

```
1 from google.colab import files
2 uploaded = files.upload() #로컬PC에서 파일 선택
```

파일 선택 gapminder.tsv

- **gapminder.tsv**(n/a) - 83637 bytes, last modified: 2021. 4. 19. - 100% done
Saving gapminder.tsv to gapminder (1).tsv

```
1 ! pip list #설치되어있는 라이브러리 확인
```

pydot-ng	2.0.0
pydotplus	2.0.2
PyDrive	1.3.1
pyemd	0.5.1
pyerfa	1.7.2
pyglet	1.5.0
Pygments	2.6.1
pygobject	3.26.1
pymc3	3.7
PyMeeus	0.5.11
pymongo	3.11.3
pymystem3	0.2.0
PyOpenGL	3.1.5
pyarsing	2.4.7
pyrsistent	0.17.3
pysndfile	1.3.8
PySocks	1.7.1
pystan	2.19.1.1
pytest	3.6.4
python-apt	0.0.0
python-chess	0.23.11
python-dateutil	2.8.1
python-louvain	0.15
python-slugify	4.0.1
python-utils	2.5.6
pytz	2018.9
pyviz-comms	2.0.1
PyWavelets	1.1.1
PyYAML	3.13
pyzmq	22.0.3
qdlidl	0.1.5.post0
qtconsole	5.0.3
QtPy	1.9.0
regex	2019.12.20
requests	2.23.0
requests-oauthlib	1.3.0
resampy	0.2.2
retrying	1.3.3
rpy2	3.4.3
rsa	4.7.2
scikit-image	0.16.2
scikit-learn	0.22.2.post1
scipy	1.4.1
screen-resolution-extra	0.0.0
scs	2.1.3
seaborn	0.11.1
Send2Trash	1.5.0
setuptools	54.2.0
setuptools-git	1.2

Shapely	1.7.1
simplegeneric	0.8.1
six	1.15.0
sklearn	0.0
sklearn-pandas	1.8.0
smart-open	5.0.0
snowballstemmer	2.1.0
sortedcontainers	2.3.0
SoundFile	0.10.3.post1
spacy	2.2.4
shiny	1.8.5

```

1 import pandas as pd
2 df = pd.read_csv('gapminder.tsv', sep = '\t') #파일을 df라는 변수에 저장
3 df.head(10) #데이터를 0부터 9번까지 확인
4 df.info() #데이터의 정보 확인

```

```

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1704 entries, 0 to 1703
Data columns (total 6 columns):
#   Column      Non-Null Count  Dtype
---  -
0   country     1704 non-null   object
1   continent   1704 non-null   object
2   year        1704 non-null   int64
3   lifeExp     1704 non-null   float64
4   pop         1704 non-null   int64
5   gdpPercap   1704 non-null   float64
dtypes: float64(2), int64(2), object(2)
memory usage: 80.0+ KB

```

```

1 # count=df['country'] #열 이름으로 검색 (시작과 끝부분) #count라는 변수에 저장
2 count=df[['country', 'continent', 'year']] #내가 원하는 항목들만 보기
3 count
4 # df.tail() #끝부분 5개 출력

```

```
1 count.iloc[0] # iloc= 행번호로 뽑기, loc = 인덱스로 뽑기
```

```
country    Afghanistan
continent   Asia
year       1952
Name: 0, dtype: object
```

```
3    Afghanistan    Asia    1967
```

```
1 count.iloc[-1] #행 번호이기 때문에 마지막 번호 출력
```

```
2 # count.loc[-1] #인덱스 번호이기 때문에 에러 발생
```

```
country    Zimbabwe
continent   Africa
year       2007
Name: 1703, dtype: object
```

```
1701    Zimbabwe    Africa    1997
```

```
1 df.iloc[[0,99,199]] #여러개의 항목을 선택할때 [[]] <-- 이중 대괄호
```

	country	continent	year	lifeExp	pop	gdpPercap
0	Afghanistan	Asia	1952	28.801	8425333	779.445314
99	Bangladesh	Asia	1967	43.453	62821884	721.186086
199	Burkina Faso	Africa	1987	49.557	7586551	912.063142

```
1 df.loc[[0],['year']]
```

```
2 # df.iloc[[0],['year']]는 행번호이기 때문에 에러 발생
```

	year
0	1952

```
1 df.loc[:,['year','pop']] #처음부터 행 전체, 열
```

	year	pop
0	1952	8425333

1 df.iloc[:,list(range(3,6)))] # 행은 처음부터, 3~5번까지의 열에 해당하는 값 출력

	lifeExp	pop	gdpPer cap
0	28.801	8425333	779.445314
1	30.332	9240934	820.853030
2	31.997	10267083	853.100710
3	34.020	11537966	836.197138
4	36.088	13079460	739.981106
...
1699	62.351	9216418	706.157306
1700	60.377	10704340	693.420786
1701	46.809	11404948	792.449960
1702	39.989	11926563	672.038623
1703	43.487	12311143	469.709298

1704 rows × 3 columns

1 group_y = df.groupby('year') #연도 별로 그룹핑
2 group_y['lifeExp'].mean()

```

↳ year
1952    49.057620
1957    51.507401
1962    53.609249
1967    55.678290
1972    57.647386
1977    59.570157
1982    61.533197
1987    63.212613
1992    64.160338
1997    65.014676
2002    65.694923
2007    67.007423
Name: lifeExp, dtype: float64

```

✓ 0초 오후 3:56에 완료됨

