Data Analyst Handy Guide

Essential Tools and Concepts for Data Analysis

Statistics

Descriptive Statistics

Summarizes and describes the main features of a dataset

- **Mean:** Average value ($\Sigma x/n$)
- **Median:** Middle value when sorted
- **Mode:** Most frequent value
- **Range:** Max - Min
- **Std Dev:** $\sqrt{(\Sigma(x-\mu)^2/n)}$
- **Variance:** $\sigma^2 = \Sigma(x-\mu)^2/n$
- **Outliers:** Values beyond Q1-1.5×IQR or Q3+1.5×IQR

Use case: Summarizing customer demographics

Inferential Statistics

Makes predictions or inferences about a population based on sample data

- **Hypothesis Testing:** Determine if results are statistically significant
- **P-Values:** Probability of observing results if null hypothesis is true
- **T-Test:** Compare means between two groups
- **ANOVA:** Compare means across multiple groups
- **Confidence Intervals:** Range of values likely to contain population parameter

Use case: Testing if new website design increases conversion

Correlation vs Causation

Understanding relationships between variables

- **Correlation:** Statistical relationship (-1 to +1)
- **Causation:** One variable directly affects another

```
Pearson's r = Σ[(x-μx)(y-μy)] / √[Σ(x-μx)² × Σ(y-μy)²]
```

Use case: Analyzing relationship between ad spend and sales

Remember: Correlation ≠ Causation!

Distribution Types

Patterns of how data is distributed

- **Normal:** Bell-shaped, symmetric (mean=median=mode)
- **Skewed:** Asymmetric distribution
- **Right-skewed:** Tail extends to the right (mean>median)
- **Left-skewed:** Tail extends to the left (mean

Use case: Determining appropriate statistical tests

A/B Testing Basics

Comparing two versions to determine which performs better

- **Null Hypothesis:** No difference between versions
- **Alternative Hypothesis:** Difference exists
- **Significance Level:** Typically α=0.05
- **Sample Size:** Adequate to detect meaningful difference

Use case: Testing which email subject line generates more opens

$$\bar{X} = \frac{\Sigma fi \cdot xi}{\Sigma fi}$$

Excel

Σ Basic Functions

Fundamental operations for data calculations

- **SUM:** Adds values in a range
- **AVERAGE:** Calculates arithmetic mean
- **COUNT:** Counts cells with numbers
- **IF:** Logical test with true/false results
- **AND/OR:** Combines multiple conditions

```
=IF(AND(A1>10, B1<20), "Pass", "Fail")
```

Use case: Calculating quarterly sales totals

Aggregation

Conditional calculations based on criteria

- **SUMIF:** Sum cells meeting specific criteria
- **COUNTIF:** Count cells meeting criteria
- **AVERAGEIF:** Average cells meeting criteria
- **AGGREGATE:** Performs various calculations with option to ignore errors/hidden rows

```
=SUMIF(range, criteria, [sum_range])
```

Use case: Summing sales by region

Lookup Functions

Retrieve data from tables based on lookup values

- **VLOOKUP:** Vertical lookup (searches first column)
- **HLOOKUP:** Horizontal lookup (searches first row)
- **XLOOKUP:** Modern, flexible lookup (Excel 365)
- **INDEX-MATCH:** Powerful combination for flexible lookups

```
=VLOOKUP(lookup_value, table_array, col_index_num, [range_lookup])
```

Use case: Finding product prices by ID

PivotTables & PivotCharts

Summarize and visualize large datasets

- **PivotTable:** Interactive table that summarizes data
- **Fields:** Rows, Columns, Values, Filters
- **Calculated Fields:** Create custom calculations
- **PivotCharts:** Visual representation of PivotTable data

Use case: Analyzing sales by product, region, and time period

Conditional Formatting

Apply formatting based on cell values

- **Highlight Cells:** Color cells based on values
- **Data Bars:** Visual bars representing cell values
- **Color Scales:** Gradient colors based on value
- **Icon Sets:** Display icons based on values

Use case: Visualizing performance metrics with color coding

Data Cleaning Tools

Tools for preparing and cleaning data

- **Text to Columns:** Split text into multiple columns
- **Remove Duplicates:** Eliminate duplicate entries
- **Flash Fill:** Auto-fill patterns based on examples
- **TRIM/UPPER/LOWER:** Text formatting functions

