Project Presentation

CPE 213 Data Models

Kittipol Neamprasetporn 62070503404

Thanasit Suwanposri 62070503414

Siriphorn Jarisu 62070503448

Healthcare dataset stroke data

```r
install.packages("tidybayes")
library(dplyr)
library(tidyr)
library(tidyverse)

Data ← read.csv("healthcare-dataset-stroke-data.csv", sep = ",")
Data_summary ← summary(Data)
```

- ID
- Gender
- Age
- Hypertension
- Heart disease
- Ever married
- Work type
- Residence type

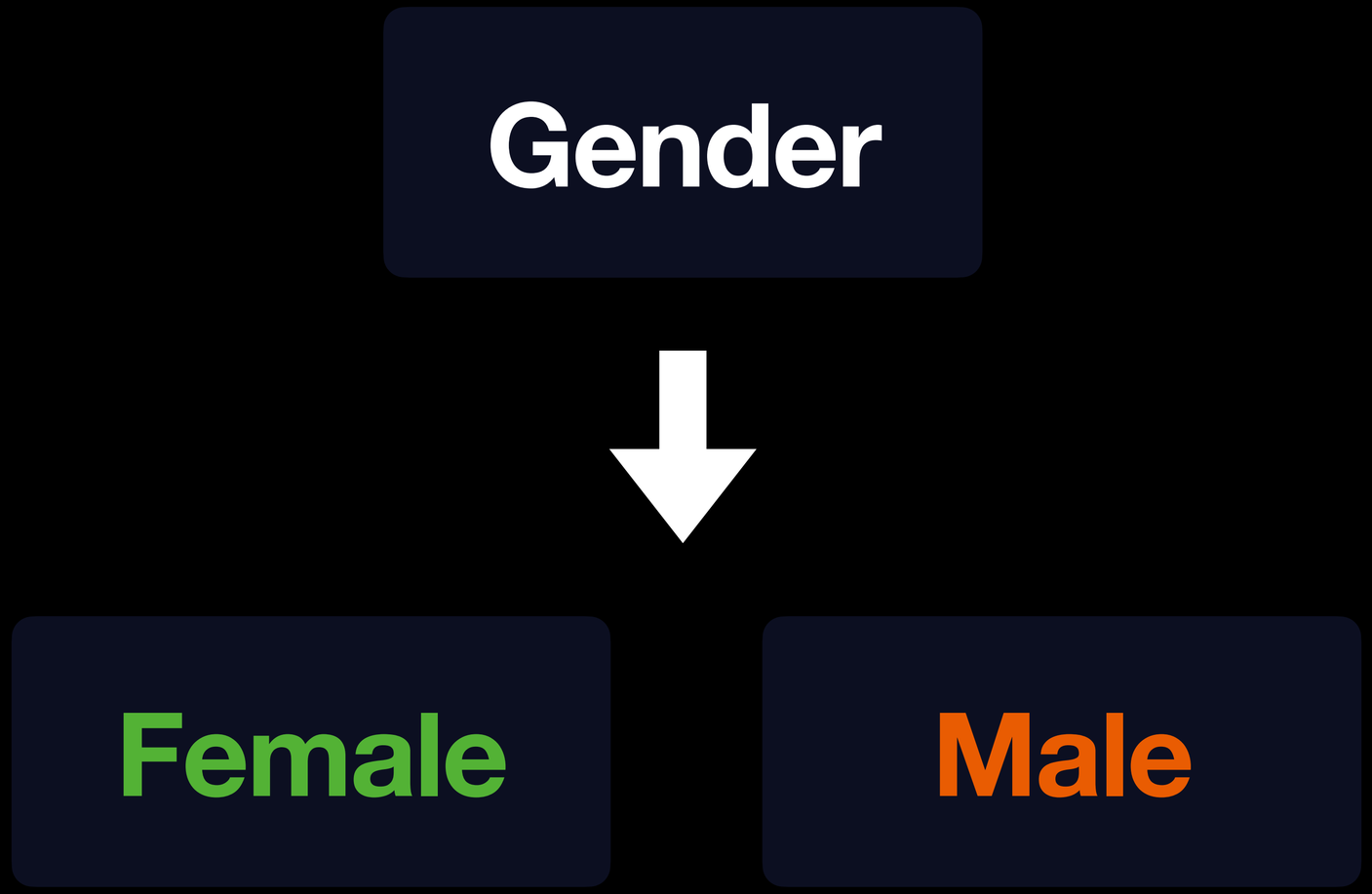
Healthcare dataset stroke data

Continued

```r
install.packages("tidybayes")
library(dplyr)
library(tidyr)
library(tidyverse)

Data ← read.csv("healthcare-dataset-stroke-data.csv", sep = ",")
Data_summary ← summary(Data)
```

- Average glucose level
- BMI
- Smoking status
- Stroke

Introduction to the problem



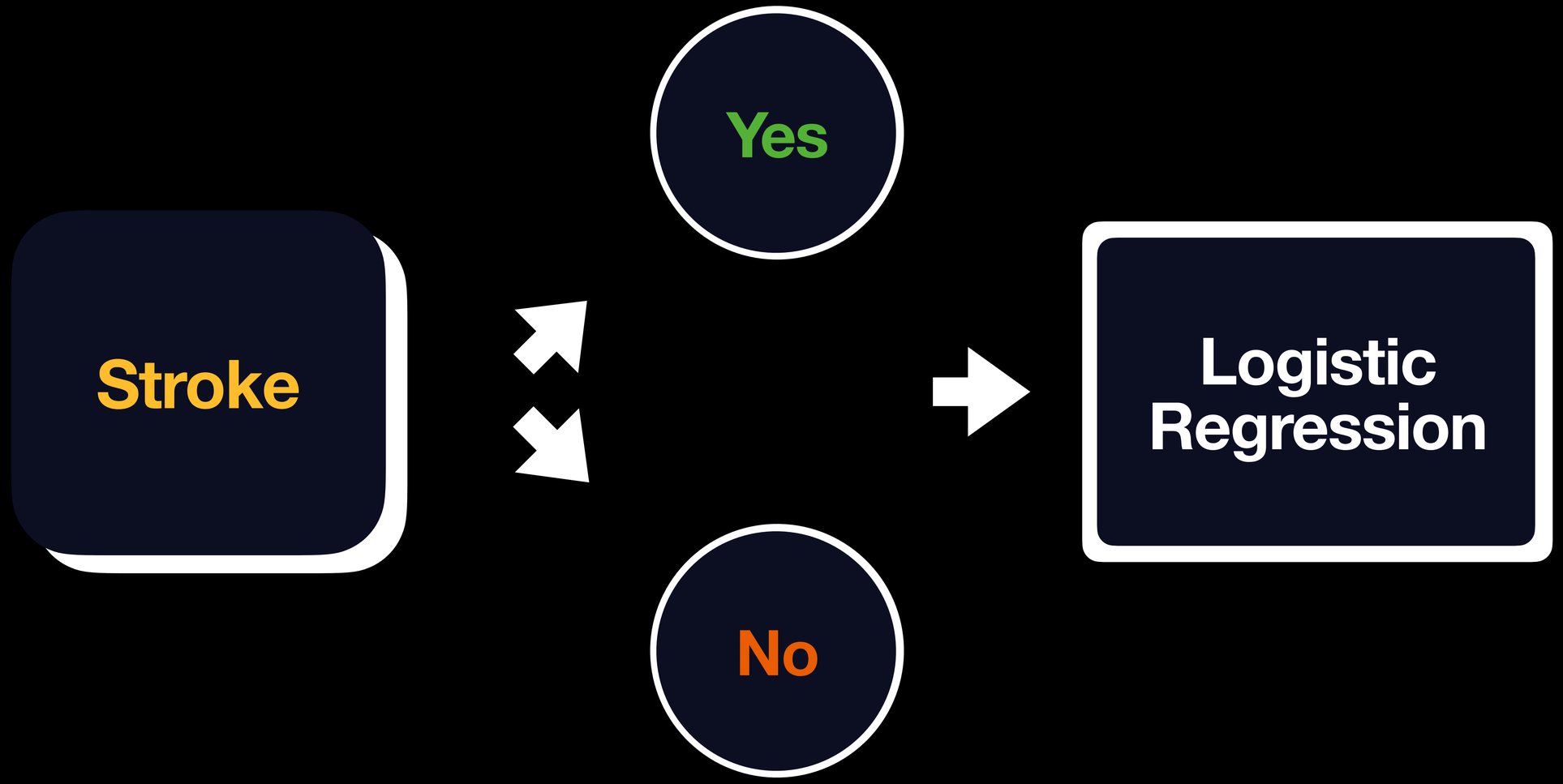
Introduction to the problem

Who have had a stroke between Female and Male ?

