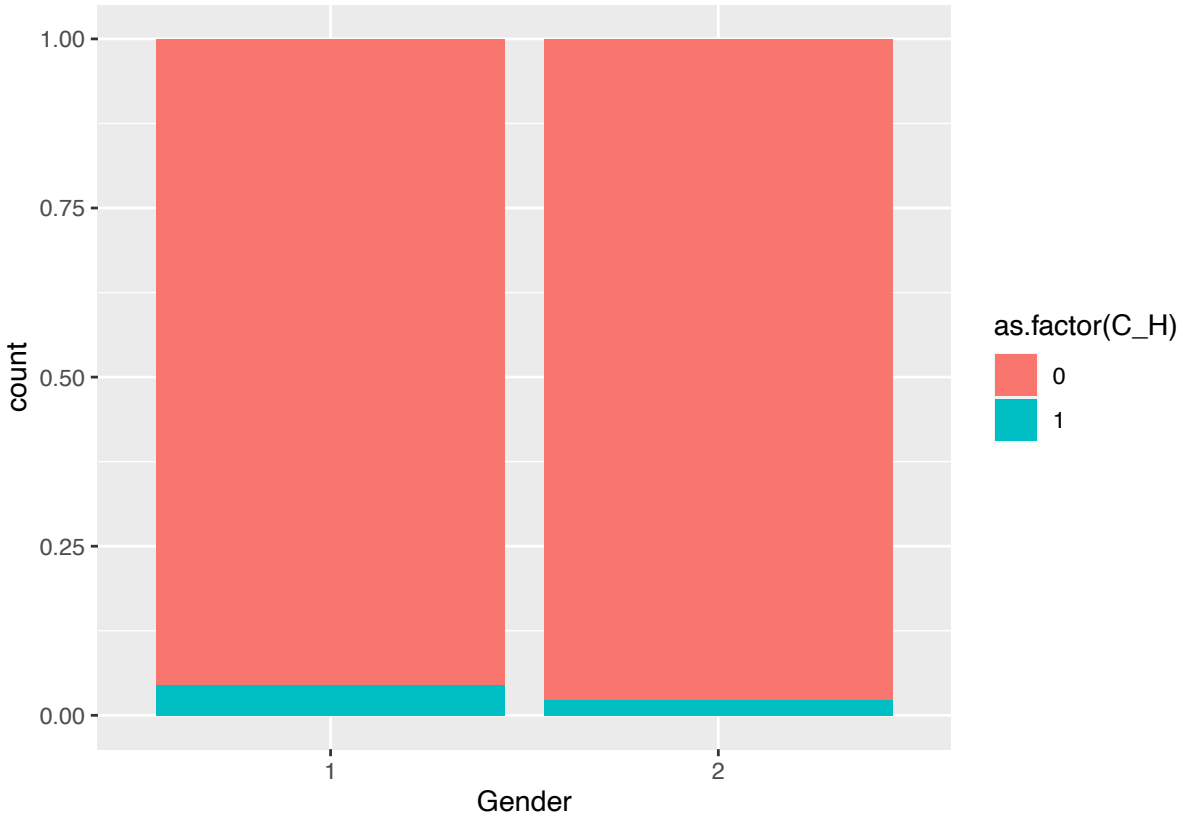
```
# ratio of getting CH conditional on each race category
coronary_heart_9 %>% group_by(Race) %>% summarize(n=n(), CH = sum(C_H), ratio = CH/n)
```

```
## # A tibble: 6 x 4
##   Race      n    CH  ratio
##   <fct> <int> <dbl>  <dbl>
## 1 1       349     4 0.0115
## 2 2       290     8 0.0276
## 3 3      1455    75 0.0515
## 4 4       675    16 0.0237
## 5 6       316     4 0.0127
## 6 7        83     2 0.0241
```

