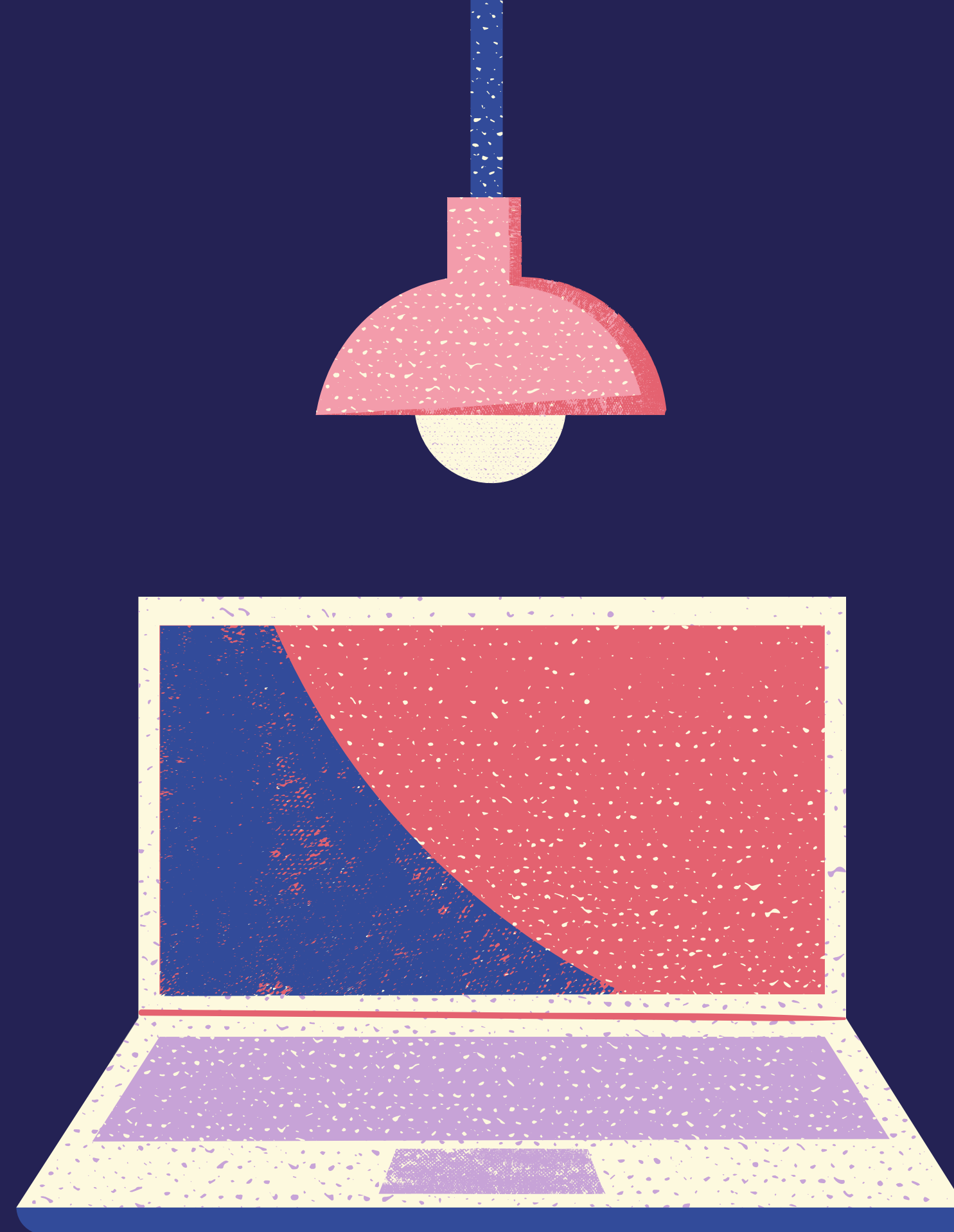


# 스마트 스토어 마케팅 전략을 위한 어플 개발

임광현



# 주제 선정 배경

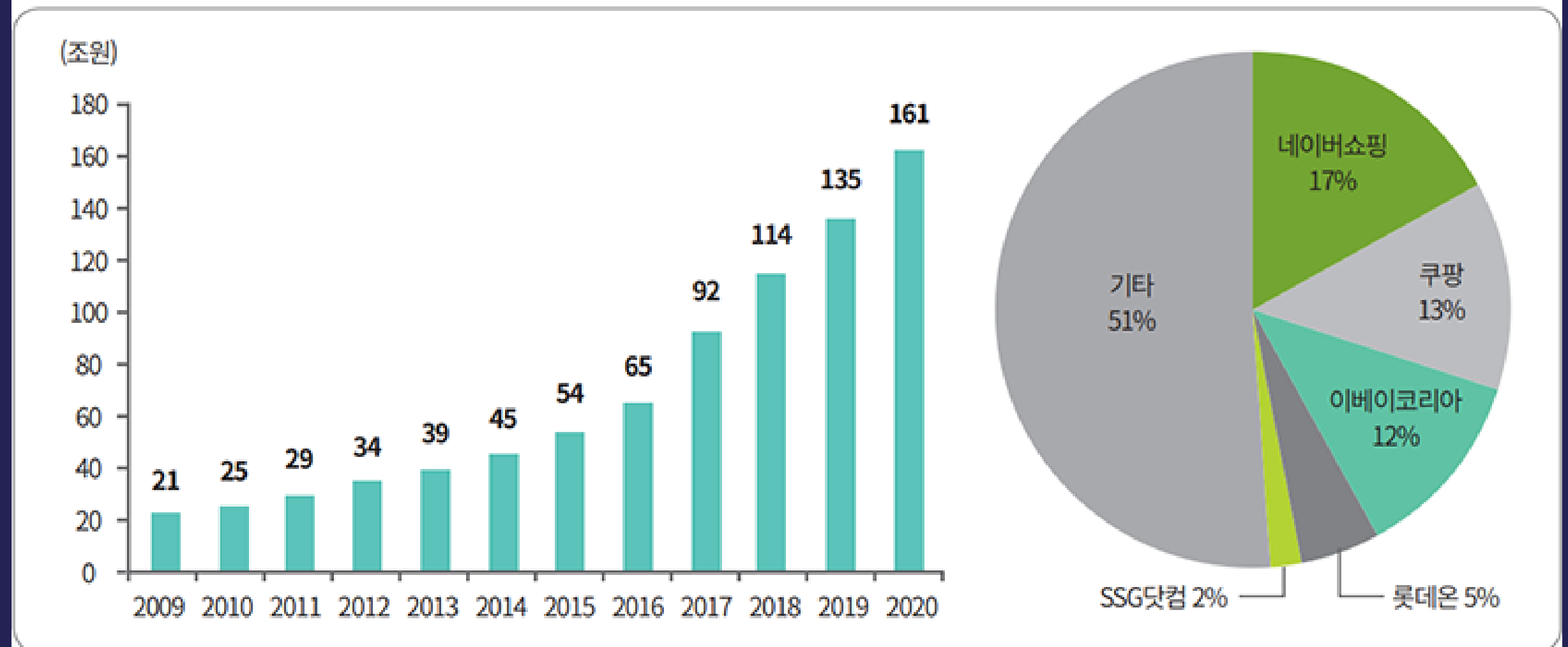
최근 몇년간 투잡에 대한 관심이 높아짐 에따라  
"스마트 스토어의 대한 관심도"가 높아짐

<직장인들이 선호하는 '투잡'>				
항목	20대	30대	40대 이상	전체
서비스직	45.40%	40.70%	47.20%	44.30%
쇼핑몰/세포마켓(SNS)	22.50%	30.80%	17.90%	24.30%
유튜버(1인 미디어)	31.70%	19.20%	8.50%	23.10%
강의/과외	17.20%	14.50%	18.90%	16.70%
디자인/개발 대행 (본업 기술을 살리는 투잡)	16.40%	17.40%	9.40%	15.40%
