TOPIC - Executing SQL and R: A Holistic Strategy for Assessing Insurance Customer Data

Table of contents

Note: The word count is from Introduction to Limitations only.

Introduction

The insurance company aims to set up an analytics unit to delve into customer nuances across various insurance products and marketing channels. With data scattered across multiple files, they're contemplating a shift from SQL to R for analysis. My strategy begins with consolidating their data using SQL followed by R. This phase includes an initial descriptive analysis to identify trends. The objective is to derive insights for personalized marketing strategies. This process ensures a seamless transition from SQL to R, maintaining uninterrupted analytical processes.

Methodology

Saltz (2021) highlights CRISP-DM as a widely adopted framework in data mining projects, providing structured guidance for data scientists. Its iterative nature aligns with business goals, facilitating effective data handling and valuable insight extraction.

The project follows a structured flow. Firstly, it aims to define details of data sources, identify and address issues affecting data quality. Once these challenges are addressed, data gets consolidated into a unified schema called ABT. This comprehensive table serves as foundation for generating insights, aiding company in enhancing its communication and marketing strategies.

Data Sources

Overview

Customer data and insurance policies are stored across four Excel files: Data 1_Customer acts as customer table with CustomerID as its primary key. Data 2_Motor_Policies represents motor policies, using MotorID as its primary key. Data 3_Health_Policies serves as health policies table, utilizing HealthID as primary key. Lastly, Data 4_Travel_Policies functions as travel policies table, employing TravelID as its primary key. These files are imported individually for further discussion.

Customer Table

The table 'Customer' contains primary and foreign keys alongside various personal details. It includes CustomerID as the primary key, capturing personal information such as Title, GivenName, MiddleInitial, Surname, CardType (Visa/MasterCard/0), Occupation, Gender, Age, and Location (Rural/Urban). Additionally, it records the preferred communication channel (SMS, Email, or Phone). Moreover, it holds Foreign Keys (MotorID, HealthID, TravelID) linking to other tables—Motor Policies, Health Policies, and Travel Policies respectively. This table has 14 fields, 4085 observations.

Travel Policies Table

The 'Travel Policies' table features TravelID as the primary key and includes policyStart and policyEnd dates. It records TravelType, specifying the type of travel insurance—options include Backpacker, Senior, Business, Premium, or Standard—held by the respective customer. This table has 4 fields, 2108 observations.

Health Policies Table

The 'Health Policies' table contains insurance policy details with HealthID as the primary key. It includes policyStart and policyEnd dates, HealthType (Level1, Level2, or Level3), HealthDependentsAdults (number of dependent adults), and DependentsKids (number of dependent children) covered under the customer's health insurance policy. This table has 6 fields, 2543 observations.

Motor policies Table

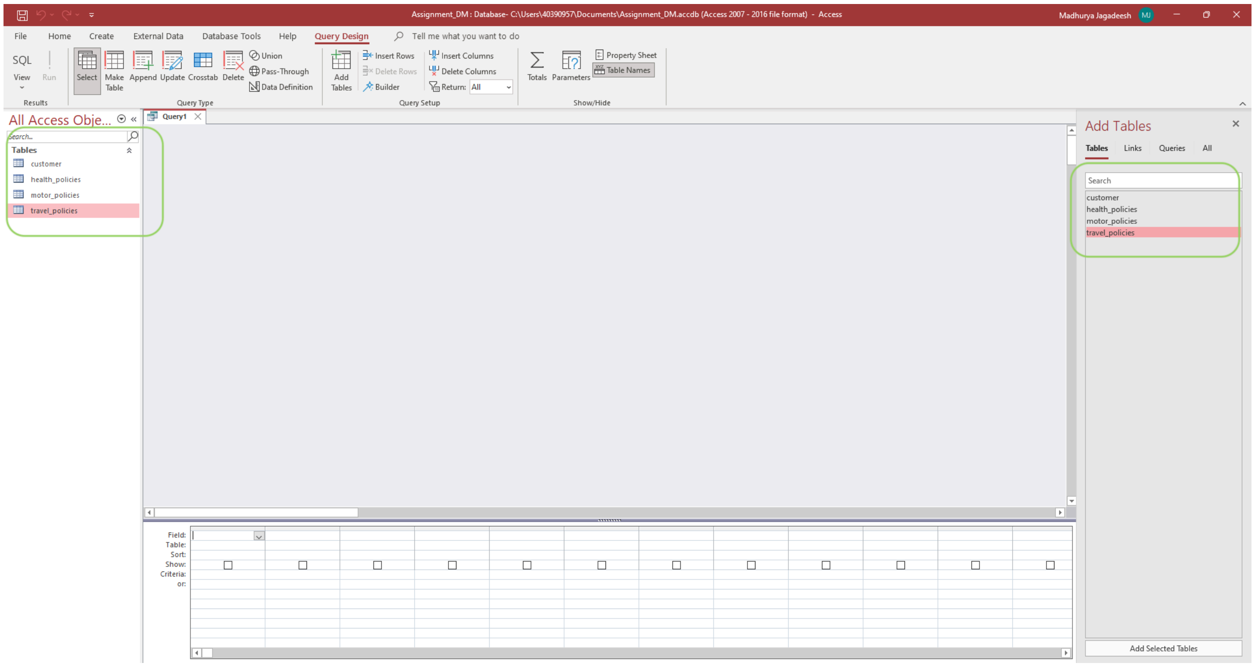
The 'Motor Policies' table contains data related to insurance policies. It includes MotorID as the primary key and details like PolicyStart and PolicyEnd dates. The table records MotorType (Single or Bundle), veh_value (vehicle value in $10,000s), Exposure (0-1), clm (claim occurrence), Numclaims, v_body (vehicle body type), v_age (vehicle age), and LastClaimDate for the most recent claim. This table has 11 fields, 3361 observations.

Import flat files

Import all flat files into DataBase

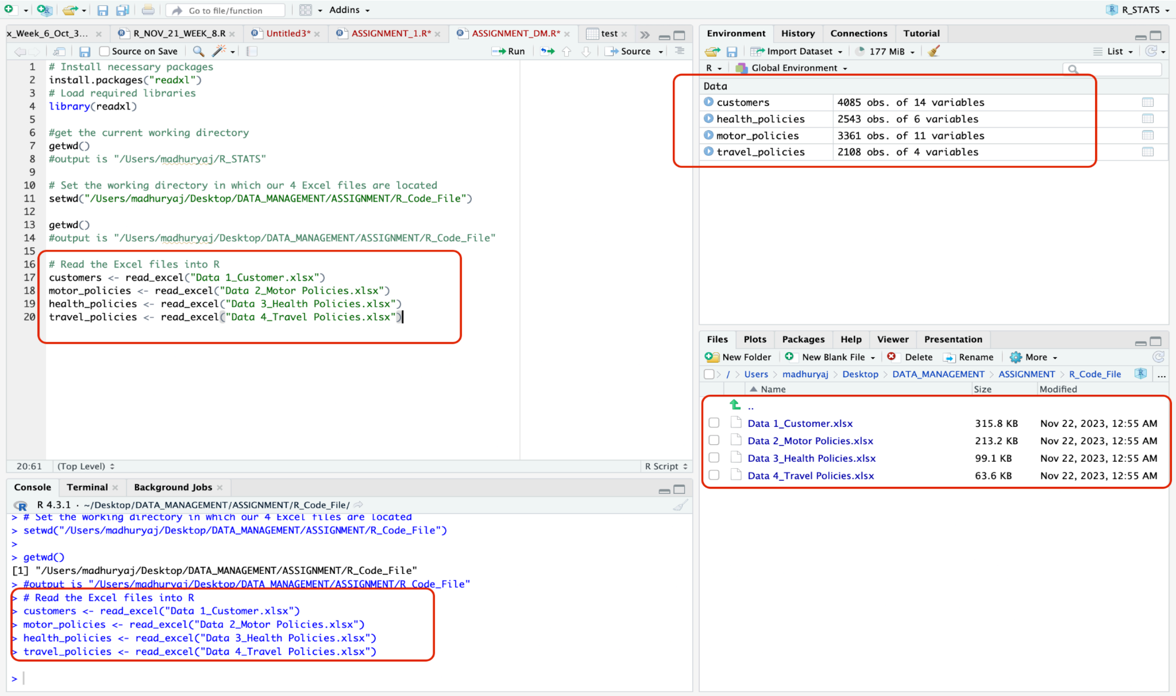
In this analysis, MS Access is used, which is a relational database management system (RDBMS) that allows users to store, manage, and manipulate data. It offers tools for creating databases, forms, queries, and reports, making it easier to organize and analyze information within a Windows environment (McFadyen, 2016).

