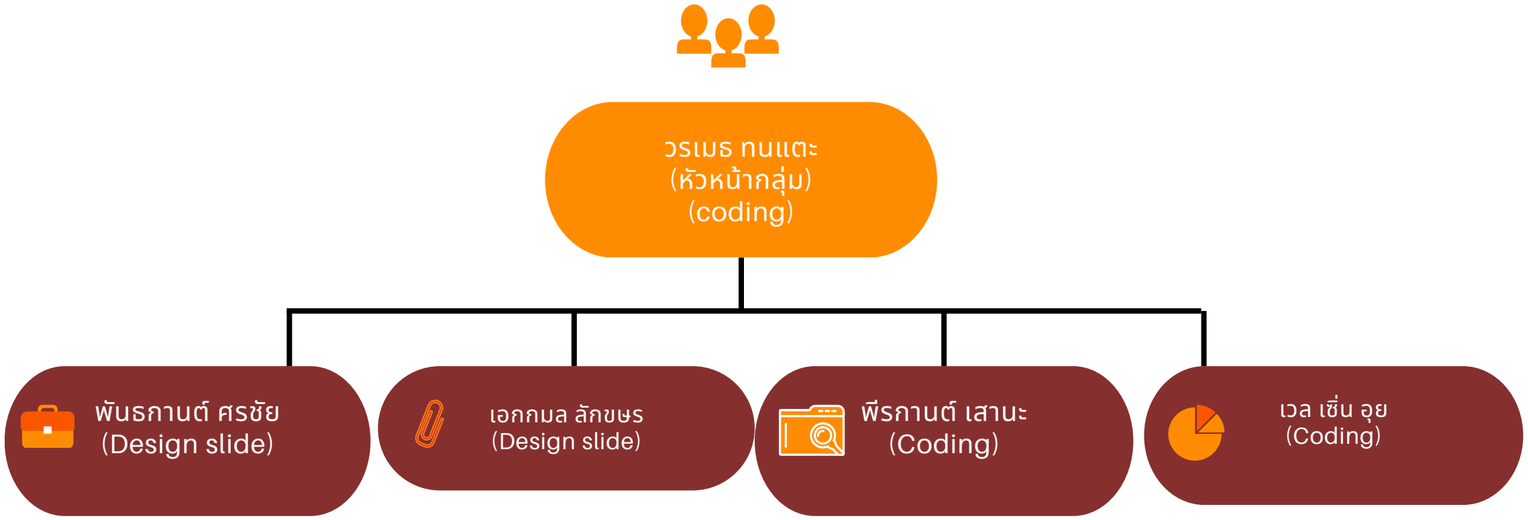
BAMBOO

Brazilian E-Commerce

BAMBOO

Group members



DATA CLEANING & EDA

วัตถุประสงค์ (Objective)

- เพื่อตรวจสอบความถูกต้องและเหมาะสมของข้อมูล (Data Audit)
- เพื่อฝึกการดำเนินงาน Data Cleaning เนื้องานด้าน
- เพื่อสร้าง Visualization สำหรับการวิเคราะห์ผลสำรวจ (EDA)

DATA AUDIT & CLEANING PLAN

IMPORT LIBRARIES

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

DOWNLOAD DATASET

```
import kagglehub

# Download latest version
path = kagglehub.dataset_download("olistbr/brazilian-ecommerce")

print("Path to dataset files:", path)
```

LOAD DATA

```
BZ_customers = pd.read_csv("/kaggle/input/brazilian-ecommerce/olist_customers_dataset.csv")
```

```
BZ_order_dataset = pd.read_csv("/kaggle/input/brazilian-ecommerce/olist_orders_dataset.csv")
```

BZ_ORDER_DATASET.INFO()

ตรวจสอบโครงสร้างและข้อมูลที่หายไป

```
BZ_order_dataset.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 99441 entries, 0 to 99440
Data columns (total 8 columns):
 #   Column                         Non-Null Count  Dtype 
---  ------                         --------------  ----- 
 0   order_id                       99441 non-null  object
 1   customer_id                    99441 non-null  object
 2   order_status                   99441 non-null  object
 3   order_purchase_timestamp       99441 non-null  object
 4   order_approved_at              99281 non-null  object
 5   order_delivered_carrier_date   97658 non-null  object
 6   order_delivered_customer_date  96476 non-null  object
 7   order_estimated_delivery_date  99441 non-null  object
dtypes: object(8)
memory usage: 6.1+ MB
```

