LECTURE 14: CURVE FITTING II

STAT 131A

OCT 14 2025

LINEAR REGRESSION: MODEL PARAMETERS

- Ultimately we build a linear regression model by working off a **sample** of the entire **population** on data points (X,Y)
- Thus the model is something that is estimated
  - And so are the parameters: β_0 and β_1

HYPOTHESIS FRAMEWORK: IS THERE LINEARITY IN THE RELATIONSHIP ?

- We test whether $\beta_1 = 0$
  - $H_0: \beta_1 = 0$
- Why ?

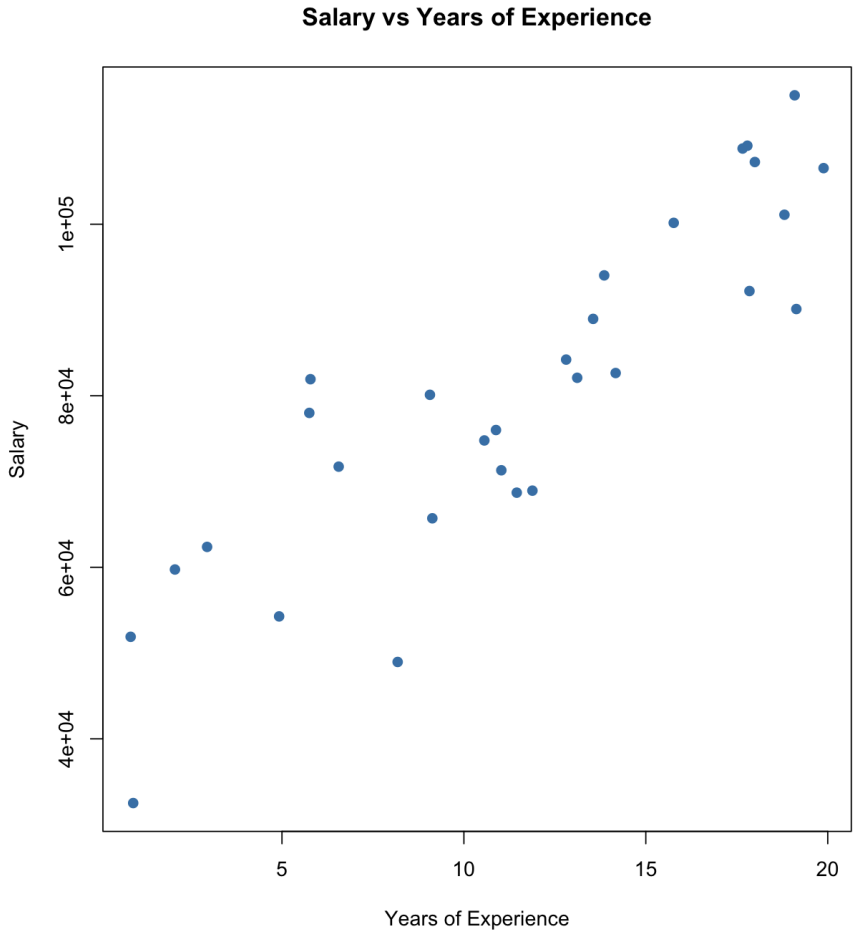
CONFIDENCE INTERVALS

- So, how **confident** are we about $\hat{\beta}_0$ and $\hat{\beta}_1$?
- Recap boostrapping confidence intervals for parameters, when comparing two groups
  - We had two groups: $X_1, \ldots, X_{n_1}$ and $Y_1, \ldots, Y_{n_2}$
  - We resampled the data within each group to get: $X_1^*, \ldots, X_{n_1}^*$ and $Y_1^*, \ldots, Y_{n_2}^*$
  - And then estimated our stat : $\hat{\delta}^*$

BOOSTRAPPING

- In constructing a (linear regression) model we have N pairs of (x_i, y_i)
- We resample N times to get: $(x_1^*, y_1^*), \ldots, (x_N^*, y_N^*)$
- Now we run regression on $(x_1^*, y_1^*), \ldots, (x_N^*, y_N^*)$ and get $\hat{\beta}_1^*$ and $\hat{\beta}_0^*$
- We repeat this B times, to get:
  - $(\hat{\beta}_0^{(1)*}, \hat{\beta}_1^{(1)*}), \ldots, (\hat{\beta}_0^{(B)*}, \hat{\beta}_1^{(B)*})$
- We have a distribution !
- Calculate confidence intervals from the percentiles of these values

