Data Science II: Final Project- RMD and Knitted Files

Read in data:

```
flu = read.csv("severe_flu.csv")
head(flu)
```

```
##   id age gender race smoking height weight  bmi diabetes hypertension SBP LDL
## 1  1  59      0    1       1  162.7   73.2 27.6        0            0 120  95
## 2  2  54      1    1       1  169.9   73.6 25.5        1            1 133  87
## 3  3  55      1    3       1  175.4   86.3 28.1        0            0 123 139
## 4  4  59      0    1       0  169.5   77.3 26.9        0            0 121 126
## 5  5  62      1    1       0  168.7   84.9 29.8        1            0 122 107
## 6  6  64      1    1       0  170.2   75.7 26.1        0            1 132  99
##   severe_flu
## 1          0
## 2          0
## 3          0
## 4          1
## 5          1
## 6          0
```

Libraries:

```
library(caret)
```

```
## Loading required package: ggplot2
```

```
## Loading required package: lattice
```

```
library(tidymodels)
```

```
## -- Attaching packages --------------------------------------- tidymodels 1.3.0 --
```

```
## v broom        1.0.7     v rsample      1.2.1
## v dials        1.4.0     v tibble       3.2.1
## v dplyr        1.1.4     v tidyr        1.3.1
## v infer        1.0.7     v tune         1.3.0
## v modeldata    1.4.0     v workflows    1.2.0
## v parsnip      1.3.0     v workflowsets 1.1.0
## v purrr        1.0.4     v yardstick    1.3.2
## v recipes      1.1.1
```

```
## -- Conflicts ---------------------------------------- tidymodels_conflicts() --
## x purrr::discard()         masks scales::discard()
## x dplyr::filter()          masks stats::filter()
## x dplyr::lag()             masks stats::lag()
## x purrr::lift()            masks caret::lift()
## x yardstick::precision()   masks caret::precision()
## x yardstick::recall()      masks caret::recall()
## x yardstick::sensitivity() masks caret::sensitivity()
## x yardstick::specificity() masks caret::specificity()
## x recipes::step()          masks stats::step()
```

```
library(splines)
library(mgcv)
```

```
## Loading required package: nlme

##
## Attaching package: 'nlme'

## The following object is masked from 'package:dplyr':
##
##     collapse

## This is mgcv 1.9-1. For overview type 'help("mgcv-package")'.
```

```
library(pdp)
```

```
##
## Attaching package: 'pdp'

## The following object is masked from 'package:purrr':
##
##     partial
```

```
library(earth)
```

```
## Loading required package: Formula

## Loading required package: plotmo

## Loading required package: plotrix

##
## Attaching package: 'plotrix'

## The following object is masked from 'package:scales':
##
##     rescale
```

```r
library(tidyverse)
```

```
## -- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 --
## v forcats   1.0.0     v readr     2.1.5
## v lubridate 1.9.3     v stringr   1.5.1

## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x readr::col_factor() masks scales::col_factor()
## x nlme::collapse()    masks dplyr::collapse()
## x purrr::discard()    masks scales::discard()
## x dplyr::filter()     masks stats::filter()
## x stringr::fixed()    masks recipes::fixed()
## x dplyr::lag()        masks stats::lag()
## x purrr::lift()       masks caret::lift()
## x pdp::partial()      masks purrr::partial()
## x readr::spec()       masks yardstick::spec()
## i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become errors
```

```r
library(ggplot2)
library(bayesQR)
library(dplyr)
library(ISLR)
library(mlbench)
library(randomForest)
```

```
## randomForest 4.7-1.2
## 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
```

```r
library(ranger)
```

```
##
## Attaching package: 'ranger'
##
## The following object is masked from 'package:randomForest':
##
##     importance
```

```r
library(gbm)
```

```
## Loaded gbm 2.2.2
## This version of gbm is no longer under development. Consider transitioning to gbm3, https://github.co
```

```r
library(rpart)
```

```
##
## Attaching package: 'rpart'
##
## The following object is masked from 'package:dials':
##
##     prune
```

```r
library(rpart.plot)
library(party)
```

```
## Loading required package: grid
## Loading required package: mvtnorm
## Loading required package: modeltools
## Loading required package: stats4
##
## Attaching package: 'modeltools'
##
## The following object is masked from 'package:bayesQR':
##
##     prior
##
## The following object is masked from 'package:workflows':
##
##     fit
##
## The following object is masked from 'package:tune':
##
##     parameters
##
## The following object is masked from 'package:parsnip':
##
##     fit
##
## The following object is masked from 'package:infer':
##
##     fit
##
## The following object is masked from 'package:dials':
##
##     parameters
##
## Loading required package: strucchange
## Loading required package: zoo
##
## Attaching package: 'zoo'
##
## The following objects are masked from 'package:base':
##
##     as.Date, as.Date.numeric
##
```

```
## Loading required package: sandwich
##
## Attaching package: 'strucchange'
##
## The following object is masked from 'package:stringr':
##
##     boundary
##
##
## Attaching package: 'party'
##
## The following object is masked from 'package:dplyr':
##
##     where
```

```
library(partykit)
```

```
## Loading required package: libcoin
##
## Attaching package: 'partykit'
##
## The following objects are masked from 'package:party':
##
##     cforest, ctree, ctree_control, edge_simple, mob, mob_control,
##     node_barplot, node_bivplot, node_boxplot, node_inner, node_surv,
##     node_terminal, varimp
```

```
library(pROC)
```

```
## Type 'citation("pROC")' for a citation.
##
## Attaching package: 'pROC'
##
## The following objects are masked from 'package:stats':
##
##     cov, smooth, var
```

```
library(e1071)
```

```
##
## Attaching package: 'e1071'
##
## The following object is masked from 'package:tune':
##
##     tune
##
## The following object is masked from 'package:rsample':
##
##     permutations
##
## The following object is masked from 'package:parsnip':
##
##     tune
```

```r
library(kernlab)
```

```
##
## Attaching package: 'kernlab'
##
## The following object is masked from 'package:modeltools':
##
##     prior
##
## The following object is masked from 'package:bayesQR':
##
##     prior
##
## The following object is masked from 'package:purrr':
##
##     cross
##
## The following object is masked from 'package:dials':
##
##     buffer
##
## The following object is masked from 'package:scales':
##
##     alpha
##
## The following object is masked from 'package:ggplot2':
##
##     alpha
```

```r
library(ggrepel)
library(corrplot)
```

```
## corrplot 0.95 loaded
```

```r
library(plotmo)
```

Factors

```r
flu <- flu %>%
  mutate(
    #gender = factor(gender, levels = c(0, 1), labels = c("Female", "Male")),
    #race = factor(race, levels = c(1, 2, 3, 4), labels = c("White", "Asian", "Black", "Hispanic")),
    #smoking = factor(smoking, levels = c(0, 1, 2), labels = c("Never smoked", "Former smoker", "Curren
    #diabetes = factor(diabetes, levels = c(0, 1), labels = c("No", "Yes")),
    #hypertension = factor(hypertension, levels = c(0, 1), labels = c("No", "Yes")),
    severe_flu = factor(severe_flu, levels = c(0, 1), labels = c("No", "Yes"))
  )
```

Exploratory analysis:

```
# observe first couple of rows
head(flu)
```

```
##   id age gender race smoking height weight  bmi diabetes hypertension SBP LDL
## 1  1  59      0    1       1  162.7   73.2 27.6        0            0 120  95
## 2  2  54      1    1       1  169.9   73.6 25.5        1            1 133  87
## 3  3  55      1    3       1  175.4   86.3 28.1        0            0 123 139
## 4  4  59      0    1       0  169.5   77.3 26.9        0            0 121 126
## 5  5  62      1    1       0  168.7   84.9 29.8        1            0 122 107
## 6  6  64      1    1       0  170.2   75.7 26.1        0            1 132  99
##   severe_flu
## 1         No
## 2         No
## 3         No
## 4        Yes
## 5        Yes
## 6         No
```

```
str(flu)
```

```
## 'data.frame':    1000 obs. of  13 variables:
##  $ id          : int  1 2 3 4 5 6 7 8 9 10 ...
##  $ age         : int  59 54 55 59 62 64 64 62 67 66 ...
##  $ gender      : int  0 1 1 0 1 1 0 1 0 1 ...
##  $ race        : int  1 1 3 1 1 1 1 4 1 1 ...
##  $ smoking     : int  1 1 1 0 0 0 0 1 0 0 ...
##  $ height      : num  163 170 175 170 169 ...
##  $ weight      : num  73.2 73.6 86.3 77.3 84.9 75.7 89.2 81.9 68.3 76.3 ...
##  $ bmi         : num  27.6 25.5 28.1 26.9 29.8 26.1 28.9 27.7 24.3 25.5 ...
##  $ diabetes    : int  0 1 0 0 1 0 0 0 0 0 ...
##  $ hypertension: int  0 1 0 0 0 1 0 0 1 1 ...
##  $ SBP         : int  120 133 123 121 122 132 122 119 138 135 ...
##  $ LDL         : int  95 87 139 126 107 99 99 123 97 111 ...
##  $ severe_flu  : Factor w/ 2 levels "No","Yes": 1 1 1 2 2 1 1 2 1 1 ...
```

```
summary(flu)
```

```
##        id              age            gender           race      
##  Min.   :   1.0   Min.   :46.00   Min.   :0.000   Min.   :1.00  
##  1st Qu.: 250.8   1st Qu.:57.00   1st Qu.:0.000   1st Qu.:1.00  
##  Median : 500.5   Median :60.00   Median :0.000   Median :1.00  
##  Mean   : 500.5   Mean   :60.08   Mean   :0.478   Mean   :1.72  
##  3rd Qu.: 750.2   3rd Qu.:63.00   3rd Qu.:1.000   3rd Qu.:3.00  
##  Max.   :1000.0   Max.   :72.00   Max.   :1.000   Max.   :4.00  
##     smoking          height          weight           bmi       
##  Min.   :0.000   Min.   :151.5   Min.   : 59.10   Min.   :20.10  
##  1st Qu.:0.000   1st Qu.:165.2   1st Qu.: 75.10   1st Qu.:25.90  
##  Median :0.000   Median :169.7   Median : 80.10   Median :27.70  
##  Mean   :0.519   Mean   :169.7   Mean   : 80.03   Mean   :27.86  
```

```
##  3rd Qu.:1.000   3rd Qu.:174.0   3rd Qu.: 84.80   3rd Qu.:29.60
##  Max.   :2.000   Max.   :191.9   Max.   :103.70   Max.   :36.70
##     diabetes      hypertension        SBP             LDL        severe_flu
##  Min.   :0.000   Min.   :0.000   Min.   :108.0   Min.   : 41.0   No :747
##  1st Qu.:0.000   1st Qu.:0.000   1st Qu.:124.0   1st Qu.: 98.0   Yes:253
##  Median :0.000   Median :0.000   Median :130.0   Median :111.0
##  Mean   :0.145   Mean   :0.464   Mean   :129.9   Mean   :110.5
##  3rd Qu.:0.000   3rd Qu.:1.000   3rd Qu.:135.0   3rd Qu.:123.0
##  Max.   :1.000   Max.   :1.000   Max.   :154.0   Max.   :174.0
```

```r
#checking for missing
colSums(is.na(flu))
```

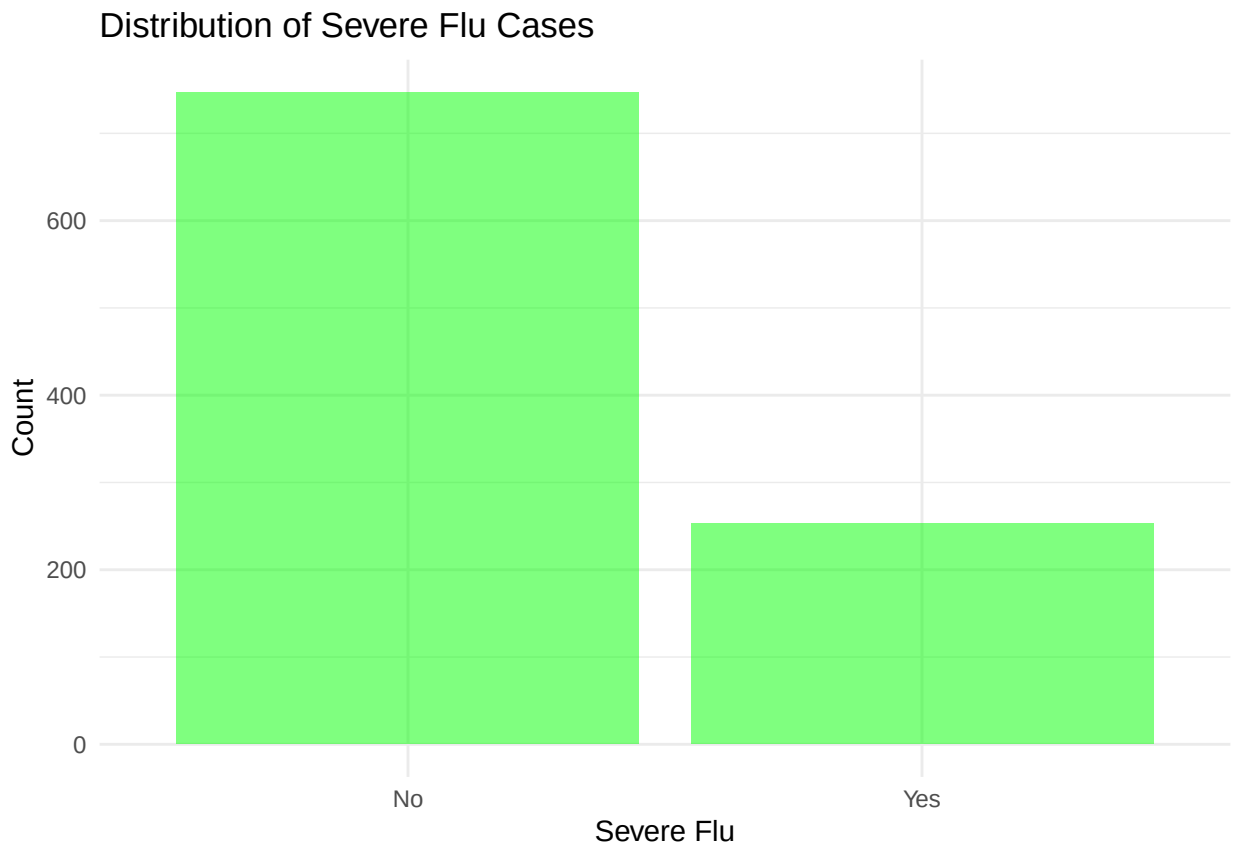
```
##           id          age       gender         race      smoking       height
##            0            0            0            0            0            0
##       weight          bmi     diabetes hypertension          SBP          LDL
##            0            0            0            0            0            0
##   severe_flu
##            0
```

```r
# checking for duplicates
sum(duplicated(flu))
```

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

Bar Plot for distribution of severe flu

```r
ggplot(flu, aes(x = severe_flu)) +
  geom_bar(fill = "green", alpha = 0.5) +
  theme_minimal() +
  labs(title = "Distribution of Severe Flu Cases",
       x = "Severe Flu",
       y = "Count")
```



Summarizing continous variables

```
summary(flu[, c("age", "height", "weight", "bmi", "SBP", "LDL")])
```

```
##       age             height          weight           bmi
##  Min.   :46.00   Min.   :151.5   Min.   : 59.10   Min.   :20.10
##  1st Qu.:57.00   1st Qu.:165.2   1st Qu.: 75.10   1st Qu.:25.90
##  Median :60.00   Median :169.7   Median : 80.10   Median :27.70
##  Mean   :60.08   Mean   :169.7   Mean   : 80.03   Mean   :27.86
##  3rd Qu.:63.00   3rd Qu.:174.0   3rd Qu.: 84.80   3rd Qu.:29.60
##  Max.   :72.00   Max.   :191.9   Max.   :103.70   Max.   :36.70
##       SBP             LDL
##  Min.   :108.0   Min.   : 41.0
##  1st Qu.:124.0   1st Qu.: 98.0
##  Median :130.0   Median :111.0
##  Mean   :129.9   Mean   :110.5
##  3rd Qu.:135.0   3rd Qu.:123.0
##  Max.   :154.0   Max.   :174.0
```

Summarizing categorical variables distribution among those with severe fu vs. without severe flu

```
# For gender
flu %>%
  group_by(severe_flu, gender) %>%
  summarize(count = n(), .groups = "drop") %>%
  group_by(severe_flu) %>%
  mutate(proportion = count / sum(count))
```

```
## # A tibble: 4 x 4
## # Groups:   severe_flu [2]
##   severe_flu gender count proportion
##   <fct>       <int> <int>      <dbl>
## 1 No              0   400      0.535
## 2 No              1   347      0.465
## 3 Yes             0   122      0.482
## 4 Yes             1   131      0.518
```

```
# For race
flu %>%
  group_by(severe_flu, race) %>%
  summarize(count = n(), .groups = "drop") %>%
  group_by(severe_flu) %>%
  mutate(proportion = count / sum(count))
```

```
## # A tibble: 8 x 4
## # Groups:   severe_flu [2]
##   severe_flu  race count proportion
##   <fct>      <int> <int>      <dbl>
## 1 No             1   492     0.659
## 2 No             2    48     0.0643
## 3 No             3   143     0.191
## 4 No             4    64     0.0857
## 5 Yes            1   164     0.648
## 6 Yes            2    16     0.0632
## 7 Yes            3    41     0.162
## 8 Yes            4    32     0.126
```

```
# For smoking
flu %>%
  group_by(severe_flu, smoking) %>%
  summarize(count = n(), .groups = "drop") %>%
  group_by(severe_flu) %>%
  mutate(proportion = count / sum(count))
```

```
## # A tibble: 6 x 4
## # Groups:   severe_flu [2]
##   severe_flu smoking count proportion
##   <fct>        <int> <int>      <dbl>
## 1 No               0   439     0.588
```

```
## 2 No               1   243     0.325
## 3 No               2    65     0.0870
## 4 Yes              0   145     0.573
## 5 Yes              1    70     0.277
## 6 Yes              2    38     0.150
```

```
# For diabetes
flu %>%
  group_by(severe_flu, diabetes) %>%
  summarize(count = n(), .groups = "drop") %>%
  group_by(severe_flu) %>%
  mutate(proportion = count / sum(count))
```

```
## # A tibble: 4 x 4
## # Groups:   severe_flu [2]
##   severe_flu diabetes count proportion
##   <fct>         <int> <int>      <dbl>
## 1 No                0   654      0.876
## 2 No                1    93      0.124
## 3 Yes               0   201      0.794
## 4 Yes               1    52      0.206
```

```
# For hypertension
flu %>%
  group_by(severe_flu, hypertension) %>%
  summarize(count = n(), .groups = "drop") %>%
  group_by(severe_flu) %>%
  mutate(proportion = count / sum(count))
```

```
## # A tibble: 4 x 4
## # Groups:   severe_flu [2]
##   severe_flu hypertension count proportion
##   <fct>             <int> <int>      <dbl>
## 1 No                    0   409      0.548
## 2 No                    1   338      0.452
## 3 Yes                   0   127      0.502
## 4 Yes                   1   126      0.498
```