BZ_ORDER_DATASET.DESCRIBE()

ดูสถิติเบื้องต้นของข้อมูล (แบบ TEXT)

```
BZ_order_dataset.describe()
```

	order_id	customer_id	order_status	order_purchase_timestamp	order_approved_at	order_delivered_carrier_date	order_delivered_customer_date	order_estimated_delivery_date
count	99441	99441	99441	99441	99281	97658	96476	99441
unique	99441	99441	8	98875	90733	81018	95664	459
top	66dea50a8b16d9b4dee7af250b4be1a5	edb027a75a1449115f6b43211ae02a24	delivered	2018-08-02 12:06:07	2018-02-27 04:31:10	2018-05-09 15:48:00	2018-05-14 20:02:44	2017-12-20 00:00:00
freq	1	1	96478	3	9	47	3	522

คำนวณเปอร์เซ็นต์ข้อมูลที่หายไป (MISSING DATA)

โค้ดนี้จะตรวจสอบทุกคอลัมน์ในตาราง BZ_ORDER_DATASET แล้วคำนวณว่ามีข้อมูลที่ว่าง (NULL/MISSING) อยู่กี่เปอร์เซ็นต์

```
# คำนวณเปอร์เซ็นต์ของ Missing Values ในแต่ละคอลัมน์
missing_percentage = (BZ_order_dataset.isnull().sum() / len(BZ_order_dataset)) * 100

# แสดงผลเฉพาะคอลัมน์ที่มีข้อมูลหาย
print(missing_percentage[missing_percentage > 0].sort_values(ascending=False))
```

```
order_delivered_customer_date    2.981668
order_delivered_carrier_date     1.793023
order_approved_at                0.160899
dtype: float64
```

ผลลัพธ์แสดงให้เห็นว่ามี 3 คอลัมน์ที่มีข้อมูลหาย โดยคอลัมน์ ORDER_DELIVERED_CUSTOMER_DATE (วันที่ส่งถึงลูกค้า) มีข้อมูลหายไปมากที่สุดเกือบ 3%

แก้ไขประเภทข้อมูล (DATA TYPE CLEANING)

แก้ Data type จาก object เป็น Date time

```
date_columns = ['order_purchase_timestamp', 'order_approved_at',
                'order_delivered_carrier_date', 'order_delivered_customer_date',
                'order_estimated_delivery_date']

for col in date_columns:
    BZ_order_dataset[col] = pd.to_datetime(BZ_order_dataset[col], errors='coerce')

BZ_order_dataset.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 99441 entries, 0 to 99440
Data columns (total 8 columns):
 #   Column                         Non-Null Count  Dtype
---  ------                         --------------  -----
 0   order_id                       99441 non-null  object
 1   customer_id                    99441 non-null  object
 2   order_status                   99441 non-null  object
 3   order_purchase_timestamp       99441 non-null  datetime64[ns]
 4   order_approved_at              99281 non-null  datetime64[ns]
 5   order_delivered_carrier_date   97658 non-null  datetime64[ns]
 6   order_delivered_customer_date  96476 non-null  datetime64[ns]
 7   order_estimated_delivery_date  99441 non-null  datetime64[ns]
dtypes: datetime64[ns](5), object(3)
memory usage: 6.1+ MB
```

โค้ดนี้จะแปลงคอลัมน์ที่ควรจะเป็น "วันที่" (เช่น ORDER_PURCHASE_TIMESTAMP) ซึ่งปัจจุบันถูกเก็บเป็น "ข้อความ" (OBJECT) ให้กลายเป็นประเภทข้อมูล "วันที่และเวลา" (DATETIME) ที่ถูกต้อง

ผลลัพธ์: การใช้ .INFO() อีกครั้งในตอนท้าย ช่วยยืนยันว่าคอลัมน์เหล่านั้นถูกเปลี่ยนประเภทเป็น DATETIME64[NS] สำเร็จแล้ว พร้อมสำหรับการวิเคราะห์ด้านเวลาต่อไป

```
print("Missing values per column after datetime conversion:")
print(BZ_order_dataset[date_columns].isnull().sum())
```

```
Missing values per column after datetime conversion:
order_purchase_timestamp            0
order_approved_at                 160
order_delivered_carrier_date     1783
order_delivered_customer_date    2965
order_estimated_delivery_date       0
dtype: int64
```