R EXAMPLE (BOOTSTRAPPED CI)



	lower	estimate	upper
(Intercept)	36089.604	45566.387	54280.757
years_exp	2394.733	3035.037	3697.124

CI: PARAMETRIC MODEL APPROACH

- We can also consider a **parameteric** model approach to estimating the confidence intervals for the regression model parameters
- Why are there two p-values ?

```
> summary(fit)

Call:
lm(formula = salary ~ years_exp, data = data)

Residuals:
   Min     1Q Median     3Q    Max
-29316  -6466   1886   6908  16459

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  29655.7     3091.6   9.592 9.72e-13 ***
years_exp     2480.3      230.7  10.751 2.24e-14 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 9807 on 48 degrees of freedom
Multiple R-squared:  0.7066,    Adjusted R-squared:  0.7005
F-statistic: 115.6 on 1 and 48 DF,  p-value: 2.244e-14
```

PARAMETRIC APPROACH

- From the line of fit we have (assume):
  - $y = \beta_0 + \beta_1 x + e$.
- We will need to assume a probability distribution for the errors e
  - Nothing random about Y !
- $e \sim N(0, \sigma^2)$
  - Normal with **same** (as y) (unknown) variance σ^2
- $e_1, \ldots, e_n$ are independent
- $y_i \,|\, x_i \sim N(\beta_0 + \beta_1 x_i, \sigma^2)$

DISTRIBUTION

- Since each y_i is normally distributed, it follows that
  - $\hat{\beta}_1 \sim N(\beta_1, \nu_1^2)$
  - where
    - $\nu_1^2 = var(\hat{\beta}_1) = \dfrac{\sigma^2}{\sum_{i=1}^{n} (x_i - \bar{x})^2}$

VARIANCE AND T-STAT

- $\hat{\nu}_1^2 = \hat{var}(\hat{\beta}_1) = \dfrac{\hat{\sigma}^2}{\sum_{i=1}^{n}(x_i - \bar{x})^2}$
- $T_1 = \dfrac{\hat{\beta}_1}{\sqrt{\hat{var}(\hat{\beta}_1)}}$

```
> summary(fit)

Call:
lm(formula = salary ~ years_exp, data = data)

Residuals:
   Min     1Q Median     3Q    Max
-29316  -6466   1886   6908  16459

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  29655.7     3091.6   9.592 9.72e-13 ***
years_exp     2480.3      230.7  10.751 2.24e-14 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 9807 on 48 degrees of freedom
Multiple R-squared:  0.7066,    Adjusted R-squared:  0.7005
F-statistic: 115.6 on 1 and 48 DF,  p-value: 2.244e-14
```

CONFIDENCE INTERVAL

- $\hat{\beta}_1 \pm 1.96\hat{\nu}_1$

```
> confint(fit)
                 2.5 %     97.5 %
(Intercept) 23439.500 35871.804
years_exp    2016.412   2944.122
```

ESTIMATING σ^2

- Would be : $\frac{1}{n-1}\sum(e_i - \bar{e})^2$
- $r_i = y_i - (\hat{\beta}_0 + \hat{\beta}_1 x_i)$
- The r_i are called **residuals**
- $r_i \neq e_i$!!

