Week 1:
EDA (NumPy, Pandas) and Basic Tableau

Kuggle

Contents

Kuggle

1. Machine Learning & Visualization

- Machine Learning: Algorithms that learn patterns from data and make predictions.

- **EDA**
- Preprocessing
- Learning
- Evaluation
- Classification
- Regression Analysis

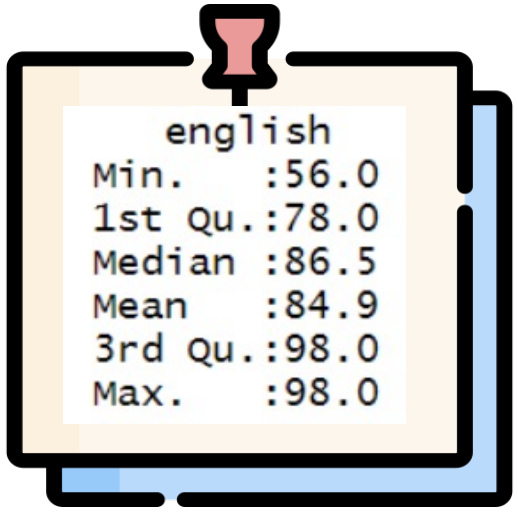
EDA

EDA (Exploratory Data Analysis): Understanding data through visualization and statistics

Types of EDA

(Graphic)

-> Visualization based

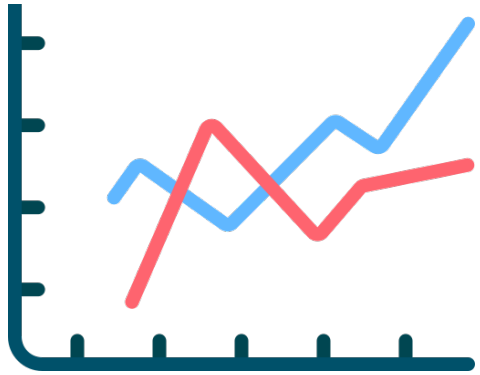


(Non-Graphic)

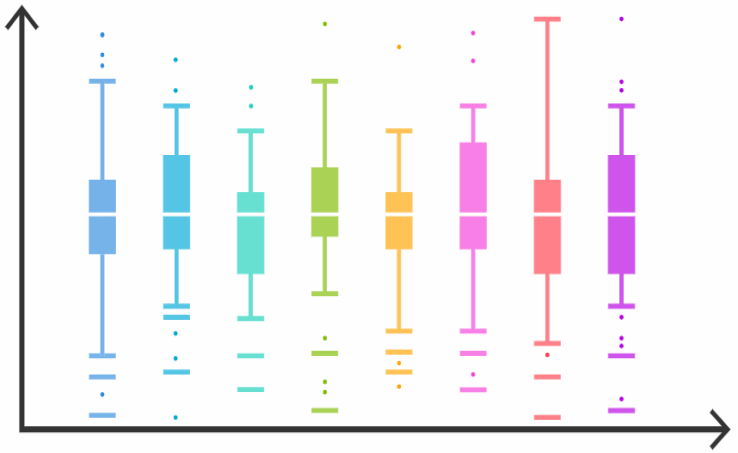
-> Summary Statistics based

EDA

Necessity of EDA



Understanding data shape



Discover data problems

Understand relationships between variables

Used for data preprocessing, feature selection, machine learning algorithm selection, etc.

Reference

Let's 태블로, 쉽게 따라하는 데이터 시각화

태블로 퍼블릭 무료버전 활용, 실습 중심의 콘텐츠 구성, 다수의 실전 연습문제 수록

종이책 27,000원 | eBook 22,500원

최정민 , 류민호 저자(글)

생능북스 · 2023년 04월 10일

주간베스트 컴퓨터/IT 432위

10.0 (16개의 리뷰)

도움돼요 (56%의 구매자)