To import Excel flat files into Microsoft Access, open Access and go to 'External Data' tab. Click 'Excel,' choose your file, and decide to import it into a new table. Set import options, i.e., select primary keys appropriately for every table, preview and confirm data, then complete import by giving an appropriate name for every table.



Import all flat files into R Studio

To read flat-file Excel data into R, use 'readxl' package. Start by installing it if needed: install.packages("readxl") and load package library(readxl). The `read_excel()` function is used to import data, specifying file path or name. Data is stored as a dataframe as customers, motor_policies, health_policies, travel_policies respectively. Refer Code:R_1.



```
# Install necessary packages
install.packages("readxl")
# Load required libraries
library(readxl)

#get the current working directory
getwd()
#output is "/Users/madhuryaj/R_STATS"

# Set the working directory in which our 4 Excel files are located
setwd("/Users/madhuryaj/Desktop/DATA_MANAGEMENT/ASSIGNMENT/R_Code_File")

getwd()
#output is "/Users/madhuryaj/Desktop/DATA_MANAGEMENT/ASSIGNMENT/R_Code_File"

# Read the Excel files into R
customers <- read_excel("Data 1_Customer.xlsx")
motor_policies <- read_excel("Data 2_Motor Policies.xlsx")
health_policies <- read_excel("Data 3_Health Policies.xlsx")
travel_policies <- read_excel("Data 4_Travel Policies.xlsx")
```

```
Data
customers          4085 obs. of 14 variables
health_policies    2543 obs. of 6 variables
motor_policies     3361 obs. of 11 variables
travel_policies    2108 obs. of 4 variables
```

Name	Size	Modified
..		
Data 1_Customer.xlsx	315.8 KB	Nov 22, 2023, 12:55 AM
Data 2_Motor Policies.xlsx	213.2 KB	Nov 22, 2023, 12:55 AM
Data 3_Health Policies.xlsx	99.1 KB	Nov 22, 2023, 12:55 AM
Data 4_Travel Policies.xlsx	63.6 KB	Nov 22, 2023, 12:55 AM

```
R 4.3.1 · ~/Desktop/DATA_MANAGEMENT/ASSIGNMENT/R_Code_File/
> # Set the working directory in which our 4 Excel files are located
> setwd("/Users/madhuryaj/Desktop/DATA_MANAGEMENT/ASSIGNMENT/R_Code_File")
>
> getwd()
[1] "/Users/madhuryaj/Desktop/DATA_MANAGEMENT/ASSIGNMENT/R_Code_File"
> #output is "/Users/madhuryaj/Desktop/DATA_MANAGEMENT/ASSIGNMENT/R_Code_File"
> # Read the Excel files into R
> customers <- read_excel("Data 1_Customer.xlsx")
> motor_policies <- read_excel("Data 2_Motor Policies.xlsx")
> health_policies <- read_excel("Data 3_Health Policies.xlsx")
> travel_policies <- read_excel("Data 4_Travel Policies.xlsx")

> 
```

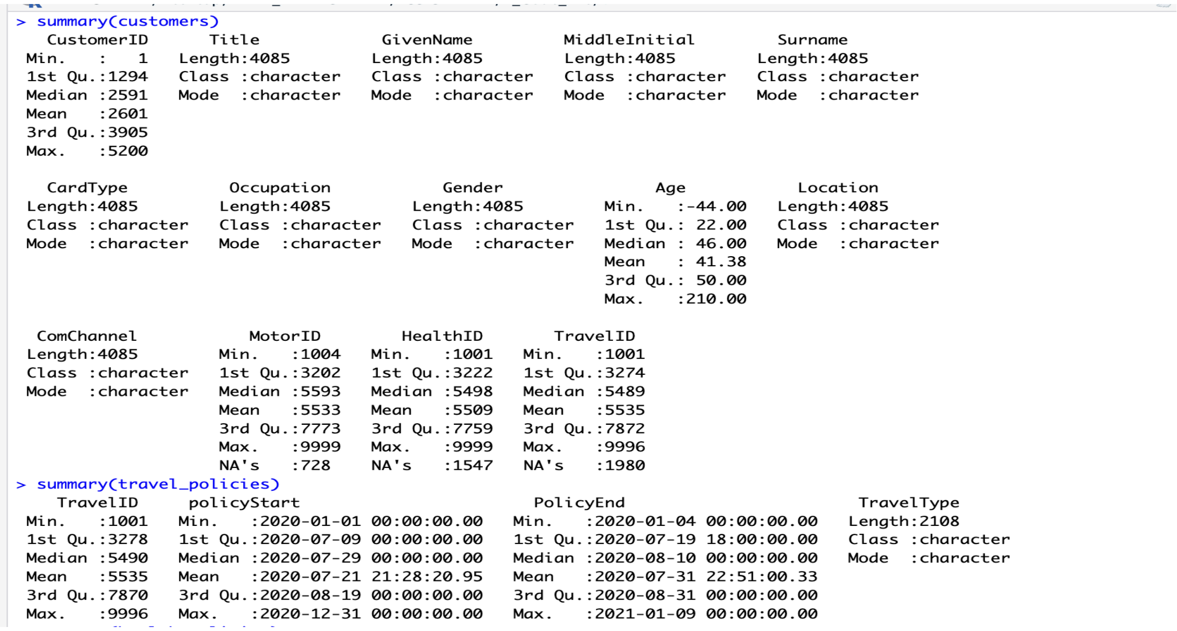
Analysis of the data

Initial descriptive statistics in SQL

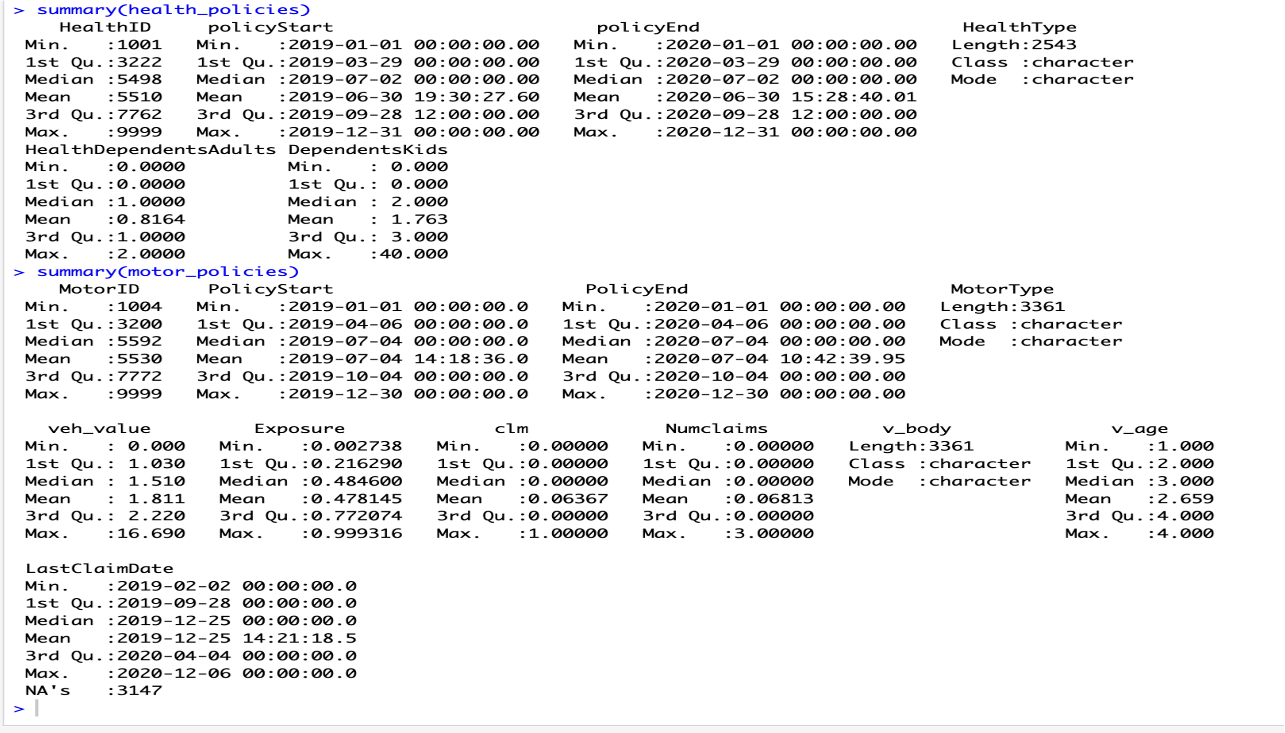
- As per Code:SQL_13, there are 4085 customers in this analysis, out of which only 975 brought all three insurances as per Code:SQL_12.
- The average age of customers is ~42 years in this dataset, as per Code:SQL_2.
- There are almost balanced male and female customers as per Code:SQL_10.