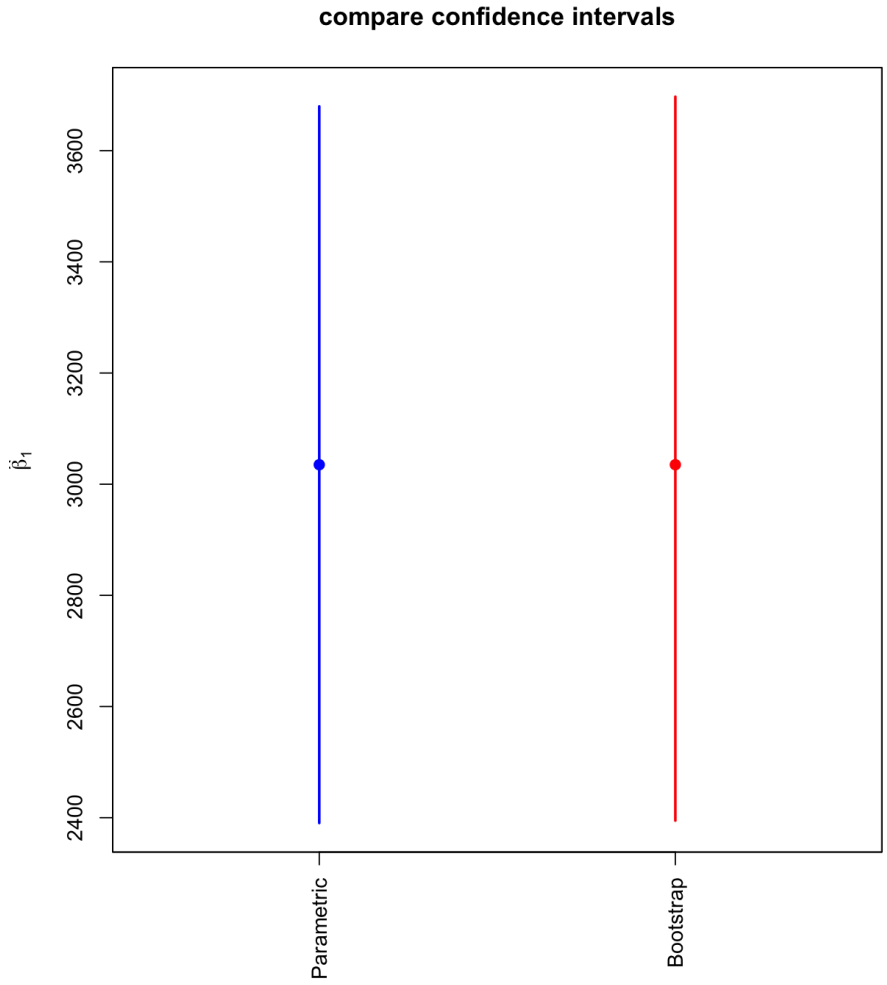
σ^2

- $\frac{1}{n-1}\sum (r_i - \bar{r})^2$
- $\frac{1}{n-1}\sum r_i^2$
- $\hat{\sigma}^2 = \frac{1}{n-2}\sum_i r_i^2 .$

ASSUMPTIONS

- The parametric linear model makes the following assumptions
  - Errors are independent
  - Errors are i.i.d, meaning they have the same variance
  - Errors are normally distributed
- The bootstrap makes the same kind of assumptions as in the two group comparisons:
  - The i.i.d resampling of the bootstrapped data mirrors how the actual data was generated (i.e. the actual data is i.i.d)
  - The sample size is large enough that the sample distribution is close to the real distribution.
  - The test statistic is well behaved (e.g. unbiased)
    - this *is* true for regression

