Regresión Polinómica

#Regresión Polinómica (R)

La regresión linear utiliza el método de mínimos cuadrados para encontrar la recta que resulta en la menor suma de errores al cuadrado (RMSE: Root Mean Square Error). La palabra simple se refiere a que la variable respuesta solo depende de 1 variable independiente: Y = f(X)

Escenario del problema

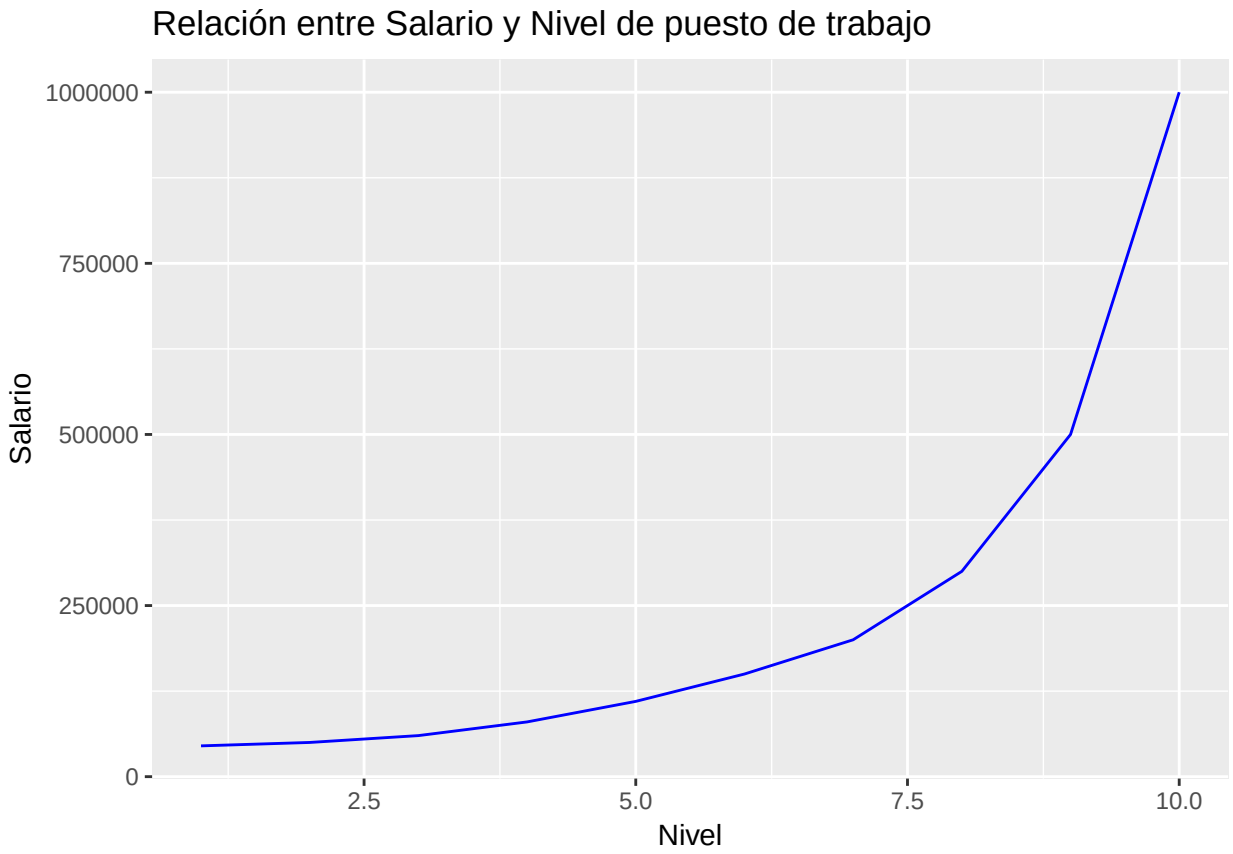
Vamos a contratar un nuevo empleado. Nos ha dicho que en su anterior empresa fue Manager Regional durante 2 años y que cobraba 170.000€ al año. Queremos determinar hasta que punto nos dice la verdad para poder negociar con él el salario que queremos ofrecerle en su nuevo puesto.
¡Vamos a ello!

```
# 1. Importar librerías
library(caTools)
library(ggplot2)
```

```
# 2. Importar datos
datos <- read.csv('../Datos/4.1.Salarios2.csv')
datas = datos[2:3] # Eliminamos la columna del título del puesto y nos quedamos con el nivel
head(datos, 10)
```

```
##            Posicion Nivel Salario
## 1          Analista     1   45000
## 2  Consultor Junior     2   50000
## 3  Consultor Senior     3   60000
## 4           Manager     4   80000
## 5   Manager General     5  110000
## 6  Manager Regional     6  150000
## 7            Socio     7  200000
## 8      Socio Senior     8  300000
## 9           Nivel-C     9  500000
## 10              CEO    10 1000000
```

```
ggplot() +
  geom_line(aes(x=datos$Nivel, y=datos$Salario), colour='blue') +
  ggtitle('Relación entre Salario y Nivel de puesto de trabajo') +
  xlab('Nivel') +
  ylab('Salario')
```



