


다이렉트 예금 가입 고객 분석

팀명: 최대박 / 팀원: 최대훈, 박성실

2022.06.11



Index

1. 프로젝트 일정
 2. 요구사항 정의서
 3. 분석단계별 수행 계획
 4. 최종분석결과
 5. Lessons Learned
- 



프로젝트 일정

6/7

요구사항 정리/기획

요구사항 파악/정리
요구사항 정의서 작성
분석단계별 수행계획 작성
데이터 정의서 작성

6/8

분석단계

파이썬 코드 구현

6/9~10

분석단계

파이썬 코드 수정 및 보완
최종분석 보고서 작성

6/11

최종보고

최종분석 보고서 최종 작성
프로젝트 리뷰



요구사항 정의서

22.3월말 기준 다이렉트 예금 보유고객 특성별 현황

성별 : M(남성) / F(여성)

연령별 : 10대/20대/30대/40대/50대/60대이상

지역별 : 서울/경인/전북/전남/기타



2022년 1~3월말 기준
예/적금좌수 현황

고객 특성

리포트
형태

1Paper리포트 형태 출력

분석단계별 수행계획

00

기초테이블 생성

Data susin & data gds cd 조인하여 예적금 구분, 상품명 열 생성

Data susin & data cstno 조인하여 성별, 연령, 지역 열 생성

01

22년1~3월말 기준 예/적금좌수현황

정상계좌 카운트 → 예/적금막대그래프 시각화

02

22.3월말 기준 다이렉트예금 고객 현황

성별: M(남성)/F(여성) 구분 / 지역별: 지역별 구분

연령별: 10대, 20대, 30대, 40대, 50대, 60대 이상으로 구분

→ 고객 중복제거한 후 그룹화하여 고객수 카운트하여 파이차트 시각화

03

한페이지에 리포트 형태 출력

생성한 차트들을 한페이지에 시각화



최종 분석 결과

Step0. 기초 테이블 생성

엑셀파일 불러오기

```
mission_data_xls = './data/mission_data.xlsx'
```

```
data_susin = pd.read_excel(mission_data_xls)
data_susin.head()
```

```
data_cstno = pd.read_excel(mission_data_xls, sheet_name=1)
data_cstno.head()
```

```
data_gds_cd = pd.read_excel(mission_data_xls, sheet_name=2)
data_gds_cd.head()
```

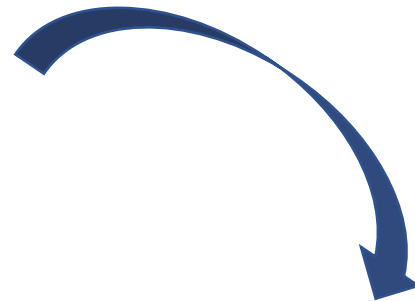
테이블 조인

```
left = data_susin
right = data_cstno
mission_data1 = pd.merge(left, right, on='cstno')
mission_data1.head()
```

```
left = mission_data1
right = data_gds_cd
mission_data2 = pd.merge(left, right, on='gds_cd')
mission_data2.head()
```

정상계좌만 추출

```
data1 = mission_data2[mission_data2.stcd == '정상']
data1
```



	base_dt	acno	gds_cd	stcd	cstno	age	sex	add	gds_nm	gds_dvcd
0	20220131	1111021010800000	ZZZZAA	정상	300000000111	11	F	전북	JB다이렉트예금	예금
1	20220228	1111021010800000	ZZZZAA	정상	300000000111	11	F	전북	JB다이렉트예금	예금
2	20220331	1111021010800000	ZZZZAA	정상	300000000111	11	F	전북	JB다이렉트예금	예금
3	20220228	1111021010800039	ZZZZAA	정상	300000000117	16	M	전북	JB다이렉트예금	예금
4	20220228	1111021010800044	ZZZZAA	정상	300000000117	16	M	전북	JB다이렉트예금	예금



