hearthdisease

Ming Shen

11/17/2019

Load packages and datasets

```
library(tidyverse)
```

```
## -- Attaching packages --------------------------------- tidyverse 1.2.1 --

## v ggplot2 3.3.3     v purrr   0.3.4
## v tibble  3.0.6     v dplyr   1.0.4
## v tidyr   1.1.2     v stringr 1.4.0
## v readr   1.4.0     v forcats 0.5.1

## Warning: package 'ggplot2' was built under R version 3.6.2

## Warning: package 'tibble' was built under R version 3.6.2

## Warning: package 'tidyr' was built under R version 3.6.2

## Warning: package 'readr' was built under R version 3.6.2

## Warning: package 'purrr' was built under R version 3.6.2

## Warning: package 'dplyr' was built under R version 3.6.2

## Warning: package 'forcats' was built under R version 3.6.2

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

```
library(MASS)
```

```
##
## Attaching package: 'MASS'

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

```
heart_Df <- read.csv("/Users/mingshen/Desktop/CSUEB/Fall 2020/Project/heartdisease/heart_cleveland_uplo
```

Data cleaning

```
## Check for missing Values
colSums(is.na(heart_Df)) #=> colSums:0
```

```
##       age       sex        cp  trestbps      chol       fbs   restecg
##         0         0         0         0         0         0         0
##   thalach     exang   oldpeak     slope        ca      thal condition
##         0         0         0         0         0         0         0
```

