Tidy Models Parameter Tuning

Setup

```
library(readr)
library(tidyverse)
```

```
-- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 --
v dplyr     1.1.4     v purrr     1.0.2
v forcats   1.0.0     v stringr   1.5.1
v ggplot2   3.5.1     v tibble    3.2.1
v lubridate 1.9.3     v tidyr     1.3.1
-- Conflicts ------------------------------------------ tidyverse_conflicts() --
x dplyr::filter() masks stats::filter()
x dplyr::lag()    masks stats::lag()
i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to becor
```

```
library(tidymodels)
```

```
-- Attaching packages --------------------------------------- tidymodels 1.2.0 --
v broom        1.0.6     v rsample      1.2.1
v dials        1.3.0     v tune         1.2.1
v infer        1.0.7     v workflows    1.1.4
v modeldata    1.4.0     v workflowsets 1.1.0
v parsnip      1.2.1     v yardstick    1.3.1
v recipes      1.1.0
-- Conflicts ------------------------------------------ tidymodels_conflicts() --
x scales::discard() masks purrr::discard()
x dplyr::filter()   masks stats::filter()
x recipes::fixed()  masks stringr::fixed()
x dplyr::lag()      masks stats::lag()
```

```
x yardstick::spec() masks readr::spec()
x recipes::step()   masks stats::step()
* Search for functions across packages at https://www.tidymodels.org/find/
```

```
library(readr)
library(dplyr)
library(ggplot2)
library(glmnet)
```

```
Loading required package: Matrix

Attaching package: 'Matrix'

The following objects are masked from 'package:tidyr':

    expand, pack, unpack

Loaded glmnet 4.1-8
```

```
library(MASS)
```

```

Attaching package: 'MASS'

The following object is masked from 'package:dplyr':

    select
```

```
library(GGally)
```

```
Registered S3 method overwritten by 'GGally':
  method from
  +.gg   ggplot2
```

```
library(discrim)
```

```

Attaching package: 'discrim'
```

```
The following object is masked from 'package:dials':

    smoothness
```

```
library(poissonreg)
```

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, ~
```

```r
# 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"
```

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

```
NULL
```

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

```
No Diabetes    Diabetes
        500         268
```

```r
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
```

Exploratory Analysis

```r
glimpse(diabetes)
```

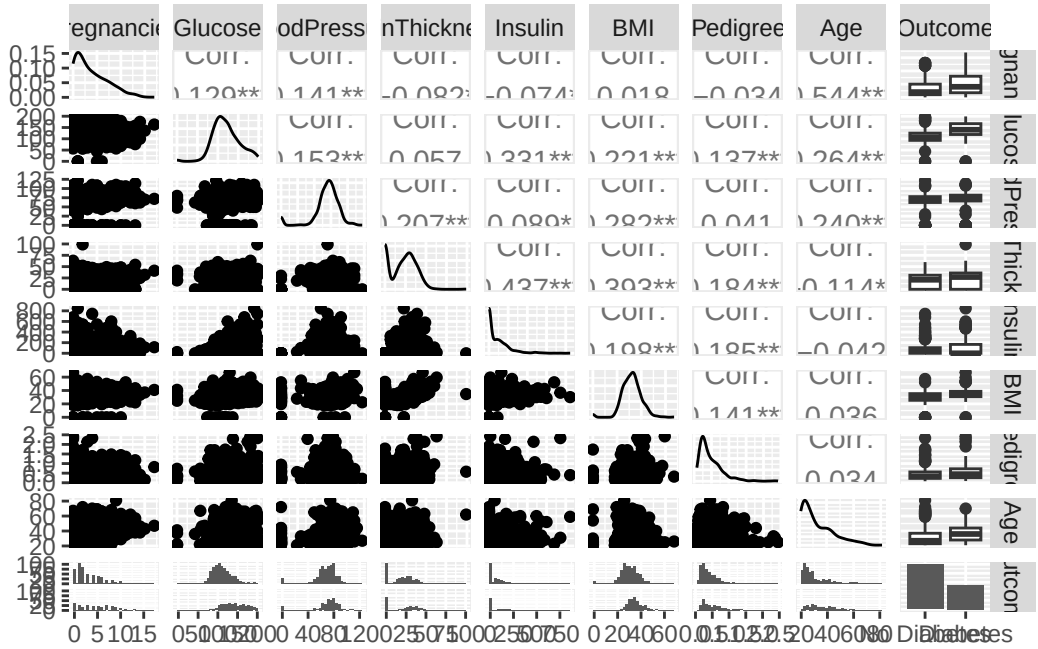
```
Rows: 768
Columns: 9
$ Pregnancies              <dbl> 6, 1, 8, 1, 0, 5, 3, 10, 2, 8, 4, 10, 10, 1, ~
```

