Report of predictive model of WIF data

Yige Wang

2024-04-03

Introduction

Dataset

The data comes form experiments (provided by Karl, my collaborator). Here shows original data.

```
# A tibble: 6 x 5
  cell_line treatment name    conc gene_expression
  <chr>     <chr>     <chr>  <dbl>           <dbl>
1 Wild-type Placebo   GL-XIb     0            5.05
2 Wild-type Placebo   GL-cDZ     0            5.92
3 Wild-type Placebo   GL-XIb     1            4.15
4 Wild-type Placebo   GL-cDZ     1            3.34
5 Wild-type Placebo   GL-XIb     2            6.67
6 Wild-type Placebo   GL-cDZ     2            5.54
```

The meaning of column names is shown in Table 1

Table 1: Meaning of column name

Name	Meaning
cell_line	cell type
treatment	treatment to sample, placebo or using Activating Factor
name	name of each sample
conc	concentration of Activating Factor or saline
gene_expression	rate of gene expression

Research question

The report is about how to analyze the effect of a new treatment on gene expression, specifically looking at how the treatment influences the effect of a growth factor on gene expression and how to build a predictive model of gene expression.

Methods

Clean Data

First of all I clean the data by correct the name of category variables, because there is same Letter capitalization error. I also add a column called "case" which is combination of variable "cell_line" and variable "treatment"

```
# A tibble: 6 x 6
  cell_line treatment name    conc gene_expression case
  <chr>     <chr>     <chr>  <dbl>           <dbl> <chr>
1 Wild-Type Placebo   Gl-Xib     0            5.05 Wild-Type&Placebo
2 Wild-Type Placebo   Gl-Cdz     0            5.92 Wild-Type&Placebo
3 Wild-Type Placebo   Gl-Xib     1            4.15 Wild-Type&Placebo
4 Wild-Type Placebo   Gl-Cdz     1            3.34 Wild-Type&Placebo
5 Wild-Type Placebo   Gl-Xib     2            6.67 Wild-Type&Placebo
6 Wild-Type Placebo   Gl-Cdz     2            5.54 Wild-Type&Placebo
```

Analysis

First of all, to build a predictive model, I draw plots to have a overview of the relationship between variables.

The difference among groups

Here I use t-test and Analysis of Variance to find the main relationship of variables.

term	df	sumsq	meansq	statistic	p.value
cell_line	1.00	104.53	104.53	2.84	9.59×10^{-2}
treatment	1.00	2,141.69	2,141.69	58.16	4.35×10^{-11}
name	5.00	3,088.32	617.66	16.77	2.65×10^{-11}
Residuals	80.00	2,946.05	36.83	NA	NA

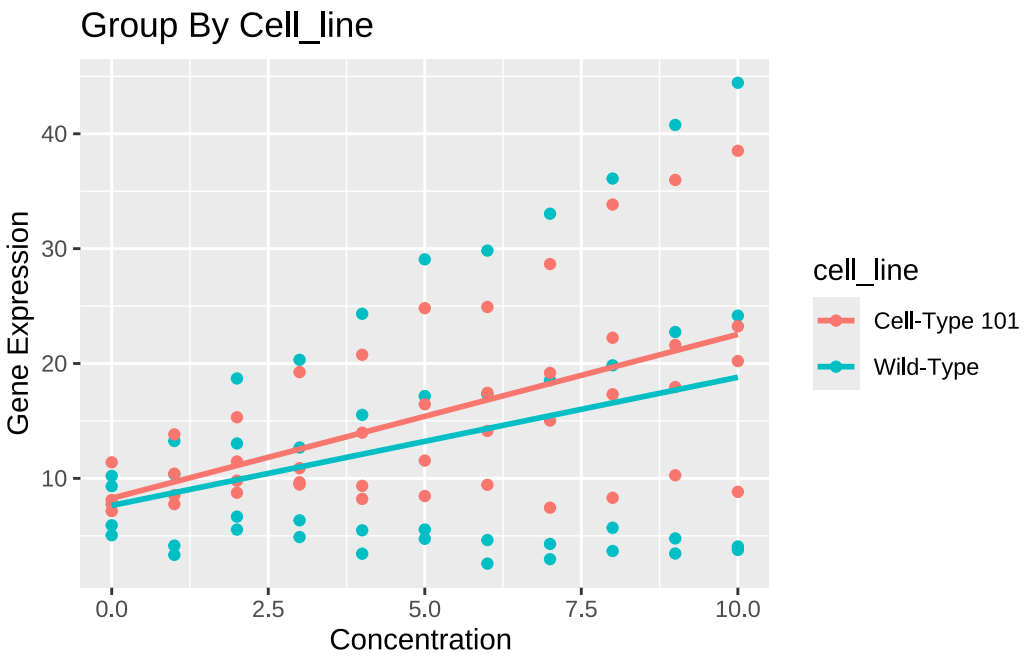


Figure 1: The linear relationship between concentration and gene expression grouped by cell line

It shows cell_line is not significant factor while other variables have a significant effort on the gene expression.

