



X 2



X 1



## EDA : Analysis for Refund Defense (based on FastCampus classes in 2022)-최종본

	유형	고객id	포맷	카테고리	코스(상품) 이름	쿠폰이름	판매가격	결제수단	실거래금액	쿠폰할인액	거래금액	환불금액
0	PAYMENT	20053	음인원	업무 생산성	음인원 패키지 : 김원손의 파이썬 업무자동화 유치원	No Data	189000	TRANSFER	159000	0	159000	0
1	PAYMENT	58309	음인원	부동산/금융	초격차 패키지 : 한 번에 끝내는 부동산 금융(PF) 실무	No Data	501000	TRANSFER	501000	0	501000	0
2	PAYMENT	18075	음인원	영상/3D	편집하는여자의 영상편집 마스터클래스 - 제 6강 다양한 효과를 응용하여 애프터이펙트...	No Data	549000	CARD	549000	0	549000	0
3	PAYMENT	41898	음인원	프로그래밍	초격차 패키지 : 85개 프로젝트로 배우는 Android 앱 개발 feat. Jet...	[WELCOME] 프로그래밍 3만원할인	165000	CARD	135000	30000	135000	0
4	PAYMENT	94328	음인원	프로그래밍	초격차 패키지 : 개발자 취업 합격 패스 With 코딩테스트, 기술 면접	No Data	239000	CARD	239000	0	239000	0
...	...	...	...	...	...	...	...	...	...	...	...	...
159323	PAYMENT	39658	음인원	부업/창업	음인원 패키지 : 제 2의 월급을 만들어주는 직장인 스몰비즈니스 가이드	[NEW22] 온라인 강의 1만원 즉시 추가 할인 쿠폰	217000	CARD	207000	10000	207000	0
159324	PAYMENT	95136	음인원	영상/3D	[Kih] 닷필의 미디어아트 그래픽 클래스 : 모션그래픽 뉴패러다임 Creative...	No Data	385000	CARD	385000	0	385000	0
159325	PAYMENT	12764	음인원	데이터 사이언스	음인원 패키지 : 파이썬을 활용한 데이터 전처리 Level UP	[WELCOME] 패캠프 저율이지? 3만원 할인쿠폰	166000	CARD	136000	30000	136000	0
159326	PAYMENT	19606	음인원	업무 생산성	음인원 패키지 : 100억을 움직이는 50가지 PPT 디자인 원칙	No Data	289000	VBANK	289000	0	289000	0
159327	REFUND	62567	음인원	데이터 사이언스	음인원 패키지 : 머신러닝 서비스 구축을 위한 실전 MLOps	No Data	370000	TRANSFER	-345240	0	0	-345240

157753 rows x 12 columns

### Mini Project 발표자료

by 23th TEAM of YearDream 3rd (a.k.a 2ki4mi)

## 0. 데이터 분석 목차

### 0. 데이터 분석 목차

#### 1. 서론 : 문제 정의

##### 1.1 분석 목표

##### 1.2 분석 배경

#### 2. 본론 : 데이터 탐색

##### 2.1 데이터 탐색 개요

##### 2.2 사용 Library

##### 2.3 데이터셋 기본정보 파악

#### 3. 데이터 전처리

##### 3.1 전처리 목표

##### 3.2 전처리 개요

###### 3.2.1. 필요 없는 Column 제거 : 거래id, 코스ID, 사이트, 거래일자

###### 3.2.2. 제거 해야 할 행 확인 → 필요 없는 Row 제거

###### 3.2.3 결측치 확인 및 처리

###### 3.2.4 데이터타입 확인

#### 4. 시각화 개요

##### 4.1 시각화 목적

##### 4.2 시각화

###### 4.2.1 기본 시각화

###### 4.2.2. 구매 확정(PAYMENT) 비율 알아보기

###### 4.2.3 환불(REFUND) 비율 알아보기

###### 4.2.4 구매확정/환불 비교하기

###### 4.2.5 강의 Format에 대한 분석

###### 4.2.6 환불에 대한 인사이트 도출

###### 4.2.7 상관관계 분석

#### 5. 마무리

##### 5.1 분석 결과

##### 5.2 방안 제시

##### 5.3 한계점


#### 6. 피드백

##### 6.1 분석을 통해 느낀 점


##### 6.2 최종소감

# 1. 서론 : 문제 정의

## 1.1 분석 목표

 소비자의 최종적 소비 행위(구매 및 환불)에 영향을 미치는 요인을 분석하기 위해 2022년 패스트캠퍼스 내 강의 구매 및 환불 현황을 살펴보고, 이를 시각화하고자 한다. 또한 분석한 데이터를 통해 소비행위 고취 및 환불행위 방어 등의 수익화 방안을 탐색하고자 한다.

## 1.2 분석 배경



### 패스트캠퍼스 (fastcampus.co.kr)

2013년 패스트트랙아시아의 사내벤처로 시작하여 2014년부터 '패스트캠퍼스'라는 브랜드명을 사용하였으며, 2021년 데이원컴퍼니로 법인명을 변경하였다. 빠르고 유연한 사업의 운영을 위해 4개의 사업부를 사내독립기업(CIC: Company In Company)으로 승격하여, 데이원컴퍼니라는 레이블 아래 패스트캠퍼스CIC, 레모네이드CIC, 콜로소CIC, 스노우볼CIC가 독립적인 회사로 운영되고 있습니다.

# 2. 본론 : 데이터 탐색

## 2.1 데이터 탐색 개요

거래id	유형	고객id	코스id	사이트	포맷	커태고려	코스(상표) 이름	거래일자	주문이름	판매가격	결제수단	실거래금액	주문항의액	거래금액	환불금액
0	2417084 PAYMENT	20053	209016	FASTCAMPUS	출인원	업무 생산성	출인원 패키지 : 김원준의 파이썬 업무자동화 유지법	2022. 12. 31. 오후 12:12:33	-	189000	TRANSFER	159000	-	159000	-
1	2415408 PAYMENT	58309	210664	FASTCAMPUS	출인원	부동산/금융	초격차 패키지 : 한 번에 끝내는 부동산 금융(P) 실무	2022. 12. 30. 오후 6:12:57	-	501000	TRANSFER	501000	-	501000	-
2	2413897 PAYMENT	18075	211782	FASTCAMPUS	출인원	영상/3D	편입하는여자의 영상편집 마스터클래스 - 제 6강 다양한 포...	2022. 12. 31. 오후 11:12:15	-	549000	CARD	549000	-	549000	-
3	2413891 PAYMENT	41898	213450	FASTCAMPUS	출인원	프로그래밍	초격차 패키지 : 95개 프로젝트로 배우는 Android 앱 개발 feat. ...	2022. 12. 31. 오후 11:12:03	[WELCOME] 프로그래밍 3만원할인	165000	CARD	135000	30000	135000	-
4	2413887 PAYMENT	94328	210773	FASTCAMPUS	출인원	프로그래밍	초격차 패키지 : 개발자 취업 합격 레스 With 코딩테스트 기...	2022. 12. 31. 오후 11:12:10	-	239000	CARD	239000	-	239000	-
5	2413886 PAYMENT	47039	210773	FASTCAMPUS	출인원	프로그래밍	초격차 패키지 : 개발자 취업 합격 레스 With 코딩테스트 기...	2022. 12. 31. 오후 11:12:58	-	239000	POINT	239000	-	239000	-
6	2413879 PAYMENT	81643	207628	FASTCAMPUS	출인원	마케팅	초격차 패키지 : 피코먼스 마케팅, 데이터로 완전 정복	2022. 12. 31. 오후 11:12:40	-	213000	CARD	213000	-	213000	-
7	2413876 PAYMENT	86273	214558	FASTCAMPUS	출인원	프로그래밍	[6이] 입문자를 위한 플스택 웹 개발 Kit : 기획부터 프로젝트...	2022. 12. 31. 오후 11:12:54	[20% 할인] 입문자를 위한 플스택 웹 개발 Kit : 기...	195000	CARD	156000	39000	156000	-
8	2413872 PAYMENT	81432	214298	FASTCAMPUS	출인원	업무 생산성	출인원 패키지 : 연신사수 피도리와 함께하는 생존 PPT 마스...	2022. 12. 31. 오후 11:12:01	-	130000	CARD	130000	-	130000	-
9	2413869 PAYMENT	21825	213315	FASTCAMPUS	출인원	알라스트	네오아카데미 : 인티의 50개의 에เจ를 통해 배우는 카주얼 게...	2022. 12. 31. 오후 11:12:51	[WELCOME] 디자인 3만원할인	188000	CARD	158000	30000	158000	-
10	2413865 PAYMENT	50713	203328	FASTCAMPUS	출인원	디자인	출인원 패키지 : 디자인에 몰려 뜬는 압축 포토샵	2022. 12. 31. 오후 11:12:26	-	134000	CARD	134000	-	134000	-
11	2413863 PAYMENT	58980	214558	FASTCAMPUS	출인원	프로그래밍	[6이] 입문자를 위한 플스택 웹 개발 Kit : 기획부터 프로젝트...	2022. 12. 31. 오후 11:12:03	-	195000	CARD	195000	-	195000	-



#### 데이터 탐색 내용

- 실습데이터 : 2022년 패스트캠퍼스 강의 거래내역 데이터
- 데이터 출처 : 패스트캠퍼스
- 데이터 유형 : csv 파일
- 데이터 품목
  - 거래id : 거래행위 고유 ID
  - **유형 : 소비자 최종 행위 (Payment / Refund)**
  - 고객id : 고객 고유 ID
  - 코스ID : 코스(상품) 고유 ID
  - 사이트 : 거래가 이루어진 사이트
  - **포맷 : 강의 유형**
  - **카테고리 : 강의 상세 카테고리**
  - 코스(상품) : 강의 이름
  - 거래일자 : 거래한 일자
  - 쿠폰이름 : 사용 된 쿠폰의 이름
  - **판매가격 : 사이트에 공시된 강의 판매 가격**
  - 결제수단 : 결제 방식
  - **실거래금액 : 기타 혜택을 사용하여 실제로 거래한 금액**
  - **쿠폰할인액 : 강의 결제시 쿠폰에 의해 할인되는 금액**
  - **거래금액 : 거래 완료된 최종 금액**
  - **환불금액 : 강의를 환불 시 환불되는 금액**
- 데이터 사이즈 : 전체 159328 rows × 16 columns 개의 데이터에서 전처리 시행

## 2.2 사용 Library

: 데이터 분석을 위해 사용한 파이썬 패키지

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

import matplotlib as mlp
import matplotlib.colors as mcl
import matplotlib.patches as mpt

import openpyxl
import time
import re

import matplotlib.font_manager as fm
from collections import Counter

from wordcloud import WordCloud
from PIL import Image
from plotnine import *
```

## 2.3 데이터셋 기본정보 파악

- 기본 데이터 셋 확인

```
# 1. 데이터 모양 확인
df.shape

# 2. 데이터 기본 정보 확인
df.info()

# 3. 데이터 기술통계적 정보 확인
df.describe()

# 4. 결측치 확인
for i in new_df.columns :
    missingValueRate = new_df[i].isna().sum() / len(new_df) * 100
    if missingValueRate > 0 :
        print("{} null rate: {}".format(i,round(missingValueRate, 2)))
```

### 1. 데이터 모양확인

Data Shape: (159328, 16)

.shape

### 2. 데이터 기본정보 확인

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 159328 entries, 0 to 159327
Data columns (total 16 columns):
#   Column              Non-Null Count  Dtype
---  -
0   거래id              159328 non-null  int64
1   유형                159328 non-null  object
2   고객id              159328 non-null  int64
3   코스ID              159328 non-null  int64
4   사이트              159328 non-null  object
5   포맷                159328 non-null  object
6   카테고리            159328 non-null  object
7   코스(상품) 이름     159328 non-null  object
8   거래일자            159328 non-null  object
9   쿠폰이름            159328 non-null  object
10  판매가격            159328 non-null  object
11  결제수단            159328 non-null  object
12  실거래금액          159328 non-null  int64
13  쿠폰할인액          159328 non-null  object
14  거래금액            159328 non-null  object
15  환불금액          159328 non-null  object
dtypes: int64(4), object(12)
memory usage: 19.4+ MB

.info()
```

3. 기술통계적 정보 확인

	거래id	고객id	코스ID	실거래금액
count	1.593280e+05	159328.000000	159328.000000	1.593280e+05
mean	1.968703e+06	50061.549903	207679.355123	1.418337e+05
std	2.516892e+05	28850.738273	3541.408820	1.397796e+05
min	1.486194e+06	0.000000	2204.000000	-1.054400e+06
25%	1.743640e+06	25081.500000	204373.000000	7.800000e+04
50%	1.971748e+06	50082.500000	207161.000000	1.620000e+05
75%	2.188854e+06	75071.000000	210796.000000	2.176000e+05
max	2.417084e+06	99999.000000	214834.000000	1.339200e+06

4. 결측치 확인

유형 null rate: 0.0%  
고객id null rate: 0.0%  
포맷 null rate: 0.0%  
카테고리 null rate: 0.0%  
코스(상품) 이름 null rate: 0.0%  
쿠폰이름 null rate: 0.0%  
판매가격 null rate: 0.0%  
결제수단 null rate: 0.0%  
실거래금액 null rate: 0.0%  
쿠폰할인액 null rate: 0.0%  
거래금액 null rate: 0.0%  
환불금액 null rate: 0.0%

.describe() & 결측치 확인

- 쿠폰 내용 wordcloud

```
img = Image.open('/content/drive/MyDrive/Colab Notebooks/data/500_500.jpg')
img_array = np.array(img)

wordcloud = WordCloud(width = 500, height = 500, font_path=FONT_PATH, random_state=21,max_font_size=120, mask=img_array, background_color='white').generate_from_frequencies(word_counts)

plt.figure(figsize=(12, 8))
plt.imshow(wordcloud, interpolation="bilinear")
plt.axis('off')
plt.show()
```



### 3. 데이터 전처리

#### 3.1 전처리 목표

💡 분석 목적에 적합한 데이터를 확보하기 위하여 불필요한 데이터를 수정 및 삭제하고, 변수 타입 재정의, 결측치 확인, 이상치(잡음) 제거 등 다양한 방법으로 데이터를 정제하였다. 이는 목적에 부합한 분석 과정과 신뢰 할 수 있는 분석 결과를 얻도록 함에 있다.