Assessing correlation among continuous variables

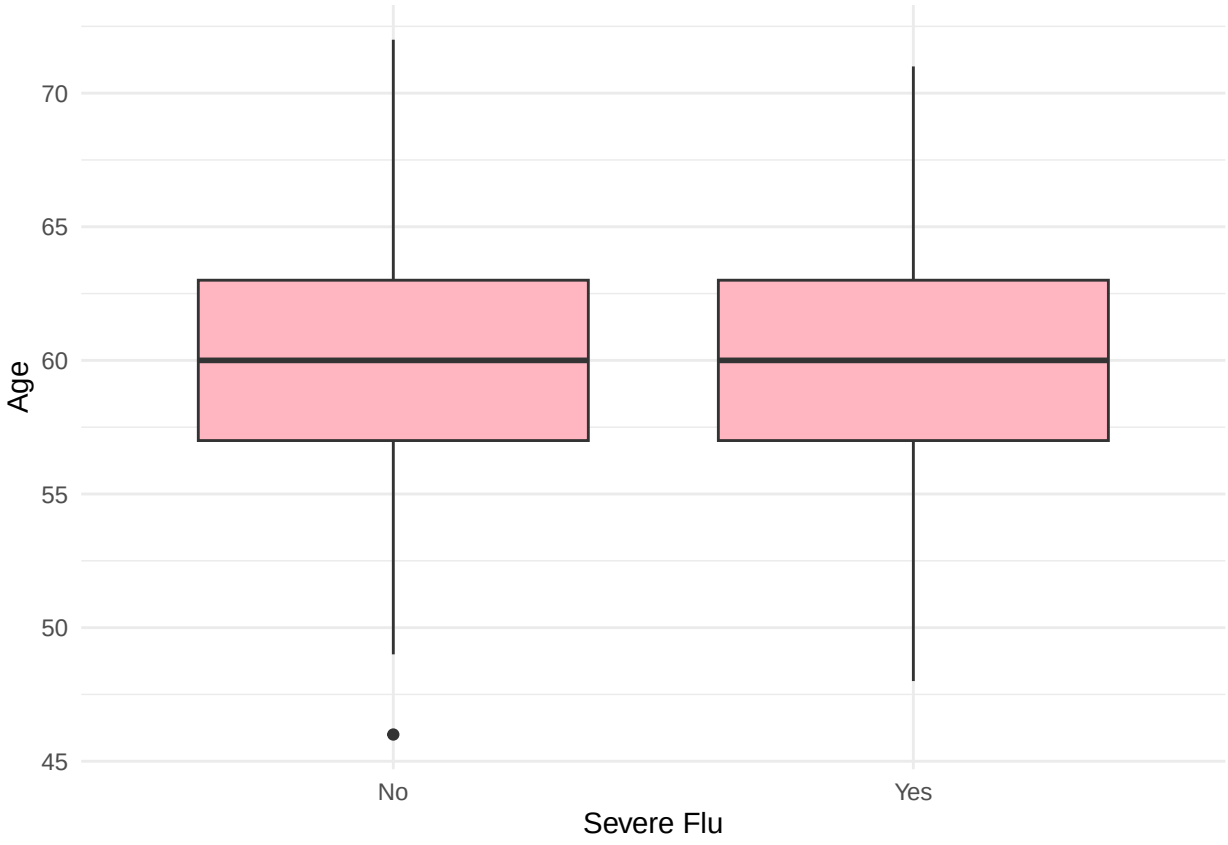
```
cor(flu[, c("age", "height", "weight", "bmi", "SBP", "LDL")])
```

```
##                age      height       weight          bmi          SBP
## age     1.00000000  0.01794456 -0.029940909 -0.04158528  0.44027512
## height  0.01794456  1.00000000  0.267637393 -0.48933944  0.03143295
## weight -0.02994091  0.26763739  1.000000000  0.70666669 -0.01929686
## bmi    -0.04158528 -0.48933944  0.706666689  1.00000000 -0.04120141
## SBP     0.44027512  0.03143295 -0.019296863 -0.04120141  1.00000000
## LDL     0.20742590  0.01832110 -0.001534474 -0.01566285  0.24444416
##                LDL
```

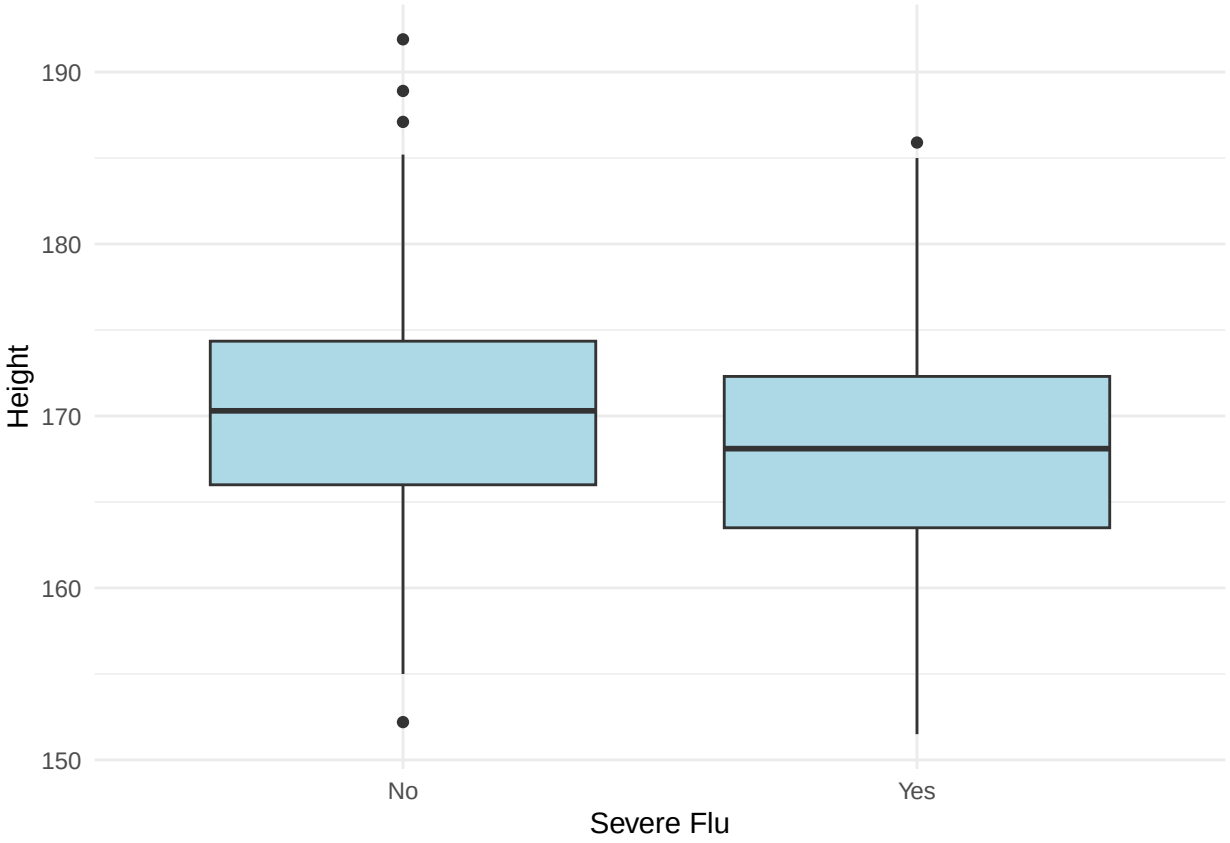
```
## age     0.207425901
## height  0.018321100
## weight -0.001534474
## bmi    -0.015662850
## SBP     0.244444156
## LDL     1.000000000
```

Asses relationship between severe flu and continous variables

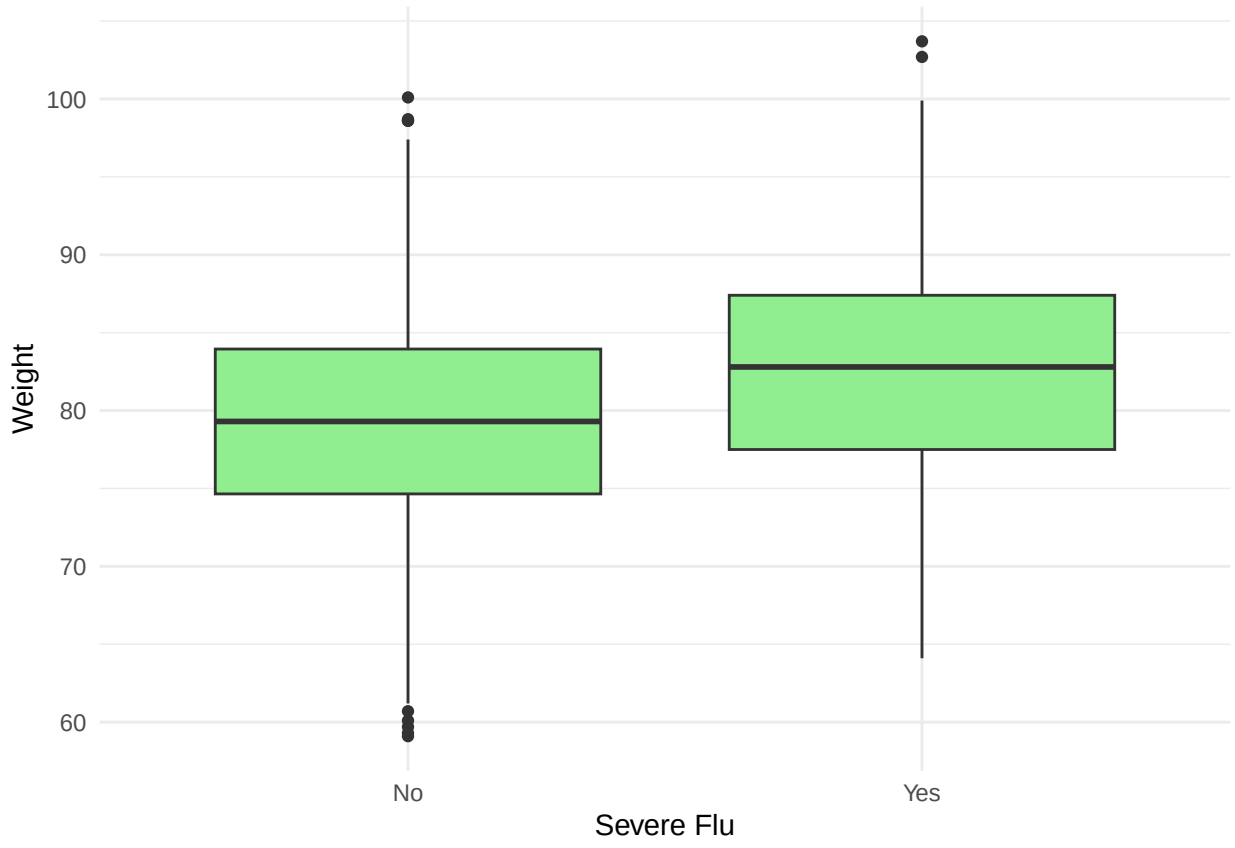
```
ggplot(flu, aes(x = severe_flu, y = age)) +
  geom_boxplot(fill = "lightpink") +
  theme_minimal() +
  labs(x = "Severe Flu", y = "Age")
```



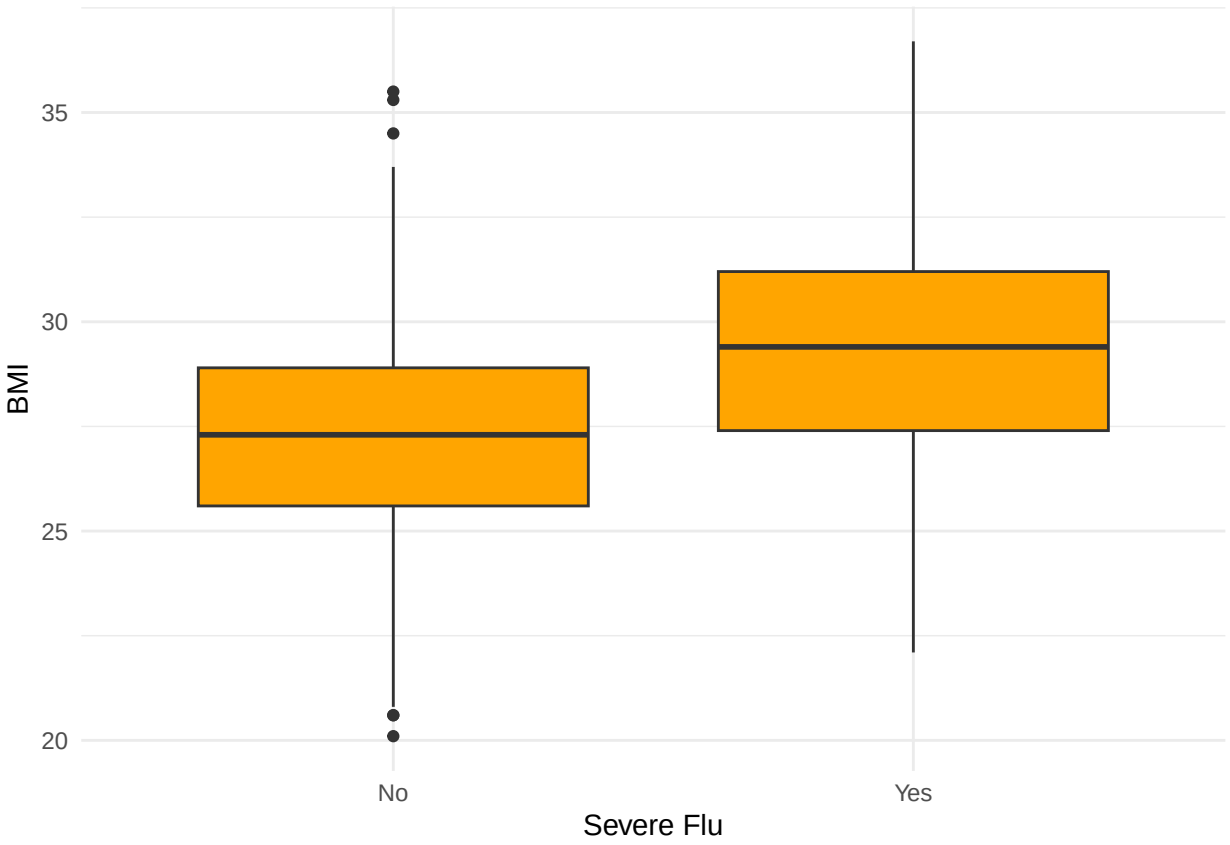

```
ggplot(flu, aes(x = severe_flu, y = height)) +
  geom_boxplot(fill = "lightblue") +
  theme_minimal() +
  labs(x = "Severe Flu", y = "Height")
```



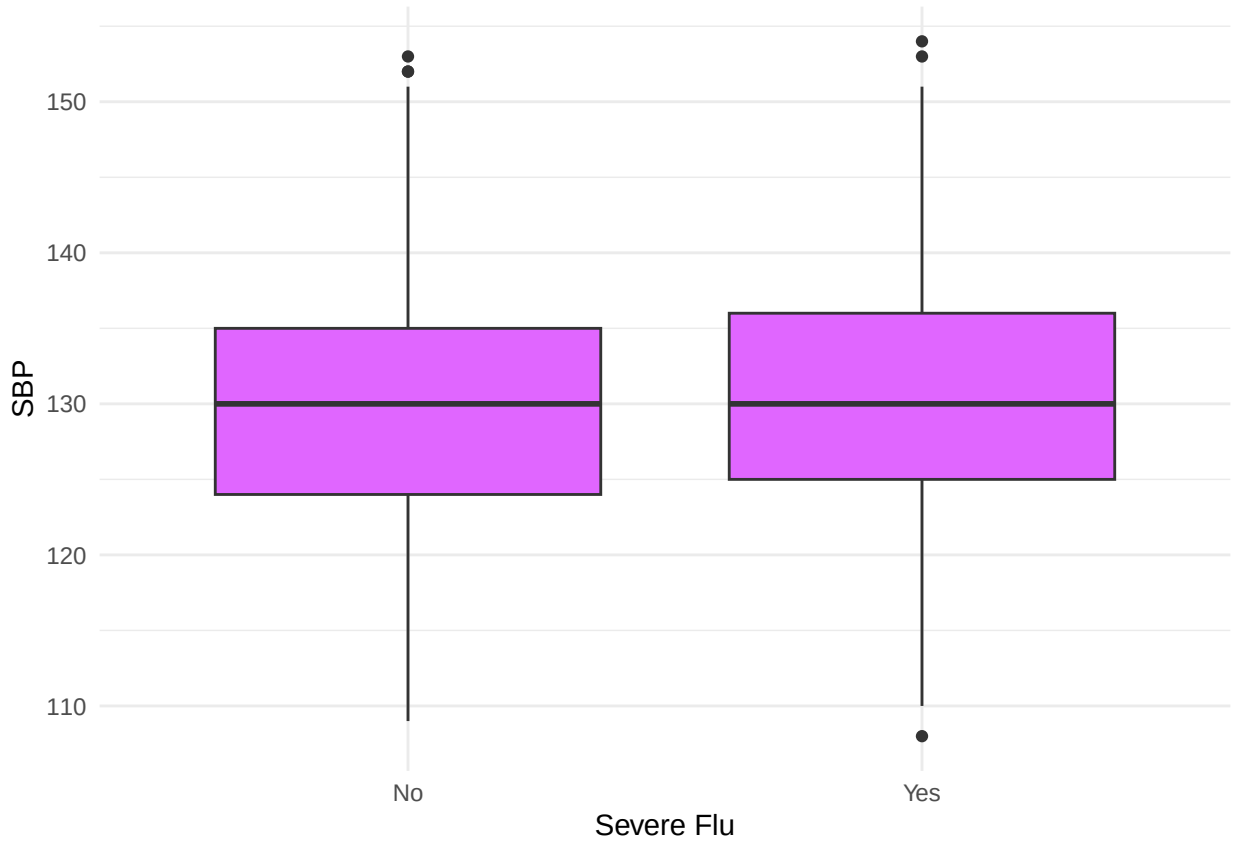

```
ggplot(flu, aes(x = severe_flu, y = weight)) +
  geom_boxplot(fill = "lightgreen") +
  theme_minimal() +
  labs(x = "Severe Flu", y = "Weight")
```



