

라이브 코칭 2회차

Data Science 2024

잠시 후 오후 8시, 코칭스터디(Data Science 2024) LIVE가 진행됩니다

오늘의 순서

#2회차 라이브
서울 종합병원 분포 확인하기

코칭스터디 공지사항

- ✓ 팀 재조정 안내
- ✓ 슬랙 채널 안내
- ✓ 부스터 이벤트
- ✓ 2주차 학습계획/미션

라이브 코칭

- ✓ 라이브 코칭

QnA

- ✓ 라이브 코치에게 물어봐
- ✓ 실시간 QnA

공지사항

슬랙 채널 안내

#00_운영진 공지사항

#01_운영관련 qna

#01_자주묻는질문 faq

#02_자유게시판

#03_코치에게 물어봐

#04_주차별 우수미션

#05_리드부스터 공지사항

#99_코딩코치 자기소개

 00코치_00팀

 00코치_미션제출

 00코치_자유게시판

#01_운영관련 qna

- 부스트코스-학습강좌/스터디강좌 관련
- 팀활동 관련 질문
- 과업 인증 관련 질문

#03_코치에게-물어봐

- 학습방법: 어떻게 공부해야 할까요?
- 진로: 대학원은 가는 것이 좋을까요?
- 현업: 프로들은 어떤 것에 관심이 많나요?

#00코치_자유게시판

- 프로그램 설치 관련
- 미션/퀴즈 공부 중 막힐 때!



팀 재조정 안내

끝까지 함께 달려보아요🍀

- ✓ 코칭스터디는 무료/온라인 형태의 교육으로 프로그램 **초반 탈락자가 발생합니다.**
- ✓ 그렇지만, 열심히 학습해주시는 리드부스터&부스터분들의 학습은 계속 되어야 합니다!
- ✓ 리드부스터가 제출하는 **활동일지+팀 모니터링**으로 활동을 파악하고 있습니다.
- ✓ 팀 활동이 불가능한 팀에 대하여 우선적으로 **팀 재조정 및 리드부스터 재선발**을 실시하겠습니다.
- ✓ 팀원이 조금씩 변동되더라도, 이는 학습의지가 높은 분들의 학습을 지원하고자 함이니 너그라이 이해해주시면 감사하겠습니다.
- ✓ 추가로 팀 조정 관련 공지가 나간 팀은 내일부터 순차적으로 조정이 이루어질 예정이니 조금만 기다려 주세요!
- ✓ 문의사항은 #01_운영관련qna 채널에 남겨주시거나, 운영진에게 슬랙DM 보내주세요.

우수 참가자 리워드

#열정적인 여러분을 위한 깜짝 이벤트



#팀활동왕(팀)

우수미션 & MVP
최다 선정 팀!

#소통왕(개인) 3명

자유게시판, QnA 등
동료부스터 질문에 적극적으로 답변
좋은 학습 자료를 아낌없이 공유
슬랙에서 열심히 소통



열심히 달려갈 여러분을 위해
수료 리워드까지
준비했으니, 끝까지 함께
달려보아요! 

2주차 학습계획/미션

부스트코스 - [스터디강좌]에서 확인하기

필독 | 오리엔테이션

FAQ | 자주묻는 질문

학습 :: 미션 :: 라이브 안내

2주차 | 서울 종합병원 분포 확인하기

○ 2주차 | 학습 계획 & 학습 범위

○ 2주차 | 미션

○ 2주차 | 라이브 코칭 :: 실시간 시청

○ 2주차 | 라이브 코칭 :: VOD 다시보기

♥ ♥ 2주차 학습 계획 ♥ ♥

1) 파이썬으로 시작하는 데이터 사이언스 강좌 수강하기(아래의 학습 범위)

- (1) 서울 종합병원 분포 확인하기
- (2) QUIZ 2 풀기
 ►► Quiz 2 : <https://www.boostcourse.org/ds112/joinLectures/28141>

2) 퀴즈 인증 제출하기(① 08월 11일 일요일 23:59까지!)

- 위의 QUIZ 2 풀이 후 화면 캡쳐해서 슬랙에 업로드하기
- 슬랙(코칭스터디 <Data Science>) → 본인 팀 채널에 업로드 (00코치_01~10팀)

3) 라이브 코치님께 질문 남기기(❤ 08월 08일 목요일 18:00까지!)

- 라이브 코치님께 궁금하신 사항이 있으신 분들은 자유롭게 남겨주세요!
- 슬랙(코칭스터디 <Data Science>) → 03-코치에게-들어봐 채널에 남겨주세요!

⌚ 📣 2주차 미션 내용을 알려드립니다 📣 ⌚

부스터 여러분들, 2주차 강의는 잘 들으셨나요?!

학습한 내용을 토대로 풀어야 할 2주차 미션 내용을 아래와 같이 공개합니다!

미션 내용은 팀원들과 함께 풀이해주세요!(적극적인 토론이 필요합니다!)

▣ 매주 일요일 23:59까지 리드부스터가 제출해주세요! 모두들 화이팅입니다!

◆ 미션에 도전하기 전에 먼저!!

이번 미션에 활용되는 데이터를 다운로드 받기 위해, 주피터 노트북에서 다음 코드를 먼저 실행해주세요.

나의 컴퓨터 환경에 데이터를 저장하지 않아도, 웹사이트에서 바로 데이터를 받아올 수 있습니다.

잘못한 피드백을 위해 미션을 제출할 때에도 아래 코드를 꼭 포함해서 제출해주세요!

```
import pandas as pd
df = pd.read_csv("https://raw.githubusercontent.com/corazzon/boostcourse-ds-510/master/data/medical_201909
df.shape
```

◆ Q1. 전국 시도별 약국수를 구해주세요!

- 상권업종소분명이 약국인 것을 찾아 빈도수를 구해주세요. 이 때, value_counts, groupby, pivot_table 등 다양한 집계 방법을 통해 구해볼 수 있습니다. 각자 구하기 편한 방법을 통해 빈도수를 구합니다.
- 다음의 결과가 나오도록 구합니다.

▣ 출력 예시

경기도	4510
서울특별시	3579

리드부스터 활동일지

8월 11일 일요일 23시 59분까지

💡 슬랙에서 과업을 인증하면 해당 주차에 체크✓ 해주세요.

❗ 작성하는 공간이 아닙니다

퀴즈	1주차	2주차	3주차	4주차	수료 기준: 75% 이상
홍길동_리더	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	0%
	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
미션	1주차	2주차	3주차	4주차	수료 기준: 75% 이상
홍길동_리더	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	0%
	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
라이브	1주차	2주차	3주차	4주차	수료 기준: 75% 이상
홍길동_리더	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	0%
	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	

1주차 미션 정리

미션 간단 풀이

1주차 미션 솔루션 공유

1주차 미션 출제 의도

1번 [파이썬 기초] 리스트, 반복문, 조건문 이해하기

2번 [판다스 집계 연산] groupby, pivot_table로 데이터 집계하기

3번 [마크다운] Jupyter에서 문서화하기

4번 [파일 경로 확인] 판다스로 파일 불러오기

1주차 미션 - 1번

[파이썬 기초] 리스트, 반복문, 조건문 이해하기

💡 Q1. 여러분은 파이썬을 통해 설문조사 문항의 응답내역을 분석하게 되었습니다. 문항별 응답내용에는 하나의 응답만 할 수 있는 single choice 문제와 여러 응답을 선택할 수 있는 multiple choice 문제가 있습니다. 2개를 구분하기 위해 single choice 문항 번호에 "_"를 표기하지 않기로 했습니다.
문항별 응답내역이 'question'에 담겨 있을 때, 조건문과 반복문을 사용하여 아래와 같은 결과가 출력되도록 코드를 작성해보세요.

