Tidy Models Parameter Tuning

Setup

```r
suppressPackageStartupMessages(library(tidyverse))
suppressPackageStartupMessages(library(tidymodels))
suppressPackageStartupMessages(library(ggformula))
suppressPackageStartupMessages(library(GGally))
suppressPackageStartupMessages(library(future))
library(ISLR2)

## Multitasking setup (old)
#suppressPackageStartupMessages(library(doMC))
#cores <- detectCores(logical=TRUE)
#cat('Available cores: ',cores,'\n')
#registerDoMC(cores=cores-1)

# New multitasking setup
cores <- availableCores()
cat('Available cores: ',cores,'\n')
```

```
Available cores:  8
```

```r
plan(strategy=multisession, workers=cores-1)

# Load data and setup training/test/fold partitions
set.seed(2025)
hitters <- na.omit(Hitters)

hitters_split <- initial_split(hitters, strata = "Salary")
```

```
hitters_train <- training(hitters_split)
hitters_test <- testing(hitters_split)

hitters_fold <- vfold_cv(hitters_train, v = 10)
```

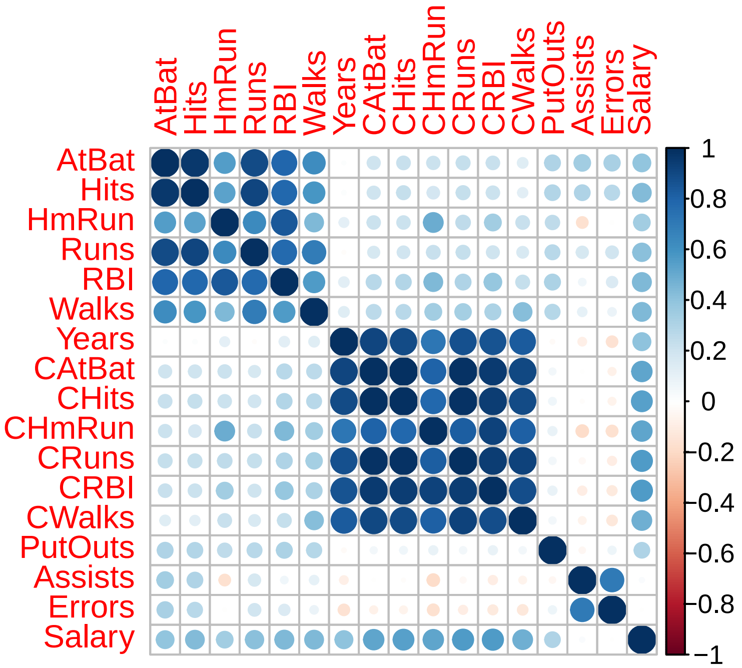
Source

Exploratory analysis

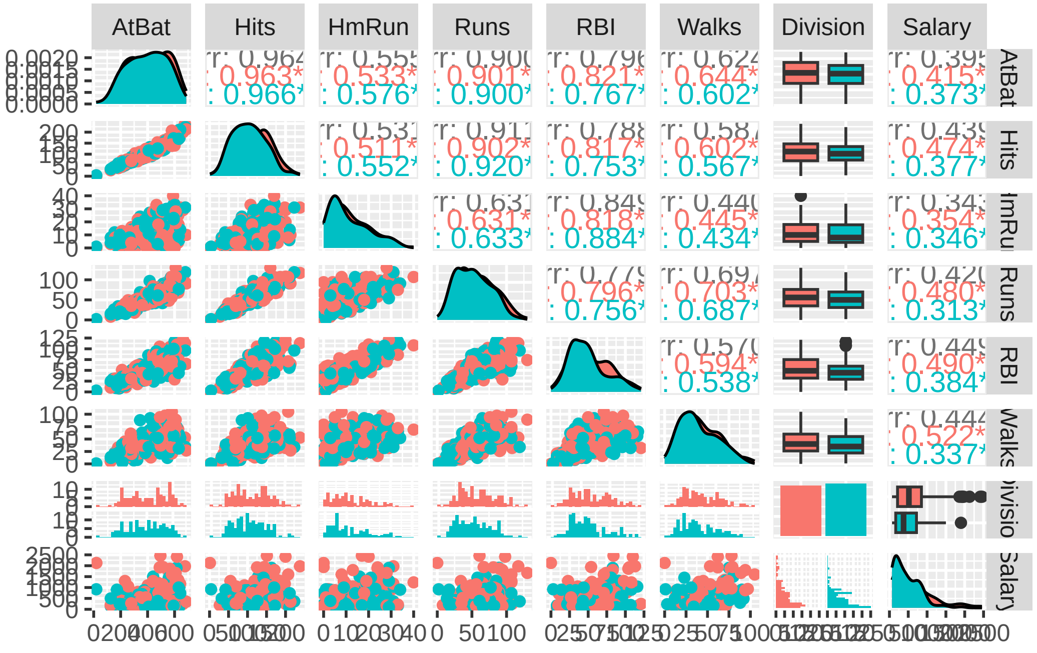
```
glimpse(hitters)
```

```
Rows: 263
Columns: 20
$ AtBat     <int> 315, 479, 496, 321, 594, 185, 298, 323, 401, 574, 202, 418, ~
$ Hits      <int> 81, 130, 141, 87, 169, 37, 73, 81, 92, 159, 53, 113, 60, 43,~
$ HmRun     <int> 7, 18, 20, 10, 4, 1, 0, 6, 17, 21, 4, 13, 0, 7, 20, 2, 8, 16~
$ Runs      <int> 24, 66, 65, 39, 74, 23, 24, 26, 49, 107, 31, 48, 30, 29, 89,~
$ RBI       <int> 38, 72, 78, 42, 51, 8, 24, 32, 66, 75, 26, 61, 11, 27, 75, 8~
$ Walks     <int> 39, 76, 37, 30, 35, 21, 7, 8, 65, 59, 27, 47, 22, 30, 73, 15~
$ Years     <int> 14, 3, 11, 2, 11, 2, 3, 2, 13, 10, 9, 4, 6, 13, 15, 5, 8, 1,~
$ CAtBat    <int> 3449, 1624, 5628, 396, 4408, 214, 509, 341, 5206, 4631, 1876~
$ CHits     <int> 835, 457, 1575, 101, 1133, 42, 108, 86, 1332, 1300, 467, 392~
$ CHmRun    <int> 69, 63, 225, 12, 19, 1, 0, 6, 253, 90, 15, 41, 4, 36, 177, 5~
$ CRuns     <int> 321, 224, 828, 48, 501, 30, 41, 32, 784, 702, 192, 205, 309,~
$ CRBI      <int> 414, 266, 838, 46, 336, 9, 37, 34, 890, 504, 186, 204, 103, ~
$ CWalks    <int> 375, 263, 354, 33, 194, 24, 12, 8, 866, 488, 161, 203, 207, ~
$ League    <fct> N, A, N, N, A, N, A, N, A, A, N, N, A, N, N, A, N, N, A, N, ~
$ Division  <fct> W, W, E, E, W, E, W, W, E, E, W, E, E, E, W, W, W, E, W, W, ~
$ PutOuts   <int> 632, 880, 200, 805, 282, 76, 121, 143, 0, 238, 304, 211, 121~
$ Assists   <int> 43, 82, 11, 40, 421, 127, 283, 290, 0, 445, 45, 11, 151, 45,~
$ Errors    <int> 10, 14, 3, 4, 25, 7, 9, 19, 0, 22, 11, 7, 6, 8, 10, 16, 2, 5~
$ Salary    <dbl> 475.000, 480.000, 500.000, 91.500, 750.000, 70.000, 100.000,~
$ NewLeague <fct> N, A, N, N, A, A, A, N, A, A, N, N, A, N, N, A, N, N, N, N, ~
```