Summary statistics of the datasets in R

The summary() function in R provides basic statistical summaries such as minimum, 1st quartile, median, mean, 3rd quartile, maximum, and counts of non-missing values for each variable in dataset (Wickham *et al*., 2023). So for all 4 tables, summary statistics are checked. Refer Code:R_2.



```
> summary(customers)
   CustomerID       Title             GivenName          MiddleInitial        Surname
 Min.   :   1   Length:4085        Length:4085        Length:4085        Length:4085
 1st Qu.:1294   Class :character   Class :character   Class :character   Class :character
 Median :2591   Mode  :character   Mode  :character   Mode  :character   Mode  :character
 Mean   :2601
 3rd Qu.:3905
 Max.   :5200

   CardType          Occupation           Gender               Age           Location
 Length:4085        Length:4085        Length:4085        Min.   :-44.00   Length:4085
 Class :character   Class :character   Class :character   1st Qu.: 22.00   Class :character
 Mode  :character   Mode  :character   Mode  :character   Median : 46.00   Mode  :character
                                                          Mean   : 41.38
                                                          3rd Qu.: 50.00
                                                          Max.   :210.00

  ComChannel           MotorID        HealthID       TravelID
 Length:4085        Min.   :1004   Min.   :1001   Min.   :1001
 Class :character   1st Qu.:3202   1st Qu.:3222   1st Qu.:3274
 Mode  :character   Median :5593   Median :5498   Median :5489
                    Mean   :5533   Mean   :5509   Mean   :5535
                    3rd Qu.:7773   3rd Qu.:7759   3rd Qu.:7872
                    Max.   :9999   Max.   :9999   Max.   :9996
                    NA's   :728    NA's   :1547   NA's   :1980
> summary(travel_policies)
    TravelID     policyStart                      PolicyEnd                        TravelType
 Min.   :1001   Min.   :2020-01-01 00:00:00.00   Min.   :2020-01-04 00:00:00.00   Length:2108
 1st Qu.:3278   1st Qu.:2020-07-09 00:00:00.00   1st Qu.:2020-07-19 18:00:00.00   Class :character
 Median :5490   Median :2020-07-29 00:00:00.00   Median :2020-08-10 00:00:00.00   Mode  :character
 Mean   :5535   Mean   :2020-07-21 21:28:20.95   Mean   :2020-07-31 22:51:00.33
 3rd Qu.:7870   3rd Qu.:2020-08-19 00:00:00.00   3rd Qu.:2020-08-31 00:00:00.00
 Max.   :9996   Max.   :2020-12-31 00:00:00.00   Max.   :2021-01-09 00:00:00.00
```

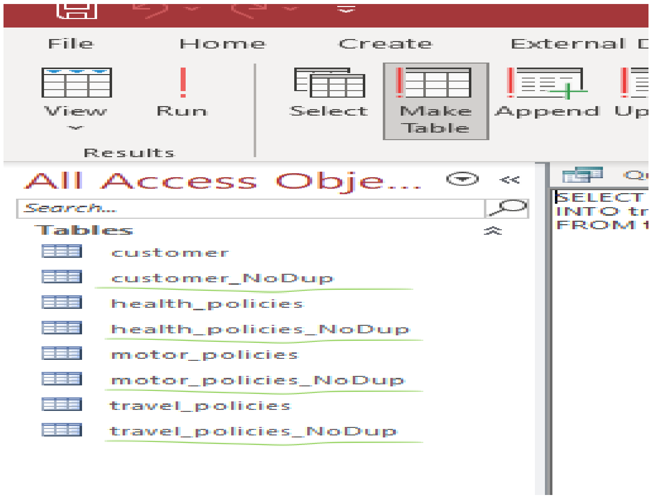


```
> summary(health_policies)
    HealthID      policyStart                      policyEnd                        HealthType
 Min.   :1001   Min.   :2019-01-01 00:00:00.00   Min.   :2020-01-01 00:00:00.00   Length:2543
 1st Qu.:3222   1st Qu.:2019-03-29 00:00:00.00   1st Qu.:2020-03-29 00:00:00.00   Class :character
 Median :5498   Median :2019-07-02 00:00:00.00   Median :2020-07-02 00:00:00.00   Mode  :character
 Mean   :5510   Mean   :2019-06-30 19:30:27.60   Mean   :2020-06-30 15:28:40.01
 3rd Qu.:7762   3rd Qu.:2019-09-28 12:00:00.00   3rd Qu.:2020-09-28 12:00:00.00
 Max.   :9999   Max.   :2019-12-31 00:00:00.00   Max.   :2020-12-31 00:00:00.00
 HealthDependentsAdults DependentsKids
 Min.   :0.0000         Min.   : 0.000
 1st Qu.:0.0000         1st Qu.: 0.000
 Median :1.0000         Median : 2.000
 Mean   :0.8164         Mean   : 1.763
 3rd Qu.:1.0000         3rd Qu.: 3.000
 Max.   :2.0000         Max.   :40.000
> summary(motor_policies)
    MotorID       PolicyStart                     PolicyEnd                        MotorType
 Min.   :1004   Min.   :2019-01-01 00:00:00.0   Min.   :2020-01-01 00:00:00.00   Length:3361
 1st Qu.:3200   1st Qu.:2019-04-06 00:00:00.0   1st Qu.:2020-04-06 00:00:00.00   Class :character
 Median :5592   Median :2019-07-04 00:00:00.0   Median :2020-07-04 00:00:00.00   Mode  :character
 Mean   :5530   Mean   :2019-07-04 14:18:36.0   Mean   :2020-07-04 10:42:39.95
 3rd Qu.:7772   3rd Qu.:2019-10-04 00:00:00.0   3rd Qu.:2020-10-04 00:00:00.00
 Max.   :9999   Max.   :2019-12-30 00:00:00.0   Max.   :2020-12-30 00:00:00.00

   veh_value         Exposure             clm             Numclaims          v_body              v_age
 Min.   : 0.000   Min.   :0.002738   Min.   :0.00000   Min.   :0.00000   Length:3361        Min.   :1.000
 1st Qu.: 1.030   1st Qu.:0.216290   1st Qu.:0.00000   1st Qu.:0.00000   Class :character   1st Qu.:2.000
 Median : 1.510   Median :0.484600   Median :0.00000   Median :0.00000   Mode  :character   Median :3.000
 Mean   : 1.811   Mean   :0.478145   Mean   :0.06367   Mean   :0.06813                      Mean   :2.659
 3rd Qu.: 2.220   3rd Qu.:0.772074   3rd Qu.:0.00000   3rd Qu.:0.00000                      3rd Qu.:4.000
 Max.   :16.690   Max.   :0.999316   Max.   :1.00000   Max.   :3.00000                      Max.   :4.000

 LastClaimDate
 Min.   :2019-02-02 00:00:00.0
 1st Qu.:2019-09-28 00:00:00.0
 Median :2019-12-25 00:00:00.0
 Mean   :2019-12-25 14:21:18.5
 3rd Qu.:2020-04-04 00:00:00.0
 Max.   :2020-12-06 00:00:00.0
 NA's   :3147
> 
```

Data observation, cleaning, and data quality issues addressed in SQL and R.

