Project

2023-08-31

Read in data

These first few exercises will run through some of the simple principles of creating a ggplot2 object, assigning aesthetics mappings and geoms.

1. Read in the healthcare-dataset-stroke-data ,into a new object called 'stroke' with function read.csv

```r
# Import Libraries
library(tidyverse)
library(dplyr)
library(modelr)
library(tidyr)
library(pROC)
library(MASS)
```

```r
stroke = read_csv("healthcare-dataset-stroke-data.csv")
stroke %<>% mutate(bmi=as.numeric(.$bmi)) %>% drop_na()
```

Splitting the data

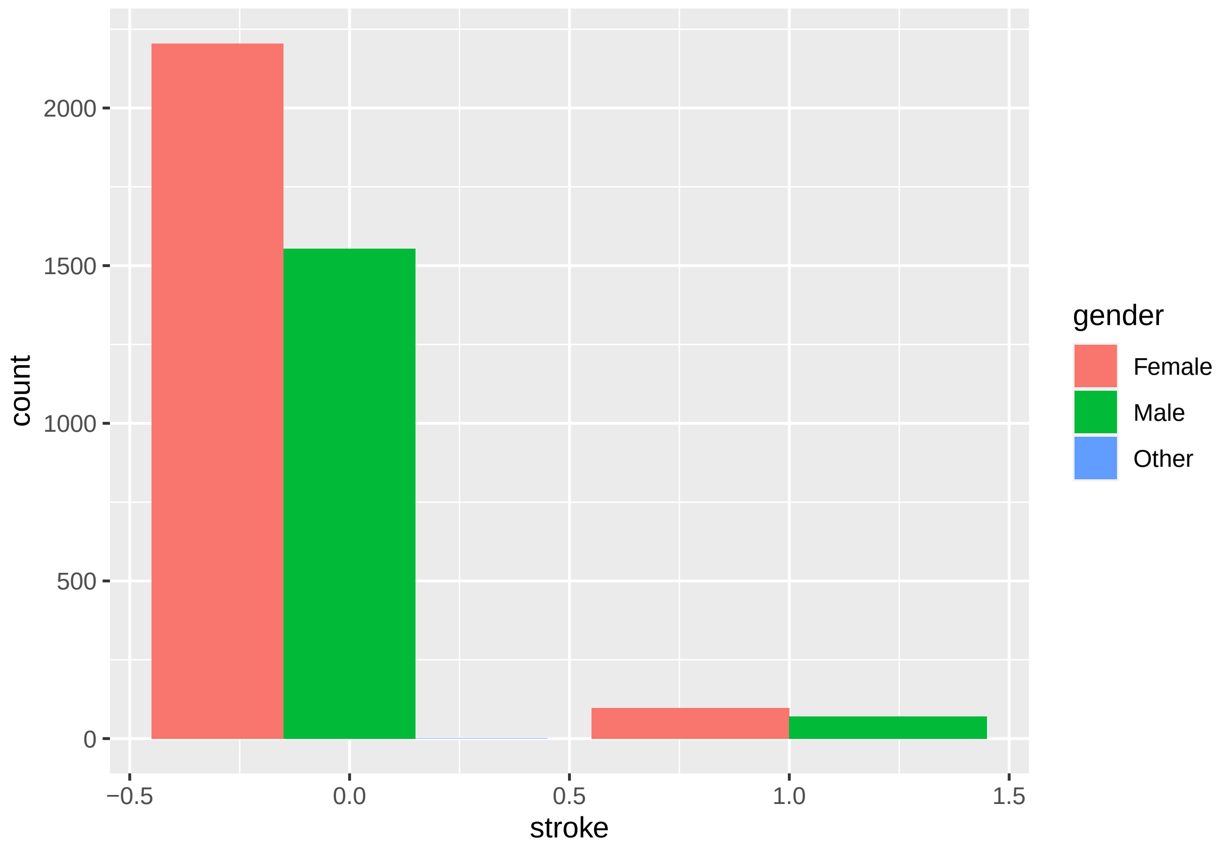
2. Split your data into training dataset and testing dataset using the ratio 80:20

```r
split <- resample_partition(stroke, c(train=0.8, test=0.2))
stroke_train <- as_tibble(split$train)
stroke_test <- as_tibble(split$test)
```

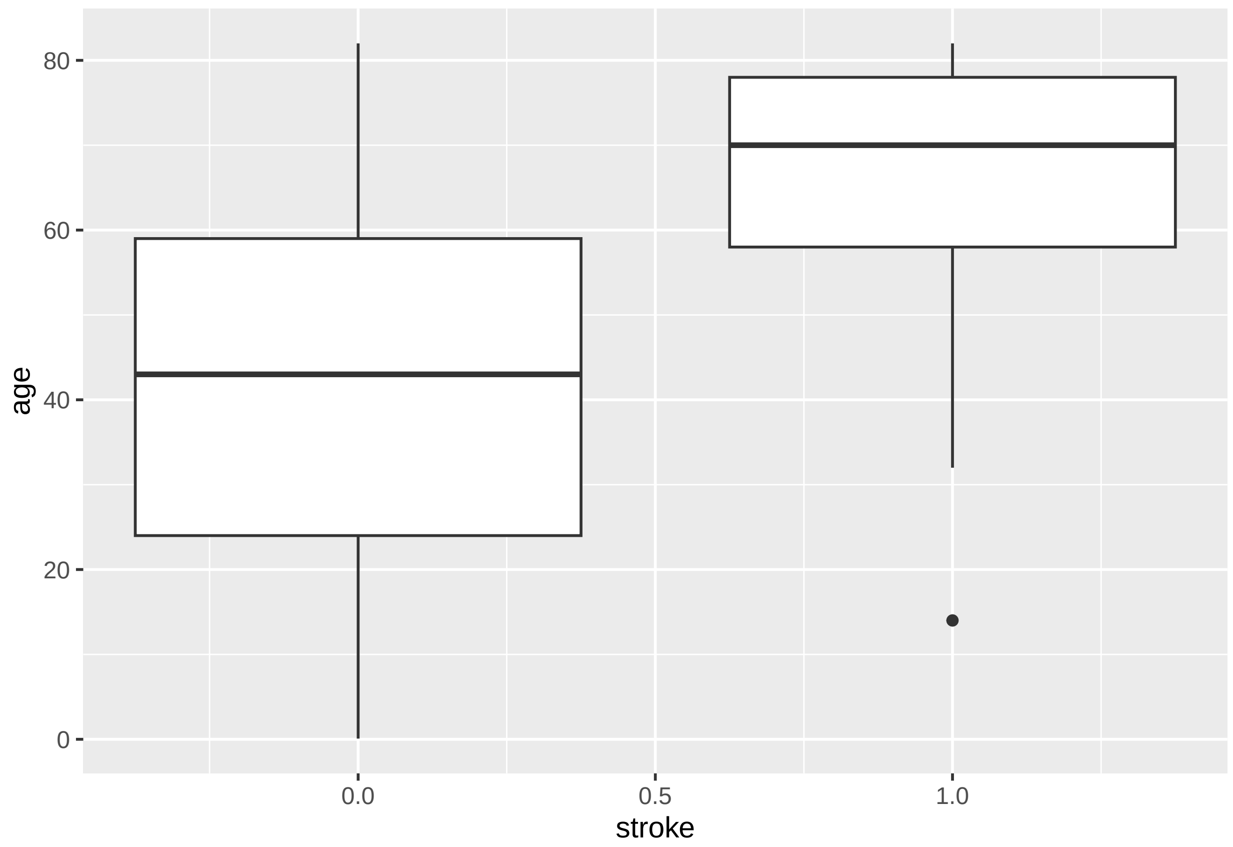
Plotting all other variables using the training dataset