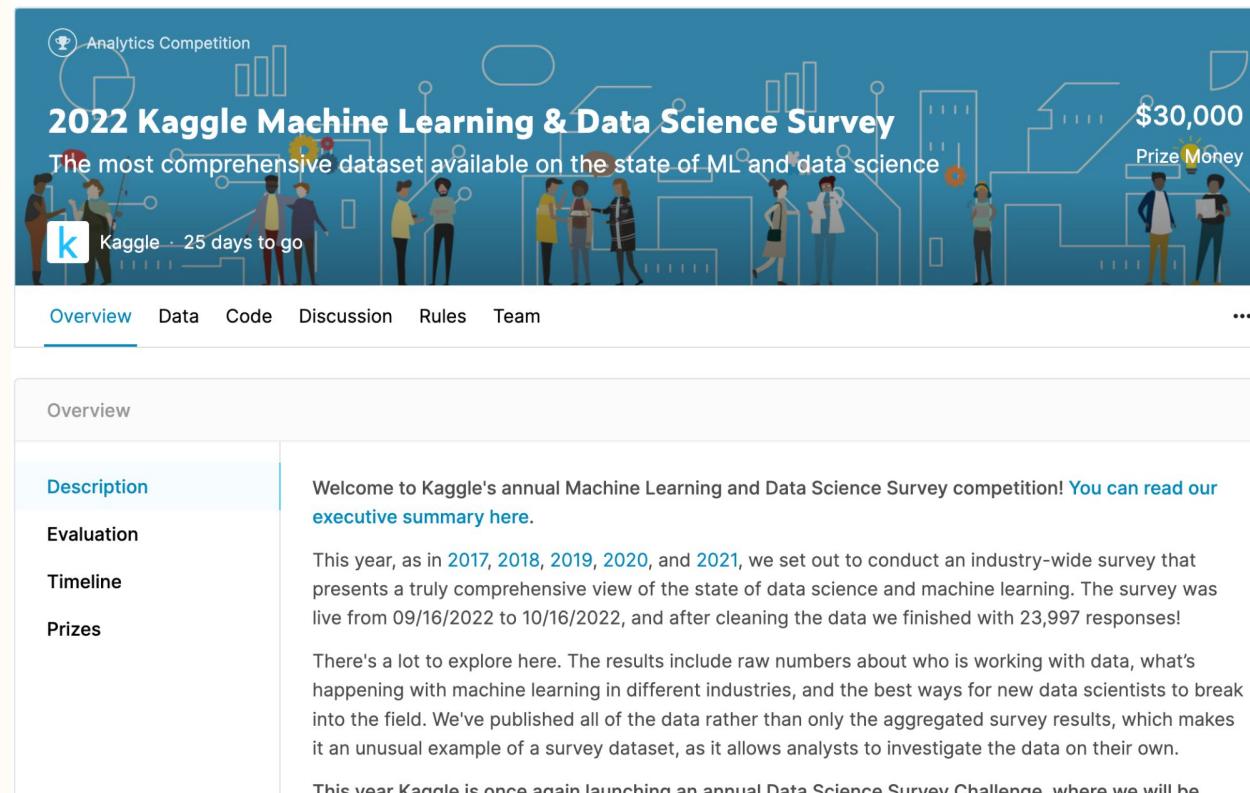
결과 출력 예시

['Q2', 'Q3', 'Q4', 'Q5', 'Q8', 'Q9']

```
question = ['Q2', 'Q3', 'Q4', 'Q5', 'Q6_1', 'Q6_2', 'Q6_3', 'Q6_4', 'Q6_5', 'Q6_6',
            'Q6_7', 'Q6_8', 'Q6_9', 'Q6_10', 'Q6_11', 'Q6_12', 'Q7_1', 'Q7_2',
            'Q7_3', 'Q7_4', 'Q7_5', 'Q7_6', 'Q7_7', 'Q8', 'Q9', 'Q10_1', 'Q10_2',
            'Q10_3']
```

1주차 미션 - 1번

[파이썬 기초] 리스트, 반복문, 조건문 이해하기



The screenshot shows the homepage of the Kaggle Machine Learning & Data Science Survey 2022. At the top, there's a banner with the text "Analytics Competition", "2022 Kaggle Machine Learning & Data Science Survey", "The most comprehensive dataset available on the state of ML and data science", "Kaggle - 25 days to go", and "\$30,000 Prize Money". Below the banner, there's a navigation bar with tabs: Overview (which is selected), Data, Code, Discussion, Rules, Team, and a "...". The main content area has a title "Overview" and sections for "Description", "Evaluation", "Timeline", and "Prizes". The "Description" section contains text about the survey's purpose and results. The "Timeline" section provides details about the survey period. The "Prizes" section mentions a challenge. To the right of the main content, there's a QR code.

<https://www.kaggle.com/competitions/kaggle-survey-2022>

1주차 미션 - 1번

[파이썬 기초] 리스트, 반복문, 조건문 이해하기

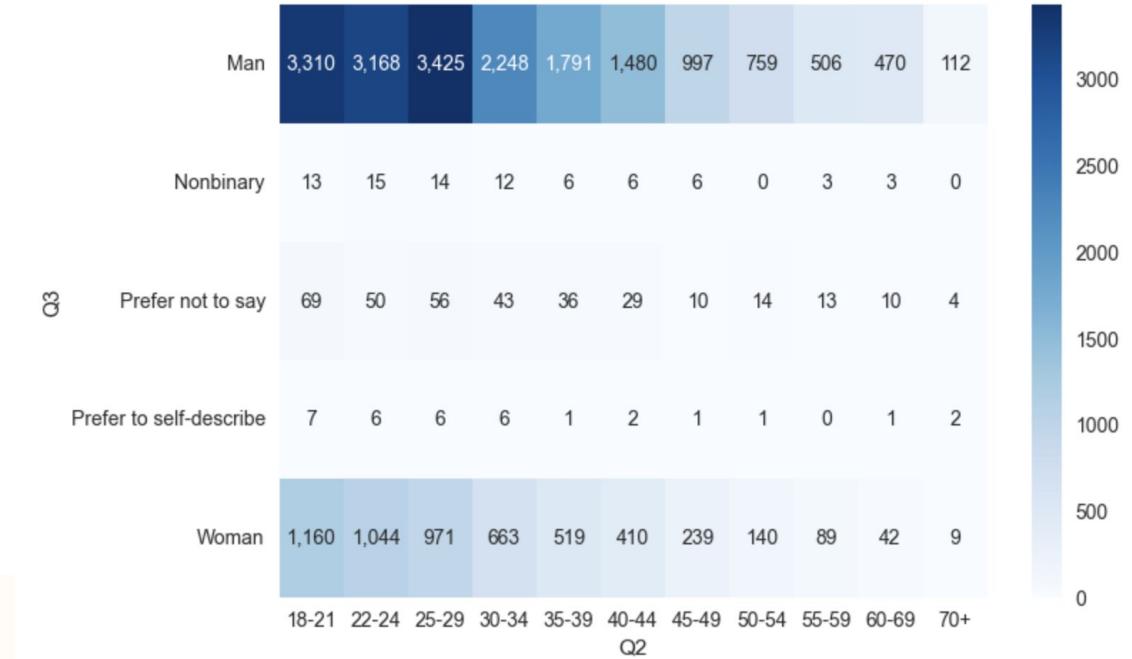
Select the title most similar to your current role (or most recent title if retired): - Selected Choice

Q23	
Data Scientist	1,929
Data Analyst (Business, Marketing, Financial, Quantitative, etc)	1,538
Currently not employed	1,432
Software Engineer	980
Teacher / professor	833
Manager (Program, Project, Operations, Executive-level, etc)	832
Other	754
Research Scientist	593
Machine Learning/ MLops Engineer	571
Engineer (non-software)	465
Data Engineer	352
Statistician	125
Data Architect	95
Data Administrator	70
Developer Advocate	61



```
sns.heatmap(age_gender.T, annot=True, cmap="Blues", fmt=".0f")
```

```
<AxesSubplot:xlabel='Q2', ylabel='Q3'>
```



분석 방법이 궁금하다면 ?

[[1/13] 📈 전세계 데이터 사이언티스트들의 연령대? 관심사? 고용상태? 임금? 📊 이 궁금하다면? - 2020 kaggle survey - YouTube](https://www.youtube.com/watch?v=8_7iou5hjVg&list=PLaTc2c6yEwmq7L8oPO57W91Vx-nP1TvSh)

1주차 미션 - 1번

[파이썬 기초] 리스트, 반복문, 조건문 이해하기

```
question = ['Q2', 'Q3', 'Q4', 'Q5', 'Q6_1', 'Q6_2', 'Q6_3', 'Q6_4', 'Q6_5', 'Q6_6',
            'Q6_7', 'Q6_8', 'Q6_9', 'Q6_10', 'Q6_11', 'Q6_12', 'Q7_1', 'Q7_2',
            'Q7_3', 'Q7_4', 'Q7_5', 'Q7_6', 'Q7_7', 'Q8', 'Q9', 'Q10_1', 'Q10_2',
            'Q10_3']
```

```
single_choice = []
for qst in question:
    if "_" not in qst:
        single_choice.append(qst)
```

```
single_choice
```

```
['Q2', 'Q3', 'Q4', 'Q5', 'Q8', 'Q9']
```

1주차 미션 - 2번

[판다스 집계 연산] 데이터 집계하기

Q2. 한스 로슬링(Hans Rosling, 1948년 7월 27일 ~ 2017년 2월 7일)은 스웨덴의 의사자 통계학자로 비영리 벤처 캡마인더 재단의 공동설립자이기도 합니다. 빅데이터를 가장 잘 활용하는 보건 통계학자로 알려져 있습니다. 베스트셀러 책인 "팩트풀니스" 저자이기도 합니다. 캡마인더 사이트에서는 연도별, 국가별 GDP와 기대수명 데이터를 제공하고 있는데, 대표적으로 파이썬 라이브러리 중 'seaborn'에서 제공되는 예제 데이터가 있습니다. 오늘은 이 데이터를 활용해 문제를 풀어보겠습니다.