Use case: Standardizing customer address formats

Charts & Visualization

Create visual representations of data

- **Column/Bar:** Compare values across categories
- **Line:** Show trends over time
- **Pie:** Show parts of a whole
- **Scatter:** Show relationships between variables

Use case: Creating monthly sales trend dashboard

Python

Python Basics

Fundamental programming concepts in Python

- **Variables:** Data containers (x = 5)
- **Loops:** for, while iterations
- **Functions:** def my_func():
- **Lists:** Ordered, mutable collection
- **Dictionaries:** Key-value pairs

```
for i in range(5):
  print(i)
```

Use case: Automating repetitive data tasks

NumPy

Fundamental package for scientific computing

- **Arrays:** Multi-dimensional data structures
- **Indexing:** Accessing array elements
- **Math Operations:** Vectorized calculations

```
import numpy as np
arr = np.array([1, 2, 3])
np.mean(arr)
```

Use case: Performing mathematical operations on large datasets

Pandas

Data manipulation and analysis library

- **DataFrames:** 2D labeled data structures
- **Filtering:** Selecting data based on conditions
- **Aggregation:** Grouping and summarizing data
- **Merge:** Combining datasets
- **GroupBy:** Split-apply-combine operations

```
df.groupby('category').sum()
```

Use case: Analyzing sales data by product category

Data Cleaning

Preparing data for analysis

- **Nulls:** Handling missing values
- **Duplicates:** Removing duplicate entries
- **Formatting:** Standardizing data types

```
df.dropna()
df.drop_duplicates()
df['date'] = pd.to_datetime(df['date'])
```

Use case: Preparing customer data for analysis

Visualization

Creating visual representations of data

- **Matplotlib:** Basic plotting library
- **Seaborn:** Statistical data visualization

```
import matplotlib.pyplot as plt
import seaborn as sns
sns.scatterplot(x='age', y='income', data=df)
```

Use case: Creating visual reports for stakeholders

Working with Dates & Strings

Handling temporal and text data

- **Dates:** Parsing, formatting, calculations
- **Strings:** Manipulation, extraction, formatting

```
df['date'].dt.strftime('%Y-%m-%d')
df['text'].str.upper()
```

Use case: Analyzing time series sales data

EDA Process with Python

Exploratory Data Analysis workflow

- **Understand:** Data structure and content
- **Clean:** Handle missing values and outliers
- **Analyze:** Statistical summaries and relationships
- **Visualize:** Create plots to identify patterns

Use case: Initial investigation of a new dataset

R Programming

R Basics

Fundamental data structures in R

- **Vectors:** Ordered collection of elements
- **DataFrames:** 2D tables with columns of different types
- **Lists:** Collections of elements of different types

```
vec <- c(1, 2, 3)
df <- data.frame(x = c(1, 2), y = c("a", "b"))
lst <- list(a = 1, b = "text")
```

Use case: Storing and manipulating datasets

Tidyverse Overview

Collection of R packages for data science

- **Core Packages:** ggplot2, dplyr, tidyr, readr, purrr
- **Design Philosophy:** Tidy data principles
- **Installation:** install.packages("tidyverse")

```
library(tidyverse)
```

Use case: Comprehensive data analysis workflow

Data Manipulation with dplyr

Grammar of data manipulation

- **filter:** Select rows based on conditions
- **select:** Choose columns by name
- **mutate:** Create new variables
- **group_by:** Group data for analysis
- **summarise:** Reduce multiple values to single summary

```
df %>% filter(x > 5) %>% group_by(category) %>%
summarise(mean_y = mean(y))
```

Use case: Calculating average sales by region

Data Cleaning

Preparing data for analysis

- **NA handling:** Removing or imputing missing values
- **Type conversion:** Changing data types
- **Outlier detection:** Identifying extreme values

```
df %>% drop_na()
df %>% mutate(date = as.Date(date_string))
df %>% filter(!is.outlier(value))
```

Use case: Preparing survey data for analysis

Visualization using ggplot2

Grammar of graphics for data visualization

- **Scatter:** geom_point()
- **Bar:** geom_bar() or geom_col()
- **Boxplot:** geom_boxplot()
- **Themes:** Customizing plot appearance

```
ggplot(df, aes(x = category, y = value)) +
  geom_bar(stat = "identity") +
  theme_minimal()
```

