

```
from google.colab import drive
drive.mount('/content/drive')
```

Drive already mounted at /content/drive; to attempt to forcibly remount, call drive.mount("/content/drive", force_remount=True)

▼ (5-1) Polar Coordinate

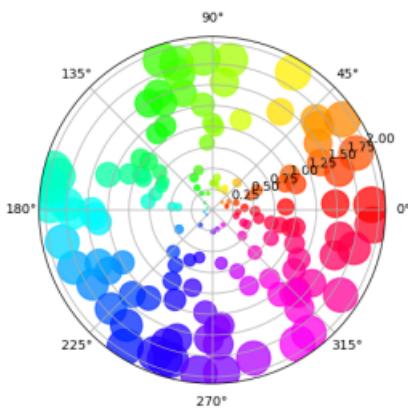
▼ Polar Plot

참고 자료 : https://matplotlib.org/stable/api/projections_api.html

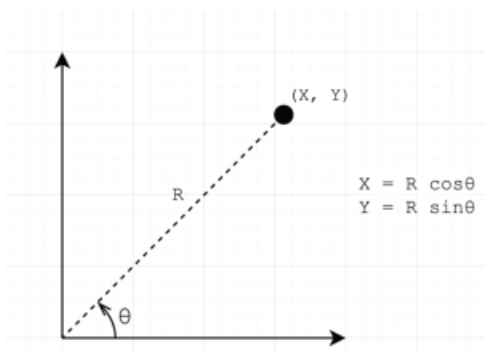
- 극 좌표계(Polar Coordinate)를 사용하는 시각화
 - 거리(R), 각(Theta)을 사용하여 plot
- 회전, 주기성 등 표현 적합
 - projection = polar 을 추가하여 사용

아래 그래프는 Scatter

- Line, Bar도 가능

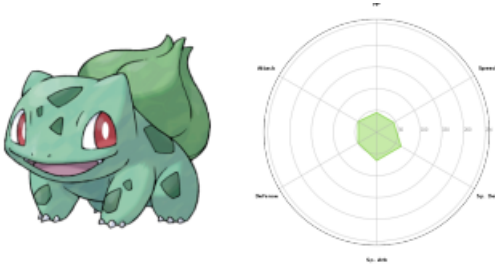


- 직교 좌표계 X, Y에서 변환 가능



▼ Radar Plot

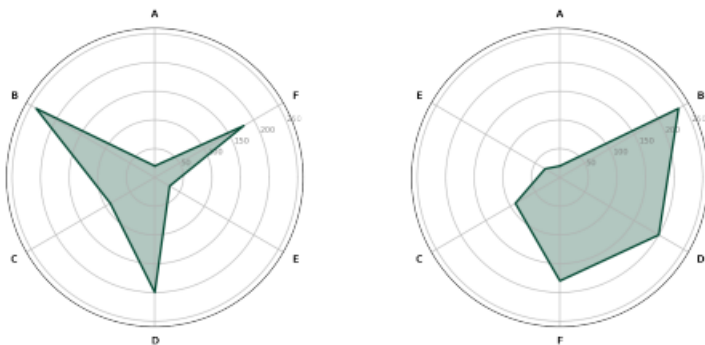
- 극좌표계를 사용하는 대표적인 차트
- Star Plot으로 불림
- 중심점을 기준으로 N개의 변수 값을 표현 가능
- 데이터의 Quality를 비교하고 표현하기에 좋음
 - 캐릭터의 강함
 - 운동 선수 분석



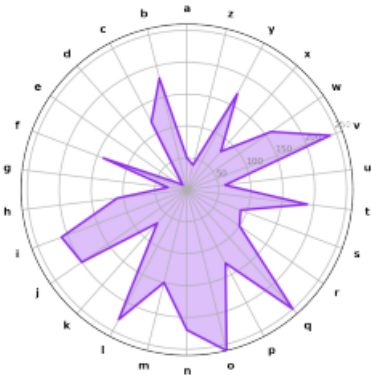
이미지 출처 : [https://bulbapedia.bulbagarden.net/wiki/Bulbasaur_\(Pok%C3%A9mon\)](https://bulbapedia.bulbagarden.net/wiki/Bulbasaur_(Pok%C3%A9mon))

▼ Rader Chart 주의사항

- 각 feature는 독립적이며, 척도가 같아야 함
 - 순서형 변수와 수치형 변수가 함께 사용하지 말자.
 - 차량 정보 예시
 - 안전성 평가, 소비자 만족도 등 5점 만점
 - 최대 속도, 연비 등 절대적 수치
 - 순서형 변수와 수치형 변수를 하나의 Radar Plot에 표현 자제
- 다각형의 면적이 중요해보인다. 하지만 feature의 순서에 따라 면적이 많이 달라진다.
- 아래 그래프는 같은 데이터를 표현한 것이다. 오른쪽은 데이터를 정렬했다.



