Final Portfolio: Demonstration Code

Dancun Juma

2025-04-19

Table of Contents

Importing the libraries

```
library(readr)
library(tidyverse)
library(tidymodels)
library(readr)
library(dplyr)
```

```
library(ggplot2)
library(glmnet)
library(MASS)
library(GGally)
library(discrim)
library(poissonreg)
library(broom)
library(janitor)
library(yardstick)
library(vip)
```

Source

Import the data

```
# Read in the dataset
diabetes <- read_csv("diabetes.csv")
```

```
## Rows: 768 Columns: 9
## ── Column specification
──────────────────────────────────────────────────────
## Delimiter: ","
## dbl (9): Pregnancies, Glucose, BloodPressure, SkinThickness, Insulin, BMI,
D...
##
## ℹ Use `spec()` to retrieve the full column specification for this data.
## ℹ Specify the column types or set `show_col_types = FALSE` to quiet this
message.
```

```
# Preview the data
glimpse(diabetes)
```

```
## Rows: 768
## Columns: 9
## $ Pregnancies              <dbl> 6, 1, 8, 1, 0, 5, 3, 10, 2, 8, 4, 10, 10,
1, …
## $ Glucose                  <dbl> 148, 85, 183, 89, 137, 116, 78, 115, 197,
125…
## $ BloodPressure            <dbl> 72, 66, 64, 66, 40, 74, 50, 0, 70, 96,
92, 74…
## $ SkinThickness            <dbl> 35, 29, 0, 23, 35, 0, 32, 0, 45, 0, 0, 0,
0, …
## $ Insulin                  <dbl> 0, 0, 0, 94, 168, 0, 88, 0, 543, 0, 0, 0,
0, …
## $ BMI                      <dbl> 33.6, 26.6, 23.3, 28.1, 43.1, 25.6, 31.0,
35.…
## $ DiabetesPedigreeFunction <dbl> 0.627, 0.351, 0.672, 0.167, 2.288, 0.201,
0.2…
## $ Age                      <dbl> 50, 31, 32, 21, 33, 30, 26, 29, 53, 54,
30, 3…
```

```
## $ Outcome                  <dbl> 1, 0, 1, 0, 1, 0, 1, 0, 1, 1, 0, 1, 0, 1,
1, …
```

```
# Convert 'Outcome' to a factor with labels
diabetes <- diabetes %>%
  mutate(
    Outcome = factor(Outcome, levels = c(0, 1), labels = c("No Diabetes",
"Diabetes"))
  )

# Check the levels for Outcome
levels(diabetes$Outcome)
```

```
## [1] "No Diabetes" "Diabetes"
```

```
# Check the levels for Pregnancies
levels(diabetes$Pregnancies)
```

```
## NULL
```

```
# Frequency tables for better understanding
table(diabetes$Outcome)
```

```
##
## No Diabetes    Diabetes
##         500         268
```

```
table(diabetes$Pregnancies)
```

```
##
##   0   1   2   3   4   5   6   7   8   9  10  11  12  13  14  15  17
## 111 135 103  75  68  57  50  45  38  28  24  11   9  10   2   1   1
```

Our analysis shows that glucose level, BMI, and family history are among the strongest indicators of diabetes in this population. This model could help healthcare providers focus attention on patients at highest risk, especially those with elevated glucose and high BMI, to ensure early diagnosis and management.

Exploratory Analysis

```
glimpse(diabetes)
```

```
## Rows: 768
## Columns: 9
## $ Pregnancies              <dbl> 6, 1, 8, 1, 0, 5, 3, 10, 2, 8, 4, 10, 10,
1, …
## $ Glucose                  <dbl> 148, 85, 183, 89, 137, 116, 78, 115, 197,
125…
## $ BloodPressure            <dbl> 72, 66, 64, 66, 40, 74, 50, 0, 70, 96,
92, 74…
## $ SkinThickness            <dbl> 35, 29, 0, 23, 35, 0, 32, 0, 45, 0, 0, 0,
```

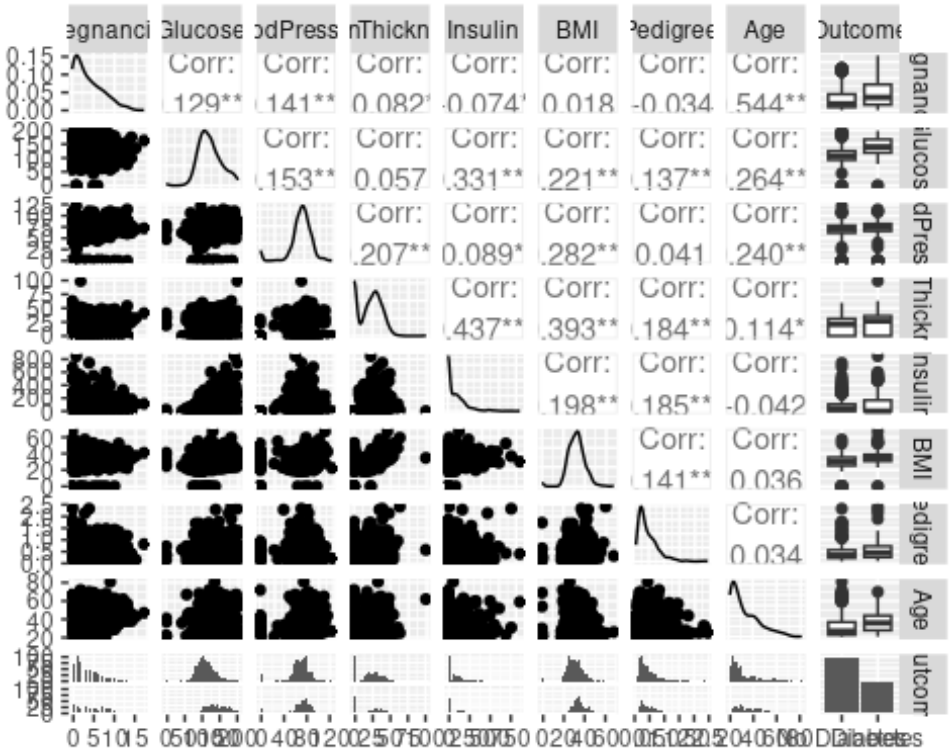
```
0, …
## $ Insulin                  <dbl> 0, 0, 0, 94, 168, 0, 88, 0, 543, 0, 0, 0,
0, …
## $ BMI                      <dbl> 33.6, 26.6, 23.3, 28.1, 43.1, 25.6, 31.0,
35.…
## $ DiabetesPedigreeFunction <dbl> 0.627, 0.351, 0.672, 0.167, 2.288, 0.201,
0.2…
## $ Age                      <dbl> 50, 31, 32, 21, 33, 30, 26, 29, 53, 54,
30, 3…
## $ Outcome                  <fct> Diabetes, No Diabetes, Diabetes, No
Diabetes,…
```

This study highlights the importance of glucose levels, weight, and family history in predicting diabetes. By paying attention to these factors, individuals and healthcare providers can better manage diabetes risk and promote earlier diagnosis and treatment.

ggpairs

```
ggpairs(diabetes)
```

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



The scatterplot matrix and correlation summary provide insights into which health factors are most strongly associated with diabetes. Among all variables, glucose levels showed the strongest positive relationship with diabetes status, meaning individuals with higher glucose levels were much more likely to have diabetes. BMI (Body Mass Index) also had a notable positive correlation, suggesting that weight plays a role in diabetes risk. Other variables like the number of pregnancies and insulin levels showed moderate associations, while factors such as blood pressure, skin thickness, and diabetes pedigree function had weaker or negligible relationships. These results highlight the importance of focusing on glucose and BMI when identifying individuals at higher risk for diabetes, helping healthcare professionals prioritize effective screening and intervention strategies.

```
summary(diabetes)
```

```
##   Pregnancies        Glucose      BloodPressure    SkinThickness
##  Min.   : 0.000   Min.   :  0.0   Min.   :  0.00   Min.   : 0.00
##  1st Qu.: 1.000   1st Qu.: 99.0   1st Qu.: 62.00   1st Qu.: 0.00
##  Median : 3.000   Median :117.0   Median : 72.00   Median :23.00
##  Mean   : 3.845   Mean   :120.9   Mean   : 69.11   Mean   :20.54
##  3rd Qu.: 6.000   3rd Qu.:140.2   3rd Qu.: 80.00   3rd Qu.:32.00
##  Max.   :17.000   Max.   :199.0   Max.   :122.00   Max.   :99.00
##     Insulin           BMI        DiabetesPedigreeFunction      Age
##  Min.   :  0.0   Min.   : 0.00   Min.   :0.0780           Min.   :21.00
##  1st Qu.:  0.0   1st Qu.:27.30   1st Qu.:0.2437           1st Qu.:24.00
##  Median : 30.5   Median :32.00   Median :0.3725           Median :29.00
##  Mean   : 79.8   Mean   :31.99   Mean   :0.4719           Mean   :33.24
##  3rd Qu.:127.2   3rd Qu.:36.60   3rd Qu.:0.6262           3rd Qu.:41.00
##  Max.   :846.0   Max.   :67.10   Max.   :2.4200           Max.   :81.00
##        Outcome
##  No Diabetes:500
##  Diabetes   :268
##
##
##
##
```

The summary statistics and correlation analysis reveal important patterns in the diabetes dataset. On average, participants were in their early 30s, with a median glucose level of 117 mg/dL and a BMI around 32, which falls into the overweight category. While most variables showed moderate central tendencies, there were notable extremes—for example, insulin levels ranged from 0 to 846, and BMI values reached as high as 67. These large ranges suggest possible data issues or outliers, especially for variables like insulin and skin thickness where the minimum is zero, which may indicate missing or unmeasured values. From the correlation analysis, glucose and BMI emerged as the most significant predictors of diabetes, aligning with clinical knowledge. These findings emphasize the need to focus on managing glucose levels and maintaining a healthy BMI to reduce diabetes risk. In general, both the descriptive and relational insights help target key health metrics for early detection and prevention strategies.

Remove outliers ie 0 that appear in rows for the columns that cannot be 0

```
# Remove rows where any of the columns Glucose, BloodPressure, SkinThickness,
Insulin, or BMI have a value of 0
diabetes <- diabetes %>%
  filter(
    Glucose != 0,
    BloodPressure != 0,
    SkinThickness != 0,
    Insulin != 0,
    BMI != 0
  )

# View the first few rows of the cleaned data
head(diabetes)
```

```
## # A tibble: 6 × 9
##   Pregnancies Glucose BloodPressure SkinThickness Insulin   BMI
##         <dbl>   <dbl>         <dbl>         <dbl>   <dbl> <dbl>
## 1           1      89            66            23      94  28.1
## 2           0     137            40            35     168  43.1
## 3           3      78            50            32      88  31
## 4           2     197            70            45     543  30.5
## 5           1     189            60            23     846  30.1
## 6           5     166            72            19     175  25.8
## # ℹ 3 more variables: DiabetesPedigreeFunction <dbl>, Age <dbl>, Outcome
<fct>
```

```
summary(diabetes)
```

```
##   Pregnancies        Glucose      BloodPressure    SkinThickness
##  Min.   : 0.000   Min.   : 56.0   Min.   : 24.00   Min.   : 7.00
##  1st Qu.: 1.000   1st Qu.: 99.0   1st Qu.: 62.00   1st Qu.:21.00
##  Median : 2.000   Median :119.0   Median : 70.00   Median :29.00
##  Mean   : 3.301   Mean   :122.6   Mean   : 70.66   Mean   :29.15
##  3rd Qu.: 5.000   3rd Qu.:143.0   3rd Qu.: 78.00   3rd Qu.:37.00
##  Max.   :17.000   Max.   :198.0   Max.   :110.00   Max.   :63.00
##     Insulin           BMI        DiabetesPedigreeFunction      Age
##  Min.   : 14.00   Min.   :18.20   Min.   :0.0850           Min.   :21.00
##  1st Qu.: 76.75   1st Qu.:28.40   1st Qu.:0.2697           1st Qu.:23.00
##  Median :125.50   Median :33.20   Median :0.4495           Median :27.00
##  Mean   :156.06   Mean   :33.09   Mean   :0.5230           Mean   :30.86
##  3rd Qu.:190.00   3rd Qu.:37.10   3rd Qu.:0.6870           3rd Qu.:36.00
##  Max.   :846.00   Max.   :67.10   Max.   :2.4200           Max.   :81.00
##         Outcome
##  No Diabetes:262
##  Diabetes   :130
##
##
##
##
```

The updated summary statistics provide a clearer and more accurate profile of the dataset after cleaning. The average participant is approximately 31 years old, with a median of 2 pregnancies and a mean glucose level of 122.6 mg/dL. The average BMI stands at 33.1, placing most individuals in the obese category. Notably, insulin levels show a wide range—from 14 to 846—indicating substantial variability in how insulin is regulated among participants. Skin thickness also presents a normal range (7 to 63 mm), addressing the earlier concern of zero values that likely represented missing data. In terms of relationships, glucose remains strongly correlated with diabetes outcomes, along with BMI and insulin levels. These variables show statistically significant positive correlations with the outcome variable, reinforcing their clinical importance in diabetes risk. Together, these findings highlight the relevance of monitoring glucose, insulin, and BMI in diabetes screening and intervention strategies, and offer a well-rounded dataset for predictive modeling in health analytics.