Use case: Creating publication-quality visualizations

Statistical Functions in R

Built-in statistical capabilities

- **Descriptive:** mean(), sd(), median(), quantile()
- **Inferential:** t.test(), aov(), cor.test()
- **Distributions:** rnorm(), pnorm(), dnorm()
- **Modeling:** lm(), glm()

```
t.test(group1, group2)
model <- lm(y ~ x1 + x2, data = df)
```

Use case: Conducting hypothesis tests and regression analysis

SQL

Basic Queries

Fundamental SQL commands for data retrieval

- **SELECT:** Retrieves data from database
- **WHERE:** Filters rows based on conditions
- **ORDER BY:** Sorts result set
- **LIMIT:** Restricts number of rows returned

```
SELECT name, age
FROM customers
WHERE age > 25
ORDER BY name ASC
LIMIT 10;
```

Use case: Retrieving customer information

Aggregation

Functions that operate on sets of values

- **COUNT:** Counts number of rows
- **SUM:** Calculates sum of values
- **AVG:** Computes average value
- **GROUP BY:** Groups rows that have same values
- **HAVING:** Filters groups based on conditions

```
SELECT department, AVG(salary) as avg_salary
FROM employees
GROUP BY department
HAVING AVG(salary) > 50000;
```

Use case: Calculating average salary by department

Joins

Combining rows from two or more tables

- **INNER JOIN:** Returns matching rows from both tables
- **LEFT JOIN:** Returns all rows from left table and matched rows from right
- **RIGHT JOIN:** Returns all rows from right table and matched rows from left
- **FULL OUTER JOIN:** Returns all rows when there's a match in either table

```
SELECT orders.id, customers.name
FROM orders
INNER JOIN customers ON orders.customer_id =
customers.id;
```

Use case: Combining order and customer data

Subqueries & CTEs

Nested queries and temporary result sets

- **Subquery:** Query nested inside another query
- **CTE:** Common Table Expression, temporary named result set

```
-- Subquery
SELECT name
FROM customers
WHERE id IN (SELECT customer_id FROM orders);

-- CTE
WITH active_customers AS (
  SELECT customer_id FROM orders
)
SELECT name FROM customers
WHERE id IN (SELECT customer_id FROM
active_customers);
```

Use case: Finding customers who have placed orders

Window Functions

Functions that operate across a set of table rows

- **ROW_NUMBER:** Assigns unique integer to rows
- **RANK:** Assigns rank to rows within partition
- **LAG:** Accesses data from previous row
- **LEAD:** Accesses data from following row

```
SELECT
  employee_id,
  salary,
  ROW_NUMBER() OVER (ORDER BY salary DESC) as
rank
FROM employees;
```

Use case: Ranking employees by salary

Data Cleaning with SQL

Preparing and standardizing data using SQL

- **TRIM:** Removes leading/trailing spaces
- **COALESCE:** Returns first non-null value
- **CASE WHEN:** Conditional logic in queries
- **CAST/CONVERT:** Changes data types

```
SELECT
  TRIM(name) as clean_name,
  COALESCE(phone, 'N/A') as phone,
  CASE WHEN age < 18 THEN 'Minor'
    WHEN age BETWEEN 18 AND 65 THEN 'Adult'
    ELSE 'Senior' END as age_group
FROM customers;
```

Use case: Standardizing customer data formats

Power BI

Data Loading & Modeling

Connecting to and structuring data sources

- **Data Sources:** Excel, SQL, databases, online services
- **Relationships:** Connecting tables with cardinality
- **Data Model:** Structuring tables for analysis
- **Refresh:** Updating data from sources

Use case: Building a sales data model from multiple sources

Visualizations

Visual representations of data insights

- **Cards:** Display single KPI values
- **Charts:** Bar, line, pie, scatter plots
- **KPIs:** Visual indicators with targets
- **Maps:** Geographic data visualization

Use case: Creating interactive sales dashboard

Power Query

Data transformation and preparation tool

- **Transform:** Clean and reshape data
- **Filter:** Select specific rows or columns
- **Merge:** Combine queries (like SQL JOINs)
- **Append:** Stack queries vertically

