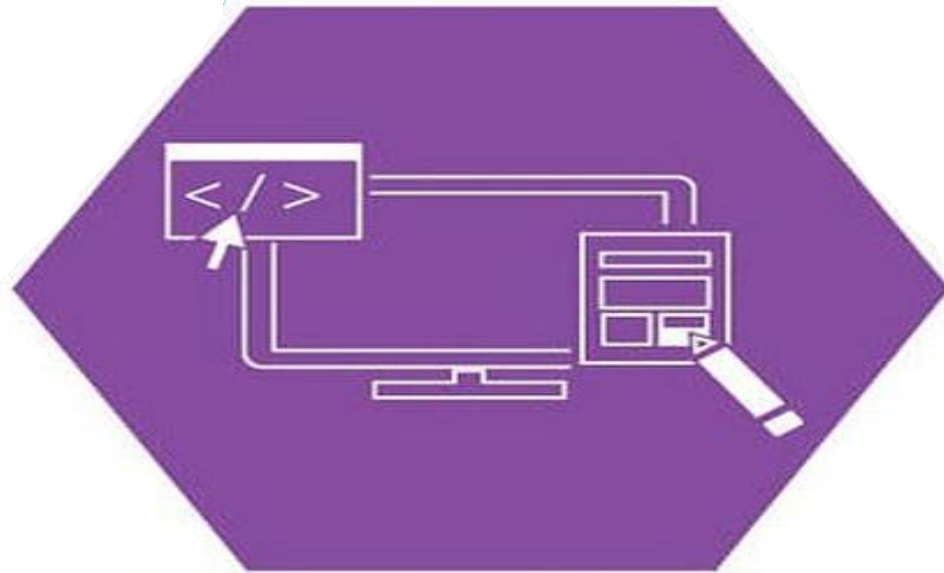


# 4243/5243: Applied Data Science

## Lecture 02: Data Pre-Processing & Feature Engineering

# Data Pre-Processing

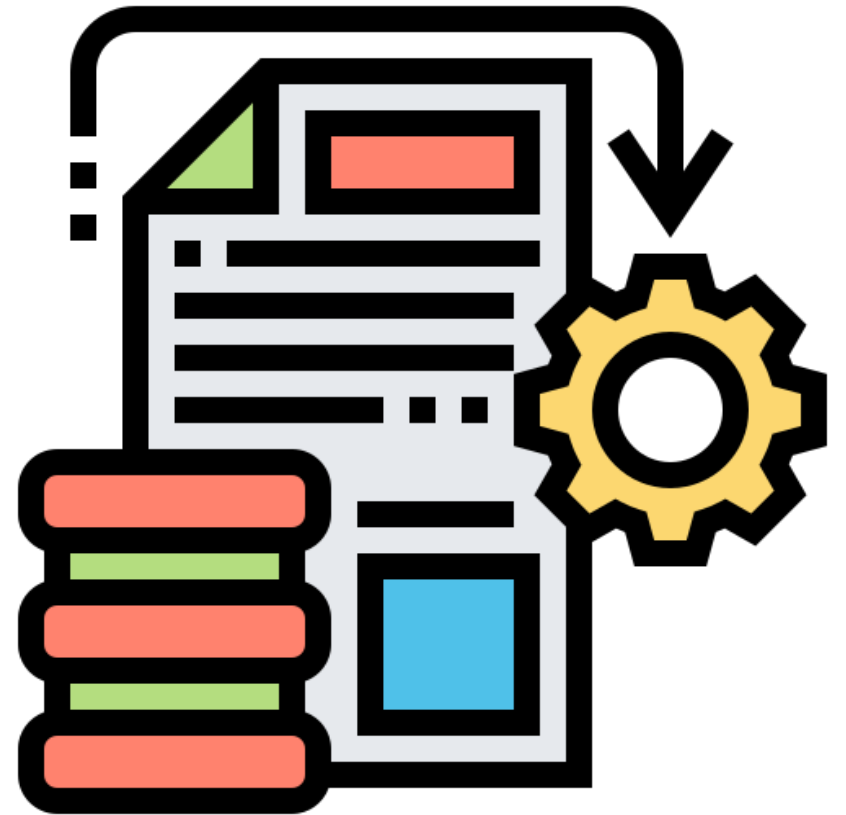
2



Clean, Prepare  
& Manipulate Data

# Data Pre-Processing

- Data pre-processing and engineering techniques generally refer to the addition, deletion, or transformation of data.
- The time spent on identifying data engineering needs can be significant and requires you to spend substantial time understanding your data.



# Data Pre-Processing

- The need for data pre-processing is determined by the type of model being used. Different models have different **sensitivities** to the type of predictors in the model.
- For instance, some methods, such as tree-based models, are notably insensitive to the characteristics of the predictor data. Other, like linear regression, are not.
- In this class, a wide array of possible transformation techniques will be discussed. We will also discuss which, if any, pre-processing techniques can be useful.

# Data Cleaning



# Data Inconsistencies

- **Inconsistent data** can arise during data collection, integration, or entry, and may lead to errors in analysis, misleading results, or failure of machine learning models.
- Ensuring **consistency** in the dataset is a vital step in data preprocessing.
- Data inconsistency occurs when the same information is represented in multiple ways or when relationships within the data do not align with expected rules.

# Data Inconsistencies

## Formatting Issues:

- Dates represented in different formats (e.g., “2025 – 01 - 28”, “01/28/2025”).
- Currency values stored inconsistently (e.g., “\$100”, “100 USD”).
- Text inconsistencies due to capitalization, spelling, or abbreviations (e.g., “New York”, “new York”, “NY”).

# Data Inconsistencies

## Categorical Data Issues:

- Different labels for the same category (e.g., “Male”, “M”).
- Categories with spelling errors or typos (e.g., “Fmale” instead of “Female”).



# Data Inconsistencies

- **Duplicate Records**: same entity recorded multiple times with slight variations (e.g., customer names: “A. Pijyan” and “Alex Pijyan”).
- **Negative values for quantities that cannot be negative** (e.g., age, weight).
- **Numeric columns stored as strings** (e.g., “100” stores as text).
- **Mixed data types in a single column** (e.g., text and numbers).

# Resolving Data Inconsistencies

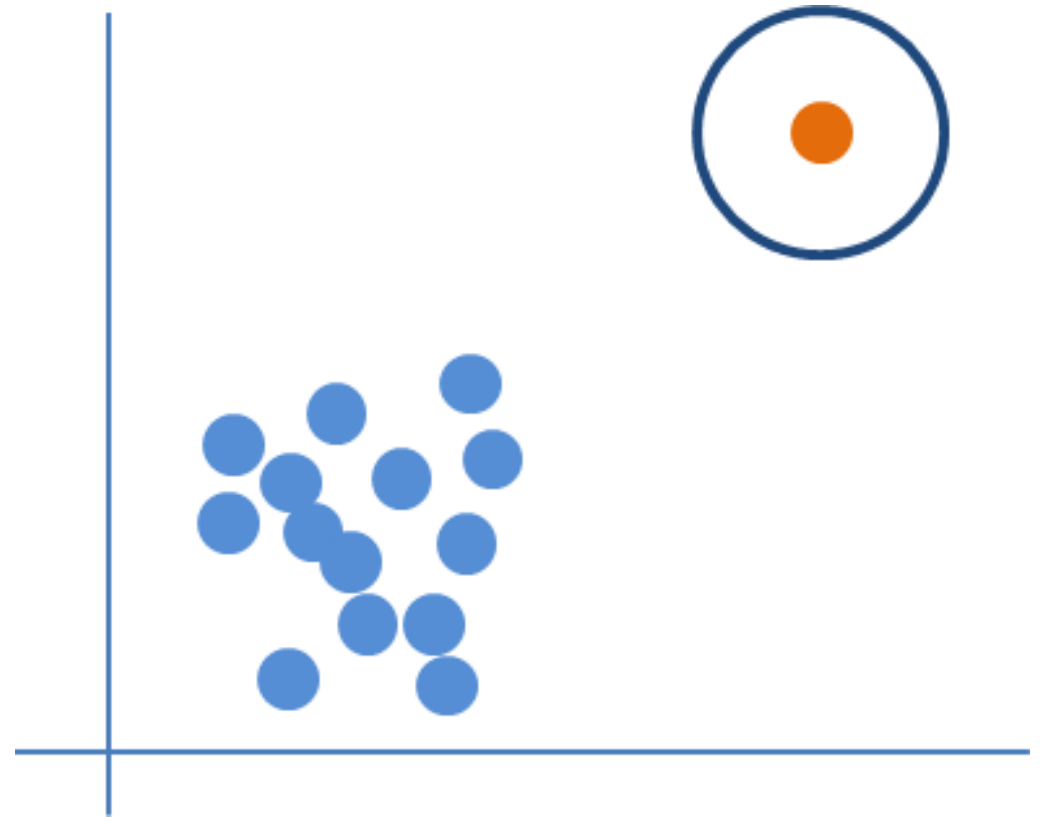
- **Dates** - use a consistent date format across the dataset (e.g., “MM-DD-YYYY”). Tools: ***lubridate*** in R or **pandas.to\_datetime()** in Python.
- **Text** – standardize capitalization and remove unnecessary whitespace. Example: convert all text to lowercase (**.str.lower()** in Python or **tolower()** in R).
- **Map Values** – replace inconsistent labels with standardized ones (e.g., map “Male” to “M”). Tools: use **fuzzywuzzy** (Python) or **stringdist** (R) to handle typos and close matches.

# Resolving Data Inconsistencies

- **Duplicate Records** – use unique identifiers (e.g., customer ID) to identify duplicates. Remove duplicates or merge records if necessary.
- **Consistent data type** – convert columns to appropriate data types.

# Outliers & Influential Points

- **Outliers** are data points that deviate significantly from the majority of the dataset.
- They may be caused by errors, rare events, or natural variability.