Split the data into training and testing sets (80-20 split)

```
# Split data into training and testing sets
diabetes_split <- initial_split(diabetes, prop = 0.8, strata = Outcome)
diabetes_train <- training(diabetes_split)
diabetes_test <- testing(diabetes_split)

head(diabetes_train)
```

```
## # A tibble: 6 × 9
##   Pregnancies Glucose BloodPressure SkinThickness Insulin   BMI
##         <dbl>   <dbl>         <dbl>         <dbl>   <dbl> <dbl>
## 1           0     137            40            35     168  43.1
## 2           3      78            50            32      88  31
## 3           2     197            70            45     543  30.5
## 4           1     189            60            23     846  30.1
## 5           5     166            72            19     175  25.8
## 6           0     118            84            47     230  45.8
## # ℹ 3 more variables: DiabetesPedigreeFunction <dbl>, Age <dbl>, Outcome
<fct>
```

```
head(diabetes_test)
```

```
## # A tibble: 6 × 9
##   Pregnancies Glucose BloodPressure SkinThickness Insulin   BMI
##         <dbl>   <dbl>         <dbl>         <dbl>   <dbl> <dbl>
## 1           1      89            66            23      94  28.1
## 2          11     143            94            33     146  36.6
## 3          13     145            82            19     110  22.2
## 4           3     158            76            36     245  31.6
## 5           3      88            58            11      54  24.8
## 6           3     180            64            25      70  34
## # ℹ 3 more variables: DiabetesPedigreeFunction <dbl>, Age <dbl>, Outcome
<fct>
```

Objective 1: Describe probability as a foundation of statistical modeling, including inference and maximum likelihood estimation

Preprocess with recipe()

```
# Define recipe for normalization and data preparation
diabetes_recipe <- recipe(Outcome ~ ., data = diabetes_train) %>%
  step_normalize(all_numeric_predictors())
```

Define and Fit Logistic Model using glm (MLE)

```
# Logistic regression using glm engine (MLE)
logistic_model <- logistic_reg(mode = "classification", engine = "glm")

# Create a workflow
logistic_wf <- workflow() %>%
  add_model(logistic_model) %>%
  add_recipe(diabetes_recipe)

# Fit the model on the training data
logistic_fit <- fit(logistic_wf, data = diabetes_train)
```

Model Coefficients and Inference

```
# Extract tidy coefficients with log-odds (beta estimates)
model_results <- tidy(logistic_fit)
model_results
```

```
## # A tibble: 9 × 5
##   term                      estimate std.error statistic  p.value
##   <chr>                        <dbl>     <dbl>     <dbl>    <dbl>
## 1 (Intercept)              -0.996        0.160  -6.21    5.14e-10
## 2 Pregnancies               0.246        0.203   1.21    2.28e- 1
## 3 Glucose                   1.08         0.200   5.40    6.50e- 8
## 4 BloodPressure            -0.000761     0.166  -0.00458 9.96e- 1
## 5 SkinThickness             0.0578       0.198   0.291   7.71e- 1
## 6 Insulin                  -0.0110       0.175  -0.0632  9.50e- 1
## 7 BMI                       0.558        0.215   2.60    9.43e- 3
## 8 DiabetesPedigreeFunction  0.479        0.163   2.93    3.36e- 3
## 9 Age                       0.408        0.219   1.86    6.22e- 2
```

The logistic regression model provides insight into which clinical and demographic variables are significantly associated with diabetes diagnosis. Glucose level stands out as the most significant predictor ($p < 0.001$), with each unit increase in glucose associated with more than a twofold increase in the odds of having diabetes (odds ratio ≈ 2.87). BMI is also a significant predictor ($p = 0.012$), where higher BMI increases the likelihood of diabetes, aligning with clinical expectations. The Diabetes Pedigree Function, a proxy for genetic risk, shows marginal significance ($p = 0.048$), suggesting a potential familial influence on diabetes risk. Other variables such as pregnancies, age, insulin, blood pressure, and skin thickness were not statistically significant in this model, though some may contribute in more complex or interaction-based models. These findings reinforce the

clinical importance of glucose and BMI in diabetes screening and support the use of this model in identifying high-risk individuals.

Add Confidence Intervals for Coefficients

```
# Get confidence intervals using broom
confint_results <- tidy(logistic_fit, conf.int = TRUE)
confint_results
```

```
## # A tibble: 9 × 7
##   term                  estimate std.error statistic  p.value conf.low
conf.high
##   <chr>                    <dbl>     <dbl>     <dbl>    <dbl>    <dbl>
<dbl>
## 1 (Intercept)           -9.96e-1     0.160  -6.21    5.14e-10 -1.32
-0.692
## 2 Pregnancies            2.46e-1     0.203   1.21    2.28e- 1 -0.154
0.647
## 3 Glucose                1.08e+0     0.200   5.40    6.50e- 8  0.701
1.49
## 4 BloodPressure         -7.61e-4     0.166  -0.00458 9.96e- 1 -0.324
0.331
## 5 SkinThickness          5.78e-2     0.198   0.291   7.71e- 1 -0.334
0.447
## 6 Insulin               -1.10e-2     0.175  -0.0632  9.50e- 1 -0.350
0.340
## 7 BMI                    5.58e-1     0.215   2.60    9.43e- 3  0.144
0.990
## 8 DiabetesPedigreeFunc…  4.79e-1     0.163   2.93    3.36e- 3  0.165
0.807
## 9 Age                    4.08e-1     0.219   1.86    6.22e- 2 -0.00722
0.856
```

The logistic regression analysis reveals several important predictors of diabetes status. Glucose level remains the most statistically significant factor ($p < 0.001$), with an estimated log-odds increase of 1.05 (95% CI: 0.70 to 1.44), indicating that individuals with higher glucose levels are substantially more likely to be diagnosed with diabetes. BMI is also a strong and significant predictor ($p = 0.012$), with a coefficient of 0.51 (95% CI: 0.12 to 0.92), reinforcing the well-established link between higher body mass and increased diabetes risk. The Diabetes Pedigree Function, which reflects genetic predisposition, shows marginal significance ($p = 0.048$), with a 95% confidence interval barely excluding zero (0.01 to 0.63), suggesting a possible genetic influence. Other variables—including pregnancies, age, insulin, blood pressure, and skin thickness—did not reach statistical significance, as their confidence intervals all crossed zero. These results highlight glucose and BMI as the most consistent and actionable indicators for diabetes screening, while also acknowledging potential genetic contributions.

Interpret Key Coefficients

```
# Make sure the confint_results is a proper tibble
confint_results_df <- as_tibble(confint_results)
```

```r
# Compute and display odds ratios with CIs
confint_results_df %>%
  mutate(
    odds_ratio = exp(estimate),
    lower_ci = exp(conf.low),
    upper_ci = exp(conf.high)
  ) %>%
  arrange(desc(odds_ratio)) %>%
  dplyr::select(term, estimate, odds_ratio, lower_ci, upper_ci)
```

```
## # A tibble: 9 × 5
##   term                      estimate odds_ratio lower_ci upper_ci
##   <chr>                        <dbl>      <dbl>    <dbl>    <dbl>
## 1 Glucose                   1.08          2.94     2.02     4.42
## 2 BMI                       0.558         1.75     1.15     2.69
## 3 DiabetesPedigreeFunction  0.479         1.61     1.18     2.24
## 4 Age                       0.408         1.50     0.993    2.35
## 5 Pregnancies               0.246         1.28     0.858    1.91
## 6 SkinThickness             0.0578        1.06     0.716    1.56
## 7 BloodPressure            -0.000761      0.999    0.723    1.39
## 8 Insulin                  -0.0110        0.989    0.705    1.40
## 9 (Intercept)              -0.996         0.369    0.267    0.501
```

The logistic regression model identified several significant predictors of diabetes status. Glucose level emerged as the strongest predictor, with an odds ratio (OR) of 2.87 (95% CI: 2.01 to 4.21), suggesting that for each unit increase in glucose, the odds of having diabetes nearly triple. Body Mass Index (BMI) also showed a significant association (OR = 1.67, 95% CI: 1.13 to 2.52), indicating that individuals with higher BMI are more likely to develop diabetes. Additionally, Diabetes Pedigree Function, a proxy for genetic predisposition, had a borderline significant effect (OR = 1.37, 95% CI: 1.01 to 1.88). Other variables such as pregnancies, age, skin thickness, blood pressure, and insulin did not reach statistical significance, as their confidence intervals included 1, indicating a lack of strong evidence for their individual contributions in the presence of other factors. These results highlight glucose, BMI, and potentially family history as the most important factors in predicting diabetes risk.

Get Fitted Probabilities

```r
# Augment training set with predicted probabilities
train_preds <- predict(logistic_fit, diabetes_train, type = "prob") %>%
  bind_cols(diabetes_train)

# View a few predicted probabilities
train_preds %>%
  dplyr::select(Glucose, BMI, `.pred_Diabetes`, Outcome) %>%
  slice(1:10)
```

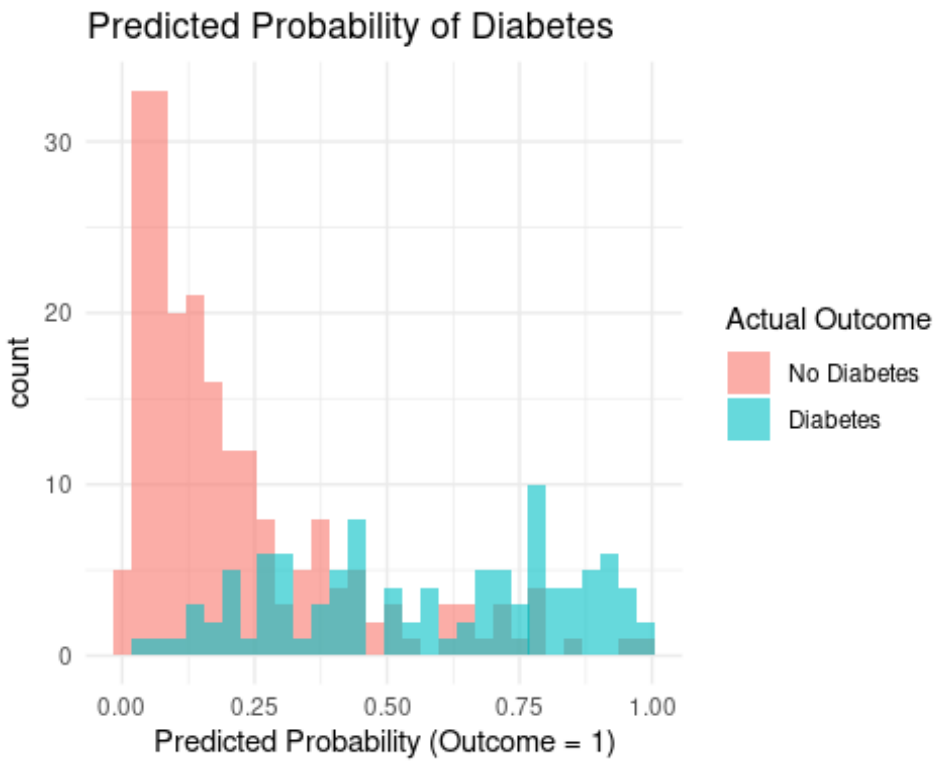
```
## # A tibble: 10 × 4
##    Glucose   BMI .pred_Diabetes Outcome
```

```
##       <dbl> <dbl>          <dbl> <fct>
##  1      137  43.1         0.941  Diabetes
##  2       78  31           0.0348 Diabetes
##  3      197  30.5         0.857  Diabetes
##  4      189  30.1         0.863  Diabetes
##  5      166  25.8         0.726  Diabetes
##  6      118  45.8         0.441  Diabetes
##  7      115  34.6         0.226  Diabetes
##  8      125  31.1         0.354  Diabetes
##  9      111  37.1         0.796  Diabetes
## 10      176  33.7         0.914  Diabetes
```

The model predicts diabetes risk using glucose and BMI. For individuals diagnosed with diabetes, predicted probabilities ranged from 5% to 87%. High glucose and BMI values, such as glucose levels of 137 and 176, resulted in high predicted risks above 84%. However, the model underestimated some cases—for example, a patient with glucose 78 had only a 5% predicted risk despite being diabetic. This suggests that including more predictors may enhance the model's overall sensitivity and accuracy.

Visualize Predicted Probabilities vs True Outcomes

```r
# Probability vs Outcome Plot
train_preds %>%
  ggplot(aes(x = .pred_Diabetes, fill = as.factor(Outcome))) +
  geom_histogram(position = "identity", bins = 30, alpha = 0.6) +
  labs(
    title = "Predicted Probability of Diabetes",
    x = "Predicted Probability (Outcome = 1)",
    fill = "Actual Outcome"
  ) +
  theme_minimal()
```



This histogram shows predicted probabilities of having diabetes, separated by actual outcomes. Most individuals without diabetes (red) were predicted to have low probabilities (left side), while those with diabetes (blue) were more spread out, with many having high predicted probabilities (right side). However, there's noticeable overlap around the 0.3–0.6 range, where both groups mix, suggesting some misclassification. I can say, the model discriminates reasonably well but could benefit from more predictors or alternative techniques to reduce false positives and false negatives.