The factors

- Age
- Hypertension
- Heart disease
- Average glucose level

Analytic Object



Data description and preparation

Data Description

```r
install.packages("tidybayes")
library(dplyr)
library(tidyr)
library(tidyverse)

Data <- read.csv("healthcare-dataset-stroke-data.csv", sep = ",")
Data_summary <- summary(Data)
```

Data Description

```
       id            gender                age          hypertension     heart_disease     ever_married        work_type         Residence_type
 Min.   :   67   Length:5110        Min.   : 0.08   Min.   :0.00000   Min.   :0.00000   Length:5110        Length:5110        Length:5110
 1st Qu.:17741   Class :character   1st Qu.:25.00   1st Qu.:0.00000   1st Qu.:0.00000   Class :character   Class :character   Class :character
 Median :36932   Mode  :character   Median :45.00   Median :0.00000   Median :0.00000   Mode  :character   Mode  :character   Mode  :character
 Mean   :36518                      Mean   :43.23   Mean   :0.09746   Mean   :0.05401
 3rd Qu.:54682                      3rd Qu.:61.00   3rd Qu.:0.00000   3rd Qu.:0.00000
 Max.   :72940                      Max.   :82.00   Max.   :1.00000   Max.   :1.00000
 avg_glucose_level     bmi            smoking_status         stroke
 Min.   : 55.12    Length:5110        Length:5110        Min.   :0.00000
 1st Qu.: 77.25    Class :character   Class :character   1st Qu.:0.00000
 Median : 91.89    Mode  :character   Mode  :character   Median :0.00000
 Mean   :106.15                                          Mean   :0.04873
 3rd Qu.:114.09                                          3rd Qu.:0.00000
 Max.   :271.74                                          Max.   :1.00000
```

Data Description

```r
install.packages("tidybayes")
library(dplyr)
library(tidyr)
library(tidyverse)

Data ← read.csv("healthcare-dataset-stroke-data.csv", sep = ",")
Data_head ← head(Data)
Data_head
```

	id	gender	age	hypertension	heart_disease	ever_married	work_type	Residence_type	avg_glucose_level	bmi	smoking_status	stroke
1	9046	Male	67	0	1	Yes	Private	Urban	228.69	36.6	formerly smoked	1
2	51676	Female	61	0	0	Yes	Self-employed	Rural	202.21	N/A	never smoked	1
3	31112	Male	80	0	1	Yes	Private	Rural	105.92	32.5	never smoked	1
4	60182	Female	49	0	0	Yes	Private	Urban	171.23	34.4	smokes	1
5	1665	Female	79	1	0	Yes	Self-employed	Rural	174.12	24	never smoked	1
6	56669	Male	81	0	0	Yes	Private	Urban	186.21	29	formerly smoked	1

Data Preparation

Coding

```r
Data ← read.csv("healthcare-dataset-stroke-data.csv", sep = ",")

# Check the unique values for Categorical Variable
unique(Data$gender)
unique(Data$ever_married)
unique(Data$work_type)
unique(Data$Residence_type)
unique(Data$smoking_status)
```

Result

‘Male’, ‘Female’, ‘Other’

‘Yes’, ‘No’

‘Private’, ‘Self-employed’, ‘Govt_job’, ‘children’, ‘Never_worked’

‘Urban’, ‘Rural’

‘formerly smoked’, ‘never smoked’, ‘smokes’, ‘Unknown’

Data Preparation

Coding

```r
Data ← read.csv("healthcare-dataset-stroke-data.csv", sep = ",")

# Check NA Values
colSums(is.na(Data))
```

Result

ID : 0	Heart disease : 0	Average glucose level : 0
Gender : 0	Ever married : 0	BMI : 0
Age : 0	Work type : 0	Smoking status : 0
Hypertension : 0	Residence type : 0	Stroke : 0

Data Preparation

```r
Data ← read.csv("healthcare-dataset-stroke-data.csv", sep = ",")

# Preparation
# remove Gender: 'Other', bmi: 'N/A'
# Change Categorical Variables → Factors
Data %>%
  drop_na() %>%
  filter(gender ≠ "Other", bmi ≠ "N/A") %>%
  mutate(hypertension = factor(ifelse(hypertension == 1, "yes", "no")),
         heart_disease = factor(ifelse(heart_disease == 1, "yes",
"no")),
         stroke = factor(ifelse(stroke == 1, "yes", "no")),
         bmi = as.double(bmi)) %>%
  mutate_if(is.character, as.factor) → stroke_mod
```

```
> summary(stroke_mod)
       id            gender          age        hypertension heart_disease ever_married
 Min.   :   77   Female:2897   Min.   : 0.08   no :4457     no :4665      No :1704
 1st Qu.:18602   Male  :2011   1st Qu.:25.00   yes: 451     yes: 243      Yes:3204
 Median :37580                 Median :44.00
 Mean   :37060                 Mean   :42.87
 3rd Qu.:55182                 3rd Qu.:60.00
 Max.   :72940                 Max.   :82.00
         work_type    Residence_type avg_glucose_level      bmi                smoking_status stroke
 children     : 671   Rural:2418     Min.   : 55.12    Min.   :10.30   formerly smoked: 836   no :4699
 Govt_job     : 630   Urban:2490     1st Qu.: 77.07    1st Qu.:23.50   never smoked   :1852   yes: 209
 Never_worked :  22                  Median : 91.68    Median :28.10   smokes         : 737
 Private      :2810                  Mean   :105.30    Mean   :28.89   Unknown        :1483
 Self-employed: 775                  3rd Qu.:113.50    3rd Qu.:33.10
                                     Max.   :271.74    Max.   :97.60
```

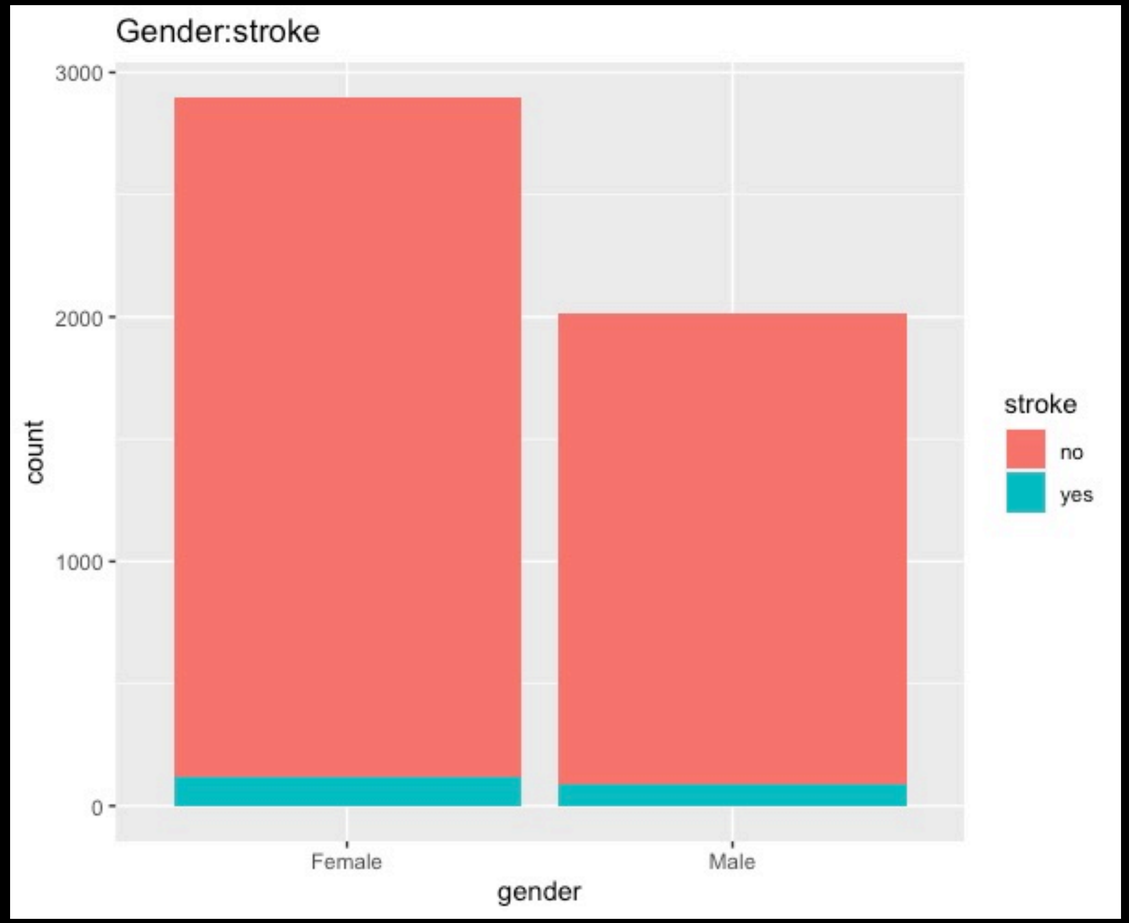
Entries: 4,908 records
Got stroke: 209 records
Not get stroke: 4,699 records

Data exploration and visualization

Data exploration and visualization

Let’s see the number of people who have Stroke for each factor

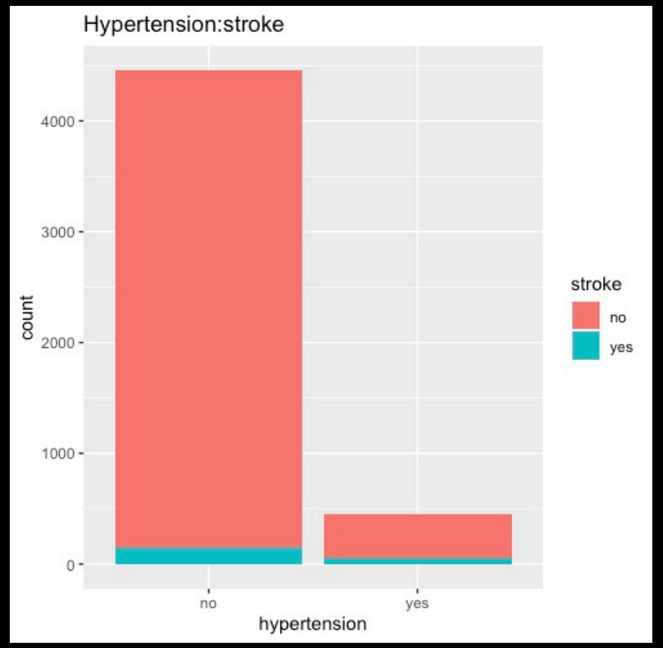
```r
# Data exploration
 ggplot(stroke_mod, aes(gender,fill = stroke)) +
 geom_bar() +
 labs(title = "Gender:stroke")
```



