

Module2 Syllabus

2.1 Exploratory Data Analysis and the Data Science Process: Basic tools (plots, graphs and summary statistics) of EDA, Philosophy of EDA, The Data Science Process,

2.2 Case Study: Real Direct (online real estate firm).

2.3 Three Basic Machine Learning Algorithms: Linear Regression, k-Nearest Neighbours (k- NN), k-means.

2.1.1 Exploratory Data Analysis (EDA)

Exploratory Data Analysis (EDA) is often overlooked in statistics textbooks, typically relegated to introductory sections that are considered basic and elementary. It's commonly taught using simple techniques like **histograms and stem-and-leaf plots**, which can seem trivial since they're introduced to children in fifth grade. However, EDA is actually a crucial part of the **data science process** and reflects a philosophical approach to statistics advocated by certain **statisticians**, particularly those influenced by the **Bell Labs** tradition.

Exploratory Data Analysis (EDA) involves using plots, graphs, and summary statistics to systematically analyze data. This includes plotting distributions of variables (such as box plots), examining time series data, transforming variables, exploring pairwise relationships between variables using scatterplot matrices, and computing summary statistics like mean, minimum, maximum, quartiles, and identifying outliers.

However, EDA is more than just a set of tools—it's a mindset focused on developing a **deep relationship** with the data. The goal is to gain intuition, understand the data's shape, and connect this understanding to the underlying process that generated the data. EDA is a personal exploration between the analyst and the data, without the need to prove anything to others at this stage.

EDA happens between you and the data and isn't about proving anything to anyone else yet.

• What is Exploratory Data Analysis?

Exploratory Data Analysis (EDA) is an approach to analyzing datasets to summarize their main characteristics, often employing visual methods. The primary goal of EDA is to gain insight into the data, understand its underlying structure, detect patterns, identify anomalies, and formulate hypotheses for subsequent analysis. EDA involves examining the data through various graphical representations, statistical summaries, and data transformation techniques without presupposing any formal hypotheses. This process helps in understanding the nature of the data, revealing relationships between variables, and guiding further analysis or modeling decisions. EDA is considered an essential initial step in data analysis, providing a preliminary understanding of the dataset before more complex statistical techniques are applied.

The basic tools of EDA are plots, graphs and summary statistics. **The terms "plots" and "graphs"** are often used interchangeably in the context of data visualization, but they can have slightly different meanings depending on the context. Here's a breakdown of the difference:

• Plots:

Definition: Plots refer specifically to visual representations of data using graphical elements such as points, lines, bars, or other symbols to display relationships, distributions, or trends within the data.

Examples: Scatter plots, line plots, bar plots, box plots, histograms, pie charts, heatmaps, etc.

Purpose: Plots are used to visually explore and analyze data, making it easier to identify patterns, trends, outliers, and relationships between variables.

• Graphs:

Definition: Graphs, in a broader sense, can refer to any visual representation of data or mathematical relationships. In some contexts, graphs specifically refer to mathematical structures represented by nodes (vertices) and edges (connections) that illustrate relationships between entities.

Examples: Line graphs, network graphs (nodes and edges), flowcharts, tree diagrams, Venn diagrams, etc.

Purpose: Graphs can be used to represent various types of information beyond just data visualization, including mathematical relationships, hierarchical structures, processes, or relationships between entities.

In summary, while "plots" typically refer to specific types of visual data representations used for analysis and exploration, "graphs" can encompass a broader range of visual tools and representations used in different contexts beyond data analysis. The specific usage of these terms can vary depending on the field and context of discussion.

• Statistical Summary

A statistical summary is a concise description or summary of key characteristics or features of a dataset or a sample of data. It provides numerical measures that describe the central tendency, variability, and distribution of the data. Common statistical summaries include:

Measures of Central Tendency:

1. **Mean:** Average value of the data points.
2. **Median:** Middle value of the dataset when arranged in order.
3. **Mode:** Most frequently occurring value(s) in the dataset.

Measures of Variability:

1. **Range:** Difference between the maximum and minimum values.
2. **Variance:** Average of the squared differences from the mean.
3. **Standard Deviation:** Square root of the variance, indicating the spread of data around the mean.

Summary of Distribution:

1. **Quartiles:** Values that divide the dataset into four equal parts.
2. **Interquartile Range (IQR):** Range of values between the first and third quartiles.
3. **Skewness and Kurtosis:** Measures of asymmetry and tail heaviness of the distribution.

Other Summary Statistics:

1. **Sum:** Total sum of all data points.
2. **Count:** Number of observations in the dataset.
3. **Percentiles:** Values below which a certain percentage of data falls.

Statistical summaries provide a compact and meaningful way to understand the essential characteristics of data, aiding in the exploratory data analysis process and informing further statistical modeling or analysis.

2.1.2 Philosophy of Exploratory Data Analysis

Quote: Long before worrying about how to convince others, you first have to understand what's happening yourself.

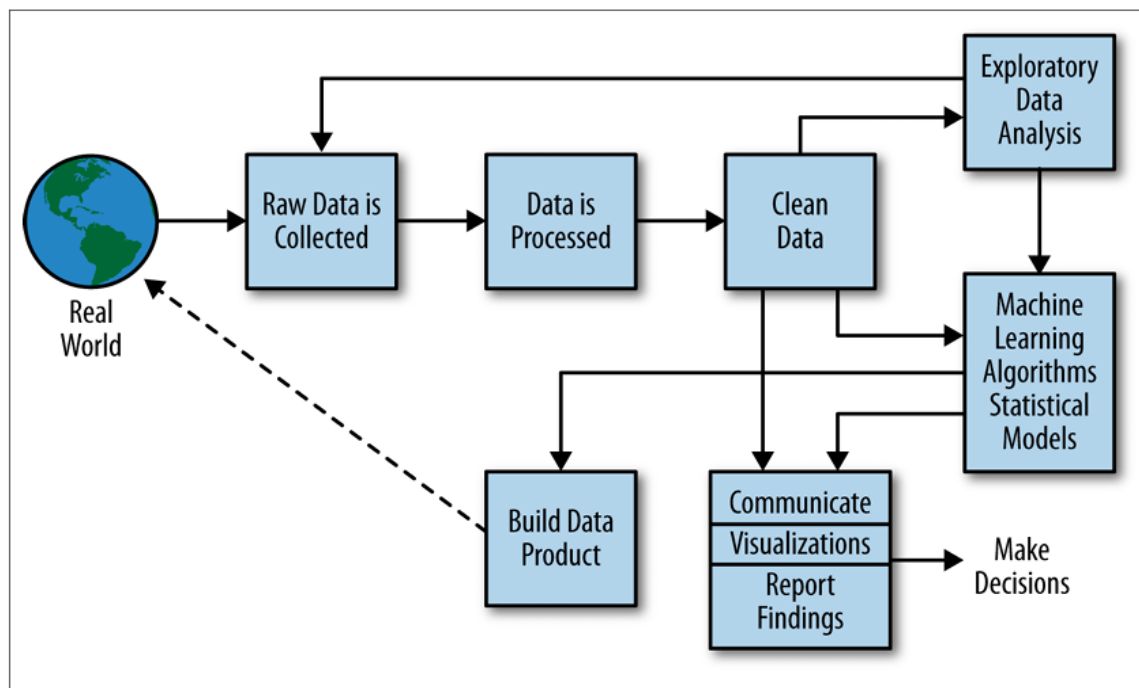
— **Andrew Gelman**

Rachel, during her time at Google, collaborated with former Bell Labs/AT&T statisticians Daryl Pregibon and Diane Lambert, who emphasized the importance of integrating Exploratory Data Analysis (EDA) into best practices. Even with vast Google-scale datasets, EDA remains essential. EDA is conducted not only to gain intuition about the data, compare distributions, and ensure data sanity, but also to identify missing data, outliers, and debug issues within the logging process.

In the context of log-generated data, EDA plays a crucial role in debugging and ensuring accurate patterns. It helps engineers identify and correct potential issues that could mislead data interpretation. Ultimately, EDA contributes to ensuring products perform as intended.

While EDA involves extensive visualization, it differs from data visualization used for communicating findings later in the analysis process. EDA focuses on personal understanding rather than external communication. It empowers algorithm development by informing decisions on metrics like "popularity" based on data behavior, guiding improvements and algorithm design. Conducting thorough EDA, including plotting and comparisons, is emphasized over premature regression analysis, ensuring a more robust and informed approach to data analysis.

2.1.3 The Data Science Process



The diagram represents the overarching data science process as described in the provided content. Let me walk through how the different components align:

Real World: This represents the real-world phenomena, activities, and sources from which raw data is collected, such as user interactions, sensor readings, transaction records, etc.

Raw Data is Collected: The first step is to gather the raw data from these real-world sources, which could be logs, records, emails, or any other form of data.

Data is Processed: The raw data is then processed, cleaned, and wrangled into a structured format suitable for analysis. This may involve joining datasets, handling missing values, removing duplicates, and more, using tools like Python, R, SQL, or shell scripts.

Clean Data: The result of the data processing step is a clean, structured dataset ready for exploration and analysis.

Exploratory Data Analysis: At this stage, the clean data is analyzed to uncover patterns, trends, and insights through techniques like visualization, statistical summaries, and initial modeling.

Machine Learning Algorithms/Statistical Models: Based on the exploratory analysis, appropriate machine learning algorithms or statistical models are

selected and applied to the data to solve problems like classification, prediction, or description.

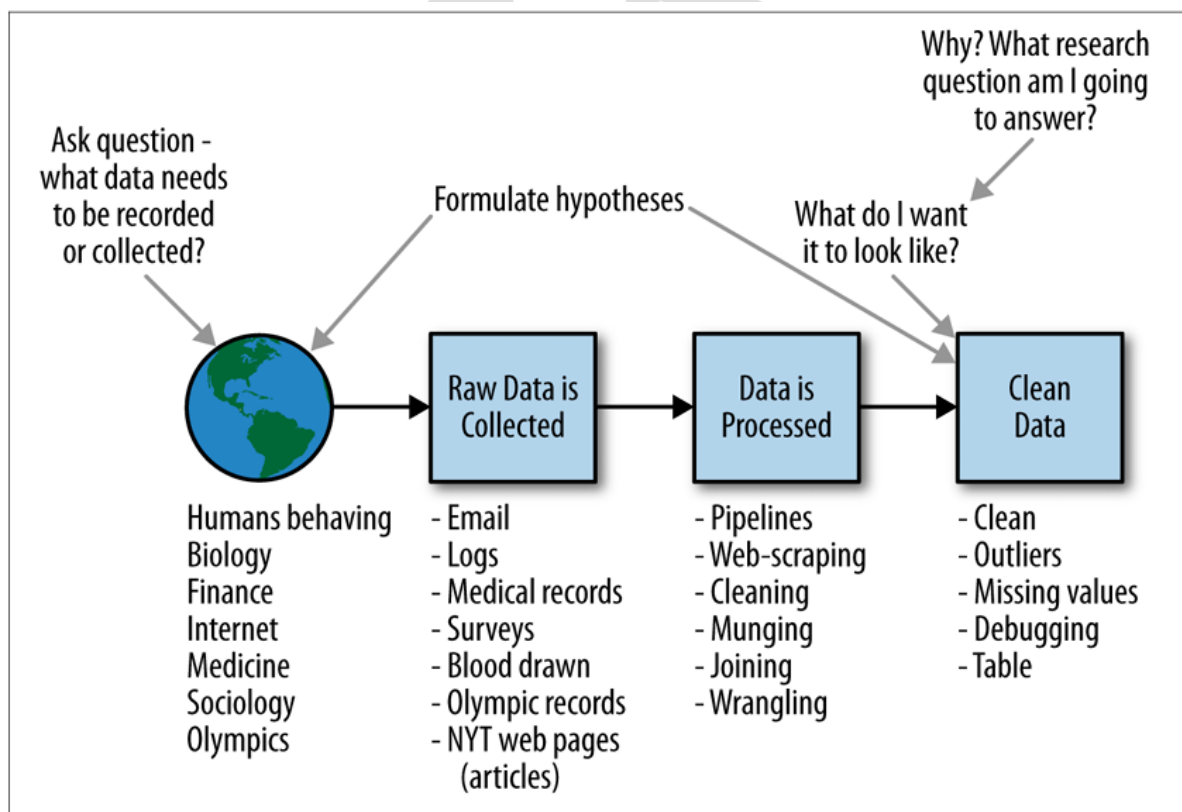
Communicate Visualizations/Report Findings: The insights and results from the analysis are communicated through visualizations, reports, presentations, or publications.

Build Data Product: In some cases, the goal is to build a data product, such as a recommendation system, spam classifier, or search ranking algorithm, which gets deployed and incorporated back into the real world.

Make Decisions: The analysis and findings ultimately inform decision-making processes.

The key aspect highlighted is the feedback loop, where the deployed data product or decisions made based on the analysis can influence and generate new data in the real world, which then becomes the input for the next iteration of the data science process.

• A Data Scientist's Role in This Process



The diagram illustrates the involvement and decision-making role of the data scientist throughout the data science process.

The data scientist starts by formulating the research question or hypothesis they want to answer or test. Based on this, they determine what kind of data needs to be collected or recorded from real-world sources like human behavior, biology, finance, internet activities, medicine, sociology, and others listed.

The raw data can come from various sources such as emails, logs, medical records, surveys, blood samples, Olympic records, web pages or articles. This raw data is then collected under the guidance of the data scientist.

Next, the data scientist decides on the appropriate data processing techniques like pipelines, web scraping, cleaning, munging, joining, and wrangling to transform the raw data into a clean, structured dataset. The clean data may involve handling outliers, missing values, deduplication, debugging, and putting the data into a tabular format.

The data scientist also considers what the ideal, clean dataset should look like to effectively answer the research question. They may need to formulate specific hypotheses to test based on the research objective.

Throughout this process, the data scientist is actively involved in decision-making, guiding the collection, processing, and shaping of the data to enable effective analysis and insights. Their role is crucial in steering the entire data science workflow from formulating questions to obtaining an analysis-ready dataset.

• **Connection to the Scientific Method**

We can think of the data science process as an extension of or variation of the scientific method:

- Ask a question.
- Do background research.
- Construct a hypothesis.
- Test your hypothesis by doing an experiment.
- Analyze your data and draw a conclusion.
- Communicate your results.

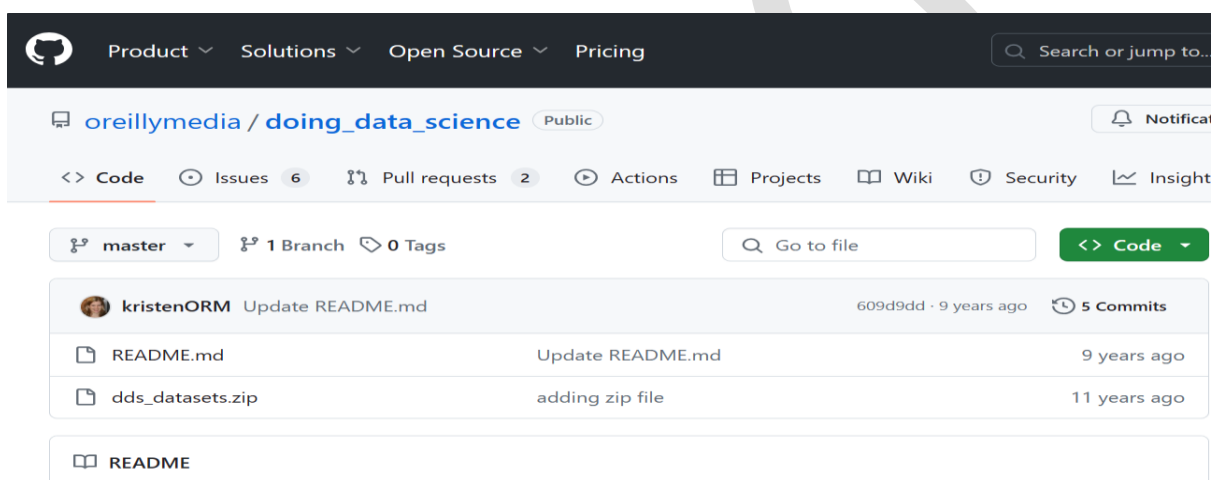
In both the data science process and the scientific method, not every problem requires one to go through all the steps, but almost all problems can be solved with some combination of the stages. For example if your end goal is a data

visualization (which itself could be thought of as a data product), it's possible you might not do any machine learning or statistical modelling, but you'd want to get all the way to a clean dataset, do some exploratory analysis, and then create the visualization.

