# Project

Jessica, Rafia, Caleb

2024-04-13

# OJECTIVE 1:

• 1. Perform a multiple logistic regression analysis and provide interpretation of the regression coefficients including hypothesis testing, and confidence intervals.

• 2. Do not need to include interactions with this model.

• 3. Comment on the practical vs statistical significance of the deemed important factors.

• 4. This does not have to be extremely fancy in terms of the model building approach, let EDA, feature selection, and/or overall intuition guide you.

• 6. The goal is the best model, not the fanciest. Note the difference between a model that is interpretable versus a model that is complex. (A model with a lot of predictors can still be interpretable) I want interpretable, explainable and one that makes accurate predictions.

• 7. Interactions models should NOT be used here. I need to display my ability to clearly interpret the regression coefficients. Effects plots may be used in addition too but not at the exclusions of coefficient interpretation.

```
# Load necessary libraries
library(dplyr)
```

```
##
## Attaching package: 'dplyr'
## The following objects are masked from 'package:stats':
##
##     filter, lag
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
library(gt)
```

```
## Warning: package 'gt' was built under R version 4.3.3
```

```
library(caret)
```

```
## Loading required package: ggplot2
## Loading required package: lattice
```

```
library(ggplot2)
```

```r
library(lattice)
library(aplore3)
```

```
## Warning: package 'aplore3' was built under R version 4.3.3
```

```r
library(tidyr)
library(tibble)
library(kableExtra)
```

```
## Warning: package 'kableExtra' was built under R version 4.3.3
##
## Attaching package: 'kableExtra'
## The following object is masked from 'package:dplyr':
##
##     group_rows
```

```r
library(effects)
```

```
## Warning: package 'effects' was built under R version 4.3.3
## Loading required package: carData
## Use the command
##     lattice::trellis.par.set(effectsTheme())
##   to customize lattice options for effects plots.
## See ?effectsTheme for details.
```

```r
library(MLmetrics)
```

```
## Warning: package 'MLmetrics' was built under R version 4.3.3
##
## Attaching package: 'MLmetrics'
## The following objects are masked from 'package:caret':
##
##     MAE, RMSE
## The following object is masked from 'package:base':
##
##     Recall
```

```r
# Load and summarize the dataset
data("glow_bonemed")
summary(glow_bonemed)
```

```
##      sub_id         site_id          phy_id       priorfrac      age
##  Min.   :  1.0   Min.   :1.000   Min.   :  1.00   No :374   Min.   :55.00
```

```
##  1st Qu.:125.8   1st Qu.:2.000   1st Qu.: 57.75   Yes:126   1st Qu.:61.00
##  Median :250.5   Median :3.000   Median :182.50             Median :67.00
##  Mean   :250.5   Mean   :3.436   Mean   :178.55             Mean   :68.56
##  3rd Qu.:375.2   3rd Qu.:5.000   3rd Qu.:298.00             3rd Qu.:76.00
##  Max.   :500.0   Max.   :6.000   Max.   :325.00             Max.   :90.00
##      weight           height           bmi        premeno   momfrac
armassist
##  Min.   : 39.90   Min.   :134.0   Min.   :14.88   No :403   No :435   No
:312
##  1st Qu.: 59.90   1st Qu.:157.0   1st Qu.:23.27   Yes: 97   Yes: 65
Yes:188
##  Median : 68.00   Median :161.5   Median :26.42
##  Mean   : 71.82   Mean   :161.4   Mean   :27.55
##  3rd Qu.: 81.30   3rd Qu.:165.0   3rd Qu.:30.79
##  Max.   :127.00   Max.   :199.0   Max.   :49.08
##  smoke        raterisk     fracscore      fracture  bonemed   bonemed_fu
##  No :465   Less   :167   Min.   : 0.000   No :375   No :371   No :361
##  Yes: 35   Same   :186   1st Qu.: 2.000   Yes:125   Yes:129   Yes:139
##            Greater:147   Median : 3.000
##                          Mean   : 3.698
##                          3rd Qu.: 5.000
##                          Max.   :11.000
##  bonetreat
##  No :382
##  Yes:118
##
##
##
##
```

```
head(glow_bonemed)
```

```
##   sub_id site_id phy_id priorfrac age weight height      bmi premeno
momfrac
## 1      1       1     14        No  62   70.3    158 28.16055      No
No
## 2      2       4    284        No  65   87.1    160 34.02344      No
No
```

```
## 3      3       6    305       Yes  88   50.8    157 20.60936      No
Yes
## 4      4       6    309        No  82   62.1    160 24.25781      No
No
## 5      5       1     37        No  61   68.0    152 29.43213      No
No
## 6      6       5    299       Yes  67   68.0    161 26.23356      No
No
##   armassist smoke raterisk fracscore fracture bonemed bonemed_fu bonetreat
## 1        No    No     Same         1       No      No         No        No
## 2        No    No     Same         2       No      No         No        No
## 3       Yes    No     Less        11       No      No         No        No
## 4        No    No     Less         5       No      No         No        No
## 5        No    No     Same         1       No      No         No        No
## 6        No   Yes     Same         4       No      No         No        No
```

```
str(glow_bonemed)
```

```
## 'data.frame':    500 obs. of  18 variables:
##  $ sub_id    : int  1 2 3 4 5 6 7 8 9 10 ...
##  $ site_id   : int  1 4 6 6 1 5 5 1 1 4 ...
##  $ phy_id    : int  14 284 305 309 37 299 302 36 8 282 ...
##  $ priorfrac : Factor w/ 2 levels "No","Yes": 1 1 2 1 1 2 1 2 2 1 ...
##  $ age       : int  62 65 88 82 61 67 84 82 86 58 ...
##  $ weight    : num  70.3 87.1 50.8 62.1 68 68 50.8 40.8 62.6 63.5 ...
##  $ height    : int  158 160 157 160 152 161 150 153 156 166 ...
##  $ bmi       : num  28.2 34 20.6 24.3 29.4 ...
##  $ premeno   : Factor w/ 2 levels "No","Yes": 1 1 1 1 1 1 1 1 1 1 ...
##  $ momfrac   : Factor w/ 2 levels "No","Yes": 1 1 2 1 1 1 1 1 1 1 ...
##  $ armassist : Factor w/ 2 levels "No","Yes": 1 1 2 1 1 1 1 1 1 1 ...
##  $ smoke     : Factor w/ 2 levels "No","Yes": 1 1 1 1 1 2 1 1 1 1 ...
##  $ raterisk  : Factor w/ 3 levels "Less","Same",..: 2 2 1 1 2 2 1 2 2 1
...
##  $ fracscore : int  1 2 11 5 1 4 6 7 7 0 ...
##  $ fracture  : Factor w/ 2 levels "No","Yes": 1 1 1 1 1 1 1 1 1 1 ...
##  $ bonemed   : Factor w/ 2 levels "No","Yes": 1 1 1 1 1 1 1 2 1 1 ...
##  $ bonemed_fu: Factor w/ 2 levels "No","Yes": 1 1 1 1 1 1 1 2 1 1 ...
##  $ bonetreat : Factor w/ 2 levels "No","Yes": 1 1 1 1 1 1 1 2 1 1 ...
```

