#Load in the packages

```
library(ggplot2)
library(tidyverse)
```

```
## -- Attaching packages --------------------------------------- tidyverse
1.3.1 --
```

```
## v tibble  3.1.6     v dplyr   1.0.7
## v tidyr   1.1.4     v stringr 1.4.0
## v readr   2.1.1     v forcats 0.5.1
## v purrr   0.3.4
```

```
## -- Conflicts ------------------------------------------
tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```
library(dplyr)
```

#Load in the data

```
data <- read.csv("C:/Users/oriri/OneDrive/Desktop/DSC680/Project
2/stroke_data.csv")
```

#cleaning the data

```
#removing column id
data <- data %>%
  select(-c(id)) %>%
  filter(gender != "Other") %>%
  mutate(stroke = as.factor(stroke))
```

#getting a count of all the stroke events

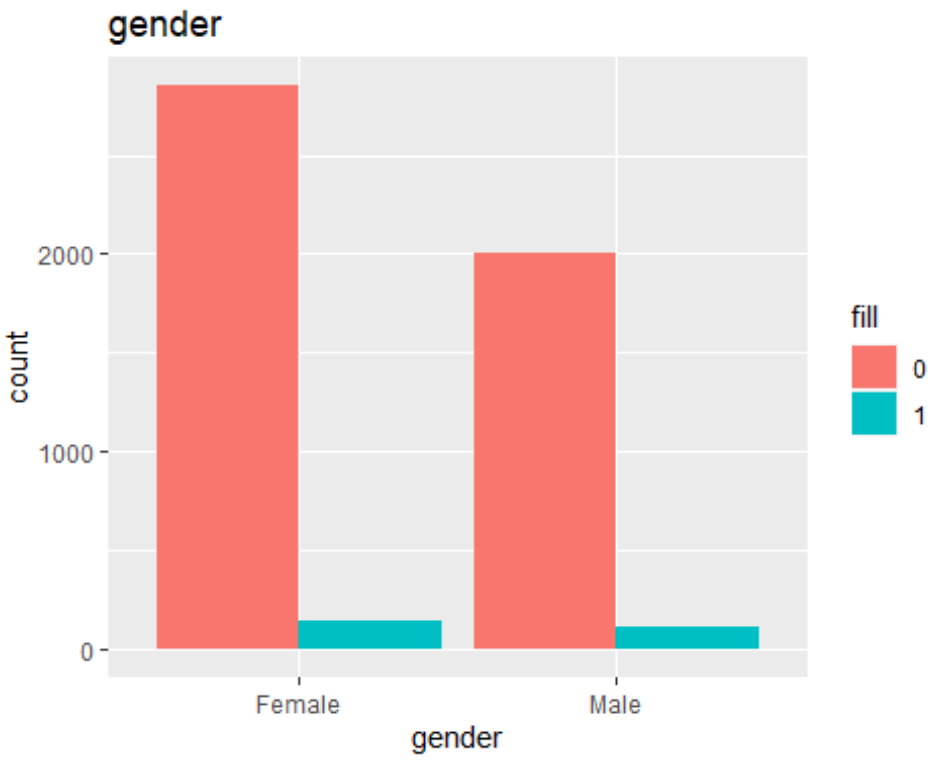
```
data %>%
  group_by(stroke) %>%
  count()
```

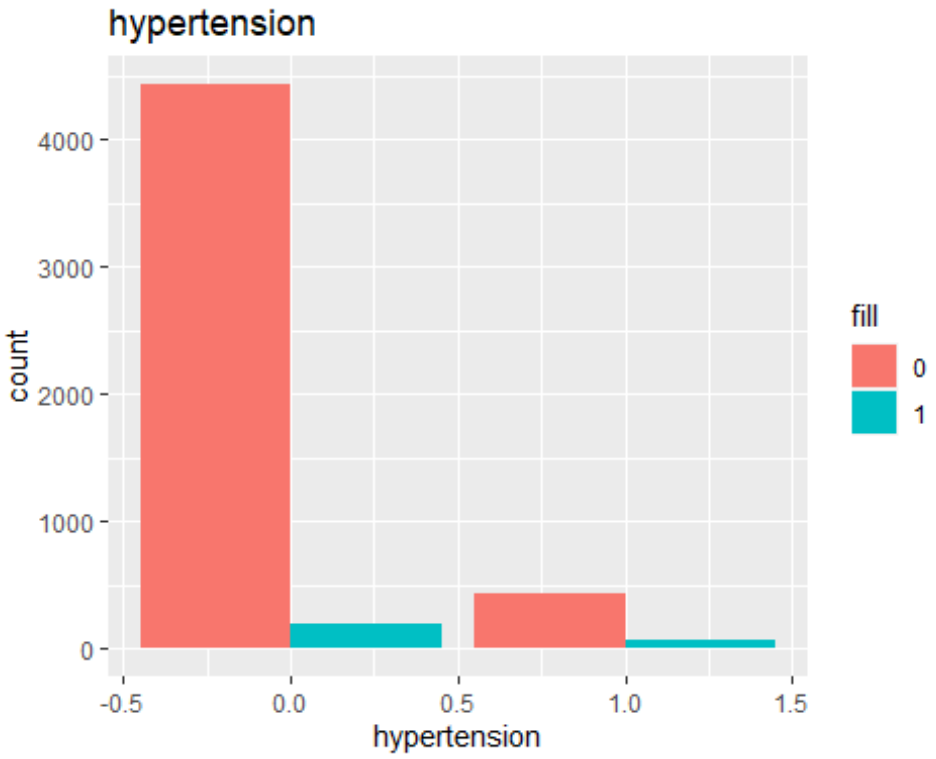
```
## # A tibble: 2 x 2
## # Groups:   stroke [2]
##   stroke     n
##   <fct>  <int>
## 1 0       4860
## 2 1        249
```

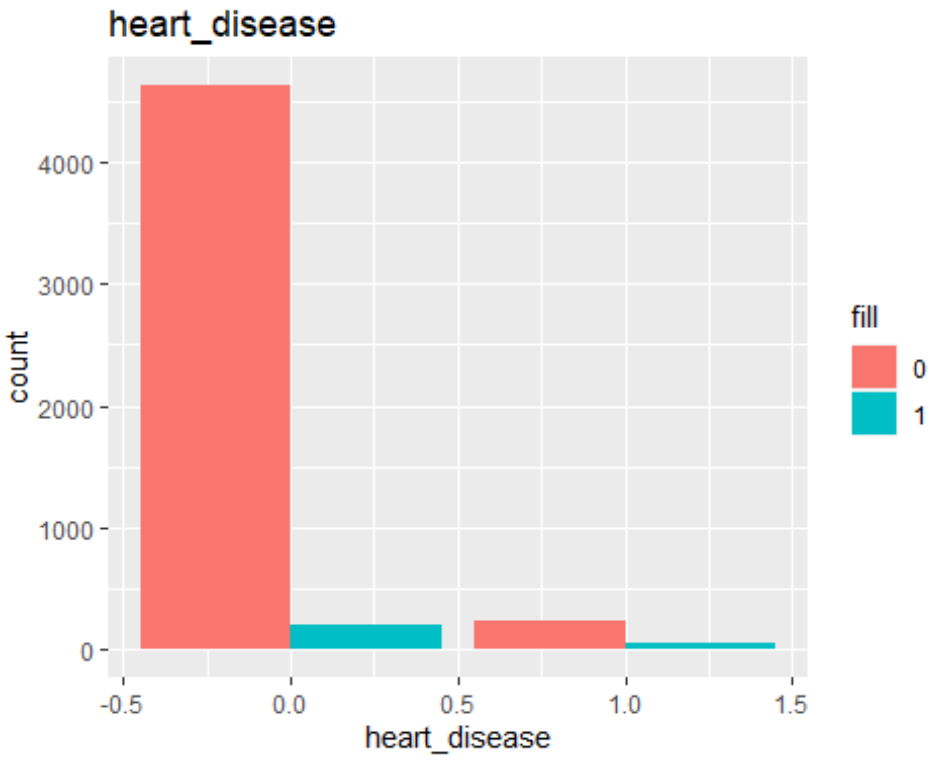
#Visualizing the categorical variables against stroke status