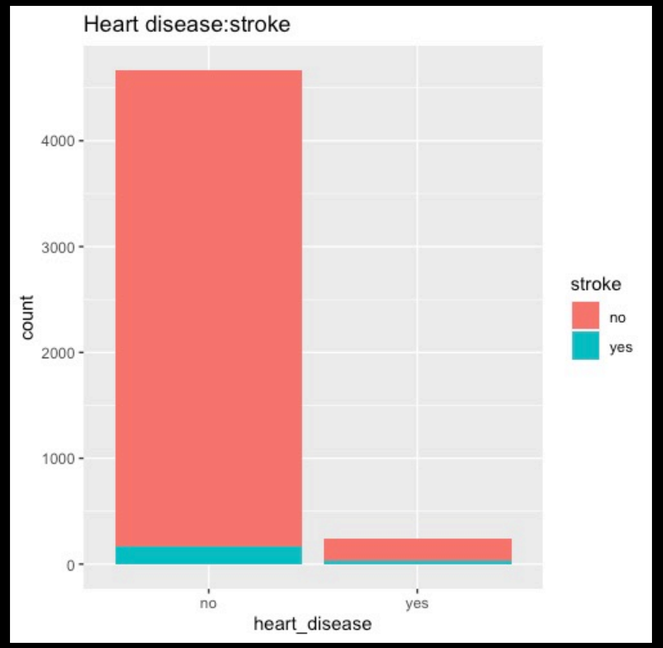
Factor : Gender

Data exploration and visualization

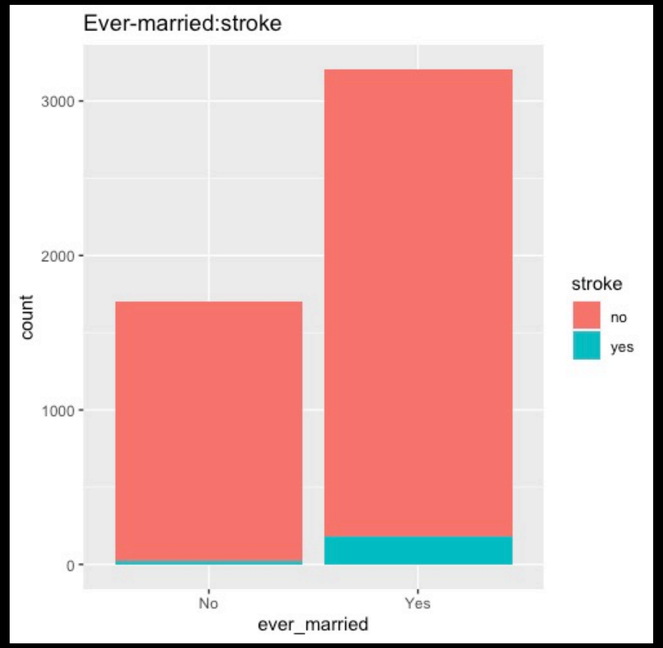
Let’s see the number of people who have Stroke for each factor



Factor : Hypertension



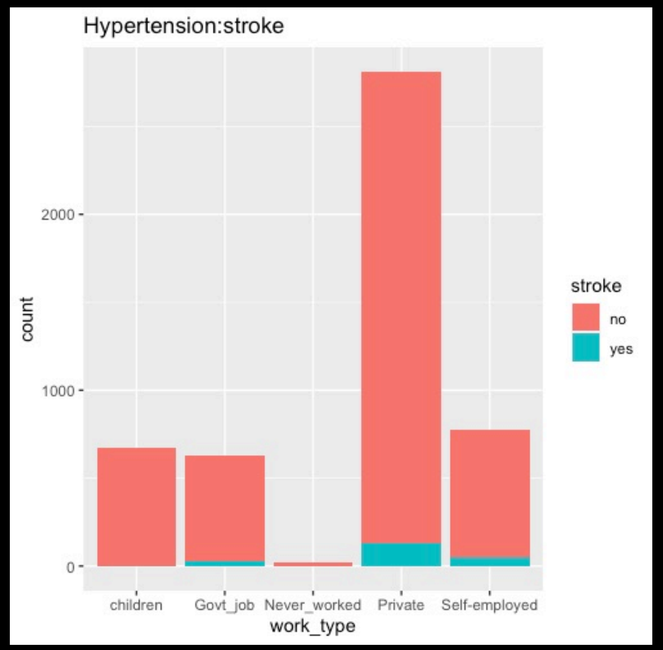
Factor : Heart Disease



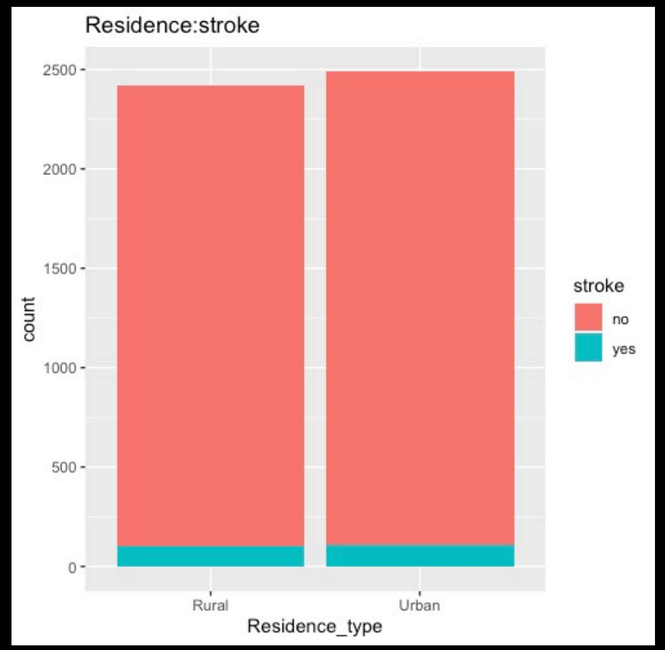
Factor : Ever married

Data exploration and visualization

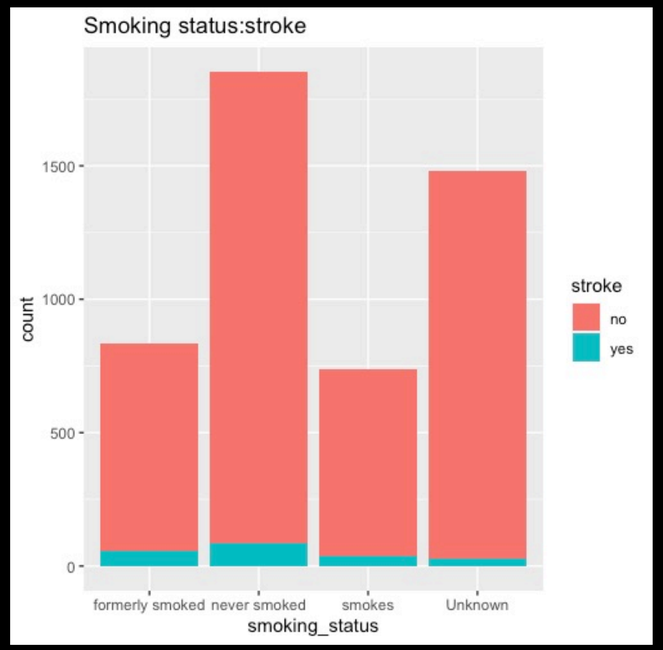
Let’s see the number of people who have Stroke for each factor



Factor : Work type



Factor : Residence type



Factor : Smoking Status

Probability

Data exploration and visualization

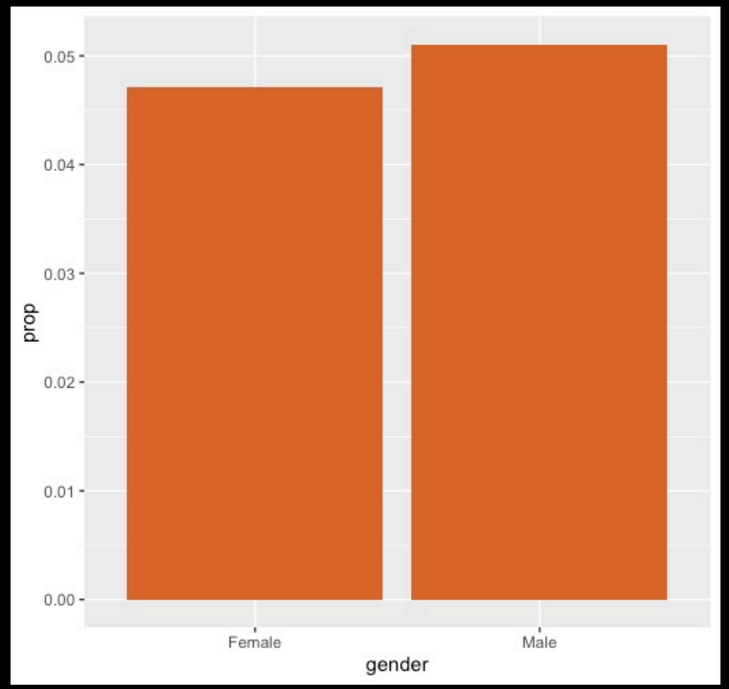
Data exploration and visualization

Let’s see the probability of people who have Stroke for each factor

```r
# Visualization
Data_Prop <- Data %>%
  group_by(gender) %>%
  summarise(prop = sum(stroke == "1")/length(gender))

# Ploting
df1 <- Data_Prop %>%
  ggplot(aes(x = gender,
             y = prop)) +
  geom_col(fill = "#dc7073")
df1
```