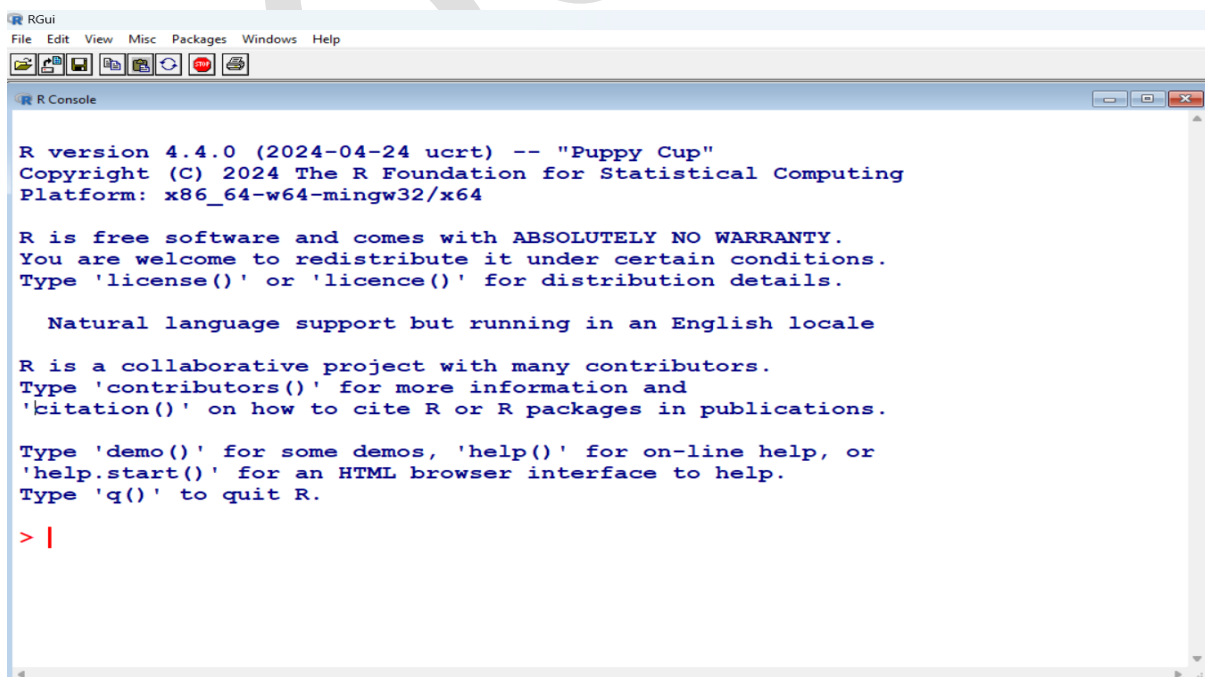
Exploratory Data Analysis Example:

There are 31 datasets named **nyt1.csv, nyt2.csv, ..., nyt31.csv**, which you can find here:

https://github.com/oreillymedia/doing_data_science

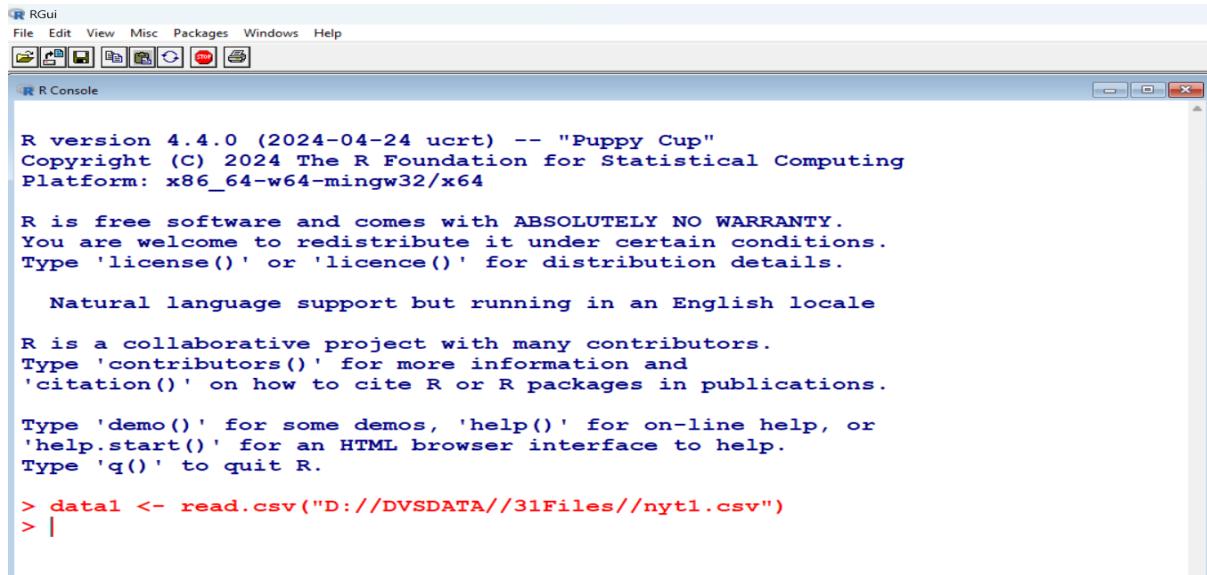


- Download and Install R: <https://www.r-project.org/>



Read csv file say nyt1.csv into a variable data1 as illustrated below:

```
data1 <- read.csv("D://DVSDATA// 31Files// nyt1.csv")
```



```
RGui
File Edit View Misc Packages Windows Help
R Console
R version 4.4.0 (2024-04-24 ucrt) -- "Puppy Cup"
Copyright (C) 2024 The R Foundation for Statistical Computing
Platform: x86_64-w64-mingw32/x64

R is free software and comes with ABSOLUTELY NO WARRANTY.
You are welcome to redistribute it under certain conditions.
Type 'license()' or 'licence()' for distribution details.

Natural language support but running in an English locale

R is a collaborative project with many contributors.
Type 'contributors()' for more information and
'citation()' on how to cite R or R packages in publications.

Type 'demo()' for some demos, 'help()' for on-line help, or
'help.start()' for an HTML browser interface to help.
Type 'q()' to quit R.

> data1 <- read.csv("D://DVSDATA//31Files//nyt1.csv")
> |
```

- Read top six rows of the data set using head () as illustrated below:

```
> data1 <- read.csv("D://DVSDATA//31Files//nyt1.csv")
> head(data1)
  Age Gender Impressions Clicks Signed_In
1  36      0           3       0         1
2  73      1           3       0         1
3  30      0           3       0         1
4  49      1           3       0         1
5  47      1          11       0         1
6  47      0          11       1         1
> |
```

- Display the datasets based on different age category as illustrated below:

```
> data1$agecat <-cut(data1$Age,c(-Inf,0,18,24,34,44,54,64,Inf))
> head(data1)
  Age Gender Impressions Clicks Signed_In  agecat
1  36      0           3       0         1 (34,44]
2  73      1           3       0         1 (64, Inf]
3  30      0           3       0         1 (24,34]
4  49      1           3       0         1 (44,54]
5  47      1          11       0         1 (44,54]
6  47      0          11       1         1 (44,54]
> |
```

- Display the statistical summary as illustrated below:

```
> # view
> summary(data1)
      Age          Gender      Impressions      Clicks
Min.   : 0.00   Min.   :0.000   Min.   : 0.000   Min.   :0.00000
1st Qu.: 0.00   1st Qu.:0.000   1st Qu.: 3.000   1st Qu.:0.00000
Median : 31.00  Median :0.000   Median : 5.000   Median :0.00000
Mean   : 29.48  Mean   :0.367   Mean   : 5.007   Mean   :0.09259
3rd Qu.: 48.00  3rd Qu.:1.000   3rd Qu.: 6.000   3rd Qu.:0.00000
Max.   :108.00  Max.   :1.000   Max.   :20.000   Max.   :4.00000

      Signed_In      agecat
Min.   :0.0000   (-Inf,0]:137106
1st Qu.:0.0000   (34,44] : 70860
Median :1.0000   (44,54] : 64288
Mean   :0.7009   (24,34] : 58174
3rd Qu.:1.0000   (54,64] : 44738
Max.   :1.0000   (18,24] : 35270
              (Other) : 48005

> |
```

- Install packages called “doBY” as illustrated below:

```
> # brackets
> install.packages("doBy")
Warning in install.packages("doBy") :
  'lib = "C:/Program Files/R/R-4.4.0/library"' is not writable
--- Please select a CRAN mirror for use in this session ---
also installing the dependencies 'crayon', 'fs', 'pkgbuild', 'rprojroot', 'diff$

trying URL 'https://cran.icts.res.in/bin/windows/contrib/4.4/crayon_1.5.2.zip'
Content type 'application/zip' length 164120 bytes (160 KB)
downloaded 160 KB

trying URL 'https://cran.icts.res.in/bin/windows/contrib/4.4/fs_1.6.4.zip'
Content type 'application/zip' length 413207 bytes (403 KB)
downloaded 403 KB

trying URL 'https://cran.icts.res.in/bin/windows/contrib/4.4/pkgbuild_1.4.4.zip'
Content type 'application/zip' length 204134 bytes (199 KB)
downloaded 199 KB
```

```

> library("doBy")
> siterange <- function(x){c(length(x), min(x), mean(x), max(x))}
> summaryBy(Age~agecat, data =data1, FUN=siterange)
  agecat Age.FUN1 Age.FUN2 Age.FUN3 Age.FUN4
1  (-Inf,0]   137106      0 0.00000      0
2   (0,18]    19252      7 16.03350     18
3  (18,24]    35270     19 21.26904     24
4  (24,34]    58174     25 29.50335     34
5  (34,44]    70860     35 39.49468     44
6  (44,54]    64288     45 49.49258     54
7  (54,64]    44738     55 59.49819     64
8 (64, Inf]    28753     65 72.98870    108
> |

```

- Display the statistical summary of signed in users as illustrated below:

```

> # so only signed in users have ages and genders
> summaryBy(Gender+Signed_In+Impressions+Clicks~agecat,data
+ =data1)
  agecat Gender.mean Signed_In.mean Impressions.mean Clicks.mean
1  (-Inf,0]  0.0000000      0      4.999657  0.14207985
2   (0,18]  0.6421151      1      4.998961  0.13105132
3  (18,24]  0.5338531      1      5.006635  0.04845478
4  (24,34]  0.5321621      1      4.993829  0.05048647
5  (34,44]  0.5316963      1      5.021507  0.05167937
6  (44,54]  0.5289790      1      5.010406  0.05027377
7  (54,64]  0.5361885      1      5.022308  0.10183736
8 (64, Inf]  0.3632664      1      5.012347  0.15128856
> |

```

Drawing Plots

- Install Package: `install.packages("ggplot2")`

```

> # plot
> install.packages("ggplot2")
Installing package into 'C:/Users/proft/AppData/Local/R/win-library$
(as 'lib' is unspecified)
trying URL 'https://cran.icts.res.in/bin/windows/contrib/4.4/ggplot$
Content type 'application/zip' length 5011490 bytes (4.8 MB)
downloaded 4.8 MB

package 'ggplot2' successfully unpacked and MD5 sums checked

```

The downloaded binary packages are in

```

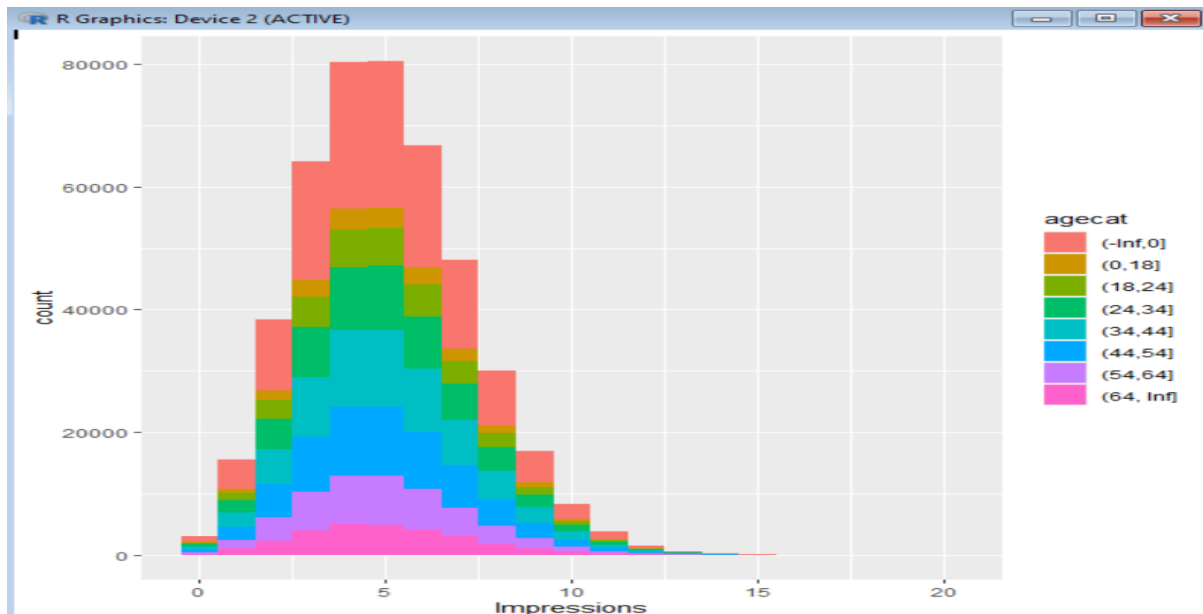
C:\Users\proft\AppData\Local\Temp\RtmpaAfXim\downloaded_pac$

```

Load Library: `library(ggplot2)`

Draw Histogram:

```
ggplot(data1, aes(x=Impressions, fill=agecat)) +geom_histogram(binwidth=1)
```



Box Blot

A box plot, also known as a box-and-whisker plot, is a graphical representation of a dataset that displays its distribution and central tendency. It is particularly useful for visualizing the spread and skewness of the data, as well as identifying outliers. Here are the main components of a box plot:

Box: The box itself represents the interquartile range (IQR), which is the range between the first quartile (Q1) and the third quartile (Q3). This range contains the middle 50% of the data.

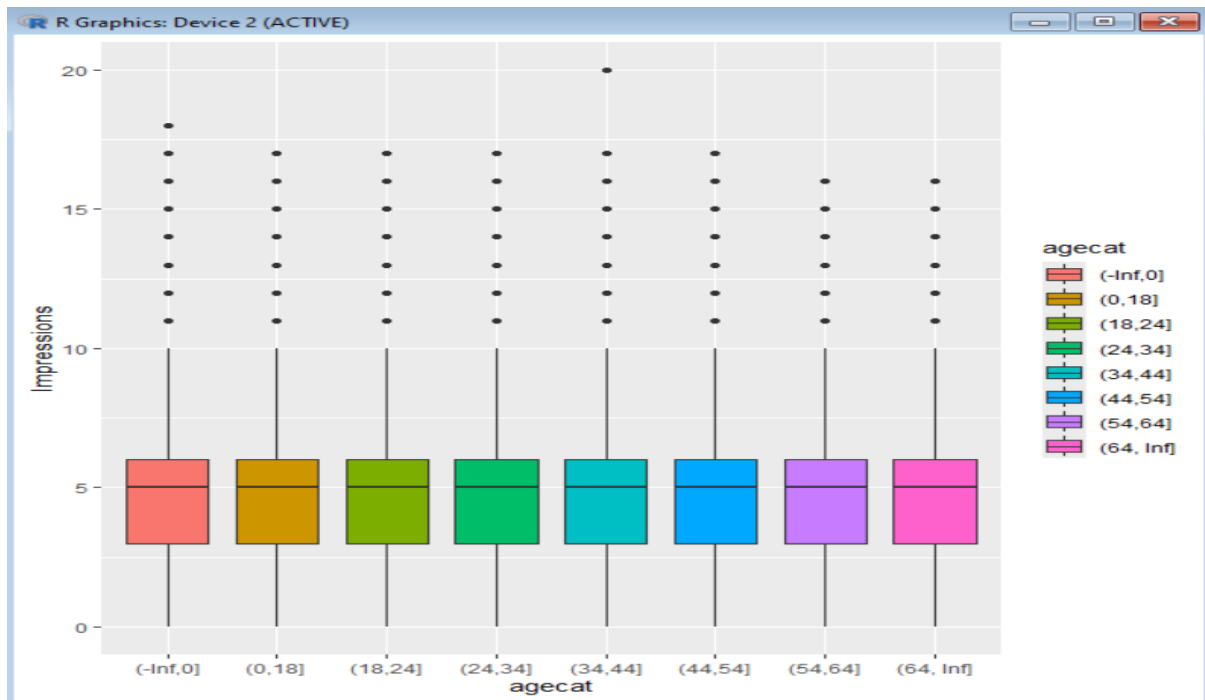
Median Line: A line inside the box marks the median (Q2) of the dataset.

Whiskers: Lines extending from the box to the smallest and largest values within 1.5 times the IQR from Q1 and Q3, respectively. These lines show the range of the data excluding outliers.

Outliers: Data points that fall outside of the whiskers. They are typically plotted as individual points beyond the whiskers.

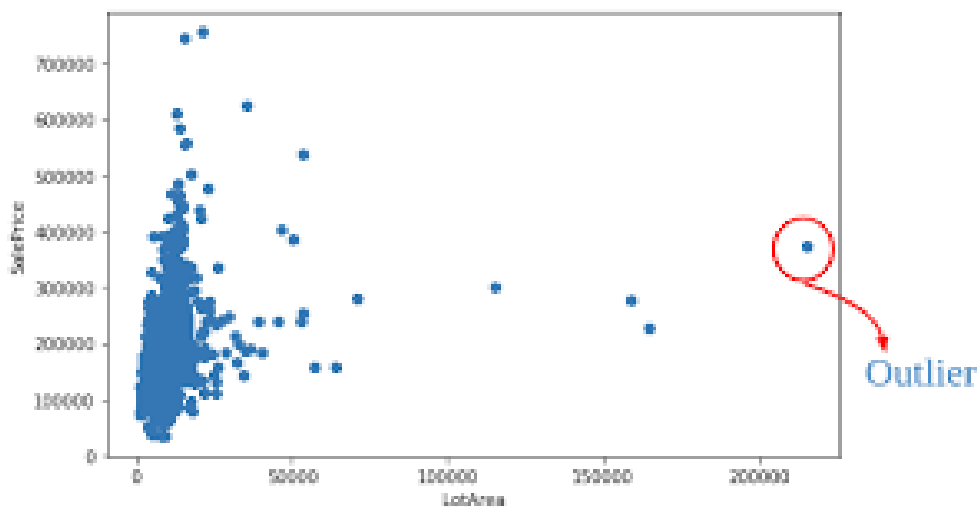
Draw Box Plot:

```
ggplot(data1, aes(x=agecat, y=Impressions, fill=agecat))+geom_boxplot()
```



Outliers

- In data science, outliers are data points that significantly deviate from the rest of the dataset. These anomalies can occur due to variability in the data or errors in the data collection process. Identifying and handling outliers is crucial because they can significantly affect the results of data analysis and statistical modeling.



Example of Outliers

Consider a dataset representing the ages of students in a university class:

Ages=[18,19,20,21,22,23,24,25,26,27,40]

In this dataset, most students' ages range from **18 to 27 years**. However, there is one student aged 40, which is significantly higher than the rest. This age of 40 is an outlier.

Thought Experiment:

How Would You Simulate Chaos? [Simulating Chaos: Exploring Order from Disorder]

- Most data problems start out with a certain amount of dirty data, ill-defined questions, and urgency. As data scientists we are, in a sense, attempting to create order from chaos.
- **Lorenzian Water Wheel: Ferris wheel-like apparatus** with rotating water buckets, exhibiting chaotic behavior influenced by molecular interactions.
- **Digital Chaos Simulations:** Binder and Jensen's paper explores **chaos using finite-state machines in digital simulations**.
- **Interdisciplinary Program:** M.I.T., Harvard, and Tufts collaborated on a **chaos simulation involving a humanitarian crisis** scenario in Chad-Sudan border region.
- **Creating Order in Startups:** Gascoigne's essay discusses strategies for **managing chaos within startup environments** to establish order amidst uncertainty and complexity.

