# Predictive Diagnosis of Knee Conditions

Huiting Wu

Sharmeen Kapoorwala

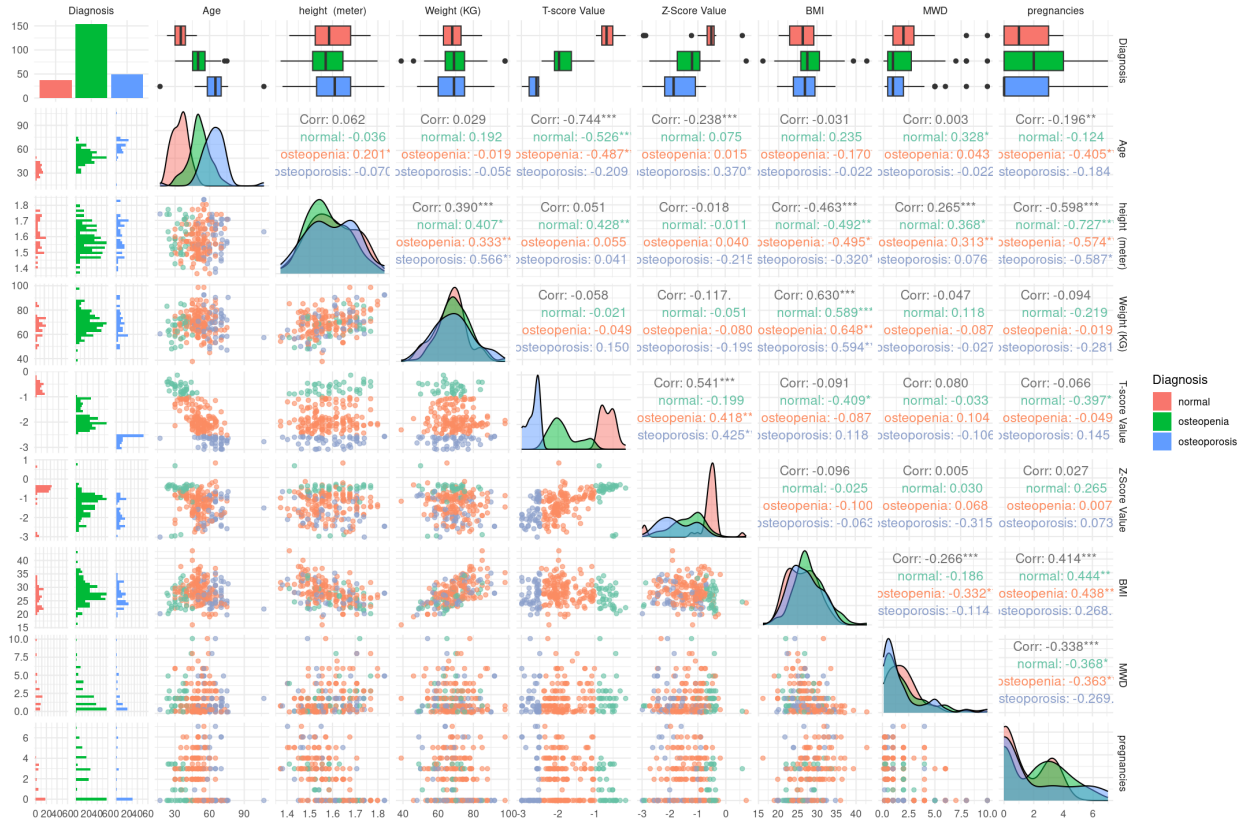
May 3, 2025

# Introduction

Osteoporosis is the leading pathological disorder of bones after arthritis, affecting millions of people worldwide. The diagnosis of osteoporosis is done by Dual Energy X-ray Absorptiometry (DEXA).Inspired by this scenario, we have selected this database for the detection of osteoporosis. The database consists of clinical factors that are responsible for osteoporosis, with the T-score values obtained from the Quantitative Ultrasound System and Knee X-ray for each participant. The research question for our project is **"Can we predict if a patient has a normal or abnormal knee diagnosis (osteopenia or osteoporosis) based on basic patient data?"** and the goal of our data analysis and statistical modeling is to create a predictive model which is cost effective, can assist with early detection so it can help the doctors use prevention strategies (like diet changes, medications, or exercise programs) before major issues develop and last but not the least factor is it will be a support for Automated Systems i.e. it can be a first step towards AI-driven diagnostic tools that could eventually assist radiologists or orthopedic specialists.The analysis includes data wrangling, analyzing patterns and corelations and eventually applying statistical modeling or machine learing to predict the diagnosis for the patients. The ultimate goal is to support the development of flexible diagnostic tools that can be particularly helpful in areas where access to conventional bone health evaluations, such as DEXA scans, is restricted.

# Data Description

The data was collected by Mendeley data and was published in August 2021 and it includes a T-score generated from the Qualitative Ultrasound System so T-score is basically a number which is generated by a bone density scan. T-score above -1 indicates that a person has healthy bone, T-scores between -1 and -2.5 indicates that a person has low bone density and if a T-score is -2.5 or lower, then a person is diagnosed with osteoporosis. Osteopenia is the term used to describe a person who has "low bone density," which indicates that their bones are weaker and that they are more likely to fracture and osteoporosis is a disorder that makes bones weak and brittle to the point where a fall or even minor stress can break them. Most often, it happens beyond the age of 50.The important variables on which we are going to focus are diagnosis (which is the response variable and we have classified as 0,1. 0 is for normal and 1 is for osteopenia or osteoporosis), Gender, Age, Weight, Height, BMI, T-Score, Z-score, Menopause Age, Number of Pregnancies and Maximum Walking Distace covered in a day. We have a total of 240 observations out of which 36 patients has normal knee x-rays, 49 has osteoporosis and 154 has osteopenia and there are 27 variables of the dataset. There are 19 female and 18 male patients diagnosed as normal. The number of female patients diagnosed with osteopenia is 36 higher than that of male patients. Among patients diagnosed with osteoporosis, there are 13 more male patients than female patients.




The correlation plot shows relationships between patient characteristics and diagnoses. Osteopenia is most common, and class distribution is imbalanced, especially when grouped as normal vs. abnormal. Age increases with diagnosis severity, indicating aging as a key factor. Height, weight, and BMI are normally distributed, with height slightly right-skewed in osteopenia due to more female patients. T-scores and Z-scores decrease as bone health worsens, with osteoporosis showing the lowest values. Age negatively correlates with T-score, and BMI, weight, and height are collinear. Z- and T-scores correlate moderately. Due to direct diagnostic influence, bone density scores were excluded to prevent data leakage.

# Methods and Results

To identify the best model for predicting abnormal diagnoses, both multiple logistic regression and decision tree models were applied. Predictor selection for logistic regression was performed using stepwise selection based on AIC and BIC criteria. The AIC-selected model included Age, Weight, Diabetic Status, BMI, Pregnancies, Career, and Medical History. In contrast, the BIC-selected model was more parsimonious, retaining only Age and Pregnancies. The decision tree model identified Age, Pregnancies, Gender, Height, Career, and BMI as key predictors. Notably, the Career variable included only two categories: housewife and others.

**Confusion Matrix (AIC)**

| | 0 | 1 | Sum |
|---|---|---|---|
| 0 | 10 | 3 | 13 |
| 1 | 1 | 58 | 59 |
| Sum | 11 | 61 | 72 |

**Confusion Matrix (BIC)**

| | 0 | 1 | Sum |
|---|---|---|---|
| 0 | 9 | 4 | 13 |
| 1 | 2 | 57 | 59 |
| Sum | 11 | 61 | 72 |

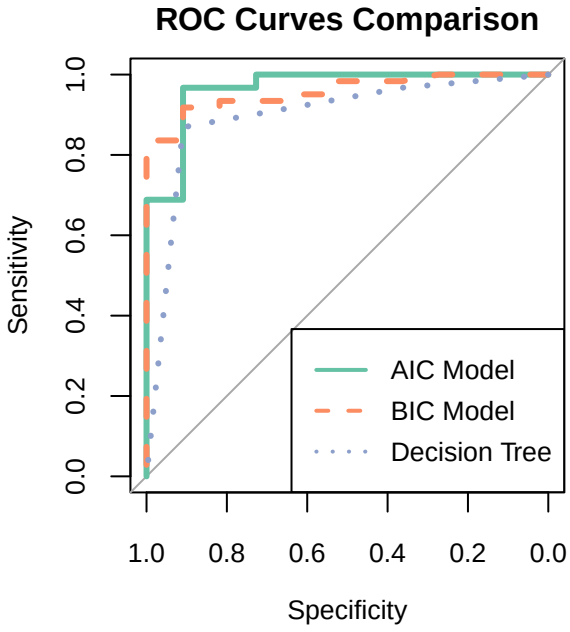
**Confusion Matrix (Decision Tree)**

| | 0 | 1 | Sum |
|---|---|---|---|
| 0 | 4 | 7 | 11 |
| 1 | 2 | 59 | 61 |
| Sum | 6 | 66 | 72 |

**Model Comparison**

| MODEL | Accuracy | Sensitivity | Specificity | AUC |
|---|---|---|---|---|
| By BIC | 91.67% | 93.44% | 81.82% | 0.9575 |
| Decision Tree Model | 87.5% | 89.39% | 66.67% | 0.8979 |
| By AIC | 94.44% | 95.1% | 90.90% | 0.9657 |

Confusion matrices and model comparisons shows the classification among AIC, BIC and Decision Tree. The accuracy of AIC model = 0.9444 so about 94% of patients in the test dataset were correctly classified. The sensitivity of AIC model = 0.9508 and it means that about 95% of patients who were diagnosed with either osteopenia or osteoporosis in the test dataset were correctly classified and the specificity = 0.9090 so we can say that about 91% of patients who were diagnosed with healthy knee bones in the test dataset were correctly classified. Also AUC = 0.9657 which is a lot closer to 1 and we can conclude that AIC is the better model which performs at classification.




ROC curve shows the comparison of different models, green represent curve for AIC, orange dashed line represents BIC model and blue dotted line represents Decision Tree which we used for classification and it is visible that curve for AIC shows better values and closer to

the goal of having sensitivity = 1 and specificity = 1.

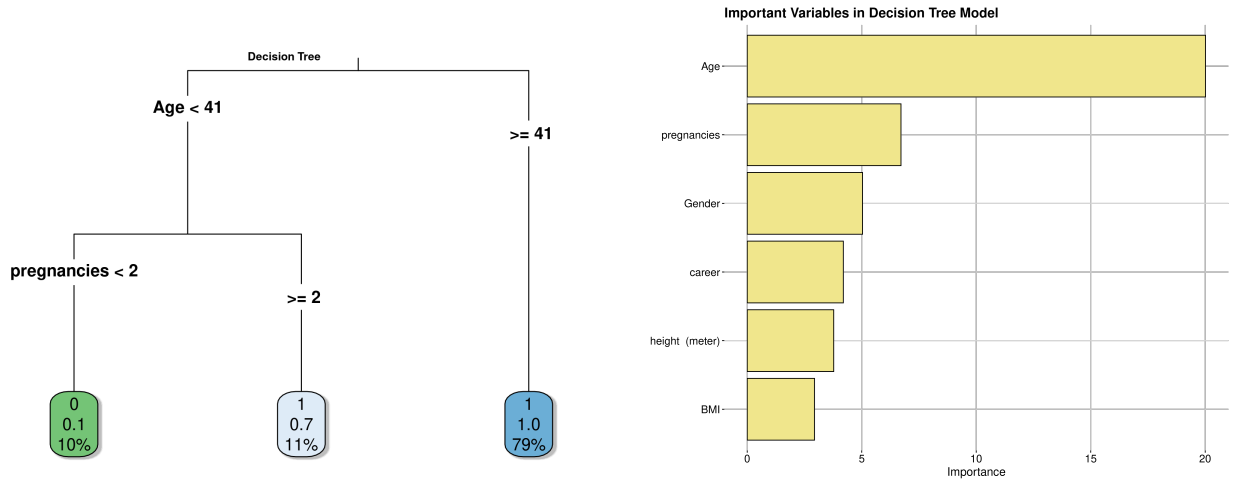



Figure 1: Decision Tree (left) and Variable Importance (right)

The decision tree predicts abnormal diagnoses ('1' = osteopenia/osteoporosis, '0' = normal). Age is the strongest predictor. Patients over 41 (79% of the sample) were all predicted as abnormal. Among those under 41 with two or more pregnancies (11%), 70% were predicted abnormal, while 90% with fewer pregnancies (10%) were predicted normal. The important variables include age, number of pregnancies, gender, career, height, and BMI.