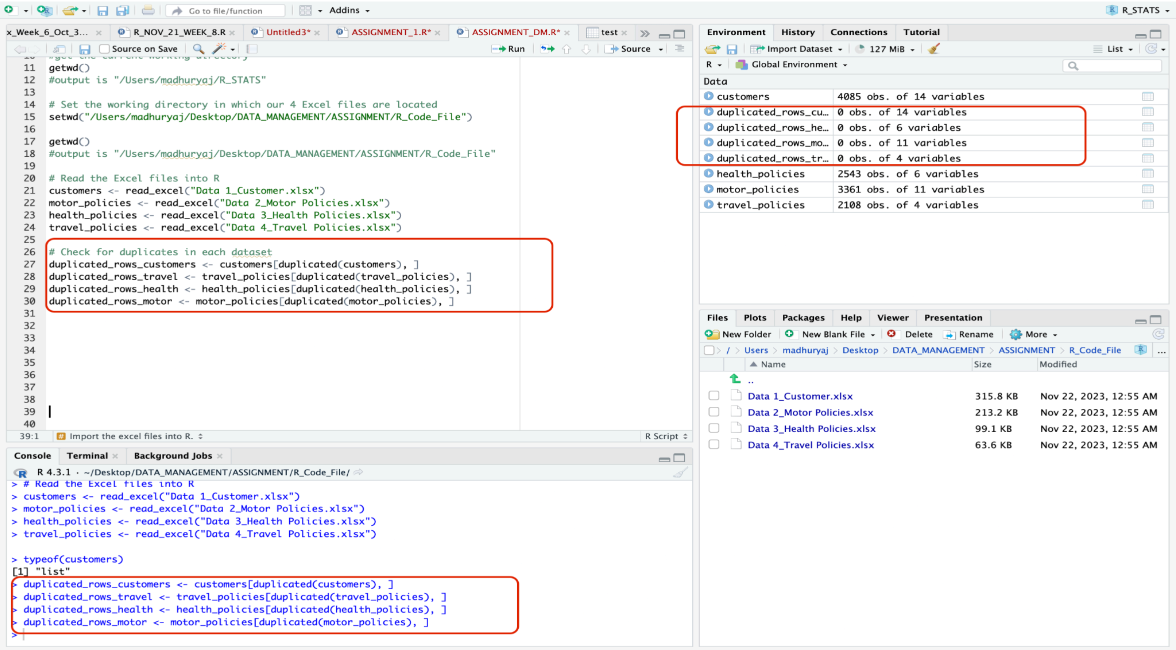
1) Duplicate records in the tables.

Tables are checked for duplicates before merging. Both validation in Code:SQL_1 and a simpler cross-check in R Code:R_3 confirm: no duplicate records exist. This ensures data integrity, free from concerns about duplicate entries.



Tables

- customer
- customer_NoDup
- health_policies
- health_policies_NoDup
- motor_policies
- motor_policies_NoDup
- travel_policies
- travel_policies_NoDup



```r
getwd()
#output is "/Users/madhuryaj/R_STATS"

# Set the working directory in which our 4 Excel files are located
setwd("/Users/madhuryaj/Desktop/DATA_MANAGEMENT/ASSIGNMENT/R_Code_File")

getwd()
#output is "/Users/madhuryaj/Desktop/DATA_MANAGEMENT/ASSIGNMENT/R_Code_File"

# Read the Excel files into R
customers <- read_excel("Data 1_Customer.xlsx")
motor_policies <- read_excel("Data 2_Motor Policies.xlsx")
health_policies <- read_excel("Data 3_Health Policies.xlsx")
travel_policies <- read_excel("Data 4_Travel Policies.xlsx")

# Check for duplicates in each dataset
duplicated_rows_customers <- customers[duplicated(customers), ]
duplicated_rows_travel <- travel_policies[duplicated(travel_policies), ]
duplicated_rows_health <- health_policies[duplicated(health_policies), ]
duplicated_rows_motor <- motor_policies[duplicated(motor_policies), ]
```

```
R 4.3.1 · ~/Desktop/DATA_MANAGEMENT/ASSIGNMENT/R_Code_File/
> # Read the Excel files into R
> customers <- read_excel("Data 1_Customer.xlsx")
> motor_policies <- read_excel("Data 2_Motor Policies.xlsx")
> health_policies <- read_excel("Data 3_Health Policies.xlsx")
> travel_policies <- read_excel("Data 4_Travel Policies.xlsx")

> typeof(customers)
[1] "list"
> duplicated_rows_customers <- customers[duplicated(customers), ]
> duplicated_rows_travel <- travel_policies[duplicated(travel_policies), ]
> duplicated_rows_health <- health_policies[duplicated(health_policies), ]
> duplicated_rows_motor <- motor_policies[duplicated(motor_policies), ]
>
```

Data

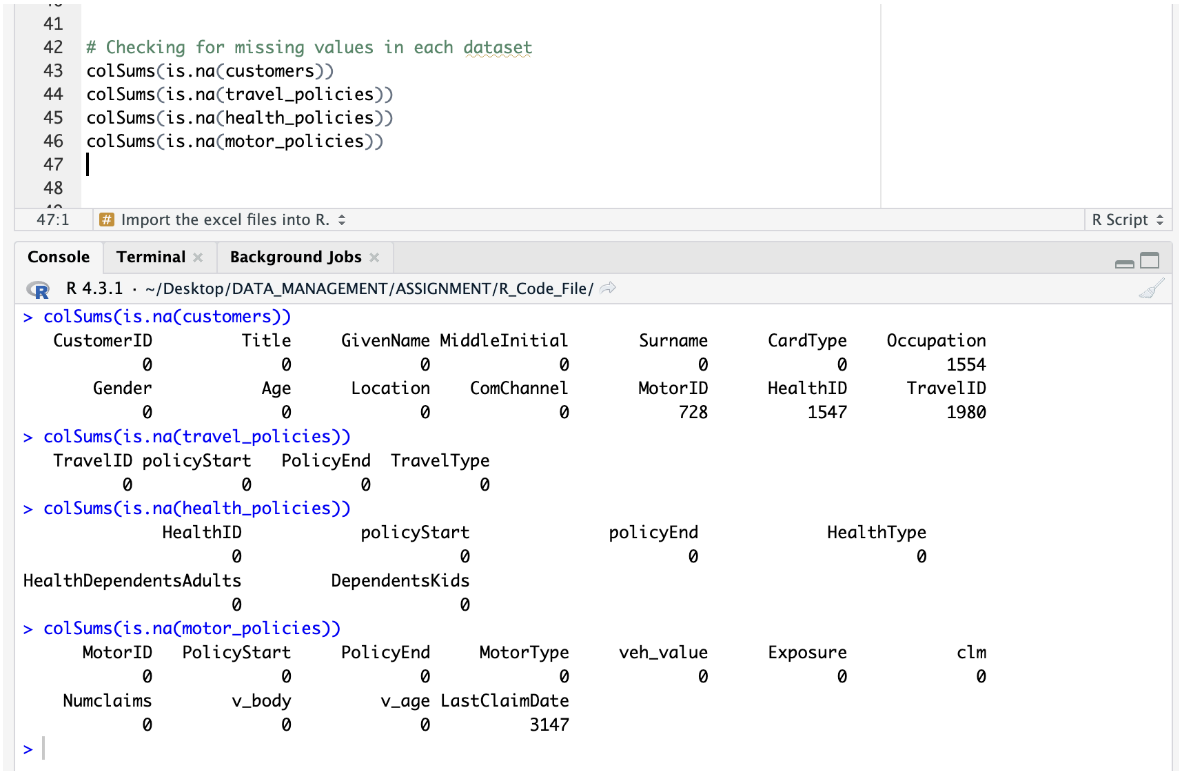
customers	4085 obs. of 14 variables
duplicated_rows_cu…	0 obs. of 14 variables
duplicated_rows_he…	0 obs. of 6 variables
duplicated_rows_mo…	0 obs. of 11 variables
duplicated_rows_tr…	0 obs. of 4 variables
health_policies	2543 obs. of 6 variables
motor_policies	3361 obs. of 11 variables
travel_policies	2108 obs. of 4 variables

Name	Size	Modified
..		
Data 1_Customer.xlsx	315.8 KB	Nov 22, 2023, 12:55 AM
Data 2_Motor Policies.xlsx	213.2 KB	Nov 22, 2023, 12:55 AM
Data 3_Health Policies.xlsx	99.1 KB	Nov 22, 2023, 12:55 AM
Data 4_Travel Policies.xlsx	63.6 KB	Nov 22, 2023, 12:55 AM

2) Missing Values

To check for missing values in datasets using R, colSums() function in conjunction with is.na(). This combination sums up occurrences of missing values in each column of dataset. This Code:R_4 calculates total count of missing values within columns across each dataset customers, travel policies, health policies, and motor policies.