Exploring Chaos and Simulation in Data Science

- **Simulation serves as a vital tool in data science**, aiding in understanding model behavior and debugging code
- **Data scientists** navigate **organizational chaos**, striving to impose order and learn from chaotic experiences.

Case Study: RealDirect

- **Doug Perlson, CEO of RealDirect, brings expertise in real estate law, startups, and online advertising.**
- His vision for RealDirect is to *leverage comprehensive real estate data to enhance the process of buying and selling homes.*
- Typically, homeowners sell their properties approximately *every seven years* with assistance from **brokers** and relying on **current data**.
- However, challenges persist within the **broker system** and **data quality**. **RealDirect** aims to tackle these issues head-on.
- Initially, **brokers operate** as independent "**free agents**" akin to home *sales consultants, guarding their data and relying heavily on experience.*
- However, in reality, even experienced **brokers have** only marginally more data than **novices**.
- **RealDirect** tackles this issue by assembling a **team of licensed real estate agents** who **collaborate and share knowledge**.
- This involves creating a **seller interface with data-driven tips for selling homes** and **utilizing interaction data to offer real-time recommendations** for next steps.
- Brokers are **trained to become proficient** in using **information-gathering tools** to monitor new and pertinent data and access publicly available information.
- For instance, **accessing data on co-op (a specific type of apartment in NYC) sales is now feasible**, although this capability is a recent development.
- Brokers are **trained to become proficient** in using **information-gathering tools** to monitor new and pertinent data and access publicly available information.
- For instance, **accessing data on co-op (a specific type of apartment in NYC) sales is now feasible**, although this capability is a recent development.
- **Publicly available data often suffers** from a **significant time delay**, with information about sales taking **up to three months** to become accessible.
- RealDirect is **actively developing real-time data feeds**, covering crucial aspects such as **home search initiation, initial offers, duration between offer and closure, and online search patterns**.
- Ultimately, access to accurate and timely information benefits both *buyers and sellers, assuming transparency and honesty in the transaction process.*

How Does RealDirect Make Money?






- First, it **offers a subscription to sellers**—about **\$395 a month**—to access the **selling tools**.
- Second, it allows sellers to use **RealDirect's agents** at a reduced commission, typically **2% of the sale instead** of the usual 2.5% or 3%.
- The site itself is best thought of as a **platform for buyers and sellers to manage their sale or purchase process**.
- There are statuses for each person on site:
 - **active,**
 - **offer made,**
 - **offer rejected,**
 - **showing,**
 - **in contract, etc.**
- Based on your status, **different actions** are suggested by the software
- There are some challenges they have to deal with as well, of course. First off, there's a law in **New York** that says you can't show all the **current housing listings unless** those listings reside behind a registration wall, so **RealDirect** requires registration. On the one hand, this is an obstacle for buyers, but serious buyers are likely willing to do it.
- **Doug** discussed **important factors** that buyers prioritize, such as
 - **proximity to parks,**
 - **subway stations, and**
 - **schools,**
 - **along with comparisons of apartment prices per square foot within the same building or block.**
- **RealDirect** aims to **expand its services** by providing comprehensive coverage of this type of data to **meet buyer preferences**.

Exercise: RealDirect Data Strategy

You have been hired as chief data scientist at **realdirect.com**, and report directly to the CEO. The company (hypothetically) does not yet have its data plan in place. It's looking to you to come up with a **data strategy**.

Here are a couple ways you could begin to approach this problem:

1. **Explore its existing website**, thinking about how buyers and sellers would navigate through it, and how the website is **structured/ organized**. Try to understand the existing business model, and think about how analysis of **RealDirect** user-behavior data could be used to inform decision-making and product development. Come up with a list of research questions you think could be answered by data:
 - What data would you advise the engineers log and what would your ideal datasets look like?
 - How would data be used for reporting and monitoring product usage?
 - How would data be built back into the product/website?
2. **Because there is no data yet for you to analyze** (typical in a start up when its still building its product), you should get some auxiliary data to **help gain intuition about this market**.
 - For example, go to https://github.com/oreillymedia/doing_data_science. Click on **Rolling Sales Update (after the fifth paragraph)**. You can use any or all of the datasets here.

 rollingsales_bronx	14-05-2024 05:32	Microsoft Excel 97...	1,508 KB
 rollingsales_brooklyn	14-05-2024 05:32	Microsoft Excel 97...	6,454 KB
 rollingsales_manhattan	14-05-2024 05:32	Microsoft Excel 97...	7,193 KB
 rollingsales_queens	14-05-2024 05:32	Microsoft Excel 97...	6,582 KB
 rollingsales_statenisland	14-05-2024 05:32	Microsoft Excel 97...	1,814 KB

Snapshot of Bronx Rolling Sales File

rollingsales_bronx [Compatibility Mode] - Excel (Product Activation Failed)

File Home Insert Page Layout Formulas Data Review View Tell me what you want to do... Sign in Share

Clipboard Font Alignment Number Styles Cells Editing

1 **Bronx Rolling Sales File. All Sales From August 2012 - August 2013.**

2 Sales File as of 08/30/2013 Coop Sales Files as of 09/18/2013

3 Neighborhood Name 09/06/13, Descriptive Data is as of 06/01/13

4 Building Class Category is based on Building Class at Time of Sale.

BOROUGH	NEIGHBORHOOD	BUILDING CLASS CATEGORY	TAX CLASS AT PRESEN	BLOCK	LOT	EASEMENT	BUILDING CLASS AT PRESEN	ADDRESS
2	BATHGATE	01 ONE FAMILY HOMES	1	3028	25		A5	412 EAST 179TH STREET
2	BATHGATE	01 ONE FAMILY HOMES	1	3039	28		A1	2329 WASHINGTON AVENUE
2	BATHGATE	01 ONE FAMILY HOMES	1	3046	39		A1	2075 BATHGATE AVENUE
2	BATHGATE	01 ONE FAMILY HOMES	1	3046	52		A1	2047 BATHGATE AVENUE
2	BATHGATE	02 TWO FAMILY HOMES	1	2900	61		S2	406 EAST TREMONT AVENUE
2	BATHGATE	02 TWO FAMILY HOMES	1	2912	158		B1	505 EAST 171ST STREET
2	BATHGATE	02 TWO FAMILY HOMES	1	2929	117		B1	3860 3 AVENUE
2	BATHGATE	02 TWO FAMILY HOMES	1	3030	60		B3	4469 PARK AVENUE
2	BATHGATE	02 TWO FAMILY HOMES	1	3035	27		B1	454 EAST 179 STREET
2	BATHGATE	02 TWO FAMILY HOMES	1	3039	65		B2	465 EAST 185 STREET
2	BATHGATE	02 TWO FAMILY HOMES	1	3040	5		S2	4654-4656 PARK AVENUE
2	BATHGATE	02 TWO FAMILY HOMES	1	3046	54		B2	2043 BATHGATE AVENUE
2	BATHGATE	02 TWO FAMILY HOMES	1	3050	85		B1	2241 BATHGATE AVENUE
2	BATHGATE	02 TWO FAMILY HOMES	1	3052	37		S2	4557 3 AVENUE

Ready

Snapshot of Staten Island Rolling Sales File

rollingsales_statenisland [Compatibility Mode] - Excel (Product Activation Failed)

File Home Insert Page Layout Formulas Data Review View Tell me what you want to do... Sign in Share

Clipboard Font Alignment Number Styles Cells Editing

1 **Staten Island Rolling Sales File. All Sales From August 2012 - August 2013.**

2 Sales File as of 08/30/2013 Coop Sales Files as of 09/18/2013

3 Neighborhood Name 09/06/13, Descriptive Data is as of 06/01/13

4 Building Class Category is based on Building Class at Time of Sale.

BOROUGH	NEIGHBORHOOD	BUILDING CLASS CATEGORY	TAX CLASS AT PRESEN	BLOCK	LOT	EASEMENT	BUILDING CLASS AT PRESEN	ADDRESS
5	ANNADALE	01 ONE FAMILY HOMES	1	5395	32		A1	541 SYCAMORE STREET
5	ANNADALE	01 ONE FAMILY HOMES	1	5401	10		A2	16 JANSEN STREET
5	ANNADALE	01 ONE FAMILY HOMES	1	5401	38		A1	27 WEAVER STREET
5	ANNADALE	01 ONE FAMILY HOMES	1	5407	11		A1	24 ELMBANK STREET
5	ANNADALE	01 ONE FAMILY HOMES	1	5425	39		A1	23 SANDGAP STREET
5	ANNADALE	01 ONE FAMILY HOMES	1	6205	16		A5	93 EAGAN AVENUE
5	ANNADALE	01 ONE FAMILY HOMES	1	6205	55		A5	36 SEGUINE PLACE
5	ANNADALE	01 ONE FAMILY HOMES	1	6205	126		A5	20 MAY PLACE
5	ANNADALE	01 ONE FAMILY HOMES	1	6211	20		A5	9 EAGAN AVENUE
5	ANNADALE	01 ONE FAMILY HOMES	1	6212	28		A1	96 JEANNETTE AVENUE
5	ANNADALE	01 ONE FAMILY HOMES	1	6212	44		A5	50 LUCY LOOP
5	ANNADALE	01 ONE FAMILY HOMES	1	6212	85		A1	1086 ARDEN AVENUE
5	ANNADALE	01 ONE FAMILY HOMES	1	6216	32		A1	8 EAGAN AVENUE
5	ANNADALE	01 ONE FAMILY HOMES	1	6217	32		A2	10 FABIAN STREET

Ready

3. Load, Clean and Conduct Exploratory Data Analysis

First challenge: load in and clean up the data. **Next,** conduct exploratory data analysis in order to find out where there are **outliers or missing values**, decide how you will treat them, make sure the dates are formatted correctly,

make sure values you think are numerical are being treated as such, etc. **Once the data is in good shape**, conduct exploratory data analysis to visualize and make comparisons

- I. across neighborhoods, and
- II. across time.

4. Summarize your findings in a brief report aimed at the CEO

5. Can you think of any other people you should talk to?

Being the “**data scientist**” often involves speaking to people who aren’t also data scientists, so it would be ideal to have a set of communication strategies for getting to the information you need about the data. Can you think of any other people you should talk to?

6. Acquire Domain Knowledge

Most of you are not “**domain experts**” in real estate or online businesses. Does stepping out of your comfort zone and figuring out how you would go about “**collecting data**” in a different setting give you insight into how you do it in your own field?

Sometimes “**domain experts**” have their own set of vocabulary. Did Doug use vocabulary specific to his domain that you didn’t understand (“**comps,**” “**open houses,**” “**CPC**”)?

7. Work on Best Practices for Data Strategy

- **Doug** mentioned the company didn’t necessarily have a data strategy. There is **no industry standard for creating one**.
- As you work through this assignment, think about whether there is a **set of best practices you would recommend** with respect to developing a data strategy for an online business, or in your own domain.

Sample R code

Here's some sample R code that takes the Brooklyn housing data in the preceding exercise, and cleans and explores it a bit.

Install packages and read the `rollingsales_brooklyn.xls` into data frame `bk`.

```
install.packages("readxl")
library(readxl)
bk <-
read.xls("D://DVSDATA//rollingsales_brooklyn.xls",pattern="BOROUGH")
```

Display Top 6 rows of Data frame bk

head(bk)

```
> head(bk)
# A tibble: 6 × 21
  BOROUGH NEIGHBORHOOD BUILDING CLASS CATEG...1 `TAX CLASS AT PRESENT` BLOCK LOT
    <dbl> <chr>          <chr>          <chr>          <dbl> <dbl>
1     3 <NA>          15 CONDOS - 2-10 UNI... <NA>      814 1103
2     3 <NA>          15 CONDOS - 2-10 UNI... <NA>      814 1105
3     3 <NA>          15 CONDOS - 2-10 UNI... <NA>     1967 1401
4     3 <NA>          15 CONDOS - 2-10 UNI... <NA>     1967 1402
5     3 <NA>          15 CONDOS - 2-10 UNI... <NA>     1967 1403
6     3 <NA>          15 CONDOS - 2-10 UNI... <NA>     1967 1404
# [i] abbreviated name: 1`BUILDING CLASS CATEGORY`
# [i] 15 more variables: `EASE-MENT` <lgl>, `BUILDING CLASS AT PRESENT` <chr>,
# `ADDRESS` <chr>, `APART\nMENT\nNUMBER` <chr>, `ZIP CODE` <dbl>,
# `RESIDENTIAL UNITS` <dbl>, `COMMERCIAL UNITS` <dbl>, `TOTAL UNITS` <dbl>,
# `LAND SQUARE FEET` <dbl>, `GROSS SQUARE FEET` <dbl>, `YEAR BUILT` <dbl>,
# `TAX CLASS AT TIME OF SALE` <dbl>, `BUILDING CLASS AT TIME OF SALE` <chr>,
# `SALE\nPRICE` <dbl>, `SALE DATE` <dtm>
```

Display the statistical Summary of the data frame bk

summary (bk)

```
> summary(bk)
  BOROUGH      NEIGHBORHOOD      BUILDING CLASS CATEGORY      TAX CLASS AT PRESENT
Min.   :3      Length:23373      Length:23373      Length:23373
1st Qu.:3      Class :character      Class :character      Class :character
Median :3      Mode  :character      Mode  :character      Mode  :character
Mean   :3
3rd Qu.:3
Max.   :3

  BLOCK      LOT      EASE-MENT      BUILDING CLASS AT PRESENT
Min.   : 20      Min.   :  1.0      Mode:logical      Length:23373
1st Qu.:1638      1st Qu.: 22.0      NA's:23373      Class :character
Median :3839      Median : 48.0      Mode  :character
Mean   :3984      Mean   : 305.4
3rd Qu.:6259      3rd Qu.: 142.0
Max.   :8955      Max.   :9039.0

  ADDRESS      APART\MENT\NUMBER      ZIP CODE      RESIDENTIAL UNITS
Length:23373      Length:23373      Min.   :  0      Min.   : 0.000
Class :character      Class :character      1st Qu.:11209      1st Qu.: 1.000
Mode  :character      Mode  :character      Median :11218      Median : 1.000
Mean   :11211      Mean   : 2.156
3rd Qu.:11230      3rd Qu.: 2.000
Max.   :11416      Max.   :509.000
```

Print Missing Values

```
na_count <- sum(is.na(bk$SALEPRICE))
print(na_count)
Output: 0
```

List all column names: names(bk)

```
> bk <- read_excel("D://DVSDATA//rollingsales_brooklyn.xls", sheet = 1)
> names(bk)
 [1] "BOROUGH"           "NEIGHBORHOOD"
 [3] "BUILDING CLASS CATEGORY" "TAX CLASS AT PRESENT"
 [5] "BLOCK"             "LOT"
 [7] "EASE-MENT"         "BUILDINGCLASSAT PRESENT"
 [9] "ADDRESS"           "APARTMENTNUMBER"
[11] "ZIPCODE"           "RESIDENTIALUNITS"
[13] "COMMERCIALUNITS"   "TOTALUNITS"
[15] "LANDSQUAREFEET"    "GROSSSQAREFEET"
[17] "YEARBUILT"         "TAXCLASSATTIMEOFSALE"
[19] "BUILDINGCLASSATTIMEOFSALE" "SALEPRICE"
[21] "SALEDATE"
```

Clean/format the data with regular expressions

```
bk$gross.sqft <- as.numeric(gsub("[^[:digit:]]", "", bk$GROSSSQAREFEET))
```

- Access the **GROSSSQAREFEET** column:
- Use **gsub** to remove non-digit characters
- **as.numeric(...)** converts the resulting cleaned string, which now contains only digits, to a numeric type.
- Assign the numeric value to a new column **gross.sqft**

Example: Suppose **bk\$GROSSSQAREFEET** has the following values:

"1,000"

"2,500"

"3,750sqft"

The `gsub("[^[:digit:]]", "", bk$GROSSSQAREFEET)` operation would transform these values to:

"1000"

"2500"

"3750"

```
> print(bk$gross.sqft)
 [1]      0      0      0      0      0      0      0      0      0
[10]      0      0      0      0      0      0      0      0      0
[19]      0      0      0      0      0      0      0      0 1492 1724
[28]  2132  1704  2640  3304  2000  1800  1281  2346 2835
[37]  1562  1328  2500  2070  4071  5107  1392  1392 1392
[46]  1392  1860  1860  1392  1479   992  1320  2700 2700
[55]  2700  3160  2224  2242  2090  1622  1935  2340 1401
[64]  2048  2745  3240  4140  2512  1200  1584  2521 2062
[73]  2890  3340  3340  3340  2544  1984  2904  2845 2845
[82]  2845  1771  2115  3240  3240  2772  2018  2160 2160
[91]  3195  3195  2448  2953  2953  4400  2365  2365 1513
[100] 2910  3254  2000  1802  1812  2560  2194  2194 2194
[109] 4105  1635  1785  2275  2620  1940  1953  1762 2490
[118] 1934  3420  2352  1995  2760  2802  2348  1674 2060
[127] 2520  1756  1998  1729  1928  1928  1782  1782 1860
[136] 2340  2160  1980  1980  2160  2160  1980  1980 1776
[145] 1645  1928  1610  1840  1720  1652  1652  1652 1584
[154] 1869  1869  1869  1869  1673  1320  1993  1584 1584
[163] 1734  1734  5300  3060  2328  3240  3826  2648 4480
```


Similarly

- `bk$land.sqft <- as.numeric(gsub("[^[:digit:]]", "", bk$LANDSQUAREFEET))`
- `bk$sale.date <- as.Date(bk$SALEDATE)`
- `bk$year.built <- as.numeric(as.character(bk$YEARBUILT))`

