BASLINE

June 8, 2024

[1]:
```
# It may take several minutes to install those libraries in Watson Studio
install.packages("rlang")
```

```
Updating HTML index of packages in '.Library'
Making 'packages.html' … done
```

[2]:
```
# It may take several minutes to install those libraries in Watson Studio
library("tidymodels")
library("tidyverse")
library("stringr")
```

```
Warning message:
"replacing previous import ‘lifecycle::last_warnings’ by ‘rlang::last_warnings’
when loading ‘tibble’"Warning message:
"replacing previous import ‘ellipsis::check_dots_unnamed’ by
‘rlang::check_dots_unnamed’ when loading ‘tibble’"Warning message:
"replacing previous import ‘ellipsis::check_dots_used’ by
‘rlang::check_dots_used’ when loading ‘tibble’"Warning message:
"replacing previous import ‘ellipsis::check_dots_empty’ by
‘rlang::check_dots_empty’ when loading ‘tibble’"  Attaching packages
                     tidymodels 0.1.0
 broom     0.5.6       recipes   0.1.12
 dials     0.0.6       rsample   0.0.5
 dplyr     0.8.5       tibble    3.0.1
 ggplot2   3.3.0       tune      0.1.0
 infer     0.5.1       workflows 0.1.1
 parsnip   0.1.0       yardstick 0.0.6
 purrr     0.3.4
  Conflicts                       tidymodels_conflicts()
 purrr::discard()  masks scales::discard()
 dplyr::filter()   masks stats::filter()
 dplyr::lag()      masks stats::lag()
 ggplot2::margin() masks dials::margin()
 recipes::step()   masks stats::step()
  Attaching packages                      tidyverse 1.3.0
 readr   1.3.1      forcats 0.5.0
 stringr 1.4.0
  Conflicts                        tidyverse_conflicts()
```

```
 readr::col_factor() masks scales::col_factor()
 purrr::discard()    masks scales::discard()
 dplyr::filter()     masks stats::filter()
 stringr::fixed()    masks recipes::fixed()
 dplyr::lag()        masks stats::lag()
 ggplot2::margin()   masks dials::margin()
 readr::spec()       masks yardstick::spec()
```

[3]:
```
# Dataset URL
dataset_url <- "https://cf-courses-data.s3.us.cloud-object-storage.appdomain.
 ↪cloud/IBMDeveloperSkillsNetwork-RP0321EN-SkillsNetwork/labs/datasets/
 ↪seoul_bike_sharing_converted_normalized.csv"
bike_sharing_df <- read_csv(dataset_url)
spec(bike_sharing_df)
```

```
Parsed with column specification:
cols(
  .default = col_double(),
  DATE = col_character(),
  FUNCTIONING_DAY = col_character()
)
See spec(…) for full column specifications.
```

```
cols(
  DATE = col_character(),
  RENTED_BIKE_COUNT = col_double(),
  TEMPERATURE = col_double(),
  HUMIDITY = col_double(),
  WIND_SPEED = col_double(),
  VISIBILITY = col_double(),
  DEW_POINT_TEMPERATURE = col_double(),
  SOLAR_RADIATION = col_double(),
  RAINFALL = col_double(),
  SNOWFALL = col_double(),
  FUNCTIONING_DAY = col_character(),
  `0` = col_double(),
  `1` = col_double(),
  `10` = col_double(),
  `11` = col_double(),
  `12` = col_double(),
  `13` = col_double(),
  `14` = col_double(),
  `15` = col_double(),
  `16` = col_double(),
  `17` = col_double(),
  `18` = col_double(),
  `19` = col_double(),
  `2` = col_double(),
```

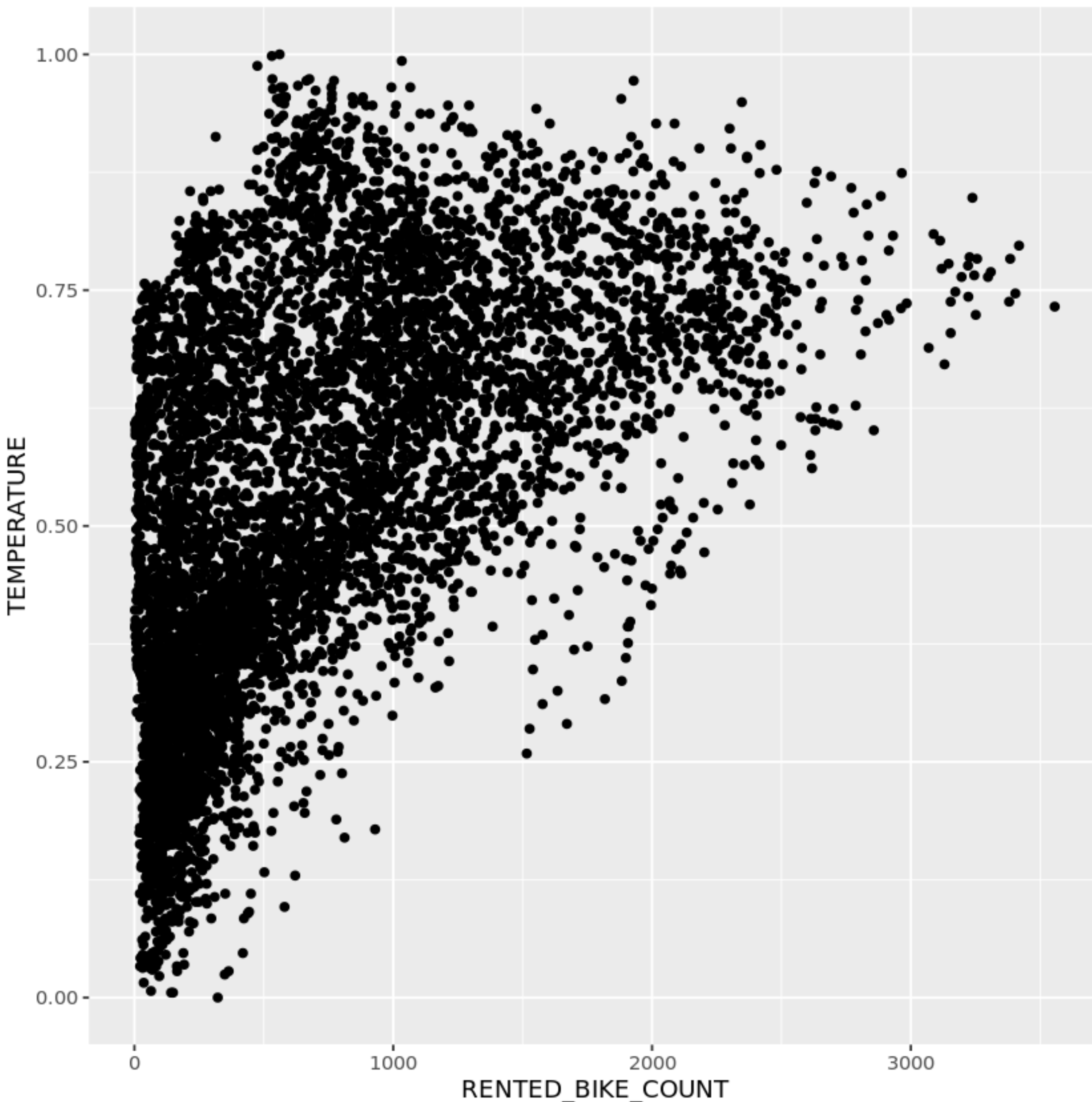
```
  `20` = col_double(),
  `21` = col_double(),
  `22` = col_double(),
  `23` = col_double(),
  `3` = col_double(),
  `4` = col_double(),
  `5` = col_double(),
  `6` = col_double(),
  `7` = col_double(),
  `8` = col_double(),
  `9` = col_double(),
  AUTUMN = col_double(),
  SPRING = col_double(),
  SUMMER = col_double(),
  WINTER = col_double(),
  HOLIDAY = col_double(),
  NO_HOLIDAY = col_double()
)
```