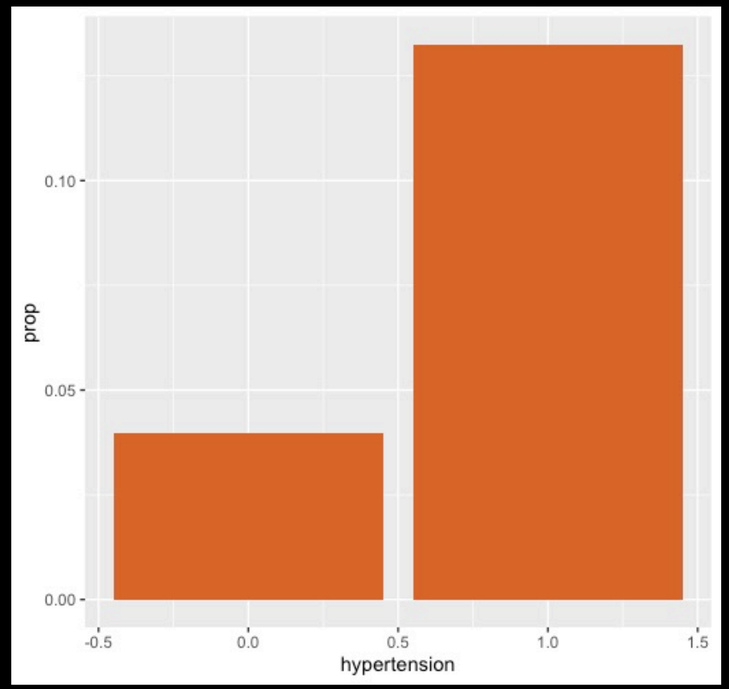
Factor : Gender

Let's see the probability of people who have Stroke for each factor



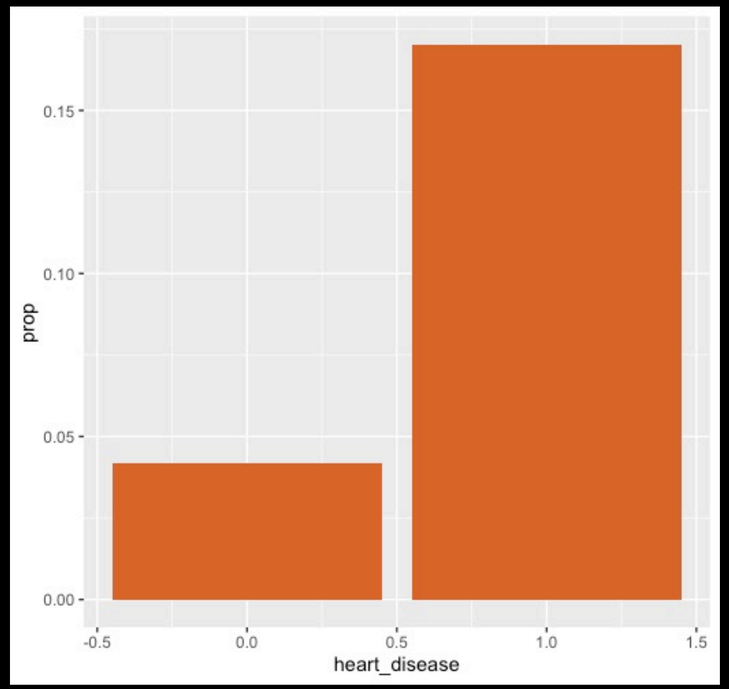
Female : 0.047 Male : 0.051

Factor : Gender



0 (No) : 0.039 1 (Yes) : 0.132

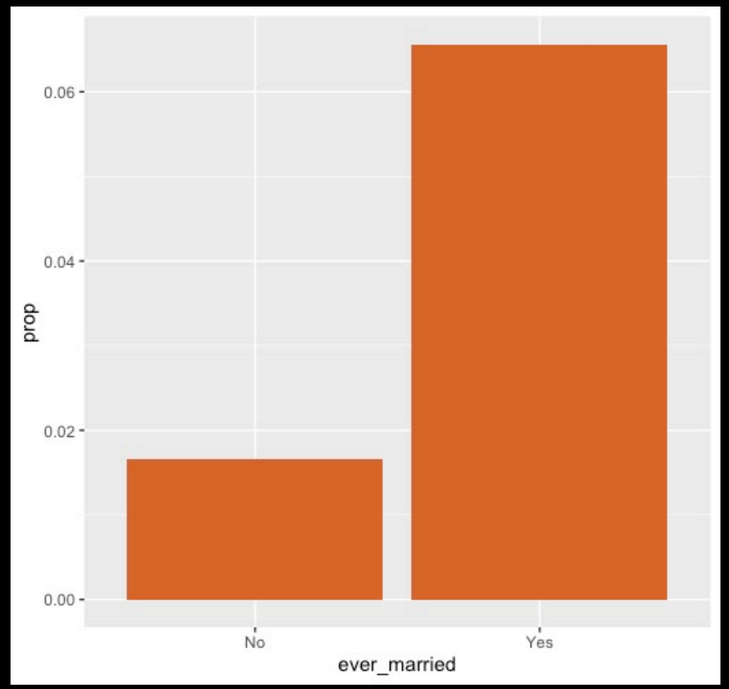
Factor : Hypertension



0 (No) : 0.042 1 (Yes) : 0.170

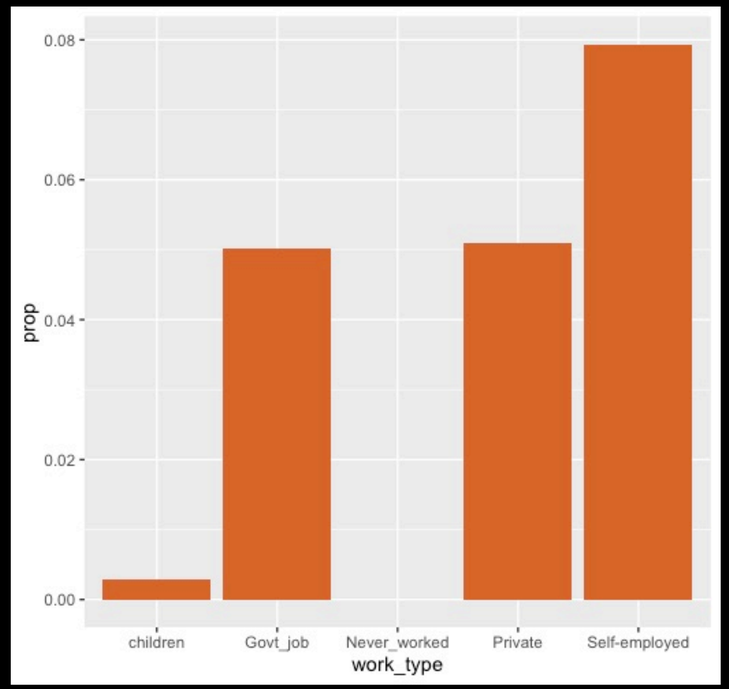
Factor : Heart Disease

Let's see the probability of people who have Stroke for each factor



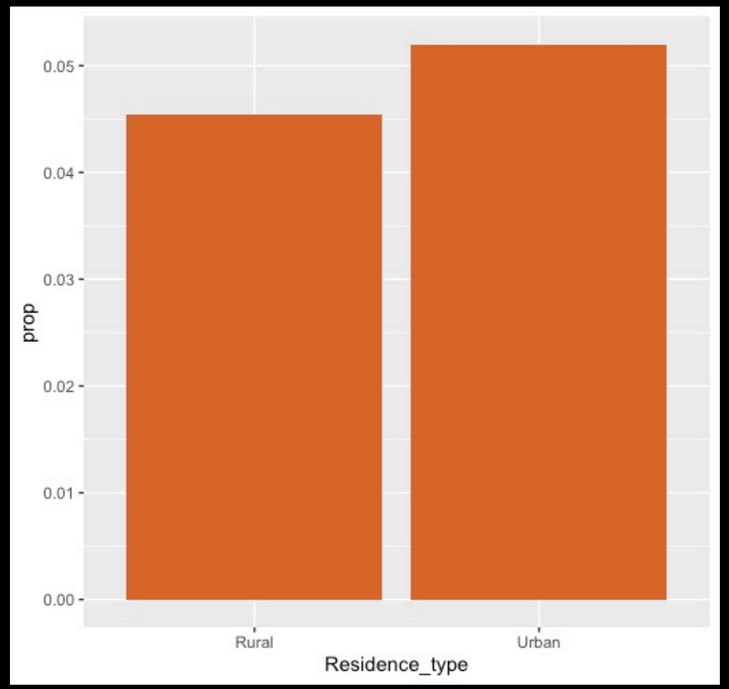
No : 0.017 Yes : 0.066

Factor : Ever married



children : 0.0029,
Govt_job: 0.050,
Never_worked : 0.0,
Private : 0.050 ,
Self-employed : 0.079