3. Plot stroke v.s. all other variables (except id), which means you would generate 10 plots totally (use the golden rules)

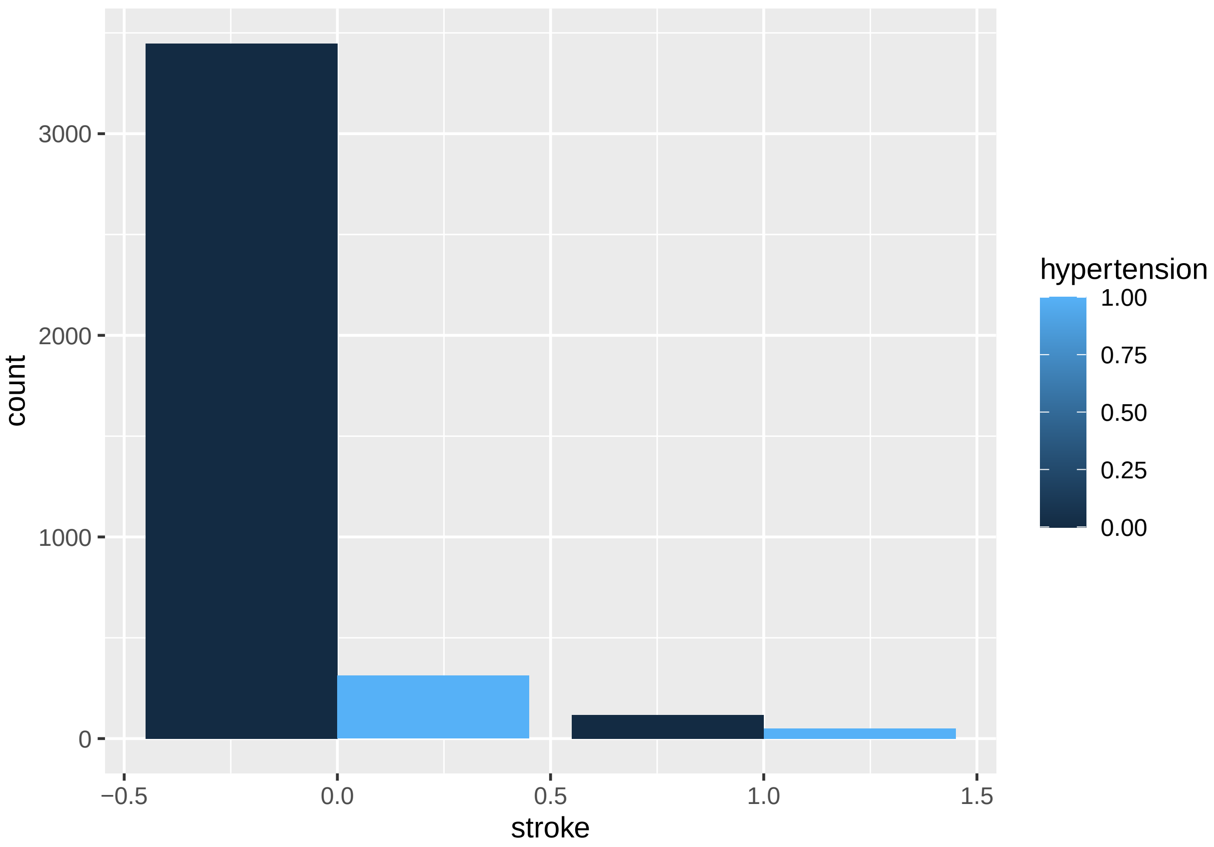
```r
# 1. Plot stroke vs. gender
ggplot(stroke_train, mapping=aes(x=stroke, fill=gender, group=gender)) + geom_bar(position="dodge")
```



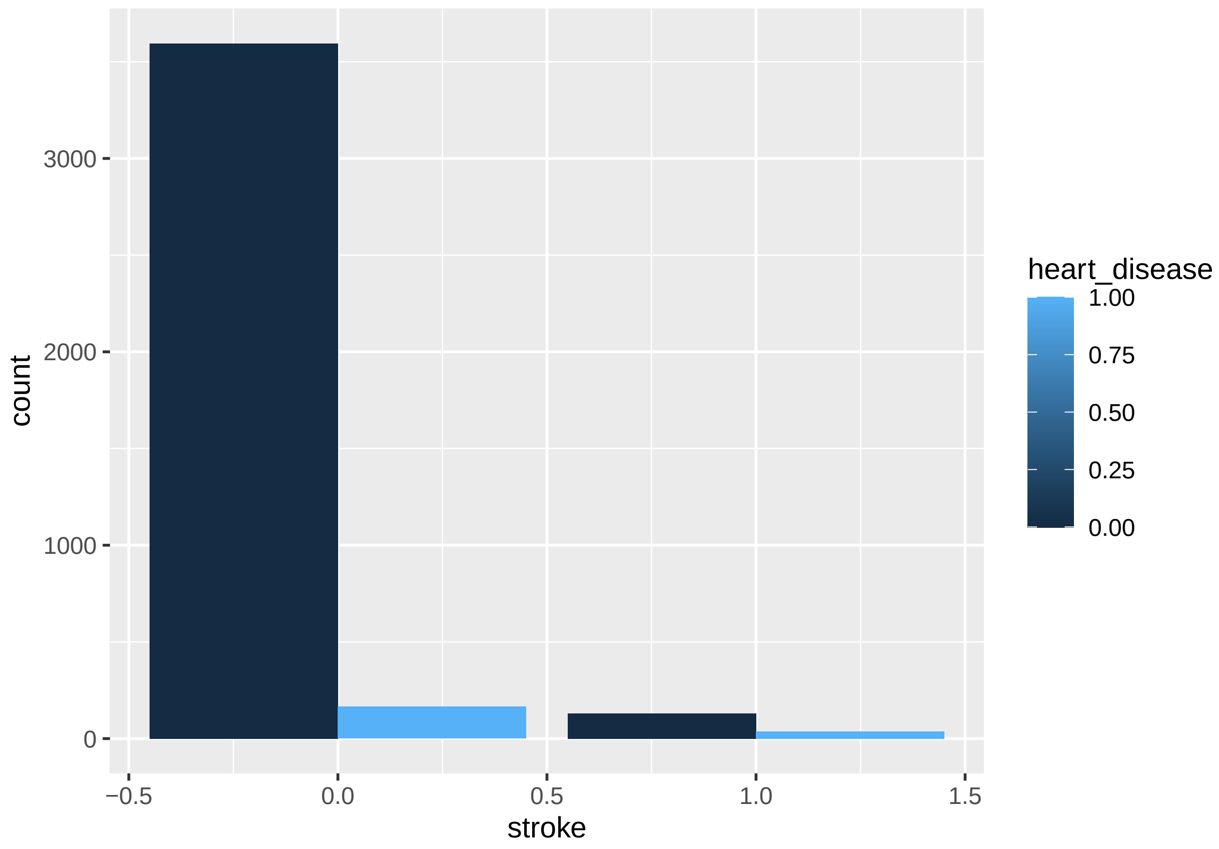
```
# 2. Plot stroke vs. age
ggplot(stroke_train, mapping=aes(x=stroke, y=age, group=stroke)) + geom_boxplot()
```



