

Unit 3

Data cleaning, also known as data cleansing or data scrubbing, is the process of identifying and correcting (or removing) errors, inconsistencies, and inaccuracies within a dataset. This crucial step in the data management and data science pipeline ensures that the data is accurate, consistent, and reliable, which is essential for effective analysis and decision-making.

1. Data Cleaning Terms

Loading the Data

- **Definition:** The process of importing a dataset into the R environment for analysis.
- **Example:** `read.csv()` loads a CSV file into a dataframe object in R.

Inspecting the Data

- **Definition:** Reviewing the structure, summary, and sample records of the data to understand its attributes.
 - `str(data)`: Shows the structure of the dataset (columns, types, etc.).
 - `summary(data)`: Provides basic descriptive statistics for numeric columns and counts for factors.
 - `head(data)`: Displays the first few rows of the dataset.

Handling Missing Data

- **Definition:** The process of identifying and resolving missing values (e.g., NA) in the dataset to avoid errors or biases in analysis.
 - **Removing Missing Data:** Use `na.omit()` to delete rows with missing values.
 - **Imputation:** Replace missing values with statistical measures like mean, median, or custom values.

Fixing Data Types

- **Definition:** Ensuring that variables have the correct type (e.g., numeric, character, factor) for analysis.

- Example: Converting character to factor when a column represents categories.

Removing Duplicates

- **Definition:** Eliminating repeated rows in the dataset to ensure the uniqueness of records.
 - Example: `distinct()` removes duplicate rows.

Renaming Columns

- **Definition:** Changing column names to improve clarity and consistency.
 - Example: Rename `old_name` to `new_name` using `rename()`.

Filtering Outliers

- **Definition:** Detecting and removing values that are unusually high or low compared to the rest of the data.
 -

2. Data Analysis Terms

Descriptive Statistics

- **Definition:** Basic summaries of data to describe its main features.
 - **Mean:** Average value.
 - **Standard Deviation (SD):** Measure of the data's spread around the mean.

Exploratory Data Analysis (EDA)

- **Definition:** An initial investigation of data to uncover patterns, spot anomalies, and form hypotheses.
 - **Correlation:** Measures the strength of the relationship between two numeric variables (values range from -1 to 1).
 - **Frequency Table:** Displays the counts of unique values in a column.

Visualization

- **Definition:** Using graphical representations to understand data distribution and relationships visually.
 - **Histogram:** A bar chart representing the frequency of numeric data values.
 - **Scatter Plot:** A plot showing the relationship between two numeric variables.
 - **Boxplot:** A graphical summary of a numeric variable's distribution and outliers.

Group-wise Summaries

- **Definition:** Aggregating data based on groups to compute metrics like mean, sum, or count.
 - Example: Finding the average sales per region using `group_by()` and `summarise()`.

