데이터 분석 기초
Fundamental of data analysis

Exercise 12

Notice

- Exercise 12는 colab과 goorm을 병행하여 사용합니다.
- Colab은 '시각화'가 가능함으로 여러분이 시각화된 그래프를 보는 용도로,
- Goorm은 채점에 사용되게 됩니다.
- Goorm에서 시각화가 불가능하여 이와 같은 방법을 쓰게 되었고, 기말 시험 때도 이와 같은 풀이 절차를 밟아야 하므로 실습으로 연습해주시기 바랍니다.

0 과목 간 상관 계수 출력

- 한 교실의 국어, 영어, 수학, 과학 점수가 주어졌습니다. 데이터프레임을 정의한 뒤 (이론 수업 자료 참고) corr() 함수로 상관 계수를 구한 뒤 input으로 두 과목이 입력되면 두과목의 상관 계수를 출력하세요.

0 과목 간 상관 계수 출력

입출력 예시

```
프로세스가 시작되었습니다.(입력값을 직접 입력해 주세요)
> korean
english
0.892973079875521
```

```
프로세스가 시작되었습니다.(입력값을 직접 입력해 주세요)
> math
science
0.8435334152290931
```

```
프로세스가 시작되었습니다.(입력값을 직접 입력해 주세요)
> korean
science
0.40734470561031916
```

1 가장 높은 상관 계수 과목 구하기

- 과목이 많을 때 상관 계수를 보기 불편할 수도 있습니다.
- 한 과목이 어떤 과목과 제일 상관 계수가 높은 지 확인하고 싶습니다.
- 주어진 과목들의 점수에 대하여 data frame을 정의하고 corr() 함수로 상관 계수를 구한 뒤 Input으로 입력된 한 과목에 대해 해당 과목을 제외하고 가장 상관 계수가 높은 과목과 그 때 상관 계수를 출력하세요.

1 가장 높은 상관 계수 과목 구하기

```
Korean=[100, 90, 90, 95, 85, 80, 75, 80, 75, 70, 65, 65, 50, 40, 50, 50]
English=[100, 95, 95 ,90, 85, 80, 85, 70, 70, 60, 65, 55, 60, 60, 50, 50]
Math=[100, 90, 95, 85, 80, 70, 60, 70, 85, 80, 95, 45, 80, 85, 50, 75]
Science=[100, 90, 90, 85, 85, 95, 50, 70, 85, 70, 80, 30, 85, 75, 50, 85]
Sports = [100, 100, 80, 100, 100, 90, 70, 60, 80, 95, 70, 70, 85, 95, 95, 100]
Music = [100, 100, 100, 95, 100, 85, 75, 90, 85, 95, 85, 70, 75, 90, 85, 85]
Moral = [100, 85, 75, 85, 90, 100, 85, 75, 90, 85, 100, 100, 75, 90, 95, 60]
```

과목 점수

1 가장 높은 상관 계수 과목 구하기

입출력 예시

```
프로세스가 시작되었습니다.(입력값을 직접 입력해 주세요)
> korean
english 0.892973079875521
```

```
프로세스가 시작되었습니다.(입력값을 직접 입력해 주세요)
> sports
music 0.5368472552189569
```

```
프로세스가 시작되었습니다.(입력값을 직접 입력해 주세요)
> music
math 0.682551385744504
```

```
프로세스가 시작되었습니다.(입력값을 직접 입력해 주세요)
> moral
korean 0.1709890230315441
```

2 WHR 데이터 준비

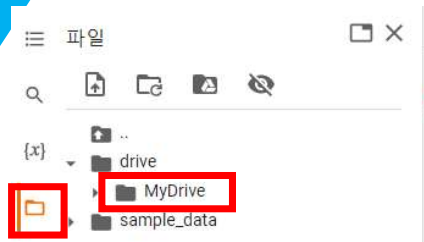
- Colab에서 WHR 데이터를 불러온 뒤 전처리까지 실행해봅니다.
- Goorm에는 데이터가 존재하며 전처리를 진행한 결과를 출력하는 것이 목표입니다.

2 Colab – WHR 데이터 준비

- Kaggle 사이트 혹은 icampus에서 데이터 셋을 다운로드 해주세요.
  https://www.kaggle.com/datasets/ajaypalsinghlo/world-happiness-report-2022?resource=download

- 데이터를 불러온 뒤 결측, 중복치가 있는 지 확인 후 있으면 제거합니다.
- 데이터의 가독성이 떨어지므로 각 지표의 이름을 바꿉니다.
- 필요 없는 데이터 열을 제거합니다.