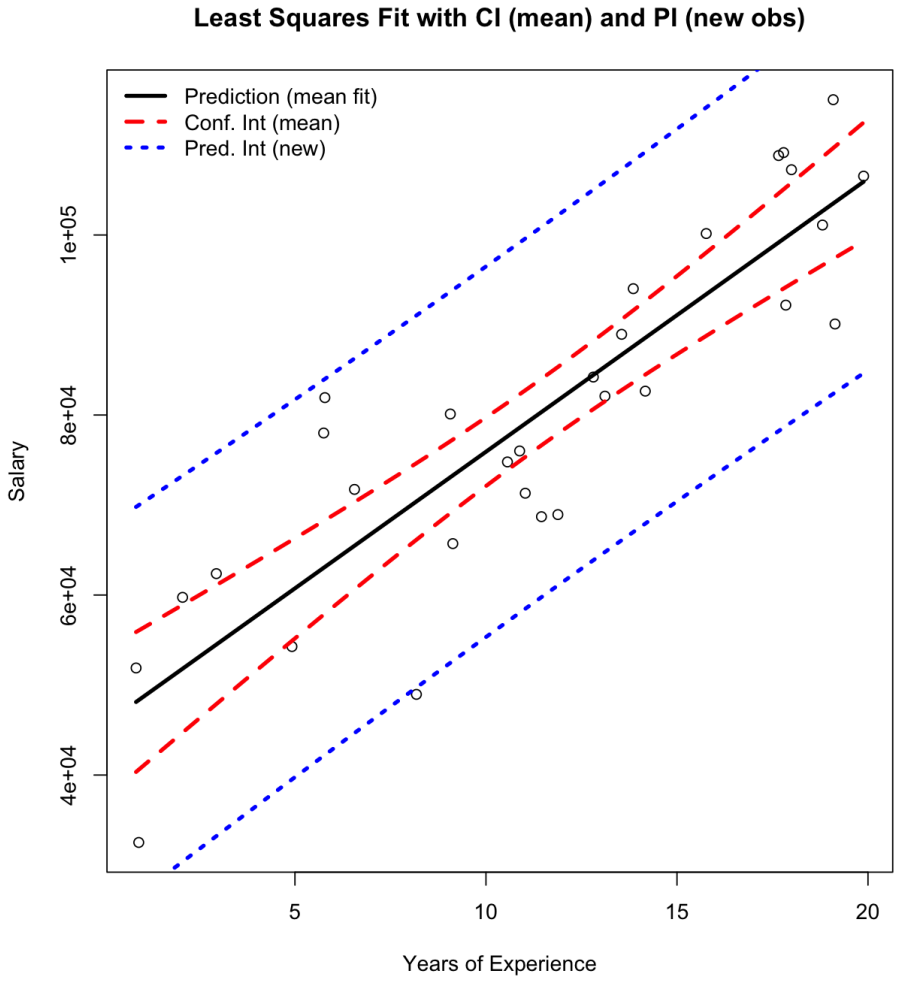
CI: BOTH METHODS



PREDICTION INTERVALS, AND CONFIDENCE INTERVALS

- Two different questions
  - A **confidence interval** asks: “What’s the likely average salary for people with this many years of experience?”
  - A **prediction interval** asks: “If I pick one individual with this many years of experience, what range might their actual salary fall in?”
- Confidence interval → about the mean
  - It reflects uncertainty in estimating the regression line itself: the mean trend across all people.
  - If you collected new data many times, the *average fitted line* would wobble slightly
    - the CI shows that uncertainty
- Prediction interval → about individual outcomes
  - Even if the line is known perfectly (how ?) , **individual salaries scatter widely** around it because of real-world variation : bonuses, industries, education, etc.
  - It answers: “For one new employee with 10 years of experience, what salary range is plausible?”