Data Format: A data.frame with 500 rows and 18 variables such as:

priorfrac - If the patient previously had a fracture
age
weight
height
bmi
premeno
momfrac
armassist
smoke
raterisk
fracscore
fracture
bonemed - Bone medications at enrollment (1: No, 2: Yes)
bonemed_fu - Bone medications at follow-up (1: No, 2: Yes)
bonetreat - Bone medications both at enrollment and follow-up (1: No, 2: Yes)

Summary statistics for numeric variables

```
summary = glow_bonemed %>%
  select(age, weight, height, bmi, fracscore) %>%
  summarise_each(
    funs(min = min,
    q25 = quantile(., 0.25),
    median = median,
    q75 = quantile(., 0.75),
    max = max,
    mean = mean,
    sd = sd,
    variance= var))
# reshape it using tidyr functions
clean.summary = summary %>%
  gather(stat, val) %>%
  separate(stat, into = c("var", "stat"), sep = "_") %>%
  spread(stat, val) %>%
  select(var, min, max, mean, sd, variance)
print(clean.summary)
```

```
##         var       min       max      mean        sd   variance
## 1       age  55.00000  90.00000  68.56200  8.989537  80.811780
## 2       bmi  14.87637  49.08241  27.55303  5.973958  35.688178
## 3 fracscore   0.00000  11.00000   3.69800  2.495446   6.227251
```

```
## 4    height 134.00000 199.00000 161.36400  6.355493  40.392289
## 5    weight  39.90000 127.00000  71.82320 16.435992 270.141825
```

Summary statistics for categorical variables

```
summary(glow_bonemed %>% select(priorfrac, premeno, momfrac, armassist,
smoke, raterisk, fracture, bonemed, bonemed_fu, bonetreat))
```

```
##  priorfrac premeno   momfrac   armassist smoke        raterisk   fracture
##  No :374   No :403   No :435   No :312   No :465   Less   :167   No :375
##  Yes:126   Yes: 97   Yes: 65   Yes:188   Yes: 35   Same   :186   Yes:125
##                                                   Greater:147
##  bonemed   bonemed_fu bonetreat
##  No :371   No :361    No :382
##  Yes:129   Yes:139    Yes:118
##
```

No missing values

```
colSums(is.na(glow_bonemed))
```

```
##     sub_id    site_id     phy_id  priorfrac        age     weight
height
##          0          0          0          0          0          0
0
##        bmi    premeno    momfrac  armassist      smoke   raterisk
fracscore
##          0          0          0          0          0          0
0
##   fracture    bonemed bonemed_fu  bonetreat
##          0          0          0          0
```

```
sum(is.na(glow_bonemed))
```

```
## [1] 0
```

```
library(kableExtra)
colSums(is.na(glow_bonemed)) %>%
  kable("html", caption = "No missing values") %>%
  kable_styling()
```

No missing values

| |
|---|
| sub_id |
| site_id |
| phy_id |
| priorfrac |
| age |
| weight |
| height |
| bmi |
| premeno |
| momfrac |
| armassist |
| smoke |
| raterisk |
| fracscore |