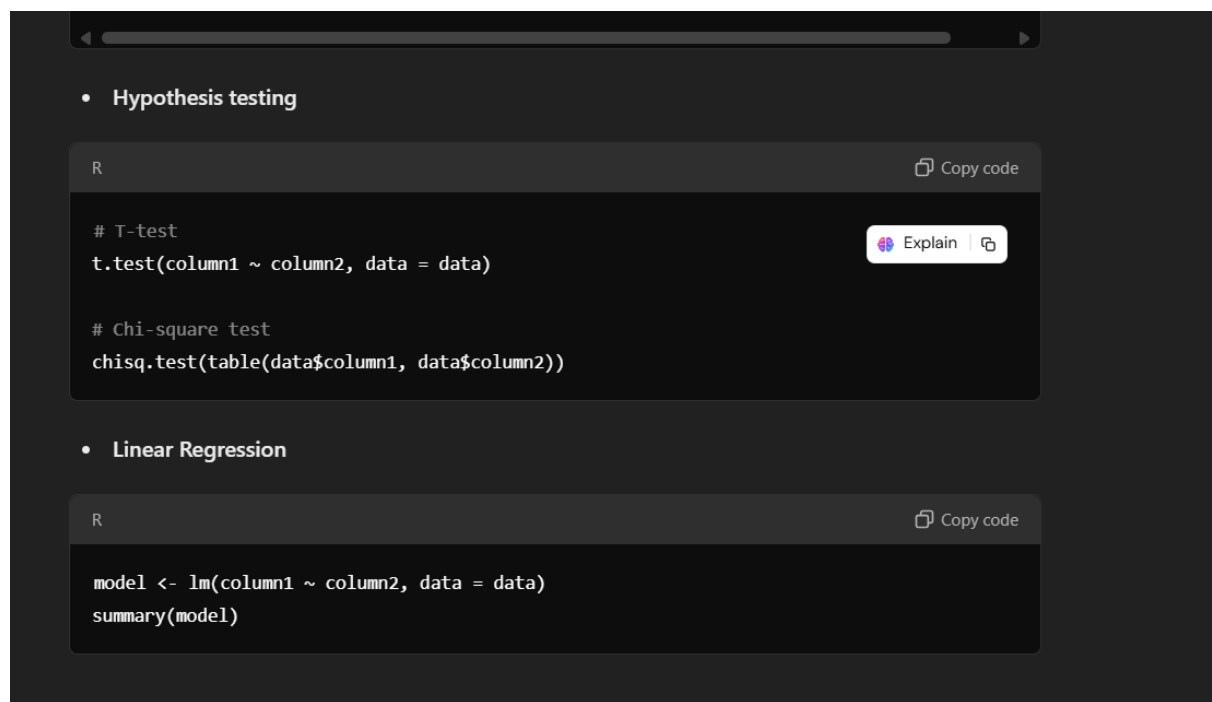
Hypothesis Testing

- **Definition:** Statistical tests to evaluate assumptions or claims about the data.
 - **T-test:** Compares the means of two groups to determine if they are significantly different.
 - **Chi-square Test:** Tests the association between two categorical variables.

Linear Regression

- **Definition:** A statistical method to model the relationship between a dependent variable and one or more independent variables.
 - **dplyr:** For data manipulation (filtering, summarising, grouping, etc.).
 - **tidyr:** For data tidying (reshaping, filling missing values, etc.).
 - **ggplot2:** For creating elegant and versatile data visualizations.

Refer to code from rhistory word docx



A probability distribution describes how the values of a random variable are distributed. It gives the probability of occurrence of each possible value or range of values for a random variable. There are two main types of distributions:

1. **Discrete Probability Distributions:** For discrete random variables, which take on countable values (e.g., 0, 1, 2), such as the outcomes of a die roll.

Binomial Distribution: Used for binary outcomes (e.g., success/failure).

Poisson Distribution: Used for the count of events occurring in a fixed interval.

2. Continuous Probability Distributions: For continuous random variables, which can take on an infinite number of values within a range (e.g., weight, height, temperature).

Normal Distribution: A bell-shaped curve for many natural phenomena

Uniform Distribution: All values in a range are equally likely.

Exponential Distribution: Models time between events in a Poisson process

Summary Table

Type	Example	R Function
Discrete	Binomial, Poisson	dbinom, dpois, rbinom
Continuous	Normal, Uniform, Exponential	dnorm, runif, dexp, rnorm

When it comes to achieving the mean of two or more population groups, ANOVA (Analysis of variance) and t-test are the two best practices preferred. Although there is a thin line of difference between both of them.

The t-test is conducted when you have to find the population means between two groups. But when there are three or more groups you go for the ANOVA test.

Both t-test and ANOVA are the statistical methods of testing a hypothesis. And they both share the assumptions:

Sample drawn from the population is normally distributed

Homogeneous variance

Random data sampling

Observations are independent

Dependent variable is measured in ratio or interval levels

ANOVA VS t-test: Definition

The definition is the best way to understand how the two differ, so let's start with that.

What is a t-test?

This method of data analysis examines how greatly the population means of two samples differ from each other.

The best use of the t-test is to test a hypothesis. The data analysis method helps determine if a process has any effect on the target population. The method should be used when you want to compare the means of two groups.

: What is ANOVA?

Analysis of Variance, developed by Ronald Fisher, helps find the statistical difference between the means of two or more groups.