# Conclusion

Overall, the model selected by AIC method which performs better at classification with the model equation **logit(estimated abnormal) = - 9.424 + 0.262 Age + 0.089 Weight - 2.780 Diabetic(yes) - 0.265 BMI + 0.665 Pregnancies - 1.142 career(others) + 1.324 medical history(yes))**. There is a positive relationship between the chance of abnormal diagnosis with age, weight, pregnancies, and having a medical history before. Also, conclude that knee health is substantially predicted by **Age** and **pregnancy**. This implies that women who have had more pregnancies and those who are aged are more likely to develop osteopenia or osteoporosis.

Our dataset was limited, with just 240 patients, and it was not balanced, with an unequal number of patients for each diagnosis. These were the difficulties we faced. By using the T-score as our response variable, we can enhance the prediction model and possibly arrive at a different conclusion than the one we now have. In order to obtain more accurate predictions, it would also be vital to explore additional machine learning models, such as random forest or XGBoost, and to have a larger and more balanced dataset.

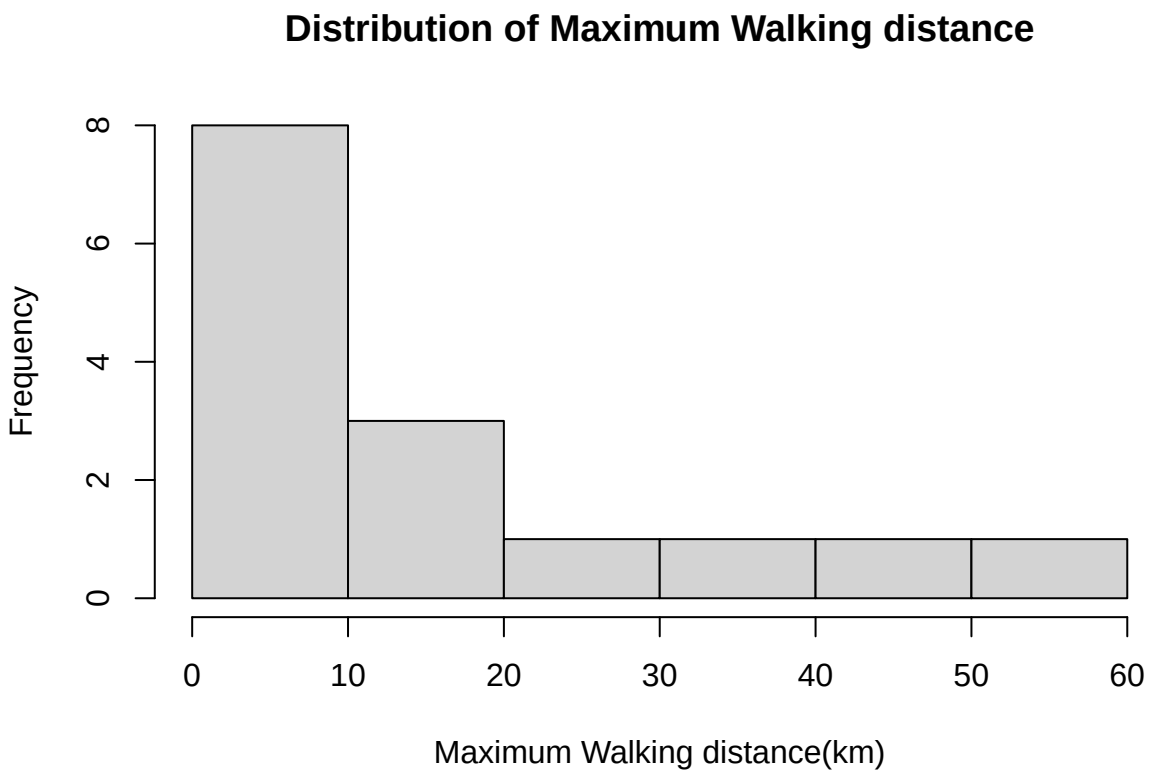
# Code Appendix

## Import the data

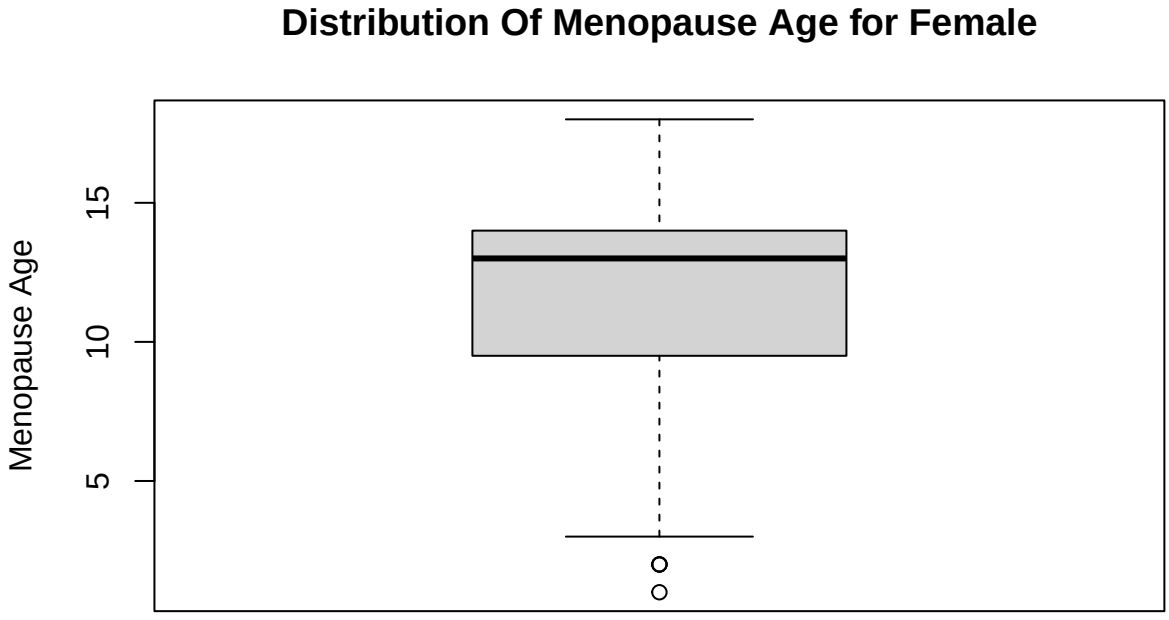
```
library(stringr)
library(readxl)
library(visdat)
library(dplyr)
library(car)
library(GGally)
library(rpart)
library(caret)
library(pROC)
library(rpart.plot)
library(ggplot2)
library(knitr)
library(rsample)
library(vip)
```

```
knee <- read_excel("patient details.xlsx")
knee <- knee[1:(nrow(knee) - 3), ]
knee <- knee |> mutate(across(where(is.character), as.factor))
```

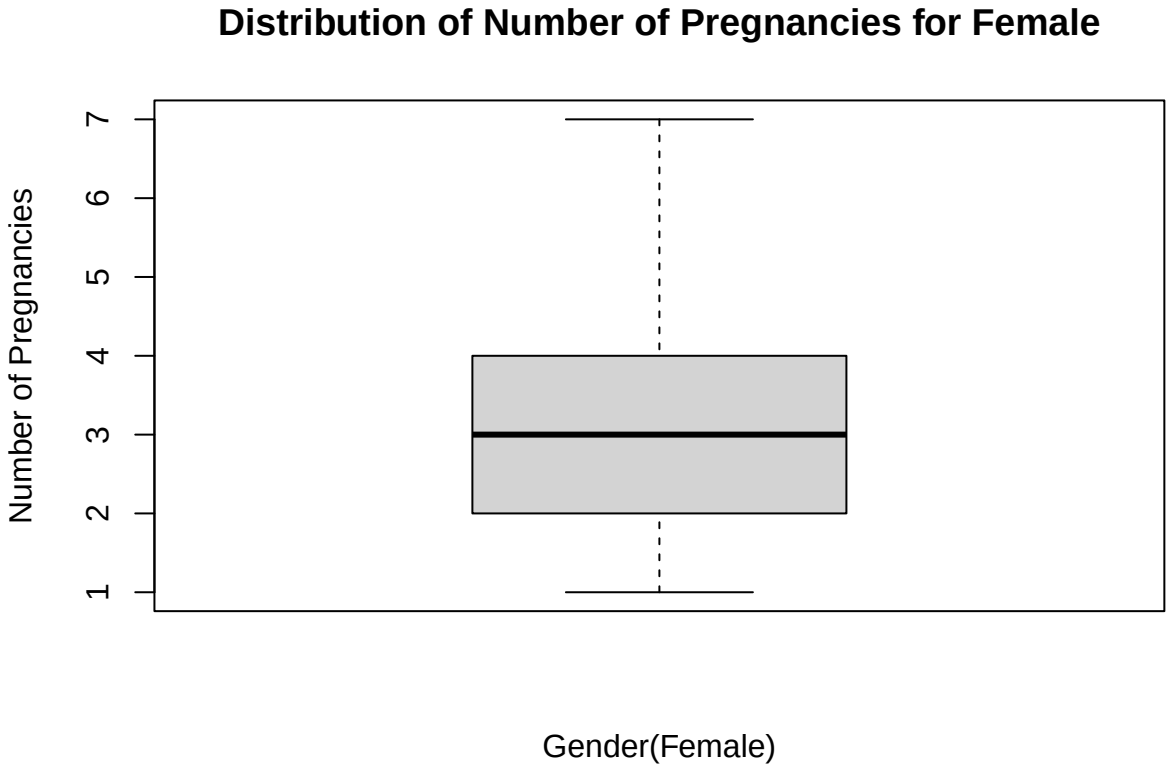
```
knee |> select(`Maximum Walking distance (km)`) |>
  table() |> hist(main = "Distribution of Maximum Walking distance",
                  xlab = "Maximum Walking distance(km)")
```




```
knee |> filter(Gender == "female") |>
  select(`Menopause Age`) |>
  boxplot(main = "Distribution Of Menopause Age for Female",
          ylab = "Menopause Age",
          xlab = "Gender in Female")
```




```
knee |> filter(Gender == "female") |>
  select(`Number of Pregnancies`) |>
  boxplot(main = "Distribution of Number of Pregnancies for Female",
          ylab = "Number of Pregnancies",
          xlab = "Gender(Female)")
```