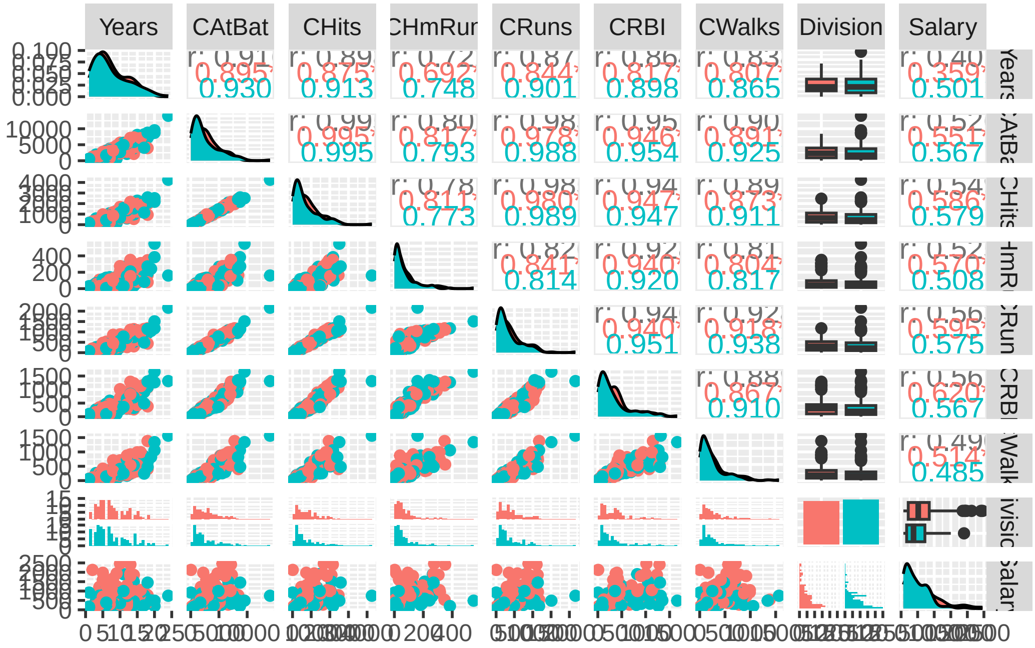
```
library(corrplot)
hitters %>%
  dplyr::select(-c(League,Division,NewLeague)) %>%
  cor() %>% corrplot()
```



```
hitters %>%
  select(AtBat:Walks,Division,Salary) %>%
  ggpairs(mapping=aes(col=Division))
```



```r
hitters %>%
  select(Years:CWalks,Division,Salary) %>%
  ggpairs(mapping=aes(col=Division))
```



```r
library(car)
vif(lm(Salary~.,data=hitters))
```

```
     AtBat       Hits      HmRun       Runs        RBI      Walks      Years
 22.944366  30.281255   7.758668  15.246418  11.921715   4.148712   9.313280
    CAtBat      CHits     CHmRun      CRuns       CRBI     CWalks     League
251.561160 502.954289  46.488462 162.520810 131.965858  19.744105   4.134115
  Division    PutOuts    Assists     Errors  NewLeague
  1.075398   1.236317   2.709341   2.214543   4.099063
```

Ridge regression

```r
ridge_spec <- linear_reg(mixture = 0, penalty = 0) %>%
  set_mode("regression") %>%
  set_engine("glmnet")
```

```
ridge_fit <- fit(ridge_spec, Salary ~ ., data = hitters)
tidy(ridge_fit)
```

```
Attaching package: 'Matrix'

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

    expand, pack, unpack

Loaded glmnet 4.1-8

# A tibble: 20 x 3
   term         estimate penalty
   <chr>           <dbl>   <dbl>
 1 (Intercept)   81.1          0
 2 AtBat         -0.682        0
 3 Hits           2.77         0
 4 HmRun         -1.37         0
 5 Runs           1.01         0
 6 RBI            0.713        0
 7 Walks          3.38         0
 8 Years         -9.07         0
 9 CAtBat        -0.00120      0
10 CHits          0.136        0
11 CHmRun         0.698        0
12 CRuns          0.296        0
13 CRBI           0.257        0
14 CWalks        -0.279        0
15 LeagueN       53.2          0
16 DivisionW   -123.           0
17 PutOuts        0.264        0
18 Assists        0.170        0
19 Errors        -3.69         0
20 NewLeagueN   -18.1          0
```

```
tidy(ridge_fit, penalty = 11498)
```

```
# A tibble: 20 x 3
   term        estimate penalty
   <chr>            <dbl>   <dbl>
 1 (Intercept) 407.         11498
 2 AtBat         0.0370     11498
 3 Hits          0.138      11498
 4 HmRun         0.525      11498
 5 Runs          0.231      11498
 6 RBI           0.240      11498
 7 Walks         0.290      11498
 8 Years         1.11       11498
 9 CAtBat        0.00314    11498
10 CHits         0.0117     11498
11 CHmRun        0.0876     11498
12 CRuns         0.0234     11498
13 CRBI          0.0242     11498
14 CWalks        0.0250     11498
15 LeagueN       0.0866     11498
16 DivisionW    -6.23       11498
17 PutOuts       0.0165     11498
18 Assists       0.00262    11498
19 Errors       -0.0206     11498
20 NewLeagueN    0.303      11498
```

```r
tidy(ridge_fit, penalty = 705)
```

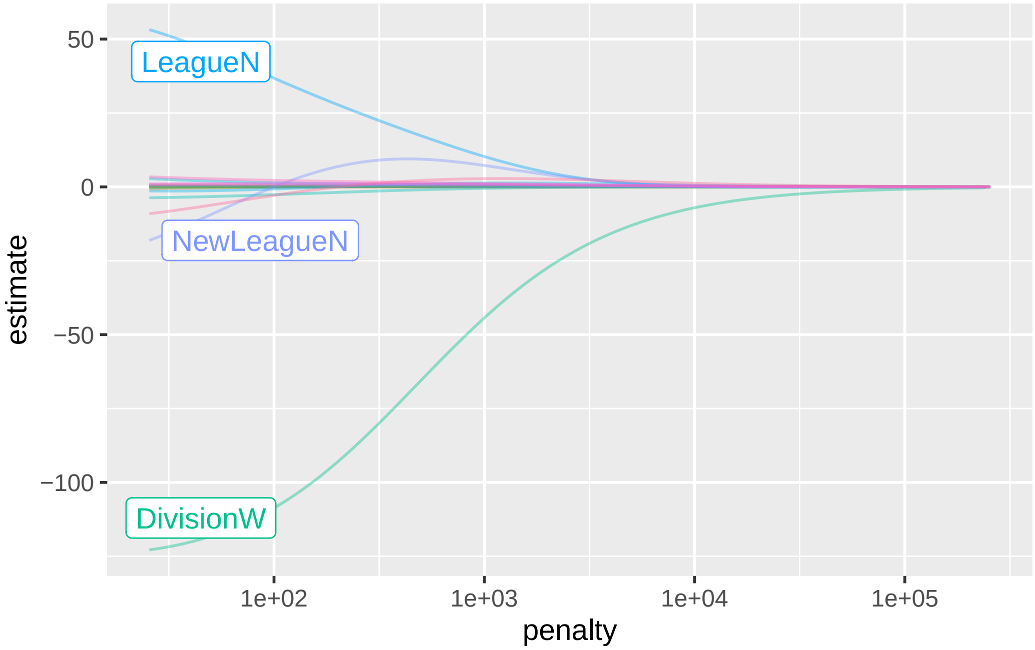
```
# A tibble: 20 x 3
   term        estimate penalty
   <chr>          <dbl>   <dbl>
 1 (Intercept)  54.4        705
 2 AtBat         0.112      705
 3 Hits          0.656      705
 4 HmRun         1.18       705
 5 Runs          0.937      705
 6 RBI           0.847      705
 7 Walks         1.32       705
 8 Years         2.58       705
 9 CAtBat        0.0108     705
10 CHits         0.0468     705
11 CHmRun        0.338      705
12 CRuns         0.0937     705
13 CRBI          0.0979     705
14 CWalks        0.0718     705
15 LeagueN      13.7        705
```

```
16 DivisionW   -54.7        705
17 PutOuts       0.119      705
18 Assists       0.0161     705
19 Errors       -0.704      705
20 NewLeagueN    8.61       705
```

```
tidy(ridge_fit, penalty = 50)
```

```
# A tibble: 20 x 3
   term          estimate penalty
   <chr>            <dbl>   <dbl>
 1 (Intercept)   48.2          50
 2 AtBat         -0.354        50
 3 Hits           1.95         50
 4 HmRun         -1.29         50
 5 Runs           1.16         50
 6 RBI            0.809        50
 7 Walks          2.71         50
 8 Years         -6.20         50
 9 CAtBat         0.00609      50
10 CHits          0.107        50
11 CHmRun         0.629        50
12 CRuns          0.217        50
13 CRBI           0.215        50
14 CWalks        -0.149        50
15 LeagueN       45.9          50
16 DivisionW   -118.           50
17 PutOuts        0.250        50
18 Assists        0.121        50
19 Errors        -3.28         50
20 NewLeagueN    -9.42         50
```

```
  ridge_fit %>%
  autoplot()
```



```
predict(ridge_fit, new_data = hitters)
```

```
# A tibble: 263 x 1
    .pred
    <dbl>
 1  442.
 2  676.
 3 1059.
 4  521.
 5  543.
 6  218.
 7   74.7
 8   96.1
 9  809.
10  865.
# i 253 more rows
```

```
predict(ridge_fit, new_data = hitters, penalty = 500)
```

```
# A tibble: 263 x 1
   .pred
```

```
    <dbl>
 1  525.
 2  620.
 3  895.
 4  425.
 5  589.
 6  179.
 7  147.
 8  187.
 9  841.
10  840.
# i 253 more rows
```

```
ridge_recipe <-
  recipe(formula = Salary ~ ., data = hitters_train) %>%
  step_novel(all_nominal_predictors()) %>%  # add 'new' level to factors and chr -> fctr
  step_dummy(all_nominal_predictors()) %>%  # factors to dummy variables
  step_zv(all_predictors()) %>%             # remove zero variance predictors
  step_normalize(all_predictors())          # normalize all predictors

  ridge_spec <-
  linear_reg(penalty = tune(), mixture = 0) %>%
  set_mode("regression") %>%
  set_engine("glmnet")

  ridge_workflow <- workflow() %>%
  add_recipe(ridge_recipe) %>%
  add_model(ridge_spec)

  penalty_grid <- grid_regular(penalty(range = c(-5, 5)), levels = 50)
penalty_grid
```

```
# A tibble: 50 x 1
     penalty
       <dbl>
 1 0.00001
 2 0.0000160
 3 0.0000256
 4 0.0000409
 5 0.0000655
 6 0.000105
 7 0.000168
 8 0.000268
 9 0.000429
10 0.000687
# i 40 more rows
```