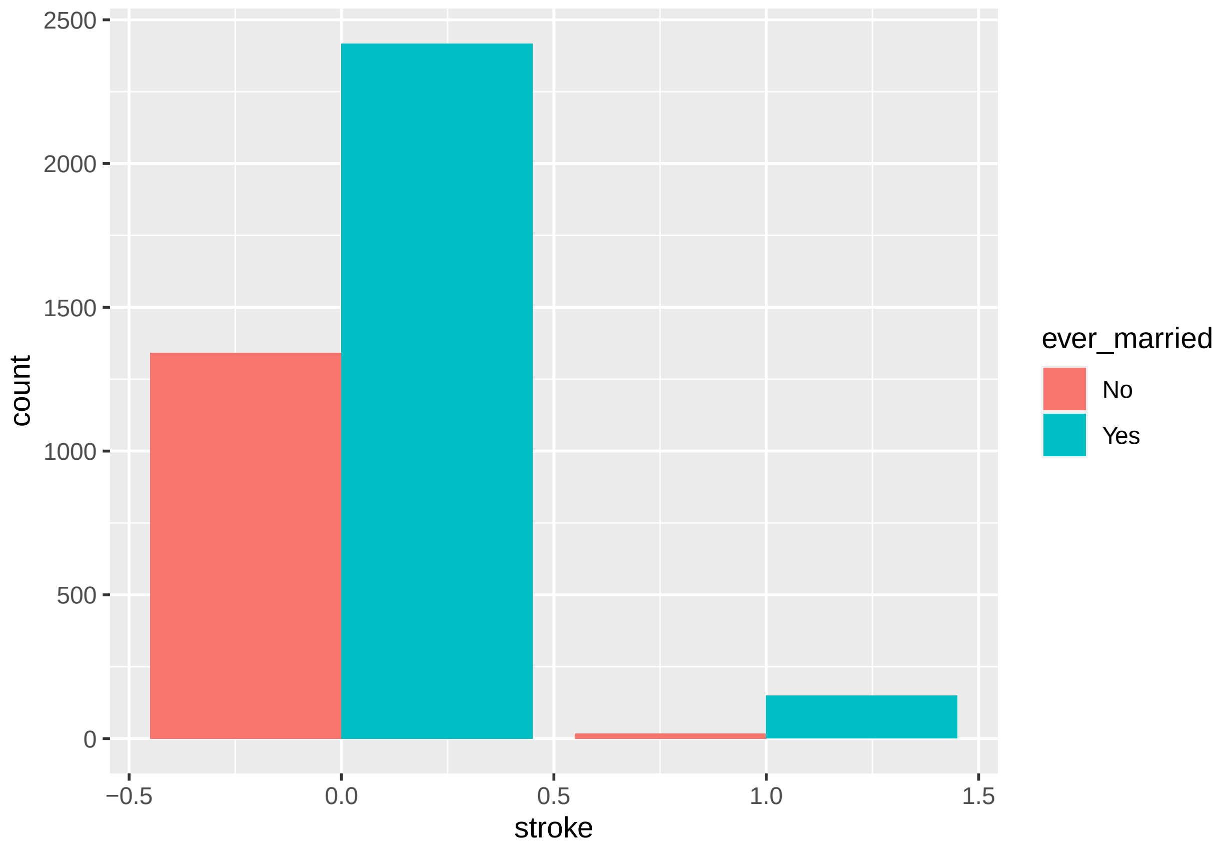
```
# 3. Plot stroke vs. hypertension
ggplot(stroke_train, mapping=aes(x=stroke, fill=hypertension, group=hypertension)) + geom_bar(position=
```



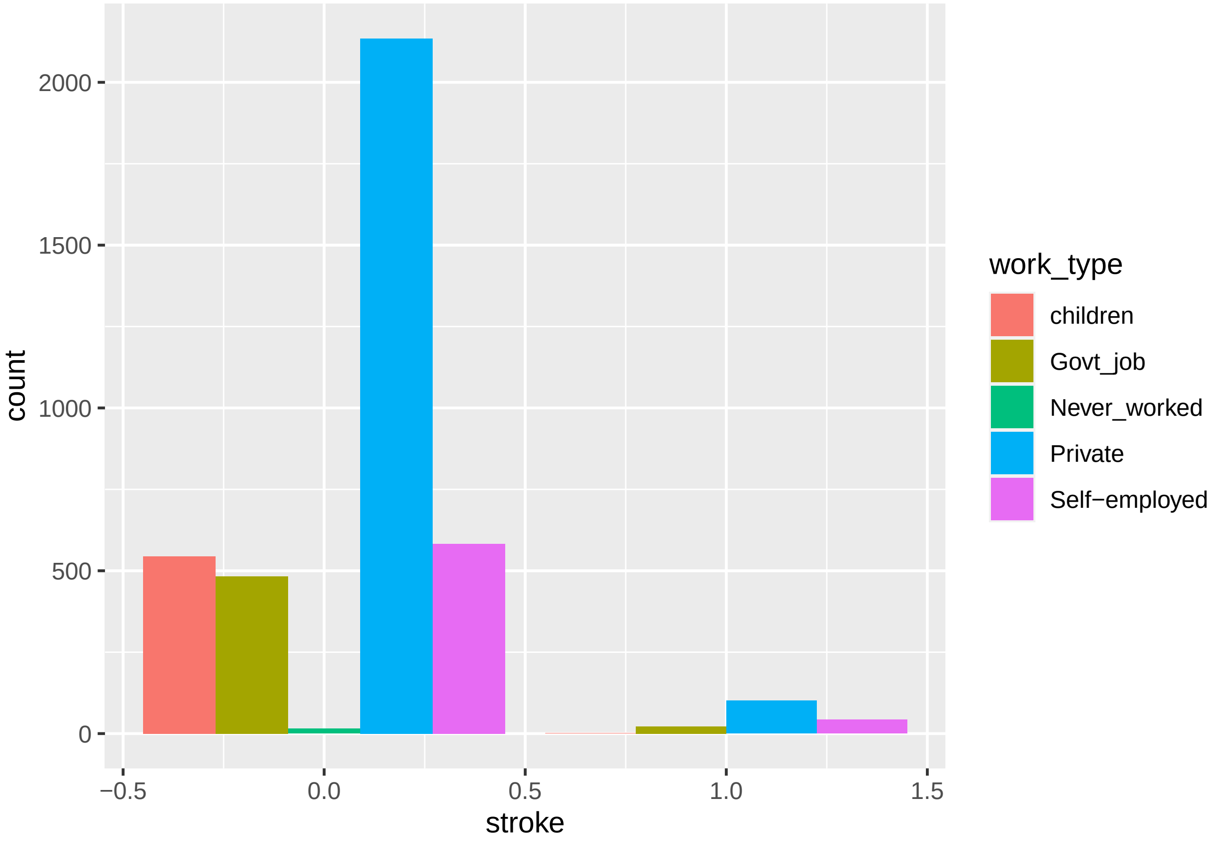
```
# 4. Plot stroke vs. heart disease
ggplot(stroke_train, mapping=aes(x=stroke, fill=heart_disease, group=heart_disease)) + geom_bar(positio
```