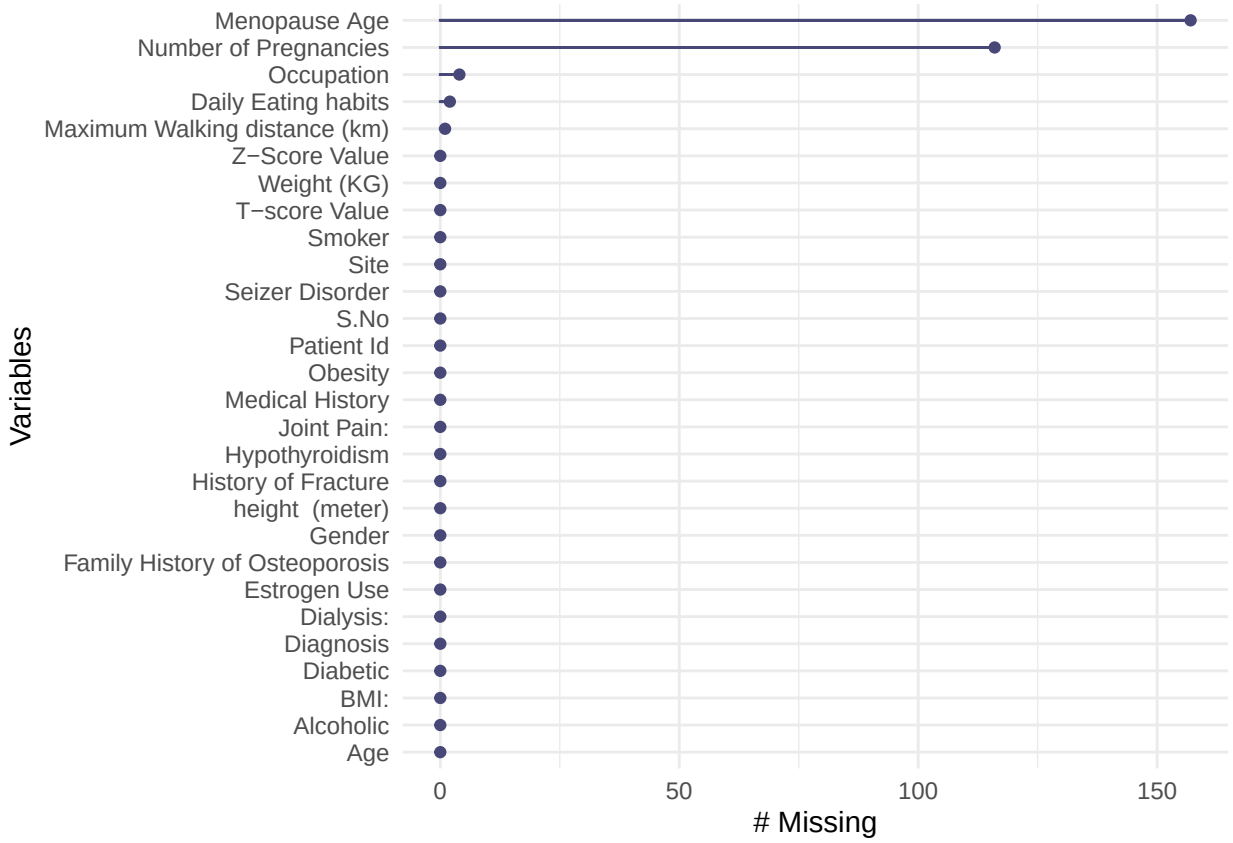

## Missing Values

```
colSums(is.na(knee))
```

```
##                             S.No                       Patient Id
##                                0                                0
##                      Joint Pain:                           Gender
##                                0                                0
##                              Age                      Menopause Age
##                                0                              157
##                  height  (meter)                     Weight (KG)
##                                0                                0
##                           Smoker                       Alcoholic
##                                0                                0
##                         Diabetic                     Hypothyroidism
##                                0                                0
##           Number of Pregnancies                    Seizer Disorder
##                              116                                0
##                    Estrogen Use                       Occupation
##                                0                                4
##              History of Fracture                         Dialysis:
##                                0                                0
## Family History of Osteoporosis  Maximum Walking distance (km)
##                                0                                1
```

```
##            Daily Eating habits                  Medical History
##                              2                                0
##                T-score Value                  Z-Score Value
##                              0                                0
##                           BMI:                             Site
##                              0                                0
##                        Obesity                        Diagnosis
##                              0                                0
```

```
naniar::gg_miss_var(knee)
```



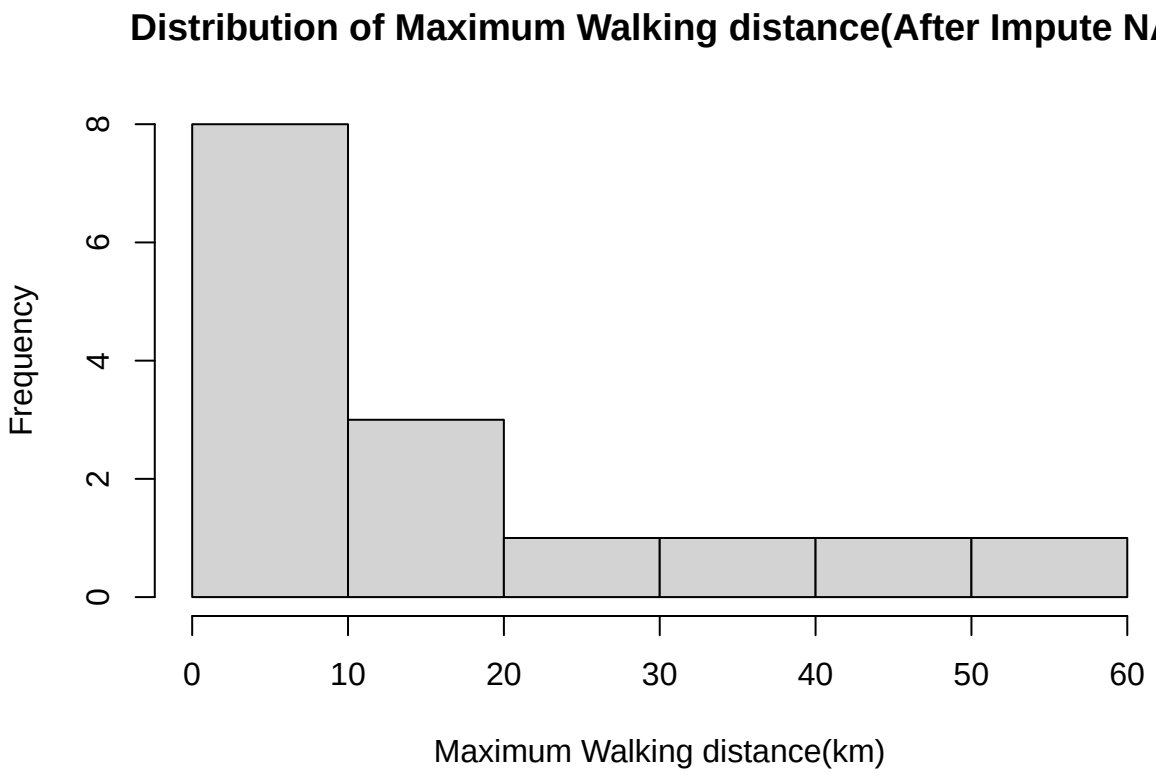

## Clean the data

### Maximum Walking distance

- There is 1 missing value in `Maximum Walking distance (km)`. The distribution is right skewed. Impute it with the median.

```
mwd_median <- median(knee$`Maximum Walking distance (km)`, na.rm = T)
knee <- knee |> mutate(
  MWD = ifelse(is.na(`Maximum Walking distance (km)`), mwd_median,
               `Maximum Walking distance (km)`)
) |> select(-`Maximum Walking distance (km)`)
```

```r
knee |> select(MWD) |>
  table() |> hist(main = "Distribution of Maximum Walking distance(After Impute NA)",
                  xlab = "Maximum Walking distance(km)")
```




### Menopause Age

- Menopause typically occurs naturally between the ages of 45 and 55, with the average age being 51. It's defined as the point when a woman has not had a menstrual period for 12 consecutive months.

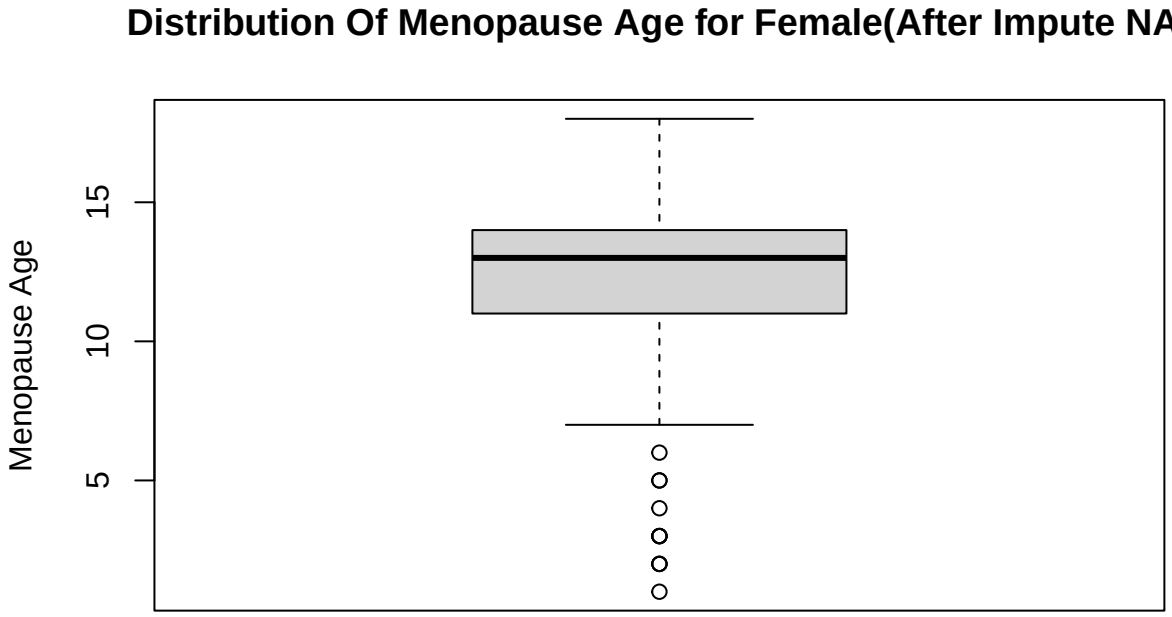
```r
knee$Menopause_Age <- as.numeric(knee$`Menopause Age`)
knee <- knee |> select(-`Menopause Age`)
```

- The Menopause Age is right skewed for female. Impute the missing value with its median for female only. For male would be not apply.

```r
female_median <- median(knee$Menopause_Age[knee$Gender == "female"], na.rm = TRUE)

knee <- knee  |>
  mutate(Menopause_Age = case_when(
    Gender == "female" & is.na(Menopause_Age) ~ female_median,
    Gender == "male" & is.na(Menopause_Age) ~ NA_real_,  # keep as NA
    TRUE ~ Menopause_Age
  ))
```

```
knee |> filter(Gender == "female") |>
  select(Menopause_Age) |>
  boxplot(main = "Distribution Of Menopause Age for Female(After Impute NA)",
          ylab = "Menopause Age",
          xlab = "Gender in Female")
```




### Number of Pregnancies

- There is only one patient gender in male with number of pregnancies=4. Others are missing values.

```
knee |>
  filter(!is.na(`Number of Pregnancies`) & Gender == "male")
```

