Polynomial Regression

Author : Rose Ellison

Formula:

$$y = b_0 + b_1x_1 + b_2x_1^2 + \ldots + b_nx_1^n$$

I will be building both a simple and polynomial regression model based off the 'Position_Salaries' data to determe which model fits the data better. In this dataset there are three columns *Position*, *Level*, and *Salary*. Salary is our dependent variable while the other two are our independent variables. We want to use regression to determine if a particuilar new hire's past salary was possibly $160,000 as a region manager.

```
# Set the seed
set.seed(1)

# Importing the data
positions <- read.csv('../../data/Position_Salaries.csv')

dim(positions)

## [1] 10  3

positions

##              Position Level  Salary
## 1    Business Analyst     1   45000
## 2   Junior Consultant     2   50000
## 3   Senior Consultant     3   60000
## 4             Manager     4   80000
## 5     Country Manager     5  110000
## 6      Region Manager     6  150000
## 7             Partner     7  200000
## 8      Senior Partner     8  300000
## 9             C-level     9  500000
## 10                CEO    10 1000000
```

Preparing the data

From the table we can see there is some redundancy between the *Position* and *Level* column. Therefor it would make sense to drop the *Position* column and just use the numeric *Level* and *Salary* columns. Since we only have 10 observations, it would not be useful to split the data into a training and test set.

```
# Saving the dataset with only the two necessary columns
positions <- positions[, 2:3]
```

Fitting Regressions to the Dataset

From the data it is not clear if we need to use simple linear or polynomial linear regression to best fit the data. Therefore, we will use both and then determine which one fits best.

Simple Regression

```
simple.regressor <- lm(formula = Salary ~ Level,
                       data = positions)
summary(simple.regressor)

##
## Call:
## lm(formula = Salary ~ Level, data = positions)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -170818 -129720  -40379   65856  386545
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  -195333     124790  -1.565  0.15615
## Level          80879      20112   4.021  0.00383 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 182700 on 8 degrees of freedom
## Multiple R-squared:  0.669,  Adjusted R-squared:  0.6277
## F-statistic: 16.17 on 1 and 8 DF,  p-value: 0.003833
```

It appears that the simple linear regression is actually not at all a bad model and we can see there is a close correlation between the variables and a p-value of 0.003833.

Polynomial

```
# Create a new column which contains the squares of the levels
positions$Level2 <- positions$Level ^ 2
positions$Level3 <- positions$Level ^ 3
positions$Level4 <- positions$Level ^ 4

# Polynomial Regressor
polynomial.regressor <- lm(formula = Salary ~ .,
                           data = positions)

summary(polynomial.regressor)

##
## Call:
## lm(formula = Salary ~ ., data = positions)
##
## Residuals:
##      1      2      3      4      5      6      7      8      9     10
```

```
##  -8357  18240   1358 -14633 -11725   6725  15997  10006 -28695  11084
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  184166.7    67768.0   2.718  0.04189 *
## Level       -211002.3    76382.2  -2.762  0.03972 *
## Level2        94765.4    26454.2   3.582  0.01584 *
## Level3       -15463.3     3535.0  -4.374  0.00719 **
## Level4          890.2      159.8   5.570  0.00257 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 20510 on 5 degrees of freedom
## Multiple R-squared:  0.9974, Adjusted R-squared:  0.9953
## F-statistic: 478.1 on 4 and 5 DF,  p-value: 1.213e-06
```

The polynomial linear regression is also a good model for the data. We can see there is a close correlation between the independent and dependent variables. The polynomial seems better than the simple because it has a lower p-value of .00001441.