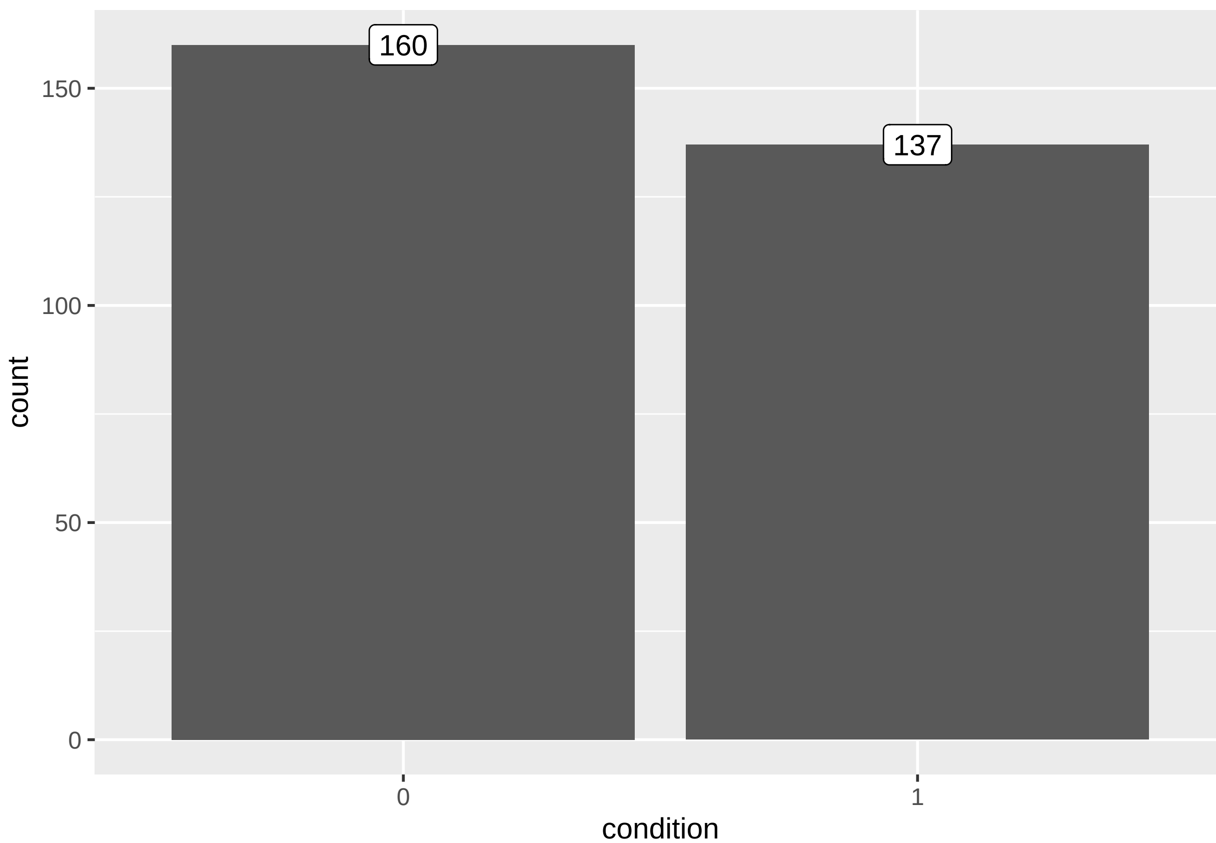
```
## Correctly convert variables to factors.
for (var in names(heart_Df)) {
  if(length(unique(heart_Df[[var]])) < 4) {
    heart_Df[[var]] <- as.factor(heart_Df[[var]])
  }
```

```
}

str(heart_Df)
```

```
## 'data.frame':    297 obs. of  14 variables:
##  $ age      : int  69 69 66 65 64 64 63 61 60 59 ...
##  $ sex      : Factor w/ 2 levels "0","1": 2 1 1 2 2 2 2 2 1 2 ...
##  $ cp       : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ trestbps : int  160 140 150 138 110 170 145 134 150 178 ...
##  $ chol     : int  234 239 226 282 211 227 233 234 240 270 ...
##  $ fbs      : Factor w/ 2 levels "0","1": 2 1 1 2 1 1 2 1 1 1 ...
##  $ restecg  : Factor w/ 3 levels "0","1","2": 3 1 1 3 3 3 3 1 1 3 ...
##  $ thalach  : int  131 151 114 174 144 155 150 145 171 145 ...
##  $ exang    : Factor w/ 2 levels "0","1": 1 1 1 1 2 1 1 1 1 1 ...
##  $ oldpeak  : num  0.1 1.8 2.6 1.4 1.8 0.6 2.3 2.6 0.9 4.2 ...
##  $ slope    : Factor w/ 3 levels "0","1","2": 2 1 3 2 2 2 3 2 1 3 ...
##  $ ca       : int  1 2 0 1 0 0 0 2 0 0 ...
##  $ thal     : Factor w/ 3 levels "0","1","2": 1 1 1 1 1 3 2 1 1 3 ...
##  $ condition: Factor w/ 2 levels "0","1": 1 1 1 2 1 1 1 2 1 1 ...
```