R EXAMPLE

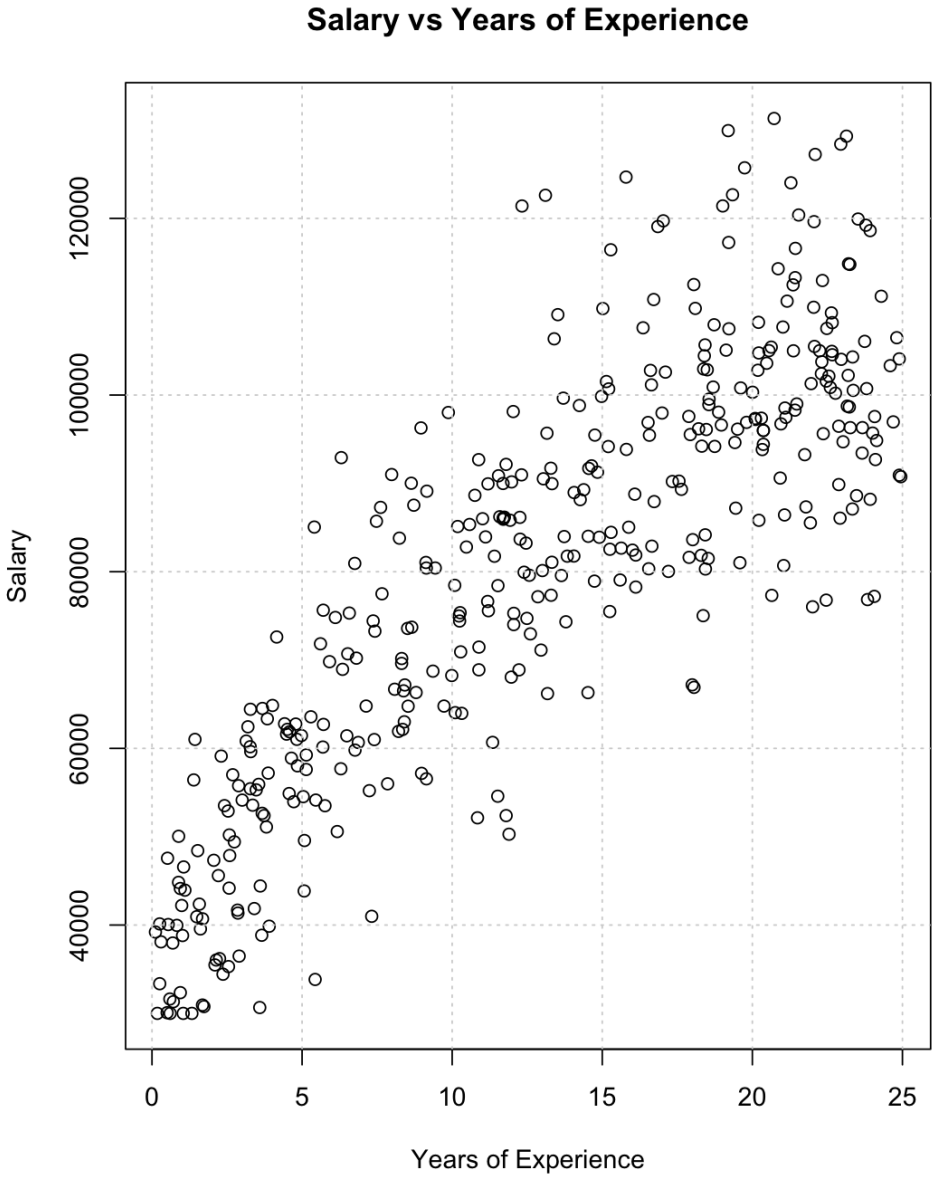


- The **red dashed band** around the line of best fit represents how much the *average salary* estimate could vary at each experience level.
- The **blue dotted band** adds this personal variability on top of the line's uncertainty.

SUMMARY

- The goal is to quantify the uncertainty in the regression slope for Y ~ X.
- Approach 1: Bootstrap CIs: use resampling to approximate the sampling distribution of β_1
- Approach 2: Parametric (t-based) CIs: assume a Normal-error model and use theory for β_1
- Both aim to produce a 95% interval for β_1 but rely on different assumptions and computations