[4]:
```
bike_sharing_df <- bike_sharing_df %>%
                   select(-DATE, -FUNCTIONING_DAY)
```

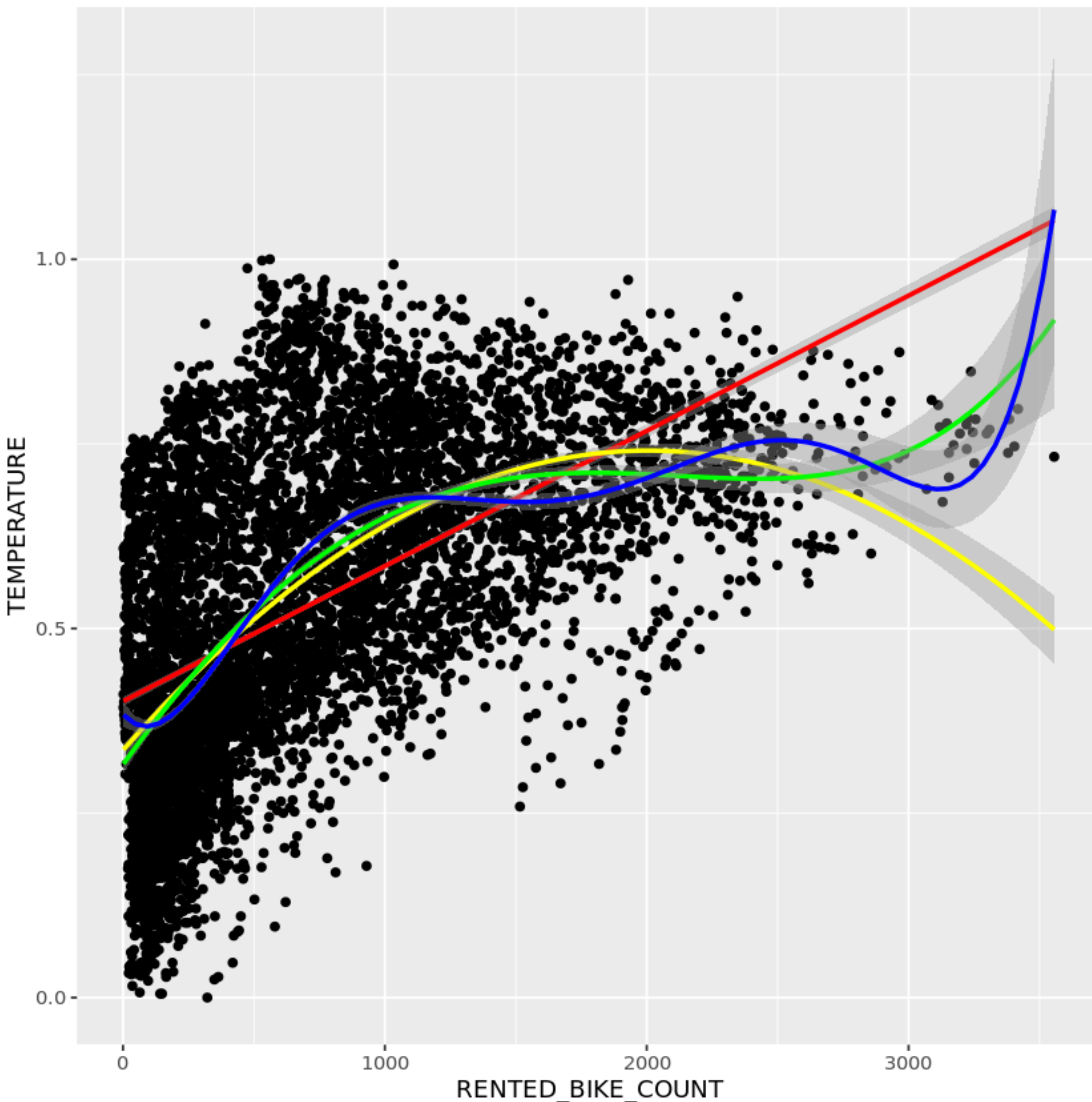
[5]:
```
lm_spec <- linear_reg() %>%
  set_engine("lm") %>%
  set_mode("regression")
```

[6]:
```
set.seed(1234)
data_split <- initial_split(bike_sharing_df, prop = 4/5)
train_data <- training(data_split)
test_data <- testing(data_split)
```

[7]:
```
###TASK: Add polynomial terms
ggplot(data = train_data, aes(RENTED_BIKE_COUNT, TEMPERATURE)) +
    geom_point()
```



[8]:
```
# Plot the higher order polynomial fits
ggplot(data=train_data, aes(RENTED_BIKE_COUNT, TEMPERATURE)) +
    geom_point() +
    geom_smooth(method = "lm", formula = y ~ x, color="red") +
    geom_smooth(method = "lm", formula = y ~ poly(x, 2), color="yellow") +
    geom_smooth(method = "lm", formula = y ~ poly(x, 4), color="green") +
    geom_smooth(method = "lm", formula = y ~ poly(x, 6), color="blue")
```



[9]:
```
# Fit a linear model with higher order polynomial on some important variables
lm_poly <- lm(RENTED_BIKE_COUNT ~ poly(TEMPERATURE, 6) +
                poly(HUMIDITY, 4)+
                poly(RAINFALL,2), data = train_data)
summary(lm_poly$fit)
```

```
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
 -714.7   354.7   745.4   732.2  1135.3  1467.7
```

[10]:
```
lm_poly_pred <- predict(lm_poly, newdata = test_data) #predict
test_results_poly = data.frame(PREDICTION = lm_poly_pred, TRUTH =
 ↪test_data$RENTED_BIKE_COUNT) #create df for test results
```

```
#convert all negative prediction to 0 (RENTED_BIKE_COUNT can't be negative)
test_results_poly <- test_results_poly %>%
  mutate(PREDICTION = ifelse(PREDICTION <0, 0, PREDICTION))
```

[11]:
```
# Calculate R-squared and RMSE from the test results
mse <- mean(lm_poly$residuals^2)
mse
```

213625.473794737

[12]:
```
rmse <- sqrt(mse)
rmse
```

462.196358482774

[13]:
```
rmse_poly <- sqrt(mean ( (test_results_poly$TRUTH -␣
↪test_results_poly$PREDICTION)^2) )
rmse_poly
```

450.449931337093

[14]:
```
summary(lm_poly)$r.squared
```

0.486041988782648

[15]:
```
model_1<- c( (lm_poly)$r.squared, rmse_poly)
model_1
```

450.449931337093

[16]:
```
##TASK: Add interaction terms
# Add interaction terms to the poly regression built in previous step

# HINT: You could use `*` operator to create interaction terms such as␣
↪HUMIDITY*TEMPERATURE and make the formula look like:
# RENTED_BIKE_COUNT ~ RAINFALL*HUMIDITY ...
lm_poly_interaction <- lm(RENTED_BIKE_COUNT ~ poly(TEMPERATURE, 6) +␣
↪poly(HUMIDITY, 4)+poly(RAINFALL,2)+
                          RAINFALL*HUMIDITY + TEMPERATURE*HUMIDITY,
                          data = train_data)
summary(lm_poly_interaction)
```

```
Call:
lm(formula = RENTED_BIKE_COUNT ~ poly(TEMPERATURE, 6) + poly(HUMIDITY,
    4) + poly(RAINFALL, 2) + RAINFALL * HUMIDITY + TEMPERATURE *
    HUMIDITY, data = train_data)

Residuals:
```

```
     Min       1Q   Median       3Q      Max
-1323.50  -250.09   -65.22   168.49  2215.00

Coefficients: (3 not defined because of singularities)
                       Estimate Std. Error t value Pr(>|t|)
(Intercept)             1572.85      60.77  25.881  < 2e-16 ***
poly(TEMPERATURE, 6)1  58733.17    1588.09  36.983  < 2e-16 ***
poly(TEMPERATURE, 6)2  -5223.35     481.14 -10.856  < 2e-16 ***
poly(TEMPERATURE, 6)3 -12742.10     484.48 -26.301  < 2e-16 ***
poly(TEMPERATURE, 6)4  -4427.89     461.62  -9.592  < 2e-16 ***
poly(TEMPERATURE, 6)5   -769.85     457.24  -1.684   0.0923 .
poly(TEMPERATURE, 6)6    628.73     458.80   1.370   0.1706
poly(HUMIDITY, 4)1      8120.71    1540.42   5.272 1.39e-07 ***
poly(HUMIDITY, 4)2     -7894.02     499.44 -15.806  < 2e-16 ***
poly(HUMIDITY, 4)3       397.04     484.91   0.819   0.4129
poly(HUMIDITY, 4)4     -2849.57     479.23  -5.946 2.88e-09 ***
poly(RAINFALL, 2)1    -46450.19   20924.49  -2.220   0.0265 *
poly(RAINFALL, 2)2      1329.92     533.63   2.492   0.0127 *
RAINFALL                     NA         NA      NA       NA
HUMIDITY                     NA         NA      NA       NA
TEMPERATURE                  NA         NA      NA       NA
RAINFALL:HUMIDITY      16016.54    7571.09   2.115   0.0344 *
HUMIDITY:TEMPERATURE   -2806.97     161.41 -17.391  < 2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 452.6 on 6758 degrees of freedom
Multiple R-squared:  0.5082,	Adjusted R-squared:  0.5072
F-statistic: 498.8 on 14 and 6758 DF,  p-value: < 2.2e-16
```

