

# Lecture 4: Multiple Linear Regression

Pilsung Kang

School of Industrial Management Engineering

Korea University

# AGENDA

**01** Multiple Linear Regression

---

**02** Evaluating Regression Models

---

**03** R Exercise

---

# Multiple Linear Regression

- Regression Example: Predict the selling price of Toyota Corolla



Dependent variable  
(target)

Independent variables  
(attributes, features)


Variable	Description
Price	Offer Price in EUROS
Age_08_04	Age in months as in August 2004
KM	Accumulated Kilometers on odometer
Fuel_Type	Fuel Type (Petrol, Diesel, CNG)
HP	Horse Power
Met_Color	Metallic Color? (Yes=1, No=0)
Automatic	Automatic (Yes=1, No=0)
CC	Cylinder Volume in cubic centimeters
Doors	Number of doors
Quarterly_Tax	Quarterly road tax in EUROS
Weight	Weight in Kilograms

# Multiple Linear Regression

- Goal

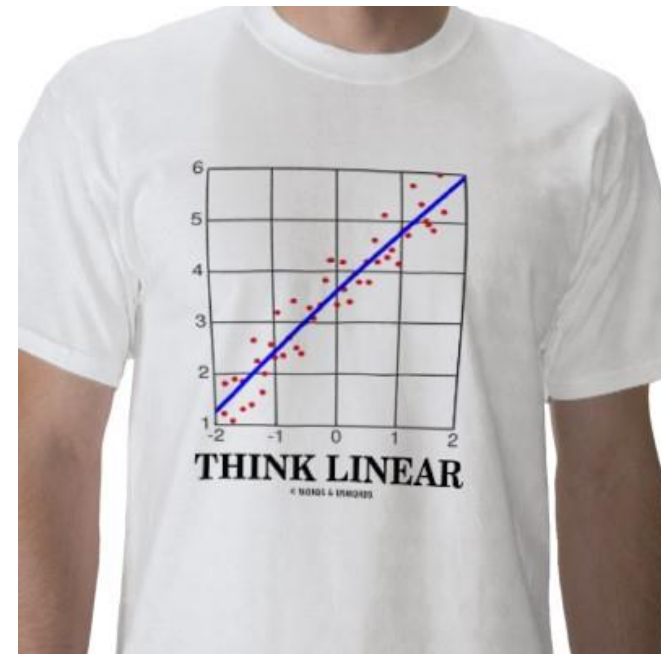
- ✓ Fit a linear relationship between a quantitative dependent variable  $Y$  and a set of predictors  $X_1, X_2, \dots, X_p$ .

$$y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 \cdots + \beta_d x_d + \epsilon$$



$$\hat{y} = \hat{\beta}_0 + \hat{\beta}_1 x_1 + \hat{\beta}_2 x_2 \cdots + \hat{\beta}_d x_d$$

coefficients



# Multiple Linear Regression

- Explanatory vs. Predictive

## Explanatory Regression

- Explain relationship between predictors (explanatory variables) and target.
- Familiar use of regression in data analysis.
- Model Goal: Fit the data well and understand the contribution of explanatory variables to the model.
- “goodness-of-fit”:  $R^2$ , residual analysis, p-values.

$$Y = \beta_0 + \beta_1 x_1 + \dots + \beta_p x_p + \varepsilon$$

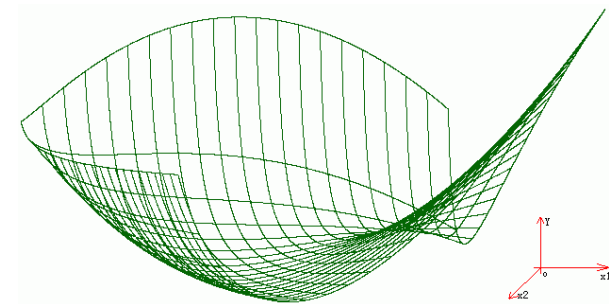
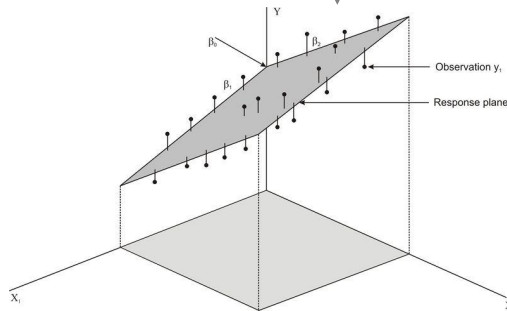
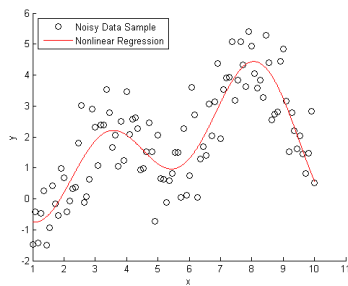
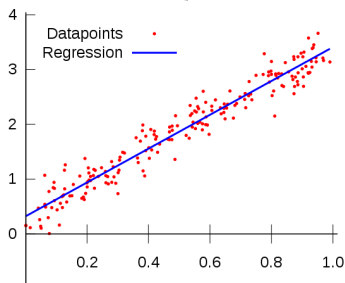
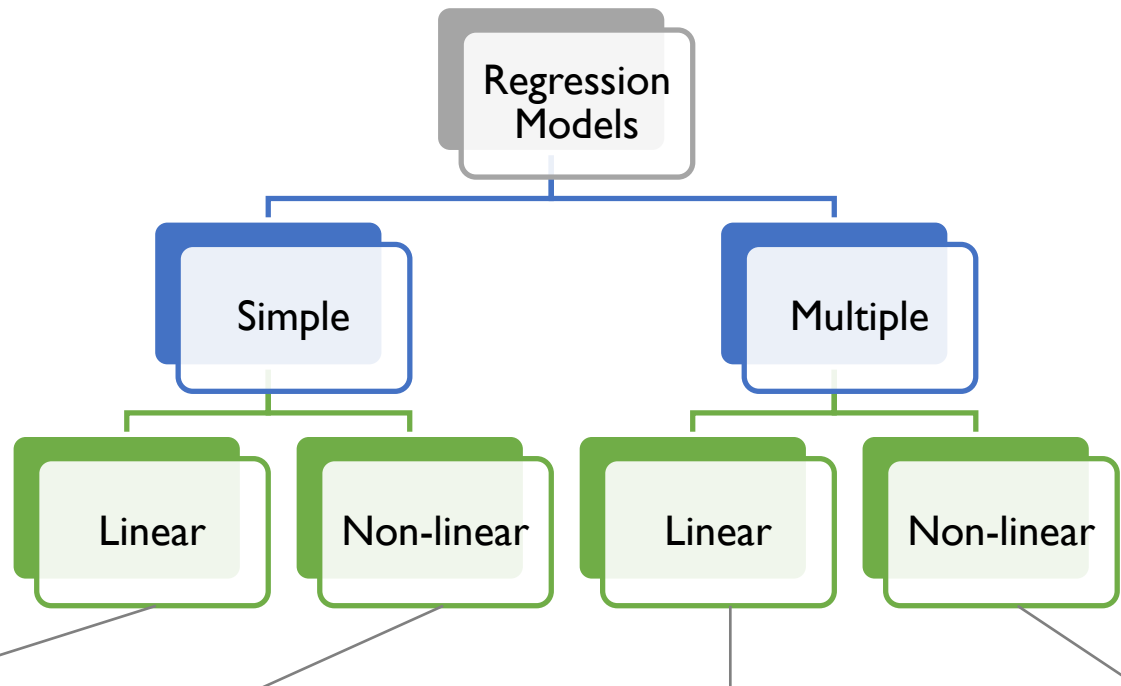
## Predictive Regression

- Predict target values in other data where we have predictor values, but not target values.
- Classic data mining context
- Model Goal: Optimize predictive accuracy
- Train model on training data
- Assess performance on validation (hold-out) data
- Explaining role of predictors is not primary purpose (but useful)

$$Y = \beta_0 + \beta_1 x_1 + \dots + \beta_p x_p + \varepsilon$$

# Multiple Linear Regression

- Type of Regression

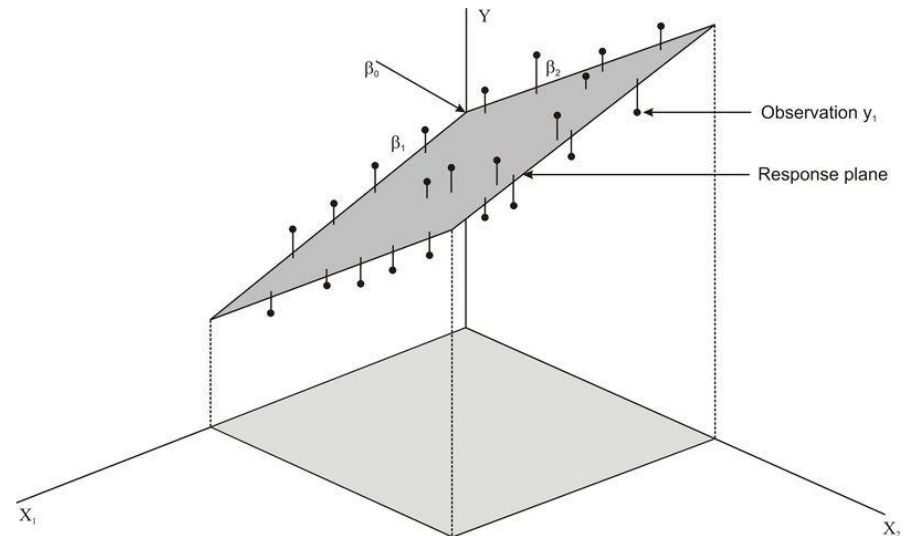
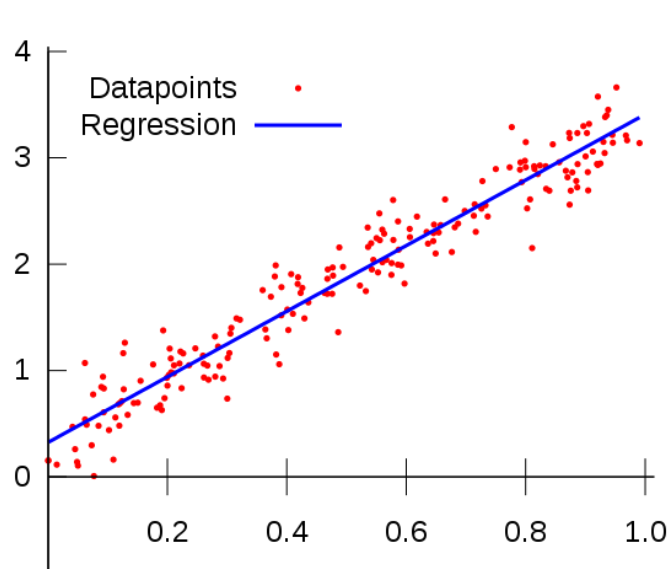