As a marketer, it is important to learn when to use ANOVA tests. You can use this data analysis to identify how different groups of customers or populations of interest respond. The one-way Analysis of Variance can show you whether the independent variables have any significant difference.

Tests:

The comparison chart for ANOVA vs t-test only gives you an overview of the differences between the two data analysis methods. In this section, we are diving deeper into the differences by comparing the definition and working of the two analysis methods.

Comparison variable	T-TEST	ANOVA
Definition	t-test is statistical hypothesis test used to compare the means of two population groups.	ANOVA is an observable technique used to compare the means of more than two population groups.
Feature	t-test compares two sample sizes (n) both below 30.	ANOVA equates three or more such groups.
Error	t-test is less likely to commit an error.	ANOVA has more error risks.
Example	Sample from class A and B students have given a mathematics course may have different mean and standard deviation.	When one crop is being cultivated from various seed varieties.
Test	t-test can be performed in a double-sided or single-sided test.	ANOVA is one-sided test due to no negative variance.
Population	t-test is used when the population is less than 30.	ANOVA is used for huge population counts.



The following diagram will give you a better understanding of when to use t-test and ANOVA.

T-Test:


- Used for comparing means of two groups.
- Assumptions:
 - Normality of data
 - Equal variances
 - Independence of observations

R Functions:

R Functions for T-Tests

1. Independent Samples T-Test: Compare means of two groups.


R

 Copy code

```
t.test(score ~ group, data = df)
```

2. Paired Samples T-Test: Compare means of two related groups (e.g., before and after).


R

 Copy code

```
t.test(before, after, paired = TRUE)
```

3. One-Sample T-Test: Compare a sample mean to a specific value.


R

 Copy code

```
t.test(df$score, mu = 0)
```

4. Pairwise Comparisons: Compare all pairs of groups.

R

 Copy code


```
pairwise.t.test(x, g, p.adjust.method = "method")
```


Example: Independent Samples T-Test

Data

We have test scores from two groups: A and B.

```
R
# Sample Data
df <- data.frame(
  group = c("A", "A", "A", "B", "B", "B"),
  score = c(85, 90, 88, 78, 82, 80)
)
```


 Copy code

 Explain 

Perform T-Test

```
R
# Perform T-Test
t_result <- t.test(score ~ group, data = df)

# View T-Test result
print(t_result)
```

 Copy code

 Explain 



- `t.test()`: Performs an independent samples t-test or paired samples t-test.


`t.test(score ~ group, data = df)`

- `t.test()`: Performs a one-sample t-test.

Compare the mean score of one group to a value (e.g., 80).

```
R
# One-Sample T-Test
t_result_one_sample <- t.test(df$score, mu = 80)

# View result
print(t_result_one_sample)
```

 Copy code

 Explain 

`t.test(df$score, mu = 0)`

- `pairwise.t.test()`: Performs pairwise comparisons.

ANOVA (Analysis of Variance):

- Used for comparing means of three or more groups.

- Assumptions:

- Normality of data
- Equal variances
- Independence of observations: Observations must be independent of each other.

R Functions:

R Functions

1. Performing ANOVA:

- Use the `aov()` function.

```
R
aov(response ~ predictor, data = dataset)
```

Copy code

- Example:

```
R
aov(score ~ group, data = df)
```

Copy code

2. Summarizing Results:

- Use `summary()` to display ANOVA results.

```
R
summary(aov_model)
```

Copy code

- ``aov()``: Performs an analysis of variance.

```
aov(score ~ group, data = df)
```

* `summary()`: Summarizes the ANOVA results.

Post-Hoc Tests:

* Used after significant ANOVA results to compare specific groups.

* R Functions:

* `TukeyHSD()`: Performs Tukey's Honest Significant Difference test.

* `pairwise.t.test()`: Performs pairwise comparisons.

