

Power BI

Fingerpointlab

CONTENTS

1. PowerBI - Python 세팅

1. Python 설치
2. Python 설정

2. Python 을 이용한 데이터 변환

1. 데이터 가져오기
2. 파이썬 스크립트 실행

3. Python을 이용한 시각화

1. 데이터 다루기

1. 데이터 다루기

1) Python 설치

1

Active Python Releases

For more information visit the Python Developer's Guide.

Python version	Maintenance status	First released	End of support	Release schedule
3.10	bugfix	2021-10-04	2026-10	PEP 619
3.9	security	2020-10-05	2025-10	PEP 596
3.8	security	2019-10-14	2024-10	PEP 569
3.7	security	2018-06-27	2023-06-27	PEP 537
2.7	end-of-life	2010-07-03	2020-01-01	PEP 373

Looking for a specific release?

Python releases by version number:

Release version	Release date		Click for more
Python 3.7.10	Feb. 15, 2021	Download	Release Notes
Python 3.8.7	Dec. 21, 2020	Download	Release Notes
Python 3.9.1	Dec. 7, 2020	Download	Release Notes
Python 3.9.0	Oct. 5, 2020	Download	Release Notes
Python 3.5.10	Sept. 5, 2020	Download	Release Notes
Python 3.7.9	Aug. 17, 2020	Download	Release Notes

[View older releases](#)

과정

설명

1

- 설치되지 않은 경우 python.org 에서 3.9.0버전 파이썬 설치
- 처음 뜨는 설치창 체크 박스 다 체크

1. 데이터 다루기

2) Python 설정

1

옵션

전역

데이터 로드
Power Query 편집기
DirectQuery
R 스크립팅
Python 스크립팅
보안
개인 정보
국가별 설정
업데이트
사용량 현황 데이터
진단
미리 보기 가능
자동 복구
보고서 설정
현재 파일
데이터 로드
국가별 설정
개인 정보
자동 복구

Python 스크립트 옵션

Python용 홈 디렉터리를 선택하려면 검색된 Python 설치물 드롭다운 목록에서 선택하거나, [기타]를 선택하고 원하는 위치를 찾아봅니다.

검색된 Python 홈 디렉터리:

C:\Users\bigda\AppData\Local\Programs\Python\Python36

Power BI Desktop에서 실행할 Python IDE(통합 개발 환경)를 선택하려면 검색된 IDE를 드롭다운 목록에서 선택하거나, [기타]를 선택하고 컴퓨터에서 다른 IDE를 찾아봅니다.

검색된 Python IDE:

.PY 파일에 대한 기본 OS 프로그램

Python IDE에 대한 자세한 정보

임시 스토리지 위치 변경

참고: 경우에 따라 Python 사용자 지정 시각적 개체가 추가 패키지를 자동으로 설치하기도 합니다. 이렇게 되려면 임시 스토리지 폴더 이름이 라틴 문자(영문자)로 되어 있어야 합니다.

확인

취소

2

명령 프롬프트

```
Microsoft Windows [Version 10.0.19043.2006]
(c) Microsoft Corporation. All rights reserved.

C:\Users\bigda>conda activate aisample

(aisample) C:\Users\bigda>
```

과정

설명

1

- 파일 > 옵션 > 전역 > python 스크립팅
- 검색된 Python 홈 디렉터리 에 설치된 파이썬 경로가 표시됨

2

- 만약 아나콘다 경로로 설정하고 싶다면
- Python 홈 디렉터리 경로를 아나콘다로 맞춰주고
- 명령 프롬프트에서 conda activate (가상환경명) 을 입력해주면 됨
- Ex) conda activate base