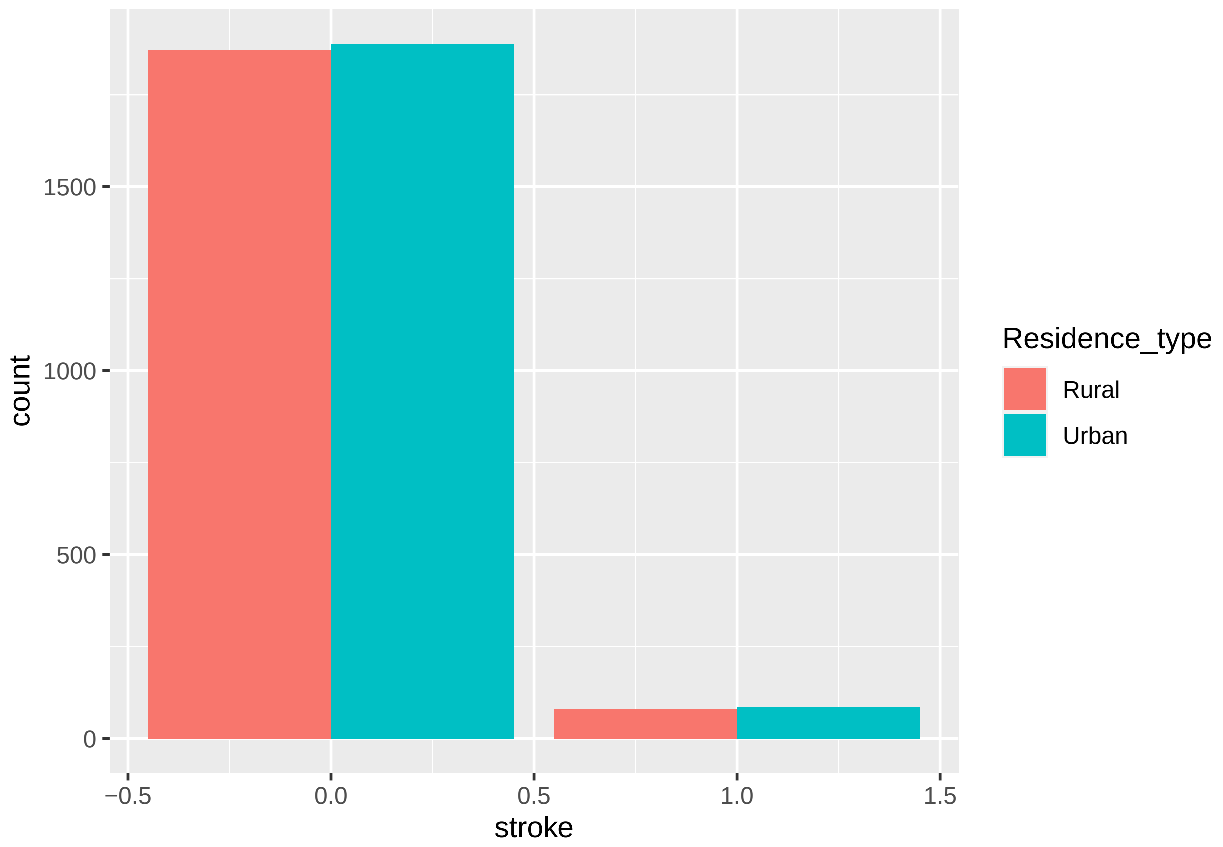
```
# 5. Plot stroke vs. ever married
ggplot(stroke_train, mapping=aes(x=stroke, fill=ever_married, group=ever_married)) + geom_bar(position=
```



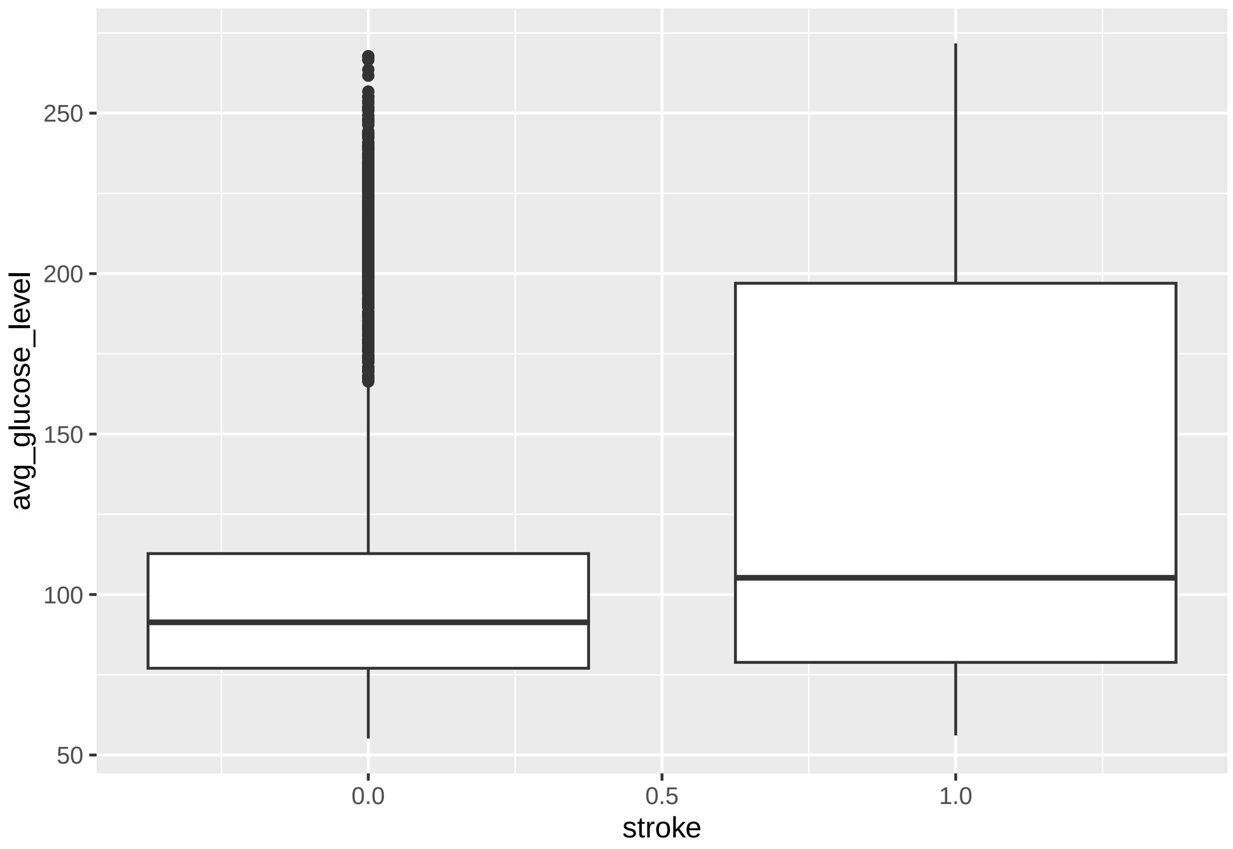
```
# 6. Plot stroke vs. work type
ggplot(stroke_train, mapping=aes(x=stroke, fill=work_type, group=work_type)) + geom_bar(position="dodge
```