# Multiple Linear Regression

- Linear Regression

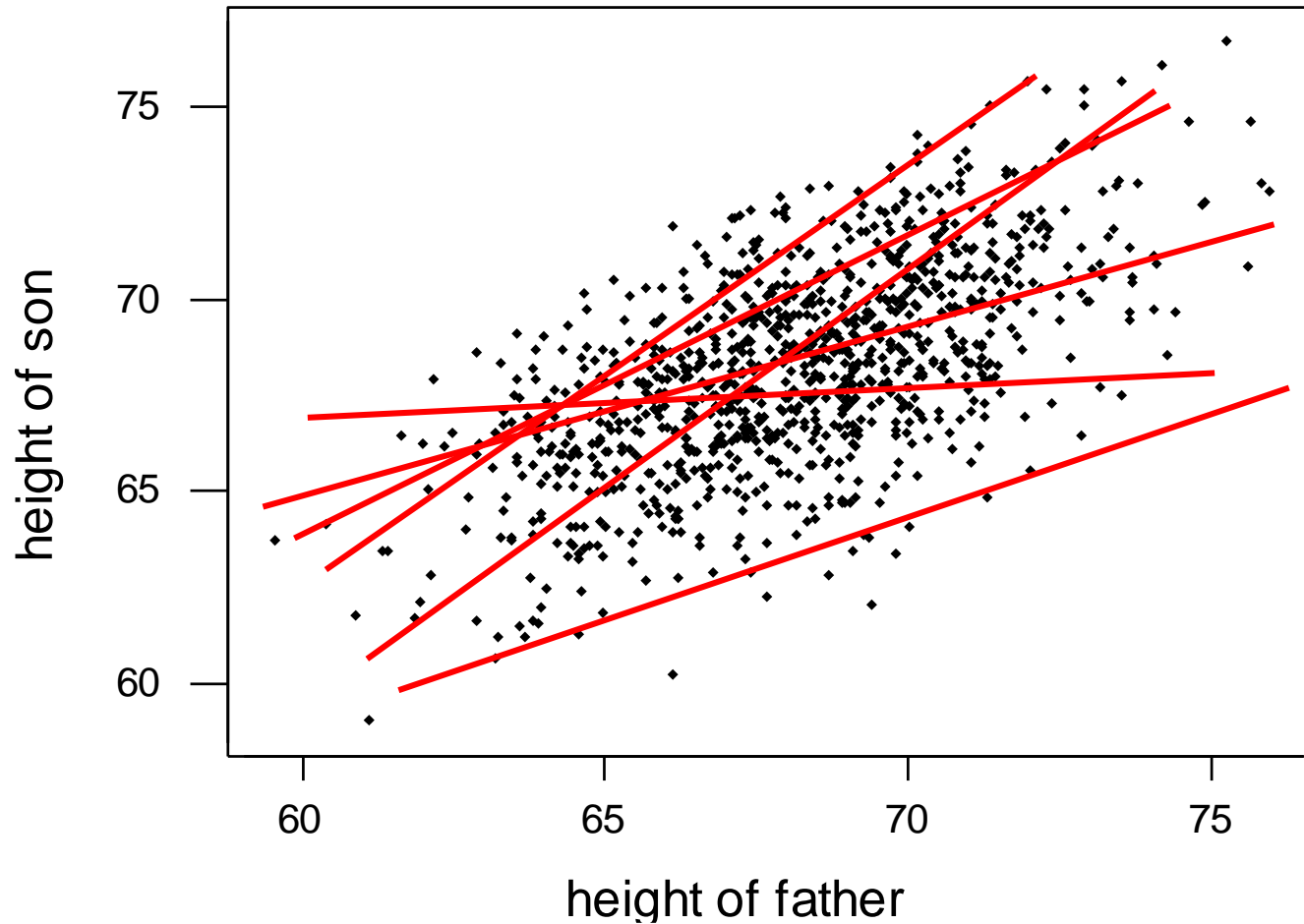
- ✓ Assume that the relationship between the input variable and the target variable is always **linear**.

$$\hat{y} = \hat{\beta}_0 + \hat{\beta}_1 x_1 + \hat{\beta}_2 x_2 \cdots + \hat{\beta}_d x_d$$



# Multiple Linear Regression

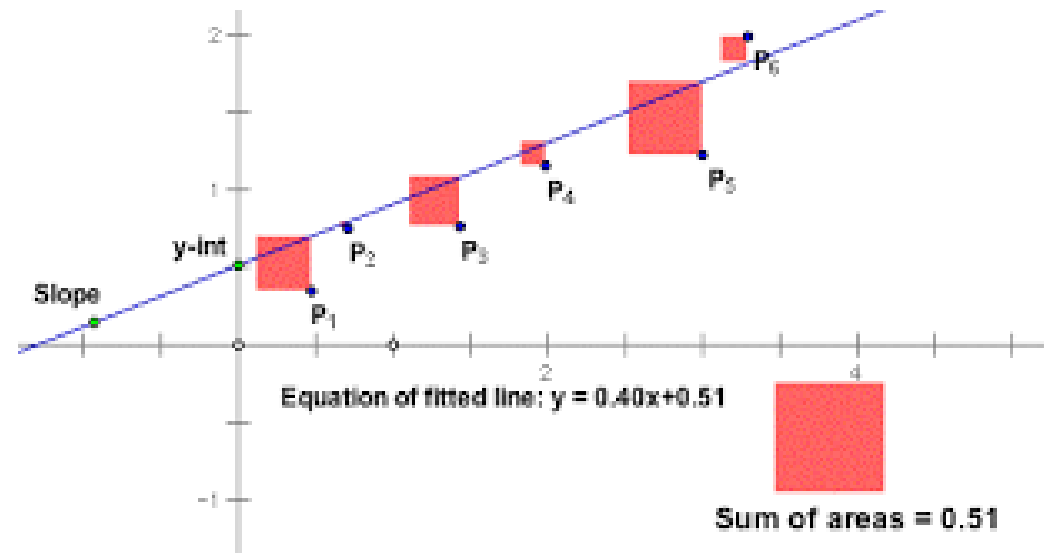
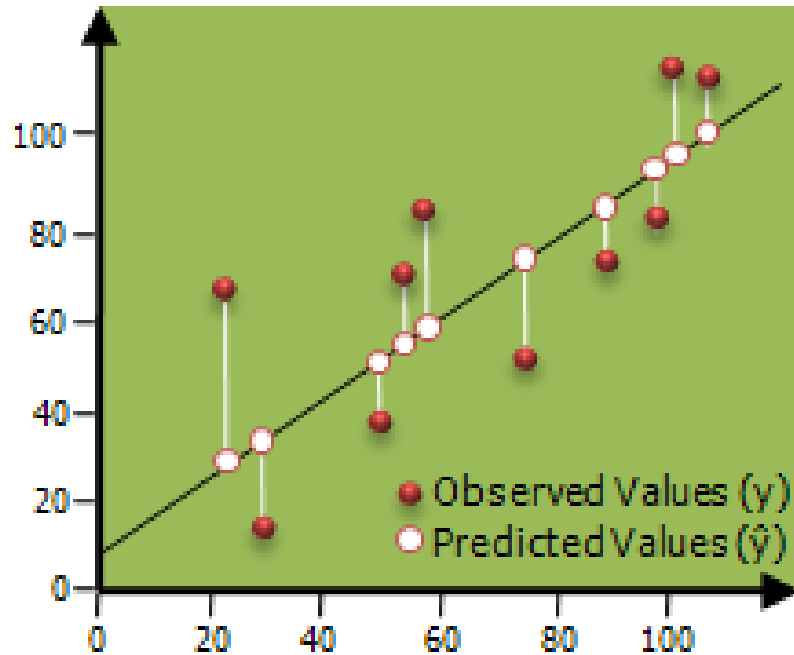
- Which line is optimal?





# Multiple Linear Regression

- Estimating the coefficients
  - ✓ Ordinary least square (OLS): Minimize the squared difference between the actual target value and the estimated value by the regression model



# Multiple Linear Regression

- Estimating the coefficients

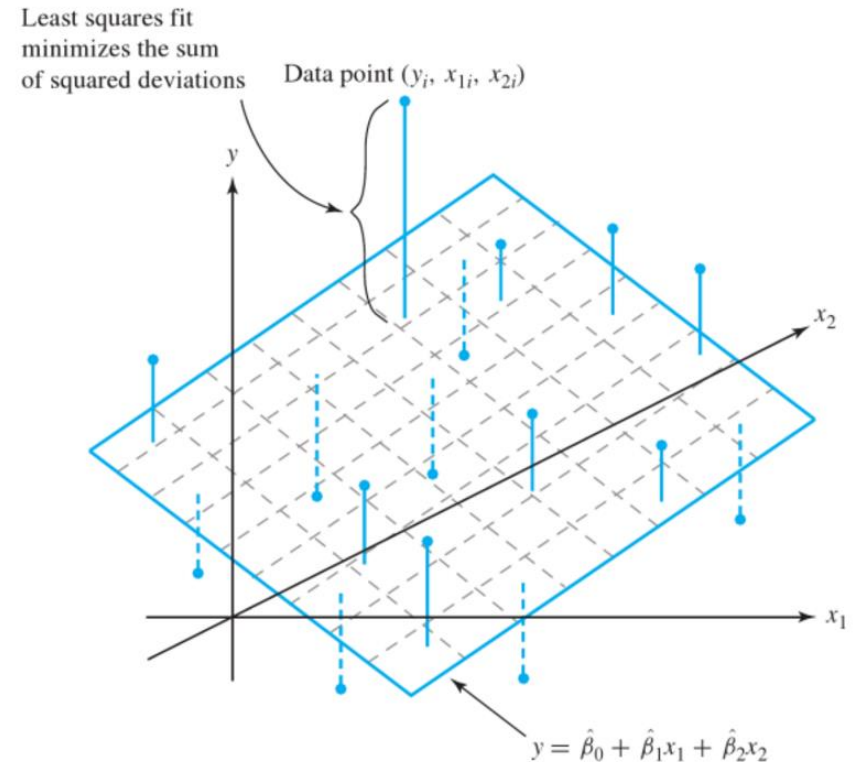
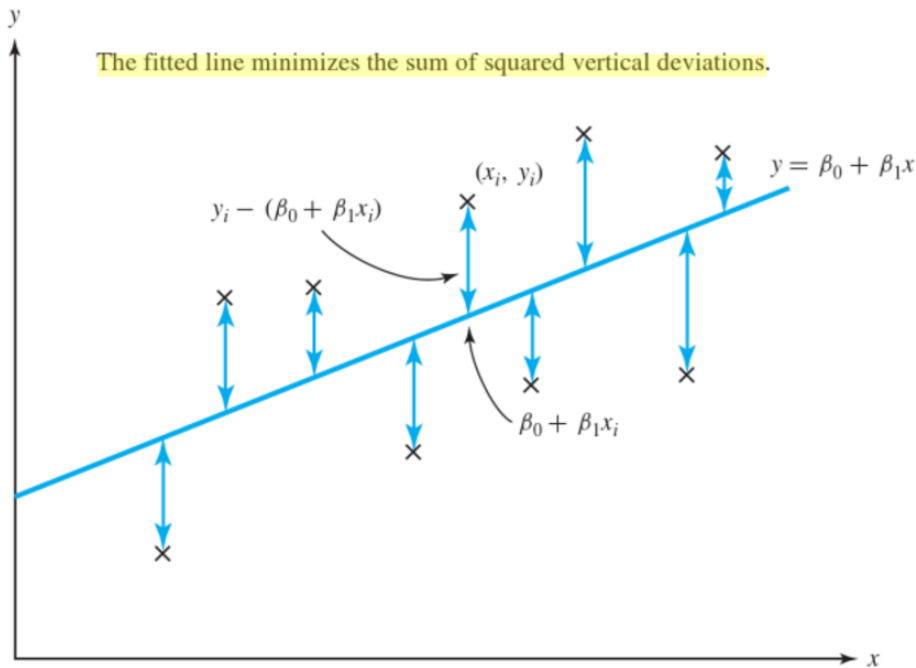
- ✓ Ordinary least square (OLS)

- Actual target:  $y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 \cdots + \beta_d x_d + \epsilon$
- Predicted target:  $\hat{y} = \hat{\beta}_0 + \hat{\beta}_1 x_1 + \hat{\beta}_2 x_2 \cdots + \hat{\beta}_d x_d$
- **Goal:** minimize the difference between the actual and predicted target.

$$\begin{aligned} \min \frac{1}{2} \sum_{i=1}^n (y_i - \hat{y}_i)^2 \\ = \frac{1}{2} (y_i - \hat{\beta}_0 + \hat{\beta}_1 x_{i1} + \hat{\beta}_2 x_{i2} \cdots + \hat{\beta}_d x_{id})^2 \end{aligned}$$

# Multiple Linear Regression

- Estimating the coefficients
  - ✓ Ordinary least square (OLS)

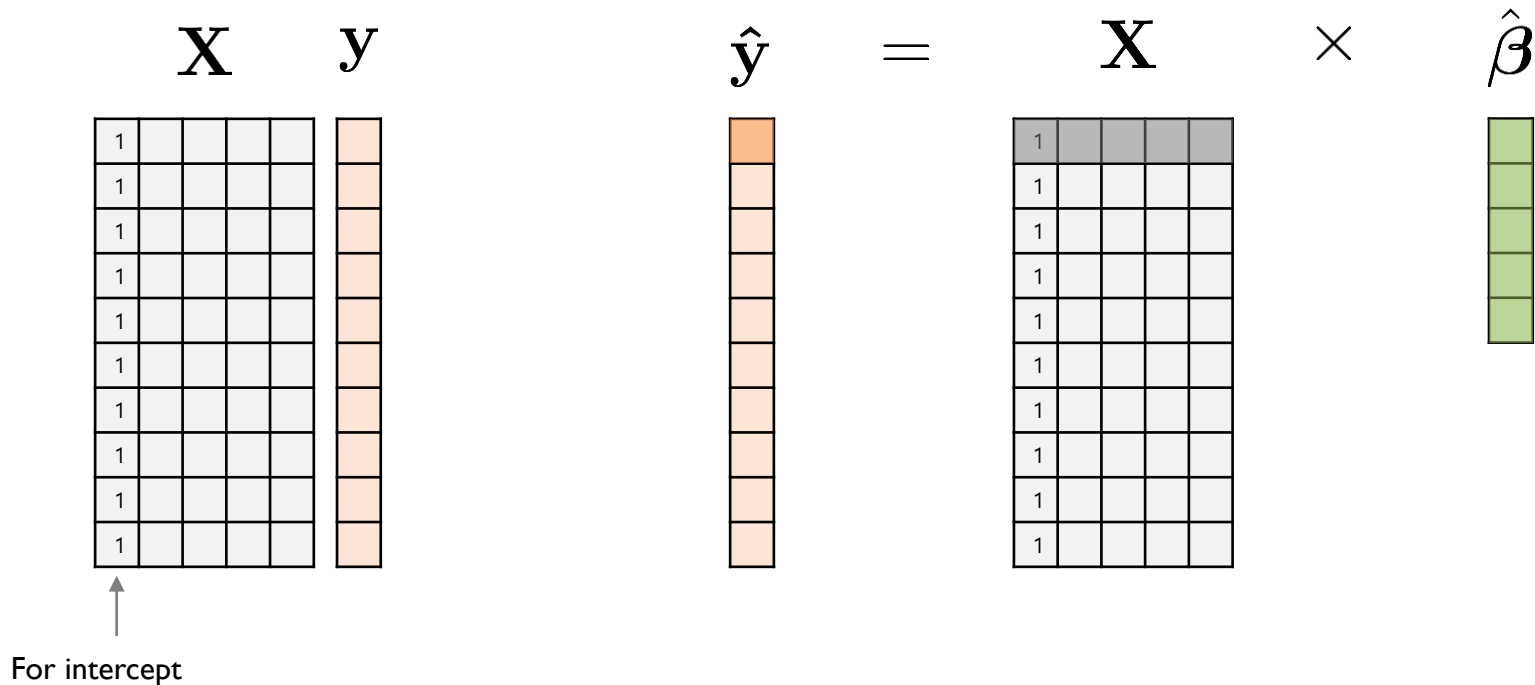


# Multiple Linear Regression

- Ordinary least square: Matrix solution

$\mathbf{X} : n \times (d + 1)$  matrix,  $\mathbf{y} : n \times 1$  vector

$\hat{\boldsymbol{\beta}} : (d + 1) \times 1$  vector



# Multiple Linear Regression

- Ordinary least square: Matrix solution

$\mathbf{X} : n \times (d + 1)$  matrix,  $\mathbf{y} : n \times 1$  vector

$\hat{\boldsymbol{\beta}} : (d + 1) \times 1$  vector

$$\min E(\mathbf{X}) = \frac{1}{2} \left( \mathbf{y} - \mathbf{X}\hat{\boldsymbol{\beta}} \right)^T \left( \mathbf{y} - \mathbf{X}\hat{\boldsymbol{\beta}} \right)$$

$$\Rightarrow \frac{\partial E(\mathbf{X})}{\partial \hat{\boldsymbol{\beta}}} = -\mathbf{X}^T \left( \mathbf{y} - \mathbf{X}\hat{\boldsymbol{\beta}} \right) = 0$$

$$\Rightarrow \mathbf{X}^T \mathbf{y} + \mathbf{X}^T \mathbf{X} \hat{\boldsymbol{\beta}} = 0$$

$$\hat{\boldsymbol{\beta}} = \left( \mathbf{X}^T \mathbf{X} \right)^{-1} \mathbf{X}^T \mathbf{y} \longrightarrow \text{Unique and explicit solution exists!}$$

# Multiple Linear Regression

- Ordinary least square: Matrix solution

$$\hat{\beta} = \left( \mathbf{X}^T \mathbf{X} \right)^{-1} \mathbf{X}^T \mathbf{y}$$

$$\hat{\beta} = \left( \mathbf{X}^T \mathbf{X} \right)^{-1} \mathbf{X}^T \mathbf{y}$$

The diagram illustrates the matrix solution for the regression coefficient. It shows the vector  $\hat{\beta}$  (green) equal to the product of the matrix  $(\mathbf{X}^T \mathbf{X})^{-1}$  (gray) and the vector  $\mathbf{X}^T \mathbf{y}$  (orange). The matrix  $\mathbf{X}^T \mathbf{X}$  is shown as a 10x10 grid with the first row containing ten 1s and the rest of the rows being empty. The vector  $\mathbf{X}^T \mathbf{y}$  is shown as a 10x1 grid with the first row containing a 1 and the rest of the rows being empty.

Closed form solution for the regression coefficient

# Multiple Linear Regression

- Ordinary least square

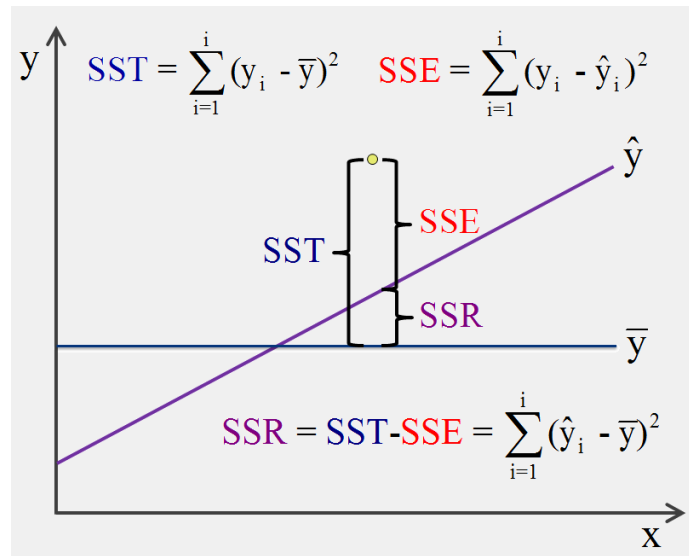
- ✓ Finds the best estimates  $\beta$  when the following conditions are satisfied:

- The noise  $\varepsilon$  follows a normal distribution.
    - The linear relationship is correct.
    - The cases are independent of each other.
    - The variability in Y values for a given set of predictors is the same regardless of the values of the predictors (homoskedasticity).

# Multiple Linear Regression

- Goodness of fit

- ✓ Sum-of-Squares Decomposition



$$\underbrace{\sum_{j=1}^n (y_j - \bar{y})^2}_{\substack{\text{(total sum of squares)} \\ \text{about mean}}} = \underbrace{\sum_{j=1}^n (\hat{y}_j - \bar{y})^2}_{\substack{\text{regression} \\ \text{sum of squares}}} + \underbrace{\sum_{j=1}^n \hat{\varepsilon}_j^2}_{\substack{\text{(residual (error))} \\ \text{sum of squares}}} .$$

**SST**
**SSR**
**SSE**



# Multiple Linear Regression

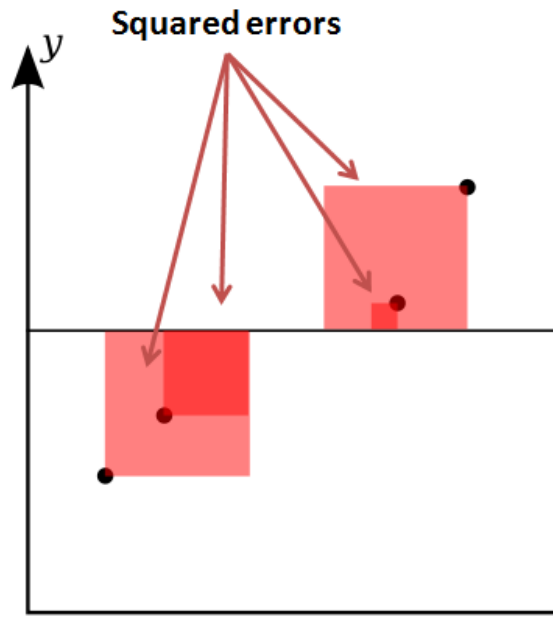
- Goodness-of-fit: (Adjusted)  $R^2$

$$R^2 = 1 - \frac{\sum_{j=1}^n \hat{\varepsilon}_j^2}{\sum_{j=1}^n (y_j - \bar{y})^2} = \frac{\sum_{j=1}^n (\hat{y}_j - \bar{y})^2}{\sum_{j=1}^n (y_j - \bar{y})^2} \quad R^2 = 1 - \frac{SSE}{SST} = \frac{SSR}{SST}$$

- ✓ Gives the proportion of the total variation in the  $y_i$ 's explained by the predictor variables
- ✓  $0 \leq R^2 \leq 1$
- ✓  $R^2 = 1 \rightarrow$  The fitted equation passes through all the data points
- ✓  $R^2 = 0 \rightarrow$  There is no linear relationship between the predictor variables and the target variable

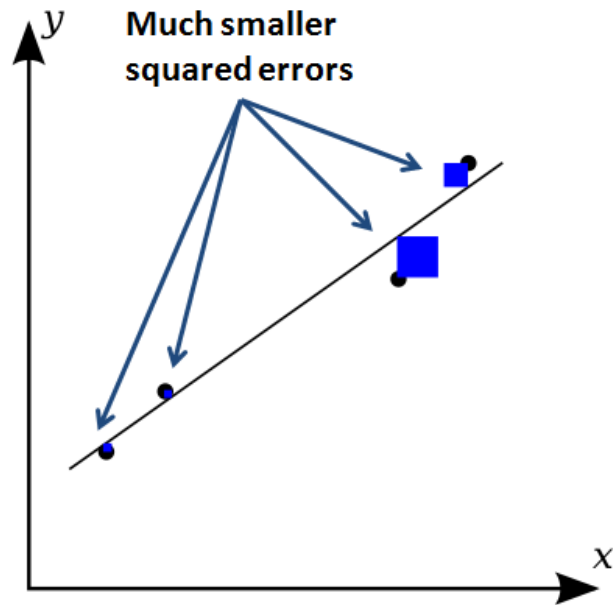
# Multiple Linear Regression

- Goodness-of-fit: (Adjusted)  $R^2$ 
  - ✓ Graphical interpretation



Computationally:

$$R\text{-squared} = 1 - \left[ \frac{SS_{\text{error}}}{SS_{\text{total}}} \right]$$



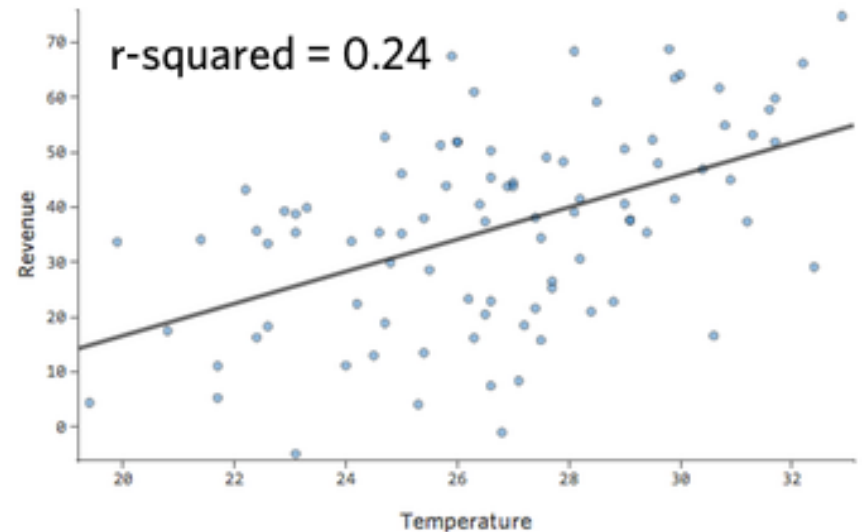
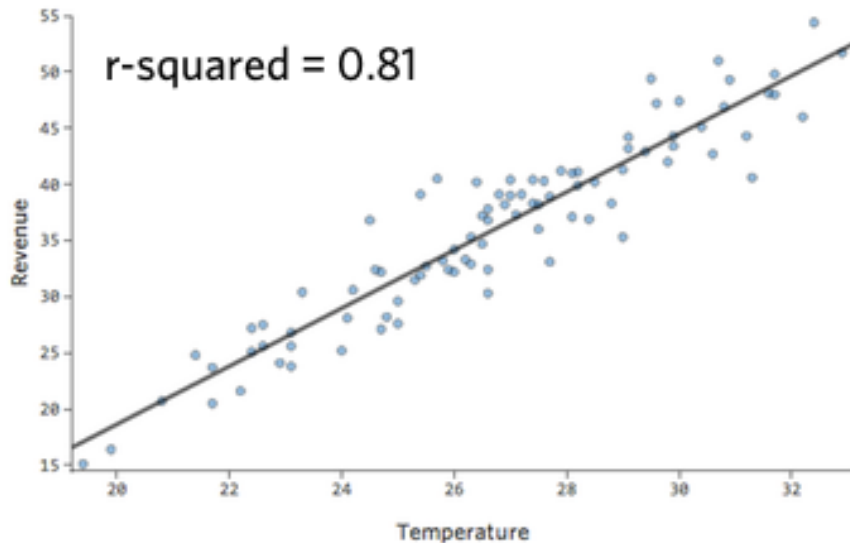
Conceptually:

Force x and y to be independent, calculate **the squared error**.

Allow for a relationship between x and y, does this reduce your **error**?

# Multiple Linear Regression

- Goodness-of-fit: (Adjusted)  $R^2$ 
  - ✓ The proportionate reduction of total variation associated with the use of the predictor variable  $Z$ .



- $R^2$  of your model is very high
  - I did a good job! (No!)
  - This dataset has a strong linear relationship between the  $X$  and  $y$
  - Because everyone can have the same solution

# Multiple Linear Regression

- Goodness-of-fit: (Adjusted)  $R^2$

- ✓ Adjusted  $R^2$

$$R_{adj}^2 = 1 - \left[ \frac{n-1}{n-(p+1)} \right] \frac{SSE}{SST} \leq 1 - \frac{SSE}{SST} = R^2$$

- ✓  $R^2$  increases monotonically when a (possibly not significant) new variable is added

- ✓ Adjusted  $R^2$  fix this problem

- ✓ If an insignificant variable is added, the adjusted  $R^2$  does not increase

- Model verification

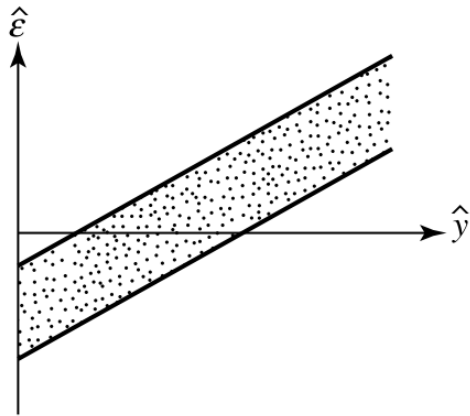
- ✓ Check whether the model satisfies the following assumptions

- Residuals are independent

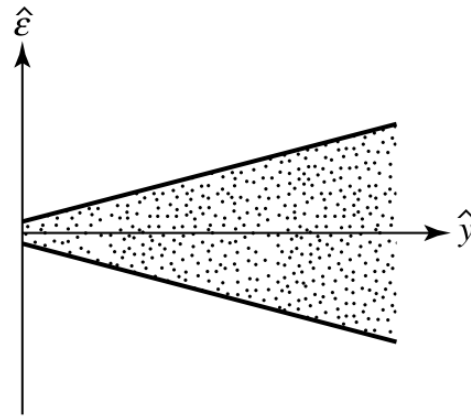
- Residuals have zero mean and a constant variance

# Multiple Linear Regression

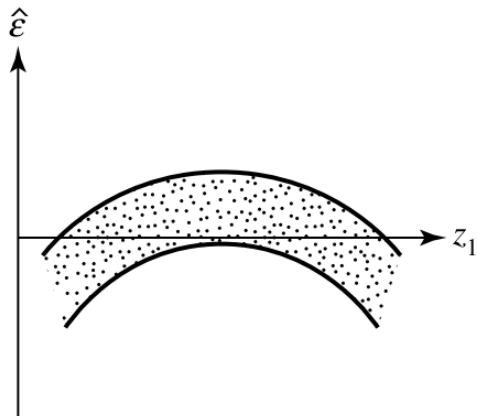
- Residual plots



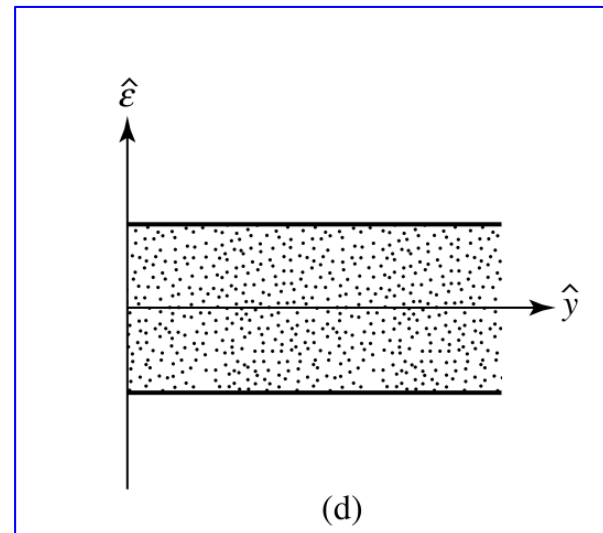
(a)



(b)



(c)

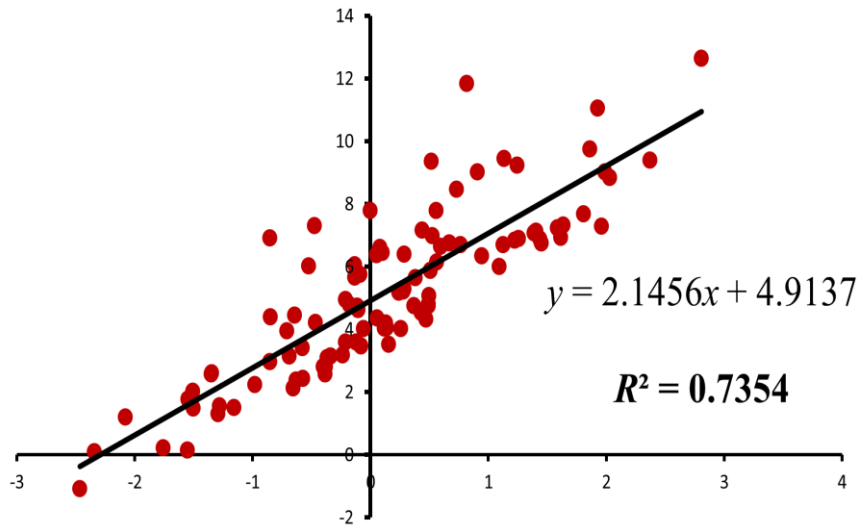


(d)

# Multiple Linear Regression

- Model checking

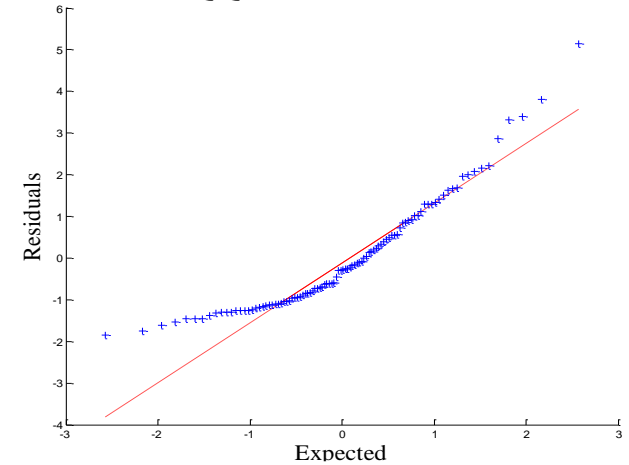
$$y = 2x + \varepsilon, \quad \varepsilon \sim \text{Gamma}(2,1)$$



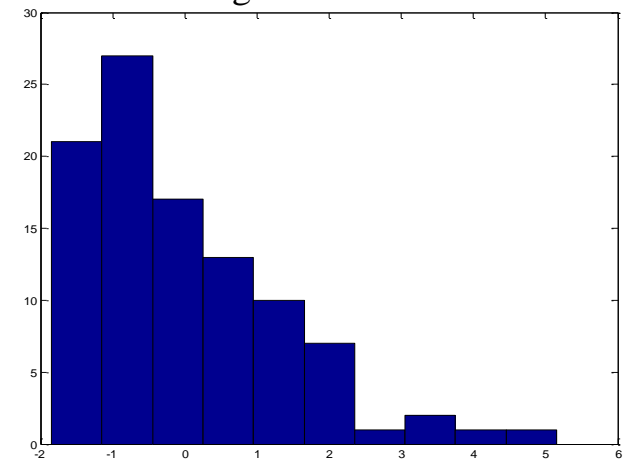
Regression model



QQ Plot of Residuals



Histogram of Residuals



# Multiple Linear Regression: Example

- Example: predict the selling price of Toyota corolla

Y

X

Price	Age_08_04	KM	Fuel_Type	HP	Met_Color	Automatic	cc	Doors	Quarterly_Tax	Weight
13500	23	46986	Diesel	90	1	0	2000	3	210	1165
13750	23	72937	Diesel	90	1	0	2000	3	210	1165
13950	24	41711	Diesel	90	1	0	2000	3	210	1165
14950	26	48000	Diesel	90	0	0	2000	3	210	1165
13750	30	38500	Diesel	90	0	0	2000	3	210	1170
12950	32	61000	Diesel	90	0	0	2000	3	210	1170
16900	27	94612	Diesel	90	1	0	2000	3	210	1245
18600	30	75889	Diesel	90	1	0	2000	3	210	1245
21500	27	19700	Petrol	192	0	0	1800	3	100	1185
12950	23	71138	Diesel	69	0	0	1900	3	185	1105
20950	25	31461	Petrol	192	0	0	1800	3	100	1185
19950	22	43610	Petrol	192	0	0	1800	3	100	1185
19600	25	32189	Petrol	192	0	0	1800	3	100	1185
21500	31	23000	Petrol	192	1	0	1800	3	100	1185
22500	32	34131	Petrol	192	1	0	1800	3	100	1185
22000	28	18739	Petrol	192	0	0	1800	3	100	1185
22750	30	34000	Petrol	192	1	0	1800	3	100	1185
17950	24	21716	Petrol	110	1	0	1600	3	85	1105
16750	24	25563	Petrol	110	0	0	1600	3	19	1065

# Multiple Linear Regression: Example

- Data preprocessing

- ✓ Create dummy variables for fuel types

	Fuel_type = Diesel	Fuel_type = Petrol	Fuel_type = CNG
Diesel	1	0	0
Petrol	0	1	0
CNG	0	0	1

- Data partitioning

- ✓ 60% training data / 40% validation data

Id	Model	Price	Age_08_04	Mfg_Month	Mfg_Year	KM	Fuel_Type_Diesel	Fuel_Type_Petrol
1	RRA 2/3-Doors	13500	23	10	2002	46986	1	0
4	RRA 2/3-Doors	14950	26	7	2002	48000	1	0
5	SOL 2/3-Doors	13750	30	3	2002	38500	1	0
6	SOL 2/3-Doors	12950	32	1	2002	61000	1	0
9	VT I 2/3-Doors	21500	27	6	2002	19700	0	1
10	RRA 2/3-Doors	12950	23	10	2002	71138	1	0
12	BNS 2/3-Doors	19950	22	11	2002	43610	0	1
17	ORT 2/3-Doors	22750	30	3	2002	34000	0	1



# Multiple Linear Regression: Example

- Fitted linear regression model

Input variables	Coefficient	Std. Error	p-value	SS
Constant term	-3608.418457	1458.620728	0.0137	97276410000
Age_08_04	-123.8319168	3.367589	0	8033339000
KM	-0.017482	0.00175105	0	251574500
Fuel_Type_Diesel	210.9862518	474.9978333	0.6571036	6212673
Fuel_Type_Petrol	2522.066895	463.6594238	0.00000008	4594.9375
HP	20.71352959	4.67398977	0.00001152	330138600
Met_Color	-50.48505402	97.85591125	0.60614568	596053.75
Automatic	178.1519013	212.0528565	0.40124047	19223190
cc	0.01385481	0.09319961	0.88188446	1272449
Doors	20.02487946	51.0899086	0.69526076	39265060
Quarterly_Tax	16.7742424	2.09381151	0	160667200
Weight	15.41666317	1.40446579	0	214696000

$\beta$

Significance  
Probability

# Multiple Linear Regression: Example

- Interpret the result
  - ✓ Regression coefficient
    - Beta value for the corresponding predictor variable
    - The amount of change when the predictor variable increases by 1
    - If it is **positive/negative**, then the predictor variable and the target variable are **positively/negatively** correlated

Input variables	Coefficient	Std. Error	p-value	SS
Constant term	-3608.418457	1458.620728	0.0137	97276410000
Age_08_04	-123.8319168	3.367589	0	8033339000
KM	-0.017482	0.00175105	0	251574500
Fuel_Type_Diesel	210.9862518	474.9978333	0.6571036	6212673
Fuel_Type_Petrol	2522.066895	463.6594238	0.00000008	4594.9375
HP	20.71352959	4.67398977	0.00001152	330138600
Met_Color	-50.48505402	97.85591125	0.60614568	596053.75
Automatic	178.1519013	212.0528565	0.40124047	19223190
cc	0.01385481	0.09319961	0.88188446	1272449
Doors	20.02487946	51.0899086	0.69526076	39265060
Quarterly_Tax	16.7742424	2.09381151	0	160667200
Weight	15.41666317	1.40446579	0	214696000

# Multiple Linear Regression: Example

- Interpret the result

- ✓ p-value

- Indicate whether the regression coefficient is statistically significant or not
- A predictor variable is important for modeling when its p-value is close to 0
- Can be used to select significant variables (e.g., use the variables with p-value less than 0.05)

Input variables	Coefficient	Std. Error	p-value	SS
Constant term	-3608.418457	1458.620728	0.0137	97276410000
Age_08_04	-123.8319168	3.367589	0	8033339000
KM	-0.017482	0.00175105	0	251574500
Fuel_Type_Diesel	210.9862518	474.9978333	0.6571036	6212673
Fuel_Type_Petrol	2522.066895	463.6594238	0.00000008	4594.9375
HP	20.71352959	4.67398977	0.00001152	330138600
Met_Color	-50.48505402	97.85591125	0.60614568	596053.75
Automatic	178.1519013	212.0528565	0.40124047	19223190
cc	0.01385481	0.09319961	0.88188446	1272449
Doors	20.02487946	51.0899086	0.69526076	39265060
Quarterly_Tax	16.7742424	2.09381151	0	160667200
Weight	15.41666317	1.40446579	0	214696000

# AGENDA

**01** Multiple Linear Regression

---

**02** Evaluating Regression Models

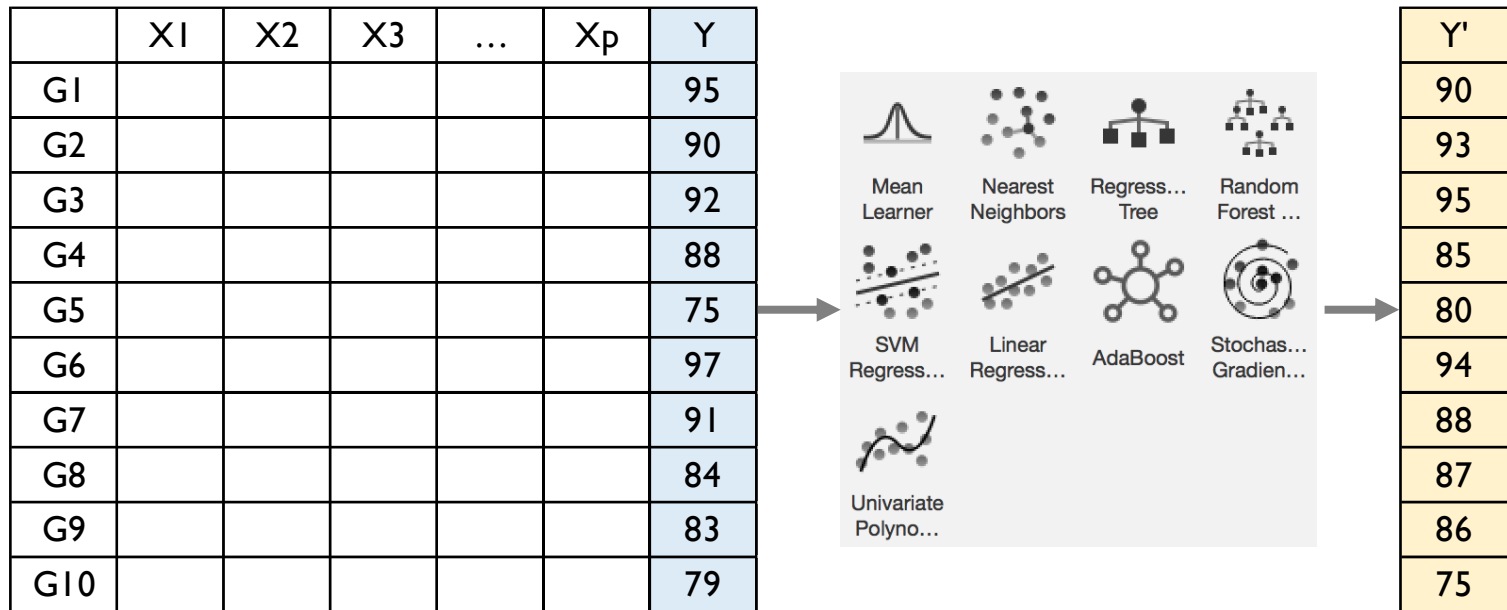
---

**03** R Exercise

---

# Evaluating Regression Models

- Example: predict a yield based on process sensors

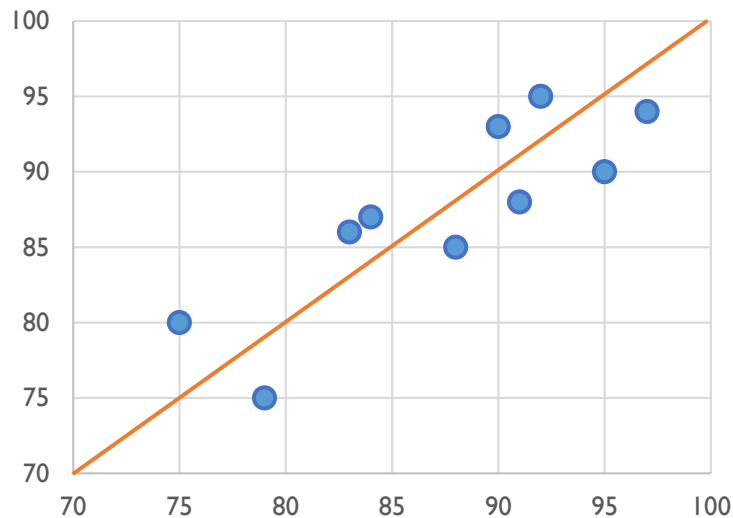


How accurate is this model?

# Evaluating Regression Models

- Performance measure I: Average Error
  - ✓ Compare the average difference between the actual and predicted y
  - ✓ Mislead to an inappropriate conclusion based on the sign effect

$$\text{Average Error} = \frac{1}{n} \sum_{i=1}^n (y_i - y'_i)$$

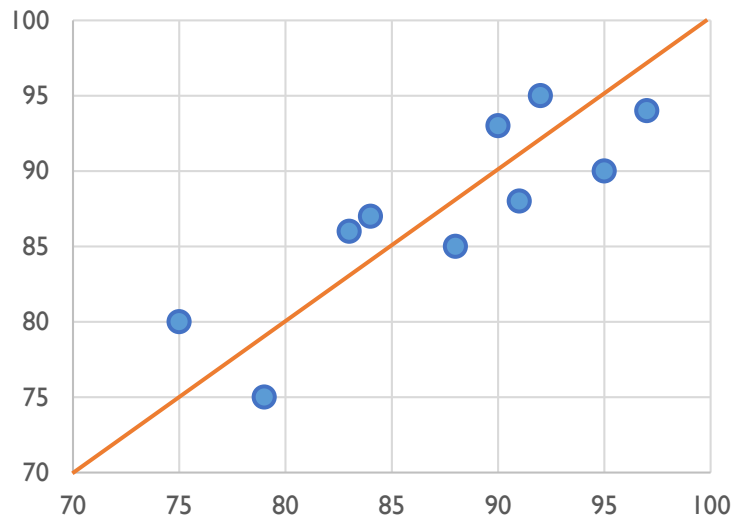


Y	Y'	Y-Y'
95	90	5
90	93	-3
92	95	-3
88	85	3
75	80	-5
97	94	3
91	88	3
84	87	-3
83	86	-3
79	75	4
Average Error		0.1

# Evaluating Regression Models

- Performance measure 2: Mean Absolute Error (MAE)
  - ✓ Compute the average of absolute value of differences between the predicted and actual  $y$

$$\text{MAE} = \frac{1}{n} \sum_{i=1}^n |y_i - y'_i|$$



Y	Y'	Y-Y'
95	90	5
90	93	3
92	95	3
88	85	3
75	80	5
97	94	3
91	88	3
84	87	3
83	86	3
79	75	4
MAE		3.5

# Evaluating Regression Models

- Performance measure 3: Mean Absolute Percentage Error (MAPE)
  - ✓ MAE can only provide the degree of absolute difference between the predicted and actual  $y$  but cannot provide the relative difference between them
  - ✓ Ex) The MAEs of the two models below are the same ( $MAE = 1$ )

Y	Y'	Y-Y'
1	0	1
1	2	1
1	0	1
1	2	1
1	0	1
1	2	1
1	0	1
1	2	1
1	0	1
1	2	1
MAE		1

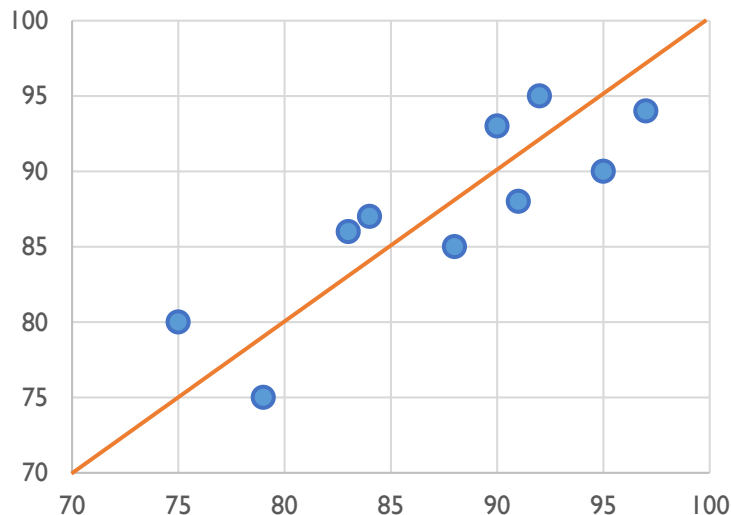
Y	Y'	Y-Y'
100	99	1
100	101	1
100	99	1
100	101	1
100	99	1
100	101	1
100	99	1
100	101	1
100	99	1
100	101	1
MAE		1



# Evaluating Regression Models

- Performance measure 3: Mean Absolute Percentage Error (MAPE)
  - ✓ Provide the relative absolute difference in terms of %
  - ✓ Commonly adopted by domains in which relative differences are more important than the absolute difference (ex: quality control in manufacturing process)

$$\text{MAPE} = \frac{1}{n} \sum_{i=1}^n \left| \frac{y_i - y'_i}{y_i} \right|$$

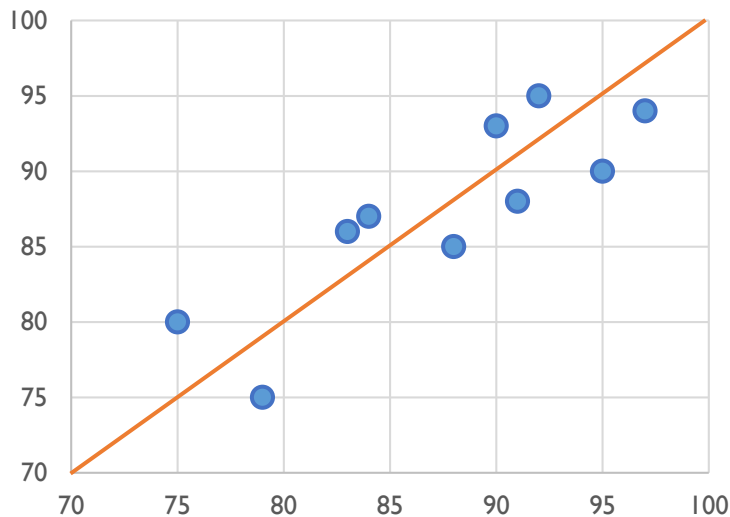


Y	Y'	Y-Y'	Y-Y' / Y
95	90	5	5.26%
90	93	3	3.33%
92	95	3	3.26%
88	85	3	3.41%
75	80	5	6.67%
97	94	3	3.09%
91	88	3	3.30%
84	87	3	3.57%
83	86	3	3.61%
79	75	4	5.06%
MAE		3.5	4.06%

# Evaluating Regression Models

- Performance measure 4 & 5: (Root) Mean Squared Error ((R)MSE)
  - ✓ Use the square instead of absolute value to resolve the effect of sign

$$\text{MSE} = \frac{1}{n} \sum_{i=1}^n (y_i - y'_i)^2, \quad \text{RMSE} = \sqrt{\frac{1}{n} \sum_{i=1}^n (y_i - y'_i)^2}$$



Y	Y'	(Y-Y') <sup>2</sup>
95	90	25
90	93	9
92	95	9
88	85	9
75	80	25
97	94	9
91	88	9
84	87	9
83	86	9
79	75	16
MSE		12.9

$$\text{RMSE} = \sqrt{12.9} = 3.59$$

# AGENDA

**01** Multiple Linear Regression

---

**02** Evaluating Regression Models

---

**03** R Exercise

---

# R Exercise I

- Data Set: Toyota Corolla Selling Price



Variable	Description	Variable	Description
Price	Offer Price in EUROS	Guarantee_Period	Guarantee period in months
Age_08_04	Age in months as in August 2004	ABS	Anti-Lock Brake System (Yes=1, No=0)
Mfg_Month	Manufacturing month (1-12)	Airbag_1	Driver_Airbag (Yes=1, No=0)
Mfg_Year	Manufacturing Year	Airbag_2	Passenger Airbag (Yes=1, No=0)
KM	Accumulated Kilometers on odometer	Airco	Airconditioning (Yes=1, No=0)
Fuel_Type	Fuel Type (Petrol, Diesel, CNG)	Automatic_airco	Automatic Airconditioning (Yes=1, No=0)
HP	Horse Power	Boardcomputer	Boardcomputer (Yes=1, No=0)
Met_Color	Metallic Color? (Yes=1, No=0)	CD_Player	CD Player (Yes=1, No=0)
Automatic	Automatic (Yes=1, No=0)	Central_Lock	Central Lock (Yes=1, No=0)
CC	Cylinder Volume in cubic centimeters	Powered_Windows	Powered Windows (Yes=1, No=0)
Doors	Number of doors	Power_Steering	Power Steering (Yes=1, No=0)
Cylinders	Number of cylinders	Radio	Radio (Yes=1, No=0)
Gears	Number of gear positions	Mistlamps	Mistlamps (Yes=1, No=0)
Quarterly_Tax	Quarterly road tax in EUROS	Sport_Model	Sport Model (Yes=1, No=0)
Weight	Weight in Kilograms	Backseat_Divider	Backseat Divider (Yes=1, No=0)
Mfr_Guarantee	Within Manufacturer's Guarantee period (Yes=1, No=0)	Metallic_Rim	Metallic Rim (Yes=1, No=0)
BOVAG_Guarantee	BOVAG (Dutch dealer network) Guarantee (Yes=1, No=0)	Radio_cassette	Radio Cassette (Yes=1, No=0)
		Parking_Assistant	Parking assistance system (Yes=1, No=0)
		Tow_Bar	Tow Bar (Yes=1, No=0)

# R Exercise I

- Define the performance evaluation function

```
# Performance evaluation function for regression -----
perf_eval_reg <- function(tgt_y, pre_y){
  # RMSE
  rmse <- sqrt(mean((tgt_y - pre_y)^2))
  # MAE
  mae <- mean(abs(tgt_y - pre_y))
  # MAPE
  mape <- 100*mean(abs((tgt_y - pre_y)/tgt_y))
  return(c(rmse, mae, mape))
}

# Initialize a performance summary table
perf_mat <- matrix(0, nrow = 2, ncol = 3)
rownames(perf_mat) <- c("Toyota Corolla", "Boston Housing")
colnames(perf_mat) <- c("RMSE", "MAE", "MAPE")
perf_mat
```

✓ perf\_eval\_reg() function

- Arguments: target values & predicted values
- Outputs: RMSE, MAE, MAPE

# R Exercise I

- Load the data

```
# Dataset 1: Toyota Corolla
corolla <- read.csv("ToyotaCorolla.csv")

# Indices for the activated input variables
nCar <- nrow(corolla)
nVar <- ncol(corolla)

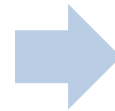
id_idx <- c(1,2)
category_idx <- 8
```

- ✓ `read.csv()`: a function that can read a csv file
- ✓ `nrow()` & `ncol()`: return the number of rows/columns in the dataframe
- ✓ `id_idx`: id-related variables, irrelevant variable for analysis, will be removed
- ✓ `category_idx`: categorical variable, will be transformed by 1-of-C coding

# R Exercise I

- Data preprocessing: I-of-C coding

Price	Age_08_04	Mfg_Month	Mfg_Year	KM	Fuel_Type	HP	Met_Color	Automatic	cc
13500	23	10	2002	46986	Diesel	90	1	0	2000
13750	23	10	2002	72937	Diesel	90	1	0	2000
13950	24	9	2002	41711	Diesel	90	1	0	2000
14950	26	7	2002	48000	Diesel	90	0	0	2000
13750	30	3	2002	38500	Diesel	90	0	0	2000
12950	32	1	2002	61000	Diesel	90	0	0	2000
16900	27	6	2002	94612	Diesel	90	1	0	2000
18600	30	3	2002	75889	Diesel	90	1	0	2000
21500	27	6	2002	19700	Petrol	192	0	0	1800
12950	23	10	2002	71138	Diesel	69	0	0	1900
20950	25	8	2002	31461	Petrol	192	0	0	1800
19950	22	11	2002	43610	Petrol	192	0	0	1800
19600	25	8	2002	32189	Petrol	192	0	0	1800
21500	31	2	2002	23000	Petrol	192	1	0	1800
22500	32	1	2002	34131	Petrol	192	1	0	1800



KM	HP	Met_Color
46986	90	1
72937	90	1
41711	90	1
48000	90	0
38500	90	0
61000	90	0
94612	90	1
75889	90	1
19700	192	0
71138	69	0
31461	192	0
43610	192	0
32189	192	0
23000	192	0
23000	192	1

...

Petrol	Diesel	CNG
0	1	0
0	1	0
0	1	0
0	1	0
0	1	0
0	1	0
0	1	0
0	1	0
0	1	0
1	0	0
0	1	0
1	0	0
1	0	0
1	0	0
1	0	0

✓ Transform one categorical variable to C binary variables

- C is the number of categories

# R Exercise I

- I-of-C Coding process

```
# Transform a categorical variable into a set of binary variables
dummy_p <- rep(0,nCar)
dummy_d <- rep(0,nCar)
dummy_c <- rep(0,nCar)

p_idx <- which(corolla$Fuel_Type == "Petrol")
d_idx <- which(corolla$Fuel_Type == "Diesel")
c_idx <- which(corolla$Fuel_Type == "CNG")

dummy_p[p_idx] <- 1
dummy_d[d_idx] <- 1
dummy_c[c_idx] <- 1
```

- ✓ `dummy_p(c/d)`: initialize a zero vector with length `nCar`
- ✓ `p_idx`: Store the row index with `Fuel_Type == "Petrol"` (do the same job for `d_idx` and `c_idx`)
- ✓ `dummy_p[p_idx] <- 1`: replace 0 by 1 for the rows in the `p_idx`



# R Exercise I

- Combine the dataset and split the data

```
Fuel <- data.frame(dummy_p, dummy_d, dummy_c)
names(Fuel) <- c("Petrol", "Diesel", "CNG")

# Prepare the data for MLR
corolla_mlr_data <- cbind(corolla[, -c(id_idx, category_idx)], Fuel)

# Split the data into the training/validation sets
set.seed(12345)
corolla_trn_idx <- sample(1:nCar, round(0.7*nCar))
corolla_trn_data <- corolla_mlr_data[corolla_trn_idx, ]
corolla_val_data <- corolla_mlr_data[-corolla_trn_idx, ]
```

- ✓ Create a new data frame “Fuel” by combining three dummy variables
- ✓ Combine the dataset with the original corolla dataset and Fuel dataset (use cbind( ) function)
- ✓ Split the data: 70% for training and 30% for validation

# R Exercise I

- Training the model

```
# Train the MLR
mlr_corolla <- lm(Price ~ ., data = corolla_trn_data)
mlr_corolla
summary(mlr_corolla)
plot(mlr_corolla)
```

- ✓ `lm( )`: linear regression

- Price ~ : Formula

- The left side of ~ is the target variable
    - The right side of ~ are the predictor variables (. means all variables except the target variable)
    - data = corolla\_trn\_data: data used to estimate the regression coefficients

- ✓ `Summary( )`: print the result of the regression model

- ✓ `plot( )`: draw four plots for the regression model

# R Exercise I

- Interpret the results
  - ✓ Estimate: estimated regression coefficients
  - ✓ Std.Error: standard error of the estimated coefficients
  - ✓ t value: t-statistic for the hypothesis test
  - ✓  $\Pr(>|t|)$ : p-value for the regression coefficient, the smaller the p-value, the more significant the variable
  - ✓ Adjusted R-squared
  - ✓ NA: variable is removed because of multicollinearity problem

```
> summary(full_model)
```

```
Call:
lm(formula = Price ~ ., data = trn_data)
```

```
Residuals:
    Min       1Q   Median       3Q      Max
-8569.6  -637.3   -42.9   650.5  5720.8
```

```
Coefficients: (3 not defined because of singularities)
              Estimate Std. Error t value Pr(>|t|)
(Intercept)  -752.65641  1775.96778  -0.424  0.671805
Age_08_04     -118.43999    4.29358  -27.585 < 2e-16 ***
Mfg_Month     -95.78658    11.11831   -8.615 < 2e-16 ***
Mfg_Year              NA           NA      NA      NA
KM             -0.01727    0.00136  -12.702 < 2e-16 ***
HP             20.46048    3.59449    5.692 1.66e-08 ***
Met_Color     -64.93066    81.74804   -0.794 0.427228
Automatic     338.02084   156.47529    2.160 0.031000 *
cc            -0.10770    0.07958   -1.353 0.176246
Doors          10.46213    43.60079    0.240 0.810418
Cylinders              NA           NA      NA      NA
Gears         183.36459   196.28703    0.934 0.350451
```