Maybe that is because we add "name" as a factor, which may reduce the influence of cell_line.

term	df	sumsq	meansq	statistic	p.value
treatment	1.00	2,141.69	2,141.69	30.17	4.05×10^{-7}
cell_line	1.00	104.53	104.53	1.47	2.28×10^{-1}
Residuals	85.00	6,034.38	70.99	NA	NA

The cell_line is totally not significant at all.

To double verify it, I built a linear regression model

term	estimate	std.error	statistic	p.value
(Intercept)	8.27	2.52	3.27	1.54×10^{-3}
conc	1.43	0.43	3.34	1.24×10^{-3}
cell_lineWild-Type	−0.63	3.57	−0.18	8.61×10^{-1}

conc:cell_lineWild-Type	−0.31	0.60	−0.51	6.08×10^{-1}

Here I find the name is related to the treatment and cell_line, which means it is redundancy to analyse total three column. Hence, there just needs to analyse the treatment, cell_line and the combined, so I did not get linear model for that.

term	estimate	std.error	statistic	p.value
(Intercept)	9.86	2.31	4.26	5.52×10^{-5}
conc	1.36	0.39	3.47	8.42×10^{-4}
caseCell-Type 101&Placebo	−3.18	3.27	−0.97	3.34×10^{-1}
caseWild-Type&Activating Factor 42	0.13	3.27	0.04	9.68×10^{-1}
caseWild-Type&Placebo	−4.56	3.27	−1.39	1.67×10^{-1}
conc:caseCell-Type 101&Placebo	0.14	0.55	0.25	8.00×10^{-1}
conc:caseWild-Type&Activating Factor 42	1.02	0.55	1.84	6.98×10^{-2}
conc:caseWild-Type&Placebo	−1.50	0.55	−2.71	8.29×10^{-3}

Hence cell_line is not significant. Just like the figure Figure 1 shows, they are close lines.

According to the relationship, I build a linear model with formula:

gene_expression ~ conc + case + case * conc.

Building model

Split data into train data and test data

Split data into train data, test data and use train data to generate the Cross-Validation for model training.

V-fold cross-validation is a robust method for assessing the performance of a statistical model.

Modeling

According to the formula:

gene_expression ~ conc + case + case * conc,

I build the model.