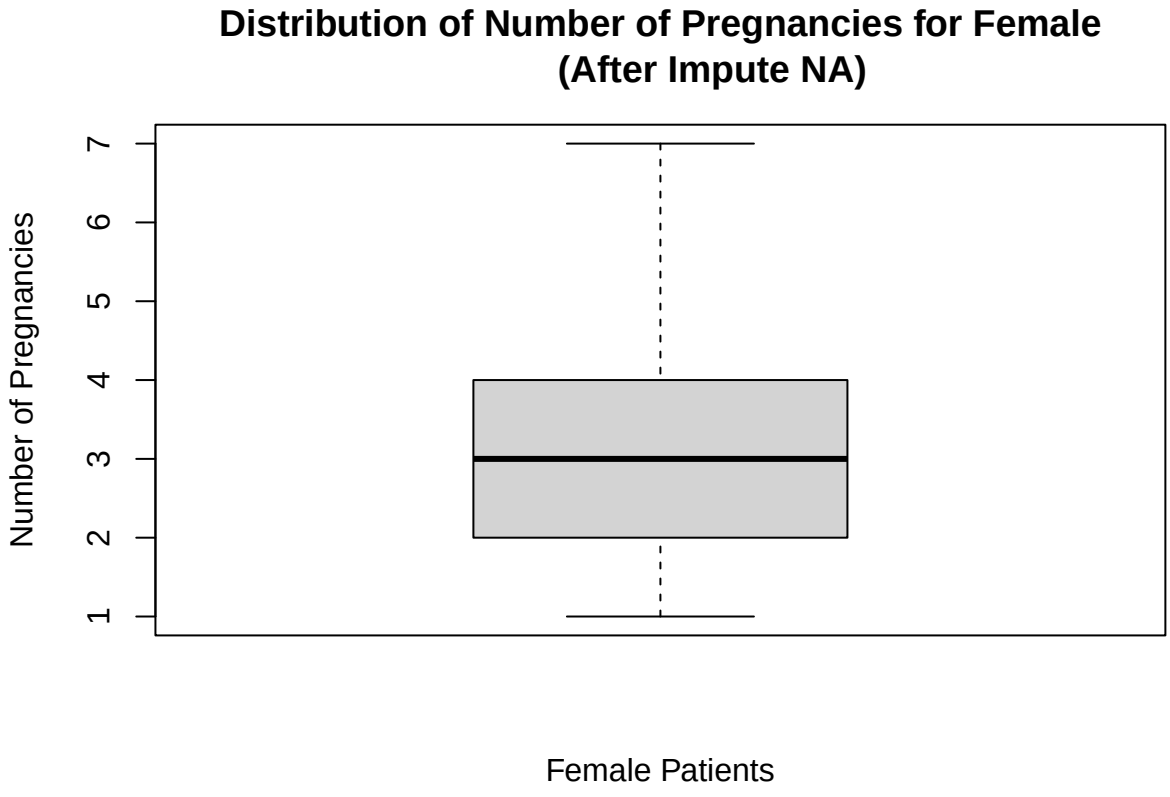
```
## # A tibble: 1 x 28
##    S.No `Patient Id` `Joint Pain:` Gender   Age `height  (meter)` `Weight (KG)`
##   <dbl> <fct>        <fct>         <fct>  <dbl>             <dbl>         <dbl>
## 1    97 OP72         yes           male      39              1.71            75
## # i 21 more variables: Smoker <fct>, Alcoholic <fct>, Diabetic <fct>,
## #   Hypothyroidism <fct>, `Number of Pregnancies` <dbl>,
## #   `Seizer Disorder` <fct>, `Estrogen Use` <fct>, Occupation <fct>,
## #   `History of Fracture` <fct>, `Dialysis:` <fct>,
## #   `Family History of Osteoporosis` <fct>, `Daily Eating habits` <fct>,
## #   `Medical History` <fct>, `T-score Value` <dbl>, `Z-Score Value` <dbl>,
## #   `BMI:` <dbl>, Site <fct>, Obesity <fct>, Diagnosis <fct>, MWD <dbl>, ...
```

- Impute the gender male for number of pregnancies is 0.

```
female_mean <- mean(knee$`Number of Pregnancies`[knee$Gender == "female"],
                    na.rm = TRUE)

knee <- knee  |>
  mutate(pregnancies = case_when(
    Gender == "female" & is.na(`Number of Pregnancies`) ~ female_mean,
    Gender == "male" & is.na(`Number of Pregnancies`) ~ 0,  # keep as NA
    TRUE ~ `Number of Pregnancies`
  )) |>
  select(-`Number of Pregnancies`)

knee |> filter(Gender == "female") |>
  select(pregnancies) |>
  boxplot(main = "Distribution of Number of Pregnancies for Female
          (After Impute NA)",
          ylab = "Number of Pregnancies",
          xlab = "Female Patients")
```




**There are some categoriacal variables with only one level and some missing values**

- `Alcoholic` only has the observation of "no". This variable will be dropped since is not helpful to put into the logistic regression.

- `Dialysis` is extremely imbalanced. There are only 1 observation with the level of "yes". The rest are "no". It is not stable in estimates.
- `Site` is not important. The data set is all about the knee.

### Other Variables

- `Occupation` with too many levels and most of the levels are only 1 observation. Need to be organized. Majority is house wife.

```
occupation_corrections <- c(
  "h.wife" = "housewife"
)

knee <- knee |>
  mutate(
     # lowercase and trim
    career_clean = tolower(trimws(Occupation)),

    # replace the names
    career_clean = str_replace_all(career_clean, occupation_corrections),

    # the missing value as "unknown"
    career_clean = ifelse(is.na(career_clean), "unknown", career_clean),

    # organize it more
    career = case_when(career_clean == "housewife" ~ "housewife",
      TRUE ~ "others"
    )
  ) |> select(-career_clean, -Occupation)
table(knee$career)
```

```
##
## housewife    others
##       115       125
```

- `Daily Eating habits` same. Need to be organized. The main limits are Low/No Fat, Low Salt, Low/No Protein, Low Sugar, No Sour Food. Organize it as 2 levels Normal and Limited. The missing value as normal(there are 2).

```
knee <- knee |>
  mutate(
    eating_clean = tolower(trimws(`Daily Eating habits`)),
    eating_habit = ifelse(
      eating_clean == "normal", "normal", "limited"
    )
  ) |>
```

```
  select(-eating_clean, -`Daily Eating habits`)
table(knee$eating_habit)
```

```
##
## limited  normal
##      51     187
```

- `Obesity` combine over weight and overweight.

```
knee <- knee |> mutate(
  obesity = ifelse(Obesity == "overweight", "over weight", Obesity)
) |> select(-Obesity)
table(knee$obesity)
```

```
##
##           1           2           3           5 over weight
##          58          67         112           1           2
```

- Clean up `History of Fracture` into injuries in lower body, upper body, other, or no injury

```
knee <- knee |>
  mutate(
    injury_type = `History of Fracture`,
    # Clean up and categorize injuries into main parts
    injury_part = case_when(
      str_detect(injury_type, "leg") ~ "leg",
      str_detect(injury_type, "foot") ~ "foot",
      str_detect(injury_type, "arm") ~ "arm",
      str_detect(injury_type, "wrist") ~ "wrist",
      str_detect(injury_type, "shoulder") ~ "shoulder",
      str_detect(injury_type, "hip") ~ "hip",
      str_detect(injury_type, "no") ~ "no injury",
      str_detect(injury_type, "knee") ~ "knee",
      str_detect(injury_type, "head") ~ "head",
      str_detect(injury_type, "neck") ~ "neck",
      TRUE ~ "Other"
    ),

    # Categorizing into lower body, upper body, and fractures
    injury = case_when(
      injury_part %in% c("leg", "foot", "hip", "knee") ~ "Lower Body",
      injury_part %in% c("arm", "wrist", "shoulder", "head", "neck") ~ "Upper Body",
      injury_part == "Other" ~ "Other",
      injury_part == "no injury" ~ "no injury",
      TRUE ~ "Fractures"
```

```
    )
  ) |> select(-injury_type, -injury_part, -`History of Fracture`)
table(knee$injury)
```

```
##
## Lower Body  no injury      Other Upper Body
##         30        169          23          18
```

- Clean `Medical History` into yes for having medical history and no for not having medical history

```
knee <- knee |> mutate(
  medical_hist = ifelse(`Medical History` %in% c("no", "normal"), "no", "yes")
) |> select(-`Medical History`)
table(knee$medical_hist)
```

```
##
##  no yes
## 114 126
```

```
knee <- knee |> mutate(across(where(is.character), as.factor)) |>
  select(-`Patient Id`, -Alcoholic, -Site, -S.No) |>
  rename(BMI = `BMI:`,
         dialysis = `Dialysis:`,
         joint_pain = `Joint Pain:`)
```

```
# Create a binary response with yes or no (1, 0)
knee_clean <- knee |> mutate(
  dummy_diagnosis = ifelse(Diagnosis == "normal", 0, 1),
  dummy_diagnosis = as.factor(dummy_diagnosis),
)
```

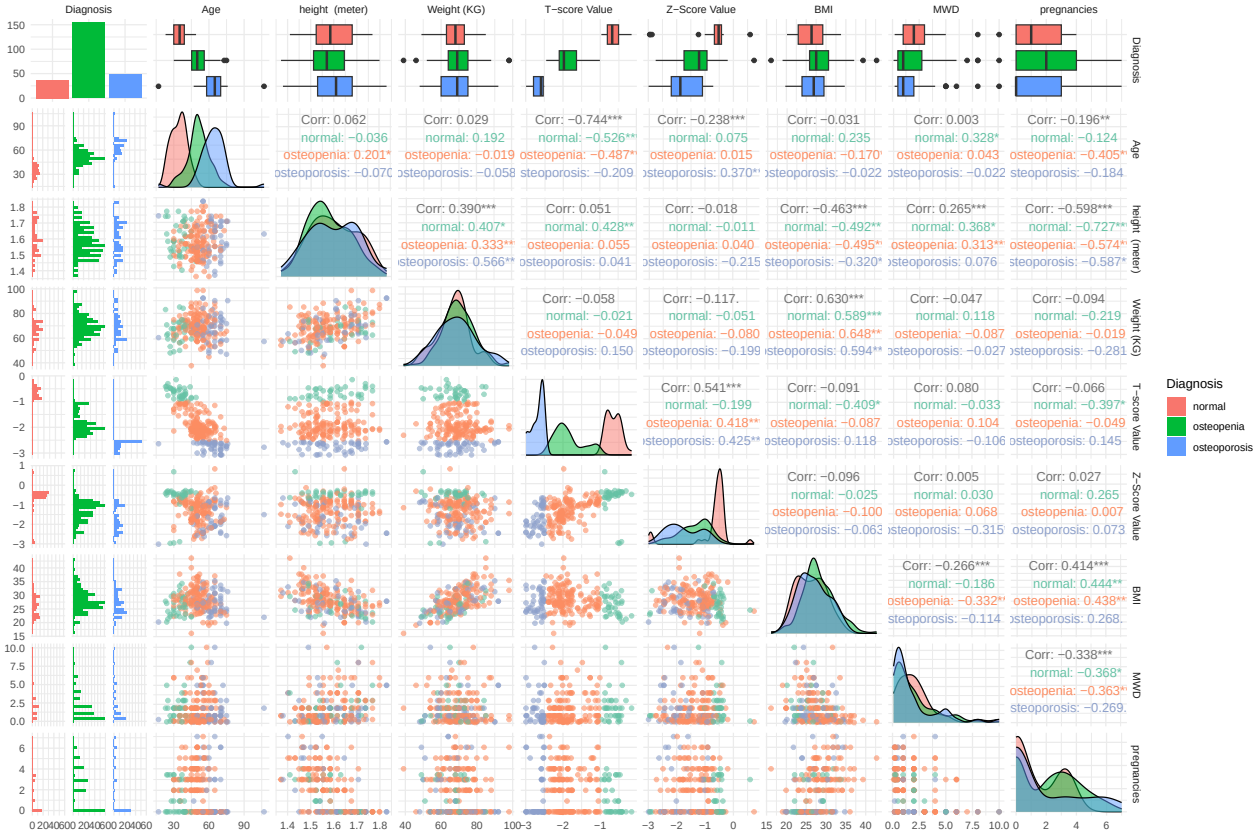
**Rename the variables and filter out not interesting variables**