무료배송 | 사은품 | 소득공제

10% 27,000원 ~~30,000원~~

적립/혜택 1,500P

배송안내 무료배송

내일(8/29,화) 도착예정

서울시 종로구 종로 1 변경

알림 신청하시면 원하시는 정보를 받아 보실 수 있습니다. 알림신청

매장 재고·위치

Importance

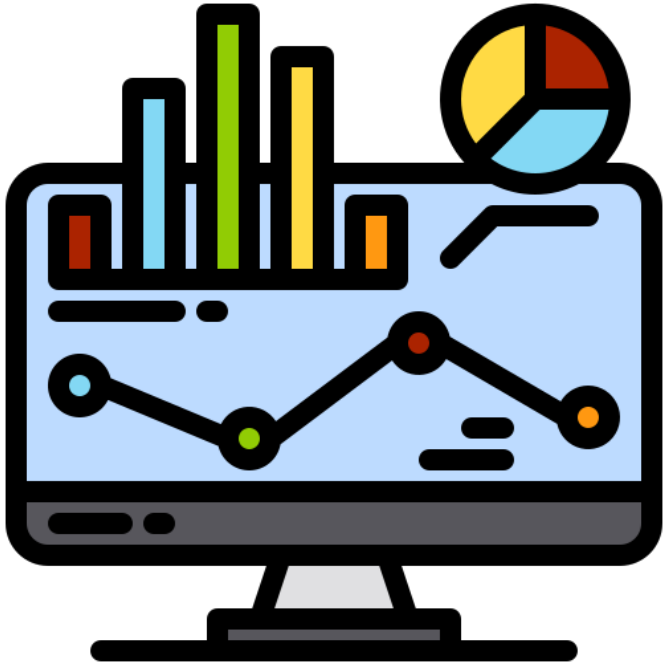
How many 9s are in the following data?

2 3 5 1 9 2 8 2 6 7
9 4 3 1 8 7 1 9 9 2
2 5 8 2 2 9 2 5 2 6
6 3 6 5 3 9 3 2 1 8
8 2 4 9 7 1 8 8 3 2
7 2 4 2 5 5 2 1 8 9
1 9 3 8 4 3 4 3 5 1
3 1 2 3 1 6 9 4 7 3
5 6 6 7 8 9 1 9 3 7
3 2 4 2 6 6 3 6 2 9

2 3 5 1 9 2 8 2 6 7
9 4 3 1 8 7 1 9 9 2
2 5 8 2 2 9 2 5 2 6
6 3 6 5 3 9 3 2 1 8
8 2 4 9 7 1 8 8 3 2
7 2 4 2 5 5 2 1 8 9
1 9 3 8 4 3 4 3 5 1
3 1 2 3 1 6 9 4 7 3
5 6 6 7 8 9 1 9 3 7
3 2 4 2 6 6 3 6 2 9