1. Tukey's Honest Significant Difference Test:

- Use `TukeyHSD()` for multiple comparisons.

```
R
TukeyHSD(aov_model)
```

Copy code

2. Pairwise Comparisons:

- Use `pairwise.t.test()` for pairwise t-tests with adjusted p-values.

```
R Copy code

# Sample Data
df <- data.frame(
  group = c("A", "A", "A", "B", "B", "B", "C", "C", "C"),
  score = c(85, 90, 88, 78, 82, 80, 92, 95, 94)
)

# Perform ANOVA
anova_result <- aov(score ~ group, data = df)
summary(anova_result)

# Post-hoc test: Tukey's HSD
TukeyHSD(anova_result)
```

Example Code:

r

Load data

data(mtcars)

T-Test

t.test(mpg ~ cyl, data = mtcars)

ANOVA

aov_model <- aov(mpg ~ cyl, data = mtcars)

summary(aov_model)

Post-Hoc Test

TukeyHSD(aov_model)

Assumptions Checking:

- Normality: shapiro.test() or qqnorm()

- Equal Variances: `var.test()` or `leveneTest()`

Non-Parametric Alternatives:

- Wilcoxon Rank-Sum Test (Mann-Whitney U Test): `wilcox.test()`

- Kruskal-Wallis Test: `kruskal.test()`

Here's a detailed explanation of the assumptions, their tests, and non-parametric alternatives with definitions:

Assumptions Checking

1. Normality

Definition: Normality refers to the assumption that the data follows a normal distribution. Many statistical tests (e.g., t-tests, ANOVA) assume normality to ensure valid inferences.

How to check:

- **Shapiro-Wilk Test (`shapiro.test()`):** This tests the null hypothesis that the data is normally distributed. If the p-value is less than the significance level (e.g., 0.05), the null hypothesis is rejected, indicating the data is not normal.
- **Q-Q Plot (`qqnorm()`):** This is a graphical method to check normality. If the points lie approximately on the straight line in a Q-Q plot, the data is likely normally distributed.

2. Equal Variances (Homogeneity of Variance)

Definition: This assumption implies that the variance within each group being compared is approximately the same. It is crucial for tests like ANOVA and t-tests when comparing multiple groups.

How to check:

- **F-Test (`var.test()`):** This is used to compare variances between two groups. It tests the null hypothesis that the variances of the two groups are equal.
- **Levene's Test (`leveneTest()` from the `car` package):** This is a more robust test that checks for equality of variances across multiple groups. It is less sensitive to deviations from normality.

Non-Parametric Alternatives

If the assumptions of normality and/or equal variances are violated, non-parametric tests can be used as they do not rely on these assumptions.

1. Wilcoxon Rank-Sum Test (Mann-Whitney U Test)

Definition: A non-parametric test used to compare the medians of two independent groups. It is an alternative to the independent two-sample t-test when the assumption of normality is violated.

- **Function:** `wilcox.test()`
 - **When to use:** Data is not normal, or the sample size is small.
-

2. Kruskal-Wallis Test

Definition: A non-parametric test used to compare medians across multiple independent groups. It is an alternative to one-way ANOVA when normality or equal variance assumptions are violated.

- **Function:** `kruskal.test()`
 - **When to use:** Data is not normal, or variances are unequal.
-

Summary

1. **Normality:** Test with `shapiro.test()` or visualize with `qqnorm()`.
2. **Equal Variances:** Test with `var.test()` or `leveneTest()`.
3. **Non-Parametric Alternatives:** Use `wilcox.test()` for two groups and `kruskal.test()` for multiple groups if assumptions are violated.

Regression

Regression is a foundational statistical technique for analyzing relationships between variables. It allows us to:

- **Describe relationships:** Understand how a dependent variable changes as one or more independent variables change.
- **Predict outcomes:** Use known values of predictors to estimate unknown outcomes.
- **Infer causal effects:** Test hypotheses about the impact of predictors under certain assumptions.

Types of Regression

1. **Simple Linear Regression:** Models a straight-line relationship between one predictor and the response.

$$Y = \beta_0 + \beta_1 X + \epsilon$$

Example: Predicting a car's mileage (YYY) based on its weight (XXX).

2. **Multiple Linear Regression:** Extends simple regression to multiple predictors.

$$Y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \dots + \beta_n X_n + \epsilon$$

Ex.: Predicting a car's mileage (YYY) based on weight (X1X_1X1) and horsepower (X2X_2X2).