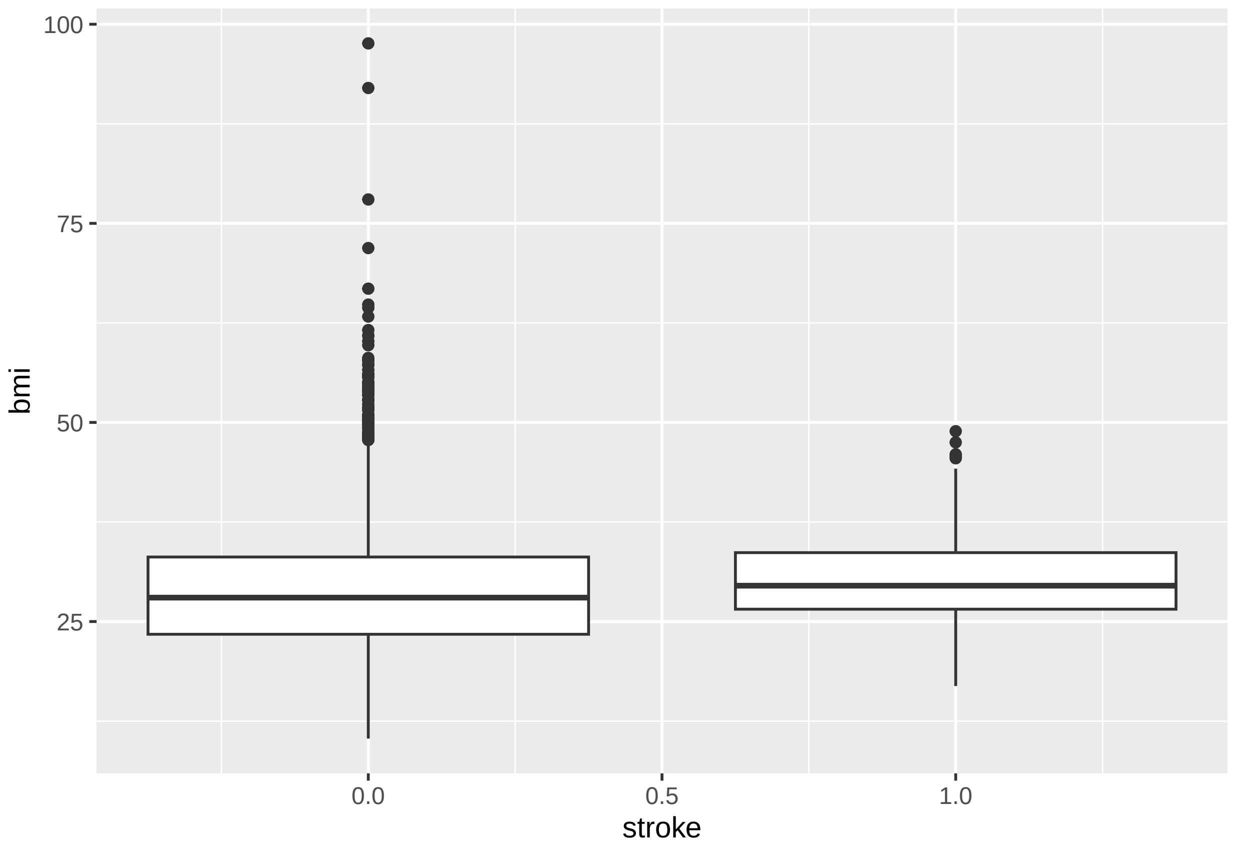
```
# 7. Plot stroke vs. residence type
ggplot(stroke_train, mapping=aes(x=stroke, fill=Residence_type, group=Residence_type)) + geom_bar(posit
```



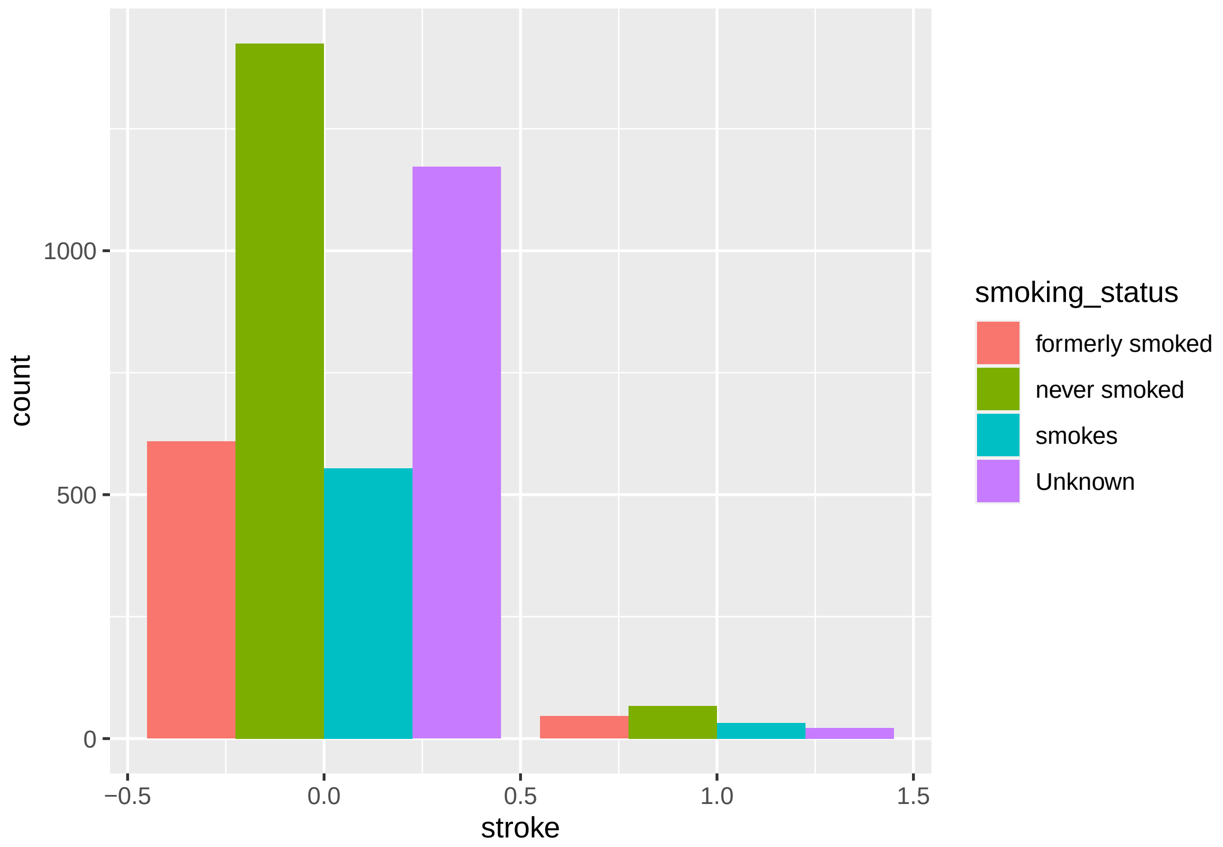
```
# 8. Plot stroke vs. average glucose level
ggplot(stroke_train, mapping=aes(x=stroke, y=avg_glucose_level, group=stroke)) + geom_boxplot()
```



```
# 9. Plot stroke vs. bmi
ggplot(stroke_train, mapping=aes(x=stroke, y=bmi, group=stroke)) + geom_boxplot()
```



```
# 10. Plot stroke vs. Smoking status
ggplot(stroke_train, mapping=aes(x=stroke, fill=smoking_status, group=smoking_status)) + geom_bar(positi
```



