Homework2

Brandon Cunningham

2024-10-22

Large Datset (Diabetes Data)

For our large dataset, we will use the diabetes dataset from kaggle.

This dataset has 100k clinical records of diabetes for health analytic purposes.

Link to Dataset

Goal: For this dataset, we want to predict whether or not the patient will have diabetes.

Importing:

```
diabetes_data=read.csv(url("https://raw.githubusercontent.com/sleepysloth12/DATA_622_HW01/refs/heads/ma
```

Exploratory Data Analysis

The column of interest, labeled `diabetes` is what we want to predict. It is an integer, 0 or 1, indicating if the patient has diabetes or not. In the current dataset, 91% of the patients have no diabetes and 8.5% of the patients have diabetes.

In order to build a predictive model, we must first go column by column and clean up the features a little bit to make this more accurate/applicable to healthcare data.

Data Cleaning

Year The first column is year. The dataset is timeseries data, collected from the years 2015-2022. However, each year has different numbers of observations. There is no way of knowing if this is longtitudnal data (one patient visited multiple year) due to the lack of unique patient identifier field. I think we can completely disregard and forget about this column.

```
diabetes_data = diabetes_data %>%
  select(-year)
```

Gender Next is gender. Gender is pretty even split, with ~60% being female and ~40% being male. There is an insignificant amount of people that answered "other" (less than 1%).

I'm going ahead and going to filter out other. Also, I am going to change the label to`is_female` so the choice is binary.

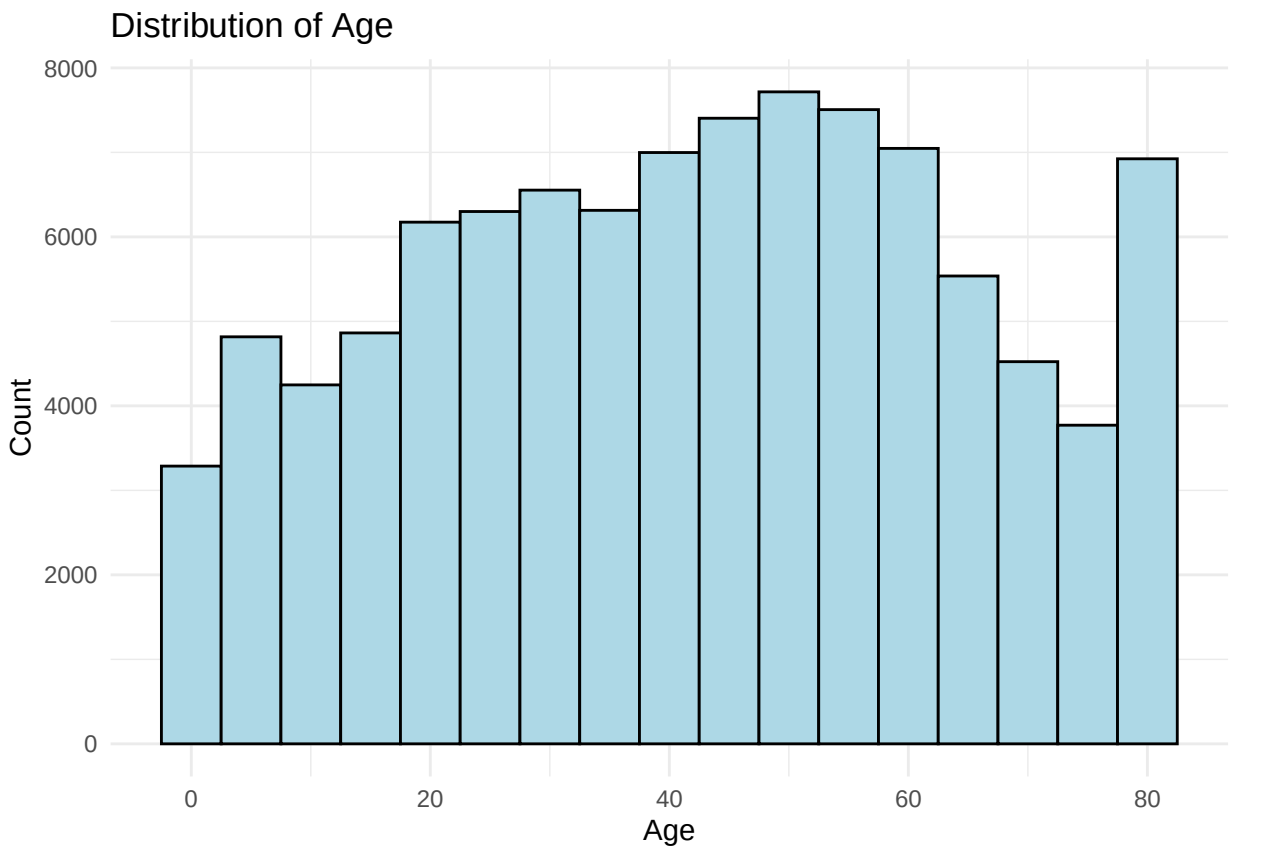
```
diabetes_data = diabetes_data %>%
  filter(gender == "Female"| gender== "Male")%>%
  mutate(is_female=ifelse(gender == "Female",1,0))%>%
  select(-gender)
```

Age Next is age. Mean age is 41.9 years old, with a standard deviation of +/-22.5 years old.