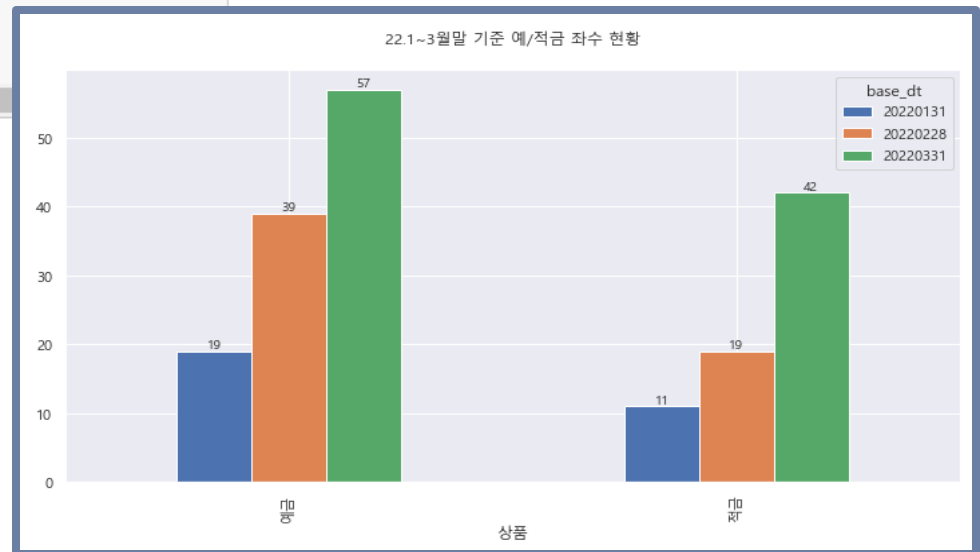
최종 분석 결과

Step1. 2022년 1~3월말 기준 예/적금 좌수 현황

```
data2 = data1.groupby(["gds_dvcd", "base_dt"]).size().unstack(1)  
data2
```

base_dt	20220131	20220228	20220331
gds_dvcd			
예금	19	39	57
적금	11	19	42

```
ax = data2.plot(kind='bar', xlabel='상품', title="\n22.1~3월말 기준 예/적금 좌수 현황")  
for c in ax.containers:  
    labels = [ x.get_height() for x in c ]  
    ax.bar_label(c, labels=labels, label_type='edge')
```



최종 분석 결과

Step2-1. 성별

```
# 22.3월기준 / 정상 / JB다이렉트예금 DF 생성
d_3m_tmp = mission_data2[mission_data2.base_dt == 20220331]
d_3m_tmp2 = d_3m_tmp[d_3m_tmp.stcd == '정상']
d_3m_tmp3 = d_3m_tmp2[d_3m_tmp2.gds_nm == 'JB다이렉트예금']
d_3m_tmp3
```

```
# 고객 중복 제거
d_3m = d_3m_tmp3.drop_duplicates(['cstno'])
d_3m
```

```
df_sex_3m = pd.Series(1, index=d_3m['sex'].values)
df_sex_3m
```

```
def get_sex_cnt(d_3m):
    sex_cnt = {}
    for idx in range(len(d_3m.index)):
        sex = d_3m.index[idx]
        count = d_3m[idx]

        if sex_cnt.get(sex):
            sex_cnt[sex] += count
        else:
            sex_cnt[sex] = count

    return sex_cnt
```

```
sex_dict = get_sex_cnt(df_sex_3m)
sex_dict
```

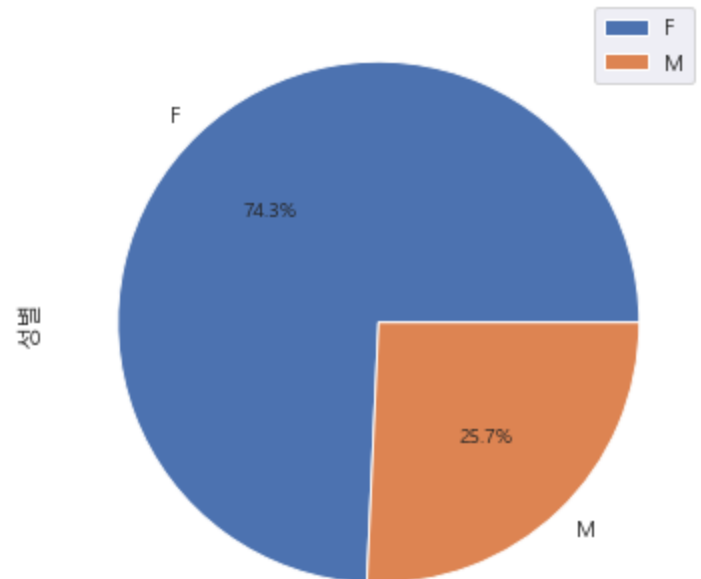
```
{'F': 26, 'M': 9}
```

```
sex_stat = Series(sex_dict)
sex_stat.name = "성별"
sex_stat
```

```
F    26
M     9
Name: 성별, dtype: int64
```

```
sex_stat.plot(kind='pie', autopct='%1f%%');
plt.rcParams["figure.figsize"] = [8,8]
plt.title("\n22.3월말 성별 다이렉트예금 현황\n", fontsize=20)
plt.legend()
```

22.3월말 성별 다이렉트예금 현황



최종 분석 결과

Step2-2. 연령별

```
age_series = d_3m.age.dropna().apply(lambda age: math.floor(age / 10) * 10)  
#NaN값 날리기 #floor -> 소숫점값 날려버리기  
age_series.name = "Age_Group"
```

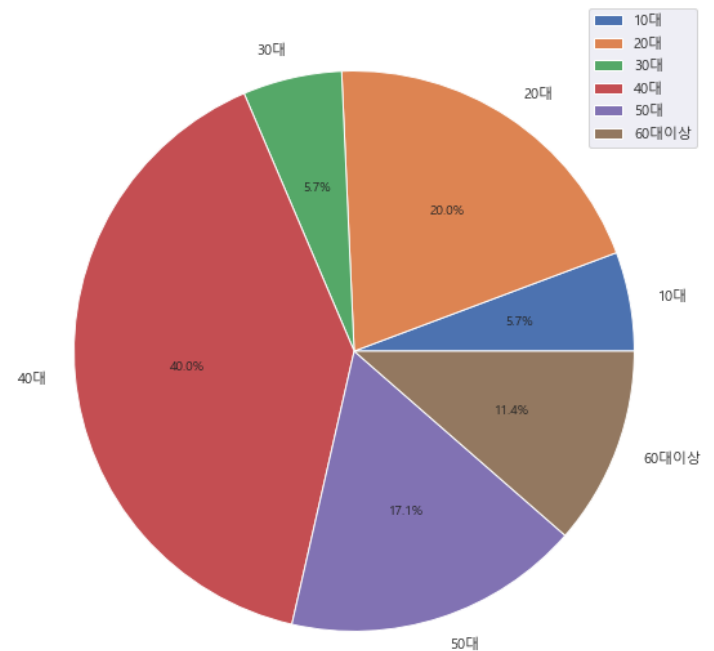
```
age_stat = age_series  
age_stat
```

```
plt.rcParams["figure.figsize"] = [8,8]
```

```
labels = ['10대', '20대', '30대', '40대', '50대', '60대이상']  
sizes = [2, 7, 2, 14, 6, 4]
```

```
fig1, ax1 = plt.subplots()  
ax1.pie(sizes, labels=labels, autopct='%1.1f%%')  
plt.title("\n22.3월말 연령별 다이렉트예금 현황\n", fontsize=20)  
plt.legend(loc=1)  
plt.show()
```

22.3월말 연령별 다이렉트예금 현황



최종 분석 결과

Step2-3. 지역별

```
d_3m = d_3m.replace('경기', '경인')
d_3m = d_3m.replace('인천', '경인')
d_3m = d_3m.replace('경북', '기타')
d_3m = d_3m.replace('경남', '기타')
d_3m = d_3m.replace('제주', '기타')
```

```
def get_add_cnt(d_3m):
    add_cnt = {}
    for idx in range(len(d_3m.index)):

        add = d_3m.index[idx]
        count = d_3m[idx]

        if add_cnt.get(add):
            add_cnt[add] += count
        else:
            add_cnt[add] = count

    return add_cnt
```

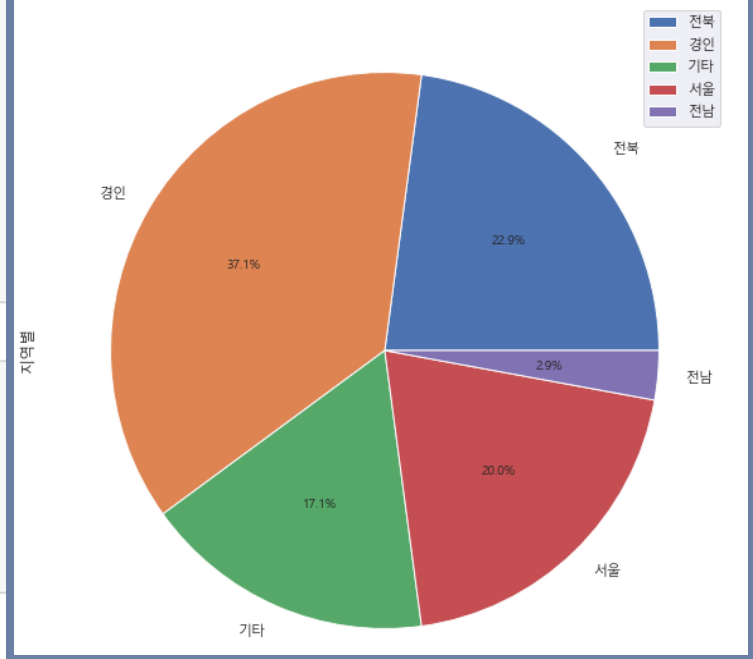
```
add_dict = get_add_cnt(df_add_3m)
add_dict
```

```
{'전북': 8, '경인': 13, '기타': 6, '서울': 7, '전남': 1}
```

```
plt.rcParams["figure.figsize"] = [10,10]
# crime_stat.plot(kind='pie');

add_stat.plot(kind='pie', autopct='%1f%%');
plt.legend(loc=1)
plt.title("\n22.3월말 지역별 다이렉트예금 현황\n", fontsize=20)
# plt.legend()
```