배달/택배/대리운전 등 (O2O서비스)	12.60%	15.70%	20.80%	15.20%
자영업	9.50%	9.90%	17.90%	11.30%
기타	3.40%	4.70%	9.40%	5.00%

# 네이버 API 사용 하게된 배경

기타를 제외한 전자상거래 규모는 "네이버 쇼핑이 가장크고",  
전자 상거래 시장 규모는 "매해 증가하는 추세" 이다.

[ 그림 1 ] 국내 전자상거래 시장규모 및 점유율(2020년 기준)



출처 : DB금융투자(2021.3)

# 스마트 스토어 마케팅 전략 을 위한 어플 소개

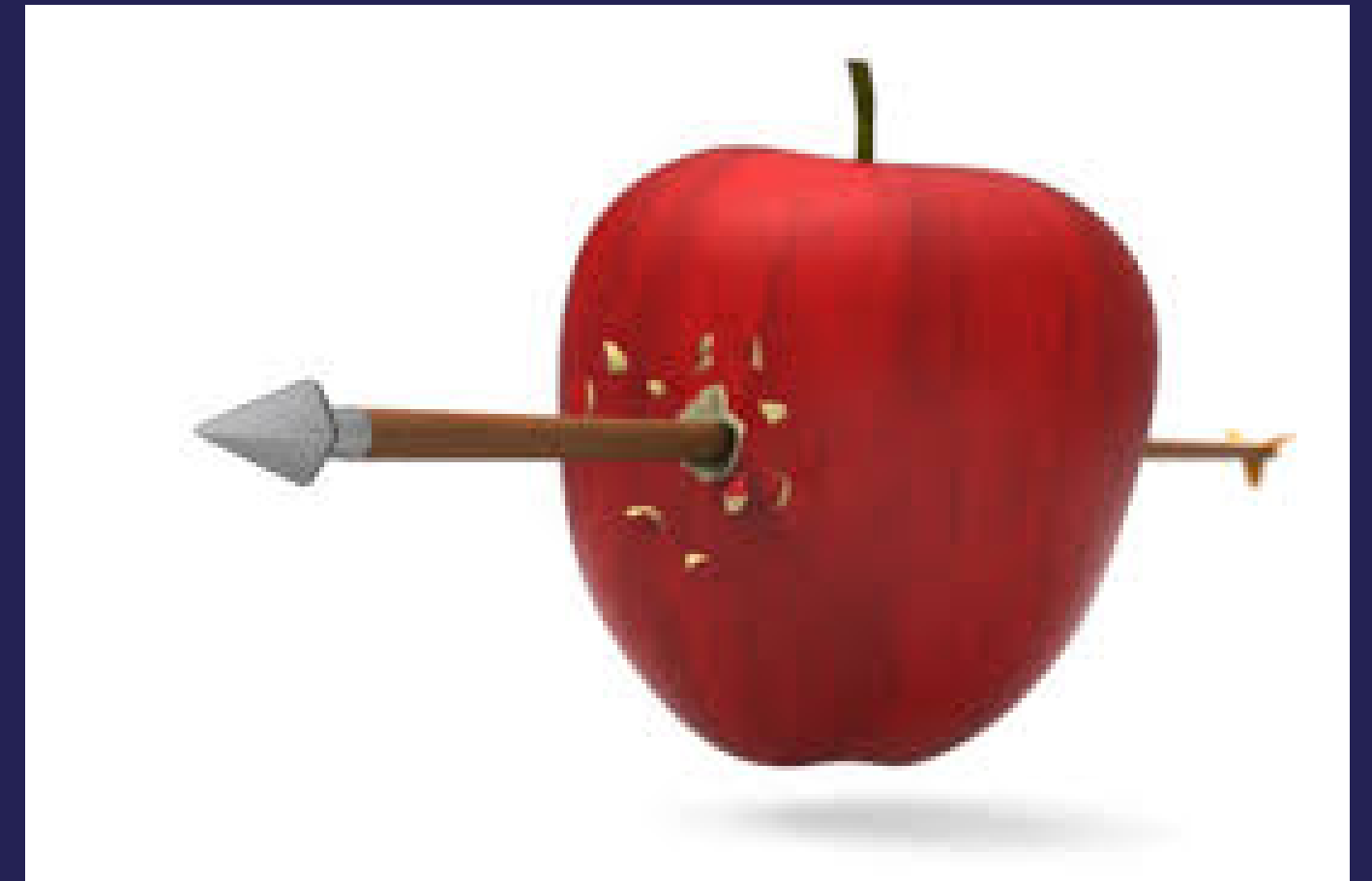
1. 사용자 타겟층 : 스마트 스토어를 하고 있는, 할려고 준비중인 예비 사업가