#### 3.2 전처리 개요

1. 고객의 최종 소비 행위, 즉 ‘구매확정 및 환불’에 대한 분석을 위해 변수 별 데이터를 탐색한다.
2. 분석에 필요없는 컬럼을 제거하고, 변수 내 분석 bias가 생길 row를 확인하고 제거한다.
3. 전처리 과정을 통해 분석의 전반적인 방향성을 다시 확인한다.

```
# Raw Data 전체범위 확인
# 1. 필요없는 column 제거 : 거래id, 코스ID, 사이트, 거래일자
new_df = fast_df.drop(columns=["거래id", "코스ID", "사이트", "거래일자"])

# 필요없는 column 확인
new_df[(new_df.쿠폰이름.str.contains("참고|검수|내부|조교|인턴|test|미리보기|확인"))]

# 필요없는 column 제거
new_df.drop(new_df[(new_df.쿠폰이름.str.contains("참고|검수|내부|조교|인턴|test|미리보기|확인"))].index, inplace=True)
new_df #1575행들 삭제
```

##### 3.2.1. 필요 없는 Column 제거 : 거래id, 코스ID, 사이트, 거래일자

	유형	고객 id	포맷	카테고리	코스(상품) 이름	쿠폰이름	판매가격	결제수단	실거래금액	쿠폰할인액	거래금액	환불금액
0	PAYMENT	20053	출인원	업무 생산성	출인원 패키지 : 김원손의 파이썬 업무자동화 유지권	-	189000	TRANSFER	159000	-	159000	-
1	PAYMENT	58309	출인원	부동산/금융	초격차 패키지 : 한 번에 끝내는 부동산 금융(PF) 실무	-	501000	TRANSFER	501000	-	501000	-
2	PAYMENT	18075	출인원	영상/3D	편집하는여자의 영상편집 마스터클래스 - 제 6강 다양한 효과를 응용하여 애프터이펙트...	-	549000	CARD	549000	-	549000	-
3	PAYMENT	41898	출인원	프로그래밍	초격차 패키지 : 35개 프로젝트로 배우는 Android 앱 개발 feat. Jet...	[WELCOME] 프로그래밍 3만원할인	165000	CARD	135000	30000	135000	-
4	PAYMENT	94328	출인원	프로그래밍	초격차 패키지 : 개발자 취업 합격 패스 With 코딩테스트, 기술면접	-	239000	CARD	239000	-	239000	-
...	...	...	...	...	...	...	...	...	...	...	...	...
159323	PAYMENT	39658	출인원	부업/창업	출인원 패키지 : 제 2의 월급을 만들어주는 직장인 소울 비즈니스 가이드	[NEW22] 온라인 강의 1만원 즉시 추가 할인 쿠폰	217000	CARD	207000	10000	207000	-
159324	PAYMENT	95136	출인원	영상/3D	[K] 5개월의 미디어아트 그래픽 클래스 : 모션그래픽 뉴 퍼러다임 Creative...	-	385000	CARD	385000	-	385000	-
159325	PAYMENT	12764	출인원	데이터 사이언스	출인원 패키지 : 파이썬을 활용한 데이터 전처리 Level UP	[WELCOME] 퍼캠프 처음이지? 3만원 할인쿠폰	166000	CARD	136000	30000	136000	-
159326	PAYMENT	19606	출인원	업무 생산성	출인원 패키지 : 100억을 움직이는 50가지 PPT 디자인 문지	-	289000	VBANK	289000	-	289000	-
159327	REFUND	62567	출인원	데이터 사이언스	출인원 패키지 : 마신러닝 서비스 구축을 위한 실전 MLOps	-	370000	TRANSFER	-345240	-	-	-345240

159328 rows x 12 columns





제거 근거 : 본 연구와 연관성이 부족한, 영향을 주지 않는 컬럼.

설명

1. 거래 id → 분석 목적에 부합하지 않은 변수
2. 코스 ID → 중복 데이터
3. 사이트 → 모두 '동일한' 값
4. 거래일자 → 분석 목적에 부합하지 않은 변수

3.2.2. 제거 해야 할 행 확인 → 필요 없는 Row 제거

	유형	고객 id	포맷	카테고리	코스(상품) 이름	쿠폰이름	판매가격	결제수단	실거래금액	쿠폰할인액	거래금액	환불금액
610	PAYMENT	72648	올인원	부동산/금융	올인원 패키지 : 권오상의 금융 아카데미 - 내부회계관리제도 (수강시간 제한)	[무료수강권] 강의 참고용 발행_강사님	345000	PROMOTION	0	345000	0	-
645	PAYMENT	20190	RED	데이터 사이언스	THE RED : Area Chair 이광무 교수의 Neural Fields & 3...	[무료수강권] 강의 참고용 발행_강사님	380000	PROMOTION	0	380000	0	-
654	PAYMENT	11854	올인원	부동산/금융	올인원 패키지 : 권오상의 금융 아카데미 - M&A 사례로 배우는 재무제표 영향 (...)	[무료수강권] 강의 참고용 발행_강사님	339000	PROMOTION	0	339000	0	-
856	PAYMENT	14565	RED	마케팅	THE RED : 광고인 박응현과 크리에이티브	[CX] 검수용 쿠폰_레드	192000	PROMOTION	0	192000	0	-
858	PAYMENT	5416	RED	마케팅	The RED : 나 자신과 여행하는 에세이 창작법 by. 이병률	[CX] 검수용 쿠폰_레드	158000	PROMOTION	0	158000	0	-
...	...	...	...	...	...	...	...	...	...	...	...	...
157850	PAYMENT	15732	올인원	영상/3D	애프터 이펙트 하나로 디자이너처럼 만드는 모션그래픽	[무료수강권] 강의참고용 발행_CX팀	165000	PROMOTION	0	165000	0	-
158050	PAYMENT	71868	올인원	디자인	초격차 패키지 : 100개의 클론디자인으로 배우는 UX/UI	[검수용] CX 검수용 쿠폰	203000	PROMOTION	0	203000	0	-
158187	PAYMENT	25851	올인원	디자인	초격차 패키지 : 100개의 클론디자인으로 배우는 UX/UI	[무료수강권] 강의 참고용 발행_강사님	203000	PROMOTION	0	203000	0	-
159170	PAYMENT	97255	올인원	프로그래밍	올인원 패키지 : 최성일의 코딩 초급자를 위한 웹 모션 디자인	[무료수강권] 강의 참고용 발행_지인할인쿠폰	196000	PROMOTION	0	196000	0	-
159171	PAYMENT	7800	올인원	프로그래밍	초격차 패키지 : 한 번에 끝내는 React의 모든 것	[무료수강권] 강의 참고용 발행_지인할인쿠폰	232000	PROMOTION	0	232000	0	-

1575 rows × 12 columns



- 쿠폰 이름에 ( 참고 | 검수 | 내부 | 조교 | 인턴 | test | 미리보기 | 확인 ) 이 포함되는 행 확인
- ⇒ 시스템 확인 차 직원 및 검수용으로 사용된 쿠폰
- ⇒ 이에, 본 행들은 분석에 방해가 되는 요소이기에 정확한 분석을 위해 제거
- [불필요한 1,575 rows 삭제]

3.2.3 결측치 확인 및 처리

	유형	고객 id	포맷	카테고리	코스(상품) 이름	쿠폰이름	판매가격	결제수단	실거래금액	쿠폰할인액	거래금액	환불금액
0	PAYMENT	20053	올인원	업무 생산성	올인원 패키지 : 김원순의 파이썬 업무자동화 유치원	-	189000	TRANSFER	159000	-	159000	-
1	PAYMENT	58309	올인원	부동산/금융	초격차 패키지 : 한 번에 끝내는 부동산 금융(PF) 실무	-	501000	TRANSFER	501000	-	501000	-
2	PAYMENT	18075	올인원	영상/3D	편집하는여자의 영상편집 마스터클래스 - 제 6강 다양한 효과를 응용하여 애프터이펙트...	-	549000	CARD	549000	-	549000	-
3	PAYMENT	41898	올인원	프로그래밍	초격차 패키지 : 35개 프로젝트로 배우는 Android 앱 개발 feat. Jet...	[WELCOME] 프로그래밍 3만원할인	165000	CARD	135000	30000	135000	-

	유형	고객 id	포맷	카테고리	코스(상품) 이름	쿠폰이름	판매가격	결제수단	실거래금액	쿠폰할인액	거래금액	환불금액
0	PAYMENT	20053	올인원	업무 생산성	올인원 패키지 : 김원순의 파이썬 업무자동화 유치원	No Data	189000	TRANSFER	159000	0	159000	0
1	PAYMENT	58309	올인원	부동산/금융	초격차 패키지 : 한 번에 끝내는 부동산 금융(PF) 실무	No Data	501000	TRANSFER	501000	0	501000	0
2	PAYMENT	18075	올인원	영상/3D	편집하는여자의 영상편집 마스터클래스 - 제 6강 다양한 효과를 응용하여 애프터이펙트...	No Data	549000	CARD	549000	0	549000	0
3	PAYMENT	41898	올인원	프로그래밍	초격차 패키지 : 35개 프로젝트로 배우는 Android 앱 개발 feat. Jet...	[WELCOME] 프로그래밍 3만원할인	165000	CARD	135000	30000	135000	0
4	PAYMENT	94328	올인원	프로그래밍	초격차 패키지 : 개발자 취업 합격 패스 With 코딩테스트, 기술면접	No Data	239000	CARD	239000	0	239000	0
...	...	...	...	...	...	...	...	...	...	...	...	...
159323	PAYMENT	39658	올인원	부업/창업	올인원 패키지 : 제 2의 월급을 만들어주는 직장인 스몰비즈니스 가이드	[NEW22] 온라인 강의 1만원 즉시 추가 할인 쿠폰	217000	CARD	207000	10000	207000	0
159324	PAYMENT	95136	올인원	영상/3D	[K1] 5개월의 미디어아트 그래픽 클래스 : 모션그래픽 뉴파라다임 Creative...	No Data	385000	CARD	385000	0	385000	0
159325	PAYMENT	12764	올인원	데이터 사이언스	올인원 패키지 : 파이썬을 활용한 데이터 전처리 Level UP	[WELCOME] 패캉은 처음이지? 3만원 할인쿠폰	166000	CARD	136000	30000	136000	0
159326	PAYMENT	19606	올인원	업무 생산성	올인원 패키지 : 100억을 움직이는 50가지 PPT 디자인 토크	No Data	289000	VBANK	289000	0	289000	0
159327	REFUND	62567	올인원	데이터 사이언스	올인원 패키지 : 머신러닝 서비스 구축을 위한 실전 MLOps	No Data	370000	TRANSFER	-345240	0	0	-345240

157753 rows × 12 columns

```
# 결측치 처리
new_df['쿠폰이름'].fillna('No Data', inplace = True)

cols = ['판매가격', '쿠폰할인액', '거래금액', '환불금액']
for i in cols:
    new_df[i].fillna(0, inplace = True)

#결측치 재확인
for i in new_df.columns :
    missingValueRate = new_df[i].isna().sum() / len(new_df) * 100
```

```
if missingValueRate >= 0 :
    print("{} null rate: {}".format(i,round(missingValueRate, 2)))
```

```
거래id null rate: 0.0%
유형 null rate: 0.0%
고객id null rate: 0.0%
코스ID null rate: 0.0%
사이트 null rate: 0.0%
포맷 null rate: 0.0%
카테고리 null rate: 0.0%
코스(상품) 이름 null rate: 0.0%
거래일자 null rate: 0.0%
쿠폰이름 null rate: 0.0%
판매가격 null rate: 0.0%
결제수단 null rate: 0.0%
실거래금액 null rate: 0.0%
쿠폰할인액 null rate: 0.0%
거래금액 null rate: 0.0%
환불금액 null rate: 0.0%
```