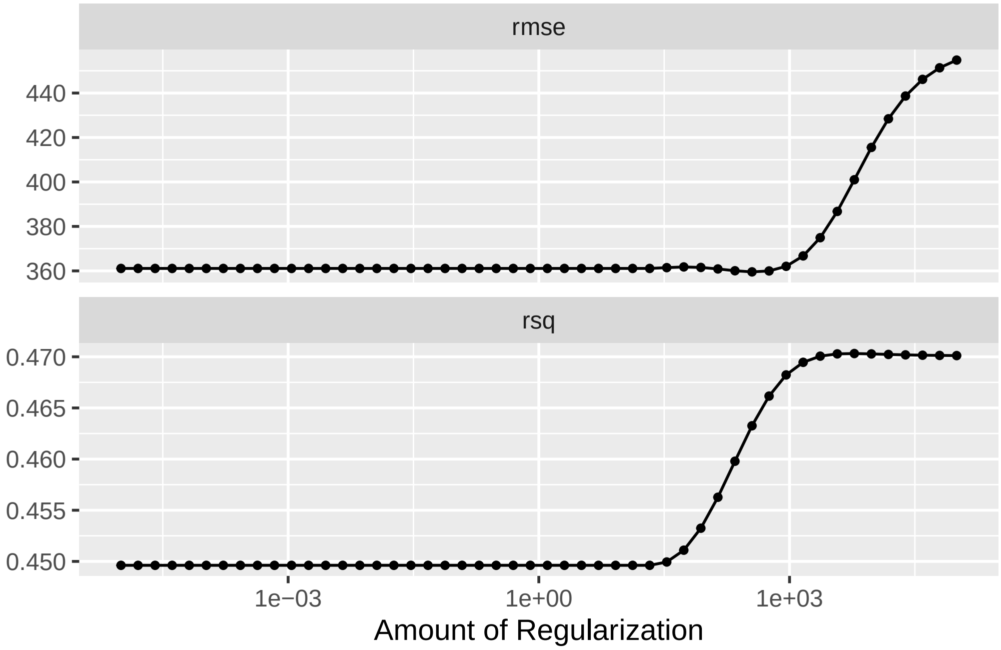
```
tune_res <- tune_grid(
  ridge_workflow,
  resamples = hitters_fold,
  grid = penalty_grid,
  control(parallel_over = "everything")
)
```

```
Warning: The `...` are not used in this function but one or more objects were
passed: ''
```

```
tune_res
```

```
# Tuning results
# 10-fold cross-validation
# A tibble: 10 x 4
   splits           id     .metrics           .notes
   <list>           <chr>  <list>             <list>
 1 <split [176/20]> Fold01 <tibble [100 x 5]> <tibble [0 x 3]>
 2 <split [176/20]> Fold02 <tibble [100 x 5]> <tibble [0 x 3]>
 3 <split [176/20]> Fold03 <tibble [100 x 5]> <tibble [0 x 3]>
 4 <split [176/20]> Fold04 <tibble [100 x 5]> <tibble [0 x 3]>
 5 <split [176/20]> Fold05 <tibble [100 x 5]> <tibble [0 x 3]>
 6 <split [176/20]> Fold06 <tibble [100 x 5]> <tibble [0 x 3]>
 7 <split [177/19]> Fold07 <tibble [100 x 5]> <tibble [0 x 3]>
 8 <split [177/19]> Fold08 <tibble [100 x 5]> <tibble [0 x 3]>
 9 <split [177/19]> Fold09 <tibble [100 x 5]> <tibble [0 x 3]>
10 <split [177/19]> Fold10 <tibble [100 x 5]> <tibble [0 x 3]>
```

```
autoplot(tune_res)
```



```
collect_metrics(tune_res) %>% filter(.metric=="rmse")
```

```
# A tibble: 50 x 7
     penalty .metric .estimator  mean     n std_err .config
       <dbl> <chr>   <chr>      <dbl> <int>   <dbl> <chr>
 1 0.00001   rmse    standard    361.    10    28.3 Preprocessor1_Model01
 2 0.0000160 rmse    standard    361.    10    28.3 Preprocessor1_Model02
 3 0.0000256 rmse    standard    361.    10    28.3 Preprocessor1_Model03
 4 0.0000409 rmse    standard    361.    10    28.3 Preprocessor1_Model04
 5 0.0000655 rmse    standard    361.    10    28.3 Preprocessor1_Model05
 6 0.000105  rmse    standard    361.    10    28.3 Preprocessor1_Model06
 7 0.000168  rmse    standard    361.    10    28.3 Preprocessor1_Model07
 8 0.000268  rmse    standard    361.    10    28.3 Preprocessor1_Model08
 9 0.000429  rmse    standard    361.    10    28.3 Preprocessor1_Model09
10 0.000687  rmse    standard    361.    10    28.3 Preprocessor1_Model10
# i 40 more rows
```

```
best_penalty <- select_best(tune_res, metric = "rmse")
best_penalty
```

```
# A tibble: 1 x 2
```

```
  penalty .config
    <dbl> <chr>
1    356. Preprocessor1_Model38
```

```
ridge_final <- finalize_workflow(ridge_workflow, best_penalty)
ridge_final_fit <- fit(ridge_final, data = hitters_train)
ridge_aug <- augment(ridge_final_fit, new_data = hitters_test)
ridge_result <- rbind(
  rmse(ridge_aug,truth = Salary, estimate = .pred),
  rsq(ridge_aug, truth = Salary, estimate = .pred))
ridge_result
```

```
# A tibble: 2 x 3
  .metric .estimator .estimate
  <chr>   <chr>          <dbl>
1 rmse    standard     277.
2 rsq     standard       0.515
```

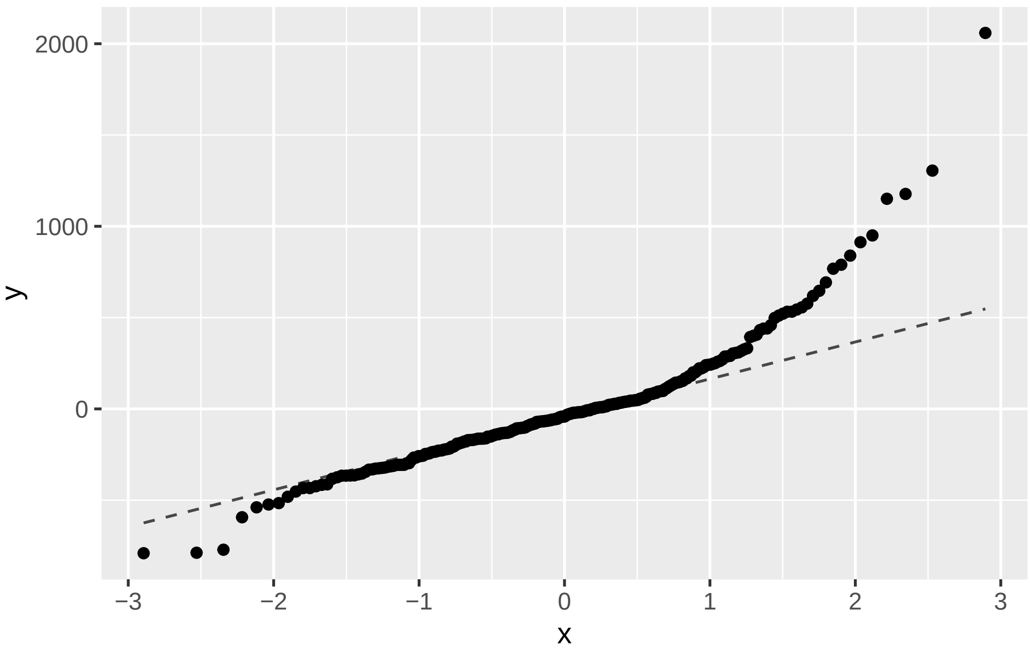
Final ridge model

```
ridge_final_fit <- fit(ridge_final, data=hitters)
tidy(ridge_final_fit,conf_int=TRUE)
```

