Data Mining

데이터 시각화 (Visualization) 과제

2021년 3월 18일

과제

실제 데이터로 시각화를 해보자.

10 Visualizations Every Data Scientist Should Know

1. **히스토그램** (Histograms)
2. **막대/파이 차트** (Bar/Pie charts)
3. **산점도/직선 그래프** (Scatter/Line plots)
4. **시계열 그래프** (Time series plots)
5. **관계 맵** (Relationship maps)
6. **히트 맵** (Heat maps)
7. **지도** (Geo Maps)
8. **3D 그래프** (3-D Plots)
9. **고차원 그래프** (Higher-Dimensional Plots)
10. **단어 클라우드** (Word clouds)

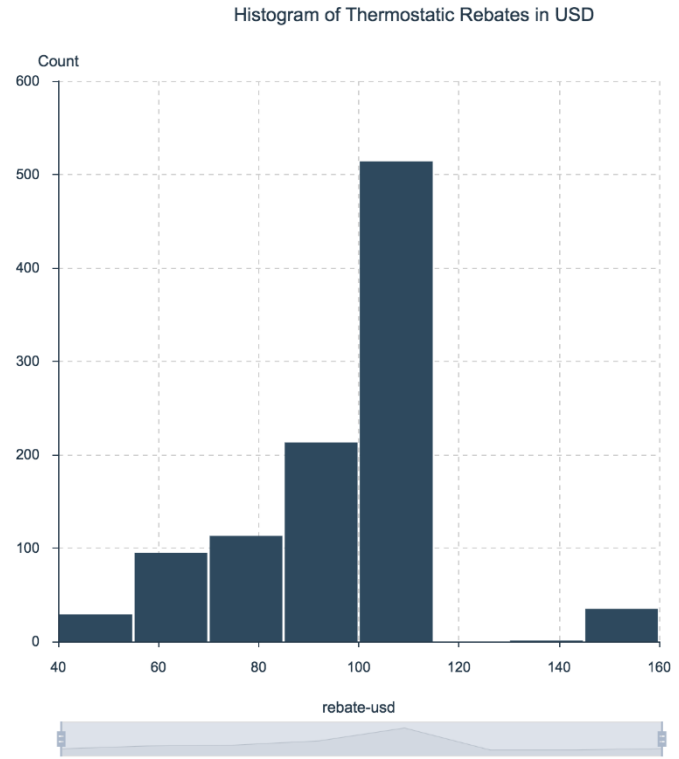
과제 범위

https://www.datasciencecentral.com/profiles/blogs/10-visualizations-every-data-scientist-should-know

1. 히스토그램 (Histograms)

문제 1 히스토그램 (Histograms)

지능형 온도 조절 장치의 할인액



- 수치형 데이터의 분포를 확인하기에 유용

데이터 읽기

```
import matplotlib.pyplot as plt
import pandas as pd

dataset_path = "data/thermostat_rebates_by_zip_1000.csv"
dataset = pd.read_csv(dataset_path)

dataset.tail()
```

	zip-code	rebate-usd	lat	lng	median-household-income	mean-household-income	population
995	40385	100	37.758499	-84.132959	43280	51428	3131
996	72433	100	36.030397	-91.049037	31934	36651	3067
997	90014	67	34.043478	-118.251931	13832	30121	7005
998	8021	90	39.807377	-75.002697	55858	63779	45515
999	68067	100	42.152506	-96.471658	39062	51461	1397

rebate_usd 컬럼으로 히스토그램을 그려본다.