```
$ 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,~
```

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`.
```



```r
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
```

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

```r
# Remove rows where any of the columns Glucose, BloodPressure, SkinThickness, Insulin, or BMI
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 x 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>
```

```r
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
```

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

```r
# 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)
```

1. 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 x 3
  .metric  .estimator .estimate
  <chr>    <chr>          <dbl>
1 accuracy binary         0.696
2 kap      binary         0.298
```

```
tidy(log_fit)
```

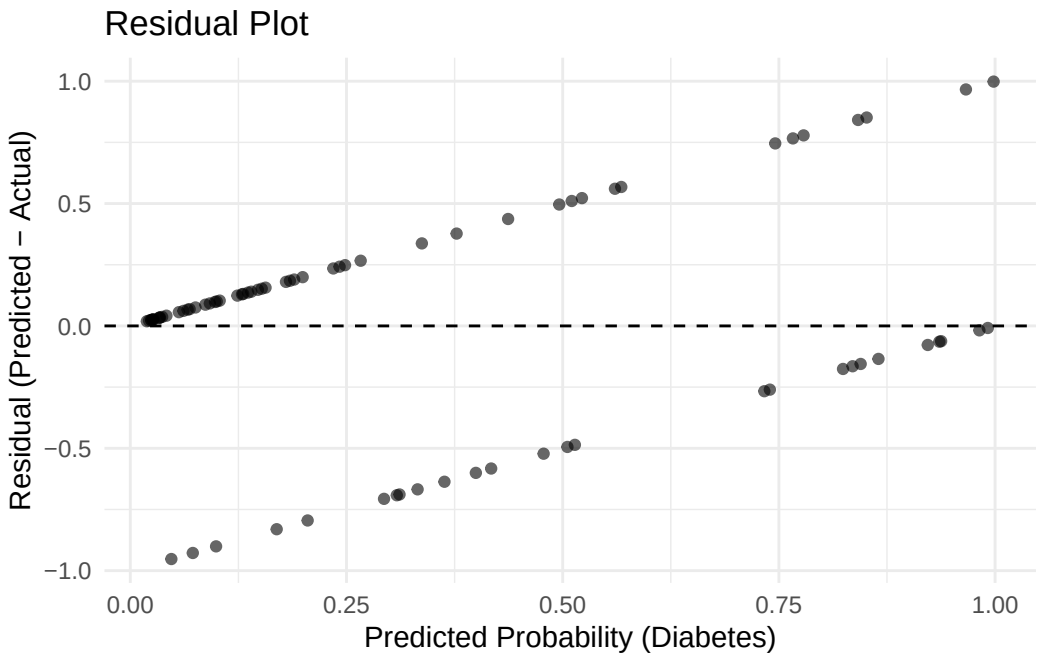
```
# A tibble: 9 x 5
  term                      estimate std.error statistic  p.value
  <chr>                        <dbl>     <dbl>     <dbl>    <dbl>
1 (Intercept)              -11.2       1.48       -7.61  2.78e-14
2 Pregnancies                0.126     0.0646      1.96  5.03e- 2
3 Glucose                    0.0428    0.00682     6.27  3.60e-10
4 BloodPressure              0.0119    0.0149      0.797 4.25e- 1
5 SkinThickness              0.00537   0.0198      0.271 7.86e- 1
```

