Olist eCommerce 데이터 분석

1기 전상택

프로젝트 개요

- 분석 주제 선정 배경: 이커머스 고객 주문 데이터를 바탕으로 다양한 EDA 진행
- 데이터셋 소개: 브라질 쇼핑몰 Olist에서 만든, 이커머스 데이터셋
- 해결하고자 하는 문제 또는 목적
  - 많이 팔리는 제품 카테고리는 무엇일까?
  - 고객이 주로 구매하는 요일은 언제일까?
  - 고객이 주로 구매하는 시간대는 언제일까?

데이터셋 설명

- 데이터셋 개요 및 출처

  Kaggle **Brazilian E-Commerce Public Dataset by Olist**

- 주요 변수 설명

  Price: 구매 가격

  Customer City: 배송 도시

  Payment Type: 결제 수단

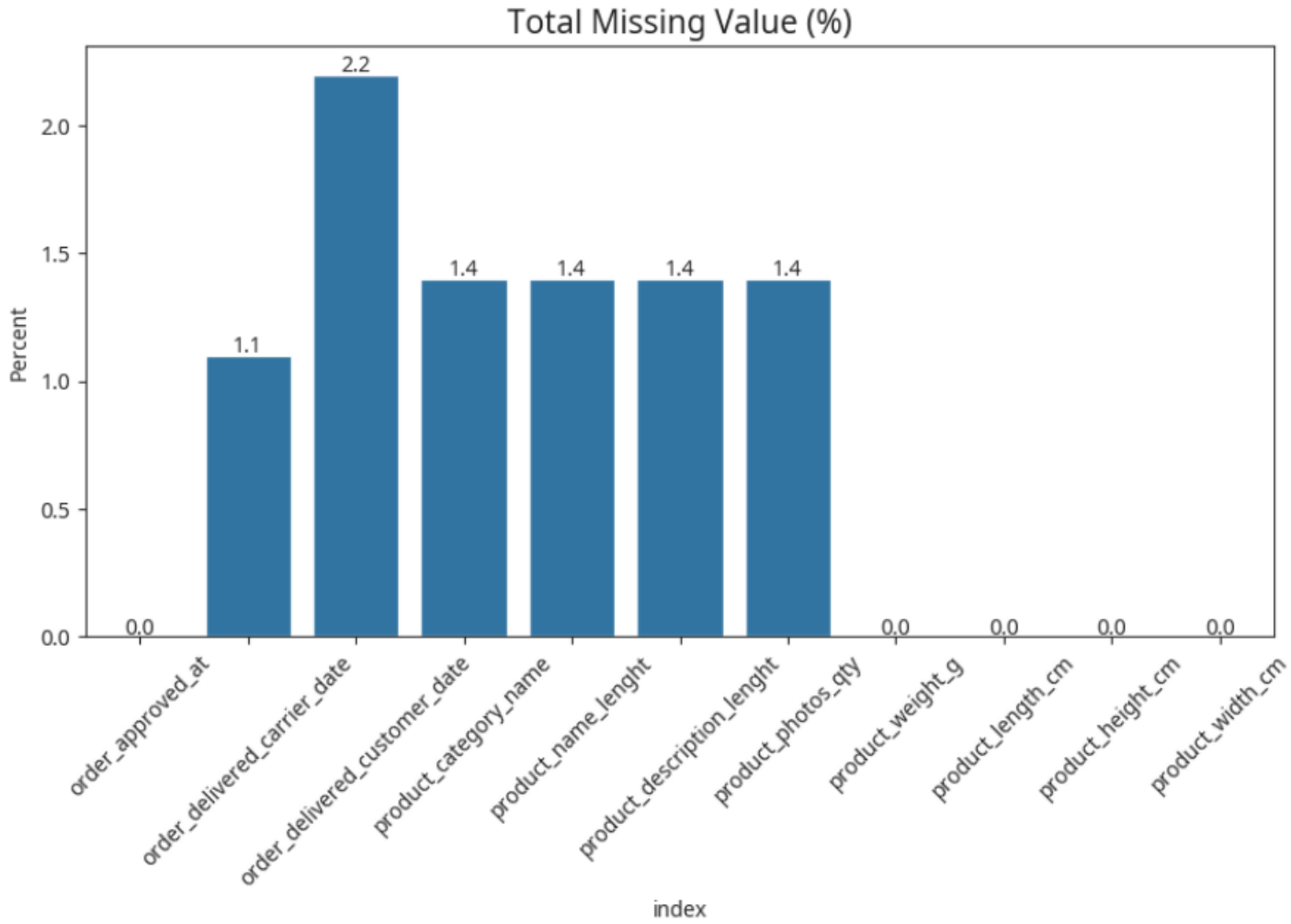
  Product_category_name: 제품 카테고리

데이터 전처리

- 전처리 기법

  결측치 처리: 결측치 10% 미만

  → 삭제 처리



데이터 전처리

- 전처리 기법

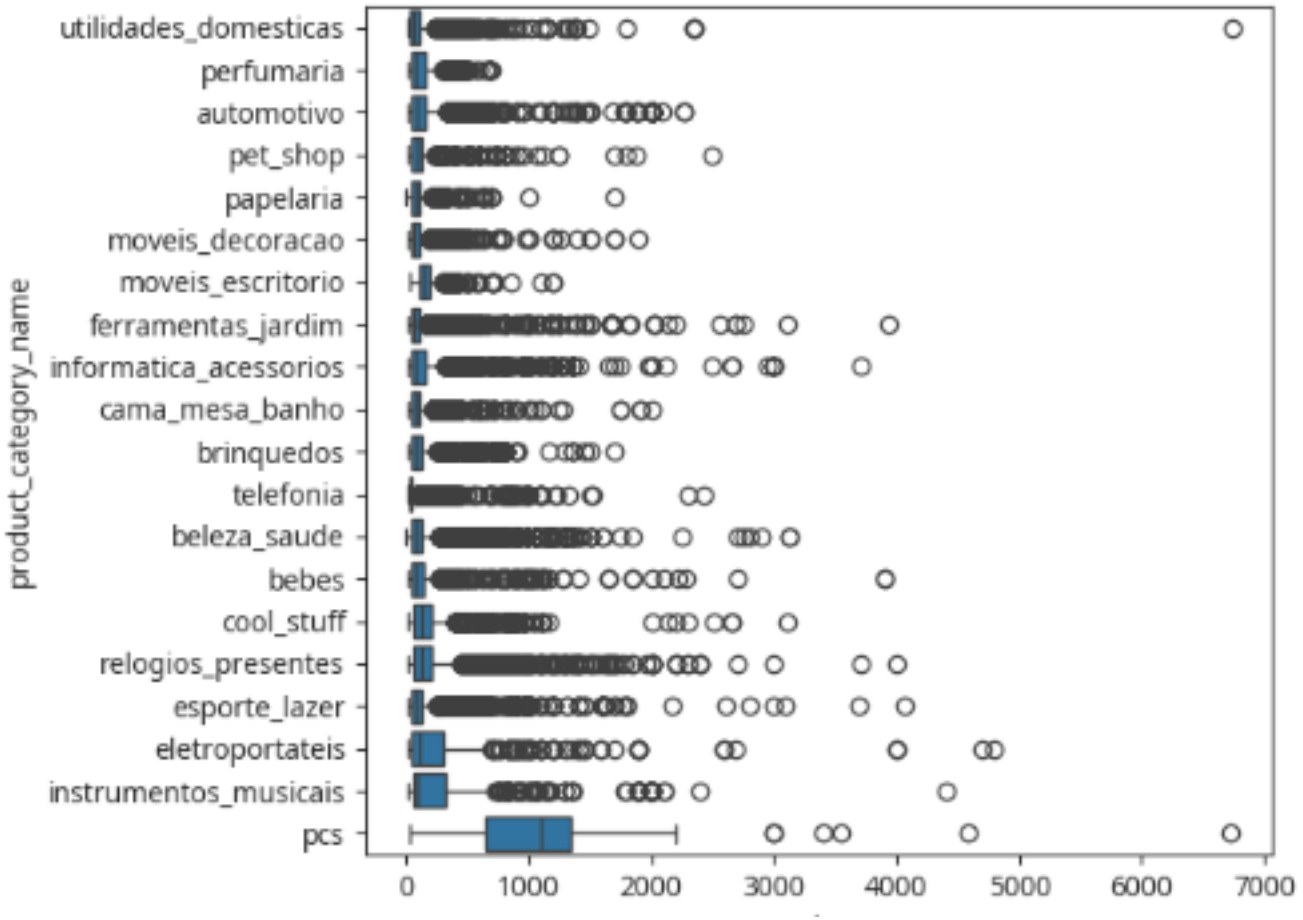
이상치 처리

굉장히 많이 나왔지만,

이커머스 쇼핑몰 특성상

가격대가 다양할 것이라고 생각

→ 따로 처리하지 않음



EDA 요약 (1)