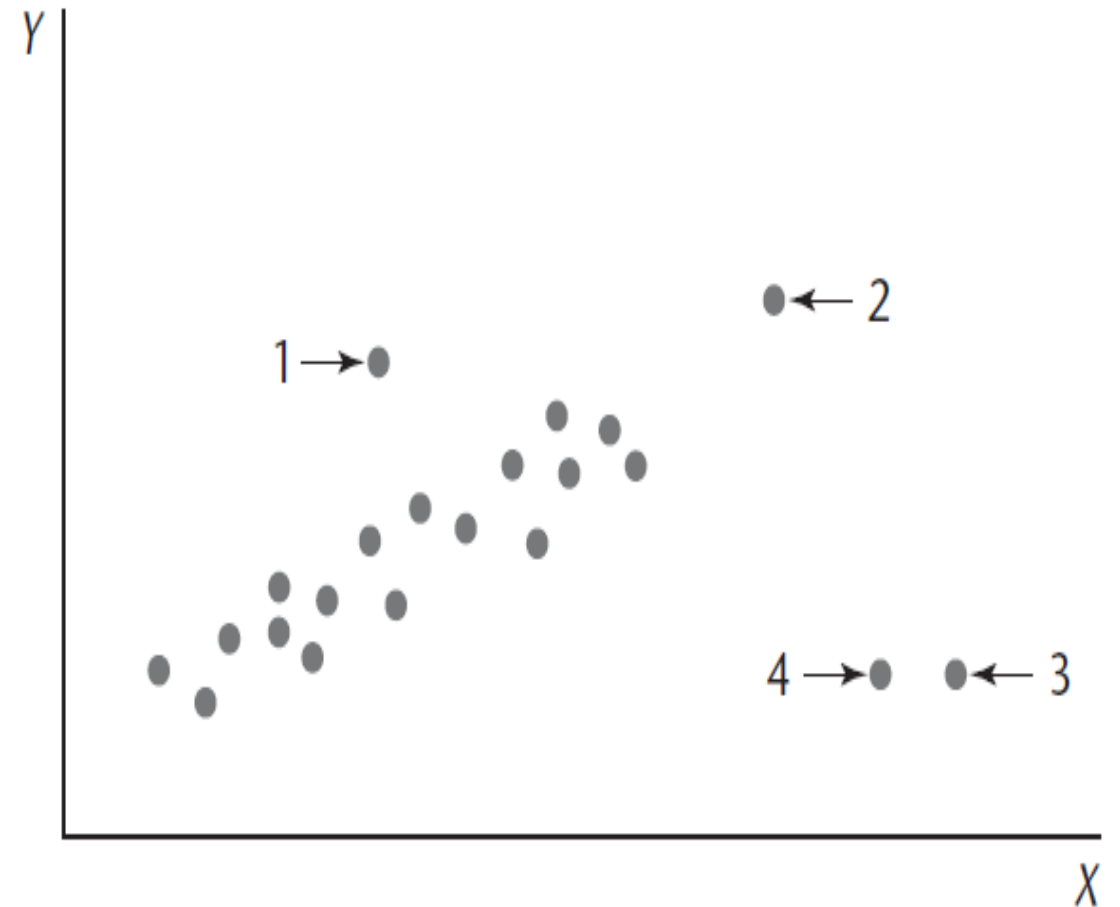


# Outliers & Influential Points

- Outliers and influential points are critical concepts in data preprocessing and feature engineering.
- They can significantly affect analysis results. For instance, outliers can skew means, variances, and regression coefficients; many models are sensitive to outliers; outliers may indicate rare events or errors that require special attention.
- Thus, understating how to detect and handle them is essential.

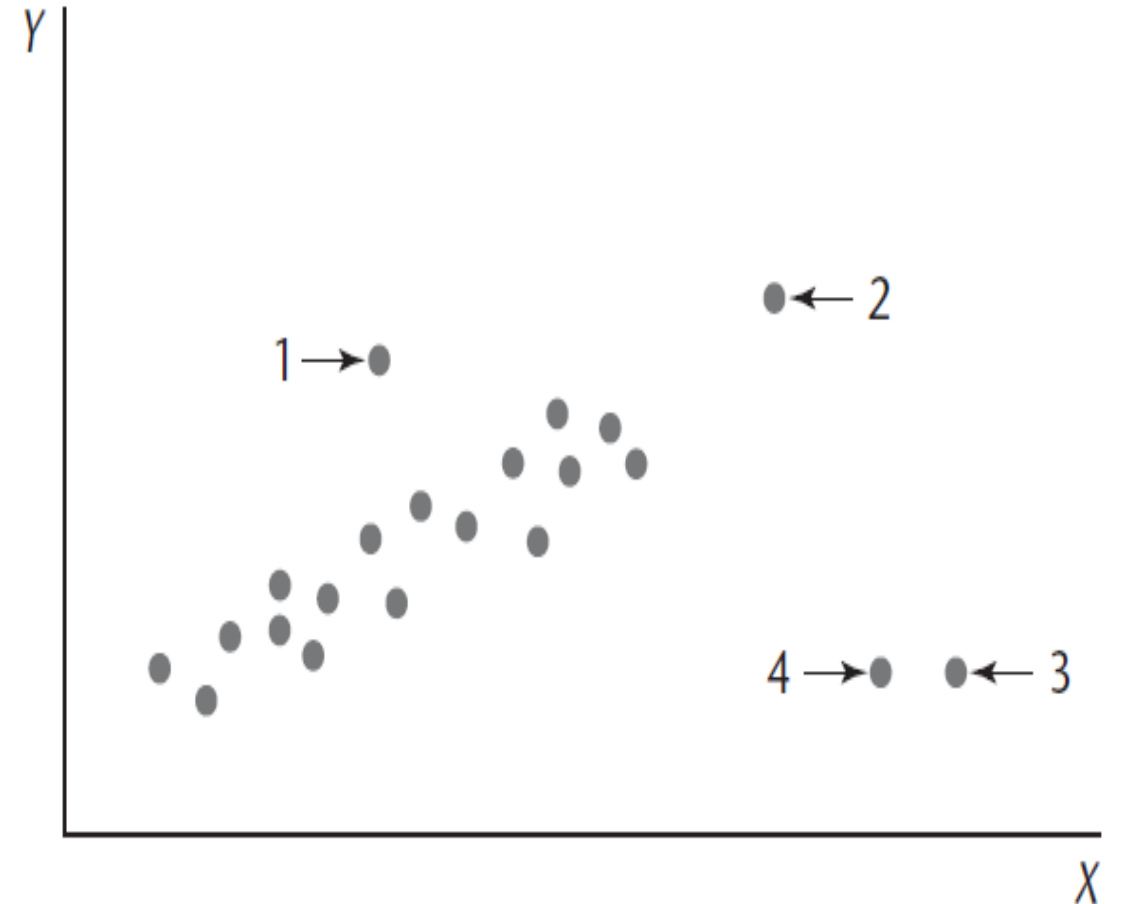
# Outliers & Influential Points

- An observation may be outlying or extreme with respect to its  $Y$  [response] value, its  $X$  [predictor(s)] values(s), or both.



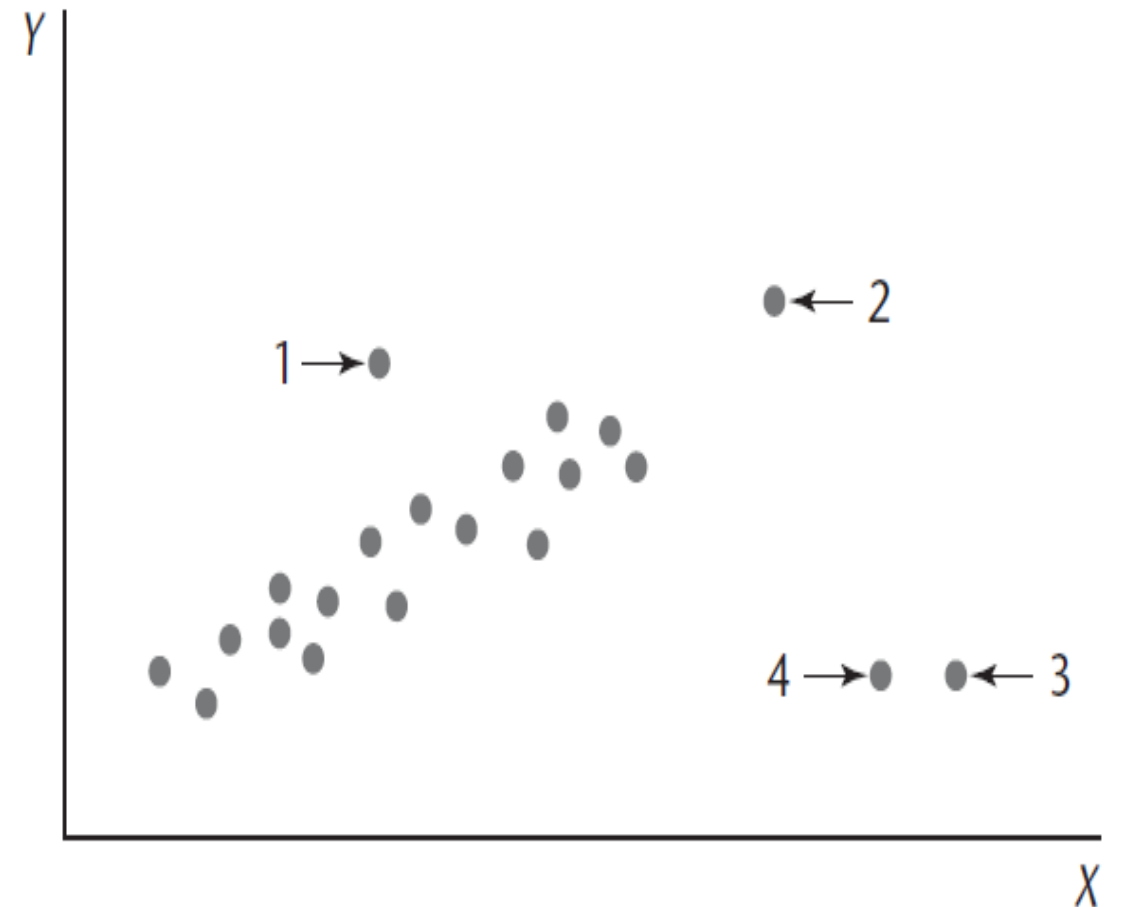
# Outliers & Influential Points

- Not all outlying cases have a strong influence on the fitted regression function.
- For instance, Case 1 may not be too influential because a number of other cases have similar  $X$  values that will keep the fitted regression function from being displaced too far by the outlying case.



# Outliers & Influential Points

- Case 2 may not be too influential because its  $Y$  value is consistent with the regression relation displayed by the nonextreme cases.
- Cases 3 and 4, on the other hand, are likely to be very influential in affecting the fit of the regression function: they are outlying with regard to their  $X$  values, and their  $Y$  values are not consistent with the regression relation.





