AE6

Brandon Leslie

1). Generate 50 X values between 0 and 5.

```
set.seed(88)

ex1 <- tibble(x = runif(50, 0, 5))
```

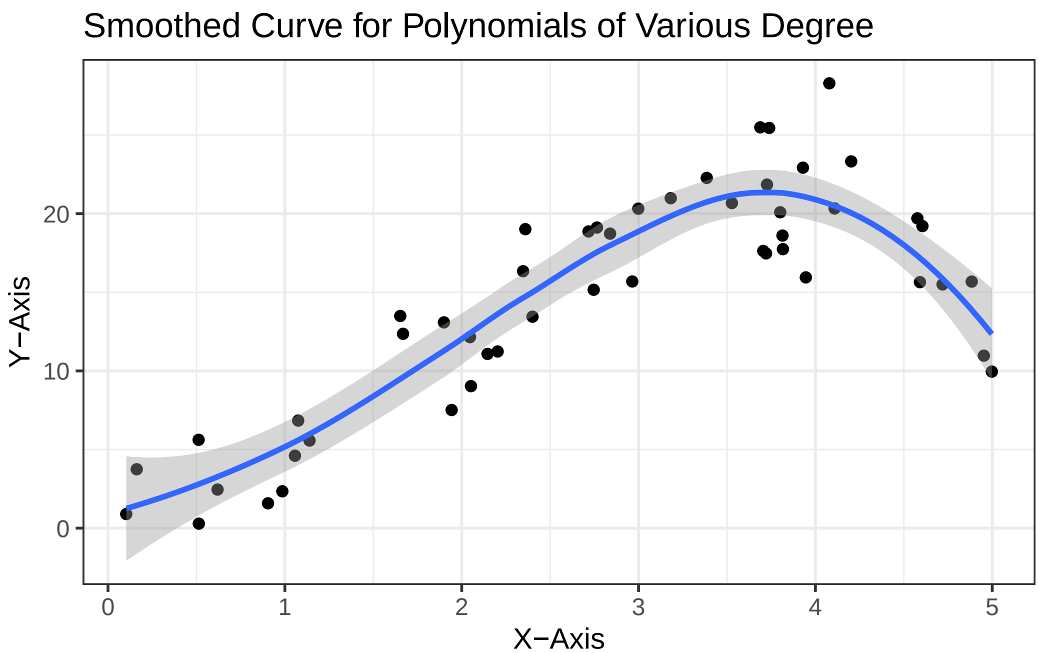
2). Generate 50 y values where $Y = -(X - 0.4)^2(X - 5.5) + 2 + \epsilon$ where ϵ is random normal with $\mu = 0, \sigma = 2.8$. There is no special significance to this form other than it gives a nice looking plot with a vaguely recognizable cubic shape.

```
ex1 <- ex1 %>%
 mutate(y = -(x - 0.4)^2 * (x - 5.5) + 2 + rnorm(50, 0, 2.8))
```

3). Plot a scatterplot of the data, and use geom_smooth() to plot a smoothed curve for polynomials of various degree.

```
ex1 %>% ggplot(aes(x = x, y = y)) +
  geom_point() +
  geom_smooth() + labs(
    x = "X-Axis",
    y = "Y-Axis",
    title = "Smoothed Curve for Polynomials of Various Degree"
```

```
  ) +
  theme_bw()
```



4). Split the data into Train and Test sets, with 70/30 split.

```
set.seed(88)

Splitting <- sample(1:nrow(ex1), size = 0.70 * nrow(ex1))

 Test <- ex1[-Splitting, ]
 Train <- ex1[Splitting, ]
```

5). Fit polynomial regression models of degree 1, 2, 3, 5, 10, 20 on the training set, predict both training and Test sets, and find rmse and r-squared values.

```r
rec <- recipe(y ~ x, data = Train) %>%
    step_poly(x, degree = 4)
rec %>%  prep() %>%  bake(Train) %>%  head()
```

```
# A tibble: 6 x 5
      y x_poly_1 x_poly_2 x_poly_3 x_poly_4
  <dbl>    <dbl>    <dbl>    <dbl>    <dbl>
1 23.3    0.199    0.145   -0.00588  -0.151
2 19.1    0.0150  -0.167   -0.0508    0.171
3 20.3    0.187    0.112   -0.0462   -0.157
4 25.4    0.140   -0.00107 -0.141    -0.0957
5 12.4   -0.126   -0.126    0.207    -0.0668
6  5.62  -0.274    0.178   -0.00116  -0.176
```

```r
rec <- recipe(y ~ x, data = Train) %>%
  step_poly(x, degree = 15)
flow <- workflow() %>%
  add_recipe(rec) %>%
  add_model(linear_reg())

flow_fit <- flow %>% fit(data = Train)

pred_train <- predict(flow_fit, new_data = Train)
pred_test <- predict(flow_fit, new_data = Test)

results_train <- bind_cols(pred_train, Train %>% select(y))
results_test <- bind_cols(pred_test, Test %>% select(y))

metrics <- metric_set(yardstick::rsq, yardstick::rmse)

print("Train Metrics")
```

```
[1] "Train Metrics"
```

```r
results_train %>% metrics(truth = y, .pred)
```

```
# A tibble: 2 x 3
  .metric .estimator .estimate
  <chr>   <chr>          <dbl>
1 rsq     standard       0.929
2 rmse    standard       1.91
```

```r
print("Test Metrics")
```

```
[1] "Test Metrics"
```

```r
results_test %>% metrics(truth = y, .pred)
```

```
# A tibble: 2 x 3
  .metric .estimator .estimate
  <chr>   <chr>          <dbl>
1 rsq     standard      0.0403
2 rmse    standard    338.
```

6). Now repeat the above but use cross validation with k = 5 folds

```r
rec <- recipe(y ~ x, data = Train) %>%
  step_poly(x, degree = 15)

folds <- vfold_cv(Train, v = 5)

flow <- workflow() %>%
  add_recipe(rec) %>%
  add_model(linear_reg())

results <- fit_resamples(flow, resamples = folds, metrics = metric_set
                         (yardstick::rsq, yardstick::rmse))

collect_metrics(results)
```

```
# A tibble: 2 x 6
  .metric .estimator    mean     n  std_err .config
  <chr>   <chr>        <dbl> <int>    <dbl> <chr>
1 rmse    standard   1400.       5 1122.    Preprocessor1_Model1
2 rsq     standard      0.362     5    0.156 Preprocessor1_Model1
```