The 'Occupation' column in dataset exhibits a notable number of missing values, making it impractical to drop these observations due to substantial loss of data. Given that these entries are non-numeric, applying statistical measures like mean is infeasible. The optimal solution is for organization to prioritize collecting complete customer data moving forward to mitigate these missing values in 'Occupation' column (Aljuaid & Sasi, 2016). This proactive step will enhance dataset's completeness and accuracy for future analyses.



```r

# Checking for missing values in each dataset
colSums(is.na(customers))
colSums(is.na(travel_policies))
colSums(is.na(health_policies))
colSums(is.na(motor_policies))

```

47:1
Import the excel files into R.
R Script
Console
Terminal
Background Jobs
R 4.3.1 · ~/Desktop/DATA_MANAGEMENT/ASSIGNMENT/R_Code_File/

```
> colSums(is.na(customers))
   CustomerID         Title     GivenName MiddleInitial       Surname      CardType    Occupation 
            0             0             0             0             0             0          1554 
       Gender           Age      Location    ComChannel       MotorID      HealthID      TravelID 
            0             0             0             0           728          1547          1980 
> colSums(is.na(travel_policies))
   TravelID policyStart   PolicyEnd  TravelType 
          0           0           0           0 
> colSums(is.na(health_policies))
              HealthID            policyStart              policyEnd             HealthType 
                     0                      0                      0                      0 
HealthDependentsAdults         DependentsKids 
                     0                      0 
> colSums(is.na(motor_policies))
      MotorID   PolicyStart     PolicyEnd     MotorType     veh_value      Exposure           clm 
            0             0             0             0             0             0             0 
    Numclaims        v_body         v_age LastClaimDate 
            0             0             0          3147 
> 
```

3) Inconsistency or unexpected values

The CardType column has values like ‘Visa’, ’MasterCard’ and 0 in dataset. The value 0 is not legitimate. Its better to have value as NA or unknown instead of wrong data. Refer Code:R_6
The 'Card Type' holds significant importance; it's essential for companies to ensure accurate data collection for this parameter in future.

4) Similar entries that can be grouped

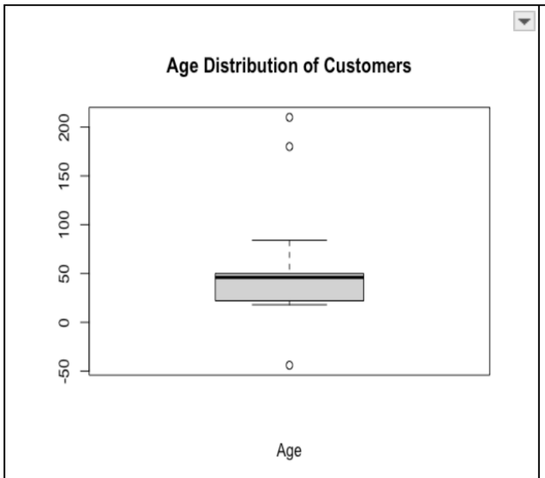
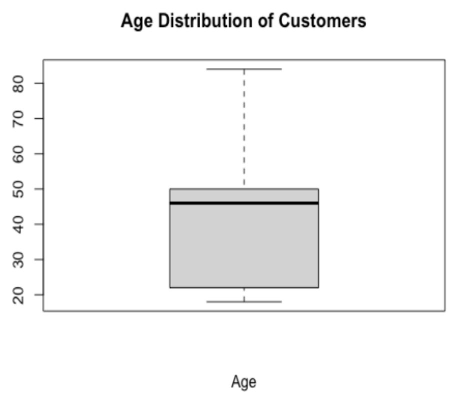
The categorization of marketing communication channel entries involves grouping similar items based on their abbreviations. Specifically, ('Email','E') are considered equivalent; ('SMS','S') are regarded same, while ('Phone','P') are grouped together. This issue is handled in SQL Code:SQL_5 through UPDATE operation.

Another column is the gender, in which (‘female’,’f’) and (‘male’,’m’) are to be synchronized via the UPDATE operation in SQL as per Code:SQL_3.

This is required, so standardizing values which aids analysis, informs tailored marketing strategies based on customer choices.

5) Outliers

The boxplot Code:R_5 serves as a tool to identify outliers within customers age distribution. Since age cannot be negative, absolute value as per Code:SQL_4 is used in those instances (Vinisha & Sujihelen, 2022). For ages exceeding 100 years, we'll replace them with dataset's mean age, approximately 42 years, following logic described in Code:SQL_2.

	
Outliers (DQ Issues)	No Outliers (DQ Issues resolved)

Once these data quality issues are resolved, we can proceed to form unified table known as ABT.

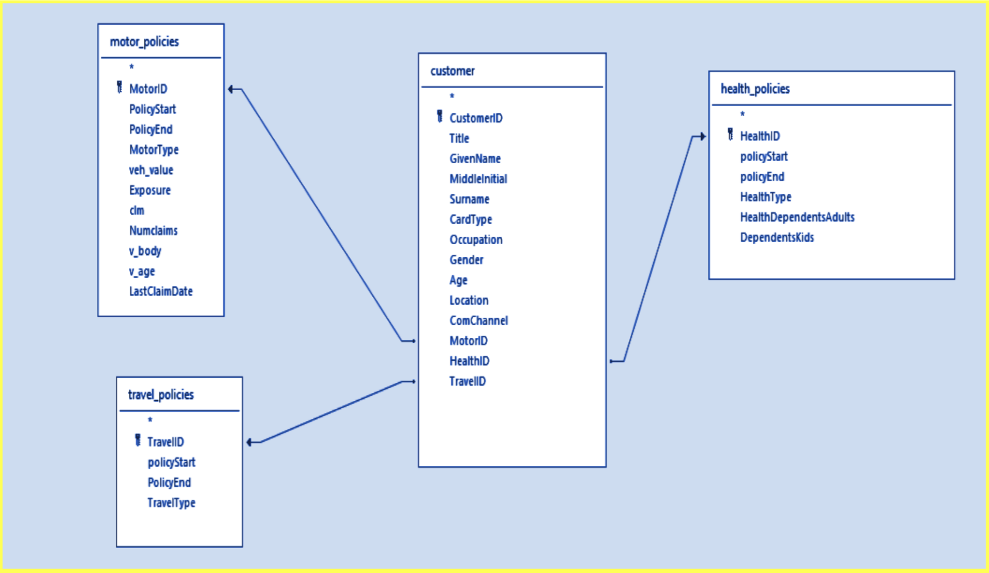
Analytical Base Table Formation

Analytical Base Table (ABT) serves as foundational step for conducting advanced analytics and modeling by presenting a consolidated dataset.

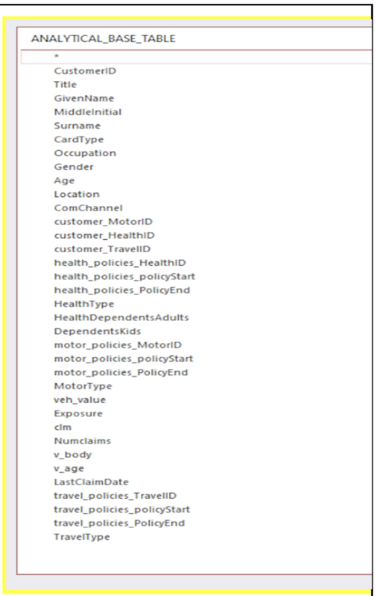
In this context, primary emphasis is on customer table. The other three tables, travel_policies, motor_policies, and health_policies, are connected via left join to customer table through foreign keys TravelID, MotorID, and HealthID, respectively, as illustrated in data diagram. This setup enables thorough analysis and extraction of insights for crafting marketing strategies for customers who bought atleast one of insurance product also.

In MS Access - SQL

In SQL's Data Manipulation Language (DML), left join merges rows from two tables based on a shared column, keeping all rows from left table and matching rows from right table. It's beneficial in constructing ABT, enabling retrieval of data from left table while accommodating unmatched data from right table. Refer Code:SQL_6