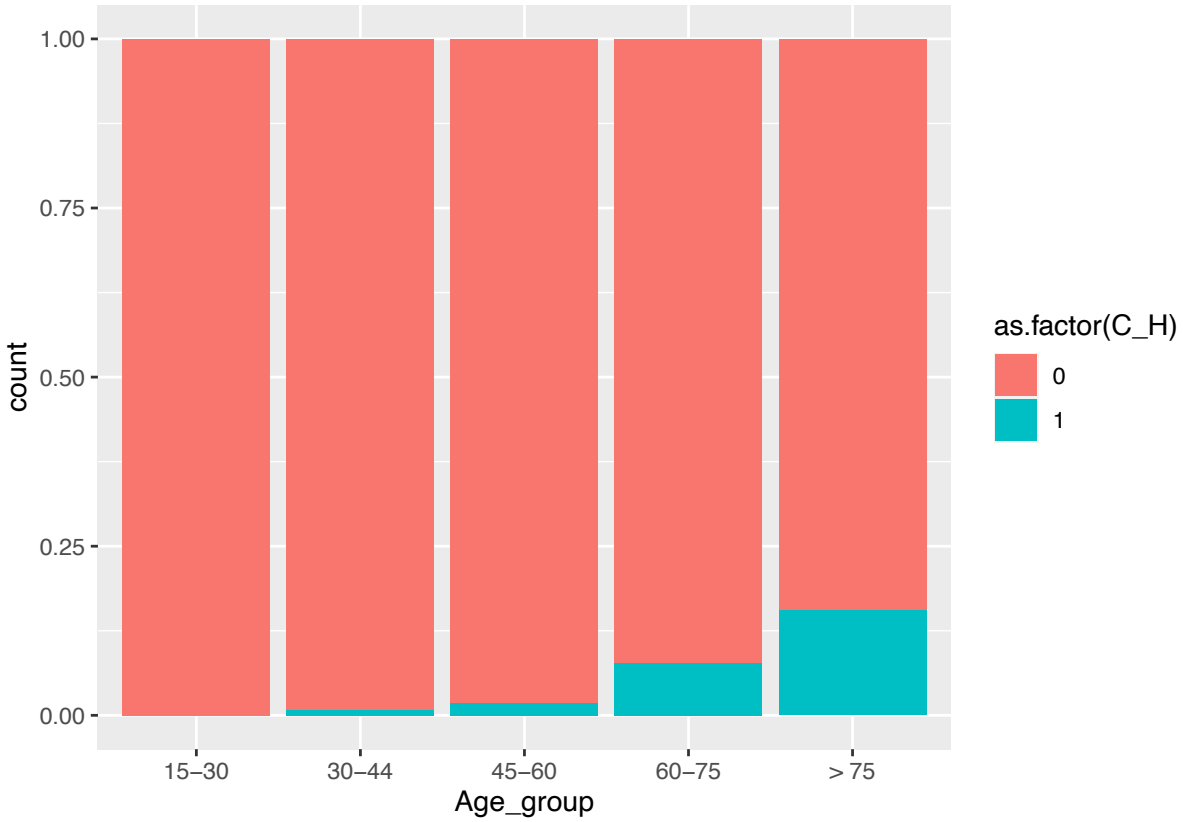
```
ggplot(coronary_heart_9) + geom_bar(aes(x = Race, fill = as.factor(C_H)), position = "fill")
```



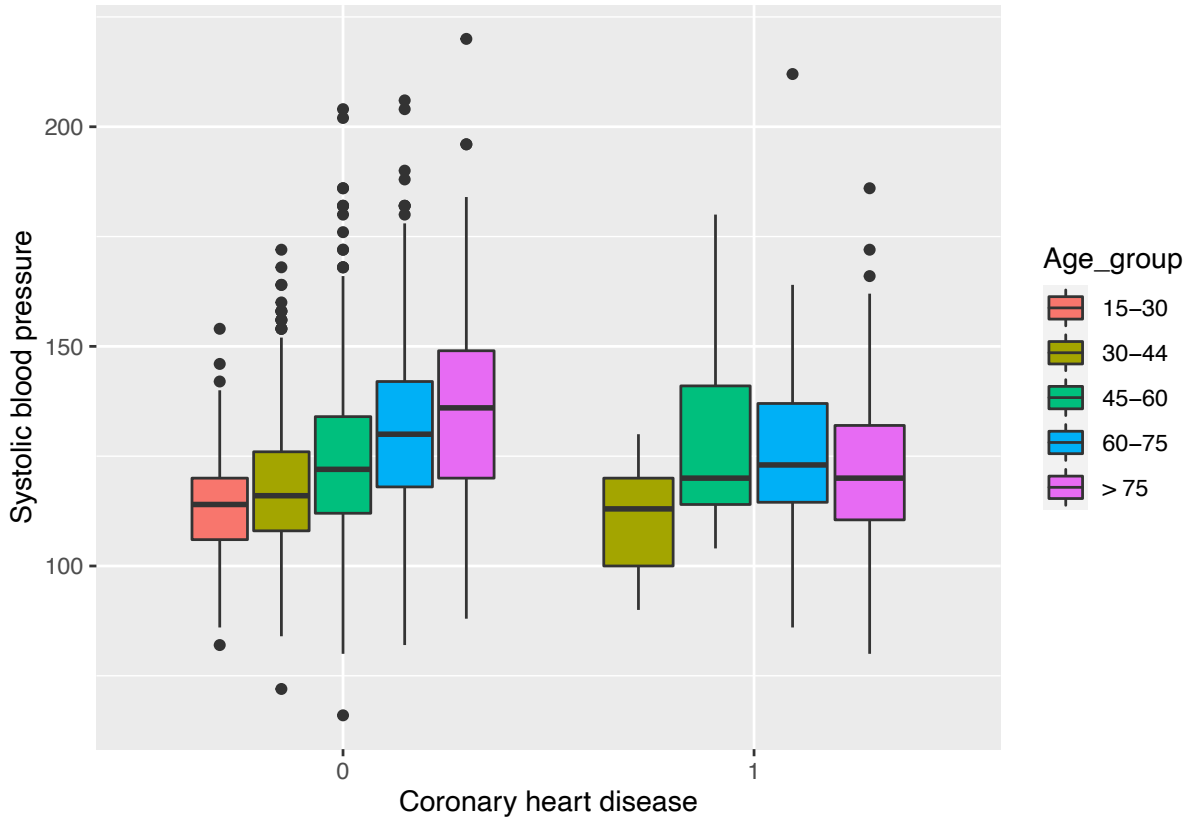

```
ggplot(coronary_heart_9) + geom_bar(aes(x = Gender, fill = as.factor(C_H)), position = "fill")
```