DATA AUDIT (CODE & OUTPUT)

- **DF.INFO(): แสดงผลและตรวจสอบ DATA TYPES (เช่น DATE เป็น OBJECT หรือไม่) และ MISSING VALUES**

จากการตรวจสอบพบว่า DATA TYPES เป็น OBJECT
และมีMISSING VALUES

```
# คำนวณเปอร์เซ็นต์ของ Missing Values ในแต่ละคอลัมน์
missing_percentage = (BZ_order_dataset.isnull().sum() / len(BZ_order_dataset)) * 100

# แสดงผลเฉพาะคอลัมน์ที่มีข้อมูลหาย
print(missing_percentage[missing_percentage > 0].sort_values(ascending=False))
```

```
order_delivered_customer_date    2.981668
order_delivered_carrier_date     1.793023
order_approved_at                0.160899
dtype: float64
```

```
print("Missing values per column after datetime conversion:")
print(BZ_order_dataset[date_columns].isnull().sum())
```

```
Missing values per column after datetime conversion:
order_purchase_timestamp            0
order_approved_at                 160
order_delivered_carrier_date     1783
order_delivered_customer_date    2965
order_estimated_delivery_date       0
dtype: int64
```

- **DF.DESCRIBE(): แสดงผลและตรวจสอบค่าสถิติ (เช่น MIN, MAX) เพื่อหา OUTLIERS**

```
BZ_order_dataset.describe()
```

	order_id	customer_id	order_status	order_purchase_timestamp	order_approved_at	order_delivered_carrier_date	order_delivered_customer_date	order_estimated_delivery_date
count	99441	99441	99441	99441	99281	97658	96476	99441
unique	99441	99441	8	98875	90733	81018	95664	459
top	66dea50a8b16d9b4dee7af250b4be1a5	edb027a75a1449115f6b43211ae02a24	delivered	2018-08-02 12:06:07	2018-02-27 04:31:10	2018-05-09 15:48:00	2018-05-14 20:02:44	2017-12-20 00:00:00
freq	1	1	96478	3	9	47	3	522

ไม่มี OUTLIERS ใน TABLE นี้

CLEANING ACTION PLAN (TEXT/MARKDOWN)

ระบุปัญหา 3-5 ข้อ ที่พบจาก DATA AUDIT (เช่น "CUSTOMERID มี MISSING 20%", "QUANTITY มีค่าติดลบ", "UNITPRICE มี OUTLIER สูงมาก")

1.ปัญหาเรื่อง DATA TYPE เป็น OBJECT ไม่สามารถนำไปใช้ต่อได้
ข้อมูลของเราเป็นเวลาแต่มีการเก็บไว้เป็น OBJECT

2.เราได้เปลี่ยนจาก DATA TYPE ที่เป็น OBJECT เป็น DATETIME จึงทำให้เกิด MISSING ERORR เพราะอาจเกิดจาก อาจมีข้อมูลบางประเภทไม่ตรงTYPE