Factor : Work type



Rural : 0.045 Urban : 0.052

Factor : Residence type

Data exploration and visualization

Let’s consider Female Versus Male

Data exploration and visualization

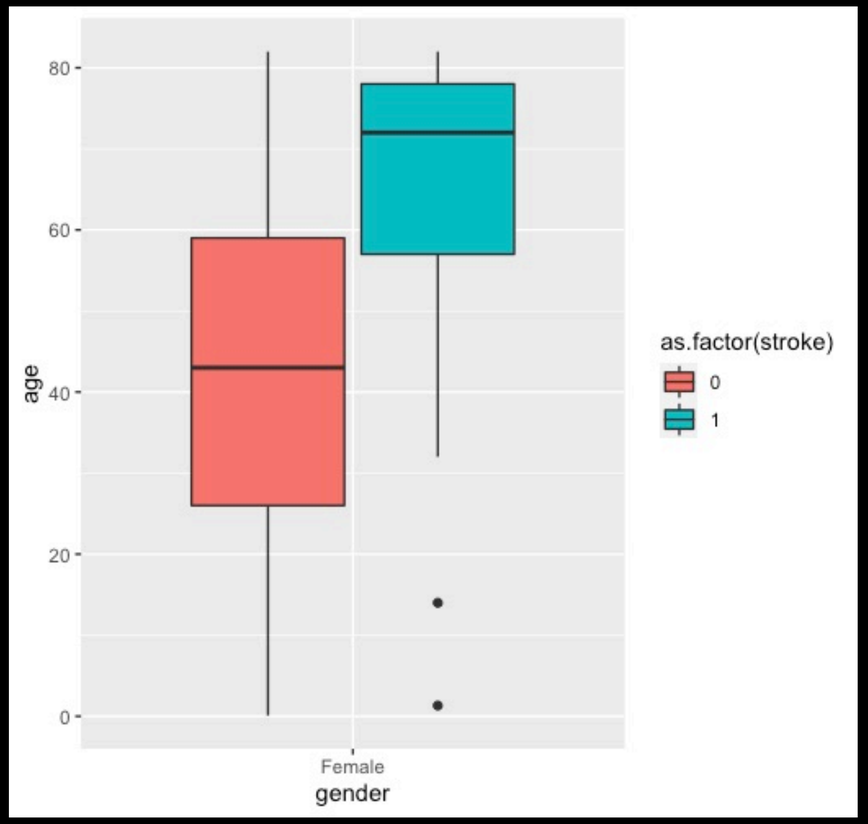
Let’s consider Female Versus Male

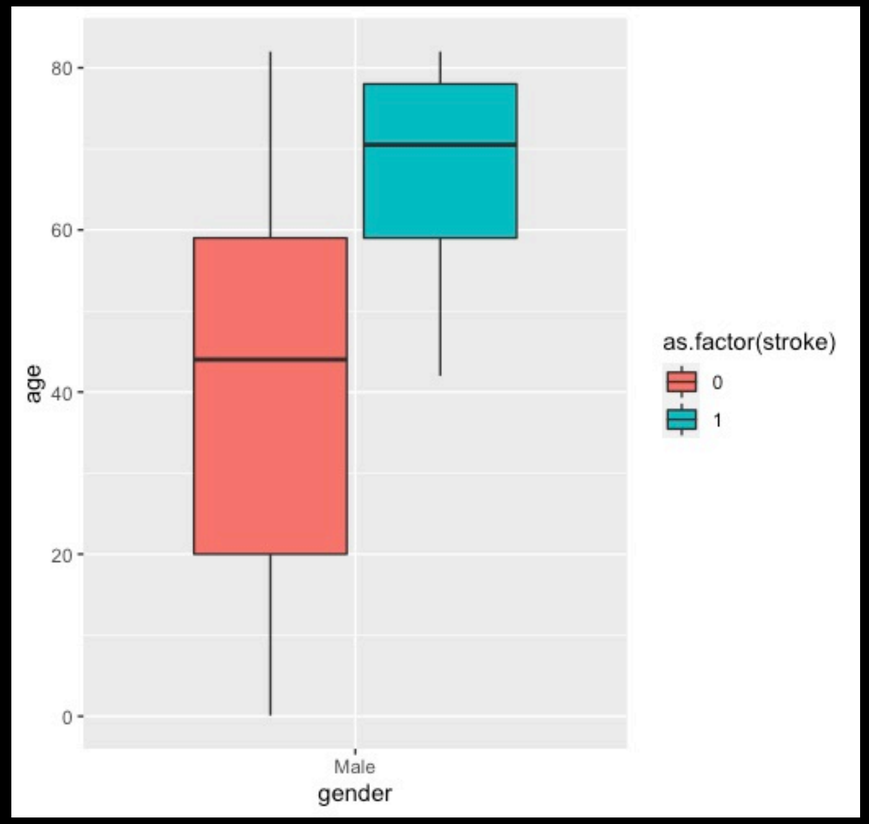
```r
# Separate by Gender
stroke_mod %>%
filter(gender == "Male") -> stroke_male

stroke_mod %>%
filter(gender == "Female") -> stroke_female

# Ploting
ggplot(stroke_male, aes(x = gender, y = age, fill = stroke)) +
geom_boxplot()

ggplot(stroke_female, aes(x = gender, y = age, fill = stroke)) +
geom_boxplot()

```

Factor : Age

Data exploration and visualization

Let’s consider Female Versus Male





Factor : Age

Data exploration and visualization

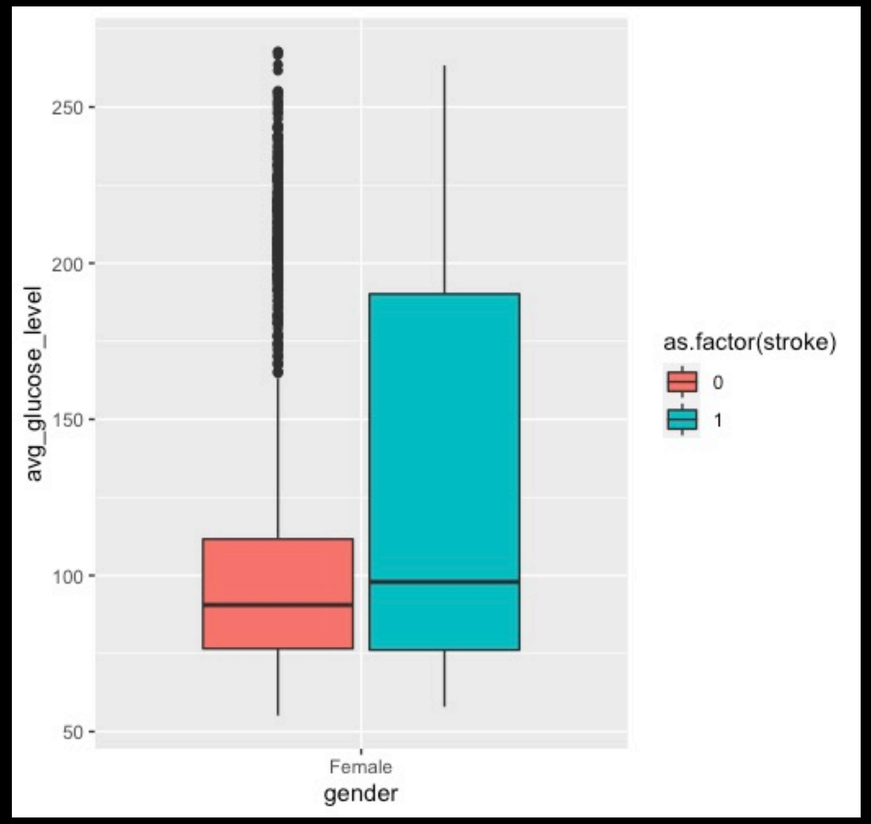
Let’s consider Female Versus Male

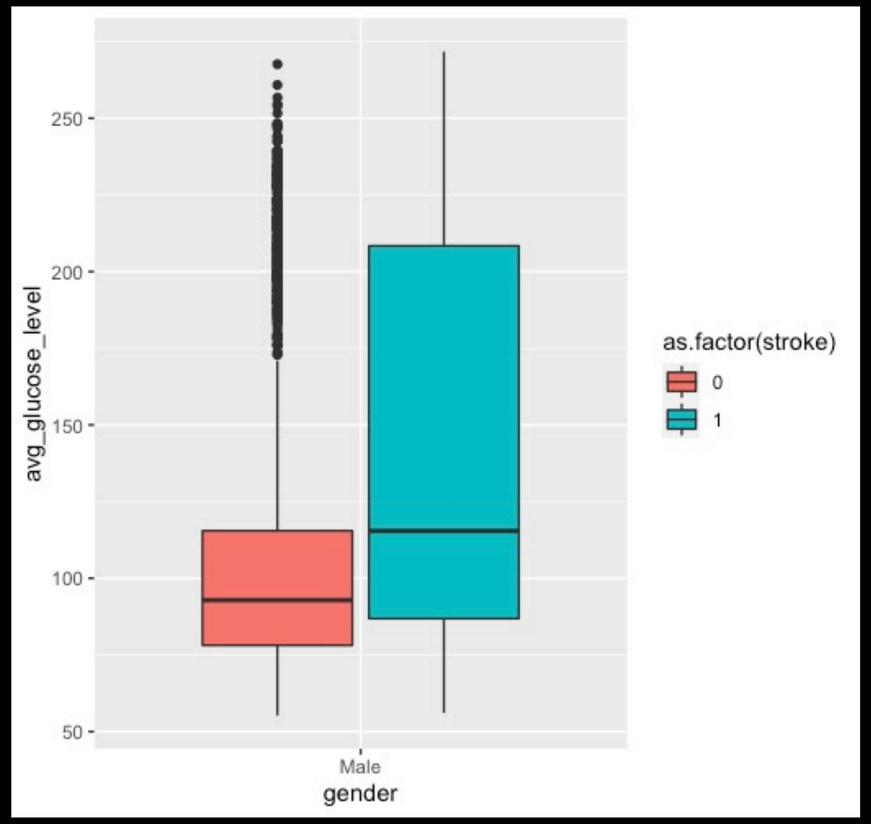
```r
# Separate by Gender
 stroke_mod %>%
 filter(gender == "Male") -> stroke_male

 stroke_mod %>%
 filter(gender == "Female") -> stroke_female

# Ploting
 ggplot(stroke_male, aes(x = gender,
                         y = avg_glucose_level,
                         fill = stroke)) +
 geom_boxplot()

 ggplot(stroke_female, aes(x = gender,
                           y = avg_glucose_level,
                           fill = stroke)) +
 geom_boxplot()