### 3.2.4 데이터타입 확인

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 157753 entries, 0 to 159327
Data columns (total 12 columns):
#   Column                Non-Null Count  Dtype
---  ---
0   유형                  157753 non-null object
1   고객id                157753 non-null int64
2   포맷                  157753 non-null object
3   카테고리              157753 non-null object
4   코스(상품) 이름      157753 non-null object
5   쿠폰이름              157753 non-null object
6   판매가격              157753 non-null object
7   결제수단              157753 non-null object
8   실거래금액            157753 non-null int64
9   쿠폰할인액            157753 non-null object
10  거래금액              157753 non-null object
11  환불금액              157753 non-null object
dtypes: int64(2), object(10)
memory usage: 15.6+ MB
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 157753 entries, 0 to 159327
Data columns (total 12 columns):
#   Column                Non-Null Count  Dtype
---  ---
0   유형                  157753 non-null object
1   고객id                157753 non-null int64
2   포맷                  157753 non-null object
3   카테고리              157753 non-null object
4   코스(상품) 이름      157753 non-null object
5   쿠폰이름              157753 non-null object
6   판매가격              157753 non-null int64
7   결제수단              157753 non-null object
8   실거래금액            157753 non-null int64
9   쿠폰할인액            157753 non-null int64
10  거래금액              157753 non-null int64
11  환불금액              157753 non-null int64
dtypes: int64(6), object(6)
memory usage: 15.6+ MB
```

```
# numeric 한 타입으로 변경하는 코드
final_df = new_df.copy()
columns_name = ['판매가격', '실거래금액', '쿠폰할인액', '거래금액', '환불금액']
for name in columns_name:
    final_df[name] = pd.to_numeric(new_df[f'{name}'], errors='coerce')

# 정보확인
final_df.info()
```

## 4. 시각화 개요

### 4.1 시각화 목적

💡 분석하고 싶은 주제에 대해 분석하고 이를 시각화하여 보다 효율적이고 직관적인 이해를 돕는다.

💡 Think

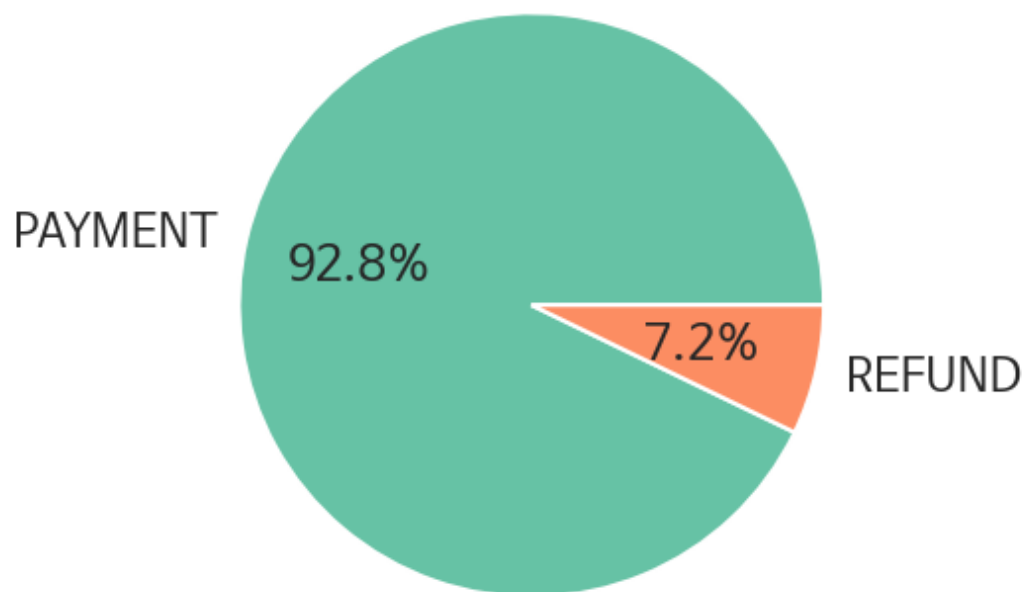
1. Payment (구매확정 행위) & Refund (환불 행위)
  - 각 명목 별로 수익 창출에 관한 인사이트를 도출한다.
  - 도출한 결과값을 시각화도구를 통해 보다 더 쉽게 파악 할 수 있도록 구현한다.
2. 세부 비교
  - Payment와 Refund에 대한 분석값을 구체화하여
  - 수익구조에 긍정적인 요인과 부정적인 요인들을 생각해보고, 시각화 한다.

### 4.2 시각화

#### 4.2.1 기본 시각화

- 소비자의 구매확정과 환불 비율

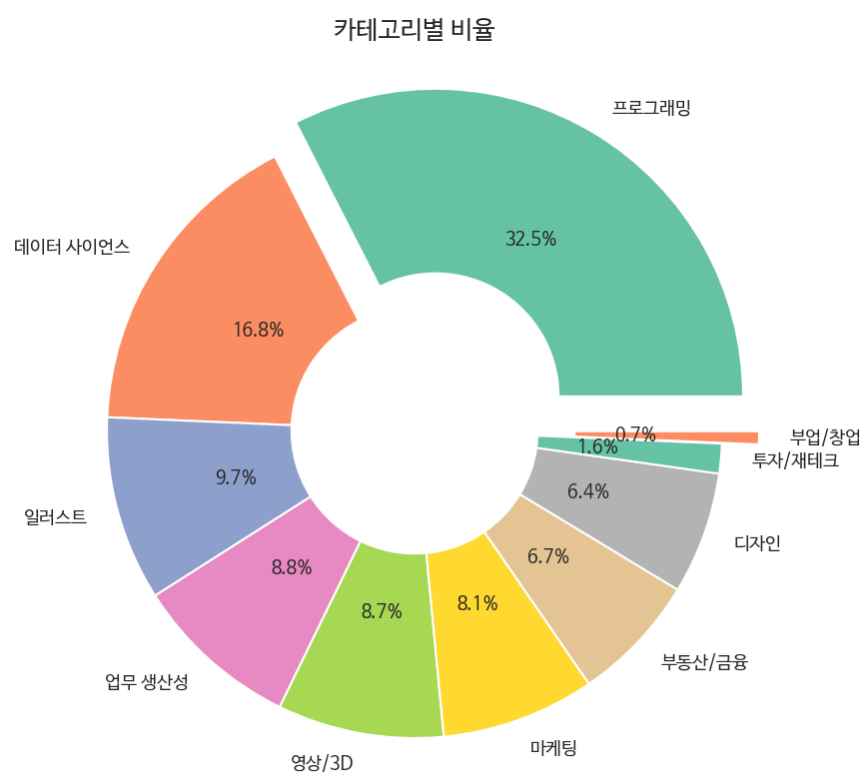
```
plt.pie(x= type_groupby_df["고객id"],labels = type_groupby_df.index, autopct='%1f%%')
plt.show()
```



- 강의 카테고리 비율

```
#카테고리 비율 확인
category_counts = final_df['카테고리'].value_counts()
category_rates = category_counts / len(final_df) * 100
category_rates = category_rates.round(3)
explode = [0.13, 0, 0, 0, 0, 0, 0, 0, 0, 0.12]

plt.figure(figsize=(10, 10))
plt.pie(category_rates, labels=category_rates.index, autopct='%1.1f%%', textprops={'fontsize': 13},
        wedgeprops={'width': 0.6, 'edgecolor': 'white'}, explode=explode)
plt.title('카테고리별 비율')
plt.show()
```



- 1) Top3 가 매출의 59% , Top5 가 매출의 76.5%
- 2) 기초 프로그래밍과 실무 적용이 가능한 업무 생산성 콘텐츠 매출 절반에 가까운 수요(41.3%)
- 3) 주력 상품을 기초 IT교육, 기초 실무 교육을 기반으로 심화 IT교육으로 Mapping 필요성 확인.

#### 4.2.2. 구매 확정(PAYMENT) 비율 알아보기

- 구매확정 테이블 확인

```
payment_groupby_df = final_df[final_df['유형'] == "PAYMENT"]
payment_groupby_df = payment_groupby_df.groupby("카테고리").count()
payment_groupby_df
```

- 카테고리 별 구매확정 건수 막대그래프

```
x = payment_groupby_df.sort_values(by="실거래금액", ascending=False).index
y = payment_groupby_df['실거래금액'].sort_values(ascending=False)

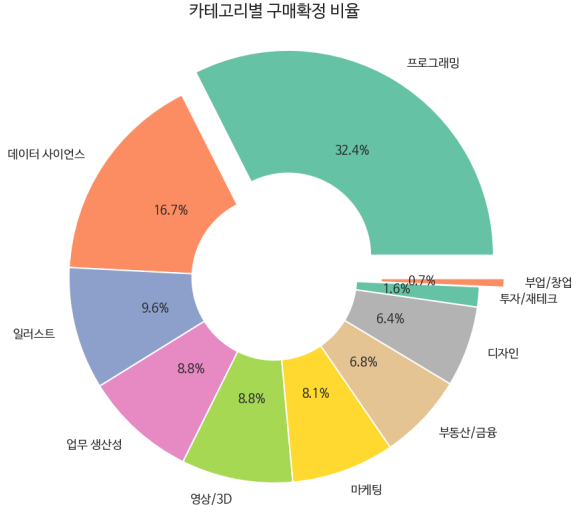
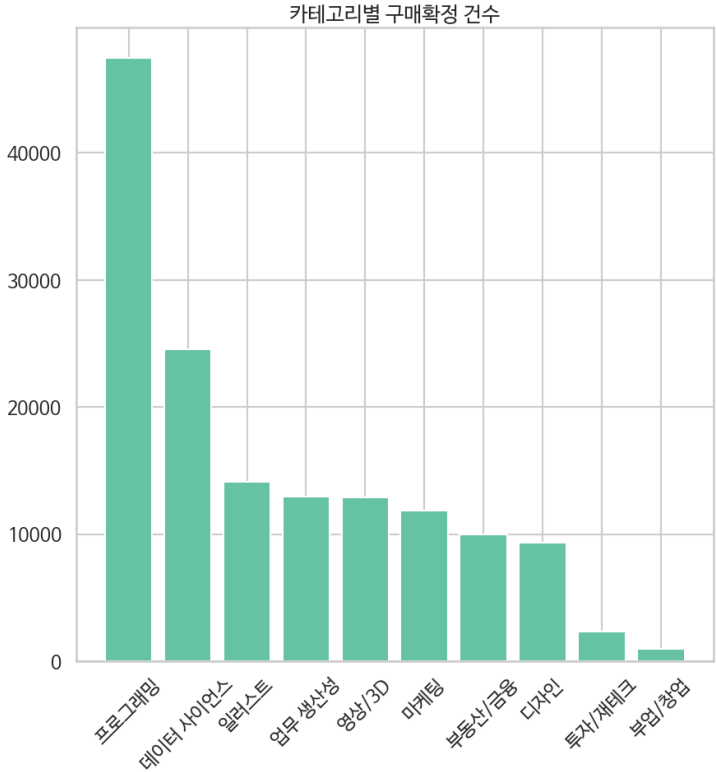
plt.figure(figsize=(10, 10))
plt.bar(x, y)
plt.xticks(rotation=45)
plt.title('카테고리별 구매확정 건수')
plt.show()
```



- 카테고리별 구매확정 비율 원형 도넛 그래프

```
explode = [0.13, 0, 0, 0, 0, 0, 0, 0, 0, 0.12]

plt.figure(figsize=(10, 10))
plt.pie(y, labels=x, autopct='%1.1f%%', textprops={'fontsize': 13},
        wedgeprops={'width': 0.6, 'edgecolor': 'white'}, explode=explode)
plt.title('카테고리별 구매확정 비율')
plt.show()
```



### 4.2.3 환불(REFUND) 비율 알아보기

- 환불테이블 확인