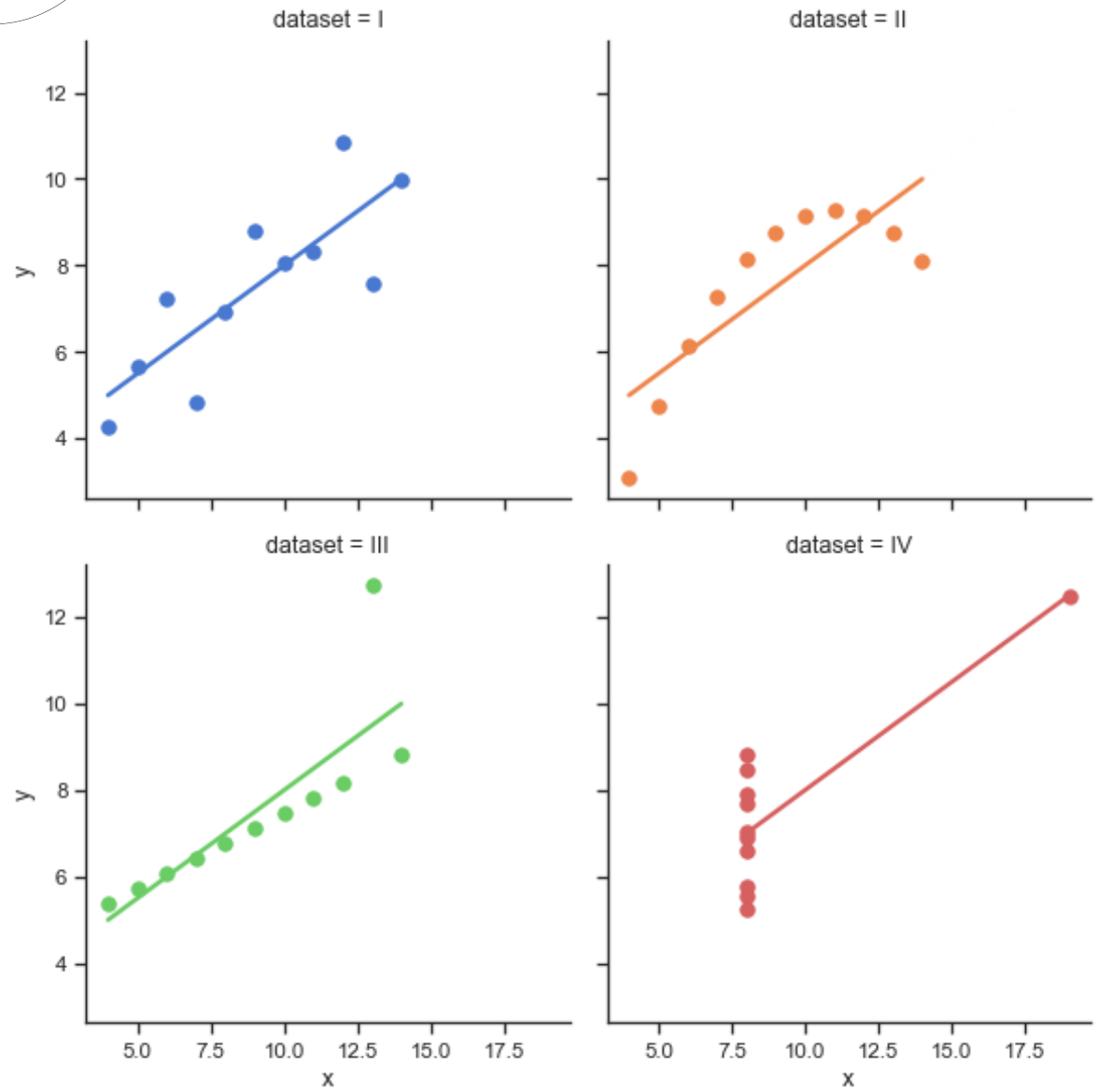
2 3 5 1 9 2 8 2 6 7
9 4 3 1 8 7 1 9 9 2
2 5 8 2 2 9 2 5 2 6
6 3 6 5 3 9 3 2 1 8
8 2 4 9 7 1 8 8 3 2
7 2 4 2 5 5 2 1 8 9
1 9 3 8 4 3 4 3 5 1
3 1 2 3 1 6 9 4 7 3
5 6 6 7 8 9 1 9 3 7
3 2 4 2 6 6 3 6 2 9