```

Factor : Average glucose level

Data exploration and visualization

Let’s consider Female Versus Male





Factor : Average glucose level

Data exploration and visualization

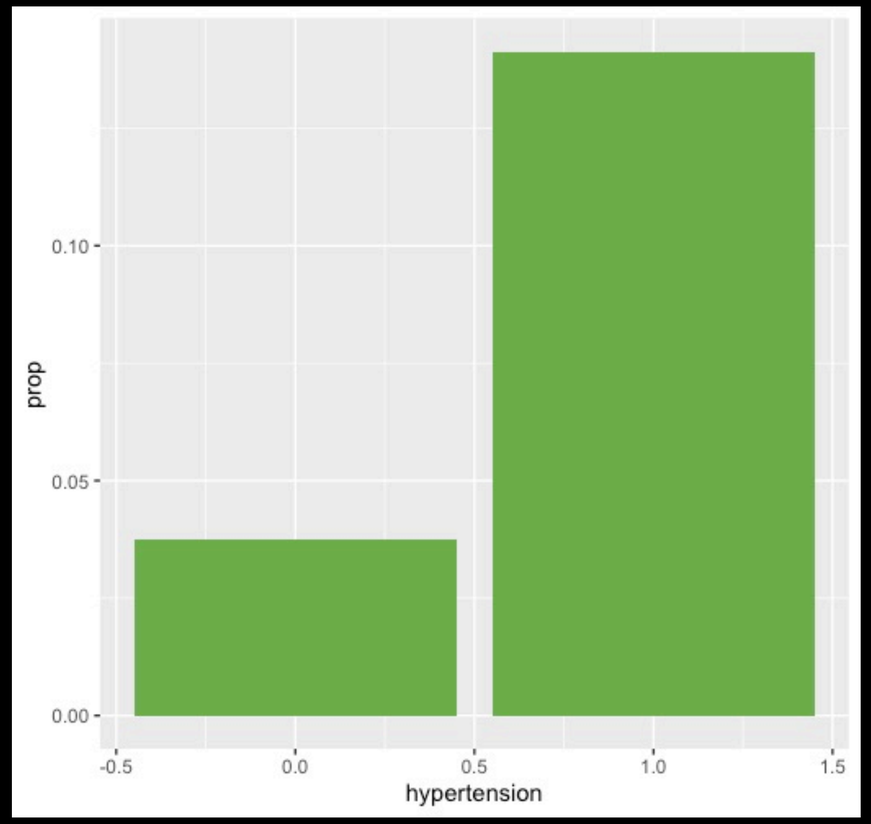
Let’s consider Female Versus Male

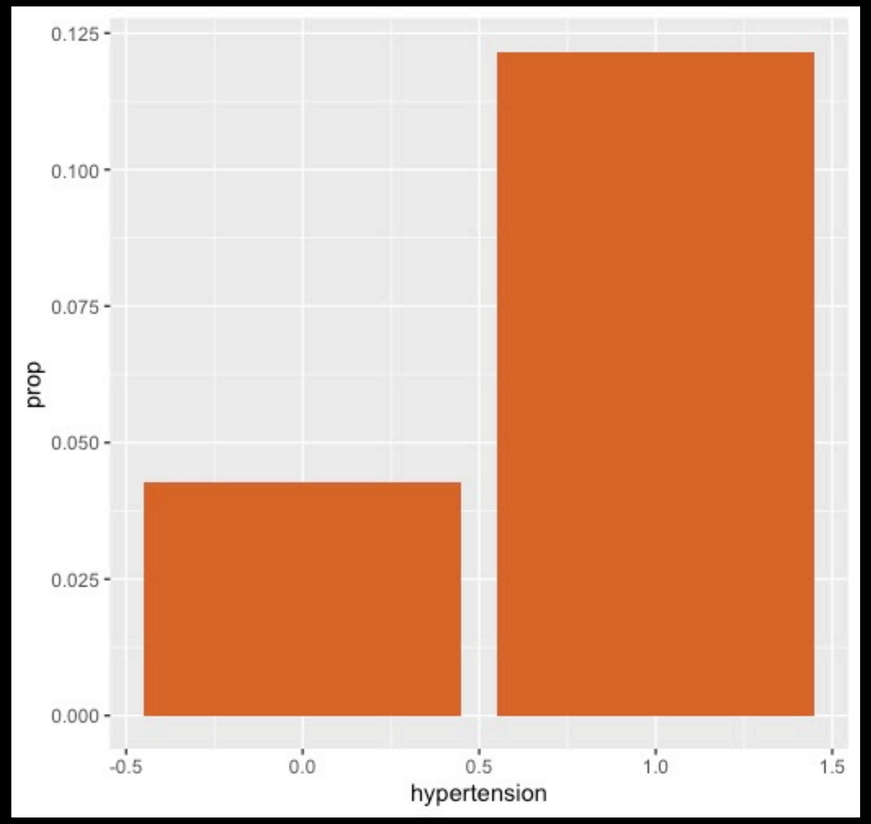
```r
# Separate by Gender
 stroke_mod %>%
 filter(gender == "Male") -> stroke_male

 stroke_mod %>%
 filter(gender == "Female") -> stroke_female

# Ploting
 ggplot(stroke_male, aes(x = gender,
                         y = hypertension,
                         fill = stroke)) +
 geom_bar()

 ggplot(stroke_female, aes(x = gender,
                           y = hypertension,
                           fill = stroke)) +
 geom_bar()

```

Factor : Hypertension

Data exploration and visualization

Let's consider term of Probability





Probability to have Stroke Female : 0.1216 Male : 0.1216 (By Hypertension)

Data exploration and visualization

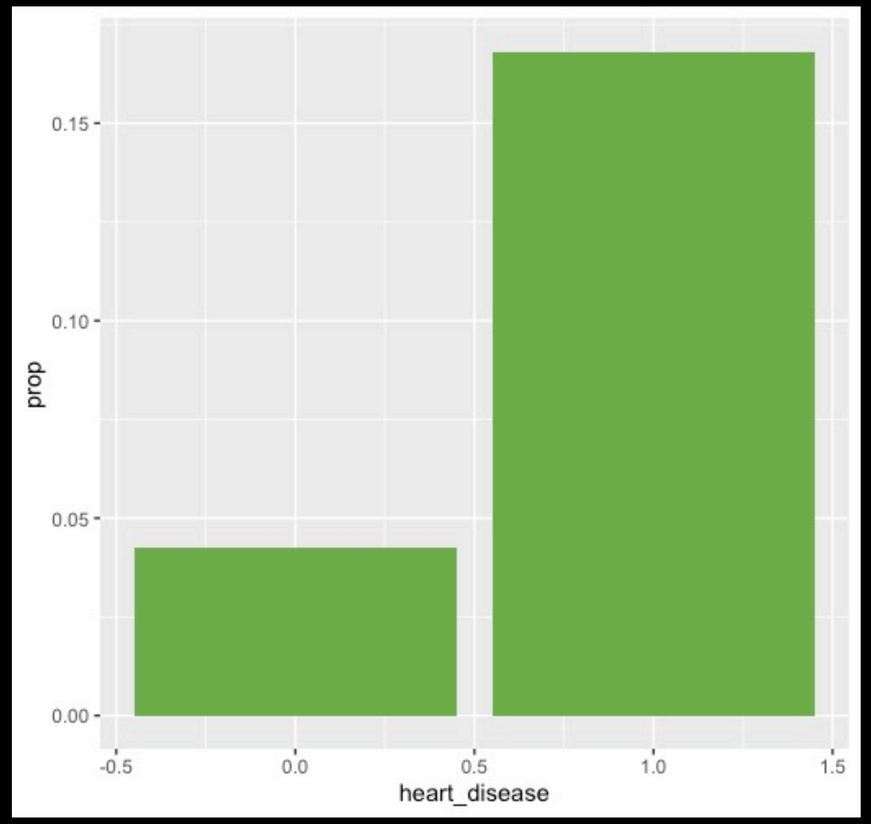
Let’s consider Female Versus Male

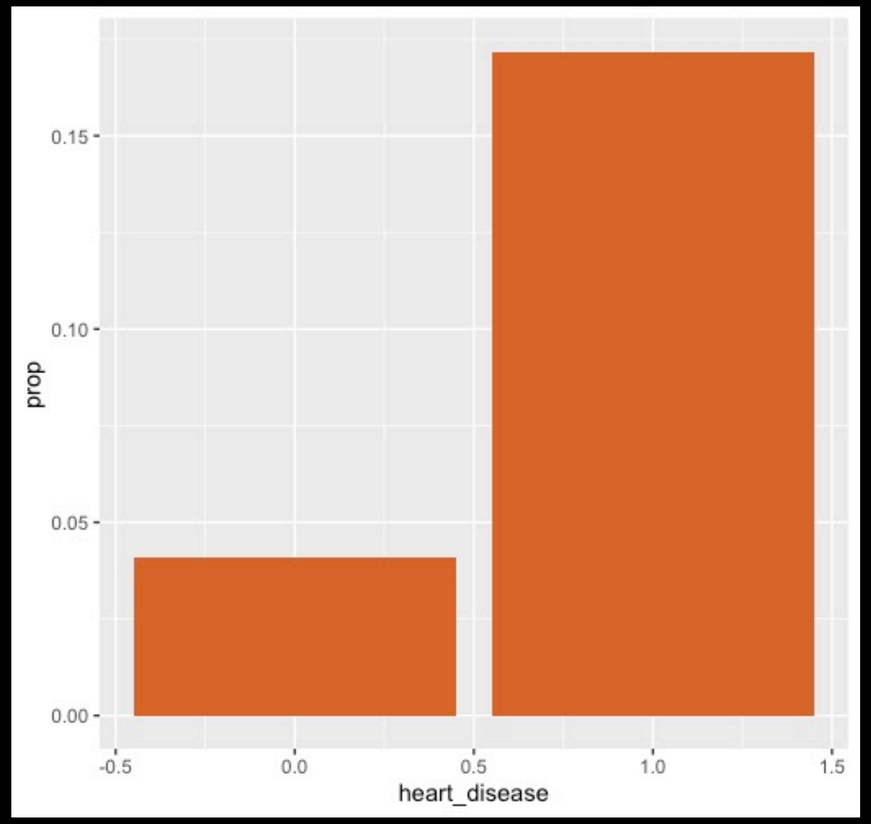
```r
# Separate by Gender
 stroke_mod %>%
 filter(gender == "Male") -> stroke_male

 stroke_mod %>%
 filter(gender == "Female") -> stroke_female

# Ploting
 ggplot(stroke_male, aes(x = gender,
                         y = heart_disease,
                         fill = stroke)) +
 geom_bar()

 ggplot(stroke_female, aes(x = gender,
                           y = heart_disease,
                           fill = stroke)) +
 geom_bar()