```
refund_groupby_df = final_df[final_df['유형'] == "REFUND"]
refund_groupby_df = refund_groupby_df.groupby("카테고리").count()
refund_groupby_df
```

	유형	고객 id	포맷	코스(상품)	이름	쿠폰이름	판매가격	결제수단	실거래금액	쿠폰할인액	거래금액	환불금액
카테 고리												
데이터 사이언스	1992	1992	1992		1992	1992	1992	1992	1992	1992	1992	1992
디자인	815	815	815		815	815	815	815	815	815	815	815
마케팅	873	873	873		873	873	873	873	873	873	873	873
부동산/금융	654	654	654		654	654	654	654	654	654	654	654
부업/창업	76	76	76		76	76	76	76	76	76	76	76
업무 생산성	864	864	864		864	864	864	864	864	864	864	864
영상/3D	942	942	942		942	942	942	942	942	942	942	942
일러스트	1185	1185	1185		1185	1185	1185	1185	1185	1185	1185	1185
투자/재테크	135	135	135		135	135	135	135	135	135	135	135
프로그래밍	3782	3782	3782		3782	3782	3782	3782	3782	3782	3782	3782

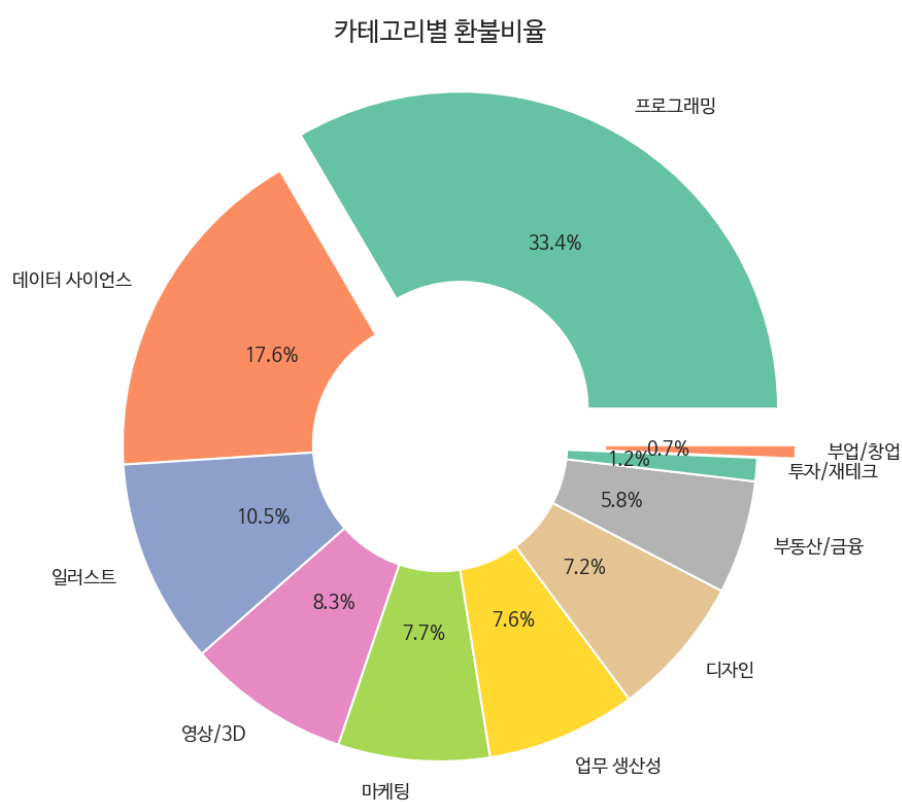
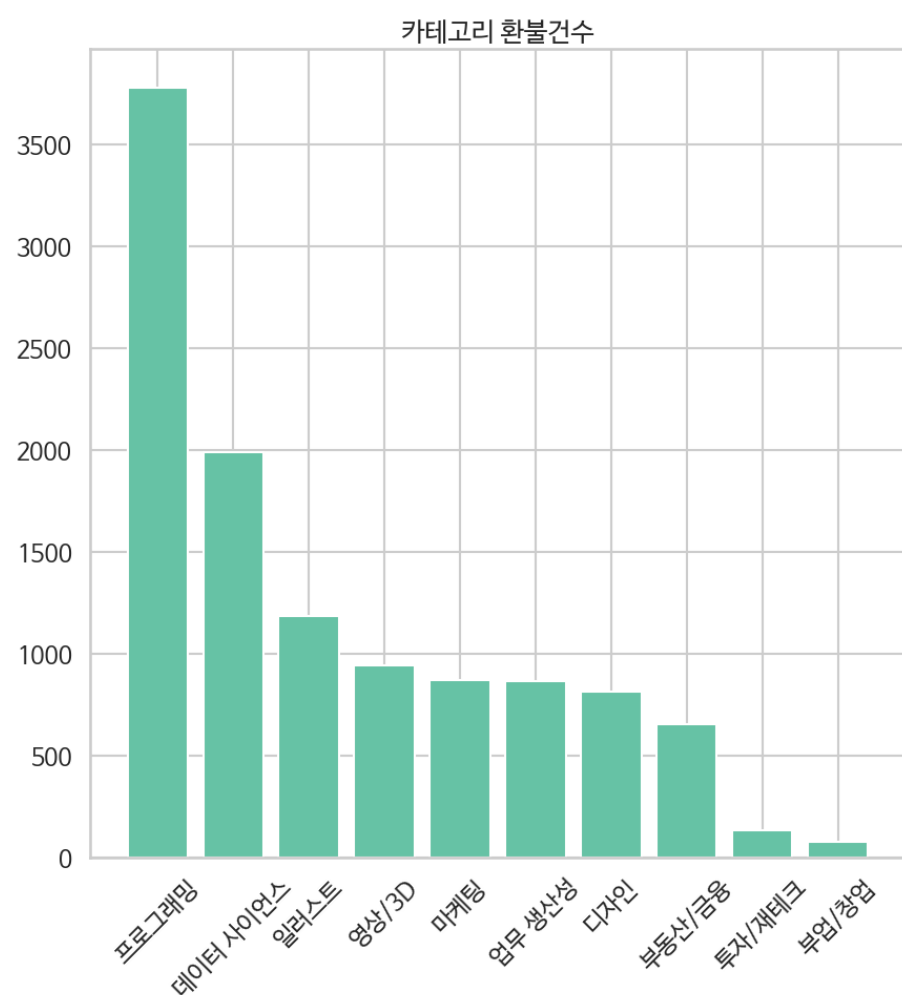
- 카테고리별 환불건수 막대그래프

```
x = refund_groupby_df.sort_values(by="실거래금액", ascending=False).index
y = refund_groupby_df['환불금액'].sort_values(ascending=False)

plt.figure(figsize=(10, 10))
plt.bar(x, y)
plt.xticks(rotation=45)
plt.title('카테고리 환불건수')
plt.show()
```

- 카테고리별 환불비율 원형 도넛그래프

```
plt.figure(figsize=(10, 10))
plt.pie(y, labels=x, autopct='%1.1f%%', textprops={'fontsize': 13},
        wedgeprops={'width': 0.6, 'edgecolor': 'white'}, explode=explode)
plt.title('카테고리별 환불비율')
plt.show()
```



#### 4.2.4 구매확정/환불 비교하기

- 구매확정 / 환불 비교할 데이터 프레임

```
groupby_df = final_df.groupby(["카테고리", "유형"]).count()
groupby_df
```

		고객 id	포맷	코스(상품)	이름	쿠폰이름	판매가격	결제수단	실거래금액	쿠폰할인액	거래금액	환불금액
카테고리	유형											
데이터 사이언스	PAYMENT	24526	24526		24526	24526	24526	24526	24526	24526	24526	24526
	REFUND	1992	1992		1992	1992	1992	1992	1992	1992	1992	1992
디자인	PAYMENT	9311	9311		9311	9311	9311	9311	9311	9311	9311	9311
	REFUND	815	815		815	815	815	815	815	815	815	815
마케팅	PAYMENT	11873	11873		11873	11873	11873	11873	11873	11873	11873	11873
	REFUND	873	873		873	873	873	873	873	873	873	873
부동산/금융	PAYMENT	9979	9979		9979	9979	9979	9979	9979	9979	9979	9979
	REFUND	654	654		654	654	654	654	654	654	654	654
부업/창업	PAYMENT	997	997		997	997	997	997	997	997	997	997
	REFUND	76	76		76	76	76	76	76	76	76	76
업무 생산성	PAYMENT	12943	12943		12943	12943	12943	12943	12943	12943	12943	12943
	REFUND	864	864		864	864	864	864	864	864	864	864
영상/3D	PAYMENT	12851	12851		12851	12851	12851	12851	12851	12851	12851	12851
	REFUND	942	942		942	942	942	942	942	942	942	942
일러스트	PAYMENT	14127	14127		14127	14127	14127	14127	14127	14127	14127	14127
	REFUND	1185	1185		1185	1185	1185	1185	1185	1185	1185	1185
투자/재테크	PAYMENT	2329	2329		2329	2329	2329	2329	2329	2329	2329	2329
	REFUND	135	135		135	135	135	135	135	135	135	135
프로그래밍	PAYMENT	47499	47499		47499	47499	47499	47499	47499	47499	47499	47499
	REFUND	3782	3782		3782	3782	3782	3782	3782	3782	3782	3782

- 더블 도넛파이로 나타낸 카테고리별 환불과 구매확정 비교 데이터셋 확인

```
#더블 도넛 파이
plt.figure(figsize=(10, 10))

#전체 강의 비율
ratio = final_df['카테고리'].value_counts().sort_values(ascending = False).values.flatten()
label = final_df['카테고리'].value_counts().sort_values(ascending = False).index
colors = plt.cm.Set3(range(len(label)))
colors_inner = ['lightsteelblue', 'firebrick', 'lightsteelblue', 'firebrick', 'lightsteelblue', 'firebrick', 'lightsteelblue', 'firebrick', 'lightsteelblue', 'firebrick',
               'lightsteelblue', 'firebrick', 'lightsteelblue', 'firebrick', 'lightsteelblue', 'firebrick', 'lightsteelblue', 'firebrick', 'lightsteelblue', 'firebrick']

explode = [0.15, 0, 0, 0, 0, 0, 0, 0, 0.09, 0.1]
```

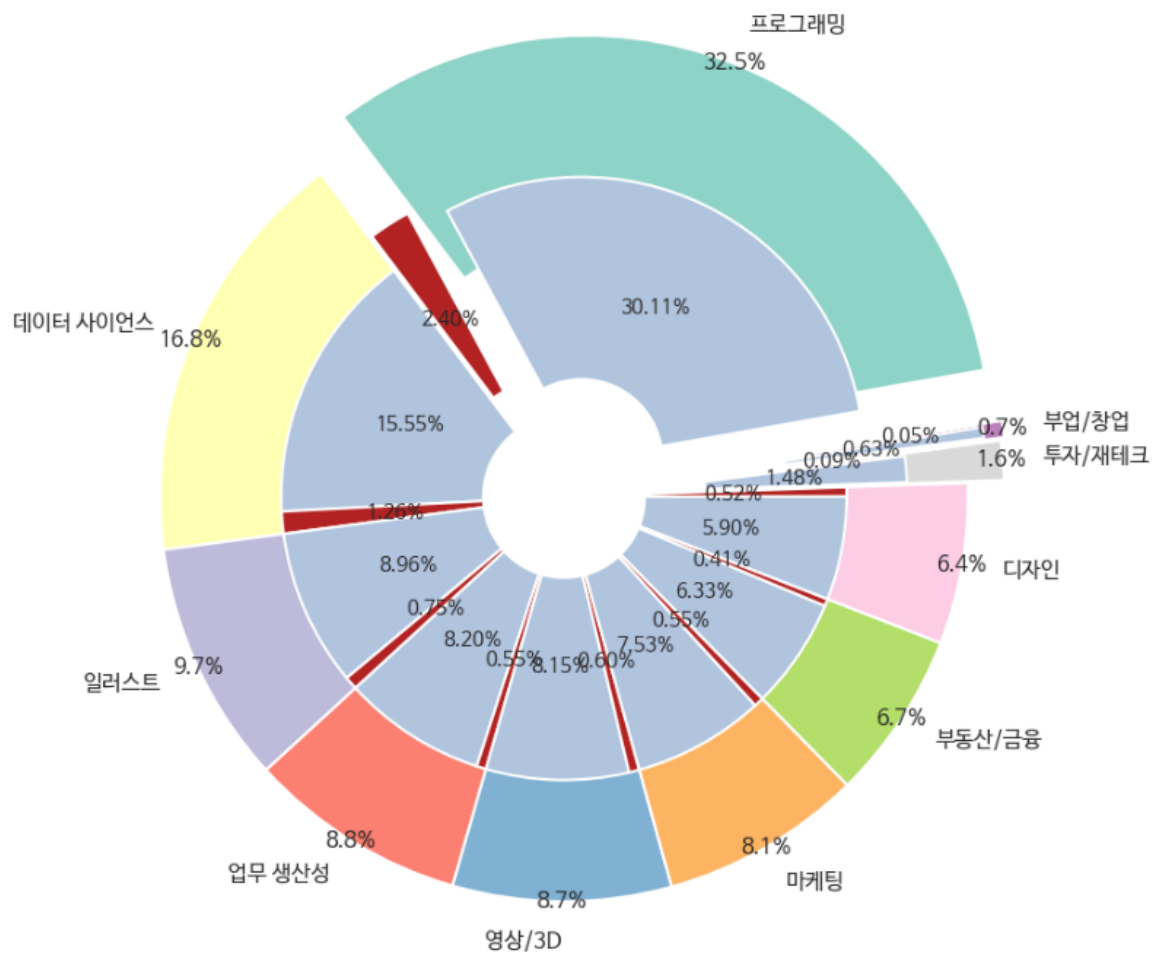
```
explode2 = [0.1, 0.1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0.15, 0.25, 0.35, 0.45]
#explode=explode

plt.pie(ratio, labels= label, colors=colors, startangle=10,frame=True,radius=1,autopct='%1.1f%%',
      pctdistance=1.,textprops={'fontsize': 12},labeldistance=1.1,
      wedgeprops={'width': 0.5, 'edgecolor': 'white'},explode=explode)

#유형의 비율 (구매확정 / 환불)
plt.pie(groupby_df['실거래금액'].values.flatten(), colors=colors_inner, autopct='%1.2f%%',
      startangle=10, radius=.7, textprops={'fontsize': 11}, labeldistance=10,
      wedgeprops={'width': 0.5, 'edgecolor': 'white'}, explode=explode2) #width :파이키기 변경, explode 특정 파이조각을 띄우는 함수.