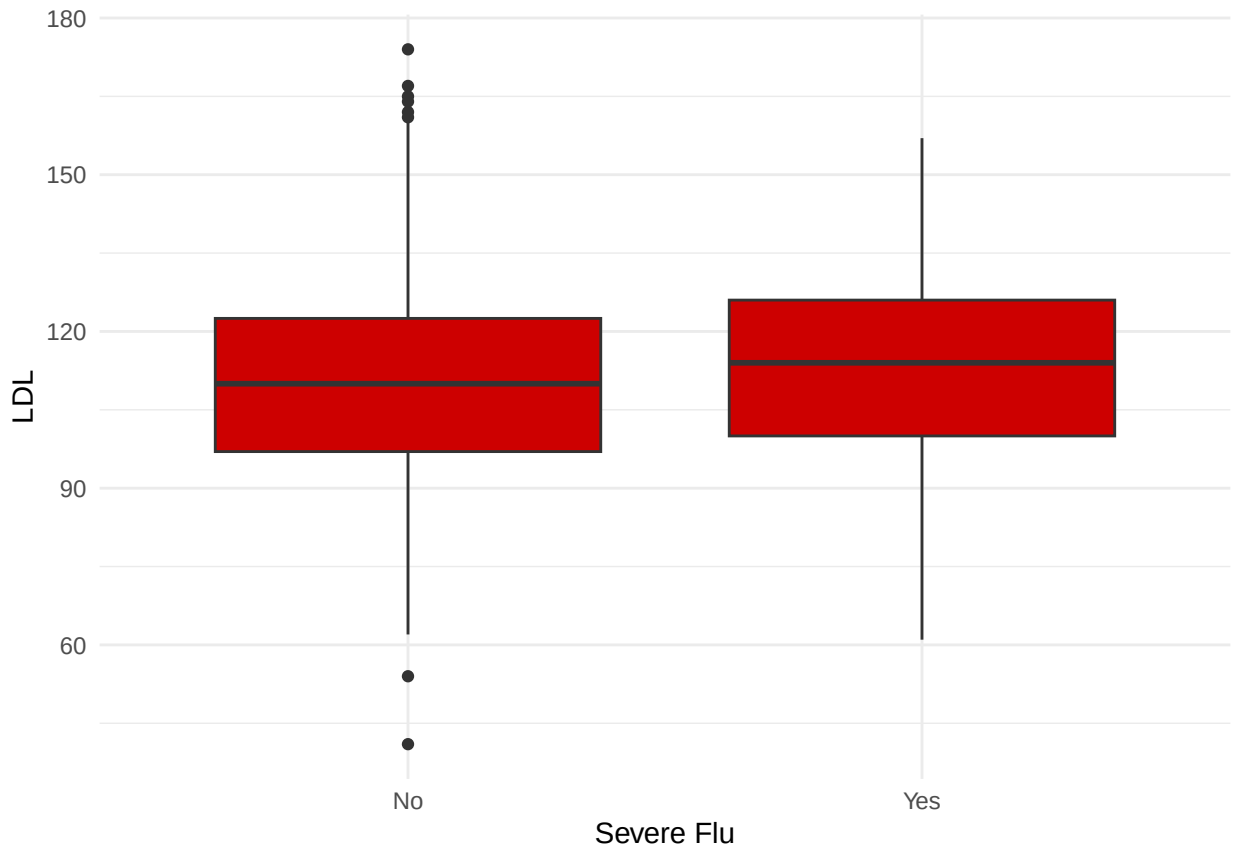

```
ggplot(flu, aes(x = severe_flu, y = bmi)) +
  geom_boxplot(fill = "orange") +
  theme_minimal() +
  labs(x = "Severe Flu", y = "BMI")
```




```
ggplot(flu, aes(x = severe_flu, y = SBP)) +
  geom_boxplot(fill = "mediumorchid1") +
  theme_minimal() +
  labs(x = "Severe Flu", y = "SBP")
```




```
ggplot(flu, aes(x = severe_flu, y = LDL)) +
  geom_boxplot(fill = "red3") +
  theme_minimal() +
  labs(x = "Severe Flu", y = "LDL")
```



Data Paritioning:

```
datSplit = initial_split(data = flu, prop = 0.8)
flu_train = training(datSplit)
flu_test = testing(datSplit)
head(flu_train)
```

```
##    id age gender race smoking height weight  bmi diabetes hypertension SBP LDL
## 1 514  57      0    1       0  164.4   81.4 30.1        0            0 125 133
## 2 520  62      1    1       0  167.8   74.6 26.5        0            1 147 103
## 3 195  56      0    1       0  169.2   72.5 25.3        0            0 117 109
## 4 758  59      0    1       0  167.0   72.3 25.9        0            0 128  78
## 5 232  67      0    3       0  164.8   78.5 28.9        1            1 132  90
## 6 443  57      1    1       0  167.8   72.4 25.7        0            0 119 107
##   severe_flu
## 1         No
## 2         No
## 3        Yes
## 4         No
## 5         No
## 6         No
```

```
head(flu_test)
```

```
##   id age gender race smoking height weight  bmi diabetes hypertension SBP LDL
## 1 11  50      1    4       1  177.7   77.5 24.5        0            0 128 132
```

```
## 2 17  61      1    1       0  162.1   83.7 31.8        0            1 138 117
## 3 35  65      0    3       0  163.4   79.9 29.9        0            1 137  73
## 4 40  64      0    1       0  177.3   79.9 25.4        0            1 133 118
## 5 42  55      0    3       1  175.2   81.3 26.5        0            0 122 114
## 6 53  62      1    4       0  170.0   76.1 26.3        0            1 133 118
##   severe_flu
## 1        Yes
## 2         No
## 3         No
## 4         No
## 5         No
## 6         No
```

Part 1: Evaluating whether boosting and SVM provide superior predictive performance compared to simpler models.

Model 1: GLM

```
set.seed(1)
#build logistic regression w/ training dataset
model_logit = glm(severe_flu ~ age + gender + race + smoking + height + weight + bmi +
               diabetes + hypertension + SBP + LDL,
             data = flu_train, family = binomial)

summary(model_logit)
```

```
##
## Call:
## glm(formula = severe_flu ~ age + gender + race + smoking + height +
##     weight + bmi + diabetes + hypertension + SBP + LDL, family = binomial,
##     data = flu_train)
##
## Coefficients:
##                Estimate Std. Error z value Pr(>|z|)
## (Intercept)  -58.383331  28.205789  -2.070  0.03846 *
## age           -0.024559   0.023089  -1.064  0.28747
## gender         0.272234   0.176702   1.541  0.12340
## race           0.004361   0.083212   0.052  0.95820
## smoking        0.338214   0.126594   2.672  0.00755 **
## height         0.297568   0.166079   1.792  0.07318 .
## weight        -0.330332   0.173022  -1.909  0.05624 .
## bmi            1.206210   0.491663   2.453  0.01415 *
## diabetes       0.634976   0.226475   2.804  0.00505 **
## hypertension   0.370369   0.295481   1.253  0.21004
## SBP           -0.003880   0.019181  -0.202  0.83971
## LDL            0.007903   0.004718   1.675  0.09390 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
```

```
##
##     Null deviance: 908.42  on 799  degrees of freedom
## Residual deviance: 793.14  on 788  degrees of freedom
## AIC: 817.14
##
## Number of Fisher Scoring iterations: 4
```

```
# Predict probabilities and eval. accuracy of predicted values
pred_probs = predict(model_logit, newdata = flu_test, type = "response")
pred_classes = ifelse(pred_probs > 0.5, 1, 0)
actuals = flu_test[["severe_flu"]]  # safer than using $
cm = table(Predicted = pred_classes, Actual = actuals)
accuracy = sum(diag(cm)) / sum(cm)
print(cm)
```

```
##          Actual
## Predicted  No Yes
##         0 146  46
##         1   5   3
```

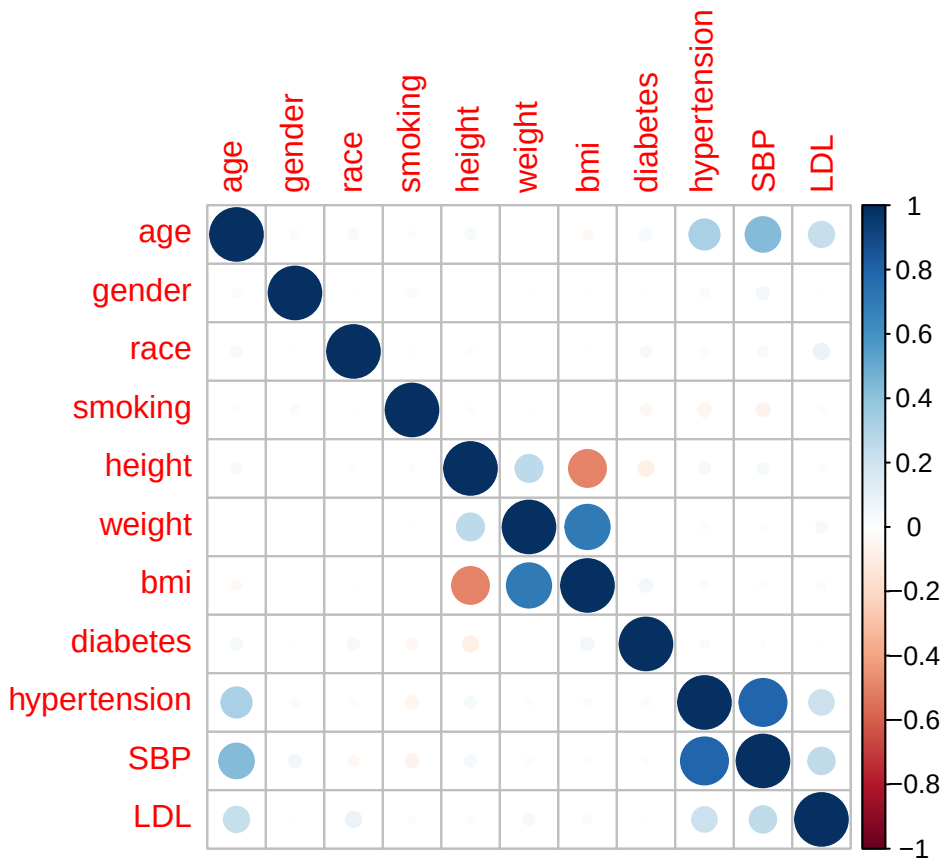
```
print(accuracy)
```

```
## [1] 0.745
```

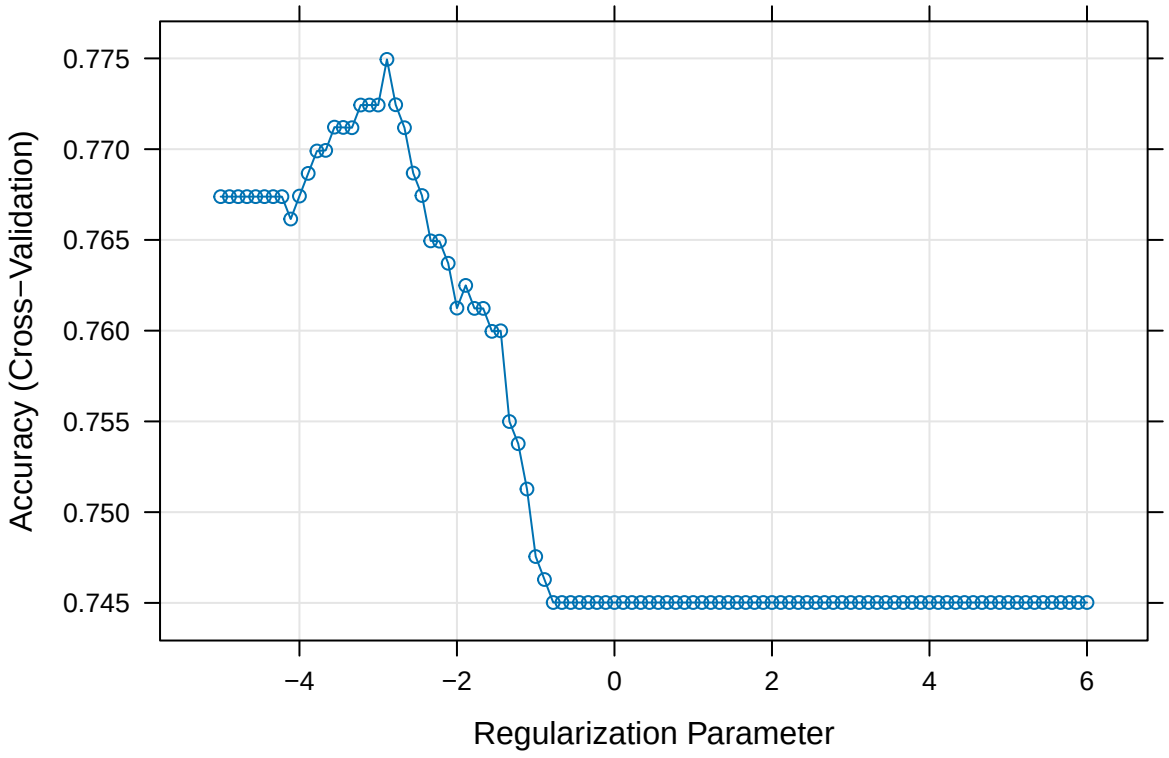
GLM has an accuracy of 0.765. Low.

Model 2: Ridge

```
x = model.matrix(severe_flu ~ . - id, flu_train)[,-1]
y = flu_train[, "severe_flu"]

corrplot(cor(x), method = "circle", type = "full")
```




```
# fitting Ridge regression using caret
ctrl1 = trainControl(method = "cv", number = 10)
set.seed(1)
ridge.fit = train(severe_flu ~ . - id,
                  data = flu_train,
                  method = "glmnet",
                  tuneGrid = expand.grid(alpha = 0,
                                         lambda = exp(seq(6, -5, length = 100))),
                  trControl = ctrl1)

plot(ridge.fit, xTrans = log)
```




```
ridge.fit$bestTune
```

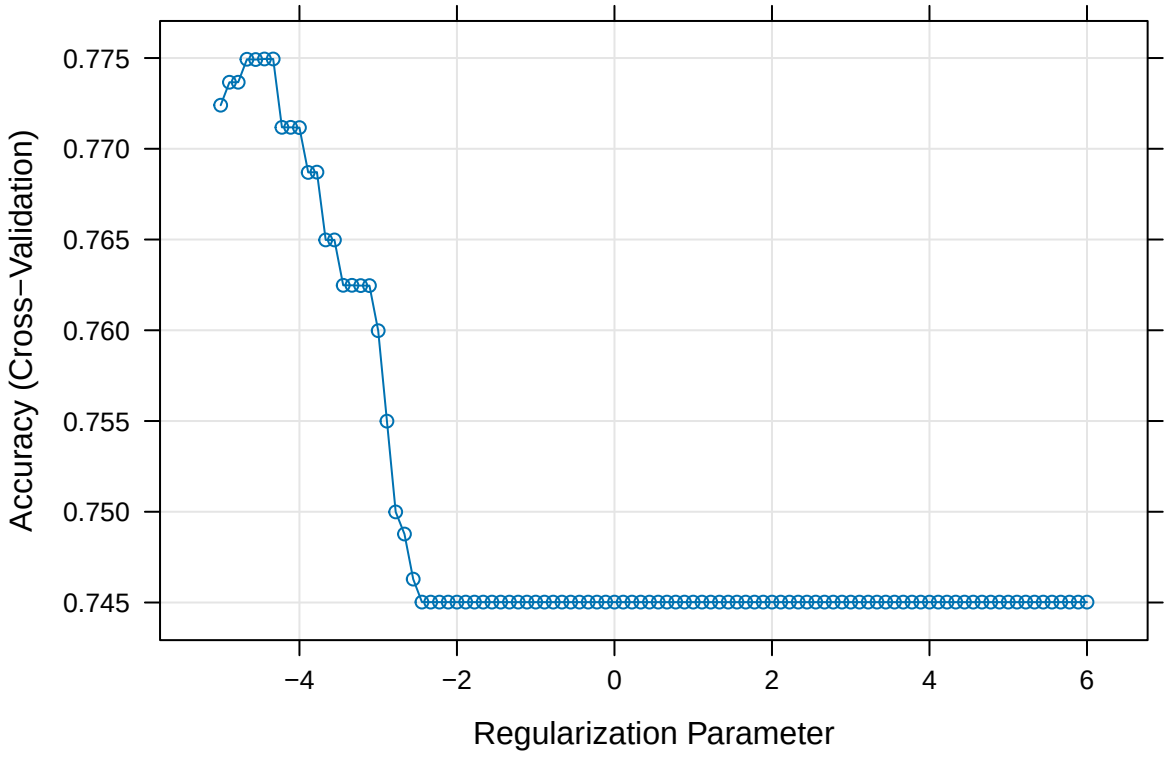
```
##    alpha   lambda
## 20     0 0.055638
```

```
coef(ridge.fit$finalModel, ridge.fit$bestTune$lambda)
```

```
## 12 x 1 sparse Matrix of class "dgCMatrix"
##                          s1
## (Intercept)  -0.735354131
## age          -0.015486774
## gender        0.182201835
## race          0.015029775
## smoking       0.231638506
## height       -0.040964161
## weight        0.027274457
## bmi           0.144368339
## diabetes      0.474760777
## hypertension  0.203960867
## SBP           0.001419630
## LDL           0.005614828
```

Model 3: Lasso

```
set.seed(1)
lasso.fit = train(severe_flu ~ . - id,
                  data = flu_train,
                  method = "glmnet",
                  tuneGrid = expand.grid(alpha = 1,
                                         lambda = exp(seq(6, -5, length = 100))),
                  trControl = ctrl1)

lasso_tune = lasso.fit$bestTune
plot(lasso.fit, xTrans = log)
```




```
lasso_coef = coef(lasso.fit$finalModel, lasso.fit$bestTune$lambda)
```

Model 4: LDA

```
ctrl3 = trainControl(method = "repeatedcv", repeats = 5,
summaryFunction = twoClassSummary,
classProbs = TRUE)

set.seed(22)
```

