Code

Emily Mittleman & Julia Rosner

2022-12-05

This file will be used for our initial code while we explore the data, different models, etc. Then we'll compile it into Report.Rmd

Load Data

```
#data <- read.csv("Data/diabetes.csv", header = TRUE)
#write.csv(diabetes_binary_5050split, "Data/diabetes_binary_5050split.csv", row.names=FALSE)
data <- read.csv("Data/diabetes_binary_5050split.csv", header = TRUE)
```

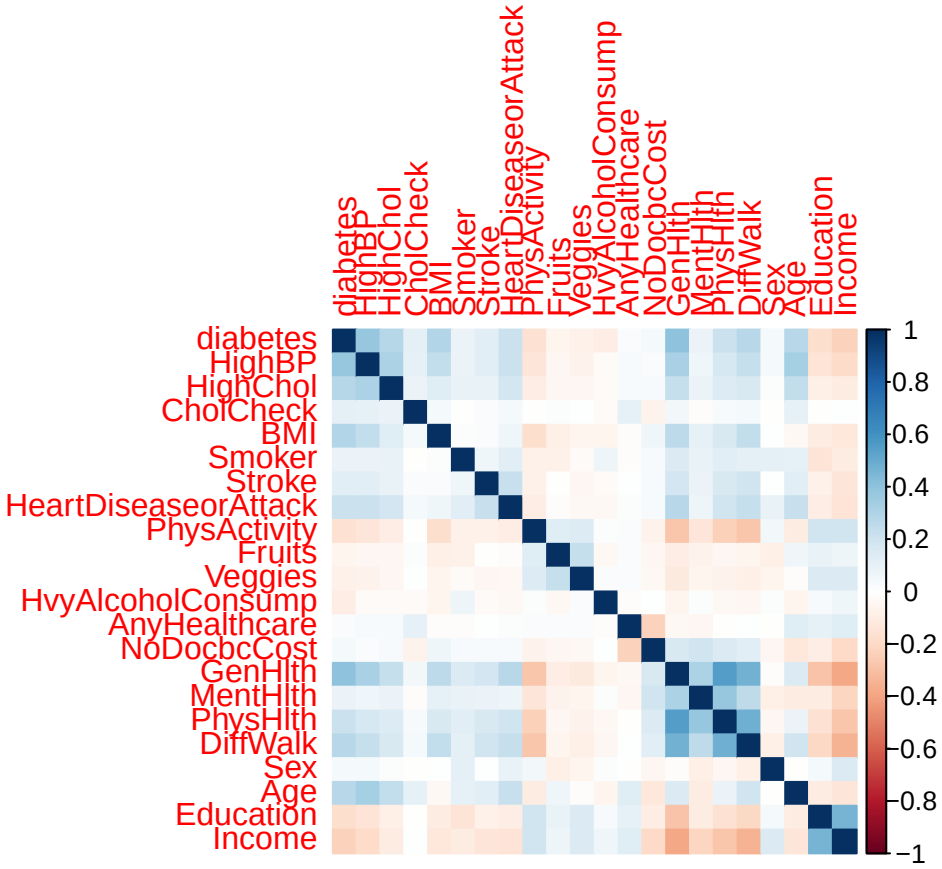
EDA

```
colnames(data)[colnames(data) == "Diabetes_binary"] = "diabetes"
head(data)
```

```
##   diabetes HighBP HighChol CholCheck BMI Smoker Stroke HeartDiseaseorAttack
## 1        0      1        0         1  26      0      0                     0
## 2        0      1        1         1  26      1      1                     0
## 3        0      0        0         1  26      0      0                     0
## 4        0      1        1         1  28      1      0                     0
## 5        0      0        0         1  29      1      0                     0
## 6        0      0        0         1  18      0      0                     0
##   PhysActivity Fruits Veggies HvyAlcoholConsump AnyHealthcare NoDocbcCost
## 1            1      0       1                 0             1           0
## 2            0      1       0                 0             1           0
## 3            1      1       1                 0             1           0
## 4            1      1       1                 0             1           0
## 5            1      1       1                 0             1           0
## 6            1      1       1                 0             0           0
##   GenHlth MentHlth PhysHlth DiffWalk Sex Age Education Income
## 1       3        5       30        0   1   4         6      8
## 2       3        0        0        0   1  12         6      8
## 3       1        0       10        0   1  13         6      8
## 4       3        0        3        0   1  11         6      8
## 5       2        0        0        0   0   8         5      8
## 6       2        7        0        0   0   1         4      7
```

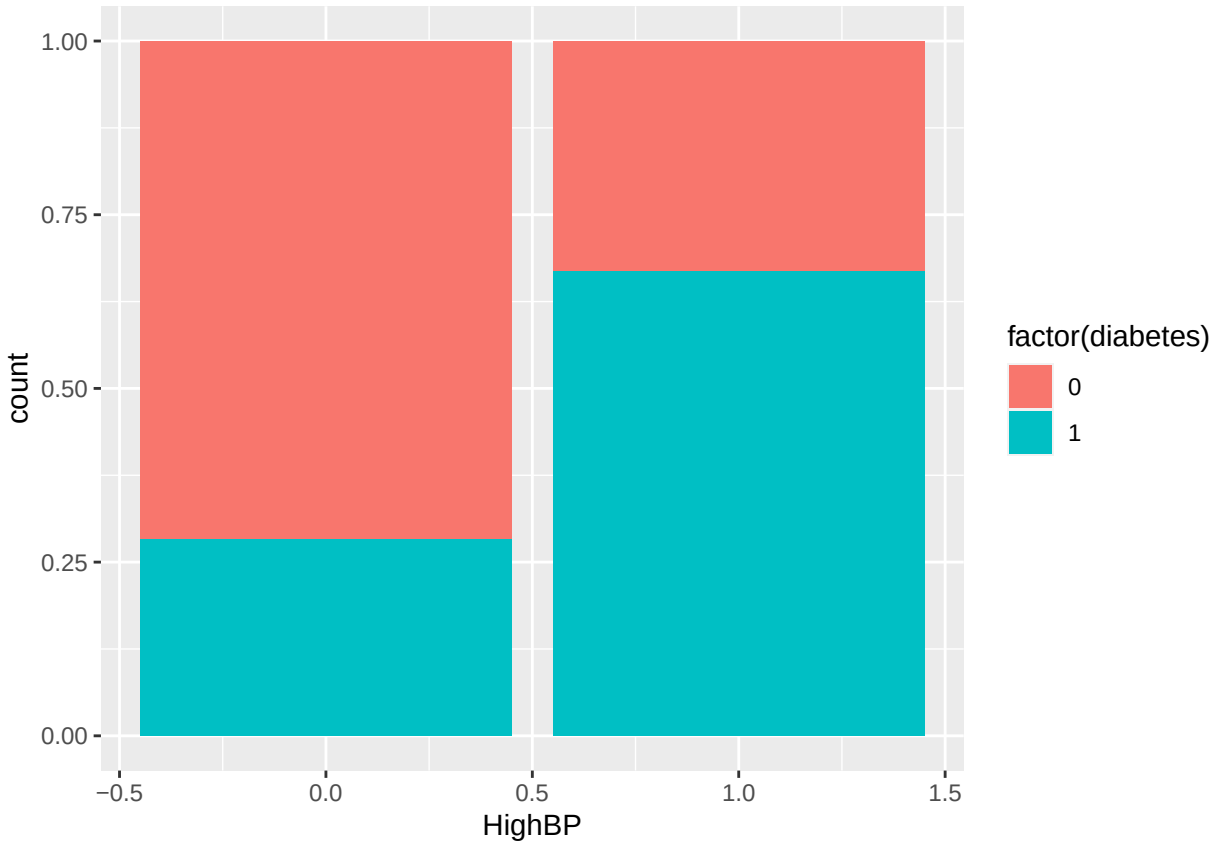
Look at correlations between variables. helps to know which attributes are highy dependent on the prediction variable

```
correlations <- cor(data)
corrplot(correlations, method="color")
```

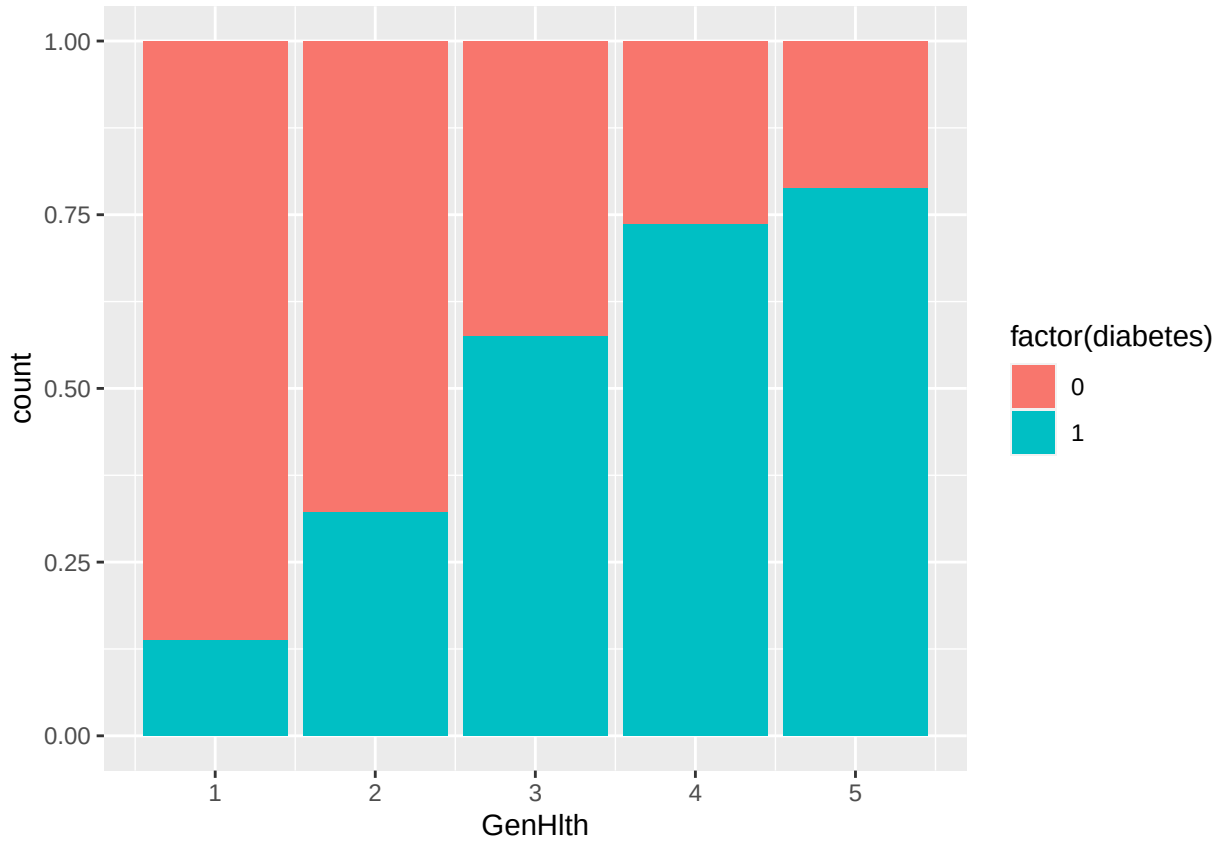


Next, look at box plots of the 2 most correlated predictors and color by outcome.

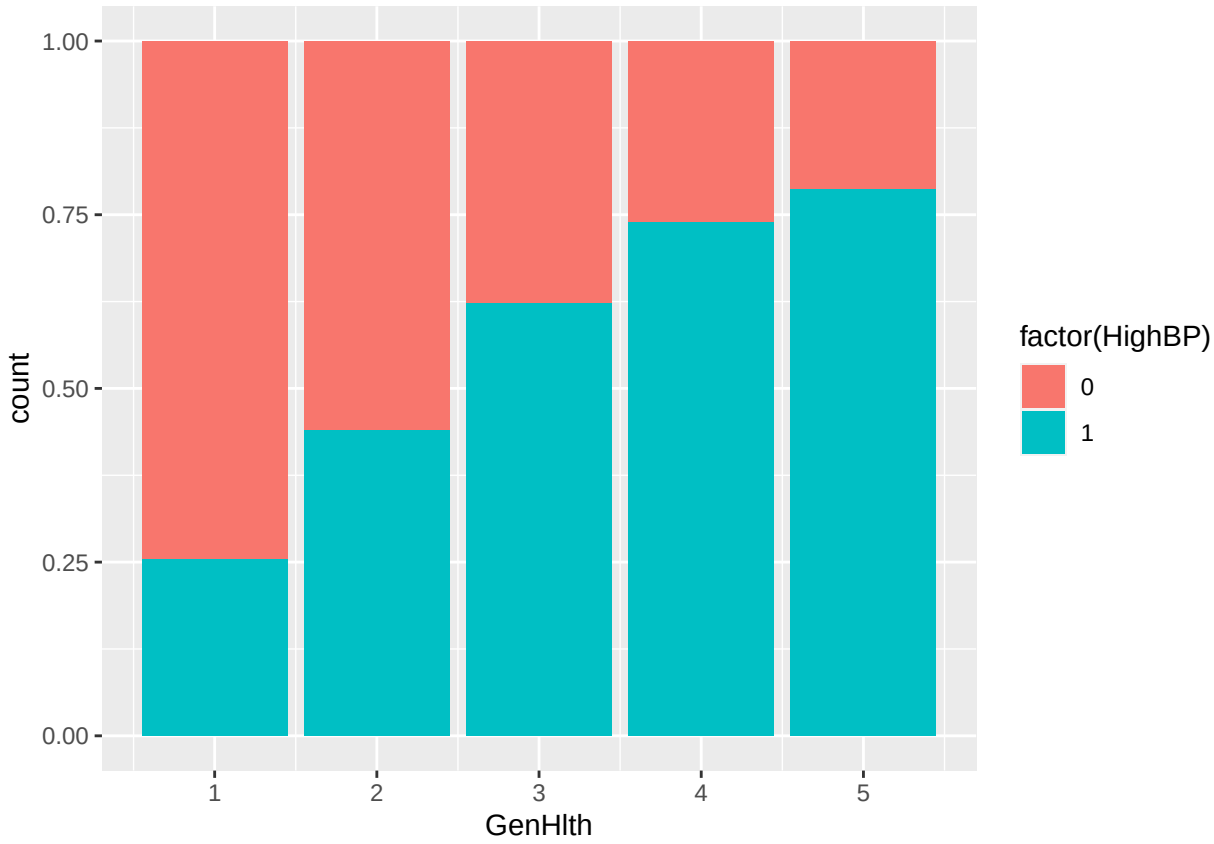
```
ggplot(data, aes(x = HighBP, fill = factor(diabetes))) +
  geom_bar(position="fill")
```




```
ggplot(data, aes(x = GenHlth, fill = factor(diabetes))) +
  geom_bar(position="fill")
```




```
ggplot(data, aes(x = GenHlth, fill = factor(HighBP))) +
  geom_bar(position="fill")
```