Data visualizing

```
## Outcome Variable
heart_Df %>%
  ggplot(aes(condition)) +
  geom_bar() +
  geom_label(stat = "count", aes(label = ..count..))
```



Top Distributions Here we will examine, basically, the circumstances of top performers or influences on the outcome variable.

```
numericVars <- which(sapply(heart_Df, is.numeric))

factorVars <- which(sapply(heart_Df, is.factor))

cat('There are', length(numericVars), 'numeric variables, and', length(factorVars), 'categoric variable
```

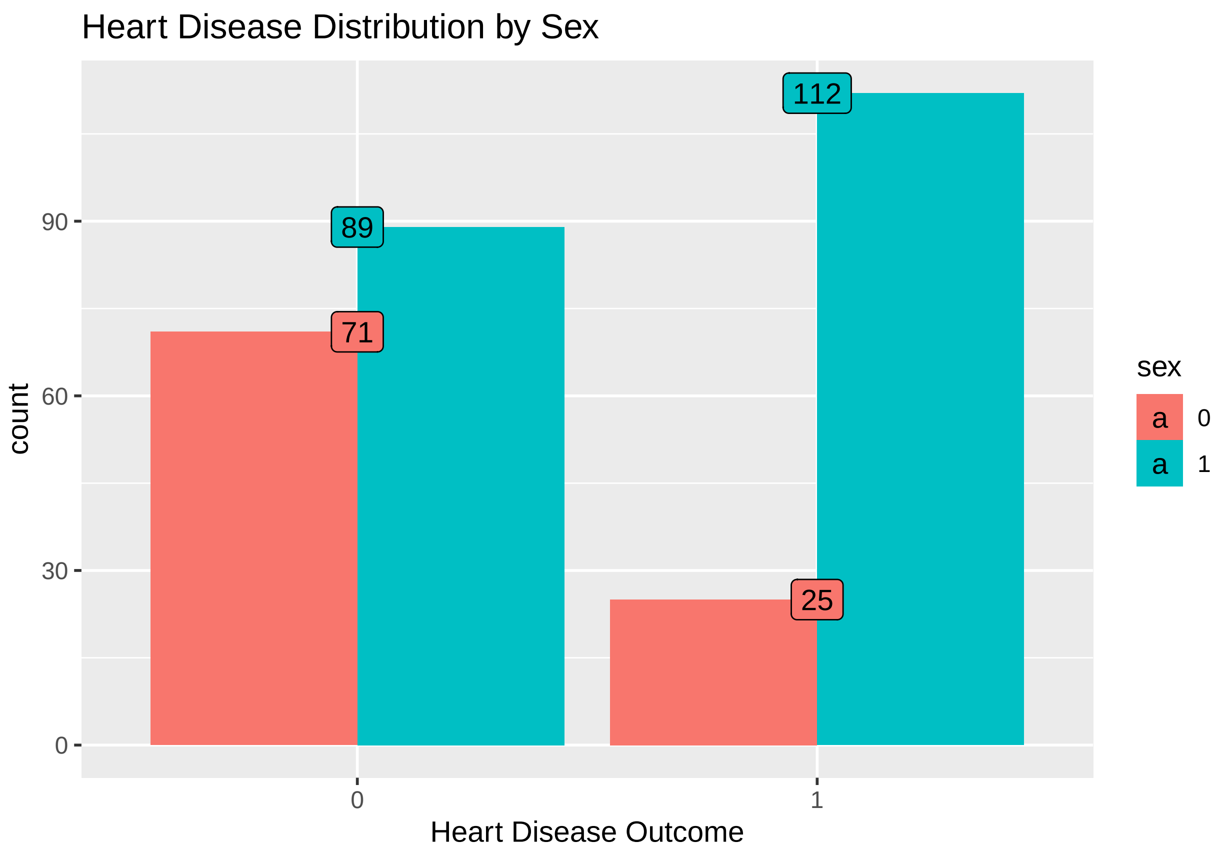
```
## There are 7 numeric variables, and 7 categoric variables
```

```
names(factorVars)
```

```
## [1] "sex"       "fbs"       "restecg"   "exang"     "slope"     "thal"
## [7] "condition"
```