Will yield the following results

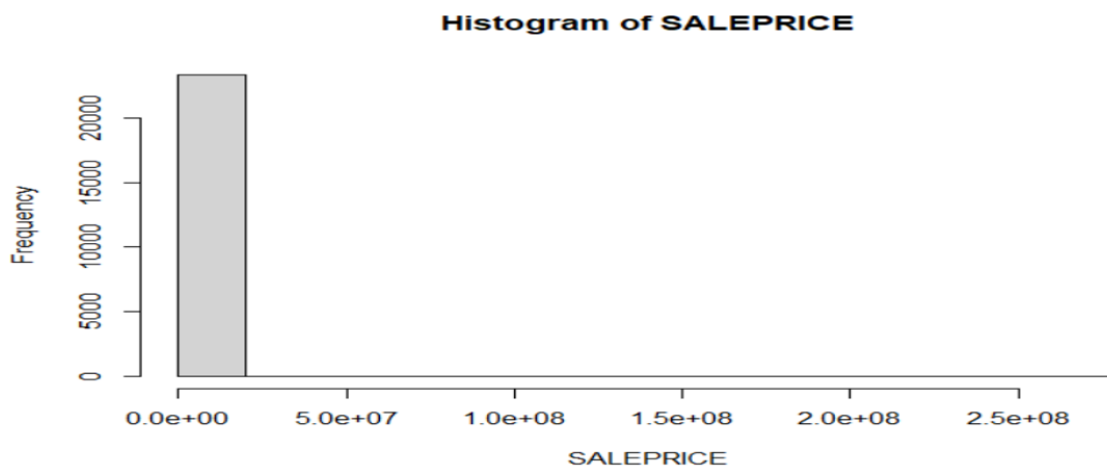
```
print(bk$land.sqft)
 [1]      0      0      0      0      0      0      0      0      0      0
[10]      0      0      0      0      0      0      0      0      0      0
[19]      0      0      0      0      0      0      0      0     2058     4833
[28]    2417    3867    1600    2707    2417    1172    1440    2513    5810
[37]    1933    1990    1713    2223    5800    7747    2511    1648    1648
[46]    1649    2062    2062    1861    1878    1372    1600    2320    2320
[55]    2320    1948    1547    2383    2500    2103    3867    2167    2140
[64]    1800    3178    1933    2610    2417     985    2417    2223    1456
[73]    5000    3853    3853    3853    2900    4833    9667    4833    4833
[82]    4833    2127    1933    2344    2344    1933    1925    1993    1933
[91]    2083    2083    2167    2167    2167    2025    1558    1558    1933
[100]    1820    5518    2900    2449    2000    1933    3867    3867    3867
[109]    3867    2147    1933    1904    2020    2331    2058    1916    1906
```

```
print(head(bk$sale.date, n= 100))
 [1] "2013-07-09" "2013-07-12" "2013-04-25" "2013-04-25" "2013-04-25"
 [6] "2013-04-25" "2013-07-25" "2013-07-25" "2012-11-19" "2013-04-22"
[11] "2012-11-12" "2013-02-08" "2012-11-13" "2012-11-13" "2012-12-07"
[16] "2013-02-15" "2013-01-09" "2013-01-07" "2012-11-07" "2013-06-25"
[21] "2013-07-03" "2013-06-19" "2013-06-03" "2013-01-16" "2013-03-18"
[26] "2013-06-06" "2012-12-18" "2012-08-24" "2013-06-18" "2012-12-14"
[31] "2012-11-29" "2012-11-14" "2013-06-03" "2013-03-11" "2013-06-06"
[36] "2012-09-27" "2013-05-07" "2012-09-25" "2013-05-31" "2012-10-22"
[41] "2012-11-29" "2013-07-25" "2013-01-03" "2013-02-21" "2013-01-11"
[46] "2012-12-12" "2013-03-22" "2013-03-01" "2013-07-24" "2012-11-16"
[51] "2013-03-18" "2013-01-24" "2012-11-07" "2013-03-22" "2012-08-30"
[56] "2013-07-17" "2012-10-20" "2013-02-20" "2013-05-10" "2012-11-13"
[61] "2012-10-12" "2013-06-19" "2013-06-17" "2013-03-21" "2013-01-02"
[66] "2013-07-03" "2013-06-24" "2013-04-03" "2013-06-28" "2013-04-02"
[71] "2013-05-01" "2013-04-23" "2013-04-05" "2013-06-18" "2013-05-17"
[76] "2013-01-15" "2013-03-01" "2012-08-13" "2012-12-28" "2012-09-19"
[81] "2012-09-19" "2012-09-19" "2012-11-29" "2012-08-24" "2012-11-20"
[86] "2012-09-27" "2013-06-25" "2012-11-16" "2013-03-26" "2013-01-14"
[91] "2012-12-20" "2012-12-19" "2012-09-28" "2012-08-15" "2012-08-15"
[96] "2013-02-13" "2013-01-25" "2013-01-25" "2012-08-17" "2013-01-15"
```

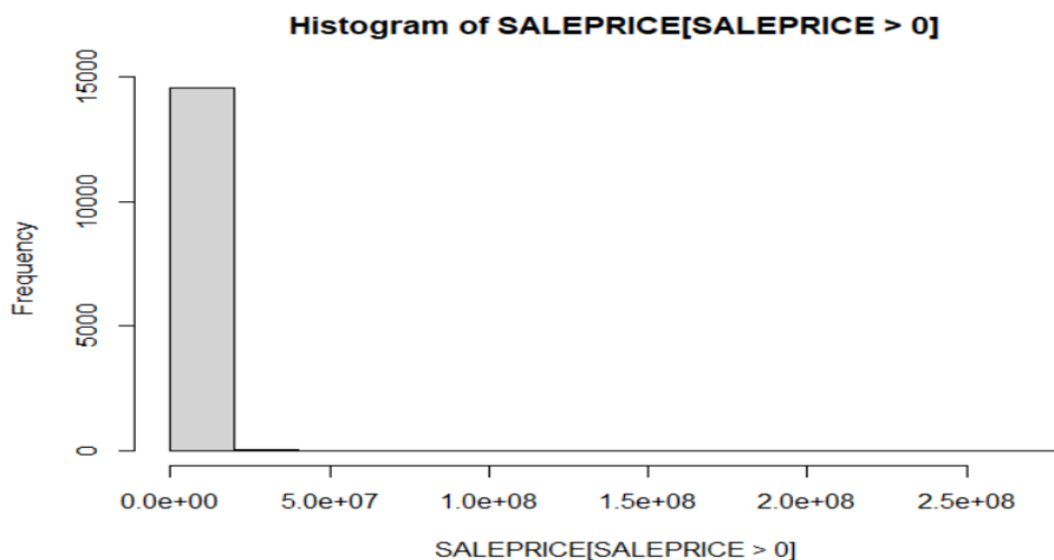
```
> print(head(bk$year.built,n=100))
 [1] 0 0 0 0 0 0 0 0 0 2011 2011 2011 2011 2011 2011 2011
[16] 2011 2011 2011 2011 2011 2011 0 0 0 0 0 1930 1930 1930 1899 1925
[31] 1960 1935 1920 1915 1935 1950 1920 1925 1930 1955 1925 1945 1950 1945 1945
[46] 1940 1940 1940 1940 1945 1950 1940 1930 1930 1930 1930 1930 1930 1930 1930
[61] 1950 1920 1960 1910 1950 1901 1950 1905 1930 1930 1930 1940 1935 1940 1940
[76] 1940 1930 1899 1899 1899 1899 1899 1950 1920 1940 1940 1935 1955 1960 1960
[91] 1930 1930 1925 1960 1960 1925 1965 1965 1910 1992
```

Exploratory Data Analysis

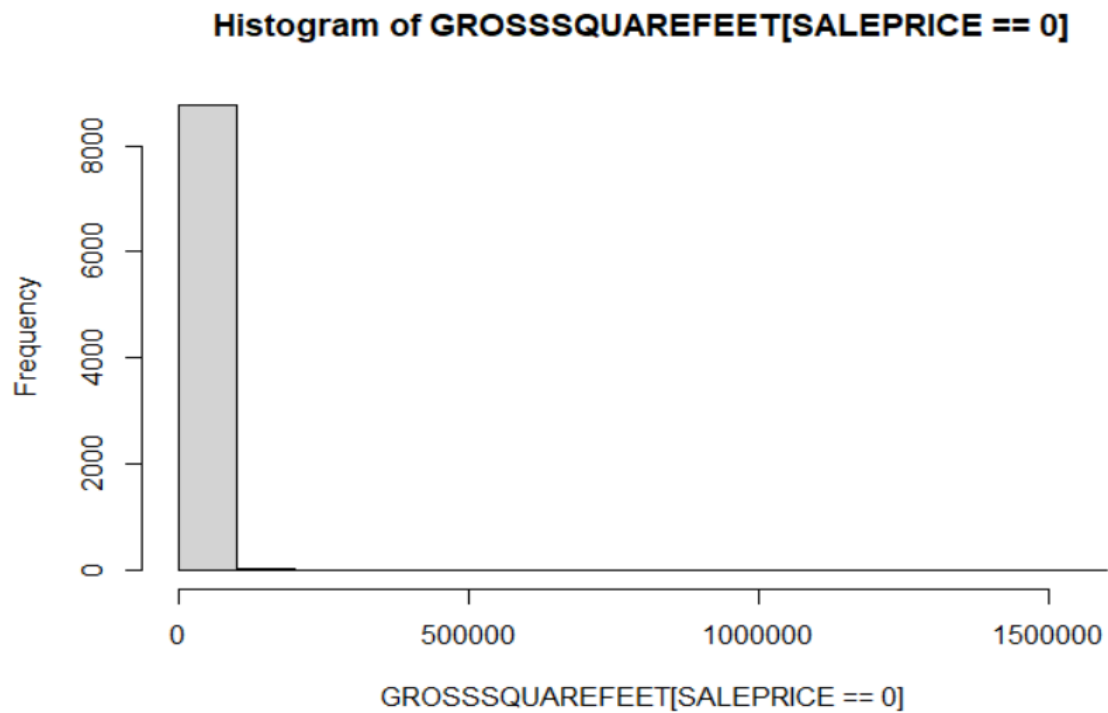
```
attach(bk)
hist(SALEPRICE
```



```
hist(SALEPRICE[SALEPRICE>0])
```



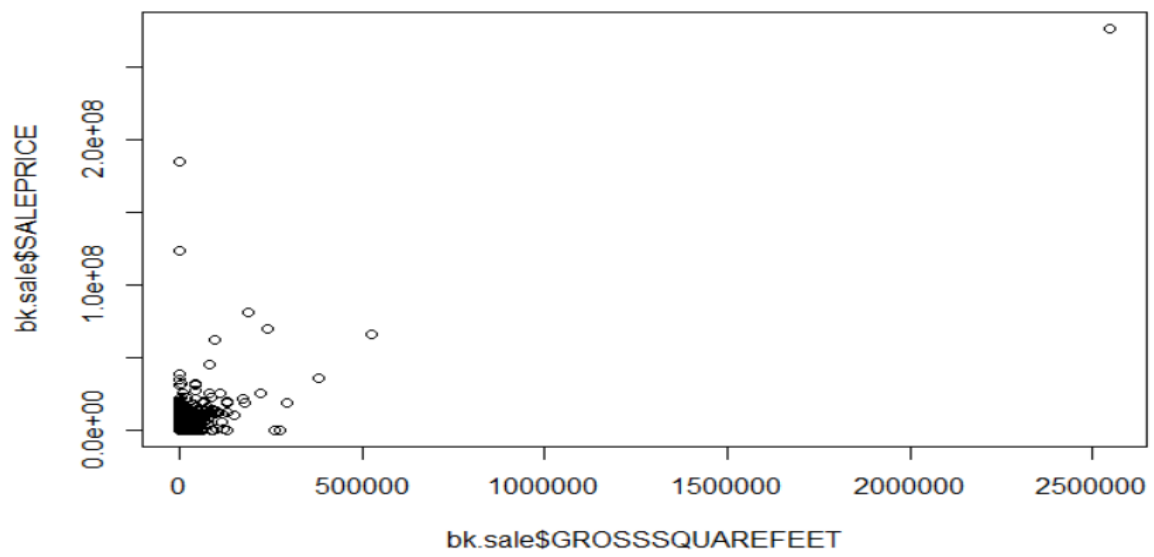

```
hist(GROSSSQUAREFEET[SALEPRICE==0])
```



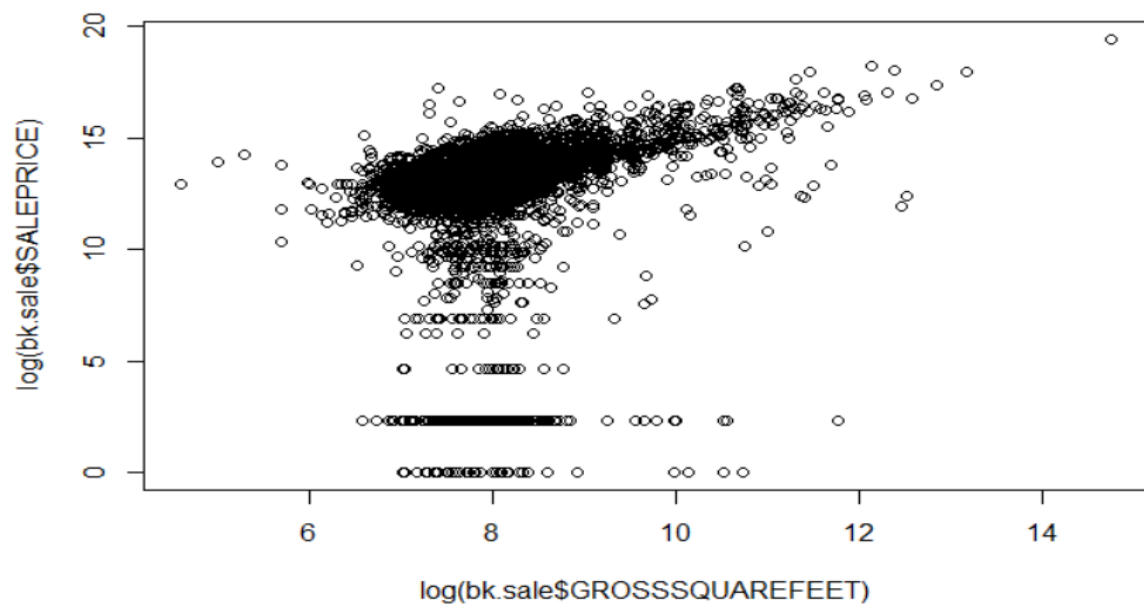
Complete Code

- `hist(SALEPRICE)`
- `hist(SALEPRICE[SALEPRICE>0])`
- `hist(GROSSSQUAREFEET[SALEPRICE==0])`
- `detach(bk)`

```
## keep only the actual sales  
bk.sale <- bk[bk$SALEPRICE!=0,]  
plot(bk.sale$GROSSSQUAREFEET,bk.sale$SALEPRICE)
```



```
plot(log(bk.sale$GROSSSQAREFEET),log(bk.sale$SALEPRICE))
```



Algorithms, ML Algorithms and Models

Algorithms

- An algorithm is a **procedure or set of steps or rules** to accomplish a task.
- Algorithms are one of the **fundamental concepts in, or building blocks of**, computer science: the basis of the **design of elegant and efficient code, data preparation and processing, and software engineering.**
- Some of the basic types of tasks that algorithms can solve are ***sorting, searching, and graph-based computational problems.***
- **Efficiency is measured in terms of memory and computational time**, which matters especially when you're dealing with massive amounts of data and building consumer facing products.

Characteristics of an Algorithm

1. **Well-Defined Instructions** : Clear and unambiguous steps.
2. **Input**: Specific data or parameters provided to the algorithm.
3. **Output** : One or more results produced by the algorithm.
4. **Finiteness** : Terminates after a finite number of steps.
5. **Effectiveness** : Each step can be performed in a finite amount of time using basic computational operations.

Additional Characteristics (Optional)

- **Deterministic**: Most algorithms are deterministic, meaning that given the same input, they will always produce the same output. However, some algorithms can be non-deterministic or probabilistic, incorporating randomness in their steps.
- **General-purpose**: Algorithms can often be designed to solve a broad class of problems, making them versatile and reusable for different applications with similar characteristics.

Types/Classes of Data Science Algorithms

1. **Data Engineering Algorithms (Data munging, preparation, and processing algorithms, such as sorting, MapReduce, or Pregel.)**
2. **Optimization algorithms** for parameter estimation, including Stochastic Gradient Descent, Newton's Method, and Least Squares.
3. **Machine learning algorithms.**

Machine Learning Algorithms

In the context of machine learning, an algorithm is a method or a process used to *build models from data*. Algorithms provide the steps needed to transform *input data into output data*. A ML model is the output or the representation generated after an algorithm processes data. In machine learning, a model is the mathematical representation of the learned patterns or relationships within the data. The model is used to make predictions or decisions without being explicitly programmed to perform the task.

Machine learning algorithms are primarily used for prediction, classification, and clustering. This overlaps with statistical modeling, which also aims to predict or classify, leading to some ambiguity between the two fields. The main distinction arises from their origins: statistical modeling comes from statistics departments, while machine learning algorithms are rooted in computer science. This distinction is evident in the techniques used and their applications.

Statistical models, such as linear regression, often focus on interpreting parameters to understand the underlying generative process of the data. In contrast, machine learning models, which power applications like image and speech recognition or recommendation systems, prioritize accuracy in predictions or classifications over understanding the data generation process.

There are several key differences between the approaches of statisticians and machine learners:

Interpreting Parameters: Statisticians seek meaningful interpretations of model parameters to describe real-world phenomena. In contrast, machine learners, especially software engineers, view models as "black boxes" optimized for predictive performance, often without interpreting the parameters.

Confidence Intervals: Statisticians provide confidence intervals to capture the variability or uncertainty of parameters. Many machine learning algorithms, like k-means or k-nearest neighbors, do not incorporate this concept.

Explicit Assumptions: Statistical models often make explicit assumptions about data distributions, while many machine learning methods, especially nonparametric ones, do not make such assumptions or do so implicitly.

Culturally, statisticians focus on understanding and quantifying uncertainty, whereas software engineers prioritize building effective predictive models,

often without concern for uncertainty. Companies like Google and Facebook emphasize rapid iteration and fixing issues as they arise.

Data scientists embody a hybrid approach, leveraging both statistical and machine learning methodologies. They combine strengths from both fields, making them adept at balancing the need for robust, interpretable models and high-performing predictive systems. This dual expertise is humorously captured in Josh Wills' quote: "Data scientist (noun): Person who is better at statistics than any software engineer and better at software engineering than any statistician."