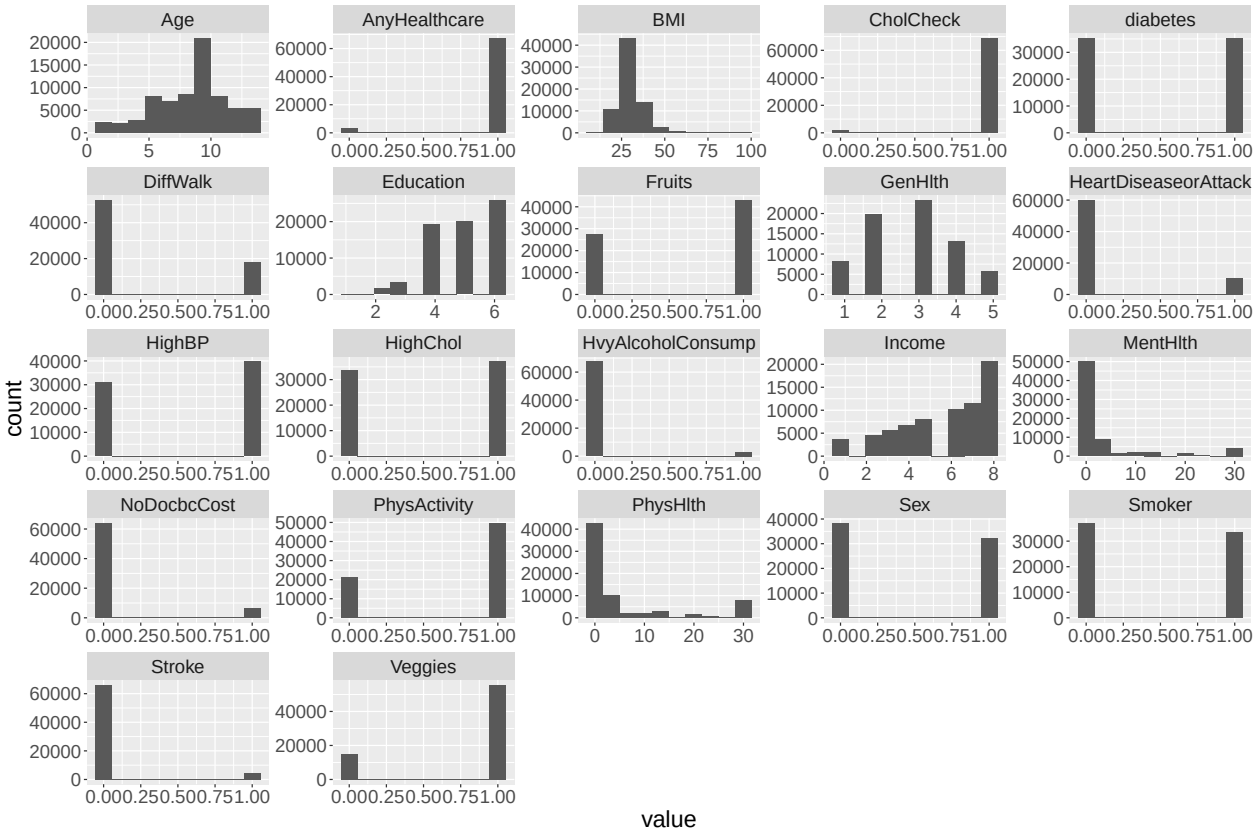
Make pivot table to make historgrams of each variable simpler

```
data_long <- data %>%                          # Apply pivot_longer function
  pivot_longer(colnames(data)) %>%
  as.data.frame()
head(data_long)
```

```
##        name value
## 1  diabetes     0
## 2    HighBP     1
## 3  HighChol     0
## 4 CholCheck     1
## 5       BMI    26
## 6    Smoker     0
```

Visualize predictor variable distrbutions:

```
ggp1 <- ggplot(data_long, aes(x = value)) +    # Draw each column as histogram
  geom_histogram(bins=10) +
  facet_wrap(~ name, scales = "free")+
  theme(text=element_text(size=20))
ggp1
```

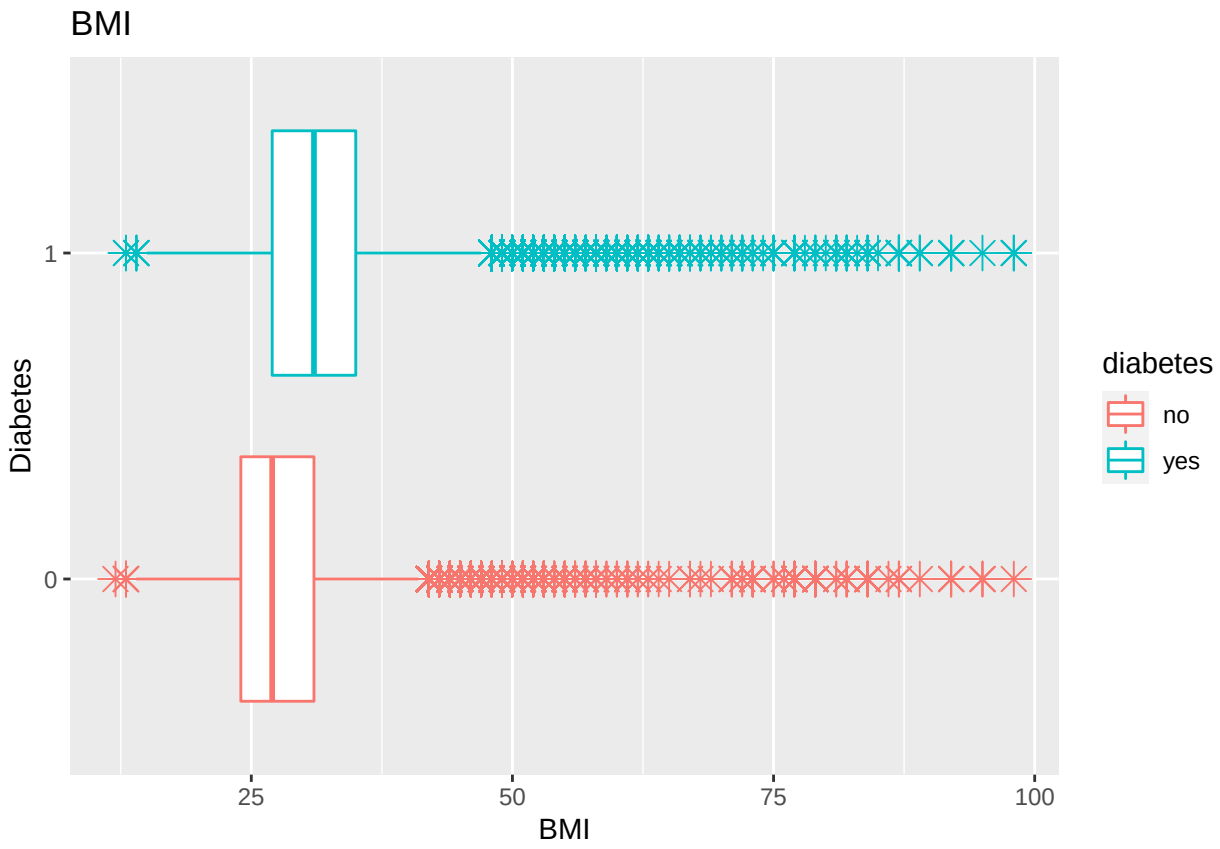


Next, look for outliers in predictors:

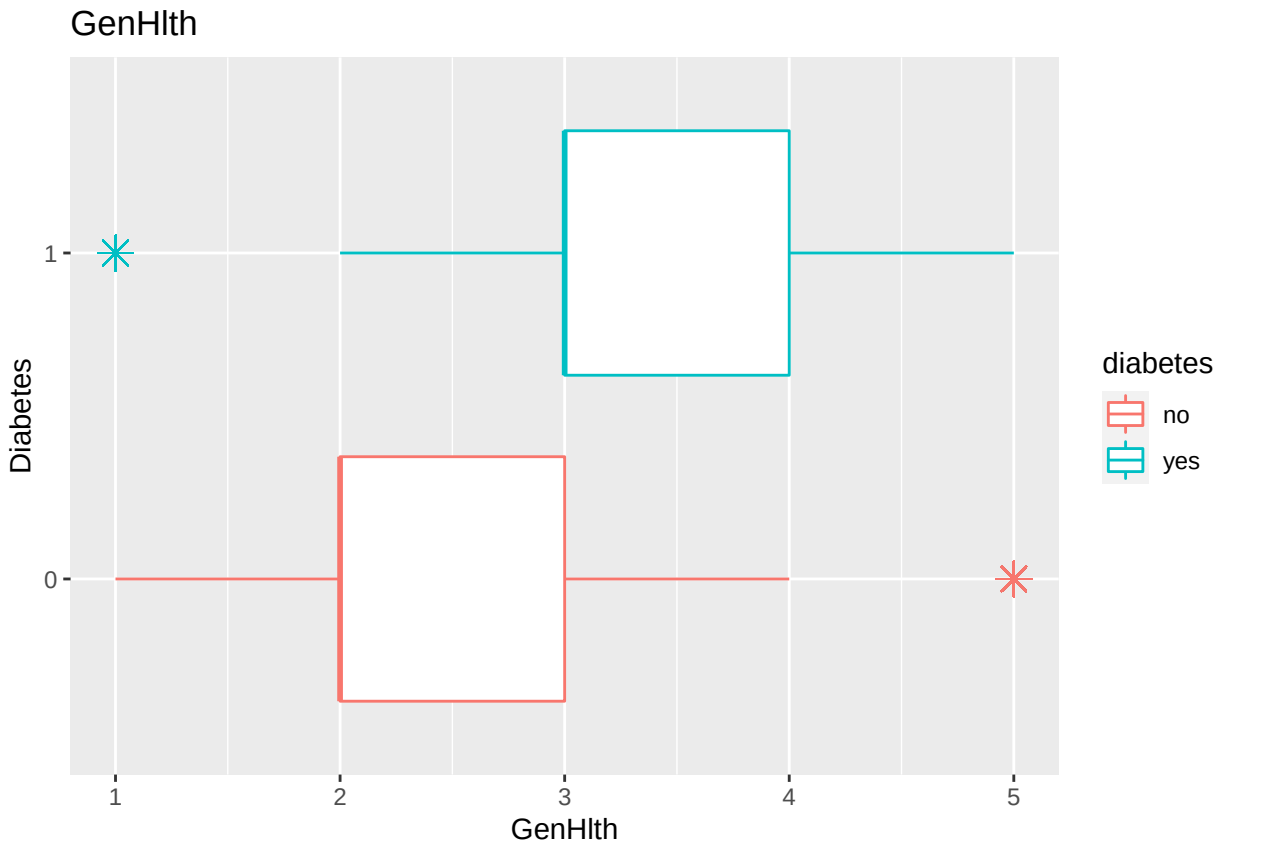
```
diabetes_labels <- c('no', 'yes')

p1 <- ggplot(data, aes(x = BMI, y=factor(diabetes), color=factor(diabetes))) +
  geom_boxplot(outlier.shape=8, outlier.size=4)+
  labs(title="BMI", y="Diabetes")+
  scale_color_discrete(name="diabetes", labels=diabetes_labels)
p2 <- ggplot(data, aes(x = GenHlth, y=factor(diabetes), color=factor(diabetes))) +
  geom_boxplot(outlier.shape=8, outlier.size=4)+
  labs(title="GenHlth",y="Diabetes")+
  scale_color_discrete(name="diabetes", labels=diabetes_labels)
p3 <- ggplot(data, aes(x = MentHlth, y=factor(diabetes), color=factor(diabetes))) +
  geom_boxplot(outlier.shape=8, outlier.size=4)+
  labs(title="MentHlth", y="Diabetes")+
  scale_color_discrete(name="diabetes", labels=diabetes_labels)
p4 <- ggplot(data, aes(x = PhysHlth, y=factor(diabetes), color=factor(diabetes))) +
  geom_boxplot(outlier.shape=8, outlier.size=4)+
  labs(title="PhysHlth",y="Diabetes")+
  scale_color_discrete(name="diabetes", labels=diabetes_labels)
p5 <- ggplot(data, aes(x = Age, y=factor(diabetes), color=factor(diabetes))) +
  geom_boxplot(outlier.shape=8, outlier.size=4)+
  labs(title="Age",y="Diabetes")+
  scale_color_discrete(name="diabetes", labels=diabetes_labels)
p6 <- ggplot(data, aes(x = Education, y=factor(diabetes), color=factor(diabetes))) +
  geom_boxplot(outlier.shape=8, outlier.size=4)+
  labs(title="Education", y="Diabetes")+
  scale_color_discrete(name="diabetes", labels=diabetes_labels)
```

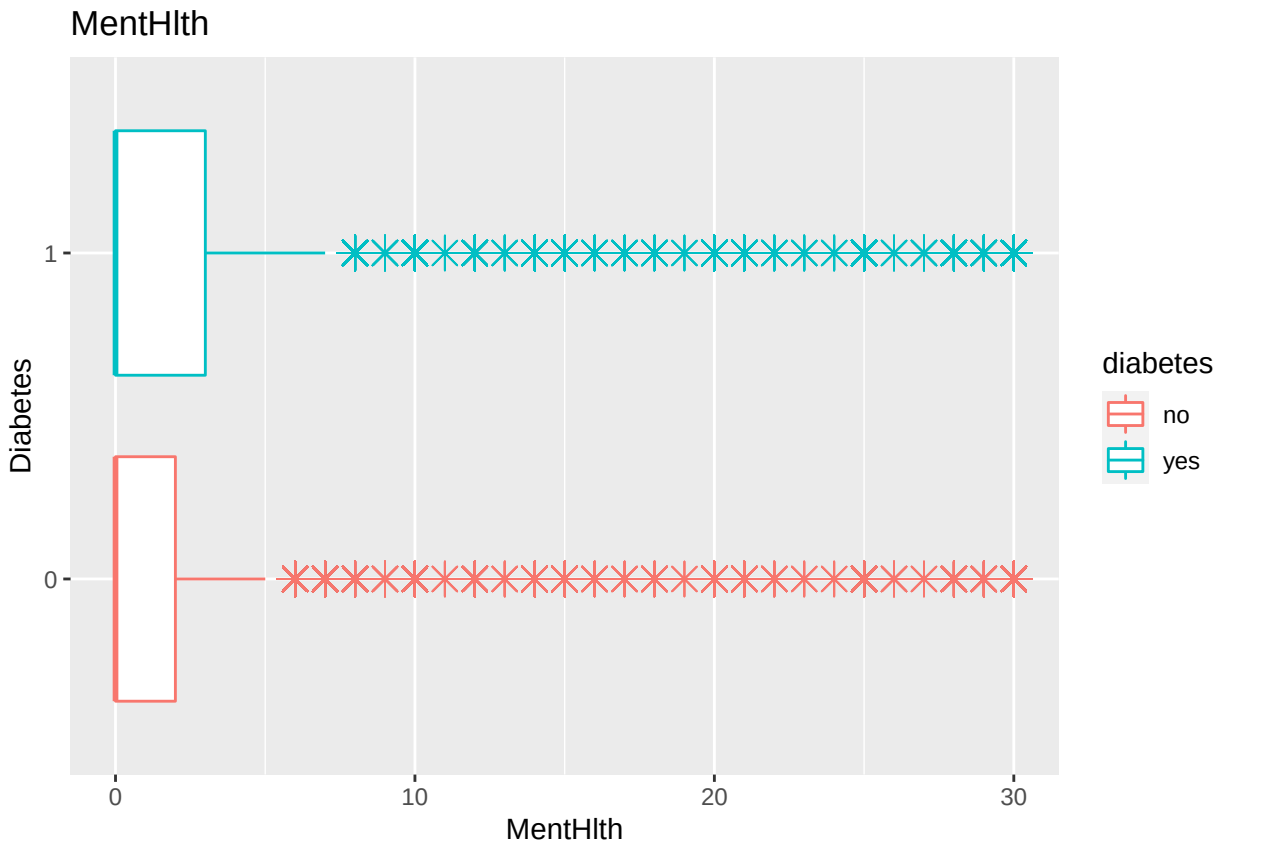
```
p7 <- ggplot(data, aes(x = Income, y=factor(diabetes), color=factor(diabetes))) +
  geom_boxplot(outlier.shape=8, outlier.size=4)+
  labs(title="Income", y="Diabetes")+
  scale_color_discrete(name="diabetes", labels=diabetes_labels)
p1
```