## Visualization

### Corelation

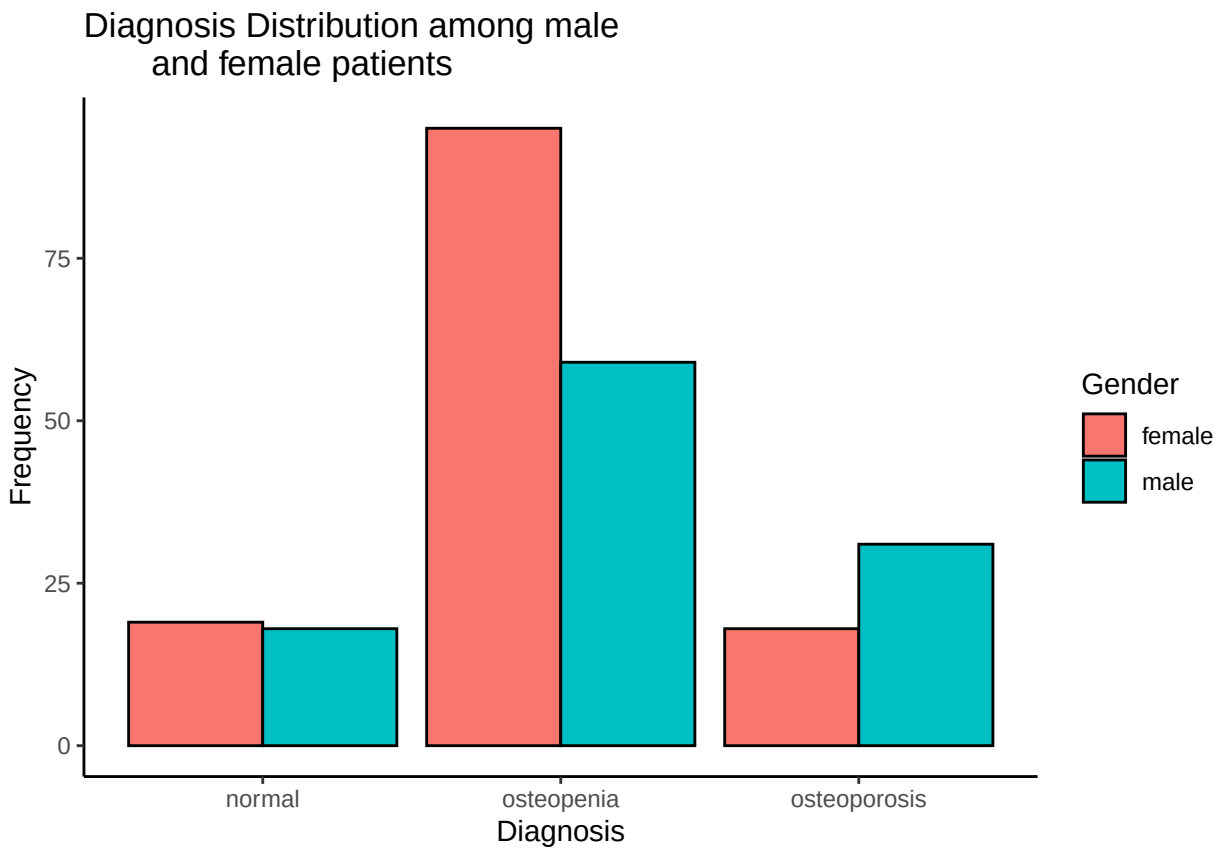
```
ggpairs(
  knee |> select(Diagnosis, where(is.numeric), -Menopause_Age),
  aes(color = Diagnosis),
  upper = list(continuous = wrap("cor", size = 4)),
  lower = list(continuous = wrap("points", alpha = 0.6)),
  diag = list(continuous = wrap("densityDiag", alpha = 0.5)),
```

```
  legend = 1
) +
  scale_color_brewer(palette = "Set2") +
  theme_minimal()
```




**Diagnoses comparing with gender male excluding [menopause age, number of pregnancies]**

```
knee_clean |> group_by(Diagnosis) |>
  ggplot(aes(x = Diagnosis, fill = Gender)) +
  geom_bar(position = "dodge", col = "black") +
  labs(title = "Diagnosis Distribution among male
       and female patients",
       y = "Frequency") +
  theme_classic()
```




```
knee_male <- knee_clean |>
  filter(Gender == "male") |>
  group_by(Diagnosis) |>
  select(-Menopause_Age) |>
    count(name = "Male Patients")
knee_male
```

```
## # A tibble: 3 x 2
## # Groups:   Diagnosis [3]
##   Diagnosis    `Male Patients`
##   <fct>                  <int>
## 1 normal                    18
## 2 osteopenia                59
## 3 osteoporosis              31
```

```
knee_female <- knee_clean |>
  filter(Gender == "female") |>
  group_by(Diagnosis) |>
    count(name = "Female Patients")
knee_female
```

```
## # A tibble: 3 x 2
## # Groups:   Diagnosis [3]
```

```
##   Diagnosis    `Female Patients`
##   <fct>                    <int>
## 1 normal                      19
## 2 osteopenia                  95
## 3 osteoporosis                18
```

## Logistic Regression

```
knee_model <- knee_clean |>
  select(-Diagnosis, -Menopause_Age, -dialysis,
         -`T-score Value`, -`Z-Score Value`, -obesity) |> na.omit()

full <- glm(dummy_diagnosis ~ ., data = knee_model, family = binomial)
summary(full)
```

```
##
## Call:
## glm(formula = dummy_diagnosis ~ ., family = binomial, data = knee_model)
##
## Coefficients:
##                                    Estimate Std. Error z value Pr(>|z|)
## (Intercept)                        11.83549 4066.85640   0.003   0.9977
## joint_painyes                     -16.89569 4066.63745  -0.004   0.9967
## Gendermale                          0.38600    1.70290   0.227   0.8207
## Age                                 0.28126    0.05386   5.222 1.77e-07 ***
## `height  (meter)`                  -1.51207   27.21611  -0.056   0.9557
## `Weight (KG)`                       0.11738    0.32820   0.358   0.7206
## Smokeryes                           0.19566    1.02371   0.191   0.8484
## Diabeticyes                        -2.97867    1.64731  -1.808   0.0706 .
## Hypothyroidismyes                   0.32425    0.92971   0.349   0.7273
## `Seizer Disorder`yes               18.44535 1813.78475   0.010   0.9919
## `Estrogen Use`yes                  -1.90459    3.53591  -0.539   0.5901
## `Family History of Osteoporosis`yes  0.50191    0.80890   0.620   0.5349
## BMI                                -0.40544    0.78035  -0.520   0.6034
## MWD                                -0.09362    0.15737  -0.595   0.5519
## pregnancies                         0.71560    0.38809   1.844   0.0652 .
## careerothers                       -1.67800    1.12826  -1.487   0.1369
## eating_habitnormal                  0.23946    0.80979   0.296   0.7675
## injuryno injury                    -1.08227    1.30905  -0.827   0.4084
## injuryOther                        -0.32545    1.73556  -0.188   0.8513
## injuryUpper Body                   -1.31655    1.92442  -0.684   0.4939
## medical_histyes                     1.12352    0.69986   1.605   0.1084
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 202.247  on 237  degrees of freedom
## Residual deviance:  78.222  on 217  degrees of freedom
## AIC: 120.22
##
## Number of Fisher Scoring iterations: 17
```

```
null <- glm(dummy_diagnosis ~ 1, data = knee_model, family = binomial)
```

### AIC for classification

```
step(full, trace = 0)
```

```
##
## Call:  glm(formula = dummy_diagnosis ~ Age + `Weight (KG)` + Diabetic +
##     BMI + pregnancies + career + medical_hist, family = binomial,
##     data = knee_model)
##
## Coefficients:
##     (Intercept)              Age    `Weight (KG)`      Diabeticyes
##        -9.42370          0.26151          0.08905         -2.77973
##             BMI      pregnancies     careerothers  medical_histyes
##        -0.26480          0.66455         -1.14204          1.32440
##
## Degrees of Freedom: 237 Total (i.e. Null);  230 Residual
## Null Deviance:       202.2
## Residual Deviance: 82.46     AIC: 98.46
```

```
select_aic <- glm(dummy_diagnosis ~ Age + `Weight (KG)` + Diabetic +
    BMI + pregnancies + career + medical_hist, family = binomial,
    data = knee_model)

summary(select_aic)
```

```
##
## Call:
## glm(formula = dummy_diagnosis ~ Age + `Weight (KG)` + Diabetic +
##     BMI + pregnancies + career + medical_hist, family = binomial,
##     data = knee_model)
##
## Coefficients:
##                 Estimate Std. Error z value Pr(>|z|)
## (Intercept)     -9.42370    2.97274  -3.170  0.00152 **
## Age              0.26151    0.04405   5.937  2.9e-09 ***
## `Weight (KG)`    0.08905    0.05658   1.574  0.11553
```

```
## Diabeticyes     -2.77973    1.55550  -1.787  0.07393 .
## BMI             -0.26480    0.13245  -1.999  0.04557 *
## pregnancies      0.66455    0.28832   2.305  0.02117 *
## careerothers    -1.14204    0.80288  -1.422  0.15490
## medical_histyes  1.32440    0.62093   2.133  0.03293 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 202.247  on 237  degrees of freedom
## Residual deviance:  82.456  on 230  degrees of freedom
## AIC: 98.456
##
## Number of Fisher Scoring iterations: 7
```

### BIC for classification

```
step(full, trace = 0, k = log(nrow(knee_model)))
```

```
##
## Call:  glm(formula = dummy_diagnosis ~ Age + pregnancies, family = binomial,
##     data = knee_model)
##
## Coefficients:
## (Intercept)          Age  pregnancies
##     -9.5510       0.2356       0.5958
##
## Degrees of Freedom: 237 Total (i.e. Null);  235 Residual
## Null Deviance:       202.2
## Residual Deviance: 94.38     AIC: 100.4
```

```
select_bic <- glm(dummy_diagnosis ~ Age + pregnancies, family = binomial,
    data = knee_model)
summary(select_bic)
```

```
##
## Call:
## glm(formula = dummy_diagnosis ~ Age + pregnancies, family = binomial,
##     data = knee_model)
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept) -9.55100    1.62787  -5.867 4.43e-09 ***
## Age          0.23555    0.03657   6.441 1.19e-10 ***
## pregnancies  0.59581    0.19036   3.130  0.00175 **
## ---
```

```
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 202.247  on 237  degrees of freedom
## Residual deviance:  94.383  on 235  degrees of freedom
## AIC: 100.38
##
## Number of Fisher Scoring iterations: 7
```

```
AIC(null, full, select_bic, select_aic)
```

```
##            df       AIC
## null        1 204.24733
## full       21 120.22221
## select_bic  3 100.38282
## select_aic  8  98.45551
```

```
BIC(null, full, select_bic, select_aic)
```

```
##            df      BIC
## null        1 207.7196
## full       21 193.1399
## select_bic  3 110.7996
## select_aic  8 126.2337
```

## Cross Validation: check the logistic model

### BIC Model

```
set.seed(333)
n <- nrow(knee_model)
floor(0.7*n)
```

```
## [1] 166
```

```
# split into training data and testing data
train <- sample(1:n, 166)

# fit the model with training data
glm_train <- glm(dummy_diagnosis ~ Age + pregnancies, family = binomial,
    data = knee_model, subset = train) # chose the smallest bic model

# summary
summary(glm_train)
```

```
##
## Call:
```

```
## glm(formula = dummy_diagnosis ~ Age + pregnancies, family = binomial,
##     data = knee_model, subset = train)
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept) -9.08912    1.92479  -4.722 2.33e-06 ***
## Age          0.21715    0.04124   5.265 1.40e-07 ***
## pregnancies  0.72453    0.23982   3.021  0.00252 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 140.69  on 165  degrees of freedom
## Residual deviance:  67.86  on 163  degrees of freedom
## AIC: 73.86
##
## Number of Fisher Scoring iterations: 7
```

```
# test set
test <- knee_model[-train, ]

# prediction on test set
pred_prob <- predict(glm_train, newdata = test, type = "response")

# classify the prediction
length(pred_prob)
```

```
## [1] 72
```

```
class_preds <- rep(0, 72)
class_preds[pred_prob > 0.5] <- 1

# confusion matrix
addmargins(table(prediction = class_preds, actual = test$dummy_diagnosis))
```

```
##           actual
## prediction  0  1 Sum
##        0    9  4  13
##        1    2 57  59
##        Sum 11 61  72
```

```
# accuracy
(9 + 57)/72 # 0.9166667
```

```
## [1] 0.9166667
```

```r
# sensitivity
57/61 # 0.9344262
```