POLYNOMIAL REGRESSION

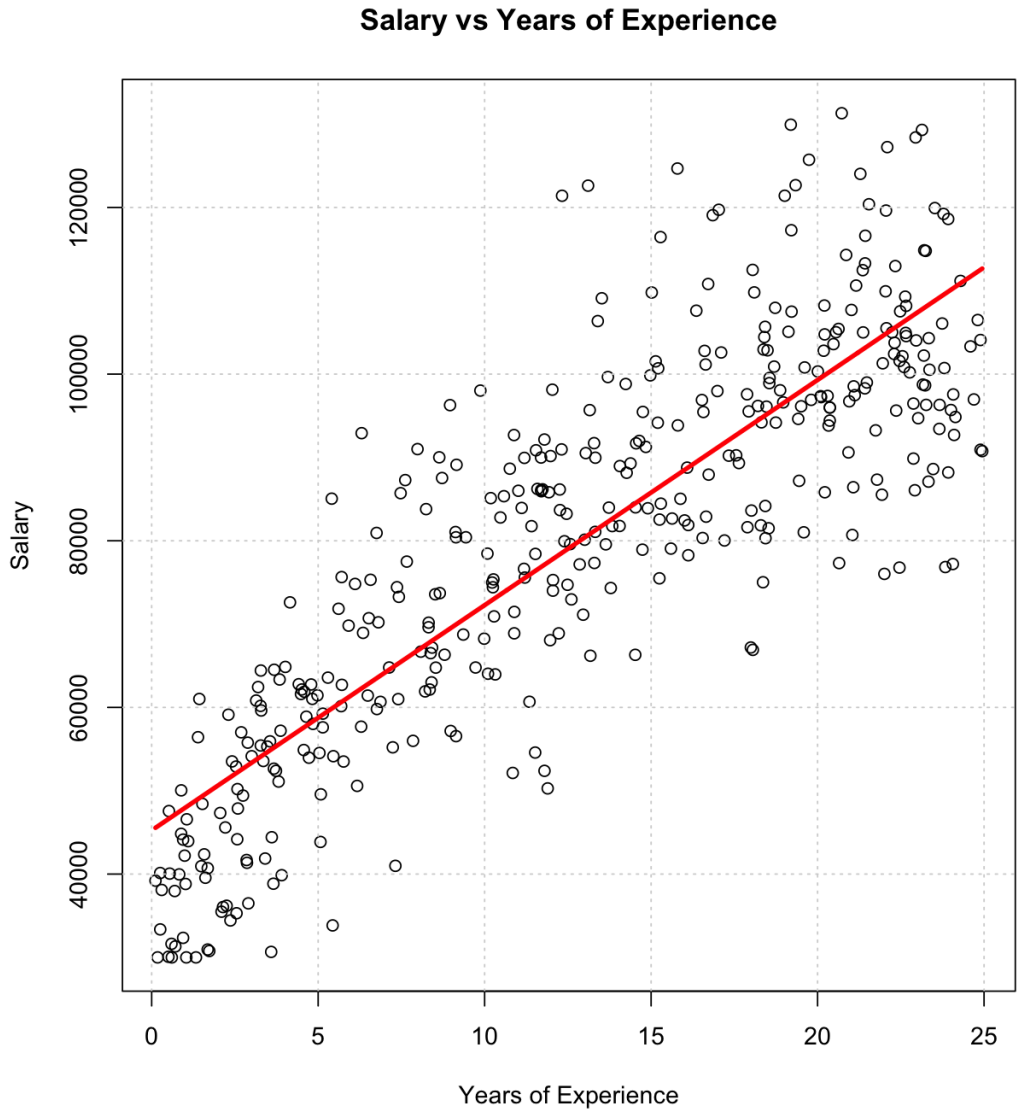

Salary vs Years of Experience
Salary
40000
60000
80000
100000
120000
0
5
10
15
20
25
Years of Experience