[17]:
```
# Calculate R-squared and RMSE for the new model to see if performance has improved
lm_poly_interaction_pred <- predict(lm_poly_interaction, newdata = test_data) #predict
test_results_poly_interaction = data.frame(PREDICTION = lm_poly_interaction_pred, TRUTH = test_data$RENTED_BIKE_COUNT) #create df for test results

#convert all negative prediction to 0 (RENTED_BIKE_COUNT can't be negative)
test_results_poly_interaction <- test_results_poly_interaction %>%
  mutate(PREDICTION = ifelse(PREDICTION <0, 0, PREDICTION))
```

```
Warning message in predict.lm(lm_poly_interaction, newdata = test_data):
"prediction from a rank-deficient fit may be misleading"
```

[18]:
```
mse <- mean(lm_poly_interaction$residuals^2)
mse
```

204422.882763984

[19]:
```
rmse <- sqrt(mse)
rmse
```

452.131488357075

[20]:
```
rmse_poly_interaction <- sqrt(mean ( (test_results_poly_interaction$TRUTH -
  ↪test_results_poly_interaction$PREDICTION)^2) )
rmse_poly_interaction
```

440.007344459465

[21]:
```
summary(lm_poly_interaction)$r.squared
```

0.50818235107276

[22]:
```
model_2<-c( (lm_poly_interaction)$r.squared, rmse_poly_interaction)
model_2
```

440.007344459465

[23]:
```
##TASK: Add regularization
#TODO: Define a linear regression model specification glmnet_spec using glmnet
  ↪engine
# HINT: Use linear_reg() function with two parameters: penalty and mixture
# - penalty controls the intensity of model regularization
# - mixture controls the tradeoff between L1 and L2 regularizations
# You could manually try different parameter combinations or use grid search to
  ↪find optimal combinations
```

[24]:
```
install.packages('glmnet')
```

```
Warning message:
"package ‘glmnet’ is not available (for R version 3.5.1)"
```

[25]:
```
library('glmnet')
```

```
Loading required package: Matrix

Attaching package: ‘Matrix’

The following objects are masked from ‘package:tidyr’:

    expand, pack, unpack

Loading required package: foreach
```

```
Attaching package: ‘foreach’

The following objects are masked from ‘package:purrr’:

    accumulate, when

Loaded glmnet 2.0-18
```

[26]:

```
lm_glmnet <- lm(RENTED_BIKE_COUNT ~ RAINFALL*HUMIDITY*TEMPERATURE + SPRING*SUMMER +
       poly(RAINFALL, 8) + poly(HUMIDITY, 5) +  poly(TEMPERATURE, 5) + poly(DEW_POINT_TEMPERATURE, 5) + poly(SOLAR_RADIATION, 5) + poly(SNOWFALL,5) +
       SPRING + SUMMER  + HOLIDAY + WIND_SPEED + VISIBILITY
       ,
      data = train_data)
summary(lm_glmnet)
```

```
Call:
lm(formula = RENTED_BIKE_COUNT ~ RAINFALL * HUMIDITY * TEMPERATURE +
    SPRING * SUMMER + poly(RAINFALL, 8) + poly(HUMIDITY, 5) +
    poly(TEMPERATURE, 5) + poly(DEW_POINT_TEMPERATURE, 5) + poly(SOLAR_RADIATION,
    5) + poly(SNOWFALL, 5) + SPRING + SUMMER + HOLIDAY + WIND_SPEED +
    VISIBILITY, data = train_data)

Residuals:
    Min      1Q  Median      3Q     Max
-1521.2  -230.7   -38.6   181.6  1926.2

Coefficients: (4 not defined because of singularities)
                                 Estimate Std. Error t value Pr(>|t|)
(Intercept)                      -1544.64     387.76  -3.984 6.86e-05 ***
RAINFALL                        -12758.35   29015.16  -0.440 0.660159
HUMIDITY                          2549.84     466.54   5.465 4.78e-08 ***
TEMPERATURE                       5814.77     578.55  10.051  < 2e-16 ***
SPRING                             -78.89      14.31  -5.511 3.69e-08 ***
SUMMER                            -116.79      22.01  -5.305 1.16e-07 ***
poly(RAINFALL, 8)1                     NA         NA      NA       NA
poly(RAINFALL, 8)2                2929.30     565.90   5.176 2.33e-07 ***
poly(RAINFALL, 8)3               -2233.48     477.67  -4.676 2.99e-06 ***
poly(RAINFALL, 8)4                1636.14     439.41   3.723 0.000198 ***
poly(RAINFALL, 8)5               -1776.17     432.00  -4.111 3.98e-05 ***
poly(RAINFALL, 8)6                1813.11     425.91   4.257 2.10e-05 ***
poly(RAINFALL, 8)7               -1207.23     438.80  -2.751 0.005953 **
poly(RAINFALL, 8)8                1284.62     424.61   3.025 0.002492 **
```

```
poly(HUMIDITY, 5)1                     NA         NA      NA       NA
poly(HUMIDITY, 5)2               -7056.55    1027.89  -6.865 7.24e-12 ***
poly(HUMIDITY, 5)3                3497.18     639.55   5.468 4.71e-08 ***
poly(HUMIDITY, 5)4               -2764.91     681.51  -4.057 5.03e-05 ***
poly(HUMIDITY, 5)5                 277.09     650.31   0.426 0.670055
poly(TEMPERATURE, 5)1                  NA         NA      NA       NA
poly(TEMPERATURE, 5)2           -14339.51    3453.18  -4.153 3.33e-05 ***
poly(TEMPERATURE, 5)3           -13326.04     630.87 -21.123  < 2e-16 ***
poly(TEMPERATURE, 5)4            -5839.59     556.00 -10.503  < 2e-16 ***
poly(TEMPERATURE, 5)5               57.08     502.72   0.114 0.909604
poly(DEW_POINT_TEMPERATURE, 5)1   8480.99    9731.91   0.871 0.383533
poly(DEW_POINT_TEMPERATURE, 5)2  16242.63    4850.64   3.349 0.000817 ***
poly(DEW_POINT_TEMPERATURE, 5)3  -1314.24     611.24  -2.150 0.031583 *
poly(DEW_POINT_TEMPERATURE, 5)4   2169.14     557.89   3.888 0.000102 ***
poly(DEW_POINT_TEMPERATURE, 5)5    654.11     466.49   1.402 0.160903
poly(SOLAR_RADIATION, 5)1       -11089.68     614.92 -18.034  < 2e-16 ***
poly(SOLAR_RADIATION, 5)2        -7146.94     444.56 -16.076  < 2e-16 ***
poly(SOLAR_RADIATION, 5)3         6033.41     423.86  14.234  < 2e-16 ***
poly(SOLAR_RADIATION, 5)4        -3549.73     419.76  -8.456  < 2e-16 ***
poly(SOLAR_RADIATION, 5)5         3302.24     419.76   7.867 4.21e-15 ***
poly(SNOWFALL, 5)1               -1326.72     483.43  -2.744 0.006078 **
poly(SNOWFALL, 5)2                 441.23     438.76   1.006 0.314634
poly(SNOWFALL, 5)3                -371.01     445.06  -0.834 0.404534
poly(SNOWFALL, 5)4                 283.04     447.23   0.633 0.526844
poly(SNOWFALL, 5)5                -200.26     438.58  -0.457 0.647970
HOLIDAY                           -133.15      24.33  -5.472 4.61e-08 ***
WIND_SPEED                         418.67      41.54  10.079  < 2e-16 ***
VISIBILITY                          12.22      22.06   0.554 0.579559
RAINFALL:HUMIDITY                15178.23   30509.23   0.497 0.618855
RAINFALL:TEMPERATURE              1925.03   44867.56   0.043 0.965779
HUMIDITY:TEMPERATURE             -7368.54     956.30  -7.705 1.49e-14 ***
SPRING:SUMMER                          NA         NA      NA       NA
RAINFALL:HUMIDITY:TEMPERATURE    -7118.67   46995.42  -0.151 0.879605
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 416.6 on 6730 degrees of freedom
Multiple R-squared:  0.585,	Adjusted R-squared:  0.5824
F-statistic: 225.9 on 42 and 6730 DF,  p-value: < 2.2e-16
```