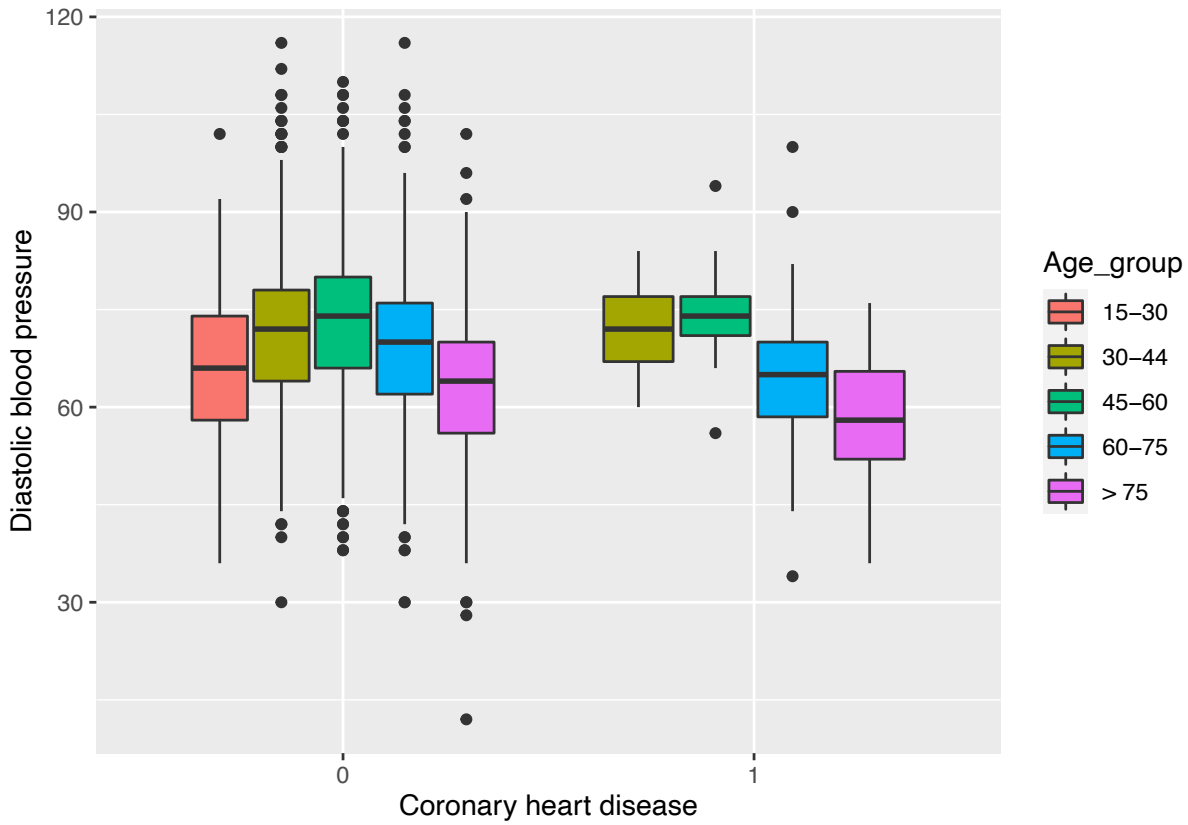

```
ggplot(coronary_heart_9) + geom_bar(aes(x = Age_group, fill = as.factor(C_H)), position = "fill")
```



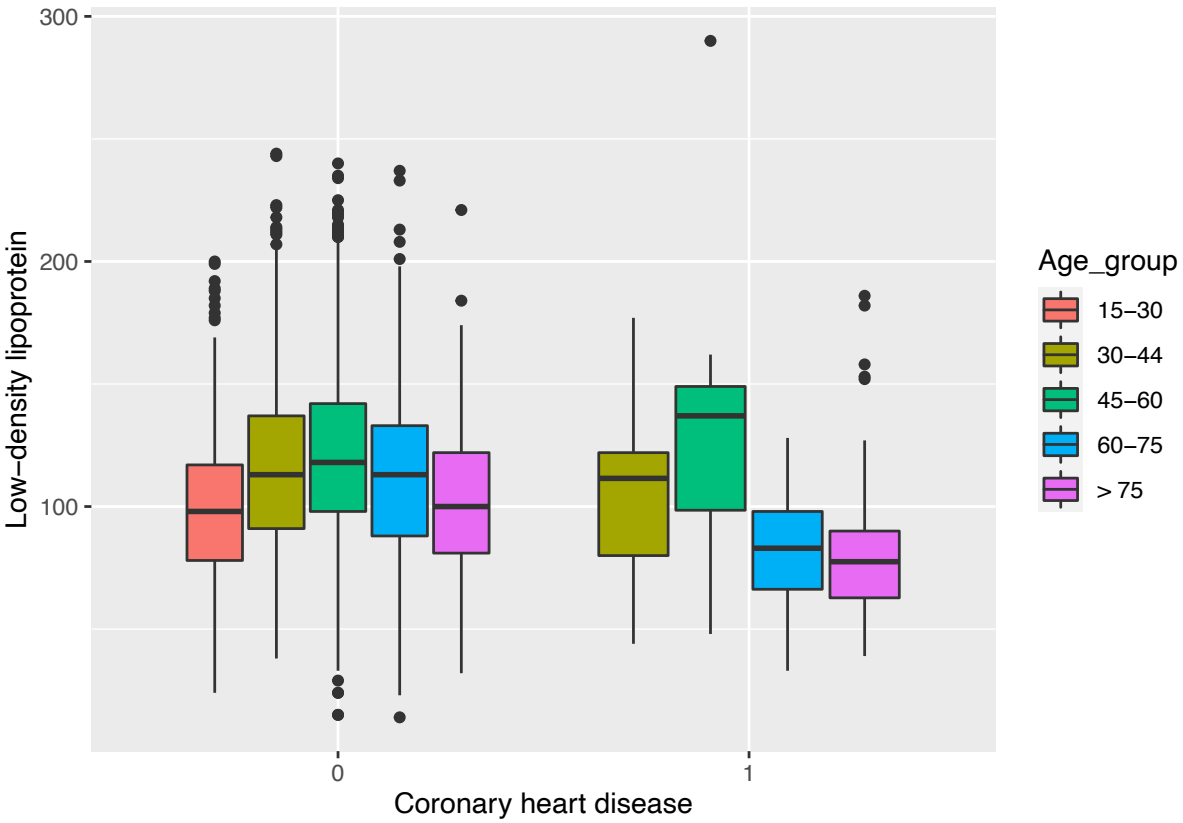

```
coronary_heart_9 %>% ggplot(aes(x = as.factor(C_H), y = BPXSY2)) +
  geom_boxplot(aes(fill = Age_group)) +
  labs(x = "Coronary heart disease", y = "Systolic blood pressure")
```