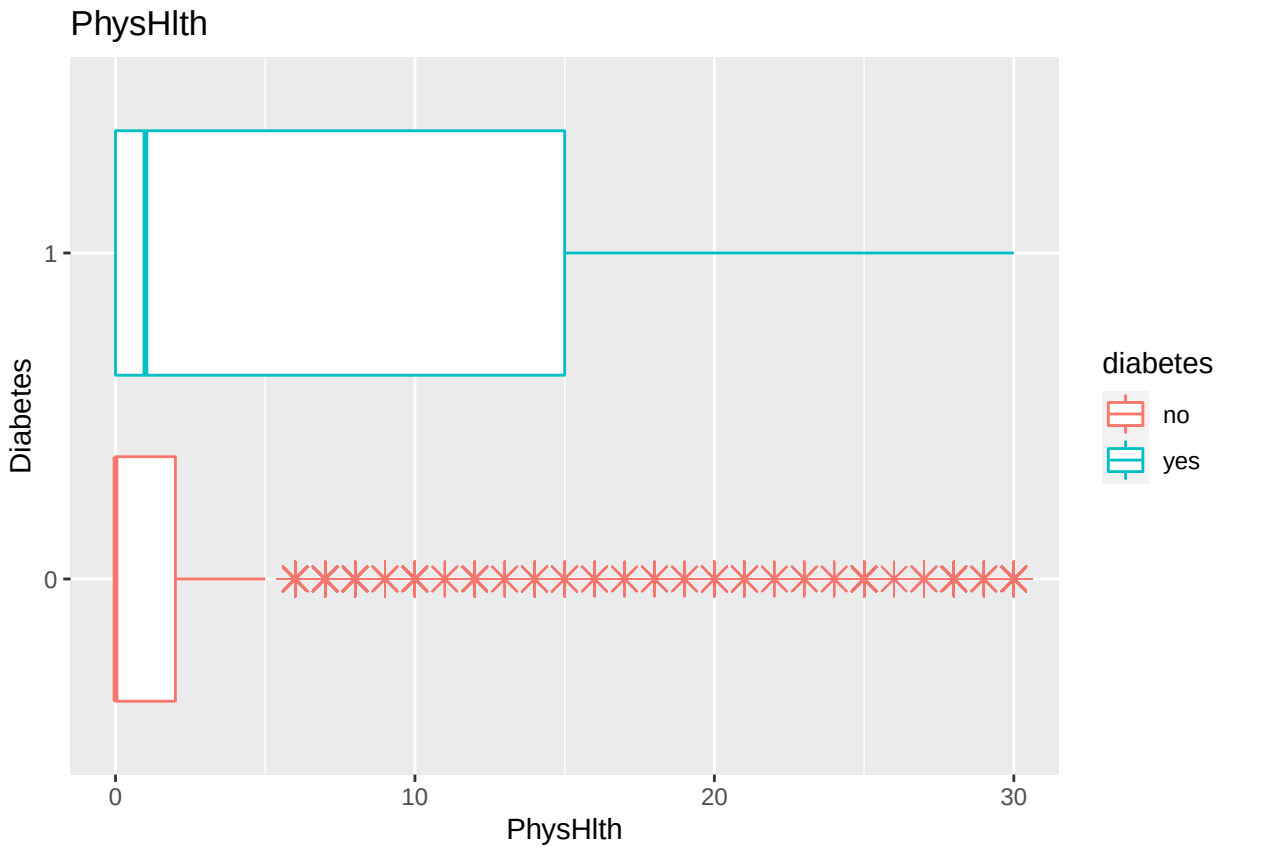

```
p2
```



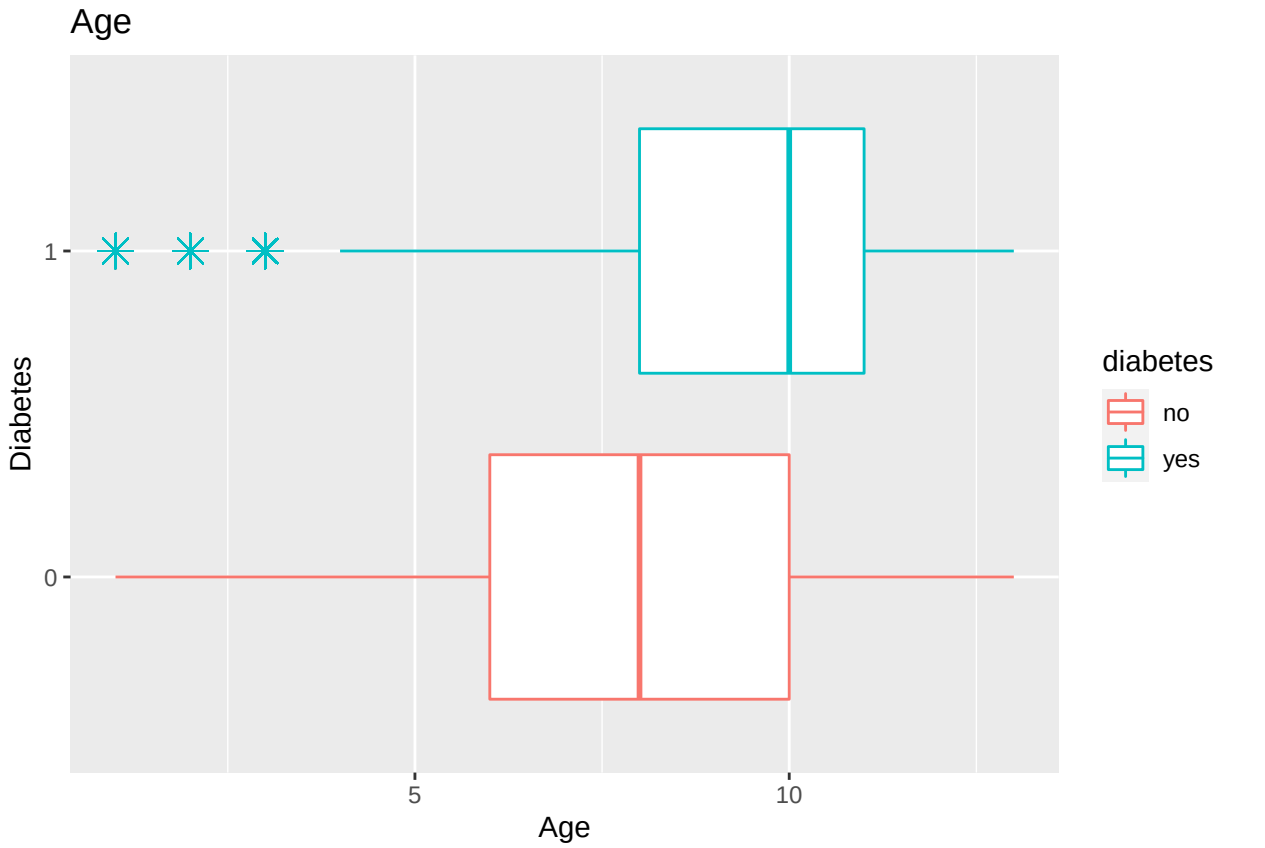

```
p3
```



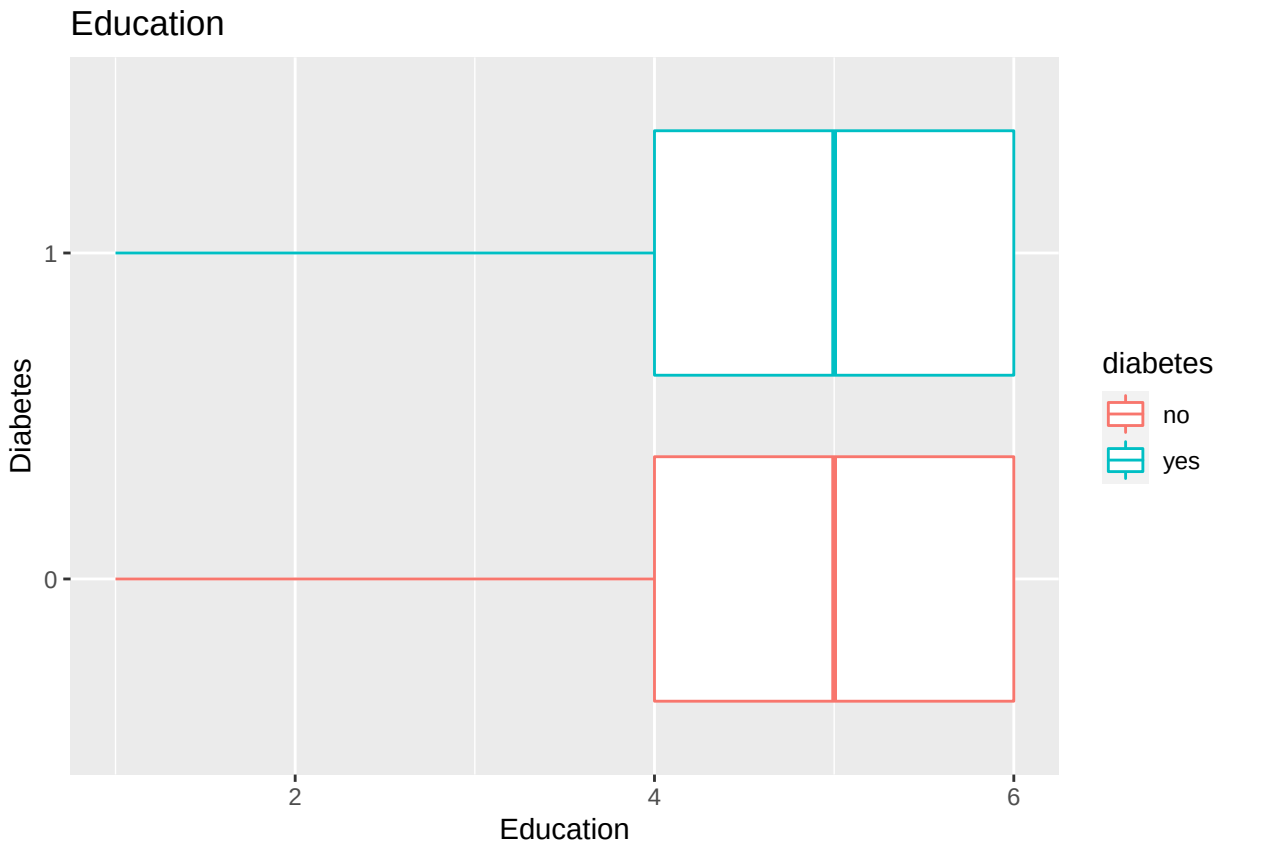

```
p4
```



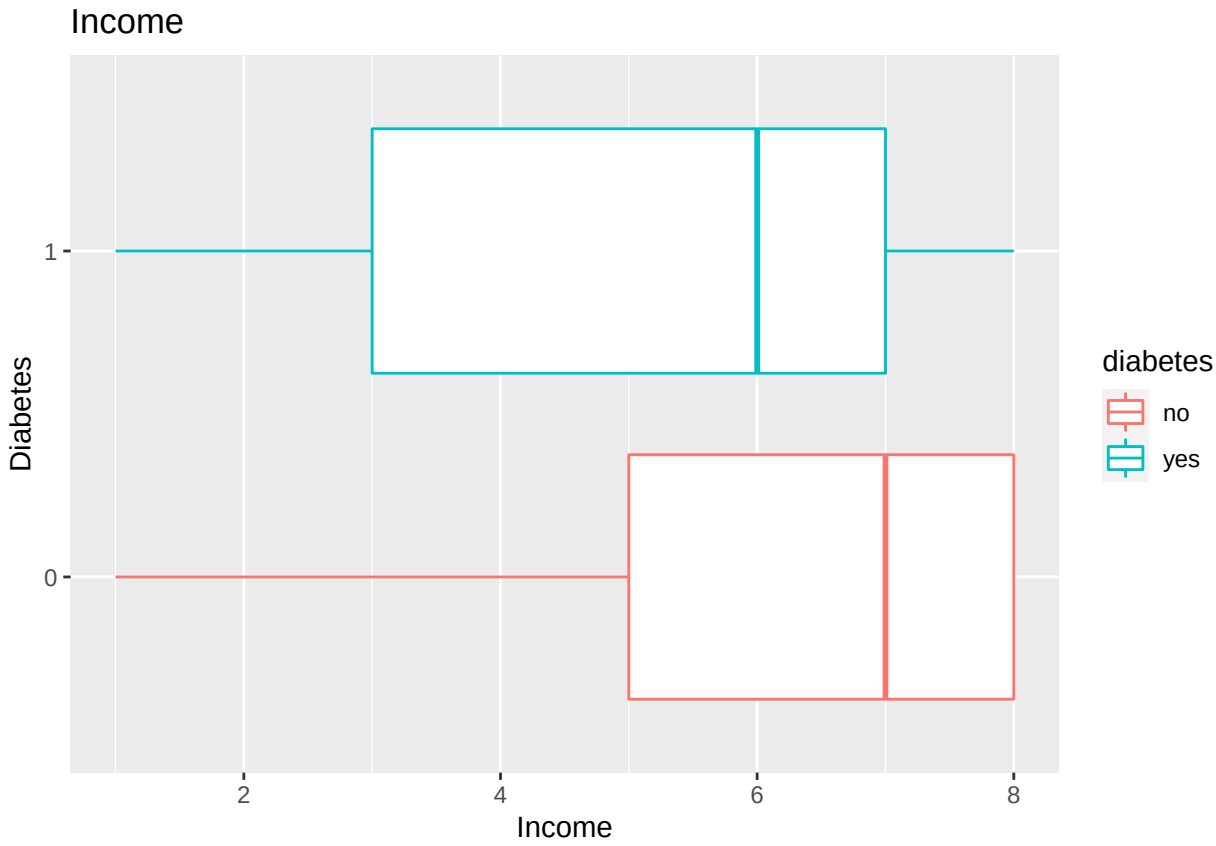

```
p5
```




```
p6
```



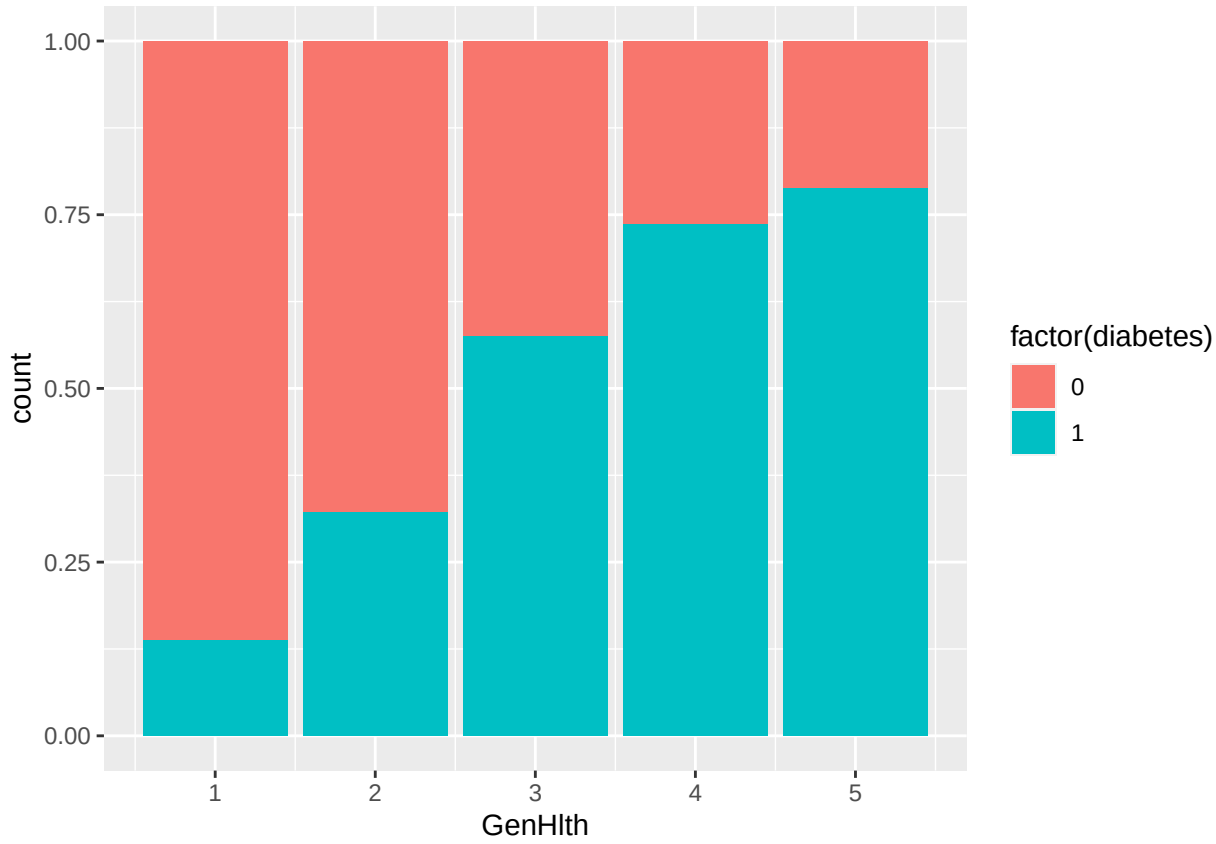

```
p7
```



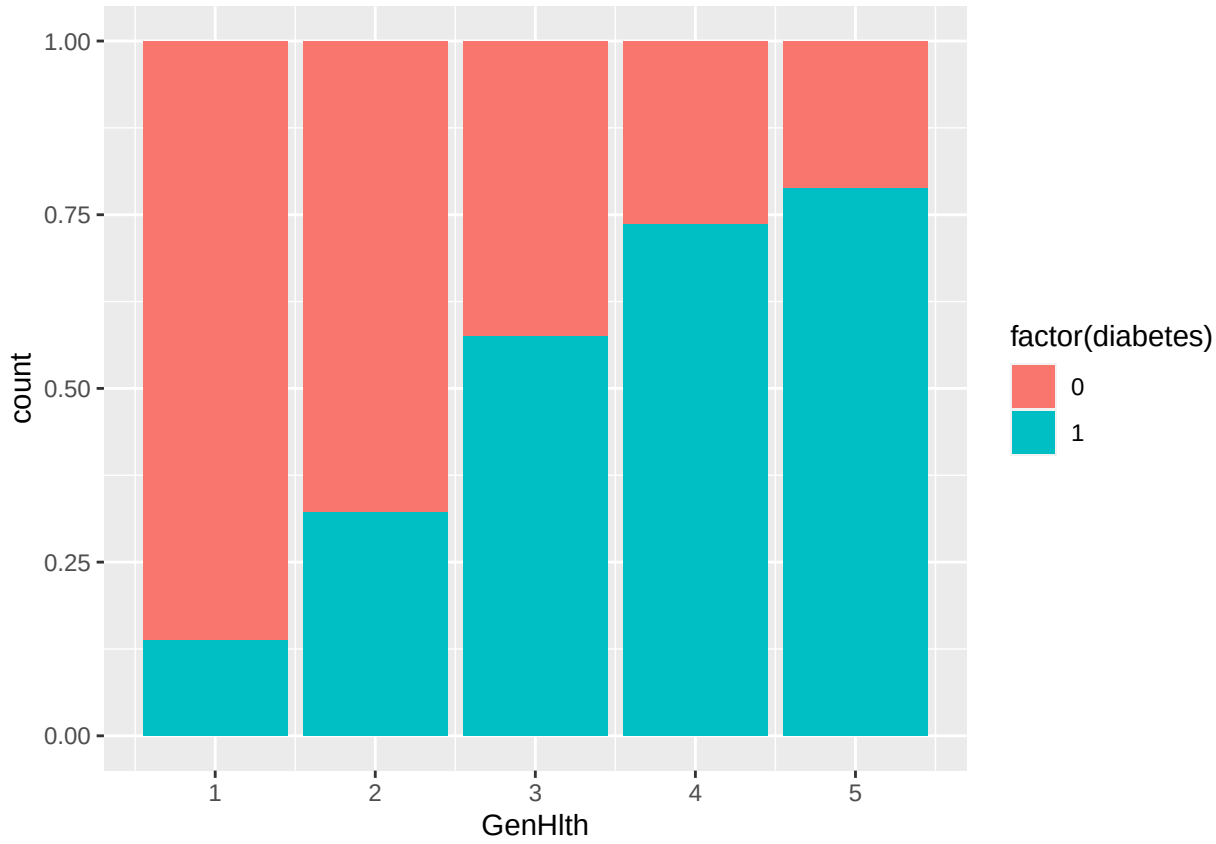
From the boxplots above, we see that the predictors BMI, MntHlth, and PhysHlth have a lot of outliers. All three distributions aee very skewed to the right. GenHlth and Age have only a couple outliers. Education and Income have none.

Now, we visulaize predictor distributions and relation to response.

```
ggplot(data, aes(x = GenHlth, fill = factor(diabetes))) +
  geom_bar(position="fill")
```

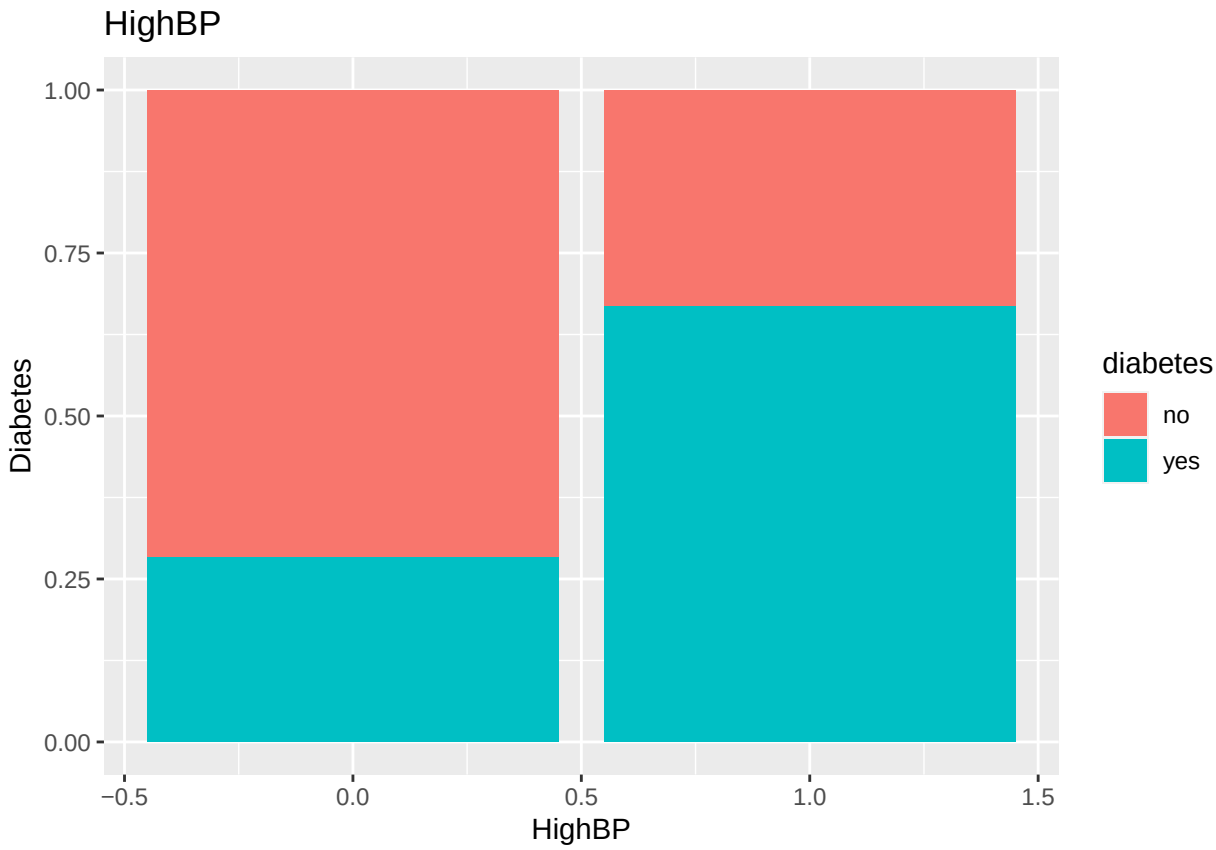



```
ggplot(data, aes(x = GenHlth, fill = factor(diabetes))) +
  geom_bar(position="fill")
```

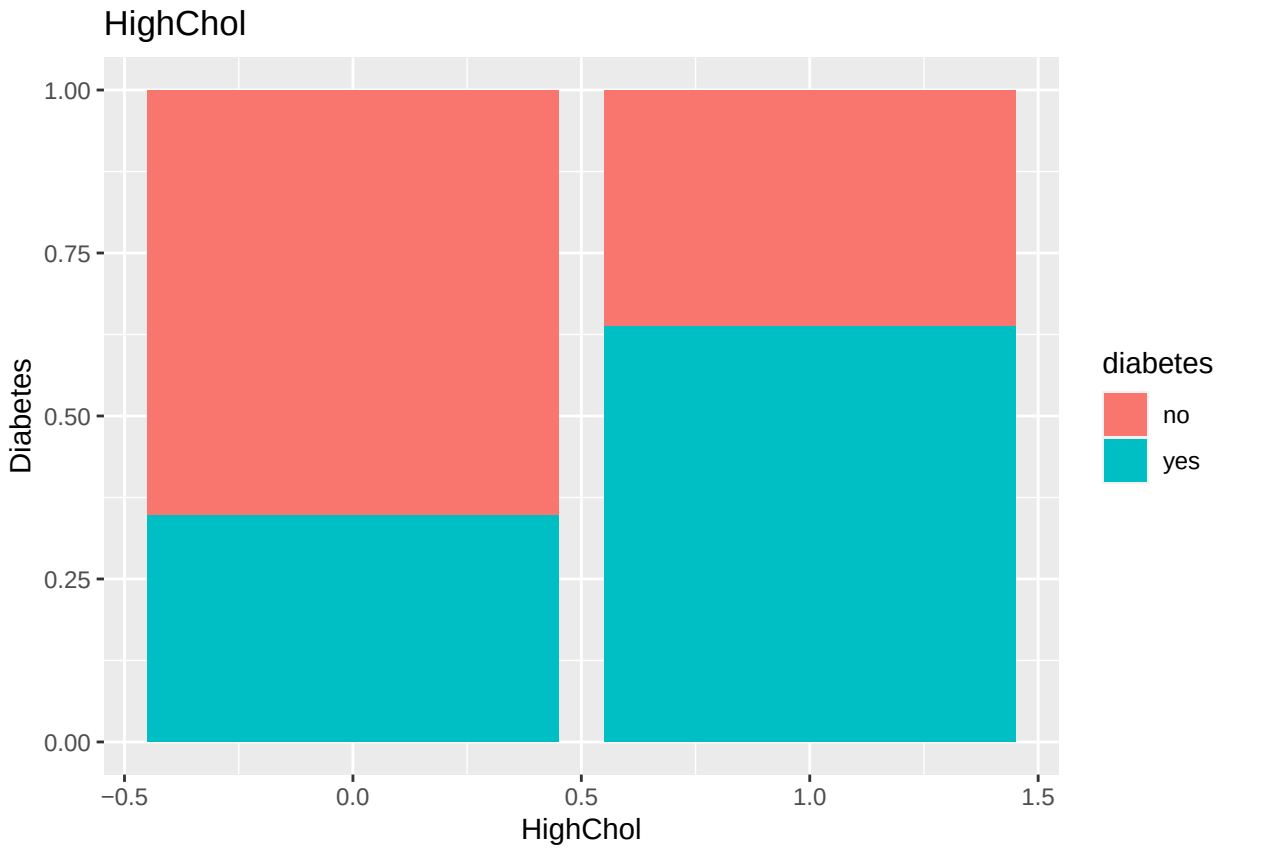



```
pbox1 <- ggplot(data, aes(x = HighBP, fill=factor(diabetes))) +
  geom_bar(position="fill")+
  labs(title="HighBP", y="Diabetes")+
  scale_fill_discrete(name="diabetes", labels=diabetes_labels)
pbox2 <- ggplot(data, aes(x = HighChol, fill=factor(diabetes))) +
  geom_bar(position="fill")+
  labs(title="HighChol", y="Diabetes")+
  scale_fill_discrete(name="diabetes", labels=diabetes_labels)
pbox3 <- ggplot(data, aes(x = CholCheck, fill=factor(diabetes))) +
  geom_bar(position="fill")+
  labs(title="CholCheck", y="Diabetes")+
  scale_fill_discrete(name="diabetes", labels=diabetes_labels)
pbox4 <- ggplot(data, aes(x = Smoker, fill=factor(diabetes))) +
  geom_bar(position="fill")+
  labs(title="Smoker", y="Diabetes")+
  scale_fill_discrete(name="diabetes", labels=diabetes_labels)