No missing values

| |
|---|
| fracture |
| bonemed |
| bonemed_fu |
| bonetreat |

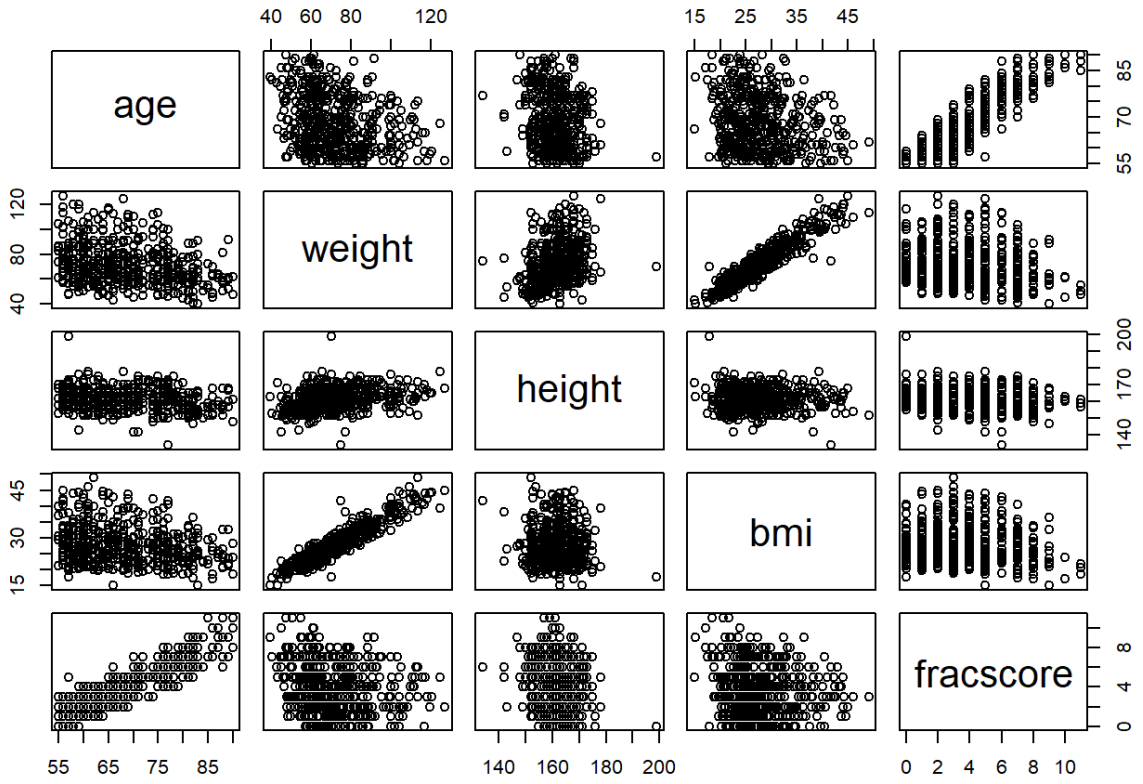
Age vs Weight: As weight increases the average age decreases
Age vs Height: Weak correlation of as height increases age decreases
Age vs BMI: As bmi increases the average age decreases
Age vs fracscore: As age increases the average fracscore increases

Weight vs Height: As height increases the average weight increases
Weight vs BMI: As bmi increases the average weight increases
Weight vs fracscore: As fracscore increases the average Weight decreases

Height vs BMI: As bmi increases the average height and variance stay the same
Height vs fracscore: As fracscore increases the average height stays the same though variance might decrease

BMI vs fracscore: As fracscore increases the average bmi decreases

```
plot(glow_bonemed[, c(5:8, 14)])
```




```
set.seed(123) # Setting seed for reproducibility


# Splitting the data
split_index <- createDataPartition(glow_bonemed$fracture, p = 0.7, list =
FALSE)
training_set <- glow_bonemed[split_index, ]
validation_set <- glow_bonemed[-split_index, ]
nrow(training_set)
## [1] 351
nrow(validation_set)
## [1] 149
```

# Perform Multiplle Logistic Reression Model

```
# Fit the logistic regression model
logistic_model <- glm(fracture ~ age + bmi + priorfrac, data = training_set,
family = binomial())
```

```
# Get a tidy summary of the model
model_summary <- broom::tidy(logistic_model)
print(model_summary)
```

```
## # A tibble: 4 × 5
##   term         estimate std.error statistic  p.value
##   <chr>           <dbl>     <dbl>     <dbl>    <dbl>
## 1 (Intercept)   -5.17      1.40      -3.70  0.000219
## 2 age            0.0464    0.0154     3.01  0.00264
## 3 bmi            0.0209    0.0235     0.892 0.373
## 4 priorfracYes   1.04      0.293      3.55  0.000378
```

# Interpretation:

1. **(Intercept)**: The log odds of experiencing a fracture when all explanatory variables (`age`, `bmi`, and `priorfrac`) are zero is -5.17. This coefficient is statistically significant with a p-value of 0.000219.
2. **age**: For each additional year of age, the log odds of having a fracture increases by 0.0464. This effect is statistically significant (p-value = 0.00264), suggesting that as age increases, the probability of having a fracture also increases.
3. **bmi**: For each unit increase in BMI, the log odds of having a fracture increases by 0.0209. However, this predictor is not statistically significant (p-value = 0.373), indicating that within the context of this model, BMI may not be a significant predictor of fracture.
4. **priorfracYes**: If a person has had a prior fracture (`priorfrac` being “Yes”), the log odds of having another fracture is increased by 1.04. This effect is statistically significant (p-value = 0.000378), strongly suggesting that a history of fractures is a significant predictor of future fractures.