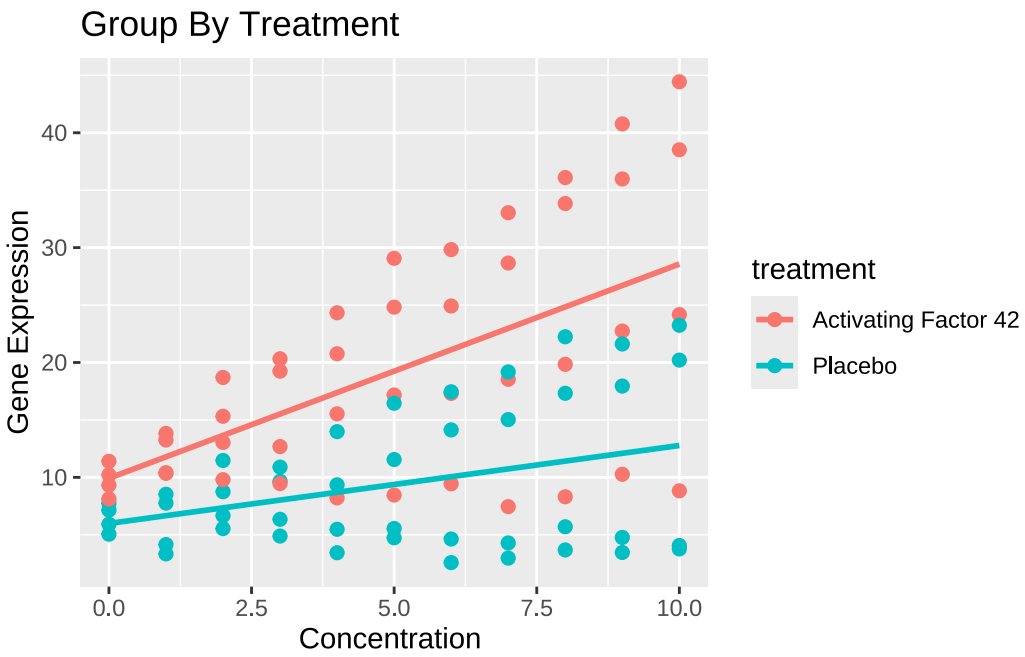


Figure 2: The linear relationship between concentration and gene expression grouped by treatment

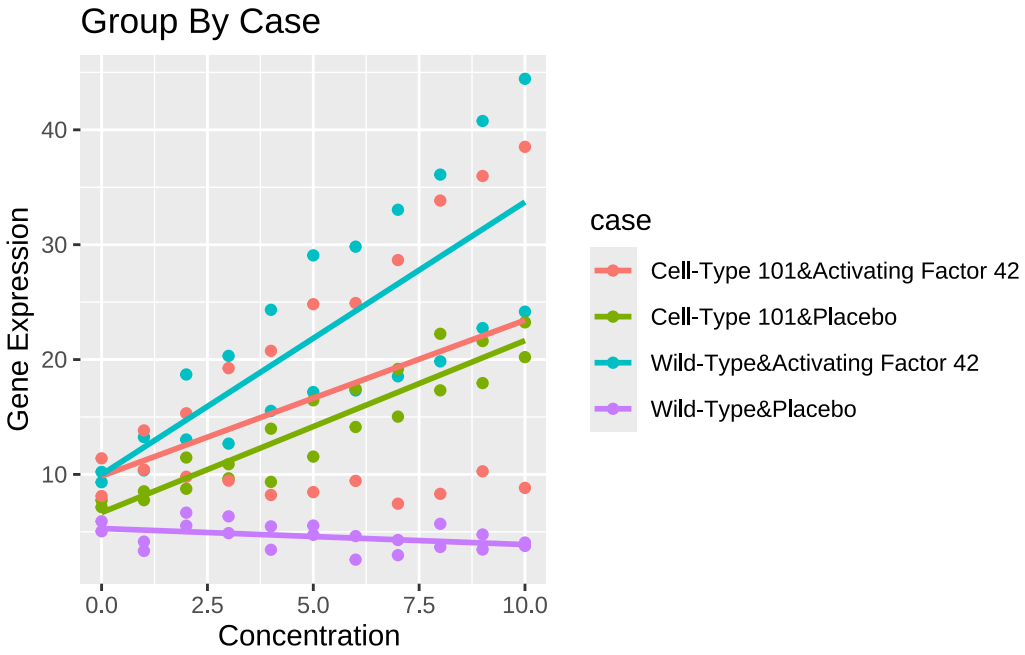
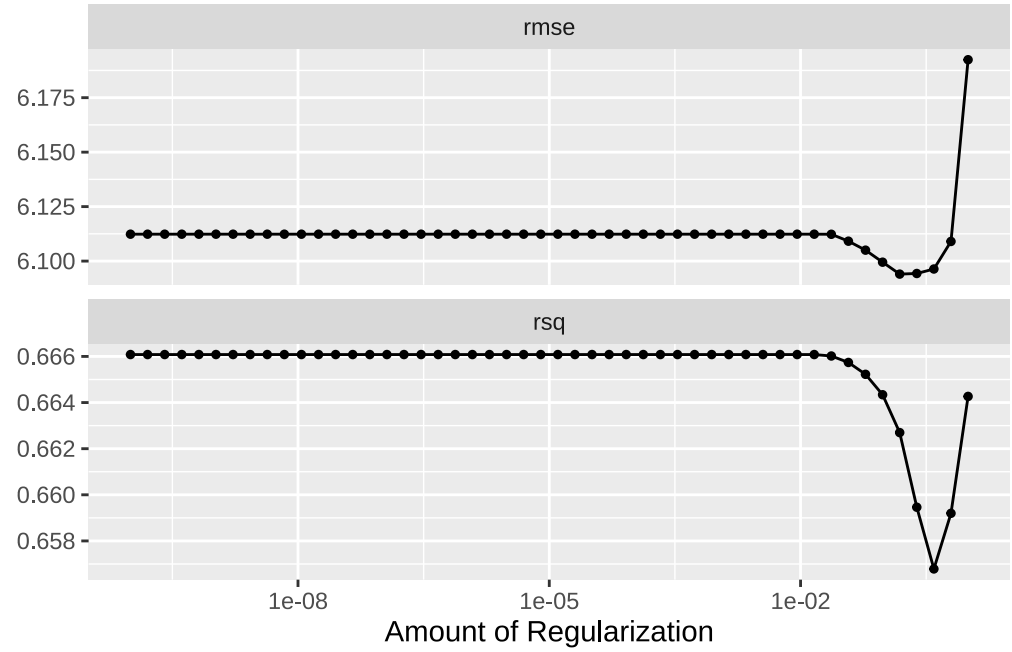