## 2. 기능

- 연령대별, 성별, 접속 환경별 상대 클릭수를 plot 하여, 상품에 맞는 맞춤 광고 전략 수립 가능
- 팔고자 하는 상품의 인기 상품 상위 100개의 평균 가격과 상품이 가장 많이 팔리는 쇼핑 플랫폼(쿠팡, 옥션 등) 및 상품이 가장 많이 팔리는 상품의 브랜드를 스크랩 하여, 팔고자 하는 상품의 대한 판매 전략 수립 가능

## 3. 활용 방안

- 제품 선정을 하지 못했을 때
- 이미 제품을 정하였지만, 판매 매출이 저조 할때
- 광고 전략을 수립하기 위해(타겟층 선정이 어려울때)



# 네이버 api 데이터를 수집하는 과정

1. Open API 요청을 위한 **인증 정보**를 설정합니다.
2. 데이터 수집에 필요한 **파라미터**(시작일, 종료일, 카테고리, 연령대, 성별, 검색환경)를 설정합니다.
3. 파라미터를 **JSON** 형태로 변환합니다.
4. 파라미터를 **HTTP POST** 방식으로 전송 합니다.
5. 응답으로 받은 **JSON** 데이터로 파싱합니다.

## -요청 파라메터

파라미터 			
파라미터를 JSON 형식으로 전달합니다.			
파라미터	타입	필수 여부	설명
startDate	string	Y	조회 기간 시작 날짜 (yyyy-mm-dd 형식). 2016년 1월 1일부터 조회할 수 있습니다.
endDate	string	Y	조회 기간 종료 날짜 (yyyy-mm-dd 형식)
timeUnit	string	Y	구간 단위 - date: 일간 - week: 주간 - month: 월간
keywordGroups	array (JSON)	Y	주제어와 주제어에 해당하는 검색어 묶음 범위 배 업. 최대 5개의 쌍을 배열로 설정할 수 있습니다.
keywordGroups.groupName	string	Y	주제어. 검색어 묶음을 대표하는 이름입니다.
keywordGroups.keywords	array (string)	Y	주제어에 해당하는 검색어. 최대 20개의 검색어를 배열로 설정할 수 있습니다.
device	string	N	범위. 검색 환경에 따른 조건입니다. - 설정 안 함: 모든 환경 - pc: PC에서 검색 추이 - mo: 모바일에서 검색 추이
gender	string	N	성별. 검색 사용자의 성별에 따른 조건입니다. - 설정 안 함: 모든 성별 - m: 남성 - f: 여성
ages	array (string)	N	연령. 검색 사용자의 연령에 따른 조건입니다. - 설정 안 함: 모든 연령 - 1: 0~12세 - 2: 13~18세 - 3: 19~24세 - 4: 25~29세 - 5: 30~34세 - 6: 35~39세 - 7: 40~44세 - 8: 45~49세 - 9: 50~54세 - 10: 55~59세 - 11: 60세 이상

# API Flow Summary

## 01. Category dictionary

네이버 쇼핑 api  
요청 파라미터 중  
**category code**  
검색

## 02. plot\_relative\_clicks

연령대 별로 접속 환경  
(PC, Mobile)에 따른  
상대 클릭수  
시계열 그래프 그리기

## 03.create\_relative\_clicks\_graph

연령대 및 성별  
접속 환경(PC,  
Mobile)에 따른 상  
대클릭수 시계열 그  
래프 그리기

## 04. get\_shop\_infot

팔고자 하는 물품 상  
위 100개의 가격 및  
상위 10개의 브랜드  
명 과 쇼핑몰 플랫폼  
(쿠팡, 네이버...)