# Calculate Odds Ratios & Confidence Intervals

```
# Calculate odds ratios
odds_ratios <- exp(coef(logistic_model))
odds_ratios
```

```
##  (Intercept)          age          bmi priorfracYes
##  0.005686664  1.047481878  1.021167839  2.829140167
```

```
# Calculate confidence intervals for the coefficients
conf_int <- exp(confint(logistic_model))
```

```
## Waiting for profiling to be done...
```

```
conf_int
```

```
##                      2.5 %     97.5 %
## (Intercept)  0.0003451074 0.08426513
## age          1.0165116691 1.08008105
## bmi          0.9746803726 1.06906086
## priorfracYes 1.5905727450 5.02135005
```

# Interpretation:

## Odds Ratios

1. **(Intercept)**: The odds of experiencing a fracture when all explanatory variables are zero is about 0.0057. This is typically not interpretable without context because age, bmi, and prior fractures cannot realistically be zero.
2. **Age**: The odds of having a fracture increase by about 4.7% for each additional year of age (`odds ratio = 1.047`). This is a small but statistically significant effect.
3. **BMI**: The odds of having a fracture increase by about 2.1% for each unit increase in BMI (`odds ratio = 1.021`). This effect is not statistically significant, as seen in the logistic regression output, which means changes in BMI do not significantly affect the odds of fracture in the context of this model.
4. **Prior Fracture (Yes)**: The odds of having a fracture are about 2.83 times higher for individuals who have had a prior fracture compared to those who have not. This is a significant and substantial effect.

## Confidence Intervals

1. **(Intercept)**: The 95% confidence interval ranges from approximately 0.000345 to 0.084, indicating high uncertainty when all predictors are zero, which is a non-practical scenario.
2. **Age**: The confidence interval ranges from approximately 1.017 to 1.080, suggesting a relatively consistent and significant effect of age across various scenarios.
3. **BMI**: The confidence interval ranges from approximately 0.975 to 1.069, including 1. This confirms that the effect of BMI is not statistically significant, as it includes the possibility of no effect (odds ratio = 1).
4. **Prior Fracture (Yes)**: The confidence interval ranges from approximately 1.59 to 5.02, indicating a significant effect. The interval does not include 1, reinforcing that having a prior fracture is a strong predictor of subsequent fractures.

## Conclusion

The results confirm that age and prior fractures are significant predictors of fractures, with prior fractures showing a particularly strong effect. BMI, while showing a positive association, does not significantly affect fracture odds in this model.

# Evaluate Model Performance

## Calculate Predicted Probabilities

```
# Assuming you've already loaded the necessary libraries and fitted your
logistic_model


# Step 1: Calculate predicted probabilities

validation_set$predicted_probabilities <- predict(logistic_model,
newdata=validation_set, type="response")


# Step 2: Convert probabilities to binary classifications

validation_set$predicted_classes <-
ifelse(validation_set$predicted_probabilities > 0.5, "Yes", "No")


# Step 3: Convert binary classifications to a factor

# Ensure that the factor levels match those of the actual outcome

validation_set$predicted_classes <- factor(validation_set$predicted_classes,
levels = c("No", "Yes"))


# Step 4: Calculate the F1 Score

# Load the MLmetrics package for F1 score calculation

library(MLmetrics)


# Calculate F1 Score using predicted classes and actual fracture outcomes

# Here we're using the MLmetrics package's F1_Score function

f1_score <- F1_Score(y_pred = validation_set$predicted_classes, y_true =
validation_set$fracture)

print(f1_score)

## [1] 0.8148148
```

## Interpreting F1 Score:

The F1 score combines precision and recall into a single metric by taking their harmonic mean, and it ranges from 0 to 1, where 1 indicates perfect precision and recall.

With an F1 score of approximately 0.815, our model demonstrates a strong balance between precision (the ability to correctly identify those with fractures) and recall (the ability to find all the fracture cases within the dataset).

# Implications of a High F1 Score:

- **Model Performance**: A high F1 score suggests that our model has a good balance between precision and recall, making it reliable for predicting fractures when applied to similar data.
- **Clinical Usefulness**: In a clinical setting, this model could potentially be used as a tool to identify individuals at high risk of fractures, allowing for targeted interventions that could prevent injuries.
- **Confidence in Predictions**: Healthcare providers might have higher confidence in the model's predictions, which can be crucial when making decisions about patient care.

```
# Compute Confusion Matrix

# Ensure that the actual outcome 'fracture' is a factor with the correct
levels

validation_set$fracture <- factor(validation_set$fracture, levels = c("No",
"Yes"))


# Compute the confusion matrix

confusion_matrix <-
caret::confusionMatrix(as.factor(validation_set$predicted_class),
validation_set$fracture)


# Print the confusion matrix

print(confusion_matrix)
```

```
## Confusion Matrix and Statistics

##

##           Reference

## Prediction No Yes

##        No  99  32

##        Yes 13   5

##

##                Accuracy : 0.698

##                  95% CI : (0.6175, 0.7704)

##     No Information Rate : 0.7517

##     P-Value [Acc > NIR] : 0.94404

##

##                   Kappa : 0.023

##

##  Mcnemar's Test P-Value : 0.00729
```

```
##
##             Sensitivity : 0.8839
##             Specificity : 0.1351
##          Pos Pred Value : 0.7557
##          Neg Pred Value : 0.2778
##              Prevalence : 0.7517
##          Detection Rate : 0.6644
##    Detection Prevalence : 0.8792
##       Balanced Accuracy : 0.5095
##
##        'Positive' Class : No
##
```

```
# Print additional statistics such as accuracy directly
print(confusion_matrix$overall['Accuracy'])
```

```
##  Accuracy
## 0.6979866
```