2 Colab – WHR 데이터 준비

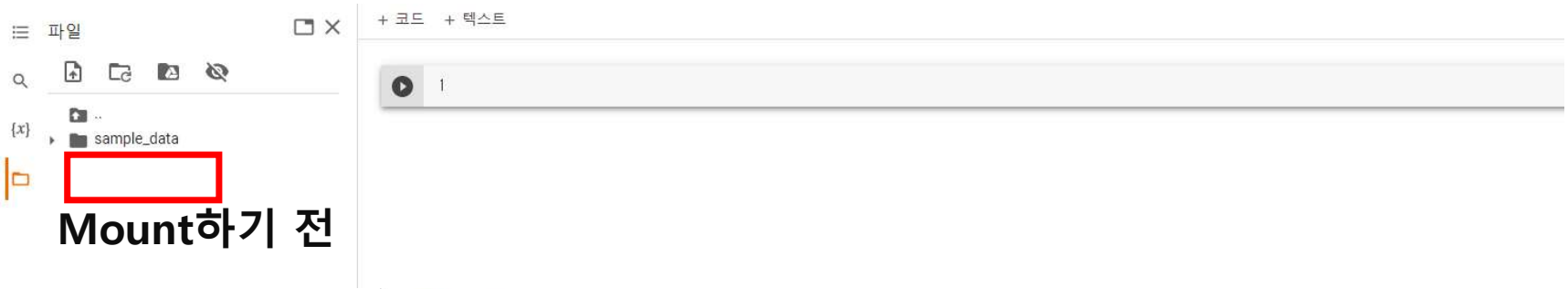


일단 drive.mount 코드를 실행하면 MyDrive 경로가 생깁니다.

```
+ 코드  + 텍스트

[2]  1 import pandas as pd
     2 from google.colab import drive                # 모듈
     3 drive.mount('/content/drive')                 # 본인의 구글 드라이브와 colab을 연결
     4 whr = pd.read_csv('/content/drive/MyDrive/etc/whr_2022.csv') #주의!! 드라이브에 world happiness report 데이터 셋을 업로드하고 그 경로를 읽어와야 합니다!!
     5 data = whr.copy()
     6 data.info()

Drive already mounted at /content/drive; to attempt to forcibly remount, call drive.mount("/content/drive", force_remount=True).
---------------------------------------------------------------------------
FileNotFoundError                         Traceback (most recent call last)
<ipython-input-2-d35ed8fleeaf> in <module>()
      2 from google.colab import drive                # 모듈
      3 drive.mount('/content/drive')                 # 본인의 구글 드라이브와 colab을 연결
----> 4 whr = pd.read_csv('/content/drive/MyDrive/etc/whr_2022.csv') #주의!! 드라이브에 world happiness report 데이터 셋을 업로드하고 그 경로를 읽어와야 합니다!!
      5 data = whr.copy()
      6 data.info()

                                    7 frames
/usr/local/lib/python3.7/dist-packages/pandas/io/common.py in get_handle(path_or_buf, mode, encoding, compression, memory_map, is_text, errors, storage_options)
    705                 encoding=ioargs.encoding,
    706                 errors=errors,
--> 707                 newline="",
    708             )
    709         else:

FileNotFoundError: [Errno 2] No such file or directory: '/content/drive/MyDrive/etc/whr_2022.csv'
```

4번째 줄에 경로를 올바르게 설정하지 않아서 뜨는 에러

2 Colab – WHR 데이터 준비



Mount하기 전

Mount한 후

2 Colab – WHR 데이터 준비

- Colab에서 따라해보세요 – 드라이브 마운트 및 데이터 불러오기

```
import pandas as pd                          # 모듈들 불러오기
from google.colab import drive
import matplotlib.pyplot as plt
import seaborn as sns
drive.mount('/content/drive')                # 본인의 구글 드라이브와 colab을 연결
whr = pd.read_csv('/content/drive/MyDrive/etc/whr_2022.csv') #주의!! 드라이브에 world happiness report 데이터 셋을 업로드하고 그 경로를 읽어와야 합니다!!
```

2 Colab – WHR 데이터 준비

- Colab에서 따라해보세요 – 데이터 copy, 필요 없는 정보 제거, 열 이름 재정의

```
 9 data = whr.copy()
10 data = data.drop(['Whisker-high', 'Whisker-low' ], axis=1).copy() #필요 없는 정보 drop 후 데이터 copy
11 data.columns = ['rank', 'country', 'happy_score', 'residual', 'gdp', 'social_support', 'health', 'freedom', 'generosity', 'trust'] #데이터 열 이름 새로 정의
```

2 Colab – WHR 데이터 준비

- Colab에서 따라해보세요 – 중복, 결측 데이터 체크 -> 중복치, 결측치 없는 거 확인

```
print( data.duplicated().sum() ) # 중복데이터 체크
print( data.isnull().sum() ) # 결측 데이터 체크
```

```
0
rank              0
country           0
happy_score       0
residual          0
gdp               0
social_support    0
health            0
freedom           0
generosity        0
trust             0
dtype: int64
```

2 Colab – WHR 데이터 준비

- Colab에서 따라해보세요 – 중복, 결측 데이터 체크 -> 중복치, 결측치 없는 거 확인

```
1 print( data.duplicated().sum() ) # 중복데이터 체크
2 print( data.isnull().sum() ) # 결측 데이터 체크
```

```
0
rank              0
country           0
happy_score       0
residual          0
gdp               0
social_support    0
health            0
freedom           0
generosity        0
trust             0
dtype: int64
```

중복, 결측 데이터 체크

```
1 data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 146 entries, 0 to 145
Data columns (total 10 columns):
 #   Column          Non-Null Count  Dtype
---  ------          --------------  -----
 0   rank            146 non-null    int64
 1   country         146 non-null    object
 2   happy_score     146 non-null    float64
 3   residual        146 non-null    float64
 4   gdp             146 non-null    float64
 5   social_support  146 non-null    float64
 6   health          146 non-null    float64
 7   freedom         146 non-null    float64
 8   generosity      146 non-null    float64
 9   trust           146 non-null    float64
dtypes: float64(8), int64(1), object(1)
memory usage: 11.5+ KB
```

데이터 정보

2 Goorm – WHR 데이터 준비

- Goorm exercise2 에서는 colab에서의 드라이브 마운트 부분은 제외하고
  1. 중복 데이터 체크 2. 결측 데이터 체크 3. 데이터 정보를 출력해주시면 됩니다.

2 Goorm – WHR 데이터 준비

출력 예시

```
> 0
rank              0
country           0
happy_score       0
residual          0
gdp               0
social_support    0
health            0
freedom           0
generosity        0
trust             0
dtype: int64
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 146 entries, 0 to 145
Data columns (total 10 columns):
 #   Column          Non-Null Count  Dtype
---  ------          --------------  -----
 0   rank            146 non-null    int64
 1   country         146 non-null    object
 2   happy_score     146 non-null    float64
 3   residual        146 non-null    float64
 4   gdp             146 non-null    float64
 5   social_support  146 non-null    float64
 6   health          146 non-null    float64
 7   freedom         146 non-null    float64
 8   generosity      146 non-null    float64
 9   trust           146 non-null    float64
dtypes: float64(8), int64(1), object(1)
memory usage: 11.5+ KB
```

3 WHR 상관 계수 분석

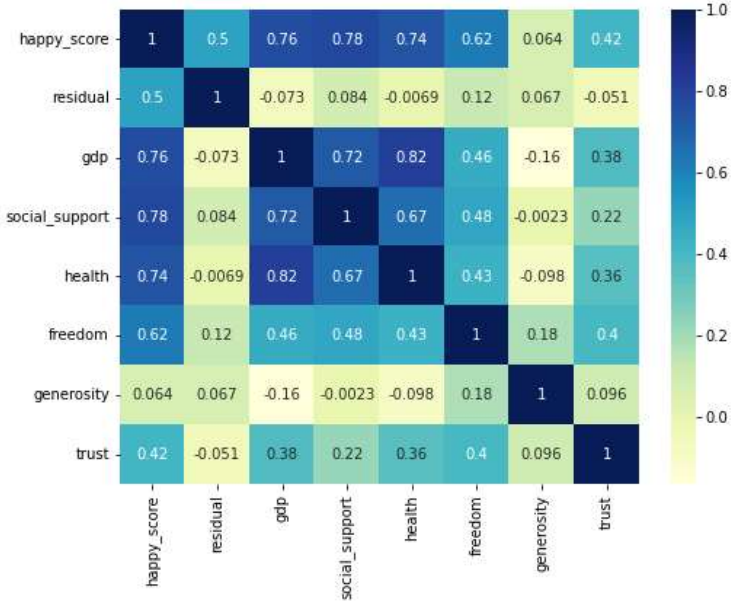
- Exercise 3은 WHR 각 지표간 상관 계수를 분석 할 것입니다. 각 지표가 어떤 지표와 가장 상관 계수가 높은지, 낮은지를 확인해 볼 것입니다. Colab은 2번의 노트북 파일을 이어서 사용해주세요.