[27]:
```
# Calculate R-squared and RMSE for the new model to see if performance has
  ↪improved
lm_glmnet_pred <- predict(lm_glmnet, newdata = test_data) #predict
test_results_lm_glmnet = data.frame(PREDICTION = lm_glmnet_pred, TRUTH =
  ↪test_data$RENTED_BIKE_COUNT) #create df for test results
```

```
#convert all negative prediction to 0 (RENTED_BIKE_COUNT can't be negative)
test_results_lm_glmnet <- test_results_lm_glmnet %>%
  mutate(PREDICTION = ifelse(PREDICTION <0, 0, PREDICTION))
```

```
Warning message in predict.lm(lm_glmnet, newdata = test_data):
"prediction from a rank-deficient fit may be misleading"
```

[28]:
```
mse <- mean(lm_glmnet$residuals^2)
mse
```

172486.797554475

[29]:
```
rmse_lm_glmnet <- sqrt(mse)
rmse_lm_glmnet
```

415.315298965106

[30]:
```
summary(lm_glmnet)$r.squared
```

0.585016852823794

[31]:
```
rsq_lm_glmnet <- summary(lm_glmnet)$r.squared
rsq_lm_glmnet
```

0.585016852823794

[32]:
```
model_3<-c( rsq_lm_glmnet, rmse_lm_glmnet)
model_3
```

1. 0.585016852823794 2. 415.315298965106

[33]:
```
penalty_value <- 10^seq(-4,4, by = 0.5) #penalty values ranging from 10^-4 to
  ↪10^4
x = as.matrix(train_data[,-1]) #define a matrix for CV
y= train_data$RENTED_BIKE_COUNT
```

[34]:
```
cv_ridge <- cv.glmnet(x,y, alpha = 0, lambda = penalty_value, nfolds = 10)
cv_lasso <- cv.glmnet(x,y, alpha = 1, lambda = penalty_value, nfolds = 10)
cv_elasticnet <- cv.glmnet(x,y, alpha = 0.5, lambda = penalty_value, nfolds =
  ↪10)
```

[35]:
```
model_prediction <- function(lm_model, test_data) {
  results <- lm_model %>%
    predict(new_data=test_data) %>%
    mutate(TRUTH=test_data$RENTED_BIKE_COUNT)
  results[results<0] <-0
  return(results)
}
```

[36]:

```
#model evaluation function
model_evaluation <- function(results) {
  rmse = rmse(results, truth=TRUTH, estimate=.pred)
  rsq = rsq(results, truth=TRUTH, estimate=.pred)
  print(rmse)
  print(rsq)
}
```

[37]:

```
glmnet_spec <- linear_reg(penalty = 0.3, mixture=0.5) %>%
  set_engine("glmnet") %>%
  set_mode("regression")
```

[38]:

```
#Ridge (L2) regularization
glmnet <- glmnet_spec %>%
  fit(RENTED_BIKE_COUNT ~ . + poly(TEMPERATURE, 6) + WINTER * `18` +␣
 ↪poly(DEW_POINT_TEMPERATURE, 6) + poly(SOLAR_RADIATION, 6) + SUMMER * `18` +␣
 ↪TEMPERATURE * HUMIDITY + poly(HUMIDITY, 6) , data = train_data)

summary(glmnet)
```

```
        Length Class      Mode
lvl      0     -none-     NULL
spec     6     linear_reg list
fit     12     elnet      list
preproc  5     -none-     list
elapsed  5     proc_time  numeric
```

[39]:

```
glmnet_pred <- model_prediction(glmnet, test_data)
model_evaluation(glmnet_pred)
```

```
# A tibble: 1 x 3
  .metric .estimator .estimate
  <chr>   <chr>          <dbl>
1 rmse    standard        310.
# A tibble: 1 x 3
  .metric .estimator .estimate
  <chr>   <chr>          <dbl>
1 rsq     standard       0.760
```

[40]:

```
glmnet_rsq = rsq(glmnet_pred, truth = TRUTH, estimate = .pred)
glmnet_rsq
glmnet_rmse = rmse(glmnet_pred, truth = TRUTH, estimate = .pred)
glmnet_rmse
```

A tibble: 1 × 3

.metric <chr>	.estimator <chr>	.estimate <dbl>
rsq	standard	0.7598974

A tibble: 1 × 3

.metric	.estimator	.estimate
<chr>	<chr>	<dbl>
rmse	standard	310.3593

[41]:
```
model_4<-c( glmnet_rsq, glmnet_rmse)
model_4
```

$.metric 'rsq'

$.estimator 'standard'

$.estimate 0.759897410578074

$.metric 'rmse'

$.estimator 'standard'

$.estimate 310.359286091261

[42]:
```
bike_recipe <-
recipe(RENTED_BIKE_COUNT ~ ., data = train_data)
```

[43]:
```
ridge_spec <- linear_reg(penalty = 0.1, mixture = 0) %>%
set_engine("glmnet")
```

[44]:
```
ridge_wf <- workflow() %>%
add_recipe(bike_recipe)
```

[45]:
```
ridge_fit <- ridge_wf %>%
add_model(ridge_spec) %>%
fit(data = train_data)
```

