CS109b Homework 2 Submission

Ihsaan Patel

February 11, 2017

Libraries & Helper Functions

```r
library(ggplot2)
library(gam)
library(boot)

# Function to compute k-fold cross-validation accuracy for a given classification model
cv_accuracy = function(data, param_val, k) {
  # Input:
  #   'data' - data frame with training data set used to fit the model
  #   'param_val' - list with parameter values to tune through cross-validation
  #   'k' - number of folds for CV
  # Output:
  #   'cv_acc' - array of model accuracies for the different parameters

  num_param = length(param_val) # Number of parameters
  cv_acc = rep(0., num_param) # list to store model accuracies

  # Iterate over parameter values
  for(i in 1:num_param){
    gam_formula <- as.formula(sprintf("HeartDisease ~ s(Age, spar=%1$f) + Sex + s(RestBP,spar=%1$f) + E
    model <- gam(gam_formula, family = binomial(link = "logit"), data = data)
    acc <- 1 - cv.glm(data, model, K = k)$delta[1]
    cv_acc[i] <- acc
  }
  return(cv_acc)
}
```

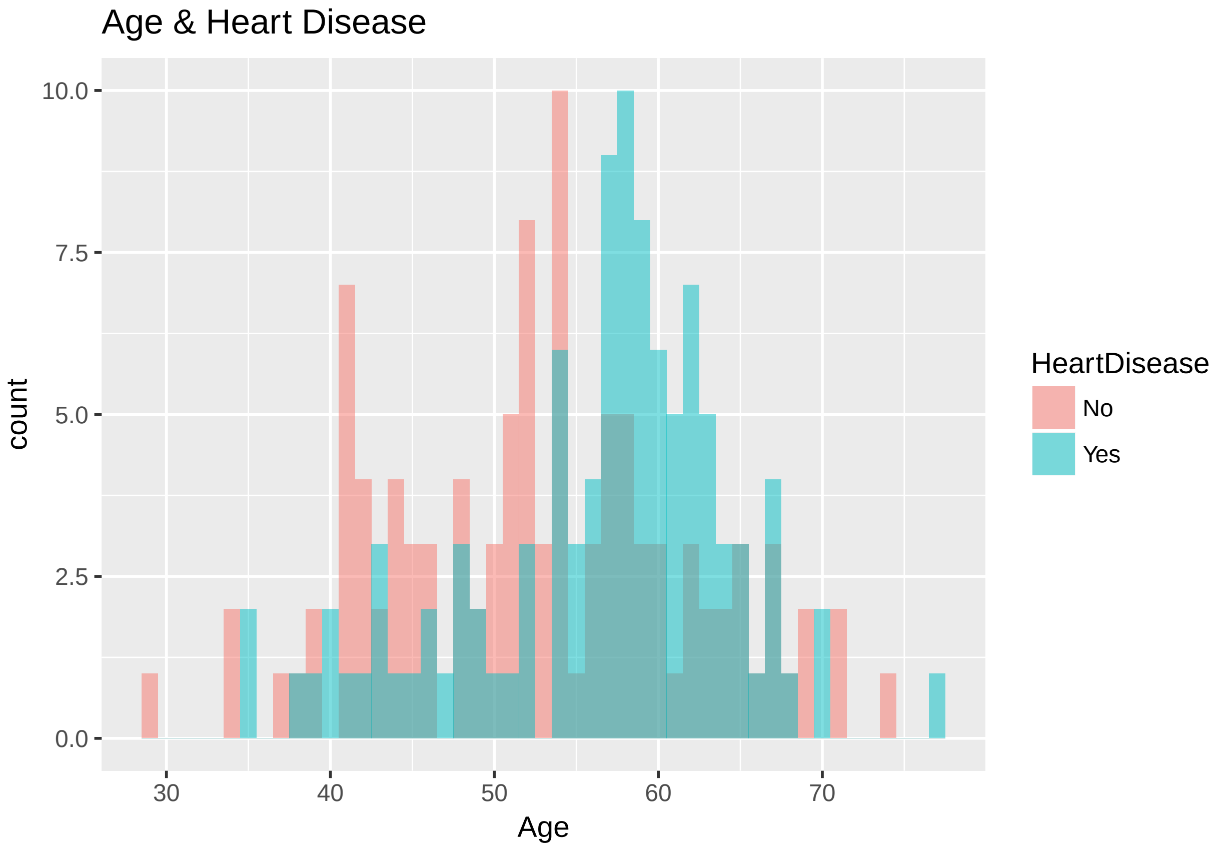
Problem 1: Heart Disease Diagnosis

Visual Inspection of Data

```r
# Load data
train_dataset1 <- read.table("dataset_1_train.txt",header = TRUE, sep = ",")
test_dataset1 <- read.table("dataset_1_test.txt",header = TRUE, sep = ",")
```