```
6 Insulin                   0.00100   0.00158     0.633 5.27e- 1
7 BMI                       0.0551    0.0325      1.70  8.96e- 2
8 DiabetesPedigreeFunction  1.83      0.521       3.51  4.40e- 4
9 Age                       0.0195    0.0211      0.924 3.55e- 1
```

```r
# 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 x 12
  `.pred_No Diabetes` .pred_Diabetes .pred_class Pregnancies Glucose
                <dbl>          <dbl> <fct>             <dbl>   <dbl>
1              0.0644          0.936 Diabetes              0     137
2              0.831           0.169 No Diabetes           1     115
3              0.963           0.0368 No Diabetes          1     101
4              0.0624          0.938 Diabetes              8     176
5              0.820           0.180 No Diabetes           2     110
6              0.734           0.266 No Diabetes           4     123
# i 7 more variables: BloodPressure <dbl>, SkinThickness <dbl>, Insulin <dbl>,
#   BMI <dbl>, DiabetesPedigreeFunction <dbl>, Age <dbl>, Outcome <fct>
```

```r
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()
```

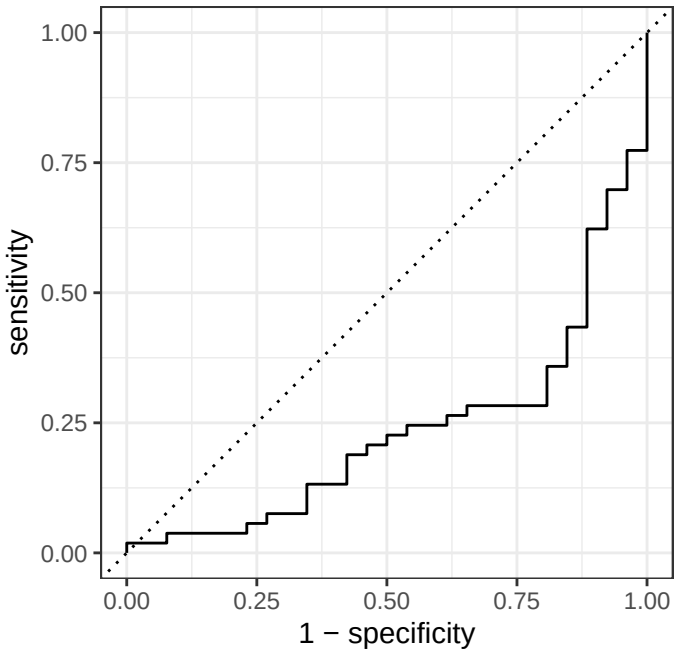


Confusion matrix

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

```
             Truth
Prediction    No Diabetes Diabetes
  No Diabetes          42       13
  Diabetes             11       13
```

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




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

```
# A tibble: 1 x 3
  .metric .estimator .estimate
  <chr>   <chr>          <dbl>
1 roc_auc binary         0.233
```

2. Multinomial Logistic Regression