[46]:
```
ridge_fit %>%
pull_workflow_fit() %>%
tidy()
```

A tibble: 3900 × 5

term	step	estimate	lambda	dev.ratio
<chr>	<dbl>	<dbl>	<dbl>	<dbl>
(Intercept)	1	7.321896e+02	360429.6	2.697382e-36
TEMPERATURE	1	1.723584e-33	360429.6	2.697382e-36
HUMIDITY	1	-6.211420e-34	360429.6	2.697382e-36
WIND_SPEED	1	5.866815e-34	360429.6	2.697382e-36
VISIBILITY	1	4.424783e-34	360429.6	2.697382e-36
DEW_POINT_TEMPERATURE	1	1.132251e-33	360429.6	2.697382e-36
SOLAR_RADIATION	1	7.082231e-34	360429.6	2.697382e-36
RAINFALL	1	-2.418017e-33	360429.6	2.697382e-36
SNOWFALL	1	-2.013842e-33	360429.6	2.697382e-36
0	1	-1.908464e-34	360429.6	2.697382e-36
1	1	-3.015741e-34	360429.6	2.697382e-36
10	1	-1.825679e-34	360429.6	2.697382e-36
11	1	-1.253324e-34	360429.6	2.697382e-36
12	1	5.270027e-36	360429.6	2.697382e-36
13	1	1.713391e-35	360429.6	2.697382e-36
14	1	5.638035e-35	360429.6	2.697382e-36
15	1	1.259089e-34	360429.6	2.697382e-36
16	1	2.603201e-34	360429.6	2.697382e-36
17	1	4.714521e-34	360429.6	2.697382e-36
18	1	8.759100e-34	360429.6	2.697382e-36
19	1	5.544040e-34	360429.6	2.697382e-36
2	1	-4.276538e-34	360429.6	2.697382e-36
20	1	3.259183e-34	360429.6	2.697382e-36
21	1	3.725101e-34	360429.6	2.697382e-36
22	1	2.157686e-34	360429.6	2.697382e-36
23	1	-3.045241e-35	360429.6	2.697382e-36
3	1	-5.512708e-34	360429.6	2.697382e-36
4	1	-6.306256e-34	360429.6	2.697382e-36
5	1	-6.229421e-34	360429.6	2.697382e-36
6	1	-4.578700e-34	360429.6	2.697382e-36
0	100	-53.33714	36.04296	0.660746
1	100	-159.56285	36.04296	0.660746
10	100	-233.72935	36.04296	0.660746
11	100	-237.74294	36.04296	0.660746
12	100	-193.71213	36.04296	0.660746
13	100	-185.18691	36.04296	0.660746
14	100	-183.16812	36.04296	0.660746
15	100	-107.12601	36.04296	0.660746
16	100	28.41071	36.04296	0.660746
17	100	277.81890	36.04296	0.660746
18	100	712.16005	36.04296	0.660746
19	100	471.91020	36.04296	0.660746
2	100	-282.50051	36.04296	0.660746
20	100	338.33928	36.04296	0.660746
21	100	386.00317	36.04296	0.660746
22	100	263.44637	36.04296	0.660746
23	100	49.12452	36.04296	0.660746
3	100	-347.25625	36.04296	0.660746
4	100	-409.55655	36.04296	0.660746
5	100	-400.85388	36.04296	0.660746

[47]:
```
ridge_fit_pred <- model_prediction(ridge_fit, test_data)
model_evaluation(ridge_fit_pred)
```

```
# A tibble: 1 x 3
  .metric .estimator .estimate
  <chr>   <chr>          <dbl>
1 rmse    standard        362.
# A tibble: 1 x 3
  .metric .estimator .estimate
  <chr>   <chr>          <dbl>
1 rsq     standard       0.676
```

[48]:
```
ridge_rsq = rsq(ridge_fit_pred, truth = TRUTH, estimate = .pred)
ridge_rsq
ridge_rmse = rmse(ridge_fit_pred, truth = TRUTH, estimate = .pred)
ridge_rmse
```

A tibble: 1 × 3

.metric <chr>	.estimator <chr>	.estimate <dbl>
rsq	standard	0.6762273

A tibble: 1 × 3

.metric <chr>	.estimator <chr>	.estimate <dbl>
rmse	standard	362.4059

[49]:
```
model_5<-c( ridge_rsq, ridge_rmse)
model_5
```

$.metric 'rsq'

$.estimator 'standard'

$.estimate 0.676227319666284

$.metric 'rmse'

$.estimator 'standard'

$.estimate 362.405922372514

[50]:
```
#Lasso (L1) regularization
lasso_spec <- linear_reg(penalty = 0.1, mixture = 1) %>%
set_engine("glmnet")
```

[51]:
```
lasso_wf <- workflow() %>%
add_recipe(bike_recipe)
```

[52]:
```
lasso_fit <- lasso_wf %>%
add_model(lasso_spec) %>%
fit(data = train_data)
```

[53]:

```
lasso_fit %>%
pull_workflow_fit() %>%
tidy()
```

A tibble: 1879 × 5

term	step	estimate	lambda	dev.ratio
<chr>	<dbl>	<dbl>	<dbl>	<dbl>
(Intercept)	1	732.189576	360.4296	0.00000000
(Intercept)	2	651.034639	328.4100	0.05306502
TEMPERATURE	2	151.587235	328.4100	0.05306502
(Intercept)	3	577.089283	299.2350	0.09712052
TEMPERATURE	3	289.707878	299.2350	0.09712052
(Intercept)	4	509.713029	272.6518	0.13369618
TEMPERATURE	4	415.558263	272.6518	0.13369618
(Intercept)	5	448.322295	248.4302	0.16406193
TEMPERATURE	5	530.228447	248.4302	0.16406193
(Intercept)	6	392.385345	226.3603	0.18927211
TEMPERATURE	6	634.711648	226.3603	0.18927211
(Intercept)	7	341.417680	206.2511	0.21020205
TEMPERATURE	7	729.912850	206.2511	0.21020205
(Intercept)	8	294.977841	187.9283	0.22757845
TEMPERATURE	8	816.656639	187.9283	0.22757845
(Intercept)	9	252.663589	171.2333	0.24200464
TEMPERATURE	9	895.694349	171.2333	0.24200464
(Intercept)	10	228.279759	156.0214	0.26649123
TEMPERATURE	10	970.894738	156.0214	0.26649123
HUMIDITY	10	-30.125273	156.0214	0.26649123
18	10	46.159676	156.0214	0.26649123
(Intercept)	11	229.807470	142.1609	0.29794699
TEMPERATURE	11	1046.718179	142.1609	0.29794699
HUMIDITY	11	-105.603742	142.1609	0.29794699
18	11	106.303536	142.1609	0.29794699
(Intercept)	12	235.946076	129.5317	0.32455931
TEMPERATURE	12	1109.929902	129.5317	0.32455931
HUMIDITY	12	-175.317166	129.5317	0.32455931
18	12	161.273590	129.5317	0.32455931
WINTER	12	-4.141401	129.5317	0.32455931
RAINFALL	77	-2053.67578	0.3062763	0.6624083
SNOWFALL	77	209.52983	0.3062763	0.6624083
0	77	28.71639	0.3062763	0.6624083
1	77	-79.24627	0.3062763	0.6624083
10	77	-176.69948	0.3062763	0.6624083
11	77	-187.40316	0.3062763	0.6624083
12	77	-147.08766	0.3062763	0.6624083
13	77	-139.58122	0.3062763	0.6624083
14	77	-140.63063	0.3062763	0.6624083
15	77	-58.85878	0.3062763	0.6624083
16	77	82.34617	0.3062763	0.6624083
17	77	350.76622	0.3062763	0.6624083
18	77	814.20083	0.3062763	0.6624083
19	77	567.70456	0.3062763	0.6624083
2	77	-209.18700	0.3062763	0.6624083
20	77	432.12483	0.3062763	0.6624083
21	77	483.77962	0.3062763	0.6624083
22	77	358.06191	0.3062763	0.6624083
23	77	133.70291	0.3062763	0.6624083
3	77	-275.17982	0.3062763	0.6624083

[54]:
```
lasso_fit_pred <- model_prediction(lasso_fit, test_data)
model_evaluation(lasso_fit_pred)
```

```
# A tibble: 1 x 3
  .metric .estimator .estimate
  <chr>   <chr>          <dbl>
1 rmse    standard        361.
# A tibble: 1 x 3
  .metric .estimator .estimate
  <chr>   <chr>          <dbl>
1 rsq     standard       0.677
```

[55]:
```
lasso_rsq = rsq(lasso_fit_pred, truth = TRUTH, estimate = .pred)
lasso_rsq
lasso_rmse = rmse(lasso_fit_pred, truth = TRUTH, estimate = .pred)
lasso_rmse
```

A tibble: 1 × 3

.metric <chr>	.estimator <chr>	.estimate <dbl>
rsq	standard	0.6773537

A tibble: 1 × 3

.metric <chr>	.estimator <chr>	.estimate <dbl>
rmse	standard	360.5281

[56]:
```
model_6 <- c( lasso_rsq, lasso_rmse)
model_6
```

$.metric 'rsq'

$.estimator 'standard'

$.estimate 0.677353737578952

$.metric 'rmse'

$.estimator 'standard'