Outcome of Heart Disease by Factor Variables

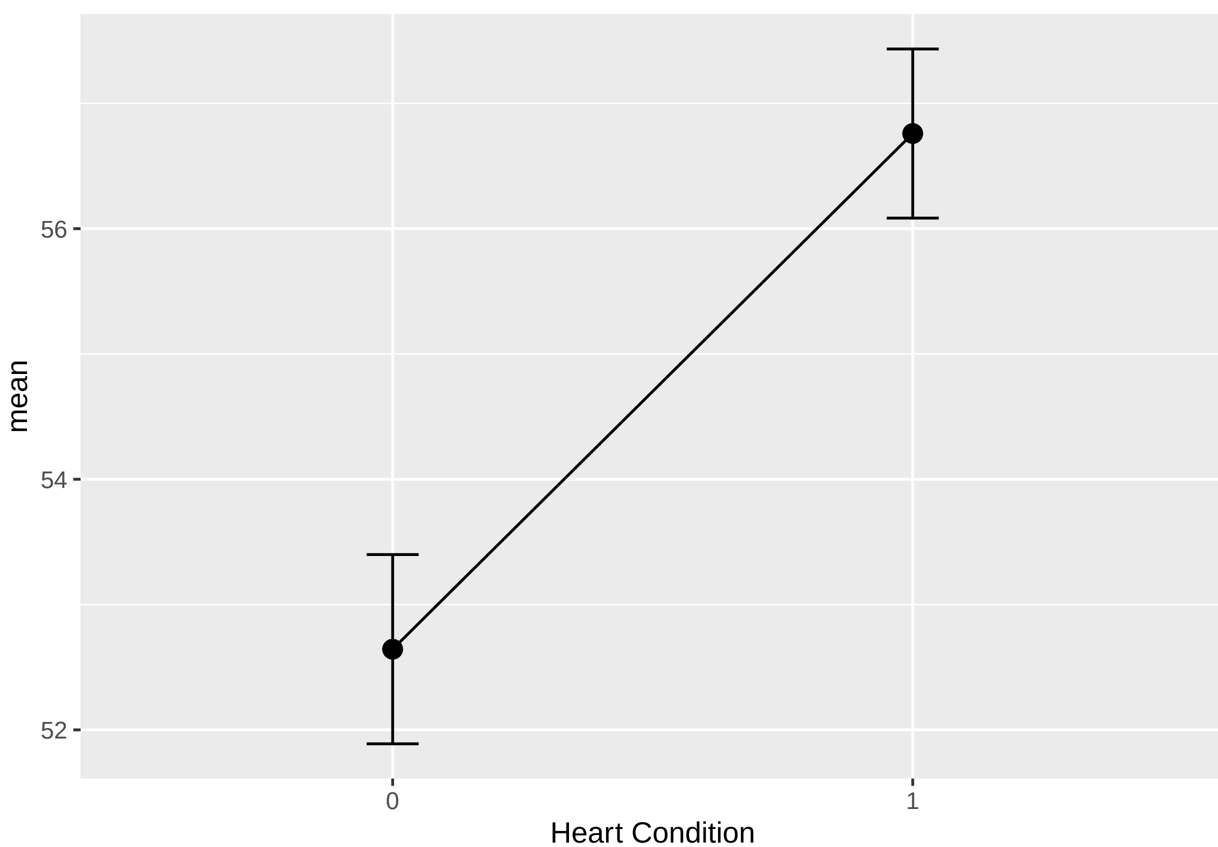
```
### Sex: Stacked Bar Chart

heart_Df %>%
  ggplot(aes(x = factor(condition), fill = sex)) +
  geom_bar(position = position_dodge(preserve = "single")) +
  labs(x = "Heart Disease Outcome", title = "Heart Disease Distribution by Sex") +
  geom_label(stat = "count", aes(label = ..count..))
```



Outcome of Heart Disease by Numeric Variables

```
### Age: Mean/SEM Plots

## Function to summarize the data by its categories and plot the Mean and Std. Error Plot.
```

```r
numeric_plot_data_function <- function(.data, var) {
  .data %>%
    # group by Outcome Variable
    group_by(condition) %>% # {{outcome}}
    # calculate means, standard deviations,
    # standard errors, and 95% confidence
    # intervals by Outcome Variable
    summarize(n = n(),
              mean = mean({{var}}),
              sd = sd({{var}}),
              se = sd / sqrt(n),
              ci = qt(0.975, df = n -1) * se,
              .groups = "drop") %>%
    # Plot the means and standard errors
    ggplot(aes(x = factor(condition), y = mean, group = 1)) +
    geom_point(size = 3) +
    geom_line() +
    xlab("Heart Condition") +
    geom_errorbar(aes(ymin = mean - se,
                      ymax = mean + se),
                  width = .1)
}


## Check
numeric_plot_data_function(heart_Df, age)
```