```
# Simulate a 3-class outcome
set.seed(123)
diabetes$Outcome3 <- factor(sample(c("Low", "Medium", "High"), nrow(diabetes), replace = TRU
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
```

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

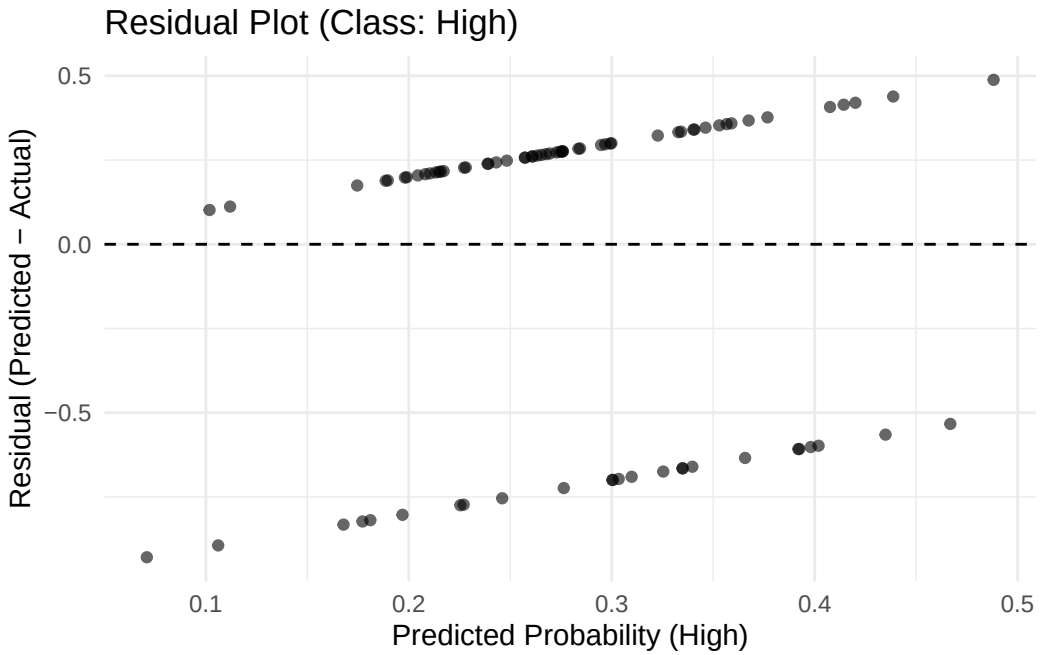
```
# A tibble: 2 x 3
  .metric  .estimator .estimate
  <chr>    <chr>          <dbl>
1 accuracy multiclass   0.338
2 kap      multiclass  -0.00165
```

```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 x 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()
```



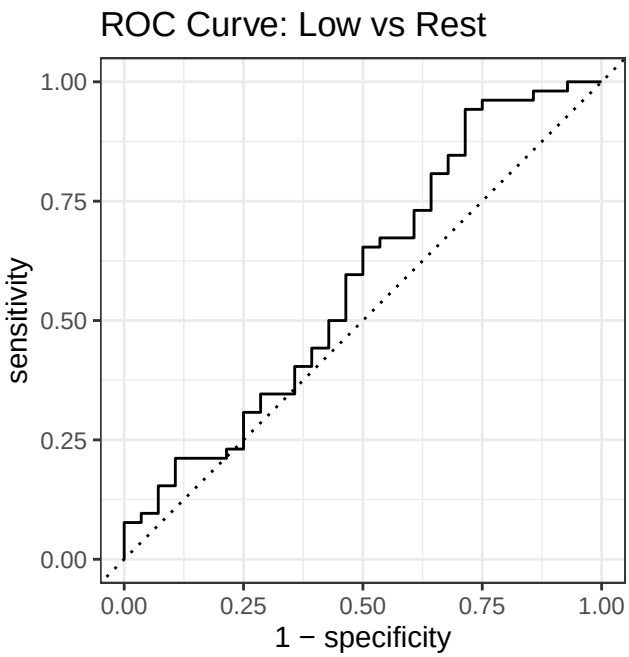
Confusion matrix

```r
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
```

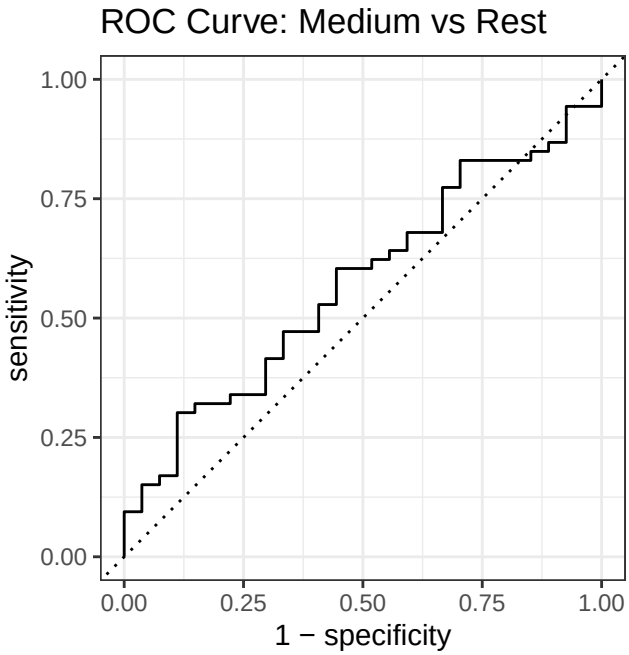
```
# Add binary columns for each class (one-vs-rest approach)
multi_preds <- multi_preds %>%
  mutate(
    truth_Low = if_else(Outcome3 == "Low", "Low", "Other") %>% factor(levels = c("Other", "L
    truth_Medium = if_else(Outcome3 == "Medium", "Medium", "Other") %>% factor(levels = c("O
    truth_High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

