2022.2학기 빅데이터 시각화

학번: 2004401

이름: 고유진

10주차. 사례연구 - 한국 복지 패널 데이터

구글 콜랩 한글 문제 해결

참고: [한글 폰트 설치]

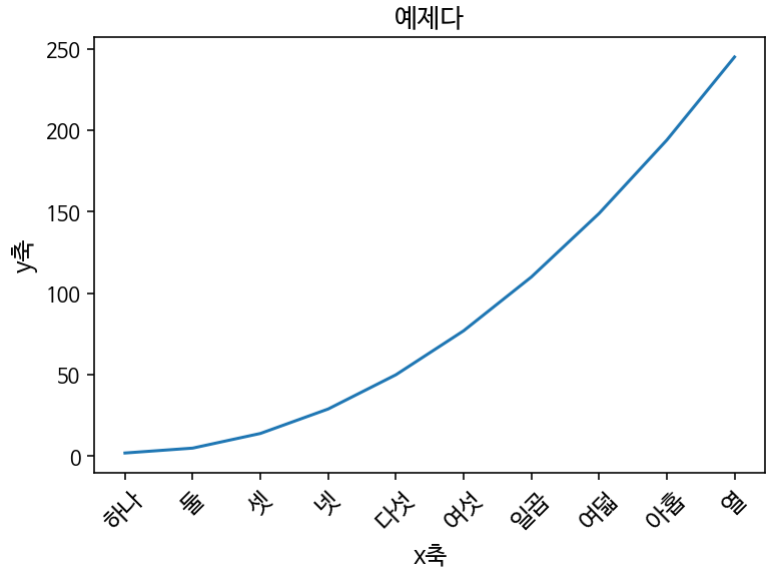
```
# 한글 문제 해결
!sudo apt-get install -y fonts-nanum
!sudo fc-cache -fv
!rm ~/.cache/matplotlib -rf
```

```
Reading package lists... Done
Building dependency tree
Reading state information... Done
fonts-nanum is already the newest version (20170925-1).
The following package was automatically installed and is no longer required:
  libnvidia-common-460
Use 'sudo apt autoremove' to remove it.
0 upgraded, 0 newly installed, 0 to remove and 5 not upgraded.
/usr/share/fonts: caching, new cache contents: 0 fonts, 1 dirs
/usr/share/fonts/truetype: caching, new cache contents: 0 fonts, 3 dirs
/usr/share/fonts/truetype/humor-sans: caching, new cache contents: 1 fonts, 0 dirs
/usr/share/fonts/truetype/liberation: caching, new cache contents: 16 fonts, 0 dirs
/usr/share/fonts/truetype/nanum: caching, new cache contents: 10 fonts, 0 dirs
/usr/local/share/fonts: caching, new cache contents: 0 fonts, 0 dirs
/root/.local/share/fonts: skipping, no such directory
/root/.fonts: skipping, no such directory
/var/cache/fontconfig: cleaning cache directory
/root/.cache/fontconfig: not cleaning non-existent cache directory
/root/.fontconfig: not cleaning non-existent cache directory
fc-cache: succeeded
```

```
# basic
import time
import random
import math

#data analytics
import numpy as np
import pandas as pd

#Math
```

```
import scipy as sp
import statsmodels.api as sm

#web crawling
import requests
from bs4 import BeautifulSoup

#visualization
import matplotlib as mpl
import matplotlib.pylab as plb
import matplotlib.pyplot as plt
import sklearn as sk
import seaborn as sns

# 브라우저에서 바로 그려지도록
%matplotlib inline

# 그래프에 retina display 적용
%config InlineBackend.figure_format = 'retina'

# Colab 의 한글 폰트 설정
plt.rc('font', family='NanumBarunGothic')

# 유니코드에서  음수 부호설정
mpl.rc('axes', unicode_minus=False)
```

```
# 테스트
plt.plot([x for x in range(0, 10)], [(3*y**2)+2 for y in range(0, 10)])
plt.title("예제다", fontsize= 13)
plt.xlabel("x축", fontsize= 12)
plt.xticks(np.arange(0, 10, 1), ['하나', '둘', '셋', '넷', '다섯', '여섯', '일곱', '여덟', '아홉',
plt.ylabel("y축", fontsize= 12)
plt.show()
```