pbox5 <- ggplot(data, aes(x = Stroke, fill=factor(diabetes))) +
  geom_bar(position="fill")+
  labs(title="Stroke", y="Diabetes")+
  scale_fill_discrete(name="diabetes", labels=diabetes_labels)
pbox6 <- ggplot(data, aes(x = HeartDiseaseorAttack, fill=factor(diabetes))) +
  geom_bar(position="fill")+
  labs(title="HeartDiseaseorAttack", y="Diabetes")+
  scale_fill_discrete(name="diabetes", labels=diabetes_labels)
pbox7 <- ggplot(data, aes(x = PhysActivity, fill=factor(diabetes))) +
  geom_bar(position="fill")+
  labs(title="PhysActivity", y="Diabetes")+
```

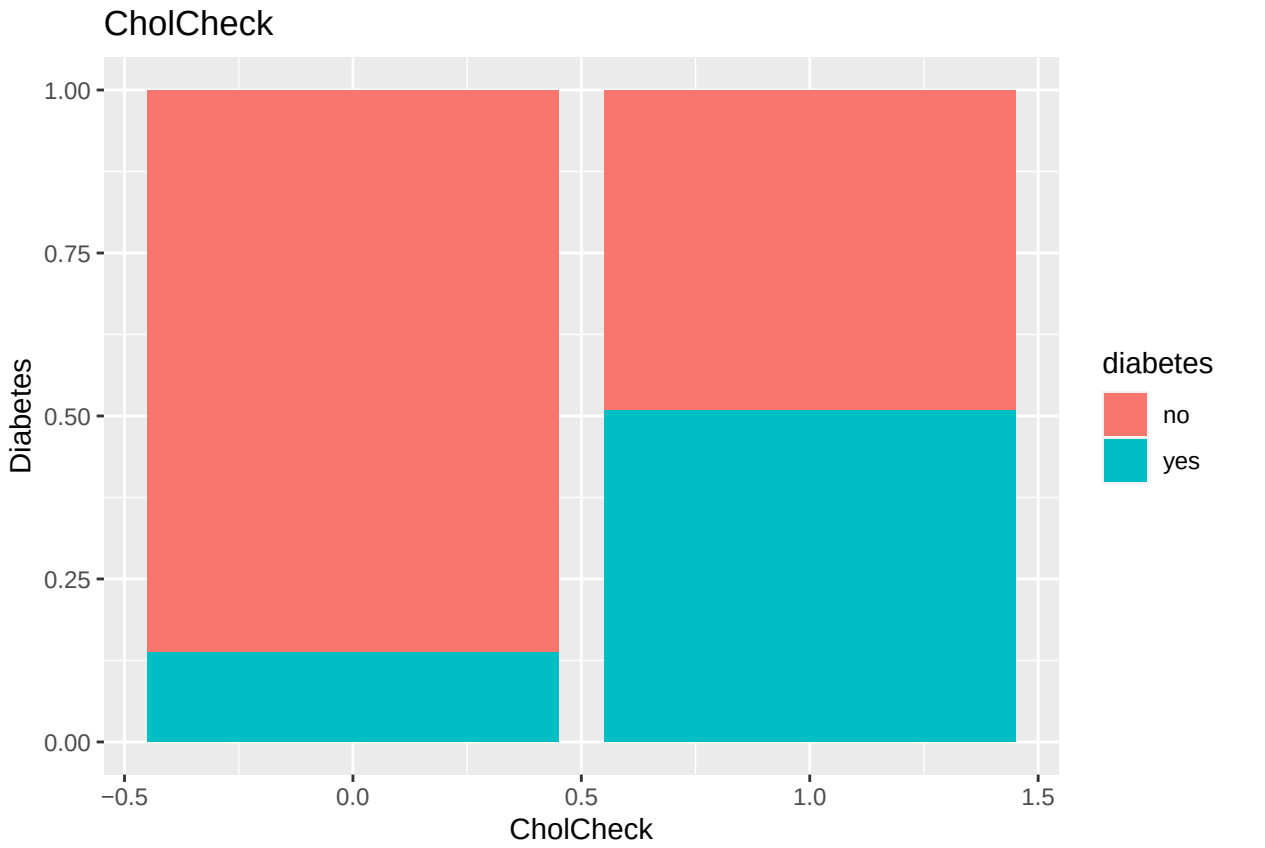
```
  scale_fill_discrete(name="diabetes", labels=diabetes_labels)
pbox8 <- ggplot(data, aes(x = Veggies, fill=factor(diabetes))) +
  geom_bar(position="fill")+
  labs(title="Veggies", y="Diabetes")+
  scale_fill_discrete(name="diabetes", labels=diabetes_labels)
pbox9 <- ggplot(data, aes(x = HvyAlcoholConsump, fill=factor(diabetes))) +
  geom_bar(position="fill")+
  labs(title="HvyAlcoholConsump", y="Diabetes")+
  scale_fill_discrete(name="diabetes", labels=diabetes_labels)
pbox10 <- ggplot(data, aes(x = AnyHealthcare, fill=factor(diabetes))) +
  geom_bar(position="fill")+
  labs(title="AnyHealthcare", y="Diabetes")+
  scale_fill_discrete(name="diabetes", labels=diabetes_labels)
pbox11 <- ggplot(data, aes(x = NoDocbcCost, fill=factor(diabetes))) +
  geom_bar(position="fill")+
  labs(title="NoDocbcCost", y="Diabetes")+
  scale_fill_discrete(name="diabetes", labels=diabetes_labels)
pbox12 <- ggplot(data, aes(x = DiffWalk, fill=factor(diabetes))) +
  geom_bar(position="fill")+
  labs(title="DiffWalk", y="Diabetes")+
  scale_fill_discrete(name="diabetes", labels=diabetes_labels)
pbox1
```