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

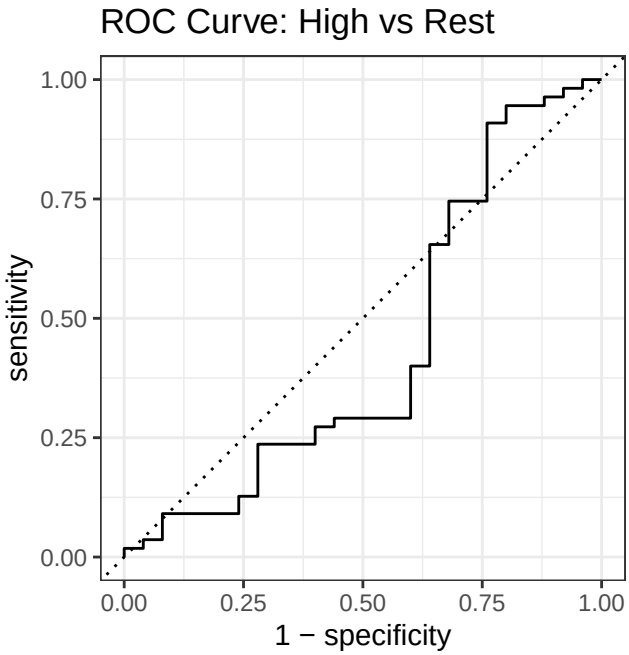



```
# ROC for "Medium"
multi_preds %>%
```

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




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



3. 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)
```

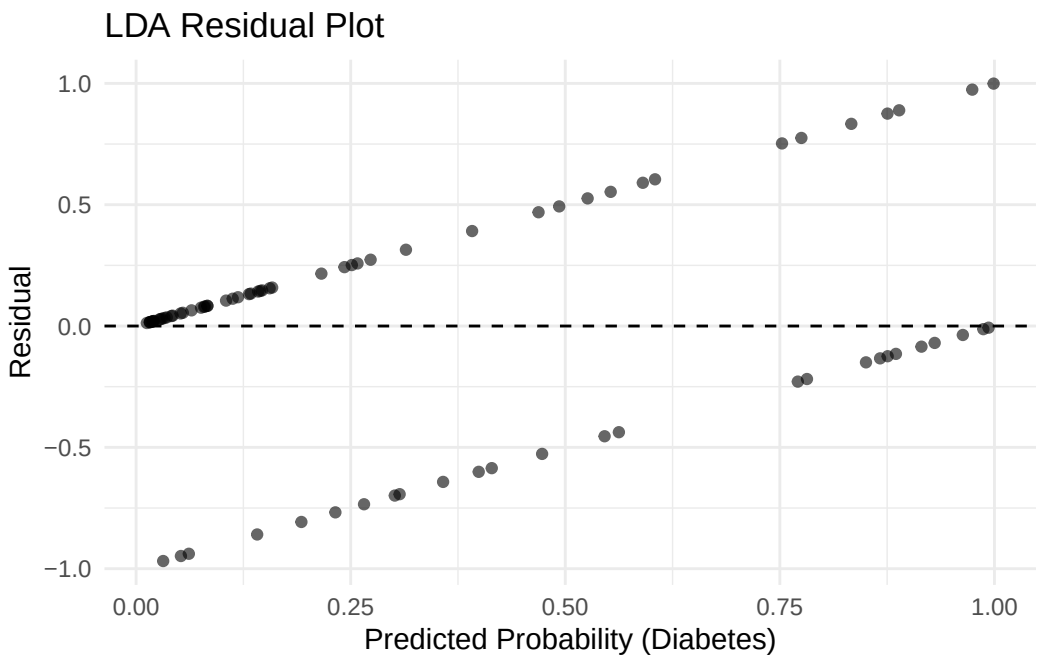
```
# A tibble: 2 x 3
  .metric  .estimator .estimate
  <chr>    <chr>          <dbl>
```