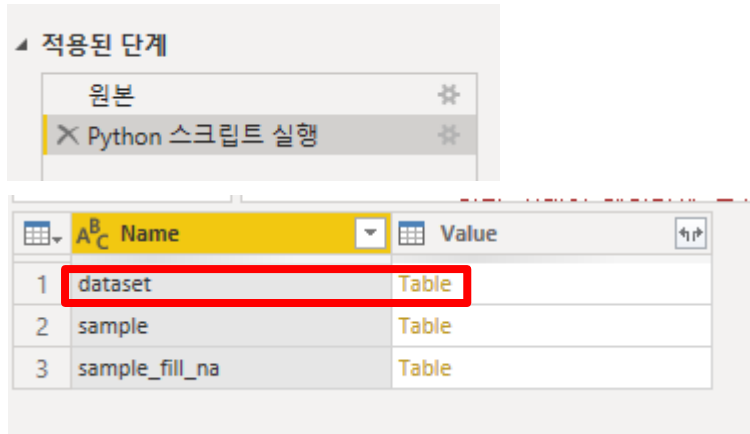
3. Python 을 이용한 시각화

3. Python 을 이용한 시각화

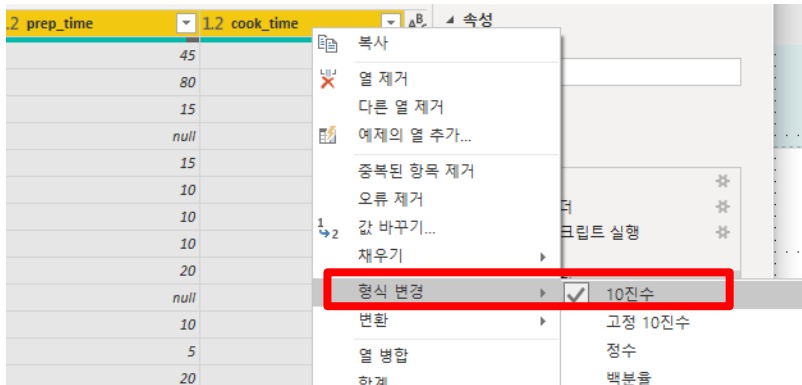
1) 데이터 정리

쿼리에디터 : 데이터 정리 툴

1



2



과정

설명

1

- 홈 > 데이터 변환 > 쿼리 에디터
- 적용된 단계 > Python 스크립트 실행 > dataset에 해당하는 Table클릭

2

데이터 형식 변경 , 텍스트 → 10진수

- Prep_time, cook_time 선택 > 우클릭 > 형식 변경 > 10진수
- 닫기 및 적용

3. Python 을 이용한 시각화

2) 파이썬 시각화 - matplotlib

The screenshot shows the JupyterLab interface. On the left, a code editor contains the following Python code:

```

1 # 데이터의 중복을 없애고 중복을 제거하기 위한 함수를 정의
2 # dataset = pandas.DataFrame({'prog_time': course, 'cook_time': time})
3 # dataset = dataset.drop_duplicates()
4
5 # 데이터를 스토일한 뒤를 출력해 보자

```

On the right, the output of the code is displayed in a table format:

course	time
1	1
1	2
1	3
1	4
1	5
1	6
1	7
1	8
1	9
1	10

과정	설명
1	<ul style="list-style-type: none"> 시각화 창 시각화 > 시각적 개체 빌드 > Py 아이콘 필드 > Indian_food 테이블 펼치고 > prep_time, cook_time, course 열들을 값에 가져다 놓기 값 prep_time, cook_time 요약 안함
2	<p>1 과정 결과</p> <ul style="list-style-type: none"> 하단의 Python 스크립트 편집기에 초록색으로 표시된 주석이 작성된 것을 볼 수 있음