Age

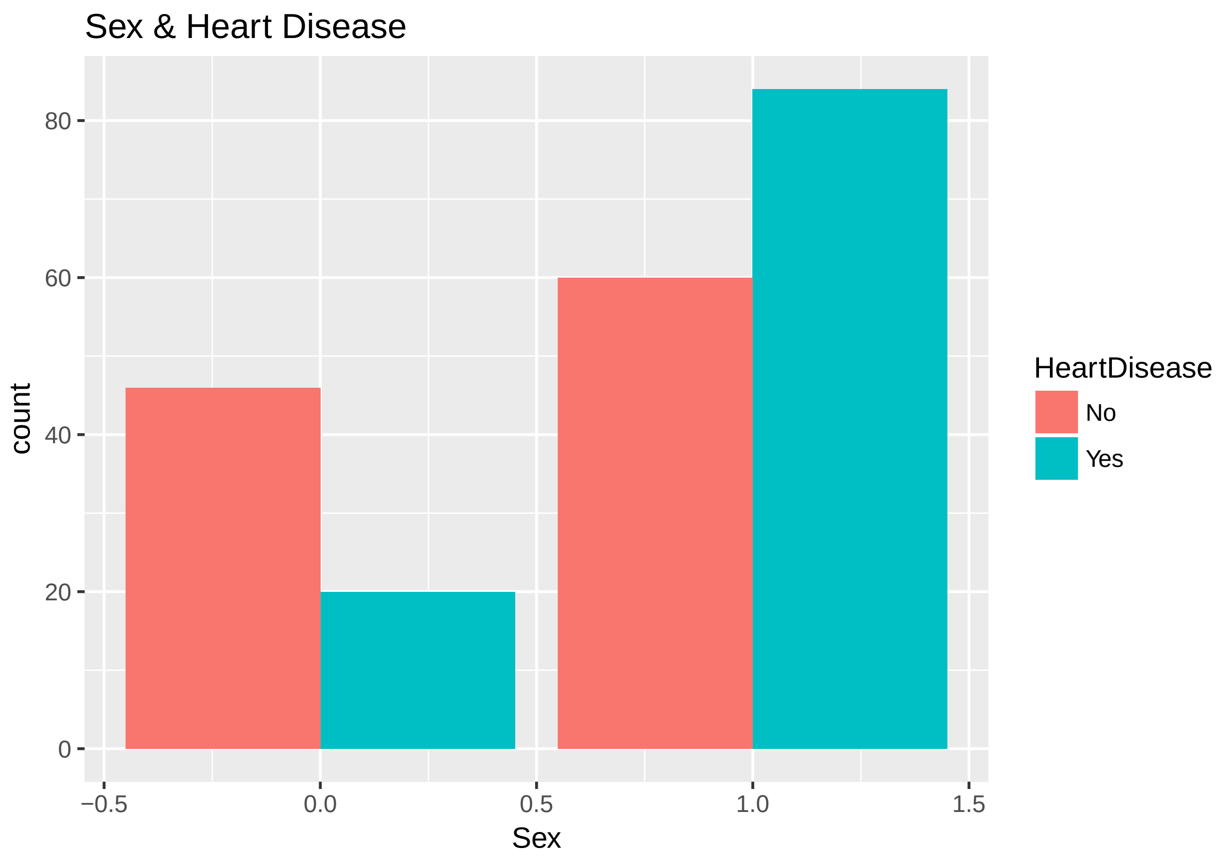
```r
# Plot and visualize Age
ggplot(train_dataset1, mapping = aes(x = Age , fill = HeartDisease)) + geom_histogram(binwidth=1, alpha
```



Older people appear to have much higher incidents of heart disease, particularly between the ages of 55 and 70.