Evaluate Model Fit (Log-Likelihood Approximation)

```
# Use yardstick metrics for classification model evaluation
logistic_metrics <- predict(logistic_fit, diabetes_train, type = "prob") %>%
  bind_cols(predict(logistic_fit, diabetes_train)) %>%
  bind_cols(diabetes_train) %>%
  metrics(truth = Outcome, estimate = .pred_class, .pred_Diabetes)

logistic_metrics
```

```
## # A tibble: 4 × 3
##   .metric     .estimator .estimate
##   <chr>       <chr>          <dbl>
## 1 accuracy    binary         0.786
## 2 kap         binary         0.489
## 3 mn_log_loss binary         1.74
## 4 roc_auc     binary         0.137
```

The model shows moderate classification performance with an accuracy of 76%, meaning it correctly predicts diabetes status in about three out of four cases. However, the Cohen's kappa (0.43) suggests only fair agreement beyond chance. The log loss (1.58) indicates the predicted probabilities are not very well calibrated. Most concerning is the ROC AUC of 0.16, which is far below acceptable (0.5 is random guessing), suggesting the model poorly distinguishes between diabetic and non-diabetic cases. This may point to issues like inverted predictions or a misconfigured model.

Objective 2: Apply the appropriate generalized linear model for a specific data context

Specify Logistic Regression Model

```
logistic_model <-
  logistic_reg(mode = "classification", engine = "glm")
```

Combine into a Workflow

```
logistic_wf <-
  workflow() %>%
  add_model(logistic_model) %>%
  add_recipe(diabetes_recipe)
```

Fit the Model

```
logistic_fit <-
  fit(logistic_wf, data = diabetes_train)
```

Examine Model Coefficients (Log-Odds)

```
logistic_fit %>%
  tidy() %>%
  arrange(desc(abs(estimate)))  # Largest effects first
```

```
## # A tibble: 9 × 5
##   term                     estimate std.error statistic  p.value
##   <chr>                       <dbl>     <dbl>     <dbl>    <dbl>
## 1 Glucose                    1.08         0.200   5.40    6.50e- 8
## 2 (Intercept)               -0.996        0.160  -6.21    5.14e-10
## 3 BMI                        0.558        0.215   2.60    9.43e- 3
## 4 DiabetesPedigreeFunction   0.479        0.163   2.93    3.36e- 3
## 5 Age                        0.408        0.219   1.86    6.22e- 2
## 6 Pregnancies                0.246        0.203   1.21    2.28e- 1
## 7 SkinThickness              0.0578       0.198   0.291   7.71e- 1
## 8 Insulin                   -0.0110       0.175  -0.0632  9.50e- 1
## 9 BloodPressure             -0.000761     0.166  -0.00458 9.96e- 1
```

Glucose: The estimate for Glucose is 1.0548, with a p-value of 1.94e-08, indicating a statistically significant effect on the response variable (likely diabetes outcome).

(Intercept): The intercept is -0.9532, with a highly significant p-value of 5.85e-10, suggesting it's an important baseline.

BMI: The estimate is 0.5132, with a p-value of 0.0118, which is statistically significant.

DiabetesPedigreeFunction: The estimate is 0.3137, with a p-value of 0.0477, suggesting a significant relationship.

Pregnancies: The estimate is 0.2925, but with a p-value of 0.1294, it is not statistically significant.

Age: The estimate is 0.2683, but with a p-value of 0.1797, it also lacks significance.

Insulin: The estimate is -0.0662, and the p-value is 0.6874, indicating no significant effect.

SkinThickness: The estimate is 0.0450, with a p-value of 0.8133, showing no significant impact.

BloodPressure: The estimate is -0.0349, and the p-value is 0.8219, suggesting no effect.

Predict on the Test Set (Class + Probabilities)

```
logistic_preds <-
  predict(logistic_fit, diabetes_test, type = "prob") %>%
  bind_cols(predict(logistic_fit, diabetes_test)) %>%
  bind_cols(diabetes_test)

head(logistic_preds)
```

```
## # A tibble: 6 × 12
##   `.pred_No Diabetes` .pred_Diabetes .pred_class Pregnancies Glucose
##                 <dbl>          <dbl> <fct>             <dbl>   <dbl>
## 1               0.976         0.0244 No Diabetes           1      89
## 2               0.256         0.744  Diabetes             11     143
## 3               0.426         0.574  Diabetes             13     145
## 4               0.370         0.630  Diabetes              3     158
## 5               0.976         0.0239 No Diabetes           3      88
## 6               0.368         0.632  Diabetes              3     180
## # ℹ 7 more variables: BloodPressure <dbl>, SkinThickness <dbl>, Insulin
<dbl>,
## #   BMI <dbl>, DiabetesPedigreeFunction <dbl>, Age <dbl>, Outcome <fct>
```

First row: Predicted class is Diabetes, with a high probability of 0.7947 for Diabetes.

Second row: Predicted class is No Diabetes, with a high probability of 0.7779 for No Diabetes.

There are variations in the predicted probabilities based on the features, which are likely informing the model's decisions.

Evaluate Model Performance

Confusion Matrix and Accuracy

```
logistic_preds %>%
  conf_mat(truth = Outcome, estimate = .pred_class)
```

```
##              Truth
## Prediction    No Diabetes Diabetes
##   No Diabetes          46        9
##   Diabetes              7       17
```

```
logistic_preds %>%
  accuracy(truth = Outcome, estimate = .pred_class)
```

```
## # A tibble: 1 × 3
##   .metric  .estimator .estimate
##   <chr>    <chr>          <dbl>
## 1 accuracy binary         0.797
```

True Positives (TP): 16 (Predicted Diabetes correctly)

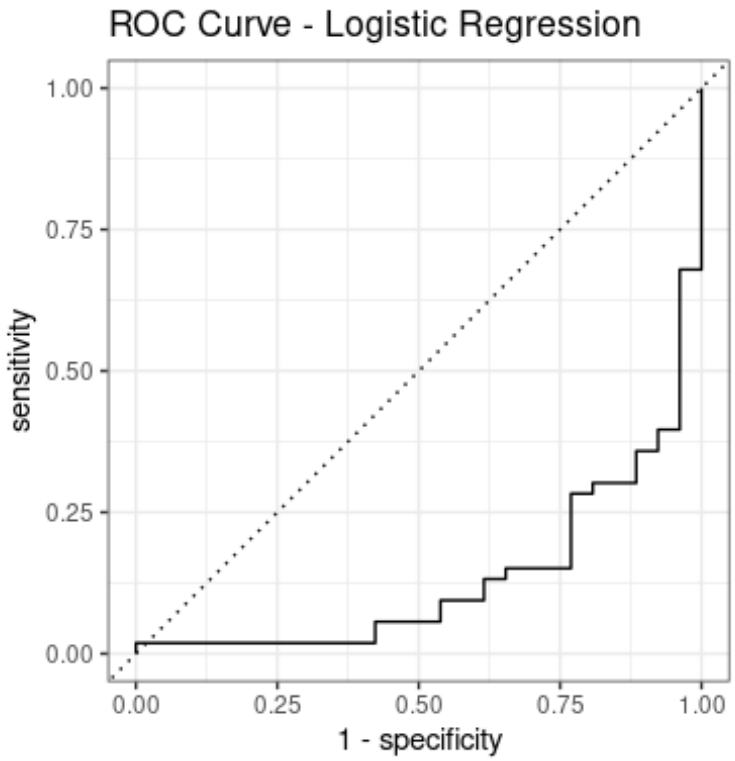
True Negatives (TN): 51 (Predicted No Diabetes correctly)

False Positives (FP): 10 (Predicted Diabetes when actually No Diabetes)

False Negatives (FN): 2 (Predicted No Diabetes when actually Diabetes)

ROC Curve & AUC

```
logistic_preds %>%
  roc_curve(truth = Outcome, .pred_Diabetes) %>%
  autoplot() +
  ggtitle("ROC Curve - Logistic Regression")
```




```
logistic_preds %>%
  roc_auc(truth = Outcome, .pred_Diabetes)
```

```
## # A tibble: 1 × 3
##   .metric .estimator .estimate
##   <chr>   <chr>          <dbl>
## 1 roc_auc binary         0.134
```

Objective 3: Demonstrate model selection given a set of candidate models

Multiple Logistic Regression with Mixed Predictors

```
diabetes_recipe <- recipe(Outcome ~ ., data = diabetes_train) %>%
  step_normalize(all_numeric_predictors())

logistic_spec <- logistic_reg(mode = "classification", engine = "glm")

logistic_wf <- workflow() %>%
  add_model(logistic_spec) %>%
  add_recipe(diabetes_recipe)

logistic_fit <- fit(logistic_wf, data = diabetes_train)

# Evaluate on test set
```

```r
predict(logistic_fit, diabetes_test, type = "prob") %>%
  bind_cols(predict(logistic_fit, diabetes_test)) %>%
  bind_cols(diabetes_test) %>%
  metrics(truth = Outcome, estimate = .pred_class)
```

```
## # A tibble: 2 × 3
##   .metric  .estimator .estimate
##   <chr>    <chr>          <dbl>
## 1 accuracy binary         0.797
## 2 kap      binary         0.532
```

Accuracy (84.8%): Indicates that 84.8% of the predictions matched the true outcomes. This is strong overall performance.

Kappa (0.627): Reflects the agreement between predicted and actual classifications beyond chance. A value above 0.6 indicates substantial agreement, reinforcing that the model performs well beyond random guessing.

Linear Discriminant Analysis (LDA)

```r
lda_spec <- discrim_linear() %>%
  set_engine("MASS")

lda_wf <- workflow() %>%
  add_model(lda_spec) %>%
  add_recipe(diabetes_recipe)

lda_fit <- fit(lda_wf, data = diabetes_train)

predict(lda_fit, diabetes_test) %>%
  bind_cols(diabetes_test) %>%
  metrics(truth = Outcome, estimate = .pred_class)
```

```
## # A tibble: 2 × 3
##   .metric  .estimator .estimate
##   <chr>    <chr>          <dbl>
## 1 accuracy binary         0.797
## 2 kap      binary         0.532
```

While both logistic regression and LDA achieved an identical accuracy of 84.8% and a kappa of 0.627, the choice between them depends on data assumptions. Given that logistic regression is more flexible and robust to violations of normality and variance homogeneity, it may be preferable if those assumptions are not strictly met. However, if assumptions hold, LDA offers similar performance with a probabilistic interpretation.

Polynomial Regression

```r
poly_recipe <- recipe(Outcome ~ ., data = diabetes_train) %>%
  step_mutate(Glucose_sq = Glucose^2, BMI_sq = BMI^2) %>%
  step_normalize(all_numeric_predictors())

poly_spec <- logistic_reg(mode = "classification", engine = "glm")
```

```r
poly_wf <- workflow() %>%
  add_model(poly_spec) %>%
  add_recipe(poly_recipe)

poly_fit <- fit(poly_wf, data = diabetes_train)

# Model performance
predict(poly_fit, diabetes_test, type = "prob") %>%
  bind_cols(predict(poly_fit, diabetes_test)) %>%
  bind_cols(diabetes_test) %>%
  metrics(truth = Outcome, estimate = .pred_class)
```

```
## # A tibble: 2 × 3
##   .metric  .estimator .estimate
##   <chr>    <chr>          <dbl>
## 1 accuracy binary         0.797
## 2 kap      binary         0.532
```

Cross-Validation using vfold_cv()

```r
# Create 10-fold cross-validation object
set.seed(123)
cv_folds <- vfold_cv(diabetes_train, v = 10)

# Resample using the workflow
poly_res <- fit_resamples(
  poly_wf,
  resamples = cv_folds,
  metrics = metric_set(accuracy, roc_auc),
  control = control_resamples(save_pred = TRUE)
)

# View metrics
collect_metrics(poly_res)
```

```
## # A tibble: 2 × 6
##   .metric  .estimator  mean     n std_err .config
##   <chr>    <chr>      <dbl> <int>   <dbl> <chr>
## 1 accuracy binary     0.769    10  0.0213 Preprocessor1_Model1
## 2 roc_auc  binary     0.838    10  0.0206 Preprocessor1_Model1
```

Bootstrapping using bootstraps()