Data Literacy



Data Literacy : The ability to discover important meaning from data

Anscombe's quartet

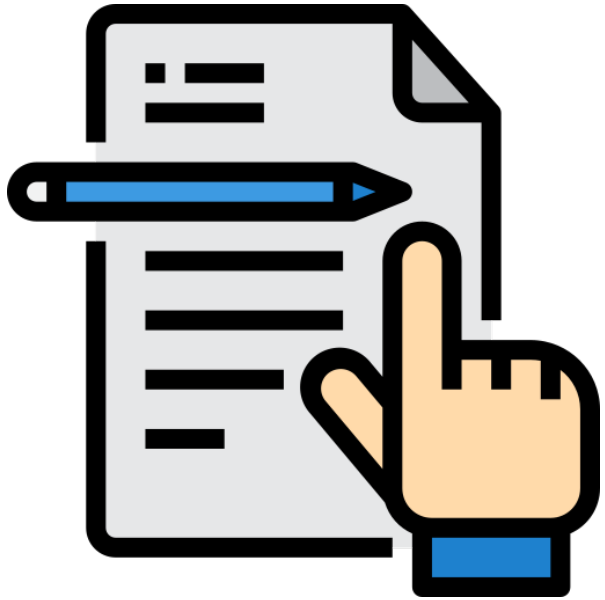


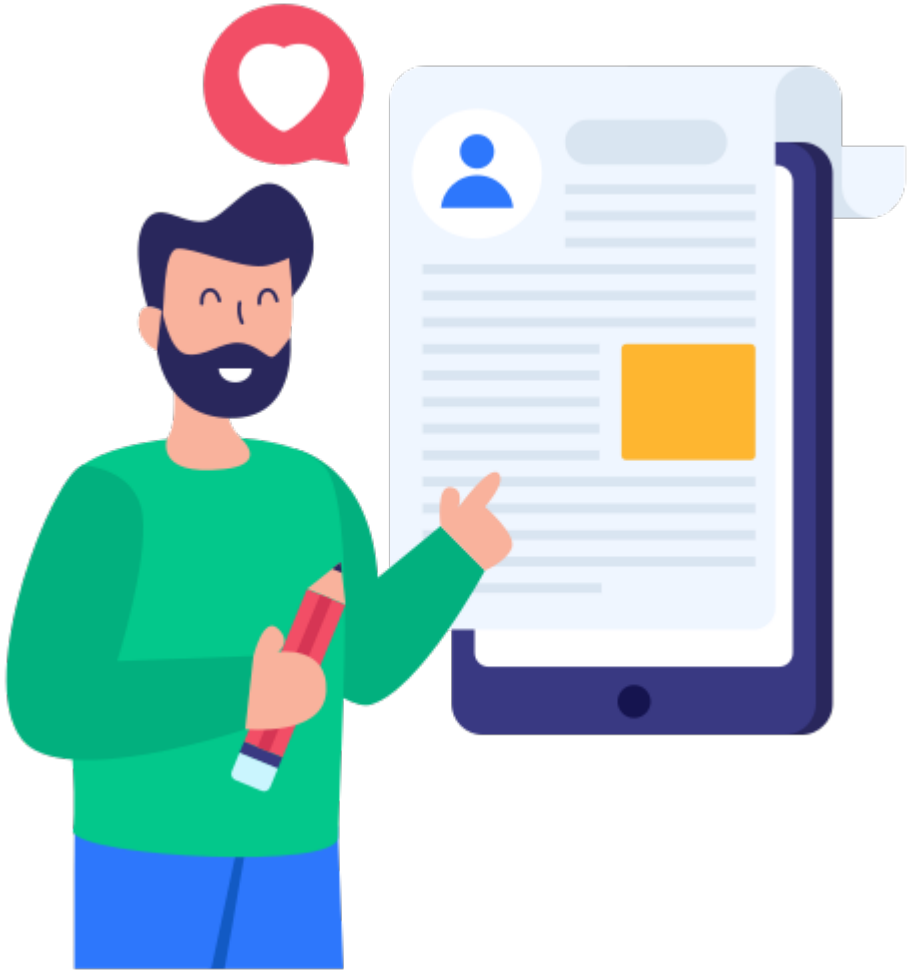
11 coordinates. 4 groups of datasets

Mean of X, Y,
Standard deviation of X,
Standard deviation of Y,
Correlation coefficient of X and Y

The 4 groups have identical values

What is Data Visualization?





Convey 'meaning' with 'context'

Increases understanding of the message

2. Tableau

2. Tableau

Tableau for Students

SV **SheerID Verification (On Behalf of Tableau) <Verify@SheerID.com>**
받는 사람: 노동환

Hi 동환,

Please retain this email for your records. You will need the product key and instructions enclosed.

Welcome to Tableau for Students! Your academic license now includes Tableau Desktop, Tableau Prep, and eLearning for free.

The product key below can be used to activate both Tableau Desktop and Tableau Prep on two separate computers, Windows or Mac. This key will expire in one year.

- Download Tableau Desktop
- Download Tableau Prep
- Activate with your product key: **TC**[redacted]
- If you're receiving the error "product key is invalid" visit the knowledge base page to resolve your issue.

Get started with free eLearning online self-paced courses:

1. Go to https://elearning.tableau.com
2. Create (or login to) your TableauID account, and confirm email address via the TableauID

Tableau for Students is a 1-year free license

다운로드 방법 참고 -https://blog.naver.com/gydnjs5238/223011468384

Tableau - 문서1

파일(F) 데이터(D) 서버(S) 도움말(H)

연결

데이터 검색

Tableau Server

파일에 연결

Microsoft Excel
텍스트 파일
JSON 파일
Microsoft Access
PDF 파일
공간 파일
통계 파일
자세히...