centre_circle = plt.Circle((0,0), .2,color='black', fc='white',linewidth=0)

fig = plt.gcf()
fig.gca().add_artist(centre_circle)
plt.text(2, 1, "Blue = PAYMENT \nRed = REFUND ", bbox=dict(facecolor='white', alpha=0.5))
plt.show()
```



Blue = PAYMENT  
Red = REFUND

# 기초 지식 필요로 소비자들 내에서 개인의 역량 부족  
# 또는 소비자가 실질적으로 원하는 '심화 과정'의 trageting miss  
데이터 사이언스/일러스트/ 디자인 92% → 92% 구매 / 8% 환불

# 진입 장벽이 다소 쉬운 과목  
프로그래밍 / 영상/3D / 마케팅 /부업/창업 → 93% 구매 / 7% 환불

# 실무 적용이 용이하고 , 실생활 내에서 체감이 높은 과목  
업무 생산성 / 부동산/금융 → 94% 구매 / 6% 환불

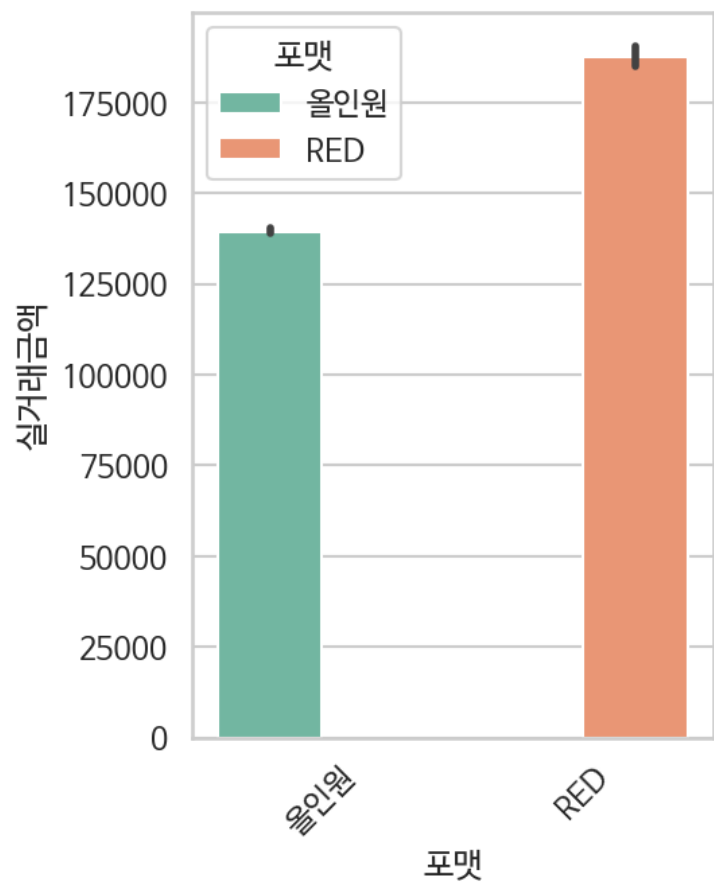
#강의 수 부족  
투자/ 재테크 → 95% 구매 /5% 환불

#### 4.2.5 강의 Format에 대한 분석

- 강의금액 분포 분석

```
# 강의 포맷에 따른 분석.
plt.figure(figsize=(5, 7))
sns.barplot(x="포맷", y="실거래금액", hue="포맷", data=final_df)
plt.xticks(rotation=45)
plt.show()

## RED 단과수업이 높은 단가의 강의가 많다.
```



- 강의 Format 에 따른 구매확정/환불 비율

```
plt.figure(figsize=(10, 10))
colors = ['#ff6666', '#ffcc99', '#99ff99', '#66b3ff']
colors_inner = ['lightskyblue', 'r', 'lightskyblue', 'r', 'lightskyblue', 'r', 'lightskyblue', 'r', 'lightskyblue', 'r', 'lightskyblue', 'r', 'lightskyblue', 'r', 'lightskyblue', 'r']

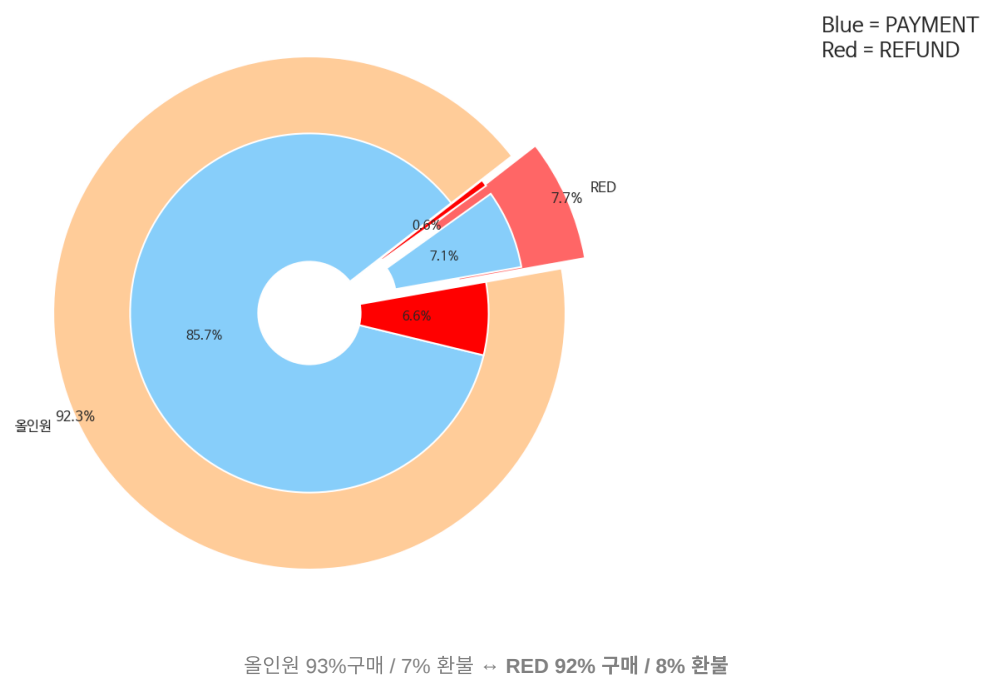
explode = [0.1, 0]
explode2 = [0.15, 0.15, 0, 0]

#포맷 비율
ratio = final_df['포맷'].value_counts().sort_values().values.flatten()
label = final_df['포맷'].value_counts().sort_values().index
plt.pie(ratio, labels= label, colors=colors, startangle=10, frame=True, radius=1, autopct='%1.1f%%',
        pctdistance=1, textprops={'fontsize': 12}, labeldistance=1.1,
        wedgeprops={'width': 0.5, 'edgecolor': 'white'}, explode = explode)

#유형의 비율 (구매확정 / 환불)
plt.pie(type_df['실거래금액'].values.flatten(), colors=colors_inner, autopct='%1.1f%%',
        startangle=10, radius=.7, textprops={'fontsize': 11}, labeldistance=10,
        wedgeprops={'width': 0.5, 'edgecolor': 'white'}, explode = explode2) #width :파이크기 변경, explode 특정 파이조각을 띄우는 함수.

centre_circle = plt.Circle((0,0), .2,color='black', fc='white',linewidth=0)

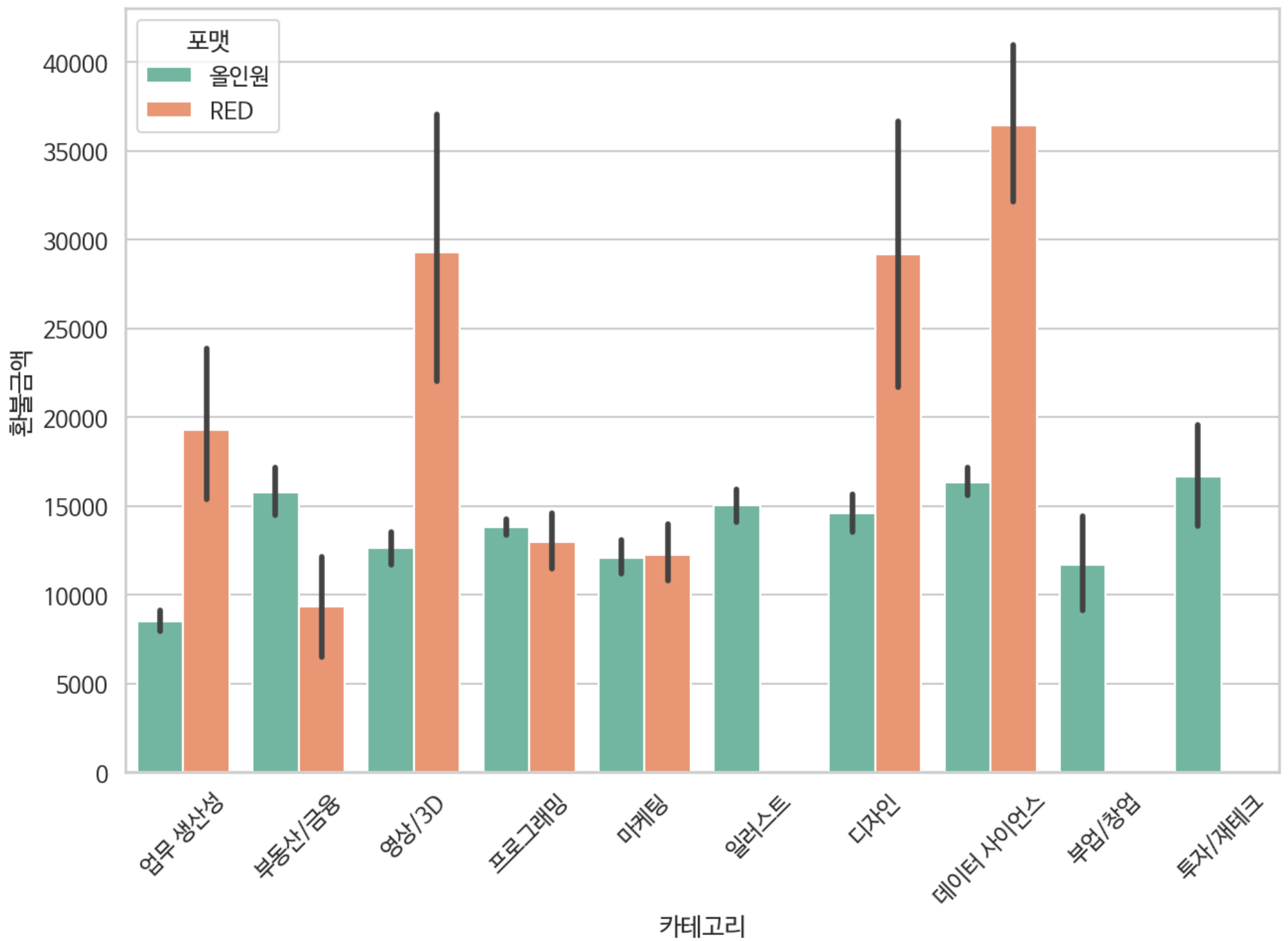
fig = plt.gcf()
fig.gca().add_artist(centre_circle)
plt.text(2, 1, "Blue = PAYMENT \nRed = REFUND ", bbox=dict(facecolor='white', alpha=0.5))
plt.show()
```



- 강의 카테고리별 강의 Format 분포

```
plt.figure(figsize=(15, 10))
sns.barplot(x=final_df["카테고리"], y=final_df["환불금액"].abs(), hue=final_df["포맷"])
plt.xticks(rotation=45)
plt.show()

##단과에 환불이 더 금액이 많다.
```



참고) RED 내 카테고리 별 강의 수:  
프로그래밍 RED 17개  
데이터 사이언스 RED 15개  
마케팅 RED 9개  
영상/3D RED 6개  
업무생산성 RED 4개  
디자인 RED 3개  
-2023-05-16일 기준-

- 카테고리 별 Format 비율

```
#단과 카테고리 더블 도넛차트

ratio = final_df['카테고리'].value_counts().sort_values(ascending = False).values.flatten()
label = final_df['카테고리'].value_counts().sort_values(ascending = False).index
#더블 도넛 파이
plt.figure(figsize=(10, 10))
colors = plt.cm.Set3(range(len(label)))
colors_inner = ['firebrick', 'lightsteelblue', 'firebrick', 'lightsteelblue', 'lightsteelblue', 'firebrick', 'lightsteelblue', 'firebrick',
               'lightsteelblue', 'firebrick', 'lightsteelblue', 'firebrick', 'lightsteelblue', 'lightsteelblue', 'lightsteelblue']

explode = [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0.1]
explode2 = [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0.1, 0.2, 0.3]