Sex

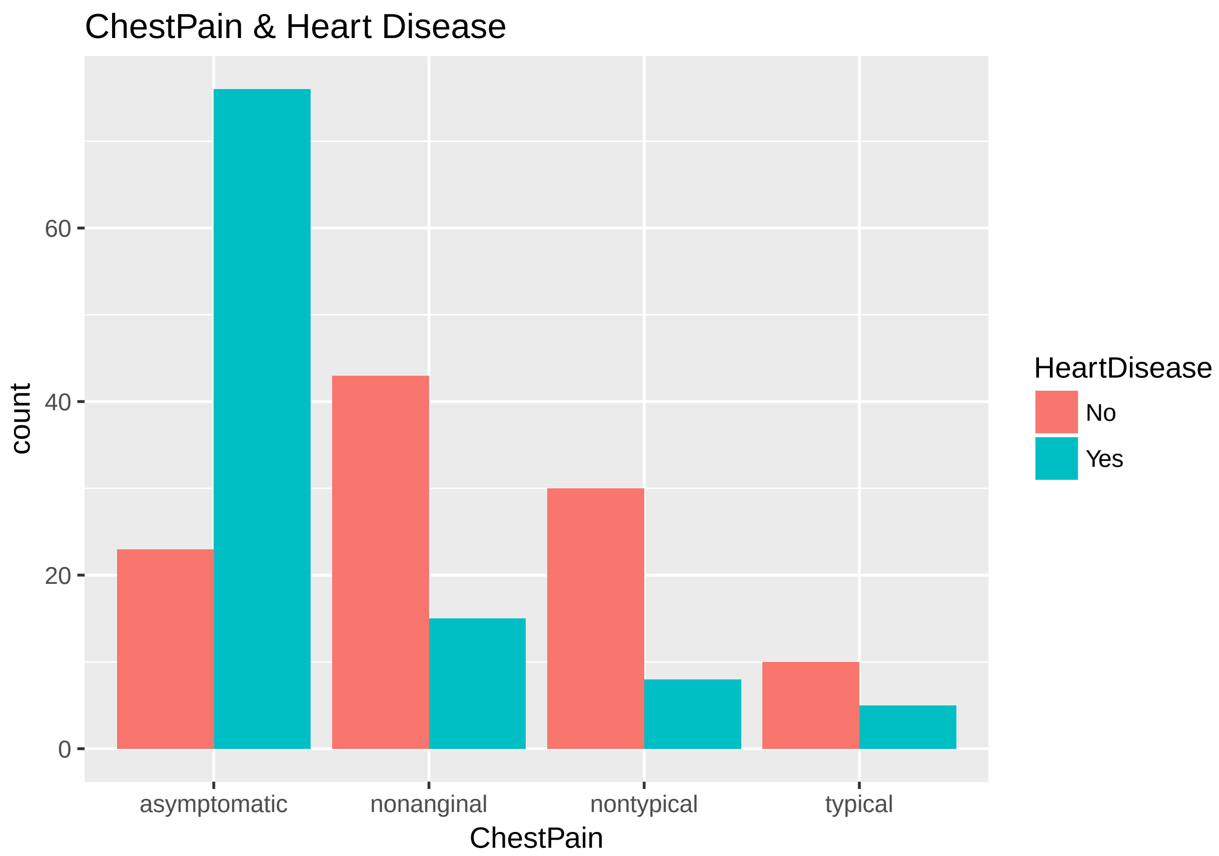
```
# Plot and visualize Sex
ggplot(train_dataset1, mapping = aes(x = Sex, fill = HeartDisease)) + geom_bar(position = "dodge") + gg
```



There appears to be a much higher incidence of heart disease in one of the sexes

ChestPain

```
# Plot and visualize Chest Pain
ggplot(train_dataset1, mapping = aes(x = ChestPain, fill = HeartDisease)) + geom_bar(position = "dodge"
```