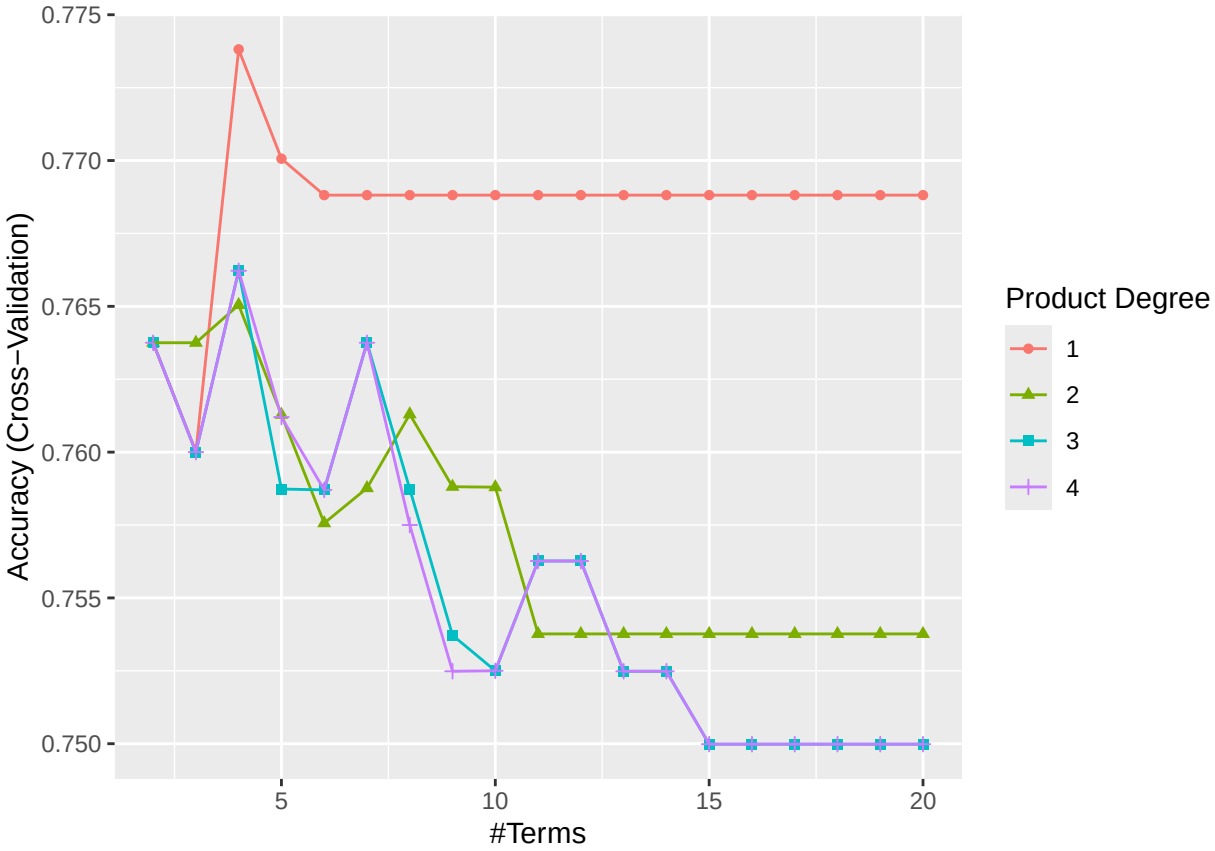
```
model.lda = train(x = flu_train[, c("age", "gender", "race", "smoking", "height", "weight", "bmi","diab
                  y = flu_train$severe_flu,
                  method = "lda",
                  metric = "ROC",
                  trControl = ctrl3)

print(model.lda)
```

```
## Linear Discriminant Analysis
##
## 800 samples
##  11 predictor
##   2 classes: 'No', 'Yes'
##
## No pre-processing
## Resampling: Cross-Validated (10 fold, repeated 5 times)
## Summary of sample sizes: 720, 721, 720, 719, 720, 721, ...
## Resampling results:
##
##   ROC       Sens      Spec
##   0.709896  0.946661  0.2634286
```

```
lda.pred2 = predict(model.lda, newdata = flu_test)
```

Model 5: MARS

```
set.seed(1)
x = model.matrix(severe_flu ~ age + gender + race + smoking + height + weight + bmi + diabetes + hyperte

y = flu_train$severe_flu
ctrl1 = trainControl(method = "cv", number = 10)
mars_grid = expand.grid(degree = 1:4,
                        nprune = 2:20)
set.seed(2)
mars.fit = train(x, y, method = "earth",
                 tuneGrid = mars_grid, trControl = ctrl1)
ggplot(mars.fit)
```




```
mars_tune = mars.fit$bestTune
mars_coef = coef(mars.fit$finalModel)

x_test = model.matrix(severe_flu ~ age + gender + race + smoking + height + weight + bmi + diabetes + hy
y_test = flu_test$severe_flu
predictions = predict(mars.fit, newdata = x_test)
mse = mean((predictions- y_test)^2)
```

```
## Warning in Ops.factor(predictions, y_test): '-' not meaningful for factors
```

Model 6: PLS

```
length(setdiff(names(flu_train), c("severe_flu", "id")))
```

```
## [1] 11
```

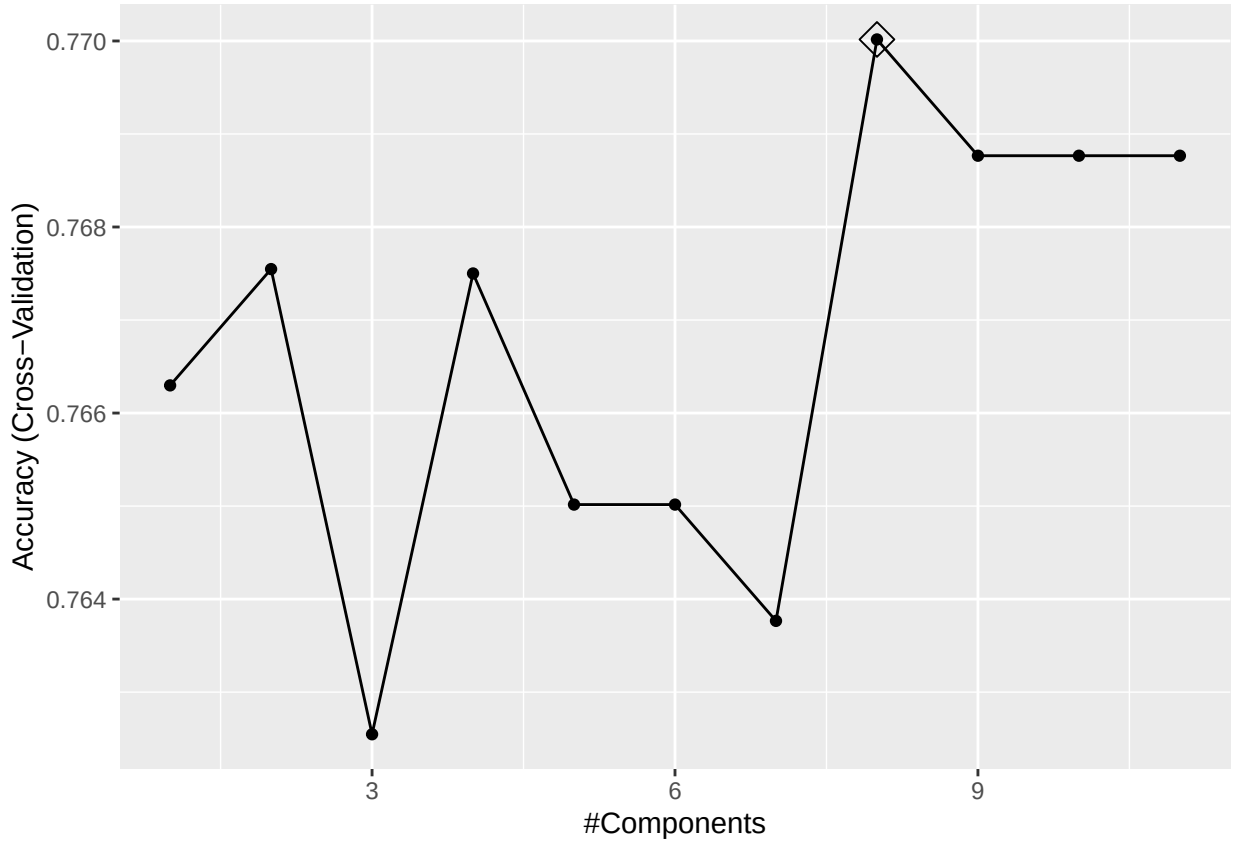
```
set.seed(2)
pls_fit <- train(severe_flu ~ . - id,
          data = flu_train,
          method = "pls",
          tuneGrid = data.frame(ncomp = 1:11),
          trControl = ctrl1,
```

```
          preProcess = c("center", "scale"))
          predy2_pls2 <- predict(pls_fit, newdata = flu_test)
          mean((flu_test$severe_flu - predy2_pls2)^2)
```

```
## Warning in Ops.factor(flu_test$severe_flu, predy2_pls2): '-' not meaningful for
## factors
```

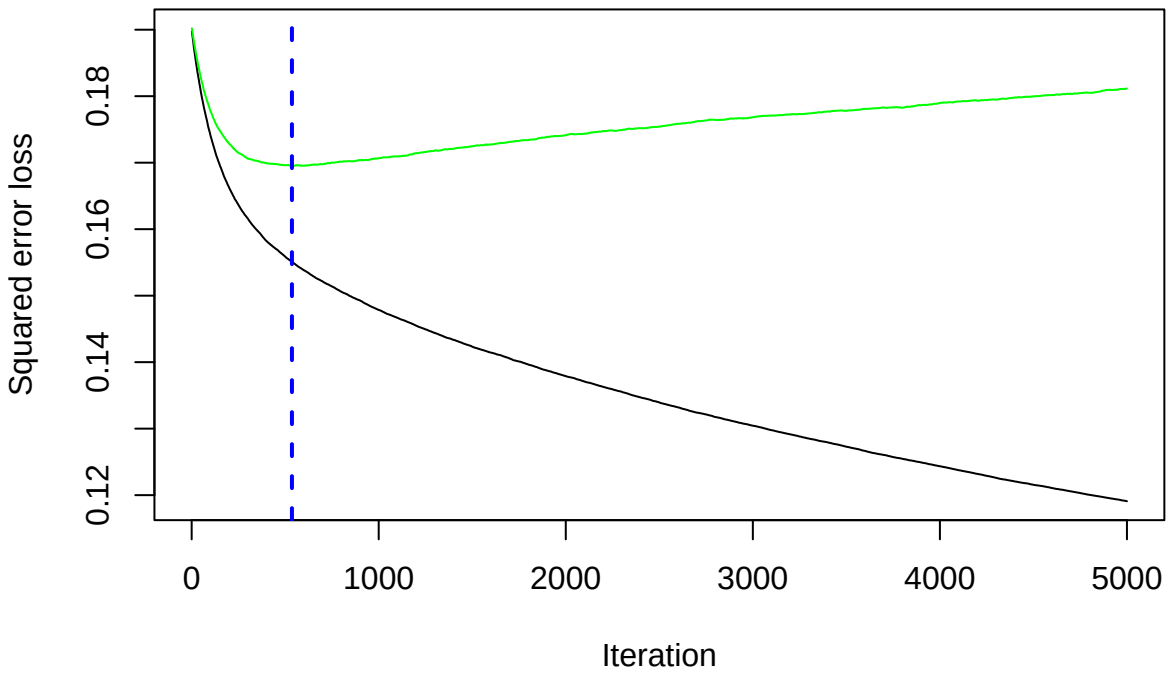
```
## [1] NA
```

```
ggplot(pls_fit, highlight = TRUE)
```



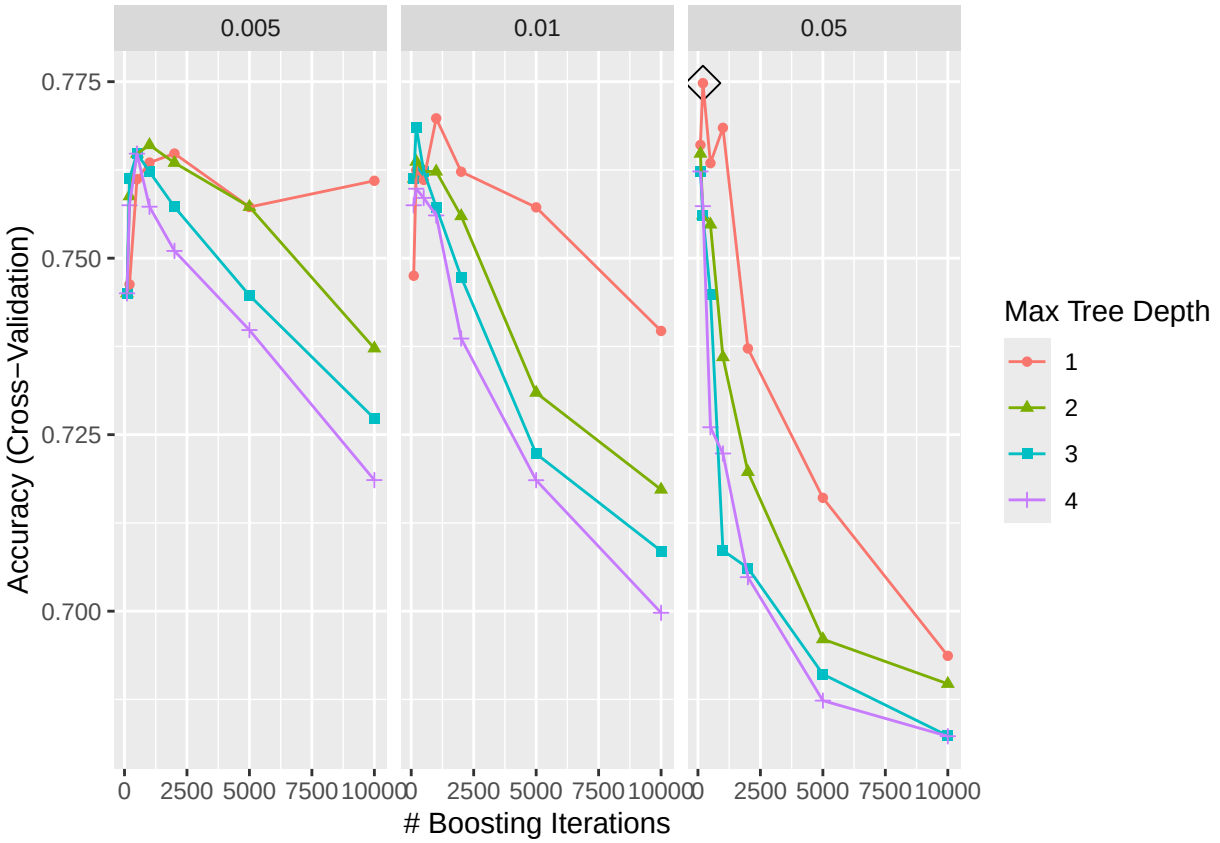
Modle 7: Boosting

```
set.seed(1)
bst = gbm(severe_flu ~ . - id,
          data = flu_train,
          distribution = "gaussian",
          n.trees = 5000,
          interaction.depth = 2,
          shrinkage = 0.005,
          cv.folds =10)

gbm.perf(bst, method = "cv")
```

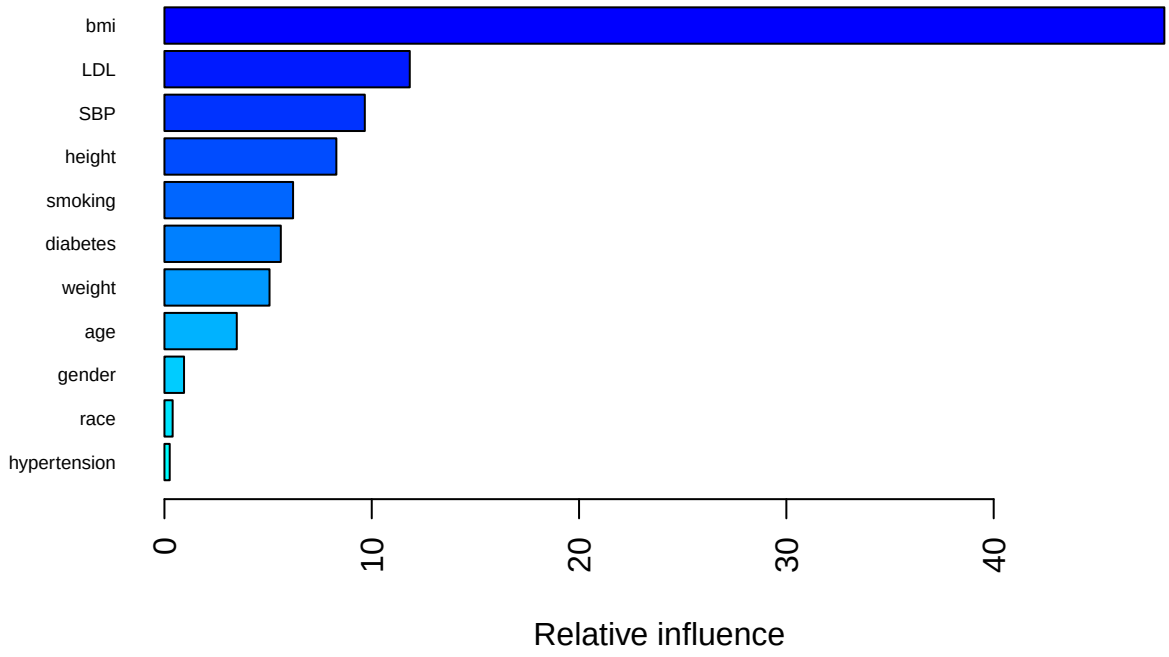



```
## [1] 536
```

```
ctrl = trainControl(method = "cv")

gbm.grid = expand.grid(n.trees = c(100,200,500,1000,2000,5000,10000),
                       interaction.depth = 1:4,
                       shrinkage = c(0.005,0.01,0.05),
                       n.minobsinnode = c(10))

set.seed(1)
gbm.fit = train(severe_flu ~ . - id,
                data = flu_train,
                method = "gbm",
                tuneGrid = gbm.grid,
                trControl = ctrl,
                verbose = FALSE
                )

ggplot(gbm.fit, highlight = TRUE)
```