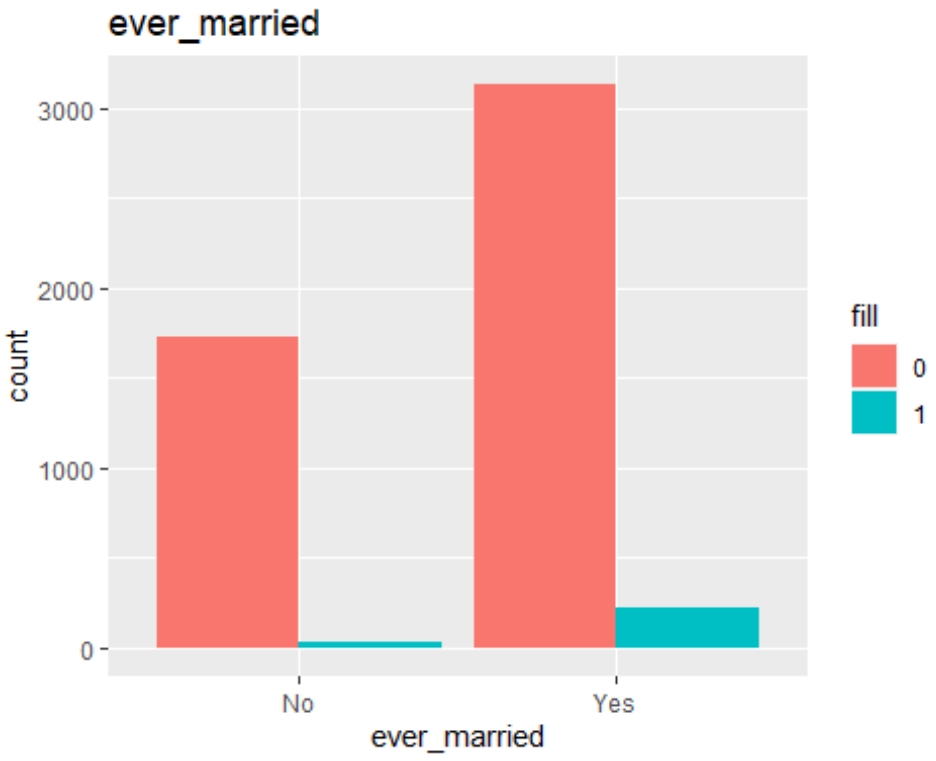
```
#putting all of the categorical variables in a list
cat_list <- c("gender", "hypertension", "heart_disease", "ever_married",
"Residence_type", "smoking_status")
```

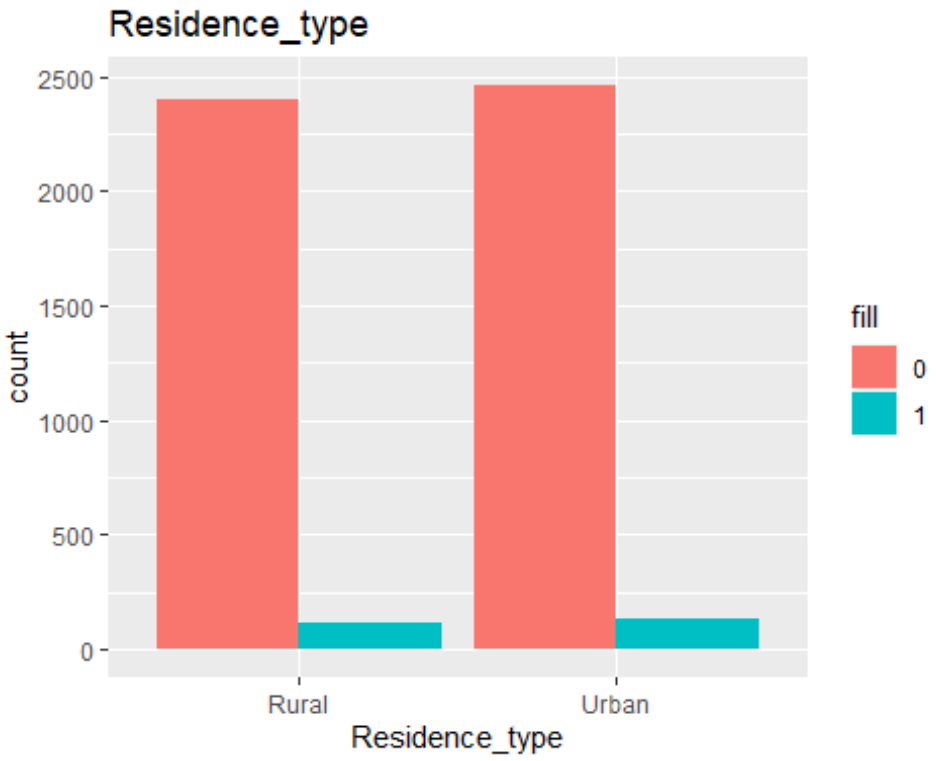
```r
for(i in cat_list){
  plot <- ggplot(data = data, aes_string(x = i, fill = data$stroke)) +
          geom_bar(position = "dodge") +
          ggtitle(i)

  print(plot)
}
```

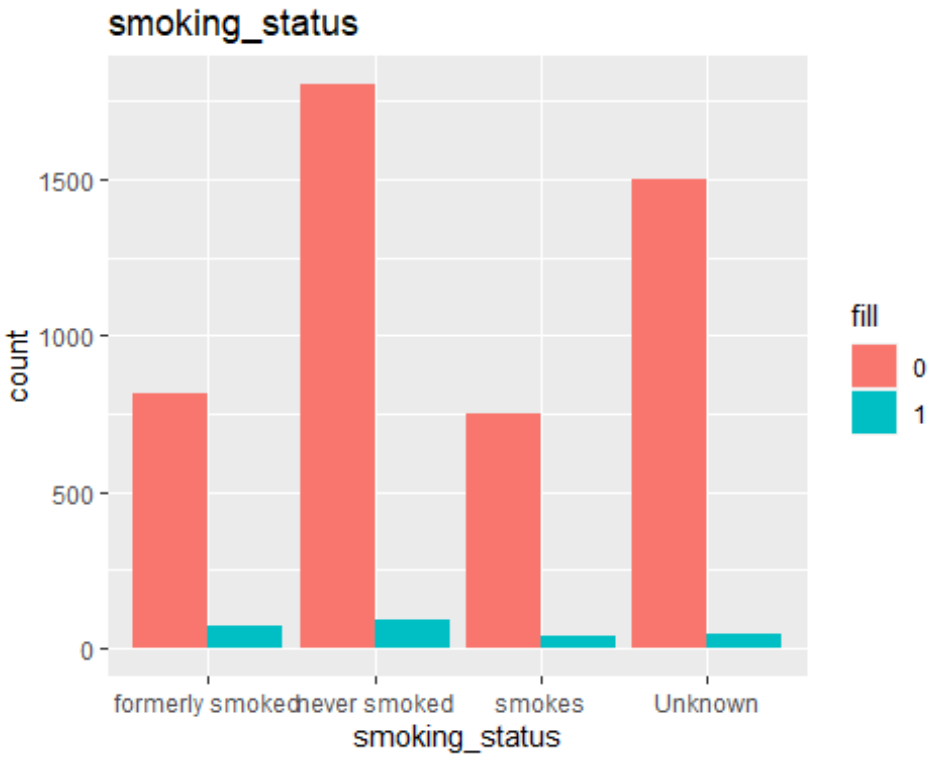

gender
2000
1000
0
count
Female
Male
gender
fill
0
1


hypertension
4000
3000
2000
1000
0
count
-0.5
0.0
0.5
1.0
1.5
hypertension
fill
0
1


heart_disease
4000
3000
2000
1000
0
count
-0.5
0.0
0.5
1.0
1.5
heart_disease
fill
0
1


ever_married
3000
2000
1000
0
count
No
Yes
ever_married
fill
0
1





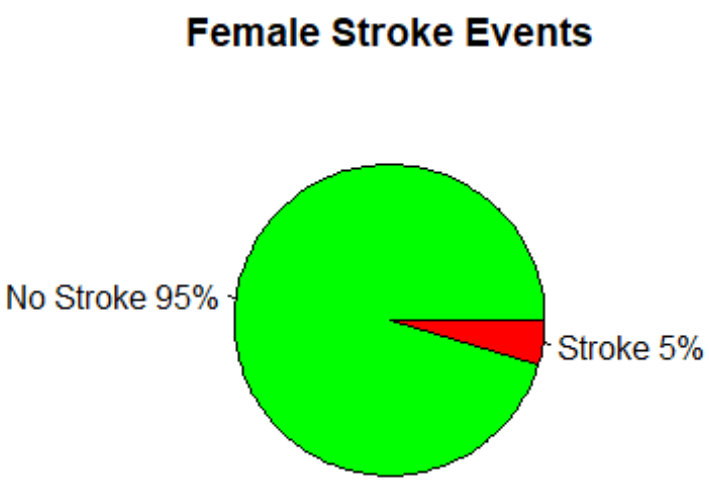
#getting proportions of the categorical variables #gender

```
#Count of stroke by gender
data %>%
```

```
  group_by(gender, stroke) %>%
  count()
```