# Outliers: Studentized Residuals

- The detection of outlying or extreme  $Y$  observations based on an examination of the residuals. We utilized either the residual  $e_i$

$$e_i = Y_i - \hat{Y}_i$$

or the studentized residuals  $e_i^*$ :

$$e_i^* = \frac{e_i}{\sqrt{MSE}}$$

# Outliers: Studentized Residuals

- Recall, the hat matrix is defined as:

$$\mathbf{H} = \mathbf{X}(\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}'$$

- The fitted values  $\hat{Y}_i$  can be expressed as:

$$\hat{\mathbf{Y}} = \mathbf{H}\mathbf{Y}$$

- The residuals  $e_i$  can be expressed as

$$\mathbf{e} = (\mathbf{I} - \mathbf{H})\mathbf{Y}$$

# Outliers: Studentized Residuals

- The variance-covariance of residuals can be expressed as

$$\sigma^2(\mathbf{e}) = \sigma^2(\mathbf{I} - \mathbf{H})$$

- Thus, the variance of residual  $e_i$  is

$$\sigma^2(e_i) = \sigma^2(1 - h_{ii})$$

- And the estimate

$$s^2(e_i) = MSE(1 - h_{ii})$$

# Outliers: Studentized Residuals

- The test statistic for the studentized deleted residual test for detecting outlying cases with respect to  $Y$  can be computed as

$$t_i = e_i^* \left[ \frac{n - p - 1}{SSE(1 - h_{ii}) - e_i^{*2}} \right]^{1/2} \sim t(n - p - 1)$$

- If the  $i$ -th observation has corresponding  $|t_i| > t(1 - \frac{\alpha}{2n}; n - p - 1)$ , then we identify this observation as an outlier with respect to  $Y$ .

# Outliers: Leverage Values

- As we saw, the hat matrix plays an important role in determining the magnitude of the studentized deleted residual and therefore in identifying outlying  $Y$  observations.
- The hat matrix is also helpful in directly identifying outlying  $X$  observations.
- In particular, the diagonal elements of the hat matrix are a useful indicator in a multivariable setting of whether a case is outlying with respect to its  $X$  values.

# Outliers: Leverage Values

- The diagonal elements  $h_{ii}$  of the hat matrix are always between 0 and 1 and their sum is  $p$ :

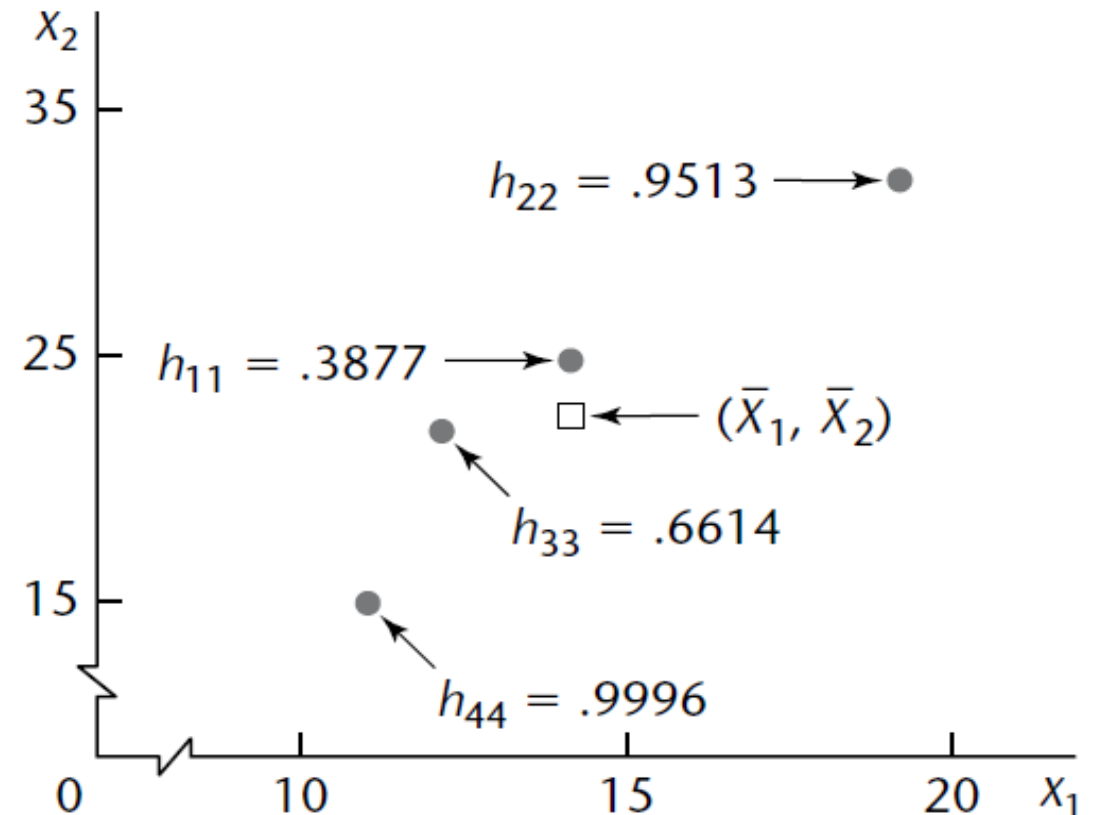
$$0 \leq h_{ii} \leq 1$$

$$\sum_{i=1}^n h_{ii} = p$$

- In addition,  $h_{ii}$  is a measure of the distance between the  $X$  values for the  $i$ th case and the means of the  $X$  values for all  $n$  cases.
- Thus, a large value of  $h_{ii}$  indicates that the  $i$ th case is distant from the center of all  $X$  observations.

# Outliers: Leverage Values

- The diagonal element  $h_{ii}$  in this context is called the **leverage** of the  $i$ th case.
- The figure on the right illustrates the role of the leverage values  $h_{ii}$  as distance measures.



# Outliers: Leverage Values

- A leverage value  $h_{ii}$  is usually considered to be large if it is more than twice as large as the mean leverage value, denoted by  $\bar{h}$ :

$$\bar{h} = \frac{\sum_{i=1}^n h_{ii}}{n} = \frac{p}{n}$$

- Hence, leverage values greater than  $2p/n$  are considered by this rule to indicate outlying cases with regard to their  $X$  values.
- Another suggested guideline is that  $h_{ii}$  values exceeding 0.5 indicate very high leverage, whereas those between 0.2 and 0.5 indicate moderate leverage.



# Influential Values: DFFITS

- A useful measure of the influence that case  $i$  has on the fitted value  $\hat{Y}_i$  is given by:

$$(DFFITS)_i = \frac{\hat{Y}_i - \hat{Y}_{i(i)}}{\sqrt{MSE_{(i)} h_{ii}}}$$

- DF stands for the difference between the fitted value  $\hat{Y}_i$  for the  $i$  case when all  $n$  cases are used in fitting the regression function and the predicted value  $\hat{Y}_{i(i)}$  for the  $i$ th case obtained when the  $i$ th case is omitted in fitting the regression function.
- The denominator is the estimated standard deviation of  $\hat{Y}_i$ .

# Influential Values: DFFITS

- It can be shown that the *DFFITS* values can be computed by using only the results from fitting the entire data set:

$$(DFFITS)_i = e_i \left[ \frac{n - p - 1}{SSE(1 - h_{ii}) - e_i^2} \right]^{1/2} \left( \frac{h_{ii}}{1 - h_{ii}} \right)^{1/2} = t_i \left( \frac{h_{ii}}{1 - h_{ii}} \right)^{1/2}$$

- As a guideline for identifying influential cases, we suggest considering a case influential if the absolute value of *DFFITS* exceeds 1 for small to medium data sets and  $2\sqrt{p/n}$  for large data sets.

# Influential Values: Cook's Distance

- In contrast to the *DFFITS* measure, Cook's distance measure considers the influence of  $i$ th case on all  $n$  fitted values. Cook's distance measure, denoted by  $D_i$ , is an aggregate influence measure, showing the effect on the  $i$ th case on all  $n$  fitted values:

$$D_i = \frac{\sum_{j=1}^n (\hat{Y}_j - \hat{Y}_{j(i)})^2}{pMSE}$$

- For interpreting Cook's distance measure, it has been found useful to relate  $D_i$  to the  $F(p, n - p)$  distribution and ascertain the corresponding percentile value. If the percentile value is less than about 10 or 20 percent, the  $i$ th case has little apparent influence on the fitted value; if it's more than 50 percent, then it is considered to be substantially influential.

# Influential Values: Cook's Distance

- Cook's distance measure  $D_i$  can be calculated without fitting a new regression function each time a different case is deleted. An algebraically equivalent expression is:

$$D_i = \frac{e_i^2}{pMSE} \left[ \frac{h_{ii}}{(1 - h_{ii})^2} \right]$$

- Thus,  $D_i$  depends mainly on two factors: the size of the residual  $e_i$  and the leverage value  $h_{ii}$ .
- The larger either of  $e_i$  or  $h_{ii}$  is, the larger  $D_i$  is.

# Outliers: Remedial Measures

- Do Nothing: if the outliers represent meaningful phenomena, retain them.
- Winsorization: Replace extreme values with a threshold (e.g. 95<sup>th</sup> percentile).
- Transformation: Apply log, square root, or power transformation to reduce the impact.

# Outliers: Remedial Measures

- Capping: Set upper and lower limits for the values.
- Removal: Exclude the outliers, especially if they result from errors.
- Segmentation: Analyze outliers separately if they represent a distinct group.

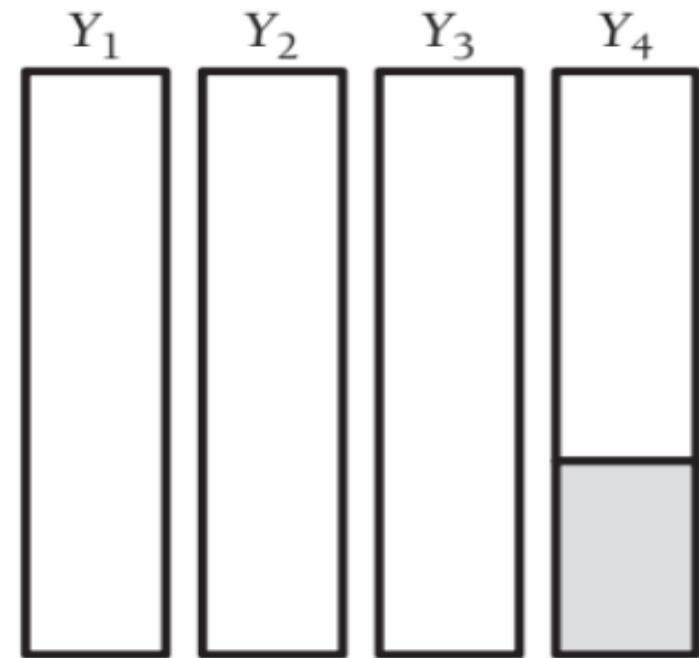
# Missing Values

- **Data quality** is an important issue for any project involving analyzing data. One of the most common data quality concerns you will run into is **missing values**.
- Data can be missing for various reasons; as a starting point, it is useful to understand different **missing data patterns**, which refer to the configuration of observed and missing values in a data set.

# Missing Data Patterns: Univariate

- A **univariate pattern** has missing values isolated to a single variable.
- A univariate pattern is relatively rare in some disciplines but can arise in experimental studies.
- For instance, suppose that  $Y_1$  through  $Y_3$  are manipulated variables (e.g., between subjects' factors in an ANOVA design) and  $Y_4$  is the incomplete outcome variable.

(A) Univariate Pattern

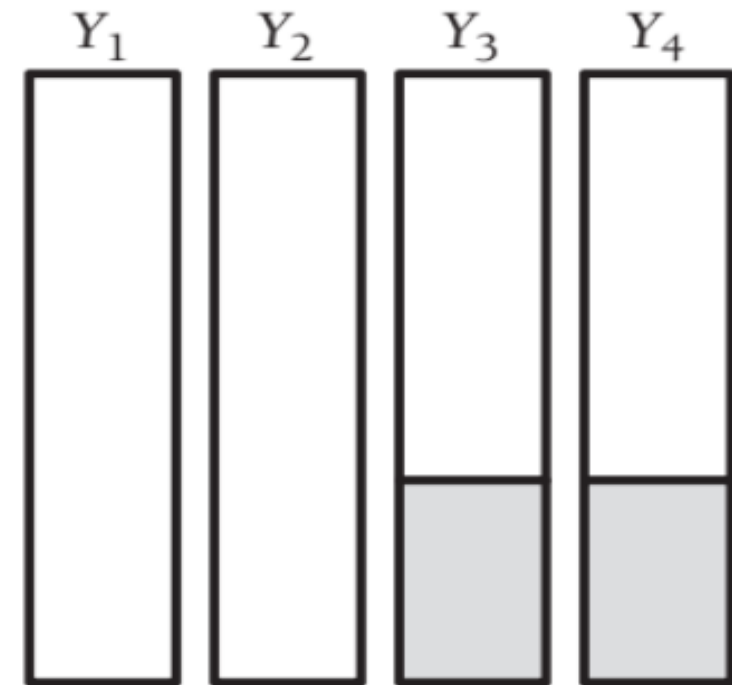




# Missing Data Patterns: Unit Nonresponse

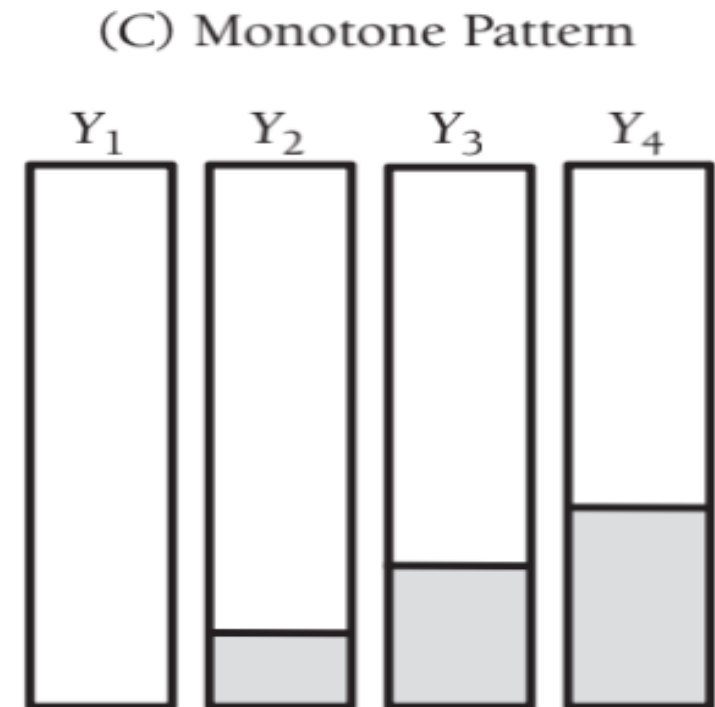
- A **unit nonresponse pattern** often occurs in survey research, where  $Y_1$  and  $Y_2$  are characteristics available for every member of the sampling frame, and  $Y_3$  and  $Y_4$  are surveys that some respondents refuse to answer.

(B) Unit Nonresponse Pattern



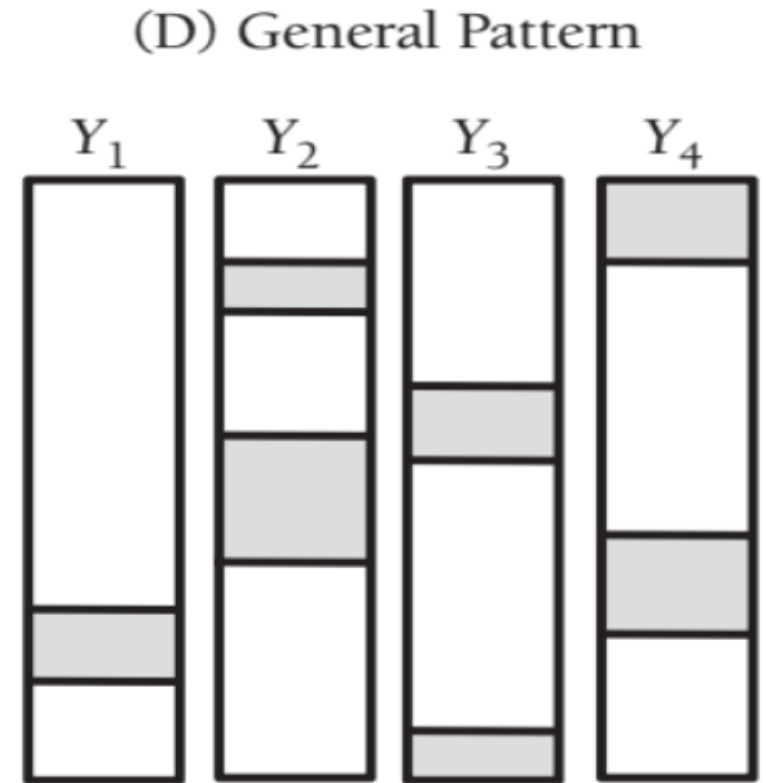
# Missing Data Patterns: Monotone Missing

- A monotone missing data pattern is typically associated with a longitudinal study where participants drop out and never return.
- For example, consider a clinical trial for a new medication in which participants quit the study because they are having adverse reactions to the drug.



# Missing Data Patterns: General

- A **general missing data pattern** is perhaps the most common configuration of missing values.
- A general pattern has missing values dispersed throughout the data matrix in a random fashion.
- The seemingly random pattern is deceptive because the values can still be systematically missing.



# Missing Data Mechanisms

- Missing values can also be described using missing data mechanisms.
- Missing data mechanisms describe possible relationships between measured variables and the probability of missing data.
- There are three commonly used missing data mechanisms that we are going to consider: Missing at Random (MAR), Missing Completely at Random (MCAR), and Missing Not at Random (MNAR).

# Missing at Random (MAR)

- Data is missing at random (MAR) when the probability of missing data on a variable  $Y$  is related to some other measured variable(s) in the model but not to the values of  $Y$  itself.
- The term missing at random is somewhat misleading because it implies that the data is missing in a random fashion.
- However, MAR actually means that a systematic relationship exists between one or more measured variables and the probability of missing data.

# Missing at Random (MAR)

- To illustrate, consider an employee selection scenario in which prospective employees complete an IQ test during their job interview and a supervisor subsequently evaluates their job performance following a 6-month probationary period.
- Suppose that the company used IQ scores as a selection measure and did not hire applicants that scored in the lower quartile of the IQ distribution.
- Thus, the probability of a missing job performance rating is solely a function of IQ scores and is unrelated to an individual's job performance.

# Missing at Random (MAR)

IQ	Job performance ratings			
	Complete	MCAR	MAR	MNAR
78	9	—	—	9
84	13	13	—	13
84	10	—	—	10
85	8	8	—	—
87	7	7	—	—
91	7	7	7	—
92	9	9	9	9
94	9	9	9	9
94	11	11	11	11
96	7	—	7	—
99	7	7	7	—
105	10	10	10	10
105	11	11	11	11
106	15	15	15	15
108	10	10	10	10
112	10	—	10	10
113	12	12	12	12
115	14	14	14	14
118	16	16	16	16
134	12	—	12	12

# Missing Completely at Random (MCAR)

- The missing completely at random (MCAR) mechanism is what researchers think of as purely random missingness.
- The formal definition of MCAR requires that the probability of missing data on a variable  $Y$  is unrelated to other measured variables and is unrelated to the values of  $Y$  itself.
- With regard to the job performance data, you can create the MCAR data by deleting scores based on the values of a random number.



# Missing Not at Random (MNAR)

- Finally, data is missing not at random (**MNAR**) when the probability of missing data on a variable  $Y$  is related to the values of  $Y$  itself, even after controlling for other variables.
- For instance, suppose the company hired all 20 applicants and subsequently terminated a number of individuals for poor performance prior to their 6-month evaluation.
- You can see that the job performance ratings are missing for the applicants with the lowest job performance ratings.

# Dealing with Missingness: Imputation

- **Imputation** is the process of replacing a missing value with a substituted, “Best Guess” value.
- Imputation should be one of the first feature engineering steps you take as it will affect any downstream pre-processing.
- In this class we will consider two imputation techniques: **estimated statistic** and **K- nearest neighbor**.

# Imputation: Estimated Statistic

- An elementary approach to imputing missing values for a feature is to compute descriptive statistics such as mean, median, or mode (for categorical features) and use that value to replace NAs (missing values).
- Although computationally efficient, this approach does not consider any other attributes for a given observation when imputing.
- For instance, a female patient that is 63 inches tall may have her weight imputed as 182lbs since that is the average weight across all observations which contains 65% males that average a height of 70 inches.

# Imputation: Estimated Statistic

Observation	Sex	Weight
Obs. 1	Male	199
Obs. 2	Female	175
Obs. 3	Male	205
Obs. 4	Male	188
Obs. 5	Female	164
Obs. 6	Male	192
Obs. 7	Female	NA
Obs. 8	Female	158
Obs. 9	Male	195
Obs. 10	Male	203
Obs. 11	Female	179


$$\frac{199+175+205+188+164+192+158+195+203+179}{10} =$$

 **185.8**

# Imputation: Estimated Statistic

Observation	Sex	Weight
Obs. 1	Male	199
Obs. 2	Female	175
Obs. 3	Male	205
Obs. 4	Male	188
Obs. 5	Female	164
Obs. 6	Male	192
Obs. 7	Female	NA
Obs. 8	Female	158
Obs. 9	Male	195
Obs. 10	Male	203
Obs. 11	Female	179

An alternative is to use grouped statistics to capture expected values for observations that fall into similar groups.


$$\frac{175 + 164 + 158 + 179}{4} = 169$$

# Imputation: $K$ -nearest Neighbor

- **$K$ -nearest neighbor (KNN)** imputes values by identifying observations with missing values, then identifying other observations that are most similar based on the other available features and using the values from these nearest neighbor observations to impute missing values.
- In KNN imputation, the missing value for a given observation is treated as the targeted response and is predicted based on the average (for quantitative values) or the mode (for qualitative values) of the  $k$  nearest neighbors.

# Measures of Similarity and Dissimilarity

- The **similarity** between two objects is a numerical measure of the degree to which the two objects are **alike**.
- Consequently, similarities are **higher** for pairs of objects that more alike.
- Similarities are usually non-negative and are often between **0** (no similarity) and **1** (complete similarity).

- The **dissimilarity** between two objects is a numerical measure of the degree to which the two objects are **different**.
- Dissimilarities are **lower** for more similar pairs of objects.
- Frequently, the term **distance** is used as synonym for dissimilarity.
- Dissimilarities range from **0 to  $\infty$** .

# Measures of Dissimilarity between Data Objects

- The **Euclidean distance**,  $d$ , between two points (observations),  $\mathbf{x}$  and  $\mathbf{y}$ , in one-, two-, or higher-dimensional space, is given by the following formula:

$$d(\mathbf{x}, \mathbf{y}) = \sqrt{\sum_{k=1}^n (\mathbf{x}_k - \mathbf{y}_k)^2}$$

where  $n$  is the number of dimensions (# of features) and  $x_k$  and  $y_k$  are, respectively, the  $k$ -th attributes (features) of  $\mathbf{x}$  and  $\mathbf{y}$ .



# Measures of Dissimilarity between Data Objects

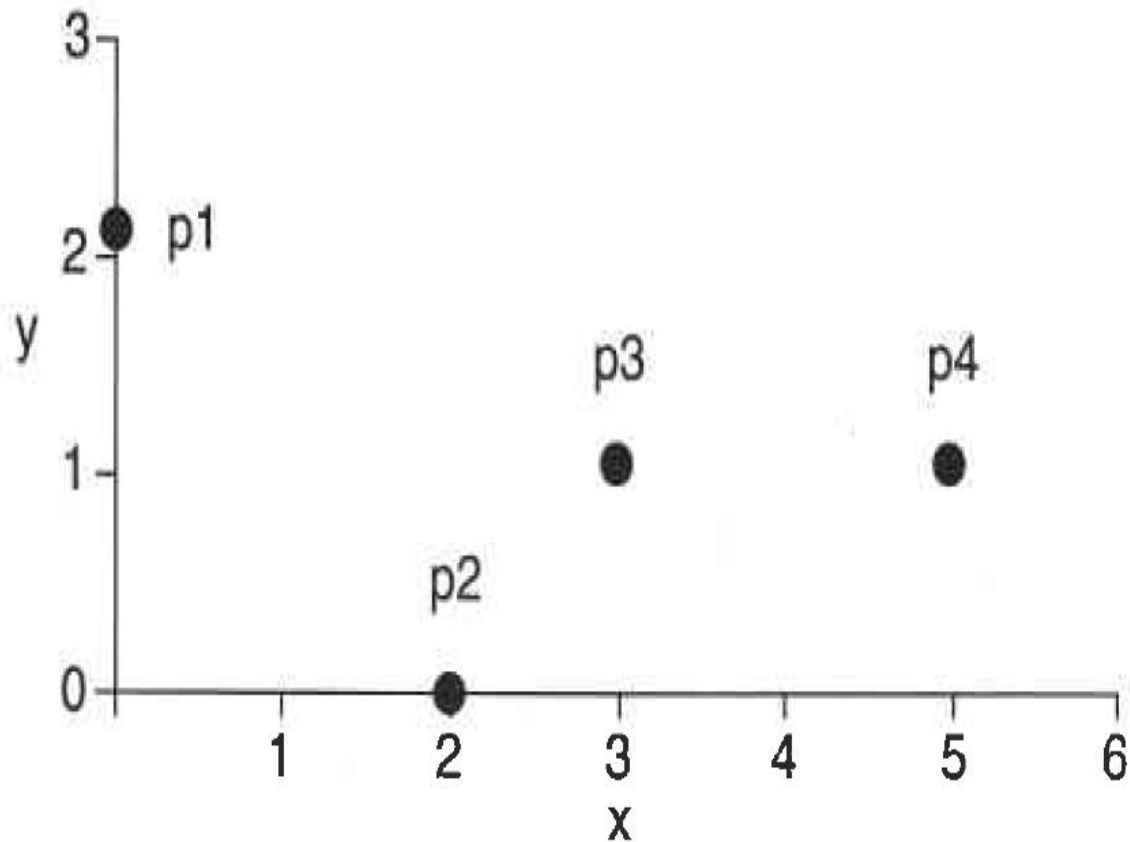
- The Euclidean distance can be generalized by the **Minkowski distance metric** that is given by

$$d(\mathbf{x}, \mathbf{y}) = \left( \sum_{k=1}^n |\mathbf{x}_k - \mathbf{y}_k|^r \right)^{1/r}$$

where  $r$  is a parameter.

- When  $r = 1$ , it's called a **City block (Manhattan) distance**.
- When  $r = 2$ , we get **Euclidean distance**.

# Measures of Dissimilarity between Data Objects: Example



Point	X coordinate	Y coordinate
P1	0	2
P2	2	0
P3	3	1
P4	5	1

# Euclidean Distance: Example

Point	X coordinate	Y coordinate
P1	0	2
P2	2	0
P3	3	1
P4	5	1

Euclidean Distance Matrix				
	P1	P2	P3	P4
P1	0	2.8	3.2	5.1
P2	2.8	0	1.4	3.2
P3	3.2	1.4	0	2.0
P4	5.1	3.2	2.0	0

$$d(P1, P3) = \sqrt{(0 - 3)^2 + (2 - 1)^2} = 3.2$$

# Manhattan Distance: Example

Point	X coordinate	Y coordinate
P1	0	2
P2	2	0
P3	3	1
P4	5	1

Manhattan Distance Matrix				
	P1	P2	P3	P4
P1	0	4	4	6
P2	4	0	2	4
P3	4	2	0	2
P4	6	4	2	0

$$d(P1, P3) = |0 - 3| + |2 - 1| = 4$$

# Similarity Measures for Binary Data

- Similarity measures between two objects that contain only binary features (yes/no or 1/0) are called similarity coefficients, and typically have values between **0** and **1**.
- A value of 1 indicates that the two objects are completely similar, while a value of 0 indicates that the objects are not similar at all.

# Similarity Measures for Binary Data

Let  $x$  and  $y$  be two objects that consist of  $n$  binary features. The comparison of two such objects, that is, two binary vectors, leads to the following quantities (frequencies):

- $f_{00}$  is the number of features where  $x = 0$  and  $y = 0$
- $f_{01}$  is the number of features where  $x = 0$  and  $y = 1$
- $f_{10}$  is the number of features where  $x = 1$  and  $y = 0$
- $f_{11}$  is the number of features where  $x = 1$  and  $y = 1$

# Simple Matching Coefficient (SMC)

- One commonly used similarity coefficient is the Simple Matching Coefficient (SMC), which is defined as

$$SMC = \frac{\text{number of matching feature values}}{\text{number of features}} = \frac{f_{00} + f_{11}}{f_{00} + f_{10} + f_{01} + f_{11}}$$

- SMC counts both presences and absences equally.

# Jaccard Coefficient

- The **Jaccard coefficient**, which is often symbolized by  $J$ , is given by

$$J = \frac{\text{number of matching presences}}{\text{number of features not involved in 00 matches}} = \frac{f_{11}}{f_{10} + f_{01} + f_{11}}$$

- The Jaccard similarity coefficient is used when the primary goal is to assess the similarity between objects based on 11 matches.



# Jaccard Coefficient and SMC: Example

- To illustrate the difference between these two similarity measures, we calculate  $SMC$  and  $J$  for the following two binary vectors:

$$\begin{aligned}x &= (1, 0, 0, 0, 0, 0, 0, 0, 0, 0) \\y &= (0, 0, 0, 0, 0, 0, 1, 0, 0, 1)\end{aligned}$$

- $f_{00} = 7$ , is the number of features where  $x = 0$  and  $y = 0$
- $f_{01} = 2$ , is the number of features where  $x = 0$  and  $y = 1$
- $f_{10} = 1$ , is the number of features where  $x = 1$  and  $y = 0$
- $f_{11} = 0$ , is the number of features where  $x = 1$  and  $y = 1$

# Jaccard Coefficient and SMC: Example

$$\begin{aligned}x &= (1, 0, 0, 0, 0, 0, 0, 0, 0, 0) \\y &= (0, 0, 0, 0, 0, 0, 1, 0, 0, 1)\end{aligned}$$

$$\bullet \text{ } SMC = \frac{f_{00} + f_{11}}{f_{00} + f_{10} + f_{01} + f_{11}} = \frac{7 + 0}{7 + 1 + 2 + 0} = 0.7$$

$$\bullet \text{ } J = \frac{f_{11}}{f_{10} + f_{01} + f_{11}} = \frac{0}{1 + 2 + 0} = 0$$

# Cosine Similarity

- Documents are often represented as vectors, where each feature represents the frequency with which a particular term (word) occurs in the document.
- Even though documents have thousands or tens of thousands of features (terms), each document is sparse since it has relatively few non-zero features.
- Therefore, a similarity measure for documents need to ignore 00 matches like the Jaccard measure, but also be able to handle non-binary vectors.

# Cosine Similarity

- The **cosine similarity** is one of the most common measure of document similarity. If  $\mathbf{x}$  and  $\mathbf{y}$  are two document vectors, then

$$\cos(\mathbf{x}, \mathbf{y}) = \frac{\mathbf{x} \cdot \mathbf{y}}{\|\mathbf{x}\| \|\mathbf{y}\|}$$

where  $\mathbf{x} \cdot \mathbf{y} = \sum_{k=1}^n x_k y_k$ , and  $\|\mathbf{x}\| = \sqrt{\sum_{k=1}^n x_k^2}$

# Cosine Similarity: Example

- This example calculates the cosine similarity for the following two data objects, which might represent document vectors:

$$x = (3, 2, 0, 5, 0, 0, 0, 2, 0, 0)$$

$$y = (1, 0, 0, 0, 0, 0, 0, 1, 0, 2)$$

$$x \cdot y = 3 * 1 + 2 * 0 + 0 * 0 + 5 * 0 + 0 * 0 + 0 * 0 + 0 * 0 + 2 * 1 + 0 * 0 + 0 * 2 = 5$$

$$\|x\| = \sqrt{3 * 3 + 2 * 2 + 0 * 0 + 5 * 5 + 0 * 0 + 0 * 0 + 0 * 0 + 2 * 2 + 0 * 0 + 0 * 0} = 6.48$$

$$\|y\| = \sqrt{1 * 1 + 0 * 0 + 0 * 0 + 0 * 0 + 0 * 0 + 0 * 0 + 0 * 0 + 1 * 1 + 0 * 0 + 2 * 2} = 2.24$$

# Cosine Similarity: Example

$$\cos(x, y) = \frac{x \cdot y}{\|x\| \|y\|} = \frac{5}{6.48 * 2.24} = 0.31$$

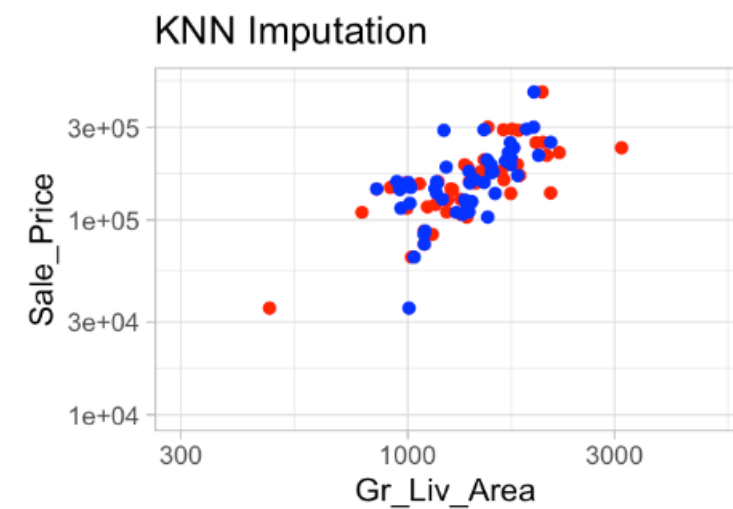
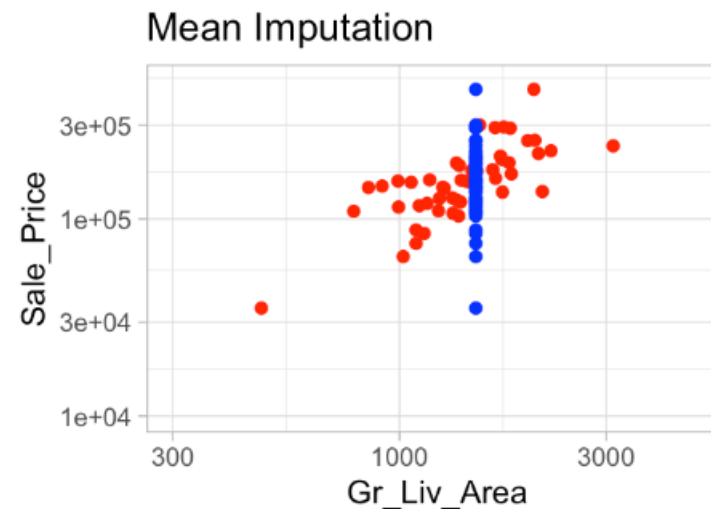
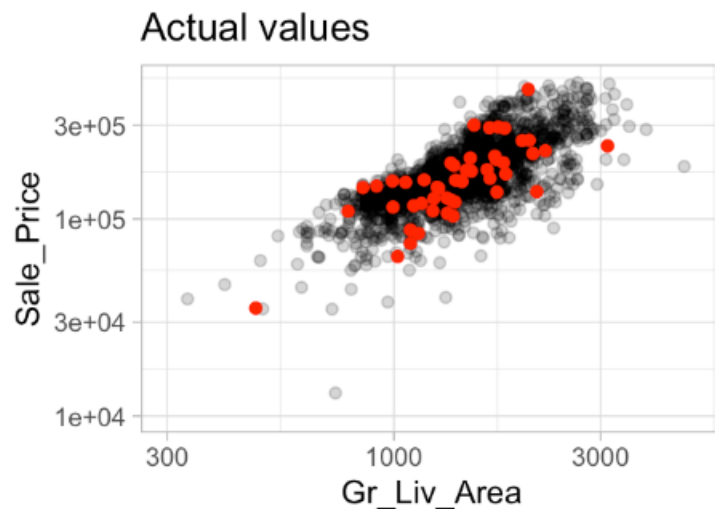
- Cosine similarity really is a measure of the (cosine of the) angle between  $x$  and  $y$ .
- Thus, the cosine similarity is 1, then  $x$  and  $y$  are the same except for magnitude (length).
- And if the cosine similarity is 0, then  $x$  and  $y$  don't share any terms (words).

# Which measure to choose?

- For many types of **dense, continuous data**, metric distance measures such as Euclidean distance are often used.
- Proximity (similarity or dissimilarity) between continuous features is most often expressed in terms of differences, and distance measures provide a well-defined way of combining these differences into an overall proximity measure.
- For **sparse data**, we typically employ similarity measures that ignore 00 matches. Thus, *Jaccard measure* is more appropriate for such data.

# Imputation Example: Ames Data

- Figures below illustrate the difference between **mean** and **KNN imputations**. It is apparent how descriptive statistic method is inferior to the KNN method.





# Numeric Feature Engineering

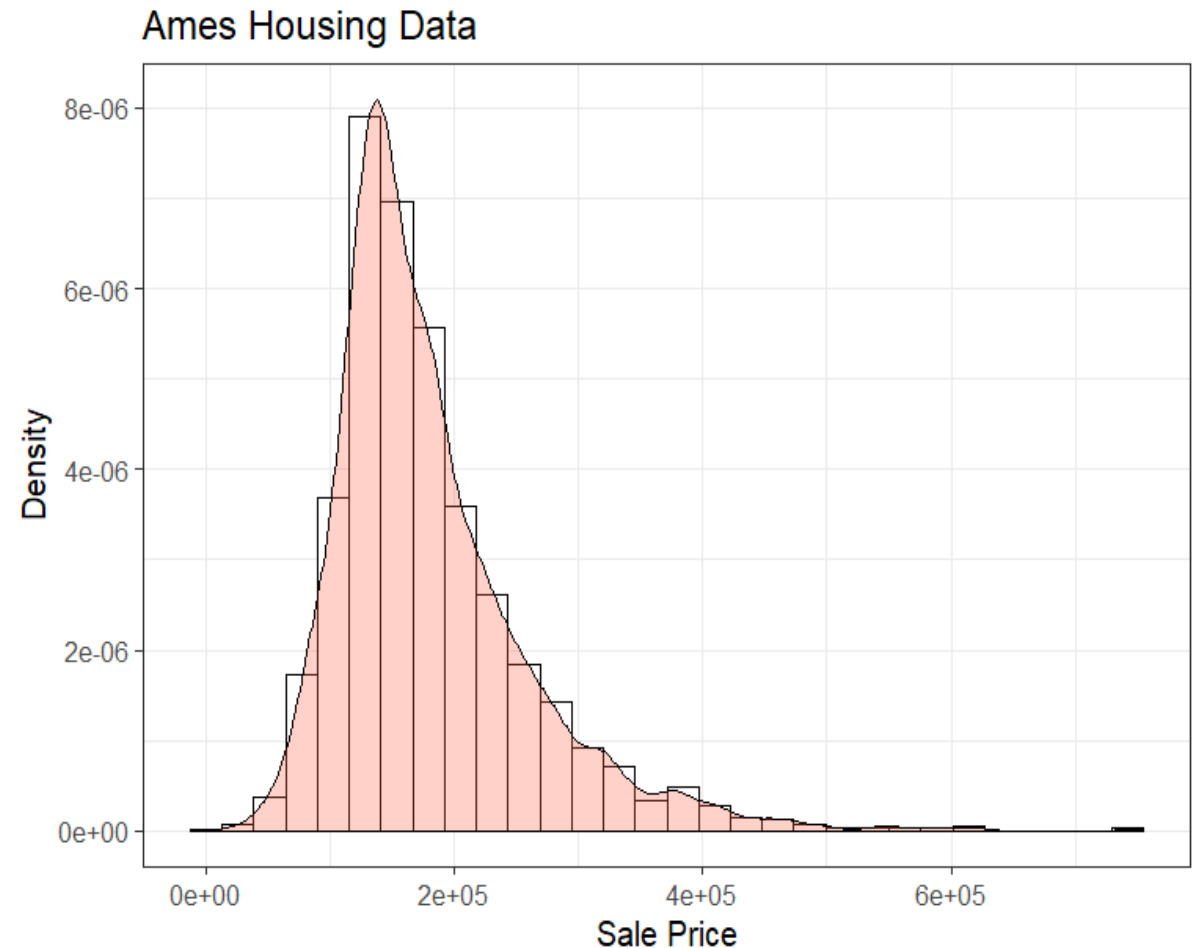
- **Numeric features** can create a host of problems for certain models when their distributions are **skewed**, contain **outliers**, or have a **wide range in magnitudes**.
- For instance, tree-based model are quite immune to these types of problems in the feature space, but many other models (e.g., GLMs, regularized regression, KNN, support vector machine) can be greatly hampered by these issues.
- Some feature engineering techniques that we are going to discuss today can help minimize these concerns.

# Numeric Feature Engineering

- Although not always a requirement, transforming the **response (target) variable** can lead to predictive improvement, especially with parametric models.
- For instance, ordinary linear regression models assume that the error terms (and hence the response) are normally distributed.
- A small violation of this condition is fine, except when the target feature has heavy tails (i.e., **outliers**) or is **skewed** in one direction or the other.