Vemos como NO existe una tendencia lineal

```
# 3. Separar en Entrenamiento y Validación
```

Recordatorio: no hacemos división de conjuntos porque tenemos muy pocos datos y nuestra intención es hacer una predicción lo más precisa posible.

```
# 4. Construir el Modelo
# 4.1 Tenemos que construir una nueva variable para cada grado de polinomio que queramos. Si por ejempl
datos$Nivel2 = datos$Nivel^2
datos$Nivel3 = datos$Nivel^3
datos$Nivel4 = datos$Nivel^4
head(datos, 5)
```

```
##           Posicion Nivel Salario Nivel2 Nivel3 Nivel4
## 1         Analista     1   45000      1      1      1
## 2 Consultor Junior     2   50000      4      8     16
## 3 Consultor Senior     3   60000      9     27     81
## 4          Manager     4   80000     16     64    256
## 5  Manager General     5  110000     25    125    625
```

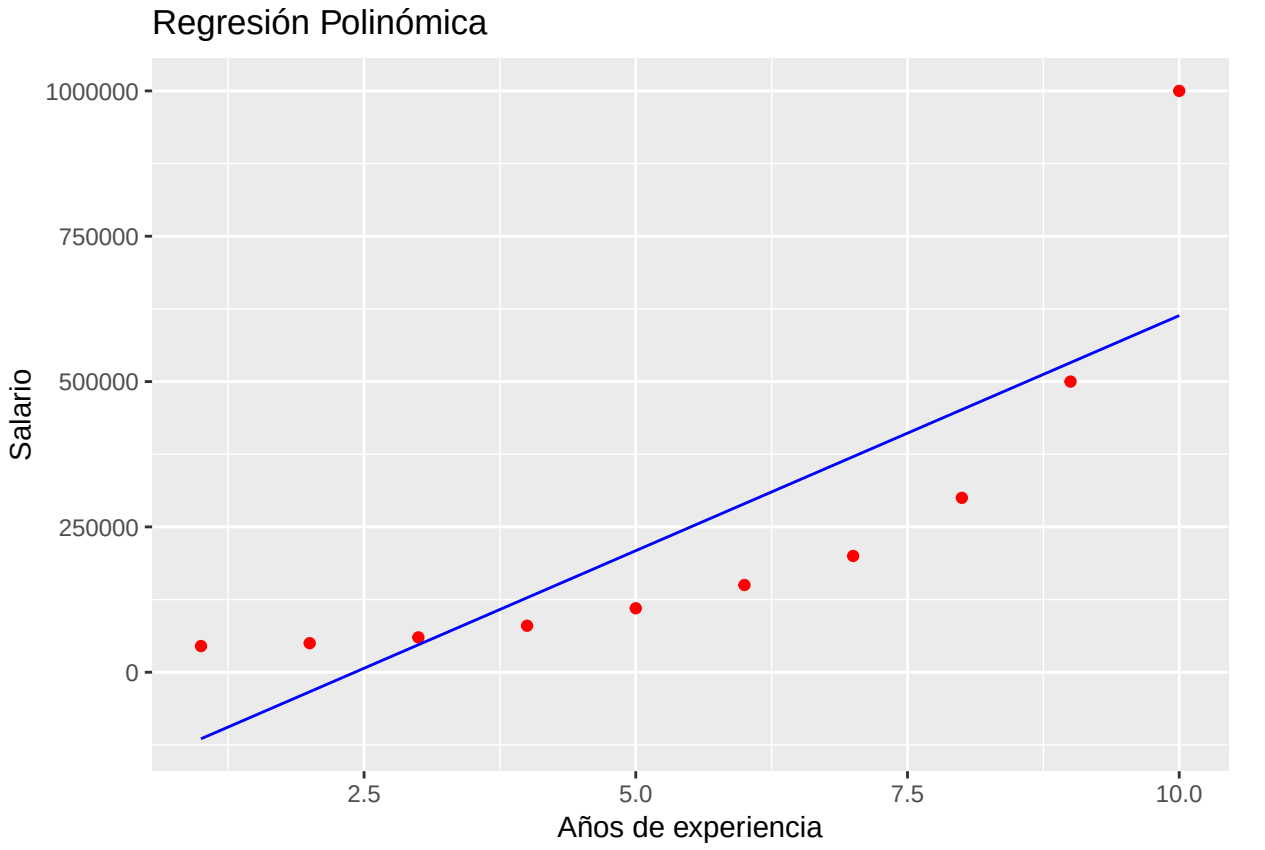
```
regresor_lineal <- lm(formula = Salario ~ Nivel, data = datos)
regresor_polino <- lm(formula = Salario ~ ., data = datos)
```

```
# 5. Hacer las prediciones para el conjunto de Validación
y_fit_lineal <- predict(regresor_lineal, newdata = datos)
y_fit_polino <- predict(regresor_polino, newdata = datos)
```