$.estimate 360.528080394777

[57]:
```
#Elastic Net (L1 and L2) Regularization
elasticnet_spec <- linear_reg(penalty = 0.1, mixture = 0.3) %>%
set_engine("glmnet")
```

[58]:
```
elasticnet_wf <- workflow() %>%
add_recipe(bike_recipe)
```

[59]:
```
elasticnet_fit <- elasticnet_wf %>%
add_model(elasticnet_spec) %>%
fit(data = train_data)
```

[60]:

```
elasticnet_fit %>%
pull_workflow_fit() %>%
tidy()
```

A tibble: 1974 × 5

term	step	estimate	lambda	dev.ratio
<chr>	<dbl>	<dbl>	<dbl>	<dbl>
(Intercept)	1	732.189576	1201.4321	0.00000000
(Intercept)	2	695.108581	1094.7001	0.02485834
TEMPERATURE	2	69.262644	1094.7001	0.02485834
(Intercept)	3	657.729345	997.4500	0.04887422
TEMPERATURE	3	139.082364	997.4500	0.04887422
(Intercept)	4	620.211394	908.8393	0.07192683
TEMPERATURE	4	209.161189	908.8393	0.07192683
(Intercept)	5	586.134311	828.1005	0.09532994
TEMPERATURE	5	275.195574	828.1005	0.09532994
WINTER	5	-5.023071	828.1005	0.09532994
(Intercept)	6	561.014883	754.5343	0.12045688
TEMPERATURE	6	330.306588	754.5343	0.12045688
WINTER	6	-22.291224	754.5343	0.12045688
(Intercept)	7	536.190397	687.5036	0.14333091
TEMPERATURE	7	384.413700	687.5036	0.14333091
WINTER	7	-38.604405	687.5036	0.14333091
(Intercept)	8	511.716667	626.4277	0.16404458
TEMPERATURE	8	437.391024	626.4277	0.16404458
WINTER	8	-53.917006	626.4277	0.16404458
(Intercept)	9	487.640743	570.7776	0.18270864
TEMPERATURE	9	489.134537	570.7776	0.18270864
WINTER	9	-68.195005	570.7776	0.18270864
(Intercept)	10	463.008704	520.0713	0.20608548
TEMPERATURE	10	538.171642	520.0713	0.20608548
18	10	42.824102	520.0713	0.20608548
WINTER	10	-81.698334	520.0713	0.20608548
(Intercept)	11	468.360688	473.8696	0.23662972
TEMPERATURE	11	587.986387	473.8696	0.23662972
HUMIDITY	11	-50.439772	473.8696	0.23662972
18	11	85.840139	473.8696	0.23662972
SNOWFALL	77	206.921714	1.020921	0.6624009
1	77	-108.770288	1.020921	0.6624009
10	77	-205.670100	1.020921	0.6624009
11	77	-216.227214	1.020921	0.6624009
12	77	-175.844534	1.020921	0.6624009
13	77	-168.337199	1.020921	0.6624009
14	77	-169.341271	1.020921	0.6624009
15	77	-87.702272	1.020921	0.6624009
16	77	53.309911	1.020921	0.6624009
17	77	321.344812	1.020921	0.6624009
18	77	784.180683	1.020921	0.6624009
19	77	537.811959	1.020921	0.6624009
2	77	-238.575058	1.020921	0.6624009
20	77	402.278845	1.020921	0.6624009
21	77	453.854354	1.020921	0.6624009
22	77	328.193591	1.020921	0.6624009
23	77	104.033526	1.020921	0.6624009
3	77	-304.538695	1.020921	0.6624009
4	77	-368.955001	1.020921	0.6624009
5	77	-358.811118	1.020921	0.6624009

[61]:
```
elasticnet_fit_pred <- model_prediction(elasticnet_fit, test_data)
model_evaluation(elasticnet_fit_pred)
```

```
# A tibble: 1 x 3
  .metric .estimator .estimate
  <chr>   <chr>          <dbl>
1 rmse    standard        361.
# A tibble: 1 x 3
  .metric .estimator .estimate
  <chr>   <chr>          <dbl>
1 rsq     standard       0.677
```

[62]:
```
elasticnet_rsq = rsq(elasticnet_fit_pred, truth = TRUTH, estimate = .pred)
elasticnet_rsq
elasticnet_rmse = rmse(elasticnet_fit_pred, truth = TRUTH, estimate = .pred)
elasticnet_rmse
```

A tibble: 1 × 3

.metric <chr>	.estimator <chr>	.estimate <dbl>
rsq	standard	0.6773348

A tibble: 1 × 3

.metric <chr>	.estimator <chr>	.estimate <dbl>
rmse	standard	360.56

[63]:
```
model_7 <- c( elasticnet_rsq, elasticnet_rmse)
model_7
```

$.metric 'rsq'

$.estimator 'standard'

$.estimate 0.677334783941434

$.metric 'rmse'

$.estimator 'standard'

$.estimate 360.559981536539

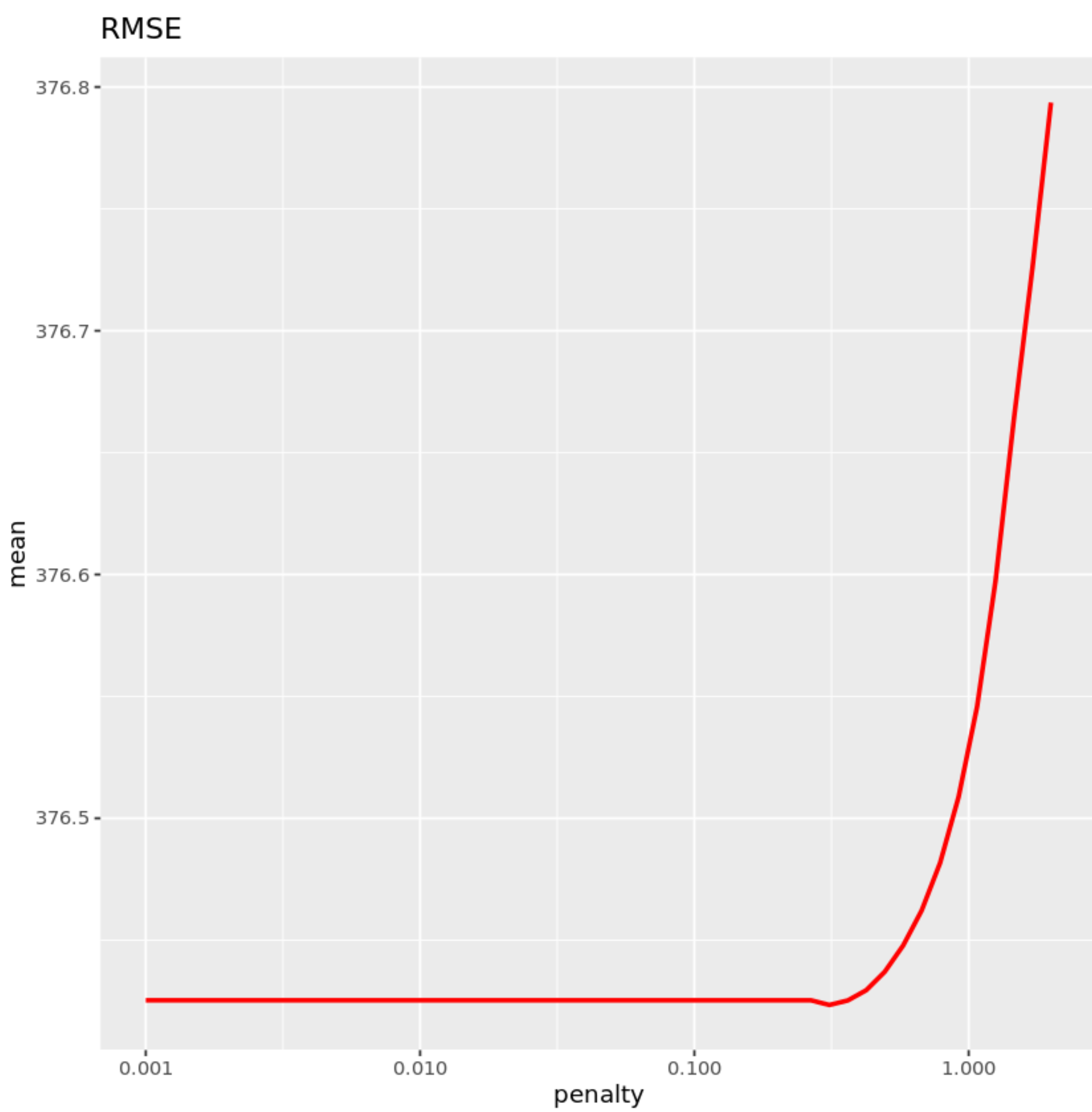
[64]:
```
#Comparing Regularization Types
#Lasso (L1)
#Ridge (L2)
#Elastic net (L1/L2)
tune_spec <- linear_reg(penalty = tune(), mixture = 1) %>%
set_engine("glmnet")
lasso_wf <- workflow() %>%
add_recipe(bike_recipe)
```