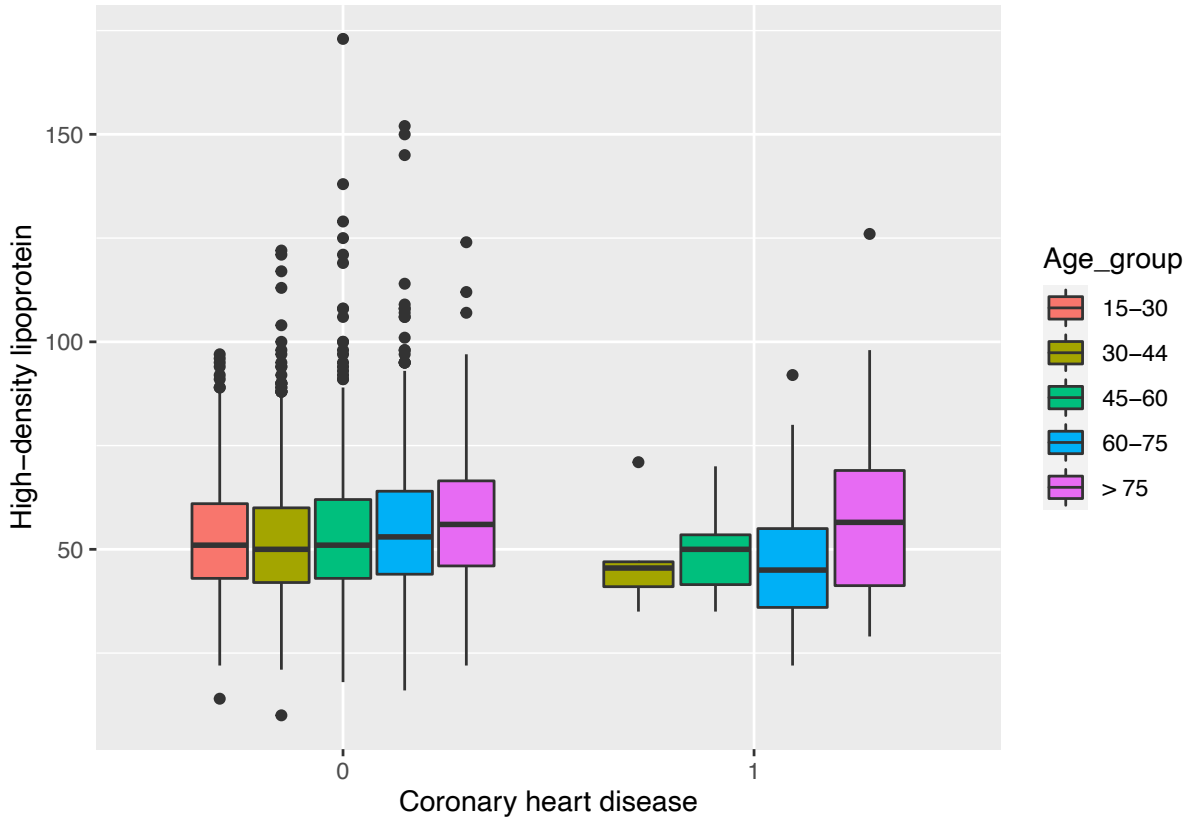

```
coronary_heart_9 %>% ggplot(aes(x = as.factor(C_H), y = BPXDI2)) +
  geom_boxplot(aes(fill = Age_group))+
  labs(x = "Coronary heart disease", y = "Diastolic blood pressure")
```