# 01. Category dictionary

```
import pandas as pd

# 엑셀 파일 경로 설정
excel_file = 'category_20230325_142255.xls'

# 엑셀 파일을 pandas DataFrame으로 변환
df = pd.read_excel(excel_file)

# 대분류 유니크한 데이터 추출
대분류 = df.iloc[:, 1].unique()

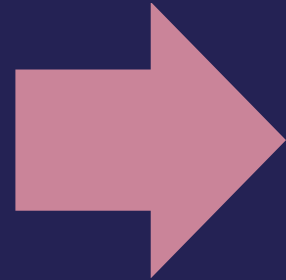
# 대분류 목록 출력
print("- 대분류 목록")
for i in range(len(대분류)):
    print(f"{i+1}. {대분류[i]}")

# 대분류 입력 받기
cat1 = input("대분류를 입력해주세요. \n : ")

# 해당 대분류에 해당하는 중분류 추출
중분류 = df[df.iloc[:, 1] == cat1].iloc[:, 2].unique()

# 중분류 목록 출력
print()
print("- 중분류 목록")
for i in range(len(중분류)):
    print(f"{i+1}. {중분류[i]}")

# 중분류 입력 받기
cat2 = input("중분류를 입력해주세요.\n : ")
```



```
# 해당 중분류에 해당하는 소분류 추출
소분류 = df[(df.iloc[:, 1] == cat1) & (df.iloc[:, 2] == cat2)].iloc[:, 3].unique()

# 소분류 목록 출력
print()
print("- 소분류 목록")
for i in range(len(소분류)):
    print(f"{i+1}. {소분류[i]}")

# 소분류 입력 받기
cat3 = input("소분류를 입력해주세요. \n : ")

# 해당 소분류에 해당하는 세분류 추출
세분류 = df[(df.iloc[:, 1] == cat1) & (df.iloc[:, 2] == cat2) & (df.iloc[:, 3] == cat3)].iloc[:, 4].values.tolist()

# 세분류가 있는 경우
if not pd.isnull(세분류).all():
    # 세분류 목록 출력
    print()
    print("- 세분류 목록")
    for i in range(len(세분류)):
        print(f"{i+1}. {세분류[i]}")

    # 세분류 입력 받기
    cat4 = input("세분류를 입력해주세요.\n : ")

    # 해당 조건에 맞는 데이터 추출 및 출력
    num = df[(df.iloc[:, 1] == cat1) & (df.iloc[:, 2] == cat2) & (df.iloc[:, 3] == cat3) & (df.iloc[:, 4] == cat4)].iloc[:, 0]
    if len(num) == 0:
        print("해당하는 데이터가 없습니다.")
    else:
        print(num)
```

**pandas** 모듈을 사용하여, 카탈로그 Excel 파일을 불러와 사용자로 부터 입력 받은  
대/중/소/세 분류에 따른 **category code** 를 out put 해주는 함수



## 02. 함수 `get_relative_clicks(category_name, category_code)`

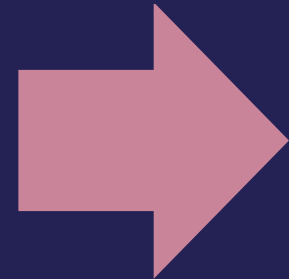
```
def get_relative_clicks(category_name, category_code):
    client_id = "JagKFffWu4Fj_OkksX_R"
    client_secret = "LD2ZmSF2j8"
    url = "https://openapi.naver.com/v1/datalab/shopping/categories"

    # ages, gender, device 변수 정의
    ages = ["10", "20", "30", "40", "50", "60"]
    genders = ["f", "m"]
    devices = ["pc", "mo"]

    # dfs 빈 리스트 선언
    dfs = []

    # for 문으로 연령별, 성별, 접속 환경별 request body 만들기
    for age in ages:
        for gender in genders:
            for device in devices:
                body = {
                    "startDate": "2022-01-01",
                    "endDate": "2022-12-31",
                    "timeUnit": "month",
                    "category": [{"name": category_name, "param": [category_code]}],
                    "device": device,
                    "ages": [age],
                    "gender": gender
                }
                body = json.dumps(body)

                request = urllib.request.Request(url)
                request.add_header("X-Naver-Client-Id", client_id)
                request.add_header("X-Naver-Client-Secret", client_secret)
                request.add_header("Content-Type", "application/json")
                response = urllib.request.urlopen(request, data=body.encode("utf-8"))
                rescode = response.getcode()
```



```
        if rescode == 200:
            response_body = response.read()
            data = json.loads(response_body.decode('utf-8'))
            df = pd.DataFrame(data['results'][0]['data'])
            df["age"] = age
            df["gender"] = gender
            df["device"] = device
            dfs.append(df)
        else:
            print("Error Code:", rescode)

    # final_df 함수에 모든 추출 데이터 합치기
    final_df = pd.concat(dfs, ignore_index=True)

    # Excel 파일의 별도 워크시트에 각 시트 작성
    with pd.ExcelWriter(f'{category_name}_relative_clicks.xlsx') as writer:
        for age in ages:
            age_df = final_df[final_df["age"] == age]
            age_df.to_excel(writer, sheet_name=f"{age}_ages", index=False)

        for gender in genders:
            gender_df = final_df[final_df["gender"] == gender]
            gender_df.to_excel(writer, sheet_name=f"{gender}_gender", index=False)

        for device in devices:
            device_df = final_df[final_df["device"] == device]
            device_df.to_excel(writer, sheet_name=f"{device}_device", index=False)
    result = f'{category_name}_relative_clicks.xlsx' 파일이 저장 완료 되었습니다."
    return result

print()
print("연령대 및 성별 등 조건에 따른 상대 클릭수 확인")
print("카테고리를 골랐다면, 카테고리 이름과 카테고리 코드를 입력해주세요.")
category_name = input("카테고리 이름을 넣어주세요. Ex)콜라\n :")
category_code = input("카테고리 코드를 입력해주세요. Ex)50002254\n :")
```