[65]:
```
bike_cvfolds <- vfold_cv(train_data)
```

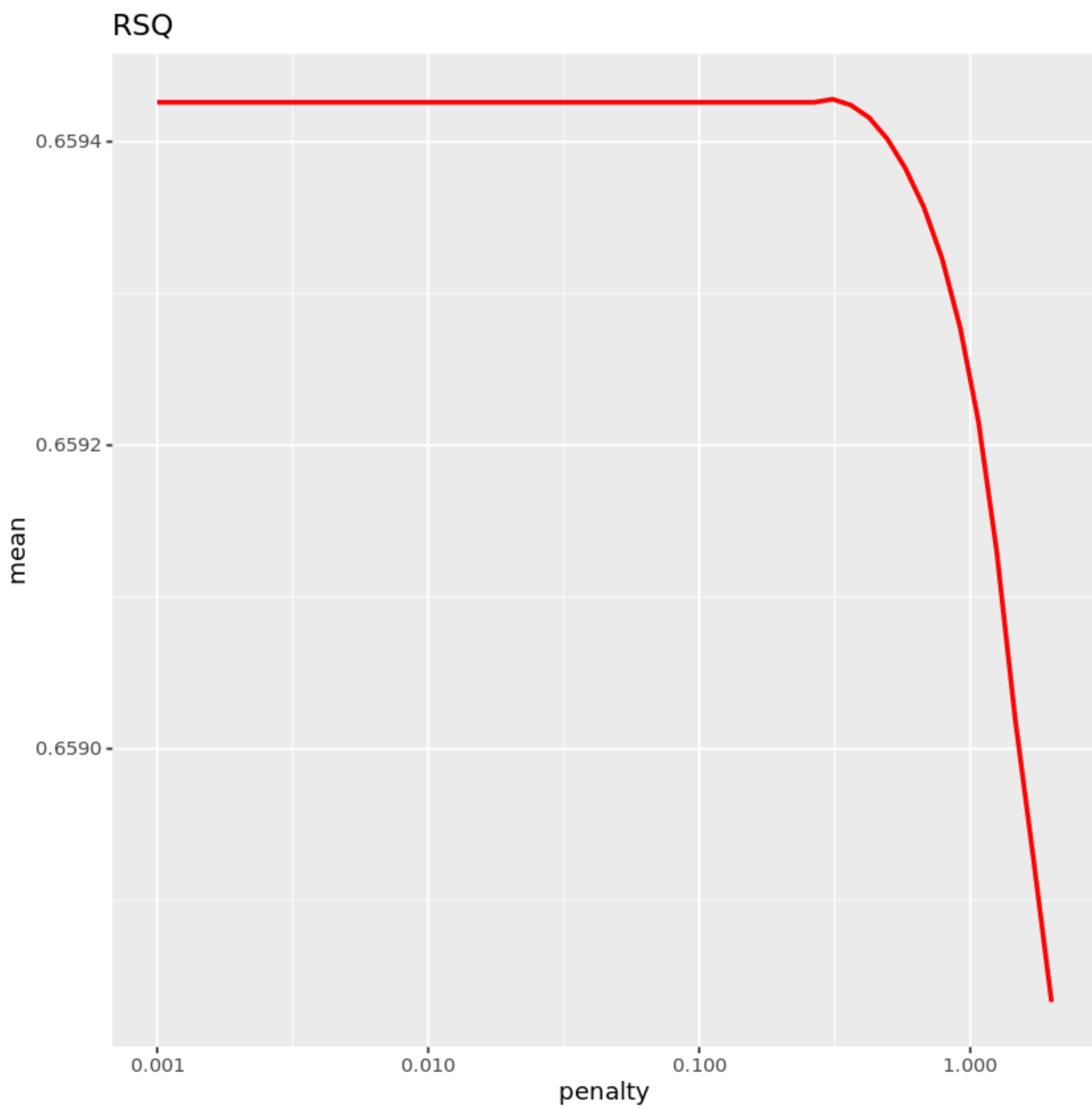
[66]:
```
lambda_grid <- grid_regular(levels = 50,
penalty(range = c(-3, 0.3)))
```

[67]:
```
lasso_grid <- tune_grid(
lasso_wf %>% add_model(tune_spec),
resamples = bike_cvfolds,
grid = lambda_grid)
```

[68]:
```
lasso_grid %>%
collect_metrics() %>%
filter(.metric == "rmse") %>%
ggplot(aes(penalty, mean)) +
geom_line(size=1, color="red") +
scale_x_log10() +
ggtitle("RMSE")
```



```
[69]: lasso_grid %>%
      collect_metrics() %>%
      filter(.metric == "rsq") %>%
      ggplot(aes(penalty, mean)) +
      geom_line(size=1, color="red") +
      scale_x_log10() +
      ggtitle("RSQ")
```




```
[70]: tune_spec <- linear_reg(
      penalty = tune(),
```

```
mixture = 0) %>%
set_engine("glmnet")
ridge_grid <- tune_grid(ridge_wf %>%
add_model(tune_spec),
resamples = bike_cvfolds,
grid = lambda_grid)
```

[71]:
```
show_best(ridge_grid, metric = "rmse")
```

A tibble: 5 × 6

penalty <dbl>	.metric <chr>	.estimator <chr>	mean <dbl>	n <int>	std_err <dbl>
0.001000000	rmse	standard	377.0972	10	5.042446
0.001167742	rmse	standard	377.0972	10	5.042446
0.001363622	rmse	standard	377.0972	10	5.042446
0.001592358	rmse	standard	377.0972	10	5.042446
0.001859464	rmse	standard	377.0972	10	5.042446

[72]:
```
#rsq     rmse
#0.486   450.4
#0.508   440.0
#0.585   415.3
#0.759   310.3
#0.676   362.4
#0.677   360.5
```

