linear_regression_Hope

```
library(ggplot2)
library(tidymodels)
library(tidyverse)
```

```
# read in data
all_data <- read.csv("~/DSCI445/project-5/CSV Files/merged_data.csv")
str(all_data)
```

```
## 'data.frame':    59388 obs. of  26 variables:
##  $ id                   : chr  "A00147" "A00360" "A00367" "A01054" ...
##  $ num_scars            : int  2 1 1 3 1 0 4 1 1 2 ...
##  $ num_tattoos          : int  1 1 0 0 0 1 1 0 2 0 ...
##  $ num_other_marks      : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ last_name            : chr  "MCCUTCHEON" "BELL" "GARVIN" "TIPTON" ...
##  $ name                 : chr  "JOHN" "HOWARD" "RAYMOND" "DARNELL" ...
##  $ weight               : chr  "185" "167" "245" "166" ...
##  $ hair                 : chr  "Brown" "Gray or Partially Gray" "Black" "Salt and Pepper" ...
##  $ sex                  : chr  "Male" "Male" "Male" "Male" ...
##  $ height               : chr  "67" "69" "72" "67" ...
##  $ race                 : chr  "White" "White" "Black" "Black" ...
##  $ eyes                 : chr  "Blue" "Green" "Brown" "Brown" ...
##  $ parent_institution   : chr  "DIXON CORRECTIONAL CENTER" "PINCKNEYVILLE CORRECTIONAL CENTER" "WESTE
##  $ year_adm             : int  1983 1988 2017 1988 1974 1983 2005 2000 2007 2013 ...
##  $ month_adm            : int  2 2 11 12 2 9 12 9 8 3 ...
##  $ total_counts         : int  1 1 1 3 1 1 2 1 1 1 ...
##  $ class                : chr  "2" "4" "2" "X" ...
##  $ offense_category     : chr  "BURGLARY" "OBSTRUCTING JUSTICE" "THEFT" "SEXUAL OFFENSE" ...
##  $ attempted            : logi  FALSE FALSE FALSE FALSE FALSE FALSE ...
##  $ aggravated           : logi  FALSE FALSE FALSE TRUE FALSE FALSE ...
##  $ armed                : logi  FALSE FALSE FALSE TRUE FALSE FALSE ...
##  $ current_sentence     : num  100 61 3 100 100 0 100 5 50 7 ...
##  $ time_sentenced_prior: num  0 82 85.3 15 100 ...
##  $ life_sentence        : logi  TRUE FALSE FALSE TRUE TRUE FALSE ...
##  $ death_sentence       : logi  FALSE FALSE FALSE FALSE FALSE FALSE ...
##  $ appx_age             : int  34 42 63 34 18 30 50 55 52 63 ...
```

```
# drop id and name columns in dataset for prediction
useful_columns <- all_data |> subset(select = -c(id, last_name, name))