3 Colab – WHR 상관 계수 분석

- Rank와 country 열 삭제 – rank는 1~146까지의 순위라서. Country는 문자열

```
data.drop(['rank', 'country'],axis=1, inplace=True)
corr_res = data.corr()
```

```
plt.figure(figsize=(8, 6)) # heatmap 크기
sns.heatmap(corr_res, annot=True, cmap='YlGnBu')
plt.show()
```

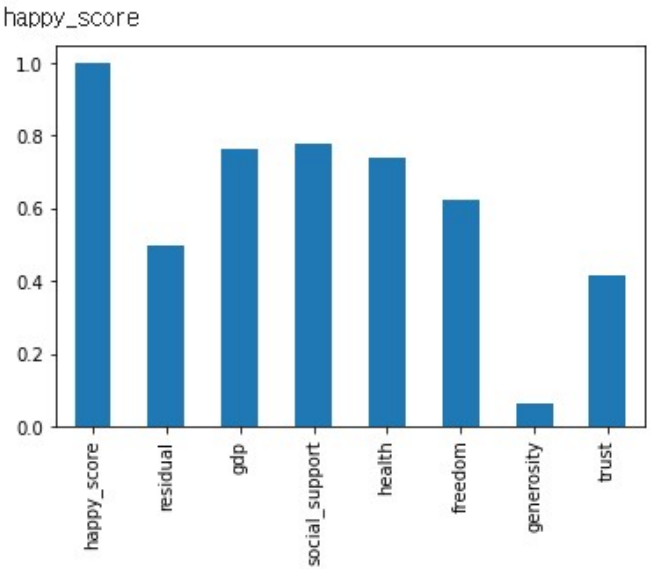


3 Colab – WHR 상관 계수 분석

- Input으로 입력 받은 지표에 대해서 상관계수 시각화. Input이 'happy_score'라서 happy_score와의 상관계수들을 확인할 수 있습니다. 가장 큰 값 social_support, 가장 낮은 값 generosity

```
1 input1 = input()
2 corr_res.loc[input1].plot.bar()
3 plt.show()
```

input



3 Goorm – WHR 상관 계수 분석

- Colab에서 수행한 시각화 과정은 제외하고 input으로 입력받은 지표에 대하여 해당 지표를 제외하고 최대, 최소 상관 계수를 갖는 지표와 값을 순서대로 출력해주세요.

3 Goorm – WHR 상관 계수 분석

입출력 예시

```
> happy_score
social_support 0.777888565189765
generosity 0.06378454924698192
```

```
> residual
happy_score 0.49899047099449895
gdp -0.07342324905295944
```

```
> gdp
health 0.8153860139557864
generosity -0.16447236608835775
```

4 WHR 파생 변수

- Exercise 4에선 주어진 데이터들을 가지고 새로운 변수를 만들어 볼 것입니다.
- 새로운 변수를 만들어 내고 그 변수를 기준으로 지표들의 평균 등을 알 수 있습니다.
- 또 새로운 변수와 기존 변수들과의 관계 또한 시각화해서 볼 수 있습니다.

4 Colab - WHR 파생 변수

```
1 nums = data.shape[0]
2 # 행복지수로 정렬되어 있으므로 1/3 지점, 2/3 지점의
3 # 행복지수 값을 가져옴
4 h_border = data.iloc[ int(nums/3) ]['happy_score']
5 m_border = data.iloc[ int(nums/3) * 2 ]['happy_score']
6 print( h_border, m_border) #각 지점 확인
```

```
6.12 5.122
```

```
1 #함수 정의
2 def encoding_group_rank(x):
3   if x >= h_border:
4     return 'H'
5   elif x >= m_border:
6     return 'M'
7   else:
8     return 'L'
```

4 Colab - WHR 파생 변수

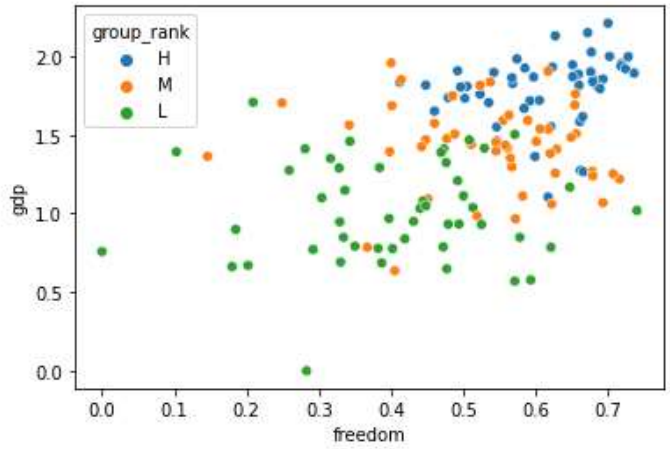
```
1 data['group_rank'] = data['happy_score'].apply(encoding_group_rank)
```

```
1 data.groupby('group_rank').mean().sort_values('happy_score', ascending=False)
```