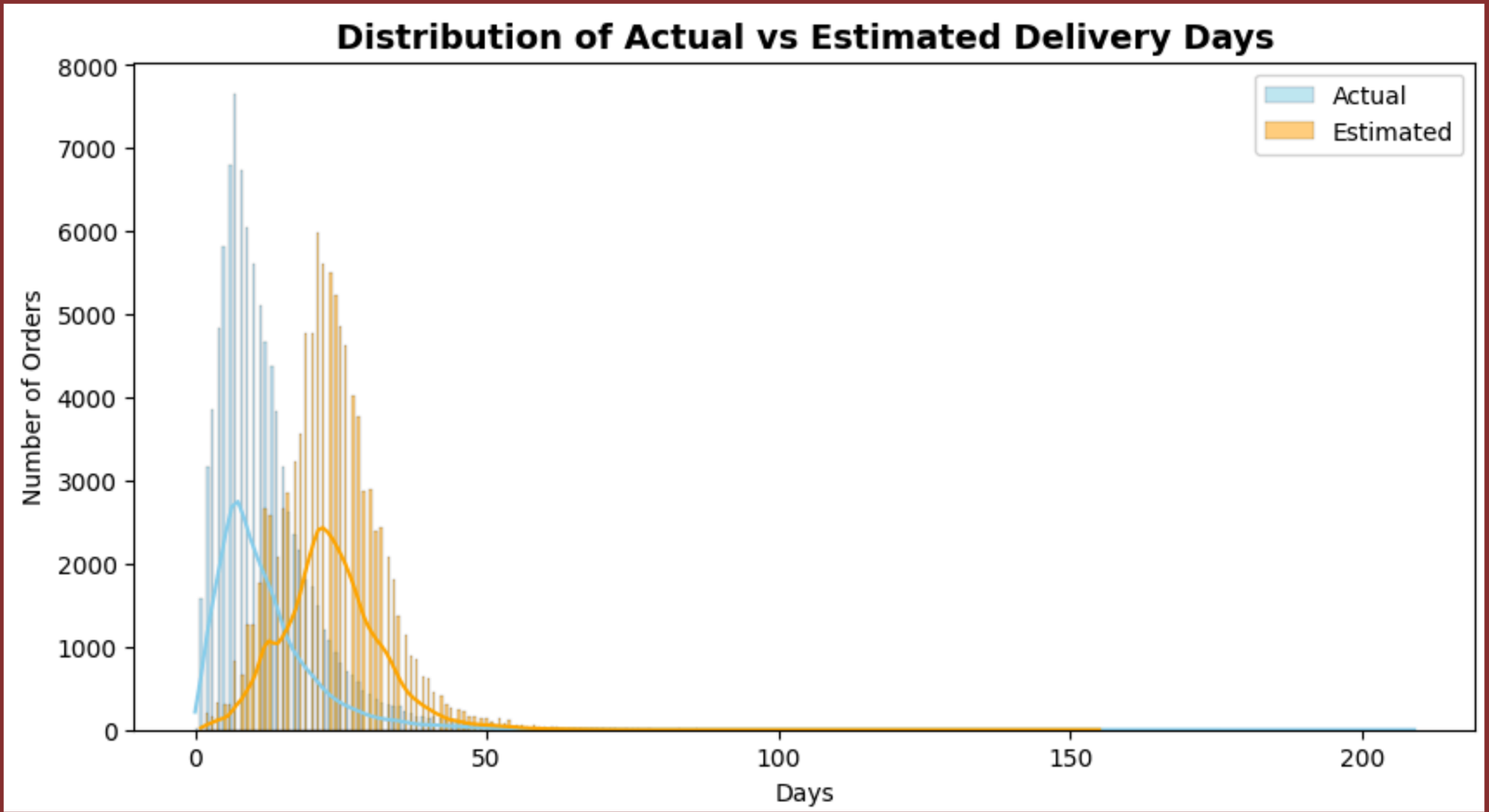
เสนอแผนการจัดการ (W3 ACTION PLAN): อธิบายว่าทีมของคุณจะจัดการกับปัญหาเหล่านี้อย่างไรในสัปดาห์หน้า (เช่น "ลบแถวที่ CUSTOMERID เป็น NULL", "ใช้ CAPPING ที่ 99TH PERCENTILE กับ QUANTITY")