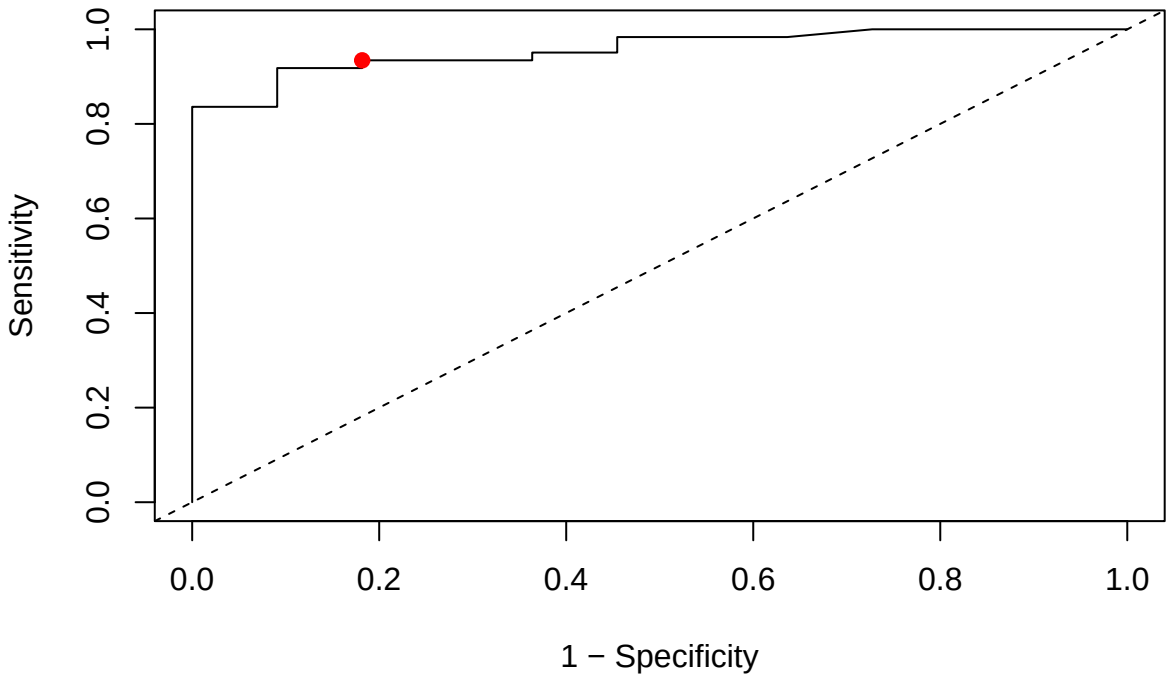
```
## [1] 0.9344262
```

```r
# specificity
9/11 # 0.8181818
```

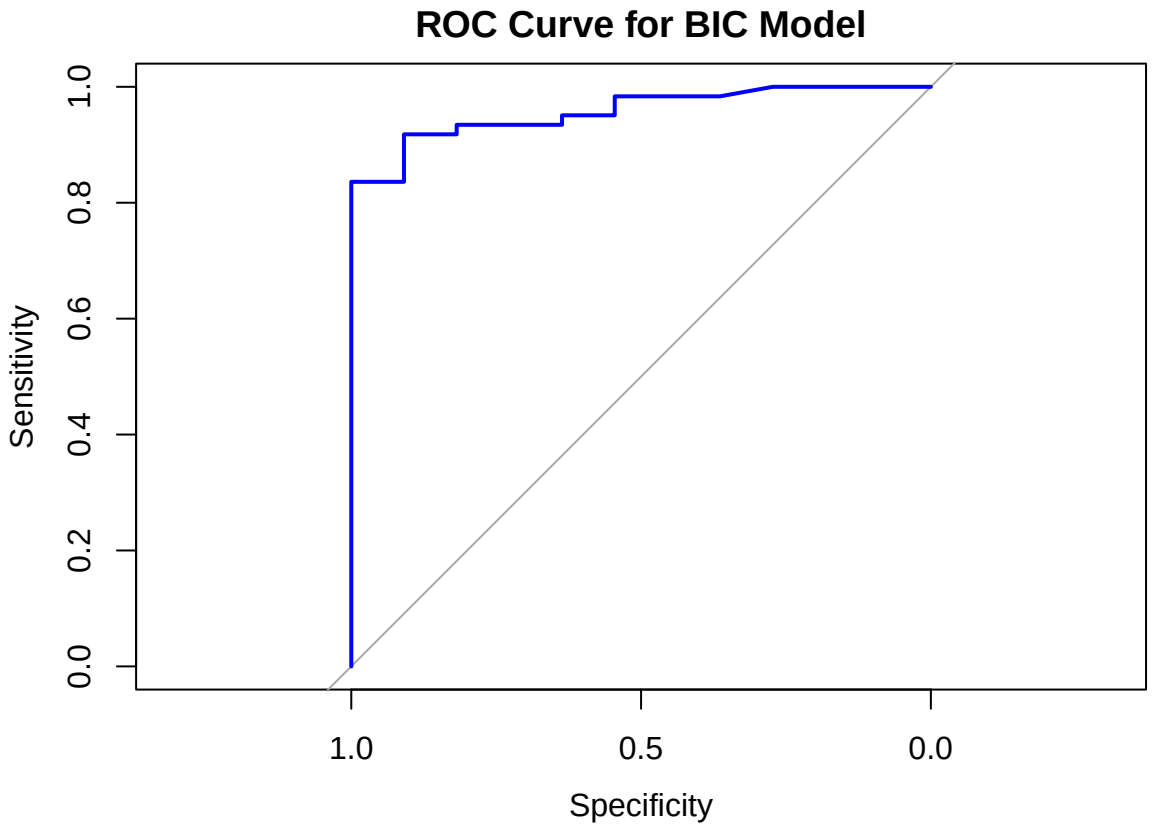
```
## [1] 0.8181818
```

```r
# ROC curve
roc_obj <- roc(test$dummy_diagnosis, pred_prob)
plot(1 - roc_obj$specificities, roc_obj$sensitivities, type = "l",
     xlab = "1 - Specificity", ylab = "Sensitivity")

abline(0, 1, lty=2)
points(x = 2/11, y = 57/61, col = "red", pch = 19)
```




```r
# ROC Curve
roc_curve <- roc(test$dummy_diagnosis, pred_prob)
plot(roc_curve, main = "ROC Curve for BIC Model", col = "blue", lwd = 2)
```




```
# AUC
auc(roc_obj) # 0.9575
```

```
## Area under the curve: 0.9575
```

### AIC model

```
# fit the model with training data
glm_train <- glm(dummy_diagnosis ~ Age + `Weight (KG)` + Diabetic +
    BMI + pregnancies + career + medical_hist, family = binomial,
    data = knee_model)

# summary
summary(glm_train)
```

```
##
## Call:
## glm(formula = dummy_diagnosis ~ Age + `Weight (KG)` + Diabetic +
##     BMI + pregnancies + career + medical_hist, family = binomial,
##     data = knee_model)
##
## Coefficients:
##                 Estimate Std. Error z value Pr(>|z|)
## (Intercept)     -9.42370    2.97274  -3.170  0.00152 **
```

```
## Age              0.26151    0.04405   5.937  2.9e-09 ***
## `Weight (KG)`    0.08905    0.05658   1.574  0.11553
## Diabeticyes     -2.77973    1.55550  -1.787  0.07393 .
## BMI             -0.26480    0.13245  -1.999  0.04557 *
## pregnancies      0.66455    0.28832   2.305  0.02117 *
## careerothers    -1.14204    0.80288  -1.422  0.15490
## medical_histyes  1.32440    0.62093   2.133  0.03293 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 202.247  on 237  degrees of freedom
## Residual deviance:  82.456  on 230  degrees of freedom
## AIC: 98.456
##
## Number of Fisher Scoring iterations: 7
```

```
# test set
test <- knee_model[-train, ]

# prediction on test set
pred_prob <- predict(glm_train, newdata = test, type = "response")

# classify the prediction
class_preds <- rep(0, 72)
class_preds[pred_prob > 0.5] <- 1

# confusion matrix
addmargins(table(prediction = class_preds, actual = test$dummy_diagnosis))
```

```
##           actual
## prediction  0  1 Sum
##        0   10  3  13
##        1    1 58  59
##        Sum 11 61  72
```

```
# accuracy
(10 + 58)/72 # 0.9444444
```

```
## [1] 0.9444444
```

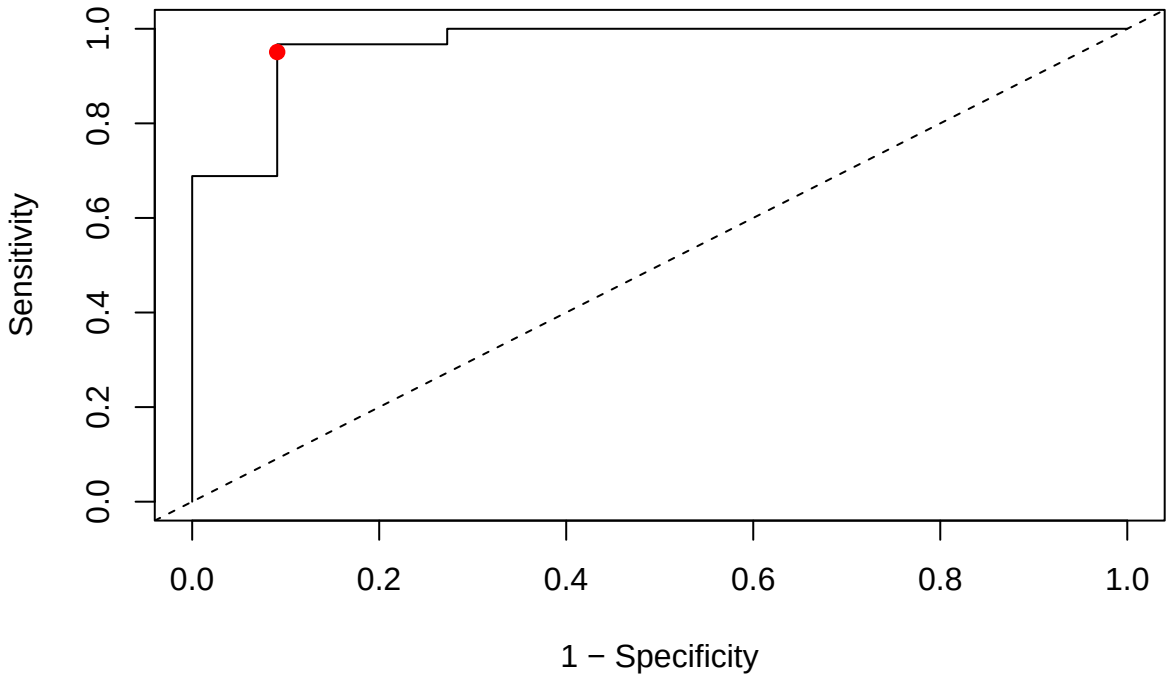
```
# sensitivity
58/61 # 0.9508197
```

```
## [1] 0.9508197
```

