Project Coversheet

Full Name	Shifana Kaleel Rahiman
Email	shifanakaleelrahiman@gmail.com
Contact Number	+4915145130613
Date of Submission	21.07.2025
Project Week	Week 1

Project Guidelines and Rules

1. Submission Format

- **Document Style**:
  - Use a clean, readable font such as *Arial* or *Times New Roman*, size 12.
  - Set line spacing to **1.5** for readability.
- **File Naming**:
  - Use the following naming format:
    Week X – [Project Title] – [Your Full Name Used During Registration]
    Example: Week 1 – Customer Sign-Up Behaviour – Mark Robb
- **File Types**:
  - Submit your report as a **PDF**.
  - If your project includes code or analysis, attach the **.ipynb notebook** as well.

2. Writing Requirements

- Use formal, professional language.
- Structure your content using headings, bullet points, or numbered lists.

3. Content Expectations

- Answer **all** parts of each question or task.

- Reference tools, frameworks, or ideas covered in the programme and case studies.
- Support your points with practical or real-world examples where relevant.
- Go beyond surface-level responses. Analyse problems, evaluate solutions, and demonstrate depth of understanding.

4. Academic Integrity & Referencing

- All submissions must be your own. Plagiarism is strictly prohibited.
- If you refer to any external materials (e.g., articles, studies, books), cite them using a consistent referencing style such as APA or MLA.
- Include a references section at the end where necessary.

5. Evaluation Criteria

Your work will be evaluated on the following:

- Clarity: Are your answers well-organised and easy to understand?
- Completeness: Have you answered all parts of the task?
- Creativity: Have you demonstrated original thinking and thoughtful examples?
- Application: Have you effectively used programme concepts and tools?
- Professionalism: Is your presentation, language, and formatting appropriate?

6. Deadlines and Extensions

- Submit your work by the stated deadline.
- If you are unable to meet a deadline due to genuine circumstances (e.g., illness or emergency), request an extension **before the deadline** by emailing: **support@uptrail.co.uk**
  Include your full name, week number, and reason for extension.

7. Technical Support

- If you face technical issues with submission or file access, contact our support team promptly at **support@uptrail.co.uk.**

8. Completion and Certification

- Certificate of Completion will be awarded to participants who submit at least two projects.
- Certificate of Excellence will be awarded to those who:
  - Submit all four weekly projects, and
  - Meet the required standard and quality in each.
- If any project does not meet expectations, you may be asked to revise and resubmit it before receiving your certificate.

YOU CAN START YOUR PROJECT FROM HERE

1. Introduction

1.1 Project Overview

This project focuses on auditing and analyzing customer sign-up data to support Rapid Scale's Monthly Business Review. The goal is to identify data quality issues and uncover trends in user acquisition, demographics, and marketing engagement to help improve marketing and onboarding strategies.

1.2 Dataset Description

The analysis is based on the customer_signups.csv file, which contains customer-level sign-up information.

Rows: 300, **Columns:** 10

Each row represents a new user and includes the following key fields:

- **customer_id**: Unique identifier for each user
- **name, email**: User contact information
- **signup_date**: Date of account creation
- **source**: User acquisition channel (Google,Instagram,Facebook,LinkedIn,Youtube, Referral)
- **region**: User's geographic region
- **plan_selected**: Subscription tier chosen (e.g., Basic, Pro, Premium – with potential inconsistencies)
- **marketing_opt_in**: Indicates if user opted in to marketing (Yes/No)
- **age, gender**: Demographic information (may contain inconsistent or missing values)

2. Data Cleaning Summary

To prepare the `customer_signup` dataset for analysis, several data cleaning steps were carried out to ensure reliability and consistency. These steps were essential for drawing accurate insights from the customer sign-up data.

2.1 Standardization

- **Date format**:

  Initially the datatypes of all given columns as object

```
df.dtypes          #Displays the datatypes of each column
```

```
customer_id         object
name                object
email               object
signup_date         object
source              object
region              object
plan_selected       object
marketing_opt_in    object
age                 object
gender              object
dtype: object
```

  The `signup_date` column was converted to a consistent `datetime` format to allow easy filtering by month and analysis over time.

1.3 - Convert signup_date to datetime

```
df['signup_date'] = pd.to_datetime(df['signup_date'], dayfirst=True ,errors = 'coerce')
df['signup_date']
```

```
0           NaT
1    2024-01-02
2    2024-01-03
3    2024-01-04
4    2024-01-05
        ...
295  2024-10-22
296  2024-10-23
297  2024-10-24
298  2024-10-25
299  2024-10-26
Name: signup_date, Length: 300, dtype: datetime64[ns]
```

Converting marketing_opt_in as boolean values and age as Int64 and all other columns to pythonstring data type.

```
df.dtypes
```

```
customer_id          string[python]
name                 string[python]
email                string[python]
signup_date          datetime64[ns]
source               string[python]
region               string[python]
plan_selected        string[python]
marketing_opt_in            boolean
age                           Int64
gender               string[python]
dtype: object
```

- **Customer_id field:** Identified the null values using .isnull() and summarised using .sum(). Then, fill the null values using index_to_series() and lambda function.
- **Lowercase**: Standardizing text data in source, region, plan_selected, gender columns by converting all values to lowercase.