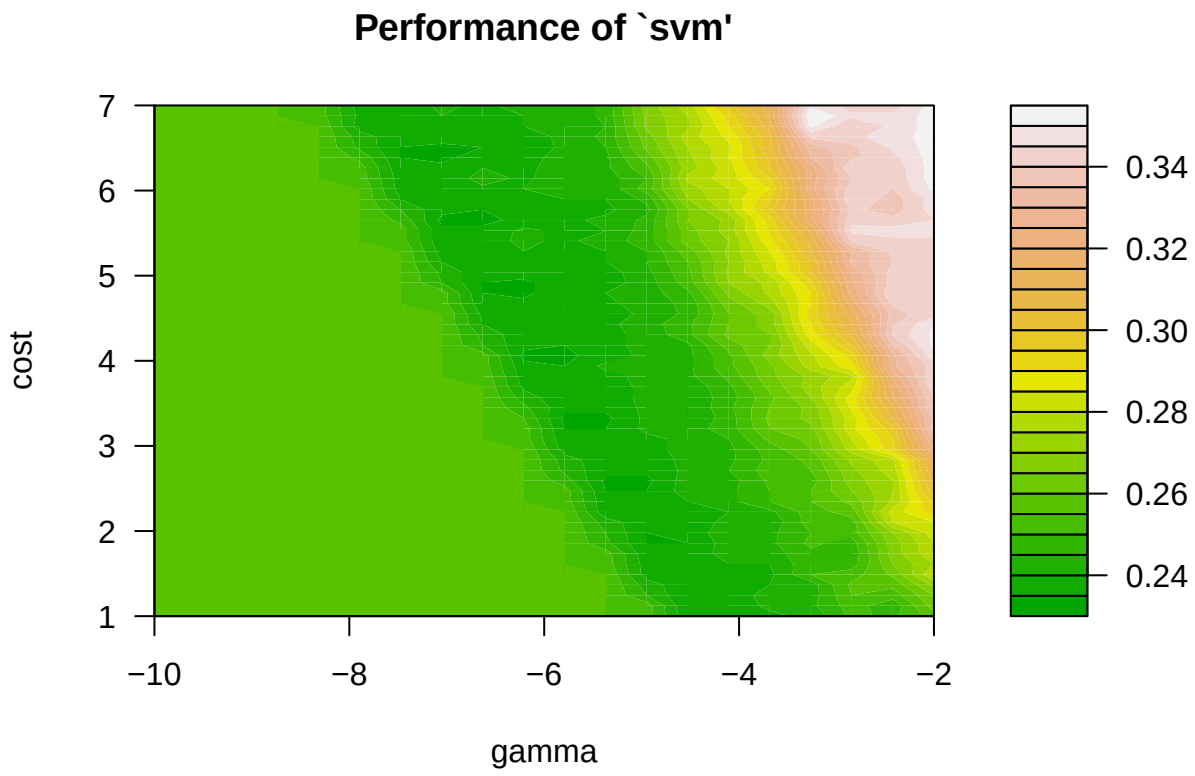

```
summary(gbm.fit$finalModel, las = 2, cBars = 19, cex.names = 0.6)
```




```
##                          var    rel.inf
## bmi                      bmi 48.2328670
## LDL                      LDL 11.8339109
## SBP                      SBP  9.6659378
## height                height  8.2931012
## smoking              smoking  6.2049675
## diabetes            diabetes  5.6129193
## weight                weight  5.0695907
## age                      age  3.4895723
## gender                gender  0.9459542
## race                    race  0.3961478
## hypertension hypertension  0.2550313
```

Model 8; SVM-Linear

```
set.seed(1)
radial.tune = tune.svm(severe_flu ~ . - id,
                       data = flu_train,
                       kernel = "radial",
                       cost = exp(seq(1, 7, len = 50)),
                       gamma = exp(seq(-10, -2,len = 20)))
                       plot(radial.tune, transform.y = log, transform.x = log,
                       color.palette = terrain.colors)
```




```
radial.tune$best.parameters
```

```
##           gamma     cost
## 272 0.004661486 13.35428
```

```
best.radial = radial.tune$best.model
summary(best.radial)
```

```
##
## Call:
## best.svm(x = severe_flu ~ . - id, data = flu_train, gamma = exp(seq(-10,
##     -2, len = 20)), cost = exp(seq(1, 7, len = 50)), kernel = "radial")
##
##
## Parameters:
##    SVM-Type:  C-classification
##  SVM-Kernel:  radial
##        cost:  13.35428
##
## Number of Support Vectors:  426
##
##  ( 223 203 )
##
##
## Number of Classes:  2
```

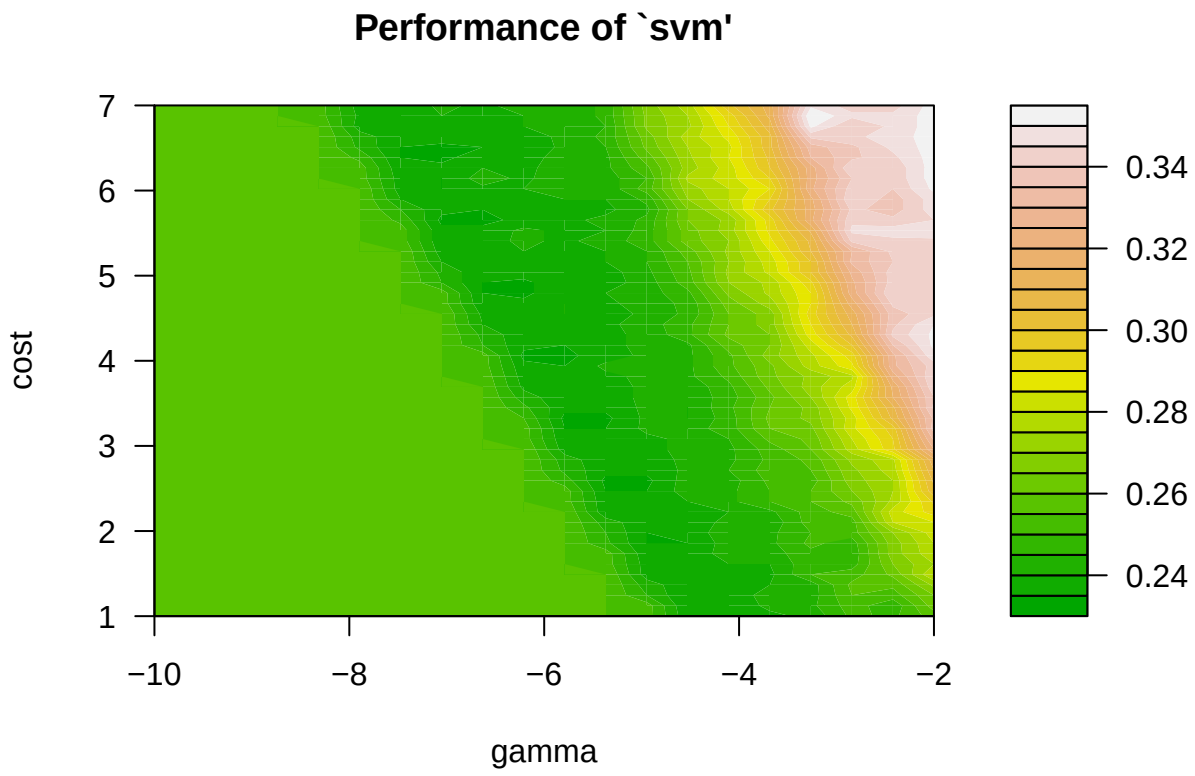
```
##
## Levels:
##  No Yes
```

```
pred.radial = predict(best.radial, newdata = flu_test)
confusionMatrix(data = pred.radial,
                reference = flu_test$severe_flu)
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction  No Yes
##        No  150  48
##        Yes   1   1
##
##                Accuracy : 0.755
##                  95% CI : (0.6894, 0.8129)
##     No Information Rate : 0.755
##     P-Value [Acc > NIR] : 0.5383
##
##                   Kappa : 0.0204
##
##  Mcnemar's Test P-Value : 4.983e-11
##
##             Sensitivity : 0.99338
##             Specificity : 0.02041
##          Pos Pred Value : 0.75758
##          Neg Pred Value : 0.50000
##              Prevalence : 0.75500
##          Detection Rate : 0.75000
##    Detection Prevalence : 0.99000
##       Balanced Accuracy : 0.50689
##
##        'Positive' Class : No
##
```

Model 9: SVM-Radial

```
set.seed(1)
radial.tune = tune.svm(severe_flu ~ . - id,
                data = flu_train,
                kernel = "radial",
                cost = exp(seq(1, 7, len = 50)),
                gamma = exp(seq(-10, -2,len = 20)))
plot(radial.tune, transform.y = log, transform.x = log,
color.palette = terrain.colors)
```




```
radial.tune$best.parameters
```

```
##           gamma     cost
## 272 0.004661486 13.35428
```

```
best.radial = radial.tune$best.model
summary(best.radial)
```

```
##
## Call:
## best.svm(x = severe_flu ~ . - id, data = flu_train, gamma = exp(seq(-10,
##     -2, len = 20)), cost = exp(seq(1, 7, len = 50)), kernel = "radial")
##
##
## Parameters:
##    SVM-Type:  C-classification
##  SVM-Kernel:  radial
##        cost:  13.35428
##
## Number of Support Vectors:  426
##
##  ( 223 203 )
##
##
## Number of Classes:  2
```

```
##
## Levels:
##  No Yes
```

```
pred.radial = predict(best.radial, newdata = flu_test)
confusionMatrix(data = pred.radial,
reference = flu_test$severe_flu)
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction  No Yes
##        No  150  48
##        Yes   1   1
##
##                Accuracy : 0.755
##                  95% CI : (0.6894, 0.8129)
##     No Information Rate : 0.755
##     P-Value [Acc > NIR] : 0.5383
##
##                   Kappa : 0.0204
##
##  Mcnemar's Test P-Value : 4.983e-11
##
##             Sensitivity : 0.99338
##             Specificity : 0.02041
##          Pos Pred Value : 0.75758
##          Neg Pred Value : 0.50000
##              Prevalence : 0.75500
##          Detection Rate : 0.75000
##    Detection Prevalence : 0.99000
##       Balanced Accuracy : 0.50689
##
##        'Positive' Class : No
##
```

Cross Validation for comparison:

```
set.seed(1)

ctrl_cv = trainControl(
  method = "cv",
  number = 10,
  classProbs = TRUE,
  summaryFunction = twoClassSummary,
  savePredictions = "final"
)

flu_train = flu_train %>% select(-id)

model_list = list()
```

```r
results = data.frame(Model = character(), ROC = numeric(), stringsAsFactors = FALSE)

# Model 1: GLM
model_list$GLM = train(severe_flu ~ ., data = flu_train, method = "glm", family = "binomial",
                       trControl = ctrl_cv, metric = "ROC")
results = rbind(results, data.frame(Model = "GLM", ROC = max(model_list$GLM$results$ROC)))

# Model 2: Ridge
model_list$Ridge = train(severe_flu ~ ., data = flu_train, method = "glmnet",
                         tuneGrid = expand.grid(alpha = 0, lambda = exp(seq(6, -5, length = 100))),
                         trControl = ctrl_cv, metric = "ROC")
results = rbind(results, data.frame(Model = "Ridge", ROC = max(model_list$Ridge$results$ROC)))

# Model 3: Lasso
model_list$Lasso = train(severe_flu ~ ., data = flu_train, method = "glmnet",
                         tuneGrid = expand.grid(alpha = 1, lambda = exp(seq(6, -5, length = 100))),
                         trControl = ctrl_cv, metric = "ROC")
results = rbind(results, data.frame(Model = "Lasso", ROC = max(model_list$Lasso$results$ROC)))

# Model 4: LDA
model_list$LDA = train(severe_flu ~ ., data = flu_train, method = "lda",
                       trControl = ctrl_cv, metric = "ROC")
results = rbind(results, data.frame(Model = "LDA", ROC = max(model_list$LDA$results$ROC)))

# Model 5: MARS
model_list$MARS = train(severe_flu ~ ., data = flu_train, method = "earth",
                        tuneLength = 10, trControl = ctrl_cv, metric = "ROC")
results = rbind(results, data.frame(Model = "MARS", ROC = max(model_list$MARS$results$ROC)))

# Model 6: PLS
model_list$PLS = train(severe_flu ~ ., data = flu_train, method = "pls",
                       tuneLength = 15, preProcess = c("center", "scale"),
                       trControl = ctrl_cv, metric = "ROC")
results = rbind(results, data.frame(Model = "PLS", ROC = max(model_list$PLS$results$ROC)))

# Model 7: Boosting (GBM)
model_list$GBM = train(severe_flu ~ ., data = flu_train, method = "gbm",
                       trControl = ctrl_cv, metric = "ROC", verbose = FALSE)
results = rbind(results, data.frame(Model = "Boosting", ROC = max(model_list$GBM$results$ROC)))

# Model 8: SVM - Linear
model_list$SVM_Linear = train(severe_flu ~ ., data = flu_train, method = "svmLinear",
                              trControl = ctrl_cv, metric = "ROC")
results = rbind(results, data.frame(Model = "SVM-Linear", ROC = max(model_list$SVM_Linear$results$ROC)))

# Model 9: SVM - Radial
model_list$SVM_Radial = train(severe_flu ~ ., data = flu_train, method = "svmRadial",
                              trControl = ctrl_cv, metric = "ROC")
results = rbind(results, data.frame(Model = "SVM-Radial", ROC = max(model_list$SVM_Radial$results$ROC)))

results = results %>%
  arrange(desc(ROC))
```

```r
print(results)
```

```
##        Model       ROC
## 1      Ridge 0.7187702
## 2        PLS 0.7175067
## 3 SVM-Linear 0.7115384
## 4        LDA 0.7081202
## 5      Lasso 0.7072877
## 6        GLM 0.7069560
## 7 SVM-Radial 0.6997513
## 8   Boosting 0.6964448
## 9       MARS 0.6919931
```

Part 2: Developing a predictive risk score (i.e., the predicted probability) that quantifies the chance of experiencing severe flu based on individual participant characteristics.

Given the Ridge regression model had the best ROC, that is the model that will be used to build the predictive risk score.

```r
library(glmnet)
```

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

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

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

```
## Loaded glmnet 4.1-8
```

```r
x = model.matrix(severe_flu ~ . - id, flu)[,-1]
y = flu_train[, "severe_flu"]

ctrl1 = trainControl(method = "cv", number = 10)
set.seed(1)

ridge.fit = train(severe_flu ~ . - id,
                  data = flu,
                  method = "glmnet",
                  tuneGrid = expand.grid(alpha = 0,
                                         lambda = exp(seq(6, -5, length = 100))),
                  trControl = ctrl1)

flu$predicted_risk_prob = predict(ridge.fit, newx = x, s = "lambda.min", type = "prob")[, "Yes"]
```

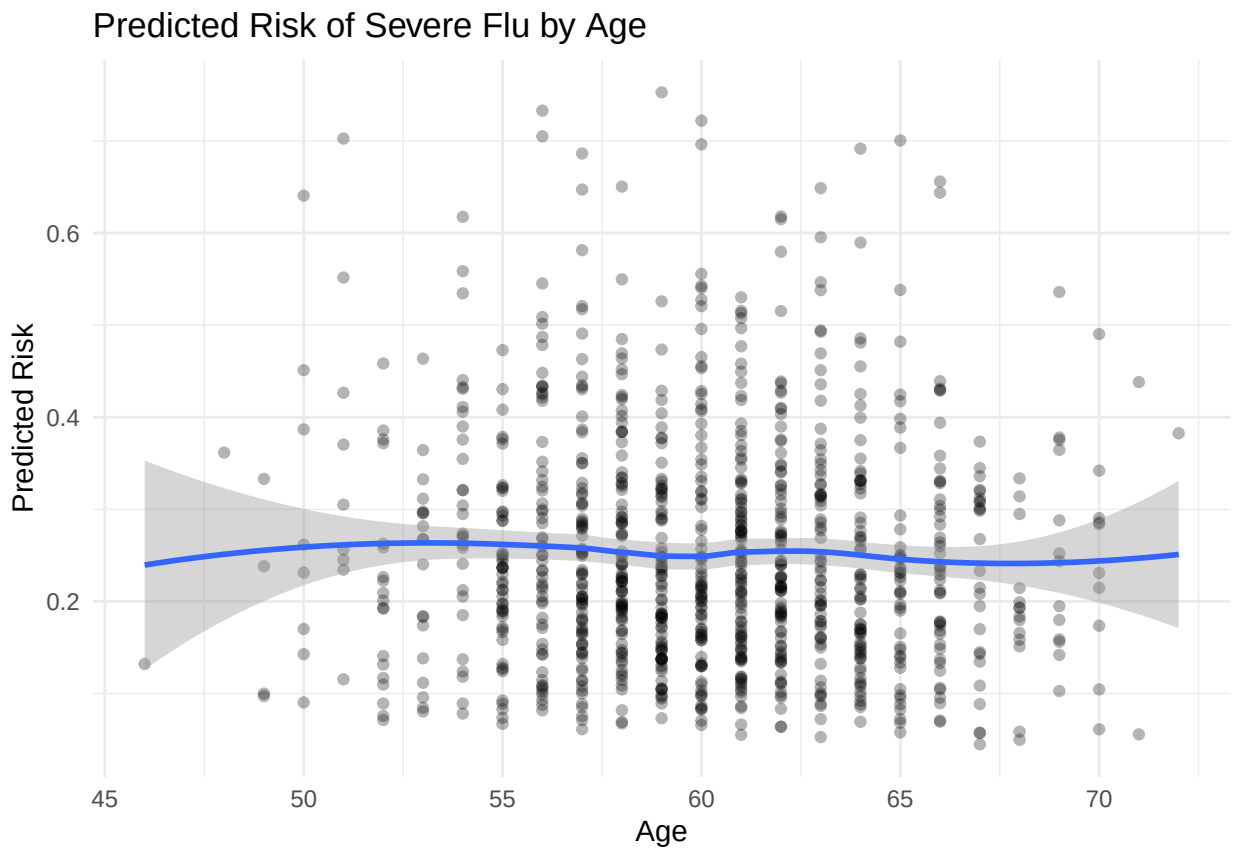
Part 3: Identifying key demographic and clinical factors that predict the risk of severe flu and assessing how these factors influence the risk.

```
# Load library
library(glmnet)

best_lambda = ridge.fit$bestTune$lambda

ridge_glmnet = ridge.fit$finalModel

coef_ridge = coef(ridge_glmnet, s = best_lambda)

coef_df = as.data.frame(as.matrix(coef_ridge))
coef_df$variable <- rownames(coef_df)
colnames(coef_df)[1] <- "coefficient"
coef_df = coef_df %>%
  filter(variable != "(Intercept)") %>%
  arrange(desc(abs(coefficient)))

head(coef_df, 10)
```

```
##                coefficient     variable
## diabetes      0.504482960     diabetes
## gender        0.190885326       gender
## smoking       0.182575139      smoking
## hypertension  0.162644748 hypertension
## bmi           0.142508352          bmi
## height       -0.034815758       height
## race          0.029121222         race
## weight        0.028194464       weight
## age          -0.020681228          age
## LDL           0.006200411          LDL
```