**이 코드는 네이버 쇼핑 데이터랩 API를 사용하여, 특정 카테고리의 연령, 성별, 접속환경별 상대 클릭수를 추출하고, 추출한 데이터를 엑셀 파일에 저장하는 코드입니다.**



## 02-1. 함수 `get_relative_clicks(category_name, category_code)`

### • Result

```
print()
print("연령대 및 성별 등 조건에 따른 상대 클릭수 확인")
print("카테고리를 골랐다면, 카테고리 이름과 카테고리 코드를 입력해주세요.")
category_name = input("카테고리 이름을 넣어주세요. Ex)콜라\n :")
category_code = input("카테고리 코드를 입력해주세요. Ex)50002254\n :")

get_relative_clicks(category_name, category_code)
```

연령대 및 성별 등 조건에 따른 상대 클릭수 확인  
카테고리를 골랐다면, 카테고리 이름과 카테고리 코드를 입력해주세요.  
카테고리 이름을 넣어주세요. Ex)콜라  
:자동우산  
카테고리 코드를 입력해주세요. Ex)50002254  
:50002254

	10_ages	20_ages	30_ages	40_ages	50_ages	60_ages
	A	B	C	D	E	
1	period	ratio	age	gender	device	
2	2022-01-01	8.33	10	f	pc	
3	2022-02-01	27.50	10	f	pc	
4	2022-03-01	63.33	10	f	pc	
5	2022-04-01	58.33	10	f	pc	
6	2022-05-01	46.39	10	f	pc	
7	2022-06-01	100	10	f	pc	
8	2022-07-01	72.78	10	f	pc	
9	2022-08-01	60	10	f	pc	
10	2022-09-01	60.28	10	f	pc	
11	2022-10-01	25.56	10	f	pc	
12	2022-11-01	19.72	10	f	pc	
13	2022-12-01	29.44	10	f	pc	

`get_relative_clicks` 함수 실행시, 오른쪽 **result** 와 같이 **엑셀 파일이 생성**

**period = 월별, ratio = 상대 클릭수, age = 연령, gender = 성별, device = 접속환경**

### 03. 함수 `def plot_relative_clicks(category_name):`

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

def plot_relative_clicks(category_name):
    # 데이터프레임 생성
    df_list = []
    for age in range(10, 70, 10):
        sheet_name = f"{age}_ages"
        df = pd.read_excel(f"{category_name}_relative_clicks.xlsx", sheet_name=sheet_name)
        df['age'] = age
        df_list.append(df)

    df_all = pd.concat(df_list)

    # 날짜 형식 변경
    df_all['period'] = pd.to_datetime(df_all['period'], format='%Y-%m')

    # 피벗 테이블 생성
    pt = pd.pivot_table(df_all, values='ratio', index=['age', 'period'], columns='device',
                        aggfunc='sum', fill_value=0)

    # 연령대 별로 접속 환경(PC, Mobile)에 따른 시계열 그래프 그리기
    fig, axes = plt.subplots(nrows=2, ncols=3, figsize=(15,8), sharex=True, sharey=True)
    for i, ax in enumerate(axes.flat):
        age = (i+1)*10
        pt_age = pt.loc[age]
        ax.plot(pt_age.index, pt_age['pc'], label='PC')
        ax.plot(pt_age.index, pt_age['mo'], label='Mobile')
        ax.set_title(f'{age} ages')
        ax.legend()

    # x축 라벨 세로로 표시하고 눈금을 12개 만들기
    fig.autofmt_xdate(rotation=90)
    plt.xticks(pd.date_range(start=pt.index.get_level_values(1).min(), end=pt.index.get_level_values(1).max(), freq='MS'),
               rotation=90)

    plt.suptitle('Relative clicks by device and age group')
```



```
# 그래프 저장
plt.savefig(f'{category_name}_pc_mo_relative_clicks.png')