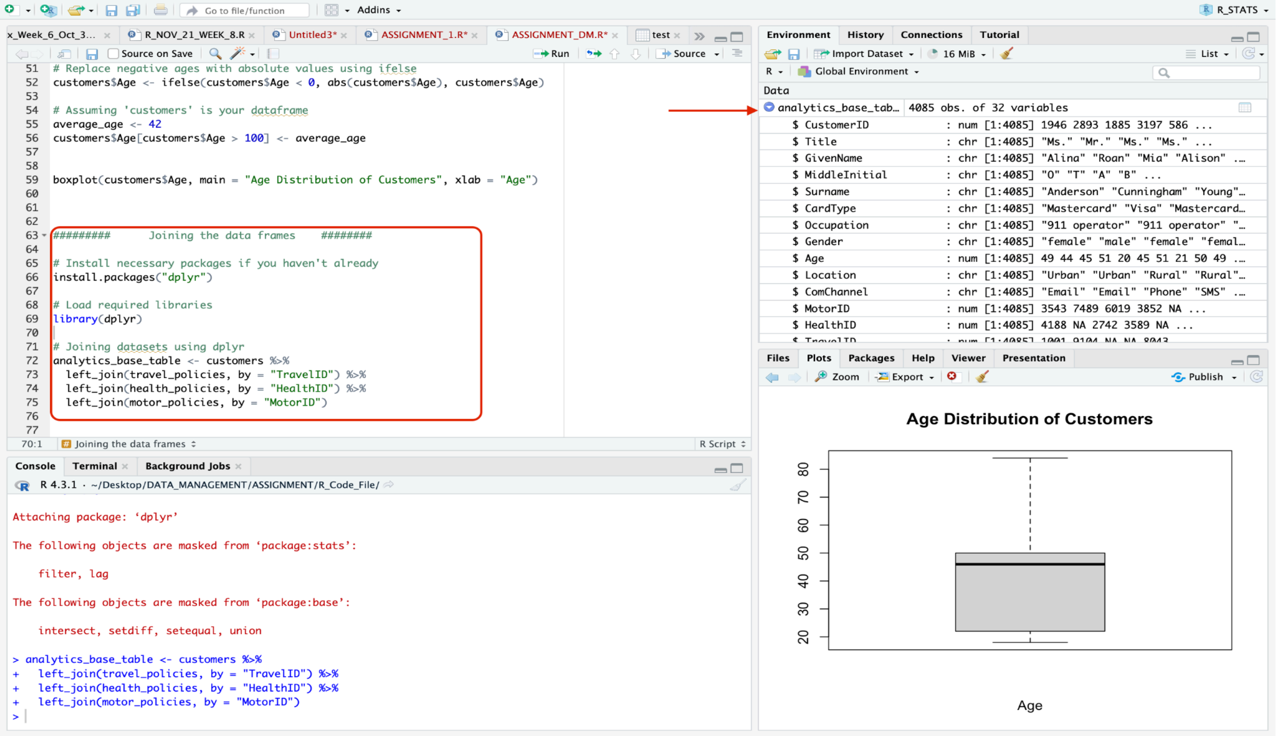
Individual tables joined via the primary and foreign keys


ANALYTICAL_BASE_TABLE
*
CustomerID
Title
GivenName
MiddleInitial
Surname
CardType
Occupation
Gender
Age
Location
ComChannel
customer_MotorID
customer_HealthID
customer_TravelID
health_policies_HealthID
health_policies_policyStart
health_policies_PolicyEnd
HealthType
HealthDependentsAdults
DependentsKids
motor_policies_MotorID
motor_policies_policyStart
motor_policies_PolicyEnd
MotorType
veh_value
Exposure
clm
Numclaims
v_body
v_age
LastClaimDate
travel_policies_TravelID
travel_policies_policyStart
travel_policies_PolicyEnd
TravelType

Analytical Base Table

In R Studio

In R, it is simpler to join these tables, which can be done using dplyr library. The dplyr package provides consistent, user-friendly syntax for data manipulation tasks, making it easier to perform complex operations like joins, filtering, summarizing, and more on your data. Refer Code:R_7



```r
# Replace negative ages with absolute values using ifelse
customers$Age <- ifelse(customers$Age < 0, abs(customers$Age), customers$Age)

# Assuming 'customers' is your dataframe
average_age <- 42
customers$Age[customers$Age > 100] <- average_age


boxplot(customers$Age, main = "Age Distribution of Customers", xlab = "Age")



#########       Joining the data frames      ########

# Install necessary packages if you haven't already
install.packages("dplyr")

# Load required libraries
library(dplyr)

# Joining datasets using dplyr
analytics_base_table <- customers %>%
  left_join(travel_policies, by = "TravelID") %>%
  left_join(health_policies, by = "HealthID") %>%
  left_join(motor_policies, by = "MotorID")
```

```
Data
analytics_base_tab…  4085 obs. of 32 variables
   $ CustomerID       : num [1:4085] 1946 2893 1885 3197 586 ...
   $ Title            : chr [1:4085] "Ms." "Mr." "Ms." "Ms." ...
   $ GivenName        : chr [1:4085] "Alina" "Roan" "Mia" "Alison" ..…
   $ MiddleInitial    : chr [1:4085] "O" "T" "A" "B" ...
   $ Surname          : chr [1:4085] "Anderson" "Cunningham" "Young"…
   $ CardType         : chr [1:4085] "Mastercard" "Visa" "Mastercard…
   $ Occupation       : chr [1:4085] "911 operator" "911 operator" "…
   $ Gender           : chr [1:4085] "female" "male" "female" "femal…
   $ Age              : num [1:4085] 49 44 45 51 20 45 51 21 50 49 ..…
   $ Location         : chr [1:4085] "Urban" "Urban" "Rural" "Rural"…
   $ ComChannel       : chr [1:4085] "Email" "Email" "Phone" "SMS" ..…
   $ MotorID          : num [1:4085] 3543 7489 6019 3852 NA ...
   $ HealthID         : num [1:4085] 4188 NA 2742 3589 NA ...
   $ TravelID         : num [1:4085] 1001 9104 NA NA 8043
```

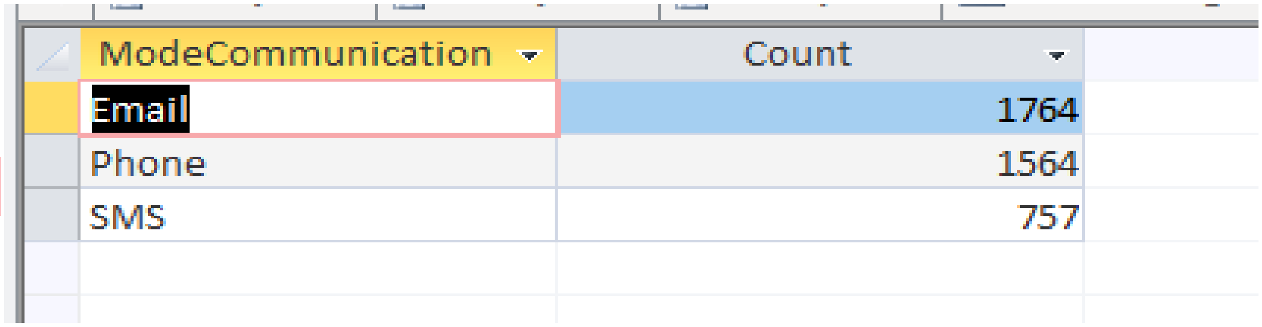
```
R 4.3.1 · ~/Desktop/DATA_MANAGEMENT/ASSIGNMENT/R_Code_File/

Attaching package: ‘dplyr’

The following objects are masked from ‘package:stats’:

    filter, lag

The following objects are masked from ‘package:base’:

    intersect, setdiff, setequal, union

> analytics_base_table <- customers %>%
+   left_join(travel_policies, by = "TravelID") %>%
+   left_join(health_policies, by = "HealthID") %>%
+   left_join(motor_policies, by = "MotorID")
> 
```

Insights Report

Insights from ABT empower business strategies and data-based choices, driving business growth through targeted marketing, identifying top-customers, and spotting market trends.

Insight 1: Communication Mode Preferred by customers

This query, Code:SQL_7 retrieves result and orders results by count of occurrences of each communication channel in descending order.



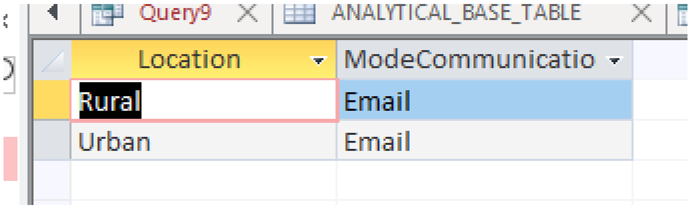
ModeCommunication	Count
Email	1764
Phone	1564
SMS	757

The majority of customers, approximately 44% of total, displayed preference for email communications, closely trailed by phone communication, which constituted around 38%. Interestingly, least favored communication channel appeared to be SMS, accounting for approximately 18% of total customer base. This inclination might suggest perceived formality associated with email as communication medium, potentially influencing customers' preferences over other channels available (Sabbagh, 2021).

Marketing Strategy: Company can focus on email communications as primary method to entice customers into purchasing insurance policies (Sabbagh, 2021).

Insight 2: Top communication channel preference by location

From Code:SQL_8 its evident that both rural and urban locations indicate common trend that customers across these areas predominantly favor email communication over other channels. This suggests that individuals in both regions had access to emails and actively anticipated official communications through this medium (Lee *et al*., 2010).

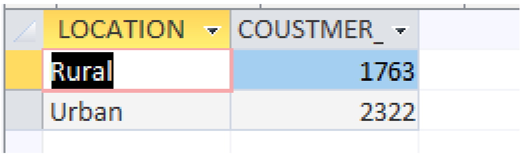


Location	ModeCommunicatio
Rural	Email
Urban	Email

Marketing Strategy: Company can prioritize using email communications as main approach to persuade customers to buy insurance policies in both areas (Sabbagh, 2021).