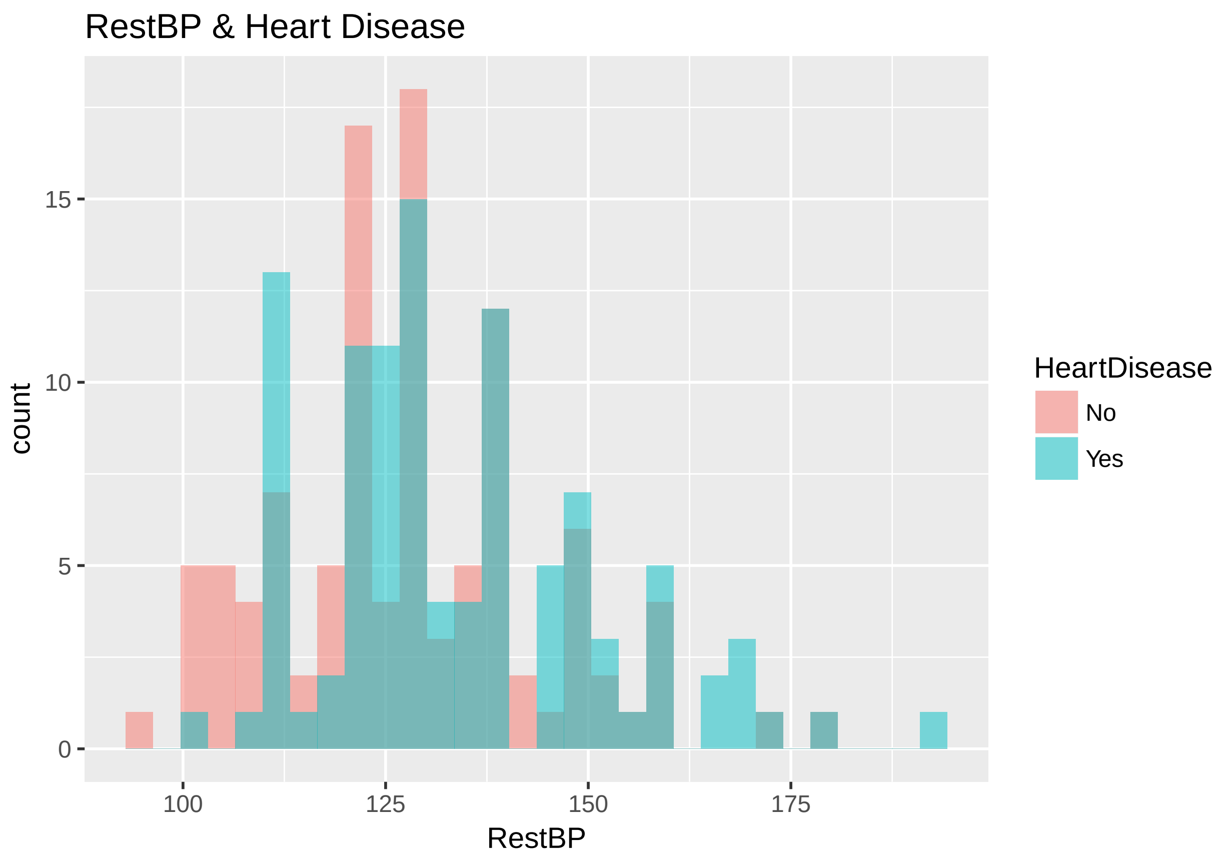


There's a much higher incidence of heart disease in individuals with asymptomatic chest pain.

RestBP

```r
# Plot and visualize RestBP
ggplot(train_dataset1, mapping = aes(x = RestBP, fill = HeartDisease)) + geom_histogram( alpha=.5, posi
```