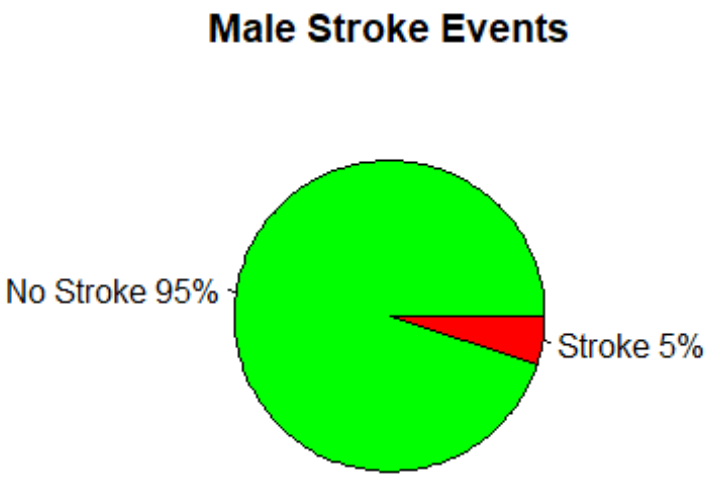
```
## # A tibble: 4 x 3
## # Groups:   gender, stroke [4]
##   gender stroke     n
##   <chr>  <fct>  <int>
## 1 Female 0       2853
## 2 Female 1        141
## 3 Male   0       2007
## 4 Male   1        108
```

```
#Female
slices <- c(2853, 141)
lbls <- c("No Stroke", "Stroke")
pct <- round(slices/sum(slices) * 100)
lbls <- paste(lbls, pct)
lbls <- paste(lbls,"%", sep = "")
pie(slices,labels = lbls, col=c("green", "red"),
   main="Female Stroke Events")
```




```
#Male
slices <- c(2007, 108)
lbls <- c("No Stroke", "Stroke")
pct <- round(slices/sum(slices) * 100)
lbls <- paste(lbls, pct)
lbls <- paste(lbls,"%", sep = "")
```

```
pie(slices,labels = lbls, col=c("green", "red"),
   main="Male Stroke Events")
```



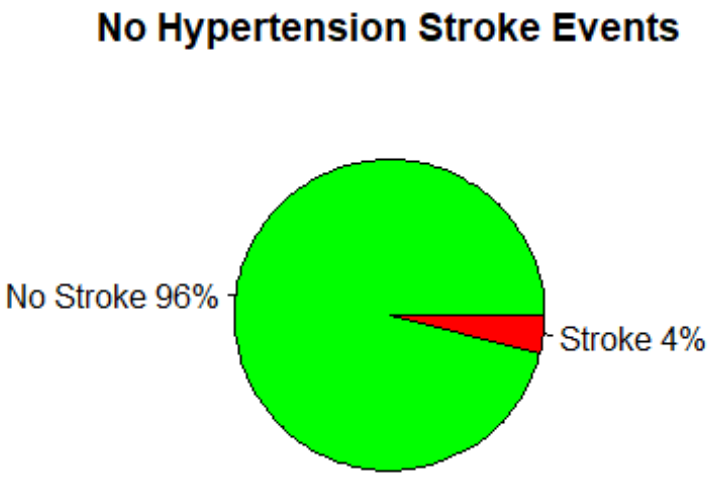

```
#2 proportion test
prop.test(x = c(141, 108), n = c(2994, 2115))

##
##  2-sample test for equality of proportions with continuity correction
##
## data:  c(141, 108) out of c(2994, 2115)
## X-squared = 0.34, df = 1, p-value = 0.5598
## alternative hypothesis: two.sided
## 95 percent confidence interval:
##  -0.016439097  0.008499814
## sample estimates:
##     prop 1     prop 2
## 0.04709419 0.05106383
```

#hypertension

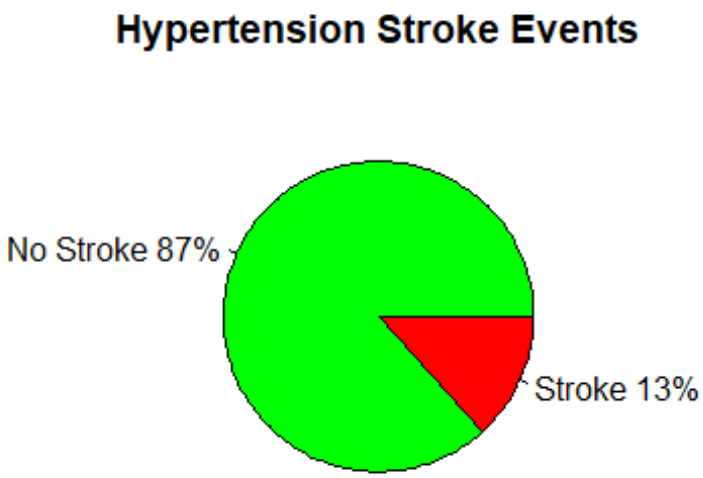
```
#Getting counts of stroke events by hyper tension
data %>%
  group_by(hypertension, stroke) %>%
  count()

## # A tibble: 4 x 3
## # Groups:   hypertension, stroke [4]
##   hypertension stroke     n
```

```
##          <int> <fct>  <int>
## 1            0 0       4428
## 2            0 1        183
## 3            1 0        432
## 4            1 1         66
```

```
#No Hypertension
slices <- c(4428, 183)
lbls <- c("No Stroke", "Stroke")
pct <- round(slices/sum(slices) * 100)
lbls <- paste(lbls, pct)
lbls <- paste(lbls,"%", sep = "")
pie(slices,labels = lbls, col=c("green", "red"),
   main="No Hypertension Stroke Events")
```




```
#Hypertension
slices <- c(432, 66)
lbls <- c("No Stroke", "Stroke")
pct <- round(slices/sum(slices) * 100)
lbls <- paste(lbls, pct)
lbls <- paste(lbls,"%", sep = "")
pie(slices,labels = lbls, col=c("green", "red"),
   main="Hypertension Stroke Events")
```




```
#2 proportion test
prop.test(x = c(183, 66), n = c(4611, 498))

##
##  2-sample test for equality of proportions with continuity correction
##
## data:  c(183, 66) out of c(4611, 498)
## X-squared = 81.573, df = 1, p-value < 2.2e-16
## alternative hypothesis: two.sided
## 95 percent confidence interval:
##  -0.1242628 -0.0614220
## sample estimates:
##    prop 1    prop 2
## 0.0396877 0.1325301
```

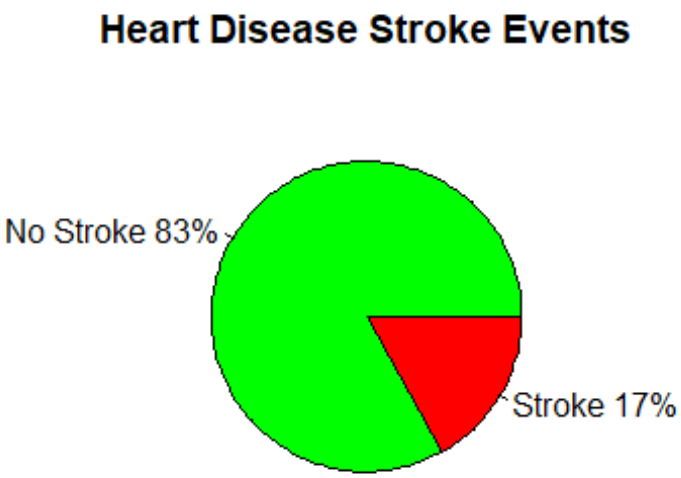
#heart_disease

```
#Getting counts of stroke by if a patient has heart disease
data %>%
  group_by(heart_disease, stroke) %>%
  count()

## # A tibble: 4 x 3
## # Groups:   heart_disease, stroke [4]
##   heart_disease stroke     n
##           <int> <fct>  <int>
## 1             0 0       4631
```

```
## 2             0 1        202
## 3             1 0        229
## 4             1 1         47
```

```
#No Hypertension
slices <- c(4631, 202)
lbls <- c("No Stroke", "Stroke")
pct <- round(slices/sum(slices) * 100)
lbls <- paste(lbls, pct)
lbls <- paste(lbls,"%", sep = "")
pie(slices,labels = lbls, col=c("green", "red"),
   main="No Heart Disease Stroke Events")
```


```
#Hypertension
slices <- c(229, 47)
lbls <- c("No Stroke", "Stroke")
pct <- round(slices/sum(slices) * 100)
lbls <- paste(lbls, pct)
lbls <- paste(lbls,"%", sep = "")
pie(slices,labels = lbls, col=c("green", "red"),
   main="Heart Disease Stroke Events")
```




```
#2 proportion test
prop.test(x = c(202, 47), n = c(4833, 276))

##
##  2-sample test for equality of proportions with continuity correction
##
## data:  c(202, 47) out of c(4833, 276)
## X-squared = 90.229, df = 1, p-value < 2.2e-16
## alternative hypothesis: two.sided
## 95 percent confidence interval:
##  -0.17511205 -0.08187568
## sample estimates:
##     prop 1     prop 2
## 0.04179599 0.17028986
```

#ever_married

```
#Getting counts of if a patient was married by stroke
data %>%
  group_by(ever_married, stroke) %>%
  count()

## # A tibble: 4 x 3
## # Groups:   ever_married, stroke [4]
##   ever_married stroke     n
##   <chr>        <fct>  <int>
## 1 No           0       1727
```

```
## 2 No           1         29
## 3 Yes          0       3133
## 4 Yes          1        220
```

```
#Never Married
slices <- c(1727, 29)
lbls <- c("No Stroke", "Stroke")
pct <- round(slices/sum(slices) * 100)
lbls <- paste(lbls, pct)
lbls <- paste(lbls,"%", sep = "")
pie(slices,labels = lbls, col=c("green", "red"),
   main="Never Married Stroke Events")
```


```
#Married
slices <- c(3133, 220)
lbls <- c("No Stroke", "Stroke")
pct <- round(slices/sum(slices) * 100)
lbls <- paste(lbls, pct)
lbls <- paste(lbls,"%", sep = "")
pie(slices,labels = lbls, col=c("green", "red"),
   main="Married Stroke Events")
```