1.แก้ DATA TYPE จาก OBJECT เป็น DATE TIME

2.การเปลี่ยนข้อมูลแล้วเกิดMISSING แก้โดยการเปลี่ยนค่าเป็นค่าว่าง

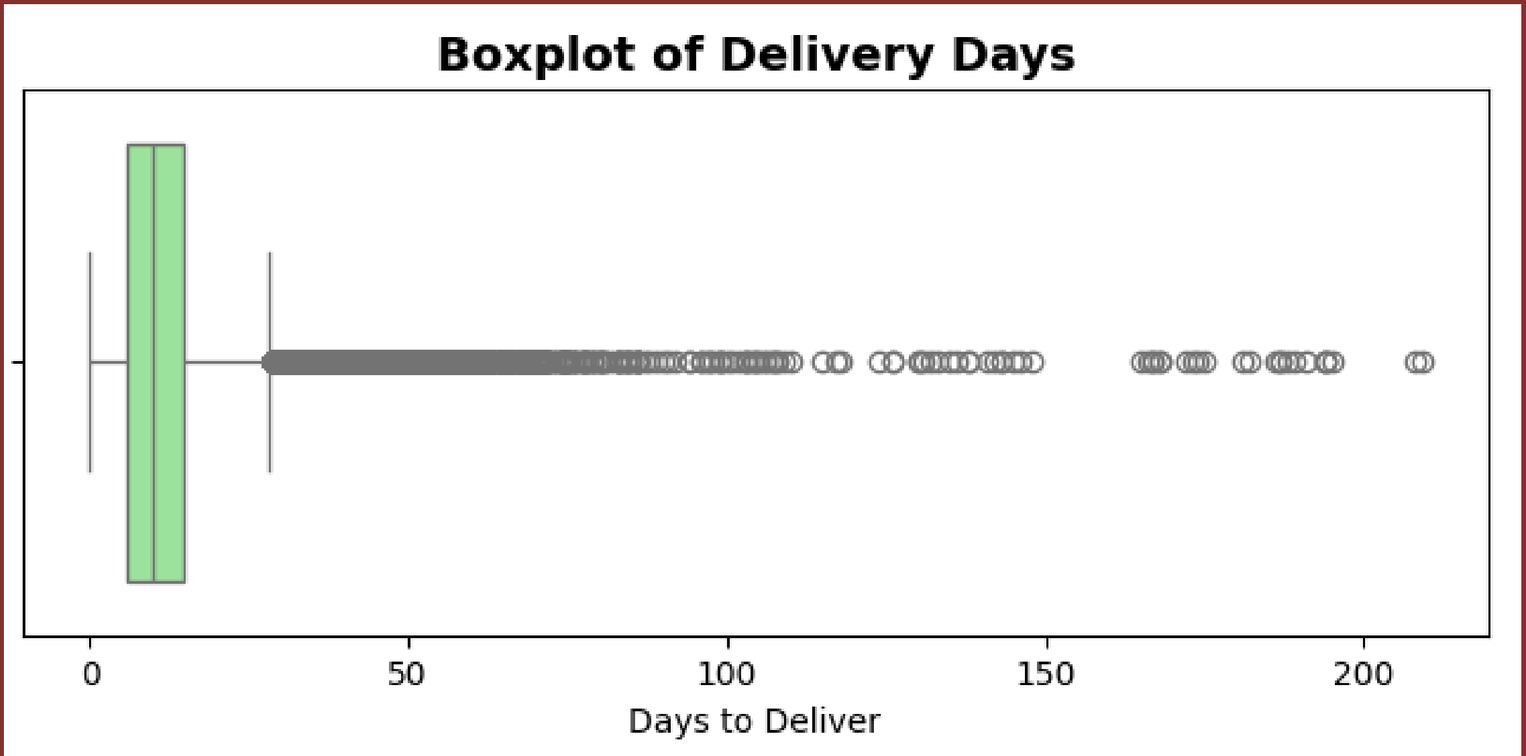
5 KEY VISUALIZATIONS (EDA)