- 주요 시각화 결과

- **customer_city**

  대도시 위주로 구매가 많이 발생했다

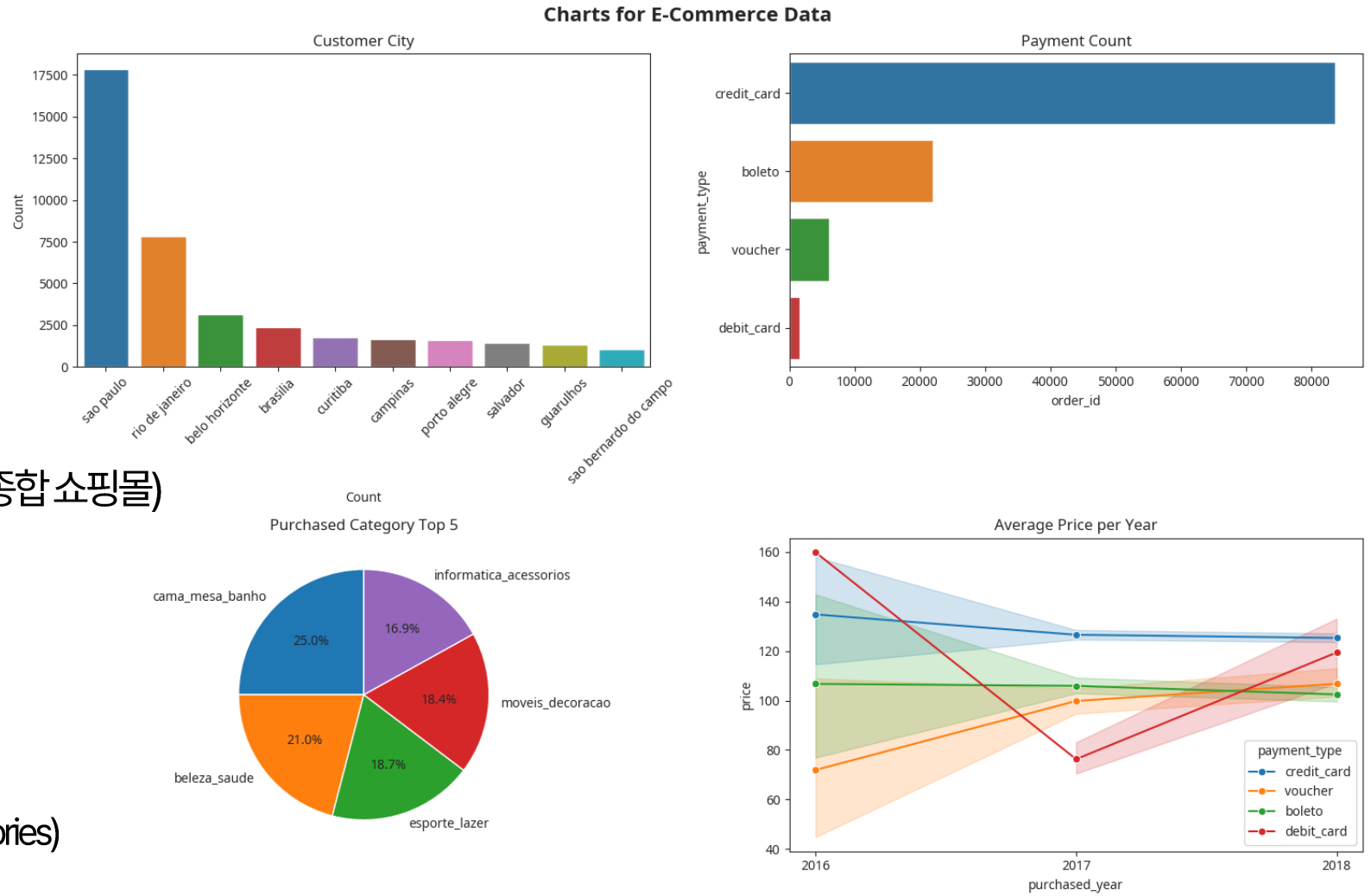
- **payment_type**

  credit card로 결제한 고객이 압도적

- **category_counts**

  다양한 분야의 카테고리들이 판매되었다 (종합 쇼핑몰)