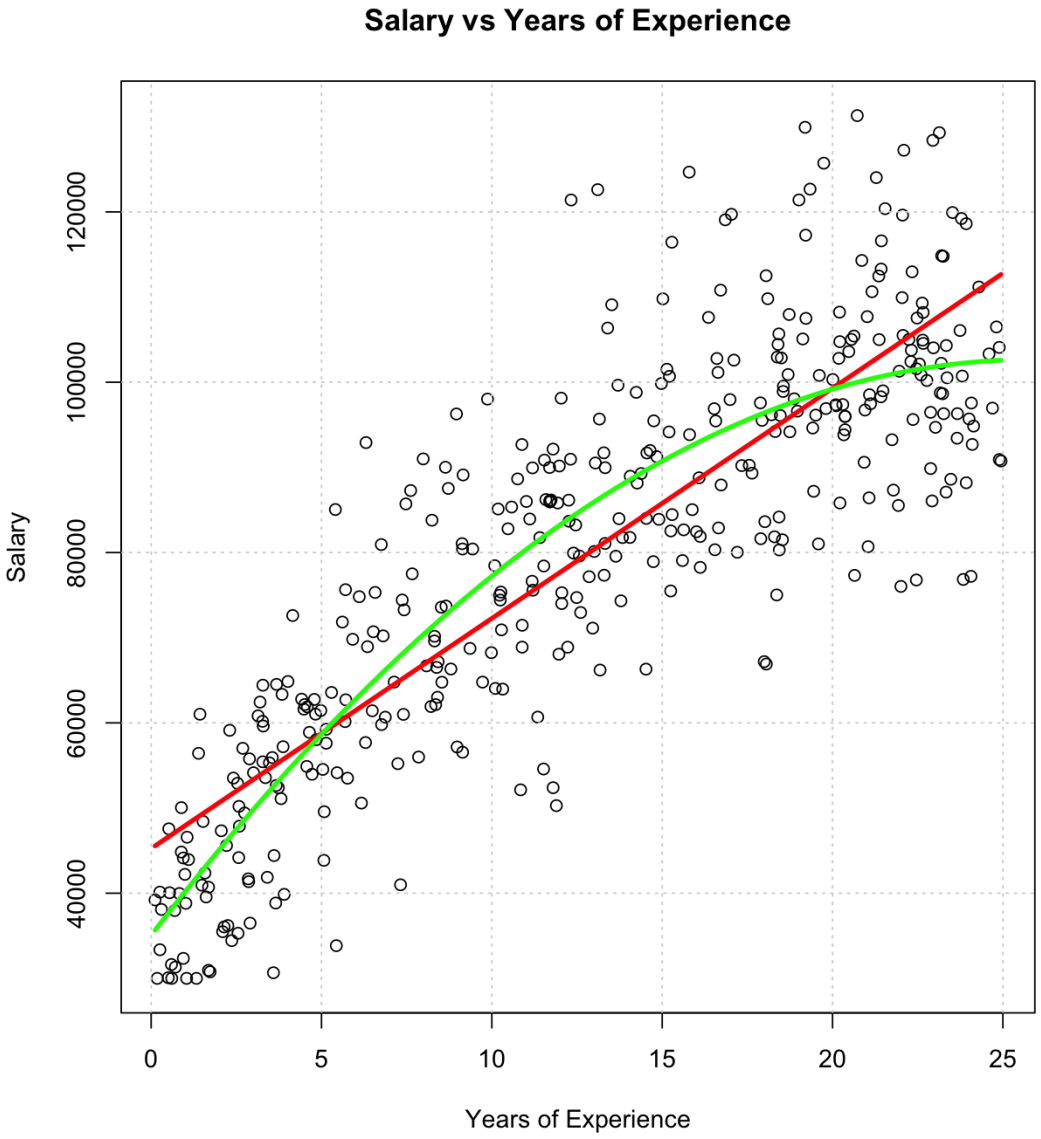
POLYNOMIAL REGRESSION

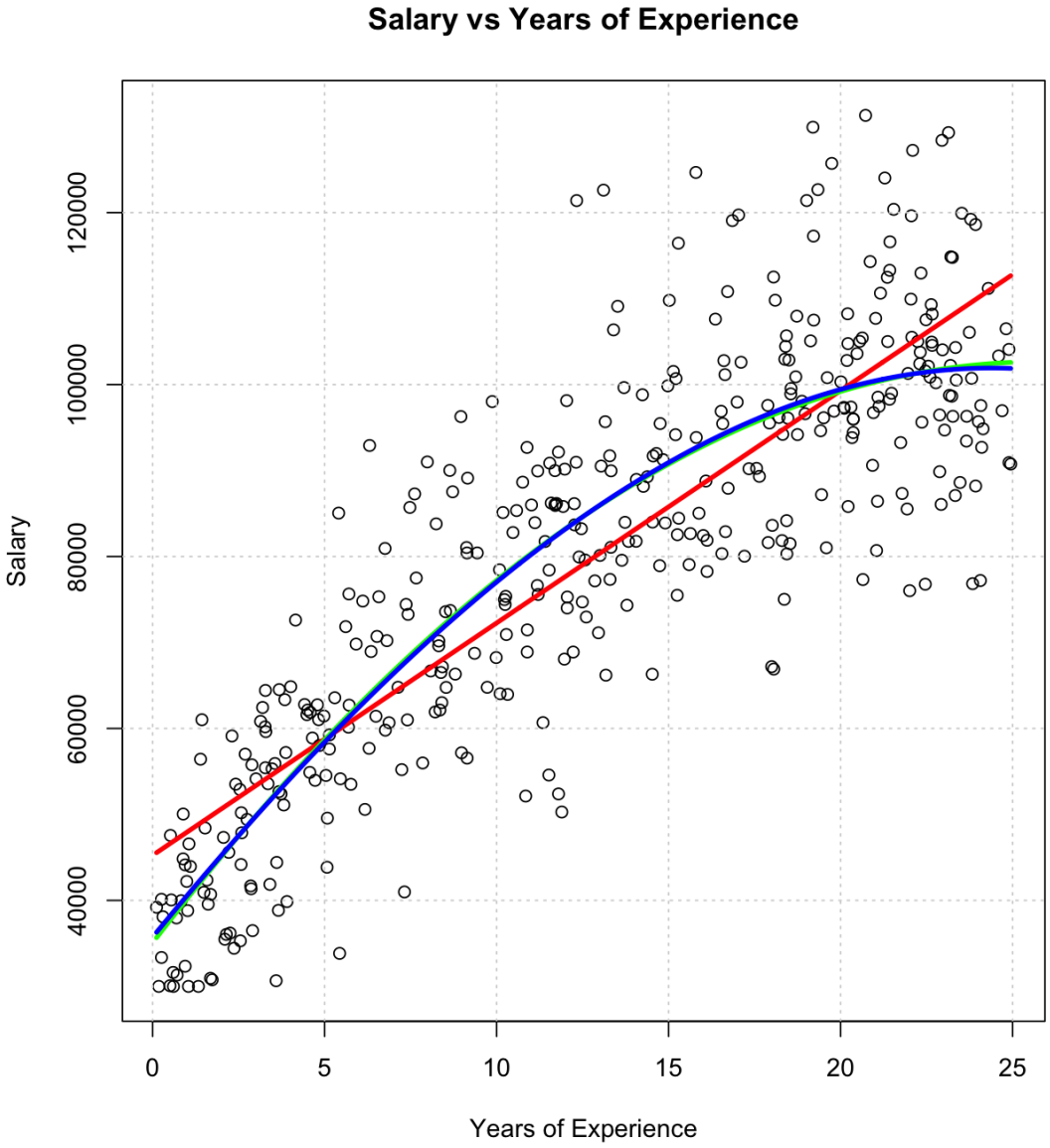
- We could a quadratic function:
  - $y = \beta_0 + \beta_1 x + \beta_2 x^2 + e$
- Determine optimal coefficents:
  - $\hat{y}_i(\beta_0, \beta_1, \beta_2) = \beta_0 + \beta_1 x_i + \beta_2 x_i^2,$
- Error:
  - $\ell(y_i, \hat{y}_i(\beta_0, \beta_1, \beta_2))$
- Apply least squares
  - $\frac{1}{n} \sum_{i=1}^{n} (y_i - \beta_0 - \beta_1 x_i - \beta_2 x_i^2)^2$