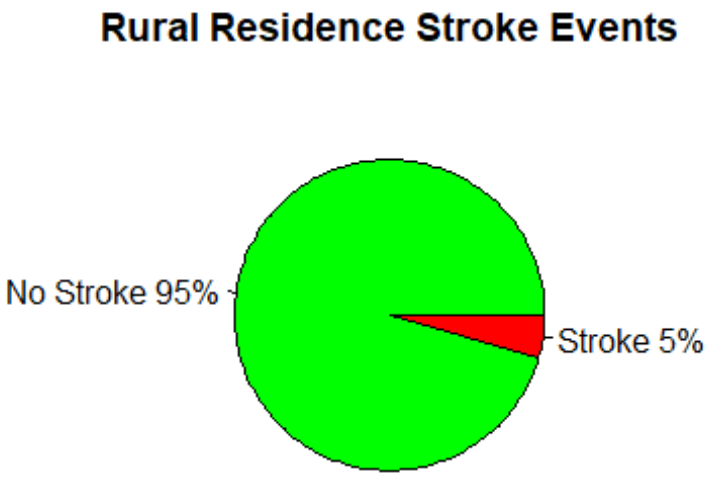

```
#2 proportion test
prop.test(x = c(29, 220), n = c(1156, 3353))

##
##  2-sample test for equality of proportions with continuity correction
##
## data:  c(29, 220) out of c(1156, 3353)
## X-squared = 26.289, df = 1, p-value = 2.939e-07
## alternative hypothesis: two.sided
## 95 percent confidence interval:
##  -0.05341703 -0.02763572
## sample estimates:
##     prop 1     prop 2
## 0.02508651 0.06561288
```

#Residence_type

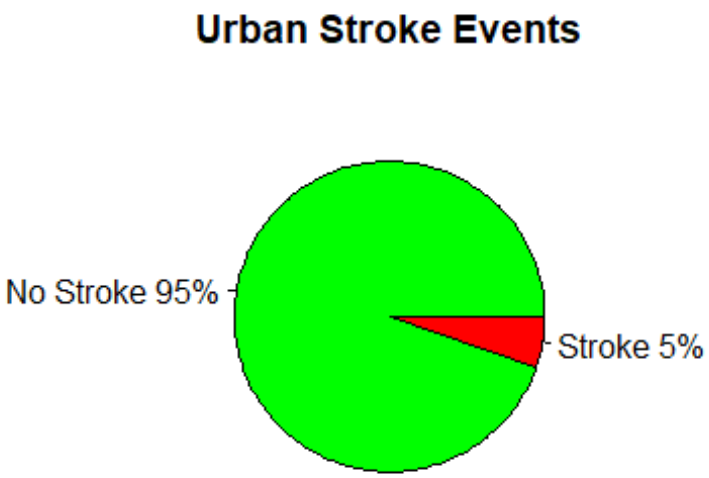
```
#Getting Counts of stroke events by residence type
data %>%
  group_by(Residence_type, stroke) %>%
  count()

## # A tibble: 4 x 3
## # Groups:   Residence_type, stroke [4]
##   Residence_type stroke     n
##   <chr>          <fct>  <int>
## 1 Rural          0       2399
```

```
## 2 Rural          1        114
## 3 Urban          0       2461
## 4 Urban          1        135
```

```
#Rural Residence
slices <- c(2399, 114)
lbls <- c("No Stroke", "Stroke")
pct <- round(slices/sum(slices) * 100)
lbls <- paste(lbls, pct)
lbls <- paste(lbls,"%", sep = "")
pie(slices,labels = lbls, col=c("green", "red"),
   main="Rural Residence Stroke Events")
```




```
#Urban Residence
slices <- c(2461, 135)
lbls <- c("No Stroke", "Stroke")
pct <- round(slices/sum(slices) * 100)
lbls <- paste(lbls, pct)
lbls <- paste(lbls,"%", sep = "")
pie(slices,labels = lbls, col=c("green", "red"),
   main="Urban Stroke Events")
```




```
#2 proportion test
prop.test(x = c(114, 220), n = c(2513, 2596))
```

```
##
##  2-sample test for equality of proportions with continuity correction
##
## data:  c(114, 220) out of c(2513, 2596)
## X-squared = 31.77, df = 1, p-value = 1.735e-08
## alternative hypothesis: two.sided
## 95 percent confidence interval:
##  -0.05322594 -0.02553738
## sample estimates:
##     prop 1     prop 2
## 0.04536411 0.08474576
```

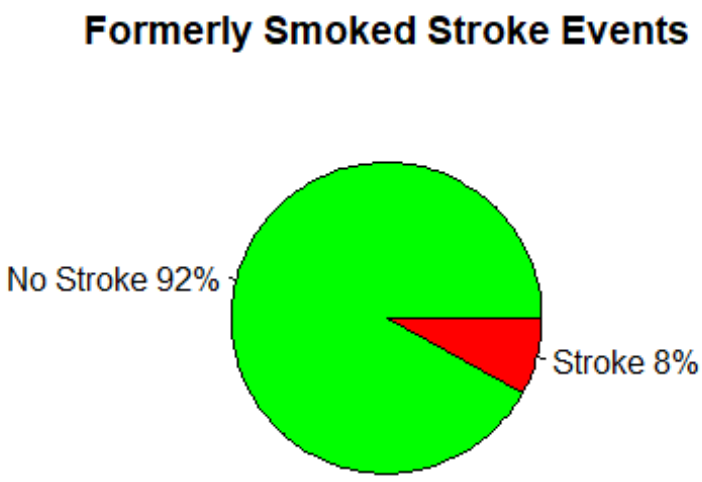
#smoking_status

```
data %>%
  group_by(smoking_status, stroke) %>%
  count()
```

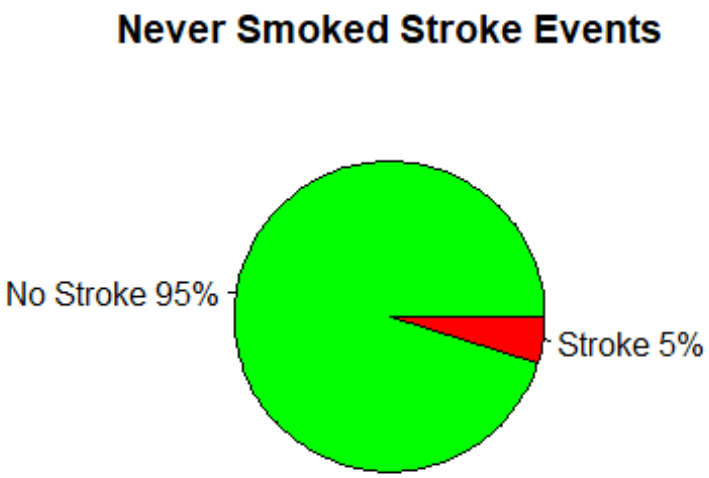
```
## # A tibble: 8 x 3
## # Groups:   smoking_status, stroke [8]
##   smoking_status  stroke     n
##   <chr>           <fct>  <int>
## 1 formerly smoked 0        814
## 2 formerly smoked 1         70
```

```
## 3 never smoked    0       1802
## 4 never smoked    1         90
## 5 smokes          0        747
## 6 smokes          1         42
## 7 Unknown         0       1497
## 8 Unknown         1         47
```

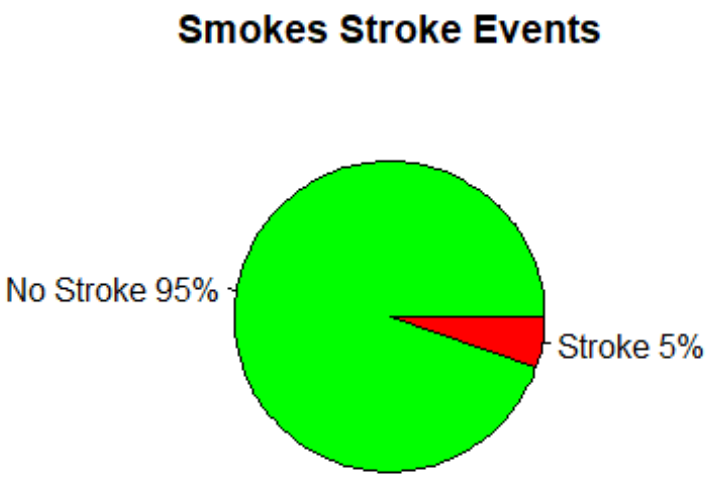
```
#Formerly Smoked
slices <- c(814, 70)
lbls <- c("No Stroke", "Stroke")
pct <- round(slices/sum(slices) * 100)
lbls <- paste(lbls, pct)
lbls <- paste(lbls,"%", sep = "")
pie(slices,labels = lbls, col=c("green", "red"),
   main="Formerly Smoked Stroke Events")
```