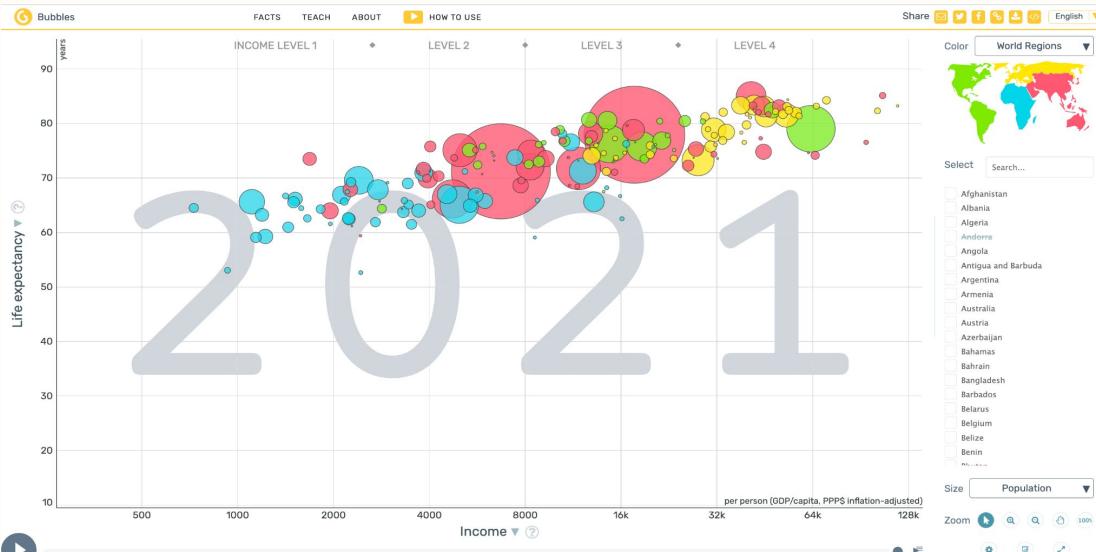
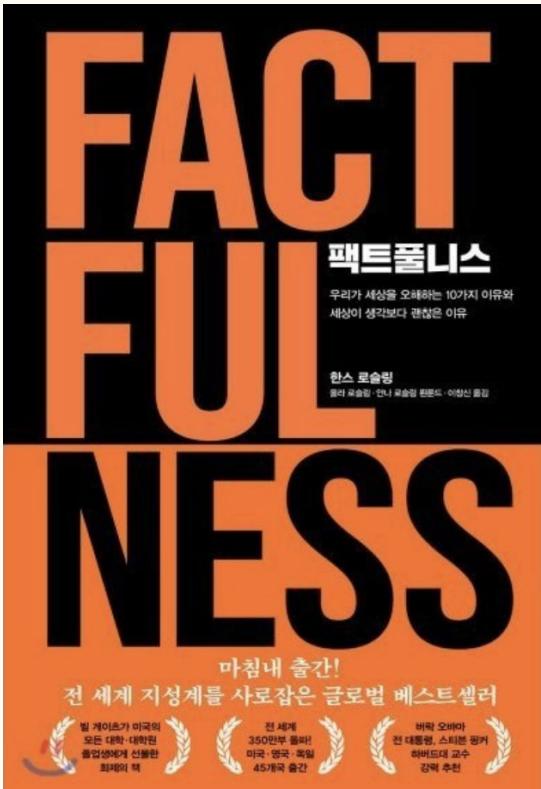
다음의 데이터는 연도, 국가별 기대수명을 나타내고 있는 데이터입니다. 2011년부터의 연도별, 국가별 **평균** 기대수명을 구해주세요. (2011년도 포함되게 구합니다.)

* [필수 조건] groupby 나 pivot_table을 활용합니다. groupby로 구할 때 unstack()이라는 기능을 사용하여 아래와 같이 컬럼에 인덱스 값을 올려서 표기할 수 있습니다.

* 만약 행에는 '연도'가, 열에 '국가'가 들어있고 수치 데이터의 결과값이 아래와 같다면, 출력형태는 조금 달라도 괜찮습니다.

1주차 미션 - 2번

[판다스 집계 연산] 데이터 집계하기



1주차 미션 - 2번

[판다스 집계 연산] 데이터 집계하기

```
import pandas as pd

df = pd.read_csv("https://raw.githubusercontent.com/mwaskom/seaborn-0.10.1/_examples/final.csv")

df
```

	Year	Country	Spending_USD	Life_Expectancy
0	1970	Germany	252.311	70.6
1	1970	France	192.143	72.2
2	1970	Great Britain	123.993	71.9
3	1970	Japan	150.437	72.0
4	1970	USA	326.961	70.9
...
269	2020	Germany	6938.983	81.1
270	2020	France	5468.418	82.3
271	2020	Great Britain	5018.700	80.4
272	2020	Japan	4665.641	84.7
273	2020	USA	11859.179	77.0

274 rows × 4 columns

```
# groupby 로 구하는 방법
df[df["Year"] > 2010].groupby(["Year", "Country"])["Life_Expectancy"].mean().unstack()
```

Country	Canada	France	Germany	Great Britain	Japan	USA
Year						
2011	81.4	82.3	80.5	81.0	82.7	78.7
2012	81.6	82.1	80.6	81.0	83.2	78.8
2013	81.7	82.3	80.6	81.1	83.4	78.8

```
# pivot_table 로 구하는 방법
pd.pivot_table(data=df[df["Year"] > 2010], index="Year", columns="Country", values="Life_Expectancy")
```

Country	Canada	France	Germany	Great Britain	Japan	USA
Year						
2011	81.4	82.3	80.5	81.0	82.7	78.7
2012	81.6	82.1	80.6	81.0	83.2	78.8
2013	81.7	82.3	80.6	81.1	83.4	78.8

1주차 미션 - 3번

[마크다운 문법 익히기] 마크다운으로 배운 내용 정리해 보기

📌 Q3. Jupyter notebook 은 문서와 코드를 함께 작성할 수 있다는 점이 장점입니다.
Jupyter notebook에서 지원하는 Markdown 문법을 사용하여, 이번 주에 배운 내용을 정리해 보세요!

1주차 미션 - 3번

[마크다운 문법 익히기] 마크다운으로 배운 내용 정리해 보기

- Jupyter 는 문서와 코드를 함께 작성할 수 있다는 점이 장점입니다.
- Jupyter 에서 지원하는 Markdown 문법을 사용하여, 배운 내용을 정리해 보세요!

답안 예시

- Pandas를 통한 파일 저장과 불러오기
 - `to_csv("파일명", index=False)` : csv 파일로 저장하기
 - `read_csv("파일명")` : csv 파일 불러오기
 - `shape`을 통한 행과 열의 수 보기
 - `head`, `tail`, `sample` 을 통한 일부 데이터 가져오기
- DataFrame의 `info()`, `describe()` 등을 통한 요약과 기술통계 값 구하기
 - `info()`
 - `describe()`
 - `nunique()`
 - `index`
 - `columns`
 - `values`
- Pandas의 DataFrame과 Series의 이해
 - Series : 1차원 벡터구조
 - DataFrame : 2차원 행렬구조

1주차 미션 - 4번

[파일 경로 확인] 판다스로 파일 불러오기

📌 Q4. 앞으로 우리는 공공데이터포털에서 데이터를 다운로드 받아 모든 과정을 진행할 예정입니다. 본격적인 학습 이전에! 데이터를 다루는 방법이 익숙해지도록 한번 더 연습해보고, 어떤 문제를 풀 수 있을지도 함께 고민해보아요!

공공데이터포털에서 원하는 데이터를 다운로드 받아 경로를 설정하고, 주피터 노트북과 판다스를 통해 불러와 보세요!

어떤 데이터를 사용해야 할지 고민된다면 다음 링크의 데이터를 다운로드 받아도 좋습니다.

[참고 예시] 공공데이터포털 - 서울특별시 강남구_생활폐기물배출량

이 때, 인코딩 오류가 발생한다면 encoding="cp949" 옵션을 사용해 주세요!

cp949는 한글 윈도우에서 사용하는 인코딩 방식이랍니다.