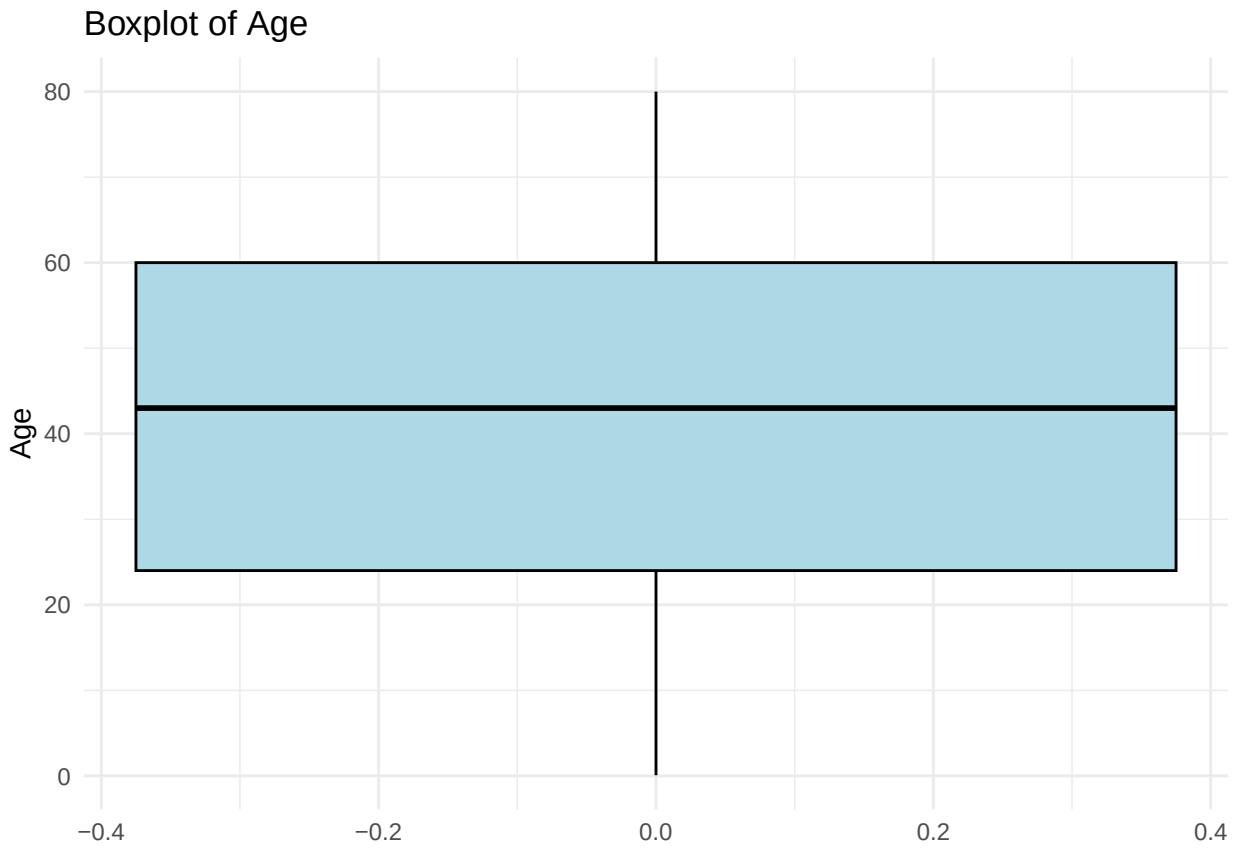
Max age is 80.

Minimum recorded age is 0.08. This might be an outlier. Therefore, lets visualize this distribution in both box plot and bar plot.

```
ggplot(diabetes_data, aes(x = age)) +
  geom_histogram(binwidth = 5, color = "black", fill = "lightblue") +
  labs(title = "Distribution of Age", x = "Age", y = "Count") +
  theme_minimal()
```




```
ggplot(diabetes_data, aes(y = age)) +
  geom_boxplot(fill = "lightblue", color = "black") +
  labs(title = "Boxplot of Age", y = "Age") +
  theme_minimal()
```



Seems like the minimum age is an outlier. In medical research, we tend to separate adult populations from pediatric populations so lets go ahead and do that here. Lets only look at 18+.

In terms of the age distribution, it looks relatively normal. Diabetes incidence seem to increase as you get closer to middle age, then decrease. There is a spike at 80 years old.

I am going to bin age/ convert it into different categories:

`is_young` = Age 18-35

`is_middle_age` = Age 36-64

`is_old` = Age 65+

```
diabetes_data = diabetes_data %>%
  filter(age>=18)%>%
  mutate(is_young=ifelse(age>=18 & age <=35, 1,0),
         is_middle_age=ifelse(age>35 & age<65 , 1 , 0),
         is_old=ifelse(age>65,1,0))%>%
  select(-age)
```

```
length(unique(diabetes_data$location))
```

State

```
## [1] 55
```

For the location column, there are 55 different locations, corresponding to the 50 different states and territory.