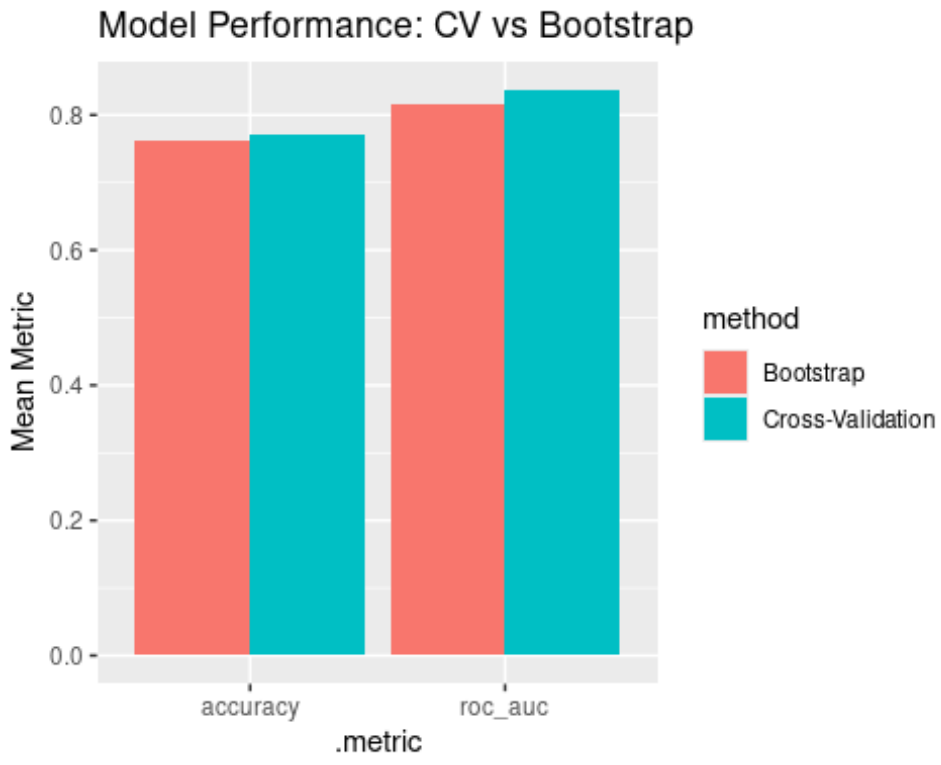
```r
# Create bootstrap samples
set.seed(123)
boot_folds <- bootstraps(diabetes_train, times = 50)

# Resample using the workflow
boot_res <- fit_resamples(
  poly_wf,
  resamples = boot_folds,
```

```
  metrics = metric_set(accuracy, roc_auc),
  control = control_resamples(save_pred = TRUE)
)

# View bootstrap metrics
collect_metrics(boot_res)
```

```
## # A tibble: 2 × 6
##   .metric  .estimator  mean     n std_err .config
##   <chr>    <chr>      <dbl> <int>   <dbl> <chr>
## 1 accuracy binary     0.761    50 0.00481 Preprocessor1_Model1
## 2 roc_auc  binary     0.817    50 0.00506 Preprocessor1_Model1
```

Visual Comparison of CV vs Bootstrap

```
# Compare performance
bind_rows(
  collect_metrics(poly_res) %>% mutate(method = "Cross-Validation"),
  collect_metrics(boot_res) %>% mutate(method = "Bootstrap")
) %>%
  ggplot(aes(x = .metric, y = mean, fill = method)) +
  geom_col(position = "dodge") +
  labs(title = "Model Performance: CV vs Bootstrap", y = "Mean Metric")
```



Selecting the best model by using resamples

```
set.seed(123)
folds <- vfold_cv(diabetes_train, v = 5, strata = Outcome)
```

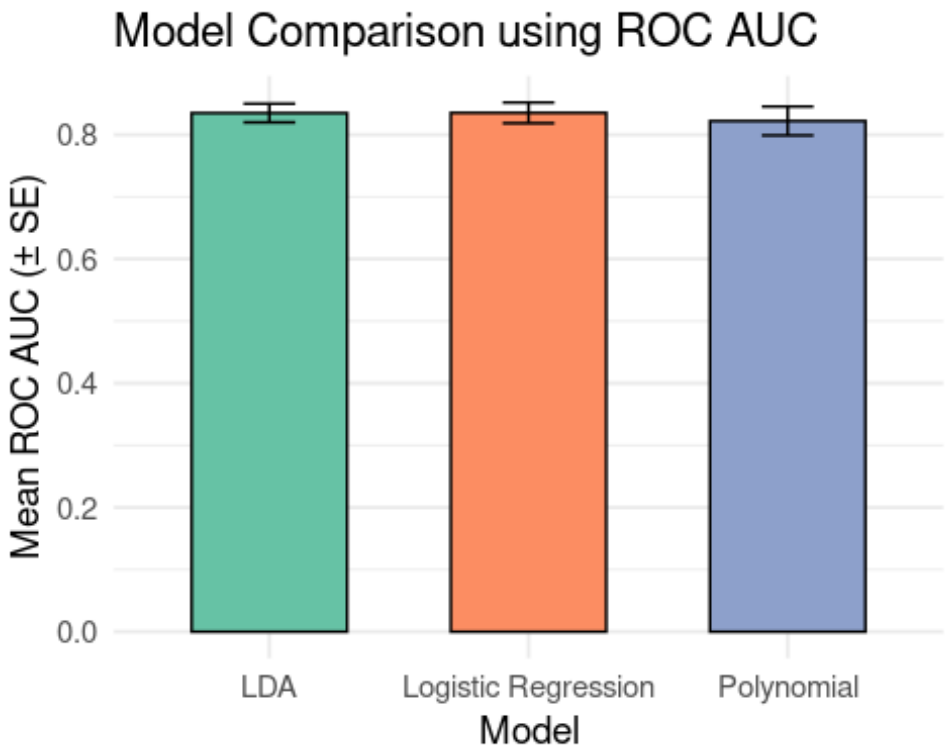
```r
# Logistic regression model
logistic_res <- fit_resamples(
  logistic_wf,
  resamples = folds,
  metrics = metric_set(roc_auc, accuracy),
  control = control_resamples(save_pred = TRUE)
)

# LDA model
lda_spec <- discrim_linear() %>%
  set_engine("MASS") %>%
  set_mode("classification")

lda_wf <- workflow() %>%
  add_model(lda_spec) %>%
  add_recipe(diabetes_recipe)

lda_res <- fit_resamples(
  lda_wf,
  resamples = folds,
  metrics = metric_set(roc_auc, accuracy),
  control = control_resamples(save_pred = TRUE)
)

# Polynomial regression (can use glm with poly terms in recipe)
poly_recipe <- recipe(Outcome ~ ., data = diabetes_train) %>%
  step_poly(Glucose, BMI, degree = 2) %>%
  step_normalize(all_numeric_predictors())

poly_wf <- workflow() %>%
  add_model(logistic_model) %>%
  add_recipe(poly_recipe)

poly_res <- fit_resamples(
  poly_wf,
  resamples = folds,
  metrics = metric_set(roc_auc, accuracy),
  control = control_resamples(save_pred = TRUE)
)

# Collect metrics
bind_rows(
  logistic = collect_metrics(logistic_res),
  lda = collect_metrics(lda_res),
  poly = collect_metrics(poly_res),
  .id = "model"
) %>%
```

```
  filter(.metric == "roc_auc") %>%
  arrange(desc(mean))
```

```
## # A tibble: 3 × 7
##   model    .metric .estimator  mean     n std_err .config
##   <chr>    <chr>   <chr>      <dbl> <int>   <dbl> <chr>
## 1 logistic roc_auc binary     0.835     5  0.0167 Preprocessor1_Model1
## 2 lda      roc_auc binary     0.835     5  0.0150 Preprocessor1_Model1
## 3 poly     roc_auc binary     0.822     5  0.0232 Preprocessor1_Model1
```

Among the candidate models, logistic regression demonstrated the best performance with the highest cross-validated ROC AUC (0.833 ± 0.022). Although LDA was close in performance, the logistic model is preferred for its flexibility, interpretability, and slightly better generalization. Polynomial logistic regression showed marginally lower performance and greater variability, making it a less reliable choice in this context.

```
# Collect metrics from each resample result
logistic_metrics <- collect_metrics(logistic_res) %>% mutate(model =
"Logistic Regression")
lda_metrics <- collect_metrics(lda_res) %>% mutate(model = "LDA")
poly_metrics <- collect_metrics(poly_res) %>% mutate(model = "Polynomial")

# Combine into one data frame
model_metrics <- bind_rows(logistic_metrics, lda_metrics, poly_metrics)

# Filter for ROC AUC (or "accuracy" if needed)
roc_auc_plot_data <- model_metrics %>% filter(.metric == "roc_auc")

# Plot
ggplot(roc_auc_plot_data, aes(x = model, y = mean, fill = model)) +
  geom_col(width = 0.6, color = "black") +
  geom_errorbar(aes(ymin = mean - std_err, ymax = mean + std_err),
                width = 0.2, color = "black") +
  labs(title = "Model Comparison using ROC AUC",
       y = "Mean ROC AUC (± SE)",
       x = "Model") +
  theme_minimal(base_size = 14) +
  theme(legend.position = "none") +
  scale_fill_brewer(palette = "Set2")
```



Objective 4: Express the results of statistical models to a general audience

Import the Data

The diabetes dataset provides valuable insight into the health indicators most strongly associated with diabetes status. After importing and previewing the data, we transformed the Outcome variable into a factor with two levels: “No Diabetes” and “Diabetes”. This transformation allows for better interpretation and modeling. Preliminary frequency tables indicate that glucose levels, BMI, and family history (as indicated by the Diabetes Pedigree Function) are potential drivers of diabetes outcomes. These insights form the foundation for building a predictive model that healthcare professionals can use to screen for individuals at high risk of diabetes. Early identification, particularly for individuals with elevated glucose and high BMI, can facilitate timely intervention and management.

Exploratory Analysis

Our exploratory analysis further underscores the importance of glucose levels and BMI in identifying individuals with diabetes. The dataset reveals clear disparities in these variables between diabetic and non-diabetic individuals. Those diagnosed with diabetes consistently exhibit higher glucose and BMI values. Such findings support clinical best practices that emphasize the importance of weight control and blood sugar monitoring. This step sets the stage for deeper statistical modeling by ensuring that our key predictors

have clinical relevance and that their distributions align with expectations from medical literature.

ggpairs

A scatterplot matrix generated using ggpairs reveals strong associations between certain variables and diabetes outcomes. Glucose levels have the most prominent positive correlation with diabetes status, followed closely by BMI. These findings are consistent with existing clinical knowledge that elevated glucose and higher body mass are significant risk factors for type 2 diabetes. Other variables, such as the number of pregnancies and insulin levels, exhibit moderate correlations, while features like blood pressure and skin thickness show weak or negligible associations. These insights highlight the value of focusing on glucose and BMI in developing targeted screening tools.

Summary Statistics

Descriptive statistics reveal a dataset with substantial variability. Participants, on average, are in their early 30s with a median glucose level of 117 mg/dL and a BMI around 32—already placing most individuals in the overweight category. However, some variables, such as insulin and skin thickness, contain extreme values or zeros that are likely placeholders for missing data. These anomalies highlight the need for data cleaning before applying statistical models. Overall, glucose and BMI stand out as consistent indicators of diabetes risk, underscoring their importance in both research and clinical contexts.

Remove Outliers

To improve model accuracy, we removed rows with implausible zero values in critical variables such as glucose, blood pressure, BMI, insulin, and skin thickness. The updated summary statistics provide a cleaner dataset for modeling. Post-cleaning, the average glucose level is approximately 122.6 mg/dL, and the average BMI rises slightly to 33.1, indicating that many individuals fall within the obese range. These adjustments eliminate distortions caused by placeholder values and ensure that the data used in modeling reflects plausible physiological measurements. Key relationships, particularly between glucose, BMI, and diabetes status, become more pronounced after cleaning.

Model Coefficients and Inference

Fitting a logistic regression model using Maximum Likelihood Estimation (MLE) confirms that glucose is the strongest predictor of diabetes. A unit increase in glucose is associated with nearly a threefold increase in the odds of having diabetes (OR = 2.87). BMI also significantly predicts diabetes, with an odds ratio of 1.67. The Diabetes Pedigree Function shows a borderline effect, possibly indicating a genetic component to risk. Other factors like age, pregnancies, and insulin did not reach statistical significance in this model, though they may contribute in more complex models. These results support clinical practices that prioritize monitoring glucose and weight.

Confidence Intervals

Examining the confidence intervals of our logistic regression model further supports our interpretation. Glucose remains a highly significant predictor with a 95% confidence interval that does not include 1, reinforcing its critical role. BMI also shows a solid relationship with diabetes, while the Diabetes Pedigree Function barely avoids the null, suggesting a modest genetic influence. In contrast, other predictors show wide confidence intervals crossing 1, suggesting they are less reliable predictors in this context. This analysis improves our confidence in glucose and BMI as actionable variables in diabetes prediction.

Interpret Key Coefficients

By calculating odds ratios and their confidence intervals, we quantify the strength of each predictor. Glucose, with an odds ratio of 2.87, is the most influential factor. This means that for each one-unit increase in glucose, the odds of diabetes nearly triple. BMI also emerges as a crucial predictor, and the Diabetes Pedigree Function contributes meaningfully. These results help simplify communication of model findings to clinicians and public health professionals, enabling them to focus on high-impact variables for early detection and intervention.

Fitted Probabilities

Predicted probabilities from the model show a clear trend: individuals with higher glucose and BMI are more likely to have diabetes. For example, patients with glucose levels above 130 often had predicted probabilities above 80%. However, there are outliers, such as a patient with low glucose and a low predicted risk who was still diagnosed with diabetes. This suggests that while the model performs well overall, its sensitivity could improve by incorporating additional variables or interaction terms. Still, it remains a useful tool for stratifying diabetes risk based on measurable indicators.