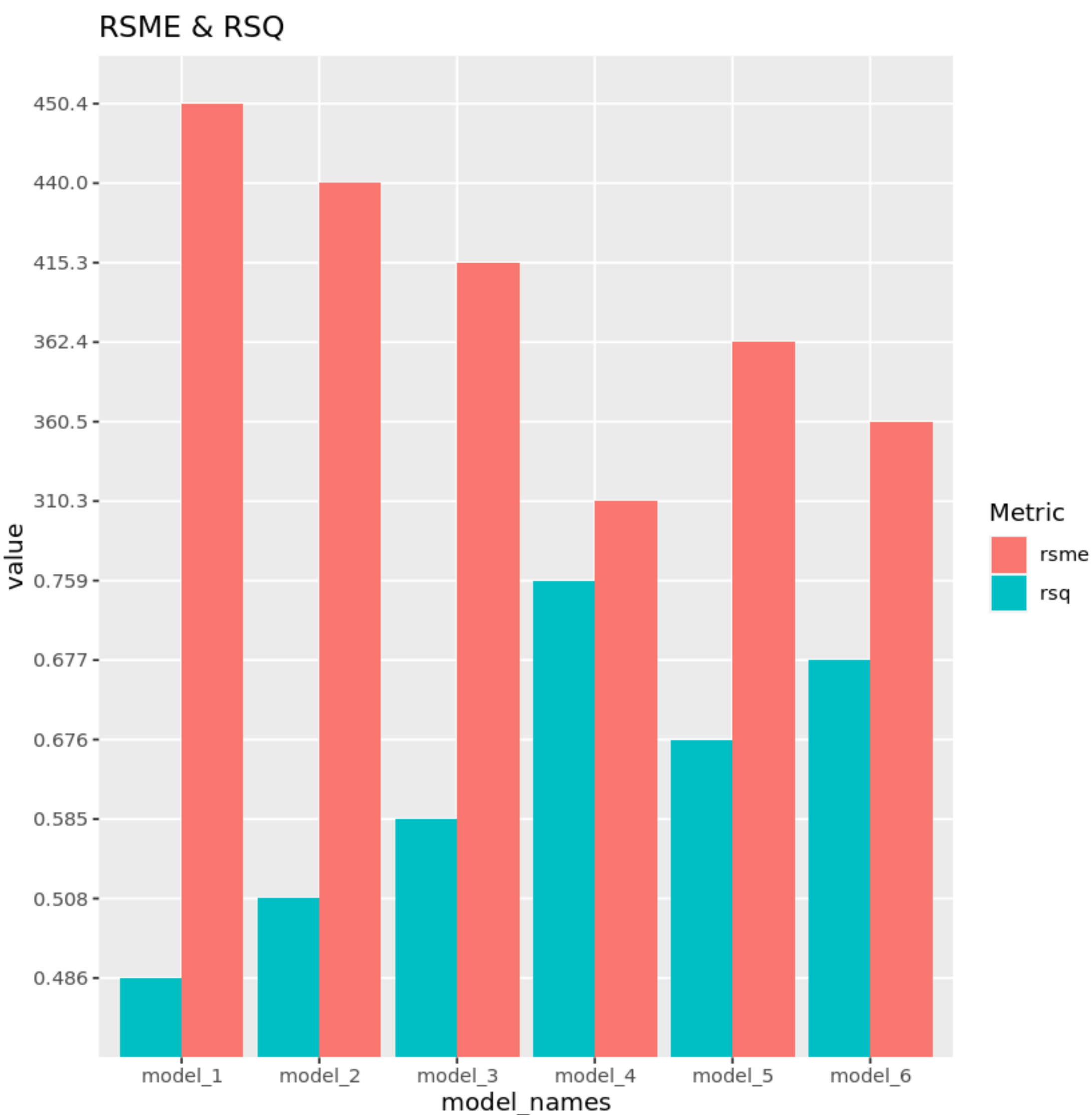
[74]:
```
##TODO: Visualize the saved RMSE and R-squared values using a grouped barchart
# HINT: Use ggplot() + geom_bar()
model_names <- c("model_1", "model_2", "model_3", "model_4", "model_5",
 ↪"model_6")
rsq <- c("0.486", "0.508", "0.585", "0.759", "0.676", "0.677")
rsme <- c( "450.4", "440.0", "415.3", "310.3", "362.4", "360.5" )
comparison_df <- data.frame(model_names, rsq, rsme)
```

[75]:
```
print(comparison_df)
```

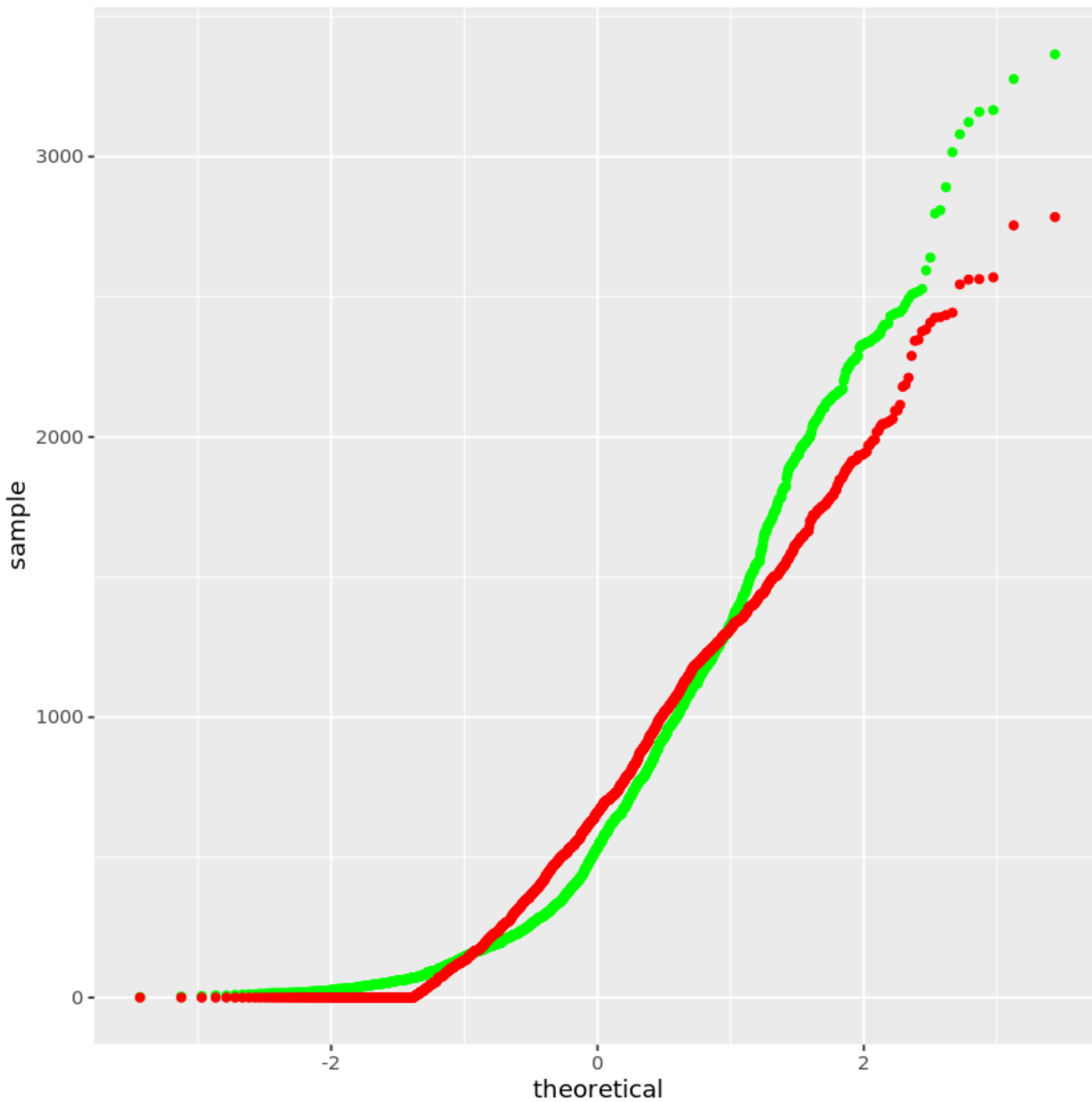
```
  model_names   rsq  rsme
1     model_1 0.486 450.4
2     model_2 0.508 440.0
3     model_3 0.585 415.3
4     model_4 0.759 310.3
5     model_5 0.676 362.4
6     model_6 0.677 360.5
```

[76]:
```
##TODO: Visualize the saved RMSE and R-squared values using a grouped barchart
# HINT: Use ggplot() + geom_bar()
comparison_df %>%
  pivot_longer(!model_names) %>%
```

```
ggplot(aes(x = model_names, y = value, fill = name)) +
geom_bar(stat = "identity", position = "dodge") +
labs(title = "RSME & RSQ", fill = "Metric")
```



[81]:
```
glmnet_pred %>%
  ggplot() +
  stat_qq(aes(sample=TRUTH), color='green') +
  stat_qq(aes(sample=.pred), color='red')
```


3000
2000
1000
0
sample
-2
0
2
theoretical

[ ]: