LECTURE 16: CURVE FITTING IV

ALGEBRA OF SOLVING FOR POLYNOMIAL REGRESSION, LOESS (MATRICES) MULTIPLE LINEAR REGRESSION: START OF

OCT 21 2025

MATRICES: ELEMENTARY OPERATIONS

- Multiplication
  - Defined when the **number of columns in the first matrix equals the number of rows in the second**
  - Each entry is the **dot product** of the corresponding row and column
  - Note that BA ≠ AB, in general !

$$A = \begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix}, \quad B = \begin{bmatrix} 5 & 6 \\ 7 & 8 \end{bmatrix}$$

$$AB = \begin{bmatrix} (1 \times 5 + 2 \times 7) & (1 \times 6 + 2 \times 8) \\ (3 \times 5 + 4 \times 7) & (3 \times 6 + 4 \times 8) \end{bmatrix} = \begin{bmatrix} 19 & 22 \\ 43 & 50 \end{bmatrix}$$

MATRICES: ELEMENTARY OPERATIONS

- Transpose: **flip** rows and columns

$$A = \begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \end{bmatrix} \quad \Rightarrow \quad A^T = \begin{bmatrix} 1 & 4 \\ 2 & 5 \\ 3 & 6 \end{bmatrix}$$

- The **Identity** matrix

$$I_3 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

MATRICES: ELEMENTARY OPERATIONS

- Inverse
- Only "invertible" matrices have an inverse
  - **Not all matrices have an inverse**

$$AA^{-1} = A^{-1}A = I$$

BACK TO WHAT WAS PROVIDED FOR PLR

- Solving for β gives $\hat{\beta} = (X^T X)^{-1} X^T Y$
- $(X^T X)^{-1} X^T$: the "pseudo inverse"
  - An approximate ("best we can achieve") inverse
  - As, not all matrices have an exact inverse !

SIMPLE LINEAR REGRESSION

$$X = \begin{bmatrix} 1 & 2 \\ 1 & 3 \\ 1 & 4 \end{bmatrix}, \quad Y = \begin{bmatrix} 3 \\ 5 \\ 7 \end{bmatrix}$$

Compute step by step:

$$X^T X = \begin{bmatrix} 3 & 9 \\ 9 & 29 \end{bmatrix}, \quad X^T Y = \begin{bmatrix} 15 \\ 49 \end{bmatrix}$$

$$(X^T X)^{-1} = \frac{1}{3 \times 29 - 9 \times 9} \begin{bmatrix} 29 & -9 \\ -9 & 3 \end{bmatrix} = \frac{1}{6} \begin{bmatrix} 29 & -9 \\ -9 & 3 \end{bmatrix}$$

Then

$$\hat{\beta} = \boxed{(X^T X)^{-1} X^T} Y = \frac{1}{6} \begin{bmatrix} 29 & -9 \\ -9 & 3 \end{bmatrix} \begin{bmatrix} 15 \\ 49 \end{bmatrix} = \begin{bmatrix} -1 \\ 2 \end{bmatrix}$$

The equivalent of "*1/X*" (division by matrix *X*)

So the fitted model is:

$$\hat{y} = -1 + 2x$$

FITTED

$$X = \begin{bmatrix} 1 & 2 \\ 1 & 3 \\ 1 & 4 \end{bmatrix}, \quad Y = \begin{bmatrix} 3 \\ 5 \\ 7 \end{bmatrix}$$

$$\hat{Y} = \begin{bmatrix} 1 & 2 \\ 1 & 3 \\ 1 & 4 \end{bmatrix} \begin{bmatrix} -1 \\ 2 \end{bmatrix}$$

FITTED

$$X = \begin{bmatrix} 1 & 2 \\ 1 & 3 \\ 1 & 4 \end{bmatrix}, \quad Y = \begin{bmatrix} 3 \\ 5 \\ 7 \end{bmatrix}$$

$$\hat{Y} = \begin{bmatrix} 1 & 2 \\ 1 & 3 \\ 1 & 4 \end{bmatrix} \begin{bmatrix} -1 \\ 2 \end{bmatrix} = \begin{bmatrix} 3 \\ 5 \\ 7 \end{bmatrix}$$

- What are the residuals ?
- What should they sum to (always) ?

FITTED VALUES: ANOTHER EXAMPLE

$$X = \begin{bmatrix} 1 & 2 \\ 1 & 3 \\ 1 & 4 \end{bmatrix}, \quad Y = \begin{bmatrix} -5 \\ -5 \\ 1 \end{bmatrix}$$

$$\hat{Y} = \begin{bmatrix} 1 & 2 \\ 1 & 3 \\ 1 & 4 \end{bmatrix} \begin{bmatrix} -12 \\ 3 \end{bmatrix} = \begin{bmatrix} -6 \\ -3 \\ 0 \end{bmatrix}$$

FITTED VALUES: ANOTHER EXAMPLE

$$X = \begin{bmatrix} 1 & 2 \\ 1 & 3 \\ 1 & 4 \end{bmatrix}, \quad Y = \begin{bmatrix} -5 \\ -5 \\ 1 \end{bmatrix}$$

$$\hat{Y} = \begin{bmatrix} 1 & 2 \\ 1 & 3 \\ 1 & 4 \end{bmatrix} \begin{bmatrix} -12 \\ 3 \end{bmatrix} = \begin{bmatrix} -6 \\ -3 \\ 0 \end{bmatrix}$$

$$\begin{bmatrix} 1 \\ -2 \\ 1 \end{bmatrix}$$

POLYNOMIAL REGRESSION

- In polynomial regression, the design matrix X has columns $[1, x, x^2, x^3, \ldots]$
- **Model setup**
  - Polynomial regression model: $Y = X\beta + \varepsilon$, where $X = [1, x, x^2, \ldots, x^d]$.
    - The goal is to find the coefficients β that minimize total squared error.
- **Define the loss function**
  - We measure total squared residuals using $L(\beta) = (Y - X\beta)^T (Y - X\beta)$.
    - This is like adding up all squared differences between the observed and predicted values.
- **Expand the expression**
  - When expanded, the loss becomes $L = Y^T Y - 2\beta^T X^T Y + \beta^T X^T X\beta$.
    - Each term reflects a different interaction between data, model, and parameters

ORDINARY LEAST SQUARES: OLS

Step	Simple Linear Regression (SLR)	Polynomial Linear Regression (PLR)
Design matrix X	$X = \begin{bmatrix} 1 & x_1 \\ \vdots & \vdots \\ 1 & x_n \end{bmatrix}$	$X = \begin{bmatrix} 1 & x_1 & {x_1}^2 & \cdots & {x_1}^d \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & x_n & {x_n}^2 & \cdots & {x_n}^d \end{bmatrix}$
Parameters	$\beta = \begin{bmatrix} \beta_0 \\ \beta_1 \end{bmatrix}$	$\beta = \begin{bmatrix} \beta_0 \\ \beta_1 \\ \vdots \\ \beta_d \end{bmatrix}$
Model	$y = X\beta + \varepsilon$ (i.e., $y_i = \beta_0 + \beta_1 x_i + \varepsilon_i$)	$y = X\beta + \varepsilon$ (i.e., $y_i = \beta_0 + \beta_1 x_i + \cdots + \beta_d {x_i}^d + \varepsilon_i$)
Loss to minimize	$L\left(\beta\right) = \left(y - X\beta\right)^T\left(y - X\beta\right)$	$L\left(\beta\right) = \left(y - X\beta\right)^T\left(y - X\beta\right)$
Differentiate and set to 0	$\frac{\partial L}{\partial \beta} = -2X^T\left(y - X\beta\right) = 0$	$\frac{\partial L}{\partial \beta} = -2X^T\left(y - X\beta\right) = 0$
Closed-form solution	$\boxed{\hat{\beta} = \left(X^TX\right)^{-1}X^Ty\text{; explicitly:}\hat{\beta}_1 = \frac{\sum\left(x_i - \bar{x}\right)\left(y_i - \bar{y}\right)}{\sum\left(x_i - \bar{x}\right)^2}, \hat{\beta}_0 = \bar{y} - \hat{\beta}_1\bar{x}}$	$\boxed{\hat{\beta} = \left(X^TX\right)^{-1}X^Ty \text{ with } \hat{\beta} = \begin{bmatrix} \hat{\beta}_0 \\ \hat{\beta}_1 \\ \vdots \\ \hat{\beta}_d \end{bmatrix}}$
Fitted values	$\hat{y} = X\hat{\beta}$	$\hat{y} = X\hat{\beta}$

BACK TO LOESS

LOESS

- LOESS (Locally Estimated Scatterplot Smoothing) fits **many small regressions** instead of one global line
- At each target point x_0 , it fits a **local polynomial** (usually linear or quadratic) using **weighted nearby points**
- Nearby points get **more weight**, distant points get **less weight**
- The fitted value at x_0 is the prediction from that local weighted regression
- It is a **non-parametric** method, we don’t assume one global β for the entire dataset

THE MATH

- At each x_0
  - We minimize the **weighted** **least-squares loss**:
    - $L(\beta \mid x_0) = \Sigma_i\, w_i(x_0)\, [y_i - (\beta_0 + \beta_1 x_i + \beta_2 x_i^2 \ldots)]^2$
  - The weight $w_i(x_0)$ is **largest near x_0**
- The **Tricube** kernel

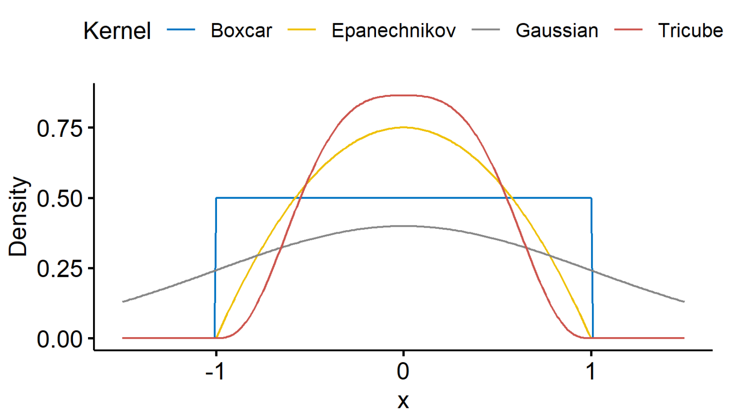
  $w_i(x_0) = (1 - |x_i - x_0|^3 / d^3)^3$ for $|x_i - x_0| < d$, else 0

- The solution at x_0 gives the local coefficients:

  $$\hat{\beta}(x_0) = (X^T W(x_0) X)^{-1} X^T W(x_0) Y$$

- The fitted value:

  $$\hat{y}(x_0) = [1, x_0, x_0^2, \ldots]\ \hat{\beta}(x_0)$$



SOLUTION

- We solve: $\hat{\beta}(x_0) = (X^T W X)^{-1} X^T W Y$

$$X = \begin{bmatrix} 1 & x_1 \\ 1 & x_2 \\ 1 & x_3 \end{bmatrix}, \quad W = \begin{bmatrix} w_1 & 0 & 0 \\ 0 & w_2 & 0 \\ 0 & 0 & w_3 \end{bmatrix}, \quad Y = \begin{bmatrix} y_1 \\ y_2 \\ y_3 \end{bmatrix}$$

Then:

$$\hat{\beta}(x_0) = (X^T W X)^{-1} X^T W Y$$

This yields local slope and intercept (β_0, β_1) at point x_0.

EXAMPLE

- Suppose we have 3 nearby points around $\mathbf{x_0} = \mathbf{5}$
  - x = [4, 5, 6]
  - y = [10, 15, 18]
- Weights using the tricube kernel):
  - w = [0.25, 1.0, 0.25]
- Then $\mathbf{x_0} = \mathbf{5}$ gets the most weight !

$$X = \begin{bmatrix} 1 & 4 \\ 1 & 5 \\ 1 & 6 \end{bmatrix}, \quad W = \begin{bmatrix} 0.25 & 0 & 0 \\ 0 & 1.0 & 0 \\ 0 & 0 & 0.25 \end{bmatrix}$$

$$\hat{\beta} = (X^T W X)^{-1} X^T W Y$$

$$\hat{\beta} \approx \begin{bmatrix} -5.0 \\ 3.9 \end{bmatrix}$$

$$\hat{y}(5) = -5.0 + 3.9 \times 5 = 14.5$$

LOESS SUMMARY

- LOESS repeats a local weighted regression at each x_0
- Each local fit has its own $\beta_0(x_0)$, $\beta_1(x_0)$, etc.
- Nearby points dominate each local line; faraway points barely influence it
- The collection of all local predictions forms a **smooth curve** that adapts to the data
  - The final "curve" is a combination of small/short straight lines (chain links)

OLS ENGINE

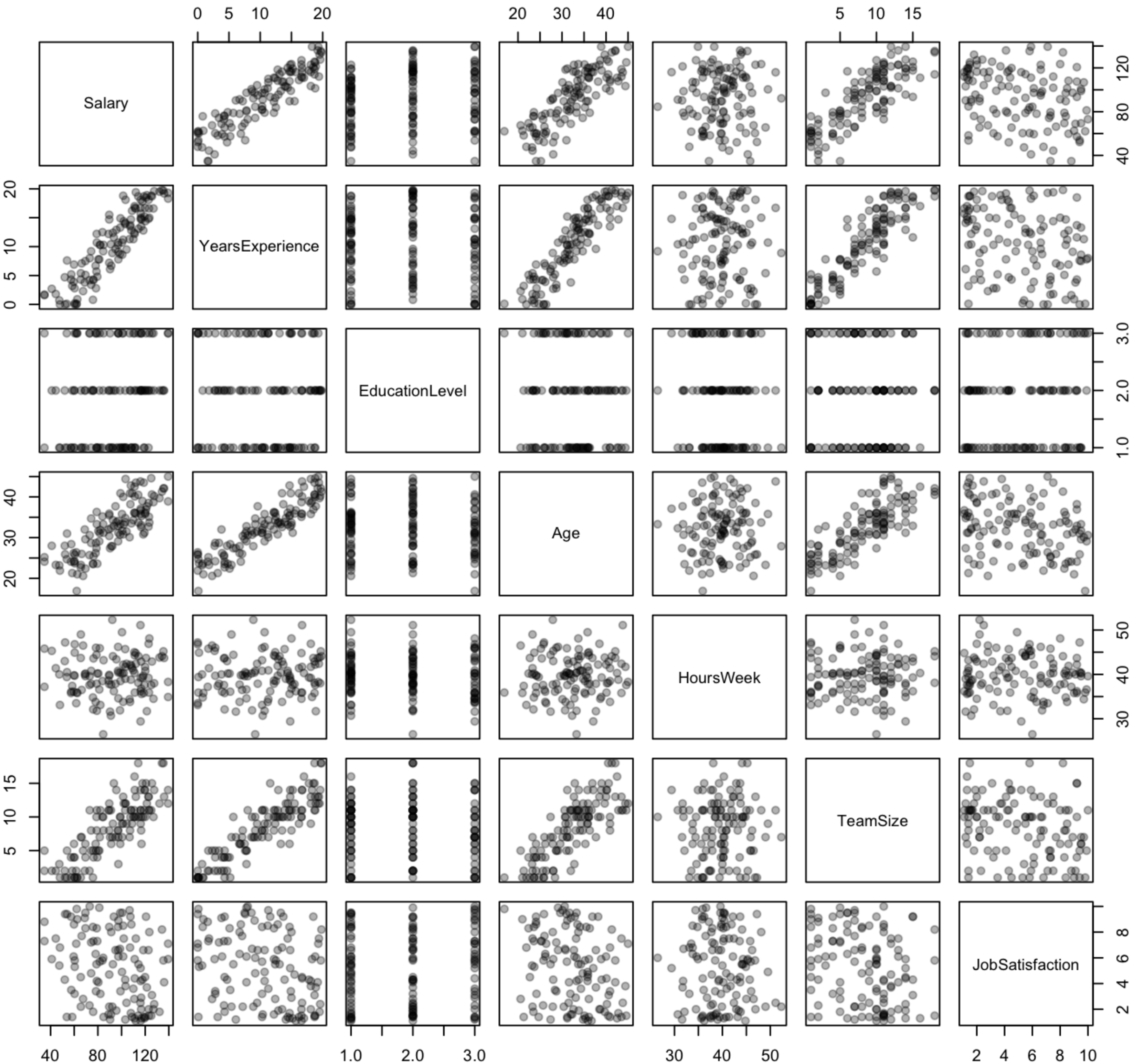
Step	Simple Linear Regression (SLR)	Polynomial Linear Regression (PLR)	LOESS (local weighted simple LR)
Design matrix X	$X = \begin{bmatrix} 1 & x_1 \\ \vdots & \vdots \\ 1 & x_n \end{bmatrix}$	$X = \begin{bmatrix} 1 & x_1 & x_1{}^2 & \cdots & x_1{}^d \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & x_n & x_n{}^2 & \cdots & x_n{}^d \end{bmatrix}$	$X = \begin{bmatrix} 1 & x_1 \\ \vdots & \vdots \\ 1 & x_n \end{bmatrix}$ with weights $W\left(x_0\right) = \begin{bmatrix} w_1 & 0 & \cdots & 0 \\ 0 & w_2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & w_n \end{bmatrix}$
Parameters	$\beta = \begin{bmatrix} \beta_0 \\ \beta_1 \end{bmatrix}$	$\beta = \begin{bmatrix} \beta_0 \\ \beta_1 \\ \vdots \\ \beta_d \end{bmatrix}$	$\beta\left(x_0\right) = \begin{bmatrix} \beta_0(x_0) \\ \beta_1(x_0) \end{bmatrix}$
Model	$y = X\beta + \varepsilon$ (i.e., $y_i = \beta_0 + \beta_1 x_i + \varepsilon_i$)	$y = X\beta + \varepsilon$ (i.e., $y_i = \beta_0 + \beta_1 x_i + \cdots + \beta_d x_i{}^d + \varepsilon_i$)	$y = X\beta\left(x_0\right) + \varepsilon$ (local weights around x_0)
Loss to minimize	$L\left(\beta\right) = \left(y - X\beta\right)^T\left(y - X\beta\right)$	$L\left(\beta\right) = \left(y - X\beta\right)^T\left(y - X\beta\right)$	$L\left(\beta; x_0\right) = \left(y - X\beta\right)^T W\left(x_0\right)\left(y - X\beta\right)$
Differentiate and set to 0	$\frac{\partial L}{\partial \beta} = -2X^T\left(y - X\beta\right) = 0$	$\frac{\partial L}{\partial \beta} = -2X^T\left(y - X\beta\right) = 0$	$\frac{\partial L}{\partial \beta} = -2X^T W\left(x_0\right)\left(y - X\beta\right) = 0$
Closed-form solution	$\hat{\beta} = \left(X^T X\right)^{-1} X^T y; \hat{\beta}_1 = \frac{\sum\left(x_i - \bar{x}\right)\left(y_i - \bar{y}\right)}{\sum\left(x_i - \bar{x}\right)^2}, \hat{\beta}_0 = \bar{y} - \hat{\beta}_1 \bar{x}$	$\hat{\beta} = \left(X^T X\right)^{-1} X^T y$, with $\hat{\beta} = \begin{bmatrix} \hat{\beta}_0 \\ \hat{\beta}_1 \\ \vdots \\ \hat{\beta}_d \end{bmatrix}$	$\hat{\beta}\left(x_0\right) = \left(X^T W\left(x_0\right) X\right)^{-1} X^T W\left(x_0\right) y$
Fitted values	$\hat{y} = X\hat{\beta}$	$\hat{y} = X\hat{\beta}$	$\hat{y}\left(x_0\right) = \hat{\beta}_0\left(x_0\right) + \hat{\beta}_1\left(x_0\right) x_0$

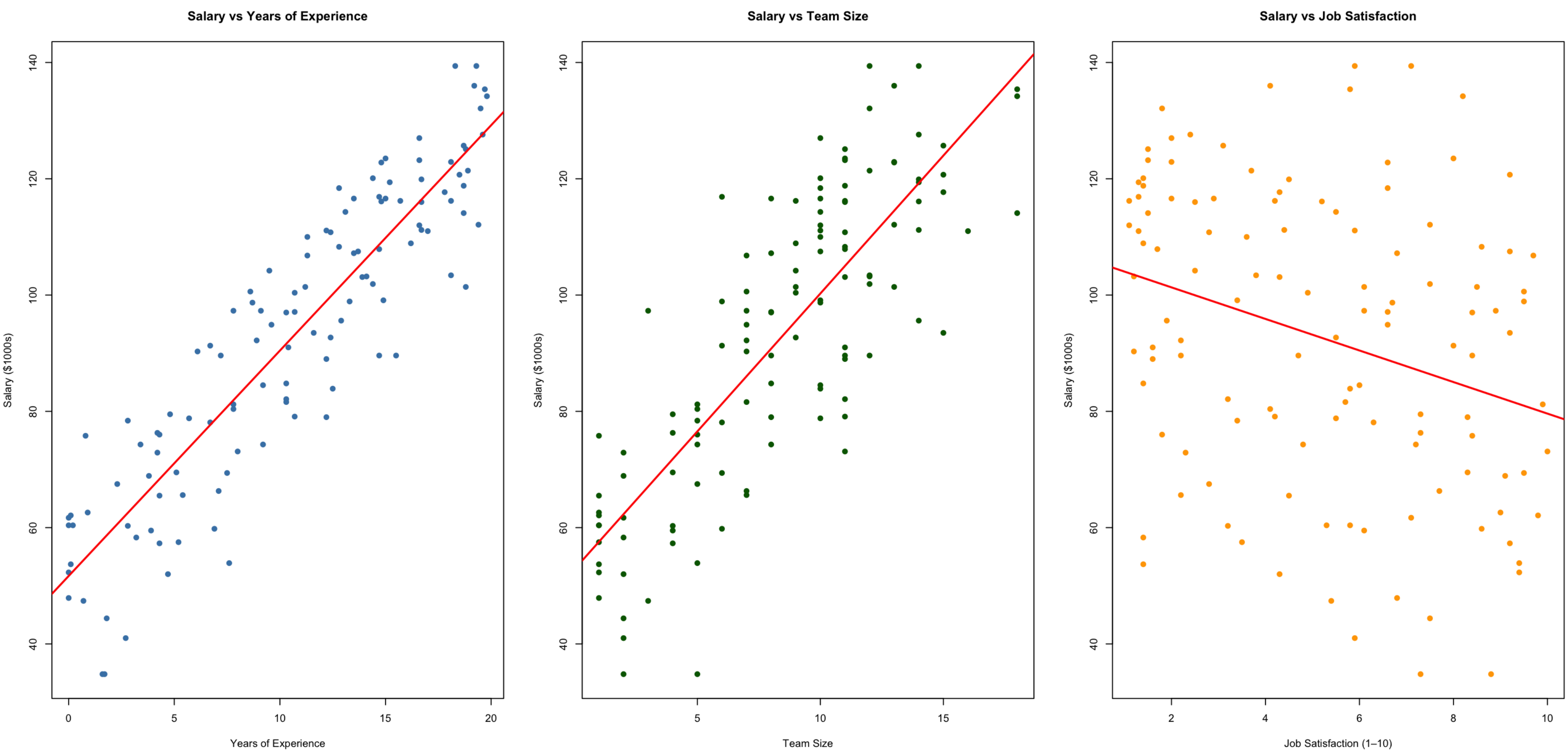
MULTIPLE LINEAR REGRESSION

MULTIPLE FACTORS

- How does SALARY relate to:
    1. YearsExperience
    2. Education Level
    3. Age
    4. HoursWeek
    5. TeamSize
    6. JobSatisfaction

Pairwise Scatterplots: Salary and Predictors

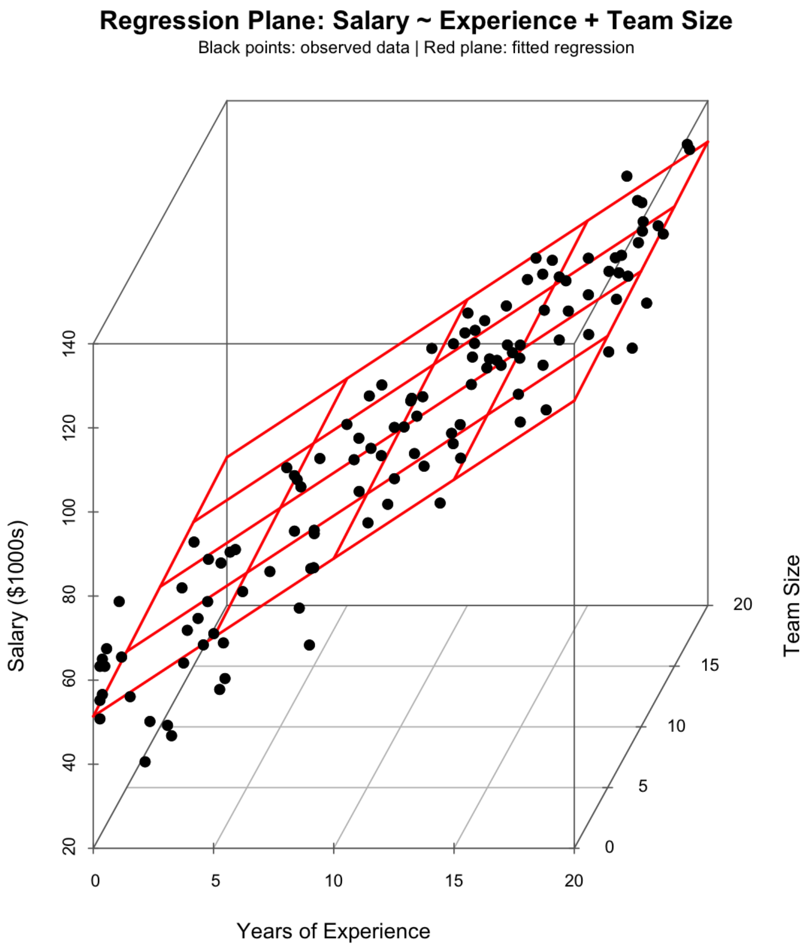



Salary vs Years of Experience
Salary ($1000s)
Years of Experience
Salary vs Team Size
Salary ($1000s)
Team Size
Salary vs Job Satisfaction
Salary ($1000s)
Job Satisfaction (1–10)

FORMALLY

- $y = \beta_0 + \beta_1 x^{(1)} + \beta_2 x^{(2)} + \ldots$

REGRESSION **PLANE**



MLR: COEFFICIENTS

lm(formula = Salary ~ YearsExperience + TeamSize + EducationLevel + JobSatisfaction + Age, data = df)

Residuals:

Min	1Q	Median	3Q	Max
-30.0833	-6.0215	0.5906	7.0405	21.8689

Coefficients:

	Estimate	Std. Error	t value	Pr(>ltl)
(Intercept)	44.9255	8.6548	5.191	9.23e-07 ***
YearsExperience	3.7244	0.4894	7.611	8.54e-12 ***
TeamSize	0.3341	0.4607	0.725	0.470
EducationLevel	5.9496	1.1656	5.104	1.34e-06 ***
JobSatisfaction	-0.4402	0.3441	-1.279	0.203
Age	-0.1063	0.3415	-0.311	0.756

MLR: COEFFICIENTS

lm(formula = Salary ~ YearsExperience + TeamSize + EducationLevel + JobSatisfaction + Age, data = df)

Residuals:

Min	1Q	Median	3Q	Max
-30.0833	-6.0215	0.5906	7.0405	21.8689

Coefficients:

	Estimate	Std. Error	t value	Pr(>|t|)
(Intercept)	44.9255	8.6548	5.191	9.23e-07 ***
YearsExperience	3.7244	0.4894	7.611	8.54e-12 ***
TeamSize	0.3341	0.4607	0.725	0.470
EducationLevel	5.9496	1.1656	5.104	1.34e-06 ***
JobSatisfaction	-0.4402	0.3441	-1.279	0.203
Age	-0.1063	0.3415	-0.311	0.756