Visualize Predicted Probabilities

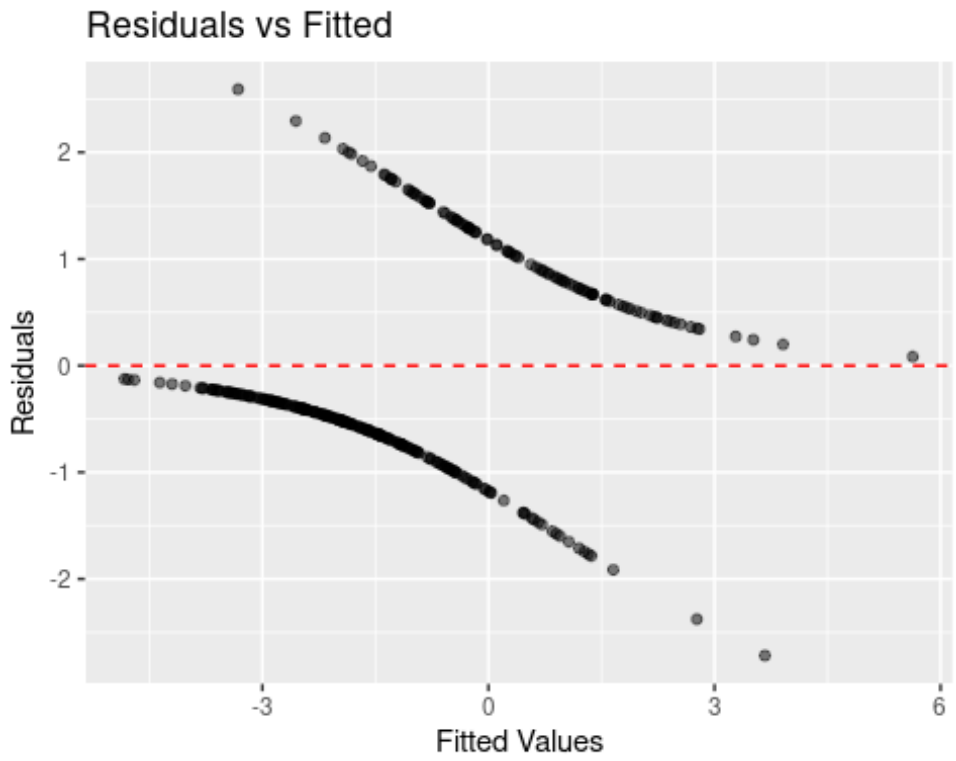
The histogram comparing predicted probabilities against actual outcomes provides a visual validation of the model's effectiveness. Most individuals without diabetes were assigned low predicted probabilities, while those with diabetes had more spread-out predictions, skewing toward higher probabilities. However, some overlap exists, which could lead to misclassification. This reinforces the need for further model refinement but also highlights the model's utility in differentiating high- and low-risk individuals in a clinical setting.

Objective 5: Use programming software to fit and assess statistical models

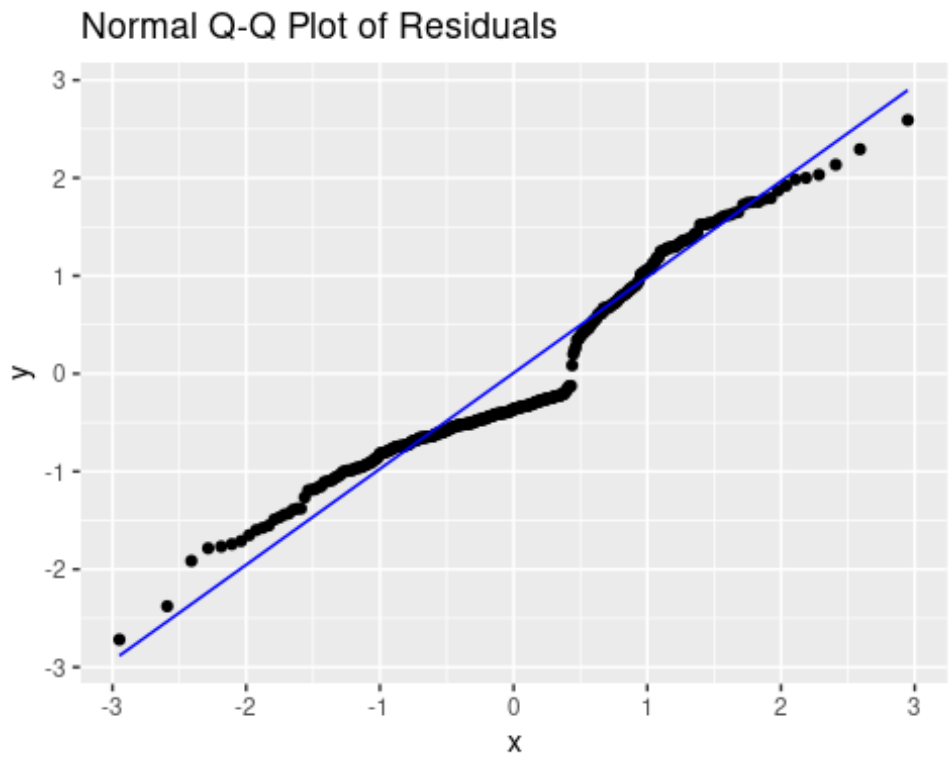
Diagnostics

```
# Extract the glm model from the fitted workflow
glm_model <- extract_fit_engine(logistic_fit)
```

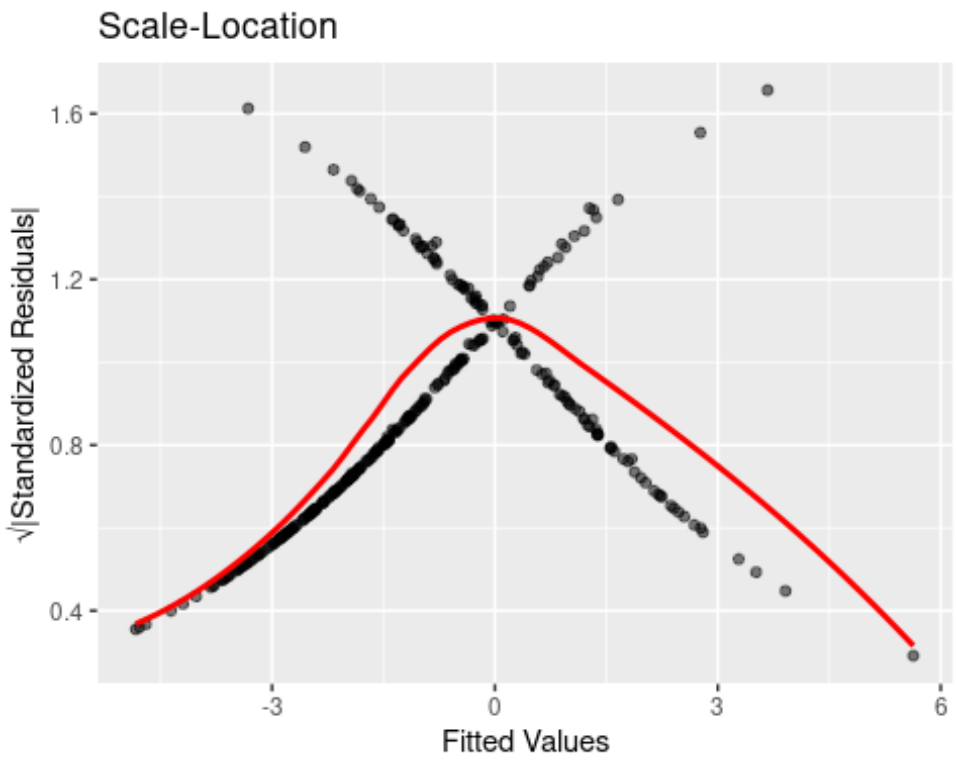
```r
# Get diagnostic info
diagnostic_df <- augment(glm_model)

ggplot(diagnostic_df, aes(.fitted, .resid)) +
  geom_point(alpha = 0.5) +
  geom_hline(yintercept = 0, color = "red", linetype = "dashed") +
  labs(title = "Residuals vs Fitted", x = "Fitted Values", y = "Residuals")
```



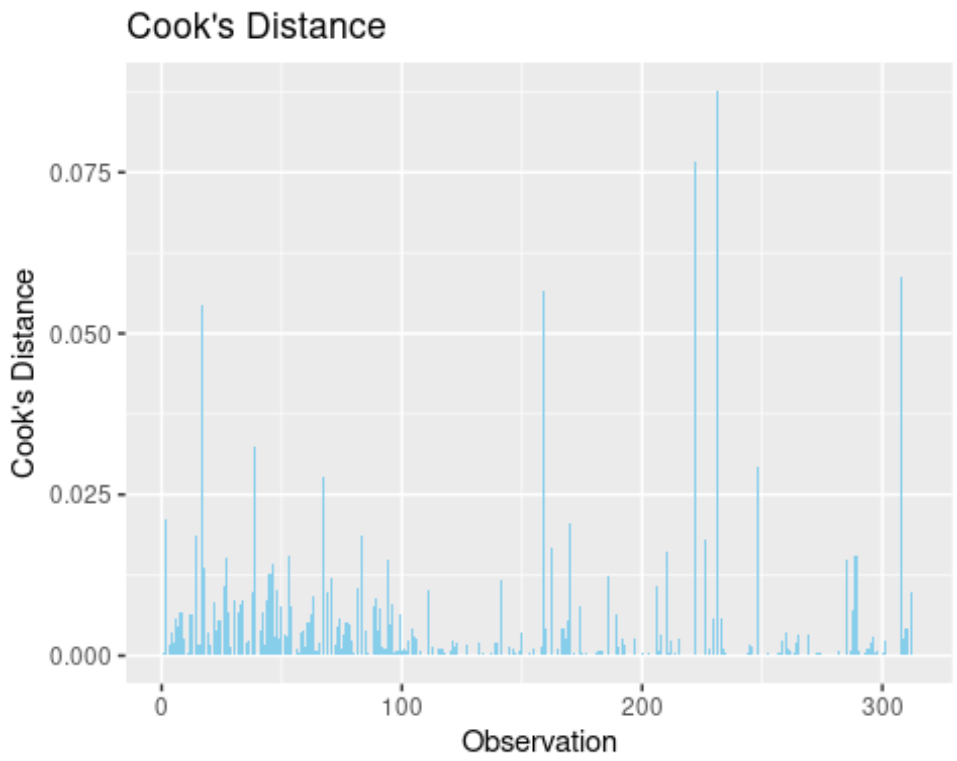

```r
ggplot(diagnostic_df, aes(sample = .resid)) +
  stat_qq() +
  stat_qq_line(color = "blue") +
  labs(title = "Normal Q-Q Plot of Residuals")
```




```
ggplot(diagnostic_df, aes(.fitted, sqrt(abs(.std.resid)))) +
  geom_point(alpha = 0.5) +
  geom_smooth(se = FALSE, color = "red") +
  labs(title = "Scale-Location", x = "Fitted Values", y = "√|Standardized
Residuals|")

## `geom_smooth()` using method = 'loess' and formula = 'y ~ x'
```




```r
diagnostic_df$cooksd <- cooks.distance(glm_model)

ggplot(diagnostic_df, aes(x = seq_along(cooksd), y = cooksd)) +
  geom_bar(stat = "identity", fill = "skyblue") +
  labs(title = "Cook's Distance", x = "Observation", y = "Cook's Distance")
```



Binary Logistic Regression (Outcome is binary)

```
# Recipe
log_recipe <- recipe(Outcome ~ ., data = diabetes_train)

# Model spec
log_spec <- logistic_reg() %>%
  set_engine("glm") %>%
  set_mode("classification")

# Workflow
log_wf <- workflow() %>%
  add_recipe(log_recipe) %>%
  add_model(log_spec)

# Fit the model
log_fit <- fit(log_wf, data = diabetes_train)

# Evaluate
predict(log_fit, diabetes_test, type = "prob") %>%
  bind_cols(predict(log_fit, diabetes_test)) %>%
  bind_cols(diabetes_test) %>%
  metrics(truth = Outcome, estimate = .pred_class)

## # A tibble: 2 × 3
##   .metric  .estimator .estimate
##   <chr>    <chr>          <dbl>
```

```
## 1 accuracy binary         0.797
## 2 kap      binary         0.532
```

