Data Cleaning & Integrity Procedures

In [3]:
```
# Import necessary libraries

import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
```

In [4]:
```
# Read the file

df = pd.read_csv("train.csv", delimiter=";")
df.head()
```

Out[4]:

	age	job	marital	education	default	balance	housing	loan	cont
0	58	management	married	tertiary	no	2143	yes	no	unkn
1	44	technician	single	secondary	no	29	yes	no	unkn
2	33	entrepreneur	married	secondary	no	2	yes	yes	unkn
3	47	blue-collar	married	unknown	no	1506	yes	no	unkn
4	33	unknown	single	unknown	no	1	no	no	unkn

In [5]:
```
# Rows & Columns of data

df.shape
```

Out[5]:
```
(45211, 17)
```

In [6]:
```
# Distribution of Values & Data types

df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 45211 entries, 0 to 45210
Data columns (total 17 columns):
 #   Column     Non-Null Count  Dtype 
---  ------     --------------  ----- 
 0   age        45211 non-null  int64 
 1   job        45211 non-null  object
 2   marital    45211 non-null  object
 3   education  45211 non-null  object
 4   default    45211 non-null  object
 5   balance    45211 non-null  int64 
 6   housing    45211 non-null  object
 7   loan       45211 non-null  object
 8   contact    45211 non-null  object
 9   day        45211 non-null  int64 
 10  month      45211 non-null  object
 11  duration   45211 non-null  int64 
 12  campaign   45211 non-null  int64 
 13  pdays      45211 non-null  int64 
 14  previous   45211 non-null  int64 
 15  poutcome   45211 non-null  object
 16  y          45211 non-null  object
dtypes: int64(7), object(10)
memory usage: 5.9+ MB
```

In [7]:
```
# Checking for Missing(null) values

df.isnull().sum()
```

Out[7]:
```
age          0
job          0
marital      0
education    0
default      0
balance      0
housing      0
loan         0
contact      0
day          0
month        0
duration     0
campaign     0
pdays        0
previous     0
poutcome     0
y            0
dtype: int64
```

In [8]:
```
# Descriptive statistics for numerical columns (detects outliers or un

df.describe()
```

Out[8]:

	age	balance	day	duration	campaign
count	45211.000000	45211.000000	45211.000000	45211.000000	45211.000000
mean	40.936210	1362.272058	15.806419	258.163080	2.763841
std	10.618762	3044.765829	8.322476	257.527812	3.098021
min	18.000000	-8019.000000	1.000000	0.000000	1.000000
25%	33.000000	72.000000	8.000000	103.000000	1.000000
50%	39.000000	448.000000	16.000000	180.000000	2.000000
75%	48.000000	1428.000000	21.000000	319.000000	3.000000
max	95.000000	102127.000000	31.000000	4918.000000	63.000000

In [19]:
```
# Check for duplicate rows

duplicate_rows = df.duplicated()
print("Total duplicate rows:", duplicate_rows.sum())
```

```
Total duplicate rows: 0
```

Data Validity Check

In [36]:
```
unknown_contact = df[df['contact'] == "unknown"]
unknown_contact.head()
```

Out[36]:

	age	job	marital	education	default	balance	housing	loan	cont
0	58	management	married	tertiary	no	2143	yes	no	unkn
1	44	technician	single	secondary	no	29	yes	no	unkn
2	33	entrepreneur	married	secondary	no	2	yes	yes	unkn
3	47	blue-collar	married	unknown	no	1506	yes	no	unkn
4	33	unknown	single	unknown	no	1	no	no	unkn

In [38]:
```
unknown_new = unknown_contact[unknown_contact['campaign'] == 32]
unknown_new.head()
```

Out[38]:

	age	job	marital	education	default	balance	housing	loan	
3331	50	entrepreneur	married	primary	no	461	yes	no	u
3483	59	management	married	tertiary	no	2319	yes	no	u
3529	53	blue-collar	married	secondary	no	1140	yes	no	u
4020	42	self-employed	married	tertiary	no	1932	yes	no	u
8238	38	admin.	married	primary	no	0	yes	no	u