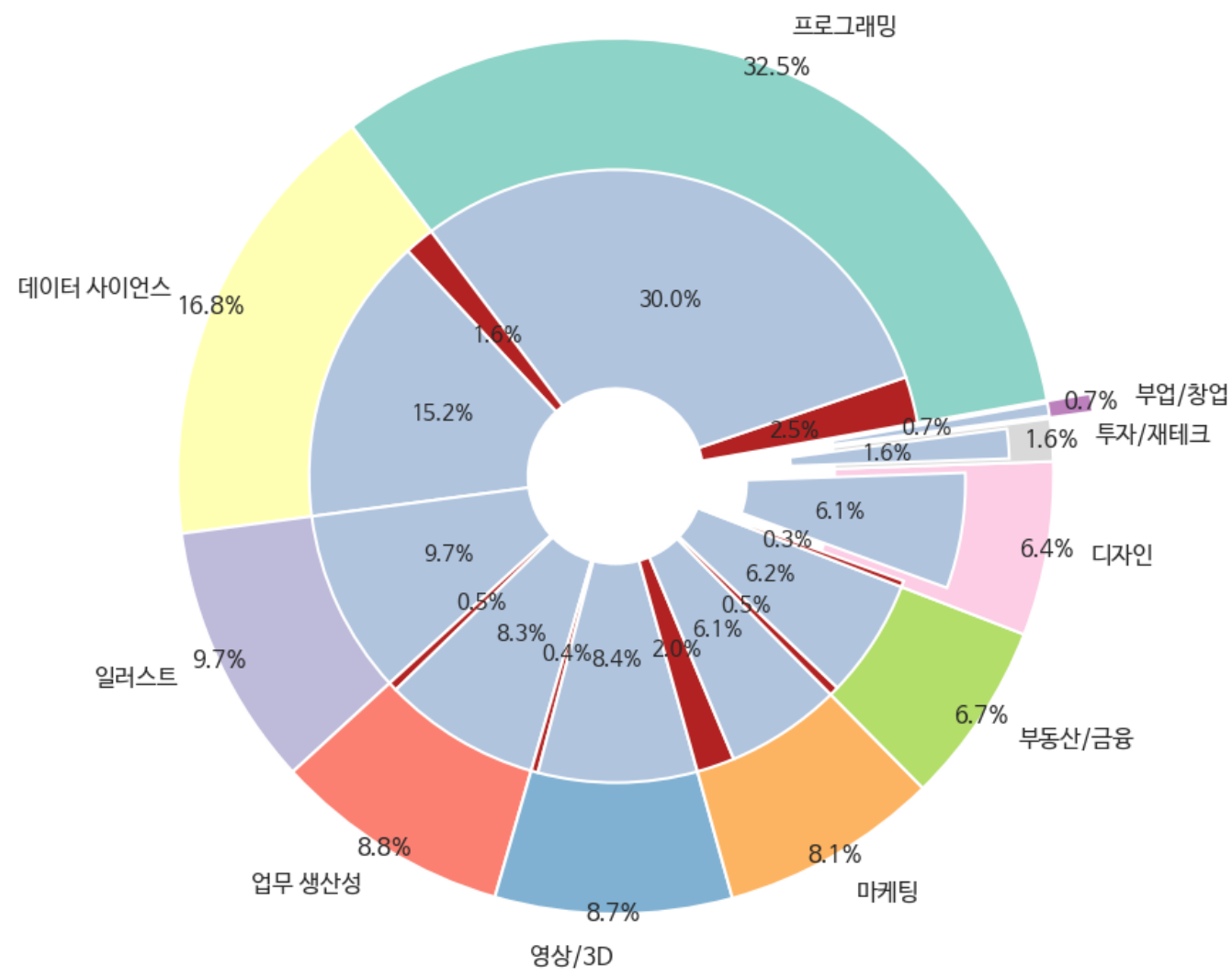
#전체 강의 비율
plt.pie(ratio, labels= label, colors=colors, startangle=10, frame=True, radius=1, autopct='%1.1f%%',
        pctdistance=1., textprops={'fontsize': 12}, labeldistance=1.1,
        wedgeprops={'width': 0.5, 'edgecolor': 'white'}, explode=explode)

#유형의 비율 (완강 / 환불)
plt.pie(type_df2['실거래금액'].values.flatten() , colors=colors_inner, autopct='%1.1f%%',
        startangle=10, radius=.7, textprops={'fontsize': 11}, labeldistance=10,
        wedgeprops={'width': 0.5, 'edgecolor': 'white'}, explode=explode2) #width :파이크기 변경, explode 특정 파이조각을 띄우는 함수.

centre_circle = plt.Circle((0,0), .2,color='black', fc='white',linewidth=0)

fig = plt.gcf()
fig.gca().add_artist(centre_circle)
plt.text(2, 1, "Red = RED \nBlue = 올인원 ", bbox=dict(facecolor='white', alpha=0.5))
plt.show()
```

Red = RED  
Blue = 올인원



#### 4.2.6 환불에 대한 인사이트 도출

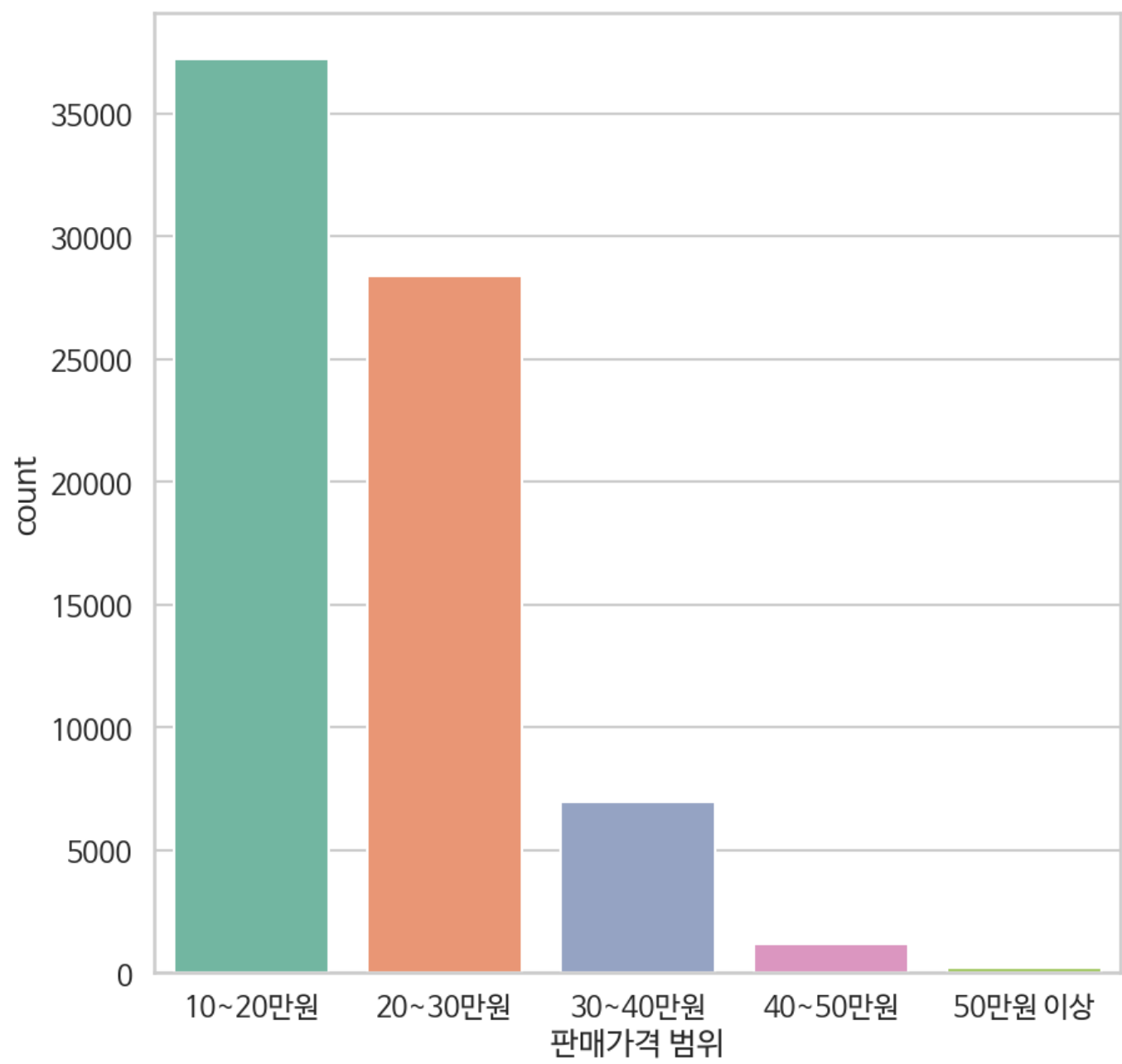
- 쿠폰 사용 유무를 기준으로 데이터의 판매가격 별 범위를 분류
- 쿠폰 사용한 데이터들에 한해 판매가격 범위 수를 분류한 막대그래프

```
plt.figure(figsize=(10, 10))
sns.despine()
# 쿠폰을 사용한 행들만 데이터프레임으로 만들
df_coupon = final_df[final_df["쿠폰이름"] != "No Data"]
df_coupon

# 데이터프레임에 판매가격을 통해 새로운 판매가격 범위 열을 만들
df_coupon.loc[df_coupon["판매가격"] < 100000, "판매가격 범위"] = "10만원 이하"
df_coupon.loc[(df_coupon["판매가격"] >= 100000) & (df_coupon["판매가격"] < 200000), "판매가격 범위"] = "10~20만원"
df_coupon.loc[(df_coupon["판매가격"] >= 200000) & (df_coupon["판매가격"] < 300000), "판매가격 범위"] = "20~30만원"
df_coupon.loc[(df_coupon["판매가격"] >= 300000) & (df_coupon["판매가격"] < 400000), "판매가격 범위"] = "30~40만원"
df_coupon.loc[(df_coupon["판매가격"] >= 400000) & (df_coupon["판매가격"] < 500000), "판매가격 범위"] = "40~50만원"
df_coupon.loc[(df_coupon["판매가격"] >= 500000), "판매가격 범위"] = "50만원 이상"

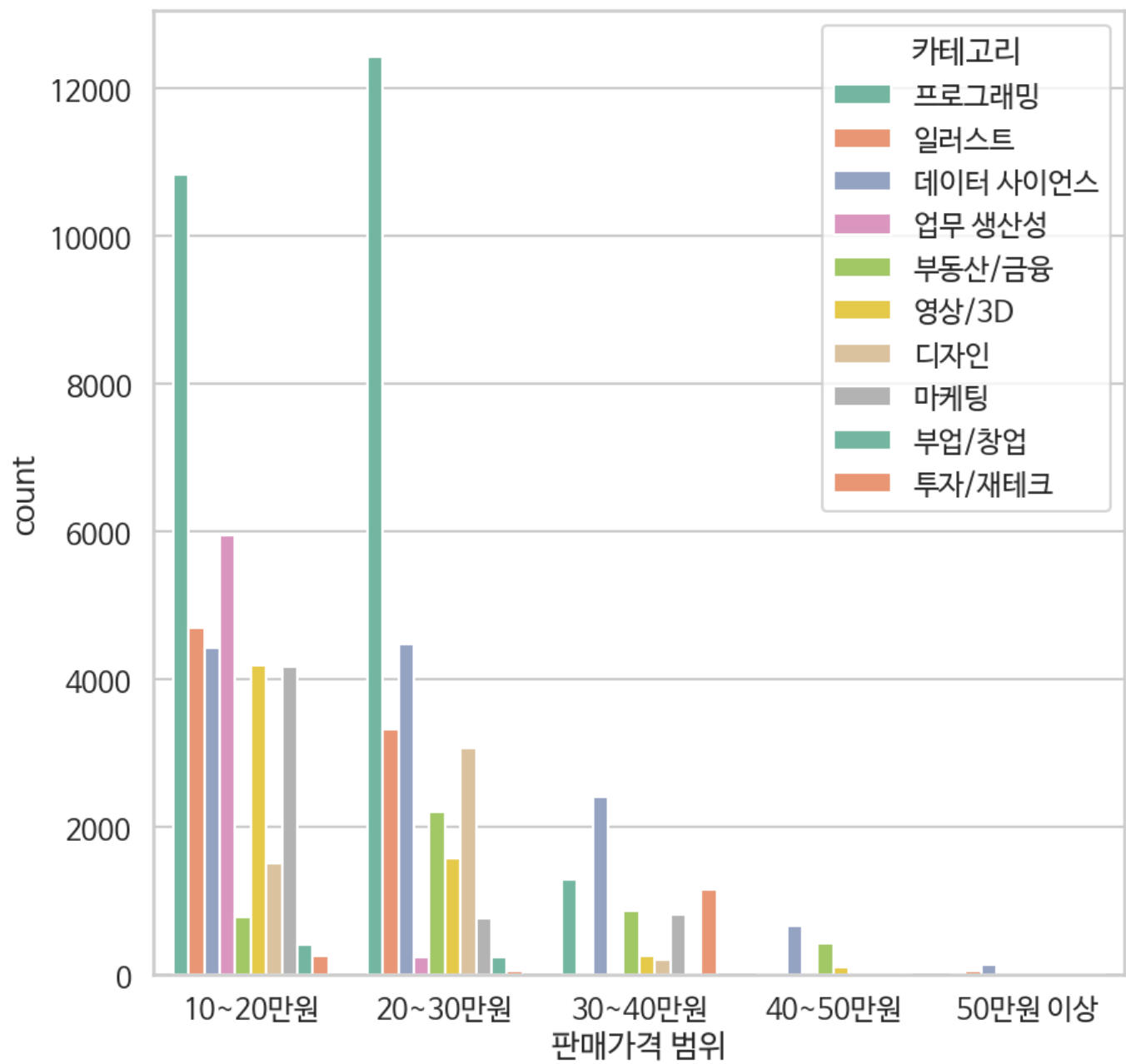
#쿠폰이 사용된 데이터들에 판매가격 범위를 countplot 그래프로 그림
sns.countplot(data=df_coupon, x="판매가격 범위", order = sale_price_order)
plt.show()
```





- 카테고리 별 판매가격 범위 막대그래프