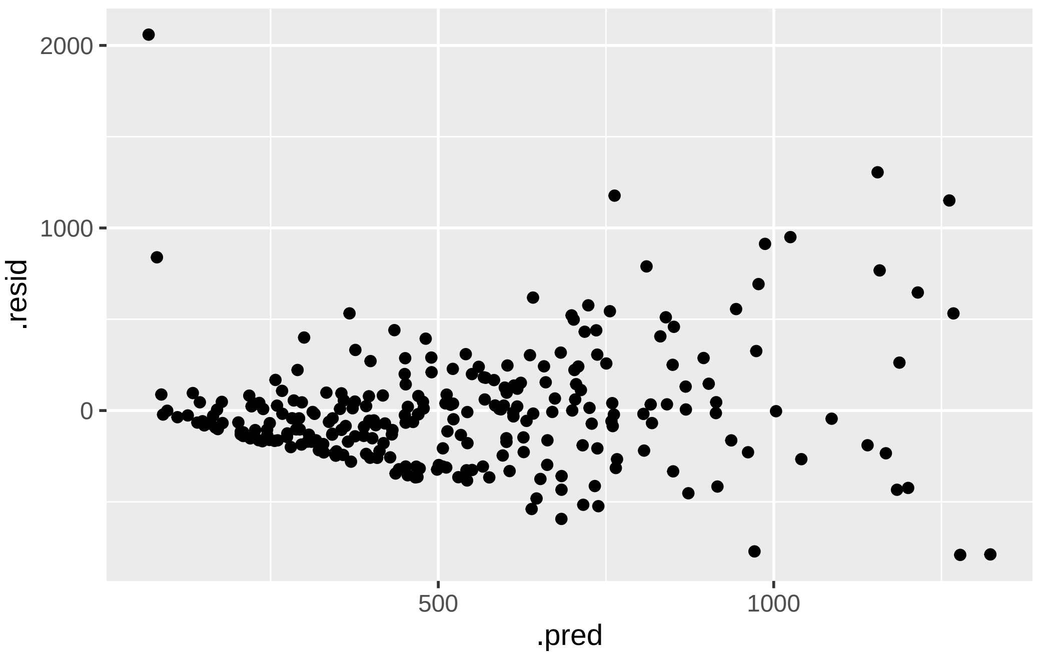
```
# A tibble: 20 x 3
   term        estimate penalty
   <chr>          <dbl>   <dbl>
 1 (Intercept)   536.      356.
 2 AtBat          12.3     356.
 3 Hits           37.6     356.
 4 HmRun           6.00    356.
 5 Runs           26.9     356.
 6 RBI            22.8     356.
 7 Walks          34.5     356.
 8 Years           7.44    356.
 9 CAtBat         25.9     356.
10 CHits          36.4     356.
11 CHmRun         32.7     356.
12 CRuns          37.1     356.
13 CRBI           38.2     356.
14 CWalks         14.9     356.
15 PutOuts        45.0     356.
```

```
16 Assists          3.96     356.
17 Errors          -8.43     356.
18 League_N        10.6      356.
19 Division_W     -38.2      356.
20 NewLeague_N      4.65     356.
```

```
ridge_aug <- augment(ridge_final_fit, new_data=hitters) %>% mutate(.resid=Salary-.pred)
ridge_aug %>%
  gf_qq(~.resid) %>% gf_qqline()
```



```
ridge_aug %>%
  gf_point(.resid~.pred)
```

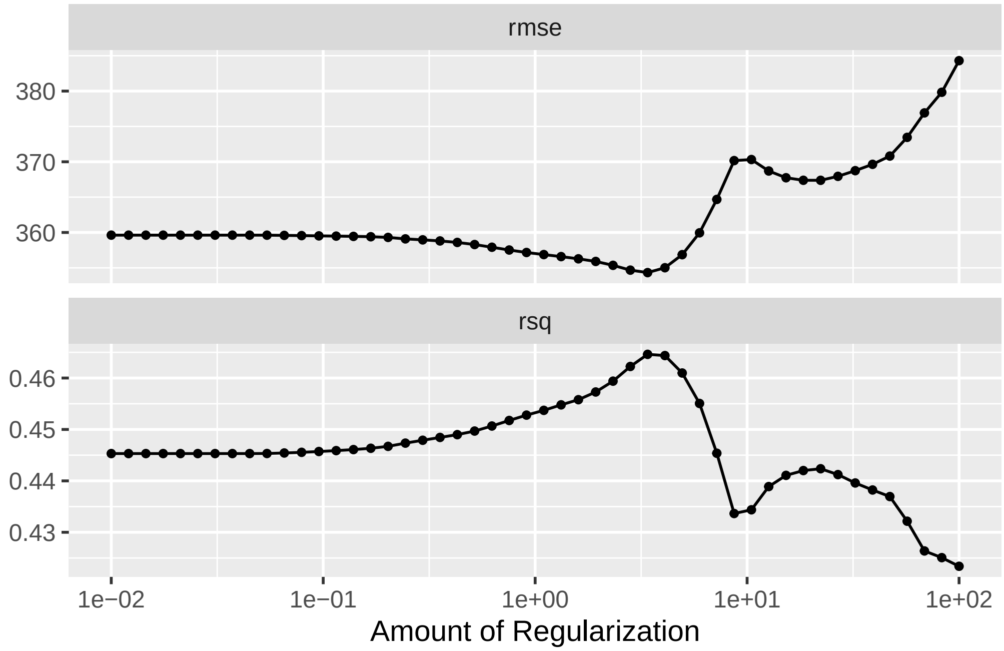


The Lasso

```
lasso_recipe <-
  recipe(formula = Salary ~ ., data = hitters_train) %>%
  step_novel(all_nominal_predictors()) %>%
  step_dummy(all_nominal_predictors()) %>%
  step_zv(all_predictors()) %>%
  step_normalize(all_predictors())

  lasso_spec <-
  linear_reg(penalty = tune(), mixture = 1) %>%
  set_mode("regression") %>%
  set_engine("glmnet")

lasso_workflow <- workflow() %>%
  add_recipe(lasso_recipe) %>%
  add_model(lasso_spec)

  penalty_grid <- grid_regular(penalty(range = c(-2, 2)), levels = 50)
  tune_res <- tune_grid(
  lasso_workflow,
  resamples = hitters_fold,
  grid = penalty_grid,
  control(parallel_over = "everything")
)
```

```
Warning: The `...` are not used in this function but one or more objects were
passed: ''
```

```
autoplot(tune_res)
```



```
best_penalty <- select_best(tune_res, metric = "rsq")
lasso_final <- finalize_workflow(lasso_workflow, best_penalty)
lasso_final_fit <- fit(lasso_final, data = hitters_train)
lasso_aug <- augment(lasso_final_fit, new_data = hitters_test)
lasso_result <- rbind(
  rmse(lasso_aug,truth = Salary, estimate = .pred),
  rsq(lasso_aug, truth = Salary, estimate = .pred))
lasso_result
```

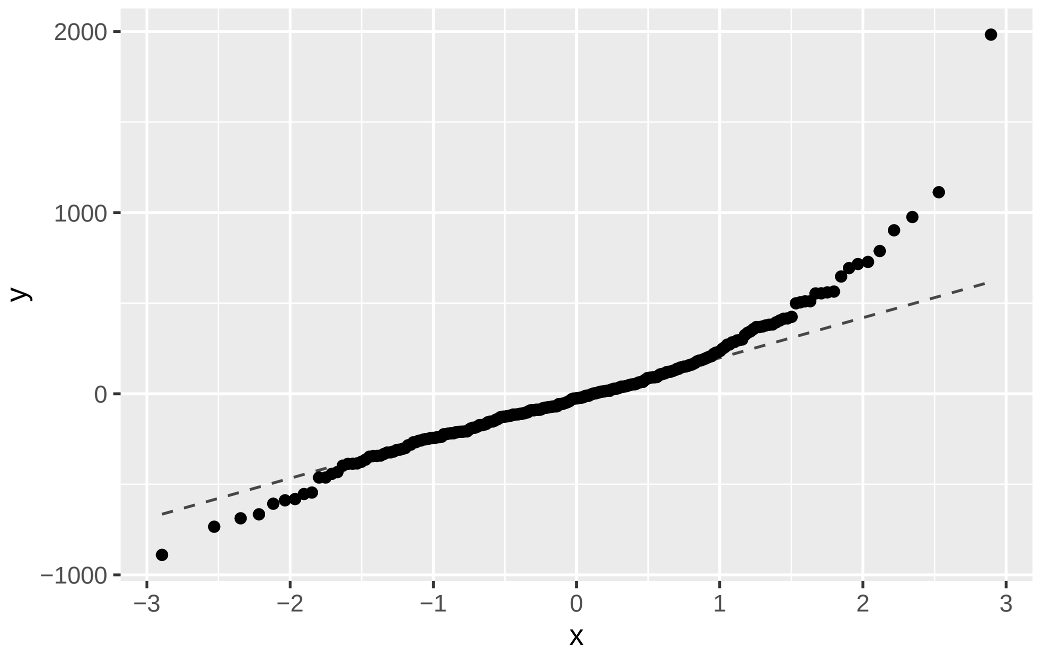
```
# A tibble: 2 x 3
  .metric .estimator .estimate
  <chr>   <chr>          <dbl>
1 rmse    standard     277.   
2 rsq     standard       0.493
```