Location is important for diabetes prediction. Some areas are probably more likely to develop diabetes than others. Like age, I want to create categories and bin them based on the location. Then, will create dummy

variables.

```
diabetes_data = diabetes_data %>%
  mutate(

    is_new_england = if_else(location %in% c("Connecticut", "Maine", "Massachusetts",
                                             "New Hampshire", "Rhode Island", "Vermont"), 1, 0),


    is_south = if_else(location %in% c("Alabama", "Arkansas", "Delaware", "Florida", "Georgia",
                                       "Kentucky", "Louisiana", "Maryland", "Mississippi",
                                       "North Carolina", "Oklahoma", "South Carolina",
                                       "Tennessee", "Texas", "Virginia", "West Virginia"), 1, 0),


    is_midwest = if_else(location %in% c("Illinois", "Indiana", "Iowa", "Kansas", "Michigan",
                                         "Minnesota", "Missouri", "Nebraska", "North Dakota",
                                         "Ohio", "South Dakota", "Wisconsin"), 1, 0),


    is_west = if_else(location %in% c("Alaska", "Arizona", "California", "Colorado", "Hawaii",
                                      "Idaho", "Montana", "Nevada", "New Mexico", "Oregon",
                                      "Utah", "Washington", "Wyoming"), 1, 0),


    is_northeast = if_else(location %in% c("New Jersey", "New York", "Pennsylvania"), 1, 0),


    is_territories = if_else(location %in% c("Guam", "Puerto Rico", "Virgin Islands",
                                             "District of Columbia", "United States"), 1, 0)
  )%>%
  select(-location)
```

Race, Ethnicity, Hypertension, & Heart Disease Race and ethnicity is already binned and with their individual dummy variables. Race and ethnicity are both factors that influence diabetes so will leave these columns untouched.

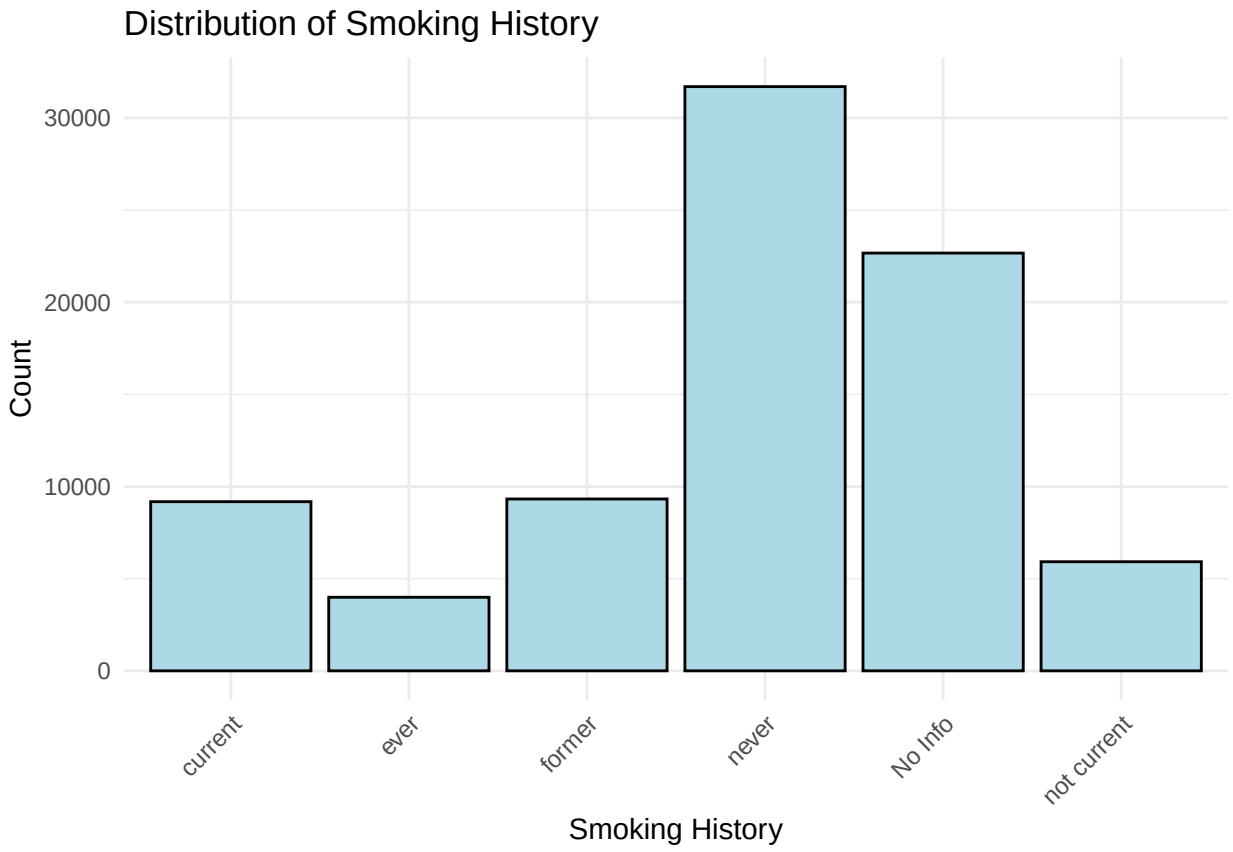
Same with the columns of hypertension and heart disease.