- Feature가 많으면 가독성이 떨어진다. => Bar, Line, Scatter 사용하자.



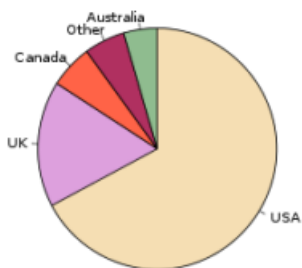
▶ 5-1. Polar Coordinate - 코드 실습

[] ↳ 숨겨진 셀 41개

▼ (5-2) Pie Charts

▼ Pie Charts

- 원을 부채꼴로 분할하여 표현하는 통계 차트
 - 전체를 백분위로 나타낼 때 사용
- 가장 많이 사용하는 차트지만 사용은 줄이는 것이 좋음
 - 비교 어려움
 - 유용성 떨어짐
 - bar plot이 더 유용 (각도보다 길이가 차이를 더 인지하기 쉬움)
 - pie 와 bar chart는 함께 사용할 것을 권장
- waffle chart가 더 비교에 유용



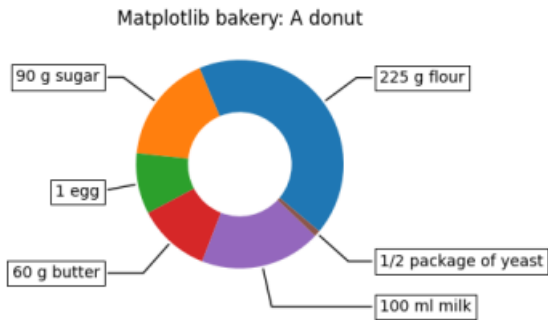
Pie chart of populations of English native speakers | wikipedia

▼ Pie Charts의 응용

▼ Donut Chart

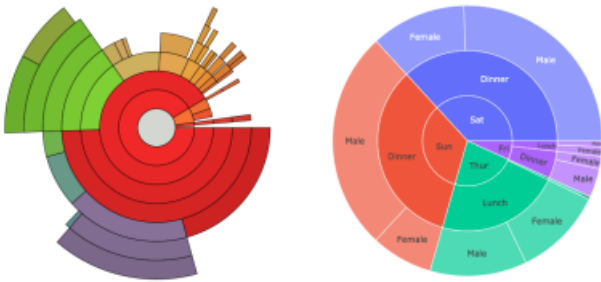
- 중간이 비어있는 도넛 모양 Pie Chart
 - EDA에서 사용하지 말자.
 - 디자인적으로 선호되서 포그래픽에서 종종 사용

- Plotly에서 쉽게 사용 가능



▼ Sunburst Chart

- 햇살(sunburst) 모양 차트
 - 계층적 데이터를 시각화하는 데 사용
 - 구현 난이도에 비해 화려하나 가독성과 유용성 떨어짐
 - Sunburst보다 Treemap을 추천
 - Plotly로 쉽게 사용 가능



▶ 5-2. Pie Chart - 코드 실습

[] ↳ 숨겨진 셀 27개

▼ (5-3) 다양한 시각화 라이브러리

[Multiple Pie Charts: Unreadable, Inefficient, and Over-Used](#)