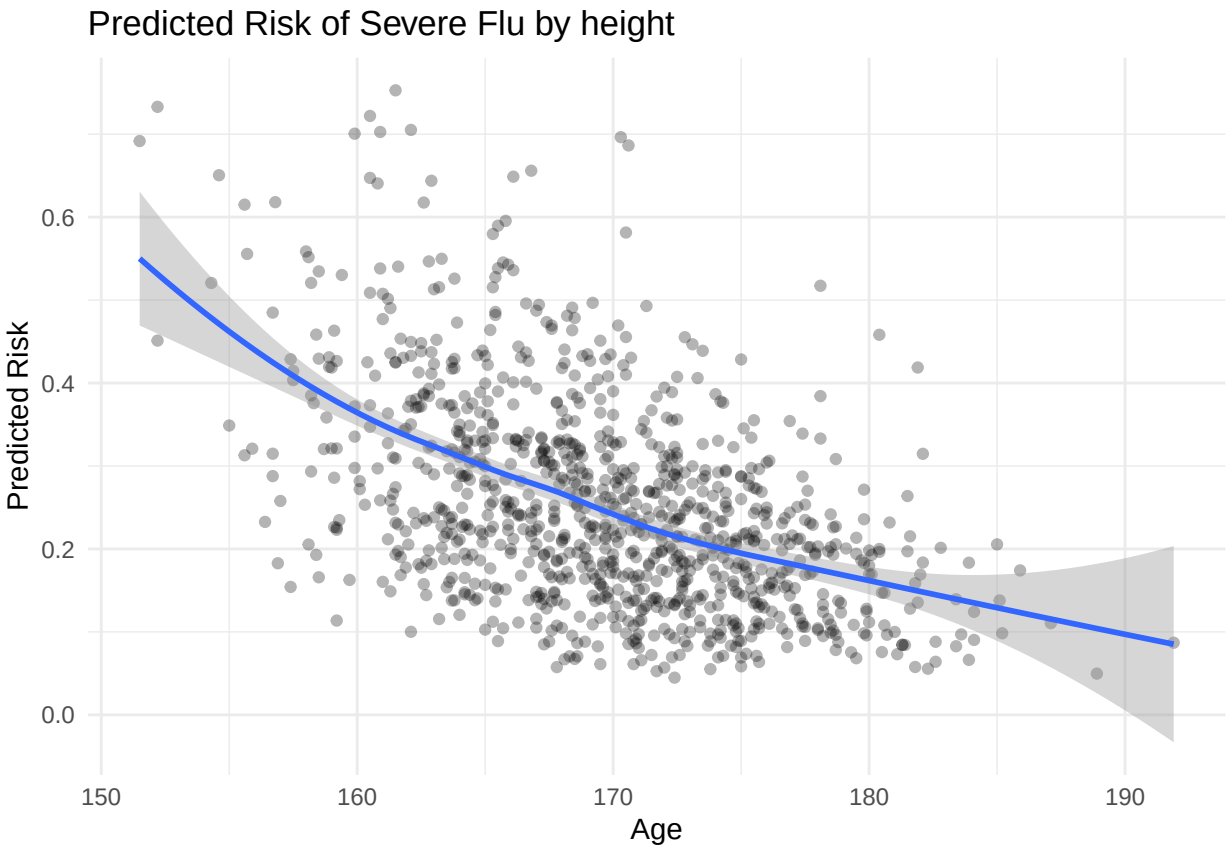
```
library(ggplot2)

# Predicted riskby continuous variable
ggplot(flu, aes(x = age, y = predicted_risk_prob)) +
  geom_point(alpha = 0.3) +
  geom_smooth(method = "loess") +
  labs(title = "Predicted Risk of Severe Flu by Age",
       x = "Age", y = "Predicted Risk") +
  theme_minimal()
```

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



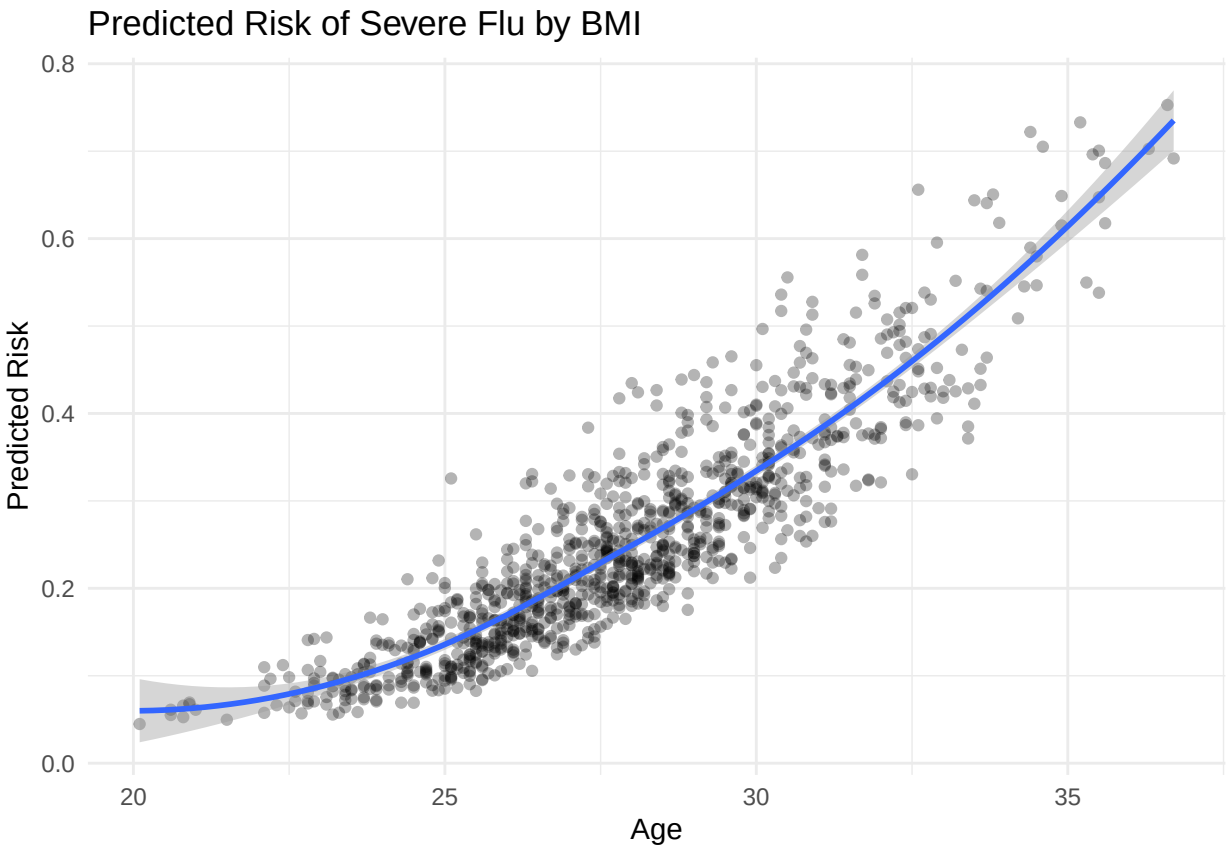

```
ggplot(flu, aes(x = height, y = predicted_risk_prob)) +
  geom_point(alpha = 0.3) +
  geom_smooth(method = "loess") +
  labs(title = "Predicted Risk of Severe Flu by height ",
       x = "Age", y = "Predicted Risk") +
  theme_minimal()
```

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



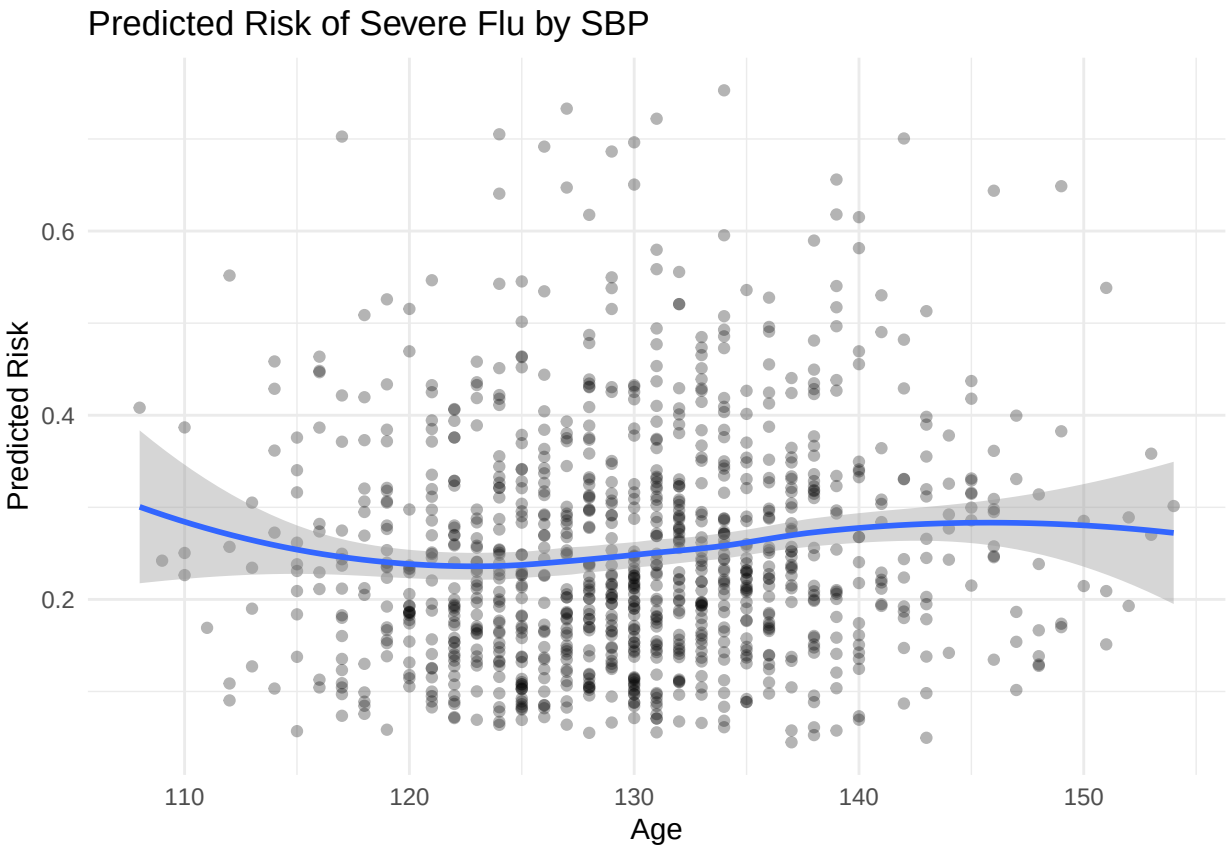

```
ggplot(flu, aes(x = bmi, y = predicted_risk_prob)) +
  geom_point(alpha = 0.3) +
  geom_smooth(method = "loess") +
  labs(title = "Predicted Risk of Severe Flu by BMI ",
       x = "Age", y = "Predicted Risk") +
  theme_minimal()
```

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



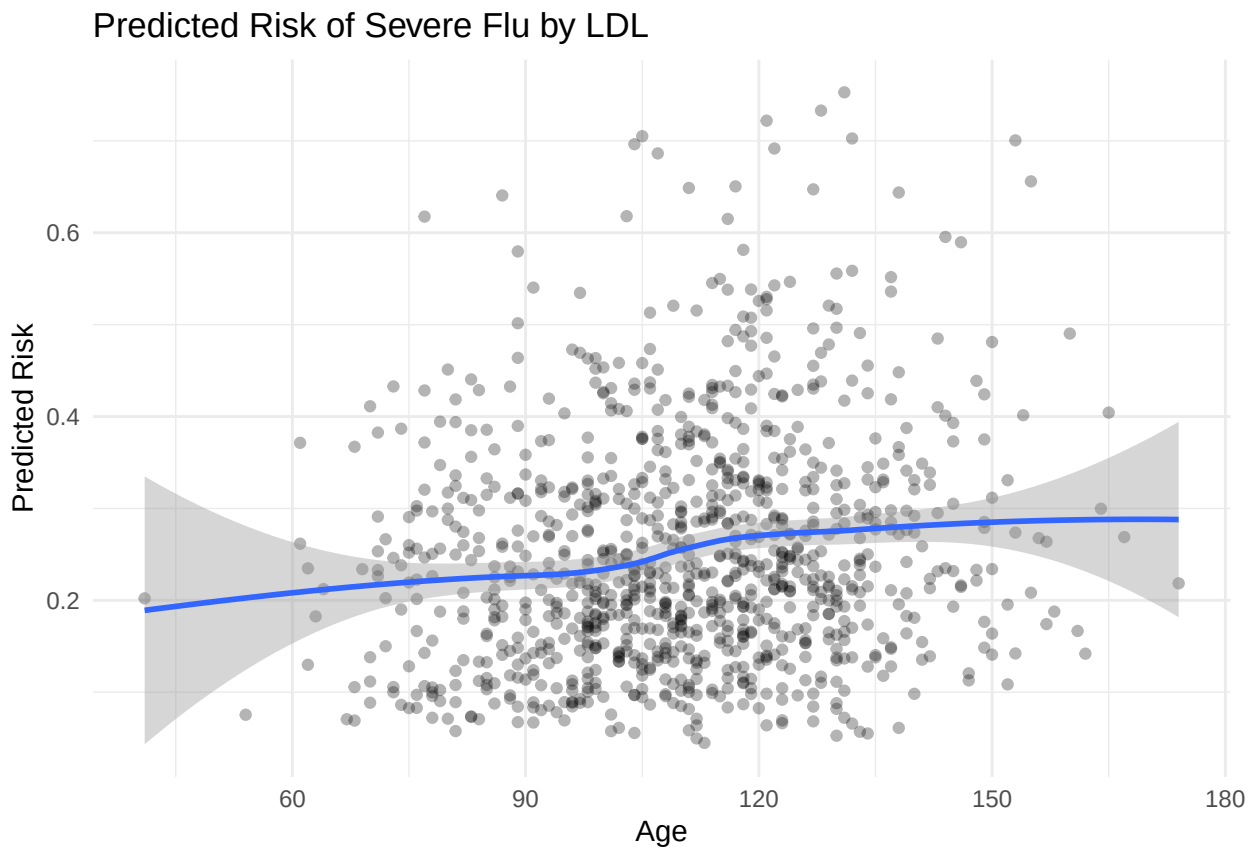

```
ggplot(flu, aes(x = SBP, y = predicted_risk_prob)) +
  geom_point(alpha = 0.3) +
  geom_smooth(method = "loess") +
  labs(title = "Predicted Risk of Severe Flu by SBP",
       x = "Age", y = "Predicted Risk") +
  theme_minimal()
```

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

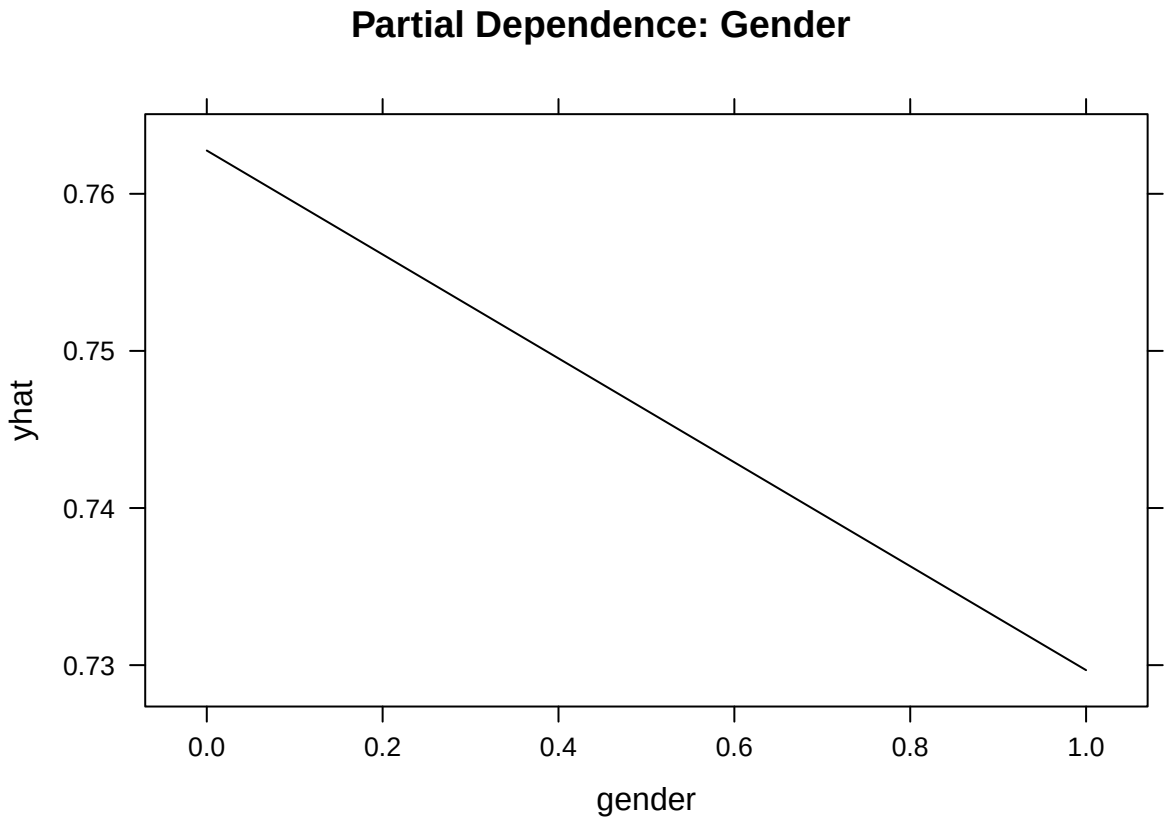



```
ggplot(flu, aes(x = LDL, y = predicted_risk_prob)) +
  geom_point(alpha = 0.3) +
  geom_smooth(method = "loess") +
  labs(title = "Predicted Risk of Severe Flu by LDL",
       x = "Age", y = "Predicted Risk") +
  theme_minimal()
```