- **Source**: Replaced `'??'` in the `source` column with `NaN` to mark missing values.
- **Plan_selected**: Corrected a typo in `plan_selected`, replacing `'prem'` with `'premium'`.
- **Gender**: Standardized entries in the `gender` column by replacing `'non-binary'`, `'123'`, and `'other'` with `'others'`.
- **Marketing_opt_in**: Converted `marketing_opt_in` values from `'Yes'`, `'No'`, and `'Nil'` to boolean (`True`/`False`).Also,changed the `marketing_opt_in` column data type to `boolean` for consistency.

2.2 Removal of Duplicates

1.5 Remove duplicate rows based on customer_id

```
df.duplicated().sum()
```

```
0
```

Used df.duplicated().sum() and returns 0, it means there are no duplicate rows in the dataset.

2.3 Handling Missing Data

```
df[['region', 'email', 'age']] = df[['region', 'email', 'age']].replace(r'^\s*$',np.nan, regex=True)
df[['region', 'email', 'age']]
```

```
#Replace all 'nan' to null values

df['source'] = df['source'].replace('nan',np.nan)
df['region'] = df['region'].replace('nan',np.nan)
df['plan_selected'] = df['plan_selected'].replace('nan',np.nan)
df['marketing_opt_in'] = df['marketing_opt_in'].replace('nan',np.nan)
df['age'] = df['age'].replace('nan',np.nan)
df['gender'] = df['gender'].replace('nan',np.nan)
```

- Replace empty strings or whitespace-only strings in `region`, `email`, and `age` columns with `NaN`.

- Replace string `'nan'` with actual `np.nan` in `source`, `region`, `plan_selected`, `marketing_opt_in`, `age`, and `gender`columns.

These steps ensured that the dataset accurately reflects user sign-ups by source and time, allowing for trustworthy insights in the later stages of the project

3.Key Findings & Trends

Here are some key insights:

- **Consistent Weekly Sign-ups:** Sign-ups per week are fairly steady, mostly around 6-7 customers weekly, indicating stable customer acquisition over time.

```
## Sign-ups per week (grouped by signup_date)

weekly_count = df.groupby(df['signup_date'].dt.strftime('%Y-W%V'))['customer_id'].count()
weekly_count
```

```
signup_date
2024-W01    6
2024-W02    7
2024-W03    7
2024-W04    7
2024-W05    8
2024-W06    7
2024-W07    7
2024-W08    7
2024-W09    7
2024-W10    7
2024-W11    7
2024-W12    6
2024-W13    6
2024-W14    7
2024-W15    7
2024-W16    7
2024-W17    7
2024-W18    6
2024-W19    7
2024-W20    7
2024-W21    7
2024-W22    7
2024-W23    6
2024-W24    7
2024-W25    7
2024-W26    7
2024-W27    7
2024-W28    7
2024-W29    6
2024-W30    7
2024-W31    7
2024-W32    6
2024-W33    7
2024-W34    7
2024-W35    7
2024-W36    7
2024-W37    7
2024-W38    7
2024-W39    7
2024-W40    7
2024-W41    7
2024-W42    7
2024-W43    6
Name: customer_id, dtype: int64
```

- **Source and Plan Preferences:** Facebook and Google are the main sources driving sign-ups, with a diverse range of plans chosen, but premium and pro plans are notably popular across regions.

```
## Sign-ups by source, region, and plan_selected

df.groupby(['source','region','plan_selected'])['signup_date'].count().head(20)
```

```
source    region   plan_selected
facebook  central  premium          4
                   pro              3
          east     basic            3
                   premium          3
                   pro              4
          north    basic            1
                   premium          2
                   pro              4
          south    basic            1
                   premium          4
                   pro              1
          west     premium          1
                   pro              1
google    central  basic            2
                   premium          2
                   pro              6
          east     basic            1
                   premium          4
                   pro              1
                   unknownplan      1
Name: signup_date, dtype: int64
```

- **Age Data Anomalies:** The maximum age value of 206 is unusually high, suggesting potential data entry errors or outliers needing review.

```
#Age summary: min, max, mean, median, null count

print("Min age of customers is ",df['age'].min())
print("Max age of customers is ",df['age'].max())
print("Mean of age customers is ",df['age'].mean())
print("Median of age of customers is ",df['age'].median())
print("Count of nulls in age is ",df['age'].isna().sum())
```

```
Min age of customers is  21
Max age of customers is  206
Mean of age customers is  36.11347517730496
Median of age of customers is  34.0
Count of nulls in age is  18
```

- **Marketing Opt-In Distribution:** Marketing opt-in counts are relatively balanced across genders, with 'others' category having slightly higher counts, reflecting diverse customer engagement.

```
## Marketing opt-in counts by gender

df.groupby(['gender'])['marketing_opt_in'].count()
```

```
gender
female     92
male       89
others    101
Name: marketing_opt_in, dtype: Int64
```