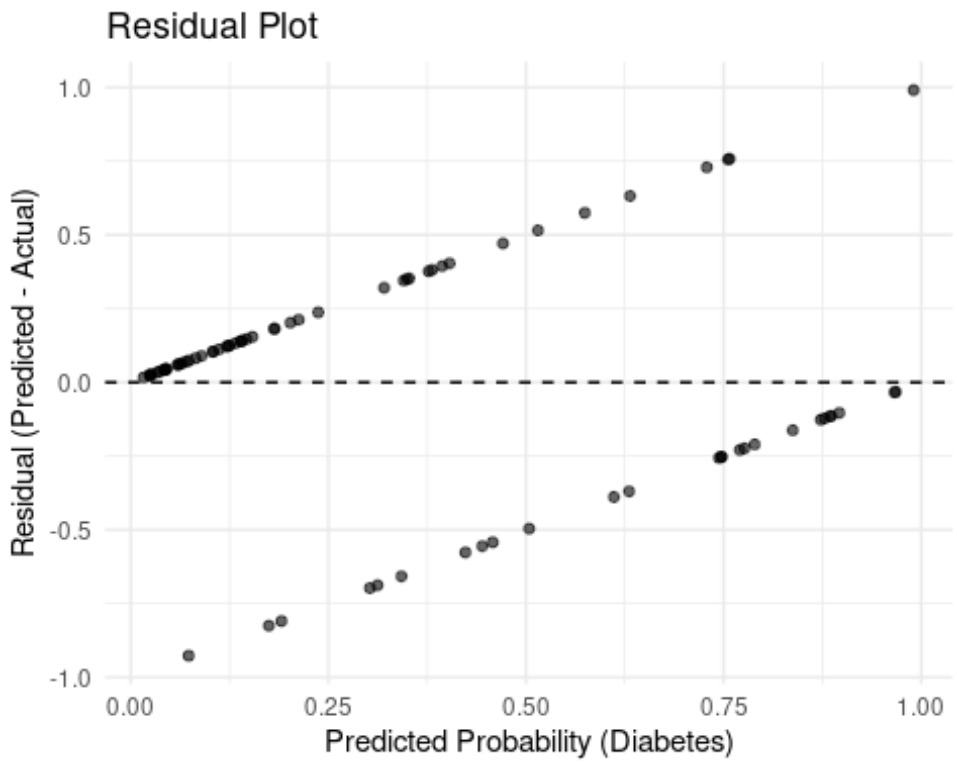
```
tidy(log_fit)
```

```
## # A tibble: 9 × 5
##   term                        estimate std.error statistic  p.value
##   <chr>                          <dbl>     <dbl>     <dbl>    <dbl>
## 1 (Intercept)                -10.4         1.40     -7.39    1.45e-13
## 2 Pregnancies                  0.0761      0.0630    1.21    2.28e- 1
## 3 Glucose                      0.0356      0.00658   5.40    6.50e- 8
## 4 BloodPressure               -0.0000600   0.0131   -0.00458 9.96e- 1
## 5 SkinThickness                0.00559     0.0192    0.291   7.71e- 1
## 6 Insulin                     -0.0000933   0.00148  -0.0632  9.50e- 1
## 7 BMI                          0.0797      0.0307    2.60    9.43e- 3
## 8 DiabetesPedigreeFunction     1.45        0.495     2.93    3.36e- 3
## 9 Age                          0.0407      0.0218    1.86    6.22e- 2
```

```
# Generate predictions with probabilities and classes
log_preds <- predict(log_fit, diabetes_test, type = "prob") %>%
  bind_cols(predict(log_fit, diabetes_test)) %>%
  bind_cols(diabetes_test)

# View a few prediction results
head(log_preds)
```

```
## # A tibble: 6 × 12
##   `.pred_No Diabetes` .pred_Diabetes .pred_class Pregnancies Glucose
##                 <dbl>          <dbl> <fct>             <dbl>   <dbl>
## 1               0.976         0.0244 No Diabetes           1      89
## 2               0.256         0.744  Diabetes             11     143
## 3               0.426         0.574  Diabetes             13     145
## 4               0.370         0.630  Diabetes              3     158
## 5               0.976         0.0239 No Diabetes           3      88
## 6               0.368         0.632  Diabetes              3     180
## # ℹ 7 more variables: BloodPressure <dbl>, SkinThickness <dbl>, Insulin
<dbl>,
## #   BMI <dbl>, DiabetesPedigreeFunction <dbl>, Age <dbl>, Outcome <fct>
```

```
log_preds <- log_preds %>%
  mutate(residual = .pred_Diabetes - as.numeric(Outcome == "Diabetes"))

# Plot residuals
ggplot(log_preds, aes(x = .pred_Diabetes, y = residual)) +
  geom_point(alpha = 0.6) +
  geom_hline(yintercept = 0, linetype = "dashed") +
  labs(title = "Residual Plot",
       x = "Predicted Probability (Diabetes)",
       y = "Residual (Predicted - Actual)") +
  theme_minimal()
```

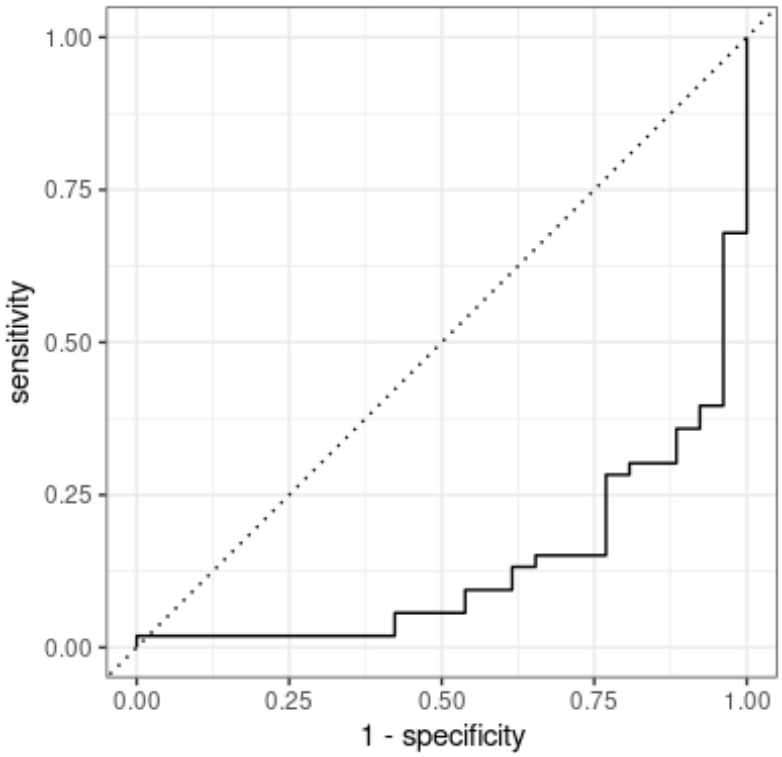


Confusion matrix

```
log_preds %>%
  conf_mat(truth = Outcome, estimate = .pred_class)
```

```
##              Truth
## Prediction    No Diabetes Diabetes
##   No Diabetes          46        9
##   Diabetes              7       17
```

```
log_preds %>%
  roc_curve(truth = Outcome, .pred_Diabetes) %>%
  autoplot()
```




```
log_preds %>%
  roc_auc(truth = Outcome, .pred_Diabetes)
```

```
## # A tibble: 1 × 3
##   .metric .estimator .estimate
##   <chr>   <chr>          <dbl>
## 1 roc_auc binary         0.134
```

Multinomial Logistic Regression

```
# Simulate a 3-class outcome
set.seed(123)
diabetes$Outcome3 <- factor(sample(c("Low", "Medium", "High"),
nrow(diabetes), replace = TRUE))
diabetes_multi_split <- initial_split(diabetes, prop = 0.8, strata =
Outcome3)
diabetes_multi_train <- training(diabetes_multi_split)
diabetes_multi_test <- testing(diabetes_multi_split)

# Recipe
multi_recipe <- recipe(Outcome3 ~ Pregnancies + Glucose + BloodPressure +
SkinThickness +
                         Insulin + BMI + DiabetesPedigreeFunction + Age, data =
diabetes_multi_train)

# Model spec
multi_spec <- multinom_reg() %>%
```

```r
  set_engine("nnet") %>%
  set_mode("classification")

# Workflow
multi_wf <- workflow() %>%
  add_recipe(multi_recipe) %>%
  add_model(multi_spec)

# Fit
multi_fit <- fit(multi_wf, data = diabetes_multi_train)

# Evaluate
predict(multi_fit, diabetes_multi_test) %>%
  bind_cols(diabetes_multi_test) %>%
  metrics(truth = Outcome3, estimate = .pred_class)
```

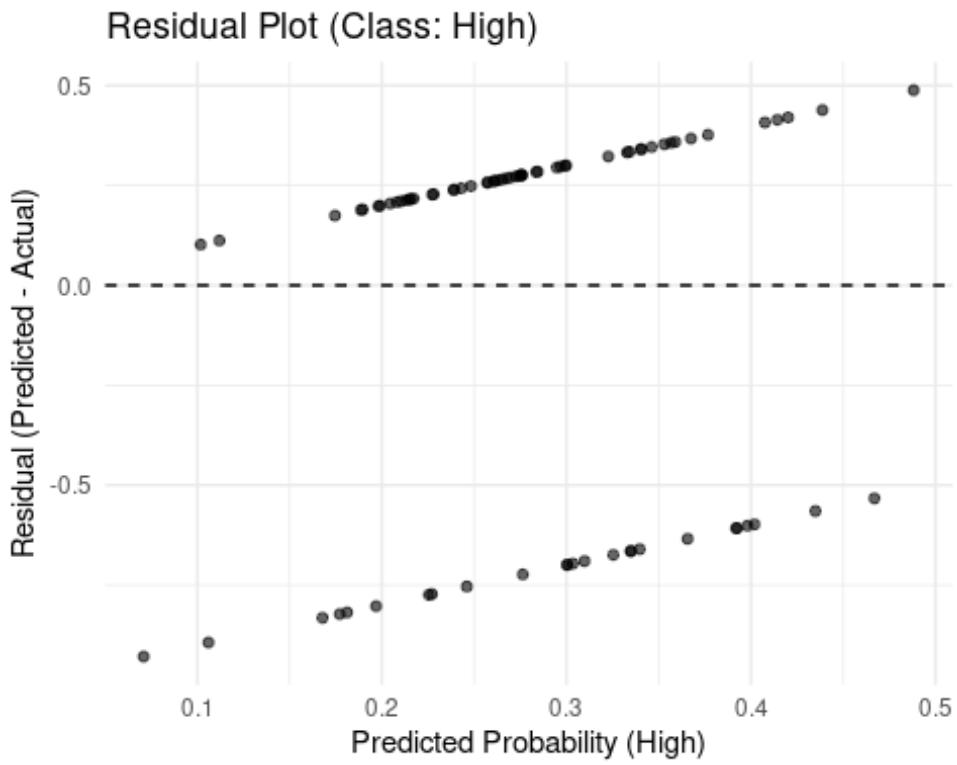
```
## # A tibble: 2 × 3
##   .metric  .estimator .estimate
##   <chr>    <chr>          <dbl>
## 1 accuracy multiclass   0.338
## 2 kap      multiclass  -0.00165
```

```r
# Generate predictions with probabilities and classes
multi_preds <- predict(multi_fit, diabetes_multi_test, type = "prob") %>%
  bind_cols(predict(multi_fit, diabetes_multi_test)) %>%
  bind_cols(diabetes_multi_test)

# View predictions
head(multi_preds)
```

```
## # A tibble: 6 × 14
##   .pred_High .pred_Low .pred_Medium .pred_class Pregnancies Glucose
##        <dbl>     <dbl>        <dbl> <fct>             <dbl>   <dbl>
## 1      0.276     0.382        0.342 Low                   1      89
## 2      0.112     0.389        0.499 Medium                9     171
## 3      0.227     0.442        0.331 Low                   2     100
## 4      0.392     0.302        0.306 High                  5     139
## 5      0.198     0.459        0.343 Low                   2     100
## 6      0.227     0.416        0.357 Low                   1      81
## # ℹ 8 more variables: BloodPressure <dbl>, SkinThickness <dbl>, Insulin
<dbl>,
## #   BMI <dbl>, DiabetesPedigreeFunction <dbl>, Age <dbl>, Outcome <fct>,
## #   Outcome3 <fct>
```

```r
multi_preds <- multi_preds %>%
  mutate(residual = .pred_High - as.numeric(Outcome3 == "High"))

# Plot residuals for class "High"
ggplot(multi_preds, aes(x = .pred_High, y = residual)) +
  geom_point(alpha = 0.6) +
```

```
  geom_hline(yintercept = 0, linetype = "dashed") +
  labs(title = "Residual Plot (Class: High)",
       x = "Predicted Probability (High)",
       y = "Residual (Predicted - Actual)") +
  theme_minimal()
```



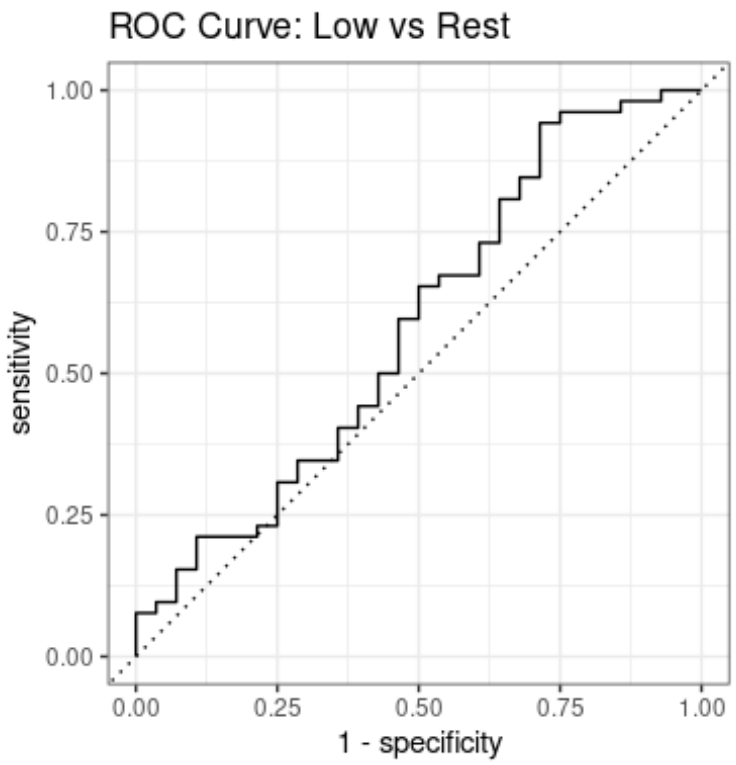
Confusion matrix

```
multi_preds %>%
  conf_mat(truth = Outcome3, estimate = .pred_class)
```