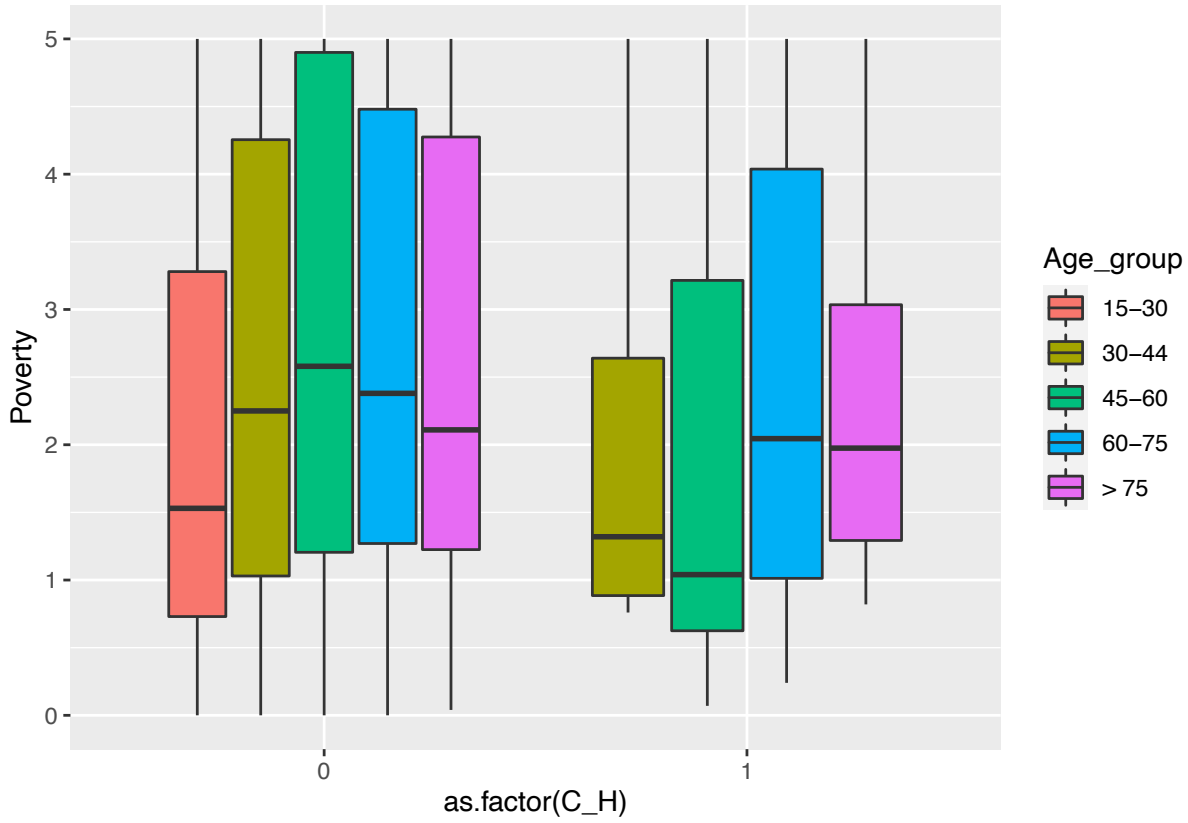

```
coronary_heart_9 %>% ggplot(aes(x = as.factor(C_H), y = LDL)) + geom_boxplot(aes(fill = Age_group))  +
  labs(x = "Coronary heart disease", y = "Low-density lipoprotein")
```




```
coronary_heart_9 %>% ggplot(aes(x = as.factor(C_H), y = HDL)) + geom_boxplot(aes(fill = Age_group)) +
  labs(x = "Coronary heart disease", y = "High-density lipoprotein")
```




```
coronary_heart_9 %>% ggplot(aes(x = as.factor(C_H), y = Poverty)) +
  geom_boxplot(aes(fill = Age_group))
```



Model fitting

```
set.seed(415)
train_idx <- sample(1:nrow(coronary_heart_9), floor(0.7*nrow(coronary_heart_9)))
train_df <- coronary_heart_9[train_idx, ]
test_df <- coronary_heart_9[-train_idx, ]
```

Best subset selection

```
n_predictors <- ncol(coronary_heart_9) - 1
regfit.full <- regsubsets(as.factor(C_H) ~ .-SEQN-Age_group,
                          data = coronary_heart_9, nvmax = n_predictors)
reg.summary <- summary(regfit.full)
#names(reg.summary)
which.min(reg.summary$bic)
```

```
## [1] 9
```

```
reg.summary$which[9,]
```

```
##   (Intercept)           Age         Race2         Race3         Race4
##          TRUE          TRUE         FALSE          TRUE         FALSE
##         Race6         Race7       Gender2       Poverty     diabetes1
##         FALSE         FALSE          TRUE          TRUE          TRUE
##        BPXSY2        BPXDI2  Triglyceride           LDL           HDL
##          TRUE          TRUE         FALSE          TRUE         FALSE
##      Smoking1       Alcohol           OW1      BMXWAIST hypertension1
##         FALSE         FALSE         FALSE         FALSE          TRUE
##       Glucose
##         FALSE
```