Insight 3: Customers across locations

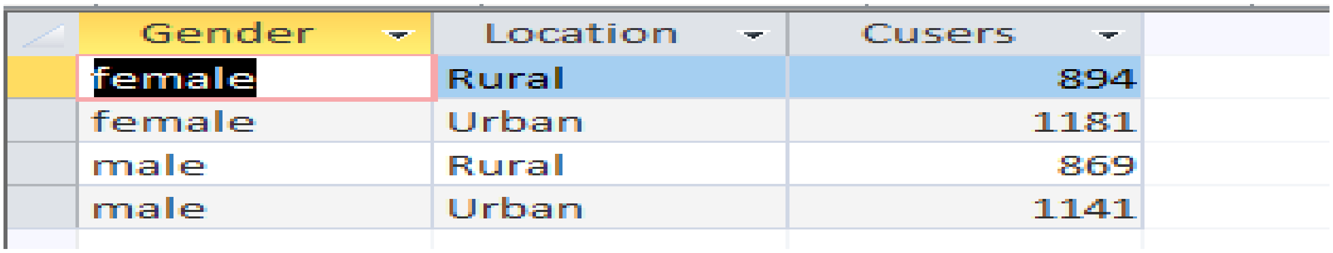
From query Code:SQL_9 - Data categorizes customers into Rural and Urban areas, revealing trend where individuals from urban areas showed greater inclination towards purchasing insurance compared to those in rural areas. This inclination might stem from higher education levels in urban areas, where people understand significance of insurance and potentially have more exposure to insurance-related promotions (Lee *et al*., 2010).



LOCATION	COUSTMER_
Rural	1763
Urban	2322

Marketing Strategy - To bridge this gap, company could strategize by focusing on educating individuals in rural areas about importance of insurance. This could serve as an opportunity to attract and assist them in enrolling in insurance policies (Cai & Deng, 2010).

Insight 4: Relationship between number of insurance buyers with respect to gender and location



Gender	Location	Cusers
female	Rural	894
female	Urban	1181
male	Rural	869
male	Urban	1141

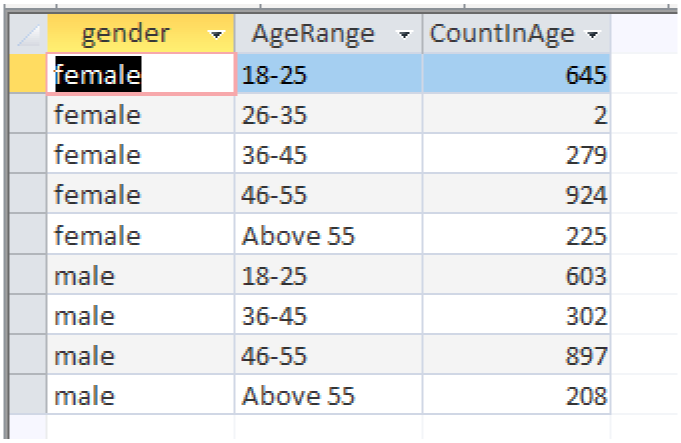
From query Code:SQL_10 - In both urban and rural settings, there was remarkably balanced distribution between male and female individuals purchasing insurance. This equitable representation across both locations indicates company's concerted effort to attract customers regardless of gender or geographical area, maintaining an approximate 50% count for both male and female customers.

Marketing Strategy

Company can maintain its current successful strategy in this area, as it's yielding positive results.

Insight 5: Age vs count of insurance holders

From query Code:SQL_11, when age is categorized into groups (18-25, 26-35, 36-45, 46-55, and above 55 years), notable observation is low insurance uptake among female customers aged 26 to 35, with only 2 individuals availing of insurance. This presents significant risk considering this age range typically signifies stable phase in person's life, often marked by substantial commitments.



gender	AgeRange	CountInAge
female	18-25	645
female	26-35	2
female	36-45	279
female	46-55	924
female	Above 55	225
male	18-25	603
male	36-45	302
male	46-55	897
male	Above 55	208

Marketing Strategy

To mitigate this gap, company should investigate underlying reasons and address them. One potential suggestion, as outlined by Li (2022), could involve encouraging individuals within this age group, particularly those employed, to consider insurance options provided by their employers, such as medical insurance. This approach may help bridge gap by emphasizing benefits of employer-offered insurance plans tailored to this demographic.

Insight 6: Customers who took all three insurance

From query Code:SQL_12, Out of 4085 customers, roughly 24% or around a quarter, equivalent to 975 people, opted for all three insurance policies. This is common, as not everyone needs motor insurance if they don't own vehicles, or travel insurance if they don't travel much. But surprisingly, there's notably low adoption of health insurance, which might be less than recommended based on records as per research by Lee and Lee (2020) .

Marketing Strategy

Health insurance is critical because unexpected events can have significant consequences. It's vital for customers to understand its importance. According to Lee and Lee (2020) company should collaborate with medical experts to raise awareness about health insurance. This approach aims to improve customer understanding and promote its adoption.

Embracing Technology's Potential: Overcoming Limitations with Positive Strategies

R surpasses MS Access in advanced statistical analysis, boasting wide array of tools, packages for complex modeling, hypothesis testing, data visualization (Zhou & Ordonez, 2021). Its adaptability allows tailored analysis pipelines and vivid graphical representations. While MS Access shines in user-friendly database management—like table creation, querying, and basic app development—it lacks robust statistical depth of R, making R top choice for in-depth analysis, visualization.

The collaboration of R, Python, SQL, and cloud-based data storage has revolutionized data analysis (Chen *et al*., 2023). Utilizing platforms like Amazon S3 and Google Cloud Storage, these technologies offer secure, scalable storage seamlessly integrated with R, Python, and SQL. This fusion optimizes handling extensive datasets through cloud computing, unifying data sources and leveraging the strengths of each language. It combines Python's machine learning with R's statistics, enabling complex real-time analysis of streaming data. This integration enhances scalability, simplifies analysis, and extracts comprehensive insights from cloud-stored data (Chen *et al.*, 2023).

References

Aljuaid, T. and Sasi, S. (2016) ‘Proper imputation techniques for missing values in data sets’, *2016 International Conference on Data Science and Engineering (ICDSE)* [Preprint]. doi:10.1109/icdse.2016.7823957.

Cai, X. and Deng, D. song (2010) ‘Some questions about the new rural endowment insurance’, *2010 International Conference on System Science, Engineering Design and Manufacturing Informatization* [Preprint]. doi:10.1109/icsem.2010.42.

Chen, W. *et al.* (2023) ‘Real-time analytics: Concepts, architectures, and ML/AI considerations’, *IEEE Access*, 11, pp. 71634–71657. doi:10.1109/access.2023.3295694.

Lee, S.-J., Kwon, S.I. and Chung, S.Y. (2010) ‘Determinants of household demand for insurance: The case of Korea’, *The Geneva Papers on Risk and Insurance - Issues and Practice*, 35(S1). doi:10.1057/gpp.2010.29.

Lee, E.K. and Lee, J. (2020) ‘Competition strategy for healthcare insurance plans’, *2020 IEEE International Conference on Bioinformatics and Biomedicine (BIBM)* [Preprint]. doi:10.1109/bibm49941.2020.9313177.

Li, D. *et al.* (2022) ‘Exploring the intention of middle-aged and elderly consumers to participate in inclusive medical insurance’, *IEEE Access*, 10, pp. 71398–71413. doi:10.1109/access.2022.3187711.

McFadyen, R. (2016) *Relational databases and Microsoft Access* \. Minneapolis, MN: Open Textbook Library.

Sabbagh, F. (2021) ‘Email marketing: The most important advantages and disadvantages’, *Business, Management and Economics Research*, (71), pp. 1–8. doi:10.32861/bmer.71.1.8.

Saltz, J.S. (2021) ‘CRISP-DM for data science: Strengths, weaknesses and potential next steps’, *2021 IEEE International Conference on Big Data (Big Data)* [Preprint]. doi:10.1109/bigdata52589.2021.9671634.

Vinisha, F.A. and Sujihelen, L. (2022) ‘Study on missing values and outlier detection in concurrence with data quality enhancement for efficient data processing’, *2022 4th International Conference on Smart Systems and Inventive Technology (ICSSIT)* [Preprint]. doi:10.1109/icssit53264.2022.9716355.

Wickham, H., Çetinkaya-Rundel, M. and Grolemund, G. (2023) *R for Data Science: Import, Tidy, transform, visualize, and model data*. Beijing: O’Reilly.