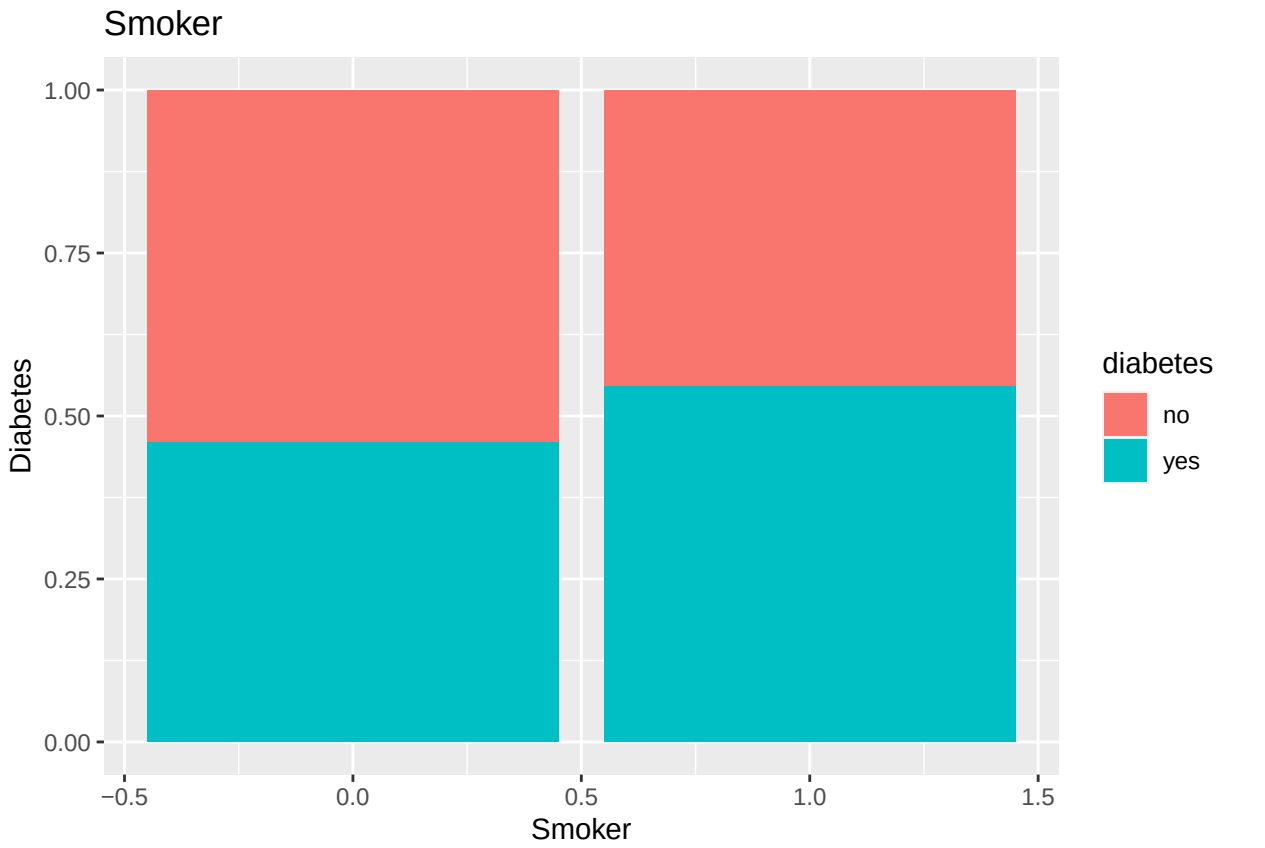

```
pbox2
```


HighChol
1.00
0.75
0.50
0.25
0.00
Diabetes
−0.5
0.0
0.5
1.0
1.5
HighChol
diabetes
no
yes

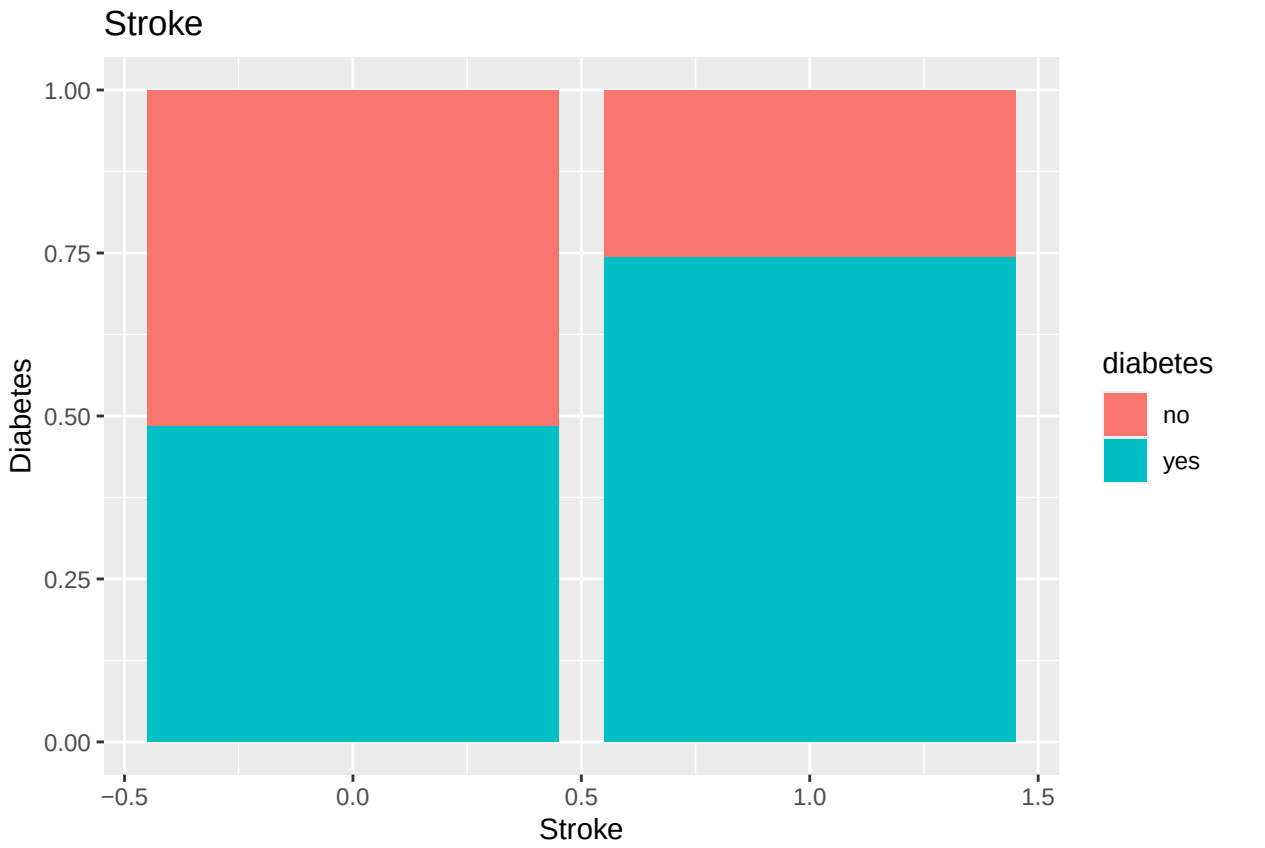
```
pbox3
```


CholCheck
1.00
0.75
0.50
0.25
0.00
Diabetes
−0.5
0.0
0.5
1.0
1.5
CholCheck
diabetes
no
yes

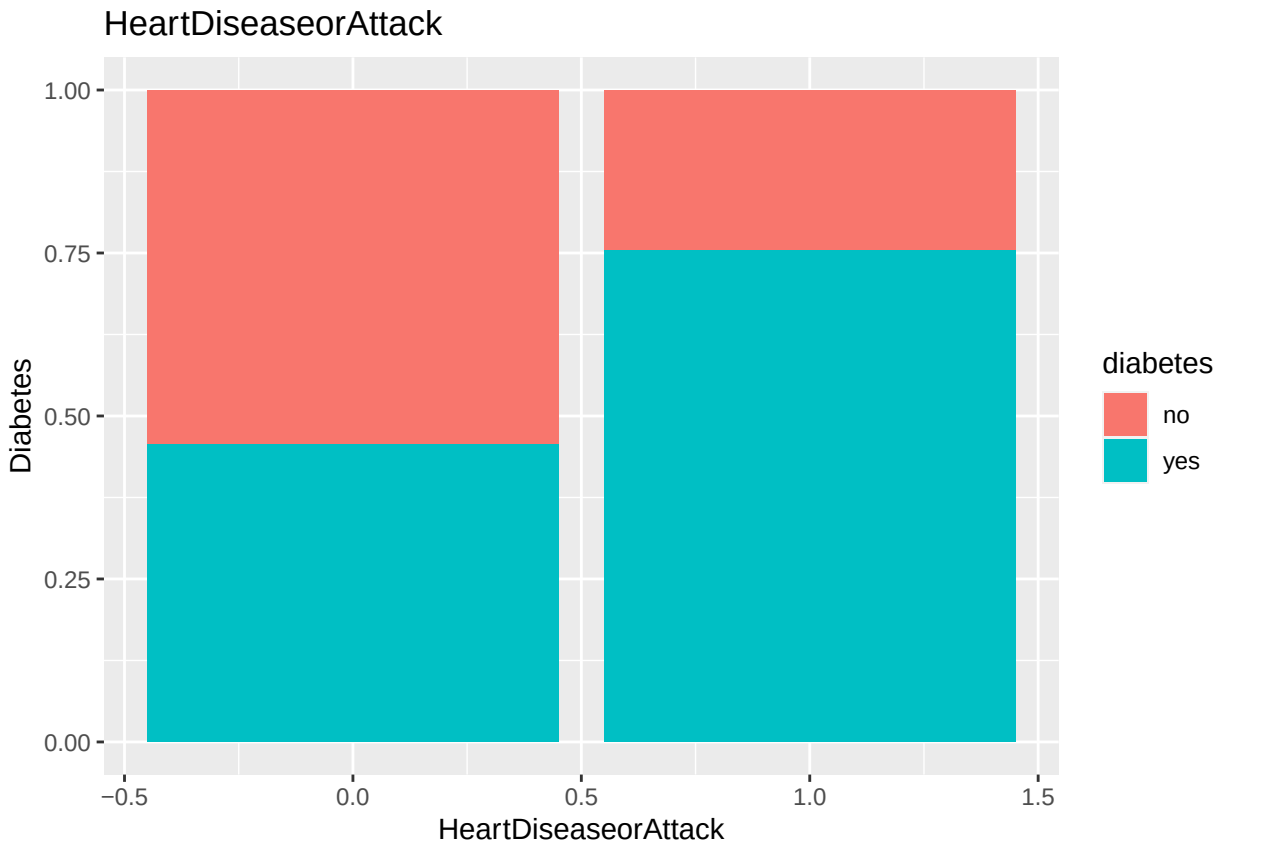
```
pbox4
```


Smoker
1.00
0.75
0.50
0.25
0.00
Diabetes
−0.5
0.0
0.5
1.0
1.5
Smoker
diabetes
no
yes

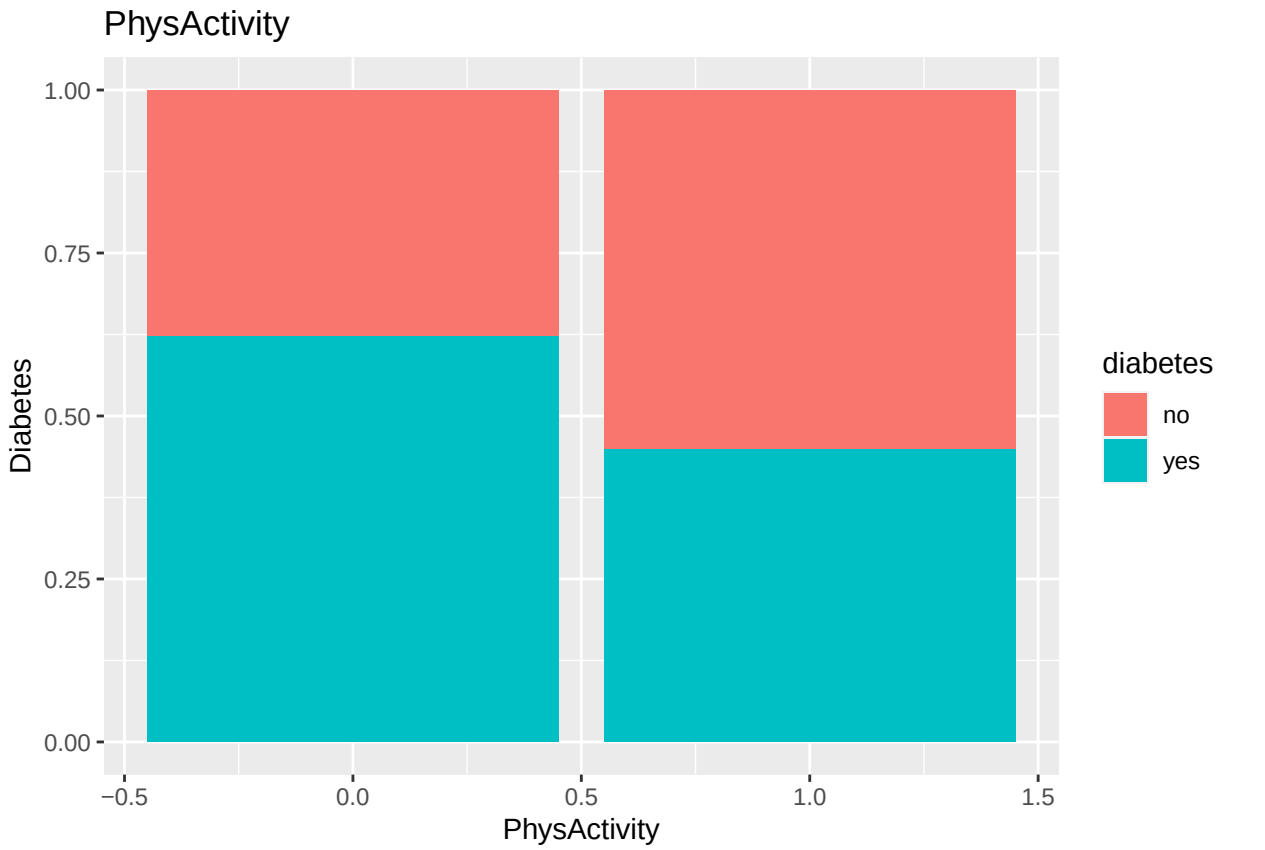
```
pbox5
```


Stroke
1.00
0.75
0.50
0.25
0.00
Diabetes
−0.5
0.0
0.5
1.0
1.5
Stroke
diabetes
no
yes