•  
•  
•

```
Tow_Bar       -217.67837    85.11397   -2.557 0.010694 *
Petrol        2280.57458   387.15749    5.891 5.30e-09 ***
Diesel        1004.02078   377.75296    2.658 0.007993 **
CNG              NA           NA      NA      NA
```

```
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

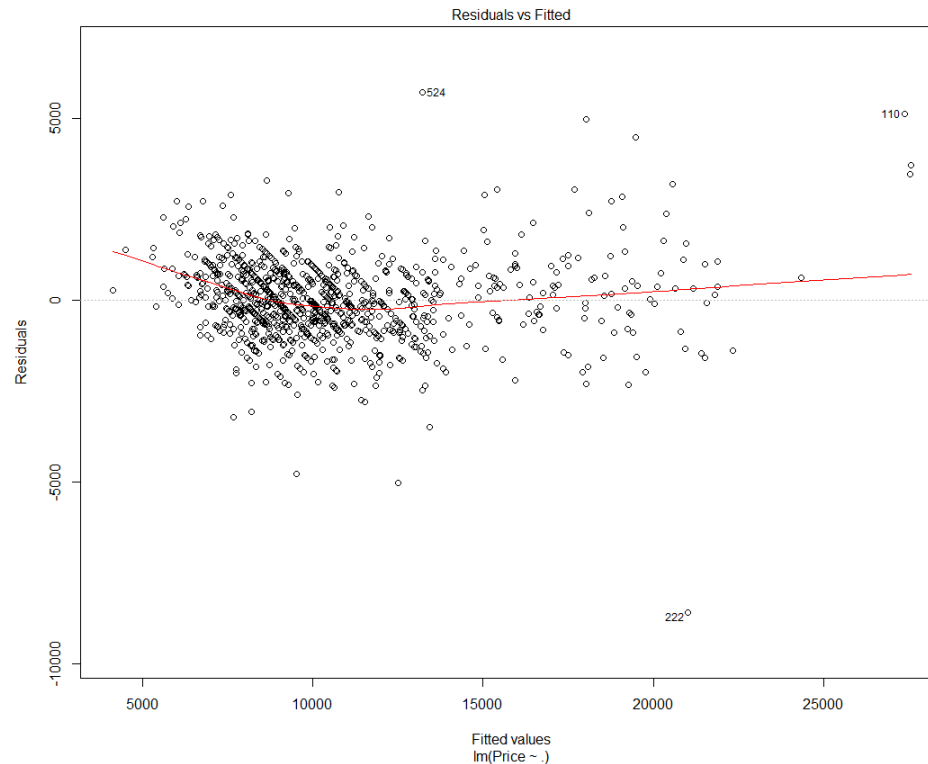
```
Residual standard error: 1128 on 971 degrees of freedom
Multiple R-squared:  0.9046,    Adjusted R-squared:  0.9014
F-statistic: 279.1 on 33 and 971 DF,  p-value: < 2.2e-16
```

# R Exercise I

- Interpret the result

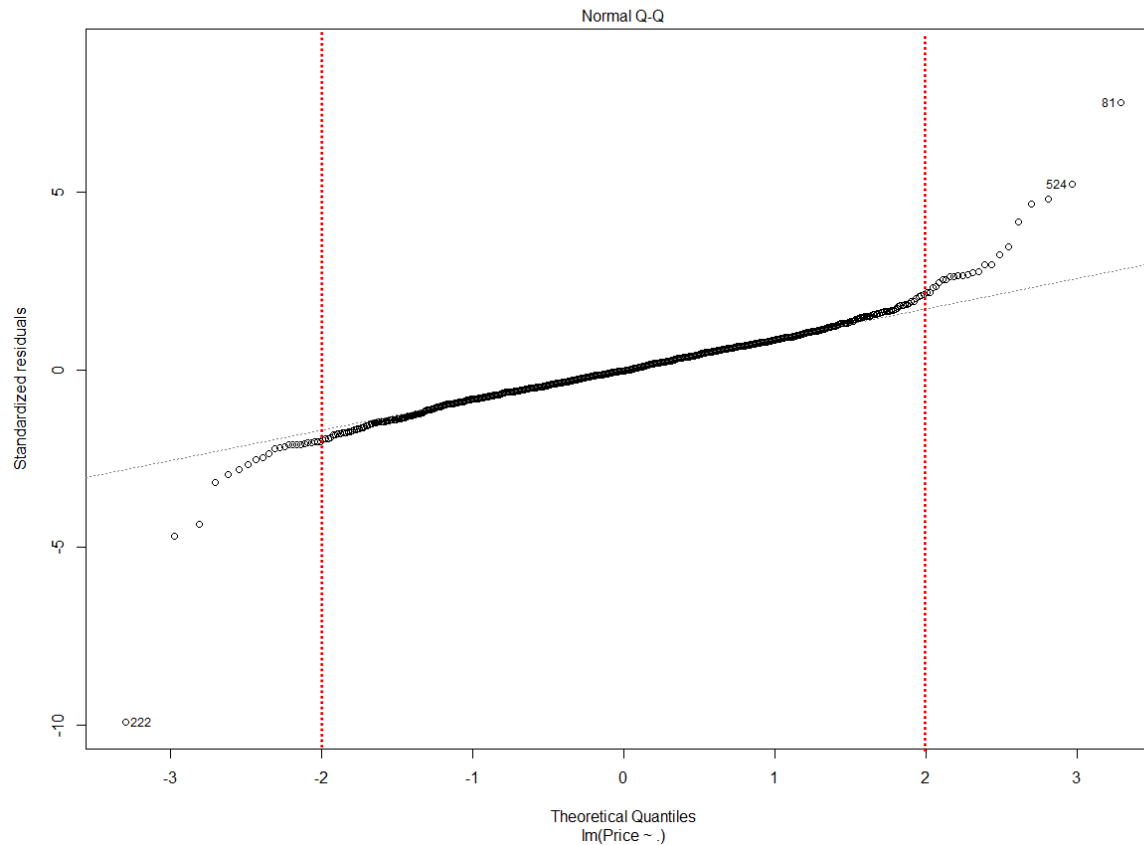
- ✓ Figure 1: used to check the following assumption

- The variability in Y values for a given set of predictors is the same regardless of the values of the predictors (homoskedasticity)



# R Exercise I

- Interpret the result
  - ✓ Figure 2: used to check the following assumption
    - The noise  $\varepsilon$  follows a normal distribution

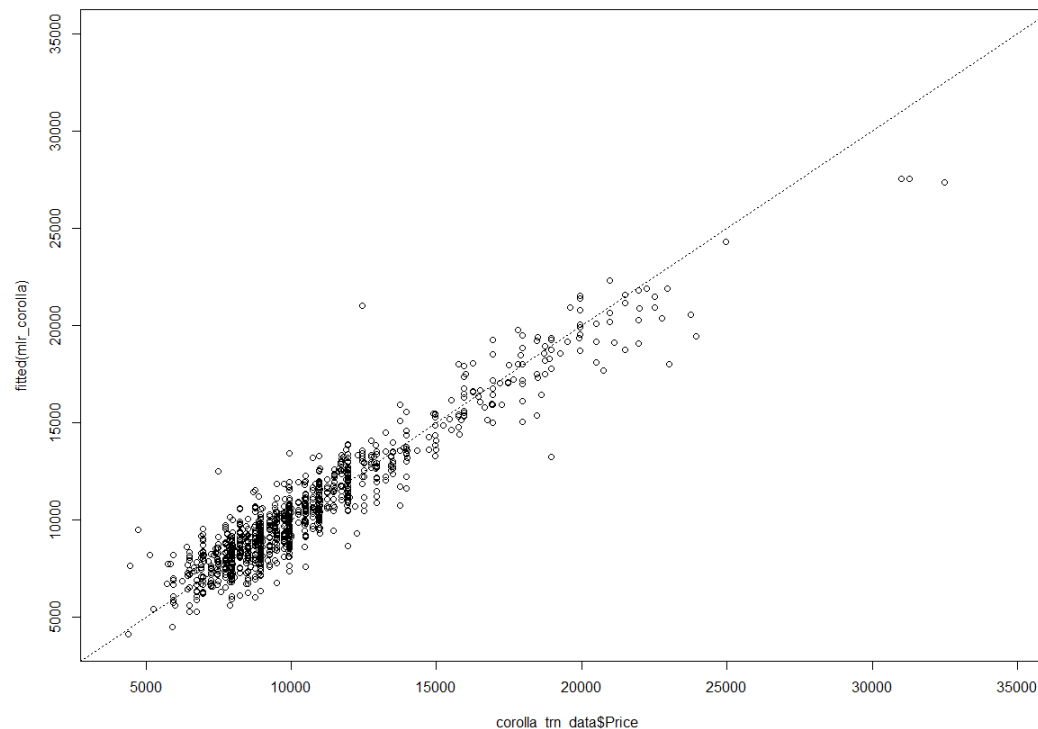


# R Exercise I

- Interpret the result

```
# Plot the result
plot(corolla_trn_data$Price, fitted(mlr_corolla), xlim = c(4000,35000),
     ylim = c(4000,35000))
abline(0,1,lty=3)
```

- ✓ Plot the relationship between the actual target values (x-axis) and the predicted values (y-axis)



# R Exercise I

- Normality check for the residuals

```
# normality test of residuals
corolla_resid <- resid(mlr_corolla)
m <- mean(corolla_resid)
std <- sqrt(var(corolla_resid))

hist(corolla_resid, density=20, breaks=50, prob=TRUE, xlab="x-variable",
      main="normal curve over histogram")
curve(dnorm(x, mean=m, sd=std), col="darkblue", lwd=2, add=TRUE, yaxt="n")

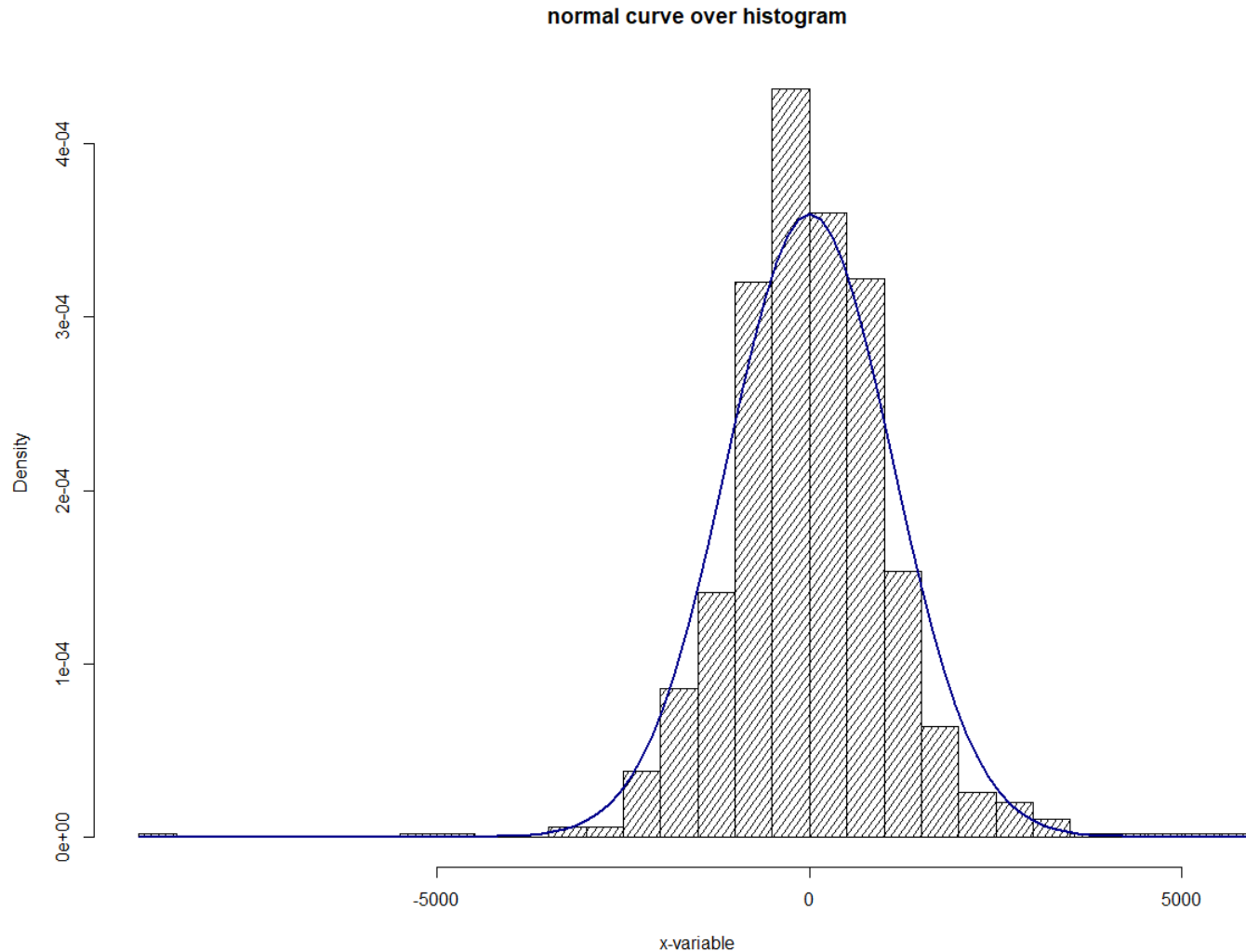
skewness(corolla_resid)
kurtosis(corolla_resid)
```

- ✓ `hist( )`: draw a histogram
- ✓ `curve( )`: draw a curve for a certain probability density function
- ✓ skewness: 0 if the dataset follows a normal distribution
- ✓ kurtosis: 3 if the dataset follows a normal distribution

```
> skewness(corolla_resid)
[1] -0.1355675
> kurtosis(corolla_resid)
[1] 8.630819
```

# R Exercise I

- Histogram & Normal Density Curve





# R Exercise I

- Prediction performance of the regression model

```
# Performance Measure
mlr_corolla_haty <- predict(mlr_corolla, newdata = corolla_val_data)
perf_mat[1,] <- perf_eval_reg(corolla_val_data$Price, mlr_corolla_haty)
perf_mat
```