3. **Polynomial Regression:** Models non-linear relationships by adding polynomial terms:

$$Y = \beta_0 + \beta_1 X + \beta_2 X^2 + \dots + \epsilon$$

Example: Modeling a curvilinear trend in sales over time.

4. **Logistic Regression:** Used when the response variable is binary (e.g., yes/no). It models the probability of the event as a logistic function:

$$\text{logit}(p) = \ln \left(\frac{p}{1-p} \right) = \beta_0 + \beta_1 X$$

Example: Predicting whether a customer will buy a product (yes/no) based on income (XXX).

5. **Regularized Regression (Ridge, Lasso, Elastic Net):** These are extensions of linear regression that include penalties to reduce overfitting, especially in high-dimensional data.
6. **Non-parametric Regression:** Methods like splines or Generalized Additive Models (GAMs) that make minimal assumptions about the relationship between variables.

Linear Models in Detail

1. Understanding Linear Regression

Linear regression assumes that the relationship between the predictor variables (X) and the outcome variable (Y) can be expressed as a linear equation:

$$Y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \dots + \beta_p X_p + \epsilon$$

- Y : Outcome variable (dependent variable).
- X_1, X_2, \dots, X_p : Predictor variables (independent variables).
- β_0 : Intercept (baseline value of Y when all predictors are 0).
- β_i : Coefficients or slopes, representing the change in Y for a one-unit increase in X_i , keeping other predictors constant.
- ϵ : Error term, capturing variability in Y not explained by the predictors.

2. Key Components

Intercept (β_0)

- Represents the predicted value of Y when all predictors (X_1, X_2, \dots) are 0.
- Example: In a house price model, β_0 could represent the base price of a house with no additional features.

Slope (β_i)

- Measures how much Y changes for a one-unit increase in X_i , holding other predictors constant.
- Example: If $\beta_1 = 500$, it means that for every additional square foot, the house price increases by \$500.

Error Term (ϵ)

- Accounts for random noise or variability in Y not explained by the model.
- Example: Variations in house prices due to factors like market conditions or personal preferences.

3. Key Outputs from a Linear Model

When you fit a linear model, R provides the following critical outputs:

(a) Coefficients

- Represents the effect size of predictors on the outcome variable.
- Example: A coefficient of 3.5 for `hours` means that for every additional hour of study, the test score increases by 3.5 points.

(b) R^2 : Proportion of Variance Explained

- Indicates how well the predictors explain the variability in Y .
- Ranges from 0 to 1:
 - $R^2 = 0$: Predictors explain none of the variance in Y .
 - $R^2 = 1$: Predictors explain all the variance in Y .
- Higher R^2 is generally better, but overly high values might indicate overfitting.

(c) p-values

- Tests the null hypothesis that the coefficient is zero (no effect).
- A small p-value (< 0.05) suggests the predictor significantly affects Y .

Dataset: Test Scores vs Study Hours

Hours Studied (X)	Test Score (Y)
1	50
2	55
3	65
4	70
5	80

R Code: Simple Linear Regression

R

Copy code

```
# Data
data <- data.frame(
  hours = c(1, 2, 3, 4, 5),
  score = c(50, 55, 65, 70, 80)
)

# Fit the linear model
model <- lm(score ~ hours, data = data)

# View model summary
summary(model)
```

Explain



Linear regression is a special case of regression that assumes a linear relationship between predictors and the outcome variable. Its simplicity and interpretability make it one of the most widely used methods.

- **Intercept (β_0 or β_0):** Baseline value of YYY when all predictors are zero.
- **Slope (β_1 or β_1):** Change in YYY for a one-unit increase in X_i , holding other predictors constant.
- **Error term (ϵ or ϵ):** Accounts for the variability in YYY not explained by predictors.

According to code Key Output:

- Coefficients: Interpret the effect size of predictors.
- R^2 : Proportion of variance explained by the model (higher is better).
- ppp-values: Statistical significance of each predictor.

Smoothing Techniques

Smoothing is a flexible approach for understanding non-linear relationships or reducing noise in data. It is particularly useful when the relationship between variables is not well-captured by straight lines.