```
plt.figure(figsize=(10, 10))
sns.countplot(data=df_coupon, x="판매가격 범위", order=sale_price_order, hue='카테고리', palette='Set2')
plt.show()
```



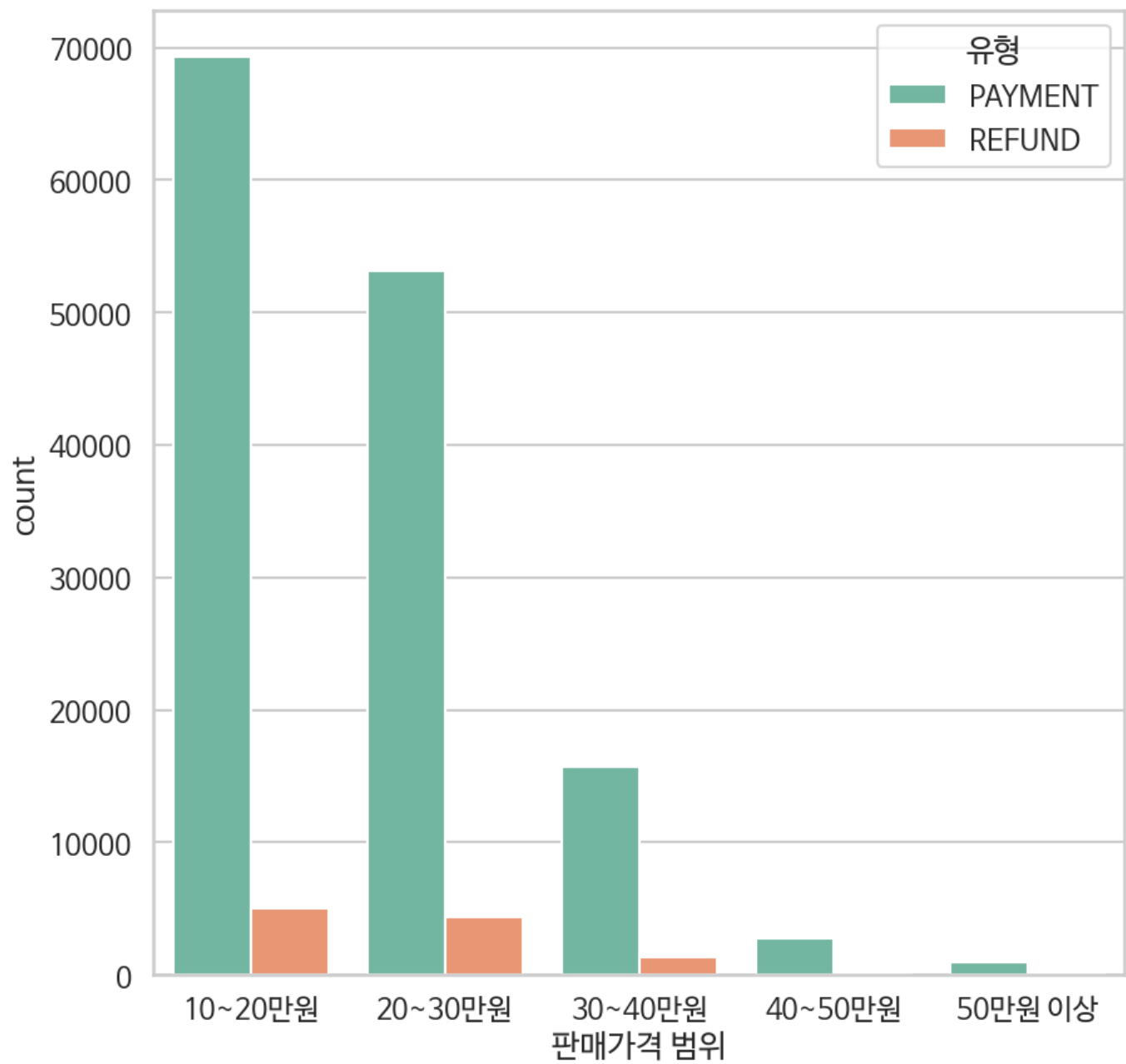
대부분 10~30만원대에서 프로그래밍이 전체 매출의 32.5%를 낸다는 것을 알 수 있다.  
 데이터 사이언스는 가격 범위 전체 퍼져있으며, 수준 별로 높은 강의료를 통해 매출을 창출한다.  
 일러스트는 10~30만원대에서 그리고 50만원대에서 하이엔드 강의로서 매출을 낸다.  
 -이는 RED가 아니어도 매출창출이 가능함을 보여준다-  
 업무생산성이 만족도가 높은것을 토대로 너무 높은 단가는 형성하지 않으나 환불이 적어 매출이 좋은편으로 나온다.

#### • 판매가격 범위 환불

```
plt.figure(figsize=(10, 10))
sns.despine()

#전체 데이터 프레임에서 판매가격을 통해 판매 가격 범위를 만들
final_df.loc[final_df["판매가격"] < 100000, "판매가격 범위"] = "10만원 이하"
final_df.loc[(final_df["판매가격"] >= 100000) & (final_df["판매가격"] < 200000), "판매가격 범위"] = "10~20만원"
final_df.loc[(final_df["판매가격"] >= 200000) & (final_df["판매가격"] < 300000), "판매가격 범위"] = "20~30만원"
final_df.loc[(final_df["판매가격"] >= 300000) & (final_df["판매가격"] < 400000), "판매가격 범위"] = "30~40만원"
final_df.loc[(final_df["판매가격"] >= 400000) & (final_df["판매가격"] < 500000), "판매가격 범위"] = "40~50만원"
final_df.loc[(final_df["판매가격"] >= 500000), "판매가격 범위"] = "50만원 이상"

#판매가격 범위를 hue를 통해 PAYMENT와 REFUND로 나눔
#데이터프레임을 판매가격 범위에 대해 countplot을 그림
sns.countplot(data=final_df, x="판매가격 범위", order=sale_price_order, hue="유형")
plt.show()
```



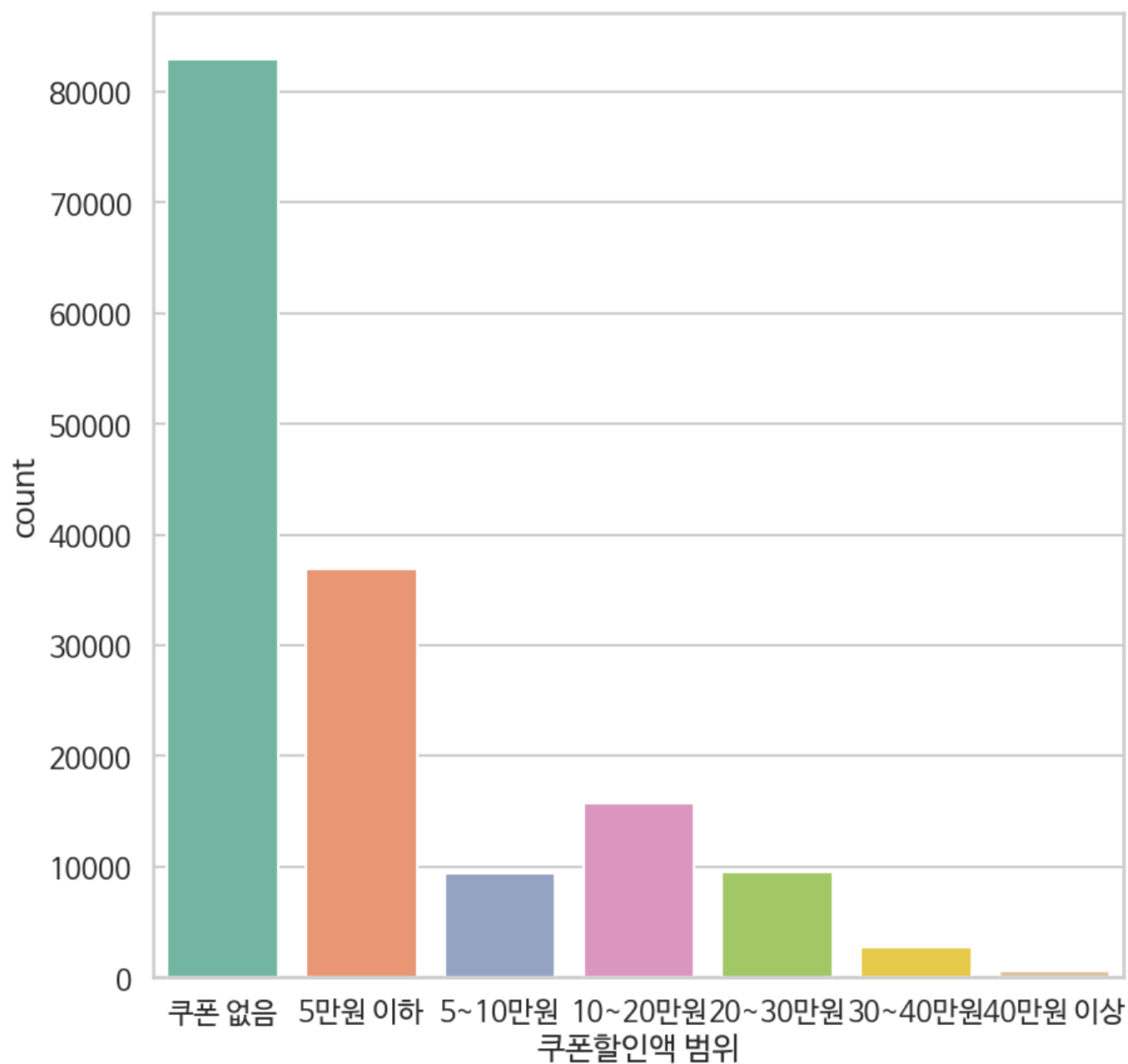
수업단가가 높을 수록, 환불 수도 적다 → 만족도 내지는 정책

#### • 쿠폰 할인액 범위 그래프

```
plt.figure(figsize=(10, 10))
#쿠폰 할인액이 int 타입인 데이터프레임에 쿠폰할인액 범위 컬럼을 추가함

final_df["쿠폰할인액 범위"] = "쿠폰 없음"
final_df.loc[(final_df["쿠폰할인액"] > 0) & (final_df["쿠폰할인액"] < 50000), "쿠폰할인액 범위"] = "5만원 이하"
final_df.loc[(final_df["쿠폰할인액"] >= 50000) & (final_df["쿠폰할인액"] < 100000), "쿠폰할인액 범위"] = "5~10만원"
final_df.loc[(final_df["쿠폰할인액"] >= 100000) & (final_df["쿠폰할인액"] < 200000), "쿠폰할인액 범위"] = "10~20만원"
final_df.loc[(final_df["쿠폰할인액"] >= 200000) & (final_df["쿠폰할인액"] < 300000), "쿠폰할인액 범위"] = "20~30만원"
final_df.loc[(final_df["쿠폰할인액"] >= 300000) & (final_df["쿠폰할인액"] < 400000), "쿠폰할인액 범위"] = "30~40만원"
final_df.loc[(final_df["쿠폰할인액"] >= 400000), "쿠폰할인액 범위"] = "40만원 이상"