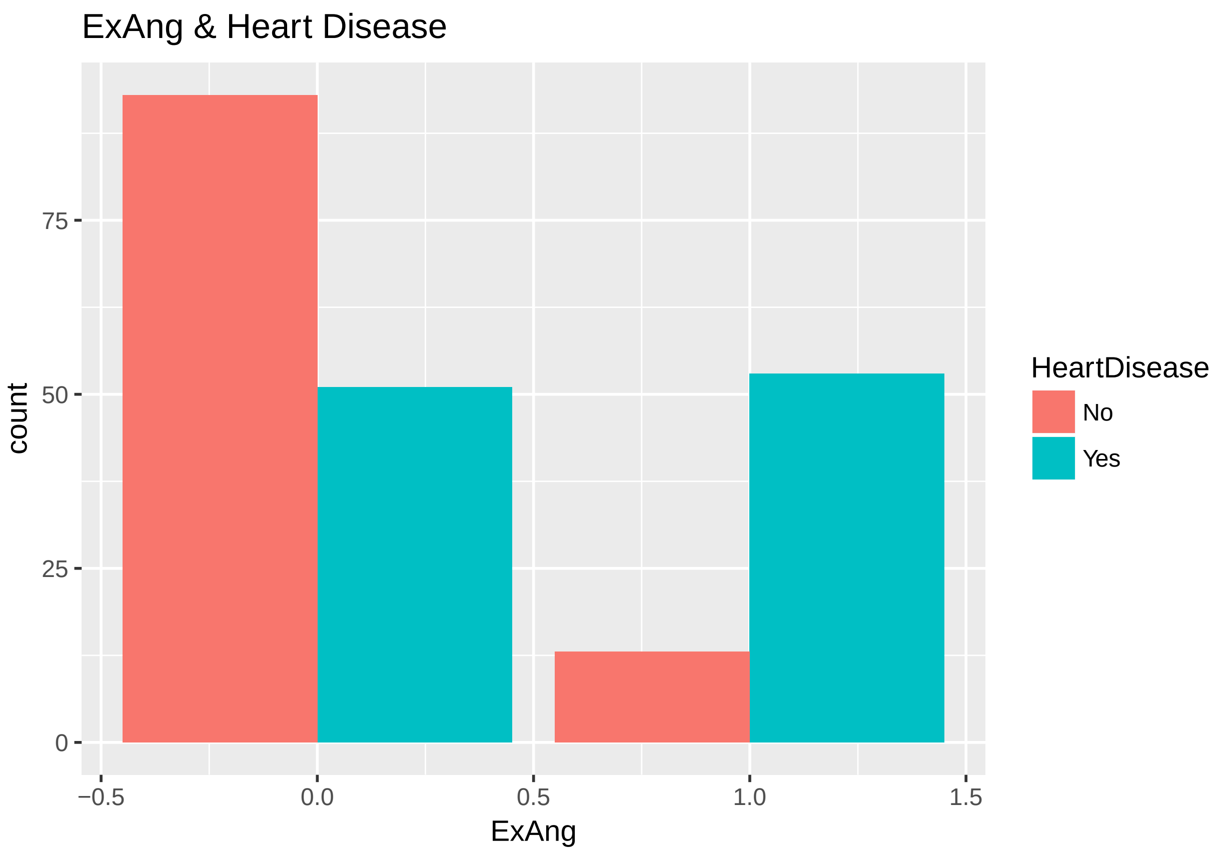
```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



Higher values of resting BP appear to be associated with heart disease

ExAng

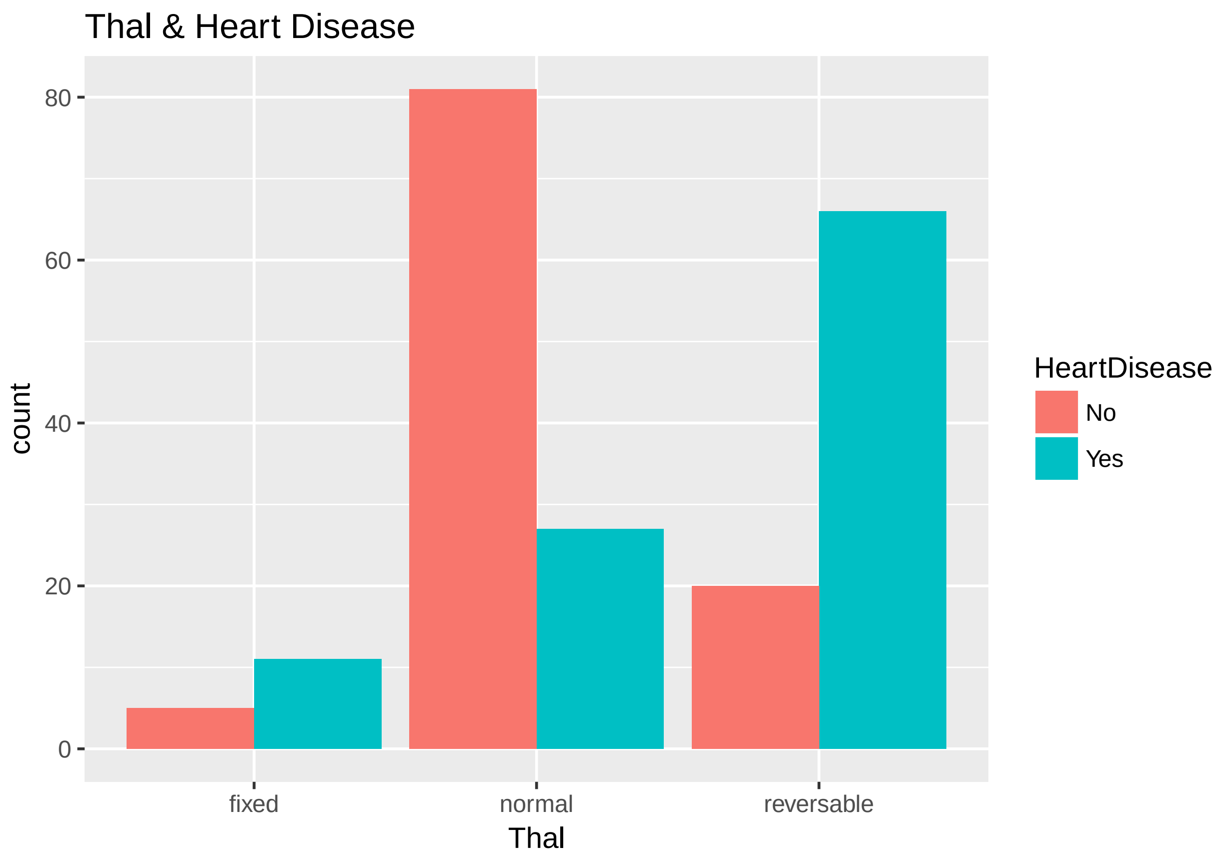
```
# Plot and visualize ExAng
ggplot(train_dataset1, mapping = aes(x = ExAng, fill = HeartDisease)) + geom_bar(position = "dodge") + g
```



People who have ExAng appeart to have a much higher incidence of heart disease

Thal

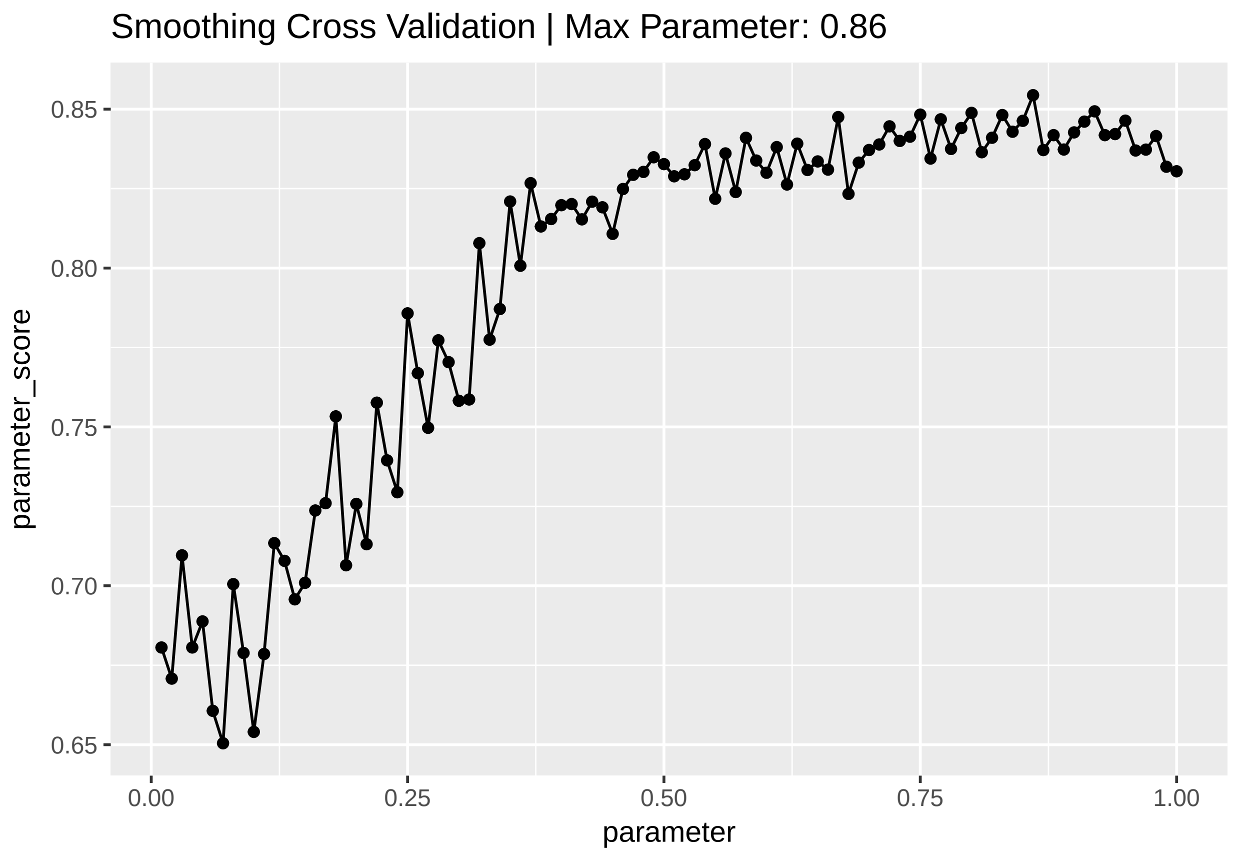
```
# Plot and visualize Thal
ggplot(train_dataset1, mapping = aes(x = Thal, fill = HeartDisease)) + geom_bar(position = "dodge") + gg
```



People with both fixed and particularly reversable Thal have much incidences of heart disease

GAM Model