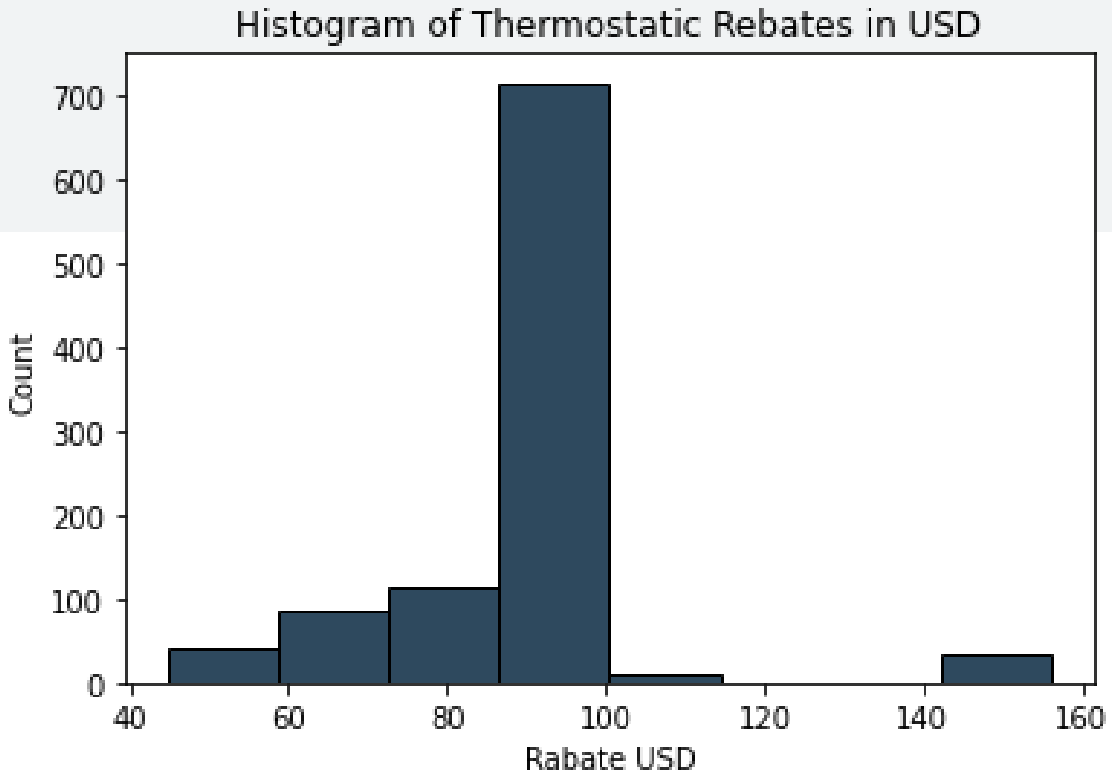
컬럼 데이터 확인

```
rebate = dataset["rebate-usd"]
rebate
```

```
0      88
1      88
2     100
3     100
4     100
     ...
995    100
996    100
997     67
998     90
999    100
Name: rebate-usd, Length: 1000, dtype: int64
```