서버에 연결

Microsoft SQL Server
MySQL
Oracle
Amazon Redshift
자세히...

저장된 데이터 원본

세계 지표
슈퍼스토어 - 샘플
Sample - Superstore

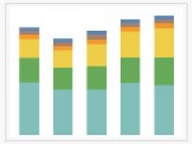
열기

통합 문서 열기

액셀러레이터

추가 액셀러레이터


슈퍼스토어


세계 지표

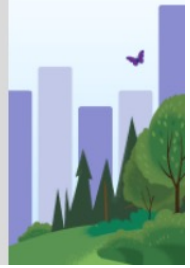
더 알아보기

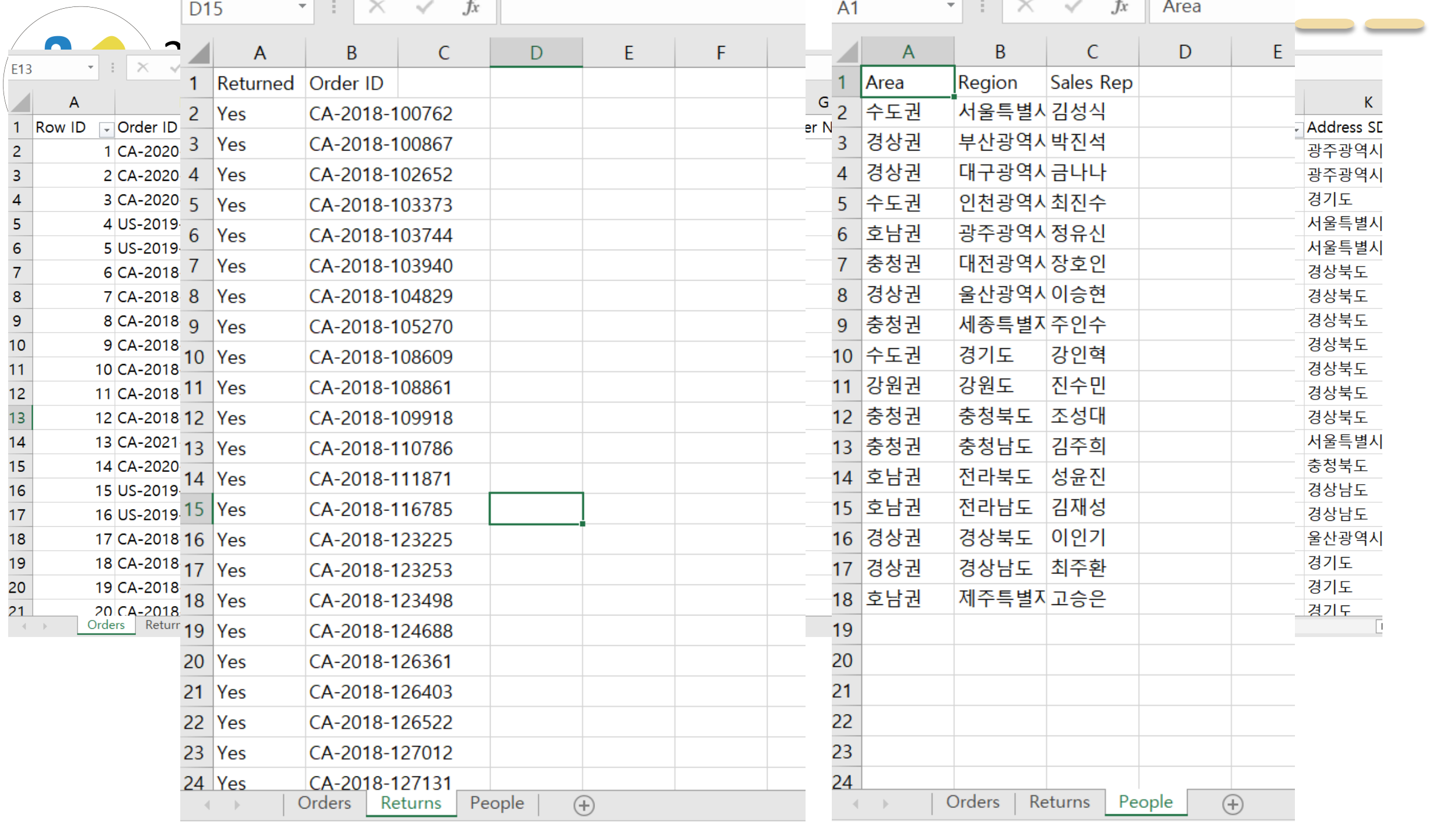
Tableau를 만나보십시오

시작하기
Tableau 환경 둘러보기
데이터 연결 및 준비
자세히 알아보기...