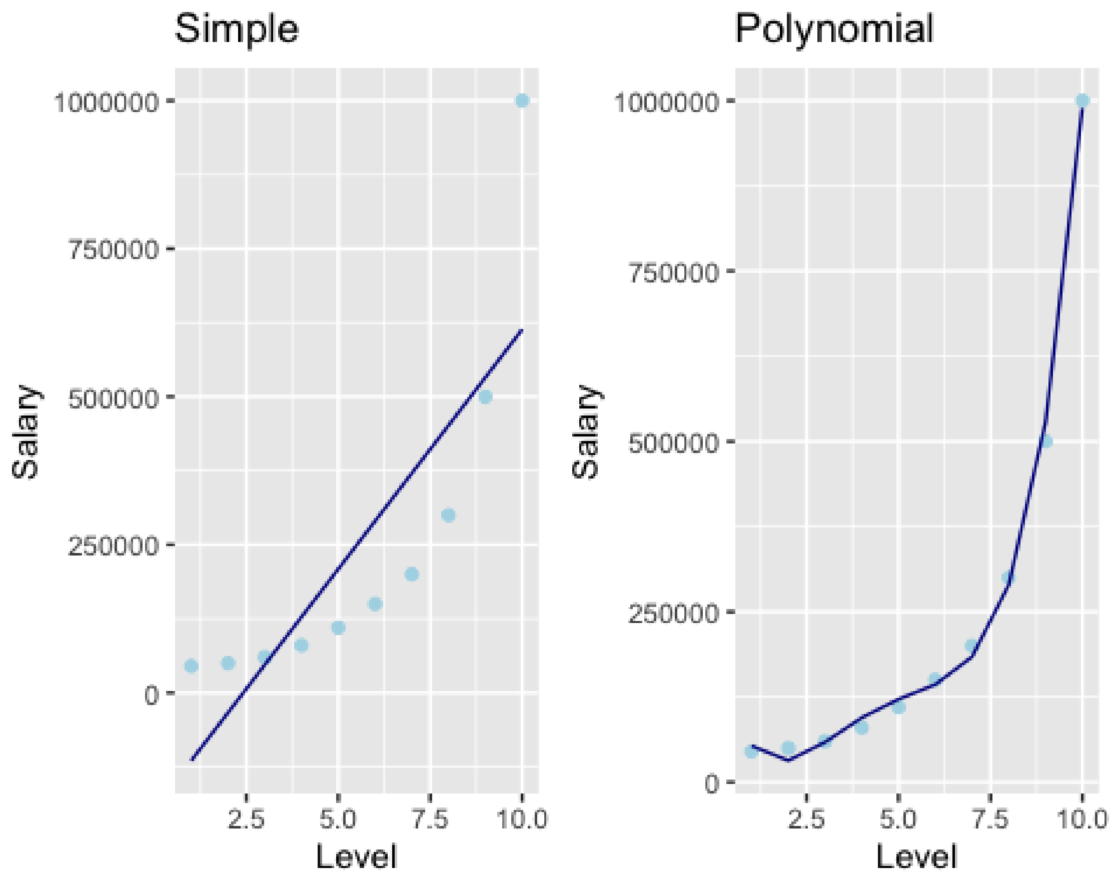
Visualizing the Regressions

```
par(mfrow = c(2,1))

# Visualizing the Simple Linear Regression
simple.plot <- ggplot() +
  geom_point(data = positions, aes(x = Level, y = Salary), col = 'lightblue')
+
  geom_line(aes(x = positions$Level, y = predict(simple.regressor, newdata =
positions)), col = 'darkblue') +
  ggtitle('Simple') +
  xlab('Level') +
  ylab('Salary')

# Visualizing the Polynomial Linear Regression
poly.plot <- ggplot() +
  geom_point(data = positions, aes(x = Level, y = Salary), col = 'lightblue')
+
  geom_line(aes(x = positions$Level, y = predict(polynomial.regressor,
newdata = positions)), col = 'darkblue') +
  ggtitle('Polynomial') +
  xlab('Level') +
  ylab('Salary')


grid.arrange(simple.plot, poly.plot, ncol = 2)
```



It is important to note that all of the light blue points are the actual data points while the dark blue line is our prediction. We can see that the polynomial regression does a much better job of predicting these points for this dataset.

Is is likely the new hire is telling the truth about his past salary?

The new hire stated he was earning a salary of $160,000 as a level 6.5.

```
poly.y.pred <- predict(polynomial.regressor, data.frame(Level = 6.5,
                                                         Level2 = 6.5^2,
                                                         Level3 = 6.5^3,
                                                         Level4 = 6.5^4))

poly.y.pred

##        1
## 158862.5
```

Conclusion

Our polynomial regressor predicted at a 6.5 level the salary would be $158,000. Therefore, it is likely the new hire was telling the truth about his salary since that number is very close to $160,000.