4.2.2 데이터 분석 준비하기

STEP 01. 경고 메시지 안나타나게 하기

```
import warnings
warnings.simplefilter(action='ignore', category=FutureWarning)

#########################################################################
import matplotlib.pyplot as plt
plt.rc('font', family='NanumBarunGothic')
plt.rcParams["font.size"] = 12
plt.rcParams['xtick.labelsize'] = 12.
plt.rcParams['ytick.labelsize'] = 12.
plt.rcParams['axes.unicode_minus'] = False
```

STEP 02. 변수 설정

```
CHART_NAME = 'seabornWelfare_exam'
cnt, PNG, UNDERBAR = 0, '.png', '_'
filename = 'welfare_python.csv'
```

STEP 03. 파일 읽어오기

```
import pandas as pd
welfare = pd.read_csv(filename, encoding='utf-8')

print(welfare.columns)
```

```
Index(['gender', 'birth', 'marriage', 'religion', 'code_job', 'income',
       'code_religion'],
      dtype='object')
```

4.2.3. 복지 데이터 전처리

STEP 01. 데이터 전처리 설명

STEP 02. gender 컬럼의 숫자값을 문자열 값으로 치환

```
welfare.loc[welfare['gender'] == 1, ['gender']] = '남성'
welfare.loc[welfare['gender'] == 2, ['gender']] = '여성'
print('₩n# 나이 컬럼은 존재하지 않으므로 생일 컬럼을 이용하여 산술 연산합니다.')
thisyear = 2020
welfare['age'] = thisyear - welfare['birth'] + 1
```

```
    # 나이 컬럼은 존재하지 않으므로 생일 컬럼을 이용하여 산술 연산합니다.
```

STEP 03. marriage 컬럼을 apply()함수를 적용시켜 문자열로 변환

```
def setMarriage( x ):
    if x == 1 :
        return '결혼'
    elif x == 3 :
        return '이혼'
    else :
        return '무응답' # 결측치

# 결혼 : 숫자 1이면 결혼, 3이면 이혼, 이외에는 결측치로 처리
welfare['marriage'] = welfare['marriage'].apply(setMarriage)
```

STEP 04. 월급 결측치 찾아서 평균 값을 적용

```
print('₩n# 월급 결측치 개수 구하기 before')
print(sum(welfare['income'].isnull()))

welfare.loc[welfare['income'].isnull(), 'income'] = welfare['income'].mean()

print('₩n# 월급 결측치 개수 구하기 after')
print(sum(welfare['income'].isnull()))
```

```
# 월급 결측치 개수 구하기 before
12030

# 월급 결측치 개수 구하기 after
0
```

STEP 05. religion 컬럼을 문자열로 변경

```
def setReligion_txt( x ):
    if int(x) == 1 :
        return '있음'
    else :
        return '없음'

print("welfare['religion'].unique()")
print(welfare['religion'].unique())

welfare['religion'] = welfare['religion'].apply(setReligion_txt)
```

```
welfare['religion'].unique()
[2 1]
```

STEP 06. 직업 코드 정보를 읽어 들이고, 직업코드 목록을 출력

```
job_file = 'welfare_job.csv'
jobframe = pd.read_csv(job_file, encoding='cp949')

print("welfare['code_job'].unique()")
print(welfare['code_job'].unique())
```

```
welfare['code_job'].unique()
[  nan  942.  762.  530.  999.  312.  254.  510.  286.  521.  773.  314.
  941.  951.  274.  873.  320.  952.  151.  152.  772.  852.  442.  991.
  422.  313.  710.  522.  399.  753.  851.  235.  231.  311.  721.  953.
  930.  863.  910.  392.  761.  922.  285.  875.  862.  421.  243.  223.
  252.  259.  771.  135.  245.  221.  751.  251.  141.  722.  246.  289.
  281.  741.  261.  247.  441.  864.  222.  411.  799.  743.  780.  149.
  891.  823.  159.  248.  874.  892.  241.  239.  791.  271.  871.  391.
  620.  131.  431.  811.  272.  429.  213.  842.  283.  284.  134.  611.
  236.  792.  855.  234.  861.  921.  253.  752.  841.  330.  233.  899.
  139.  432.  212.  423.  730.  273.  211.  412.  120.  992.  854.  822.
  831.  853.  832.  612.  821.  613.  774.  132. 1011.  237.  153.  133.
  224.  882.  242.  244.  232.  630.  742.  843. 1012.  881.  812.  819.
  111.  876.]
```