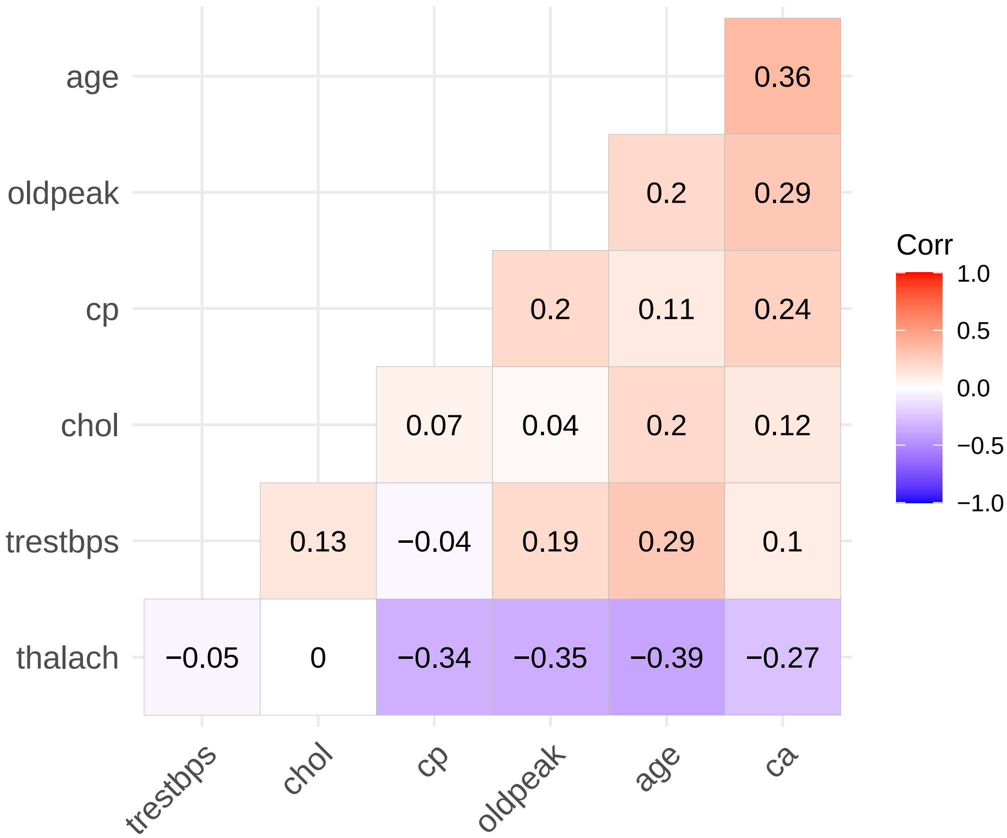


Underlying Relationships Numeric Variables: Pairwise-Correlation Correlation plots help us to visualize the pairwise relationships between a set of quantitative variables by displaying their correlations using color or shading. It is important to know that this is applicable to only numeric Variables. Hence, we are checking for the pair relationship within our dataset, although we know the outcome variable is not numerically - we will get to that later.

```r
## Correlation Set

numericVars <- select_if(heart_Df, is.numeric) # select only Column-set numeric vars

corr <- cor(numericVars, use = "pairwise.complete.obs")
round(corr, 2)
```

```
##            age    cp trestbps chol thalach oldpeak    ca
## age       1.00  0.11     0.29 0.20   -0.39    0.20  0.36
## cp        0.11  1.00    -0.04 0.07   -0.34    0.20  0.24
## trestbps  0.29 -0.04     1.00 0.13   -0.05    0.19  0.10
## chol      0.20  0.07     0.13 1.00    0.00    0.04  0.12
## thalach  -0.39 -0.34    -0.05 0.00    1.00   -0.35 -0.27
## oldpeak   0.20  0.20     0.19 0.04   -0.35    1.00  0.29
## ca        0.36  0.24     0.10 0.12   -0.27    0.29  1.00
```

```r
## Correlation Visualisation
library(ggcorrplot)

ggcorrplot(corr,
           hc.order = TRUE, # reorders the variables, placing variables with similar correlation patter
           type = "lower", lab = TRUE)
```



Categorical Predictors On Categorical Outcome: ChiSquare Chi-square statistics is used to investigate whether distributions of categorical variables differ from one another. Chi-square test is also useful while comparing the tallies or counts of categorical responses between two(or more) independent groups.

Our aim is to test the hypothesis whether the categorical predictor variable is independent of their heart Condition at .05 significance level.