useful_columns$weight <- as.numeric(useful_columns$weight)
```

```
## Warning: NAs introduced by coercion
```

```
useful_columns$height <- as.numeric(useful_columns$height)
```

```
## Warning: NAs introduced by coercion
```

```
useful_columns$attempted <- as.numeric(useful_columns$attempted)
useful_columns$aggravated <- as.numeric(useful_columns$aggravated)
useful_columns$armed <- as.numeric(useful_columns$armed)
```

```r
useful_columns$life_sentence <- as.numeric(useful_columns$life_sentence)
useful_columns$death_sentence <- as.numeric(useful_columns$death_sentence)

useful_columns |> drop_na() -> useful_columns
# convert categorical variables to dummy variables and normalize numerical variables
prep_data <- recipe(current_sentence ~ ., data = useful_columns) |>
  step_dummy(all_nominal_predictors()) |>
  step_normalize(all_predictors())
```

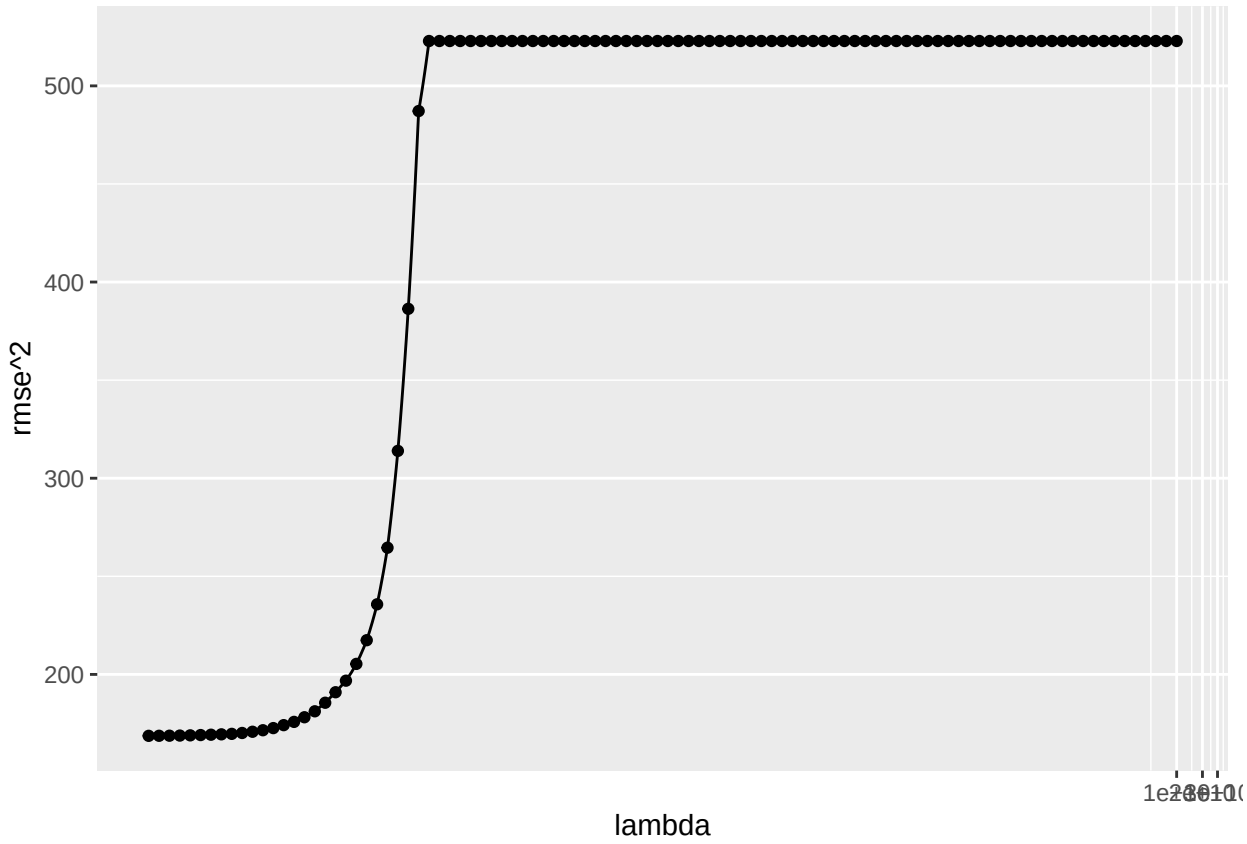
```r
# set up lasso tuning
cv_10fold <- vfold_cv(useful_columns, v = 10)

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

lambda <- lambda <- 10^seq(-2, 10, length.out = 100)
tune_df <- data.frame(lambda = lambda)
```

```r
workflow() |>
  add_model(lasso_spec) |>
  add_recipe(prep_data) |>
  tune_grid(resamples = cv_10fold, grid = tune_df) -> lasso_tune
```

```
## > A | warning: A correlation computation is required, but `estimate` is constant and has 0
##                standard deviation, resulting in a divide by 0 error. `NA` will be returned.
```

```
## There were issues with some computations   A: x1There were issues with some computations   A: x2There
```

```r
lasso_tune |>
  collect_metrics() |>
  select(lambda, .metric, mean) |>
  pivot_wider(names_from = .metric, values_from = mean) |>
  ggplot() +
  geom_line(aes(lambda, rmse^2)) +
  geom_point(aes(lambda, rmse^2)) +
  coord_trans(x = "log10")
```




```
## the penalty I would choose is
show_best(lasso_tune, metric = "rmse", n = 1)
```

```
## # A tibble: 1 x 7
##   lambda .metric .estimator  mean     n std_err .config
##    <dbl> <chr>   <chr>      <dbl> <int>   <dbl> <chr>
## 1   0.01 rmse    standard    13.0    10   0.198 Preprocessor1_Model001
```