SOLVING FOR POLYNOMIALS

- Polynomial regression is still *linear in parameters*: we just include X^2, X^3, … as extra columns in X.
- The goal is the same: find coefficients β_0, β_1, β_2, … that minimize the sum of squared errors.
- Matrix form: $Y = X\beta + \varepsilon$ where $X = [1, X, X^2, \ldots]$.
- Least-squares solution: $\hat{\beta} = (X^T X)^{-1} X^T Y$.
- So polynomial regression simply solves a **larger linear system** using the same principle as simple linear regression.


Salary vs Years of Experience
Salary
40000
60000
80000
100000
120000
0
5
10
15
20
25
Years of Experience


Salary vs Years of Experience
Salary
40000
60000
80000
100000
120000
0
5
10
15
20
25
Years of Experience


Salary vs Years of Experience
Salary
40000
60000
80000
100000
120000
0
5
10
15
20
25
Years of Experience

INTERPRETATION OF POLYNOMIAL TERMS

- β_1, β_2,… describe curvature, not direct linear effects
- Each higher-order term refines fit near extremes of X
- Interpretation focuses on overall curve shape, not single coefficient

SELECTING POLYNOMIAL DEGREE

- Higher degrees increase flexibility but risk overfitting.
- Choose degree using residual plots or cross-validation.
- Underfitting: trend missed; Overfitting: noise captured.

ORTHOGONAL POLYNOMIALS IN R

- In polynomial regression, the design matrix X has columns [1, x, x^2, x^3, …]
- These columns are **highly correlated**, which makes X^TX hard to invert (numerically unstable)
- `poly(x, degree)` creates a new set of **orthogonal columns:** mathematically independent directions in X
- This orthogonal basis makes the matrix calculations more stable and avoids large rounding errors.
- The fitted curve $\hat{\mathbf{y}} = \mathbf{X}\hat{\boldsymbol{\beta}}$ stays identical
  - only the internal representation of X changes.
- So `poly()` doesn't change the model's shape
  - The matrix algebra (operations) get easier to do