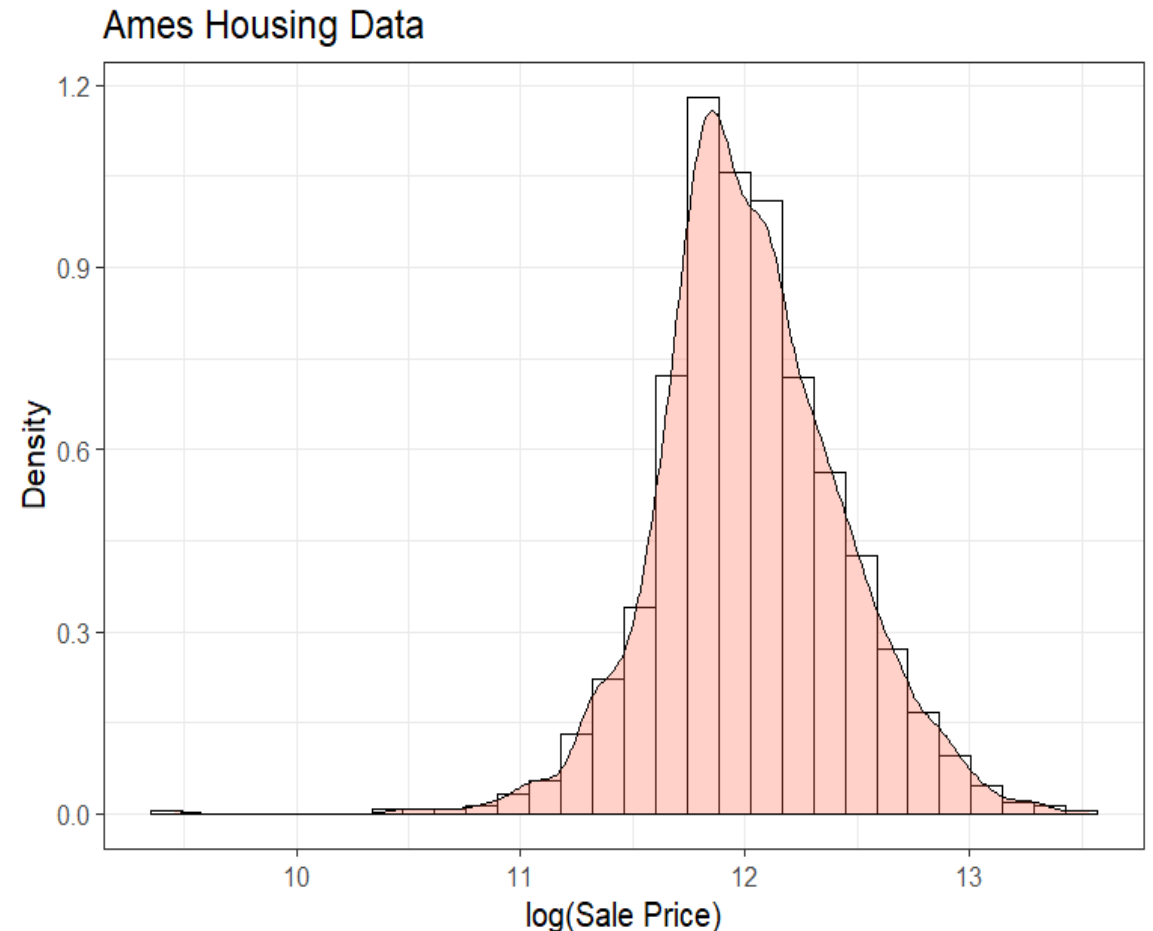
# Numeric Feature Engineering

- For example, the response variable (**Sale Price**) for homes is right (positively) skewed.
- Its values range from \$12,789 to \$755,000.
- Question: what are the approaches to help correct for positively skewed target variables?



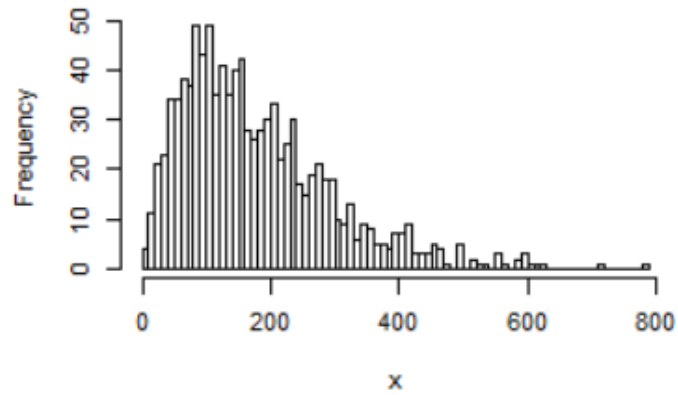
# Target Feature Engineering: Log Transformation

- **Option 1**: Use a **Log transformation**.
- A simple log transformation will help fix the issue with a right-skewed target feature and make its distribution look more “**normal**”.
- The figure to the right illustrates results of log transformation applied to the **Sale Price** variable.

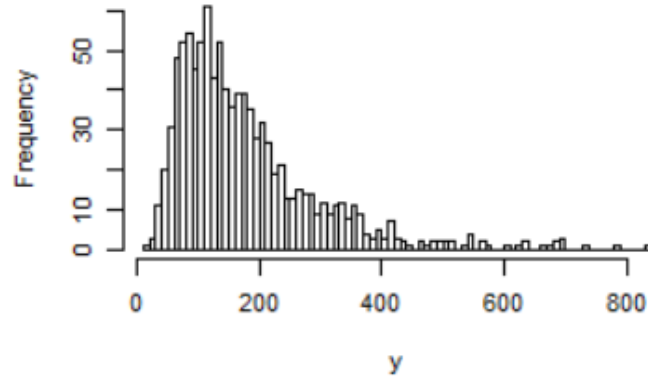


# Target Feature Engineering: Log Transformation

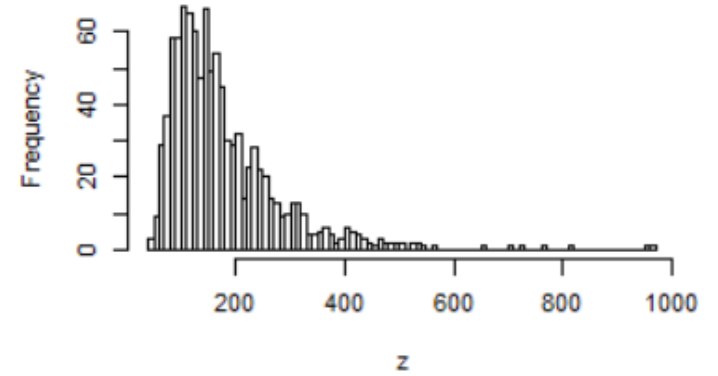
Histogram of x



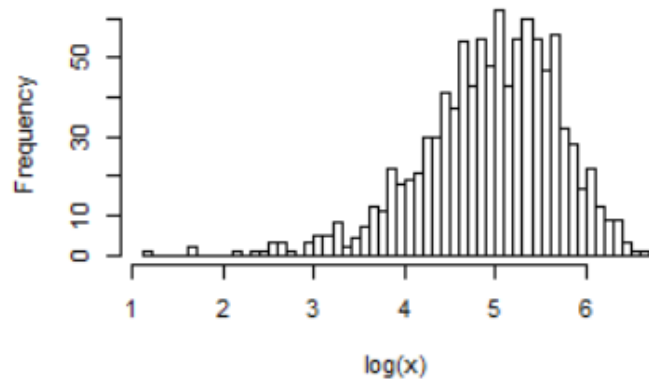
Histogram of y



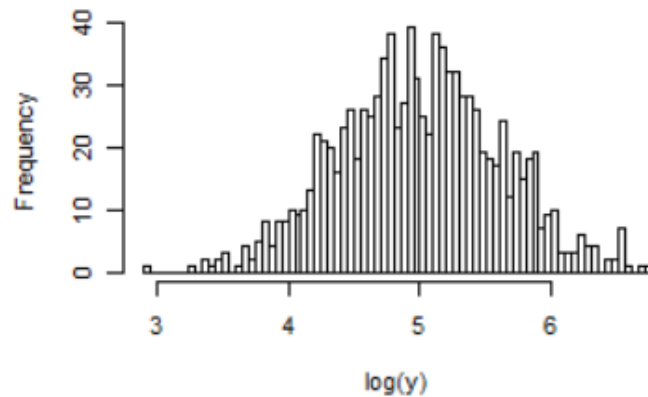
Histogram of z



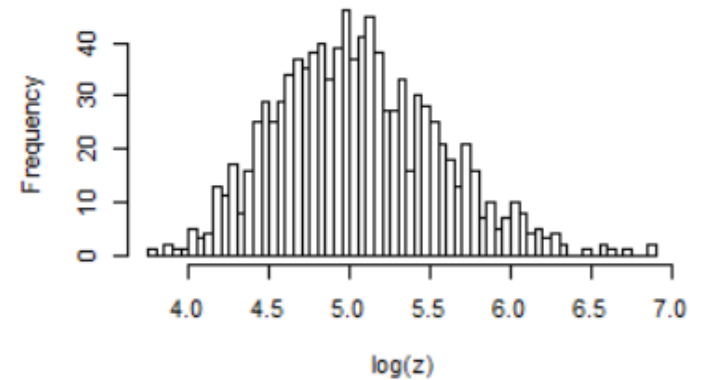
Histogram of log(x)



Histogram of log(y)



Histogram of log(z)



# Target Feature Engineering: Log Transformation

- If you apply the log transformation to a **symmetric** distribution, it will tend to make it **left-skewed** for the same reason it often makes a **right-skewed** distribution more **symmetric** and **normal**.
- If your response has negative values or zeros, then a log transformation will produce  $Na$ 's and  $-INF$ 's, respectively (you cannot take the logarithm of a negative number).
- If the nonpositive values are small (say between -0.99 and 0), then you can apply a **small offset**, which adds 1 to the value prior to applying a log transformation.

# Target Feature Engineering: Log Transformation

- Finally, if you apply the log transformation to something that is already left-skewed, it will tend to make it even more left-skewed. Thus, it wouldn't be helpful.
- In such cases power transformation might be more suitable and useful (discussed next).