# Understanding the Confusion Matrix and Statistics

1. **Confusion Matrix:**
   - **True Negatives (TN)**: 99 - The model correctly predicted 'No fracture' 99 times.
   - **False Positives (FP)**: 32 - The model incorrectly predicted 'fracture' when there was none.
   - **False Negatives (FN)**: 13 - The model incorrectly predicted 'No fracture' when there was a fracture.
   - **True Positives (TP)**: 5 - The model correctly predicted 'fracture' 5 times.
2. **Accuracy:** 0.698 (69.8%)
   - This metric tells us that about 69.8% of all predictions were correct.
3. **95% CI for Accuracy:** (0.6175, 0.7704)
   - This interval indicates where the true accuracy of the model lies with 95% confidence.
4. **No Information Rate:** 0.7517
   - This is the accuracy that would be achieved by always predicting the most frequent class. Since it's higher than the model accuracy, it suggests the model is not very effective compared to a naive baseline.
5. **Kappa:** 0.023
   - Cohen's Kappa measures the agreement between prediction and reality, corrected for chance. A value close to 0 suggests minimal agreement beyond what would be expected by random chance.
6. **Mcnemar's Test:** P-value = 0.00729
   - This test checks the symmetry of the confusion matrix, specifically, it tests whether the row and column marginal frequencies are equal. The small p-value here suggests a significant difference between the FP and FN rates.

7. **Sensitivity (Recall) for 'No':** 0.8839
    - This is the proportion of actual negatives (no fracture) that were correctly identified, a high value.
8. **Specificity for 'No':** 0.1351
    - This is the proportion of actual positives (fracture) that were correctly identified as such, which is quite low.
9. **Positive Predictive Value (Precision) for 'No':** 0.7557
    - This is the probability that subjects with a positive screening test truly have the condition.
10. **Negative Predictive Value for 'No':** 0.2778
    - This is the probability that subjects with a negative screening test truly don't have the condition, which is quite low.
11. **Balanced Accuracy:** 0.5095
    - This is the average of recall obtained on each class. It's close to 0.5, indicating poor overall effectiveness of the model across both classes.

# Summary and Next Steps

The model shows reasonable accuracy, but it performs poorly in terms of balanced accuracy, specificity, and negative predictive value. This could be due to class imbalance or other factors influencing the model's ability to predict fractures effectively.

# Detailed Interpretation of Model Results

## Regression Coefficients Interpretation

1. **Age (Coefficient = 0.0464, p-value = 0.00264)**:
    - **Statistical Interpretation**: Each additional year of age increases the odds of having a fracture by approximately 4.7% when all other variables are held constant.
    - **Clinical Implications**: This finding suggests that age is a significant risk factor for fractures. In a clinical setting, older patients might require more frequent screening for osteoporosis or other bone health assessments. Programs aimed at preventing falls could be particularly important in older populations. Additionally, this might influence decisions regarding the timing of interventions such as calcium and vitamin D supplementation or more aggressive therapies like bisphosphonates.
2. **BMI (Coefficient = 0.0209, p-value = 0.373)**:
    - **Statistical Interpretation**: The coefficient for BMI indicates a slight increase in fracture risk with higher BMI, although this finding was not statistically significant.
    - **Clinical Implications**: While BMI is not a significant predictor in this model, the direction of the coefficient aligns with some clinical expectations where higher BMI could potentially protect against certain types of fractures due to increased bone mass. However, the lack of statistical significance suggests that BMI alone should not be used as a predictor for fracture risk without other clinical factors.
3. **Prior Fracture (Yes) (Coefficient = 1.04, p-value = 0.000378)**:
    - **Statistical Interpretation**: Individuals with a history of fractures are approximately 2.83 times more likely to experience another fracture compared to those without such a history.
    - **Clinical Implications**: This result has strong clinical implications. A history of fracture significantly increases the risk of subsequent fractures, indicating that patients with past fractures should be considered high-risk. This can influence clinical decisions such as more aggressive osteoporosis treatment and increased

monitoring. It also underscores the importance of secondary prevention strategies in individuals who have already experienced a fracture.

### Practical vs. Statistical Significance

- **Practical Significance**: While statistical significance indicates that we are confident the observed effects are not due to random chance, practical significance tells us whether these effects are large enough to matter in real-world applications.
    - For example, the age-related increase in fracture risk, while statistically significant, is relatively small per year. However, accumulated over many years, this risk becomes clinically significant, especially in the planning of long-term health interventions.
    - The effect of prior fractures is both statistically significant and practically significant due to the high odds ratio. This suggests that interventions targeting patients with a history of fractures could have substantial benefits in preventing further fractures.

### Implications for Public Health Policy and Medical Practice

- **Public Health Policy**: These findings could inform public health initiatives, such as campaigns to raise awareness about the importance of bone health in older adults and the development of community-based programs to reduce fall risk.
- **Medical Practice**: In clinical practice, these results could help in stratifying patients based on their fracture risk. High-risk patients (e.g., older adults and those with a history of fractures) can be targeted for earlier and more intensive interventions.
- **Research Directions**: Future research could explore more detailed interactions between these factors and other potential predictors not included in the model, such as genetic predispositions, lifestyle factors like physical activity and diet, and medication use that may affect bone density and fracture risk.