```
1 accuracy binary         0.696
2 kap      binary         0.298
```

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

head(lda_preds)
```

```
# A tibble: 6 x 12
  `.pred_No Diabetes` .pred_Diabetes .pred_class Pregnancies Glucose
                <dbl>          <dbl> <fct>             <dbl>   <dbl>
1              0.0850         0.915  Diabetes              0     137
2              0.859          0.141  No Diabetes           1     115
3              0.973          0.0272 No Diabetes           1     101
4              0.0368         0.963  Diabetes              8     176
5              0.853          0.147  No Diabetes           2     110
6              0.742          0.258  No Diabetes           4     123
# 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()
```

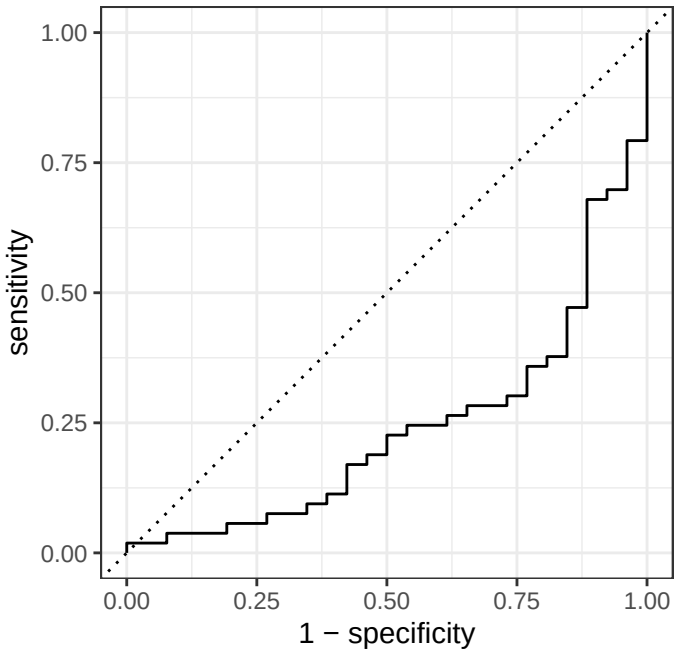


Confusion matrix

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

```
             Truth
Prediction    No Diabetes Diabetes
  No Diabetes          42       13
  Diabetes             11       13
```

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




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

```
# A tibble: 1 x 3
  .metric .estimator .estimate
  <chr>   <chr>          <dbl>
1 roc_auc binary         0.239
```

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

```
# A tibble: 2 x 3
  .metric  .estimator .estimate
  <chr>    <chr>          <dbl>
1 accuracy binary         0.696
2 kap      binary         0.298
```

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

```
# A tibble: 1 x 3
  .metric   .estimator .estimate
  <chr>     <chr>          <dbl>
1 precision binary         0.764
```

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

```
# A tibble: 1 x 3
  .metric .estimator .estimate
  <chr>   <chr>          <dbl>
1 recall  binary         0.792
```

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

```
# A tibble: 1 x 3
  .metric .estimator .estimate
  <chr>   <chr>          <dbl>
1 f_meas  binary         0.778
```