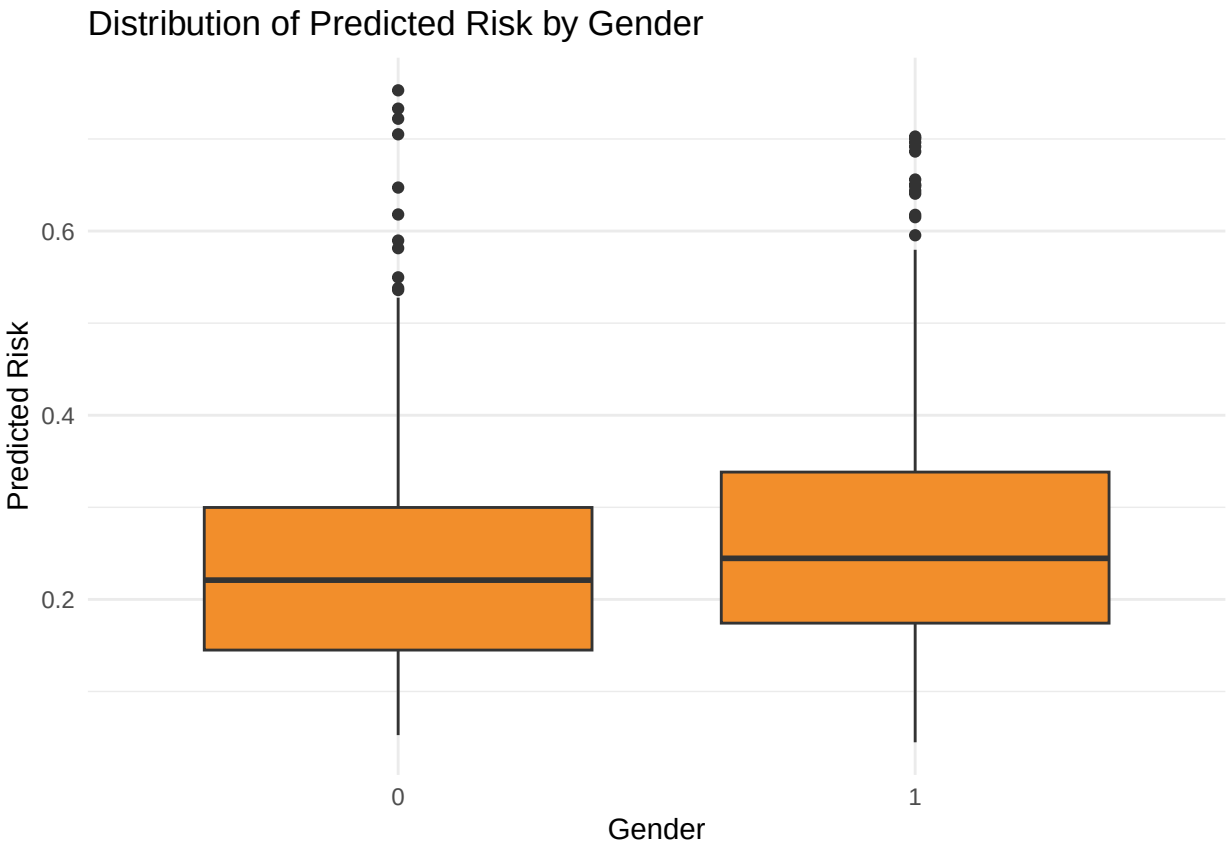
```
## `geom_smooth()` using formula = 'y ~ x'
```



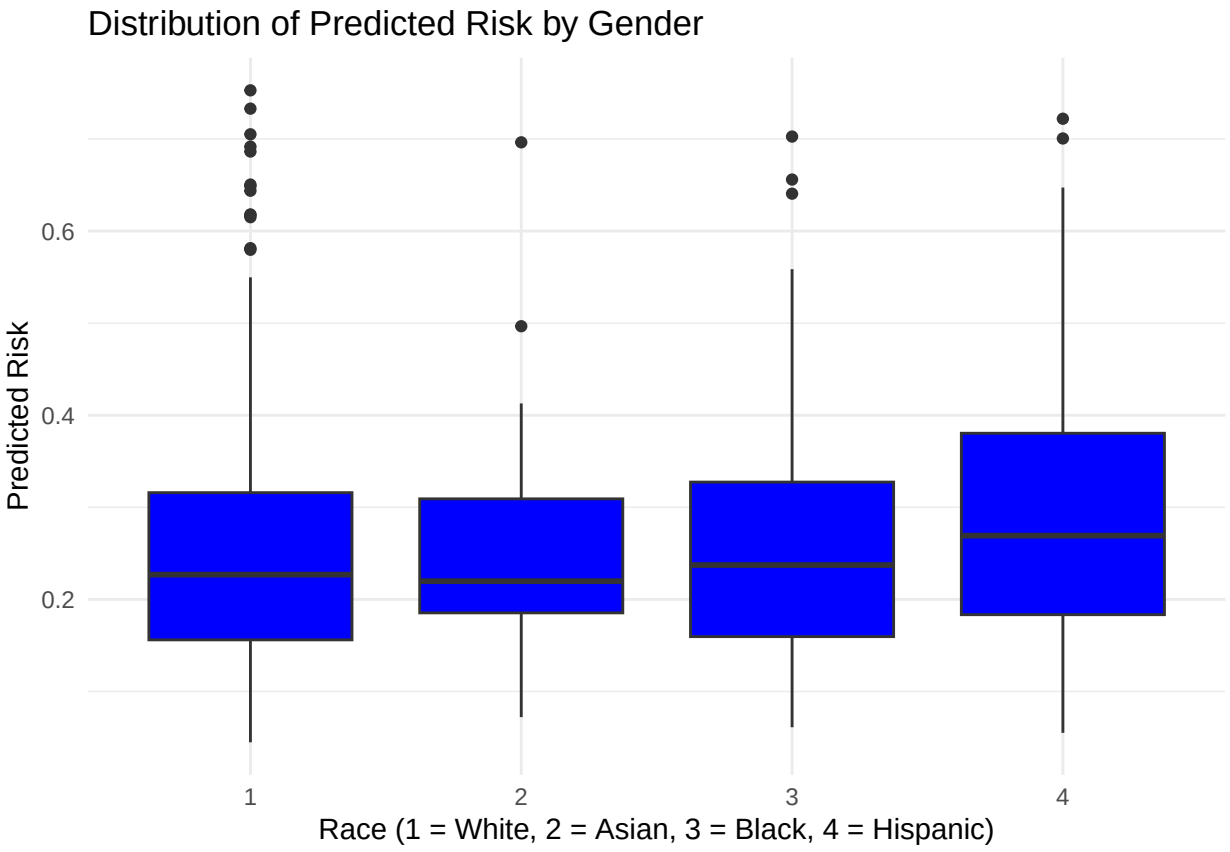

```r
library(pdp)

# Partial dependence plots (continous)
pdp_age = partial(ridge.fit, pred.var = "age", train = flu, type = "classification", prob = TRUE)

pdp_height = partial(ridge.fit, pred.var = "height", train = flu, type = "classification", prob = TRUE)

pdp_weight = partial(ridge.fit, pred.var = "weight", train = flu, type = "classification", prob = TRUE)

pdp_bmi = partial(ridge.fit, pred.var = "bmi", train = flu, type = "classification", prob = TRUE)

pdp_SBP=partial(ridge.fit, pred.var = "SBP", train = flu, type = "classification", prob = TRUE)

pdp_LDL=partial(ridge.fit, pred.var = "LDL", train = flu, type = "classification", prob = TRUE)

pdp_gender = partial(ridge.fit, pred.var = "gender", train = flu, type = "classification", prob = TRUE)
plotPartial(pdp_gender, main = "Partial Dependence: Gender")
```



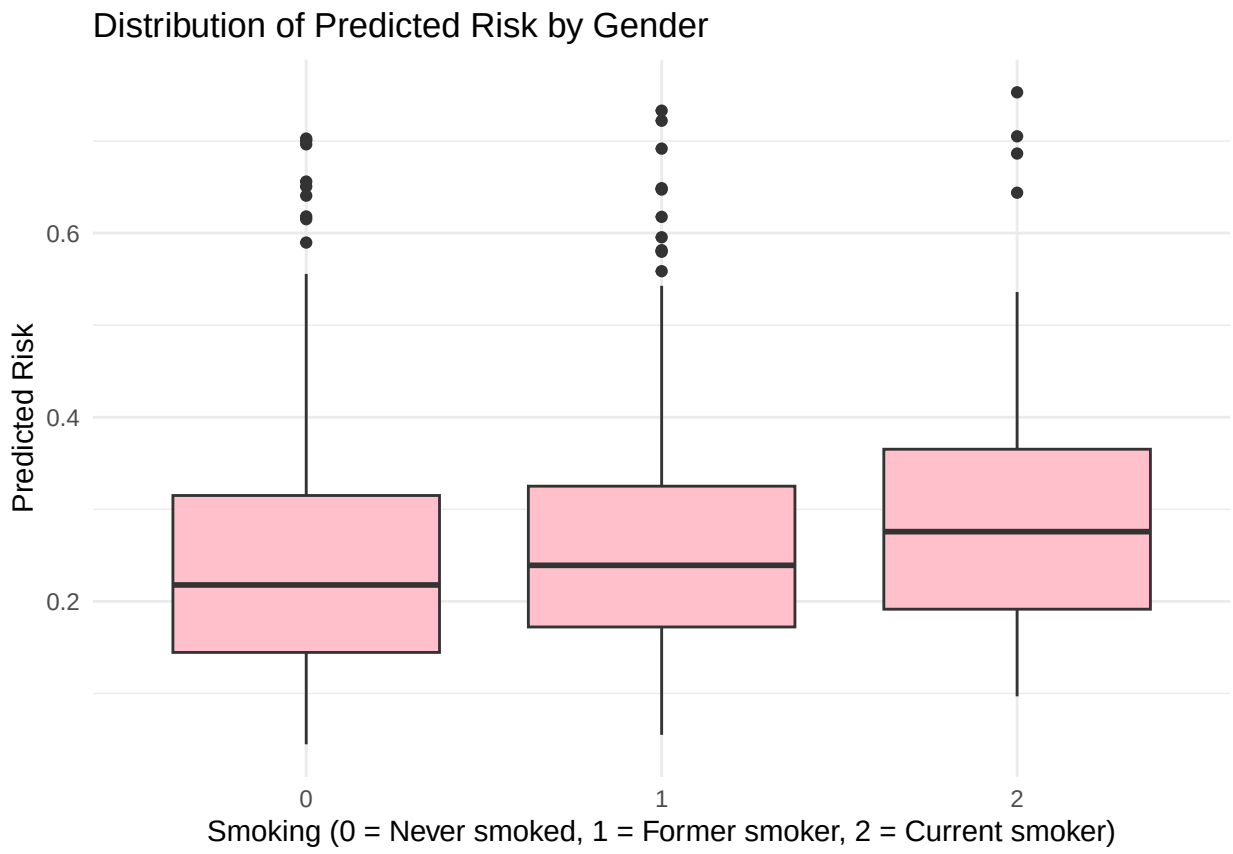

```
ggplot(flu, aes(x = factor(gender), y = predicted_risk_prob)) +
  geom_boxplot(fill = "#F28E2B") +
  labs(title = "Distribution of Predicted Risk by Gender",
       x = "Gender", y = "Predicted Risk") +
  theme_minimal()
```



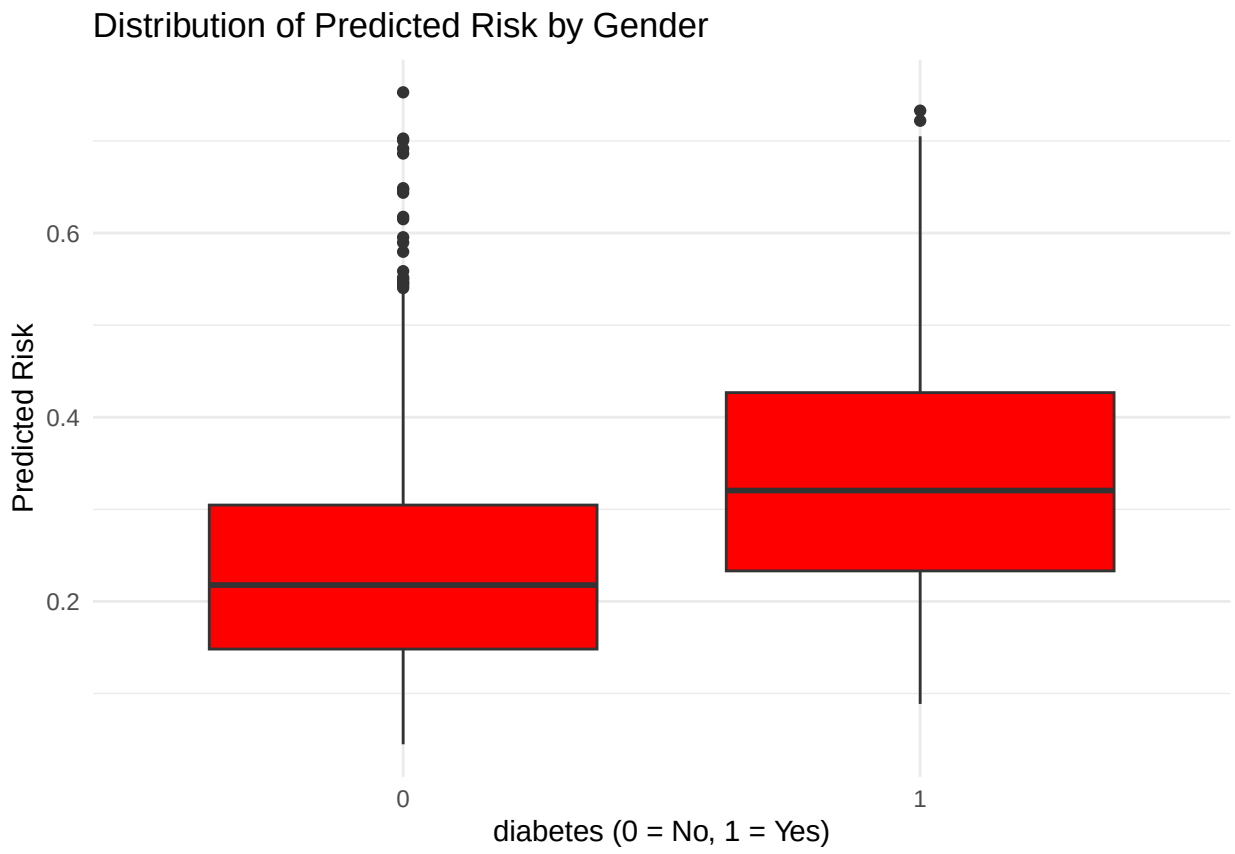

```
ggplot(flu, aes(x = factor(race), y = predicted_risk_prob)) +
  geom_boxplot(fill = "blue") +
  labs(title = "Distribution of Predicted Risk by Gender",
       x = "Race (1 = White, 2 = Asian, 3 = Black, 4 = Hispanic)", y = "Predicted Risk") +
  theme_minimal()
```



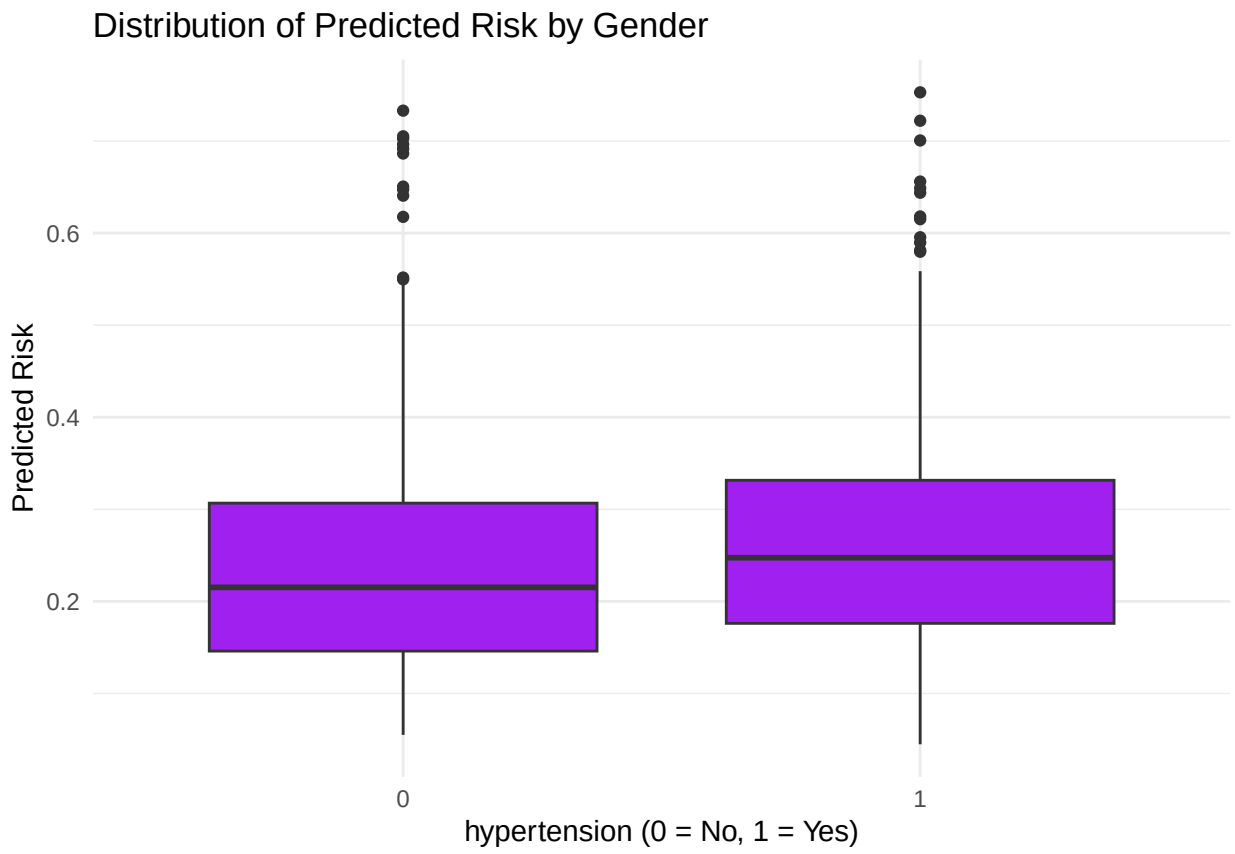

```
ggplot(flu, aes(x = factor(smoking), y = predicted_risk_prob)) +
  geom_boxplot(fill = "pink") +
  labs(title = "Distribution of Predicted Risk by Gender",
       x = "Smoking (0 = Never smoked, 1 = Former smoker, 2 = Current smoker)", y = "Predicted Risk") +
  theme_minimal()
```