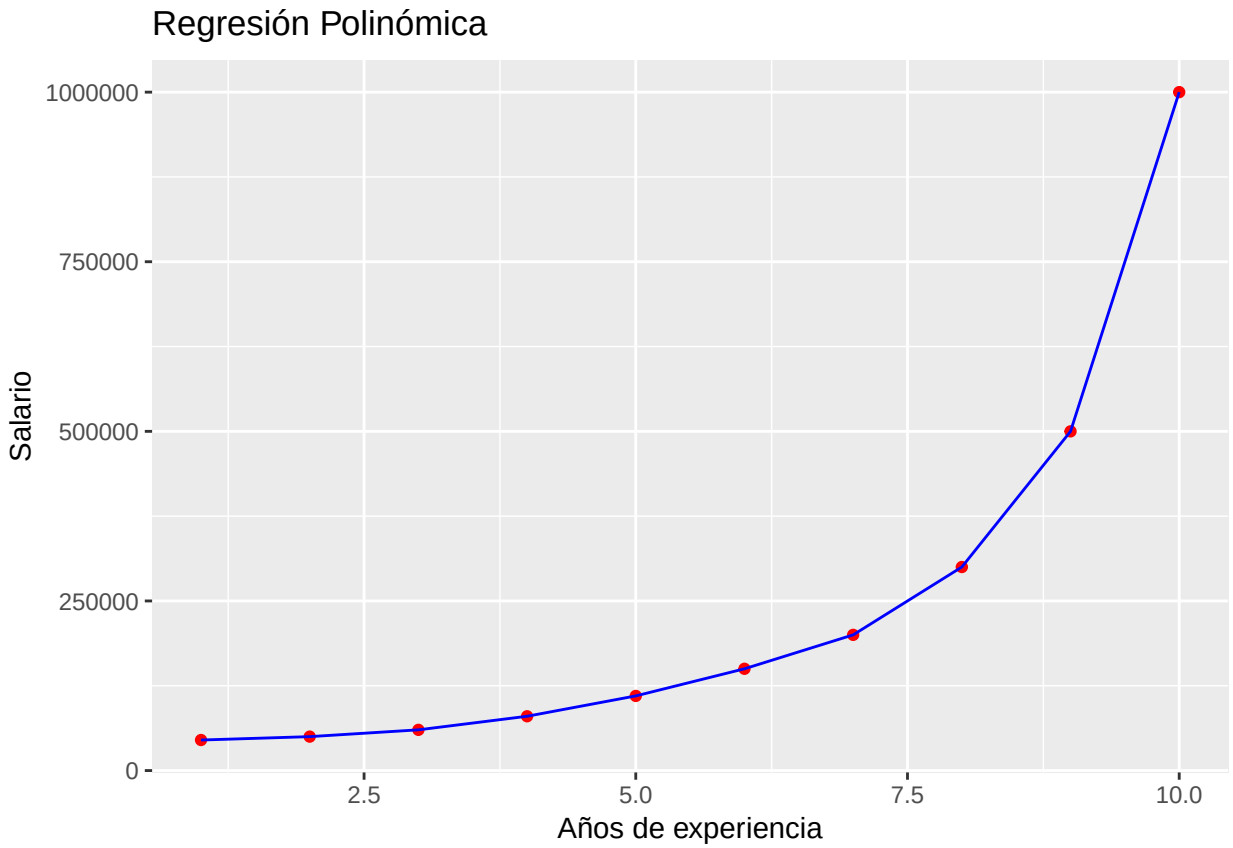
```
## Warning in predict.lm(regresor_polino, newdata = datos): prediction from a
## rank-deficient fit may be misleading
```

No estamos prediciendo en este ejemplo, sino determinando los parámetros para que el modelo se **ajuste** lo mejor posible a los datos del conjunto de entrenamiento (que constitutye todos los datos)

```
# 6. Echemos un vistazo a la pinta que tienen las predicciones
# 6.1. Para el conjunto de entrenamiento
ggplot() +
  geom_point(aes(datos$Nivel, datos$Salario), colour='red') +
  geom_line(aes(datos$Nivel, y_fit_lineal), colour='blue') +
  ggtitle('Regresión Polinómica') +
  xlab('Años de experiencia') +
  ylab('Salario')
```




```
# 6.2. Para el conjunto de validación
ggplot() +
  geom_point(aes(datos$Nivel, datos$Salario), colour='red') +
  geom_line(aes(datos$Nivel, y_fit_polino), colour='blue') +
  ggtitle('Regresión Polinómica') +
  xlab('Años de experiencia') +
  ylab('Salario')
```




```
# 7. Calcular el error
library(Metrics)
y_real <- datos$Salario
RMSE_lineal <- rmse(y_real, y_fit_lineal)
RMSE_polino <- rmse(y_real, y_fit_polino)
print(RMSE_lineal)
```

```
## [1] 163388.7
```

```
print(RMSE_polino)
```

```
## [1] 1.311894e-10
```