4.Business Question

1. **Which acquisition source brought in the most users last month?**

YouTube brought in the highest number of users last month (September), indicating it was the most effective channel for user acquisition during that period.

```
## Which acquisition source brought in the most users last month?
#Which acquisition source brought in the most users last month?
month = df.groupby([df['signup_date'].dt.strftime('%B'),'source'])['customer_id'].count()
month
last_month = month.index.get_level_values(0).unique()[-1]  # Get the last month
last_month_data = month.xs(last_month, level=0)  # Filter for just that month
max_source = last_month_data.idxmax()  # Find the source with max users
print("The highest number of users last month (",last_month,") came from the source '",str.upper(max_source),"'")
```

```
The highest number of users last month ( September ) came from the source ' YOUTUBE '
```

2. **Which region shows signs of missing or incomplete data?**

There are **30 missing values** in the `region` column, suggesting some user records lack region information. Among the known regions, **north (65 users)** and **east (61 users)** have the largest counts.

```
# Which region shows signs of missing or incomplete data?

print("\nCount of null values in region column:", df['region'].isnull().sum())

## Group by Region
region_groups = df.groupby('region')
print("\nGroup counts by region:")
print(region_groups.size())
```

```
Count of null values in region column: 30

Group counts by region:
region
central    39
east       61
north      65
south      59
west       46
dtype: int64
```

3. **Are older users more or less likely to opt in to marketing?**

- Older users (age > 50) who opted in: **18**
- Younger users (age < 50) who opted in: **106**
- This suggests that **younger users are more likely to opt in** to marketing than older users.

```
## Are older users more or less likely to opt in to marketing?

old = df[(df['age'] > 50) & (df['marketing_opt_in']== True)][['marketing_opt_in']]
print("No.of old users ",old.count())
young =  df[(df['age'] < 50) & (df['marketing_opt_in']==True)][['marketing_opt_in']]
print("No.of young users ",young.count())
```

```
No.of old users  marketing_opt_in    18
dtype: int64
No.of young users  marketing_opt_in    106
dtype: int64
```

4. **Which plan is most commonly selected, and by which age group?**

- The **26–35 age group** selects the most plans overall, with a fairly balanced spread across **premium (31)** and **pro (32)** plans.
- The **18–25 group** also shows strong uptake, especially for the **basic (28)** plan.
- The **60+ group** has no recorded plan selections, possibly due to missing or no data in this category.

```python
##Which plan is most commonly selected, and by which age group?

##cretaing bins for age group and grouping of ages
bins = [0, 25, 35, 45, 60, 100]
labels = ['18–25', '26–35', '36–45', '46–60', '60+']

df['age_group'] = pd.cut(df['age'], bins=bins, labels=labels, right=True, include_lowest=True)
df['age_group']

df.groupby(['age_group','plan_selected'])['customer_id'].count()
```

```
/var/folders/cp/ppr7gj1102lcjn2shl1ggyn40000gn/T/ipykernel_6807/3132018193.py:10: FutureWarning: The default of observed=False is deprecate
d and will be changed to True in a future version of pandas. Pass observed=False to retain current behavior or observed=True to adopt the f
uture default and silence this warning.
  df.groupby(['age_group','plan_selected'])['customer_id'].count()
```

age_group	plan_selected	
18–25	basic	28
	premium	23
	pro	24
	unknownplan	0
26–35	basic	23
	premium	31
	pro	32
	unknownplan	2
36–45	basic	11
	premium	23
	pro	13
	unknownplan	2
46–60	basic	19
	premium	19
	pro	21
	unknownplan	2
60+	basic	0
	premium	0
	pro	0
	unknownplan	0

Name: customer_id, dtype: int64

5. Recommendations

- **Focus Campaigns on Younger Age Groups (18–35):** These users show the highest engagement with marketing opt-ins and plan selections. Tailoring promotions and offers to this demographic could improve conversion and retention.
- **Invest More in YouTube Marketing:** Since YouTube brought in the most users last month, increasing budget or targeting efforts on this platform could further enhance acquisition performance.
- **Improve Region Data Collection:** With 30 missing region values, ensure that the signup process requires region input or automatically detects it (e.g., via IP geolocation) to support better regional analysis.

6.Data Issues or Risks

- **Issue: Invalid Age Values (e.g., age = 206)**
- **Risk:** Outliers like this distort age-related insights and lead to inaccurate targeting or segmentation.
- **Fix:** Implement **input validation at the source**, such as setting an acceptable age range (e.g., 18–100) in signup forms, and flagging out-of-range values in automated data quality checks during data ingestion or ETL processes.

Other Data Issues:

Data Entry Errors: Outlier ages and potential mislabeling (e.g., 'unknownplan') can skew analysis and decision-making.

Missing Data: Nulls in key columns like `region` and `age` reduce dataset completeness and may bias results.

Marketing Opt-In Ambiguity: Counting opt-ins without verifying true consent or updated preferences might misrepresent customer engagement.