Key Differences and Cultural Approaches

Parameters	Statisticians	Machine Learning /Software Engineers
Interpreting Parameters	Real-world interpretations	Focus on predictive power
Confidence Intervals	Use confidence intervals and posterior distributions	Often lack this notion
Explicit Assumptions	Make explicit assumptions about data distributions	Often nonparametric with implicit assumptions
Cultural Approaches	Investigate uncertainty, never 100% confident	Focus on building predictive models, emphasize rapid iteration and fixing issues

Three Basic Algorithms

Many business or real-world problems can be mathematically framed as classification and prediction tasks. A variety of models and algorithms exist for these purposes, but the real challenge for a data scientist is selecting the appropriate method based on the problem context and assumptions. This expertise develops with experience, enabling one to recognize the nature of a problem and choose suitable algorithms like **logistic regression, Naive Bayes, or k-means**.

Initially, understanding which algorithm to apply can be difficult when learning methods in isolation. It's important not to approach every problem with a single familiar technique, like **linear regression**, but instead to consider the mathematical attributes of the problem and how different algorithms might address these.

Collaboration and discussion with others can be valuable for determining the best approach. Maintaining a methodical attitude and recognizing that the solution is not always obvious is crucial. Textbooks often simplify this by pairing techniques with problems, but in practice, mastering the technique is only part of the challenge; knowing when to use it is equally important.

This chapter introduces three basic algorithms

1. **Linear regression,**
2. **K-nearest neighbours (k-NN), and**
3. **K-means—as foundational tools.**

What is Linear Regression?

Linear regression is a statistical method used to model the relationship between a dependent variable (also known as the outcome or response variable) and one or more independent variables (also known as predictors or features). The basic idea is to fit a line through a scatter plot of data points in such a way that the sum of the squares of the vertical distances (errors) from the points to the line is minimized. This method is known as least squares estimation.

Why Use Linear Regression?

Linear regression is used for two primary purposes:

Prediction: To predict future outcomes based on the relationship between variables.

Understanding Relationships: To understand and describe the relationship between variables.

For example, you might use **linear regression** to predict a company's sales based on its advertising budget, or to understand the relationship between the number of friends a person has on a social networking site and the amount of time they spend on that site daily.

Example1: Modelling the Given Data Set

Suppose you run a social networking site that charges a monthly subscription fee of \$25, and that this is your only source of revenue. Each month you collect data and count your number of users and total revenue. You've done this daily over the course of two years, recording it all in a spreadsheet. You could express this data as a series of points.

Here are the first four:

$$S=\{(x,y)=(1,25),(10,250),(100,2500),(200,5000)\}$$

By observing the data one can derive the Equation/Model mentally to represent the relationship of x and y: $y = 25X$. Figure below illustrate the plot of $y = 25x$ for given set of data and it can be extended to predict the future values.

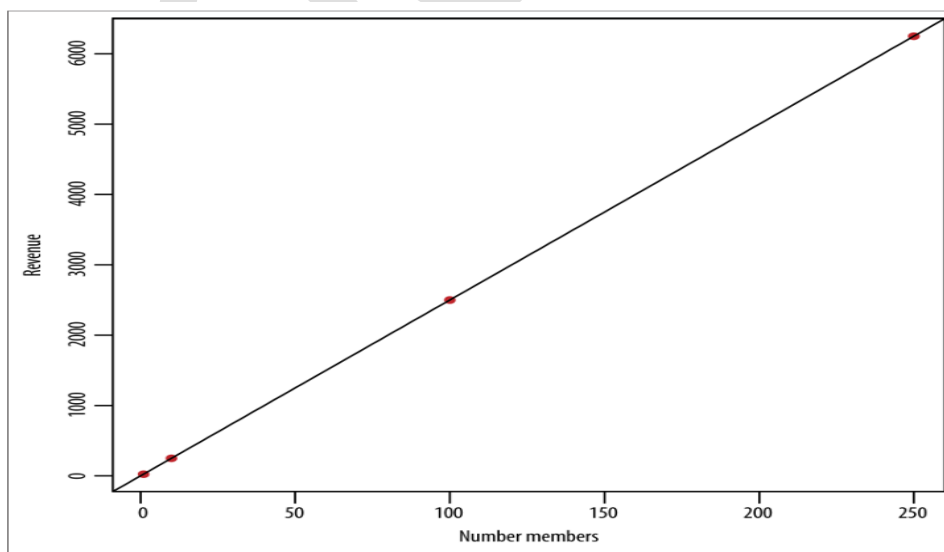


Figure 3-1. An obvious linear pattern

Example 2: Consider the following Sample Data and plot the scatter plot .

Sample Data and Plot:

new_friends	time_spent (seconds)
7	276
3	43
4	82
6	136
10	417
9	269

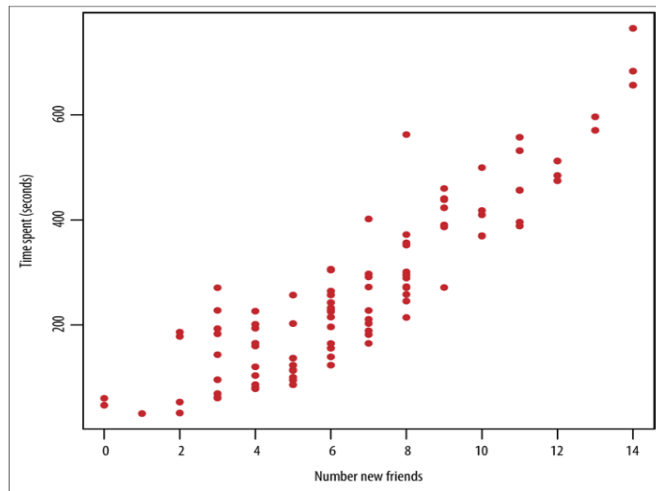


Figure 3-2. Looking kind of linear

By observing the plot one can come out with multiple lines as illustrated below

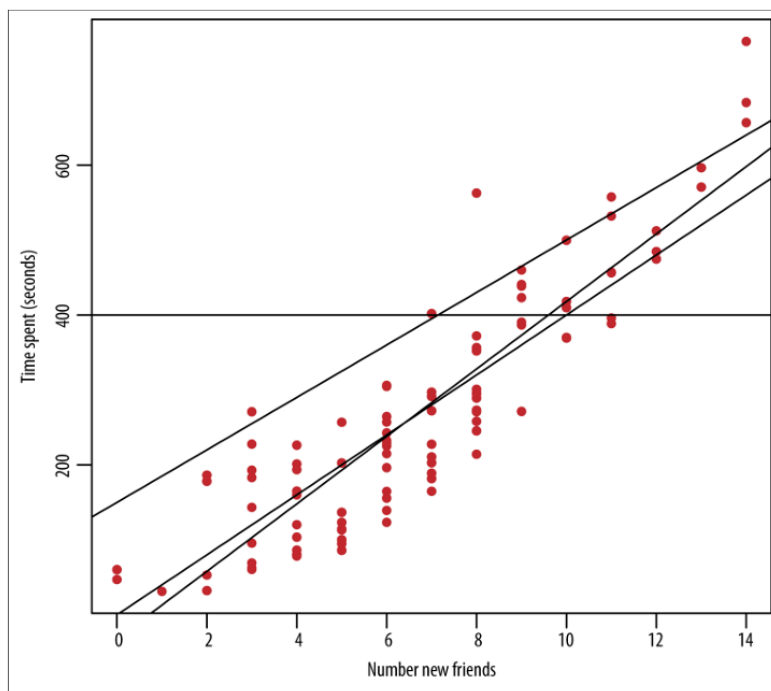


Figure 3-3. Which line is the best fit?

In order to identify which Line is the best fit, assume a Linear Relationship, start with the assumption: $y = \beta_0 + \beta_1 x$, where β_0 and β_1 represent the intercept and slope, respectively. Estimate the parameters (β_0 and β_1) using observed data pairs: $(x_1, y_1), (x_2, y_2), \dots, (x_n, y_n)$.

Fitting the model: The next stage involves fitting the model to the data to determine the optimal parameter values. By applying least squares estimation, Linear regression seeks to find the line that minimizes the **sum of the squares** of the vertical distances between the **approximated or predicted** \hat{y}_i s and the **observed** y_i s.

To find this line, you define the "**residual sum of squares**" (RSS), denoted as: $RSS(\beta) = \sum_i (y_i - \beta x_i)^2$, where i ranges over the various data points. It is the sum of all the squared vertical distances between the observed points and any given line. Figure illustrates the best fitted line.

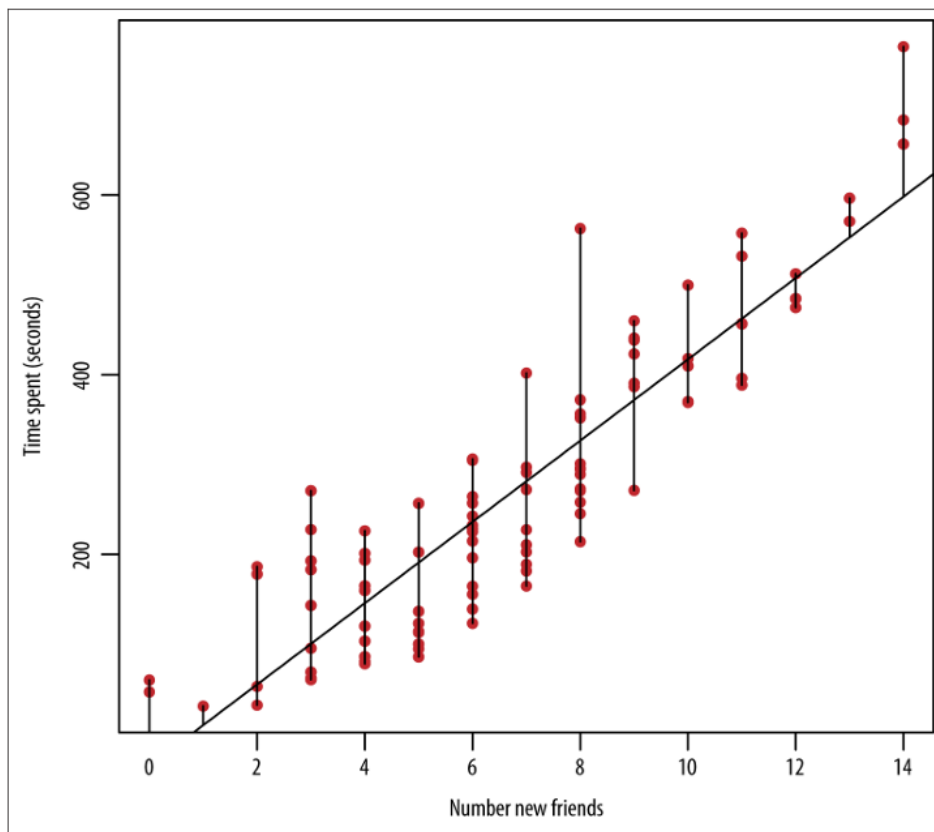


Figure 3-4. The line closest to all the points

Key Concepts in Linear Regression

Dependent Variable (y): The outcome or the variable you're trying to predict.

Independent Variable (x): The variable(s) you use to predict the dependent variable.

Linear Equation: The relationship is modeled by the equation

$$y = \beta_0 + \beta_1x + \epsilon$$

where:

- β_0 is the intercept.
- β_1 is the slope (coefficient) of the independent variable.
- ϵ is the error term (residual).

Steps in the Linear Regression Algorithm

1. **Collect Data:** Gather data for the dependent and independent variables.
2. **Visualize Data:** Plot the data to see if there is an apparent linear relationship.
3. **Fit the Model:** Use the least squares method to find the best-fitting line.
4. **Evaluate the Model:** Check how well the line fits the data using metrics like Mean Squared Error (MSE).
5. **Make Predictions:** Use the fitted model to make predictions on

Evaluation Metrics in Linear Regression

1. **R-squared (R^2)** : R^2 measures the proportion of the variance in the dependent variable that is predictable from the independent variable(s).

Formula:

$$R^2 = 1 - \frac{\sum_i (y_i - \hat{y}_i)^2}{\sum_i (y_i - \bar{y})^2}$$

- $\sum_i (y_i - \hat{y}_i)^2$: Sum of squared residuals (errors between observed and predicted values).
- $\sum_i (y_i - \bar{y})^2$: Total sum of squares (variance of the observed values from their mean).

R^2 ranges from **0 to 1**. A higher R^2 value indicates a better fit of the model to the data. In the example output, $R^2=0.8244$, meaning about **82.44%** of the variance in the dependent variable (y) is explained by the independent variable (x).

2. **p-values:** The p-value tests the null hypothesis that a coefficient (β) is equal to zero (no effect). A low p-value (**< 0.05**) indicates that you can reject the null hypothesis, meaning the coefficient is significantly different from zero. In the output, the p-value for the **slope (x)** is very small (**< 2e-16**), indicating that x is a significant predictor of y.

The **p-value** for the intercept is **0.0565**, which is slightly above **0.05**, suggesting it is not significantly different from zero at the **5%** significance level.

Example of Code in R

```
# Example dataset
x <- c(7, 3, 4, 6, 10, 9)
y <- c(276, 43, 82, 136, 417, 269)
# Fit the linear model
model <- lm(y ~ x)
# Print the model summary
summary(model)
```

Output:

Call:

```
lm(formula = y ~ x)
```

Residuals:

```
  1      2      3      4      5      6
47.547 11.507 1.267 -43.213 40.827 -57.933
```

Coefficients:

	Estimate	Std. Error	t value	Pr(> t)
(Intercept)	-116.227	54.792	-2.121	0.10120
x	49.240	7.868	6.258	0.00332

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

Residual standard error: 48.18 on 4 degrees of freedom

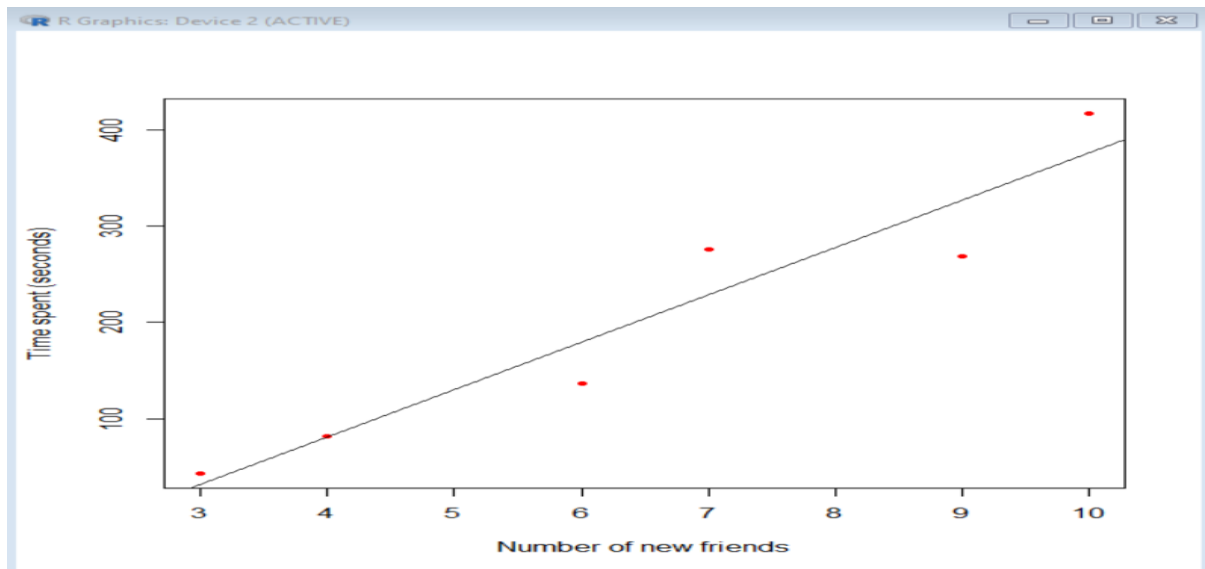
Multiple R-squared: 0.9073, Adjusted R-squared: 0.8842

F-statistic: 39.17 on 1 and 4 DF, p-value: 0.003325

Plot the data and the fitted line

```
plot(x, y, pch=20, col="red", xlab="Number of new friends", ylab="Time spent (seconds)")  
abline(model)
```

Output:



In this example, the fitted line might look like $y = -32.08 + 45.92x$ indicating that for each new friend, the time spent on the site increases by approximately 46 seconds.

