# Polynomial Regression

GR 5205 / GU 4205
Section 3

Columbia University
Xiaofei Shi

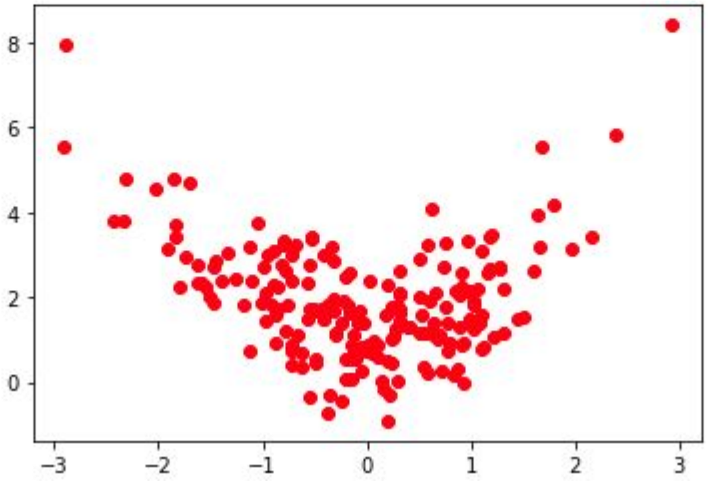
# So far...

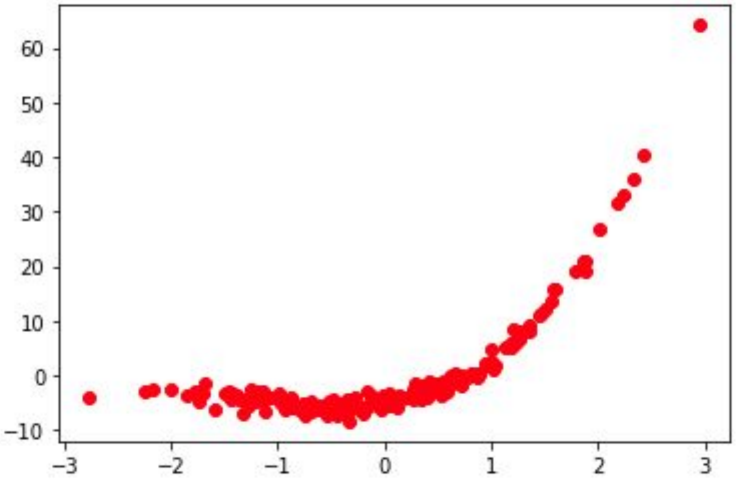
We predict a scalar random variable Y as a linear function of p-1 different predictor variables x with intercept, plus noise:

- Uncorrelated noise: unbiased estimator
- Gaussian noise: sampling distribution, hypothesis testing used in all packages
- All results are based on the linear relationship holds.

What if the ground truth is something different?

# Does the linear relationship holds?...




Solution: adding curvature!

# Adding Curvature: Polynomial Regression

- If the relationship between Y and X is non-linear, we could try to capture that fact with a polynomial. For example:

- Instead of Y being linearly related to X1, it's polynomially related, with the degree of the polynomial being d
- Treat $x_1^2, x_1^3, \ldots x_1^d$ as additional "predictors" and include them in the design matrix.
- Estimators are of the same form!

# Potential Problem?

# Realization: polynomial degree = 2

$$a_0 + a_1 x + a_2 x^2$$

- In R:

```
out = lm(y ~ poly(x,2))
```

- In Python:

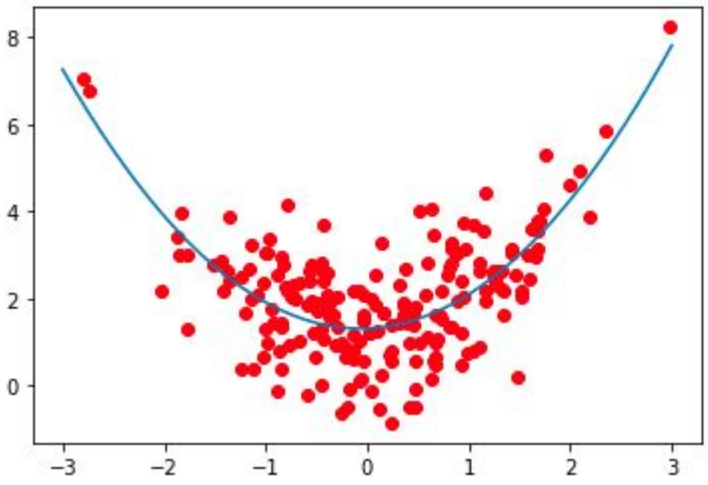
```
x = np.append(x,(x[:,1]**2).reshape(-1,1), 1)
x = statsmodels.tools.tools.add_constant(x)
model = sm.OLS(y, x).fit()
```