Manually select variables and train the models

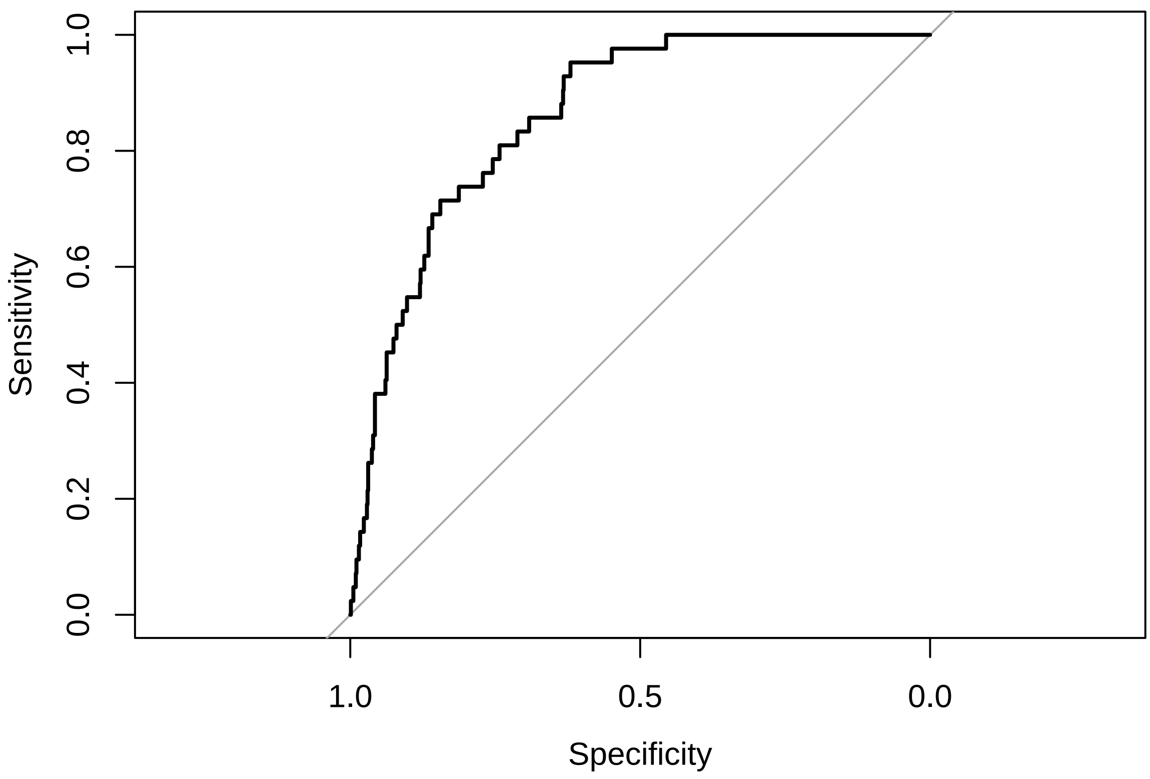
4. Pick some variables as your independent variables (it could be from 2 ~ 10), and explain why you need to pick them. Then train a logistic regression using this variables, report the significant variables and the ROC on testing set.

- ***Variables of Choice with Reasons: gender, age, hypertension, ever_married, work_type, avg_glucose_level*****, smoking_status**
- **gender: In the bar plot, there is an obvious difference of stroke count with different genders**
- **age: In the box plot, the mean of age of stroke status are apparently different**
- **hypertension: In the bar plot, there is an obvious difference of stroke count with hypertension/No hypertension. Since hypertension and heart disease bar plot looks very similar. Only "hypertension" variable is used instead.**
- **ever_married: In the bar plot, there is a significant Difference of stroke count with different married status.**
- **work_type: In the bar plot, there is a significant Difference of stroke count with different work types.**
- **avg_glucose_level: In the box plot, although the mean of average glucose level are similar, the interquartile range of sample who DO have stroke is much larger than the sample who DO NOT have stroke. Despite the box plot of bmi is similar to this plot, the interquartile range of both bmi status in the bmi plot are the same. Therefore, only the variable "avg_glucose_level" is chosen.**
- **smoking_status: In the box plot, just like the variable "work_type", it has significant difference for count with statuses**

```
# Model Training
fit1 <- glm(stroke ~ gender + age + hypertension + ever_married + work_type + avg_glucose_level + smokin
           data=stroke_train,
           family="binomial")

# Investigate Variables of Significance
summary(fit1)
```

```
##
## Call:
## glm(formula = stroke ~ gender + age + hypertension + ever_married +
##     work_type + avg_glucose_level + smoking_status, family = "binomial",
##     data = stroke_train)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.0593  -0.2959  -0.1536  -0.0771   3.4732
##
## Coefficients:
##                              Estimate Std. Error z value Pr(>|z|)
## (Intercept)                -6.875e+00  1.047e+00  -6.565 5.22e-11 ***
## genderMale                 -3.867e-02  1.722e-01  -0.225  0.82236
## genderOther                -1.047e+01  1.455e+03  -0.007  0.99426
## age                         7.431e-02  6.775e-03  10.967  < 2e-16 ***
## hypertension                5.817e-01  1.927e-01   3.019  0.00254 **
## ever_marriedYes            -1.350e-01  2.811e-01  -0.480  0.63114
## work_typeGovt_job          -9.256e-01  1.124e+00  -0.824  0.41014
## work_typeNever_worked      -1.001e+01  3.746e+02  -0.027  0.97868
## work_typePrivate           -7.591e-01  1.107e+00  -0.686  0.49302
## work_typeSelf-employed     -1.165e+00  1.131e+00  -1.030  0.30303
## avg_glucose_level           4.213e-03  1.401e-03   3.007  0.00264 **
## smoking_statusnever smoked -1.417e-01  2.100e-01  -0.675  0.49993
## smoking_statussmokes        3.262e-01  2.534e-01   1.287  0.19797
## smoking_statusUnknown      -4.383e-01  2.800e-01  -1.566  0.11744
## ---
## Signif. codes:  0 ’***’ 0.001 ’**’ 0.01 ’*’ 0.05 ’.’ 0.1 ’ ’ 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 1381.4  on 3926  degrees of freedom
## Residual deviance: 1096.3  on 3913  degrees of freedom
## AIC: 1124.3
##
## Number of Fisher Scoring iterations: 14
```

```
# Model Evaluation
pred_result <- stroke_test %>%  # Test dataset as input
  add_predictions(fit1, "stroke_pred") %>%  # Make prediction
  mutate(stroke_predicted = exp(stroke_pred)/(1+exp(stroke_pred))) # calculate result of logistic regre

# Plot ROC curve and calculate AUC score
stroke_roc = roc(pred_result$stroke, pred_result$stroke_predicted)
plot.roc(stroke_roc)
```



```
stroke_auc = auc(pred_result$stroke, pred_result$stroke_predicted)
```

- ***Significant Variables:***
- **age: 3 stars**
- **hypertension: 1 star**
- **avg_glucose_level: 3 stars**

AUC score: 0.8604357

Use feature selection to determine the significant variables

5. Instead of picking the variables by eyeballing, use feature selection to determine the variables you will use in the logistic regression. Pick one of the following options:

A.Put all the variables from 'stroke' into the stepwise selection, report the significant variables