STEP 07. merge 함수를 이용헤서 데이터 프레임 합치기

```
print('₩n# merge() 함수의 left_on과 right_on 사용하기')
welfare = pd.merge( welfare, jobframe, left_on='code_job', right_on='code_job')
print(welfare)
```

```
# merge() 함수의 left_on과 right_on 사용하기
     gender  birth marriage religion  code_job       income  code_religion  ₩
0        남성   1948       무응답       없슴     942.0   120.000000              1
1        남성   1945        이혼       없슴     942.0   220.200000              1
2        남성   1946        결혼       없슴     942.0   139.000000              1
3        남성   1953        결혼       없슴     942.0   150.000000              1
4        남성   1960        결혼       있슴     942.0   166.000000              1
...     ...    ...       ...       ...       ...          ...            ...
7524     여성   1950        결혼       있슴     819.0   241.619016              6
7525     남성   1960        결혼       있슴     111.0   250.000000              7
7526     남성   1960        결혼       없슴     111.0  1250.000000              1
7527     남성   1992       무응답       있슴     876.0   280.000000              3
7528     남성   1935        결혼       있슴     876.0   156.000000              3

      age                  job
0      73            경비원 및 검표원
1      76            경비원 및 검표원
2      75            경비원 및 검표원
3      68            경비원 및 검표원
4      61            경비원 및 검표원
...   ...                  ...
7524   71      기타 식품가공관련 기계조작원
```

```
7525   61  의회의원 고위공무원 및 공공단체임원
7526   61  의회의원 고위공무원 및 공공단체임원
7527   29    선박 갑판승무원 및 관련 종사원
7528   86    선박 갑판승무원 및 관련 종사원

[7529 rows x 9 columns]
```

STEP 08. code_religion 정보 확인

```
print("welfare['code_religion'].unique()")
print(welfare['code_religion'].unique())
```

```
welfare['code_religion'].unique()
[1 2 3 4 7 5 6]
```

STEP 09. code_religion 데이터를 문자열로 변경

```
def setReligion_txt( x ):
    if int(x) == 1 :
        return '서울'
    elif int(x) == 2 :
        return '수도권'
    elif int(x) == 3:
        return '부산/경남/울산'
    elif int(x) == 4 :
        return '대구/경북'
    elif int(x) == 5 :
        return '대전/충남'
    elif int(x) == 6 :
        return '강원/충북'
    elif int(x) == 7 :
        return '광주/전남/전북/제주도'

welfare['code_religion'] = welfare['code_religion'].apply(setReligion_txt)
```

STEP 10. 연령대를 분류

```
def newAge(x):
    if x < 30:
        return '청년'
    elif x >= 30 and x < 60:
        return '중년'
    else :
        return '노년'

welfare['ageg'] = welfare['age'].apply(newAge)

print(welfare[['age', 'ageg']].head())
```

```
   age ageg
```

```
0   73   노년
1   76   노년
2   75   노년
3   68   노년
4   61   노년
```

STEP 11. 데이터 전처리 완료 후 최종 결과물 저장

```
welfare['ageg'] = welfare['age'].apply(newAge)

print(welfare[['age', 'ageg']].head())


col_mapping = {'gender':'성별', 'birth':'생일', 'marriage':'결혼 유무', 'religion':'종교 유무', 'co
welfare = welfare.rename(columns = col_mapping)

welfare.to_csv('welfareClean.csv', index=False, encoding='cp949')
```

```
   age ageg
0   73   노년
1   76   노년
2   75   노년
3   68   노년
4   61   노년
```