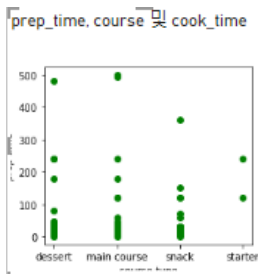
3. Python 을 이용한 시각화

2) 파이썬 시각화 - matplotlib

3

```
Python 스크립트 편집기
△ 중복된 행은 데이터에서 제거됩니다.
1 # 데이터 프레임을 만들고 중복된 행을 제거하기 위한 다음 코드는 항상 실행
2
3 # dataset = pandas.DataFrame(preptime, course, cook_time)
4 # dataset = dataset.drop_duplicates()
5
6 # 여기에 스크립트 코드를 붙여 넣거나 입력:
7
8 import matplotlib.pyplot as plt
9
10 plt.plot(dataset.course, dataset.preptime, 'go')
11 plt.xlabel('course type')
12 plt.ylabel('prep time')
13
14 plt.show()
```

4



과정

설명

3

Python 스크립트 편집기에서 코드 작성
• 기본적인 차트 표현 with matplotlib

```
import matplotlib.pyplot as plt

# 'go' : g는 색상 (green), o는 데이터 모양
plt.plot(dataset.course, dataset.preptime, 'go')
plt.xlabel('course type')
plt.ylabel('prep time')

plt.show()
```

- 위의 코드 입력
- 검은색 막대 가장 오른쪽의 재생 버튼 클릭

4

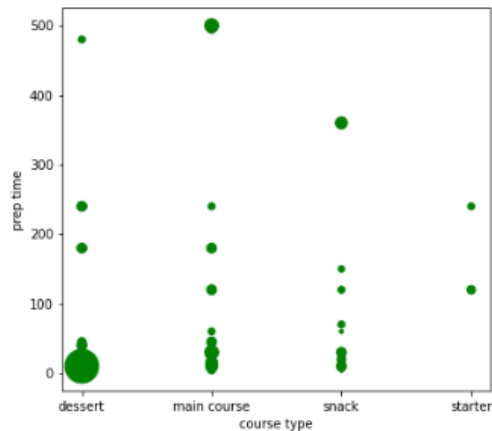
3번 결과

3. Python 을 이용한 시각화

2) 파이썬 시각화 - matplotlib

5

cook_time, course 및 prep_time



과정

설명

5

- Python 스크립트 편집기에서 코드 작성
- 기본적인 차트 표현 with matplotlib
- 마커 사이즈 다른 데이터 값으로 조정

```
import matplotlib.pyplot as plt
```

```
# matplotlib.pyplot.scatter(x, y, s=None, c=None, marker=None, cmap=None, norm=None, vmin=None, vmax=None, alpha=None, linewidths=None, *, edgecolors=None, plotnonfinite=False, data=None, **kwargs)
```

```
plt.scatter(dataset.course, dataset.prep_time, data=dataset.cook_time, 'green', 'o')
```

```
plt.xlabel('course type')
```

```
plt.ylabel('prep time')
```

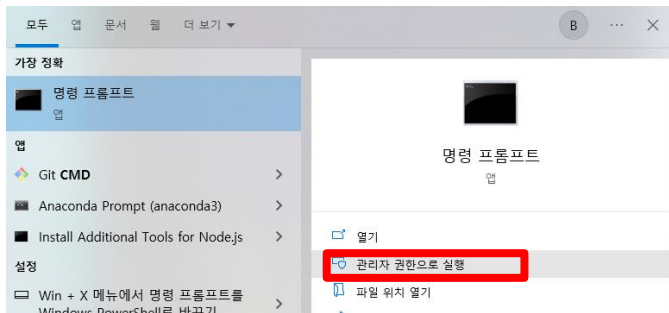
```
plt.show()
```

- 위의 코드 입력
- 검은색 막대 가장 오른쪽의 재생 버튼 클릭

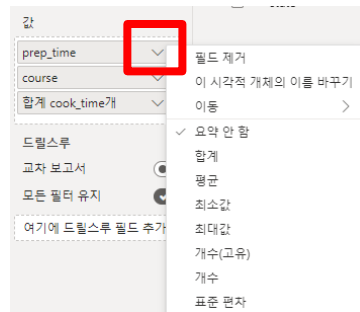
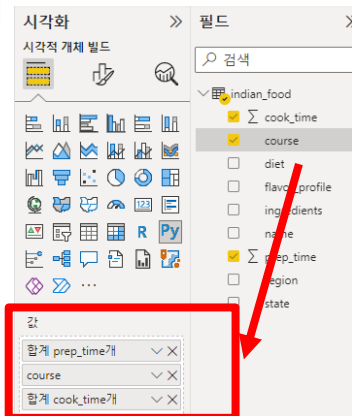
3. Python 을 이용한 시각화