```
> perf_mat
```

	RMSE	MAE	MAPE
Toyota Corolla	1085.124	814.0598	8.007297
Boston Housing	0.000	0.0000	0.000000

# R Exercise 2



- Boston Housing Price

✓ Predict the median price of houses in a unit district

Variable	Description
CRIM	per capita crime rate by town.
ZN	proportion of residential land zoned for lots over 25,000 sq.ft.
INDUS	proportion of non-retail business acres per town.
NOX	nitrogen oxides concentration (parts per 10 million).
RM	average number of rooms per dwelling.
AGE	proportion of owner-occupied units built prior to 1940.
DIS	weighted mean of distances to five Boston employment centres.
TAX	full-value property-tax rate per \ \$10,000.
PTRATIO	pupil-teacher ratio by town.
Black	$1000(B_k - 0.63)^2$ where $B_k$ is the proportion of blacks by town.
LSTAT	lower status of the population (percent).
MEDV	median value of owner-occupied homes in \ \$1000s.

# R Exercise 2

- Load the dataset and preprocess the data

```
# Dataset 2: Boston Housing
boston_housing <- read.csv("BostonHousing.csv")
nHome <- nrow(boston_housing)
nVar <- ncol(boston_housing)

# Split the data into the training/validation sets
boston_trn_idx <- sample(1:nHome, round(0.7*nHome))
boston_trn_data <- boston_housing[boston_trn_idx,]
boston_val_data <- boston_housing[-boston_trn_idx,]
```

- ✓ Unlike “corolla” dataset, all variables are numerical variables
- ✓ No special data preprocessing is required

# R Exercise 2

- Training the model and plot the results

```
# Train the MLR
mlr_boston <- lm(MEDV ~ ., data = boston_trn_data)
mlr_boston

summary(mlr_boston)
plot(mlr_boston)

# Plot the result
plot(boston_trn_data$MEDV, fitted(mlr_boston), xlim = c(-5,50), ylim = c(-5,50))
abline(0,1,lty=3)
```

# R Exercise 2

- Fitted model

- ✓ Adjusted  $R^2$ : 0.7347 (smaller than that of corolla model)

- ✓ All variables except INDUS, AGE, and TAX are statistically significant ( $\alpha = 0.05$ )

```
> summary(mlr_boston)
```

```
Call:
```

```
lm(formula = MEDV ~ ., data = boston_trn_data)
```

```
Residuals:
```

Min	1Q	Median	3Q	Max
-13.7098	-2.6401	-0.5686	1.3348	27.3746

```
Coefficients:
```

	Estimate	Std. Error	t value	Pr(> t )	
(Intercept)	2.779e+01	5.744e+00	4.839	1.98e-06	***
CRIM	-7.268e-02	3.435e-02	-2.115	0.0351	*
ZN	3.485e-02	1.710e-02	2.038	0.0423	*
INDUS	-8.059e-03	6.755e-02	-0.119	0.9051	
NOX	-1.218e+01	4.643e+00	-2.623	0.0091	**
RM	4.287e+00	4.920e-01	8.714	< 2e-16	***
AGE	-2.460e-02	1.492e-02	-1.649	0.1000	
DIS	-1.573e+00	2.456e-01	-6.404	5.00e-10	***
TAX	7.515e-04	2.755e-03	0.273	0.7852	
PTRATIO	-8.879e-01	1.495e-01	-5.941	6.98e-09	***
B	1.237e-02	3.067e-03	4.034	6.78e-05	***
LSTAT	-4.910e-01	6.139e-02	-7.999	1.97e-14	***

```
---
```

```
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

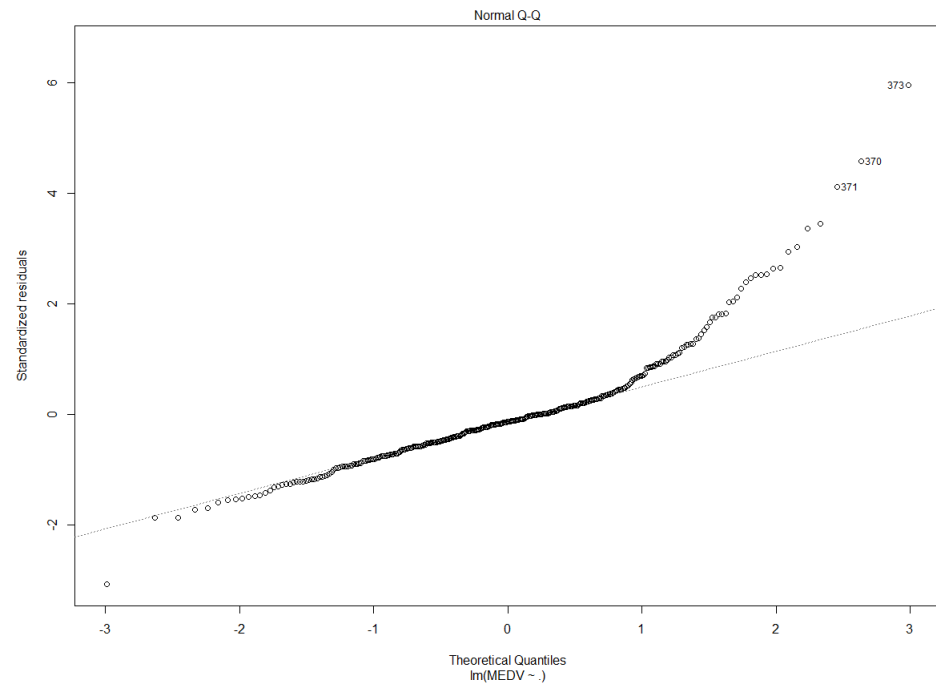
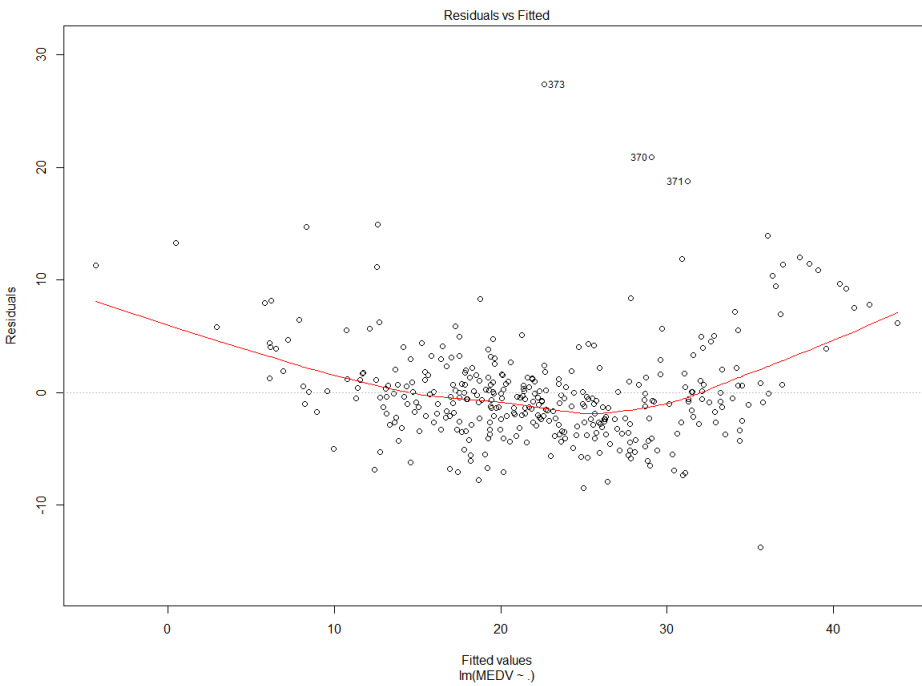
```
Residual standard error: 4.658 on 342 degrees of freedom
```

```
Multiple R-squared:  0.743,    Adjusted R-squared:  0.7347
```

```
F-statistic: 89.89 on 11 and 342 DF,  p-value: < 2.2e-16
```

# R Exercise 2

- Residual plot and normal QQ plot



✓ Residuals might not follow a normal distribution

# R Exercise 2

- Normality check for the residuals

```
# normality test of residuals
boston_resid <- resid(mlr_boston)
m <- mean(boston_resid)
std <- sqrt(var(boston_resid))

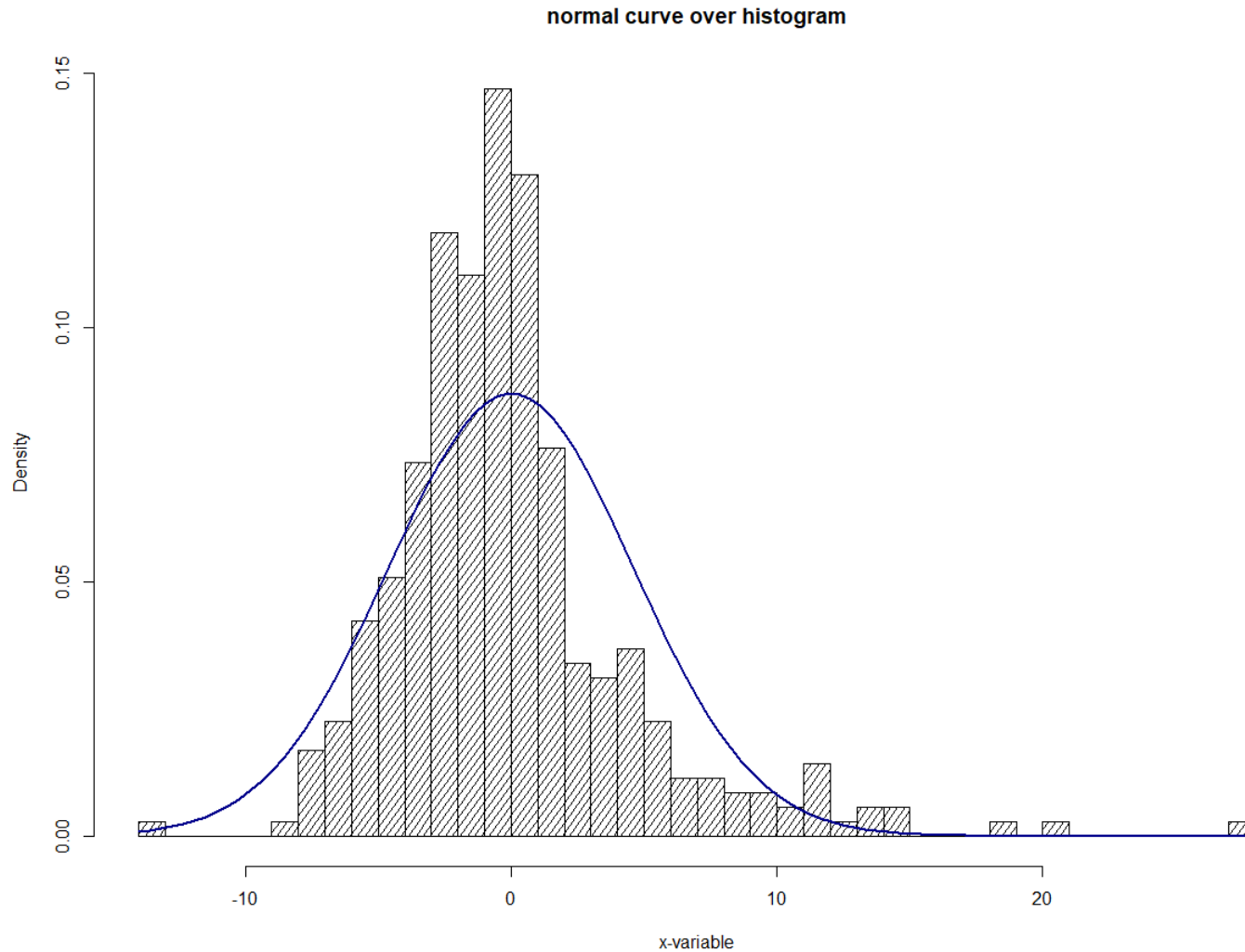
hist(boston_resid, density=20, breaks=50, prob=TRUE, xlab="x-variable",
      main="normal curve over histogram")
curve(dnorm(x, mean=m, sd=std), col="darkblue", lwd=2, add=TRUE, yaxt="n")

skewness(boston_resid)
kurtosis(boston_resid)
```

```
> skewness(boston_resid)
[1] 1.656679
> kurtosis(boston_resid)
[1] 8.669806
```

# R Exercise 2

- Normality check for the residuals





# R Exercise 2

- Prediction performance of the regression model

```
# Performance Measure
mlr_boston_haty <- predict(mlr_boston, newdata = boston_val_data)
perf_mat[2,] <- perf_eval_reg(boston_val_data$MEDV, mlr_boston_haty)
perf_mat
```

```
> perf_mat
```

	RMSE	MAE	MAPE
Toyota Corolla	1085.124496	814.059765	8.007297
Boston Housing	5.476226	3.834064	19.276196