Figure 3: The linear relationship between concentration and gene expression grouped by case

```
== Workflow ====================================================================
Preprocessor: Recipe
Model: linear_reg()

-- Preprocessor ----------------------------------------------------------------
5 Recipe Steps

* step_dummy()
* step_normalize()
* step_interact()
* step_interact()
* step_interact()

-- Model -----------------------------------------------------------------------
Linear Regression Model Specification (regression)

Main Arguments:
  penalty = tune()
  mixture = 1

Computational engine: glmnet
```

Tune the model

```
# A tibble: 100 x 7
    penalty .metric .estimator  mean     n std_err .config
      <dbl> <chr>   <chr>      <dbl> <int>   <dbl> <chr>
 1 1   e-10 rmse    standard   6.11     10  0.839  Preprocessor1_Model01
 2 1   e-10 rsq     standard   0.666    10  0.0824 Preprocessor1_Model01
 3 1.60e-10 rmse    standard   6.11     10  0.839  Preprocessor1_Model02
 4 1.60e-10 rsq     standard   0.666    10  0.0824 Preprocessor1_Model02
 5 2.56e-10 rmse    standard   6.11     10  0.839  Preprocessor1_Model03
 6 2.56e-10 rsq     standard   0.666    10  0.0824 Preprocessor1_Model03
 7 4.09e-10 rmse    standard   6.11     10  0.839  Preprocessor1_Model04
 8 4.09e-10 rsq     standard   0.666    10  0.0824 Preprocessor1_Model04
 9 6.55e-10 rmse    standard   6.11     10  0.839  Preprocessor1_Model05
10 6.55e-10 rsq     standard   0.666    10  0.0824 Preprocessor1_Model05
# i 90 more rows
```



Find best model we get

```
# A tibble: 5 x 7
  penalty .metric .estimator  mean     n std_err .config
    <dbl> <chr>   <chr>      <dbl> <int>   <dbl> <chr>
1  0.153  rmse    standard    6.09    10   0.796 Preprocessor1_Model46
2  0.244  rmse    standard    6.09    10   0.769 Preprocessor1_Model47
3  0.391  rmse    standard    6.10    10   0.736 Preprocessor1_Model48
4  0.0954 rmse    standard    6.10    10   0.814 Preprocessor1_Model45
5  0.0596 rmse    standard    6.11    10   0.826 Preprocessor1_Model44
```

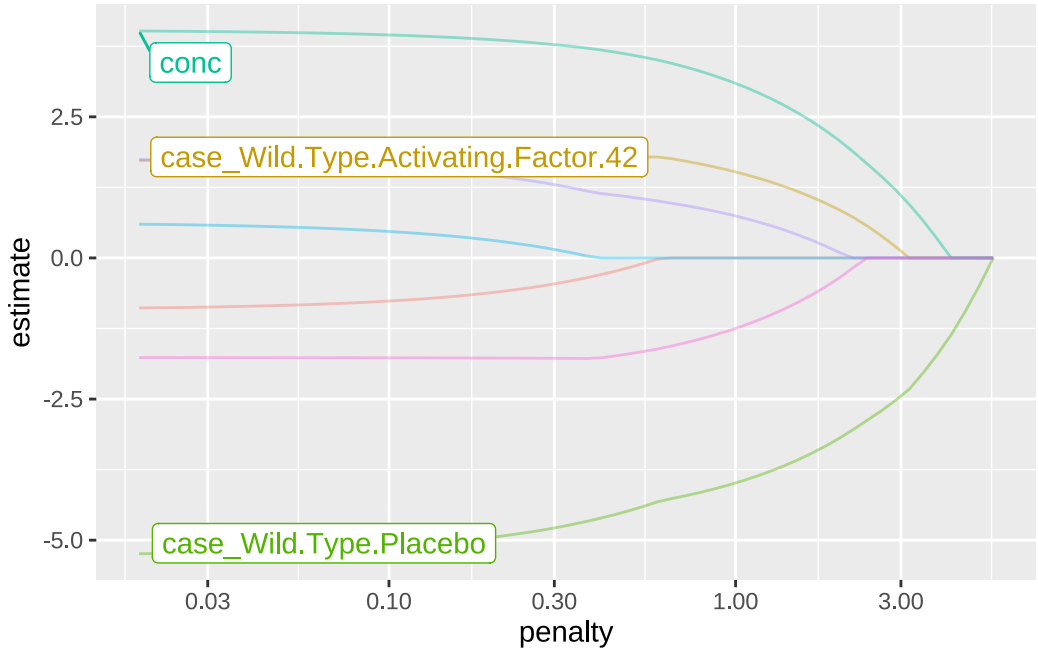
```
# A tibble: 1 x 2
  penalty .config
    <dbl> <chr>
1   0.153 Preprocessor1_Model46
```