Or

```
model = np.poly1d(np.polyfit(x, y, 2))
```

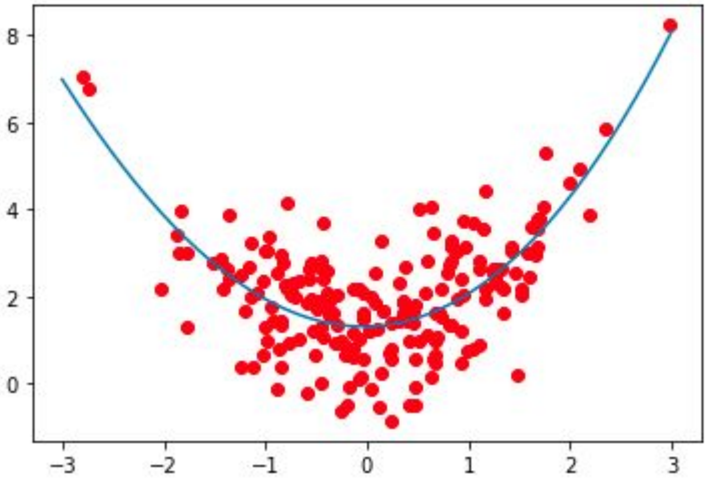
# Using a polynomial with degree = 2

```
                            OLS Regression Results
==============================================================================
Dep. Variable:                      y   R-squared:                       0.487
Model:                            OLS   Adj. R-squared:                  0.482
Method:                 Least Squares   F-statistic:                     93.69
Date:                Sat, 17 Oct 2020   Prob (F-statistic):           2.54e-29
Time:                        18:15:47   Log-Likelihood:                -279.21
No. Observations:                 200   AIC:                             564.4
Df Residuals:                     197   BIC:                             574.3
Df Model:                           2
Covariance Type:            nonrobust
==============================================================================
                 coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------
const          1.3041      0.087     15.006      0.000       1.133       1.475
x1             0.0921      0.070      1.316      0.190      -0.046       0.230
x2             0.6920      0.052     13.397      0.000       0.590       0.794
==============================================================================
Omnibus:                        0.242   Durbin-Watson:                   1.895
Prob(Omnibus):                  0.886   Jarque-Bera (JB):                0.184
Skew:                          -0.074   Prob(JB):                        0.912
Kurtosis:                       2.992   Cond. No.                         2.45
==============================================================================
```



# Using a polynomial with order = 3

```
                            OLS Regression Results
==============================================================================
Dep. Variable:                      y   R-squared:                       0.488
Model:                            OLS   Adj. R-squared:                  0.480
Method:                 Least Squares   F-statistic:                     62.31
Date:                Sat, 17 Oct 2020   Prob (F-statistic):           2.46e-28
Time:                        18:18:04   Log-Likelihood:                -279.08
No. Observations:                 200   AIC:                             566.2
Df Residuals:                     196   BIC:                             579.4
Df Model:                           3
Covariance Type:            nonrobust
==============================================================================
                 coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------
const          1.3042      0.087     14.979      0.000       1.132       1.476
x1             0.0497      0.110      0.453      0.651      -0.167       0.266
x2             0.6923      0.052     13.376      0.000       0.590       0.794
x3             0.0150      0.030      0.503      0.616      -0.044       0.074
==============================================================================
Omnibus:                        0.214   Durbin-Watson:                   1.900
Prob(Omnibus):                  0.898   Jarque-Bera (JB):                0.148
Skew:                          -0.067   Prob(JB):                        0.929
Kurtosis:                       2.999   Cond. No.                         6.07
==============================================================================
```



# How to choose the polynomials?

- Smoothness:

  Polynomials are very smooth, meaning th

  they and all their derivatives exist and ar

  continuous.