```

Factor : Heart Disease

Data exploration and visualization

Let's consider term of Probability



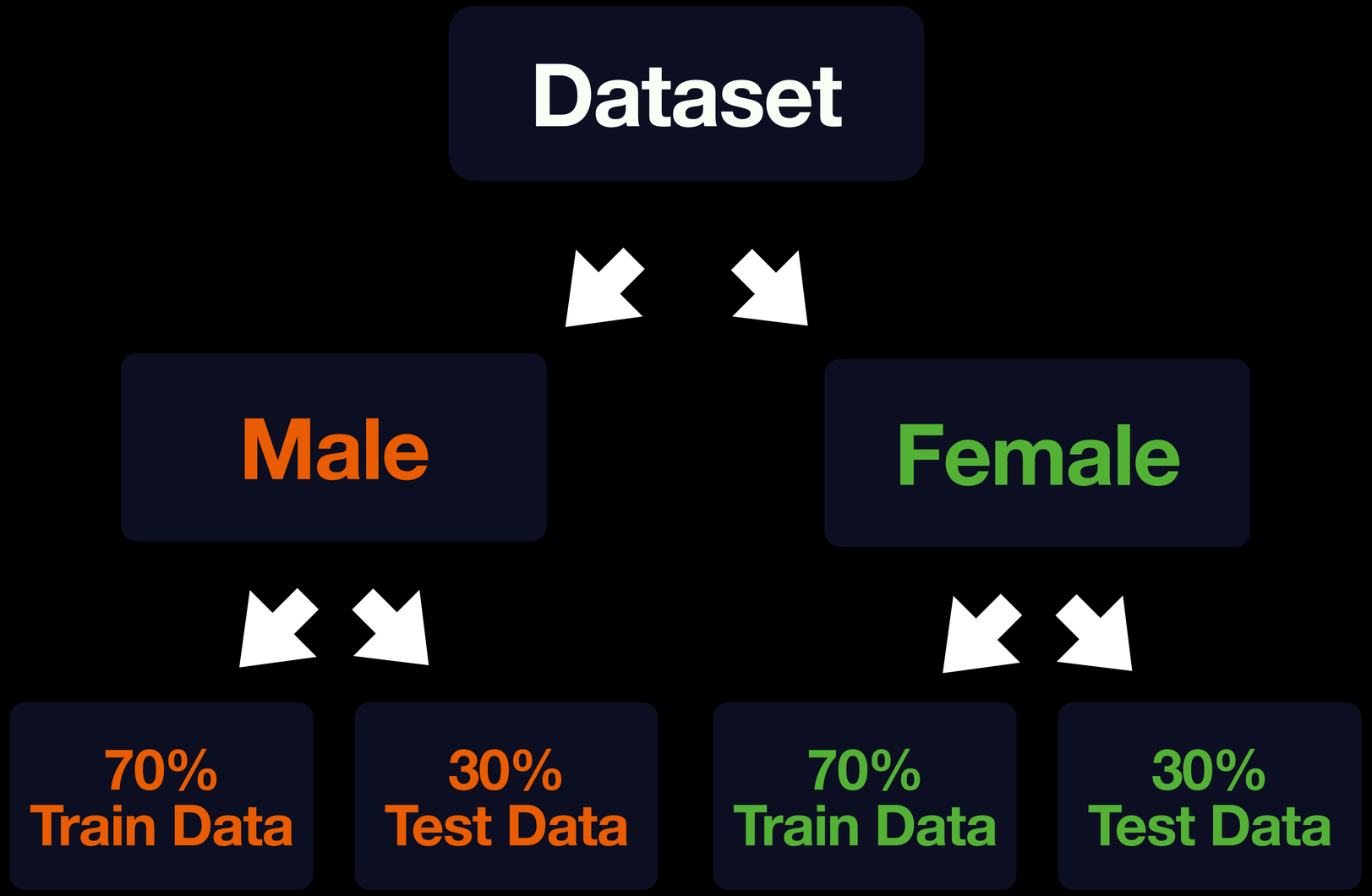


Probability to have Stroke : Female : 0.168 Male : 0.172 (By Heart Disease)

Data Modeling

Model Explanation

Model Explanation



Model implementation (Male)

```r
# Male
summary(stroke_male$stroke)
# By this, we can tell that the probability
# that male will get stroke is 0.044

# Lets separate the training and testing set.
# We will train by using 70% of all rows from stroke_male
set.seed(5000)
test_ind_male ← sample(nrow(stroke_male), 0.3*nrow(stroke_male))

stroke_male_train ← stroke_male[-test_ind_male,]
stroke_male_test ← stroke_male[test_ind_male,]

model_male ← glm(stroke ~ age * hypertension * heart_disease *
                   avg_glucose_level,
                   data = stroke_male_train, family = binomial)

res_male ← predict(model_male, stroke_male_test,
                     type = "response")
summary(res_male)

# Factor that if male have a chance to get stroke more than 30% or not,
# depends on predictors
res_male_c ← factor(ifelse(res_male ⩾ 0.3, "yes", "no"))
summary(res_male_c)
```

No	Yes
1922	89

Min	1st Qu.	Median	Mean	3rd Qu.	Max
0.0000447	0.0022303	0.0133585	0.0418238	0.0464734	0.5370234

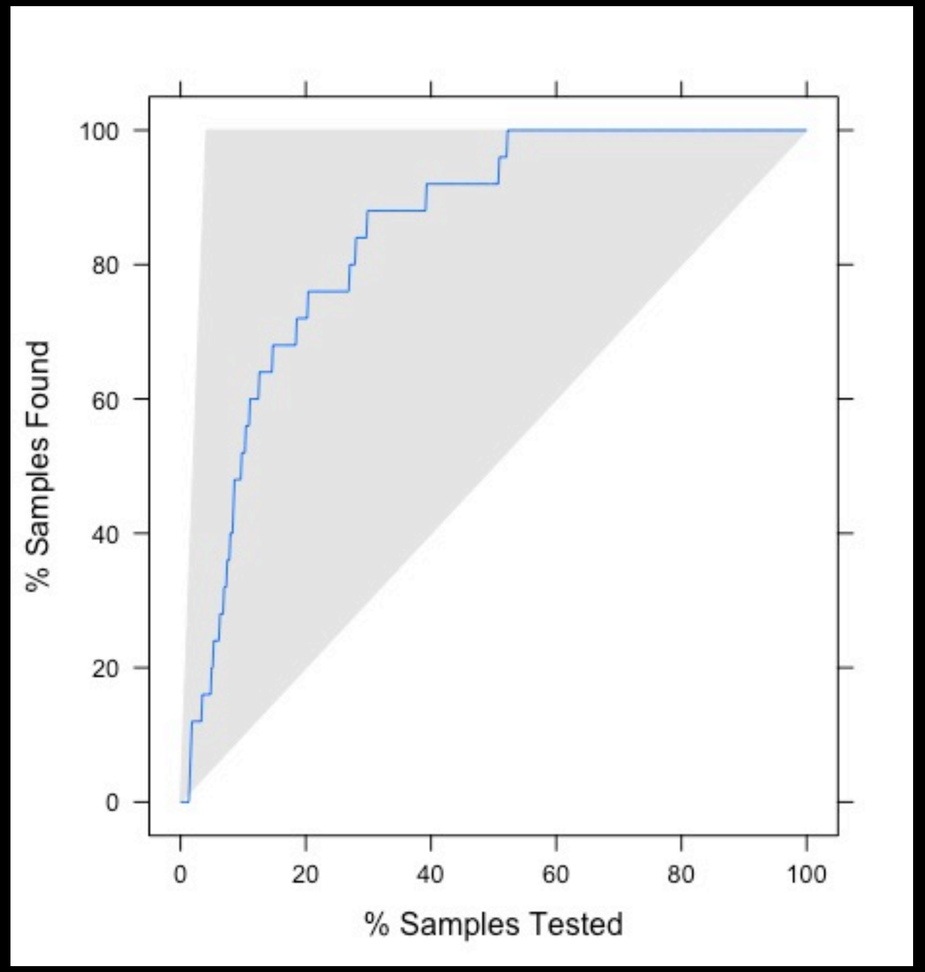
No	Yes
594	9

Evaluation code

```r
# Evaluation
# Male
confusionMatrix(res_male_c, stroke_male_test$stroke,
                mode = 'prec_recall',
                positive = "yes")

lift_male ← data.frame(prob = res_male,
                       stroke = stroke_male_test$stroke)

lift_obj_male ← lift(stroke ~ prob,
                     data = lift_male,
                     class = 'yes')
plot(lift_obj_male)
```

Evaluation result (Male)

```
Confusion Matrix and Statistics

          Reference
Prediction  no yes
       no  570  24
       yes   8   1

               Accuracy : 0.9469
                 95% CI : (0.9259, 0.9634)
    No Information Rate : 0.9585
    P-Value [Acc > NIR] : 0.93275

                  Kappa : 0.0377

 Mcnemar's Test P-Value : 0.00801

              Precision : 0.111111
                 Recall : 0.040000
                     F1 : 0.058824
             Prevalence : 0.041459
         Detection Rate : 0.001658
   Detection Prevalence : 0.014925
      Balanced Accuracy : 0.513080

       'Positive' Class : yes
```



Model implementation (Female)

```r
# Female
summary(stroke_female$stroke)
# By this, we can tell that the probability
# that female will get stoke is 0.041
# Lets separate the training and testing set.
# We will train by using 70% of all rows from stroke_female
set.seed(5000)
test_ind_female <- sample(nrow(stroke_female), 0.3*nrow(stroke_female))

stroke_female_train <- stroke_female[-test_ind_female,]
stroke_female_test <- stroke_female[test_ind_female,]

model_female <- glm(stroke ~ age * hypertension * heart_disease *
                    avg_glucose_level,
                    data = stroke_female_train, family = binomial)

res_female <- predict(model_female, stroke_female_test,
                      type = 'response')
summary(res_female)

# Factor that if female have a chance to get stroke more than 30% or
not,
#  depends on predictors
res_female_c <- factor(ifelse(res_female >= 0.3, "yes", "no"))
summary(res_female_c)
```

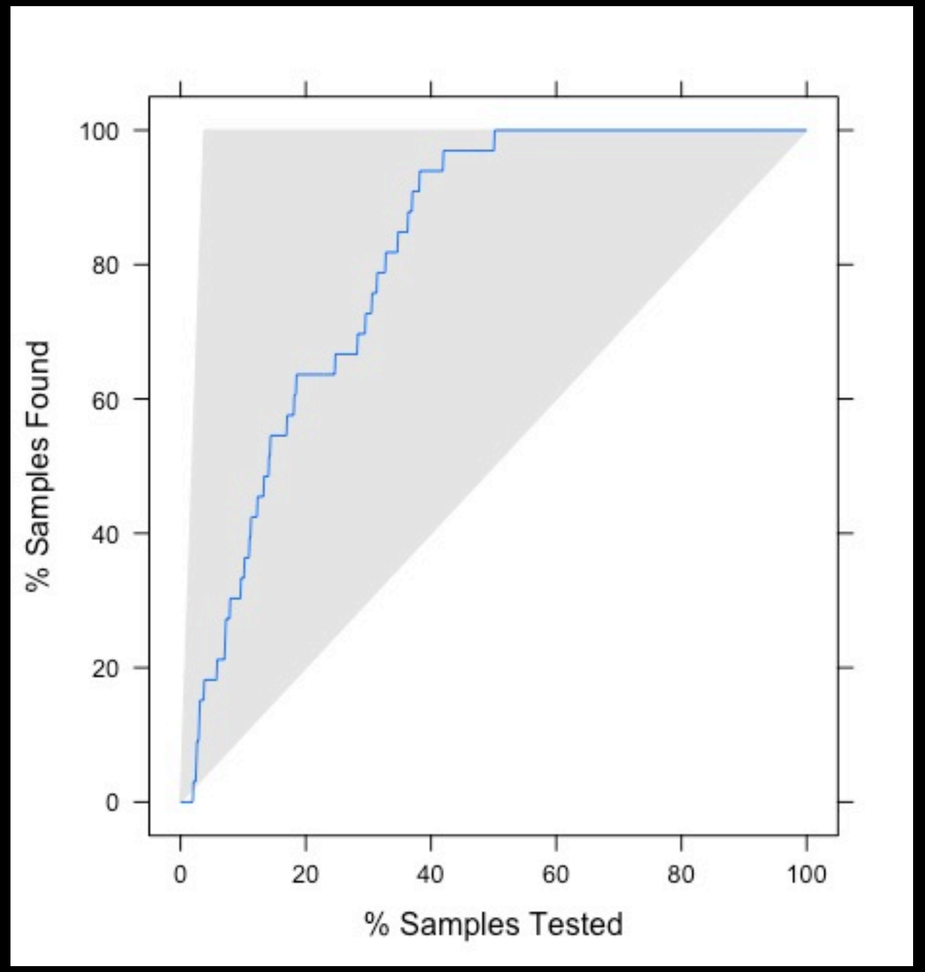
No	Yes
2777	120

Min	1st Qu.	Median	Mean	3rd Qu.	Max
0.0006781	0.0047661	0.0155278	0.0427028	0.0520295	0.4717323