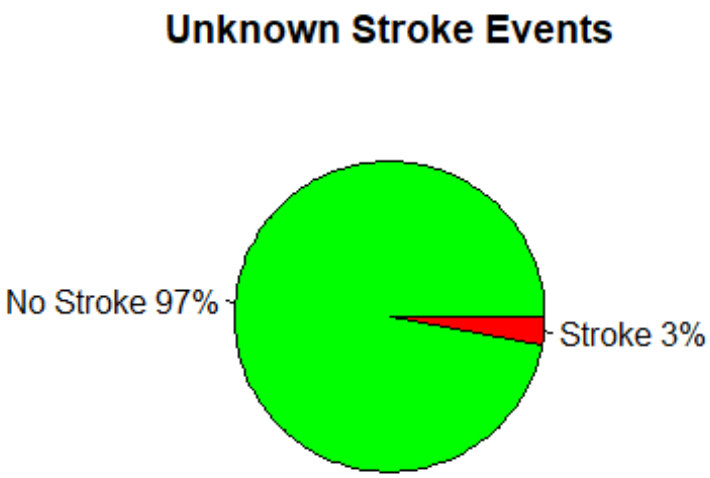

```
#Never Smoked
slices <- c(1802, 90)
lbls <- c("No Stroke", "Stroke")
pct <- round(slices/sum(slices) * 100)
lbls <- paste(lbls, pct)
lbls <- paste(lbls,"%", sep = "")
pie(slices,labels = lbls, col=c("green", "red"),
   main="Never Smoked Stroke Events")
```




```r
#Smokes
slices <- c(747, 42)
lbls <- c("No Stroke", "Stroke")
pct <- round(slices/sum(slices) * 100)
lbls <- paste(lbls, pct)
lbls <- paste(lbls,"%", sep = "")
pie(slices,labels = lbls, col=c("green", "red"),
   main="Smokes Stroke Events")
```




```r
#Unknown
slices <- c(1497, 47)
lbls <- c("No Stroke", "Stroke")
pct <- round(slices/sum(slices) * 100)
lbls <- paste(lbls, pct)
lbls <- paste(lbls,"%", sep = "")
pie(slices,labels = lbls, col=c("green", "red"),
   main="Unknown Stroke Events")
```



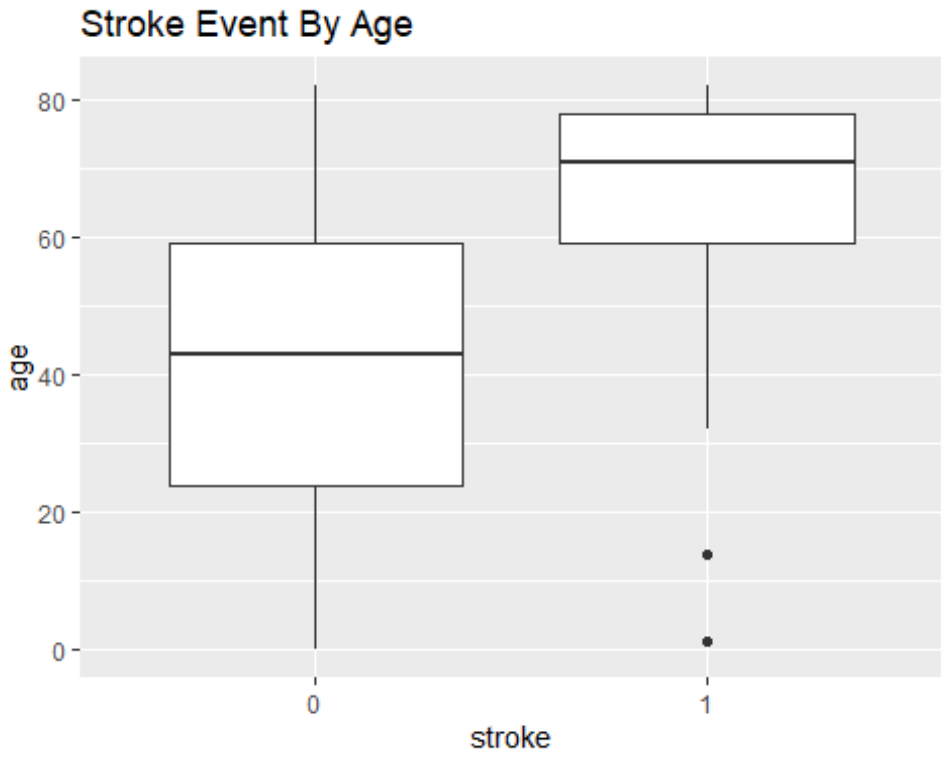
#Investigating Continuous Variables

```
numeric_vars <- data %>%
  select(age, avg_glucose_level, bmi, stroke)

str(numeric_vars)
```

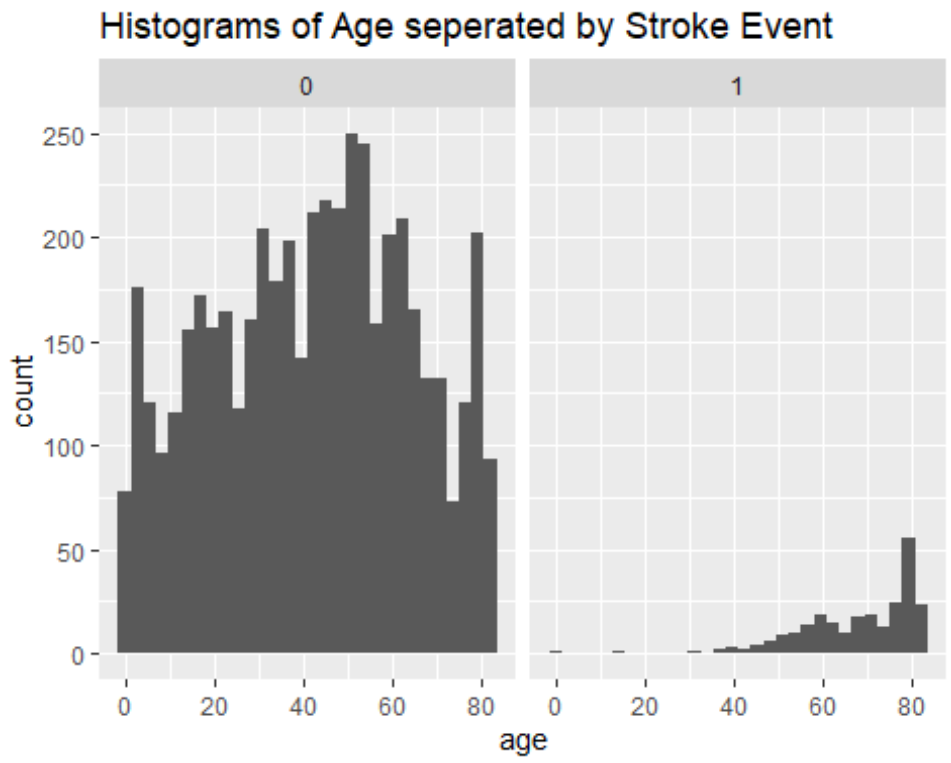
```
## 'data.frame':    5109 obs. of  4 variables:
##  $ age              : num  67 61 80 49 79 81 74 69 59 78 ...
##  $ avg_glucose_level: num  229 202 106 171 174 ...
##  $ bmi              : chr  "36.6" "N/A" "32.5" "34.4" ...
##  $ stroke           : Factor w/ 2 levels "0","1": 2 2 2 2 2 2 2 2 2 2 ...
```

#age

```
numeric_vars %>%
  ggplot(aes(x = stroke, y = age)) +
  geom_boxplot() +
  ggtitle("Stroke Event By Age")
```