Smoking History There are currently 6 categories/ choices patients could respond when asked about smoking history:

```
unique(diabetes_data$smoking_history)
```

```
## [1] "never"       "not current" "current"     "No Info"     "ever"
## [6] "former"
```

```
ggplot(diabetes_data, aes(x = smoking_history)) +
  geom_bar(fill = "lightblue", color = "black") +
  labs(title = "Distribution of Smoking History", x = "Smoking History", y = "Count") +
  theme_minimal() +
  theme(axis.text.x = element_text(angle = 45, hjust = 1))
```



The biggest group is Never smoked accounting for 35% of the data.

There is a category, 'ever' which is 'Never' mislabeled. Will fix this. Once combined, never smoked will account for40% of the data.

The second biggest is 'No info' with near 35% of the data. Since the people in 'No info' may or may not be smokers, if we leave this category in it might make our predictions inaccurate. We want to capture how smoking can influence diabetes, therefore we ill remove this group.

Also, the 'not current' and 'former' group can be combined.

```
diabetes_data = diabetes_data %>%
  filter(smoking_history!="No Info")%>%
  mutate(never_smoked=ifelse(smoking_history %in% c("ever","never"),1,0),
         former_smoker=ifelse(smoking_history %in% c("former","not current"),1,0),
         current_smoker=ifelse(smoking_history=="current",1,0)) %>%
  select(-smoking_history)
```

Biomarker Columns The distribution of BMI is normal. It is numeric and continuous. We are leaving this as is.

The hbA1c_level biomarker, although numeric, has 18 unique values. In healthcare, this biomarker is usually used to determine diabetes. We will bin this biomarker for the following categories:

A1c $<$ 5.7% –> Normal A1C

A1c between 5.7-6.4 % –> PreDiabetes

A1C over 6.5% –> diabetes

Although, correlation analysis is needed. There might be multicollinearity between these biomarker variables.

I say this because blood glucose variable and A1c directly related to each other.

Actually going to remove blood glucose because having that and A1C is repetitive/ multicollinearity.

```
diabetes_data = diabetes_data %>%
  mutate(normal_a1c=ifelse(hbA1c_level<5.7,1,0),
         prediabetic_a1c=ifelse(hbA1c_level>=5.7 & hbA1c_level <= 6.4,1,0),
         diabetic_a1c=ifelse(hbA1c_level>6.4,1,0))%>%
  select(-c(hbA1c_level,blood_glucose_level))
```

Model Selection

Now that our dataset is clean, we can discuss what model we want to use.

The target variable to predict is diabetes (binary choice whether or not patient will have diabetes).

I think the best algorithm to use in this case is logistic regression. Logistic regression provides interpretable results. The coefficients in the model can be easily interpreted as the change in log-odds of having diabetes for a one-unit change in the predictor, holding other variables constant. This interpretability is important in healthcare.

Correlation Matrix

Before begining the logistic regression model, I want to run a correlation matrix to look for multicolinearity

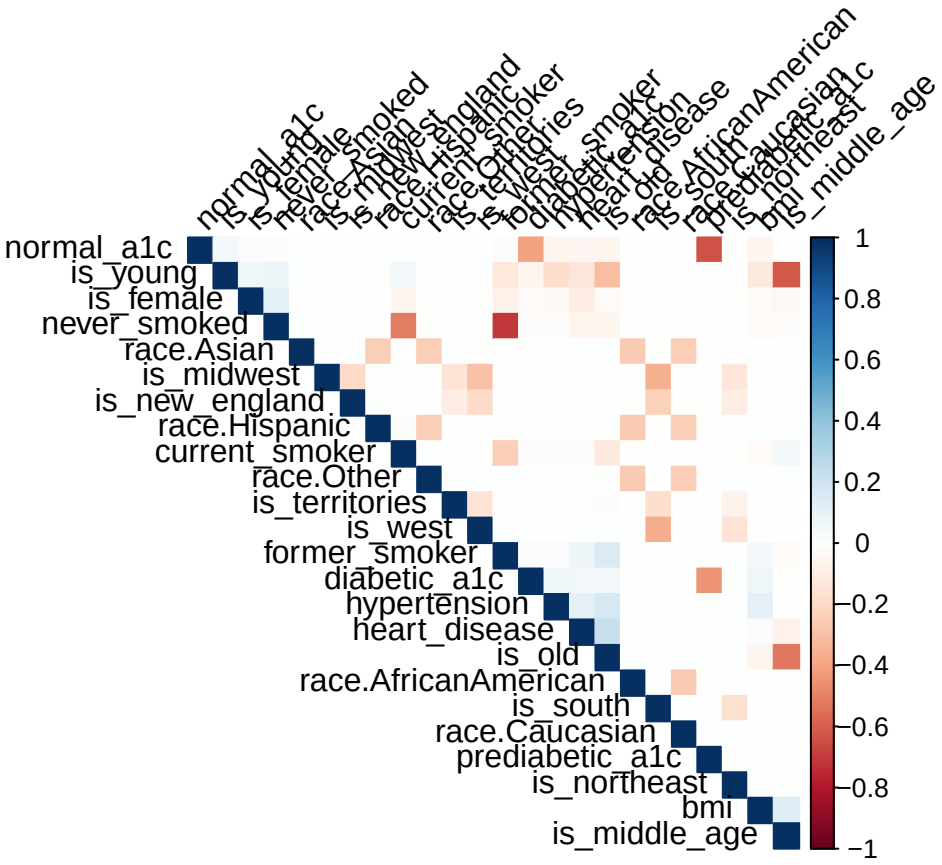
```
predictors <- diabetes_data %>%
  select(-diabetes)

cor_matrix <- cor(predictors)

high_cor <- findCorrelation(cor_matrix, cutoff = 0.7, verbose = TRUE)
```

```
## Compare row 19  and column  20 with corr  0.705
##   Means:  0.07 vs 0.049 so flagging column 19
## All correlations <= 0.7
```

```
cat("Highly correlated variables:", paste(names(predictors)[high_cor], collapse = ", "), "\n")
```

```
## Highly correlated variables: never_smoked
```

```
corrplot(cor_matrix, method = "color", type = "upper", order = "hclust",
         tl.col = "black", tl.srt = 45)
```