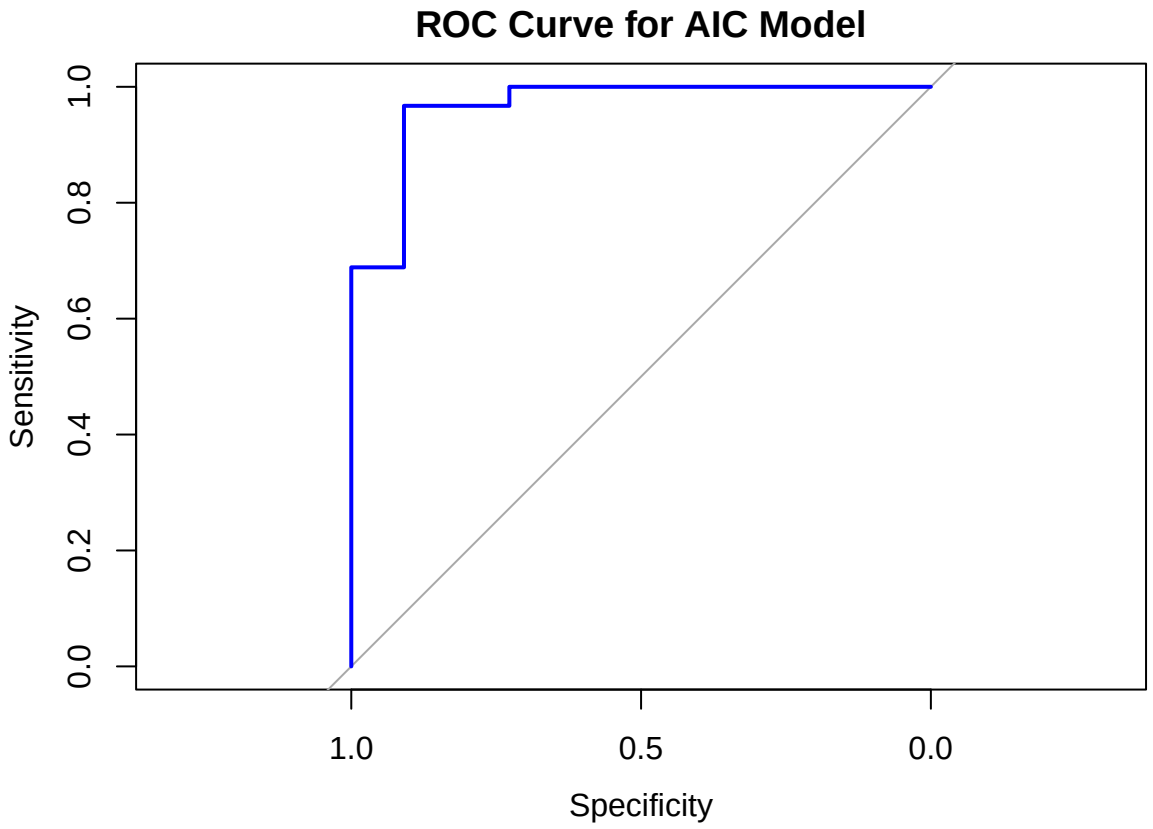
```r
# specificity
10/11 # 0.9090909
```

```
## [1] 0.9090909
```

```r
# ROC curve
roc_obj <- roc(test$dummy_diagnosis, pred_prob)
plot(1 - roc_obj$specificities, roc_obj$sensitivities, type = "l",
     xlab = "1 - Specificity", ylab = "Sensitivity")
abline(0, 1, lty=2)
points(x = 1/11, y = 58/61, col = "red", pch = 19)
```




```r
# ROC Curve
roc_curve <- roc(test$dummy_diagnosis, pred_prob)
plot(roc_curve, main = "ROC Curve for AIC Model", col = "blue", lwd = 2)
```




```
# AUC
auc(roc_obj) # 0.9657
```

```
## Area under the curve: 0.9657
```

## Desicion Tree

```
# Split data into training and testing sets
set.seed(333)

data_split <- knee_model |> initial_split(prop = 0.7)
train_data <- data_split |> training()
test_data <- data_split |> testing()

# Fit the Decision Tree Model
tree_model <- rpart(dummy_diagnosis ~ ., data = train_data, method = "class")

# Predict on the test set
tree_pred <- predict(tree_model, test_data, type = "class")
tree_pred_prob <- predict(tree_model, test_data, type = "prob")

# Confusion Matrix
conf_matrix <- table(test_data$dummy_diagnosis, tree_pred)
```

```r
addmargins(conf_matrix)
```

```
##      tree_pred
##        0  1 Sum
##   0    4  7  11
##   1    2 59  61
##   Sum  6 66  72
```

```r
# Calculate Accuracy, Sensitivity, and Specificity

(4 + 59)/72 # 0.875
```

```
## [1] 0.875
```

```r
59/66 # 0.8939394
```

```
## [1] 0.8939394
```

```r
4/6 # 0.6666667
```

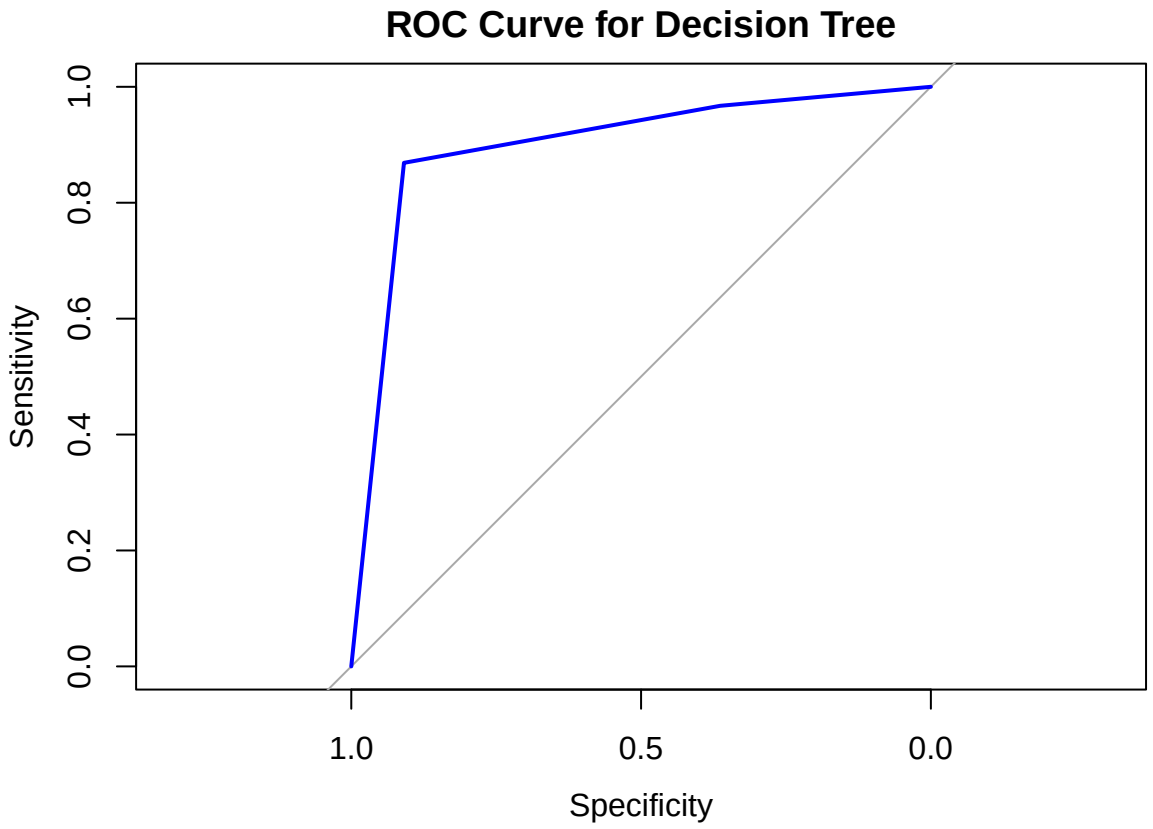
```
## [1] 0.6666667
```

```r
# ROC Curve
roc_curve <- roc(test_data$dummy_diagnosis, tree_pred_prob[,2])
```

```
## Setting levels: control = 0, case = 1
```

```
## Setting direction: controls < cases
```

```r
plot(roc_curve, main = "ROC Curve for Decision Tree", col = "blue", lwd = 2)
```



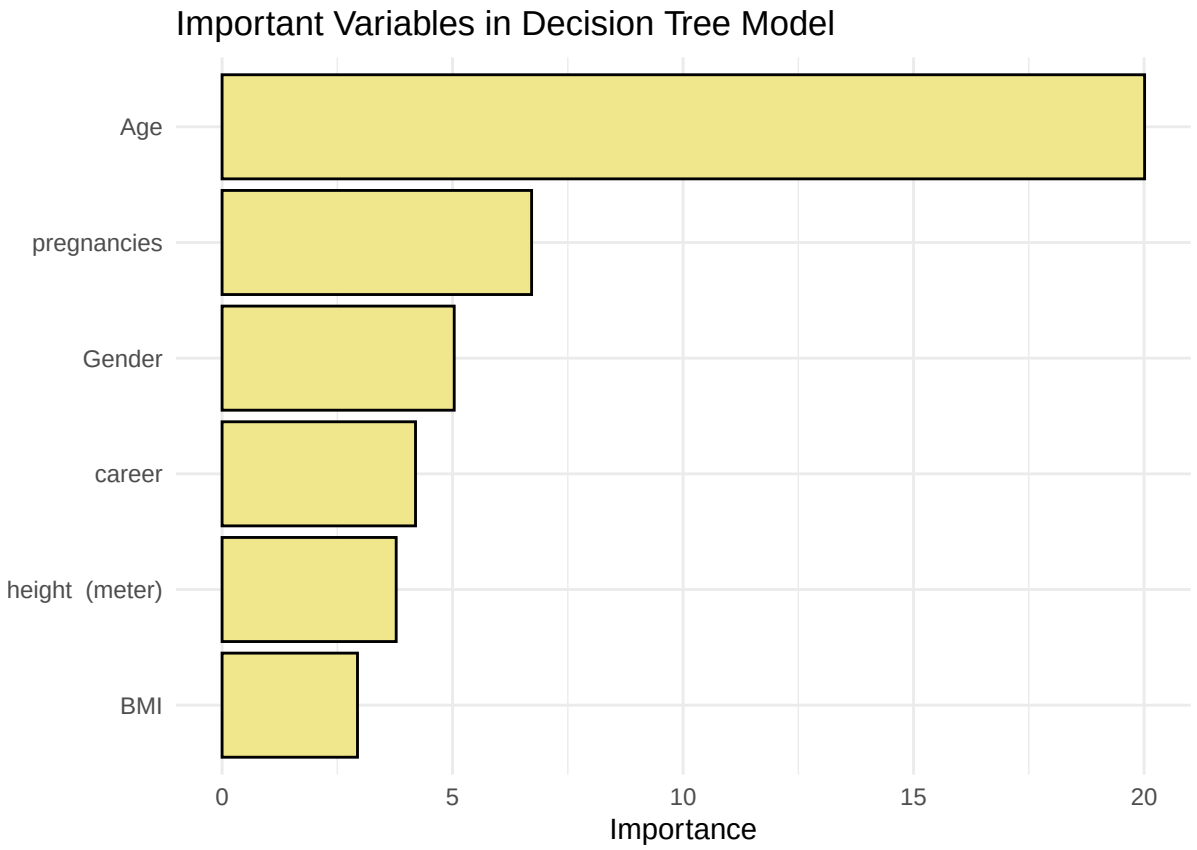

```
# Calculate AUC (Area Under the Curve)
auc_value <- auc(roc_curve)
auc_value #  0.8979
```

```
## Area under the curve: 0.8979
```