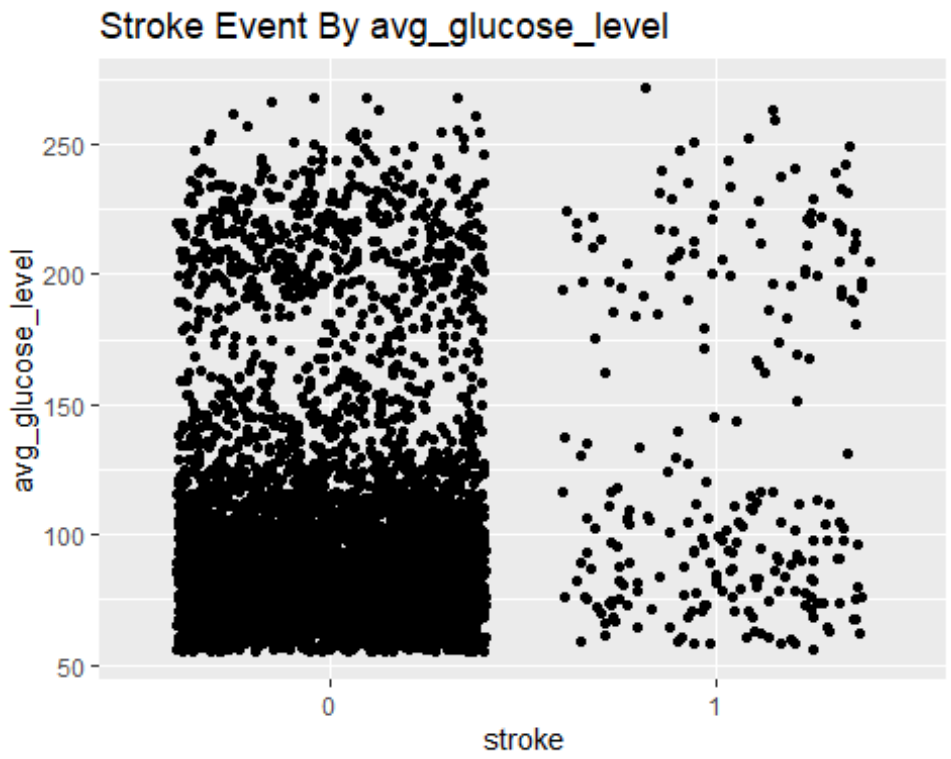



```
numeric_vars %>%
  ggplot(aes(x = age)) +
  geom_histogram() +
  facet_wrap(~stroke) +
  ggtitle("Histograms of Age seperated by Stroke Event")

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

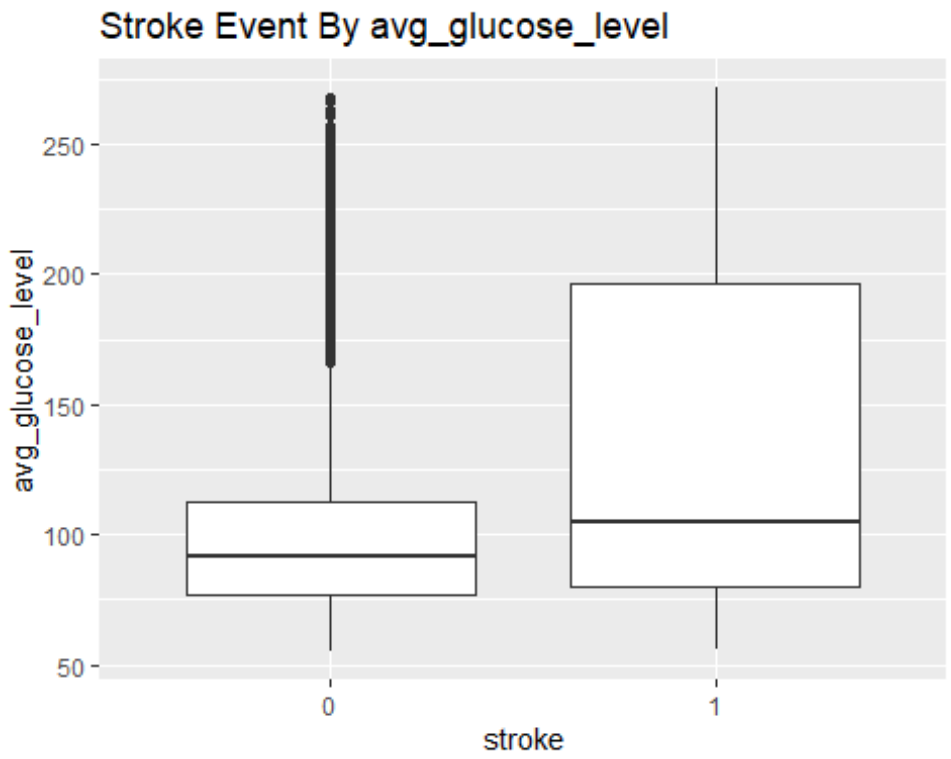


#avg_glucose_level

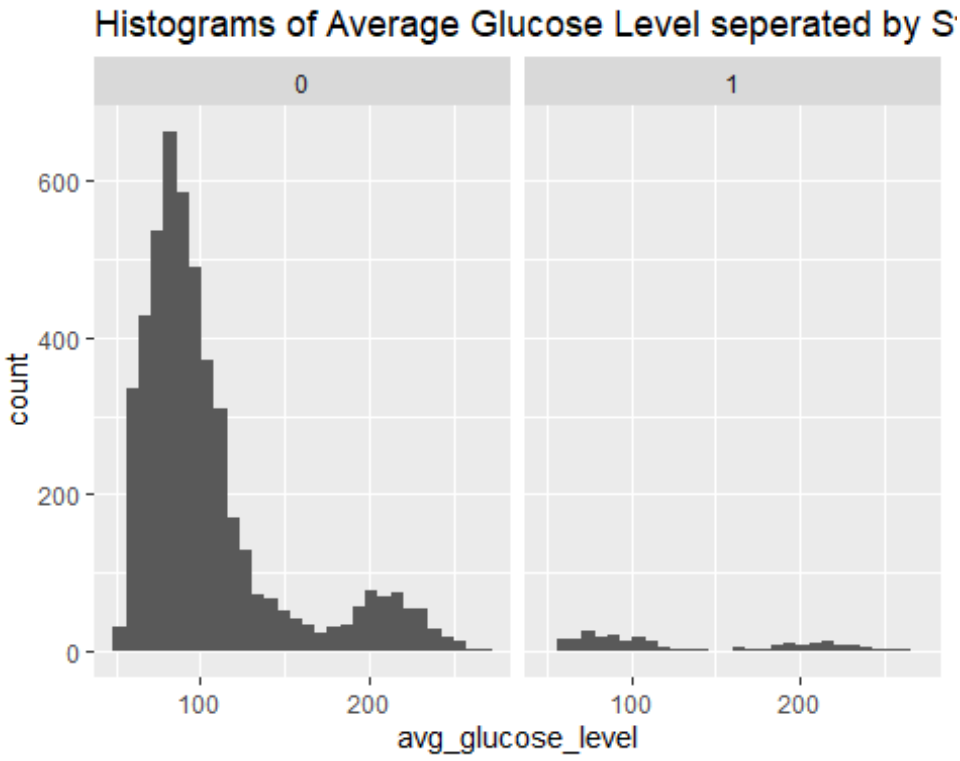
```
numeric_vars %>%
  ggplot(aes(x = stroke, y = avg_glucose_level)) +
  geom_jitter() +
  ggtitle("Stroke Event By avg_glucose_level")
```




```
numeric_vars %>%
  ggplot(aes(x = stroke, y = avg_glucose_level)) +
  geom_boxplot() +
  ggtitle("Stroke Event By avg_glucose_level")
```

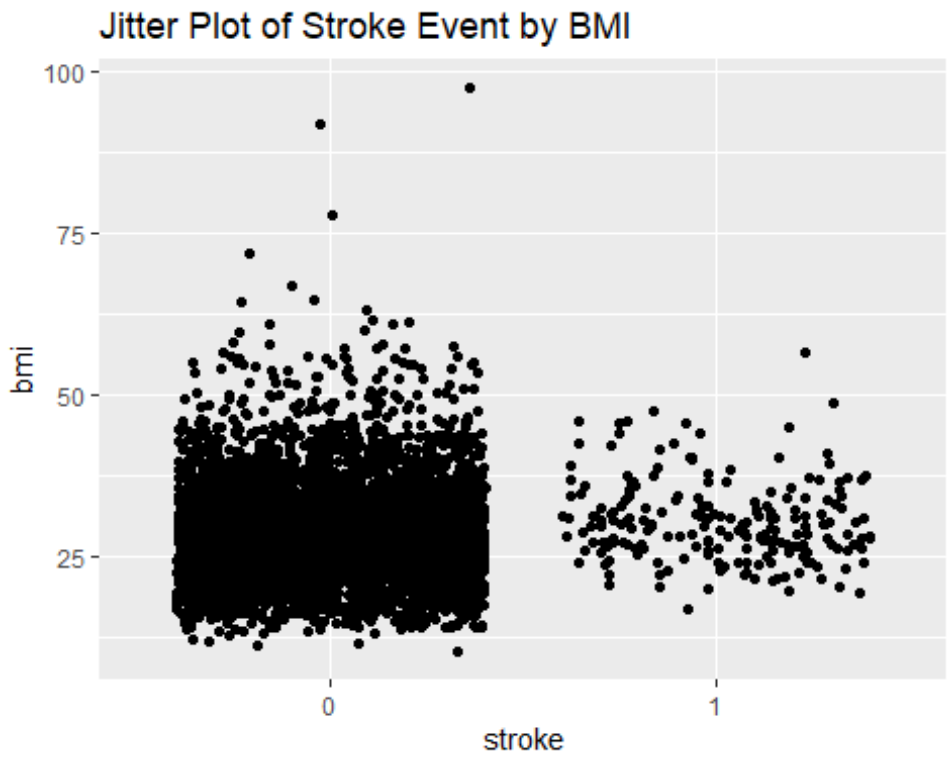



```
numeric_vars %>%
  ggplot(aes(x = avg_glucose_level)) +
  geom_histogram() +
  facet_wrap(~stroke) +
  ggtitle("Histograms of Average Glucose Level seperated by Stroke Event")