4. Poisson Regression (predict count outcome: Pregnancies)

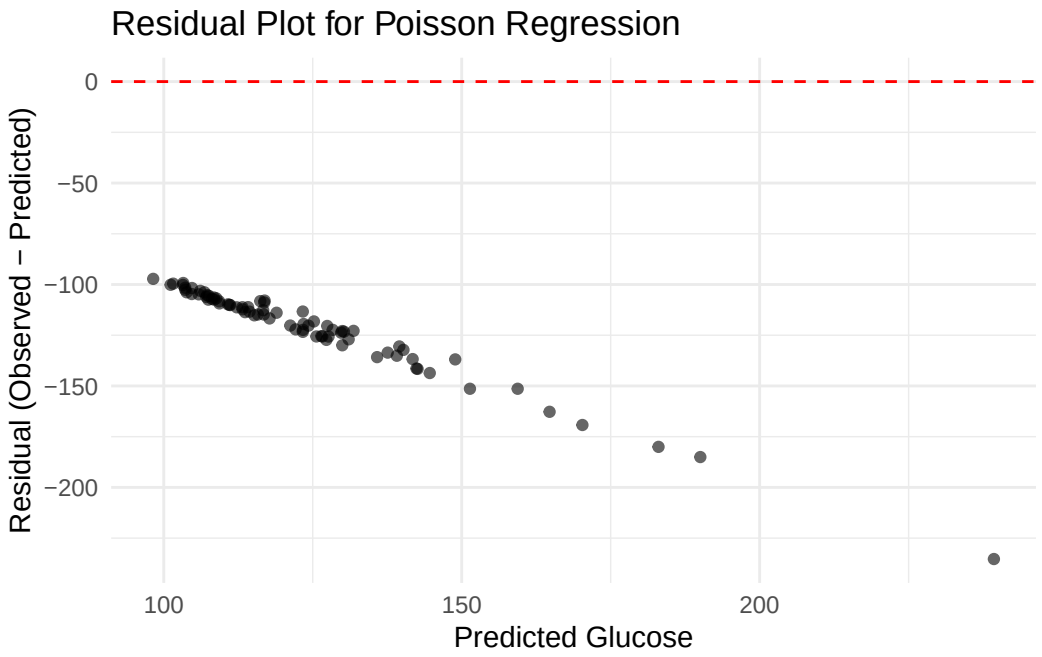
```
poisson_recipe <- recipe(Glucose ~ Pregnancies + BloodPressure + SkinThickness +
                         Insulin + BMI + DiabetesPedigreeFunction + Age, data = diabetes_tra

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 x 3
  .metric .estimator .estimate
  <chr>   <chr>          <dbl>
1 rmse    standard    123.
2 rsq     standard      0.0686
3 mae     standard    121.
```

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

poisson_preds
```

```
# A tibble: 79 x 10
   .pred Pregnancies Glucose BloodPressure SkinThickness Insulin   BMI
   <dbl>       <dbl>   <dbl>         <dbl>         <dbl>   <dbl> <dbl>
 1 127.            0     137            40            35     168  43.1
 2 113.            1     115            70            30      96  34.6
 3  98.2           1     101            50            15      36  24.2
 4 159.            8     176            90            34     300  33.7
 5 117.            2     110            74            29     125  32.4
 6 123.            4     123            80            15     176  32
 7 105.            0      95            85            25      36  37.4
 8 125.            7     160            54            32     175  30.5
 9 108.            1      88            30            42      99  55
10 123.            1     117            88            24     145  34.5
# i 69 more rows
# i 3 more variables: DiabetesPedigreeFunction <dbl>, Age <dbl>, Outcome <fct>
```