There is some multicollinearity

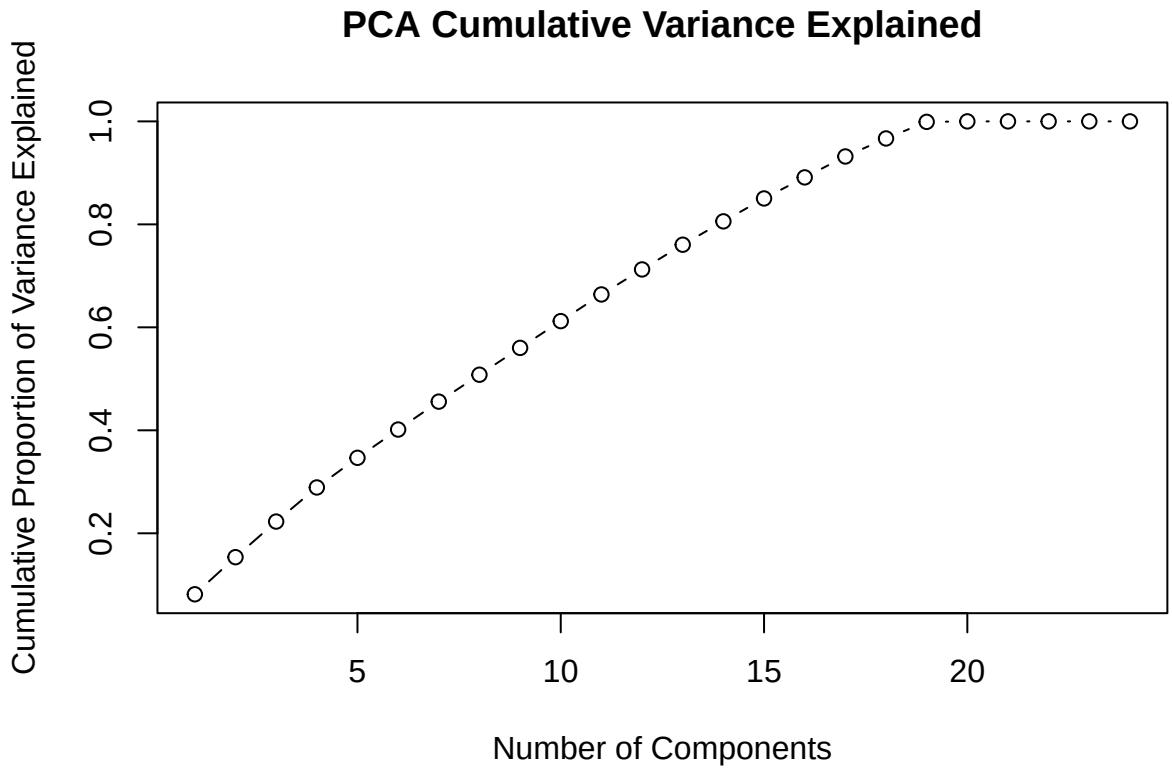
Principal Component Analysis

Conducting a PCA to determine the important components

```
pca_result <- prcomp(predictors, scale. = TRUE)
summary(pca_result)
```

```
## Importance of components:
##                           PC1    PC2     PC3     PC4     PC5    PC6     PC7
## Standard deviation     1.39898 1.3155 1.28787 1.26107 1.17547 1.1478 1.14083
## Proportion of Variance 0.08155 0.0721 0.06911 0.06626 0.05757 0.0549 0.05423
## Cumulative Proportion  0.08155 0.1537 0.22276 0.28902 0.34659 0.4015 0.45572
##                           PC8     PC9    PC10    PC11    PC12    PC13   PC14
## Standard deviation     1.11981 1.11834 1.11722 1.11510 1.07939 1.07351 1.0438
## Proportion of Variance 0.05225 0.05211 0.05201 0.05181 0.04855 0.04802 0.0454
## Cumulative Proportion  0.50797 0.56008 0.61209 0.66390 0.71244 0.76046 0.8059
##                          PC15    PC16    PC17    PC18    PC19    PC20
## Standard deviation     1.03300 0.99101 0.98754 0.91645 0.87923 0.14869
## Proportion of Variance 0.04446 0.04092 0.04063 0.03499 0.03221 0.00092
## Cumulative Proportion  0.85032 0.89124 0.93187 0.96687 0.99908 1.00000
##                            PC21      PC22      PC23      PC24
## Standard deviation     2.199e-13 2.56e-14 2.422e-14 1.844e-14
## Proportion of Variance 0.000e+00 0.00e+00 0.000e+00 0.000e+00
## Cumulative Proportion  1.000e+00 1.00e+00 1.000e+00 1.000e+00
```

```r
plot(cumsum(pca_result$sdev^2 / sum(pca_result$sdev^2)),
     type = "b",
     xlab = "Number of Components",
     ylab = "Cumulative Proportion of Variance Explained",
     main = "PCA Cumulative Variance Explained")
```



Our first principal component only accounts for about 8.16% of the total variance. That's not a lot. It means no single factor dominates in predicting diabetes. This makes sense given the complex nature of the disease and the variety of factors we've included in our dataset.