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

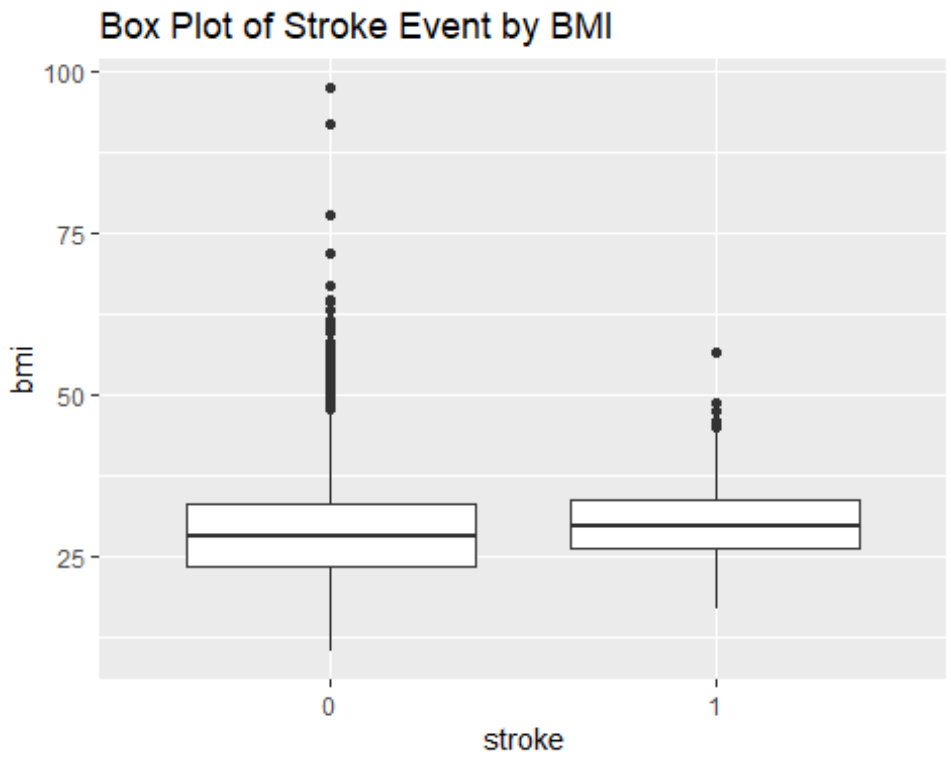


#bmi

```
numeric_vars %>%
  filter(bmi != "N/A") %>%
  mutate(bmi = as.numeric(bmi)) %>%
  ggplot(aes(x = stroke, y = bmi))+
  geom_jitter() +
  ggtitle("Jitter Plot of Stroke Event by BMI")
```




```
numeric_vars %>%
  filter(bmi != "N/A") %>%
  mutate(bmi = as.numeric(bmi)) %>%
  ggplot(aes(x = stroke, y = bmi))+
  geom_boxplot() +
  ggtitle("Box Plot of Stroke Event by BMI")
```

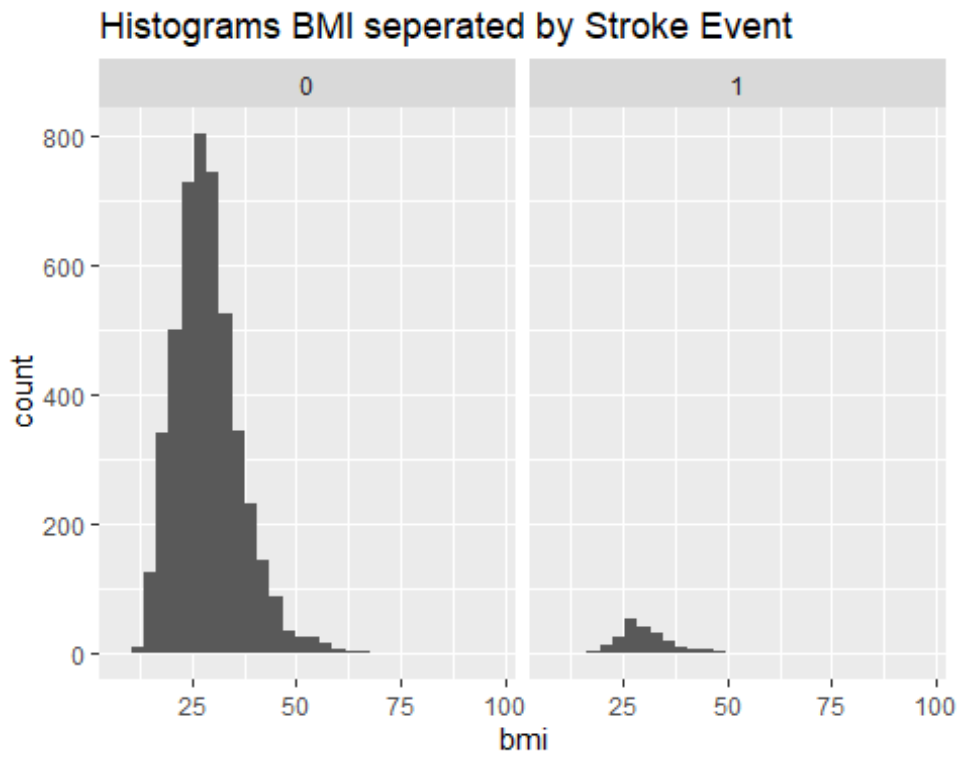



```
numeric_vars %>%
  mutate(bmi = as.numeric(bmi)) %>%
  ggplot(aes(x = bmi)) +
  geom_histogram() +
  facet_wrap(~stroke) +
  ggtitle("Histograms BMI seperated by Stroke Event")
```

```
## Warning in mask$eval_all_mutate(quo): NAs introduced by coercion
```

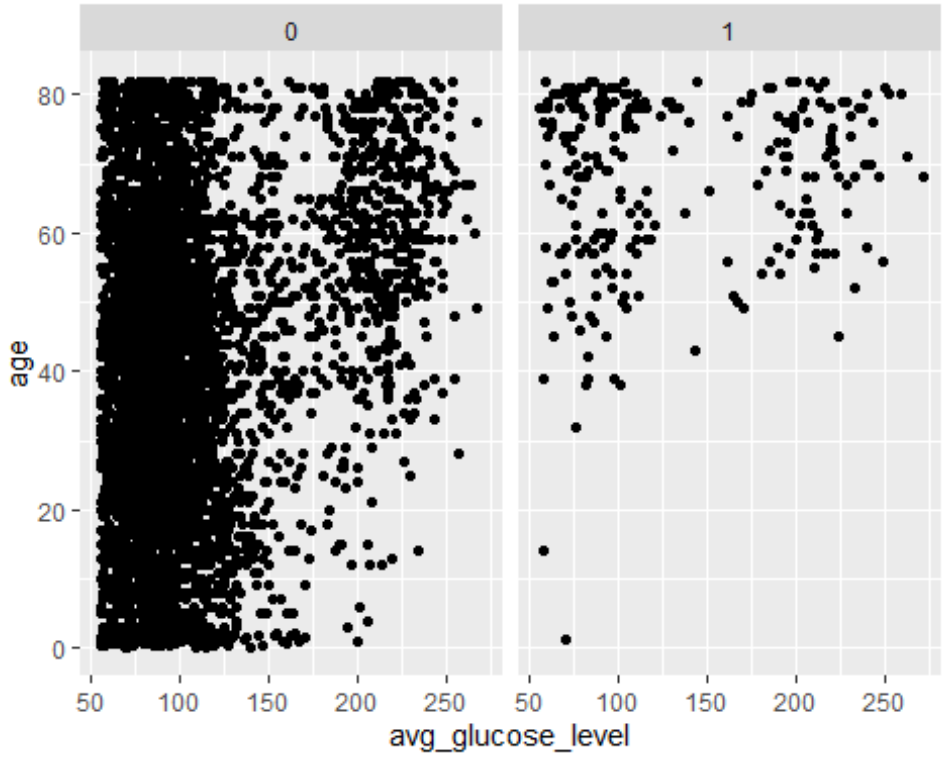
```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```