1주차 미션 - 4번

[파일 경로 확인] 판다스로 파일 불러오기

```
pd.read_csv("서울특별시 강남구_생활폐기물배출량_20221019.csv", encoding="cp949")
```

	강남구 2019년(생활폐기물)-톤	2020년(생활폐기물)-톤
0	01월 7550	7350
1	02월 6387	6688
2	03월 7198	6973
3	04월 6820	6448
4	05월 7767	6470
5	06월 7275	7923
6	07월 8164	8015
7	08월 7807	7859
8	09월 6811	7452
9	10월 7222	6875
10	11월 7592	7337
11	12월 7298	6911

2주차 학습 내용

어떤 내용을 배우게 될까요?

2주차 학습 내용 정리

matplotlib 한글 폰트

The screenshot shows the GitHub repository page for `ychoi-kr/koreanize-matplotlib`. The repository is public and was forked from `uehara1414/japanize-matplotlib`. The main navigation bar includes links for Code, Pull requests (1), Actions, Projects, Security, and Insights. The Code tab is selected. On the left, there's a dropdown for the master branch, showing 5 branches and 5 tags. A message indicates that this branch is 9 commits ahead of the upstream master. Below this, a list of recent commits is shown:

Author	Commit Message	Date
ychoi-kr	docs: add pip command	May 13, 2018
	initial commit	6 months ago
	Update .gitignore	4 years ago

On the right, there's an 'About' section with the following details:

- install & import하는 것만으로 matplotlib에서 한국어를 표시할 수 있습니다.
- Readme
- MIT license
- 31 stars
- 0 watching
- 19 forks

2주차 학습 내용 정리

matplotlib 한글 폰트

```
# 한글폰트 사용을 위해 설치  
# 아래 모듈을 설치하고 불러오면 별도의 한글폰트 설정이 필요 없습니다.  
# !pip install koreanize-matplotlib  
  
# import koreanize_matplotlib
```

2주차 학습 내용 정리

Pandas Cheat Sheet

Data Wrangling
with pandas Cheat Sheet
<http://pandas.pydata.org>

Pandas API Reference Pandas User Guide

Creating DataFrames

```
df = pd.DataFrame({
    "a": [4, 5, 6],
    "b": [7, 8, 9],
    "c": [10, 11, 12],
    index=[1, 2, 3],
})
Specify values for each column.

df = pd.DataFrame([
    [4, 7, 10],
    [5, 8, 11],
    [6, 9, 12],
], index=[1, 2, 3],
columns=['a', 'b', 'c'])
Specify values for each row.

df = pd.DataFrame([
    {"n": 1, "v": 4, "x": 7, "y": 10},
    {"n": 2, "v": 5, "x": 8, "y": 11},
    {"n": 3, "v": 6, "x": 9, "y": 12}
])
Create DataFrame with a MultiIndex.
```

Method Chaining

```
Most pandas methods return a DataFrame so that another pandas method can be applied to the result. This improves readability of code.
df = (pd.read_csv('...'))
    .rename(columns={'variable': 'var',
                     'value': 'val'})
    .query('val >= 200')
)
```

Tidy Data – A foundation for wrangling in pandas

In a tidy data set: $\text{F} \leftrightarrow \text{M} \leftrightarrow \text{A}$

Each variable is saved in its own column & Each observation is saved in its own row

Tidy data complements pandas's **vectorized operations**. pandas will automatically preserve observations as you manipulate variables. No other format works as intuitively with pandas. $\text{M} * \text{A} \rightarrow \text{F}$

Reshaping Data – Change layout, sorting, reindexing, renaming

`df.sort_values('mpg')`
Order rows by values of a column (low to high).

`df.sort_values('mpg', ascending=False)`
Order rows by values of a column (high to low).

`df.rename(columns = {'y': 'year'})`
Rename the columns of a DataFrame

`df.sort_index()`
Sort the index of a DataFrame

`df.reset_index()`
Reset index of DataFrame to row numbers, moving index to columns.

`df.drop(columns=['Length', 'Height'])`
Drop columns from DataFrame

Subset Observations - rows

`df[df.Length > 7]`
Extract rows that meet logical criteria.

Subset Variables - columns

`df[['width', 'length', 'species']]`
Select multiple columns with specific names.

Subsets - rows and columns

`df.loc[1, 2]`
Select single row with specific name.

`df.loc[:, 1:2]`
Select columns in positions 1, 2 and 5 (first column is 0).

`df.loc[1:2, 1:2]`
Select all columns between x2 and x4 (inclusive).

`df.loc[[1, 2], [1, 2]]`
Select rows meeting logical condition, and only the specific columns.

Using query

Logic in Python (and pandas)	regex (Regular Expressions) Examples
< Less than	<code>^.</code> Matches strings containing a period.'
> Greater than	<code>Length\$</code> Matches strings ending with word 'Length'
== Equals	<code>^Sepal</code> Matches strings beginning with the word 'Sepal'
<= Less than or equals	<code>^x(1-5)\$</code> Matches strings beginning with 'x' and ending with 1,2,3,4,5
>= Greater than or equals	<code>^(?!Species\$).*</code> Matches strings except the string 'Species'

Cheatsheet for pandas (<http://pandas.pydata.org>) originally written by ivy Lustig. Princeton Consultants. Inspired by RStudio Data Wrangling CheatSheet

Summarize Data

`df['w'].value_counts()`
Count number of rows with each unique value of variable

`len(df)`
of rows in DataFrame.

`df.shape`
Tuple of # of rows, # of columns in DataFrame.

`df['w'].nunique()`
of distinct values in a column.

`df.describe()`
Basic descriptive and statistics for each column (or GroupBy).

Handling Missing Data

`df.dropna()`
Drop rows with any column having NA/null data.

`df.fillna(value)`
Replace all NA/null data with value.

Make New Columns

`df.assign(Area=lambda df: df.Length*df.Height)`
Compute a new column Area.

Group Data

`df.groupby(by='col1')`
Return a GroupBy object, grouped by values in column named "col1".

`df.groupby(level='ind')`
Return a GroupBy object, grouped by values in index level named "ind".

`df.groupby(by='col1').shift(1)`
Copy with values shifted by 1.

All of the summary functions listed above can be applied to a group. Additional GroupBy functions:

- `size()` Size of each group.
- `agg(function)` Aggregate group using function.

Windows

`df.expanding()`
Return an Expanding object allowing summary functions to be applied cumulatively.

`df.rolling(n)`
Return a Rolling object allowing summary functions to be applied to windows of length n.

Plotting

`df.plot.hist()`
Histogram for each column

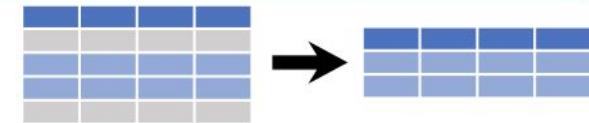
`df.plot.scatter(x='w', y='h')`
Scatter chart using pairs of points

2주차 학습 내용 정리

판다스 기초 - info()

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 91335 entries, 0 to 91334
Data columns (total 39 columns):
 #   Column      Non-Null Count  Dtype  
--- 
 0   상가업소번호      91335 non-null   int64  
 1   상호명          91335 non-null   object  
 2   지점명          1346 non-null   object  
 3   상권업종대분류코드 91335 non-null   object  
 4   상권업종대분류명 91335 non-null   object  
 5   상권업종중분류코드 91335 non-null   object  
 6   상권업종중분류명 91335 non-null   object  
 7   상권업종소분류코드 91335 non-null   object  
 8   상권업종소분류명 91335 non-null   object  
 9   표준산업분류코드  86413 non-null   object
```

Subset Observations - rows



`df[df.Length > 7]`

Extract rows that meet logical criteria.

`df.drop_duplicates()`

Remove duplicate rows (only considers columns).

`df.sample(frac=0.5)`

Randomly select fraction of rows.

`df.sample(n=10)` Randomly select n rows.

`df.nlargest(n, 'value')`

Select and order top n entries.

`df.nsmallest(n, 'value')`

Select and order bottom n entries.

`df.head(n)`

Select first n rows.

`df.tail(n)`

Select last n rows.

2주차 학습 내용 정리

판다스 기초 - index, columns, dtypes

컬럼명 보기