STEP 12. 데이터 확인

```
print(welfare.columns)

print(welfare.head(10))
```

```
Index(['성별', '생일', '결혼 유무', '종교 유무', '직업 코드', '소득', '지역구', '나이', '직업
  성별    생일 결혼 유무 종교 유무  직업 코드         소득  지역구  나이        직업 연령
0  남성  1948   무응답    없슴  942.0  120.000000   서울  73  경비원 및 검표원  노년
1  남성  1945    이혼    없슴  942.0  220.200000   서울  76  경비원 및 검표원  노년
2  남성  1946    결혼    없슴  942.0  139.000000   서울  75  경비원 및 검표원  노년
3  남성  1953    결혼    없슴  942.0  150.000000   서울  68  경비원 및 검표원  노년
4  남성  1960    결혼    있슴  942.0  166.000000   서울  61  경비원 및 검표원  노년
5  남성  1939    결혼    있슴  942.0  241.619016   서울  82  경비원 및 검표원  노년
6  남성  1947    결혼    있슴  942.0  150.000000  수도권  74  경비원 및 검표원  노년
7  남성  1952    이혼    없슴  942.0  170.000000   서울  69  경비원 및 검표원  노년
8  남성  1949    결혼    있슴  942.0  100.000000   서울  72  경비원 및 검표원  노년
9  남성  1942    결혼    있슴  942.0  120.000000   서울  79  경비원 및 검표원  노년
```

STEP 13. 통계정보 확인

```
print(welfare.describe())
```

```
                생일         직업 코드           소득          나이
count  7529.000000  7529.000000  7529.000000  7529.000000
mean   1964.012087   591.243724   241.619016    56.987913
std      15.524029   255.793317   144.679991    15.524029
min    1919.000000   111.000000     0.000000    23.000000
25%    1952.000000   314.000000   162.600000    45.000000
50%    1965.000000   611.000000   241.619016    56.000000
75%    1976.000000   863.000000   241.619016    69.000000
max    1998.000000  1012.000000  2400.000000   102.000000
```

STEP 14. 각각의 컬럼의 unique한 값 확인

```
print(welfare['결혼 유무'].unique())

print(welfare['종교 유무'].unique())

print(welfare['지역구'].unique())

print(welfare['직업'].unique())

print(welfare['연령대'].unique())
```

```
['무응답' '이혼' '결혼']
['없음' '있음']
['서울' '수도권' '부산/경남/울산' '대구/경북' '광주/전남/전북/제주도' '대전/충남' '강원/충북'
['경비원 및 검표원' '전기공' '방문 노점 및 통신 판매 관련 종사자' '기타 서비스관련 단순 종사
 '문리 기술 및 예능 강사' '영업 종사자' '스포츠 및 레크레이션 관련 전문가' '매장 판매 종사자
 '비서 및 사무 보조원' '청소원 및 환경 미화원' '가사 및 육아 도우미' '기술영업 및 중개 관련
 '금융 및 보험관련 사무 종사자' '음식관련 단순 종사원' '판매 및 운송 관리자' '고객서비스 관리
 '냉 난방 관련 설비 조작원' '음식서비스 종사자' '농림어업관련 단순 종사원' '이 미용 및 관련
 '회계 및 경리 사무원' '식품가공관련 기능 종사자' '상품 대여 종사자' '고객 상담 및 기타 사무
 '기계장비 설치 및 정비원' '금속공작기계 조작원' '전기 전자 및 기계 공학 기술자 및 시험원'
 '건축 및 토목 공학 기술자 및 시험원' '행정 사무원' '섬유 및 가죽 관련 기능 종사자' '판매관련
 '제조관련 단순 종사원' '전기 전자 부품 및 제품 제조장치 조작원' '건설 및 광업 단순 종사원'
 '전기 및 전자기기 설치 및 수리원' '배달원' '디자이너' '건설 및 채굴 기계운전원' '전기 및 전
 '의료 복지 관련 서비스 종사자' '간호사' '정보 시스템 운영자' '학교 교사' '기타 교육 전문가'
 '정보통신관련 관리자' '치료사 및 의료기사' '컴퓨터 하드웨어 및 통신공학 전문가' '자동차 정비
 '건설 전기 및 생산 관련 관리자' '의복 제조관련 기능 종사자' '보건의료관련 종사자' '매니저 및
 '작가 기자 및 출판 전문가' '금형 주조 및 단조원' '법률 전문가' '사회복지관련 종사자' '주방장
 '전기 전자 부품 및 제품 조립원' '정보시스템 개발 전문가' '경찰 소방 및 교도 관련 종사자' '기
 '용접원' '영상 및 통신 장비 관련 설치 및 수리원' '기타 건설 전기 및 생산 관련 관리자' '목재
 '세탁관련 기계조작원' '기타 판매 및 고객서비스 관리자' '종교관련 종사자' '물품이동 장비 조작
 '인쇄 및 사진현상 관련 기계조작원' '의료진료 전문가' '기타 공학 전문가 및 관련 종사자' '공예
 '인사 및 경영 전문가' '철도 및 전동차 기관사' '통계관련 사무원' '임업관련 종사자' '연구 교육
 '운송 서비스 종사자' '식품가공관련 기계조작원' '금융 및 보험 전문가' '기타 이미용 예식 및 의
 '생명 및 자연 과학 관련 시험원' '도장 및 도금기 조작원' '연극 영화 및 영상 전문가' '화가 사
 '문화 예술 디자인 및 영상 관련 관리자' '작물재배 종사자' '안전 관리 및 검사원' '배관공' '금
 '환경공학 기술자 및 시험원' '발전 및 배전 장치 조작원' '하역 및 적재 단순 종사원' '유치원 교
 '주조 및 금속 가공관련 기계조작원' '법률 및 감사 사무 종사자' '금속 재료 공학 기술자 및 시험
 '기타 제조관련 기계조작원' '기타 전문서비스 관리자' '여가 및 스포츠 관련 종사자' '인문 및 사
 '혼례 및 장례 종사자' '목재 가구 악기 및 간판 관련 기능 종사자' '상품기획 홍보 및 조사 전문
 '생명 및 자연 과학 관련 전문가' '경호 및 보안 관련 종사자' '행정 및 경영지원 관리자'
 '계기검침 수금 및 주차 관련 종사원' '운송차량 및 기계 관련 조립원' '직물 및 신발 관련 기계조
 '석유 및 화학물 가공장치 조작원' '자동조립라인 및 산업용 로봇 조작원' '화학 고무 및 플라스틱
```

```
 '원예 및 조경 종사자' '섬유제조 및 가공 기계조작원' '축산 및 사육 관련 종사자' '채굴 및 토목
 '보험 및 금융 관리자' '장교' '항공기 선박 기관사 및 관제사' '환경 청소 및 경비 관련 관리자'
 '보건 및 사회복지 관련 관리자' '통신 및 방송송출 장비 기사' '재활용 처리 및 소각로 조작원'
 '화학공학 기술자 및 시험원' '어업관련 종사자' '제관원 및 판금원' '비금속 제품 생산기 조작원
 '상 하수도 처리장치 조작원' '음료제조관련 기계조작원' '기타 식품가공관련 기계조작원' '의회의
 '선박 갑판승무원 및 관련 종사원']
['노년' '중년' '청년']
```