```r
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 x 3
  .metric .estimator .estimate
  <chr>   <chr>          <dbl>
1 rmse    standard        123.
```

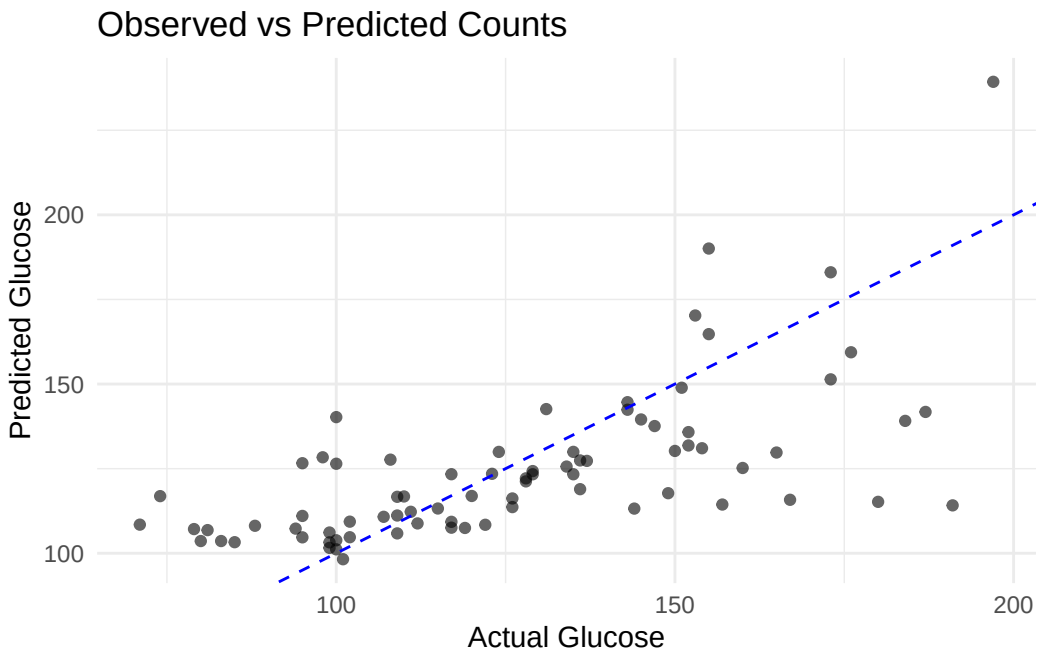
```
poisson_preds %>% mae(truth = Pregnancies, estimate = .pred)
```

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

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

```
# A tibble: 1 x 3
  .metric .estimator .estimate
  <chr>   <chr>          <dbl>
1 rsq     standard      0.0686
```

```r
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()
```




```r
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)             4.319e+00  3.498e-02 123.457  < 2e-16 ***
Pregnancies             4.915e-03  2.130e-03   2.307 0.021043 *
BloodPressure           1.714e-03  4.738e-04   3.617 0.000298 ***
SkinThickness           1.494e-03  6.551e-04   2.280 0.022587 *
Insulin                 9.790e-04  4.229e-05  23.152  < 2e-16 ***
```

```
BMI                      9.615e-04  1.038e-03   0.926 0.354467
DiabetesPedigreeFunction 4.809e-02  1.601e-02   3.004 0.002667 **
Age                      2.750e-03  7.148e-04   3.847 0.000120 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for poisson family taken to be 1)

    Null deviance: 2406.3  on 312  degrees of freedom
Residual deviance: 1529.0  on 305  degrees of freedom
AIC: 3613.6

Number of Fisher Scoring iterations: 4
```

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

```
[1] 5.14721
```

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

```r
# Create the recipe using step_poly for Age
poly_recipe <- recipe(Glucose ~ Pregnancies + BloodPressure + SkinThickness +
                        Insulin + BMI + DiabetesPedigreeFunction + Age, data = diabetes_tra
  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)
```

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

```
# A tibble: 3 x 3
  .metric .estimator .estimate
  <chr>   <chr>          <dbl>
1 rmse    standard      22.0
2 rsq     standard       0.466
3 mae     standard      16.3
```

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

poly_preds
```

```
# A tibble: 79 x 10
   .pred Pregnancies Glucose BloodPressure SkinThickness Insulin   BMI
   <dbl>       <dbl>   <dbl>         <dbl>         <dbl>   <dbl> <dbl>
 1 131.            0     137            40            35     168  43.1
 2 117.            1     115            70            30      96  34.6
 3  97.6           1     101            50            15      36  24.2
 4 154.            8     176            90            34     300  33.7
 5 119.            2     110            74            29     125  32.4
 6 129.            4     123            80            15     176  32
 7 102.            0      95            85            25      36  37.4
 8 130.            7     160            54            32     175  30.5
 9 108.            1      88            30            42      99  55
10 129.            1     117            88            24     145  34.5
# 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()
```