group_rank	happy_score	residual	gdp	social_support	health	freedom	generosity	trust
H	6.711204	2.080980	1.789449	1.131143	0.721816	0.612204	0.149531	0.226163
M	5.597313	1.789542	1.428333	0.960167	0.619542	0.535229	0.145750	0.118813
L	4.353102	1.624041	1.013918	0.627388	0.417837	0.404612	0.146816	0.118633

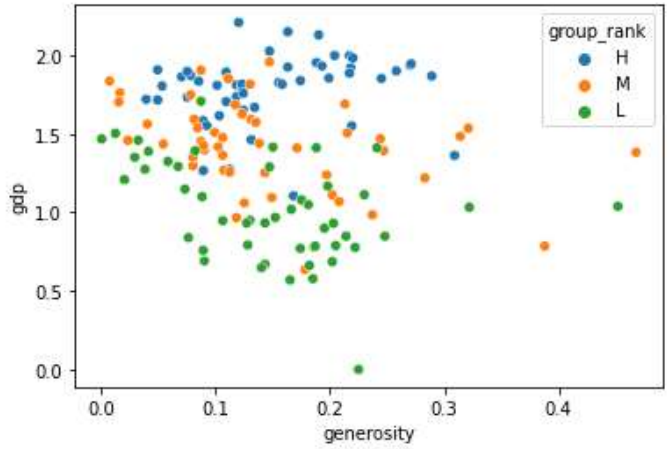
4 Colab - WHR 파생 변수

```
#그래프 분석
sns.scatterplot( x='freedom', y='gdp',hue='group_rank', data=data )
plt.show()
```



4 Colab - WHR 파생 변수

```
1 sns.scatterplot( x='generosity', y='gdp', hue='group_rank', data=data )
2 plt.show()
3 #group_rank가 freedom, gdp와는 크게 상관이 있고, generosity와는 비교적 큰 관계는 없는 것을 확인 할 수 있습니다.
```



4 Goorm - WHR 파생 변수

- Colab에서와 한 것 같이 데이터를 3그룹 H, M, L로 나눕니다.
- 그룹_H의 mean을 그룹_L의 mean으로 나눈 값 중 가장 높은 값을 갖는 변수 2개를 높은 순서대로 변수 명과 값을 출력해주세요.

group_rank	happy_score	residual	gdp	social_support	health	freedom	generosity	trust
H	6.711204	2.080980	1.789449	1.131143	0.721816	0.612204	0.149531	0.226163
L	4.353102	1.624041	1.013918	0.627388	0.417837	0.404612	0.146816	0.118633

H/L ? ? ? ? …

4 Goorm - WHR 파생 변수

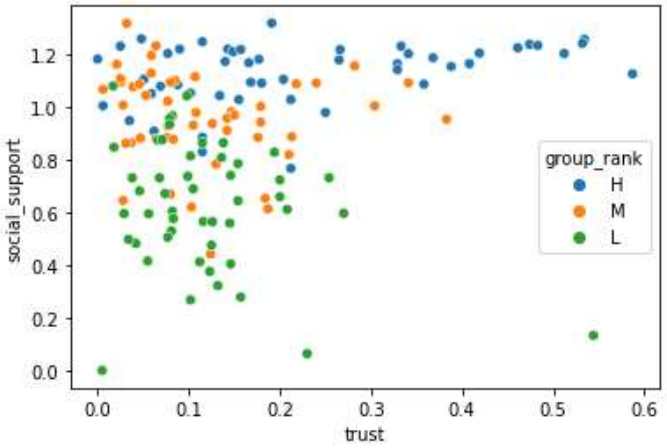
출력 예시

```
trust 1.9064166523309825
social_support 1.8029406024331536
```

4 Colab - WHR 파생 변수

- 코랩에서 가장 높은 변수 2가지를 기준으로 그래프를 그려봅시다.

```
1 sns.scatterplot( x='trust', y='social_support', hue='group_rank', data=data )
2 plt.show()
```



Thank you