Final lasso model

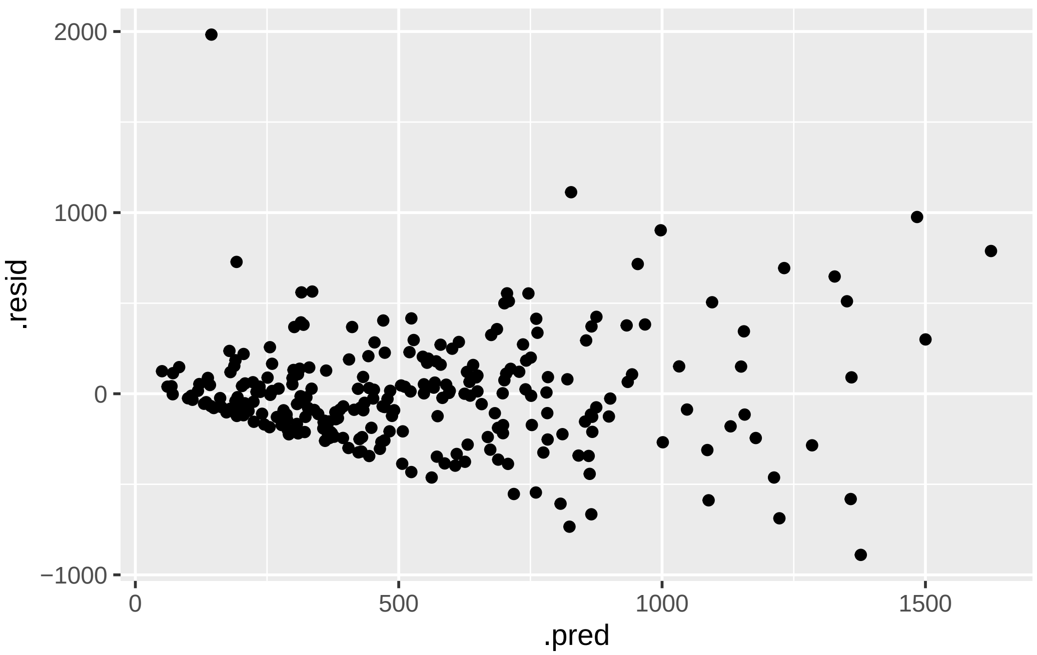
```
lasso_final_fit <- fit(lasso_final, data=hitters)
tidy(lasso_final_fit,conf_int=TRUE)
```

```
# A tibble: 20 x 3
   term        estimate penalty
   <chr>          <dbl>   <dbl>
 1 (Intercept)    536.     3.39
 2 AtBat         -198.     3.39
 3 Hits           235.     3.39
 4 HmRun            0      3.39
 5 Runs             0      3.39
 6 RBI              0      3.39
 7 Walks           94.9    3.39
 8 Years          -40.7    3.39
 9 CAtBat           0      3.39
10 CHits            0      3.39
11 CHmRun          35.2    3.39
12 CRuns          197.     3.39
13 CRBI           130.     3.39
14 CWalks        -118.     3.39
15 PutOuts         74.7    3.39
16 Assists         19.6    3.39
17 Errors         -11.5    3.39
18 League_N        15.4    3.39
19 Division_W     -59.7    3.39
20 NewLeague_N      0      3.39
```

```
lasso_aug <- augment(lasso_final_fit, new_data=hitters) %>% mutate(.resid=Salary-.pred)
lasso_aug %>%
  gf_qq(~.resid) %>% gf_qqline()
```



```
lasso_aug %>%
  gf_point(.resid~.pred)
```



Principal components regression

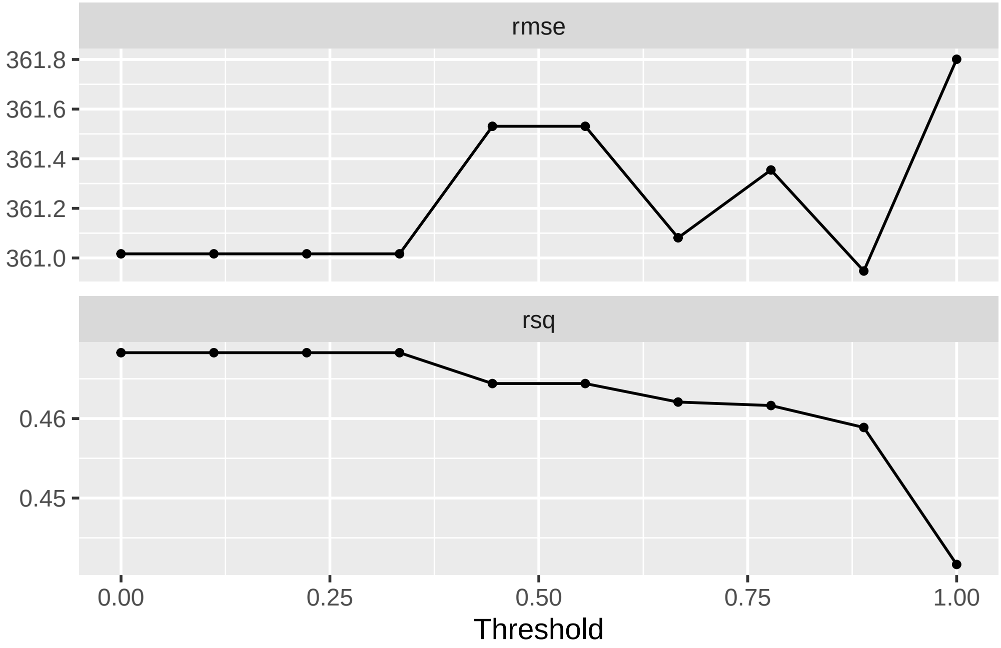
```
lm_spec <-
  linear_reg() %>%
  set_mode("regression") %>%
  set_engine("lm")

pca_recipe <-
  recipe(formula = Salary ~ ., data = hitters_train) %>%
  step_novel(all_nominal_predictors()) %>%
  step_dummy(all_nominal_predictors()) %>%
  step_zv(all_predictors()) %>%
  step_normalize(all_predictors()) %>%
  step_pca(all_predictors(), threshold = tune())

pca_workflow <-
  workflow() %>%
  add_recipe(pca_recipe) %>%
  add_model(lm_spec)

  threshold_grid <- grid_regular(threshold(), levels = 10)
threshold_grid
```

```
# A tibble: 10 x 1
   threshold
       <dbl>
 1     0
 2     0.111
 3     0.222
 4     0.333
 5     0.444
 6     0.556
 7     0.667
 8     0.778
 9     0.889
10     1
```

```
tune_res <- tune_grid(
  pca_workflow,
  resamples = hitters_fold,
  grid = threshold_grid,
  control(parallel_over = "everything")
)
```

```
Warning: The `...` are not used in this function but one or more objects were
passed: ''
```

```
autoplot(tune_res)
```



```
best_threshold <- select_best(tune_res, metric = "rmse")
pca_final <- finalize_workflow(pca_workflow, best_threshold)
pca_final_fit <- fit(pca_final, data = hitters_train)
pca_aug <- augment(pca_final_fit, new_data = hitters_test)
pca_result <- rbind(
  rmse(pca_aug,truth = Salary, estimate = .pred),
  rsq(pca_aug, truth = Salary, estimate = .pred))
pca_result
```

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

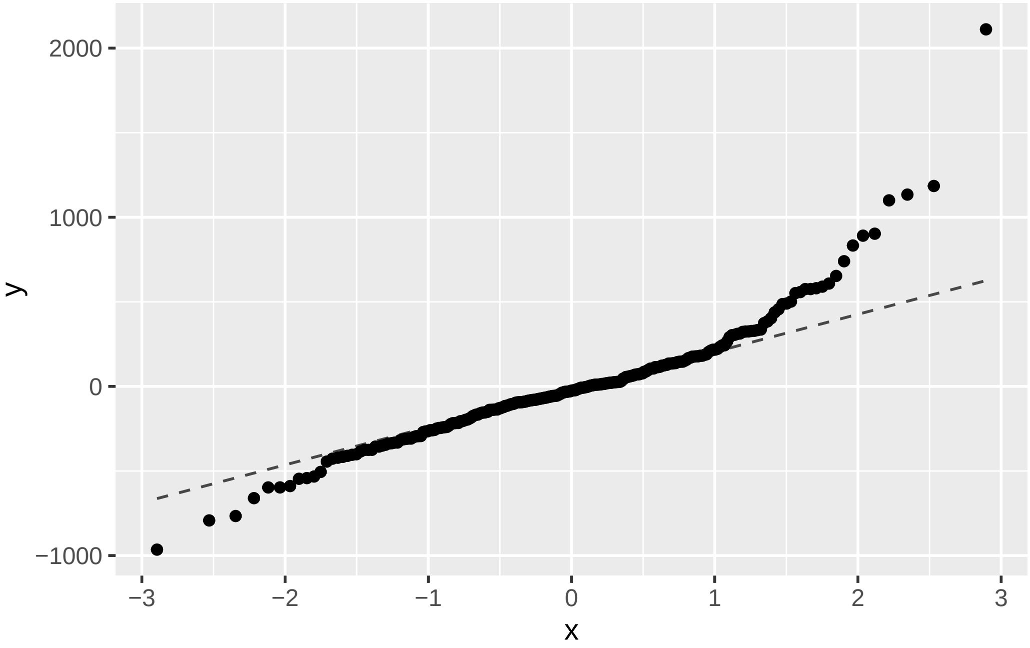
```
1 rmse    standard     271.
2 rsq     standard       0.521
```