```
## Warning: Removed 201 rows containing non-finite values (stat_bin).
```

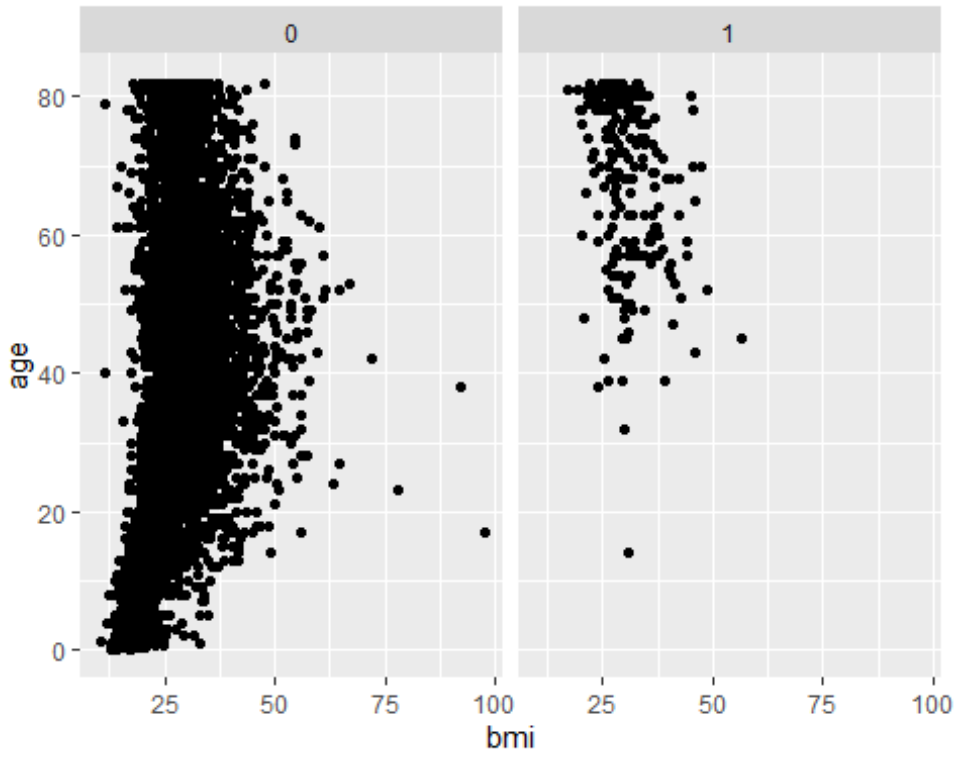


#relationships between numeric variables

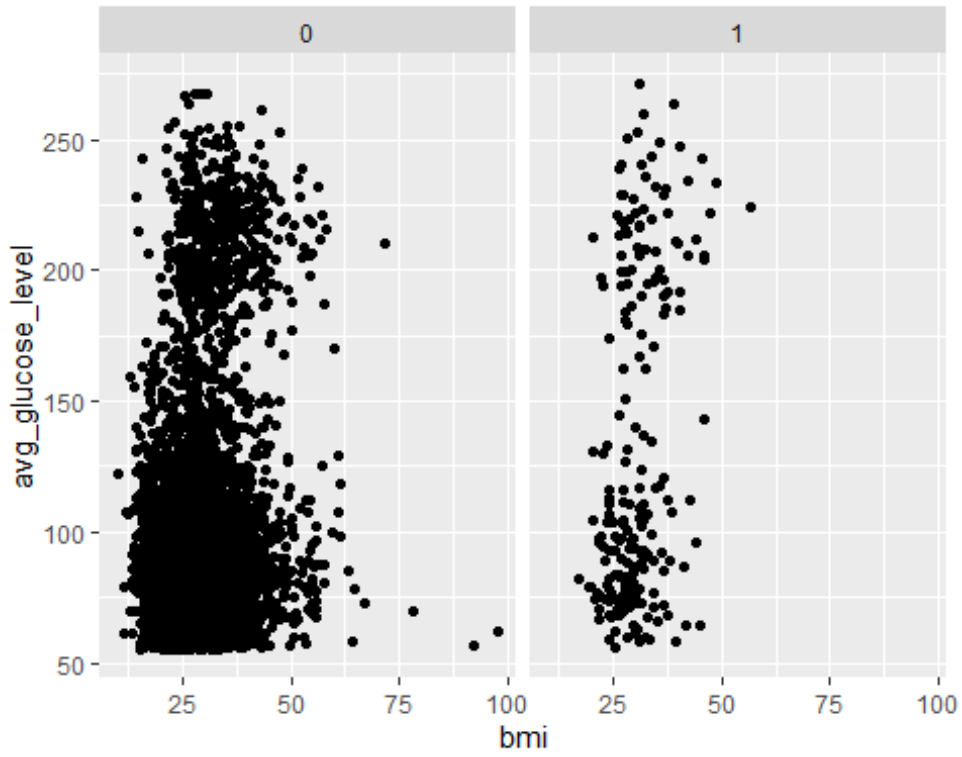
```
ggplot(data, aes(x = avg_glucose_level, y = age)) +
  geom_point() +
  facet_wrap(~stroke)
```




```
numeric_vars %>%
  filter(bmi != "N/A") %>%
  mutate(bmi = as.numeric(bmi)) %>%
  ggplot(aes(x = bmi, y = age)) +
  geom_point() +
  facet_wrap(~stroke)
```




```
numeric_vars %>%
  filter(bmi != "N/A") %>%
  mutate(bmi = as.numeric(bmi)) %>%
  ggplot(aes(x = bmi, y = avg_glucose_level)) +
  geom_point() +
  facet_wrap(~stroke)
```



#New dataset for the updated models

```
model_df <- data %>%
  filter(gender != "Other") %>%
  mutate(stroke = as.factor(stroke)) %>%
  filter(bmi != "N/A") %>%
  mutate(bmi = as.numeric(bmi)) %>%
  filter(age > 45) %>%
  filter(smoking_status != "Unknown") %>%
  select(-c(bmi, work_type))
```

#Random Forest modeling

```
library(randomForest)

## Warning: package 'randomForest' was built under R version 4.1.3

## randomForest 4.7-1

## 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

library(datasets)
library(caret)

## Warning: package 'caret' was built under R version 4.1.3

## Loading required package: lattice

##
## Attaching package: 'caret'

## The following object is masked from 'package:purrr':
##
##     lift

set.seed(11)
#creating a train and test set
stroke <- model_df %>%
  filter(stroke == '1')
dt <- sort(sample(nrow(stroke), nrow(stroke)*.8))
train <- stroke[dt,]
test <- stroke[-dt,]

no_stroke <- model_df %>%
  filter(stroke == '0') %>%
  sample_n(500)

dt2 <- sort(sample(nrow(no_stroke), nrow(no_stroke)*.8))
train2 <- no_stroke[dt2,]
test2 <- no_stroke[-dt2,]

training <- rbind(train, train2)
testing <- rbind(test, test2)


#training the model
rf <- randomForest(stroke~., data = training)

#Prediction and Confusion Matrix
p1 <- predict(rf, training)
confusionMatrix(p1,training$stroke)

## Confusion Matrix and Statistics
##
##           Reference
## Prediction   0   1
##          0 400  63
##          1   0  74
```

```
##
##                Accuracy : 0.8827
##                  95% CI : (0.8524, 0.9087)
##     No Information Rate : 0.7449
##     P-Value [Acc > NIR] : 1.778e-15
##
##                   Kappa : 0.6363
##
##  Mcnemar's Test P-Value : 5.662e-15
##
##             Sensitivity : 1.0000
##             Specificity : 0.5401
##          Pos Pred Value : 0.8639
##          Neg Pred Value : 1.0000
##              Prevalence : 0.7449
##          Detection Rate : 0.7449
##    Detection Prevalence : 0.8622
##       Balanced Accuracy : 0.7701
##
##        'Positive' Class : 0
##
```

```
p2 <- predict(rf,testing)
confusionMatrix(p2, testing$stroke)
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction  0  1
##          0 97 33
##          1  3  2
##
##                Accuracy : 0.7333
##                  95% CI : (0.6504, 0.8057)
##     No Information Rate : 0.7407
##     P-Value [Acc > NIR] : 0.6213
##
##                   Kappa : 0.0376
##
##  Mcnemar's Test P-Value : 1.343e-06
##
##             Sensitivity : 0.97000
##             Specificity : 0.05714
##          Pos Pred Value : 0.74615
##          Neg Pred Value : 0.40000
##              Prevalence : 0.74074
##          Detection Rate : 0.71852
##    Detection Prevalence : 0.96296
##       Balanced Accuracy : 0.51357
##
```

```
##        'Positive' Class : 0
##
```

#logistic Regression

```
library(fastDummies)

## Warning: package 'fastDummies' was built under R version 4.1.3

set.seed(11)
#creating a train and test set
stroke <- model_df %>%
  filter(stroke == '1')

stroke <- dummy_cols(stroke,select_columns = c('gender', 'ever_married',
'Residence_type'))
dt <- sort(sample(nrow(stroke), nrow(stroke)*.8))
train <- stroke[dt,]
test <- stroke[-dt,]

no_stroke <- model_df %>%
  filter(stroke == '0') %>%
  sample_n(500)
no_stroke <- dummy_cols(no_stroke,select_columns = c('gender',
'ever_married', 'Residence_type'))

dt2 <- sort(sample(nrow(no_stroke), nrow(no_stroke)*.8))
train2 <- no_stroke[dt2,]
test2 <- no_stroke[-dt2,]

training <- rbind(train, train2)
testing <- rbind(test, test2)

log <- glm(stroke~hypertension + heart_disease, data = training, family =
"binomial")
summary(log)

##
## Call:
## glm(formula = stroke ~ hypertension + heart_disease, family = "binomial",
##     data = training)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.1961  -0.6761  -0.6761   1.1588   1.7822
##
## Coefficients:
##               Estimate Std. Error z value Pr(>|z|)
## (Intercept)    -1.3596     0.1271 -10.697  < 2e-16 ***
## hypertension    0.6488     0.2205   2.942  0.00326 **
```

```
## heart_disease   0.7547     0.2718   2.777  0.00549 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 609.92  on 536  degrees of freedom
## Residual deviance: 592.80  on 534  degrees of freedom
## AIC: 598.8
##
## Number of Fisher Scoring iterations: 4
```

```
glm.probs <- predict(log, type = 'response')

glm.pred <- ifelse(glm.probs > .5, "1", "0")

table(glm.pred, training$stroke)
```

```
##
## glm.pred   0   1
##        0 387 127
##        1  13  10
```

```
mean(glm.pred == training$stroke)
```

```
## [1] 0.7392924
```

```
glm.probs2 = predict(log, newdata = testing, type = 'response')
glm.pred2 = ifelse(glm.probs2 > .5, "1", "0")
table(glm.pred2, testing$stroke)
```

```
##
## glm.pred2  0  1
##         0 97 34
##         1  3  1
```

```
mean(glm.pred2 == testing$stroke)
```

```
## [1] 0.7259259
```