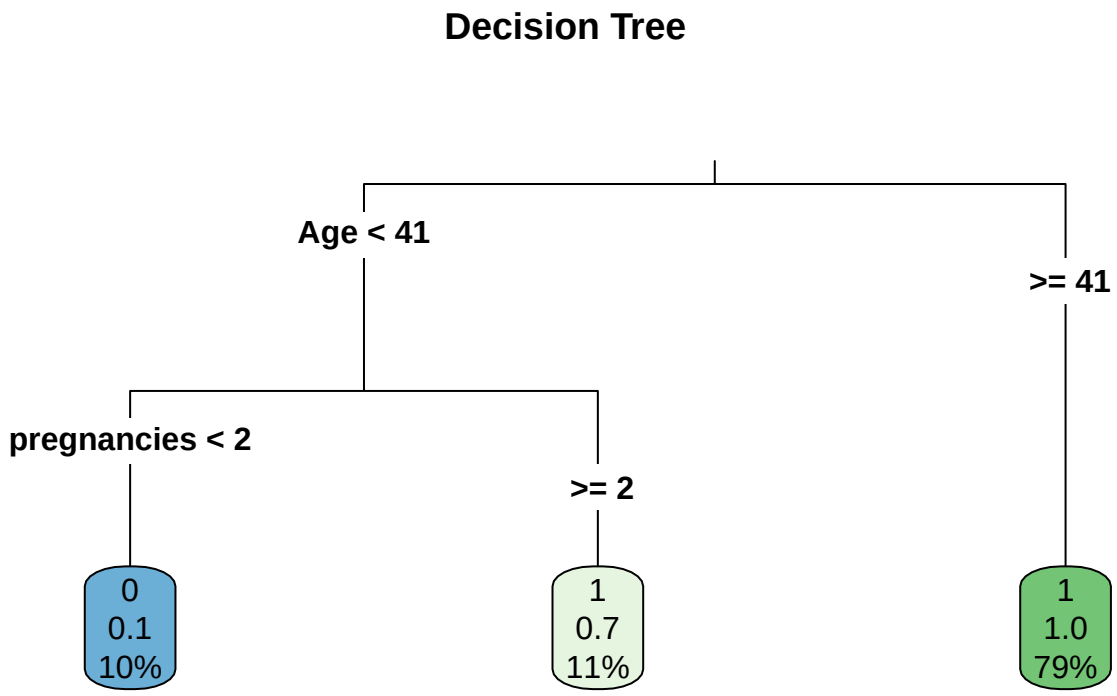
```
# Variable Importance
var_imp <- tree_model$variable.importance
print(var_imp)
```

```
##             Age     pregnancies          Gender          career height  (meter)
##       20.012427        6.714474        5.035855        4.196546        3.776891
##             BMI
##        2.937582
```

```
v_plot <- vip(tree_model) + geom_col(fill = "khaki", col = "black") +
  theme_minimal() +
  labs(title = "Important Variables in Decision Tree Model")
v_plot
```




```
# Plot the Decision Tree
rpart.plot(tree_model, type = 3, main = "Decision Tree", digits = 1)
```



```r
knee_tree <- knee_clean |>
  select(-dummy_diagnosis, -dialysis,
         -`T-score Value`, -`Z-Score Value`) |> na.omit()
# Split data into training and testing sets
set.seed(333)

data_split1 <- knee_tree |> initial_split(prop = 0.7)
train_data1 <- data_split1 |> training()
test_data1 <- data_split1 |> testing()

# Fit the Decision Tree Model
tree_model1 <- rpart(Diagnosis ~ ., data = train_data1, method = "class")

# Predict on the test set
tree_pred1 <- predict(tree_model1, test_data1, type = "class")
tree_pred_prob1 <- predict(tree_model1, test_data1, type = "prob")

# Confusion Matrix
conf_matrix1 <- table(test_data1$Diagnosis, tree_pred1)

addmargins(conf_matrix1)
```

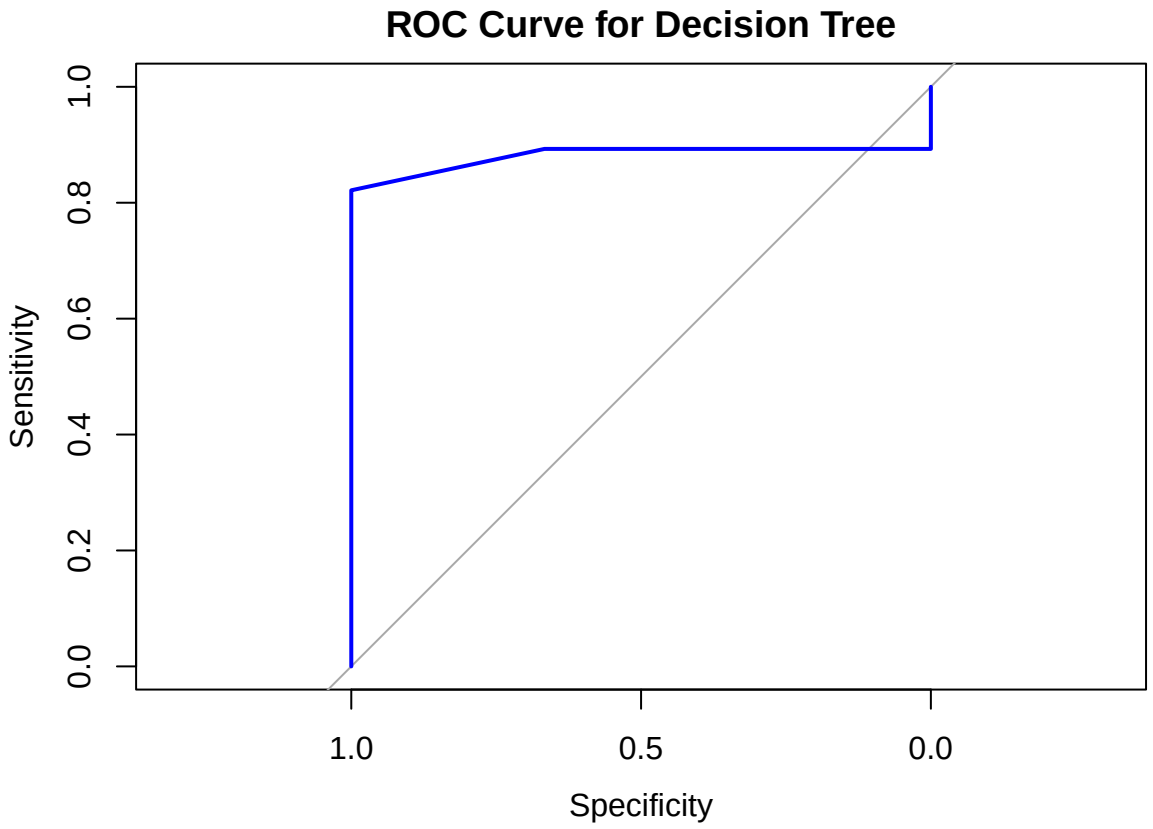
```
##               tree_pred1
##                normal osteopenia osteoporosis Sum
##   normal            4          2            0   6
##   osteopenia        0         25            3  28
##   osteoporosis      0          3            3   6
##   Sum               4         30            6  40
```

```r
# ROC Curve
roc_curve1 <- roc(test_data1$Diagnosis, tree_pred_prob1[,2])
```

```
## Setting levels: control = normal, case = osteopenia
```

```
## Setting direction: controls < cases
```

```r
plot(roc_curve1, main = "ROC Curve for Decision Tree", col = "blue", lwd = 2)
```



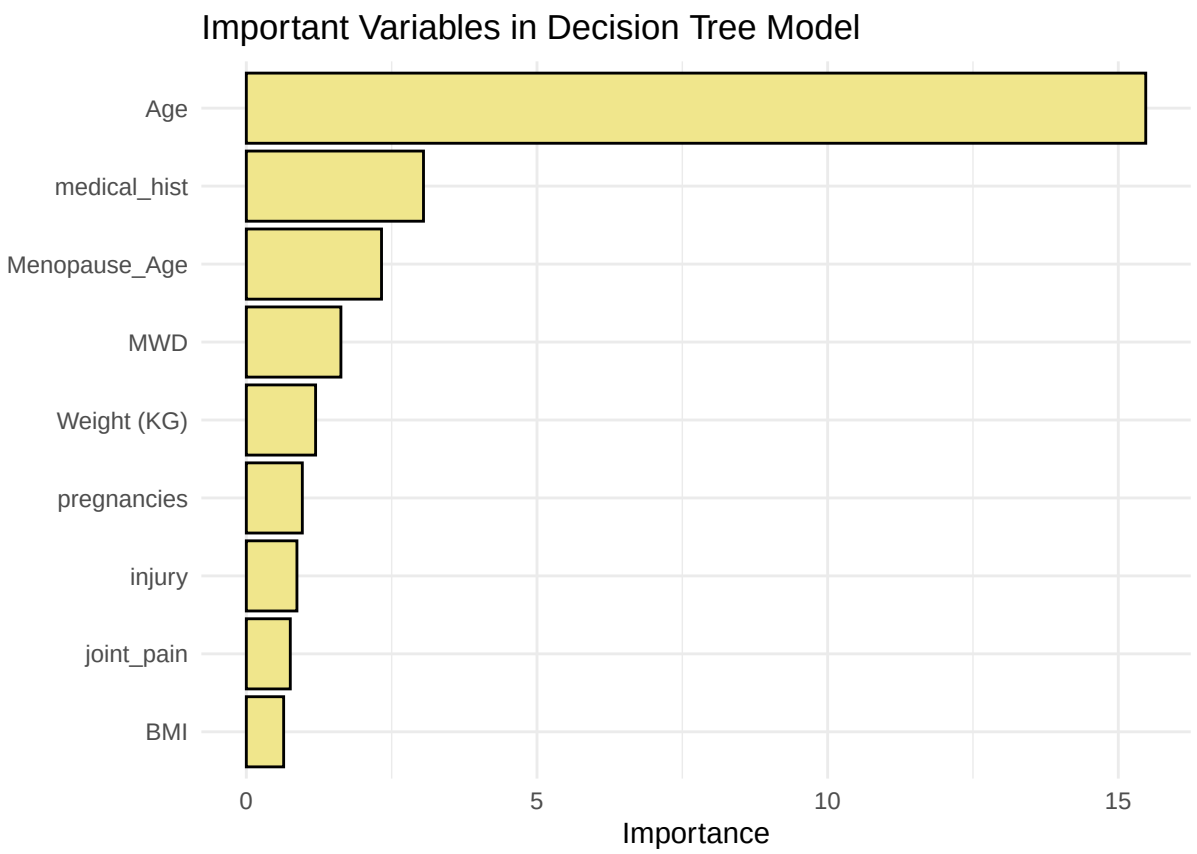

```
# Calculate AUC (Area Under the Curve)
auc_value1 <- auc(roc_curve1)
auc_value1 # 0.881
```

```
## Area under the curve: 0.881
```

```
# Variable Importance
var_imp1 <- tree_model1$variable.importance
print(var_imp1)
```

```
##           Age  medical_hist Menopause_Age           MWD   Weight (KG)
##    15.4722251     3.0476190     2.3256396     1.6272118     1.1918377
##   pregnancies        injury    joint_pain           BMI
##     0.9632682     0.8707483     0.7564635     0.6421788
```

```
v_plot1 <- vip(tree_model1) + geom_col(fill = "khaki", col = "black") +
  theme_minimal() +
  labs(title = "Important Variables in Decision Tree Model")
v_plot1
```




```r
# Plot the Decision Tree
rpart.plot(tree_model1, type = 3, main = "Decision Tree", digits = 1)
```