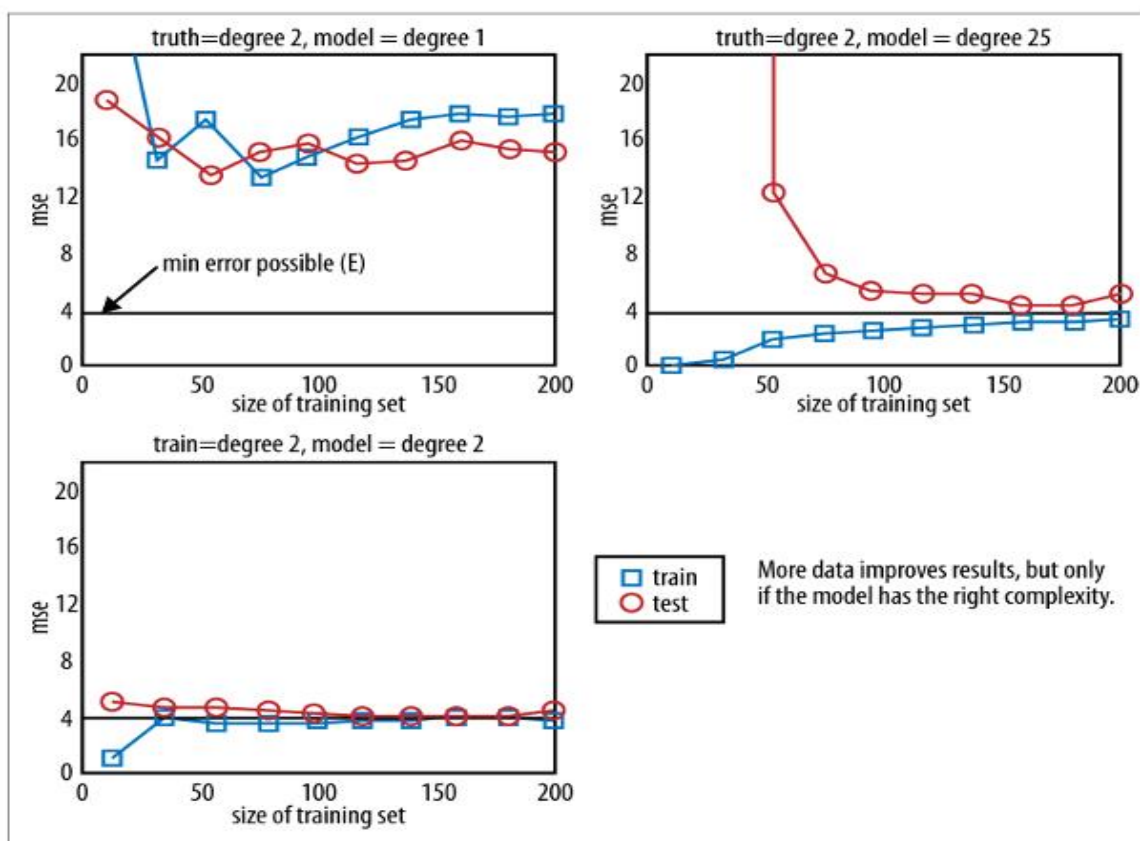
Example: Comparing mean squared error in training and testing set

Figure 3-6. Comparing mean squared error in training and test set, taken from a slide of Professor Nando de Freitas; here, the ground truth is known because it came from a dataset with data simulated from a known distribution

The image shows three graphs that illustrate the relationship between the size of the training set and the mean squared error for both the training and test sets, under different model complexities (degrees 1, 2, and 25).

The top left graph represents a model with degree 2 (quadratic) for both the truth and the model. As the size of the training set increases, the mean squared error decreases for both the training and test sets, eventually converging to the minimum possible error.

The top right graph represents a model with degree 25, where the model's complexity is higher than the true underlying function (degree 2). In this case, as the training set size increases, the training error continues to decrease, but the test error does not improve significantly after a certain point, indicating overfitting.

The bottom graph represents a model with degree 2 for both the truth and the model. In this case, where the model complexity matches the true underlying function, increasing the training set size leads to a consistent decrease in the mean squared error for both the training and test sets.

The key message conveyed by these graphs is that more data improves results, but only if the model has the right complexity. If the model is too simple (degree 1 in this case), it cannot capture the underlying pattern adequately, leading to high errors. If the model is too complex (degree 25), it can overfit the training data, resulting in poor generalization to the test set. The optimal model complexity (degree 2) strikes a balance, allowing the model to learn the underlying pattern effectively while avoiding overfitting.

In the context of the graphs shown, the terms "truth = degree 2", "training = degree 2", and "model = degree 2" refer to the complexity or degree of the underlying true function, the training model, and the model used for prediction, respectively.

Specifically:

1. "truth = degree 2" means that the underlying true function or pattern in the data is a quadratic function (a polynomial of degree 2).
2. "training = degree 2" means that during the training process, a quadratic model (polynomial of degree 2) is used to fit the training data.
3. "model = degree 2" means that the final model used for making predictions is a quadratic model (polynomial of degree 2).

In the bottom graph, where "train = degree 2, model = degree 2", it indicates that both the training model and the final prediction model are quadratic functions, matching the complexity of the underlying true function ("truth = degree 2"). This alignment of complexities allows the model to effectively learn the underlying pattern from the training data and generalize well to the test data, as evidenced by the decreasing mean squared error for both training and test sets as the size of the training set increases.

Loss Function: The mean squared error is the differences between the predicted values and the actual values, giving you an overall sense of how close your predictions are to the real data.

A loss function measures how well a model's predictions match the actual data. In the context of regression, where we're trying to predict a continuous outcome, one commonly used loss function is the Mean Squared Error (MSE).

Here's the equation for the Mean Squared Error (MSE):

$$\text{MSE} = \frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)^2$$

- n is the number of data points.
- y_i is the actual value of the outcome for the i th data point.
- \hat{y}_i is the predicted value of the outcome for the i th data point.

Let's break it down:

- $(y_i - \hat{y}_i)$ represents the difference between the actual outcome and the predicted outcome for each data point.
- $(y_i - \hat{y}_i)^2$ squares this difference, ensuring that negative and positive differences don't cancel each other out.
- $\frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)^2$ takes the average of these squared differences over all data points, giving us the mean squared error.

The MSE essentially gives us an average of the squared differences between the actual and predicted values. A lower MSE indicates that the model's predictions are closer to the actual data, while a higher MSE indicates greater discrepancies.

Adding Other Predictors

$$y = \beta_0 + \beta_1x_1 + \beta_2x_2 + \beta_3x_3 + \varepsilon$$

This equation represents multiple linear regression, where x_1 , x_2 , and x_3 are additional predictors (e.g., age and gender of users) added to the model. Each predictor has its own coefficient (e.g., $\beta_1, \beta_2, \beta_3$) representing the effect of that predictor on the dependent variable y . The model allows for more complexity by considering multiple predictors simultaneously.

Multiple Linear Regression: $y = \beta_0 + \beta_1x_1 + \beta_2x_2 + \dots + \beta_nx_n + \varepsilon$

This equation extends the concept of linear regression to include multiple predictors (up to n). Each predictor (x_1, x_2, \dots, x_n) has its own coefficient ($\beta_1, \beta_2, \dots, \beta_n$) representing its contribution to the dependent variable y . The model is useful for situations where the outcome is influenced by multiple factors.

Linear Regression Example: Predicting House Prices

Let's go through a simple example to predict house prices based on the size of the house.

Collect Data:

Suppose we have data on house prices and their sizes.

Size (sq ft)	Price (\$)
1400	245,000
1600	312,000
1700	279,000
1875	308,000
1100	199,000
1550	219,000
2350	405,000
2450	324,000
1425	319,000
1700	255,000

Visualize Data: Plot the data points on a scatter plot to see if there's a linear relationship.

Fit the Model: Use the least squares method to find the best-fitting line. The goal is to minimize the residual sum of squares (RSS).

$$RSS(\beta) = \sum (y_i - \beta_0 - \beta_1 x_i)^2$$

The solution to minimizing RSS gives us the estimates for β_0 (intercept) and β_1 (slope).

Example Calculation: Using statistical software (like R, Python, or even a calculator), we fit the linear regression model.

Code Snippet1:

```
# Using R to fit the model
```

```
sizes <- c(1400, 1600, 1700, 1875, 1100, 1550, 2350, 2450, 1425, 1700)
```

```
prices <- c(245000, 312000, 279000, 308000, 199000, 219000, 405000, 324000, 319000, 255000)
```

```
model <- lm(prices ~ sizes) # Fit the Model
```


Code Snippet2:

```
summary(model)
```

```
# Output:
```

```
Call:
```

```
lm(formula = prices ~ sizes, data = data)
```

```
Residuals:
```

Min	1Q	Median	3Q	Max
-49388	-27388	-6388	29577	64333

```
Coefficients:
```

	Estimate	Std. Error	t value	Pr(> t)
(Intercept)	98248.33	58033.48	1.693	0.1289
sizes	109.77	32.97	3.329	0.0104 *

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

Residual standard error: 41330 on 8 degrees of freedom

Multiple R-squared: 0.5808, Adjusted R-squared: 0.5284

F-statistic: 11.08 on 1 and 8 DF, p-value: 0.01039

Explanation of the output:**Residuals**

The residuals represent the differences between the observed values and the predicted values from the regression model. The summary statistics provided are:

- **Min:** -49388
- **1Q (First Quartile):** -27388
- **Median:** -6388
- **3Q (Third Quartile):** 29577
- **Max:** 64333

These values give us a sense of the distribution of the residuals:

- **Min:** The smallest residual, indicating the largest negative deviation from the predicted values.
- **1Q:** The 25th percentile of the residuals, meaning 25% of the residuals are less than or equal to this value.
- **Median:** The middle value of the residuals, indicating that half of the residuals are above this value and half are below.

- **3Q:** The 75th percentile of the residuals, meaning 75% of the residuals are less than or equal to this value.
- **Max:** The largest residual, indicating the largest positive deviation from the predicted values.

Coefficients

The table of coefficients provides information about the estimated parameters of the regression model. The key columns are:

- **Estimate:** The estimated value of the coefficient.
- **Std. Error:** The standard error of the coefficient estimate.
- **t value:** The t-statistic, which is the ratio of the Estimate to the Std. Error.
- **Pr(>|t|):** The p-value associated with the t-statistic, used to test the null hypothesis that the coefficient is zero.

For each coefficient:

1. (Intercept):

- **Estimate:** 98248.33
- **Std. Error:** 58033.48
- **t value:** 1.693
- **Pr(>|t|):** 0.1289

This suggests that the intercept is not statistically significant at the 0.05 level since the p-value is greater than 0.05.

2. sizes:

- **Estimate:** 109.77
- **Std. Error:** 32.97
- **t value:** 3.329
- **Pr(>|t|):** 0.0104

This indicates that the coefficient for the predictor variable "sizes" is statistically significant at the 0.05 level since the p-value is less than 0.05.

Significance Codes

The significance codes provide a shorthand to interpret the p-values:

- ***: p-value < 0.001
- **: p-value < 0.01
- *: p-value < 0.05
- .: p-value < 0.1
- : p-value \geq 0.1

In this case, the coefficient for "sizes" has a *, indicating it is significant at the 0.05 level.

Residual Standard Error

- **Residual standard error:** 41330 on 8 degrees of freedom

This is the standard deviation of the residuals. It measures the average amount that the observed values deviate from the predicted values.

R-Squared and Adjusted R-Squared

- **Multiple R-squared:** 0.5808
- **Adjusted R-squared:** 0.5284

These statistics indicate the proportion of the variance in the dependent variable that is explained by the independent variables:

Multiple R-squared: The proportion of variance explained by the model.

Adjusted R-squared: Adjusted for the number of predictors in the model, providing a more accurate measure when multiple predictors are used.

F-Statistic

F-statistic: 11.08 on 1 and 8 DF

p-value: 0.01039

The F-statistic tests the overall significance of the model. The p-value associated with the F-statistic indicates whether the model as a whole is statistically significant. In this case, the model is significant at the 0.05 level since the p-value is less than 0.05.

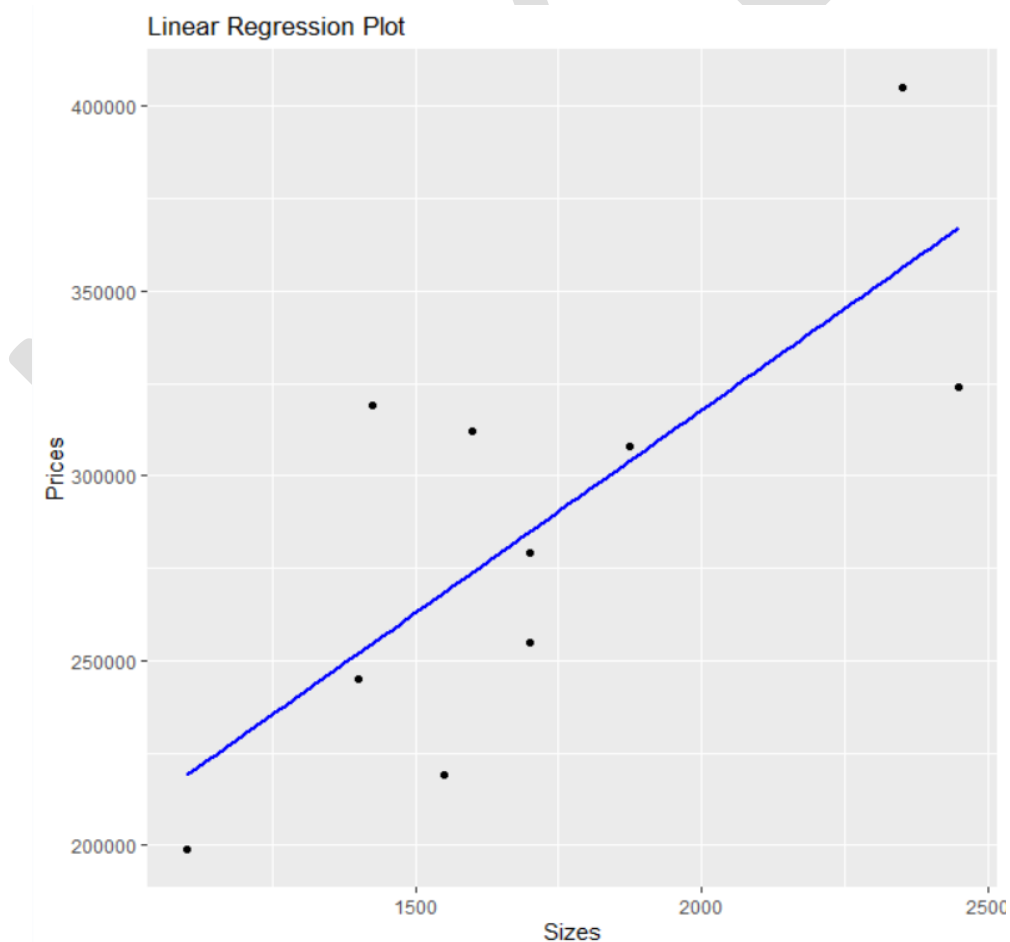
Summary

The coefficient for "sizes" is statistically significant, suggesting it has a meaningful effect on the dependent variable. The intercept is not statistically significant. The residuals have a substantial spread, as indicated by the residual standard error. The model explains about 58% of the variance in the dependent variable (Multiple R-squared), with an adjusted value accounting for the number of predictors being around 52.84%. The overall model is statistically significant, as indicated by the F-statistic's p-value.

Code Snippet3

Plot

```
ggplot(data, aes(x = sizes, y = prices)) + geom_point() + # Add points
geom_smooth(method = "lm", se = FALSE, color = "blue") + # Add regression
line
labs(title = "Linear Regression Plot", x = "Sizes", y = "Prices") # Add labels
```



Note : Remaining notes will be updated Soon

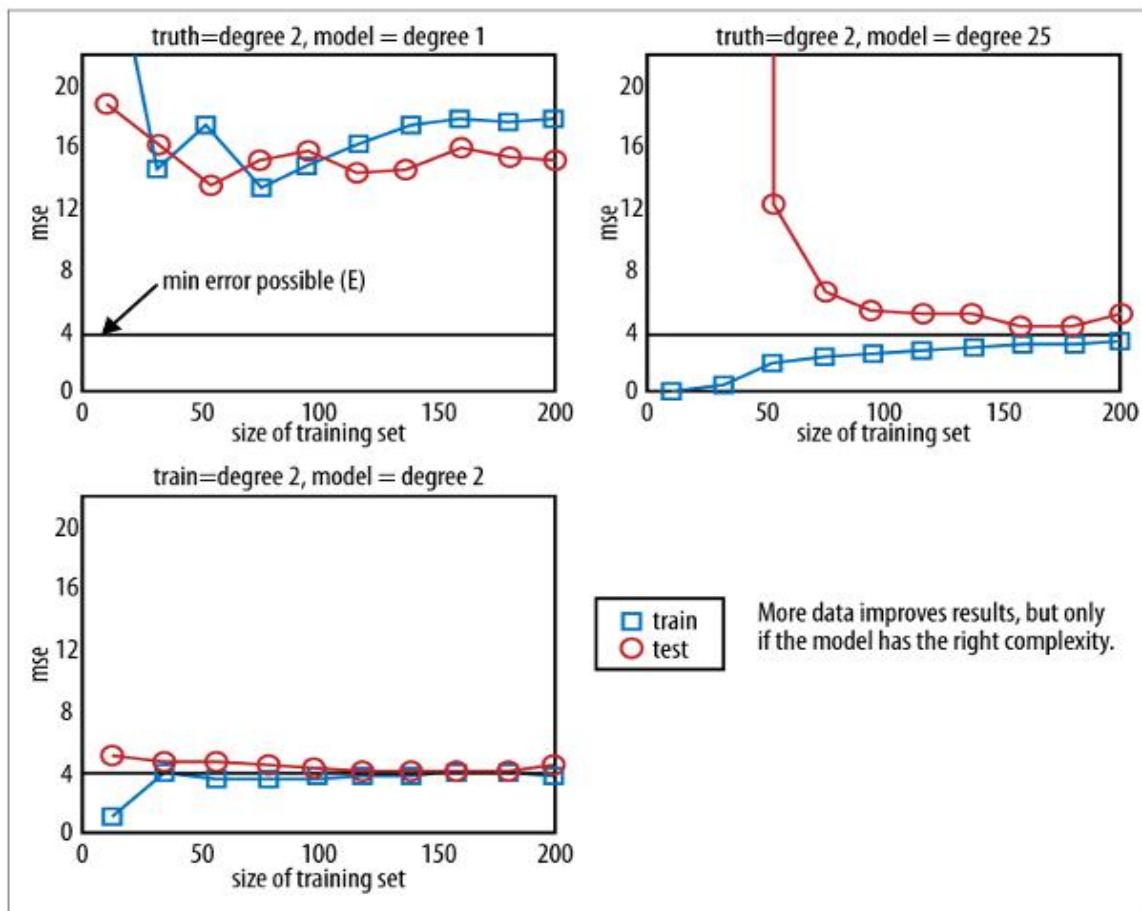


Figure 3-6. Comparing mean squared error in training and test set, taken from a slide of Professor Nando de Freitas; here, the ground truth is known because it came from a dataset with data simulated from a known distribution



2. KNN Algorithm

- The K-Nearest Neighbors (KNN) algorithm is a simple, supervised machine learning algorithm that can be used for both classification and regression tasks.
- It works based on the principle that similar data points (neighbors) are likely to have similar outcomes.

KNN Algorithm Steps

1. **Data Preparation:** Collect and prepare the dataset, ensuring it's in a suitable format for analysis.
2. **Choosing K:** Select the number of neighbors (K) to consider for making predictions.
3. **Distance Calculation:** For a given data point that needs to be classified or predicted, calculate the distance between this point and all other points in the dataset. Common distance metrics include Euclidean, Manhattan, and Minkowski distances.
4. **Identify Neighbors:** Identify the K closest data points (neighbors) to the data point in question.
5. **Voting/Prediction:**
 - **Classification:** The class with the majority among the K neighbors is assigned to the data point.
 - **Regression:** The average value of the K neighbors is assigned to the data point.

Key Points of KNN Algorithm

- **Distance Metrics:** Euclidean distance is most common, but other metrics can be used depending on the problem.
- **Choosing K:** The value of K can significantly impact the algorithm's performance. A common approach is to use cross-validation to find the optimal K.
- **Data Scaling:** KNN is sensitive to the scale of the data. It's important to normalize or standardize the data before applying KNN.

Example Data Set:

X1	X2	Label
1	2	A
2	3	A
3	3	B
6	5	B
7	8	B
8	8	A

Step-by-Step Example:

We want to predict the label of a new point (4, 4) using KNN with K=3.

1. **Choose K:** We choose K=3.
2. **Calculate Distances:** Compute the Euclidean distance from (4, 4) to all points in the dataset.
 - Distance to (1, 2): $\sqrt{(4-1)^2 + (4-2)^2} = \sqrt{9+4} = \sqrt{13} \approx 3.61$
 - Distance to (2, 3): $\sqrt{(4-2)^2 + (4-3)^2} = \sqrt{4+1} = \sqrt{5} \approx 2.24$
 - Distance to (3, 3): $\sqrt{(4-3)^2 + (4-3)^2} = \sqrt{1+1} = \sqrt{2} \approx 1.41$
 - Distance to (6, 5): $\sqrt{(4-6)^2 + (4-5)^2} = \sqrt{4+1} = \sqrt{5} \approx 2.24$
 - Distance to (7, 8): $\sqrt{(4-7)^2 + (4-8)^2} = \sqrt{9+16} = \sqrt{25} = 5$
 - Distance to (8, 8): $\sqrt{(4-8)^2 + (4-8)^2} = \sqrt{16+16} = \sqrt{32} \approx 5.66$
3. **Identify Neighbors:** The 3 closest points to (4, 4) are:
 - (3, 3) with distance ≈ 1.41 (Label B)
 - (2, 3) with distance ≈ 2.24 (Label A)
 - (6, 5) with distance ≈ 2.24 (Label B)
4. **Voting/Prediction:**
 - Labels of the 3 closest points: B, A, B
 - Majority label is B

Thus, the predicted label for the new point (4, 4) is **B**.

Implementation in R Program

```
# Install and load the required package
install.packages("class")
library(class)

# Example dataset
data <- data.frame(
  X1 = c(1, 2, 3, 6, 7, 8),
  X2 = c(2, 3, 3, 5, 8, 8),
  Label = factor(c('A', 'A', 'B', 'B', 'B', 'A'))
)

# Extract features and labels
features <- data[, 1:2]
labels <- data$Label

# Set the number of neighbors
k <- 3

# Apply KNN
predicted_label <- knn(train = features, test = new_point, cl = labels, k = k)
print(paste("The predicted label for point (4, 4) is", predicted_label))
```

KNN Applications

1. Image Classification
2. Recommendation Systems
3. Medical Diagnosis
4. Fraud Detection
5. Customer Segmentation

Distance/ Similarity Metrics

When working with various types of data, understanding and defining **similarity or distance between data points** is crucial. Here we discuss several common similarity and distance metrics, each with its unique definition of "**closeness.**" These metrics are used in different contexts and types of data to determine how similar or different data points are.

1. Euclidean Distance :

Euclidean distance is the straight-line distance between two points in a plane. It is commonly used for real-valued attributes that can be plotted in multidimensional space.

Formula:

$$d(x, y) = \sqrt{\sum_{i=1}^k (x_i - y_i)^2}$$

Example:

Consider two points in 2D space, $x = (1, 2)$ and $y = (4, 6)$. The Euclidean distance between these points is:

$$d(x, y) = \sqrt{(1 - 4)^2 + (2 - 6)^2} = \sqrt{9 + 16} = \sqrt{25} = 5$$

2. Cosine Similarity:

Cosine similarity measures the cosine of the angle between two non-zero vectors of an inner product space. It is used to determine how similar two vectors are, yielding a value between **-1 (exact opposite) and 1 (exactly the same), with 0 meaning no correlation.**

Formula:

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

Example:

For vectors $x = (1, 0, -1)$ and $y = (-1, 0, 1)$:

$$\cos(x, y) = \frac{(1 \cdot -1) + (0 \cdot 0) + (-1 \cdot 1)}{\sqrt{1^2 + 0^2 + (-1)^2} \cdot \sqrt{(-1)^2 + 0^2 + 1^2}} = \frac{-1 - 1}{\sqrt{2} \cdot \sqrt{2}} = \frac{-2}{2} = -1$$

3. Jaccard Distance/Similarity

Jaccard similarity measures the similarity between finite sets. It is defined as the size of the intersection divided by the size of the union of the sets.

Formula:

$$J(A, B) = \frac{|A \cap B|}{|A \cup B|}$$

Example:

For sets $A = \{1, 2, 3\}$ and $B = \{2, 3, 4\}$:

$$J(A, B) = \frac{|A \cap B|}{|A \cup B|} = \frac{2}{4} = 0.5$$

4. Mahalanobis Distance

Mahalanobis distance is a measure of the distance between a point and a distribution. It accounts for the correlations of the data set and is scale-invariant.

Formula:

$$d(x, y) = \sqrt{(x - y)^T S^{-1} (x - y)}$$

where S is the covariance matrix of the data.

Example:

Consider vectors $x = (1, 2)$ and $y = (2, 3)$, and covariance matrix $S = \begin{bmatrix} 1 & 0.5 \\ 0.5 & 1 \end{bmatrix}$:

$$d(x, y) = \sqrt{(1 - 2, 2 - 3) \begin{bmatrix} 1 & 0.5 \\ 0.5 & 1 \end{bmatrix}^{-1} \begin{bmatrix} 1 - 2 \\ 2 - 3 \end{bmatrix}}$$

5. Hamming Distance

Hamming distance measures, the number of positions at which the corresponding symbols differ. It is used for strings of the same length.

Example:

For strings "olive" and "ocean":

The Hamming distance is **4 (differences at positions 2, 3, 4, and 5)**.

6. Manhattan Distance

Manhattan distance (also known as L1 distance) measures the sum of the absolute differences of their Cartesian coordinates. It is like a taxi driving on a grid of streets.

Formula:

$$d(x, y) = \sum_{i=1}^k |x_i - y_i|$$

Example:

For points $x = (1, 2)$ and $y = (4, 6)$:

$$d(x, y) = |1 - 4| + |2 - 6| = 3 + 4 = 7$$

Custom Distance Metrics

Custom distance metrics are often created to handle specific needs in datasets that contain mixed types of data (e.g., numerical and categorical). Here is an example to illustrate how a custom distance metric can be designed and implemented.

Scenario

You are working with a movie dataset that includes the following attributes for each movie:

- Budget (in millions of dollars) - Numerical
- Number of actors - Numerical
- Genre - Categorical

You need to define a custom distance metric that appropriately handles these mixed data types.

Example Movies:

Let's consider two movies:

- **Movie A:**

Budget = \$50 million, Number of Actors = 10, Genre = Action

- **Movie B:**

Budget = \$80 million, Number of Actors = 12, Genre = Comedy

Step-by-Step Calculation

Step 1: Normalize the Numerical Attributes

Normalization scales the numerical values to a common range, usually [0, 1], to ensure fair contribution to the overall distance.

For simplicity, assume the following normalization (based on min-max normalization):

Budget: min = \$0 million, max = \$100 million

Number of Actors: min = 0, max = 20

Normalized values:

Movie A:

Normalized Budget = $(50 - 0) / (100 - 0) = 0.5$

Normalized Number of Actors = $(10 - 0) / (20 - 0) = 0.5$

Movie B:

Normalized Budget = $(80 - 0) / (100 - 0) = 0.8$

Normalized Number of Actors = $(12 - 0) / (20 - 0) = 0.6$

Step 2: Calculate Normalized Euclidean Distance for Numerical Attributes

$$d_{\text{num}} = \sqrt{(0.5 - 0.8)^2 + (0.5 - 0.6)^2} = \sqrt{0.09 + 0.01} = \sqrt{0.1} \approx 0.316$$

Step 3: Calculate Categorical Distance for Genre

Since the genres are different (Action vs. Comedy), assign a distance of 10:

$$d_{\text{cat}} = 10$$

Step 4: Combine the Distances

Combine the distances from numerical and categorical attributes. Since the categorical distance is significantly larger, it will dominate unless we scale it appropriately.

For the purpose of this example, assume we directly add them:

$$d_{\text{total}} = d_{\text{num}} + d_{\text{cat}} = 0.316 + 10 = 10.316$$

Custom Distance Metric Formula

In general, you can define your custom distance metric as follows:

$$d_{\text{custom}}(x, y) = w_{\text{num}} \cdot d_{\text{num}}(x, y) + w_{\text{cat}} \cdot d_{\text{cat}}(x, y)$$

Where:

- $d_{\text{num}}(x, y)$ is the normalized Euclidean distance for numerical attributes.
- $d_{\text{cat}}(x, y)$ is the categorical distance.
- w_{num} and w_{cat} are weights that balance the contributions of numerical and categorical distances.

Adjusting these weights allows fine-tuning the distance metric to best fit the specific needs of your data and application.

Training and Test Sets

When you are working with a machine learning algorithm, the general process involves two main phases: training and testing. Here's a simple explanation of each phase:

Training Phase

Purpose: To create and train a model using known data.

Process:

- You use a portion of your data, called the training set, where the outcomes (or labels) are already known.
- The model learns from this data. For example, in a dataset of people with their ages, incomes, and credit scores labeled as "high" or "low," the model learns the patterns and relationships between these attributes.
- In the k-NN (k-Nearest Neighbors) algorithm, the training phase involves simply reading in the data with the known labels (e.g., "high" or "low" credit).

Testing Phase

- **Purpose:** To evaluate how well the trained model performs on new, unseen data.
- **Process:**
 - You use a different portion of your data, called the test set, which was not used during training.
 - In the test set, you pretend that you don't know the outcomes (or labels). The model makes predictions based on what it learned during training.
 - You then compare the model's predictions to the actual known outcomes in the test set to see how accurate the model is.

Example with Steps

Consider you have a dataset with 1,000 rows, each containing information about age, income, and credit score (either "high" or "low").

Here's how you could split this data into training and test sets and use them:

1. Load the Data:

```
> head(data)
```

```
  age income credit
1  69    79  low
2  66    17  low
3  49    26  low
4  49    71  low
5  58    57 high
6  44    79 high
```

2. Define the Number of Rows and Sampling Rate:

```
n.points <- 1000 # Total number of rows in the dataset
sampling.rate <- 0.8 # 80% for training, 20% for testing
```

3. Calculate the Number of Test Set Labels:

```
num.test.set.labels <- n.points * (1 - sampling.rate) # 20% of the data
```

4. Randomly Sample Training Set Rows:

```
training <- sample(1:n.points, sampling.rate * n.points, replace=FALSE)
```

5. Create Training Set:

```
train <- subset(data[training, ], select = c(age, income))  
cl <- data$credit[training] # Labels for the training set
```

6. Define Test Set Rows and Create Test Set:

```
testing <- setdiff(1:n.points, training)  
test <- subset(data[testing, ], select = c(age, income))  
true.labels <- data$credit[testing] # True labels for the test set
```

How It Works

Training Set: 80% of the data (randomly selected) used to train the model. This set includes the "credit" labels.

Test Set: The remaining 20% of the data used to test the model. This set does not provide the "credit" labels to the model during testing but keeps them for evaluating the model's performance.

By comparing the model's predictions on the test set with the true labels (which are known but not used during prediction), you can assess how well your model is likely to perform on new, unseen data.

Summary

Training Set: Data used to train the model (with known outcomes).

Test Set: Data used to test the model's performance (with known outcomes but hidden from the model during prediction).

This approach ensures that the model is evaluated on data it hasn't seen before, giving a good indication of how it will perform in real-world scenarios.

Picking an Evaluation Metric:

Evaluating a model's performance isn't straightforward and varies based on the specific problem and its context. Here's a brief overview of how to choose and understand evaluation metrics:

Customizing Evaluation Metrics:

- Different kinds of misclassifications may have different consequences. For example, in medical diagnosis, a false negative (failing to identify a disease) can be more harmful than a false positive (incorrectly diagnosing a disease).
- Collaborating with domain experts can help tailor the evaluation metric to reflect the real-world implications of errors.

Sensitivity and Specificity:

- **Sensitivity (True Positive Rate):** The probability of correctly identifying a condition (e.g., diagnosing an ill patient as ill). High sensitivity minimizes false negatives.
- **Specificity (True Negative Rate):** The probability of correctly identifying the absence of a condition (e.g., diagnosing a well patient as well). High specificity minimizes false positives.
- Balancing sensitivity and specificity is crucial, as overemphasizing one can negatively impact the other.

Precision and Recall:

- **Precision:** The ratio of true positive results to the total predicted positive results. It measures the accuracy of the positive predictions.
- **Recall:** Another term for sensitivity, indicating the model's ability to identify all relevant cases within a dataset.

These terms are often used in information retrieval and other fields with different names for similar concepts.

Accuracy and Misclassification Rate:

Accuracy: The ratio of correct predictions (both true positives and true negatives) to the total number of predictions.

Misclassification Rate: Calculated as $1 - \text{accuracy}$, it represents the proportion of incorrect predictions.

While accuracy is a common metric, it might not be sufficient in cases with imbalanced datasets where one class dominates.

In summary, selecting the right evaluation metric depends on the specific context and the consequences of different types of errors. Metrics such as sensitivity, specificity, precision, recall, accuracy, and misclassification rate provide various lenses to assess and balance model performance based on the problem at hand.

Formulas for Evaluation Metrics

Here are the formulas for the commonly used evaluation metrics in machine learning:

1. Sensitivity (True Positive Rate or Recall):

$$\text{Sensitivity} = \frac{\text{True Positives (TP)}}{\text{True Positives (TP)} + \text{False Negatives (FN)}}$$

Sensitivity measures the proportion of actual positives that are correctly identified.

2. Specificity (True Negative Rate):

$$\text{Specificity} = \frac{\text{True Negatives (TN)}}{\text{True Negatives (TN)} + \text{False Positives (FP)}}$$

Specificity measures the proportion of actual negatives that are correctly identified.

3. **Precision (Positive Predictive Value):**

$$\text{Precision} = \frac{\text{True Positives (TP)}}{\text{True Positives (TP)} + \text{False Positives (FP)}}$$

Precision measures the proportion of positive identifications that are actually correct.

4. **Recall:**

$$\text{Recall} = \frac{\text{True Positives (TP)}}{\text{True Positives (TP)} + \text{False Negatives (FN)}}$$

Recall is another term for sensitivity, measuring the proportion of actual positives that are correctly identified.

5. **Accuracy:**

$$\text{Accuracy} = \frac{\text{True Positives (TP)} + \text{True Negatives (TN)}}{\text{Total Population}}$$

Accuracy measures the proportion of true results (both true positives and true negatives) among the total number of cases examined.

6. **Misclassification Rate:**

$$\text{Misclassification Rate} = \frac{\text{False Positives (FP)} + \text{False Negatives (FN)}}{\text{Total Population}}$$

Misclassification rate is the proportion of incorrect predictions (both false positives and false negatives) among the total number of cases examined.

Alternatively, misclassification rate can be expressed as:

$$\text{Misclassification Rate} = 1 - \text{Accuracy}$$

Summary of the Metrics

- Sensitivity (Recall): Focuses on correctly identifying positive cases.
- Specificity: Focuses on correctly identifying negative cases.
- Precision: Focuses on the accuracy of positive predictions.
- Recall: Focuses on the ability to find all relevant positive cases.
- Accuracy: Overall correctness of the model.
- Misclassification Rate: Overall incorrectness of the model.

Modeling Assumptions in k-NN Algorithm:

In the context of the k-NN (k-Nearest Neighbours) algorithm, several assumptions are made despite it being a nonparametric approach (which means there are no assumptions about the underlying data-generating distributions and no parameters to estimate). Here are the key assumptions:

Feature Space and Distance:

The data exists in a feature space where a notion of "distance" between data points is meaningful. This distance is crucial for determining the neighbors of a point.

Labeled Training Data:

The training data has labels or classifications for two or more classes. This labeling is necessary for the algorithm to make predictions about new, unlabeled data points.

Choosing the Number of Neighbors (k):

You must select the number of neighbors (k) to use in the algorithm. The choice of k can significantly impact the performance of the model.

Association Between Features and Labels:

There is an implicit assumption that the observed features and the labels are somehow related. The algorithm relies on this relationship to predict labels for new data points. The strength and validity of this assumption can be evaluated using an appropriate evaluation metric.

Additional Considerations

Feature and k Tuning: You might need to experiment with different features and values of k to optimize the model's performance, but this brings a risk of overfitting, where the model performs well on training data but poorly on new, unseen data.

Context in Supervised Learning

Both linear regression and k-NN are forms of supervised learning, where the goal is to learn a function that maps input features (x) to output labels (y) based on observed data. This contrasts with unsupervised learning, where the goal is to find patterns or structures in data without predefined labels.

These assumptions are fundamental to understanding how the k-NN algorithm operates and what considerations need to be made to ensure its effective application and evaluation.

3.K Means Algorithm

The K-Means algorithm is a popular clustering technique used to partition a dataset into K clusters. The goal is to minimize the variance within each cluster, thereby ensuring that data points within a cluster are as similar as possible. Here's a step-by-step explanation of how the K-Means algorithm works, along with an example dataset.

Steps of K-Means Algorithm

1. Initialize Centroids:

- Randomly select **K initial centroids** from the dataset. These centroids will represent the initial cluster centers.

2. Assign Data Points to Clusters:

- For each data point, calculate the **Euclidean distance** to each centroid.
- Assign each data point to the cluster with the nearest centroid.

3. Update Centroids:

- For each cluster, calculate the new centroid by taking the mean of all data points assigned to that cluster.

4. Repeat:

- Repeat steps 2 and 3 until the centroids no longer change significantly, or for a fixed number of iterations.

5. Convergence:

- The algorithm converges when the assignments of data points to clusters no longer change.