Logistic regression

```
fit1 <- glm(C_H ~ .-SEQN-Triglyceride-HDL-Smoking-Alcohol-OW-BMXWAIST-Glucose-Age_group,
            family = binomial, data = train_df)
summary(fit1)
```

```
##
## Call:
## glm(formula = C_H ~ . - SEQN - Triglyceride - HDL - Smoking -
##     Alcohol - OW - BMXWAIST - Glucose - Age_group, family = binomial,
##     data = train_df)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.5163  -0.2174  -0.1097  -0.0676   3.3250
##
## Coefficients:
##                Estimate Std. Error z value Pr(>|z|)
## (Intercept)   -5.113175   1.489376  -3.433 0.000597 ***
## Age            0.065048   0.011385   5.713 1.11e-08 ***
## Race2          1.497924   0.827989   1.809 0.070434 .
## Race3          1.465715   0.750298   1.954 0.050759 .
## Race4          0.671343   0.798438   0.841 0.400448
## Race6          0.955724   0.956300   0.999 0.317602
## Race7          1.453791   1.098761   1.323 0.185796
## Gender2       -1.099218   0.294386  -3.734 0.000189 ***
## Poverty       -0.137791   0.084768  -1.625 0.104057
## diabetes1      0.813110   0.289158   2.812 0.004924 **
## BPXSY2        -0.011506   0.007294  -1.577 0.114702
## BPXDI2        -0.018339   0.010996  -1.668 0.095359 .
## LDL           -0.006239   0.004205  -1.484 0.137895
## hypertension1  1.068930   0.314541   3.398 0.000678 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 648.68  on 2216  degrees of freedom
## Residual deviance: 470.88  on 2203  degrees of freedom
## AIC: 498.88
##
```

```
## Number of Fisher Scoring iterations: 8
```

```
logit_train_pred <- predict(fit1, train_df, type = "response")
logit_test_pred <- predict(fit1, test_df, type = "response")

# standard threshold
logit_train <- ifelse(logit_train_pred > 0.5, 1, 0)
logit_test <- ifelse(logit_test_pred > 0.5, 1, 0)
```

```
# standard threshold
train_err = mean(logit_train != train_df$C_H)
train_err
```

```
## [1] 0.03337844
```

```
test_err = mean(logit_test != test_df$C_H)
test_err
```

```
## [1] 0.03785489
```

```
logit_0.5th = table(predicted = logit_test, actual = test_df$C_H)
logit_0.5th
```

```
##          actual
## predicted   0   1
##         0 913  33
##         1   3   2
```

```
TPR <- logit_0.5th[2,2]/(logit_0.5th[2,2]+logit_0.5th[1,2])
TPR
```

```
## [1] 0.05714286
```

```
TNR <- logit_0.5th[1,1]/(logit_0.5th[1,1]+logit_0.5th[2,1])
TNR
```

```
## [1] 0.9967249
```

```
# setting threshold manually
logit_train_y <- ifelse(logit_train_pred > 0.04, 1, 0)
logit_test_y <- ifelse(logit_test_pred > 0.04, 1, 0)
```

```
# manually adjusted threshold
train_err = mean(logit_train_y != train_df$C_H)
train_err
```

```
## [1] 0.1858367
```

```
test_err = mean(logit_test_y != test_df$C_H)
test_err
```

```
## [1] 0.1861199
```

```
logit_manual = table(predicted = logit_test_y, actual = test_df$C_H)
logit_manual
```

```
##          actual
## predicted   0   1
##         0 746   7
##         1 170  28
```

```
TPR <- logit_manual[2,2]/(logit_manual[2,2]+logit_manual[1,2])
TPR
```

```
## [1] 0.8
```

```
TNR <- logit_manual[1,1]/(logit_manual[1,1]+logit_manual[2,1])
TNR
```

```
## [1] 0.8144105
```

LDA

```
fit_lda = lda(as.factor(C_H) ~ .-SEQN-Triglyceride-HDL-Smoking-Alcohol-OW
              -BMXWAIST-Glucose-Age_group,
              data = train_df)
fit_lda
```

```
## Call:
## lda(as.factor(C_H) ~ . - SEQN - Triglyceride - HDL - Smoking -
##     Alcohol - OW - BMXWAIST - Glucose - Age_group, data = train_df)
##
## Prior probabilities of groups:
##          0          1
## 0.96662156 0.03337844
##
## Group means:
##        Age      Race2     Race3     Race4      Race6      Race7   Gender2
## 0 47.13392 0.09472702 0.4540364 0.2132524 0.10172655 0.02566496 0.4708353
## 1 68.21622 0.10810811 0.6486486 0.1486486 0.04054054 0.02702703 0.2432432
##    Poverty diabetes1   BPXSY2   BPXDI2       LDL hypertension1
## 0 2.561633 0.1063929 122.1017 70.02706 113.01353     0.3443770
## 1 2.380676 0.3648649 126.2432 63.59459  96.04054     0.7837838
##
## Coefficients of linear discriminants:
##                         LD1
## Age            0.036091987
```

```
## Race2          0.488884811
## Race3          0.559497133
## Race4          0.087502124
## Race6          0.397693462
## Race7          0.446411172
## Gender2       -0.659296512
## Poverty       -0.097691801
## diabetes1      0.867385762
## BPXSY2        -0.011756388
## BPXDI2        -0.022872246
## LDL           -0.006563871
## hypertension1  0.699343502
```

```
lda_train_pred = predict(fit_lda, train_df)$class
lda_test_pred = predict(fit_lda, test_df)$class

lda_test = predict(fit_lda, test_df)$posterior[,2]
lda_train = predict(fit_lda, train_df)$posterior[,2]
```

```
# standard threshold
train_err = mean(lda_train_pred != train_df$C_H)
train_err
```

```
## [1] 0.03698692
```

```
test_err = mean(lda_test_pred != test_df$C_H)
test_err
```

```
## [1] 0.05047319
```

```
b <- table(predicted = lda_test_pred, actual = test_df$C_H)
b
```

```
##          actual
## predicted   0   1
##         0 900  32
##         1  16   3
```

```
# standard threshold
TPR <- b[2,2]/(b[2,2]+b[1,2])
TPR
```

```
## [1] 0.08571429
```

```
TNR <- b[1,1]/(b[1,1]+b[2,1])
TNR
```

```
## [1] 0.9825328
```

```r
# setting threshold manually
lda_train_y = ifelse(lda_train > 0.04, 1, 0)
lda_test_y = ifelse(lda_test > 0.04, 1, 0)
```

```r
# setting threshold manually
tr_err = mean(lda_train_y != train_df$C_H)
tr_err
```

```
## [1] 0.1763645
```

```r
te_err = mean(lda_test_y != test_df$C_H)
te_err
```

```
## [1] 0.1861199
```

```r
e <- table(predicted = lda_test_y, actual = test_df$C_H)
e
```

```
##          actual
## predicted   0   1
##         0 746   7
##         1 170  28
```

```r
TPR <- e[2,2]/(e[2,2]+e[1,2])
TPR
```

```
## [1] 0.8
```

```r
TNR <- e[1,1]/(e[1,1]+e[2,1])
TNR
```

```
## [1] 0.8144105
```