Use case: Standardizing customer data from different regions

Dashboard Design

Creating effective and user-friendly dashboards

- **Themes:** Consistent color schemes and fonts
- **Filters:** Slicers for interactive data exploration
- **Navigation:** Buttons and drill-through capabilities
- **Layout:** Logical arrangement of visuals

Use case: Executive dashboard with drill-down capabilities

DAX Basics

Data Analysis Expressions for calculations

- **Measures:** Dynamic calculations aggregated at runtime
- **Calculated Columns:** Static calculations in data model
- **Common Functions:** SUM, AVERAGE, CALCULATE, FILTER

```
Total Sales = SUM(Sales[Amount])
YoY Growth = DIVIDE([This Year], [Last Year]) - 1
```

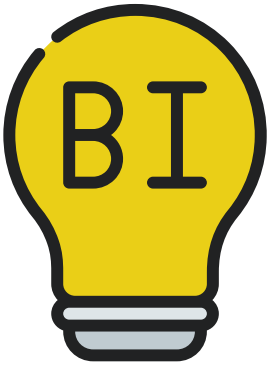
Use case: Creating year-over-year growth metrics

Real-Time Reports

Displaying up-to-the-minute data

- **Streaming Datasets:** Direct connections to live data
- **Automatic Refresh:** Scheduled data updates
- **Push Datasets:** Programmatically update data
- **Gateways:** Connect to on-premises data sources

Use case: Monitoring live production metrics



Git & GitHub

Version Control Basics

System that records changes to files over time

- **What:** Tracks file modifications, allows reverting to previous states
- **Why:** Collaboration, backup, experiment safely
- **How:** Commits, branches, repositories

Use case: Tracking changes in analysis scripts

Git Commands

Essential Git operations

- **init:** Create new repository
- **clone:** Copy existing repository
- **add:** Stage changes for commit
- **commit:** Save changes to repository
- **push:** Upload changes to remote
- **pull:** Download changes from remote

```
git add .
git commit -m "Added analysis script"
git push origin main
```

Use case: Saving and sharing analysis progress

Branching & Merging

Working on separate features simultaneously

- **Branch:** Independent line of development
- **Merge:** Combine branches together
- **Conflict Resolution:** Handle competing changes

```
git branch feature-x
git checkout feature-x
# Make changes
git checkout main
git merge feature-x
```

Use case: Developing new analysis features separately

Creating & Collaborating on Repositories

Working with others on shared code

- **Remote Repository:** Central storage on GitHub
- **Fork:** Copy repository to your account
- **Clone:** Download repository to local machine
- **Collaboration:** Team access with permissions

Use case: Team project on customer segmentation analysis

Using Git with Jupyter/VS Code

Integrating version control with development environments

- **Jupyter:** Track notebooks with .gitignore
- **VS Code:** Built-in Git integration
- **Extensions:** Enhanced Git functionality

```
# .gitignore for Jupyter
.ipynb_checkpoints/*
*.pyc
__pycache__/*
```

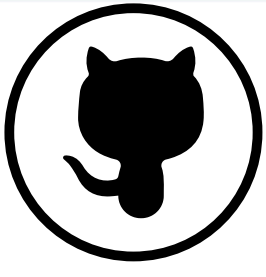
Use case: Version controlling data analysis notebooks

GitHub Issues, PRs & Contribution Workflow

Collaborative development process

- **Issues:** Track bugs, enhancements, tasks
- **Pull Requests (PRs):** Propose changes to repository
- **Code Review:** Collaborative feedback process
- **Fork → Clone → Edit → PR:** Standard contribution flow

Use case: Contributing to open-source data analysis tools



Other Tools & Concepts

Google Sheets

Cloud-based spreadsheet application

- **QUERY:** SQL-like data manipulation
- **ImportRange:** Import data from other sheets
- **ARRAYFORMULA:** Apply formula to entire range

```
=QUERY(A1:D10, "SELECT A, SUM(D) WHERE B > 100
GROUP BY A")
```

Use case: Collaborative sales tracking dashboard

Tableau

Data visualization and business intelligence tool

- **Basics:** Drag-and-drop interface
- **Charts:** Bar, line, pie, scatter, maps
- **Filters:** Interactive data exploration
- **Dashboards:** Multiple visualizations in one view