Where P-Value is > 0.05 we accept our hypothesis that the variables are independent and there is little or weak correlation between these variable, and vice versa.

```
for (var in names(factorVars)) {
  if (var != "condition"){
    cat("+----------------------+")
    cat("\n","Data Table for",var, "Variable","\n")
    # contigency table
    factor_hd <- table(heart_Df[[var]], heart_Df$condition)
    cat("----------------------")
    print(factor_hd)
    cat("----------------------")
    cat("\n","Chi Square Test @ 0.05")
    # chiSq
    print(chisq.test(factor_hd))
    cat("+----------------------+","\n\n")
  }

}
```

```
## +----------------------+
##  Data Table for sex Variable
## ----------------------
##       0   1
##   0  71  25
##   1  89 112
## ----------------------
##  Chi Square Test @ 0.05
##  Pearson's Chi-squared test with Yates' continuity correction
##
## data:  factor_hd
## X-squared = 21.852, df = 1, p-value = 2.946e-06
##
## +----------------------+
##
## +----------------------+
##  Data Table for fbs Variable
## ----------------------
##       0   1
##   0 137 117
##   1  23  20
## ----------------------
##  Chi Square Test @ 0.05
##  Pearson's Chi-squared test with Yates' continuity correction
##
## data:  factor_hd
## X-squared = 1.9997e-31, df = 1, p-value = 1
##
## +----------------------+
```

```
##
## +----------------------+
##  Data Table for restecg Variable
## ----------------------
##       0  1
##   0 92 55
##   1  1  3
##   2 67 79
## ----------------------
##  Chi Square Test @ 0.05

## Warning in chisq.test(factor_hd): Chi-squared approximation may be
## incorrect

##
##  Pearson's Chi-squared test
##
## data:  factor_hd
## X-squared = 9.5755, df = 2, p-value = 0.008331
##
## +----------------------+
##
## +----------------------+
##  Data Table for exang Variable
## ----------------------
##        0   1
##   0 137  63
##   1  23  74
## ----------------------
##  Chi Square Test @ 0.05
##  Pearson's Chi-squared test with Yates' continuity correction
##
## data:  factor_hd
## X-squared = 50.943, df = 1, p-value = 9.511e-13
##
## +----------------------+
##
## +----------------------+
##  Data Table for slope Variable
## ----------------------
##        0   1
##   0 103  36
##   1  48  89
##   2   9  12
## ----------------------
##  Chi Square Test @ 0.05
##  Pearson's Chi-squared test
##
## data:  factor_hd
## X-squared = 43.473, df = 2, p-value = 3.63e-10
##
## +----------------------+
##
## +----------------------+
##  Data Table for thal Variable
```

```
## ----------------------
##       0   1
##   0 127  37
##   1   6  12
##   2  27  88
## ----------------------
##  Chi Square Test @ 0.05
##  Pearson's Chi-squared test
##
## data:  factor_hd
## X-squared = 82.46, df = 2, p-value < 2.2e-16
##
## +----------------------+
```

Numeric Predictor Variables on Categorical Outcome: Logistic Regression¶ Naturally since the Outcome Variable is Boolean, the choice for analysing the influence of Numeric Predictors is by using Logistic Regression. Logistic regression can be used to explore the relationship between a binary response variable and an explanatory variable while other variables are held constant. Binary response variables have two levels (yes/no, lived/died, pass/fail, malignant/benign).