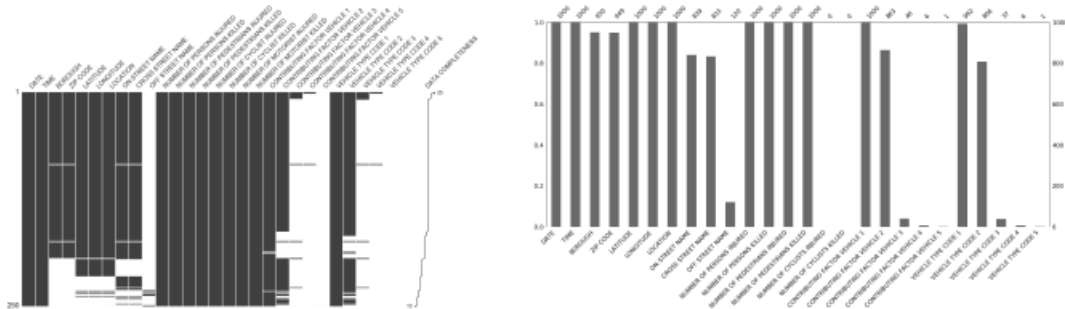
▼ Missingno

- 결측치 시각화 라이브러리
- 빠르게 결측치 분포 확인 가능
- 결측치를 정렬 가능

```
pip install missingno
```

```
Requirement already satisfied: missingno in /usr/local/lib/python3.7/dist-packages (0.5.0)
Requirement already satisfied: seaborn in /usr/local/lib/python3.7/dist-packages (from missingno) (0.11.1)
Requirement already satisfied: matplotlib in /usr/local/lib/python3.7/dist-packages (from missingno) (3.2.2)
Requirement already satisfied: numpy in /usr/local/lib/python3.7/dist-packages (from missingno) (1.19.5)
Requirement already satisfied: scipy in /usr/local/lib/python3.7/dist-packages (from missingno) (1.4.1)
```

Requirement already satisfied: python-dateutil>=2.1 in /usr/local/lib/python3.7/dist-packages (from matplotlib->missingno)
 Requirement already satisfied: cyclar>=0.10 in /usr/local/lib/python3.7/dist-packages (from matplotlib->missingno) (0.10.0)
 Requirement already satisfied: kiwisolver>=1.0.1 in /usr/local/lib/python3.7/dist-packages (from matplotlib->missingno)
 Requirement already satisfied: pyparsing!=2.0.4,!=2.1.2,!=2.1.6,>=2.0.1 in /usr/local/lib/python3.7/dist-packages (from matplotlib->missingno)
 Requirement already satisfied: six in /usr/local/lib/python3.7/dist-packages (from cyclar>=0.10->matplotlib->missingno)
 Requirement already satisfied: pandas>=0.23 in /usr/local/lib/python3.7/dist-packages (from seaborn->missingno) (1.1.5)
 Requirement already satisfied: pytz>=2017.2 in /usr/local/lib/python3.7/dist-packages (from pandas>=0.23->seaborn->missingno)



▼ Treemap

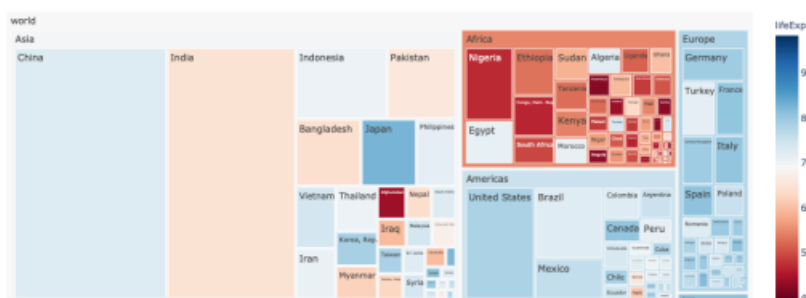
- 계층적 데이터를 작사각형을 사용하여 포함 관계를 표현한 시각화 방법
- 사각형을 분할하는 타일링 알고리즘에 따라 형태가 다양해짐
- 큰 사각형을 분할하여 전체를 나타내는 모자이크 플롯(Mosaic plot)과도 유사
- 라이브러리 : squarify / plotly

`pip install squarify`

Collecting squarify
 Downloading squarify-0.4.3-py3-none-any.whl (4.3 kB)
 Installing collected packages: squarify
 Successfully installed squarify-0.4.3