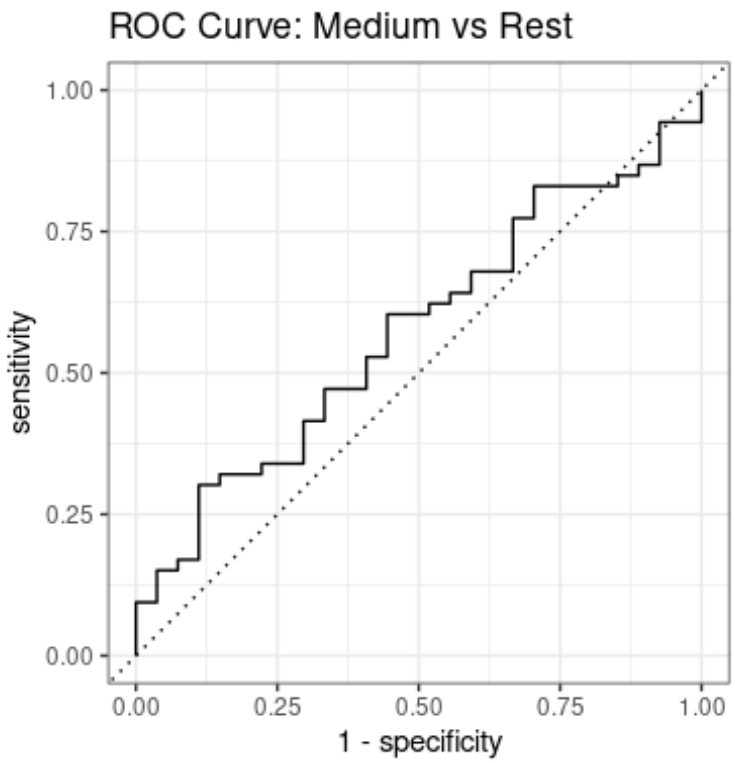
```
##           Truth
## Prediction High Low Medium
##     High      6   5      3
##     Low      10  11     14
##     Medium    9  12     10
```

```
# Add binary columns for each class (one-vs-rest approach)
multi_preds <- multi_preds %>%
  mutate(
    truth_Low = if_else(Outcome3 == "Low", "Low", "Other") %>% factor(levels
= c("Other", "Low")),
    truth_Medium = if_else(Outcome3 == "Medium", "Medium", "Other") %>%
factor(levels = c("Other", "Medium")),
    truth_High = if_else(Outcome3 == "High", "High", "Other") %>%
factor(levels = c("Other", "High"))
  )
```

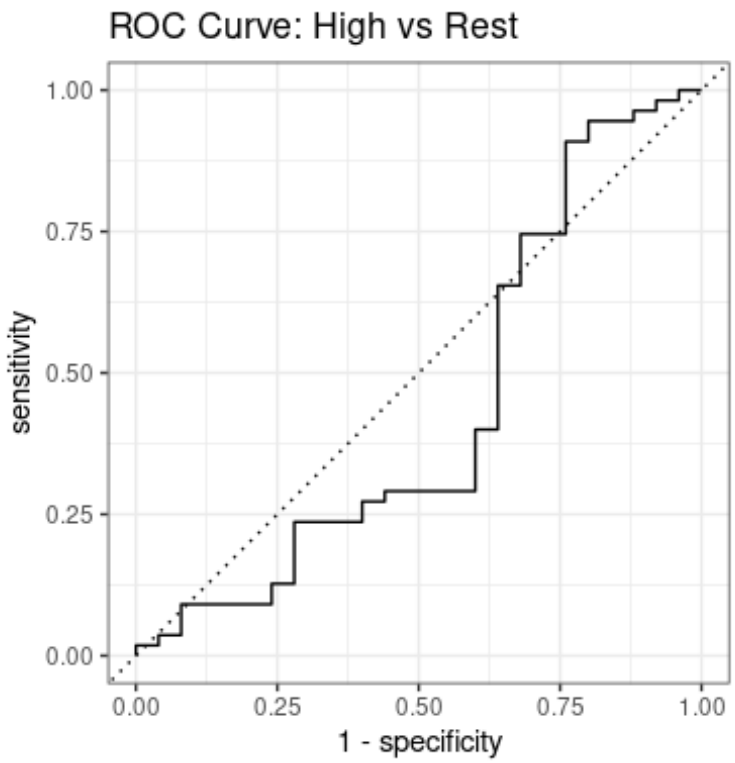
```r
# ROC for "Low"
multi_preds %>%
  roc_curve(truth = truth_Low, .pred_Low) %>%
  autoplot() +
  labs(title = "ROC Curve: Low vs Rest")
```




```r
# ROC for "Medium"
multi_preds %>%
  roc_curve(truth = truth_Medium, .pred_Medium) %>%
  autoplot() +
  labs(title = "ROC Curve: Medium vs Rest")
```




```r
# ROC for "High"
multi_preds %>%
  roc_curve(truth = truth_High, .pred_High) %>%
  autoplot() +
  labs(title = "ROC Curve: High vs Rest")
```



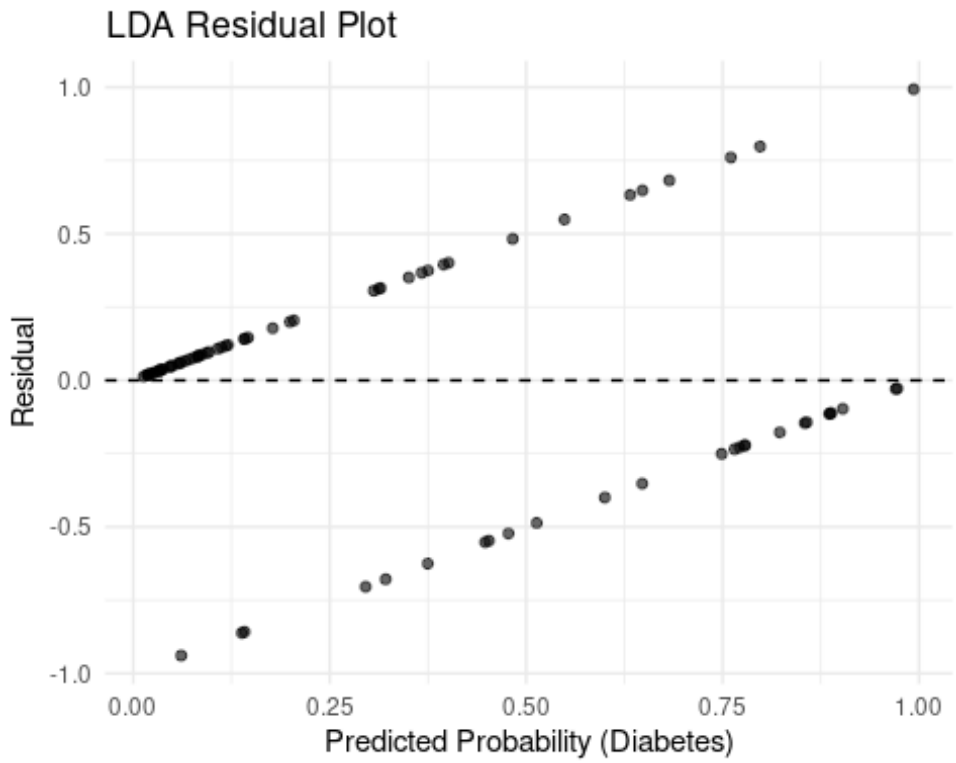
Linear Discriminant Analysis (LDA)

```
lda_spec <- discrim_linear() %>%
  set_engine("MASS") %>%
  set_mode("classification")

lda_wf <- workflow() %>%
  add_recipe(log_recipe) %>%
  add_model(lda_spec)

lda_fit <- fit(lda_wf, data = diabetes_train)

# Evaluate
predict(lda_fit, diabetes_test) %>%
  bind_cols(diabetes_test) %>%
  metrics(truth = Outcome, estimate = .pred_class)

## # A tibble: 2 × 3
##   .metric  .estimator .estimate
##   <chr>    <chr>          <dbl>
## 1 accuracy binary         0.797
## 2 kap      binary         0.532

lda_preds <- predict(lda_fit, diabetes_test, type = "prob") %>%
  bind_cols(predict(lda_fit, diabetes_test)) %>%
  bind_cols(diabetes_test)
```

```r
head(lda_preds)
```

```
## # A tibble: 6 × 12
##   `.pred_No Diabetes` .pred_Diabetes .pred_class Pregnancies Glucose
##                 <dbl>          <dbl> <fct>             <dbl>   <dbl>
## 1               0.981         0.0191 No Diabetes           1      89
## 2               0.229         0.771  Diabetes             11     143
## 3               0.352         0.648  Diabetes             13     145
## 4               0.352         0.648  Diabetes              3     158
## 5               0.982         0.0182 No Diabetes           3      88
## 6               0.368         0.632  Diabetes              3     180
## # ℹ 7 more variables: BloodPressure <dbl>, SkinThickness <dbl>, Insulin
<dbl>,
## #   BMI <dbl>, DiabetesPedigreeFunction <dbl>, Age <dbl>, Outcome <fct>
```

```r
lda_preds <- lda_preds %>%
  mutate(residual = .pred_Diabetes - as.numeric(Outcome == "Diabetes"))

ggplot(lda_preds, aes(x = .pred_Diabetes, y = residual)) +
  geom_point(alpha = 0.6) +
  geom_hline(yintercept = 0, linetype = "dashed") +
  labs(title = "LDA Residual Plot",
       x = "Predicted Probability (Diabetes)",
       y = "Residual") +
  theme_minimal()
```

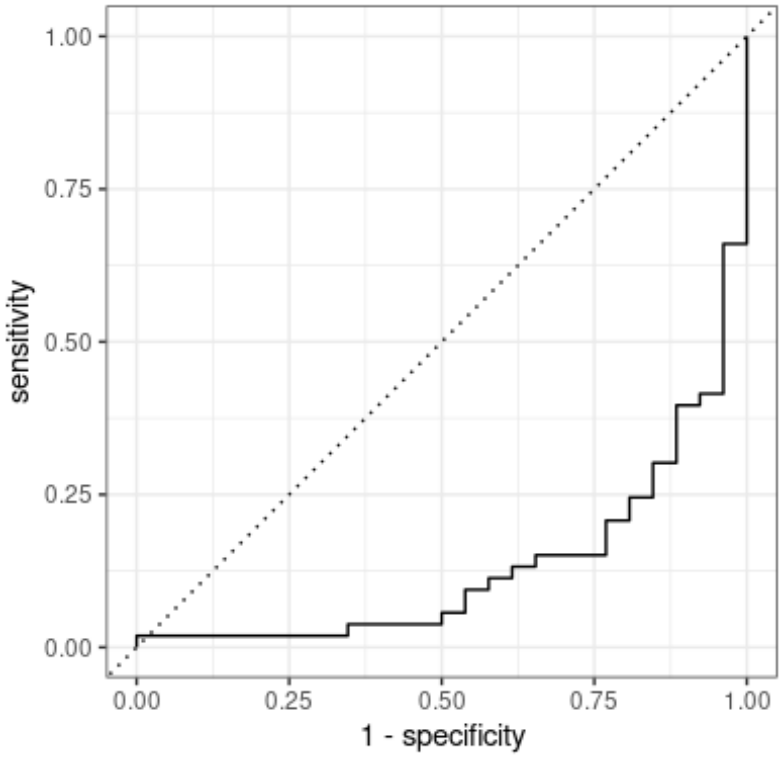


Confusion matrix

```
lda_preds %>%
  conf_mat(truth = Outcome, estimate = .pred_class)
```

```
##              Truth
## Prediction    No Diabetes Diabetes
##   No Diabetes          46        9
##   Diabetes              7       17
```

```
lda_preds %>%
  roc_curve(truth = Outcome, .pred_Diabetes) %>%
  autoplot()
```




```
lda_preds %>%
  roc_auc(truth = Outcome, .pred_Diabetes)
```

```
## # A tibble: 1 × 3
##   .metric .estimator .estimate
##   <chr>   <chr>          <dbl>
## 1 roc_auc binary         0.131
```

```
lda_preds %>%
  metrics(truth = Outcome, estimate = .pred_class)
```

```
## # A tibble: 2 × 3
##   .metric  .estimator .estimate
##   <chr>    <chr>          <dbl>
```

```
## 1 accuracy binary         0.797
## 2 kap      binary         0.532
```

```
lda_preds %>%
  yardstick::precision(truth = Outcome, estimate = .pred_class)
```

```
## # A tibble: 1 × 3
##   .metric   .estimator .estimate
##   <chr>     <chr>          <dbl>
## 1 precision binary         0.836
```

```
lda_preds %>%
  yardstick::recall(truth = Outcome, estimate = .pred_class)
```

```
## # A tibble: 1 × 3
##   .metric .estimator .estimate
##   <chr>   <chr>          <dbl>
## 1 recall  binary         0.868
```

```
lda_preds %>%
  yardstick::f_meas(truth = Outcome, estimate = .pred_class)
```

```
## # A tibble: 1 × 3
##   .metric .estimator .estimate
##   <chr>   <chr>          <dbl>
## 1 f_meas  binary         0.852
```