Final pca model

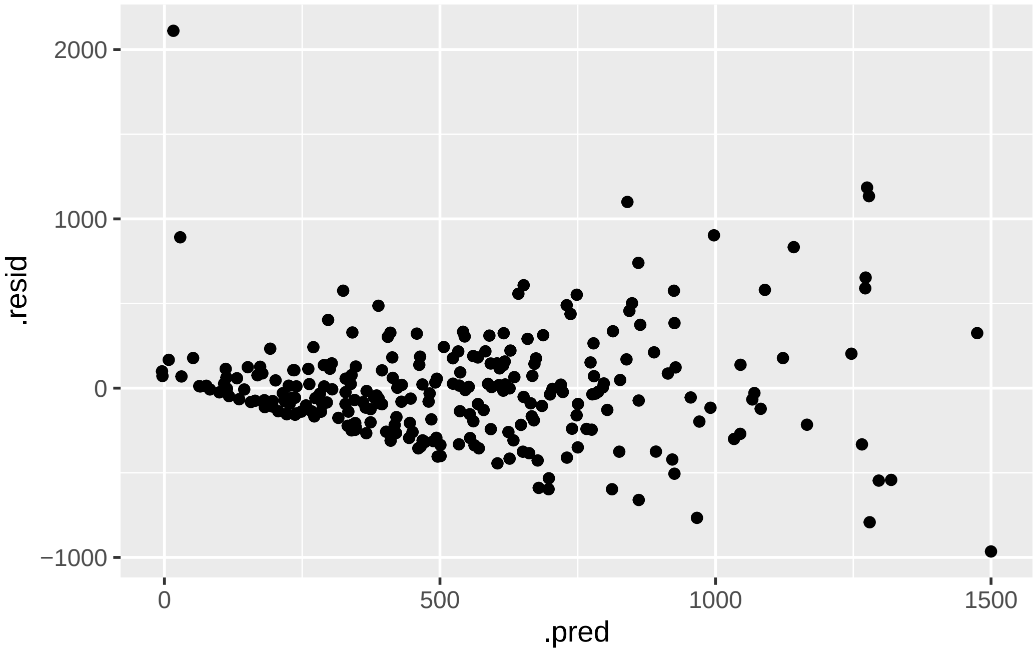
```
pca_final_fit <- fit(pca_final, data=hitters)
tidy(pca_final_fit,conf_int=TRUE)
```

```
# A tibble: 8 x 5
  term        estimate std.error statistic  p.value
  <chr>          <dbl>     <dbl>     <dbl>    <dbl>
1 (Intercept)    536.      20.6     26.0   9.20e-74
2 PC1            107.       7.65    13.9   3.24e-33
3 PC2             21.6     10.1      2.14  3.35e- 2
4 PC3            -24.3     14.5     -1.68  9.39e- 2
5 PC4            -37.1     16.5     -2.24  2.59e- 2
6 PC5             58.5     20.6      2.84  4.94e- 3
7 PC6             62.3     22.7      2.75  6.48e- 3
8 PC7            -24.7     24.8     -0.994 3.21e- 1
```

```
pca_aug <- augment(pca_final_fit, new_data=hitters) %>% mutate(.resid=Salary-.pred)
pca_aug %>%
  gf_qq(~.resid) %>% gf_qqline()
```



```
pca_aug %>%
  gf_point(.resid~.pred)
```



Partial least squares

Partial least squares in Tidy Models requires the package ‘mixOmics’ which is only available with Bioconductor which is a huge installation.

```
if (!require("BiocManager", quietly = TRUE))
    install.packages("BiocManager")

BiocManager::install("mixOmics")
```

```
pls_recipe <-
  recipe(formula = Salary ~ ., data = hitters_train) %>%
  step_novel(all_nominal_predictors()) %>%
  step_dummy(all_nominal_predictors()) %>%
  step_zv(all_predictors()) %>%
  step_normalize(all_predictors()) %>%
  step_pls(all_predictors(), num_comp = tune(), outcome = "Salary")
```

```r
lm_spec <- linear_reg() %>%
  set_mode("regression") %>%
  set_engine("lm")

pls_workflow <- workflow() %>%
  add_recipe(pls_recipe) %>%
  add_model(lm_spec)

num_comp_grid <- grid_regular(num_comp(c(1, 20)), levels = 10)

tune_res <- tune_grid(
  pls_workflow,
  resamples = hitters_fold,
  grid = num_comp_grid,
  control(parallel_over = "everything")
)
```

```
Warning: The `...` are not used in this function but one or more objects were
passed: ''
```

```r
best_threshold <- select_best(tune_res, metric = "rmse")

pls_final <- finalize_workflow(pls_workflow, best_threshold)

pls_final_fit <- fit(pls_final, data = hitters_train)

pls_aug <- augment(pls_final_fit, new_data = hitters_test)
pls_result <- rbind(
  rmse(pls_aug,truth = Salary, estimate = .pred),
  rsq(pls_aug, truth = Salary, estimate = .pred))
pls_result
```

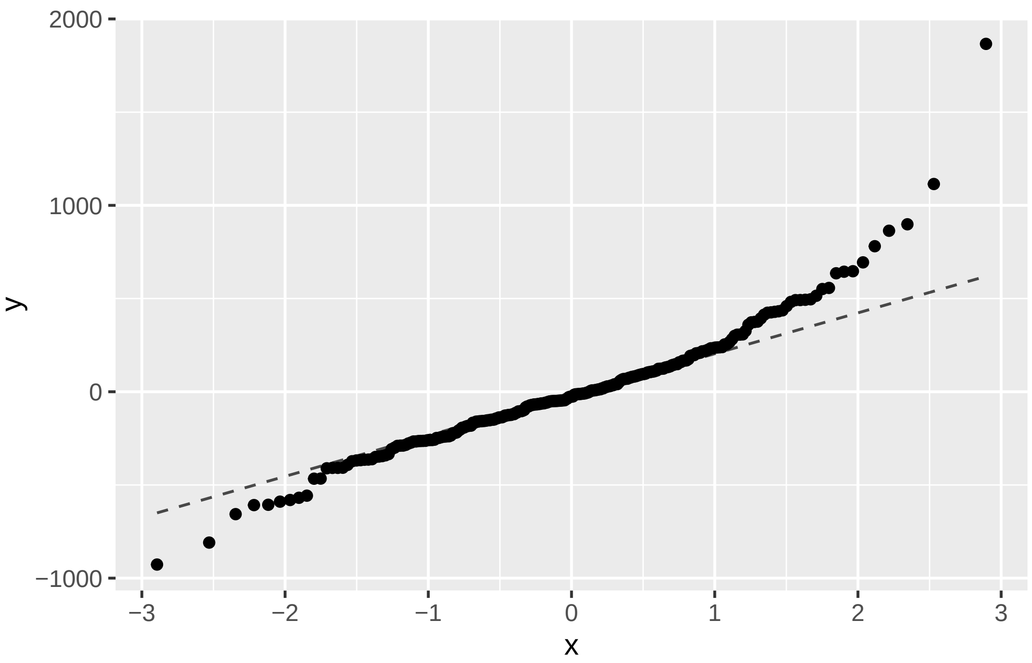
```
# A tibble: 2 x 3
  .metric .estimator .estimate
  <chr>   <chr>          <dbl>
1 rmse    standard     291.
2 rsq     standard       0.447
```