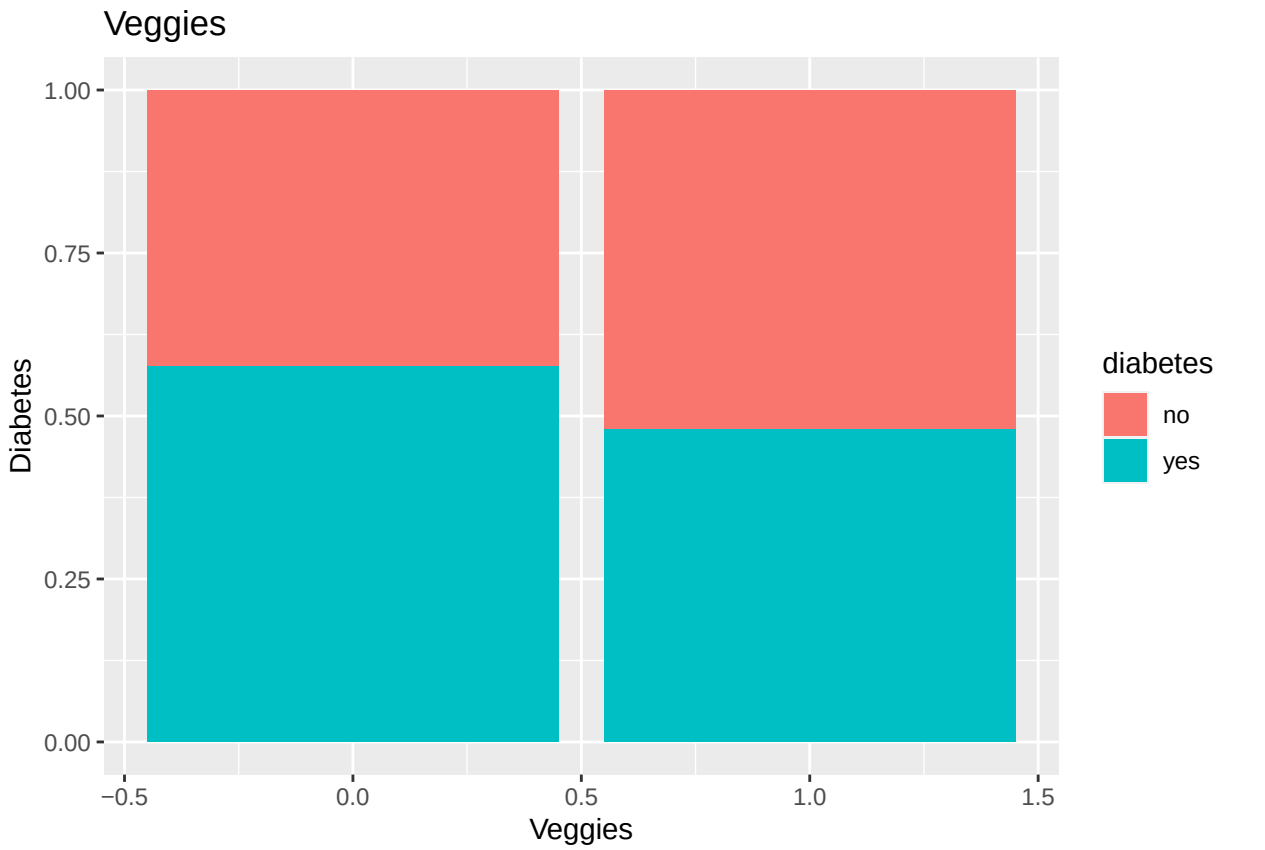
```
pbox6
```


HeartDiseaseorAttack
1.00
0.75
0.50
0.25
0.00
Diabetes
−0.5
0.0
0.5
1.0
1.5
HeartDiseaseorAttack
diabetes
no
yes

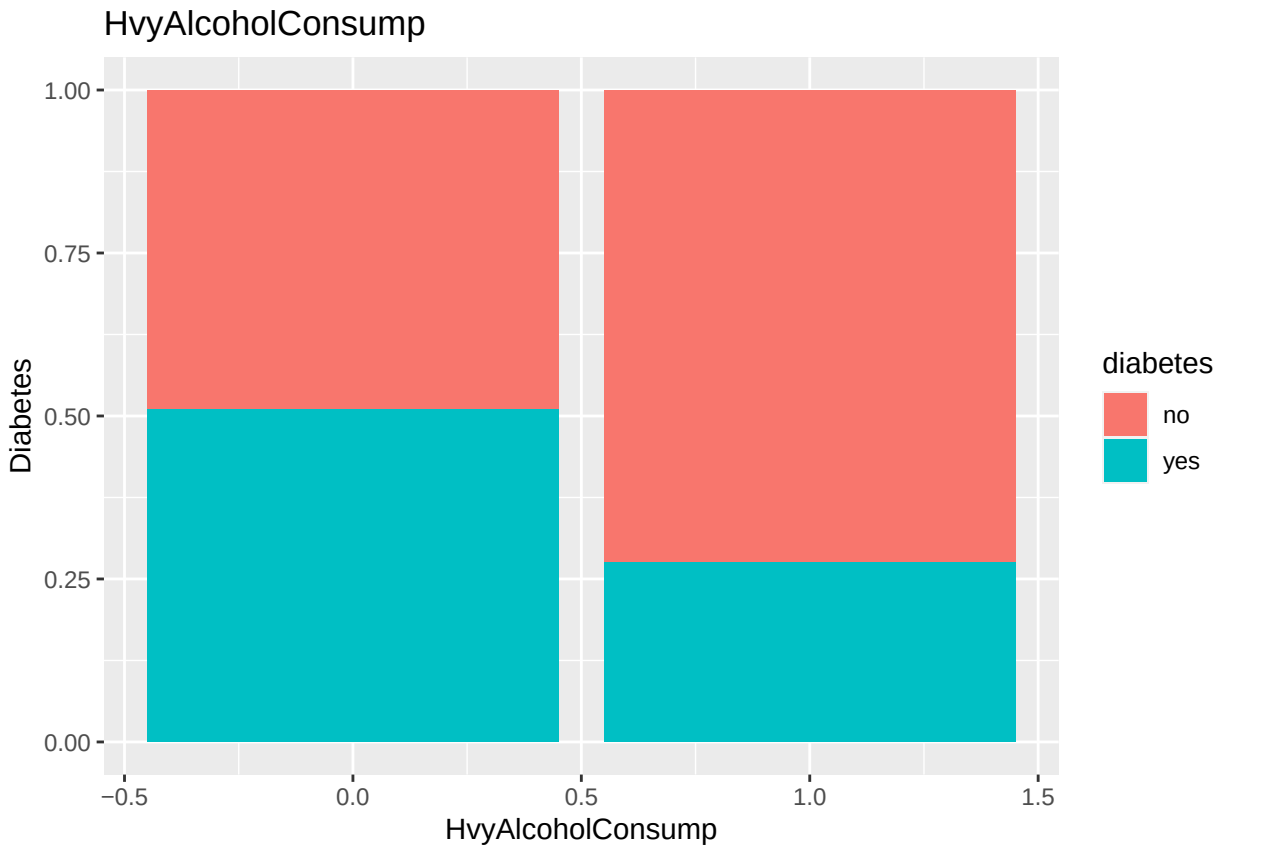
```
pbox7
```


PhysActivity
1.00
0.75
0.50
0.25
0.00
Diabetes
−0.5
0.0
0.5
1.0
1.5
PhysActivity
diabetes
no
yes

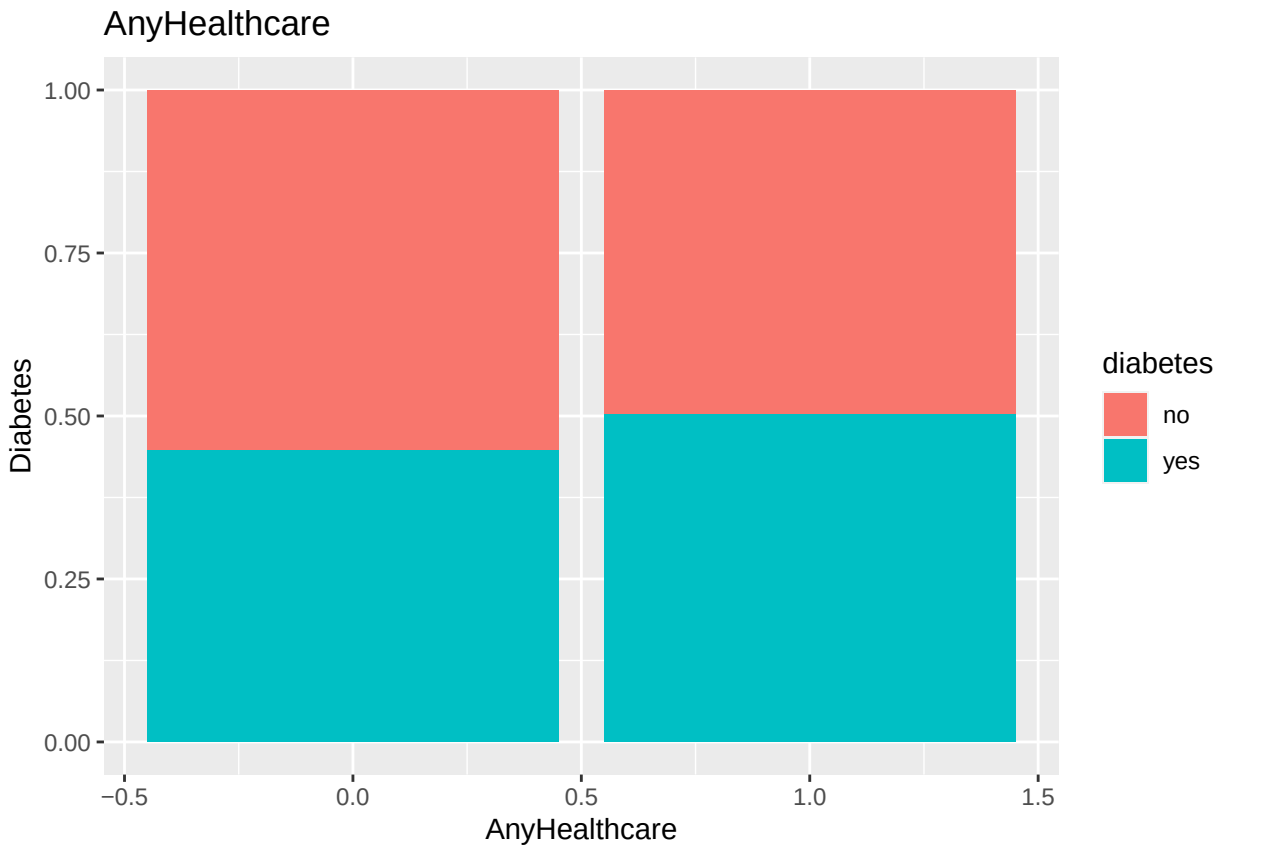
```
pbox8
```


Veggies
1.00
0.75
0.50
0.25
0.00
Diabetes
−0.5
0.0
0.5
1.0
1.5
Veggies
diabetes
no
yes

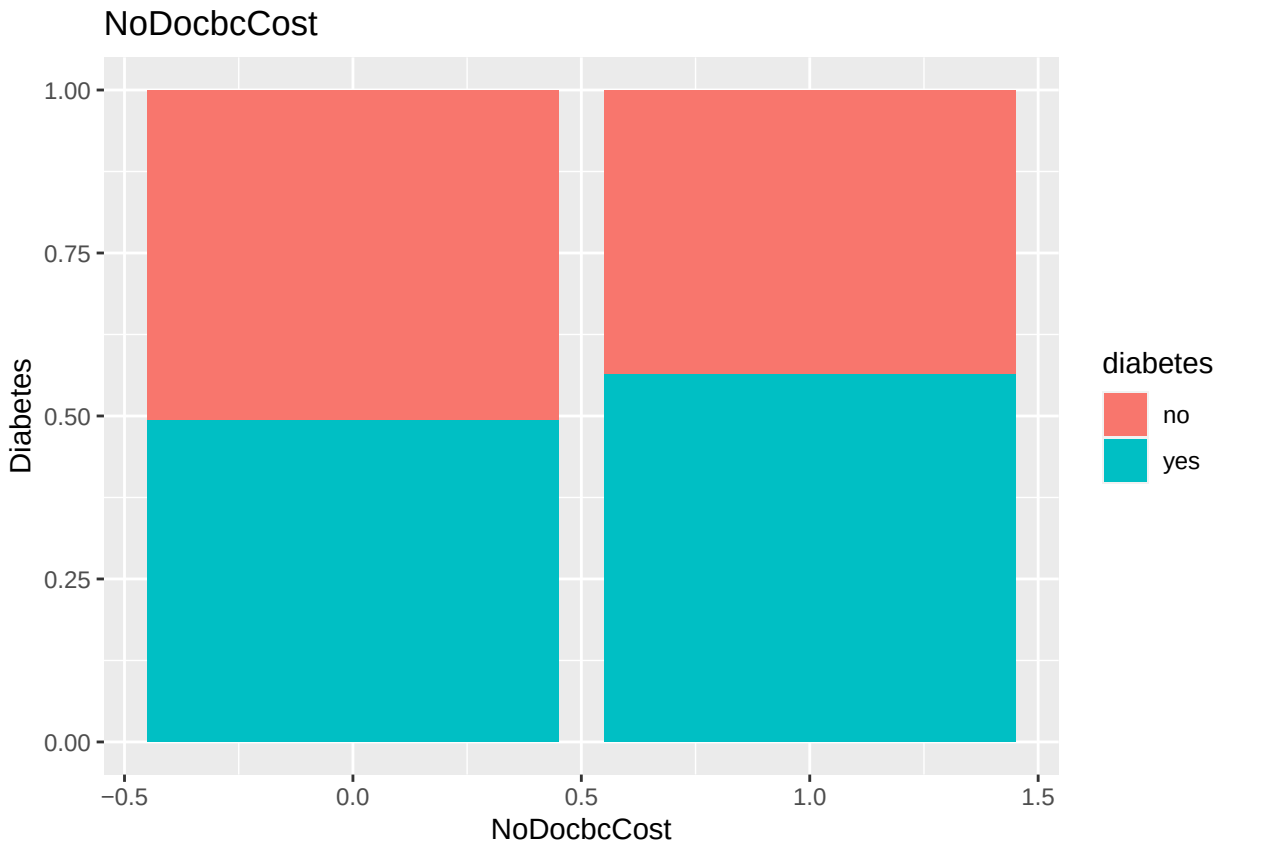
```
pbox9
```


HvyAlcoholConsump
1.00
0.75
0.50
0.25
0.00
Diabetes
−0.5
0.0
0.5
1.0
1.5
HvyAlcoholConsump
diabetes
no
yes