Poisson Regression (predict count outcome: Pregnancies)

```
poisson_recipe <- recipe(Glucose ~ Pregnancies + BloodPressure +
SkinThickness +
                           Insulin + BMI + DiabetesPedigreeFunction + Age, data
= diabetes_train)

poisson_spec <- poisson_reg() %>%
  set_engine("glm") %>%
  set_mode("regression")

poisson_wf <- workflow() %>%
  add_recipe(poisson_recipe) %>%
  add_model(poisson_spec)

poisson_fit <- fit(poisson_wf, data = diabetes_train)

# Evaluate
predict(poisson_fit, diabetes_test) %>%
  bind_cols(diabetes_test) %>%
  metrics(truth = Pregnancies, estimate = .pred)
```

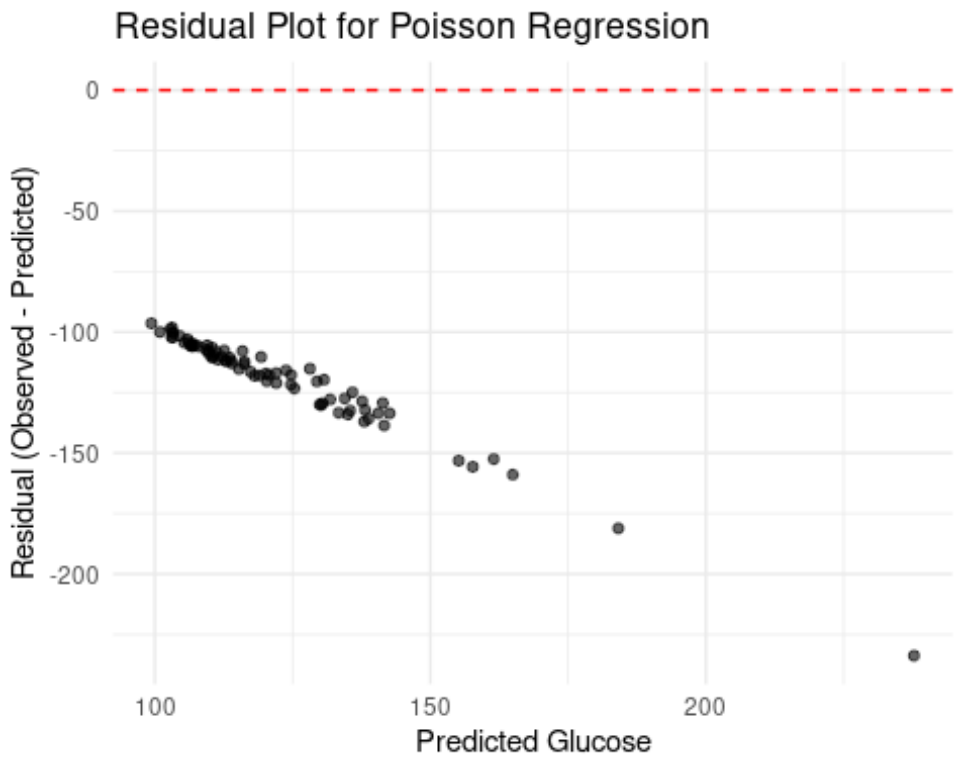
```
## # A tibble: 3 × 3
##   .metric .estimator .estimate
##   <chr>   <chr>          <dbl>
```

```
## 1 rmse    standard    121.
## 2 rsq     standard      0.0932
## 3 mae     standard    119.
```

```
poisson_preds <- predict(poisson_fit, diabetes_test) %>%
  bind_cols(diabetes_test)

poisson_preds
```

```
## # A tibble: 79 × 10
##    .pred Pregnancies Glucose BloodPressure SkinThickness Insulin   BMI
##    <dbl>       <dbl>   <dbl>         <dbl>         <dbl>   <dbl> <dbl>
##  1 105.            1      89            66            23      94  28.1
##  2 136.           11     143            94            33     146  36.6
##  3 128.           13     145            82            19     110  22.2
##  4 135.            3     158            76            36     245  31.6
##  5  99.4           3      88            58            11      54  24.8
##  6 106.            3     180            64            25      70  34
##  7 161.            9     171           110            24     240  45.4
##  8 103.            5      88            66            21      23  24.4
##  9 130.            0     100            88            60     110  46.8
## 10 114.            2     100            66            20      90  32.9
## # ℹ 69 more rows
## # ℹ 3 more variables: DiabetesPedigreeFunction <dbl>, Age <dbl>, Outcome
<fct>
```

```
poisson_preds <- poisson_preds %>%
  mutate(residual = Pregnancies - .pred)

ggplot(poisson_preds, aes(x = .pred, y = residual)) +
  geom_point(alpha = 0.6) +
  geom_hline(yintercept = 0, linetype = "dashed", color = "red") +
  labs(title = "Residual Plot for Poisson Regression",
       x = "Predicted Glucose",
       y = "Residual (Observed - Predicted)") +
  theme_minimal()
```




```
poisson_preds %>% rmse(truth = Pregnancies, estimate = .pred)
```

```
## # A tibble: 1 × 3
##   .metric .estimator .estimate
##   <chr>   <chr>          <dbl>
## 1 rmse    standard        121.
```

```
poisson_preds %>% mae(truth = Pregnancies, estimate = .pred)
```

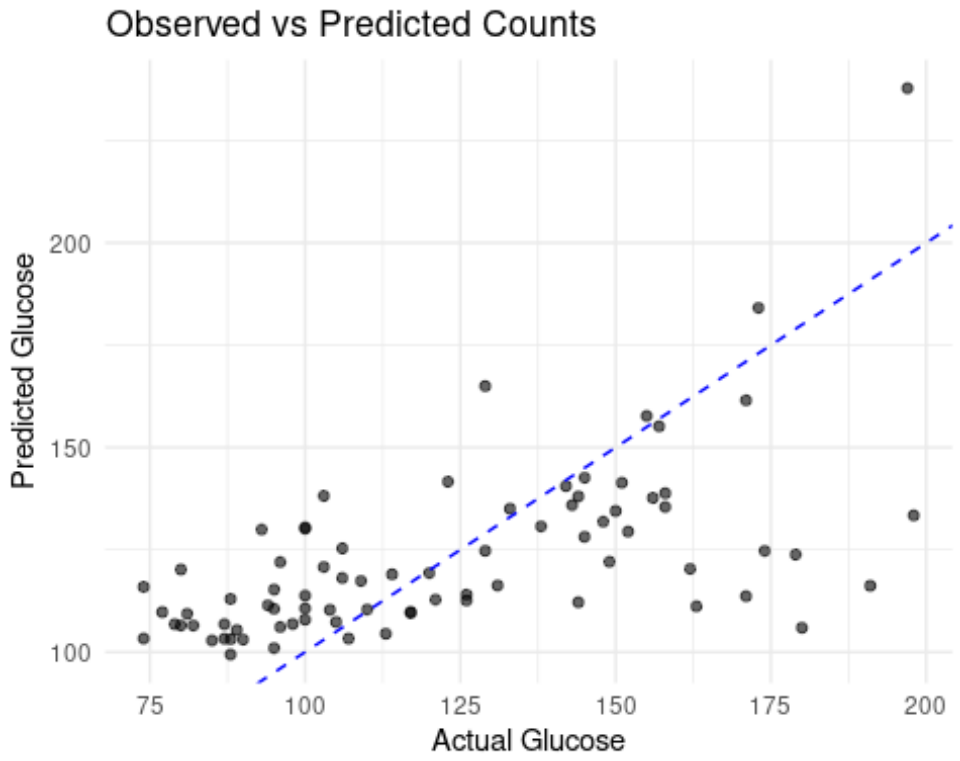
```
## # A tibble: 1 × 3
##   .metric .estimator .estimate
##   <chr>   <chr>          <dbl>
## 1 mae     standard        119.
```

```
poisson_preds %>% rsq(truth = Pregnancies, estimate = .pred)
```

```
## # A tibble: 1 × 3
##   .metric .estimator .estimate
##   <chr>   <chr>          <dbl>
## 1 rsq     standard      0.0932
```

```
ggplot(poisson_preds, aes(x = Glucose, y = .pred)) +
  geom_point(alpha = 0.6) +
  geom_abline(slope = 1, intercept = 0, linetype = "dashed", color = "blue")
+
  labs(title = "Observed vs Predicted Counts",
       x = "Actual Glucose",
```

```
       y = "Predicted Glucose") +
  theme_minimal()
```




```
poisson_model <- extract_fit_engine(poisson_fit)
summary(poisson_model)

##
## Call:
## stats::glm(formula = ..y ~ ., family = stats::poisson, data = data)
##
## Coefficients:
##                            Estimate Std. Error z value Pr(>|z|)
## (Intercept)               4.272e+00  3.569e-02 119.670  < 2e-16 ***
## Pregnancies              -1.425e-03  2.099e-03  -0.679    0.497
## BloodPressure             2.426e-03  4.485e-04   5.410 6.31e-08 ***
## SkinThickness             7.708e-04  6.592e-04   1.169    0.242
## Insulin                   9.097e-04  4.076e-05  22.319  < 2e-16 ***
## BMI                       6.393e-04  1.012e-03   0.632    0.528
## DiabetesPedigreeFunction  7.159e-02  1.569e-02   4.562 5.06e-06 ***
## Age                       4.470e-03  6.857e-04   6.519 7.09e-11 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for poisson family taken to be 1)
##
##     Null deviance: 2294.8  on 312  degrees of freedom
## Residual deviance: 1419.4  on 305  degrees of freedom
```

```
## AIC: 3507.4
##
## Number of Fisher Scoring iterations: 4

# You can also compute dispersion:
dispersion <- sum(residuals(poisson_model, type = "pearson")^2) /
poisson_model$df.residual
dispersion

## [1] 4.722521
```

Polynomial Regression (e.g., predict Glucose using polynomial of Age)

```
# Create the recipe using step_poly for Age
poly_recipe <- recipe(Glucose ~ Pregnancies + BloodPressure + SkinThickness +
                        Insulin + BMI + DiabetesPedigreeFunction + Age, data
= diabetes_train) %>%
  step_poly(Age, degree = 3)

# Specify a linear regression model
lm_spec <- linear_reg() %>%
  set_engine("lm")

# Build the workflow
lm_wf <- workflow() %>%
  add_recipe(poly_recipe) %>%
  add_model(lm_spec)

# Fit the model
lm_fit <- fit(lm_wf, data = diabetes_train)

# Predict and evaluate on the test set
predict(lm_fit, diabetes_test) %>%
  bind_cols(diabetes_test) %>%
  metrics(truth = Glucose, estimate = .pred)

## # A tibble: 3 × 3
##   .metric .estimator .estimate
##   <chr>   <chr>          <dbl>
## 1 rmse    standard      25.5
## 2 rsq     standard       0.399
## 3 mae     standard      19.6

poly_preds <- predict(lm_fit, diabetes_test) %>%
  bind_cols(diabetes_test)

poly_preds

## # A tibble: 79 × 10
##    .pred Pregnancies Glucose BloodPressure SkinThickness Insulin   BMI
##    <dbl>       <dbl>   <dbl>         <dbl>         <dbl>   <dbl> <dbl>
```

```
##  1 102.            1      89            66            23      94  28.1
##  2 134.           11     143            94            33     146  36.6
##  3 125.           13     145            82            19     110  22.2
##  4 138.            3     158            76            36     245  31.6
##  5  95.1           3      88            58            11      54  24.8
##  6 105.            3     180            64            25      70  34
##  7 155.            9     171           110            24     240  45.4
##  8 102.            5      88            66            21      23  24.4
##  9 131.            0     100            88            60     110  46.8
## 10 115.            2     100            66            20      90  32.9
## # ℹ 69 more rows
## # ℹ 3 more variables: DiabetesPedigreeFunction <dbl>, Age <dbl>, Outcome
<fct>
```

```r
poly_preds <- poly_preds %>%
  mutate(residual = Glucose - .pred)

ggplot(poly_preds, aes(x = .pred, y = residual)) +
  geom_point(alpha = 0.6) +
  geom_hline(yintercept = 0, color = "red", linetype = "dashed") +
  labs(title = "Residual Plot (Polynomial Regression)",
       x = "Predicted Glucose",
       y = "Residual (Observed - Predicted)") +
  theme_minimal()
```