컬럼 데이터 확인

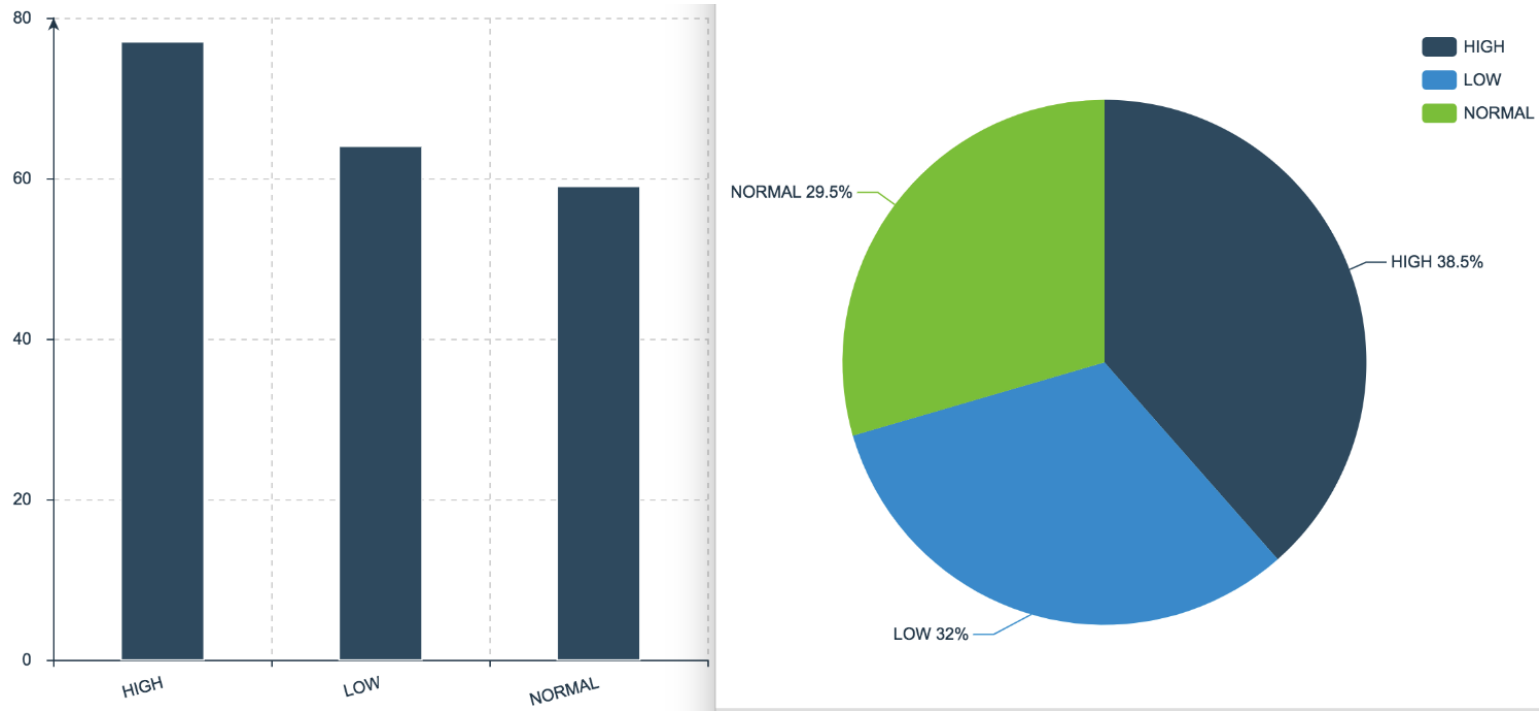
```
n, bins, patches = plt.hist(rebate,
                            8,
                            facecolor="#2E495E",
                            edgecolor=(0, 0, 0))

plt.title("Histogram of Thermostatic Rebates in USD")
plt.xlabel("Rabate USD")
plt.ylabel("Count")

plt.show()
```



2. 막대/파이 차트 (Bar/Pie charts)

문제 2 막대/파이 차트 (Bar/Pie charts)

환자의 혈압을 HIGH, NORMAL, LOW로 구분한 챠트



- 명목형 데이터를 확인할 때 유용
- 단, 카테고리가 많으면 시각화에 방해가 되므로 Top N을 뽑아서 시각화 한다.
- 파이 차트는 파이의 크기가 잘 구분이 되지 않으므로 주의할 것

데이터 읽기

```
dataset_path = "data/drugs_data.csv"
dataset = pd.read_csv(dataset_path)

dataset.tail()
```

	Age	Sex	BP	Cholesterol	NA_to_K	Drug
195	56	F	LOW	HIGH	11.566830	drugC
196	16	M	LOW	HIGH	12.006286	drugC
197	52	M	NORMAL	HIGH	9.894478	drugX
198	23	M	NORMAL	NORMAL	14.019550	drugX
199	40	F	LOW	NORMAL	11.348969	drugX

BP 컬럼으로 막대/파이 차트를 그려본다.