We need 11 components to explain about 66% of the variance, and it takes 19 to get to nearly 100%. Looking at our cumulative variance plot, we can see this gradual climb. The fact that we need so many components to explain most of the variance suggests we shouldn't try to oversimplify our model. Most of our variables are contributing unique information about diabetes risk.

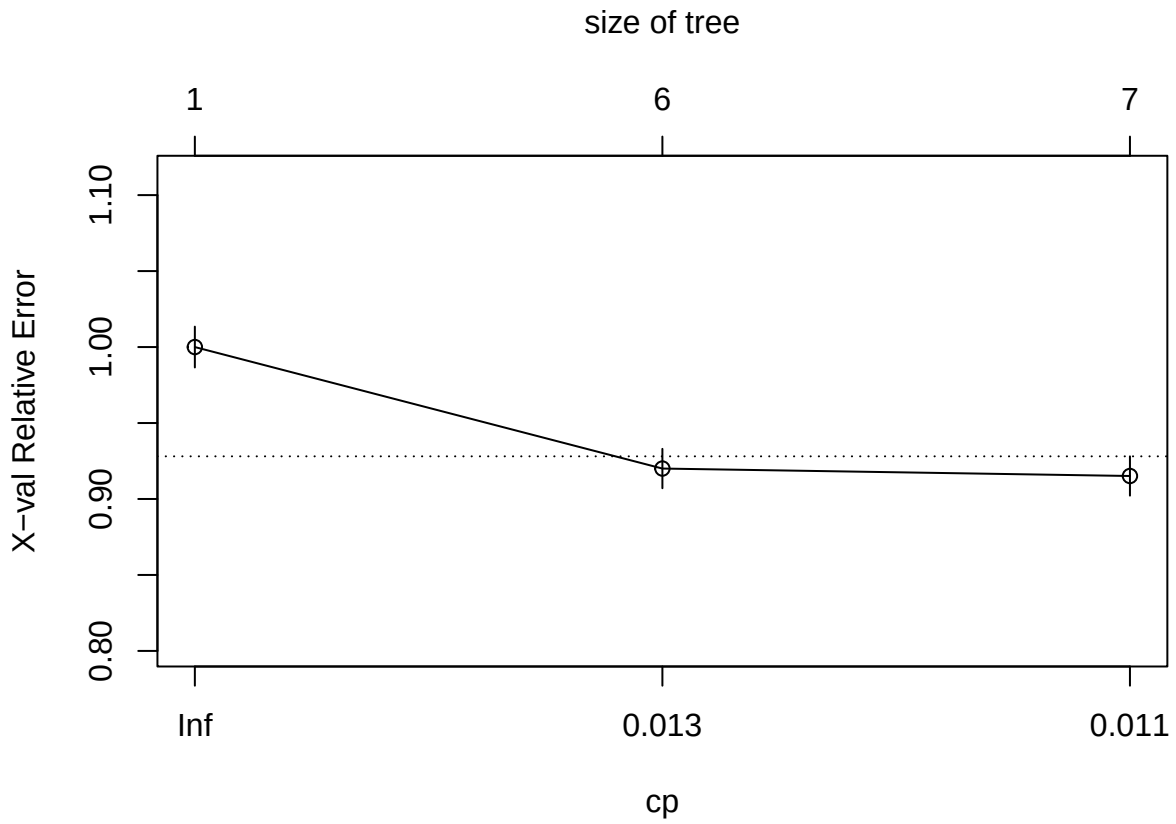
While we don't see extreme multicollinearity, there is some correlation among our variables. We can explain about 85% of the variance with 15 components, which is fewer than our original variables.

```r
set.seed(622)

train_split_idx=createDataPartition(diabetes_data$diabetes, p=0.7, list=FALSE)

train_diab = diabetes_data[train_split_idx,]
test_diab = diabetes_data[-train_split_idx,]

control <- trainControl(method = "cv", number = 5)
metric <- "RMSE"
```

Decision Tree 1:

For this tree we are throwing everything into the decision tree and

```r
set.seed(622)
fit_tree <- rpart(diabetes ~ ., method = 'class', data = train_diab)

plotcp(fit_tree)
```




```r
printcp(fit_tree)
```

```
##
## Classification tree:
## rpart(formula = diabetes ~ ., data = train_diab, method = "class")
##
## Variables actually used in tree construction:
## [1] bmi           diabetic_a1c  heart_disease hypertension  is_middle_age
## [6] is_young
##
## Root node error: 4890/42073 = 0.11623
##
## n= 42073
##
##         CP nsplit rel error  xerror     xstd
## 1 0.014656      0   1.00000 1.00000 0.013444
## 2 0.011043      5   0.92025 0.92004 0.012963
## 3 0.010000      6   0.90920 0.91513 0.012932
```

```r
summary(fit_tree)
```

```
## Call:
## rpart(formula = diabetes ~ ., data = train_diab, method = "class")
##   n= 42073
```

```
##
##           CP nsplit rel error    xerror       xstd
## 1 0.01465576      0 1.0000000 1.0000000 0.01344361
## 2 0.01104294      5 0.9202454 0.9200409 0.01296257
## 3 0.01000000      6 0.9092025 0.9151329 0.01293208
##
## Variable importance
##  diabetic_a1c      is_young           bmi is_middle_age        is_old
##            61            20            11             2             2
##  hypertension heart_disease
##             2             1
##
## Node number 1: 42073 observations,    complexity param=0.01465576
##   predicted class=0  expected loss=0.1162266  P(node) =1
##     class counts: 37183  4890
##    probabilities: 0.884 0.116
##   left son=2 (32794 obs) right son=3 (9279 obs)
##   Primary splits:
##       diabetic_a1c < 0.5    to the left,  improve=990.4332, (0 missing)
##       normal_a1c   < 0.5    to the right, improve=656.6893, (0 missing)
##       hypertension < 0.5    to the left,  improve=302.2151, (0 missing)
##       is_young     < 0.5    to the right, improve=291.6968, (0 missing)
##       is_old       < 0.5    to the left,  improve=290.4417, (0 missing)
##   Surrogate splits:
##       bmi < 70.255 to the left,  agree=0.78, adj=0.001, (0 split)
##
## Node number 2: 32794 observations
##   predicted class=0  expected loss=0.0585168  P(node) =0.7794548
##     class counts: 30875  1919
##    probabilities: 0.941 0.059
##
## Node number 3: 9279 observations,    complexity param=0.01465576
##   predicted class=0  expected loss=0.3201854  P(node) =0.2205452
##     class counts:  6308  2971
##    probabilities: 0.680 0.320
##   left son=6 (2019 obs) right son=7 (7260 obs)
##   Primary splits:
##       is_young      < 0.5    to the right, improve=332.4822, (0 missing)
##       is_old        < 0.5    to the left,  improve=248.8659, (0 missing)
##       bmi           < 30.595 to the left,  improve=243.0985, (0 missing)
##       hypertension  < 0.5    to the left,  improve=222.8035, (0 missing)
##       heart_disease < 0.5    to the left,  improve=169.4133, (0 missing)
##
## Node number 6: 2019 observations
##   predicted class=0  expected loss=0.06636949  P(node) =0.04798802
##     class counts:  1885   134
##    probabilities: 0.934 0.066
##
## Node number 7: 7260 observations,    complexity param=0.01465576
##   predicted class=0  expected loss=0.3907713  P(node) =0.1725572
##     class counts:  4423  2837
##    probabilities: 0.609 0.391
##   left son=14 (4635 obs) right son=15 (2625 obs)
##   Primary splits:
```

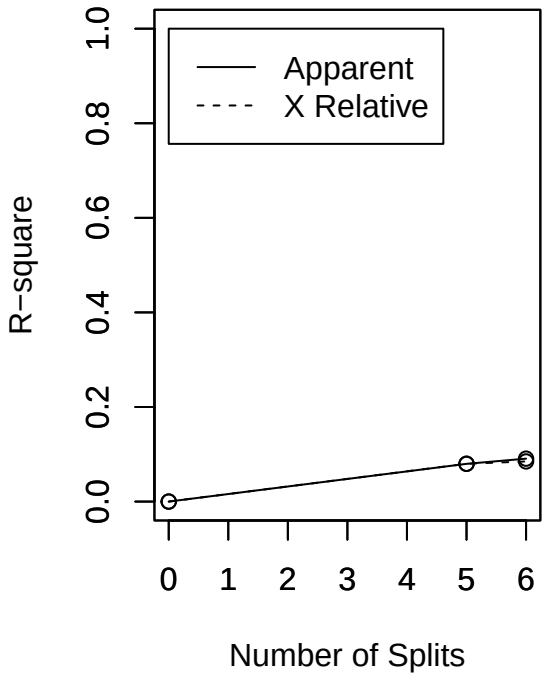
```
##       bmi           < 30.585 to the left,  improve=185.4711, (0 missing)
##       hypertension  < 0.5    to the left,  improve=132.9508, (0 missing)
##       is_middle_age < 0.5    to the right, improve=126.4846, (0 missing)
##       is_old        < 0.5    to the left,  improve=114.9070, (0 missing)
##       heart_disease < 0.5    to the left,  improve=108.2956, (0 missing)
##
## Node number 14: 4635 observations
##   predicted class=0  expected loss=0.3057174  P(node) =0.1101657
##     class counts:  3218  1417
##    probabilities: 0.694 0.306
##
## Node number 15: 2625 observations,    complexity param=0.01465576
##   predicted class=1  expected loss=0.4590476  P(node) =0.06239156
##     class counts:  1205  1420
##    probabilities: 0.459 0.541
##   left son=30 (1868 obs) right son=31 (757 obs)
##   Primary splits:
##       is_middle_age < 0.5    to the right, improve=39.77034, (0 missing)
##       hypertension  < 0.5    to the left,  improve=33.33725, (0 missing)
##       heart_disease < 0.5    to the left,  improve=33.00677, (0 missing)
##       is_old        < 0.5    to the left,  improve=32.55981, (0 missing)
##       bmi           < 37.695 to the left,  improve=25.26904, (0 missing)
##   Surrogate splits:
##       is_old        < 0.5    to the left,  agree=0.971, adj=0.900, (0 split)
##       heart_disease < 0.5    to the left,  agree=0.712, adj=0.003, (0 split)
##
## Node number 30: 1868 observations,    complexity param=0.01465576
##   predicted class=0  expected loss=0.485546  P(node) =0.04439902
##     class counts:   961   907
##    probabilities: 0.514 0.486
##   left son=60 (1475 obs) right son=61 (393 obs)
##   Primary splits:
##       hypertension  < 0.5    to the left,  improve=28.228070, (0 missing)
##       heart_disease < 0.5    to the left,  improve=25.989720, (0 missing)
##       bmi           < 37.695 to the left,  improve=17.918840, (0 missing)
##       is_female     < 0.5    to the right, improve= 5.547072, (0 missing)
##       is_midwest    < 0.5    to the left,  improve= 2.477221, (0 missing)
##
## Node number 31: 757 observations
##   predicted class=1  expected loss=0.322325  P(node) =0.01799254
##     class counts:   244   513
##    probabilities: 0.322 0.678
##
## Node number 60: 1475 observations,    complexity param=0.01104294
##   predicted class=0  expected loss=0.440678  P(node) =0.03505811
##     class counts:   825   650
##    probabilities: 0.559 0.441
##   left son=120 (1377 obs) right son=121 (98 obs)
##   Primary splits:
##       heart_disease  < 0.5    to the left,  improve=23.537950, (0 missing)
##       bmi            < 37.365 to the left,  improve=16.420940, (0 missing)
##       is_female      < 0.5    to the right, improve= 3.990932, (0 missing)
##       is_midwest     < 0.5    to the left,  improve= 1.784707, (0 missing)
##       is_territories < 0.5    to the left,  improve= 1.739569, (0 missing)
```