# Target Feature Engineering: Box-Cox Transformation

- Option 2: Use a Box-Cox transformation.
- A Box-Cox transformation is more flexible than (but also includes as a special case) the log transformation and will find an appropriate transformation from a family of power transforms that will transform the variable as close as possible to a normal distribution.
- At the core of the Box-Cox transformation is an exponent, lambda ( $\lambda$ ). All values of  $\lambda$  are considered and the optimal value for the given data is being estimated.



# Target Feature Engineering: Box-Cox Transformation

- The transformation of the response  $Y$  has the following form:

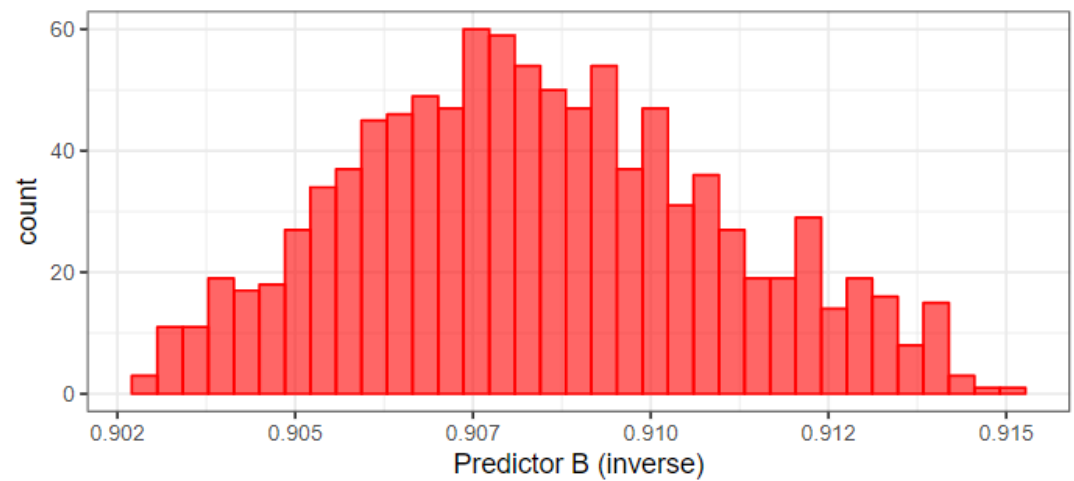
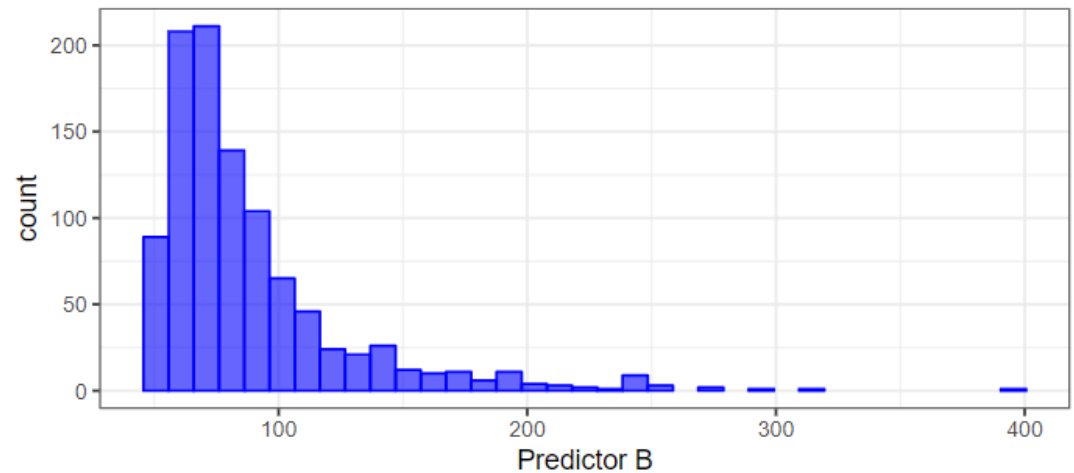
$$Y(\lambda) = \begin{cases} \frac{Y^\lambda - 1}{\lambda}, & \text{if } \lambda \neq 0 \\ \log(Y), & \text{if } \lambda = 0 \end{cases}$$

- The **optimal value** of  $\lambda$  is the one which results in the best transformation to an approximate normal distribution and can be estimate using resampling techniques (for example, cross validation).

# Target Feature Engineering: Box-Cox Transformation

Common Box-Cox Transformations	
Lambda Value ( $\lambda$ )	$\sim$ Transformed data ( $Y(\lambda)$ )
-3	$Y^{-3} = 1/Y^3$
-2	$Y^{-2} = 1/Y^2$
-1	$Y^{-1} = 1/Y$
-0.5	$Y^{-0.5} = 1/\sqrt{Y}$
0	$\text{Log}(Y)$
0.5	$Y^{0.5} = \sqrt{Y}$
1	$Y^1 = Y$
2	$Y^2$
3	$Y^3$

# Target Feature Engineering: Box-Cox Transformation



# Target Feature Engineering: Yeo-Johnson Transformation

- It is important to note that the **Box-Cox transformation** procedure can only be applied to data that is **strictly positive**. To address this problem, **Yeo and Johnson** devised an analogous procedure that can be used on any numeric data:

$$Y(\lambda) = \begin{cases} \frac{((Y + 1)^\lambda - 1)}{\lambda} & \text{if } \lambda \neq 0, Y \geq 0 \\ \ln(Y + 1) & \text{if } \lambda = 0, Y \geq 0 \\ \frac{-((-Y + 1)^{(2-\lambda)} - 1)}{(2 - \lambda)} & \text{if } \lambda \neq 2, Y < 0 \\ -\ln(-Y + 1) & \text{if } \lambda = 2, Y < 0 \end{cases}$$

# Numeric Feature Engineering: Feature Scaling

- **Feature scaling** is a critical step in building accurate and effective ML models. It can help to improve the model performance, reduce the impact of outliers, and ensure that the data is on the same scale.
- Feature scaling transforms the values of features (predictors) in a dataset to a similar scale. It ensures that all features contribute equally to the model and avoids the domination of features with larger values.
- Two commonly feature scaling techniques are **normalization** and **standardization**.

# Normalization

- **Normalization** is a data pre-processing technique employed to bring the values of features in a dataset to a **common scale**.
- It is a scaling technique in which values are **shifted** and **rescaled** so that they end up ranging between **0** and **1**, **[0,1]**. It is also known as **Min-Max Scaling**:

$$X' = \frac{X - X_{min}}{X_{max} - X_{min}}$$

# Standardization

- **Standardization** is another feature scaling method where the values are centered around the mean with a unit standard deviation.
- This means that the **mean** of the feature becomes **zero**, and the resultant distribution has a **unit standard deviation**:

$$X' = \frac{X - \mu}{\sigma}$$

# Categorical Feature Engineering

- **Categorical** (**Nominal**) predictors are those that contain **qualitative data**, that is, data that has no numeric scale. Categorical features can take a variety of forms in the data.
- With the exception of tree-based models, categorical predictors must be first converted into numeric representations so that ML models can compute.
- One of the most common ways of converting categorical features into numeric ones is called **one-hot encoding**. It transposes our categorical features so that each level of the feature is represented as a Boolean value.



# Categorical Feature Engineering: One-Hot Encoding

ID	X
1	A
2	C
3	A
4	B
5	A
6	C
7	C
8	B

One-Hot Encoding

ID	X = a	X = b	X = c
1	1	0	0
2	0	0	1
3	1	0	0
4	0	1	0
5	1	0	0
6	0	0	1
7	0	0	1
8	0	1	0

# Categorical Feature Engineering: Dummy Encoding

- One-hot encoding is called **less than full rank** encoding.
- Although one-hot encoding method is easy to implement and is straightforward to interpret, it creates perfect **collinearity** which causes problems with some predictive modeling algorithms (e.g., ordinary linear regression).
- An alternative approach is to create a **full-rank** encoding by **dropping one of the levels**. This is referred to as **dummy (indicator)** encoding.

# Categorical Feature Engineering: Dummy Encoding

ID	X
1	A
2	C
3	A
4	B
5	A
6	C
7	C
8	B

One-Hot Encoding

ID	X = a	X = b
1	1	0
2	0	0
3	1	0
4	0	1
5	1	0
6	0	0
7	0	0
8	0	1

# Categorical Feature Engineering: Rare Categories

- Sometimes features will contain levels that have very few observations.

Neighborhood	Frequency
Landmark	1
Green Hills	2
Greens	7
Blueste	9
Northpark Villa	17
Briardale	18
.....	.....

# Categorical Feature Engineering: Zero-Variance Predictors

- One potential issue is that resampling might exclude some of the rarer categories (levels) from the training set.
- This would lead to dummy variable columns that contain all zeros, and, for many models, this would become a numerical issue that will cause an error.
- When a predictor contains a single value, we call this a **zero-variance predictor** because there truly is no variation displayed by the predictor.

# Categorical Feature Engineering: Zero-Variance Predictors

- One way of handling this issue is to create the full set of dummy variables and simply remove the zero-variance or near-zero-variance predictors.
- Another solution to this problem is collapsing, or “lumping” rare categories into a lesser number of categories. In general, we may want to collapse all levels that are observed in less than 10% of the dataset into an “Other” category.

# Multicollinearity

- In data analysis, the nature and significance of the relations between predictor variables and the response variable are often of particular interest.
- When predictor variables are highly correlated among themselves, **multicollinearity** is said to exist.
- A multicollinearity among predictor variables can create a variety of interrelated problems.

# Multicollinearity

- The fact that some or all predictor variables are correlated among themselves does not, in general, inhibit our ability to obtain a good fit nor does it tend to affect inferences about mean response or predictions of new observations.
- The main issue with multicollinearity is that the estimated model coefficients tend to have **large sampling variability** (**inflated variances**).
- Thus, the **estimated coefficients** tend to **vary widely** from one sample to another. As a result, inference made about the model coefficients will be **imprecise** and **biased**.



# Multicollinearity

- When multicollinearity exists, the common interpretation or model coefficients, measuring the change in the expected value of the response variable when a given predictor variable is increased by one unit while other predictor variables are held constant, is not fully applicable.
- Additionally, interpretation of the model coefficients will not make sense anymore, because there is an infinite number of estimates available, which are equally good fit to the data.
- Thus, the effect size of predictors will widely depend on the sample data and will vary from one sample to another.

# Multicollinearity: Remedial Measures

- **Solution 1**: Delete some of the highly correlated predictors or combine them.
- **Solution 2**: Standardize predictors or apply other pre-processing procedures (for instance, **Principal Component Analysis**).
- **Solution 3**: Ridge Regression.