```
# 컬럼명만 출력해 봅니다.
```

```
df.columns
```

```
Index(['상가업소번호', '상호명', '지점명', '상권업종대분류코드', '상권업종대분류명', '상권업종중분류코드',  
       '상권업종중분류명', '상권업종소분류코드', '상권업종소분류명', '표준산업분류코드', '표준산업분류명', '시도코드',  
       '시도명', '시군구코드', '시군구명', '행정동코드', '행정동명', '법정동코드', '법정동명', '지번코드',  
       '대지구분코드', '대지구분명', '지번본번지', '지번부번지', '지번주소', '도로명코드', '도로명', '건물본번지',  
       '건물부번지', '건물관리번호', '건물명', '도로명주소', '구우편번호', '신우편번호', '동정보', '층정보',  
       '호정보', '경도', '위도'],  
      dtype='object')
```

데이터 타입

```
# 데이터 타입만 출력합니다.
```

```
df.dtypes
```

상가업소번호	int64
상호명	object
지점명	object
상권업종대분류코드	object
상권업종대분류명	object
상권업종중분류코드	object
상권업종중분류명	object
상권업종소분류코드	object
상권업종소분류명	object
표준산업분류코드	object
표준산업분류명	object

2주차 학습 내용 정리

기초 기술통계

```
# 2개의 컬럼을 describe로 요약합니다.  
df[["위도", "경도"]].describe()
```

	위도	경도
count	91335.000000	91335.000000
mean	36.624711	127.487524
std	1.041361	0.842877
min	33.219290	124.717632
25%	35.811830	126.914297
50%	37.234652	127.084550
75%	37.507463	128.108919
max	38.499659	130.909912

Summarize Data

`df['w'].value_counts()`

Count number of rows with each unique value of variable

`len(df)`

of rows in DataFrame.

`df.shape`

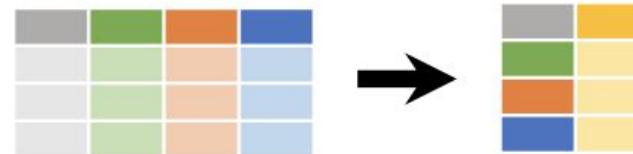
Tuple of # of rows, # of columns in DataFrame.

`df['w'].nunique()`

of distinct values in a column.

`df.describe()`

Basic descriptive and statistics for each column (or GroupBy).



2주차 학습 내용 정리

df.describe(include="object")

```
df.describe(include="object")
```

	상호명	상권업종대분류코드	상권업종대분류명	상권업종중분류코드	상권업종중분류명	상권업종소분류코드	상권업종소분류명	시도명
count	91335	91335	91335	91335	91335	91335	91335	90956
unique	56910		1	1	5	5	34	34
top	리원	S	의료	S01	병원	S02A01	약국	경기도
freq	152	91335	91335	60774	60774	18964	18964	21374

count : 빈도수
unique: 유일값의 수
top : 최빈값
freq : 최빈값의 빈도수

2주차 학습 내용 정리

기초 기술통계

기초 통계 수치

```
# 평균값  
df[ "위도" ].mean()
```

36.62471119236673

```
# 중앙값  
df[ "위도" ].median()
```

37.2346523177033

```
# 최댓값  
df[ "위도" ].max()
```

38.4996585705598

```
# 최솟값  
df[ "위도" ].min()
```

33.2192896688307

```
# 개수  
df[ "위도" ].count()
```

91335

sum()

Sum values of each object.

count()

Count non-NA/null values of each object.

median()

Median value of each object.

quantile([0.25, 0.75])

Quantiles of each object.

apply(function)

Apply function to each object.

min()

Minimum value in each object.

max()

Maximum value in each object.

mean()

Mean value of each object.

var()

Variance of each object.

std()

Standard deviation of each object.

2주차 학습 내용 정리

value_counts(normalize=True)

```
# normalize=True 옵션을 사용하면 비율을 구할 수 있습니다.  
df[ "시도명" ].value_counts(normalize=True)
```

```
경기도      0.234993  
서울특별시  0.208266  
부산광역시 0.071166  
경상남도   0.054675  
인천광역시 0.051915  
대구광역시 0.050541  
경상북도   0.045528  
전라북도   0.042812  
충청남도   0.039338  
전라남도   0.035446  
광주광역시 0.035336  
대전광역시 0.033720  
충청북도   0.029432  
강원도     0.028959  
울산광역시 0.021956  
제주특별자치도 0.012039  
세종특별자치시 0.003881  
Name: 시도명, dtype: float64
```

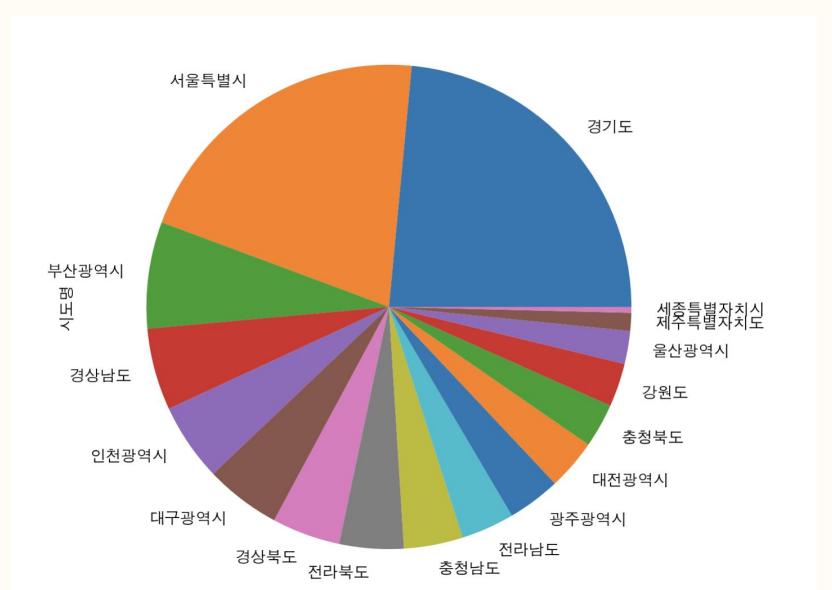
normalize라는 용어는 머신러닝, 딥러닝에서도 많이 사용되는데 보통 정규화라는 단어로 번역이 됩니다.

value_counts에서는 반환값의 전체를 더했을 때 1이 되는 값입니다. 100을 곱하면 100분위 수로 볼 수도 있습니다.

전체 빈도에서 해당 값의 빈도가 어느정도인지를 알 수 있습니다.

2주차 학습 내용 정리

Pie chart 가 seaborn에는 없는 이유?



ghost commented on Nov 16, 2015

Seaborn provides improved defaults for a lot of matplotlib functionality, but doesn't seem to address pie charts. As one of the most common visualisations, this seems like a massive oversight.

I'm aware that a lot of people seem to hate pie charts because they are badly used, but that doesn't mean there shouldn't be attractive defaults for using them correctly ;).

ghost 2

mwaskom commented on Nov 17, 2015

Sorry, no, seaborn will never support pie charts.

Owner

mwaskom closed this as completed on Nov 17, 2015

2주차 학습 내용 정리

Boolean Indexing

Logic in Python (and pandas)		
<	Less than	
>	Greater than	<code>df.column.isin(values)</code>
==	Equals	<code>pd.isnull(obj)</code>
<=	Less than or equals	<code>pd.notnull(obj)</code>
>=	Greater than or equals	<code>&, , ~, ^, df.any(), df.all()</code>
		Not equal to Group membership Is NaN Is not NaN Logical and, or, not, xor, any, all

`df.iloc[10:20]`

Select rows 10-20.

`df.iloc[:, [1, 2, 5]]`

Select columns in positions 1, 2 and 5 (first column is 0).

`df.loc[:, 'x2':'x4']`

Select all columns between x2 and x4 (inclusive).

`df.loc[df['a'] > 10, ['a', 'c']]`

Select rows meeting logical condition, and only the specific columns .

2주차 학습 내용 정리

indexing and slicing