데이터 읽기

```
BP = dataset["BP"].value_counts()
BP
```

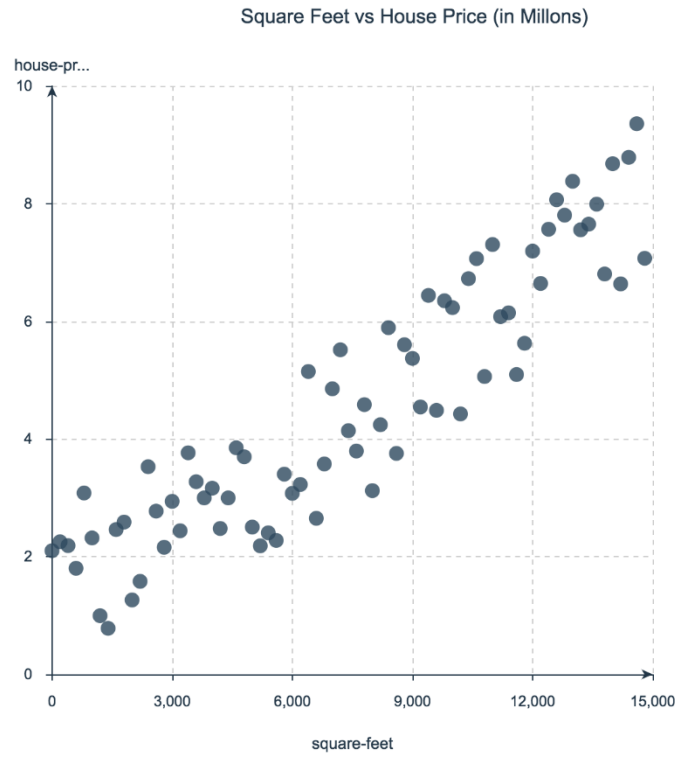
```
HIGH      77
LOW       64
NORMAL    59
Name: BP, dtype: int64
```

힌트 : BP의 키는 BP.keys()로 값은 BP로 가져올 수 있음

3. 산점도/직선 그래프 (Scatter/Line plots)

문제 3 산점도/직선 그래프 (Scatter/Line plots)

집 값과 평방 피트 간의 관계



- 두 변수의 관계를 확인할 때 유용

데이터 읽기

```
dataset_path = "data/square-feet_and_house-price.csv"
dataset = pd.read_csv(dataset_path)

dataset.tail()
```

	square-feet	house-price
70	14000.0	8.678987
71	14200.0	6.636067
72	14400.0	8.787156
73	14600.0	9.358178
74	14800.0	7.071544

4. 시계열 그래프 (Time series plot)

문제 4 시계열 그래프 (Time series plot)

2015년에서 2017년 사이에 테슬라 주식 일일 마감 가

- 시간에 따라 변수가 변화하는 트렌드를 분석하기 위한 용도

데이터 읽기

```
dataset_path = "data/tesla_stock.csv"
dataset = pd.read_csv(dataset_path)

dataset.tail()
```

	Date	Open	High	Low	Close	Volume
749	2015-01-08	212.81	213.7999	210.0100	210.615	3442509.0
750	2015-01-07	213.35	214.7800	209.7800	210.950	2968390.0
751	2015-01-06	210.06	214.2000	204.2100	211.280	6261936.0
752	2015-01-05	214.55	216.5000	207.1626	210.090	5368477.0
753	2015-01-02	222.87	223.2500	213.2600	219.310	4764443.0

날짜로 정렬

```
sorted_dataset = dataset.sort_values(by='Date')
sorted_dataset.tail()
```

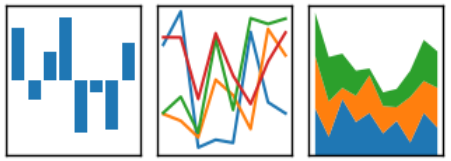
	Date	Open	High	Low	Close	Volume
4	2017-12-22	329.51	330.9214	324.82	325.20	4186131.0
3	2017-12-26	323.83	323.9400	316.58	317.29	4321909.0
2	2017-12-27	316.00	317.6800	310.75	311.64	4645441.0
1	2017-12-28	311.75	315.8200	309.54	315.36	4294689.0
0	2017-12-29	316.18	316.4100	310.00	311.35	3727621.0

5. 판다스 (Pandas)