No	Yes
860	9

Evaluation result (Female)

```
Confusion Matrix and Statistics

          Reference
Prediction  no yes
       no  827  33
       yes   9   0

               Accuracy : 0.9517
                 95% CI : (0.9352, 0.9649)
    No Information Rate : 0.962
    P-Value [Acc > NIR] : 0.9497708

                  Kappa : -0.0165

 Mcnemar's Test P-Value : 0.0003867

              Precision : 0.00000
                 Recall : 0.00000
                     F1 :     NaN
             Prevalence : 0.03797
         Detection Rate : 0.00000
   Detection Prevalence : 0.01036
      Balanced Accuracy : 0.49462

       'Positive' Class : yes
```



Model implementation (Overall)

```r
# Overall
summary(stroke_mod$stroke)
#0.043

set.seed(20000)
test_ind ← sample(nrow(stroke_mod), 0.3*nrow(stroke_mod))

stroke_train ← stroke_mod[-test_ind,]
stroke_test ← stroke_mod[test_ind,]

model ← glm(stroke ~ age * hypertension * heart_disease *
  avg_glucose_level, data = stroke_train, family = binomial)

res ← predict(model, stroke_test, type = 'response')
summary(res)

res_c ← factor(ifelse(res ⩾ 0.3, "yes", "no"))
summary(res_c)
```

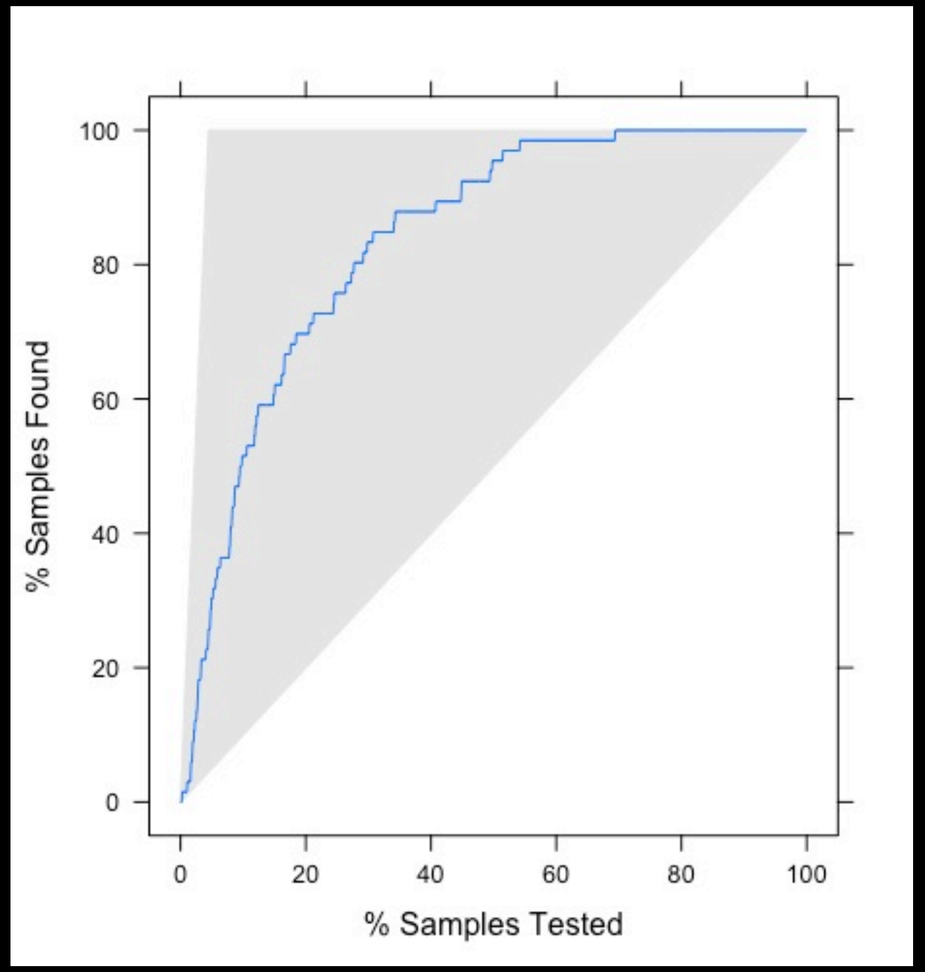
No	Yes
4699	209

Min	1st Qu.	Median	Mean	3rd Qu.	Max
0.0005477	0.0037863	0.0150500	0.0412239	0.0510362	0.4843260

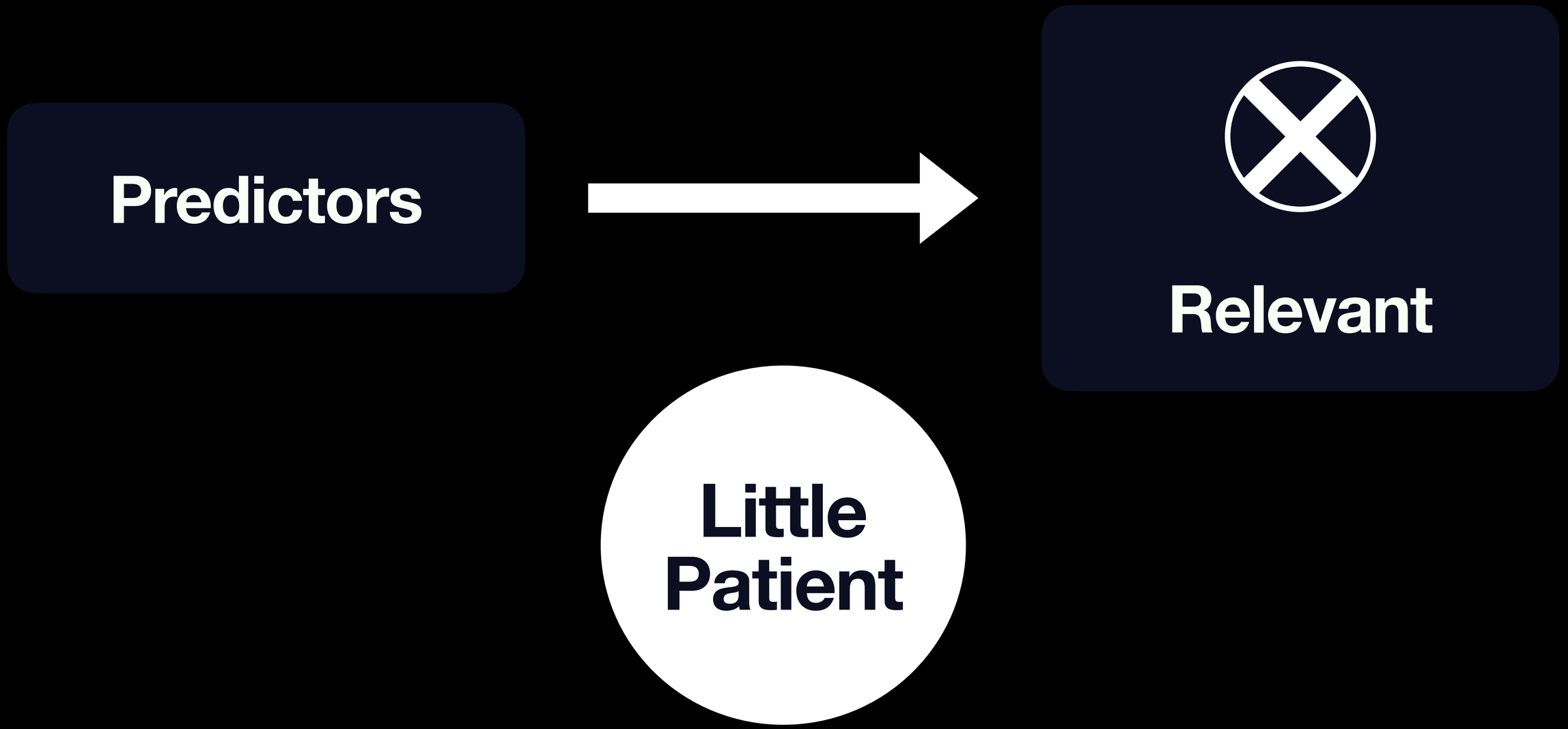
No	Yes
1466	6

Evaluation (Overall)

```
Confusion Matrix and Statistics

          Reference
Prediction   no  yes
       no  1401   65
       yes    5    1

               Accuracy : 0.9524
                 95% CI : (0.9403, 0.9627)
    No Information Rate : 0.9552
    P-Value [Acc > NIR] : 0.7189

                  Kappa : 0.0205

 Mcnemar's Test P-Value : 1.766e-12

              Precision : 0.1666667
                 Recall : 0.0151515
                     F1 : 0.0277778
             Prevalence : 0.0448370
         Detection Rate : 0.0006793
   Detection Prevalence : 0.0040761
      Balanced Accuracy : 0.5057977

       'Positive' Class : yes
```



Discussion and conclusion



THANK YOU

Kittipol Neamprasetporn 62070503404

Thanasit Suwanposri 62070503414

Siriphorn Jarisu 62070503448