DISTRIBUTION CHECK (2 กราฟ):
HISTOGRAM หรือ BOX PLOT ของตัวแปร



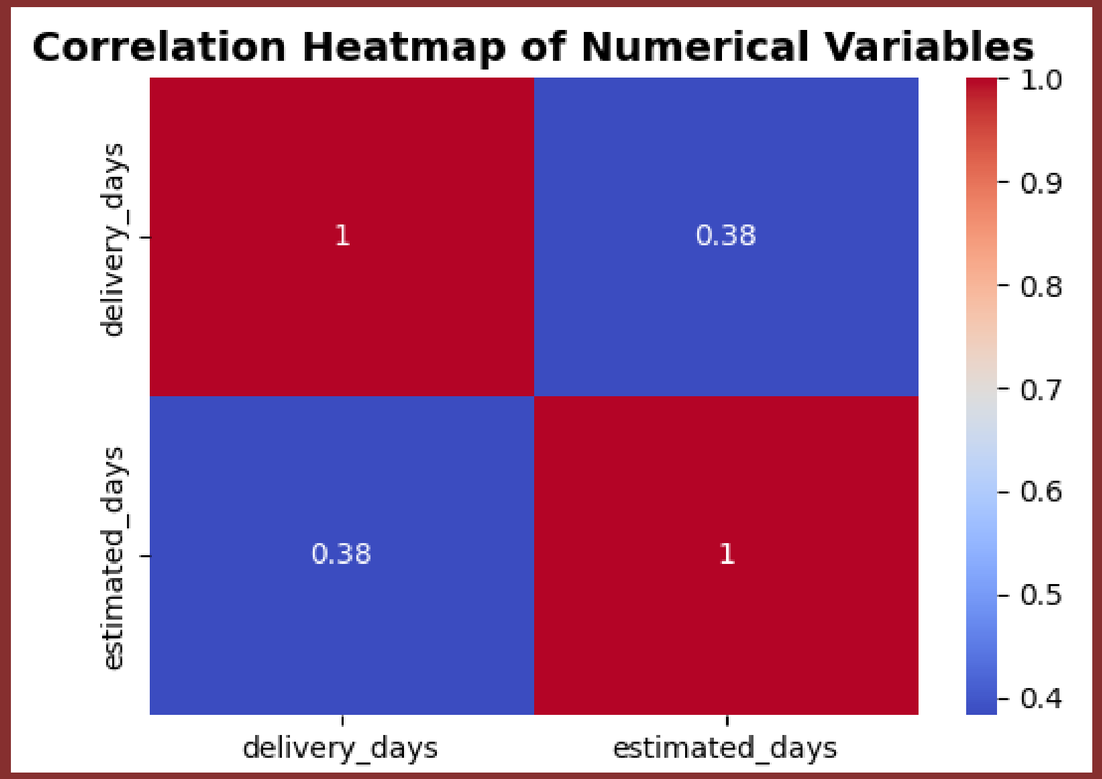
คำอธิบาย: "ตรวจสอบข้อมูลซ้ำซ้อน (DUPLICATE ROWS)"
ความหมาย: โค้ดนี้ใช้เพื่อตรวจสอบว่ามีแถว (ออเดอร์) ใดในตาราง ที่มีข้อมูล "ซ้ำกันทั้งแถว" หรือไม่
สิ่งที่พบ: ผลลัพธ์คือ 0 ซึ่งหมายความว่า ไม่มีข้อมูลที่ซ้ำซ้อนกันเลย ซึ่งเป็นสัญญาณที่ดีว่าข้อมูลมีคุณภาพ (DATA QUALITY)

NUMERICAL สำหรับ 2 ตัว (เช่น MONTHLY CHARGES, TOTAL REVENUE)



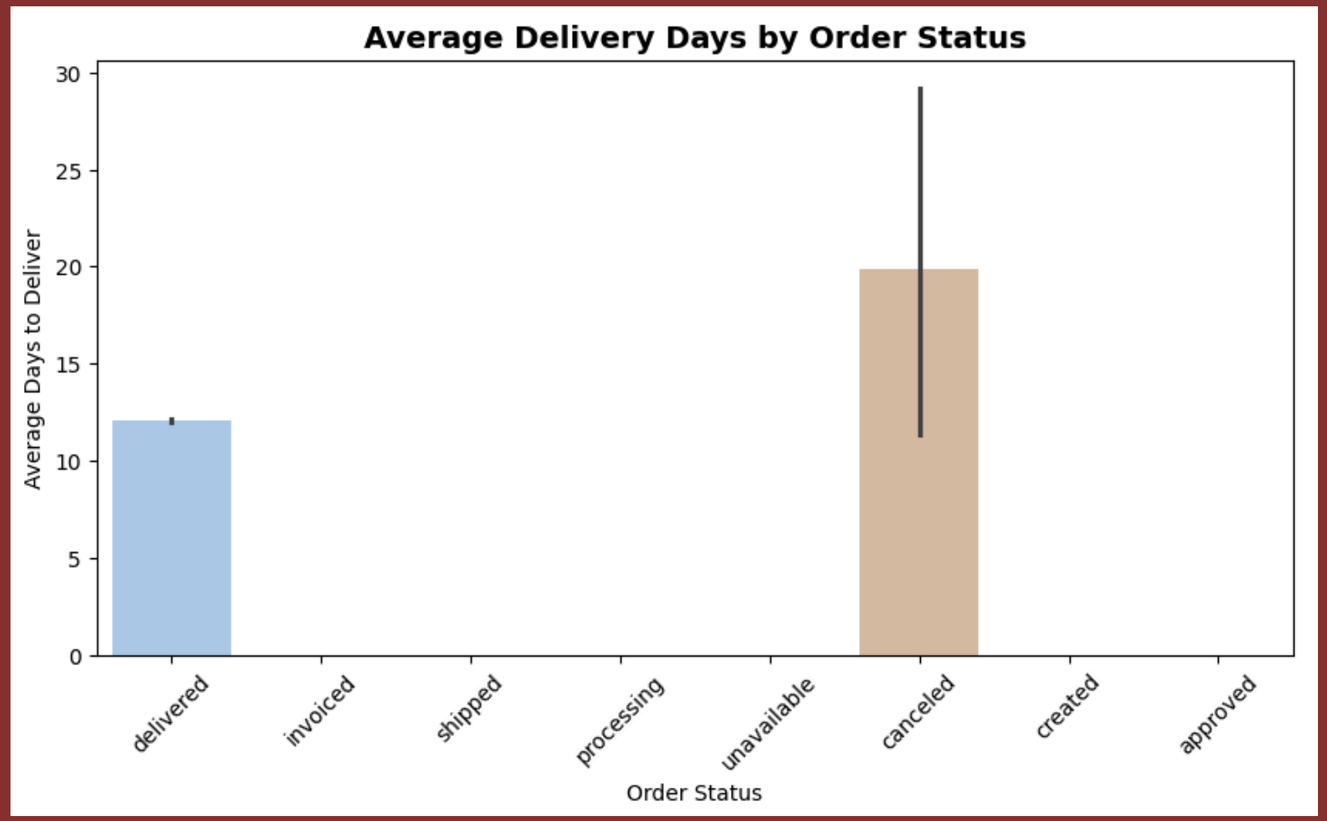
คำอธิบาย: สำรวจสถานะคำสั่งซื้อ (ORDER STATUS)"
ความหมาย: โค้ดนี้ใช้เพื่อตรวจสอบว่าในคอลัมน์ ORDER_STATUS มีค่า (สถานะ) ที่ไม่ซ้ำกันอยู่ทั้งหมดกี่แบบ และมีอะไรบ้าง