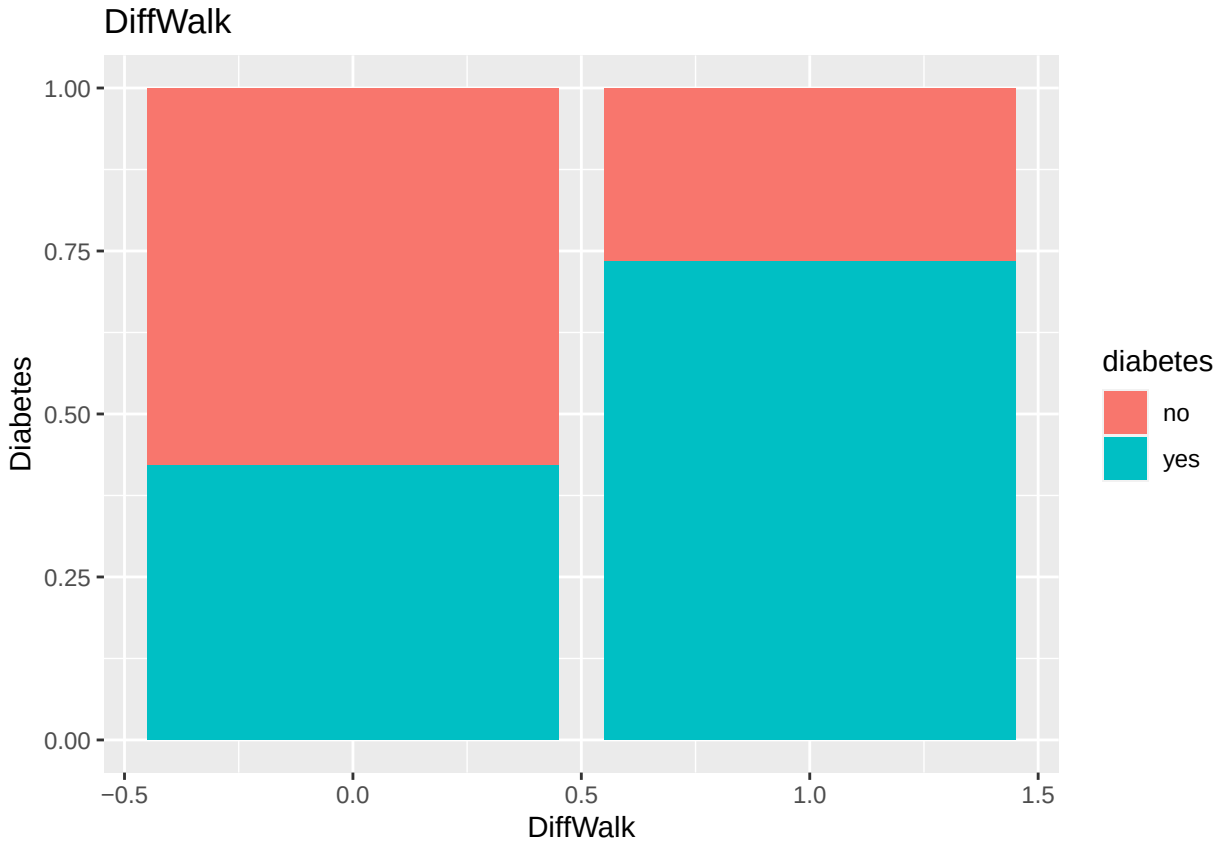
```
pbox10
```


AnyHealthcare
1.00
0.75
0.50
0.25
0.00
Diabetes
−0.5
0.0
0.5
1.0
1.5
AnyHealthcare
diabetes
no
yes

```
pbox11
```

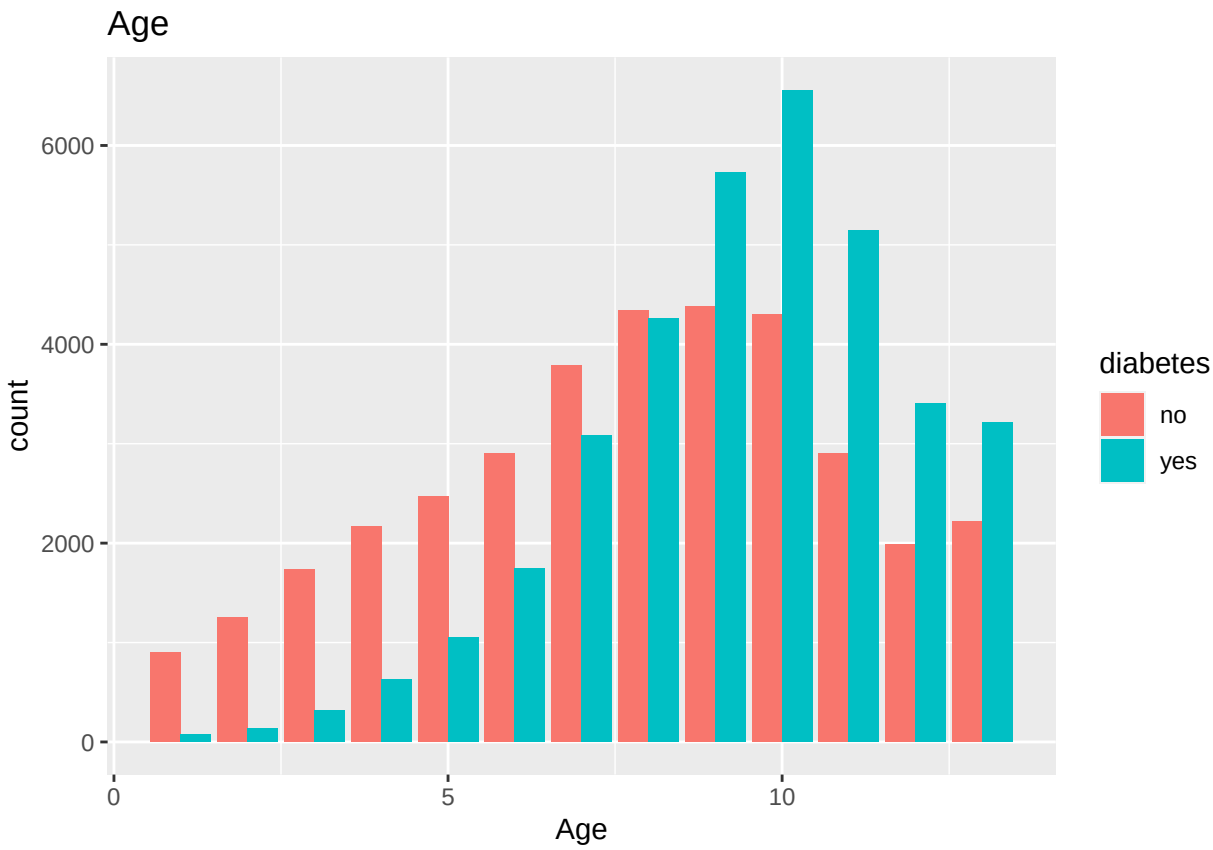

NoDocbcCost
1.00
0.75
0.50
0.25
0.00
Diabetes
−0.5
0.0
0.5
1.0
1.5
NoDocbcCost
diabetes
no
yes

```
pbox12
```



Next, look at "Age" and its relation to response (diabetes diagnosis):

```
ggplot(data, aes(x = Age, fill=factor(diabetes))) +
  geom_bar(position="dodge")+
  labs(title="Age")+
  scale_fill_discrete(name="diabetes", labels=diabetes_labels)
```



Factor Numeric Variables

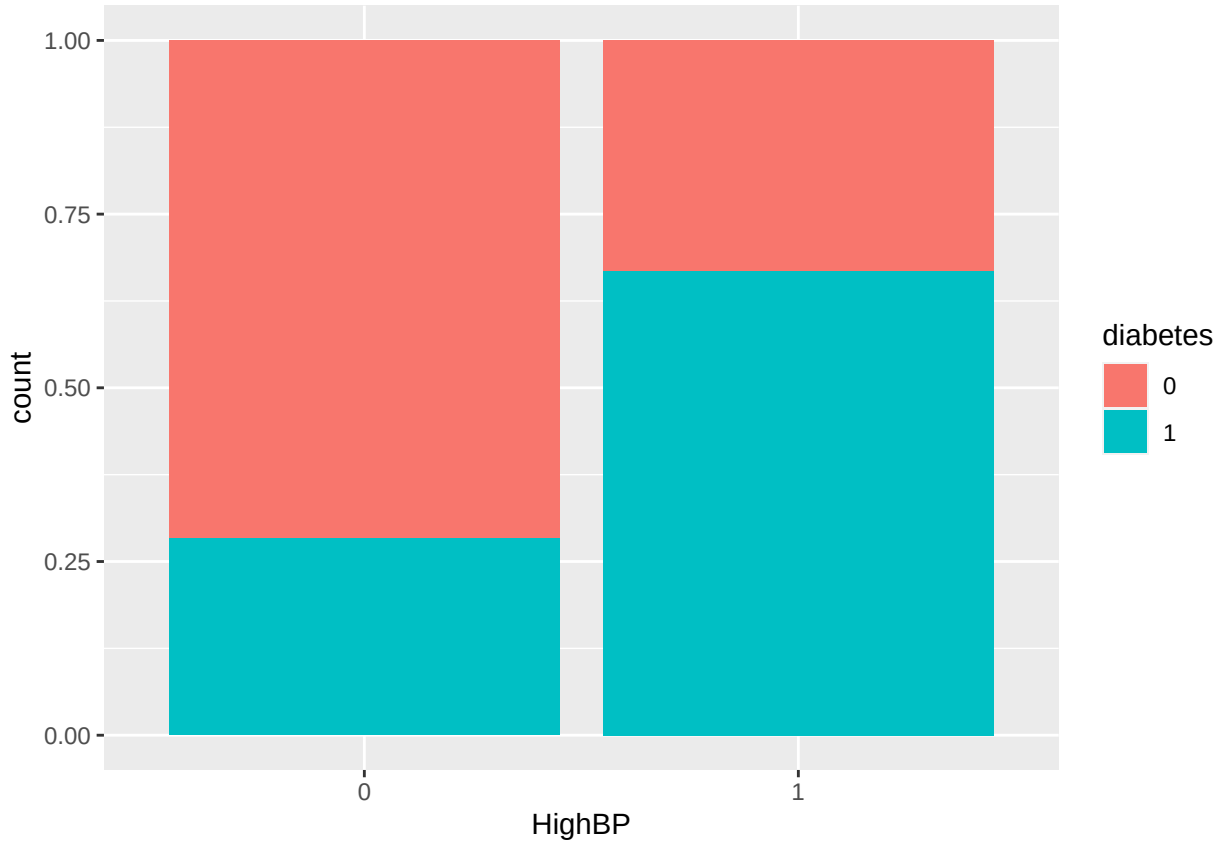
```
factored <- data
```

```
factored$diabetes <- as.factor(factored$diabetes)
factored$HighBP <- as.factor(factored$HighBP)
factored$CholCheck <- as.factor(factored$CholCheck)
factored$Smoker <- as.factor(factored$Smoker)
factored$Stroke <- as.factor(factored$Stroke)
factored$HeartDiseaseorAttack <- as.factor(factored$HeartDiseaseorAttack)
factored$PhysActivity <- as.factor(factored$PhysActivity)
factored$Fruits <- as.factor(factored$Fruits)
factored$Veggies <- as.factor(factored$Veggies)
factored$HvyAlcoholConsump <- as.factor(factored$HvyAlcoholConsump)
factored$AnyHealthcare <- as.factor(factored$AnyHealthcare)
factored$NoDocbcCost <- as.factor(factored$NoDocbcCost)
factored$GenHlth <- as.factor(factored$GenHlth)
factored$MentHlth <- as.factor(factored$MentHlth)
factored$DiffWalk <- as.factor(factored$DiffWalk)
factored$Sex <- as.factor(factored$Sex)
factored$Age <- as.factor(factored$Age)
factored$Education <- as.factor(factored$Education)
factored$Income <- as.factor(factored$Income)
```

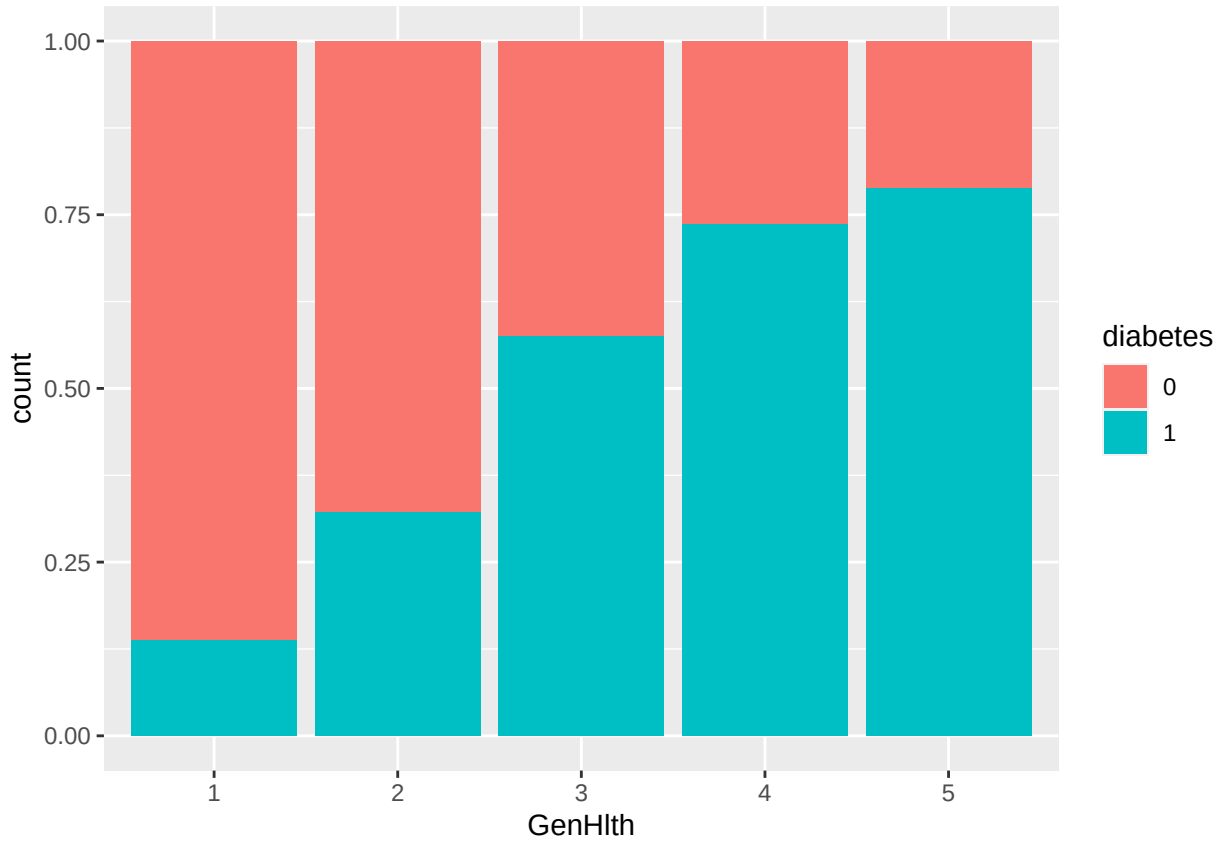
EDA of factored binary outcome dataset

Next, look at plots of 2 most correlated predictors and color by outcome.