B.[Hard one] Expand your 'stroke' by transforming your existing variables into more variables (e.g. square all the numeric variables;find the interaction term between each variables),then put all the variables (original variables plus the new variables you create) from 'stroke' into the stepwise selection, report the significant variables

```
# Option A

# Fit a linear regression for all variables and Input to Stepwise Selection Model
```

```r
fit2 <- lm(stroke ~ ., data=stroke_train)
step.model <- stepAIC(fit2, direction="both", trace=FALSE, k=2, step=5000)

# Get the summary of step model fit and Rename the coefficients
step.model.sum <- summary(step.model)
colnames(step.model.sum$coefficients) = c("est","std","T","Pr")

# Select the rows with significant variables and fetch the significant variables
sig.coef <- row.names(as.data.frame(step.model.sum$coefficients) %>%
  filter(Pr <= 0.05))
# Debugging: In case "(Intercept)" becomes part of the variables
sig.coef <- sig.coef[-1]
# Debugging: In case variable name cause bug in the later model formulation
index = which(sig.coef == "work_typeSelf-employed")
if(!identical(index, integer(0))){
  sig.coef[index] = "work_typeSelf_employed"
}
index = which(sig.coef == "never smoked")
if(!identical(index, integer(0))){
  sig.coef[index] = "never_smoked"
}
index = which(sig.coef == "formerly smoked")
if(!identical(index, integer(0))){
  sig.coef[index] = "formerly_smoked"
}
```

The significant variables are: age, hypertension, heart_disease, ever_marriedYes, work_typeGovt_job, work_typePrivate, work_typeSelf_employed, avg_glucose_level

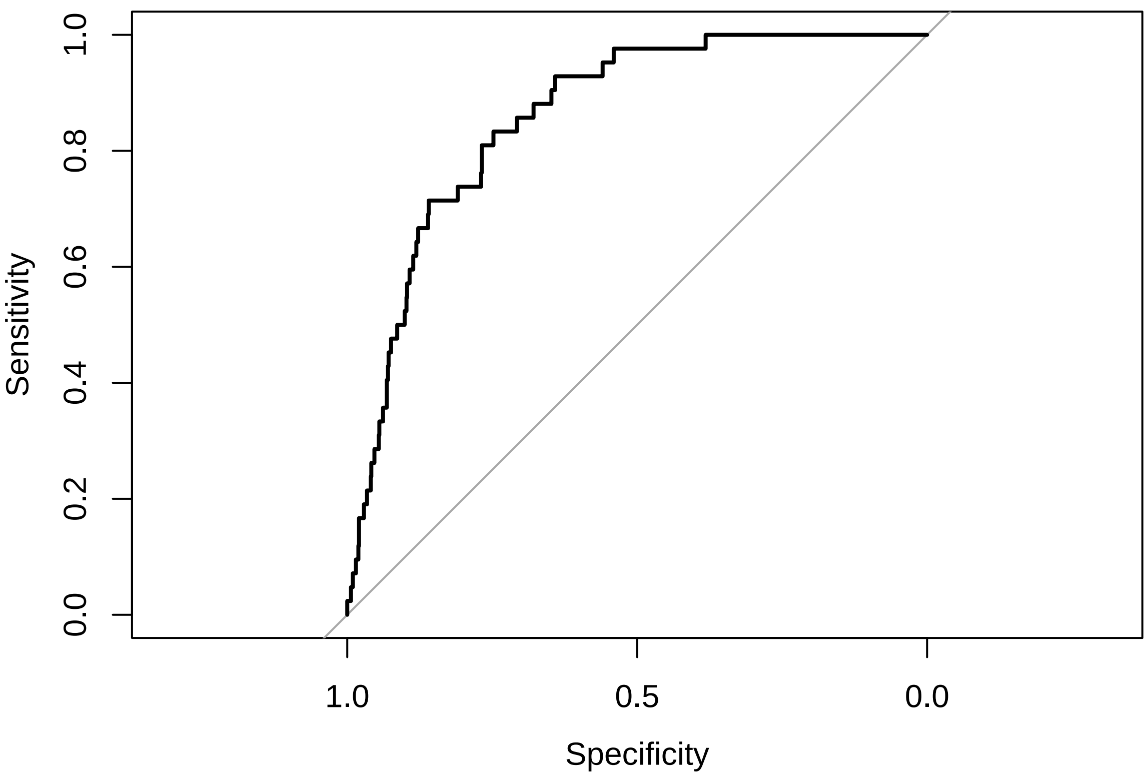
Train logistic regression

6. Use the significant variables from step 5 to train a logistic regression, report the significant variables and the ROC on testing data set.

```r
# concatenate all significant variables into formula
sig.coef.concat <- paste(sig.coef, collapse="+")

# Create model matrix to deal with the multicollinearity caused by categorical variables (remove interc
# Training Set
stroke_train_mat <- model_matrix(stroke_train, stroke ~.-1)
stroke_train_mat$stroke <- stroke_train$stroke
# Testing Set
stroke_test_mat <- model_matrix(stroke_test, stroke ~.-1)
stroke_test_mat$stroke <- stroke_test$stroke

# Debugging: In case variable name cause bug in the later model formulation
# Training Set
colnames(stroke_train_mat)[which(colnames(stroke_train_mat) == "work_typeSelf-employed")] = "work_typeS
colnames(stroke_train_mat)[which(colnames(stroke_train_mat) == "never smoked")] = "never_smoked"
colnames(stroke_train_mat)[which(colnames(stroke_train_mat) == "formerly smoked")] = "formerly_smoked"
# Testing Set
colnames(stroke_test_mat)[which(colnames(stroke_test_mat) == "work_typeSelf-employed")] = "work_typeSel
colnames(stroke_test_mat)[which(colnames(stroke_test_mat) == "never smoked")] = "never_smoked"
colnames(stroke_test_mat)[which(colnames(stroke_test_mat) == "formerly smoked")] = "formerly_smoked"

# Logistic Regression Model Training
fit3 <- glm(formula=as.formula(paste("stroke ~", sig.coef.concat, collapse="")),
            data=stroke_train_mat,
            family="binomial")

summary(fit3)
```

```
##
## Call:
## glm(formula = as.formula(paste("stroke ~", sig.coef.concat, collapse = "")),
##     family = "binomial", data = stroke_train_mat)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.2373  -0.2961  -0.1593  -0.0822   3.4823
##
## Coefficients:
##                         Estimate Std. Error z value Pr(>|z|)
## (Intercept)            -7.259224   1.010547  -7.183  6.8e-13 ***
## age                     0.068973   0.006782  10.170  < 2e-16 ***
## hypertension            0.587016   0.193110   3.040  0.00237 **
## heart_disease           0.551495   0.219670   2.511  0.01205 *
## ever_marriedYes        -0.100793   0.281107  -0.359  0.71993
## work_typeGovt_job      -0.377611   1.104985  -0.342  0.73255
## work_typePrivate       -0.212816   1.087792  -0.196  0.84489
## work_typeSelf_employed -0.607369   1.112635  -0.546  0.58515
## avg_glucose_level       0.004017   0.001407   2.855  0.00430 **
## ---
## Signif. codes:  0 ’***’ 0.001 ’**’ 0.01 ’*’ 0.05 ’.’ 0.1 ’ ’ 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 1381.4  on 3926  degrees of freedom
## Residual deviance: 1097.6  on 3918  degrees of freedom
## AIC: 1115.6
##
## Number of Fisher Scoring iterations: 8
```

```r
# Model Evaluation
pred_result2 <- stroke_test_mat %>%  # Test dataset as input
  add_predictions(fit3, "stroke_pred") %>%  # Make prediction
  mutate(stroke_predicted = exp(stroke_pred)/(1+exp(stroke_pred))) # calculate result of logistic regre

# Plot ROC curve and calculate AUC score
stroke_roc2 = roc(pred_result2$stroke, pred_result2$stroke_predicted)
plot.roc(stroke_roc2)
```



```
stroke_auc2 = auc(pred_result2$stroke, pred_result2$stroke_predicted)
```