```
# Tune smoothing parameter through k-fold cross validation
spar_to_validate <- c(1.:100.)/100.
gam_cv_scores <- cv_accuracy(train_dataset1, spar_to_validate, 5)
spar_gam_max_parameter <- spar_to_validate[which.max(gam_cv_scores)]
ggplot(data = data.frame(parameter = spar_to_validate, parameter_score = gam_cv_scores), aes(x = parame
```



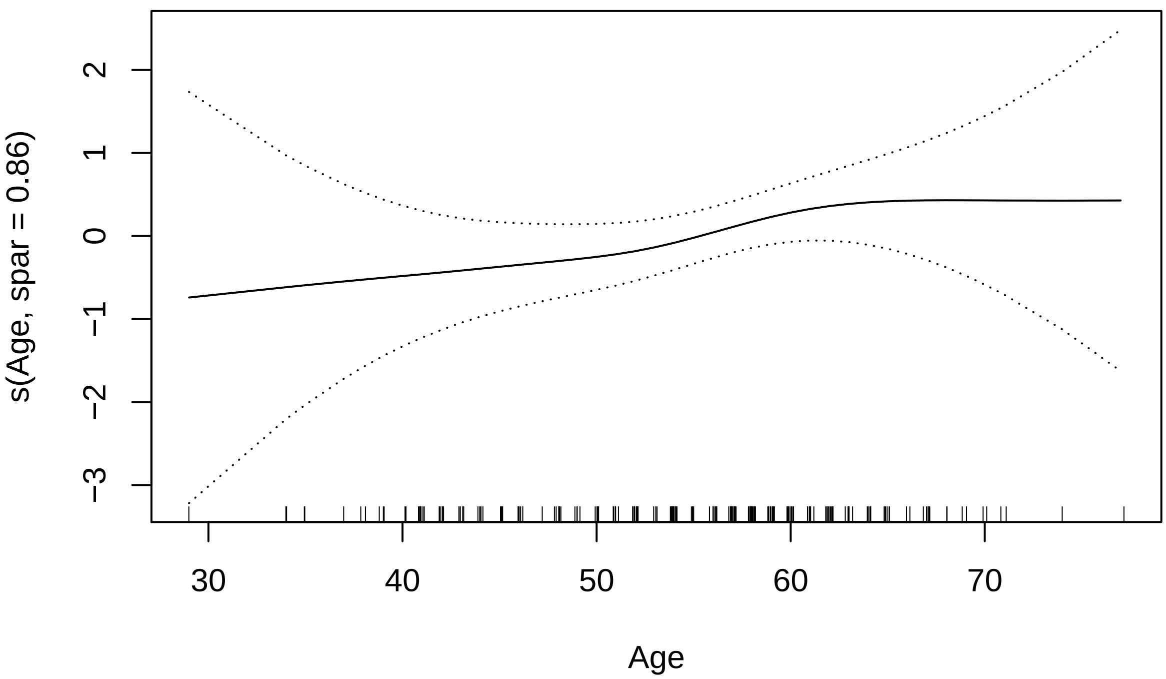
```
# Fit gam model with tuned smoothing parameter
gam_formula <- as.formula(sprintf("HeartDisease ~ s(Age, spar=%1$f) + Sex + s(RestBP,spar=%1$f) + ExAng