STEP 15. 그래프를 이미지 파일로 저장하는 함수 작성

```
def FileSave():
    global cnt
    cnt += 1
    savefile = CHART_NAME + UNDERBAR + str(cnt).zfill(2) + PNG
    plt.savefig(savefile, dpi=400)
    print(savefile + ' 파일이 저장되었습니다.')
# end def FileSave():
```

4.2.4 척도에 대한 이해

▸ 4.2.5 결혼 유무와 종교 유무에 따른 빈도(countplot)

[ ] ↳ *숨겨진 셀 14개*

▸ 4.2.6 나이에 따른 히스토그램

[ ] ↳ *숨겨진 셀 8개*

▸ 4.2.7 결혼 유무와 성별에 따른 히트맵

[ ] ↳ *숨겨진 셀 7개*

▸ 4.2.8 두 컬럼간의 짝 그래프

[ ] ↳ *숨겨진 셀 10개*

▸ 4.2.9 성별과 나이에 따른 바이올린 그래프

↳ *숨겨진 셀 1개*

4.2.10 선형 회귀 모델 그래프

↳ *숨겨진 셀 1개*

4.2.11 나이와 소득에 따른 산점도

↳ *숨겨진 셀 1개*

4.2.12 나이와 소득에 따른 산점도와 히스토그램

↳ *숨겨진 셀 1개*

4.2.13 성별에 따른 소득 그래프

↳ *숨겨진 셀 1개*

4.2.14 성별에 따른 상자 수염 그래프

↳ *숨겨진 셀 1개*

4.2.15 소득에 따른 나이

↳ *숨겨진 셀 2개*

✓ 0초 오후 4:11에 완료됨