# plot_relative_clicks("자급제폰")

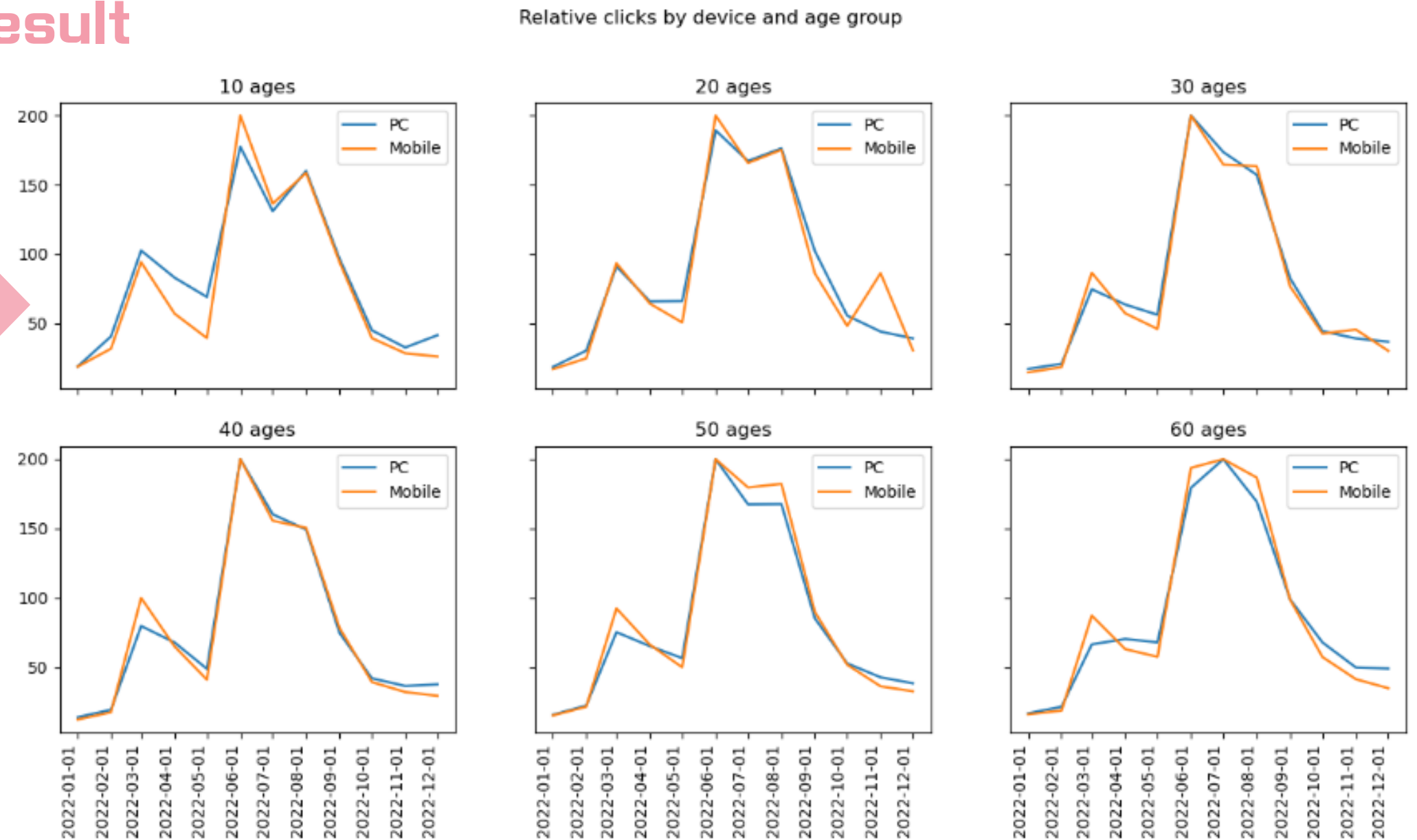
print()
print("연령대 별로 접속 환경(PC, Mobile)에 따른 시계열 그래프 그리기")
category_name = input("카테고리 이름을 입력해주세요\n : ")
plot_relative_clicks(category_name)
print(f'{category_name}_pc_mo_relative_clicks.png 저장 완료')
```

이전 함수로 **스크랩한 엑셀 파일을 다시 연령대 별로 접속 환경(PC, Mobile)에 따른 시계열 그래프 그리기**

## 03-1. 함수 `def plot_relative_clicks(category_name):`

연령대 별로 접속 환경(PC, Mobile)에 따른 시계열 그래프 그리기  
카테고리 이름을 입력해주세요  
: 자동우산  
자동우산 pc mo relative clicks.png 저장 완료

### • Result



이전 함수에 입력 했던 카테고리 이름 입력 시, 연령별 기기 환경에 따른 접속 후이 그래프 저장

## 04. 함수 create\_relative\_clicks\_graph(category\_name)

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

def create_relative_clicks_graph(category_name):
    # 데이터프레임 생성
    df_list = []
    for age in range(10, 70, 10):
        sheet_name = f"{age}_ages"
        df = pd.read_excel(f"{category_name}_relative_clicks.xlsx", sheet_name=sheet_name)
        df['age'] = age
        df_list.append(df)

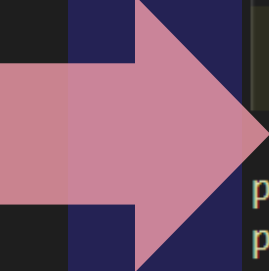
    df_all = pd.concat(df_list)

    # 날짜 형식 변경
    df_all['period'] = pd.to_datetime(df_all['period'], format='%Y-%m')

    # 피벗 테이블 생성
    pt = pd.pivot_table(df_all, values='ratio', index=['age', 'gender', 'period'], columns='device',
                        aggfunc='sum', fill_value=0)

    # 연령대 및 성별 별로 접속 환경(PC, Mobile)에 따른 시계열 그래프 그리기
    fig, axes = plt.subplots(nrows=4, ncols=3, figsize=(15,12), sharex=True, sharey=True)
    for i, ax in enumerate(axes.flat):
        age_gender = (i//2+1)*10, 'f' if i%2==0 else 'm'
        pt_age_gender = pt.loc[age_gender]
        ax.plot(pt_age_gender.index, pt_age_gender['pc'], label='PC')
        ax.plot(pt_age_gender.index, pt_age_gender['mo'], label='Mobile')
        ax.set_title(f'{age_gender[0]} ages {age_gender[1].upper()}')
        ax.legend()

    # x축 라벨 세로로 표시하고 눈금을 12개 만들기
    fig.autofmt_xdate(rotation=90)
    plt.xticks(pd.date_range(start=pt.index.get_level_values(2).min(), end=pt.index.get_level_values(2).max(),
```



```
plt.suptitle('Relative clicks by device, age group, and gender')
# plt.show()

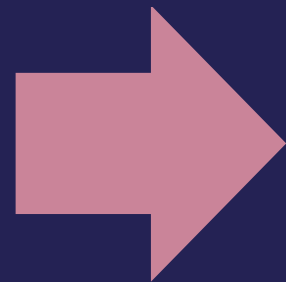
# 그래프 저장
plt.savefig(f'{category_name}_gender_relative_clicks.png')

print()
print("연령대 및 성별 별로 접속 환경(PC, Mobile)에 따른 시계열 그래프 그리기")
category_name = input("카테고리 이름을 입력해주세요\n : ")
create_relative_clicks_graph(category_name)
print(f'{category_name}_gender_relative_clicks.png 저장완료')
```