Plotly 예시

- treemap은 최하단 계층 데이터면 텍스트가 안 보일수도 있기에 interactive 라이브러리에서 사용 권장



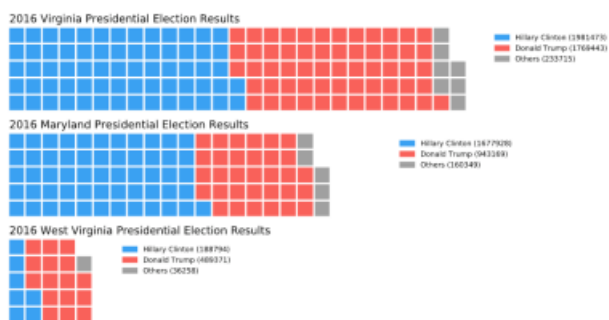
▼ Waffle Chart

- 와플 형태로 discrete하게 값을 나타내는 차트
 - 기본 형태 : 정사각형
 - 정사각형 대신 원하는 벡터 이미지로도 사용 가능

- 비율 확인에 용이 -> 파이 차트 대신에 사용하자
- Icon을 사용한 Waffle Chart도 가능(Pictogram Chart)
- 인포그래픽에서 유용

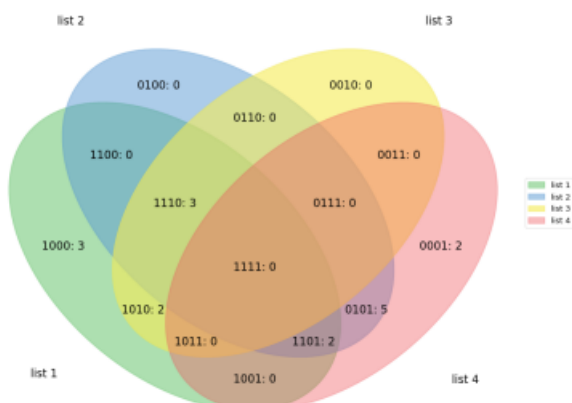
matplotlib 기반으로 만들어짐
 pip install pywaffle

```
Collecting pywaffle
  Downloading pywaffle-0.6.3-py2.py3-none-any.whl (526 kB)
    |████████████████████████████████████████| 526 kB 36.1 MB/s
Requirement already satisfied: matplotlib in /usr/local/lib/python3.7/dist-packages (from pywaffle) (3.2.2)
Requirement already satisfied: numpy>=1.11 in /usr/local/lib/python3.7/dist-packages (from matplotlib->pywaffle) (1.19.
Requirement already satisfied: pyparsing!=2.0.4,!=2.1.2,!=2.1.6,>=2.0.1 in /usr/local/lib/python3.7/dist-packages (from
Requirement already satisfied: kiwisolver>=1.0.1 in /usr/local/lib/python3.7/dist-packages (from matplotlib->pywaffle)
Requirement already satisfied: cycler>=0.10 in /usr/local/lib/python3.7/dist-packages (from matplotlib->pywaffle) (0.10
Requirement already satisfied: python-dateutil>=2.1 in /usr/local/lib/python3.7/dist-packages (from matplotlib->pywaff
Requirement already satisfied: six in /usr/local/lib/python3.7/dist-packages (from cycler>=0.10->matplotlib->pywaffle)
Installing collected packages: pywaffle
Successfully installed pywaffle-0.6.3
```



▼ Venn

- 집합 등에서 사용하는 벤 다이어그램
 - EDA 보다는 출판 및 프레젠테이션에 사용
 - 디테일한 사용이 draw.io나 ppt에 비해 어려움



```
pip install pyvenn
```

```
ERROR: Could not find a version that satisfies the requirement pyvenn (from versions: none)
ERROR: No matching distribution found for pyvenn
```

```
pip install matplotlib-venn
```

```
Requirement already satisfied: matplotlib-venn in /usr/local/lib/python3.7/dist-packages (0.11.6)
Requirement already satisfied: scipy in /usr/local/lib/python3.7/dist-packages (from matplotlib-venn) (1.4.1)
Requirement already satisfied: matplotlib in /usr/local/lib/python3.7/dist-packages (from matplotlib-venn) (3.2.2)
Requirement already satisfied: numpy in /usr/local/lib/python3.7/dist-packages (from matplotlib-venn) (1.19.5)
Requirement already satisfied: cycycler>=0.10 in /usr/local/lib/python3.7/dist-packages (from matplotlib->matplotlib-venn)
Requirement already satisfied: kiwisolver>=1.0.1 in /usr/local/lib/python3.7/dist-packages (from matplotlib->matplotlib)
Requirement already satisfied: pyparsing!=2.0.4,!=2.1.2,!=2.1.6,>=2.0.1 in /usr/local/lib/python3.7/dist-packages (from matplotlib->matplotlib)
Requirement already satisfied: python-dateutil>=2.1 in /usr/local/lib/python3.7/dist-packages (from matplotlib->matplotlib)
Requirement already satisfied: six in /usr/local/lib/python3.7/dist-packages (from cycycler>=0.10->matplotlib->matplotlib)
```