Use case: Creating interactive sales performance dashboard

BigQuery/Cloud SQL Basics

Cloud-based data warehouse and database services

- **BigQuery:** Serverless, highly scalable data warehouse
- **Cloud SQL:** Managed relational database service
- **Standard SQL:** Query language for both

```
SELECT product, SUM(sales) FROM dataset.table
GROUP BY product
```

Use case: Analyzing large-scale customer behavior data

APIs & JSON Basics

Interacting with web services and structured data

- **APIs:** Interfaces for software communication
- **REST:** Common API architecture style
- **JSON:** Lightweight data interchange format
- **Python Libraries:** requests, json

```
import requests
response = requests.get(url)
data = response.json()
```

Use case: Extracting weather data for sales analysis

Excel Power Query / Power Pivot

Advanced data transformation and modeling in Excel

- **Power Query:** Data connection and transformation
- **Power Pivot:** Data modeling and DAX formulas
- **Data Model:** Relationships between tables

Use case: Creating complex sales forecasting model in Excel

Keyboard Shortcuts

Time-saving key combinations for common tools

- **Excel:** Ctrl+Arrow (navigate), F2 (edit cell), Ctrl+Shift+L (filter)
- **SQL:** F5 (execute query), Ctrl+R (execute selected)
- **Python IDEs:** Ctrl+Enter (run cell), Shift+Enter (new cell)

Use case: Increasing productivity during data analysis

Case Studies

Sample Project Flow

Standard process for data analysis projects

- **Problem:** Define business question
- **Data:** Collect and prepare relevant data
- **Tools:** Select appropriate analytical methods
- **Solution:** Implement analysis and deliver insights

Example Flow:
Customer retention issue → Transaction data + CRM → Python + SQL → Churn prediction model

Real-World Examples

Common data analysis applications

- **Sales Analysis:** Revenue trends, product performance
- **Churn Prediction:** Identify customers likely to leave
- **A/B Testing:** Compare marketing campaign effectiveness
- **EDA:** Exploratory analysis for new datasets

Sales Analysis Example:
Identify top-performing products by region and season to optimize inventory

Visual Format Structure

Standard presentation of case study components

- **Problem Statement:** Clear business question
- **Tools Used:** Technologies and methods applied
- **Final Outcome:** Results and business impact

Example Structure:
Problem: Reduce customer acquisition cost → Tools: Python, SQL, Tableau → Outcome: 20% cost reduction

Key Business KPIs Tracked

Essential metrics for measuring business performance

- **Revenue Metrics:** Sales growth, average order value
- **Customer Metrics:** Retention rate, lifetime value
- **Operational Metrics:** Conversion rate, churn rate
- **Financial Metrics:** ROI, profit margins

E-commerce KPIs:
Conversion rate, average order value, customer acquisition cost, customer lifetime value

Interview Questions

Top 10 Technical Questions

Common technical questions by category

- **SQL:** JOINs, window functions, subqueries
- **Statistics:** Hypothesis testing, p-values, distributions
- **Python:** Pandas operations, data cleaning
- **Excel:** VLOOKUP, PivotTables, complex formulas

Example Question:
"Explain the difference between INNER JOIN and LEFT JOIN and when you would use each."

Answer Structure Tips

Effective ways to organize your responses

- **Start with overview:** Briefly explain your approach
- **Detail your process:** Step-by-step methodology
- **Discuss alternatives:** Show you've considered other options
- **Explain trade-offs:** Demonstrate critical thinking

Structure Example:
"First, I would... Then, I would... Finally, I would..."

Scenario-Based Questions

Real-world problem-solving scenarios

- **EDA:** "How would you approach exploring a new dataset?"
- **Dashboarding:** "Design a dashboard for tracking sales performance"
- **Business Decisions:** "How would you determine if a marketing campaign was successful?"

Example Scenario:
"You're given a dataset with customer information and purchase history. How would you identify customers at risk of churning?"

STAR Method & Communication

Framework for behavioral questions and soft skills

- **STAR:** Situation, Task, Action, Result
- **Situation:** Describe the context
- **Task:** Explain what needed to be done
- **Action:** Detail steps you took
- **Result:** Share outcomes and learnings

Soft Skills Tips:
"Communicate technical concepts clearly to non-technical stakeholders. Emphasize collaboration and business impact."