22.3월말 지역별 다이렉트예금 현황



최종 분석 결과

Step3. 함수 모듈화 및 1paper 구현

```
def draw_pieplot(param):  
    if param == 'Sex':  
        sex_stat.plot(kind='pie', autopct='%1.1f%%');  
        plt.rcParams["figure.figsize"] = [10,10]  
        plt.title("\n22.3월말 성별 다이렉트예금 현황\n", fontsize=20)  
        plt.legend()  
  
    elif param == 'Age':  
        plt.rcParams["figure.figsize"] = [10,10]  
        labels = ['10대', '20대', '30대', '40대', '50대', '60대이상']  
        sizes = [2, 7, 2, 14, 6, 4]  
        plt.pie(sizes, labels=labels, autopct='%1.1f%%') # 수정사항  
        plt.title("\n22.3월말 연령별 다이렉트예금 현황\n", fontsize=20)  
        plt.legend(loc=1)  
  
    elif param == 'Add':  
        plt.rcParams["figure.figsize"] = [10,10]  
        add_stat.plot(kind='pie', autopct='%1.1f%%');  
        plt.legend(loc=1)  
        plt.title("\n22.3월말 지역별 다이렉트예금 현황\n", fontsize=20)  
        plt.legend()  
  
    else :  
        print("Not Operation !!")  
        return
```

```
plt.figure(figsize=(30, 30))
```

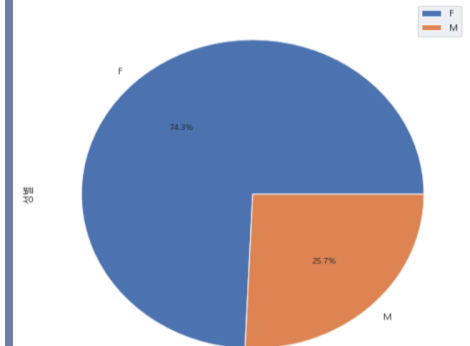
```
plt.subplot(311)  
draw_pieplot('Sex')
```

```
plt.subplot(312)  
draw_pieplot('Age')
```

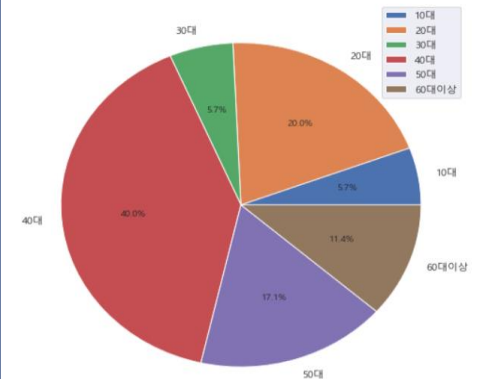
```
plt.subplot(313)  
draw_pieplot('Add')
```



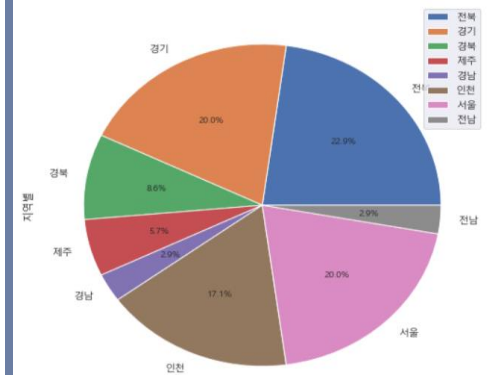
22.3월말 성별 다이렉트예금 현황



22.3월말 연령별 다이렉트예금 현황



22.3월말 지역별 다이렉트예금 현황





Lessons Learned

최대훈

실무예행 연습

실제 데이터분석 의뢰를 받았을때 이런식으로 진행이 될 수 있겠다라고 간접적으로 느낄 수 있었던 좋은 프로젝트였습니다.

박성실

끝날 때까지 끝난 게 아니다

마지막에 쉽게 해결될 거라고 생각한 부분에서 난관에 부딪혀 시간이 많이 소요되었습니다. 예상치 못한 상황에 대비한 유연한 일정 관리가 필요함을 느꼈습니다.

Thank You