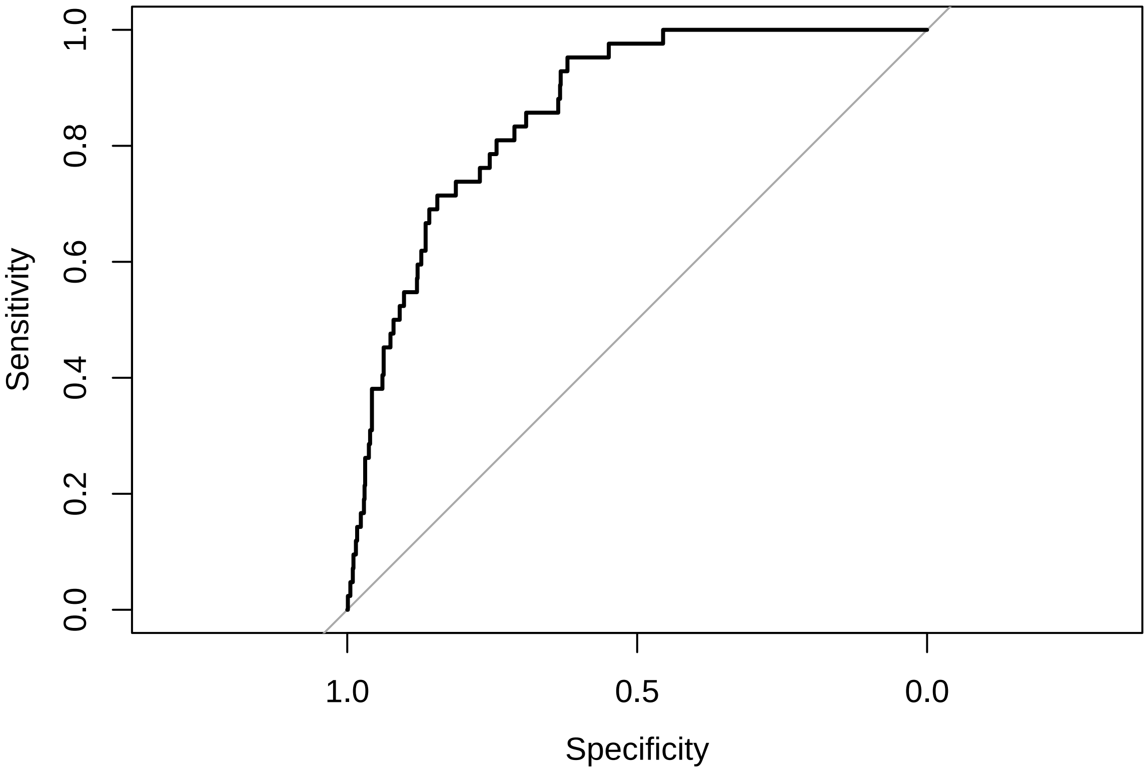
- ***Significant Variables:***
- **age: 3 stars**
- **hypertension: 2 star**
- **avg_glucose_level: 2 stars**

AUC score: 0.8590679

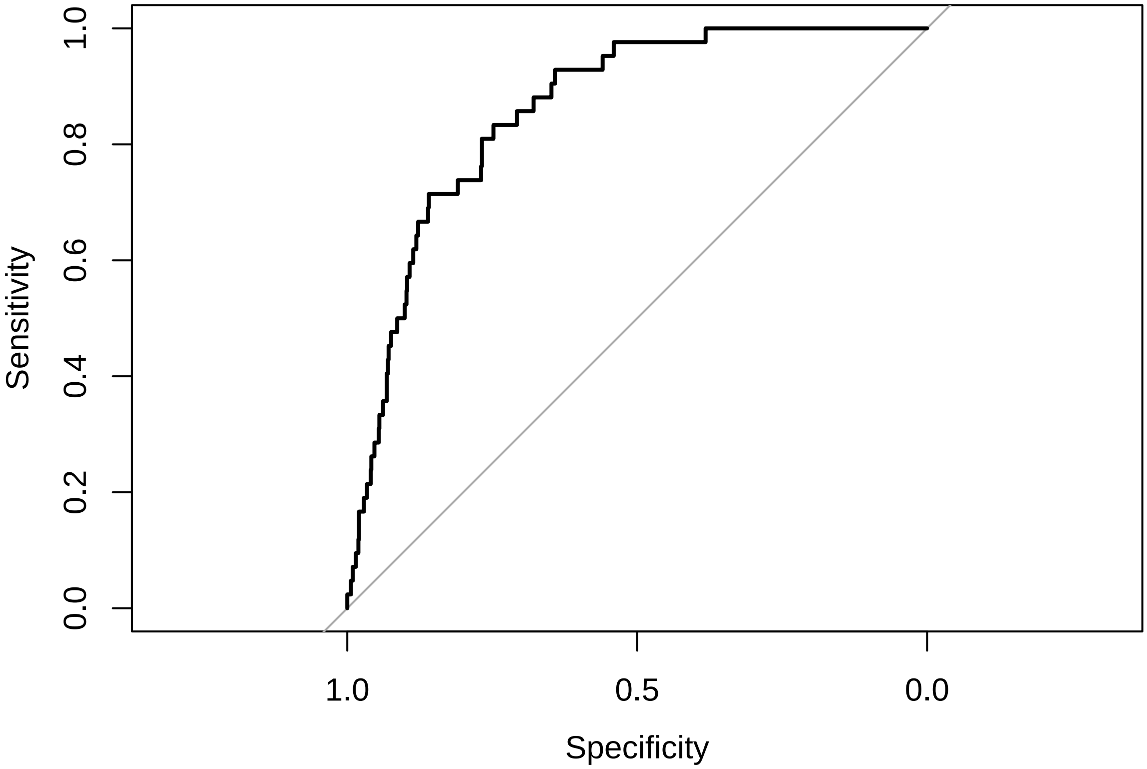
Comparison

7. Compare the ROC between the first model with manually picked variables against the second model with feature selection. Which one does better? Why do you think that's the case?

```
# Compare the ROC between two models
plot.roc(stroke_roc)
```



```
plot.roc(stroke_roc2)
```



```
# Detect the Difference between ROC with model of manual selection and ROC with model of stepwise selec
roc.test(stroke_roc, stroke_roc2)
```

```
##
##  DeLong's test for two correlated ROC curves
##
## data:  stroke_roc and stroke_roc2
## Z = 0.24739, p-value = 0.8046
## alternative hypothesis: true difference in AUC is not equal to 0
## 95 percent confidence interval:
##  -0.009468447  0.012204009
## sample estimates:
## AUC of roc1 AUC of roc2
##   0.8604357   0.8590679
```

- ***NOTE: The result is recorded before submission, result might vary each time you run it***
- ***Comparison of ROC between first and second model:***
- **AUC for stepwise selection with metric BIC: 0.8720**
- **AUC for manual feature selection: 0.8647**

Observation of the Comparison: The result shows that the model of stepwise selection with BIC metric is better than model of manual selection because it uses iterative process in a sub-optimal way to find all significant variables instead of picking by eyeballing. However, the stepwise selection with AIC metric is underperforming because all the variable "stepwise selection" algorithm choose has less significant variables which cannot to be explained all the variance of the dataset.