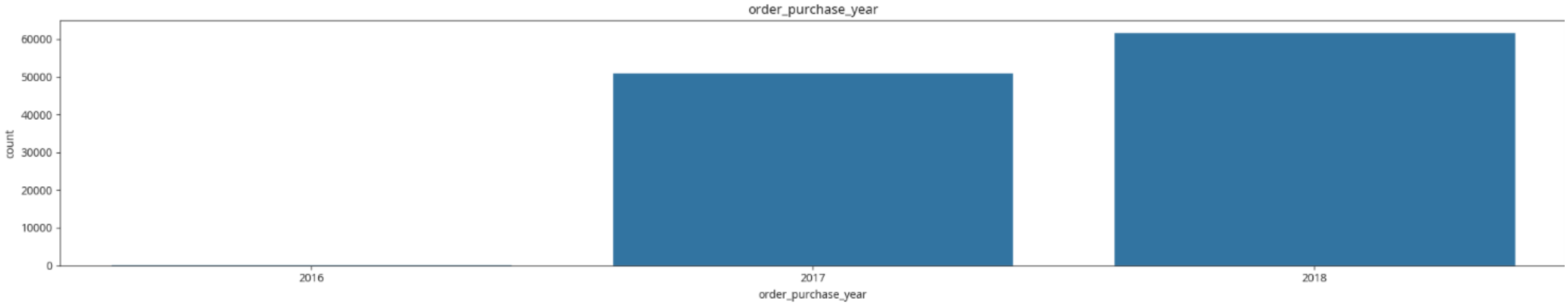
  - cama_mesa_banho (bed_bath_table)
  - beleza_saude (health_beauty)
  - esporte_lazer(sports_leisure)
  - moveis_decoracao (furniture_deco)
  - informatica_acessorios (computers_accessories)



EDA 요약 (2)

- 주요 시각화 결과

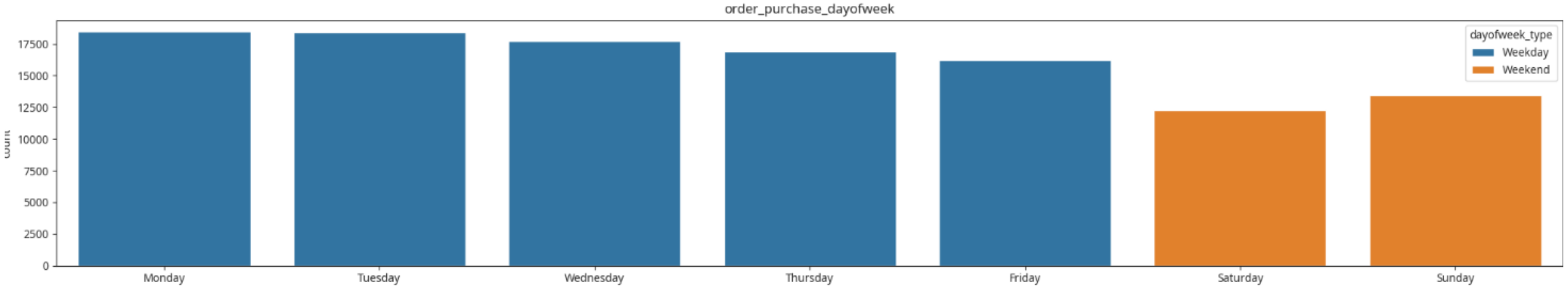
  2018년 결제 횟수가 가장 많고, 봄 ~ 여름이 가장 많다



EDA 요약 (3)