Example Dataset

Let's consider a simple 2D dataset for illustration:

Data Point	X	Y
A	1	2
B	1	4
C	3	4
D	5	2
E	5	4

Applying K-Means Algorithm

1. Initialize Centroids:

- Suppose we choose **K=2** (we want to partition the data into 2 clusters).
- Randomly select two initial centroids. For simplicity, let's choose points **A (1, 2)** and **D (5, 2)**.

2. Assign Data Points to Clusters:

- Calculate distances and assign points to the nearest centroid.

Data Point	Distance to (1, 2)	Distance to (5, 2)	Assigned Cluster
A (1, 2)	0	4	1
B (1, 4)	2	4.47	1
C (3, 4)	2.83	3.61	1
D (5, 2)	4	0	2
E (5, 4)	4.47	2	2

3. Update Centroids: Calculate the new centroids for each cluster.

Cluster 1: (A, B, C)

- New Centroid = $((1+1+3)/3, (2+4+4)/3) = (1.67, 3.33)$

Cluster 2: (D, E)

- New Centroid = $((5+5)/2, (2+4)/2) = (5, 3)$

4. Repeat

Reassign data points to the new centroids and update centroids again if necessary.

Data Point	Distance to (1.67, 3.33)	Distance to (5, 3)	Assigned Cluster
A (1, 2)	1.86	4.12	1
B (1, 4)	1.05	4.12	1
C (3, 4)	1.67	2.24	1
D (5, 2)	3.61	1	2
E (5, 4)	3.81	1	2

New centroids remain the same as above: Cluster 1 (1.67, 3.33), Cluster 2 (5, 3)

5. Convergence:

The algorithm converges when the centroids no longer change. In this example, the clusters have stabilized after the second iteration.

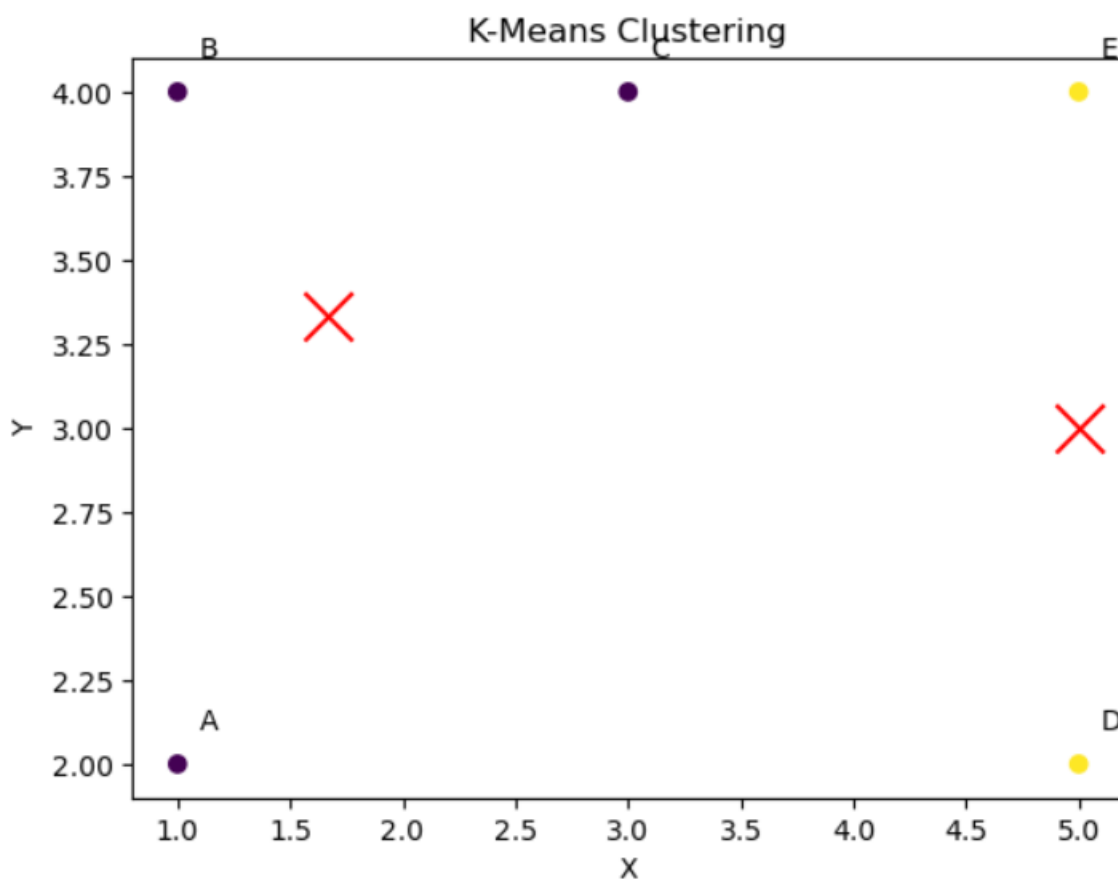
6. Visual Representation:

The final clusters can be visualized as follows:

Cluster 1: Points A (1, 2), B (1, 4), C (3, 4) with centroid (1.67, 3.33)

Cluster 2: Points D (5, 2), E (5, 4) with centroid (5, 3)

This process of iteratively assigning data points to the nearest centroid and recalculating centroids continues until the algorithm converges, resulting in well-defined clusters.



k-means has some known issues:

- Choosing k is more an art than a science, although there are bounds: $1 \leq k \leq n$, where n is number of data points.
- There are convergence issues—the solution can fail to exist, if the algorithm falls into a loop, for example, and keeps going back and forth between two possible solutions, or in other words, there isn't a single unique solution.
- Interpretability can be a problem—sometimes the answer isn't at all useful. Indeed that's often the biggest problem.

TOCX10

Module2 : Question Bank

Data Science Process and EDA

1. With a diagram, describe the data science process. (08 Marks) (L3)
2. What is Data Science Process? Write about a data scientists role in this process. (L2)
3. Describe the Exploratory Data Analysis in Data Science with example. (L2)
4. Write about the Philosophy of EDA. (L2)

Real Direct Case Study

1. Write about Real Direct case Study. (L2)
2. Discuss the real direct real estate business and their data strategy. (L4)

Linear Regression

1. Explain how to build and fit a linear regression model along with its evaluation metrics. (L3)
2. Explain the linear regression with an example in brief. (L2)

K-Nearest Neighbors (KNN)

1. Explain K – Nearest Neighbour Algorithm. (L2)
2. Discuss the various distance metrics that can be used in KNN. (L4)
3. Discuss the different similarity or distance metrics. Explain the steps involved in K Nearest Neighbour algorithm with an example. (L4)

K-Means

1. Explain the K-means algorithm. List the issues associated with it. (07 Marks) (L2)
2. Write a Short note on K – Means algorithm. (L2)

Mixed Machine Learning Algorithms

1. What are the three basic ML Algorithms? Explain the following in Linear Regression, K- nearest neighbour and K – means. (L2)

Module2: Multiple Choice Questions with Answers

Data Science Process and EDA

1. **What is the first step in the data science process?**

- A) Data Cleaning
- B) Data Collection
- C) Data Modeling
- D) Data Visualization
- **Answer:** B) Data Collection

2. **Which of the following is a key responsibility of a data scientist?**

- A) Managing databases
- B) Writing software documentation
- C) Extracting insights from data
- D) Network administration
- **Answer:** C) Extracting insights from data

3. **In the data science process diagram, what follows after data collection?**

- A) Data Cleaning
- B) Data Analysis
- C) Data Modeling
- D) Data Visualization
- **Answer:** A) Data Cleaning

4. **Which phase of the data science process involves summarizing the main characteristics of the data?**

- A) Data Collection
- B) Data Cleaning
- C) Exploratory Data Analysis
- D) Data Modeling
- **Answer:** C) Exploratory Data Analysis

5. **What is the main goal of Exploratory Data Analysis (EDA)?**

- A) To build predictive models
- B) To summarize the main characteristics of the data
- C) To clean the data
- D) To store the data
- **Answer:** B) To summarize the main characteristics of the data

6. **Which tool is commonly used for EDA in Python?**

- A) Pandas
- B) Numpy
- C) Matplotlib
- D) All of the above
- **Answer:** D) All of the above

7. **What is a common graphical method used in EDA?**

- A) Bar Chart
- B) Histogram
- C) Scatter Plot
- D) All of the above
- **Answer:** D) All of the above

8. **Which of the following best describes the philosophy of EDA?**

- A) Confirming hypotheses about the data
- B) Cleaning the data for modeling
- C) Discovering patterns and relationships in the data
- D) Storing data efficiently
- **Answer:** C) Discovering patterns and relationships in the data

9. **EDA is a/an ___ step in the data science process.**

- A) Optional
- B) Initial
- C) Intermediate
- D) Final
- **Answer:** C) Intermediate

10. **Which of the following is not a part of the data science process?**

- A) Data Collection
- B) Data Cleaning
- C) Data Modeling
- D) Data Hiding
- **Answer:** D) Data Hiding

Real Direct Case Study

11. **What is the primary focus of the Real Direct case study?**

- A) Medical data analysis
- B) Real estate business strategy
- C) Financial market prediction
- D) E-commerce analysis
- **Answer:** B) Real estate business strategy

12. **Real Direct's data strategy mainly involves which of the following?**

- A) Improving website design
- B) Enhancing data-driven decision making
- C) Reducing employee count
- D) Expanding physical store locations
- **Answer:** B) Enhancing data-driven decision making

13. **Which type of data is most crucial for Real Direct's strategy?**

- A) Customer transaction data
- B) Employee performance data
- C) Social media data
- D) Weather data
- **Answer:** A) Customer transaction data

14. **The Real Direct case study demonstrates the use of data science in which industry?**

- A) Healthcare
- B) Education

- C) Real Estate
- D) Manufacturing
- **Answer:** C) Real Estate

15. **What was a key benefit achieved by Real Direct through its data strategy?**

- A) Reduced marketing costs
- B) Increased sales and customer satisfaction
- C) Improved product development
- D) Enhanced social media presence
- **Answer:** B) Increased sales and customer satisfaction

Linear Regression

16. **What is the main purpose of linear regression?**

- A) To classify data points
- B) To cluster data points
- C) To predict a continuous target variable
- D) To reduce dimensionality
- **Answer:** C) To predict a continuous target variable

17. **In the linear regression equation $y=mx+b$, what does 'm' represent?**

- A) The y-intercept
- B) The slope of the line
- C) The dependent variable
- D) The independent variable
- **Answer:** B) The slope of the line

18. **Which metric is commonly used to evaluate the performance of a linear regression model?**

- A) Accuracy
- B) Recall
- C) Mean Squared Error (MSE)
- D) F1 Score
- **Answer:** C) Mean Squared Error (MSE)

19. What is the main advantage of using linear regression?

- A) It is easy to interpret and implement
- B) It can handle non-linear relationships
- C) It requires no assumptions about the data
- D) It is resistant to outliers
- **Answer:** A) It is easy to interpret and implement

20. What is the role of the intercept term in a linear regression model?

- A) It adjusts the slope of the line
- B) It determines the strength of the relationship
- C) It provides the starting point of the line on the y-axis
- D) It minimizes the sum of squared errors
- **Answer:** C) It provides the starting point of the line on the y-axis

21. Which of the following is an example of a real-world application of linear regression?

- A) Image classification
- B) Stock price prediction
- C) Customer segmentation
- D) Sentiment analysis
- **Answer:** B) Stock price prediction

22. What does a high R-squared value indicate in linear regression?

- A) The model fits the data poorly
- B) The model fits the data well
- C) The model is overfitting the data
- D) The model is underfitting the data
- **Answer:** B) The model fits the data well

23. Which method is typically used to estimate the coefficients in a linear regression model?

- A) Maximum Likelihood Estimation
- B) Gradient Descent
- C) Ordinary Least Squares

- D) Support Vector Machine
- **Answer:** C) Ordinary Least Squares

24. **What is a residual in linear regression?**

- A) The predicted value
- B) The difference between the observed and predicted values
- C) The slope of the regression line
- D) The y-intercept
- **Answer:** B) The difference between the observed and predicted values

25. **When should you consider using polynomial regression instead of linear regression?**

- A) When the relationship between variables is linear
- B) When the relationship between variables is non-linear
- C) When you have categorical data
- D) When you have a large number of features
- **Answer:** B) When the relationship between variables is non-linear

K-Nearest Neighbors (KNN)

26. **What is the primary purpose of the K-Nearest Neighbors algorithm?**

- A) To perform clustering
- B) To perform classification and regression
- C) To reduce dimensionality
- D) To generate random data points
- **Answer:** B) To perform classification and regression

27. **In KNN, what does the 'K' represent?**

- A) The number of features
- B) The number of nearest neighbors to consider
- C) The distance metric used
- D) The size of the dataset
- **Answer:** B) The number of nearest neighbors to consider

28. **Which of the following is a common distance metric used in KNN?**

- A) Euclidean distance
- B) Manhattan distance
- C) Minkowski distance
- D) All of the above
- **Answer:** D) All of the above

29. **What is a key disadvantage of the KNN algorithm?**

- A) It is difficult to understand
- B) It requires a large amount of storage
- C) It is not suitable for classification tasks
- D) It cannot handle missing data
- **Answer:** B) It requires a large amount of storage

30. **In KNN, how is the class of a new data point determined in classification?**

- A) By averaging the classes of the nearest neighbors
- B) By taking the majority vote of the nearest neighbors
- C) By using the maximum likelihood estimation
- D) By performing gradient descent
- **Answer:** B) By taking the majority vote of the nearest neighbors

31. **Which scenario is suitable for using K-Nearest Neighbors?**

- A) Predicting continuous values in a dataset
- B) Classifying data points based on their features
- C) Reducing the number of features in a dataset
- D) Performing unsupervised learning
- **Answer:** B) Classifying data points based on their features

32. **What is one way to handle different scales of features in KNN?**

- A) Ignore the scale difference
- B) Normalize or standardize the features
- C) Increase the value of K
- D) Use a different distance metric
- **Answer:** B) Normalize or standardize the features

K-Means

33. **What is the primary objective of the K-means algorithm?**

- A) To classify data points
- B) To cluster data points into groups
- C) To reduce dimensionality
- D) To predict continuous values
- **Answer:** B) To cluster data points into groups

34. **How does K-means determine the optimal clusters?**

- A) By maximizing the distance between data points
- B) By minimizing the sum of squared distances within clusters
- C) By using a decision tree
- D) By performing linear regression
- **Answer:** B) By minimizing the sum of squared distances within clusters

35. **Which issue is commonly associated with the K-means algorithm?**

- A) It is very slow for small datasets
- B) It does not converge to a solution
- C) It is sensitive to the initial placement of centroids
- D) It is only suitable for binary classification
- **Answer:** C) It is sensitive to the initial placement of centroids

36. **What is a centroid in the context of K-means?**

- A) A data point that is closest to the origin
- B) The geometric center of a cluster
- C) A randomly chosen data point
- D) A feature that is ignored during clustering
- **Answer:** B) The geometric center of a cluster

37. **Which metric is often used to evaluate the quality of clustering in K-means?**

- A) Accuracy
- B) Silhouette score

- C) F1 score
- D) Precision
- **Answer:** B) Silhouette score

38. **What is a common approach to choose the optimal number of clusters in K-means?**

- A) Cross-validation
- B) Elbow method
- C) Gradient descent
- D) Decision tree
- **Answer:** B) Elbow method

39. **What happens if the number of clusters (K) is set too high in K-means?**

- A) The algorithm will not converge
- B) The clusters may become too small and lose meaningful patterns
- C) The computation time will decrease
- D) The algorithm will produce the same clusters regardless of K
- **Answer:** B) The clusters may become too small and lose meaningful patterns

40. **How are the initial centroids chosen in K-means?**

- A) They are always set to the origin
- B) They are randomly selected from the data points
- C) They are determined by gradient descent
- D) They are chosen based on the maximum likelihood
- **Answer:** B) They are randomly selected from the data points

41. **What is a limitation of the K-means algorithm?**

- A) It cannot handle large datasets
- B) It requires the number of clusters to be specified beforehand
- C) It can only be used for supervised learning
- D) It does not produce interpretable results
- **Answer:** B) It requires the number of clusters to be specified beforehand

Mixed Machine Learning Algorithms

42. Which of the following is a basic machine learning algorithm used for classification?

- A) Linear Regression
- B) K-Nearest Neighbors
- C) K-Means
- D) Principal Component Analysis
- **Answer:** B) K-Nearest Neighbors

43. Which algorithm is best suited for predicting a continuous target variable?

- A) K-Means
- B) K-Nearest Neighbors
- C) Linear Regression
- D) Decision Tree
- **Answer:** C) Linear Regression

44. What is a common application of the K-means algorithm?

- A) Predicting housing prices
- B) Classifying emails as spam or not spam
- C) Segmenting customers into distinct groups
- D) Identifying handwritten digits
- **Answer:** C) Segmenting customers into distinct groups

45. Which of the following algorithms can be used for both classification and regression tasks?

- A) Linear Regression
- B) K-Nearest Neighbors
- C) K-Means
- D) Logistic Regression
- **Answer:** B) K-Nearest Neighbors

46. In which scenario would you use K-means instead of KNN?

- A) When you want to classify data points

- B) When you want to predict a continuous target variable
- C) When you want to cluster data points into groups
- D) When you need to handle large amounts of labeled data
- **Answer:** C) When you want to cluster data points into groups

47. **Which metric is not typically used to evaluate a linear regression model?**

- A) R-squared
- B) Mean Squared Error (MSE)
- C) Accuracy
- D) Root Mean Squared Error (RMSE)
- **Answer:** C) Accuracy

48. **What is the primary objective of clustering in machine learning?**

- A) To predict future values
- B) To group similar data points together
- C) To reduce the number of features
- D) To classify data points
- **Answer:** B) To group similar data points together

49. **Which algorithm uses the concept of centroids for clustering?**

- A) Linear Regression
- B) K-Nearest Neighbors
- C) K-Means
- D) Decision Tree
- **Answer:** C) K-Means

50. **Which machine learning algorithm is sensitive to the scale of features?**

- A) Linear Regression
- B) K-Nearest Neighbors
- C) K-Means
- D) All of the above
- **Answer:** D) All of the above