model.gams <- gam(gam_formula, family = binomial(link = "logit"), data = train_dataset1)

# Predict classification on test set and print the model's accuracy
pred.gams <- predict(model.gams, newdata = test_dataset1, type = "response")
cm.gams <- as.matrix(table(Actual = test_dataset1$HeartDisease, Predicted = pred.gams>.5))
accuracy.gams <- sum(diag(cm.gams))/ sum(cm.gams)

print(sprintf("GAM Model Classification Accuracy: %.4f", accuracy.gams))
```

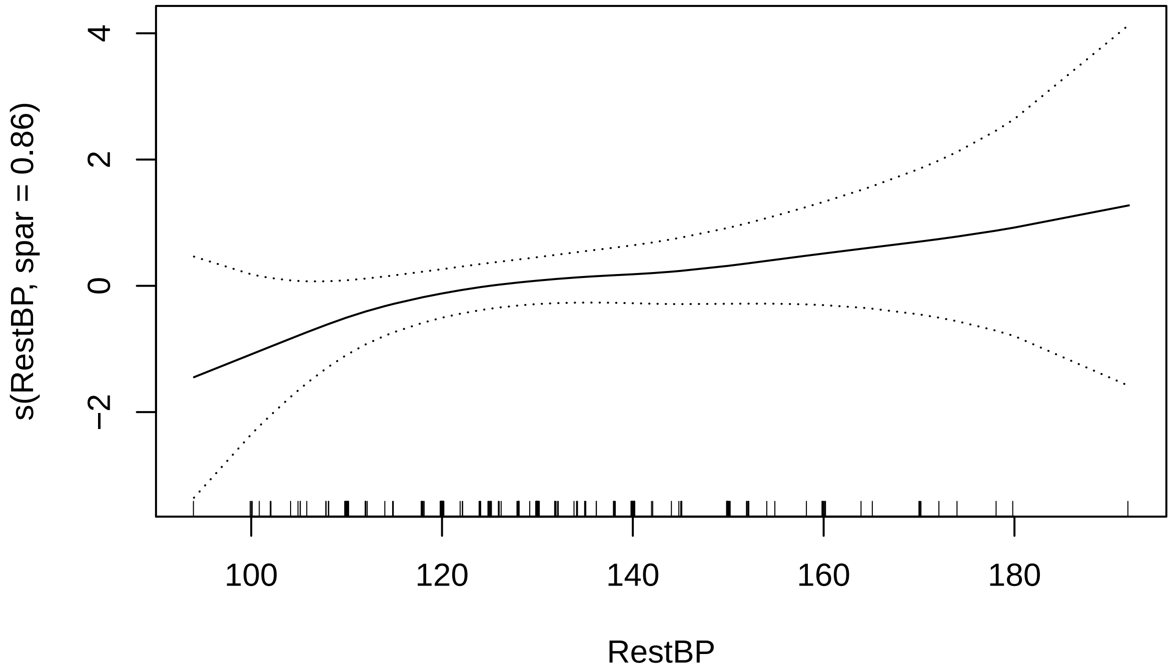
```
## [1] "GAM Model Classification Accuracy: 0.8242"
```

- The smoothing spline basis cannot be applied to categorical predictors because the weighted average would produce either the predictors actual value or some meaningless value between predictors. For instance, if 0 represents male and 1 represents female, and a weighted average produced .8, this would not be useful as it doesn't correspond to either male or female.
- R appears to one-hot encode categorical predictors automatically whereas this needed to be done manually in sklearn in Python

Predictor Smoothing