In summary, understanding the practical implications of our findings in terms of patient care, public health strategies, and future research directions is crucial for translating statistical results into effective clinical and public health interventions.

# Visualizations

```
library(pROC)
```

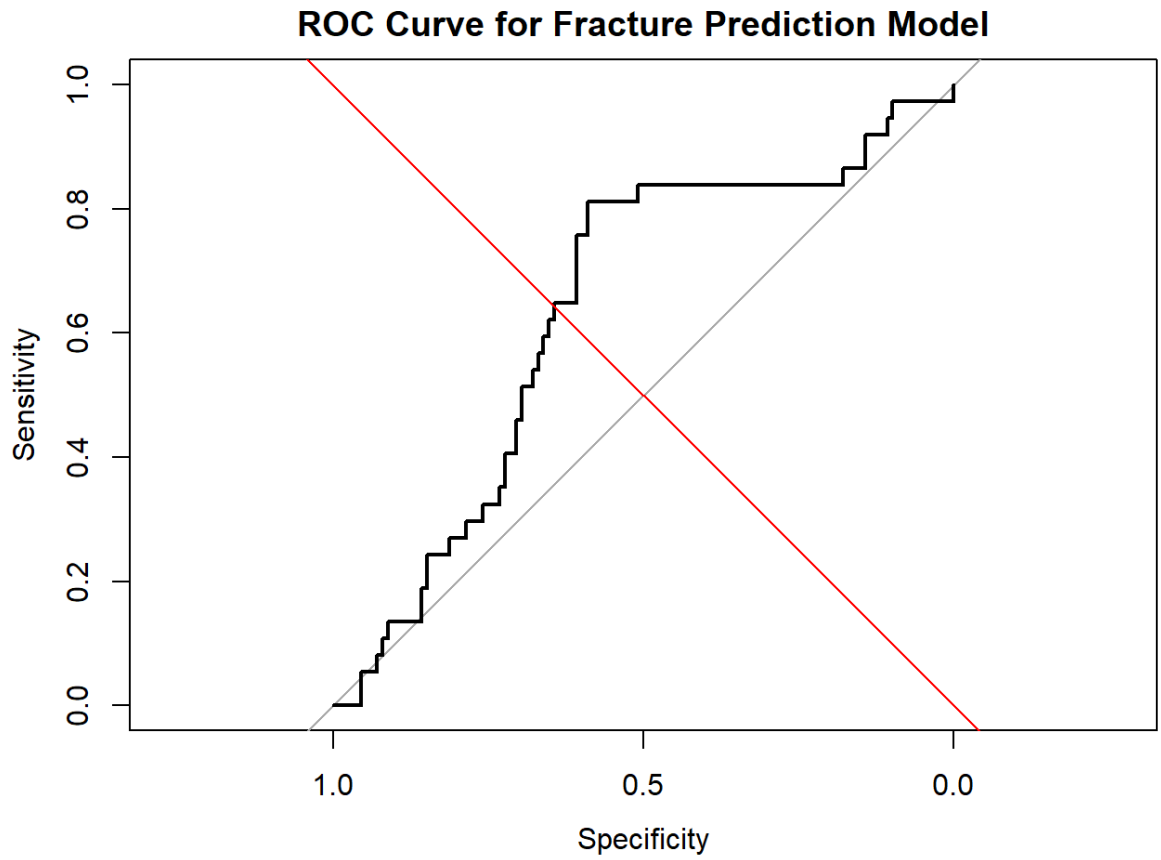
```
## Type 'citation("pROC")' for a citation.
##
## Attaching package: 'pROC'
## The following objects are masked from 'package:stats':
##
##     cov, smooth, var
```

```
# Calculate the ROC curve
roc_result <- roc(validation_set$fracture,
validation_set$predicted_probabilities)
```

```
## Setting levels: control = No, case = Yes
```

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

```
# Plot ROC curve
plot(roc_result, main="ROC Curve for Fracture Prediction Model")
abline(a=0, b=1, col="red")
```




```
# Calculate AUC
auc(roc_result)
```