QDA

```r
fit_qda = qda(as.factor(C_H) ~ .-SEQN-Triglyceride-HDL-Smoking-Alcohol-OW
              -BMXWAIST-Glucose-Age_group,
              data = train_df)
fit_qda
```

```
## Call:
## qda(as.factor(C_H) ~ . - SEQN - Triglyceride - HDL - Smoking -
##     Alcohol - OW - BMXWAIST - Glucose - Age_group, data = train_df)
##
## Prior probabilities of groups:
##          0          1
## 0.96662156 0.03337844
##
```

```
## Group means:
##        Age      Race2     Race3     Race4      Race6      Race7   Gender2
## 0 47.13392 0.09472702 0.4540364 0.2132524 0.10172655 0.02566496 0.4708353
## 1 68.21622 0.10810811 0.6486486 0.1486486 0.04054054 0.02702703 0.2432432
##    Poverty diabetes1   BPXSY2   BPXDI2       LDL hypertension1
## 0 2.561633 0.1063929 122.1017 70.02706 113.01353     0.3443770
## 1 2.380676 0.3648649 126.2432 63.59459  96.04054     0.7837838
```

```
qda_train_pred = predict(fit_qda, train_df)$class
qda_test_pred = predict(fit_qda, test_df)$class
qda_test = predict(fit_qda, test_df)$posterior[,2]
qda_train = predict(fit_qda, train_df)$posterior[,2]
```

```
# standard threshold
train_err = mean(qda_train_pred != train_df$C_H)
train_err
```

```
## [1] 0.07803338
```

```
test_err = mean(qda_test_pred != test_df$C_H)
test_err
```

```
## [1] 0.08727655
```

```
c <- table(predicted = qda_test_pred, actual = test_df$C_H)
c
```

```
##          actual
## predicted   0   1
##         0 854  21
##         1  62  14
```

```
TPR <- c[2,2]/(c[2,2]+c[1,2])
TPR
```

```
## [1] 0.4
```

```
TNR <- c[1,1]/(c[1,1]+c[2,1])
TNR
```

```
## [1] 0.9323144
```

```
# setting threshold manually
qda_train_y = ifelse(qda_train > 0.04, 1, 0)
qda_test_y = ifelse(qda_test > 0.04, 1, 0)
```

```
# setting threshold manually
tr_err = mean(qda_train_y != train_df$C_H)
tr_err
```

```
## [1] 0.2214705
```

```r
te_err = mean(qda_test_y != test_df$C_H)
te_err
```

```
## [1] 0.2229232
```

```r
d <- table(predicted = qda_test_y, actual = test_df$C_H)
d
```

```
##          actual
## predicted   0   1
##         0 713   9
##         1 203  26
```

```r
TPR <- d[2,2]/(d[2,2]+d[1,2])
TPR
```

```
## [1] 0.7428571
```

```r
TNR <- d[1,1]/(d[1,1]+d[2,1])
TNR
```

```
## [1] 0.7783843
```

Random forest classification

```r
set.seed(415)
err_tr = rep(0, 16)
err_te = rep(0, 16)
for (i in 1:16) {
  bag_mod <- randomForest(
  as.factor(C_H) ~ .-SEQN-Age_group, # Model formula
  data=train_df, # Training data
  mtry=i,
  importance=TRUE) # Return feature importance measures

  rf_pred_tr = predict(bag_mod, train_df)
  rf_pred_te = predict(bag_mod, test_df)

  err_tr[i] = mean(rf_pred_tr != train_df$C_H)
  err_te[i] = mean(rf_pred_te != test_df$C_H)
}
err_te[which.min(err_te)]
```

```
## [1] 0.03575184
```

```r
which.min(err_te)
```

```
## [1] 8
```

```
rf_best <- randomForest(
  as.factor(C_H) ~ .-SEQN-Age_group, # Model formula
  data=train_df, # Training data
  mtry=8, # Use 8 columns
  importance=TRUE) # Return feature importance measures

rf_prob_te = predict(rf_best, test_df, type = "prob")
```

```
varImpPlot(rf_best)
```


```
# standard threshold
rf_train_pred <- predict(rf_best, train_df)
rf_test_pred <- predict(rf_best, test_df)

rf_prob_tr <- predict(rf_best, train_df, type = "prob")[,2]
rf_prob_te <- predict(rf_best, test_df, type = "prob")[,2]
```

```
# standard threshold
train_err = mean(rf_train_pred  != train_df$C_H)
train_err
```

```
## [1] 0
```

```r
test_err = mean(rf_test_pred != test_df$C_H)
test_err
```

```
## [1] 0.03785489
```

```r
g = table(predicted = rf_train_pred, actual = train_df$C_H)
g
```

```
##          actual
## predicted    0    1
##         0 2143    0
##         1    0   74
```

```r
f = table(predicted = rf_test_pred, actual = test_df$C_H)
f
```

```
##          actual
## predicted   0   1
##         0 914  34
##         1   2   1
```

```r
# standard threshold
TPR <- f[2,2]/(f[2,2]+f[1,2])
TPR
```

```
## [1] 0.02857143
```

```r
TNR <- f[1,1]/(f[1,1]+f[2,1])
TNR
```

```
## [1] 0.9978166
```

```r
# setting threshold manually
rf_train_y <- ifelse(rf_prob_tr > 0.05, 1, 0)
rf_test_y <- ifelse(rf_prob_te > 0.04, 1, 0)
```

```r
# manually adjusted threshold
train_err = mean(rf_train_y != train_df$C_H)
train_err
```

```
## [1] 0.07352278
```

```r
test_err = mean(rf_test_y != test_df$C_H)
test_err
```

```
## [1] 0.2471083
```

```
rf_manual = table(predicted = rf_test_y, actual = test_df$C_H)
rf_manual
```

```
##           actual
## predicted   0   1
##         0 690   9
##         1 226  26
```

```
TPR <- rf_manual[2,2]/(rf_manual[2,2]+rf_manual[1,2])
TPR
```