3) 파이썬 시각화 - seaborn

1



2



과정

설명

1

파이썬 시각화 라이브러리 seaborn 설치

- Seaborn 설치 : 명령 프롬프트 > 관리자 권한으로 실행 > pip install seaborn

2

- 시각화 창
- 시각화 > 시각적 개체 빌드 > Py 아이콘
- 필드 > Indian_food 테이블 펼치고 > prep_time, course 열 값에 드래그드랍
- 값 prep_time 요약 안함

3. Python 을 이용한 시각화

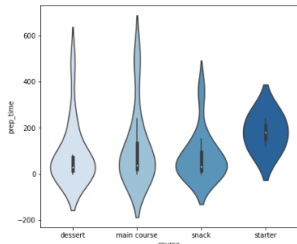
3) 파이썬 시각화 - seaborn

2

```
Python 스크립트 편집기
△ 중복된 행은 데이터에서 제거됩니다.
1 # 데이터 프레임 만들고 중복된 행을 제거하기 위한 다음 코드는 항상 실행되며 스크
2
3 # dataset = pandas.DataFrame(course, prep_time)
4 # dataset = dataset.drop_duplicates()
5 |
6 # 여기에 스크립트 코드를 붙여 넣거나 입력:
7
8 import matplotlib.pyplot as plt
9 import seaborn as sns
10
11 sns.violinplot(x=dataset["course"],y=dataset["prep_time"],palette="Blues")
12
13 plt.show()
14
```

3

course 및 prep_time



과정

설명

2

Python 스크립트 편집기에서 코드 작성

바이올린 플롯:

```
import matplotlib.pyplot as plt
import seaborn as sns
```

```
sns.violinplot(x=dataset["course"],y=dataset["prep_time"],palette="Blues")
```

```
plt.show()
```

- 위의 코드 입력
- 검은색 막대 가장 오른쪽의 재생 버튼 클릭

3

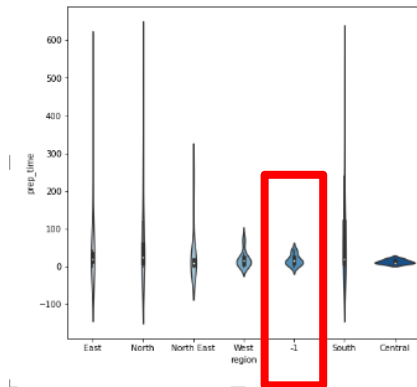
2번 결과

3. Python 을 이용한 시각화

3) 파이썬 시각화 - seaborn

4

course, prep_time 및 region



5

검색

이 시각적 개체의 필터

course
(모두)임

prep_time
(모두)임

region
-1 또는 이(가) 아님

필터 형식 ①
기본 필터링

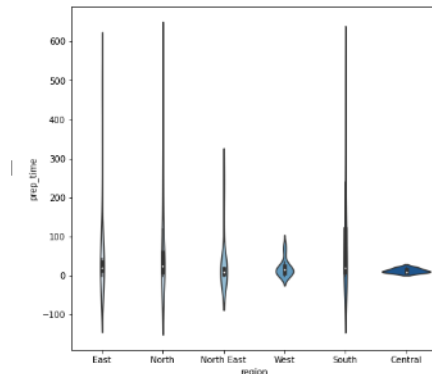
검색

☒ 모두 선택

☐ 1

☐ -1 13

☒ Central 3



과정

설명

4

Python 스크립트 편집기에서 코드 작성

- 3번 그래프 복사 붙여넣기

```
# 바이올린 플롯: region - prep_time  
import matplotlib.pyplot as plt  
import seaborn as sns
```

```
sns.violinplot(x=dataset["region"],y=dataset["prep_  
time"],palette="Blues")
```

```
plt.show()
```