RELATIONSHIP CHECK (1 กราฟ):
CORRELATION HEATMAP และความสัมพันธ์ระหว่างตัวแปร NUMERICAL กับหมวด



คำอธิบาย: ค้นหา 3 อันดับแรกของคอลัมน์ที่มีข้อมูลหาย (MISSING DATA) มากที่สุด"
ความหมาย: โค้ดนี้ใช้เพื่อตรวจสอบคุณภาพข้อมูล โดยจะนับจำนวนข้อมูลที่ว่าง (NULL) ในทุกคอลัมน์ แล้วเรียงลำดับจากมากไปน้อย

BIVARIATE COMPARISON (2 กราฟ):
BAR PLOT หรือ GROUPED BOX PLOT เพื่อเปรียบเทียบกลุ่ม
สำคัญภาพ: กราฟ 2 ชุดนี้ คือ ฮิสโตแกรมและ BOX PLOT ที่แสดงการกระจายตัวของข้อมูล NUMERICAL สำหรับ 2 ตัวแปร ตามลำดับ



กราฟนี้แสดงให้เห็นว่า ประสิทธิภาพการจัดส่งดีขึ้นอย่างต่อเนื่องในแต่ละปี โดยเวลาจัดส่งเฉลี่ย (เส้นกลางกล่อง) ลดลง และมีความ สม่ำเสมอมากขึ้น (กล่องแคบลง) เมื่อเวลาผ่านไป
จุดสังเกตหลัก (KEY OBSERVATIONS)
แกน Y (แนวตั้ง): คือ "ระยะเวลาจัดส่ง" (ยิ่งต่ำยิ่งดี)
แกน X (แนวนอน): คือ "ปีที่สั่งซื้อ" (กลุ่มที่ใช้เปรียบเทียบ)

กราฟนี้แสดงให้เห็น การเติบโตของยอดออเดอร์ ตลอดช่วงเวลา โดยเฉพาะอย่างยิ่ง มีการเติบโตแบบก้าวกระโดด ในช่วงปลายปี 2017
จุดสังเกตหลัก (KEY OBSERVATIONS)
แกน Y (แนวตั้ง): คือ จำนวนออเดอร์ ในแต่ละเดือน
แกน X (แนวนอน): คือ ช่วงเวลา เรียงลำดับตั้งแต่ปี 2016 ไปจนถึงประมาณเดือนสิงหาคม 2018