```
>>> a[0, 3:5]
```

```
array([3, 4])
```

```
>>> a[4:, 4:]
```

```
array([[44, 55],  
       [54, 55]])
```

```
>>> a[:, 2]
```

```
a([2, 12, 22, 32, 42, 52])
```

```
>>> a[2::2, ::2]
```

```
array([[20, 22, 24],  
       [40, 42, 44]])
```

0	1	2	3	4	5
10	11	12	13	14	15
20	21	22	23	24	25
30	31	32	33	34	35
40	41	42	43	44	45
50	51	52	53	54	55

2주차 학습 내용 정리

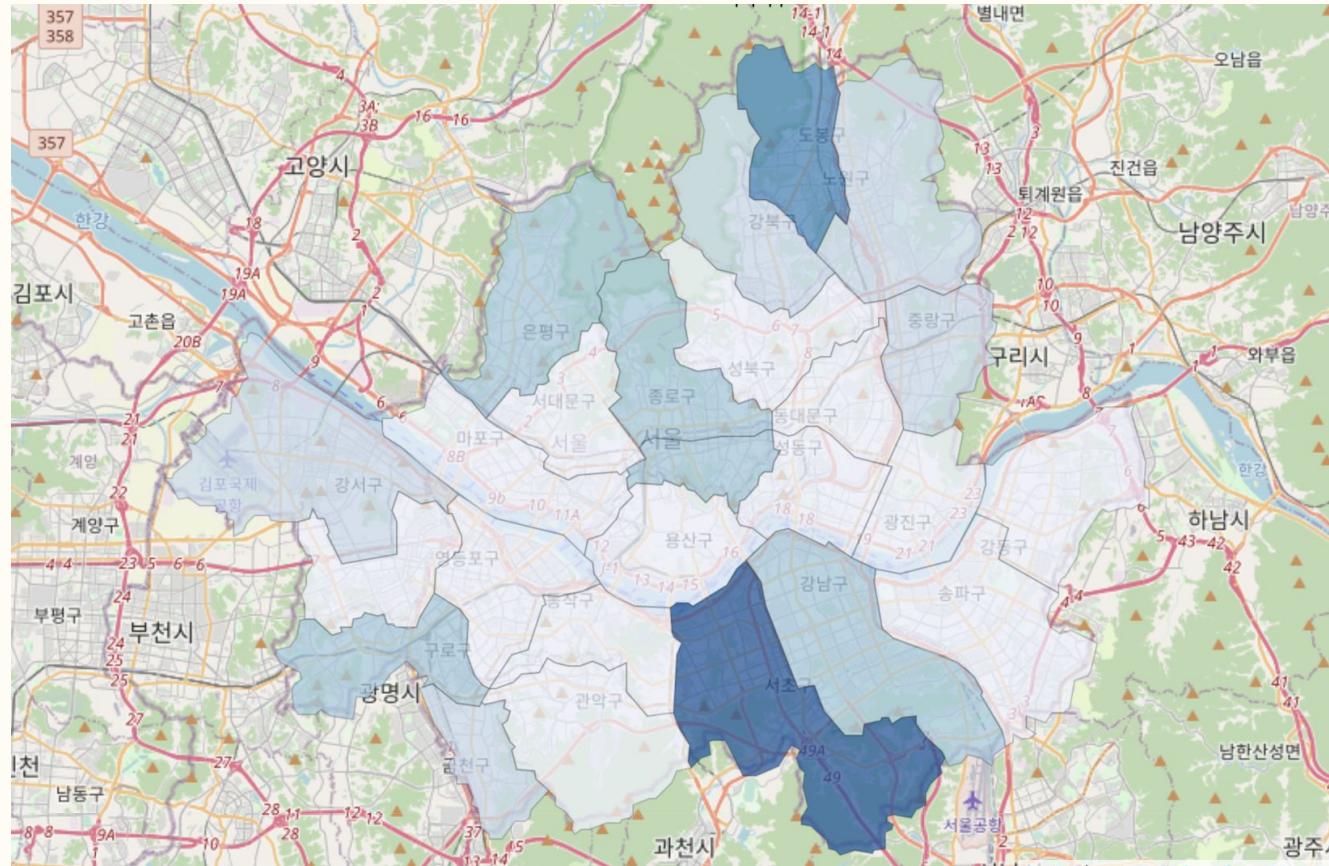
Fancy indexing

```
>>> a[(0,1,2,3,4), (1,2,3,4,5)]  
array([1, 12, 23, 34, 45])  
  
>>> a[3:, [0,2,5]]  
array([[30, 32, 35],  
       [40, 42, 45],  
       [50, 52, 55]])  
  