Types of Smoothing Techniques

Lowess/Loess (Locally Weighted Scatterplot Smoothing):

- Fits smooth curves by giving more weight to nearby points.
- Useful for exploratory data analysis.

Splines:

- Splits data into intervals and fits polynomials to each interval, ensuring smooth transitions.
- Useful for flexible modeling of non-linear trends.

Generalized Additive Models (GAMs):

- Extends linear models by using smooth functions for predictors.

Kernel Smoothing:

- Uses a kernel function to assign weights based on proximity to produce smooth estimates.

Smoothing with ggplot2: code given below

Smoothing is a data analysis technique used to highlight patterns and trends in data by reducing noise. It is especially useful for exploring relationships between variables when they exhibit non-linear patterns. Here's a detailed explanation of the common smoothing techniques:

Smoothing Techniques

1. Lowess/Loess (Locally Weighted Scatterplot Smoothing)

Definition: Lowess (for one dimension) and Loess (for two or more dimensions) are non-parametric methods that fit a smooth curve to data points by giving more weight to points closer to each observation (local weighting).

How it works:

- A regression is performed on subsets of the data within a "window" (localized region).
- The weight assigned to points decreases as their distance from the target point increases, controlled by a kernel function.

Use Case:

- Often used in **exploratory data analysis** to visualize trends without assuming a specific parametric model.
-

2. Splines

Definition: Splines are a mathematical tool that divides data into intervals and fits a polynomial function to each interval while ensuring smoothness at the boundaries (called knots).

Types:

- **Cubic Splines:** Fits cubic polynomials to each interval.
- **Natural Splines:** Constrain the spline to be linear beyond the boundary knots, reducing overfitting.
- **B-Splines:** Basis splines that are computationally efficient.

How it works:

- Knots are selected within the data range.
- Polynomials are fitted to each segment, ensuring smooth transitions at the knots.

Use Case:

- Ideal for modeling **non-linear relationships** in data when flexibility is required.

3. Generalized Additive Models (GAMs)

Definition: GAMs extend linear regression by allowing for smooth, non-linear relationships between the predictors and the response variable. They use smoothing functions (e.g., splines) to model each predictor.

How it works:

- Each predictor is modeled with a smooth function instead of a single coefficient.
- The model takes the form: $y = \beta_0 + f_1(x_1) + f_2(x_2) + \dots + \epsilon$ where f_1, f_2, \dots are smooth functions.

Use Case:

- Useful for complex **non-linear modeling** where predictors may have intricate relationships with the response.

4. Kernel Smoothing

Definition: Kernel smoothing uses a kernel function (e.g., Gaussian, Epanechnikov) to assign weights to data points based on their distance from a target point. The closer a point is, the higher its weight.

How it works:

- A kernel function (e.g., Gaussian) is centered at each point of interest.
- Weights decrease as the distance increases, producing a weighted average for the smoothed value.

Use Case:

- Used for estimating **probability densities** (e.g., kernel density estimation) or **smoothing scatterplots** in one or two dimensions.

Comparison of Techniques

Technique	Key Feature	Best For
Lowess/Loess	Local weighting, no global function	Exploratory trends in small datasets
Splines	Piecewise polynomial fits	Modeling complex non-linear data
GAMs	Combines linearity with smooth functions	Predictive modeling with smooth terms
Kernel Smoothing	Weighted average based on distance	Density estimation and scatterplots

```
# Load the dataset
data <- mtcars

# Linear Regression
linear_model <- lm(mpg ~ wt + hp, data = data)
summary(linear_model)

# Visualizing the regression line
library(ggplot2)
ggplot(data, aes(x = wt, y = mpg)) +
  geom_point() +
  geom_smooth(method = "lm", col = "blue") +
  labs(title = "Linear Regression Example")

# Lowess Smoothing
ggplot(data, aes(x = wt, y = mpg)) +
  geom_point() +
  geom_smooth(method = "loess", col = "green") +
  labs(title = "Loess Smoothing Example")

# Generalized Additive Model (GAM)
library(mgcv)
gam_model <- gam(mpg ~ s(wt) + s(hp), data = data)
summary(gam_model)
plot(gam_model)
```