- 주요 시각화 결과

  주로 주중, 낮 시간대 결제를 많이 했다

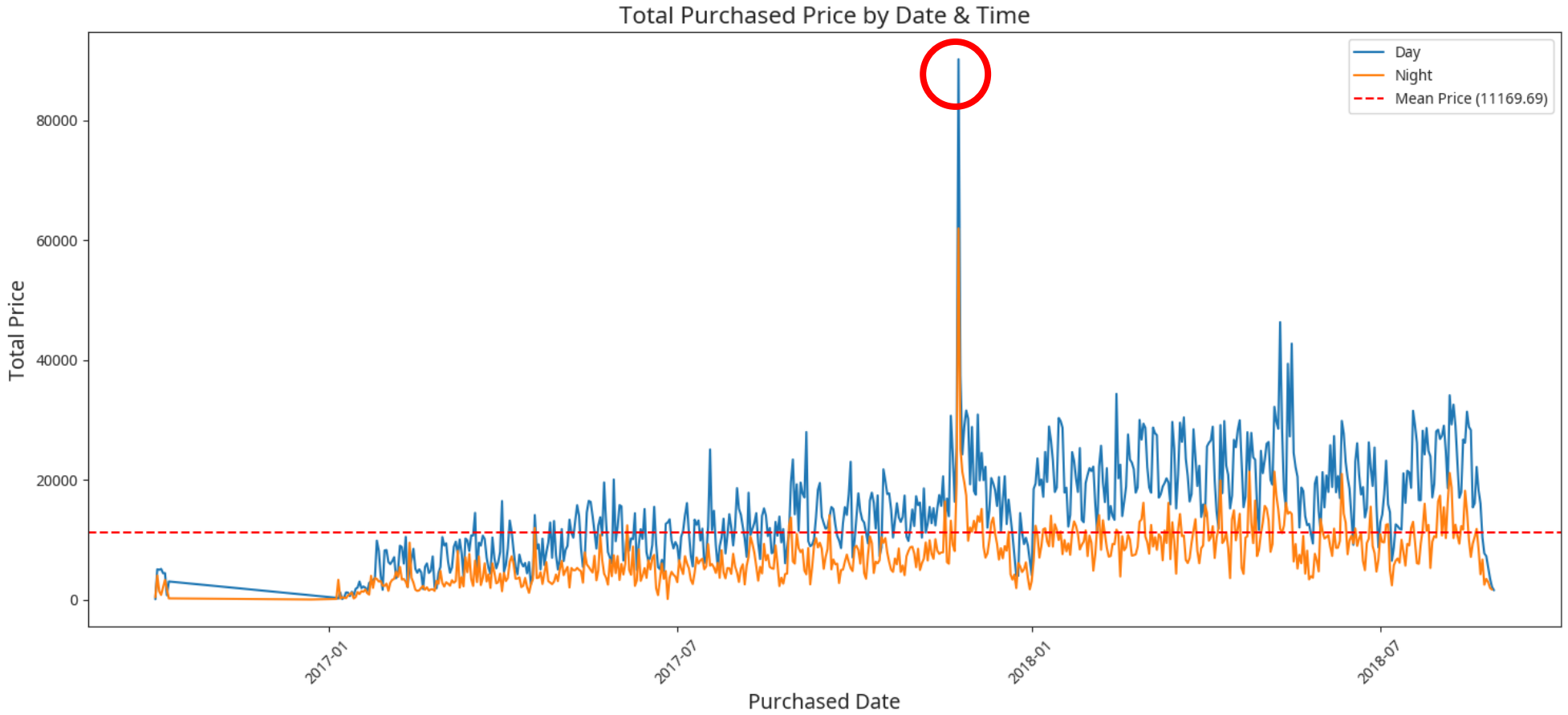


EDA 요약 (3)