이전 함수로 스크랩한 엑셀 파일을 다시 연령대 및 접속 환경(PC, Mobile) 및 성별 에 따른  
접속 후이 시계열 그래프 그리기

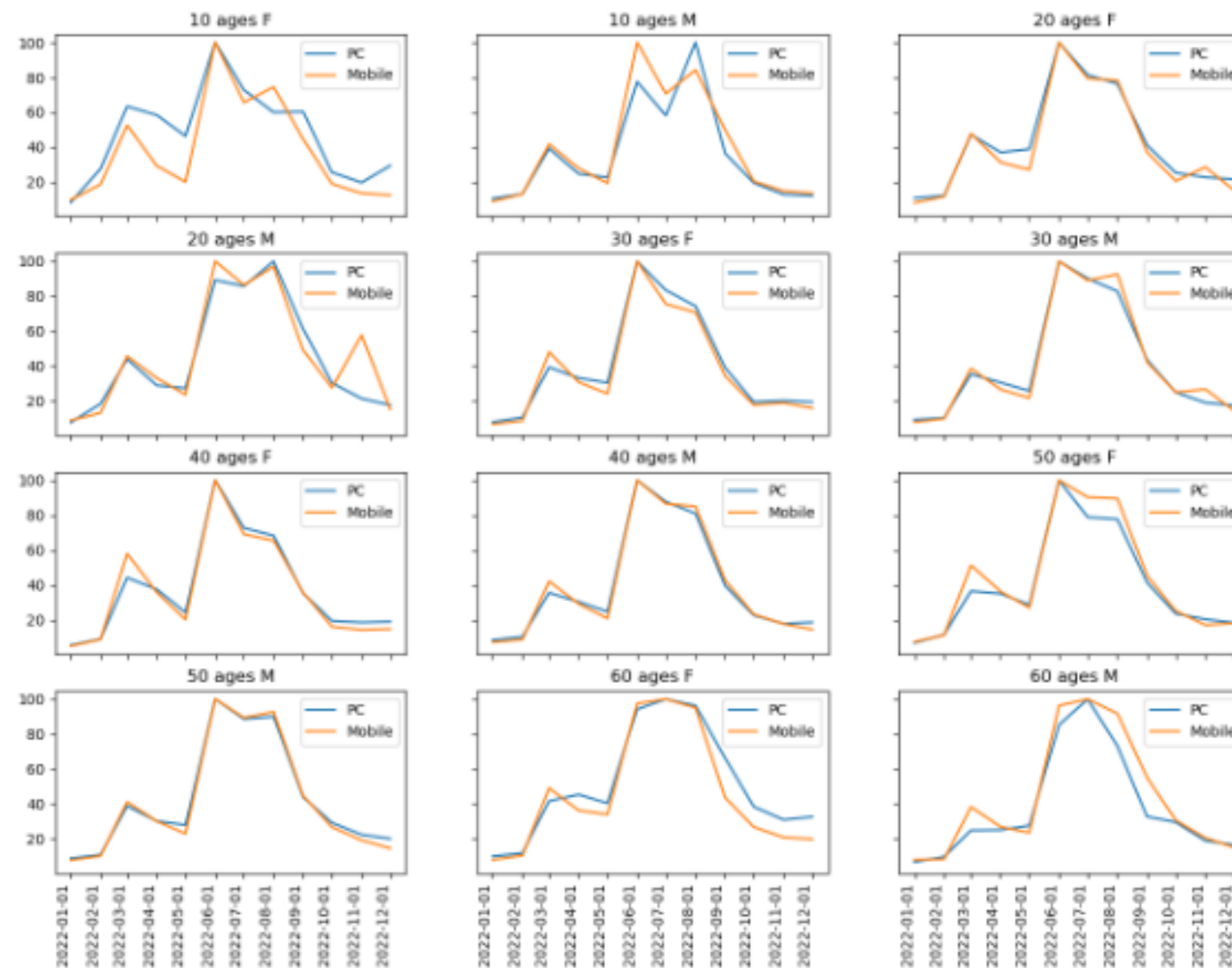
## 04.-2 함수 `create_relative_clicks_graph(category_name)`