Fit final model

```
== Workflow ====================================================================
Preprocessor: Recipe
Model: linear_reg()
```

```
-- Preprocessor ----------------------------------------------------------------
5 Recipe Steps

* step_dummy()
* step_normalize()
* step_interact()
* step_interact()
* step_interact()

-- Model -----------------------------------------------------------------------
Linear Regression Model Specification (regression)

Main Arguments:
  penalty = 0.152641796717524
  mixture = 1

Computational engine: glmnet
```

Result of fitting:



Results

Relationship between variables

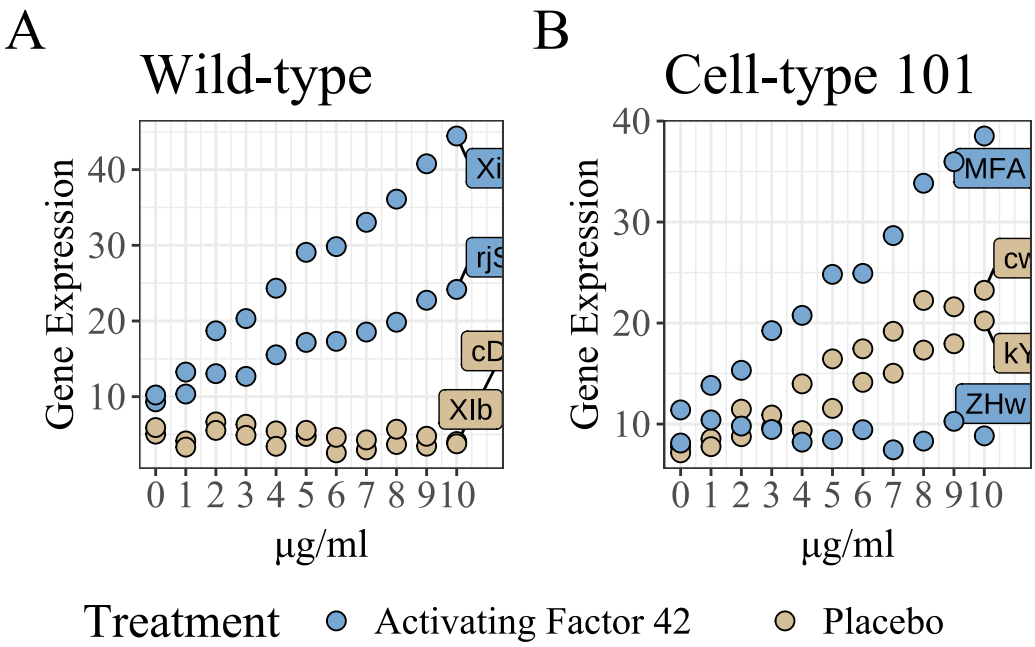


Figure 4: The linear relationship between concentration and gene expression

As the plot Figure 4 shown, obviously there is a linear relation between concentration and gene expression.

According to Figure 1 Figure 2 Figure 3, I find:

- The treatment only effects on the slope of gene expression.
- The Activating Factor will amplify the effort of concentration on gene expression.
- The Activating Factor (one of treatment) has significant influence on the Wild-Type while it does not work significantly on the Cell-Type 101.