#전체 데이터에서 쿠폰할인액 범위에 대한 countplot을 그림
sns.despine()
sns.countplot(data=final_df, x="쿠폰할인액 범위", order = coupon_order ) #상단에서 이미 객체 선언함.
plt.show()
```



First, 가장 기본적인 5만원 이하의 할인부터 하여금 간단한 강의를 맛보게 하며 수강을 유도한다.

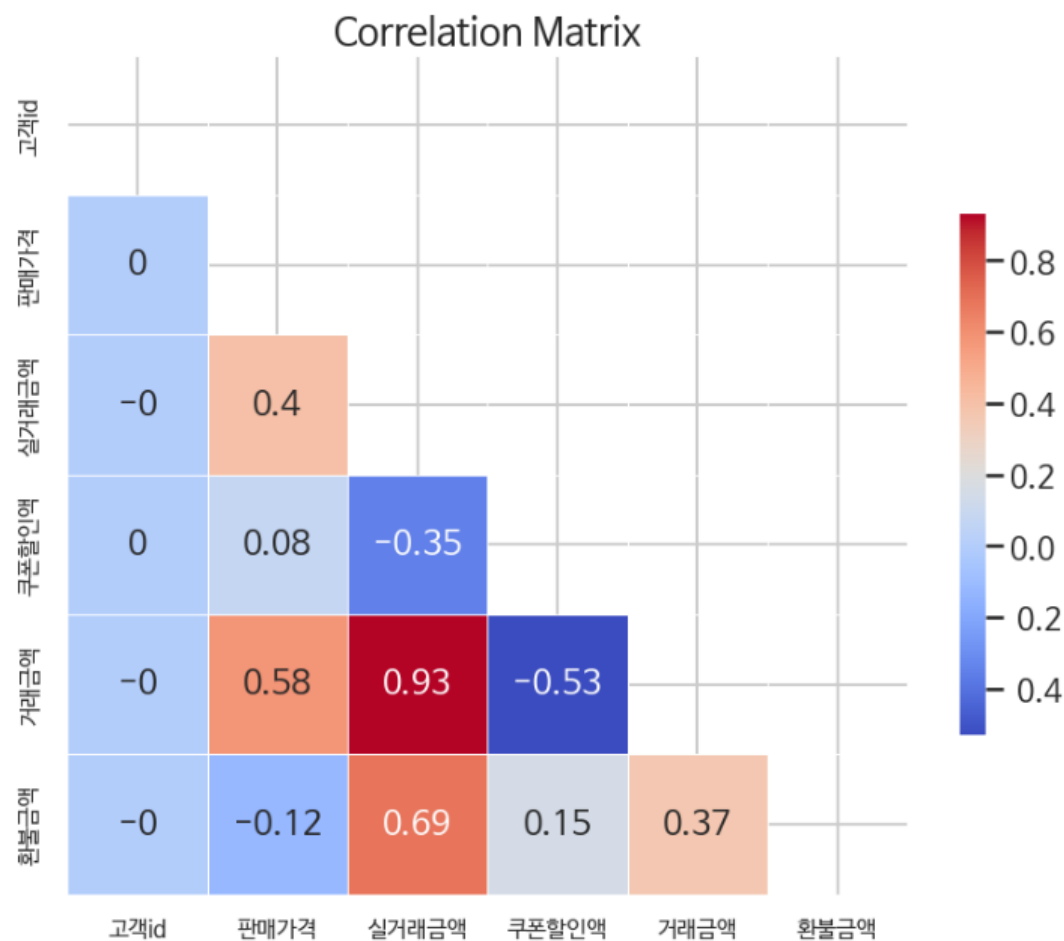
Second, 많은 10~20만원의 강의 20~30만원의 할인을 통하여 10~20만원 전체 무료 강의로 수강의 기회를 주거나, 10~20할인 이라면 20~30단가의 강의를 한 단계 낮춘 부담 없는 가격에 수강하게 하는 전략이 있어도 좋을 것이다

## 4.2.7 상관관계 분석

- correlation Matrix square 히트맵

```
final_corr = final_df.corr(numeric_only=True).round(2)
final_corr

plt.figure(figsize=(10, 10))
mask = np.triu(np.ones_like(final_corr, dtype=np.bool))
sns.heatmap(final_df.corr(numeric_only=True).round(2),annot=True, linewidths=.5, square=True, cbar_kws={"shrink": .5},cmap='coolwarm',mask = mask);
plt.title('Correlation Matrix', fontsize=20)
plt.xticks(fontsize=12)
plt.yticks(fontsize=12)
plt.show()
```



## 5. 마무리

### 5.1 분석 결과

- 상위 TOP3 (프로그래밍, 데이터사이언스, 일러스트) 카테고리가 매출의 59%를 책임진다.
- 상위 TOP5 (프로그래밍, 데이터사이언스, 일러스트, 업무생산성, 영상/3D) 카테고리가 매출의 약 77% 정도를 점유한다.
- 즉, 소비자의 큰 니즈는 '수준 높은 IT 강의, 실제 업무에 반영 가능한 업무 생산성 강의, 여타 기술 카테고리의 강의'이다.
- 범용적이거나 낮은 단가의 강의에 대해서는, 쿠폰을 이용하더라도 소비자들이 가치를 크게 느끼지 못하는 경우가 많다. 반면, 높은 가격의 경우, 강의 환불 요청은 감소하는 반비례 경향을 보이는 것으로 강의 퀄리티와 밀도에 따라 만족도가 이어지는 것으로 보인다.
- 그러나 이러한 흐름과 상반되게, 실질적으로 패스트캠퍼스 내 고 퀄리티 프리미엄 'RED' 강의에 대해서는 가격 대비 환불비 가 추세에 비해 더 높아, 만족도는 그렇게 높지 않다는 것이다.

### 5.2 방안 제시

- 수익 강화를 위해 High Quality 수업인 'RED'에서 환불이 적은 업무생산성, 부동산-금융과 같은 현장 업무에 바로 적용 가능한 수업을 많이 배치해야 한다.
- 또한, 'RED'에서 디자인과 같이 환불이 높은 수업을 줄이고 수요 자체가 많은 IT수업(프로그래밍 등)에 대해서 강의 목적과 타겟이 명확하도록 제작하는 등의 리빌딩이 필요하다.
- 그리고 'RED' 강의 영역에서 고정 소비자를 유지시켜 수익을 최대화 시키려면, ALL-IN-ONE에서 부터 RED까지 한 맥락으로 이어질 수 있는 세부적이고 구체적인 빌드업 기획이 필요하다.
- DS 강의의 경우, 소비는 많지만 구매확정으로 이어지지 않고 환불 비율이 높은 클래스이다. 즉, 소비자의 니즈를 제대로 파악하지 못하는 것으로 판단된다. 이에 좀 더 강의의 목적이 구체화되고 탄탄한 커리큘럼 방향성을 새로 제시 할 필요가 있다.
- 강의 내용의 목적성과 방향성이 세분화 되고 가격과 수업 시수가 무겁지 않은 즉 '허들이 낮은' 강의의 리스트업이 추가적으로 필요하다. 즉, 다양한 카테고리에 세부화 된 단과과목의 로드맵이 필요하다.
- 기간 구독제나 패스형의 상품을 기획할 경우 다양한 소비자들의 니즈를 충족할 수 있는 대안이 될 수 있다. 이에 수익화 증대에 긍정적인 효과를 낼 것이다. 단, 수익 구조를 해치지 않기 위해 특정 프리미엄 강의(RED) 나 인기 카테고리 별 강의는 제한 할 필요가 있다.

### 5.3 한계점

- 정제가 잘 된 데이터가 주어져 구매와 환불의 인과관계나, 환불에 영향을 미치는 파생변수를 다양하게 만들 수 있을거라 생각했지만 실질적으로 분석과정에 있어서 다양한 파생변수를 만들기에는 어렵다 판단했다. 변수 하나하나의 특징을 객관화 하기엔 설명이 부족해 오류가 발생할 위험성이 있었기 때문이다. 이에 기대했던 변수 간 인과관계를 도출 하지 못하고 상관관계 분석에만 그치는 한계가 있어서 아쉬웠다.
- 좀 더 다양한 시계열 데이터나, 소비자의 특징을 담은 변수가 있었다더라면 (나이, 성별, 직종 등의 기본정보) 더 디테일하게 분석이 가능하여 실질적으로 수익구조에 도움이 되는 요인들과 리스크들에 대해 심도깊은 분석이 가능할 수 있지 않았을까 하는 아쉬움이 있었다.

## 6. 피드백

### 6.1 분석을 통해 느낀 점

- 김민주 : 데이터 탐색과 분석 방향 설정 당시에는 다양한 주제와 소스가 나왔지만 EDA과정을 통해서 처음 설정한 연구 설계를 구현하지 못함이 (고객군 군집분석, 여러 변수와 소비자의 소비행위 간 인과관계 증명 등) 아쉬웠습니다. 보다 객관적이고 구체화 된 기준을 통해 데이터가 정의 되었다면 하는 아쉬움이 있지만, 데이터 수집과 전처리 과정이 분석의 기반이 됨을 확인할 수 있었습니다.
- 배태양 : 간단한 데이터셋 탐색시 분석과 도출결과의 다양성에 대해 고민했지만, 실질적으로 EDA 과정을 거치면서 분석에 유의미하고 다양한 결과를 낼 수 있구나. 라고 생각하게 되었습니다. 다만 아쉬운 점은 객관화가 용이하고 넓은 범위의 다양한 데이터가 주어졌더라면 보다 흥미로운 주제와 그 분석을 시행 할 수 있지 않았을 까 합니다.
- 박수영 : 정제가 잘 된 데이터를 받아 너무 편하게 작업하였습니다. 데이터를 탐색하고 분석하면서, 데이터를 하나씩 파이프라인으로 어떻게 구해오는가? 어떻게 마스킹을 하여 보내준 것일까? 경쟁사 데이터는 확보하는가? 궁금합니다.
- 박사무엘 : EDA 초기엔 다소 어렵게 느껴졌으나, 팀원들과 같이 분석하고 가설하고 짜는 부분에서 흥미를 느꼈고, 좀 더 깊게 생각하는 사고력 향상에 도움이 되었습니다. 코딩부분에선 조원들의 같은 로직임에도 다른 방식의 코드 구현을 보며 코딩에 대해 많이 배웠습니다. 이에 더욱 코딩에도 정진해야겠다 생각했습니다. 마지막으로, 내가 짠 로직이 정말로 맞는 로직인지 너무 주관적인건 아닌지 검증하는 부분이 어렵게 느껴졌습니다.
- 조인철: 특정 회사의 데이터를 활용한 EDA를 진행하여 분석에 보다 흥미를 느꼈습니다. 본 프로젝트를 통해 분석에 대한 실력과 감각을 배울 수 있었고, 다시 한번 위와 같은 프로젝트를 진행하여 보다 밀도있고 수준 높은 분석을 구현하고 싶다는 생각을 하였습니다.
- 이동근 : 길지 않은 기간의 데이터지만 분석에 필요한 다양한 결과 값을 얻을 수 있었습니다. 다만 분석 방향과 다르게 결과 값을 도출하는 데에 생각 외로 발생하는 변수가 많아 데이터 분석에 필요한 전처리 과정에서 분석 초기과정이 중요하다는 것을 느꼈습니다.

## 6.2 최종소감

- 김민주 : 프로젝트를 하면서 다른 팀원들 보다 부족한 코딩 역량을 체감하게 되어 제 자신의 실력에 대해 고찰 할 시간이 되었던 것 같습니다. 그러나 좋은 팀원들과 함께하며 브레인스토밍을 통한 분석계획, 코딩과 인사이트 도출 능력 등 많은 것들을 배울 수 있었으며, 역할 분담 안에서 서로 협업하며 각자가 잘 할 수 있는 부분을 집중하는 등 팀 프로젝트가 어떠한 작업인지 체감할 수 있었던 것 같습니다. 본 프로젝트를 통해 큰 자극을 받았고, 앞으로의 공부 방향성을 잡게 되어 소중한 시간이었습니다.
- 박사무엘 : 좋은 조원들을 만난 것 같고, 같이 작업하는데 있어서 역할분담이나 다툼이나 없었던것에 감사합니다. 또한, 프로젝트를 통해서 동기간에 끈끈함을 느낄 수 도 있는 것이 팀워크를 경험하는 데에 좋은 기회였던 것 같습니다. 프로젝트 결과 여부와 상관없이, 모두가 다같이 한 것이므로 이 프로젝트에 대해 소중함을 느낍니다.
- 배태양: 첫 데이터분석 프로젝트에서 제가 할 수있는 부분이 어떤 부분인지 느끼고 다른 팀원들의 능력을 보면서 많이 배우고 자극이 많이 되었습니다. 생각했던 것 보다 제가 할 수 있는 기술적 능력이 많이 부족함을 느낍니다. 그러나 본 프로젝트를 수행하며 부족한 부분을 채울 만한 리소스를 찾았고, 보다 열심히 수학할 수 있는 원동력과 큰 자극점이 된 것 같습니다.
- 박수영 : EDA를 진행하면서 이 과정이 데이터의 품질 향상, 깊은 이해, 그리고 더욱 효과적인 분석을 위한 기반을 마련하는 단계라는 것을 확실히 경험하였습니다. 하지만 갑자기 떠오른 인사이트를 쫓다가, 주요목적에서 벗어난 ‘끝이 보이지 않는 탐색’으로 빠질 수 있다는 점도 경험하였습니다.  
독자적으로 생각을 떠올리고 적용해보는 과정에서 '이 길이 아닌거 같은데..?' '라는 생각이 들 때마다 힘들었지만 이를 통해 EDA 경험과 다양한 시각을 갖는 것의 중요성을 느꼈습니다. 또한 동료들에게 의견을 구하는게 중요하다는 것을 느꼈습니다.  
가장 재밌었던 시간은 협업 과정에서는 서로의 아이디어를 브레인스토밍하는 과정이었습니다. 간단하게 던져진 의견들이 '오? 꽤나 의미 있는 주제네?'라는 통찰로 이어지는 것을 보며, 여러 사람의 다양한 시각이 어떻게 큰 그림을 그려나가는지 체감할 수 있었습니다.  
좀 더 더 더 더 더 더 많은 데이터 의 필요성을 확확 느낍니다! (크롤링..?)
- 이동근 : 이번 미니 프로젝트를 통해 자신의 실력을 더욱 더 정확히 알게 되었고, 부족한 부분을 서로 보완해 가면서 많은 것을 배우는 시간을 가져서 좋았습니다.
- 조인철: 이번 프로젝트를 통해 EDA가 중요하다는 것을 다시 한 번 깨달았고 EDA에 대한 공부가 추가적으로 필요하다고 느꼈습니다. 그리고 좋은 팀원들 덕분에 재미있게 프로젝트를 완료해서 좋았습니다.