- 위의 코드 입력
- 검은색 막대 가장 오른쪽의 재생 버튼 클릭

5

Region -1을 제외하고 필터링

- 필터 기능 활용
- Region 박스 클릭 > 모두 선택 > 빈값과 -1 체크해제

3. Python 을 이용한 시각화

3) 파이썬 시각화 - seaborn

과정

설명

6

Python 스크립트 편집기에서 코드 작성
• 3번 그래프 복사 붙여넣기

```
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
import numpy as np
sns.set_theme(style="white", rc={"axes.facecolor": (0, 0, 0, 0)})
```

데이터 선언

```
Rs = dataset.prep_time
X = dataset.prep_time
G = dataset.region
Df = pd.DataFrame(dict(x=x, g=g))
```

FacetGrid object 초기화

```
pal = sns.cubehelix_palette(10, rot=-.25, light=.7)
g = sns.FacetGrid(df, row="g", hue="g", aspect=15,
height=.5, palette=pal)
```

Draw the densities in a few steps

```
g.map(sns.kdeplot, "x", bw_adjust=.5, clip_on=False,
fill=True, alpha=1, linewidth=1.5)
g.map(sns.kdeplot, "x", clip_on=False, color="w", lw=2,
bw_adjust=.5)
g.map(plt.axhline, y=0, lw=2, clip_on=False)
```

과정

설명

Python 스크립트 편집기에서 코드 작성
• 왼쪽 코드 밑에 이어서 작성

각 축의 라벨 다는 함수

```
def label(x, color, label):
    ax = plt.gca()
    ax.text(0, .2, label, fontweight="bold", color=color,
ha="left", va="center", transform=ax.transAxes)
```

g.map(label, "x")

overlap 그래프 서브 플랏 설정
g.fig.subplots_adjust(hspace=-.25)

축 세부 정보 지우기

```
g.set_titles("")
g.set(yticks=[])
g.despine(bottom=True, left=True)
```

plt.show()

- 위의 코드 입력
- 검은색 막대 가장 오른쪽의 재생 버튼 클릭

3. Python 을 이용한 시각화

3) 파이썬 시각화 - seaborn

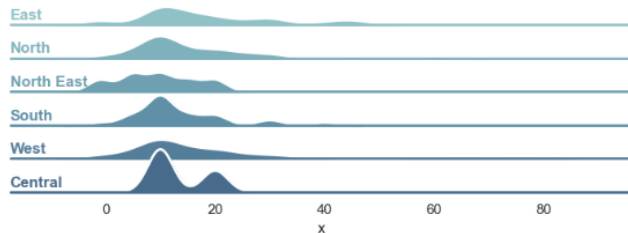
7

prep_time
100보다 작음
필터 형식 ①
고급 필터링
다음 값일 경우 항목 표시
보다 작음
100
● 맞 ○ 또는
필터 적용

region
-1 또는 이(가) 아님
필터 형식 ①
기본 필터링
검색
☒ 모두 선택
☐ -1
☒ Central
☒ East
☒ North
☒ North East
☐ 하나만 선택해야 합니다.

8

cook_time, prep_time, region 및 course



과정

설명

7

필터링 *Region* , *prep_time*

- 필터 박스 > region 클릭 > 모두 선택 > 빈값과 -1 체크 해제
- 필터 박스 > prep_time > 고급 필터링
- 보다 작거나 같음 > 100 입력 > 필터 적용

8

6~7 결과

End of Document