```
# Plot the smoothing parameters of the GAM model
plot.gam(model.gams, se = TRUE, terms = c(sprintf("s(Age, spar = %.2f)",spar_gam_max_parameter), sprint
```


s(Age, spar = 0.86)
2
1
0
−1
−2
−3
30
40
50
60
70
Age



There doesn't appear to be any benefit smoothing the splines due to the large standard errors for non-zero coefficient values.

Likelihood Tests

(i) GAM with only Intercept

```
# Fit the GAM model with only an intercept and compare to the original GAM model
model.gami <- gam(HeartDisease ~ 1, family = binomial(link = "logit"), data = train_dataset1)
print(anova(model.gams, model.gami, test = "Chi"))
```

```
## Analysis of Deviance Table
##
## Model 1: HeartDisease ~ s(Age, spar = 0.86) + Sex + s(RestBP, spar = 0.86) +
##     ExAng + ChestPain + Thal
## Model 2: HeartDisease ~ 1
##   Resid. Df Resid. Dev     Df Deviance  Pr(>Chi)
## 1    197.03     178.69
## 2    209.00     291.10 -11.97  -112.41 < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

The gam model that includes all predictors with smoothing on the continuous ones performs better than the gam model with only the intercept at a significance level of 0.001.

(ii) GAM with Categorical

```
# Fit the GAM model with only categorical predictors and compare to the original GAM model
model.gamcat <- gam(HeartDisease ~ Sex + ChestPain + ExAng + Thal, family = binomial(link = "logit"), da
print(anova(model.gams, model.gamcat, test = "Chi"))
```

```
## Analysis of Deviance Table
##
## Model 1: HeartDisease ~ s(Age, spar = 0.86) + Sex + s(RestBP, spar = 0.86) +
##     ExAng + ChestPain + Thal
## Model 2: HeartDisease ~ Sex + ChestPain + ExAng + Thal
##   Resid. Df Resid. Dev      Df Deviance Pr(>Chi)
## 1    197.03     178.69
## 2    202.00     189.75 -4.9696   -11.06   0.0493 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

The gam model that includes all predictors with smoothing on the continuous ones performs better than the gam model with only categorical predictors at a significance level of 0.05.

(iii) GAM with Linear Predictors

```
# Fit the GAM model with all predictors entered linearly and compare to the original GAM model
model.gamlin <- gam(HeartDisease ~ Age + Sex + ChestPain + RestBP + ExAng + Thal, family = binomial(lin
print(anova(model.gams, model.gamlin, test = "Chi"))
```

```
## Analysis of Deviance Table
##
## Model 1: HeartDisease ~ s(Age, spar = 0.86) + Sex + s(RestBP, spar = 0.86) +
##     ExAng + ChestPain + Thal
## Model 2: HeartDisease ~ Age + Sex + ChestPain + RestBP + ExAng + Thal
##   Resid. Df Resid. Dev      Df Deviance Pr(>Chi)
## 1    197.03     178.69
## 2    200.00     181.62 -2.9696  -2.9247   0.3981
```

The gam model that includes all predictors with smoothing on the continuous ones performs better than the gam model with all predictors entered linearly, however the improvement is not statistically significant.