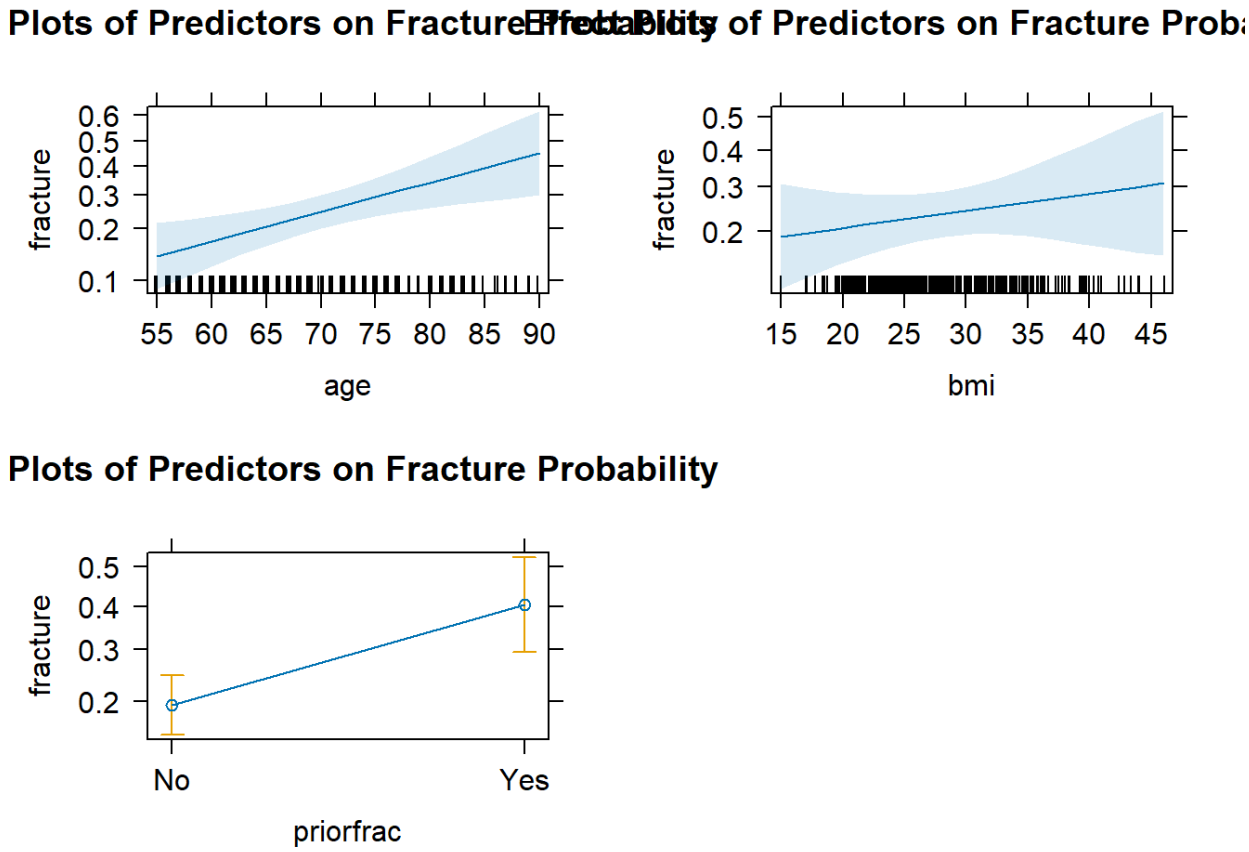
```
## Area under the curve: 0.6354
```

```
# Fit the logistic regression model (if not already fitted)
logistic_model <- glm(fracture ~ age + bmi + priorfrac, data = training_set,
family = binomial())


# Calculate effects
model_effects <- allEffects(logistic_model)


# Plot the effects
```

```
plot(model_effects, main = "Effect Plots of Predictors on Fracture
Probability")
```




# Effect Plots Interpretation

1. **Age**:
   - The plot shows an upward trend, indicating that the probability of fracture increases with age. This is consistent with clinical expectations since bone density can decrease and fracture risk may increase as people get older.
2. **BMI**:
   - The BMI plot appears to have a very slight upward slope, suggesting a small increase in fracture probability as BMI increases. However, as the effect of BMI was not statistically significant in the regression model, this plot reinforces the conclusion that BMI alone is not a strong predictor of fracture risk within the model's context.
3. **Prior Fracture (Yes/No)**:
   - The effect plot for prior fracture shows a marked increase in the probability of fracture for those with a history of fracture (Yes) compared to those without (No). This significant difference supports the conclusion that prior fractures are a strong predictor of future fracture risk, which is highly relevant for patient risk assessments and preventive healthcare strategies.

# ROC Curve Interpretation

The Area Under the Curve (AUC) for our fracture prediction model is 0.6354. This value falls between 0.5 (no better than random chance) and 1.0 (perfect classification), indicating that the model has a modest ability to differentiate between patients who will experience a fracture and those who will not.

- **Interpreting the AUC Value**: An AUC of 0.6354 suggests that the model correctly distinguishes between a true case (fracture) and a non-case (no fracture) approximately 63.54% of the time. While not close to perfect, this performance is markedly better than random guessing, which would be represented by an AUC of 0.5.
- **Model Performance**: This AUC value provides us with confidence that the model has learned some underlying patterns in the data that are predictive of fractures. However, there is room for improvement, and future iterations of the model could incorporate additional predictors, employ feature engineering, or explore more complex algorithms to enhance predictive accuracy.

## Practical Applications

Given the AUC value, the model can be tentatively used in clinical settings to prioritize patients for fracture risk assessment, but with caution due to its moderate performance. For example:

- **Risk Stratification**: Patients with a higher predicted probability might be targeted for bone density scanning or other assessments.
- **Preventative Measures**: These predictions could also be used to identify patients who would benefit from interventions such as lifestyle changes, supplements, or medications to strengthen bone density.

## Summary

The AUC value is a testament to the model's utility, yet it also signals the need for a cautious approach when applying these predictions in real-world settings. The value underlines the importance of continuing to refine our predictive models to better serve patients and healthcare professionals alike.