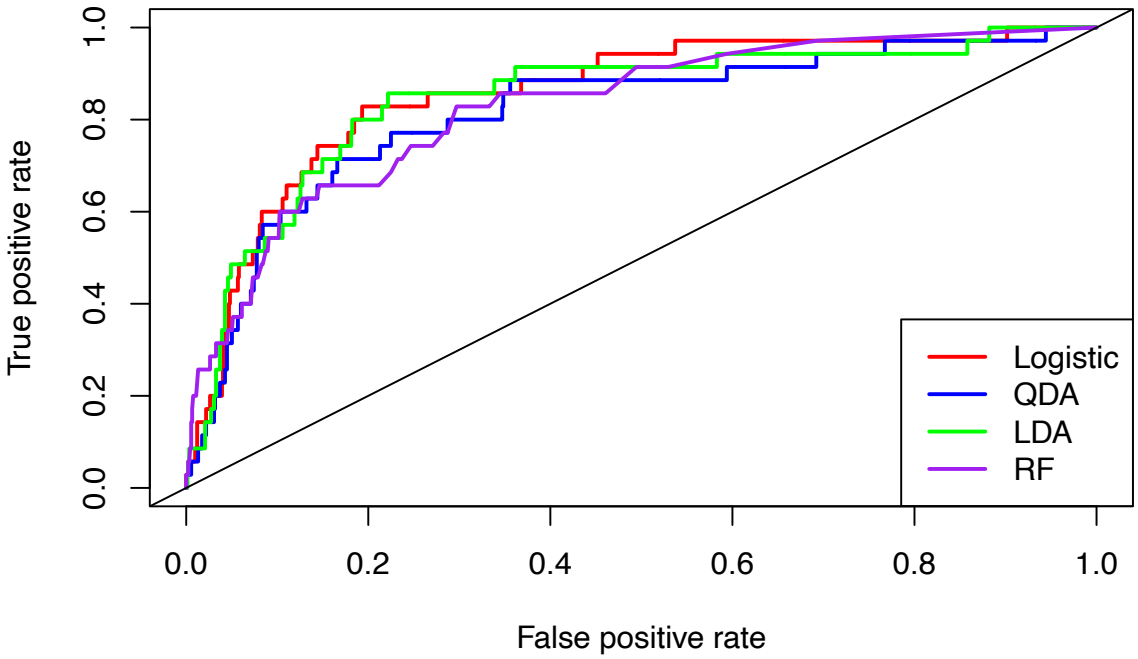
```
## [1] 0.7428571
```

```
TNR <- rf_manual[1,1]/(rf_manual[1,1]+rf_manual[2,1])
TNR
```

```
## [1] 0.7532751
```

ROC plot

```
pred1 = prediction(logit_test_pred, test_df$C_H)
pred2 = prediction(qda_test, test_df$C_H)
pred3 = prediction(lda_test, test_df$C_H)
pred4 = prediction(rf_prob_te, test_df$C_H)
roc1 = performance(pred1, "tpr", "fpr")
roc2 = performance(pred2, "tpr", "fpr")
roc3 = performance(pred3, "tpr", "fpr")
roc4 = performance(pred4, "tpr", "fpr")
plot(roc1, col='red', lwd =2)
plot(roc2, col='blue', lwd =2, add = TRUE)
plot(roc3, col='green', lwd =2, add = TRUE)
plot(roc4, col='purple', lwd =2, add = TRUE)
legend("bottomright", c("Logistic","QDA", "LDA", "RF"),
       col = c("red", "blue", "green", "purple"), lwd = 2)
#lines(roc2)
abline(0,1)
```



Classification method	Threshold	True Positive Rate	True Negative Rate
Logistic regression	0.500	0.0571	0.997
LDA	0.500	0.0857	0.982
QDA	0.500	0.400	0.932
Random forest	0.500	0.0286	0.998

Classification method	Threshold	True Positive Rate	True Negative Rate
Logistic regression	0.04	0.800	0.814
LDA	0.04	0.800	0.814
QDA	0.04	0.743	0.778
Random forest	0.04	0.743	0.753