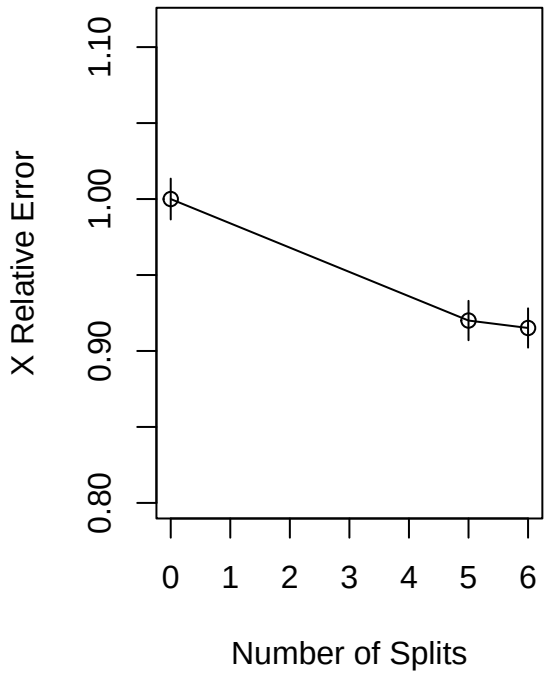
```
##
## Node number 61: 393 observations
##   predicted class=1  expected loss=0.346056  P(node) =0.009340907
##     class counts:   136   257
##    probabilities: 0.346 0.654
##
## Node number 120: 1377 observations
##   predicted class=0  expected loss=0.4168482  P(node) =0.03272883
##     class counts:   803   574
##    probabilities: 0.583 0.417
##
## Node number 121: 98 observations
##   predicted class=1  expected loss=0.2244898  P(node) =0.002329285
##     class counts:    22    76
##    probabilities: 0.224 0.776
```

```
par(mfrow = c(1, 2))
rsq.rpart(fit_tree)
```

```
##
## Classification tree:
## rpart(formula = diabetes ~ ., data = train_diab, method = "class")
##
## Variables actually used in tree construction:
## [1] bmi           diabetic_a1c  heart_disease hypertension  is_middle_age
## [6] is_young
##
## Root node error: 4890/42073 = 0.11623
##
## n= 42073
##
##         CP nsplit rel error  xerror     xstd
## 1 0.014656      0   1.00000 1.00000 0.013444
## 2 0.011043      5   0.92025 0.92004 0.012963
## 3 0.010000      6   0.90920 0.91513 0.012932
```

```
## Warning in rsq.rpart(fit_tree): may not be applicable for this method
```






```
predictions_tree <- predict(fit_tree, newdata = test_diab)
head(predictions_tree)
```

```
##            0         1
## 1  0.9414832 0.0585168
## 2  0.9414832 0.0585168
## 9  0.5831518 0.4168482
## 11 0.5831518 0.4168482
## 12 0.9414832 0.0585168
## 14 0.9414832 0.0585168
```

```
tree_rmse <- RMSE(predictions_tree, test_diab$diabetes)
print(tree_rmse)
```

```
## [1] 0.645769
```