Predictive model

The parameter of the best linear model I got are

```
Warning: `pull_workflow_fit()` was deprecated in workflows 0.2.3.
i Please use `extract_fit_parsnip()` instead.
```

```
# A tibble: 8 x 3
  term                                          estimate penalty
  <chr>                                            <dbl>   <dbl>
1 (Intercept)                                      14.1    0.153
2 conc                                              3.91   0.153
3 case_Cell.Type.101.Placebo                       -0.684  0.153
4 case_Wild.Type.Activating.Factor.42               1.75   0.153
5 case_Wild.Type.Placebo                           -5.02   0.153
6 conc_x_case_Cell.Type.101.Placebo                 0.385  0.153
7 conc_x_case_Wild.Type.Activating.Factor.42        1.53   0.153
8 conc_x_case_Wild.Type.Placebo                    -1.77   0.153
```

Here "conc_x_case_Cell.Type.101.Placebo" means the interactive term.

And the rmse and rsq of the model are

```
# A tibble: 2 x 4
  .metric .estimator .estimate .config
  <chr>   <chr>          <dbl> <chr>
1 rmse    standard       5.68  Preprocessor1_Model1
2 rsq     standard       0.728 Preprocessor1_Model1
```

Discussion

As the results shows, the root mean square error (RMSE) indicates the average deviation of the predicted gene expression values from the actual values. The R-squared value suggests that approximately 70% of the variability in gene expression is explained by the model, which is a reasonable fit given the complexity of biological data.

I just consider one linear model as predictive model there are some point could make the model better in the future:

- Including More Variables: Incorporating additional variables such as time points, different cell lines, and other treatment types could provide a more comprehensive understanding.
- Non-linear Models: Exploring non-linear models or machine learning techniques may capture more complex relationships in the data.
- Validation: Applying the model to an independent dataset to validate its predictive capability and generalizability.

Appendix

```
pacman::p_load(tidyverse, tidymodels, textrecipes, targets, showtext, readxl)
## Add font
  font_add(
    family = "times",
    regular = here::here(
      "template","Times New Roman.ttf"
    )
  )
  showtext_auto()
tar_load(WIF_file)
head(read_excel(WIF_file))
tar_load(WIF_data)
tar_load(point_plots)
tar_load(analysis_tabs)
head(WIF_data)
point_plots$GroupByCell_line
point_plots$GroupByTreatment
point_plots$GroupByCase
analysis_tabs$avo_totalGroup
analysis_tabs$avo_treatment_cell_line
analysis_tabs$lm_conc_cell_line
analysis_tabs$lm_conc_case
set.seed(114514)
WIF_data_modeling <-
  WIF_data |>
  dplyr::select(gene_expression, conc, case)

WIF_split <- initial_split(WIF_data_modeling, strata = gene_expression)
WIF_train <- training(WIF_split)
WIF_test <- testing(WIF_split)
WIF_cv <- vfold_cv(WIF_train)
WIF_recipe <-
recipe(gene_expression ~ conc + case, data = WIF_train) |>
    step_dummy(all_nominal()) |>
    step_normalize(all_numeric(), -all_outcomes()) |>
    step_interact(terms = ~ conc:case_Cell.Type.101.Placebo) |>
    step_interact(terms = ~ conc:case_Wild.Type.Activating.Factor.42) |>
    step_interact(terms = ~ conc:case_Wild.Type.Placebo)
```

```
WIF_model <- linear_reg(penalty = tune(), mixture = 1) |>
  set_mode("regression") |>
  set_engine("glmnet")

WIF_wf <- workflow(WIF_recipe, WIF_model)
WIF_wf
WIF_grid <- grid_regular(penalty(), levels = 50)

WIF_tune <- tune_grid(
  WIF_wf,
  resamples = WIF_cv,
  grid = WIF_grid
  )
collect_metrics(WIF_tune)
WIF_tune |> autoplot()
show_best(WIF_tune, metric = "rmse")
penalty <- select_best(WIF_tune, metric = "rmse")
penalty
WIF_wf <- WIF_wf |>
  finalize_workflow(penalty)
WIF_wf
WIF_fit <- WIF_wf |> fit(WIF_train)
WIF_fit |> extract_fit_engine() |> autoplot()
tar_read(conference_plot)
WIF_fit |>
  pull_workflow_fit() |>
  tidy()
last_fit(WIF_wf, WIF_split) |> collect_metrics()
pacman::p_load(tidyverse, targets, lubridate, gt)
theme_set(theme_bw())
"IMRaD_Report.qmd"
```