  Desirable if you are looking for a smooth

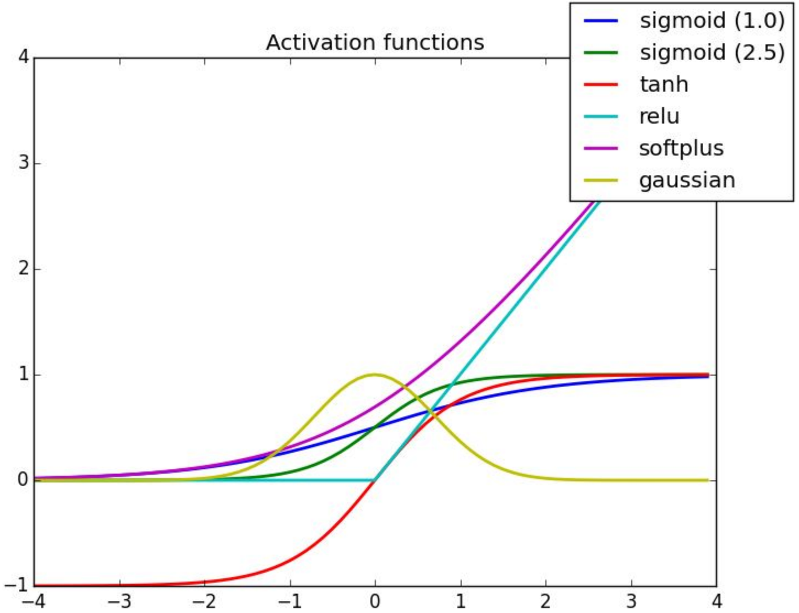
  dependence, not if there are sharp

  threshold or jumps.

  Notice that one **can** approximate thresho

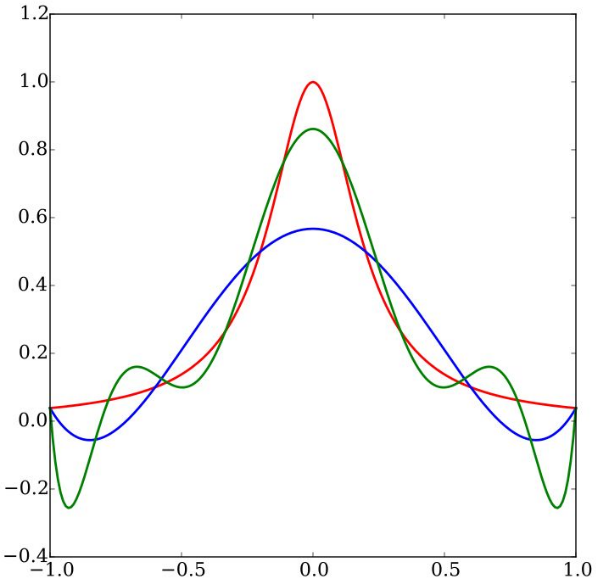
  as accurate as one wants to, but ending u

  with very high order polynomials.



# How to choose the polynomials?

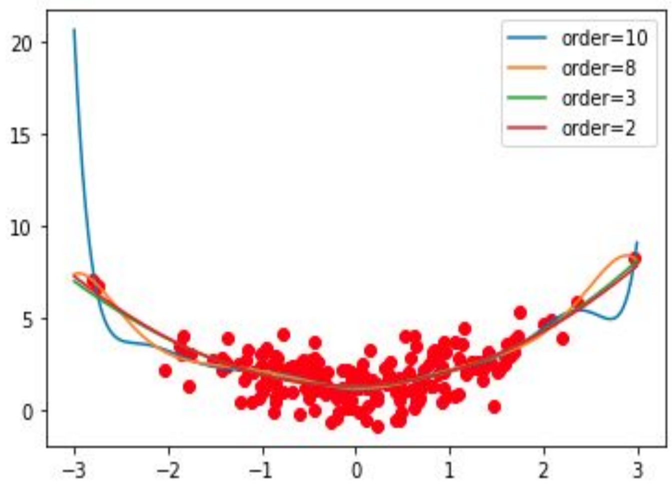
- Overfitting:
  A polynomial with degree d can fit any d+1 points. Using a high-order polynomial, or even summing a large number of low-order polynomials, can therefore lead to curves which come very close to the data we used to estimate them, but predict very badly.




Runge’s Phenomenon

# How to choose the polynomials?

- Picking the polynomial order:
  - scientific theory
  - carefully examining the diagnostics plots
  - variable and model selections



# Other choices: Orthogonal Polynomials

Suppose that $x \in [-1, 1]$

- $f_0(x) = 1, \quad f_1(x) = x, \quad f_2(x) = x^2, \quad f_3(x) = x^3, \ldots$

- Legendre polynomials:

  $g_0(x) = 1, \quad g_1(x) = x, \quad g_2(x) = \frac{1}{2}(3x^2 - 1), \quad g_3(x) = \frac{1}{2}(5x^3 x), \ldots$

  - gives the same results as the former simple polynomials;
  - least squares optimization results are more stable and the standard errors of the coefficients are smaller

# Other choices: beyond polynomials

- We are treating different powers of X as new features, and we can of course treat different functions of X as new features as well.
  - Fourier family: sines and cosines
  - ReLU and other activation functions
- Choose the functions:
  - scientific theory
  - carefully examining the diagnostics plots
  - variable and model selections

# More to think about...

- Global trend v.s. local accuracy: a trade-off
- Piecewise polynomials, i.e. splines, are widely used in interpolation and fitting to avoid Runge's phenomenon, but more parameters need to be estimated.
- Outliers