Zhou, X. and Ordonez, C. (2021) ‘Programming languages in data science: A comparison from a database angle’, *2021 IEEE International Conference on Big Data (Big Data)* [Preprint]. doi:10.1109/bigdata52589.2021.9672007.

Appendix

The code snippets have been named SQL_1, SQL_2,..., and R_1, R_2 … for reference within the documentation space.

SQL Code

SQL_1
Comment - To check for duplicates

```
a)SELECT DISTINCT * INTO customer_NoDup FROM customer;
b)SELECT DISTINCT * INTO health_policies_NoDup FROM health_policies;
c)SELECT DISTINCT * INTO motor_policies_NoDup FROM motor_policies;
d)SELECT DISTINCT * INTO travel_policies_NoDup FROM travel_policies;
```

SQL_2
Comment - Get the average age of customers in ABT

```
SELECT AVG(ANALYTICAL_BASE_TABLE.[AGE])
FROM ANALYTICAL_BASE_TABLE;
```

Output is ~ 42 years.

SQL_3
Comment - DQ Issues addressed in SQL - Gender

```
a)UPDATE ANALYTICAL_BASE_TABLE
SET ANALYTICAL_BASE_TABLE.[GENDER] = 'female'
WHERE ANALYTICAL_BASE_TABLE.[GENDER] = 'f';

b)UPDATE ANALYTICAL_BASE_TABLE
SET ANALYTICAL_BASE_TABLE.[GENDER] = 'male'
WHERE ANALYTICAL_BASE_TABLE.[GENDER] = 'm';
```

SQL_4

Comment - DQ Issues addressed in SQL - Age

Age over 100 YEARS

```
UPDATE ANALYTICAL_BASE_TABLE
SET ANALYTICAL_BASE_TABLE.[AGE]=42
WHERE ANALYTICAL_BASE_TABLE.[AGE]>100;
```

Negative Age

```
UPDATE ANALYTICAL_BASE_TABLE
SET AGE = ABS(AGE)
WHERE AGE<0;
```

SQL_5

Comment - DQ Issues addressed in SQL - Communication Channel

SMS

```
UPDATE ANALYTICAL_BASE_TABLE
SET ANALYTICAL_BASE_TABLE.[ComChannel] = "SMS"
WHERE ANALYTICAL_BASE_TABLE.[ComChannel] = "S";
```

PHONE

```
UPDATE ANALYTICAL_BASE_TABLE
SET ANALYTICAL_BASE_TABLE.[ComChannel] = "Phone"
WHERE ANALYTICAL_BASE_TABLE.[ComChannel] = "P";
```

EMAIL

```
UPDATE ANALYTICAL_BASE_TABLE
SET ANALYTICAL_BASE_TABLE.[ComChannel] = "Email"
WHERE ANALYTICAL_BASE_TABLE.[ComChannel] = "E";
```

SQL_6

Comment - Analytical Base Table formation

```
SELECT * INTO ANALYTICAL_BASE_TABLE
FROM ((customer
LEFT JOIN health_policies ON customer.HealthID = health_policies.HealthID)
LEFT JOIN motor_policies ON customer.MotorID = motor_policies.MotorID)
LEFT JOIN travel_policies ON customer.TravelID = travel_policies.TravelID;
```

SQL_7

Comment - Insights - Communication Mode Preferred

```
SELECT
ComChannel AS ModeCommunication , COUNT(*) AS Count
FROM
ANALYTICAL_BASE_TABLE
GROUP BY ComChannel
ORDER BY COUNT(*) DESC;
```

SQL_8

Comment - Insights - Top Communication Channel Preference by Location

```
SELECT
Location, First(ComChannel) AS ModeCommunication
FROM (
   SELECT Location, ComChannel, COUNT(*) AS ChannelCount
   FROM ANALYTICAL_BASE_TABLE
   GROUP BY Location, ComChannel
   ORDER BY Location, COUNT(*) DESC
) AS SubQuery
GROUP BY Location;
```

SQL_9

Comment - Insights Customer count Location wise

```
SELECT
LOCATION,COUNT(CUSTOMERID) AS COUSTMER_COUNT
FROM
ANALYTICAL_BASE_TABLE GROUP BY LOCATION;
```

SQL_10

Comment - Insights Relationship between number of insurance buyers with respect to gender and location

```
SELECT
Gender, Location, COUNT(*) AS Count
FROM ANALYTICAL_BASE_TABLE
GROUP BY Gender, Location;
```

SQL_11

Comment - Insights Age vs count of insurance holders

```
SELECT gender,
    IIf(Age BETWEEN 18 AND 25, '18-25',
        IIf(Age BETWEEN 26 AND 35, '26-35',
            IIf(Age BETWEEN 36 AND 45, '36-45',
                IIf(Age BETWEEN 46 AND 55, '46-55', 'Above 55')
            )
        )
    ) AS AgeRange,
    COUNT(*) AS CountInAgeRange
FROM
    ANALYTICAL_BASE_TABLE
GROUP BY
 gender,
    IIf(Age BETWEEN 18 AND 25, '18-25',
        IIf(Age BETWEEN 26 AND 35, '26-35',
            IIf(Age BETWEEN 36 AND 45, '36-45',
                IIf(Age BETWEEN 46 AND 55, '46-55', 'Above 55'),
            )
        )
    );
```

SQL_12 - Insights Customers who took all three insurance

```
SELECT CustomerID
FROM Analytical_Base_Table
WHERE HealthID IS NOT NULL AND MotorID IS NOT NULL AND TravelID IS
NOT NULL;
```

SQL_13

```
SELECT COUNT(*) as ‘Number_of_Customers’
FROM Customers;
```

R Code

R_1

```
########   START - Import the excel files into R . ########

# Install necessary packages
install.packages("readxl")

# Load required libraries
library(readxl)

#get the current working directory
getwd()
#output is "/Users/madhuryaj/R_STATS"

# Set the working directory in which our 4 Excel files are located
setwd("/Users/madhuryaj/Desktop/DATA_MANAGEMENT/ASSIGNMENT/R_Code_File")

getwd()
#output
"/Users/madhuryaj/Desktop/DATA_MANAGEMENT/ASSIGNMENT/R_Code_File"

# Read the Excel files into R
customers <- read_excel("Data 1_Customer.xlsx")
motor_policies <- read_excel("Data 2_Motor Policies.xlsx")
health_policies <- read_excel("Data 3_Health Policies.xlsx")
travel_policies <- read_excel("Data 4_Travel Policies.xlsx")

########   END - Import the excel files into R . ########
```

R_2

```
# Explore  Summary Statistics for numeric columns
```

```
summary(customers)
summary(travel_policies)
summary(health_policies)
summary(motor_policies)
```

R_3

```
# Check for duplicates in each dataset - Checking is any Data Quality Issues
duplicated_rows_customers <- customers[duplicated(customers), ]
duplicated_rows_travel <- travel_policies[duplicated(travel_policies), ]
duplicated_rows_health <- health_policies[duplicated(health_policies), ]
duplicated_rows_motor <- motor_policies[duplicated(motor_policies), ]
```

R_4

```
# Checking for missing values in each dataset - Checking if any Data Quality Issues
colSums(is.na(customers))
colSums(is.na(travel_policies))
colSums(is.na(health_policies))
colSums(is.na(motor_policies))
```

R_5

```
###### START - Addressing a Data Quality issue in Age column #####

# Removing the Outliers in the data
boxplot(customers$Age, main = "Age Distribution of Customers", xlab = "Age")

# Replace negative ages with absolute values using ifelse
customers$Age<-ifelse(customers$Age< 0, abs(customers$Age), customers$Age)

# Assuming 'customers' is your dataframe
average_age <- 42
customers$Age[customers$Age > 100] <- average_age

# Visualizating the Age data after removing outliers
boxplot(customers$Age, main = "Age Distribution of Customers", xlab = "Age")

###### END - Addressing a Data Quality issue in Age column #####
```

R_6

```
# Replace '0' with 'Unknown' in 'CardType' column to address Data Quality Issue
customers$CardType [customers$CardType == '0'] <- 'Unknown'
```

R_7

```
#########  START - Joining the data frames  to form ABT  ########

# Install necessary packages if you haven't already
install.packages("dplyr")

# Load required libraries
library(dplyr)

# Joining datasets using dplyr
analytics_base_table <- customers %>%
  left_join(travel_policies, by = "TravelID") %>%
  left_join(health_policies, by = "HealthID") %>%
  left_join(motor_policies, by = "MotorID")

#########  END - Joining the data frames  to form ABT  ########
```

Others

ABT - Analytical Base Table