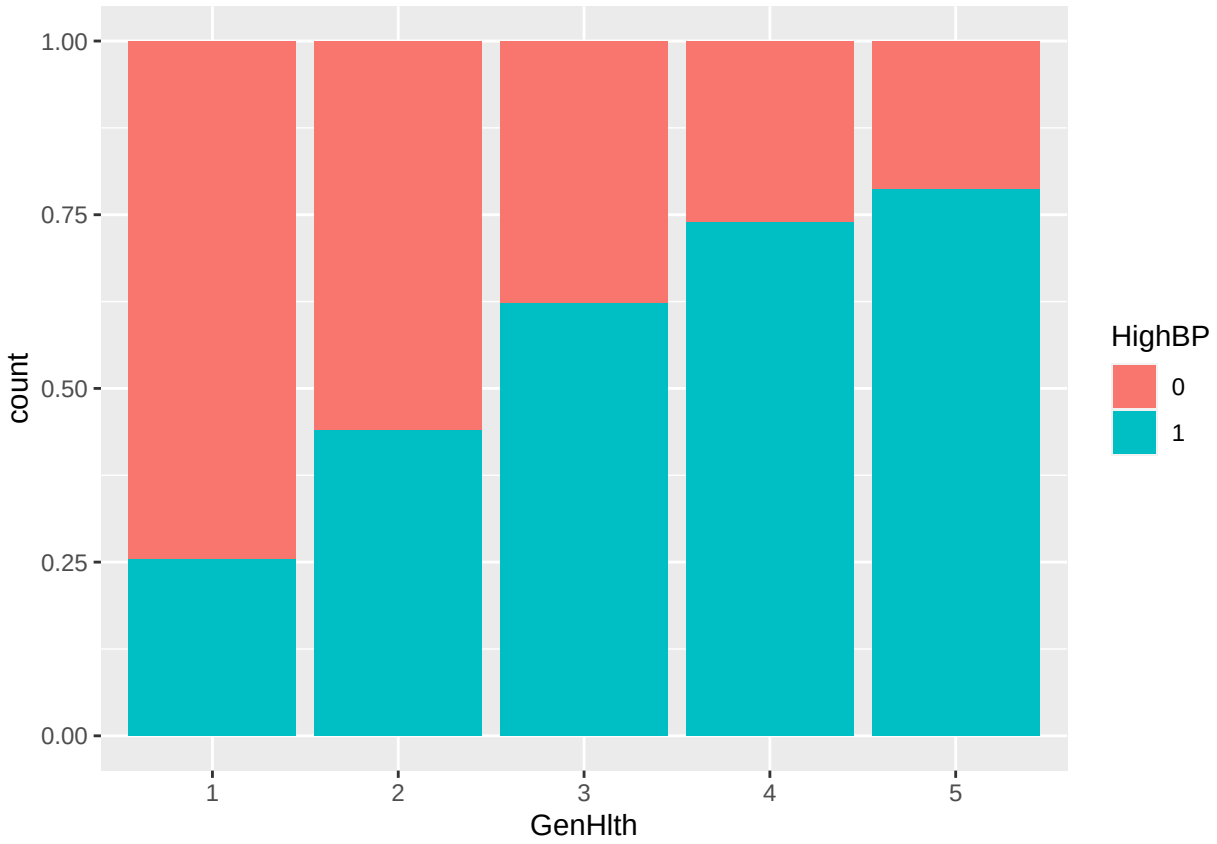
```
ggplot(factored, aes(x = HighBP, fill = diabetes)) +
  geom_bar(position="fill")
```




```
ggplot(factored, aes(x = GenHlth, fill = diabetes)) +
  geom_bar(position="fill")
```




```
ggplot(factored, aes(x = GenHlth, fill = HighBP)) +
  geom_bar(position="fill")
```



Modeling

Split data train and test

```
set.seed(17)
sample <- sample(c(TRUE, FALSE), nrow(factored), replace=TRUE, prob=c(0.7,0.3))
train  <- factored[sample, ]
test   <- factored[!sample, ]
```

Logistic Regression

```
glm.fit.all <- glm(diabetes ~ HighBP+ HighChol + CholCheck + HeartDiseaseorAttack + AnyHealthcare
+ PhysActivity + HvyAlcoholConsump + Fruits + Veggies + GenHlth + DiffWalk + Sex + Income + Education +
               data = factored, family = binomial)
summary(glm.fit.all)
```

```
##
## Call:
## glm(formula = diabetes ~ HighBP + HighChol + CholCheck + HeartDiseaseorAttack +
##     AnyHealthcare + PhysActivity + HvyAlcoholConsump + Fruits +
##     Veggies + GenHlth + DiffWalk + Sex + Income + Education +
##     BMI + PhysHlth, family = binomial, data = factored)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -3.3594  -0.8272  -0.0158   0.8429   2.9260
##
```

```
## Coefficients:
##                          Estimate Std. Error z value Pr(>|z|)
## (Intercept)            -5.667666   0.294814 -19.225  < 2e-16 ***
## HighBP1                 0.920575   0.019051  48.321  < 2e-16 ***
## HighChol                0.659898   0.018564  35.547  < 2e-16 ***
## CholCheck1              1.461079   0.079652  18.343  < 2e-16 ***
## HeartDiseaseorAttack1   0.445041   0.027861  15.974  < 2e-16 ***
## AnyHealthcare1          0.316080   0.044945   7.033 2.03e-12 ***
## PhysActivity1          -0.070266   0.021035  -3.340 0.000837 ***
## HvyAlcoholConsump1     -0.827859   0.048268 -17.151  < 2e-16 ***
## Fruits1                 0.036607   0.019284   1.898 0.057654 .
## Veggies1               -0.054404   0.023033  -2.362 0.018177 *
## GenHlth2                0.738827   0.037936  19.475  < 2e-16 ***
## GenHlth3                1.425295   0.037527  37.980  < 2e-16 ***
## GenHlth4                1.841166   0.042970  42.847  < 2e-16 ***
## GenHlth5                1.942993   0.056634  34.308  < 2e-16 ***
## DiffWalk1               0.272619   0.025425  10.722  < 2e-16 ***
## Sex1                    0.277096   0.018800  14.739  < 2e-16 ***
## Income2                 0.110355   0.055291   1.996 0.045944 *
## Income3                 0.070587   0.052618   1.341 0.179764
## Income4                 0.067226   0.051048   1.317 0.187867
## Income5                -0.020916   0.049907  -0.419 0.675146
## Income6                -0.092623   0.048909  -1.894 0.058250 .
## Income7                -0.153540   0.049106  -3.127 0.001768 **
## Income8                -0.372086   0.048247  -7.712 1.24e-14 ***
## Education2              0.312773   0.280893   1.113 0.265495
## Education3              0.158108   0.277145   0.570 0.568346
## Education4              0.072587   0.274702   0.264 0.791597
## Education5              0.081152   0.274825   0.295 0.767775
## Education6              0.004384   0.274969   0.016 0.987279
## BMI                     0.060327   0.001496  40.323  < 2e-16 ***
## PhysHlth               -0.005416   0.001198  -4.521 6.15e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 98000  on 70691  degrees of freedom
## Residual deviance: 73819  on 70662  degrees of freedom
## AIC: 73879
##
## Number of Fisher Scoring iterations: 5
```

```
glm.probs.all <- predict(glm.fit.all, type = "response")
glm.probs.all[1:10]
```

```
##          1          2          3          4          5          6          7
## 0.41243806 0.65243374 0.07210192 0.64782442 0.14415449 0.07236625 0.09010003
##          8          9         10
## 0.55558375 0.27080847 0.38466927
```

```
glm.pred.all <- rep(0, length(factored$diabetes))
glm.pred.all[glm.probs.all > 0.5] <- 1
```

```
table(glm.pred.all, factored$diabetes)
```

```
##
## glm.pred.all     0     1
##            0 25650  8581
##            1  9696 26765
```

```
accuracy <- sum(diag(table(glm.pred.all, factored$diabetes)))/nrow(factored)
accuracy
```

```
## [1] 0.7414559
```

Now make model based off of training data:

```
glm.fit.trainall <- glm(diabetes ~ HighBP+ HighChol + CholCheck + HeartDiseaseorAttack + AnyHealthcare
+ PhysActivity + HvyAlcoholConsump + Fruits + Veggies + GenHlth + DiffWalk + Sex + Income + Education +
               data = train, family = binomial)
glm.probs.trainall <- predict(glm.fit.trainall, test, type = "response")
```

```
glm.pred.trainall <- rep(0, length(test))
glm.pred.trainall[glm.probs.trainall > 0.5] <- 1
table(glm.pred.trainall, test$diabetes)
```

```
##
## glm.pred.trainall    0    1
##                 0   18    0
##                 1 2917 7957
```

```
accuracy <- sum(diag(table(glm.pred.trainall, test$diabetes)))/nrow(test)
accuracy
```

```
## [1] 0.376659
```

To improve the accuracy we will consider a subset of predictors. Look at correlations to decide. The most correlated to diabetes are GenHlth and HighBP.

```
glm.fit.cor <- glm(diabetes ~ GenHlth + HighBP, data=train, family = binomial)
glm.probs.cor <- predict(glm.fit.cor, test, type = "response")
glm.pred.cor <- rep("no diabetes", length(test))
glm.pred.cor[glm.probs.cor > 0.5] <- "diabetes"
table(glm.pred.cor, test$diabetes)
```

```
##
## glm.pred.cor     0    1
##   diabetes    2795 7243
##   no diabetes   18    0
```

```
Accuracy <- (0+5)/(1+18+5+0)
Accuracy
```

```
## [1] 0.2083333
```

The subset of predictors made our predictive performance worse.

KNN

```
#KNN wont knit but works (just takes a while to run)
#library(class)
#set.seed(1)
```

```
#knn.pred <- knn(train, test, train$diabetes, k = 10)
#table(knn.pred, test$diabetes)
```

```
#accuracy <- sum(diag(table(knn.pred, test$diabetes)))/nrow(test)
#accuracy
```

Perform CV to find best k value. . . .?

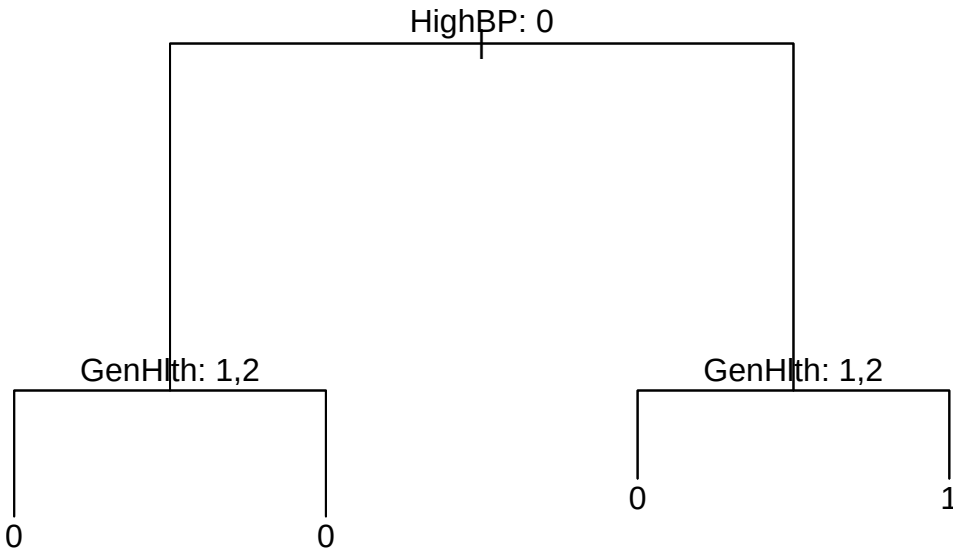
Trees

```
library(tree)
```

```
tree.all <- tree(diabetes ~ HighBP+ HighChol + CholCheck + HeartDiseaseorAttack + AnyHealthcare
+ PhysActivity + HvyAlcoholConsump + Fruits + Veggies + GenHlth + DiffWalk + Sex + Income + Education +
summary(tree.all)
```

```
##
## Classification tree:
## tree(formula = diabetes ~ HighBP + HighChol + CholCheck + HeartDiseaseorAttack +
##     AnyHealthcare + PhysActivity + HvyAlcoholConsump + Fruits +
##     Veggies + GenHlth + DiffWalk + Sex + Income + Education +
##     BMI + PhysHlth, data = factored)
## Variables actually used in tree construction:
## [1] "HighBP"  "GenHlth"
## Number of terminal nodes:  4
## Residual mean deviance:  1.144 = 80900 / 70690
## Misclassification error rate: 0.2999 = 21197 / 70692
```

```
plot(tree.all)
text(tree.all, pretty = 0)
```



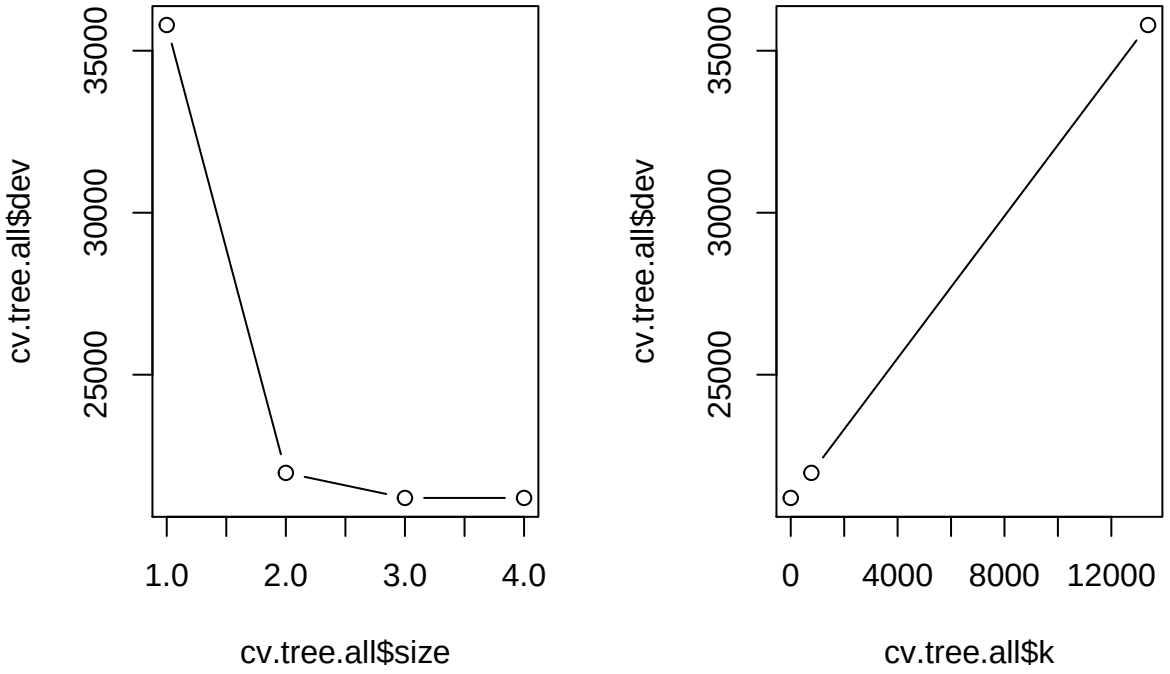

```
set.seed(3)
cv.tree.all <- cv.tree(tree.all, FUN = prune.misclass)
names(cv.tree.all)
```

```
## [1] "size"   "dev"    "k"      "method"
```

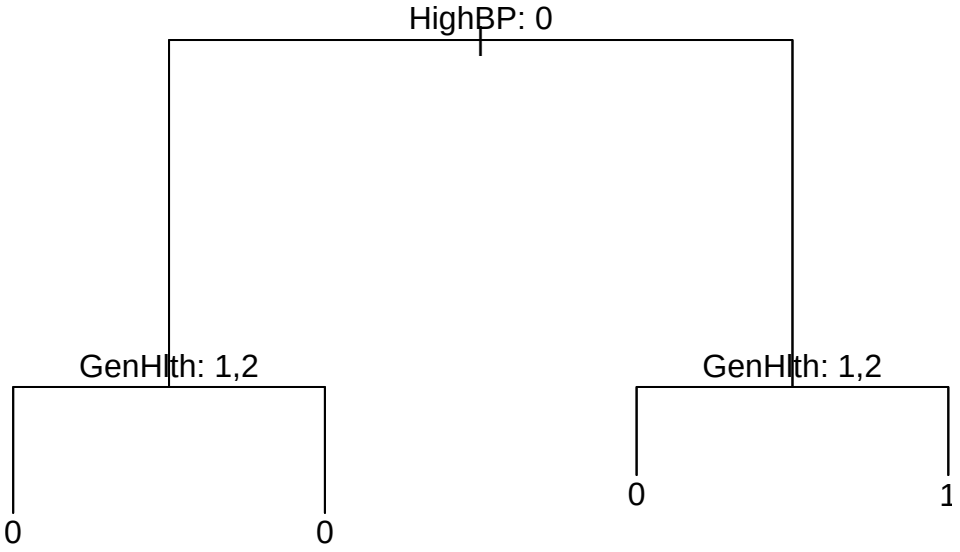
```
cv.tree.all
```

```
## $size
## [1] 4 3 2 1
##
## $dev
## [1] 21197 21197 21970 35793
##
## $k
## [1]  -Inf     0   773 13376
##
## $method
## [1] "misclass"
##
## attr(,"class")
## [1] "prune"         "tree.sequence"
```

```
par(mfrow = c(1,2))
plot(cv.tree.all$size, cv.tree.all$dev, type = "b")
plot(cv.tree.all$k, cv.tree.all$dev, type = "b")
```




```
prune.tree <- prune.misclass(tree.all, best = 4)
plot(prune.tree)
text(prune.tree, pretty = 0)
```



Neural Nets