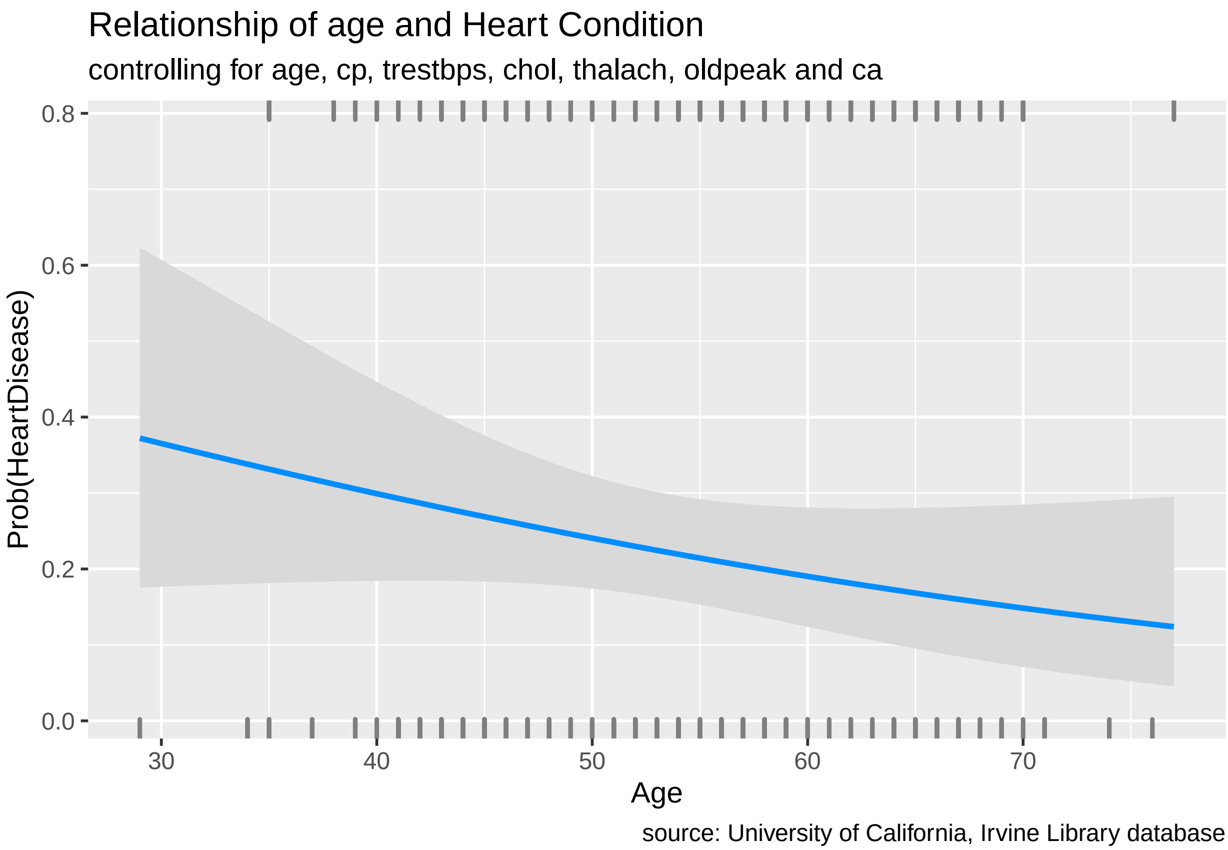
```r
heartDisease_glm <- glm(condition ~ age + cp + trestbps + chol + thalach + oldpeak + ca,
                 family = "binomial",
                 data = heart_Df)

## Check
heartDisease_glm
```

```
##
## Call:  glm(formula = condition ~ age + cp + trestbps + chol + thalach +
##     oldpeak + ca, family = "binomial", data = heart_Df)
##
## Coefficients:
## (Intercept)          age           cp     trestbps         chol
##    -0.91558     -0.02986      0.77839      0.02403      0.00225
##     thalach      oldpeak           ca
##    -0.02931      0.64635      1.18475
##
## Degrees of Freedom: 296 Total (i.e. Null);  289 Residual
## Null Deviance:       409.9
## Residual Deviance: 253.8     AIC: 269.8
```

```r
## Plot results
library(visreg)
```

```
## Warning: package 'visreg' was built under R version 3.6.2
```

```r
visreg(heartDisease_glm, "age",
       gg = TRUE,
       scale="response") +
  labs(y = "Prob(HeartDisease)",
       x = "Age",
       title = "Relationship of age and Heart Condition",
       subtitle = "controlling for age, cp, trestbps, chol, thalach, oldpeak and ca",
       caption = "source: University of California, Irvine Library database")
```



Relationship of age and Heart Condition
controlling for age, cp, trestbps, chol, thalach, oldpeak and ca
Prob(HeartDisease)
0.0
0.2
0.4
0.6
0.8
30
40
50
60
70
Age
source: University of California, Irvine Library database