판다스 (Pandas)

pandas

$y_{it} = \beta' x_{it} + \mu_i + \epsilon_{it}$



데이터 구조를 분석하기 위한 라이브러리

- **테이블 구조의 데이터**를 **인덱스**를 통해 다루는 방식
- 다양한 파일 I/O : **CSV**, **텍스트 파일**, **Excel**, **SQL DB**, **HDF5** 등
- 데이터 객체 연산 시 **인덱스 결합 방식**으로 처리
- 유연한 데이터 구조 변환 및 **통계 요약**
- 대량 데이터의 **슬라이싱**, **인덱싱**, **부분집합** 처리
- **용이한 컬럼 추가** 방식
- **Group by** 엔진으로 데이터의 요약 및 변환
- 고성능으로 데이터셋을 **결합 (Merge and Joining)**
- **계층적 인덱싱 (Hierarchical axis indexing)**
- **시계열** 기능 : 날짜 범위 생성 및 빈도 변환, 이동 통계량, 날짜 이동 및 지연, 다른 시간 간격을 갖는 시계열 결합
- 주요 코드 부분이 Cython 또는 C로 작성되어 **성능이 최적화** 됨

테이블 형태의 자료 구조

데이터 프레임(DataFrame)**은 테이블 형태의 데이터를 표현하며 '시리즈**(Series)**의 시리즈 ' 구조이다.**

열 시리즈

	MPG	Cylinders	Displacement	Horsepower	Weight	Acceleration	Model Year	Origin
393	27.0	4	140.0	86.0	2790.0	15.6	82	1
394	44.0	4	97.0	52.0	2130.0	24.6	82	2
395	32.0	4	135.0	84.0	2295.0	11.6	82	1
396	28.0	4	120.0	79.0	2625.0	18.6	82	1
397	31.0	4	119.0	82.0	2720.0	19.4	82	1

열 인덱스

행 시리즈 →

행 인덱스

dataframe = {'MPG' : {...}, 'Cylinders' : {....}, ..., 'Weight' " {...}, ...}

- **시리즈** (Series) : (인덱스, 값) 구조로 되어 있는 데이터 구조 (딕셔너리와 유사)
- **데이터 프레임** (DataFrame) : 열의 시리즈로 구성된 데이터 구조, 열마다 데이터 타입이 다를 수 있음

컬럼 이름/타입 확인

```
dataset.columns
```

```
Index(['MPG', 'Cylinders', 'Displacement', 'Horsepower', 'Weight',
       'Acceleration', 'Model Year', 'Origin'],
      dtype='object')
```
컬럼 이름의 데이터 타입

```
dataset.dtypes
```

```
MPG             float64
Cylinders         int64
Displacement    float64
Horsepower      float64
Weight          float64
Acceleration    float64
Model Year        int64
Origin            int64
dtype: object
```

컬럼 이름으로 조회

컬럼 이름으로 조회

```
dataset['MPG']
```

```
0      18.0
1      15.0
2      18.0
3      16.0
4      17.0
      ...
393    27.0
394    44.0
395    32.0
396    28.0
397    31.0

Name: MPG, Length: 398, dtype: float64
```

여러 컬럼 조회

칼럼 이름을 리스트로 명시

```
dataset[['MPG', 'Weight']]
```

	MPG	Weight
1	18.0	3504.0
2	15.0	3693.0
3	18.0	3436.0
4	16.0	3433.0
...	...	...
393	27.0	2790.0
394	44.0	2130.0
395	32.0	2295.0
396	28.0	2625.0
397	31.0	2720.0

398 rows × 2 columns

슬라이싱

```
dataset[1:3]
```

	MPG	Cylinders	Displacement	Horsepower	Weight	Acceleration	Model Year	Origin
1	15.0	8	350.0	165.0	3693.0	11.5	70	1
2	18.0	8	318.0	150.0	3436.0	11.0	70	1

컬럼 조건 검색

여러 조건으로 행 선택

```
dataset[(dataset['Origin']==2) & (dataset['Horsepower']>70)].head(4)
```

	MPG	Cylinders	Displacement	Horsepower	Weight	Acceleration	Model Year	Origin
20	25.0	4	110.0	87.0	2672.0	17.5	70	2
21	24.0	4	107.0	90.0	2430.0	14.5	70	2
22	25.0	4	104.0	95.0	2375.0	17.5	70	2
23	26.0	4	121.0	113.0	2234.0	12.5	70	2

- 차량 제조국이 Europe이고 마력이 70이상인 차량 선택
- head()는 default로 처음 5개 항목을 반환, tail()은 default로 마지막 5개 항목을 반환

Thank you!