Final pls model

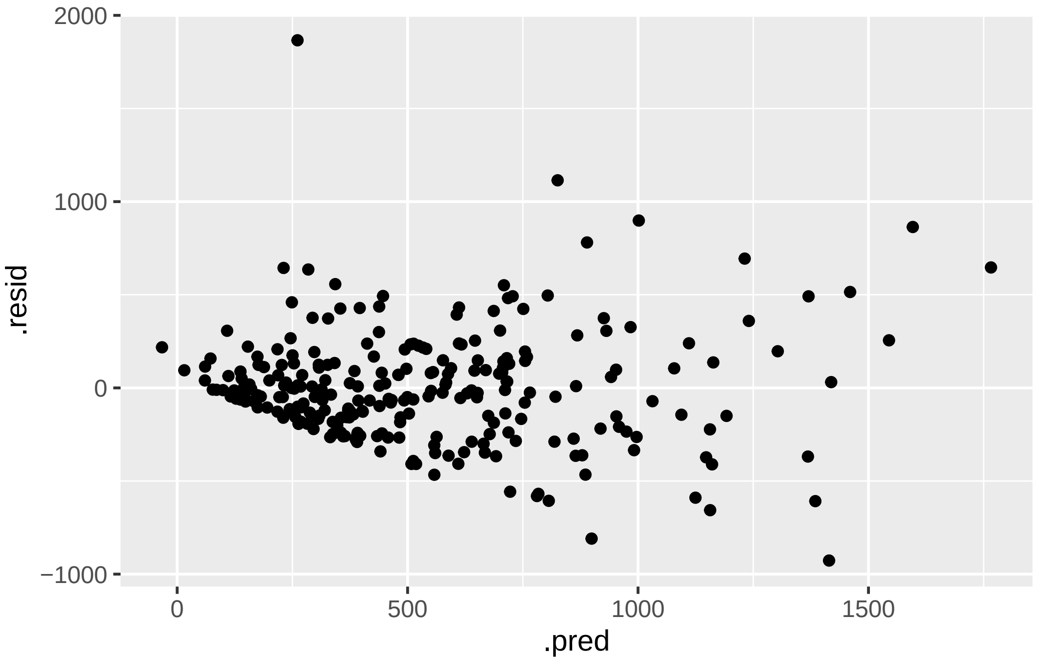
```r
pls_final_fit <- fit(pls_final, data=hitters)
tidy(pls_final_fit,conf_int=TRUE)
```

```
# A tibble: 12 x 5
   term        estimate std.error statistic  p.value
   <chr>          <dbl>     <dbl>     <dbl>    <dbl>
 1 (Intercept)    536.      19.3      27.8  1.24e-78
 2 PLS01          111.       7.24     15.3  6.78e-38
 3 PLS02           65.6     15.3       4.28 2.69e- 5
 4 PLS03           34.7     12.9       2.68 7.76e- 3
 5 PLS04           49.5     21.3       2.32 2.10e- 2
 6 PLS05           99.6     31.6       3.15 1.81e- 3
 7 PLS06           52.5     21.1       2.49 1.36e- 2
 8 PLS07           59.5     30.9       1.92 5.57e- 2
 9 PLS08           84.7     37.8       2.24 2.61e- 2
10 PLS09           37.6     31.2       1.21 2.29e- 1
11 PLS10           57.2     48.8       1.17 2.42e- 1
12 PLS11           48.6     40.5       1.20 2.31e- 1
```

```r
pls_aug <- augment(pls_final_fit, new_data=hitters) %>% mutate(.resid=Salary-.pred)
pls_aug %>%
  gf_qq(~.resid) %>% gf_qqline()
```



```
pls_aug %>%
  gf_point(.resid~.pred)
```



Blended Ridge and Lasso

```
mixed_recipe <-
  recipe(formula = Salary ~ ., data = hitters_train) %>%
  step_novel(all_nominal_predictors()) %>%
  step_dummy(all_nominal_predictors()) %>%
  step_zv(all_predictors()) %>%
  step_normalize(all_predictors())
prep(mixed_recipe) #%>% bake(hitters_train)
```

```
-- Recipe ----------------------------------------------------------------------
-- Inputs

Number of variables by role

outcome:    1
predictor: 19



-- Training information

Training data contained 196 data points and no incomplete rows.



-- Operations

* Novel factor level assignment for: League, Division, NewLeague | Trained

* Dummy variables from: League, Division, NewLeague | Trained

* Zero variance filter removed: League_new and Division_new, ... | Trained

* Centering and scaling for: AtBat, Hits, HmRun, Runs, RBI, ... | Trained
```

```r
mixed_spec <-
  linear_reg(penalty = tune("penalty"), mixture = tune("mixture")) %>%
  set_mode("regression") %>%
  set_engine("glmnet")

  extract_parameter_set_dials(mixed_spec)
```

```
Collection of 2 parameters for tuning

 identifier    type    object
    penalty penalty nparam[+]
    mixture mixture nparam[+]
```

```r
mixed_workflow <- workflow() %>%
  add_recipe(mixed_recipe) %>%
  add_model(mixed_spec)

mixed_grid <- grid_regular(penalty(range = c(-5, 5)), mixture(),levels = 50)
mixed_grid %>%
  pull(penalty) %>% range()
```

```
[1] 1e-05 1e+05
```

```r
grid_regular(penalty(), mixture(),levels = 50) %>%
  pull(penalty) %>% range()
```

```
[1] 1e-10 1e+00
```

```r
expand.grid("penalty"=seq(1e-5,1e5,2000), "mixture"=seq(0,1,1/50)) %>%
  pull(penalty) %>% range()
```

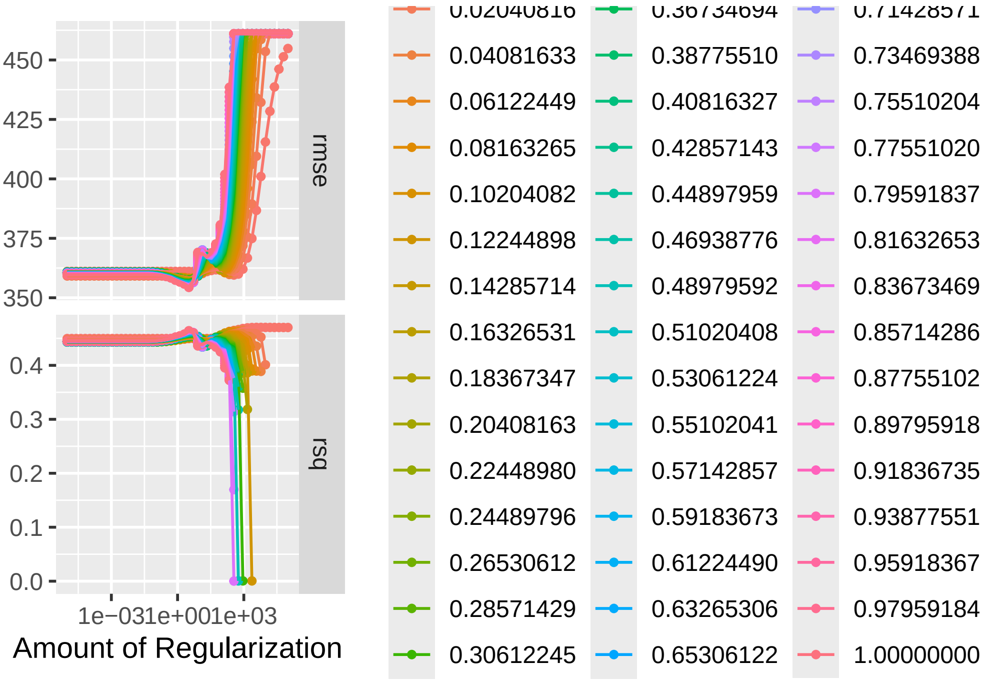
```
[1] 1.0e-05 9.8e+04
```

```r
system.time(
  suppressMessages(
    tune_res <- tune_grid(
      mixed_workflow,
      resamples = hitters_fold,
      grid = mixed_grid,
      control(parallel_over = "everything")
    )
  )
)
```

```
Warning: The `...` are not used in this function but one or more objects were
passed: ''
```

```
   user  system elapsed 
  1.769   0.042  22.779 
```

```
autoplot(tune_res)
```



```
best_penalty <- select_best(tune_res, metric = "rmse")
mixed_final <- finalize_workflow(mixed_workflow, best_penalty)
mixed_final_fit <- fit(mixed_final, data = hitters_train)
mixed_aug <- augment(mixed_final_fit, new_data = hitters_test)
mixed_result <- rbind(
  rmse(mixed_aug,truth = Salary, estimate = .pred),
  rsq(mixed_aug, truth = Salary, estimate = .pred))
mixed_result
```