연령대 및 성별 별로 접속 환경(PC, Mobile)에 따른 시계열 그래프 그리기  
카테고리 이름을 입력해주세요  
: 자동우산  
자동우산\_gender\_relative\_clicks.png 저장완료



### • Result

Relative clicks by device, age group, and gender



이전 함수에 입력 했던 카테고리 이름 입력 시 , 연령별, 기기 환경, 성별 에 따른 접속 후이 그래프 저장



## 05 . 함수 get\_shop\_info(keyword)

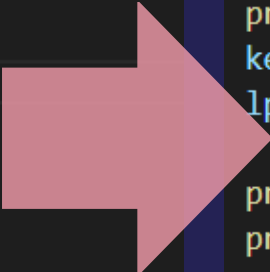
```
def get_shop_info(keyword):
    client_id = "JagKFffWu4Fj_OkksX_R"
    client_secret = "LD2ZmSF2j8"
    encText = urllib.parse.quote(keyword)
    shop_url = "https://openapi.naver.com/v1/search/shop?query=" + encText + "&display=100&start=1"
    request = urllib.request.Request(shop_url)
    request.add_header("X-Naver-Client-Id", client_id)
    request.add_header("X-Naver-Client-Secret", client_secret)
    response = urllib.request.urlopen(request)
    rescode = response.getcode()
    if rescode == 200:
        response_body = response.read()
        json_str = response_body.decode('utf-8')
    else:
        response_body.decode('utf-8')
        print("Error Code: " + rescode)
    json_data = json.loads(json_str)
    df = pd.json_normalize(json_data['items'])
    df['title'] = df['title'].str.replace('<b>', '').str.replace('</b>', '')

    lprice_mean = df.loc[df['lprice'].notnull(), 'lprice'].astype(int).mean()

    mallName_cnt = df['mallName'].str.split(expand=True).stack().value_counts()
    mallName_dup_cnt = dict(mallName_cnt[:10])

    brand_cnt = df['brand'].str.split(expand=True).stack().value_counts()
    brand_dup_cnt = dict(brand_cnt[:10])

    return lprice_mean, mallName_dup_cnt, brand_dup_cnt,
```



```
print()
print("팔려고 하는 물품 상위 100개의 가격 및 상위 10개의 브랜드 명 과 쇼핑몰 플랫폼(쿠팡, 네이버) 확인 하기")
keyword = input("keyword 를 입력해주세요.")
lprice_mean, mallName_dup_cnt, brand_dup_cnt = get_shop_info(keyword)

print(f"상위 100개의 평균 가격 입니다. \n = {lprice_mean} 원")
print(f"상위 10개 쇼핑몰명 중복횟수 입니다. \n = {mallName_dup_cnt}")
print(f"상위 10개 브랜드명 중복횟수 입니다. \n = {brand_dup_cnt}")
```

**특정 물품 상위 100개의 가격 및 상위 10개의 브랜드 명 과 쇼핑몰 플랫폼(쿠팡, 네이버)**



## 05-1. 함수 . 함수 `get_shop_info(keyword)`

```
팔려고 하는 물품 상위 100개의 가격 및 상위 10개의 브랜드 명 과 쇼핑몰 플랫폼(쿠팡, 네이버) 확인 하기
keyword 를 입력해주세요.
: 자동우산
상위 100개의 평균 가격 입니다.
= 12473.87 원
상위 10개 쇼핑몰명 중복횟수 입니다.
= {'네이버': 69, '일상생활연구소': 2, '베오라': 2, '쇼핑마니또': 2, '토스몰': 2, '부산소녀': 1, '우주파워샵': 1, '씨티라이프': 1, '툭TOP': 1, '오만거다': 1}
상위 10개 브랜드명 중복횟수 입니다.
= {'카카오프렌즈': 6, '파라체이스': 5, '일상생활연구소': 3, '기라로쉬': 3, '산리오': 3, 'MLB': 2, '예다움': 2, '스위스밀리터리': 2, '고티바쉬': 2, '질스튜어트': 2}
```

**자동우산의 상위 100개의 평균 가격은 12473.87 원 이고, 네이버 쇼핑몰에서 제일 구매를 많이하며,  
카카오 프렌즈 브랜드가 가장 인기가 많다.**



## 앞서 소개 했던 api 를 통한 자동우산의 광고 마케팅 전략

1. 모든 연령층에서 6월 > 7월 > 8월 순으로 자동우산을 구매하는 경향이 강함. <- 6월 7월 8월에 광고 빈도 수 증가 시킴
2. 접속 환경에 따른 구매 비율이 크게 차이 나지 않음. <- 모바일 / pc 광고를 50대 50
3. 자동 우산의 상위 100개의 평균 가격은 12473.87 원 <- 판매 가격은 12000원 내외가 많음.
4. 네이버 쇼핑몰에서 제일 구매를 많이 함. <- 네이버 쇼핑몰에 광고 적극적으로 공략
5. 카카오 프렌즈 브랜드가 가장 인기가 많음. <- 자동우산은 카카오프렌즈 가 제일 인기가 많음.

**Q&A**

감싸합니다!

임광현