>>> mask = np.array([1,0,1,0,0,1], dtype=bool)  
>>> a[mask, 2]  
array([2, 22, 52])
```

0	1	2	3	4	5
10	11	12	13	14	15
20	21	22	23	24	25
30	31	32	33	34	35
40	41	42	43	44	45
50	51	52	53	54	55

알아두면 쓸데있는 신비한 프로그래밍

folium Choropleth Map



알아두면 쓸데있는 신비한 프로그래밍

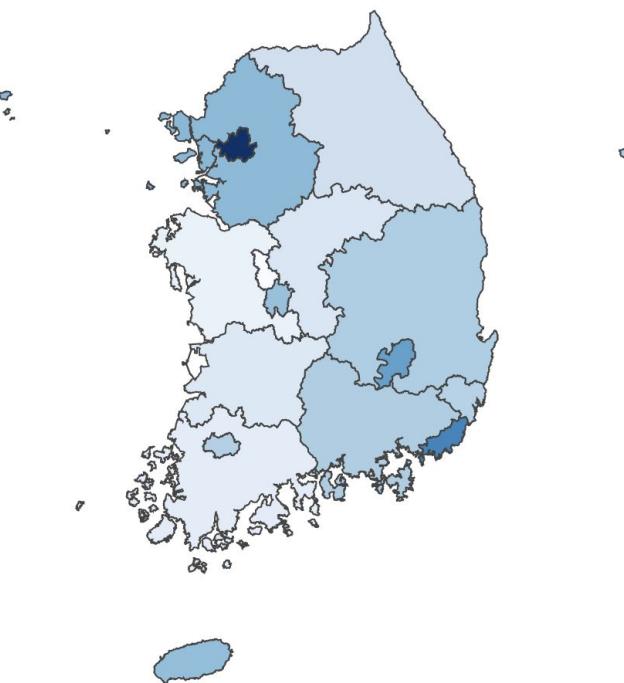
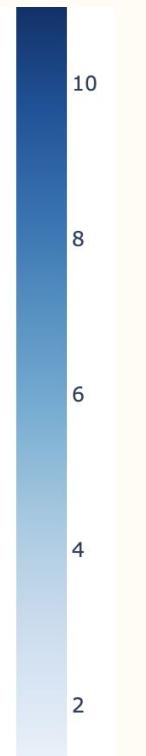
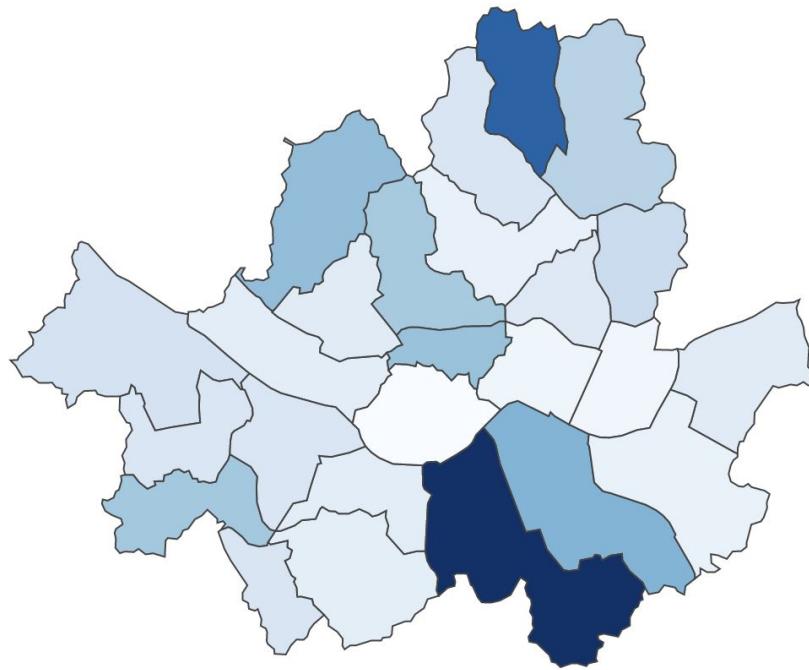
folium Choropleth Map

서울 행정 구역 : [southkorea/seoul-maps: Seoul administrative divisions in ESRI Shapefile, GeoJSON and TopoJSON formats.](<https://github.com/southkorea/seoul-maps>)

전국 행정 구역 : [southkorea/southkorea-maps: South Korea administrative divisions in ESRI Shapefile, GeoJSON and TopoJSON formats.](<https://github.com/southkorea/southkorea-maps>)

알아두면 쓸데있는 신비한 프로그래밍

plotly Choropleth Map



알아두면 쓸데있는 신비한 프로그래밍

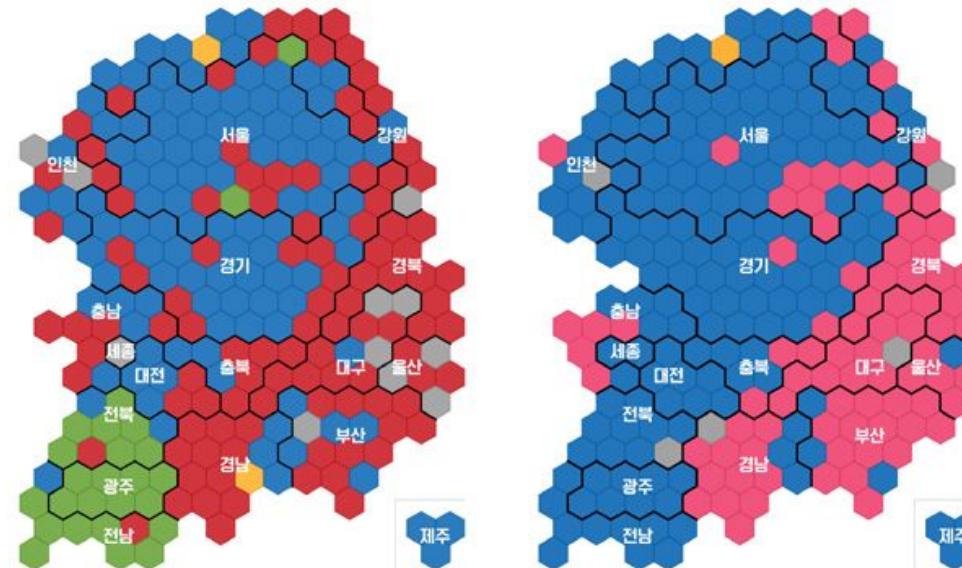
cartogram

제20대 VS 제21대 총선 지역구 카토그램

제20대 총선

VS

제21대 총선



KBS 데이터저널리즘팀 | 자료 중앙선거관리위원회 (2016년 4월 14일, 2020년 4월 16일 기준)

미션 출제 의도

2주차

2주차 미션의 출제 의도와 문제 소개

2주차 미션 출제 의도

1번 [판다스 인덱싱과 연산] 전국 시도별 약국수 구하기

2번 [판다스 인덱싱과 연산] 시도별 동물병원 빈도수 구하기

3번 [판다스 인덱싱과 연산] 피부과나 성형외과가 다른지역에 비해 많은 곳은?

4번 [folium 활용] 지도에서 노인/치매병원 위치 보기

2주차 미션 - 1번

[판다스 인덱싱과 연산] 전국 시도별 약국수 구하기

- 📌 Q1. 상권업종중분류명이 병원인 것을 찾아 빈도수를 구해주세요. 이 때, value_counts, groupby, pivot_table 등 다양한 집계 방법을 통해 구해볼 수 있습니다. 각자 구하기 편한 방법을 통해 빈도수를 구합니다.

2주차 미션 - 2번

[판다스 인덱싱과 연산] 시도별 동물병원 빈도수 구하기

Q2. 여러분은 반려동물과 관련된 사업을 하는 스타트업에 취업을 하여 상권분석을 해달라는 요청을 받았습니다.

병원이나 약국은 인구나 유동인구가 많은 지역에 주로 위치하고 있습니다. 그렇다면 동물병원도 병원이나 약국이 많은 곳에 더 많이 있을까요?

빈도수를 구하고 시각화 하여 동물병원이 어느지역에 많은지 분석해 주세요!

2주차 미션 - 3번

피부과나 성형외과가 다른지역에 비해 많은 곳은?

Q3. 강남지역에는 다른 지역에 비해 피부과나 성형외과가 많아보입니다.

실제로 해당 지역에 피부과나 성형외과가 다른지역에 비해 전체 병원 수 중에서 어느정도의 비율을 차지하고 있는지 구별로 구해주세요.

상권업종소분류명에 "피부" 나 "성형"이 들어가는 서울시에 소재한 병원을 찾아주세요.
그리고 시군구별로 피부, 성형이 들어가는 비율이 어느정도 되는지 구해주세요.

2주차 미션 - 4번

지도에서 노인/치매병원 위치 보기

Q4. 평균 기대수명이 점점 길어지면서 실버 의료 산업도 주목받고 있습니다.
여러분은 실버 의료 산업과 관련된 스타트업에 취업했습니다.

지도를 시각화하여 '노인/치매병원'이 주로 어디에 위치하고 있는지를 찾아보고자 합니다.
folium 을 통해 지도에 전국의 '노인/치매병원'을 표시해 주세요!

Colab AI 기능 사용하기

온라인 편집기 활용하기

Colab 활용하기

Colab 온라인 편집기

+ 코드 + 텍스트

연결 ▾ ♦ Gemini 🔍 ⚙️ ▾

▶ 코딩을 시작하거나 AI로 코드를 생성하세요.

{x} Colab이란?

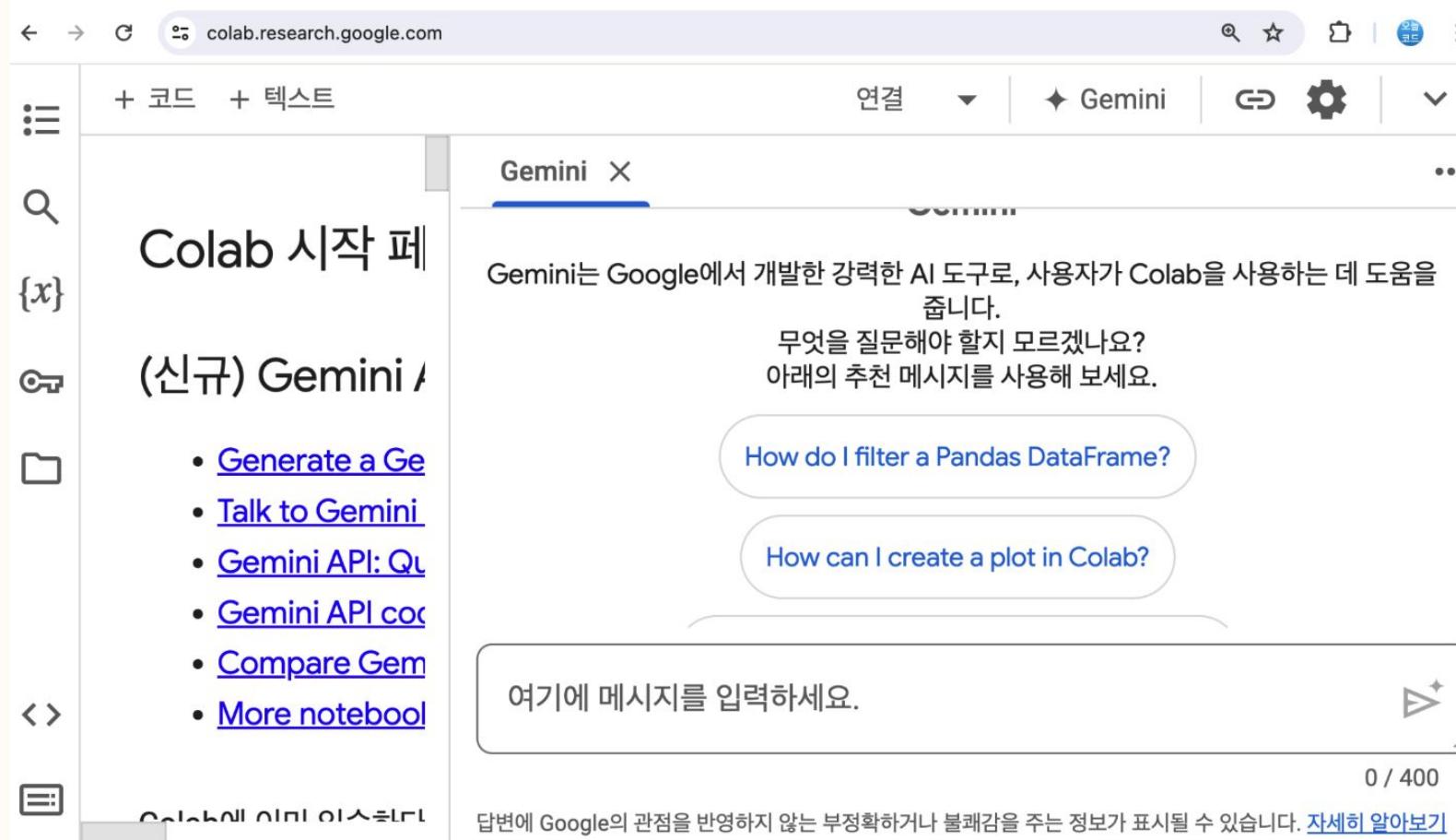
Colaboratory(줄여서 'Colab'이라고 함)을 통해 브라우저 내에서 Python 스크립트를 작성하고 실행할 수 있습니다.

- 구성이 필요하지 않음
- 무료로 GPU 사용
- 간편한 공유

<> 학생이든, 데이터 과학자든, AI 연구원이든 Colab으로 업무를 더욱 간편하게 처리할 수 있습니다. [Colab 소개 영상](#)에서 자세한 내용을 확인하거나 아래에서 시작해 보세요.

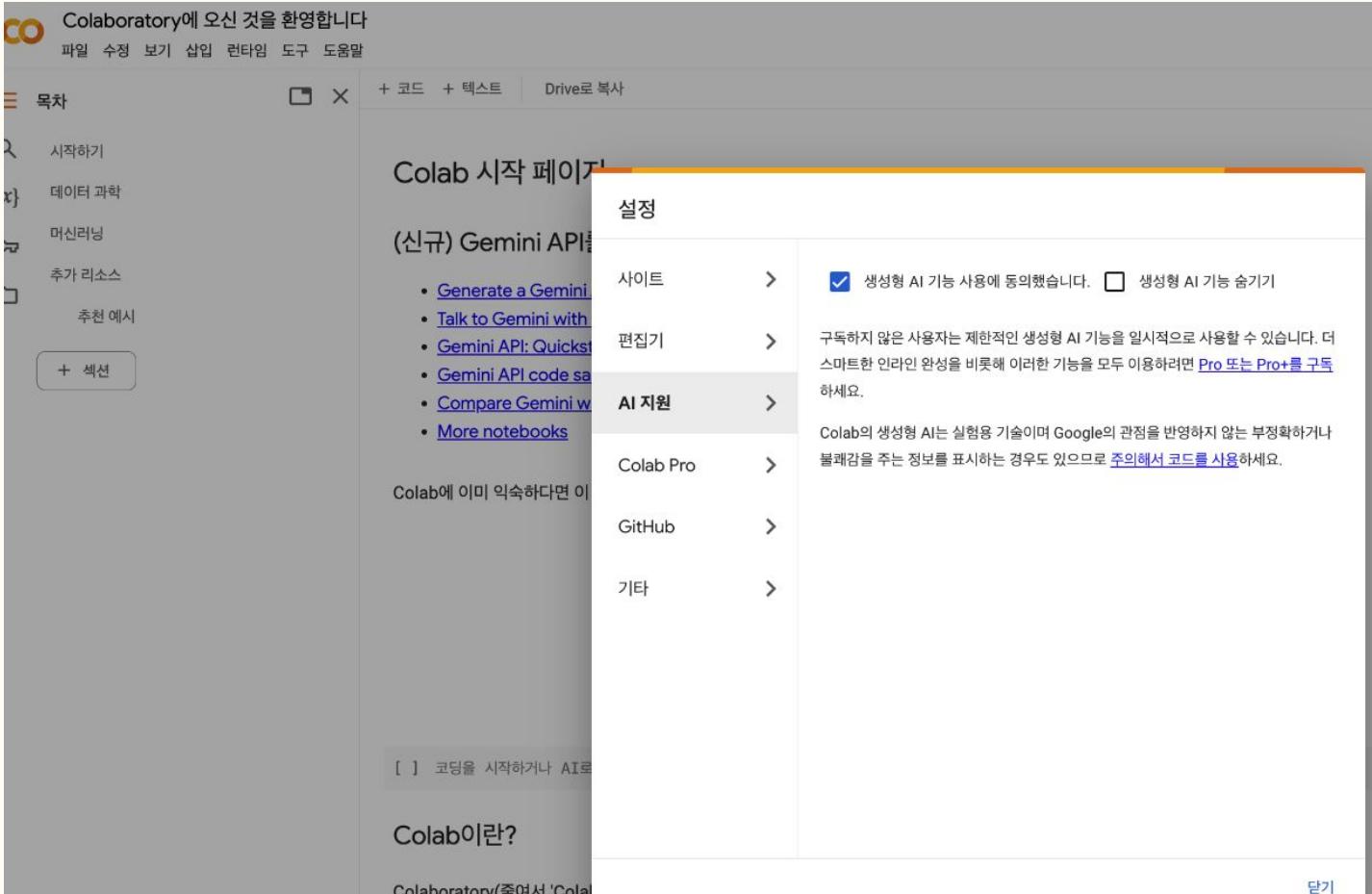
Colab 활용하기

Colab AI 기능 활용하기



Colab 활용하기

Colab AI 기능 활용하기



Colab 활용하기

Colab AI 기능 활용하기

The screenshot shows the Google Colab interface. On the left, there's a sidebar with icons for file management (list, search, {x}, key, folder, diff, and list). The main area has a toolbar at the top with buttons for '+ 코드' (+ Code), '+ 텍스트' (+ Text), and '변경사항을 저장할 수 없음' (Changes not saved). It also includes a '연결' (Connection) dropdown, a 'Gemini' button, and other standard Colab controls like up/down arrows, copy/paste, and settings.

In the center, there's a text input field with a '생성' (Create) button and a placeholder text: '판다스로 데이터프레임을 만들고 간단한 기술 통계 구현'. Below this, there are navigation controls (< 1 of 4 >), like/dislike buttons, and a note: '코드를 사용할 때는 주의가 필요합니다.' (Please be careful when using code).

The code cell contains the following Python code:

```
# prompt: 판다스로 데이터프레임을 만들고 간단한 기술 통계 구현

import pandas as pd

# Create a dataframe
df = pd.DataFrame({'col1': [1, 2, 3, 4], 'col2': [5, 6, 7, 8]})

# Print the dataframe
print(df)

# Print descriptive statistics
print(df.describe())
```

Colab 활용하기

Colab AI 기능 활용하기

장점	단점
<ul style="list-style-type: none">• 무료로 사용할 수 있음• GPU 및 TPU 지원• 간편한 설치 및 설정• 다양한 라이브러리 기본 제공• 협업 기능 지원• 자동 저장 및 버전 관리• 손쉬운 공유 및 배포	<ul style="list-style-type: none">• 작업 세션 시간 제한• 인터넷 연결 필수• 개인 정보 보호 문제• 하드웨어 성능 제한• 프리미엄 기능 제한• 커스터마이징 제약• 데이터 업로드 및 다운로드 속도 제한

Search Labs

?



Search Labs에 오신 것을 환영합니다

의견을 보내주세요



<https://labs.google.com/search?source=hp>

코치에게 물어봐

+실시간 QnA

코치에게 물어봐

슬랙 #코치에게-물어봐 채널에 남겨주신 질문에 대해 답변해드려요

동동이_리더

데이터 사이언스, 데이터 분석가 등으로 취업을 할 때 도메인을 정하는 것이 중요하다는 것으로 알고 있습니다!

혹시 현시점에서 취업하기 좋은 도메인 추천 해주실 수 있을까요?

코치에게 물어봐

슬랙 #코치에게-물어봐 채널에 남겨주신 질문에 대해 답변해드려요

코알라_리더

안녕하세요 코치님들, 데이터 분석가 취준중인데 신입으로 채용하는 곳이 별로 없어서 고민입니다.
저는 사회과학계열 석사학위(양적연구논문), 연구원 경력 1년 6개월, 국비지원 부트캠프 수료 정도 한
상태인데 지방이라 그런지 쉽지 않네요..
몇년 뒤에 경기도권으로 이동을 고려하고 있는데, 현 시점에서 DA가 아닌 데이터 품질관리 업무도 경력에
도움이 될지 코치님들 의견이 궁금합니다..!

코치에게 물어봐

슬랙 #코치에게-물어봐 채널에 남겨주신 질문에 대해 답변해드려요

오이비누

데이터 사이언티스트 취직을 희망하고 있습니다, AI의 변화로 계속해서 공부를 이어가야 하는 것과 별개로
퇴근 후에는 업무로부터 자유롭나요(일과 삶의 구분이 명확한가요)?

해당 직군의 채용이 대기업보단 스타트업에서 이뤄지는 것 같습니다. 협업 많이 하시는지 개인으로 할 때가
많은지 궁금합니다

코치에게 물어봐

슬랙 #코치에게-물어봐 채널에 남겨주신 질문에 대해 답변해드려요

이한희

? 요즘 AI 관심이 높아지고 있는데 그만큼 일자리도 많은가요? 아니면 AI에 뛰어드는 사람만 많아지는 상황인가요? 저도 취업 잘 될 것 같은 AI 융합 전공을 하고 있는 입장으로서 취업은 안 되는데 그냥 유행을 따라가고 있는 것은 아닌지 궁금합니다!

라이브 코칭 3회차

08월 12일 월요일 20시

많은 참여 부탁드려요 🍀

오늘 진행한 2회차 다시보기는 이번 주 목요일 15시에 업로드 됩니다:)
리드부스터는 08월 11일 일요일까지 활동일지 제출해주세요!