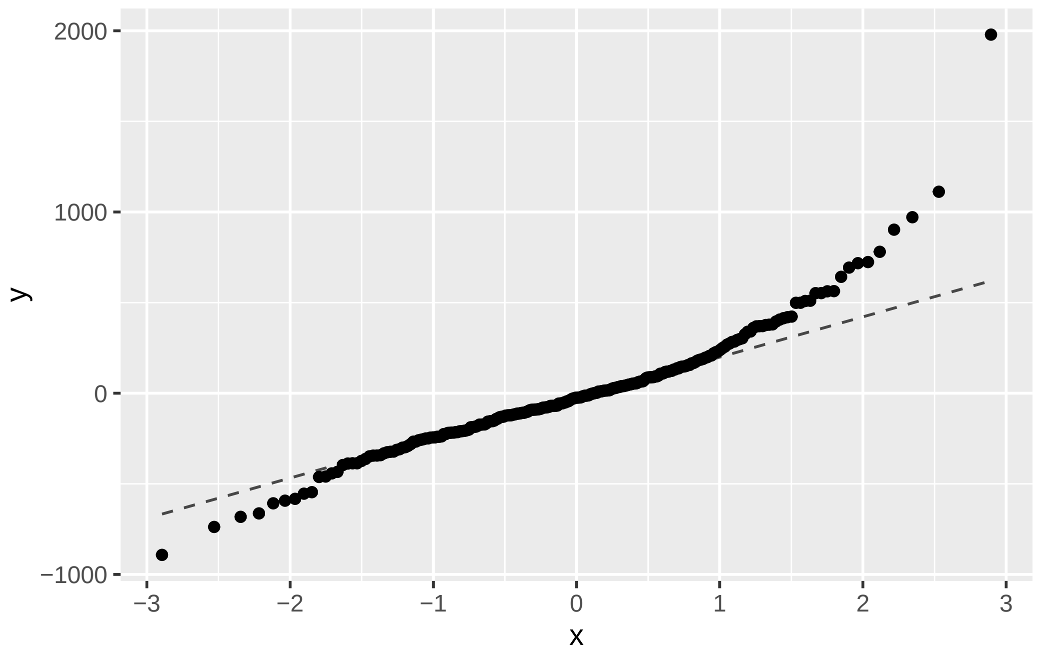
```
# A tibble: 2 x 3
  .metric .estimator .estimate
  <chr>   <chr>          <dbl>
1 rmse    standard     277.
2 rsq     standard       0.491
```

Final mixed model

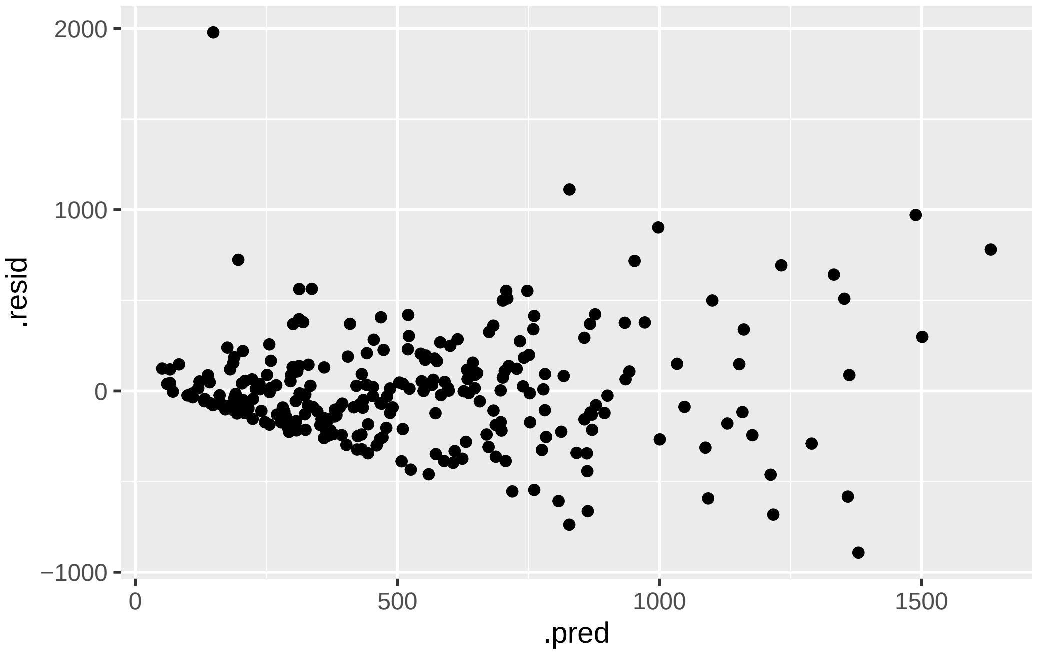
```
mixed_final_fit <- fit(mixed_final, data=hitters)
tidy(mixed_final_fit,conf_int=TRUE)
```

```
# A tibble: 20 x 3
   term        estimate penalty
   <chr>          <dbl>   <dbl>
 1 (Intercept)    536.     3.24
 2 AtBat         -205.     3.24
 3 Hits           240.     3.24
 4 HmRun            0      3.24
 5 Runs             0      3.24
 6 RBI              0      3.24
 7 Walks           96.6    3.24
 8 Years          -41.8    3.24
 9 CAtBat           0      3.24
10 CHits            0      3.24
11 CHmRun          36.6    3.24
12 CRuns          201.     3.24
13 CRBI           129.     3.24
14 CWalks        -123.     3.24
15 PutOuts         75.1    3.24
16 Assists         20.8    3.24
17 Errors         -11.9    3.24
18 League_N        15.6    3.24
19 Division_W     -59.7    3.24
20 NewLeague_N      0      3.24
```

```
mixed_aug <- augment(mixed_final_fit, new_data=hitters) %>% mutate(.resid=Salary-.pred)
mixed_aug %>%
  gf_qq(~.resid) %>% gf_qqline()
```



```
mixed_aug %>%
  gf_point(.resid~.pred)
```



Results comparison

Compare models fit on the training data and assessed on the test data. The base linear model fit wasn't done above, so do it here.

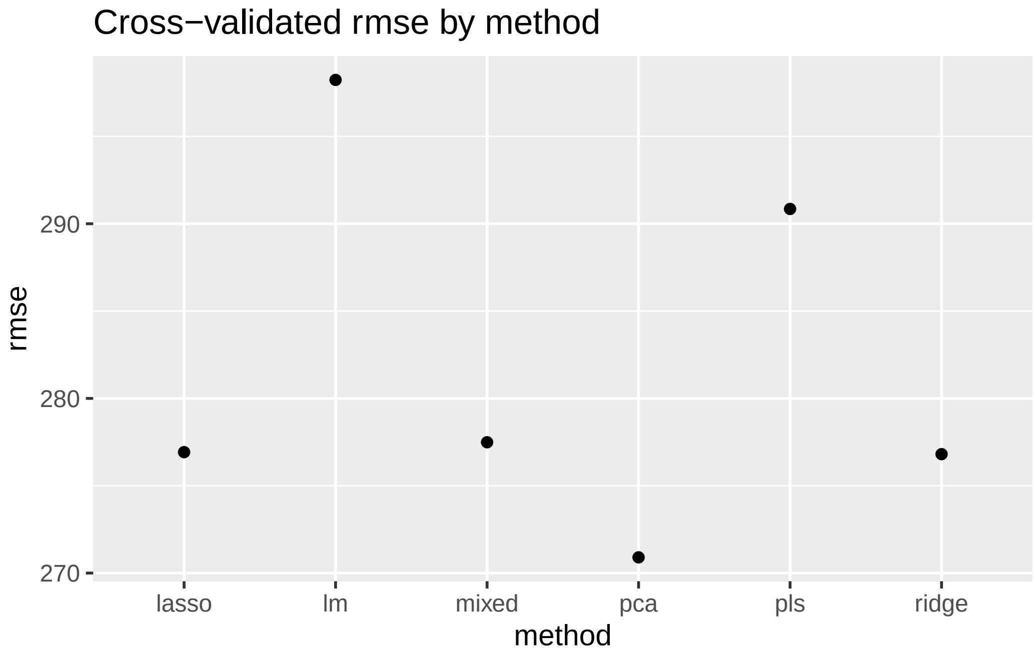
```
lm_final_fit <- fit(lm_spec, Salary ~ ., data = hitters_train)
lm_aug <- augment(lm_final_fit, new_data = hitters_test)
lm_result <- rbind(
  rmse(lm_aug,truth = Salary, estimate = .pred),
  rsq(lm_aug, truth = Salary, estimate = .pred))
lm_result
```

```
# A tibble: 2 x 3
  .metric .estimator .estimate
  <chr>   <chr>          <dbl>
1 rmse    standard     298.
2 rsq     standard       0.421
```

```
results <-
  rbind(
    lm_result %>% mutate(method='lm'),
    ridge_result %>% mutate(method='ridge'),
    lasso_result %>% mutate(method='lasso'),
    pca_result %>% mutate(method='pca'),
    pls_result %>% mutate(method='pls'),
    mixed_result %>% mutate(method='mixed')) %>%
  dplyr::select(method,.metric,.estimate) %>%
  arrange(.metric,.estimate)
results
```

```
# A tibble: 12 x 3
   method .metric .estimate
   <chr>  <chr>       <dbl>
 1 pca    rmse      271.
 2 ridge  rmse      277.
 3 lasso  rmse      277.
 4 mixed  rmse      277.
 5 pls    rmse      291.
 6 lm     rmse      298.
 7 lm     rsq         0.421
 8 pls    rsq         0.447
 9 mixed  rsq        0.491
10 lasso  rsq        0.493
11 ridge  rsq        0.515
12 pca    rsq        0.521
```

```
results %>%
  filter(.metric=='rmse') %>%
  gf_point(.estimate~method) %>%
  gf_labs(y='rmse',title="Cross-validated rmse by method")
```


Cross−validated rmse by method
rmse
lasso
lm
mixed
pca
pls
ridge
method