# Next Steps Optimitzed MLR Model

```
# Rename Columns and convert factors where needed
glow_bonemed_new <- glow_bonemed %>%
    rename(
        FRACTURE = fracture,
        AGE = age,
        HEIGHT = height,
        WEIGHT = weight,
        PREMENO = premeno,
        MOMFRAC = momfrac,
        RATERISK = raterisk,
```

```
        PRIORFRAC = priorfrac,
        ARMASSIST = armassist,
        SMOKE = smoke,
        BMI = bmi,
        BONEMED =bonemed,
        BONETREAT = bonetreat,
        BONEMED_FU = bonemed_fu,
        FRACSCORE = fracscore
    ) %>%
    mutate(
        PRIORFRAC = as.numeric(PRIORFRAC == "Yes"),
        ARMASSIST = as.numeric(ARMASSIST == "Yes"),
        MOMFRAC = as.numeric(MOMFRAC == "Yes"),
        SMOKE = as.numeric(SMOKE == "Yes"),
        FRACTURE = as.numeric(FRACTURE == "Yes"),
        RATERISK_num = as.numeric(RATERISK),
        # Create RATERISK_EQ_3 variable where '3' indicates "Greater than
others of the same age"
        # This assumes that 'RATERISK' is a factor and "Greater" is one of
the levels
        RATERISK_EQ_3 = as.numeric(RATERISK == "Greater"))


str(glow_bonemed_new)
## 'data.frame':    500 obs. of  20 variables:
##  $ sub_id       : int  1 2 3 4 5 6 7 8 9 10 ...
##  $ site_id      : int  1 4 6 6 1 5 5 1 1 4 ...
##  $ phy_id       : int  14 284 305 309 37 299 302 36 8 282 ...
##  $ PRIORFRAC    : num  0 0 1 0 0 1 0 1 1 0 ...
##  $ AGE          : int  62 65 88 82 61 67 84 82 86 58 ...
##  $ WEIGHT       : num  70.3 87.1 50.8 62.1 68 68 50.8 40.8 62.6 63.5 ...
##  $ HEIGHT       : int  158 160 157 160 152 161 150 153 156 166 ...
##  $ BMI          : num  28.2 34 20.6 24.3 29.4 ...
##  $ PREMENO      : Factor w/ 2 levels "No","Yes": 1 1 1 1 1 1 1 1 1 1 ...
##  $ MOMFRAC      : num  0 0 1 0 0 0 0 0 0 0 ...
##  $ ARMASSIST    : num  0 0 1 0 0 0 0 0 0 0 ...
```

```
##  $ SMOKE        : num  0 0 0 0 0 1 0 0 0 0 ...
##  $ RATERISK     : Factor w/ 3 levels "Less","Same",..: 2 2 1 1 2 2 1 2 2 1
...
##  $ FRACSCORE    : int  1 2 11 5 1 4 6 7 7 0 ...
##  $ FRACTURE     : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ BONEMED      : Factor w/ 2 levels "No","Yes": 1 1 1 1 1 1 1 2 1 1 ...
##  $ BONEMED_FU   : Factor w/ 2 levels "No","Yes": 1 1 1 1 1 1 1 2 1 1 ...
##  $ BONETREAT    : Factor w/ 2 levels "No","Yes": 1 1 1 1 1 1 1 2 1 1 ...
##  $ RATERISK_num : num  2 2 1 1 2 2 1 2 2 1 ...
##  $ RATERISK_EQ_3: num  0 0 0 0 0 0 0 0 0 0 ...
```

```
print(colnames(glow_bonemed_new))
```

```
##  [1] "sub_id"        "site_id"       "phy_id"        "PRIORFRAC"
##  [5] "AGE"           "WEIGHT"        "HEIGHT"        "BMI"
##  [9] "PREMENO"       "MOMFRAC"       "ARMASSIST"     "SMOKE"
## [13] "RATERISK"      "FRACSCORE"     "FRACTURE"      "BONEMED"
## [17] "BONEMED_FU"    "BONETREAT"     "RATERISK_num"  "RATERISK_EQ_3"
```

```
# Fit the logistic regression model using the updated data frame
logistic_model_new <- glm(FRACTURE ~ AGE + WEIGHT + HEIGHT + BMI + PREMENO +
MOMFRAC + ARMASSIST + SMOKE + RATERISK_num,
                          data = glow_bonemed_new, family = binomial())


# Output a summary of the logistic regression model
summary(logistic_model_new)
```

```
##
## Call:
## glm(formula = FRACTURE ~ AGE + WEIGHT + HEIGHT + BMI + PREMENO +
##     MOMFRAC + ARMASSIST + SMOKE + RATERISK_num, family = binomial(),
##     data = glow_bonemed_new)
##
## Coefficients:
##               Estimate Std. Error z value Pr(>|z|)
## (Intercept)  -14.11857   12.70467  -1.111  0.26644
## AGE            0.04916    0.01415   3.475  0.00051 ***
## WEIGHT        -0.10448    0.08672  -1.205  0.22827
## HEIGHT         0.04897    0.07820   0.626  0.53117
```

```
## BMI            0.29026    0.22384   1.297  0.19472
## PREMENOYes     0.12567    0.28278   0.444  0.65674
## MOMFRAC        0.62259    0.30400   2.048  0.04056 *
## ARMASSIST      0.40495    0.25273   1.602  0.10909
## SMOKE         -0.22062    0.45983  -0.480  0.63138
## RATERISK_num   0.45037    0.14574   3.090  0.00200 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 562.34  on 499  degrees of freedom
## Residual deviance: 511.01  on 490  degrees of freedom
## AIC: 531.01
##
## Number of Fisher Scoring iterations: 4
```