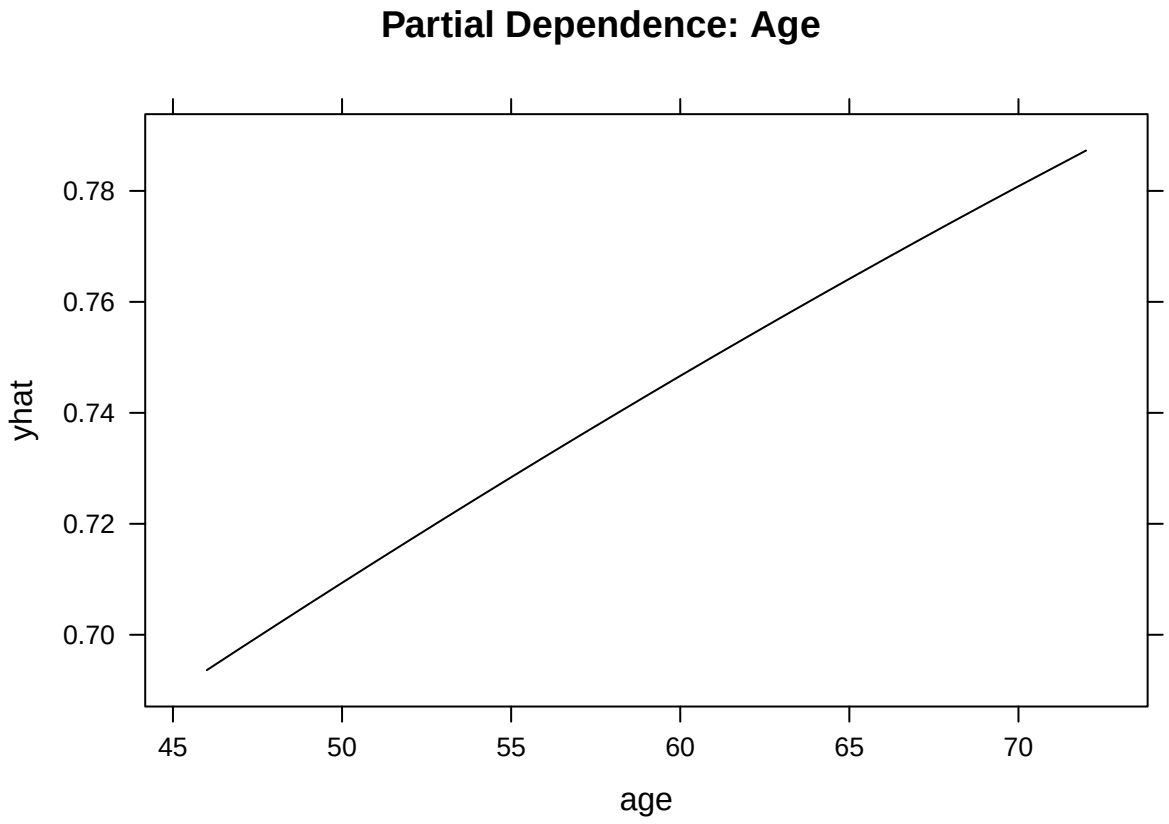

```
ggplot(flu, aes(x = factor(diabetes), y = predicted_risk_prob)) +
  geom_boxplot(fill ="red") +
  labs(title = "Distribution of Predicted Risk by Gender",
       x = "diabetes (0 = No, 1 = Yes)", y = "Predicted Risk") +
  theme_minimal()
```



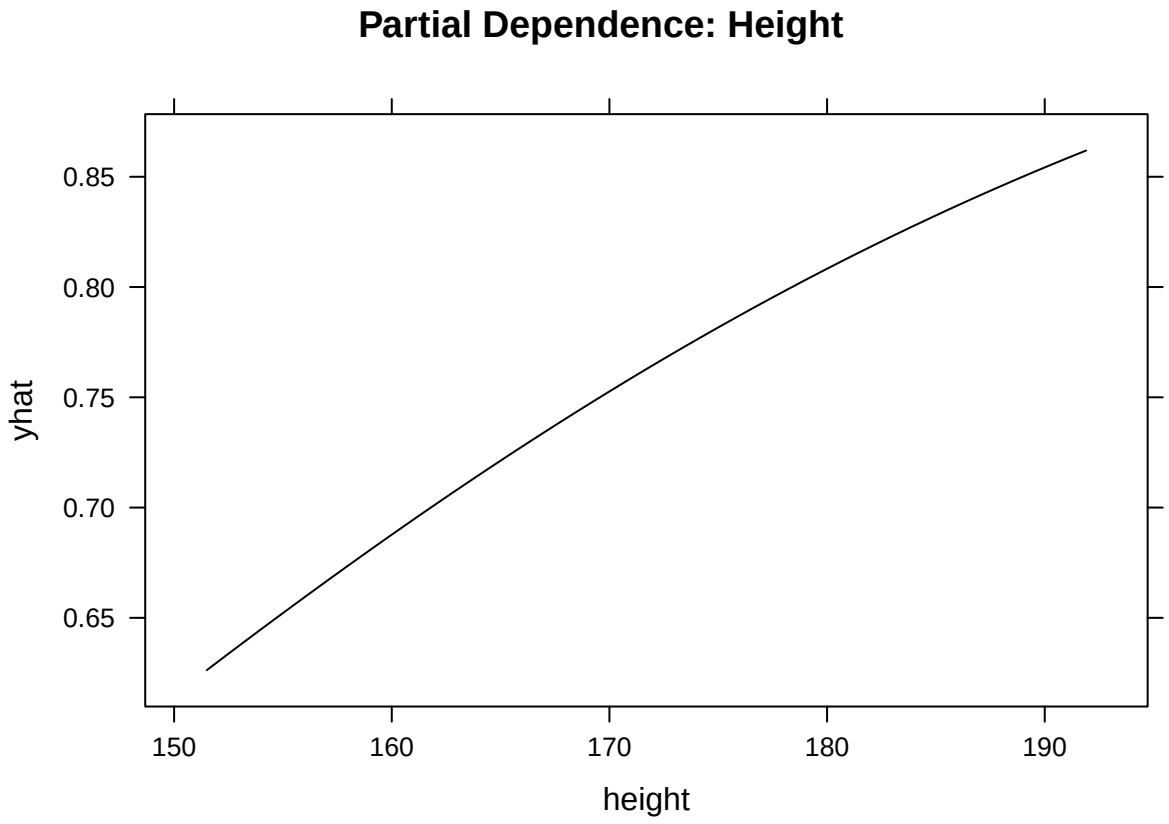

```
ggplot(flu, aes(x = factor(hypertension), y = predicted_risk_prob)) +
  geom_boxplot(fill ="purple") +
  labs(title = "Distribution of Predicted Risk by Gender",
       x = "hypertension (0 = No, 1 = Yes)", y = "Predicted Risk") +
  theme_minimal()
```



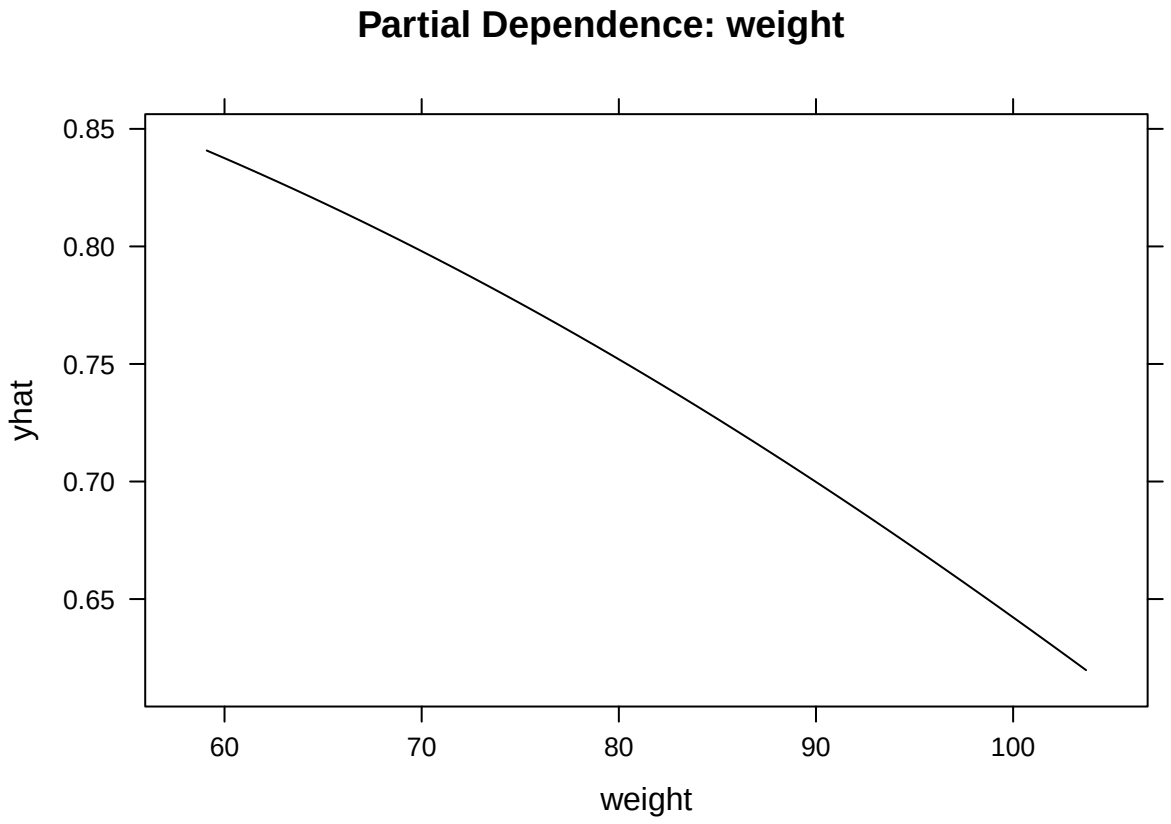

```
plotPartial(pdp_age, main = "Partial Dependence: Age")
```



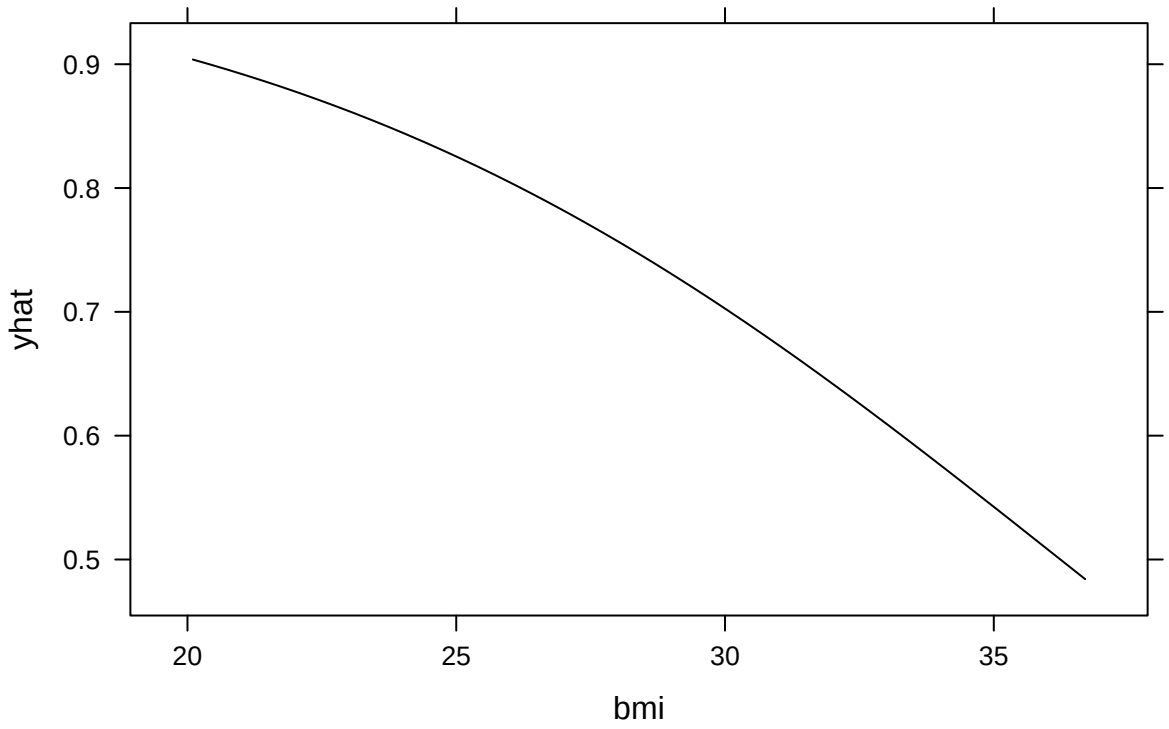

```
plotPartial(pdp_height, main = "Partial Dependence: Height")
```



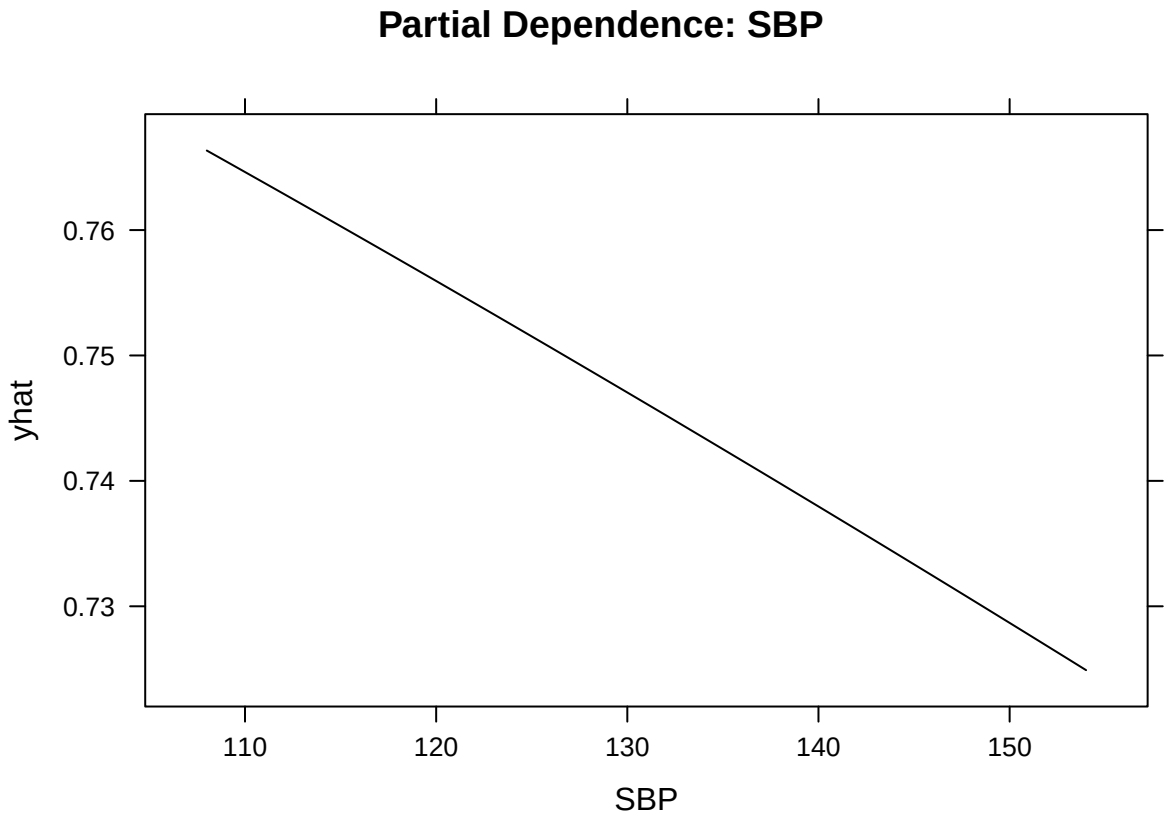

```
plotPartial(pdp_weight, main = "Partial Dependence: weight")
```



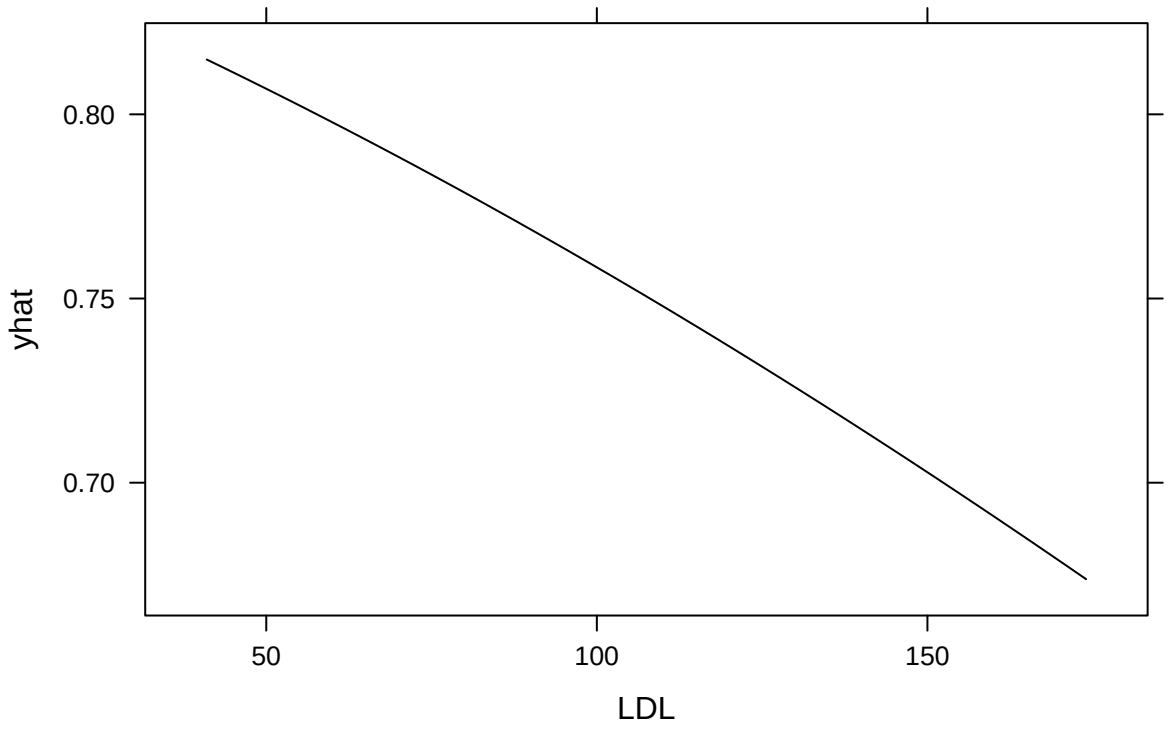

```r
plotPartial(pdp_bmi, main = "Partial Dependence: BMI")
```




```
plotPartial(pdp_SBP, main = "Partial Dependence: SBP")
```




```
plotPartial(pdp_LDL, main = "Partial Dependence: LDL")
```


Partial Dependence: LDL
0.80
0.75
0.70
yhat
50
100
150
LDL