- 주요 시각화 결과

  특정 시점, 구매 가격 급증

  → 2017년 블랙 프라이데이 때 구매가 급격히 증가



Feature Engineering

- Datetime 관련 파생 변수 생성

  연, 월, 일, 계절, 주말 여부, 낮/밤 등등 → 시간적인 요소를 고려하여 고객 행동 파악 목적

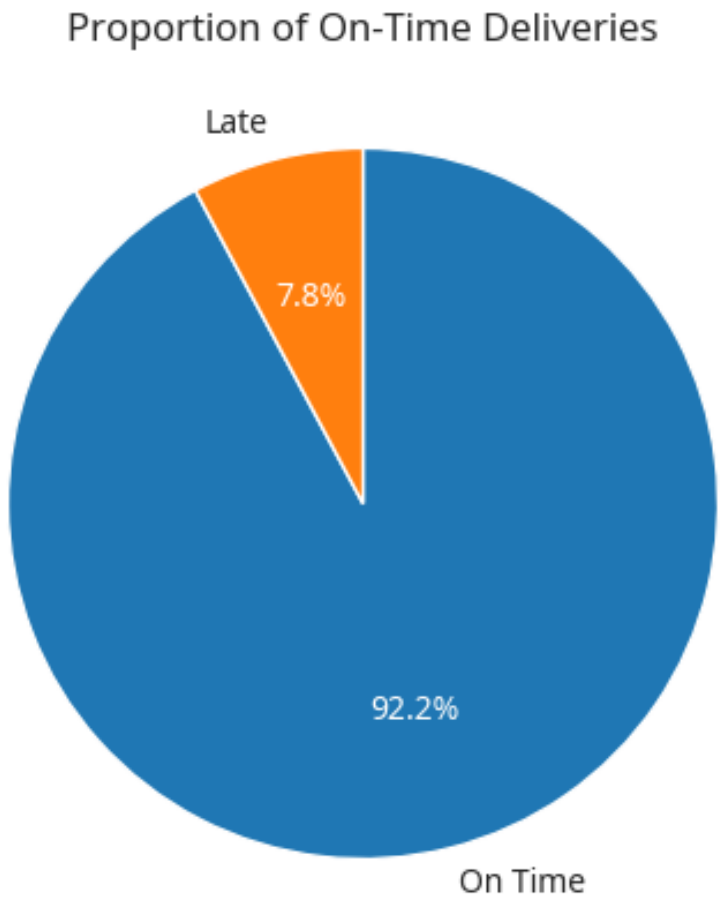
```python
# order_purchase_month에 대해 season 컬럼 생성 (3개월 단위: 봄, 여름, 가을, 겨울)
def month_to_season(m):
    if m in [3, 4, 5]:
        return 'Spring'  # 봄
    elif m in [6, 7, 8]:
        return 'Summer'  # 여름
    elif m in [9, 10, 11]:
        return 'Fall'    # 가을
    else:
        return 'Winter'  # 겨울

df_mod['season'] = df_mod['order_purchase_month'].apply(month_to_season)

# order_purchase_day에 대해 day_type 컬럼 생성
# 요일 정보(order_purchase_dayofweek)를 사용하여 주말(Saturday, Sunday)은 Weekend, 나머지는 Weekday로 분류
df_mod['day_type'] = df_mod['order_purchase_dayofweek'].apply(lambda x: 'Weekend' if x in ['Saturday', 'Sunday'] else 'Weekday')

# order_purchase_hour에 대해 hour_type 컬럼 생성
# 7시부터 18시는 "Day", 그 외는 "Night"으로 분류
def hour_to_type(h):
    return 'Day' if 7 <= h <= 18 else 'Night'

df_mod['hour_type'] = df_mod['order_purchase_hour'].apply(hour_to_type)

# order_purchase_dayofweek에 대해서도 dayofweek_type 컬럼 생성 (주말/주중)
df_mod['dayofweek_type'] = df_mod['order_purchase_dayofweek'].apply(lambda x: 'Weekend' if x in ['Saturday', 'Sunday'] else 'Weekday')
```

Feature Engineering

- 정시 배송 확인 파생 변수 생성

  예상 배송 시간, 실제 배송 시간 변수 존재

  → 실제 및 예상 시간의 차이를 확인하여, 늦게 배송된 비율 확인



결과 요약 및 인사이트

- 분석 결과 요약
  - 브라질 고객의 구매 패턴 파악
    - 대도시, 신용카드 고객 위주
    - 봄 ~ 여름 주중, 낮 시간
  - Olist 이커머스 쇼핑몰의 지연 배송 비율 약 8%

- 한계점
  - 고객 성별, 나이 데이터가 없어,
    고객에 대한 심도 있는 분석 불가능
  - 가공된 데이터이기 때문에,
    별도의 Feature Engineering 필요성 적음
  - 여러 번 구매한 고객에 대한 정보가 없어,
    코호트 및 퍼널, RFM 분석 실패
    (향후 추가 데이터를 통해 개선 예정)

감사합니다