▶ 5-3. 다양한 시각화 라이브러리 - 코드 실습

[] ↪ 숨겨진 셀 54개

▼ (6-1) Interactive Visualization

▼ Interactive를 사용하는 이유

정적 시각화의 단점

- 설득을 위한 메시지를 압축하는 것이 정적 시각화의 장점
- 정형 데이터에서 Feature가 10개 있다면 => 공간적 낭비가 큼
 - 각각을 살펴보는 데 10개의 plot
 - 각각의 관계를 살펴보는 데 $10 * 9 / 2 = 45$ 개의 plot
 - 가볍게 살펴보는 것도 약 50개의 plot이 필요
- 각각의 사용자는 원하는 인사이트가 다름
 - 인터랙션으로 원하는 정보를 얻을 수도 있다.

▼ 인터랙티브의 종류

- Select : mark something as interesting
- Explore : show me something else
- Reconfigure : show me a different arrangement
- Encode : show me a different representation
- Abstract : show me more or less detail
- Filter : show me something conditionally
- Connect : show me related items

출처 : Toward a Deeper Understanding of the Role of Interaction in Information Visualization

<https://www.cc.gatech.edu/~stasko/papers/infovis07-interaction.pdf>

대표 라이브러리

- Plotly

- Bokeh
- Altair

▼ Interactive Viz Library

Matplotlib

주피터 노트북 환경 또는 Local에서만 인터랙티브 실행

- 웹에 deploy 불가
- mpld3 라이브러리를 사용하면 웹에서 D3-based Viewer로 화면을 움직이는 기능만 제공

Plotly

- Python, R, JS에서도 제공
- 예시 많고 문서화 잘 됨
- 통계 시각화 + 지리 시각화 + 3D 시각화 + 금융 시각화 등 다양한 시각화 기능 제공
- D3.js(JS 시각화 라이브러리)를 기반으로 만들어져 웹에서 사용 가능
- 기본 색상 : 형광 Color

Plotly Express

- Plotly를 seaborn과 유사하게 만들어 쉬운 문법
- 커스텀 부분이 Plotly보다 부족하지만 다양한 함수 제공

Bokeh

- Matplotlib과 문법 유사
- 기본 Theme이 Plotly에 비해 깔끔, matplotlib이랑 유사하다.
- 비교적 부족한 문서화

Altair

- Vega 라이브러리를 사용하여 만든 인터랙티브
- 시각화를 + 연산 등으로 배치하는 것이 특징
- 문법이 Pythonic하지 않고 js 같고 커스텀이 불편
- 데이터 크기 5000개 제한되어 샘플링 시각화에 사용
- Bar, Line, Scatter, Histogram에 특화

▸ 6-2. Interactive Visualization(Plotly Express) - 코드 실습

- 과하게 사용하면 계산 부하만 생긴다.
- Multidimensional : 소비자의 행동 패턴 분석에 사용

[] ↩ 숨겨진 셀 48개

(7-1) Custom Matplotlib Theme

[Apple Human Interface Guidelines - Color](#)

[Google Material Design - Color](#)

[Color Palettes in Seaborn](#)

▶ 7-1. Custom Matplotlib Theme (Dark Mode) - 실습 코드

- matplotlib 테마 바꾸면 seaborn도 바뀐다.
- plotly는 color_palette 리스트로 변경하자.

[] ↪ 숨겨진 셀 30개

▶ (7-2) Image & Text Visualization Techniques - 실습 코드

- TSNE, UMAP: 속도 빠름
- IPython의 HTML을 사용하면 다양한 색상을 선택할 수 있음
- data-data relation network visualization은 파이썬 대신 다른 툴을 사용

[] ↪ 숨겨진 셀 41개