리소스

Tableau Prep 받기
Tableau Blueprint 평가
Tableau 커뮤니티 포럼
Tableau 액셀러레이터
블로그 - 최신 게시물 보기





E13

	A	Order ID
1	Row ID	Order ID
2	1	CA-2020
3	2	CA-2020
4	3	CA-2020
5	4	US-2019
6	5	US-2019
7	6	CA-2018
8	7	CA-2018
9	8	CA-2018
10	9	CA-2018
11	10	CA-2018
12	11	CA-2018
13	12	CA-2018
14	13	CA-2021
15	14	CA-2020
16	15	US-2019
17	16	US-2019
18	17	CA-2018
19	18	CA-2018
20	19	CA-2018
21	20	CA-2018

Orders | Returr

D15

	A	B	C	D	E	F
1	Returned	Order ID				
2	Yes	CA-2018-100762				
3	Yes	CA-2018-100867				
4	Yes	CA-2018-102652				
5	Yes	CA-2018-103373				
6	Yes	CA-2018-103744				
7	Yes	CA-2018-103940				
8	Yes	CA-2018-104829				
9	Yes	CA-2018-105270				
10	Yes	CA-2018-108609				
11	Yes	CA-2018-108861				
12	Yes	CA-2018-109918				
13	Yes	CA-2018-110786				
14	Yes	CA-2018-111871				
15	Yes	CA-2018-116785				
16	Yes	CA-2018-123225				
17	Yes	CA-2018-123253				
18	Yes	CA-2018-123498				
19	Yes	CA-2018-124688				
20	Yes	CA-2018-126361				
21	Yes	CA-2018-126403				
22	Yes	CA-2018-126522				
23	Yes	CA-2018-127012				
24	Yes	CA-2018-127131				

Orders | Returns | People

A1

Area

	A	B	C	D	E
1	Area	Region	Sales Rep		
2	수도권	서울특별시	김성식		
3	경상권	부산광역시	박진석		
4	경상권	대구광역시	금나나		
5	수도권	인천광역시	최진수		
6	호남권	광주광역시	정유신		
7	충청권	대전광역시	장호인		
8	경상권	울산광역시	이승현		
9	충청권	세종특별자	주인수		
10	수도권	경기도	강인혁		
11	강원권	강원도	진수민		
12	충청권	충청북도	조성대		
13	충청권	충청남도	김주희		
14	호남권	전라북도	성윤진		
15	호남권	전라남도	김재성		
16	경상권	경상북도	이인기		
17	경상권	경상남도	최주환		
18	호남권	제주특별자	고승은		
19					
20					
21					
22					
23					
24					

Orders | Returns | People

K
Address SD
광주광역시
광주광역시
경기도
서울특별시
서울특별시
경상북도
경상북도
경상북도
경상북도
경상북도
경상북도
경상북도
서울특별시
충청북도
경상남도
경상남도
울산광역시
경기도
경기도
경기도

Tableau - 문서1

파일(F) 데이터(D) 서버(S) 창(N) 도움말(H)

Superstore KR - VizLab 2022

연결 추가

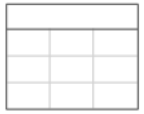
Superstore KR - VizLab 2022
Microsoft Excel

시트

- Orders
- People
- Returns

새 유니온

새 테이블 확장 프로그램



여기로 테이블 끌기

데이터 원본 시트 1

Orders(Superstore KR - VizLab 2022)

연결 추가

Superstore KR - VizLab 2022
Microsoft Excel

시트

- Orders
- People
- Returns

새 유니온

새 테이블 확장 프로그램

연결
라이브 추출

필터
0 추가

Orders

더 많은 데이터가 필요하십니까?

여기에 테이블을 끌어놓아 관계를 만드십시오. 자세히 알아보기

Orders 20개 필드 9994개 행 100 행

이름
Orders

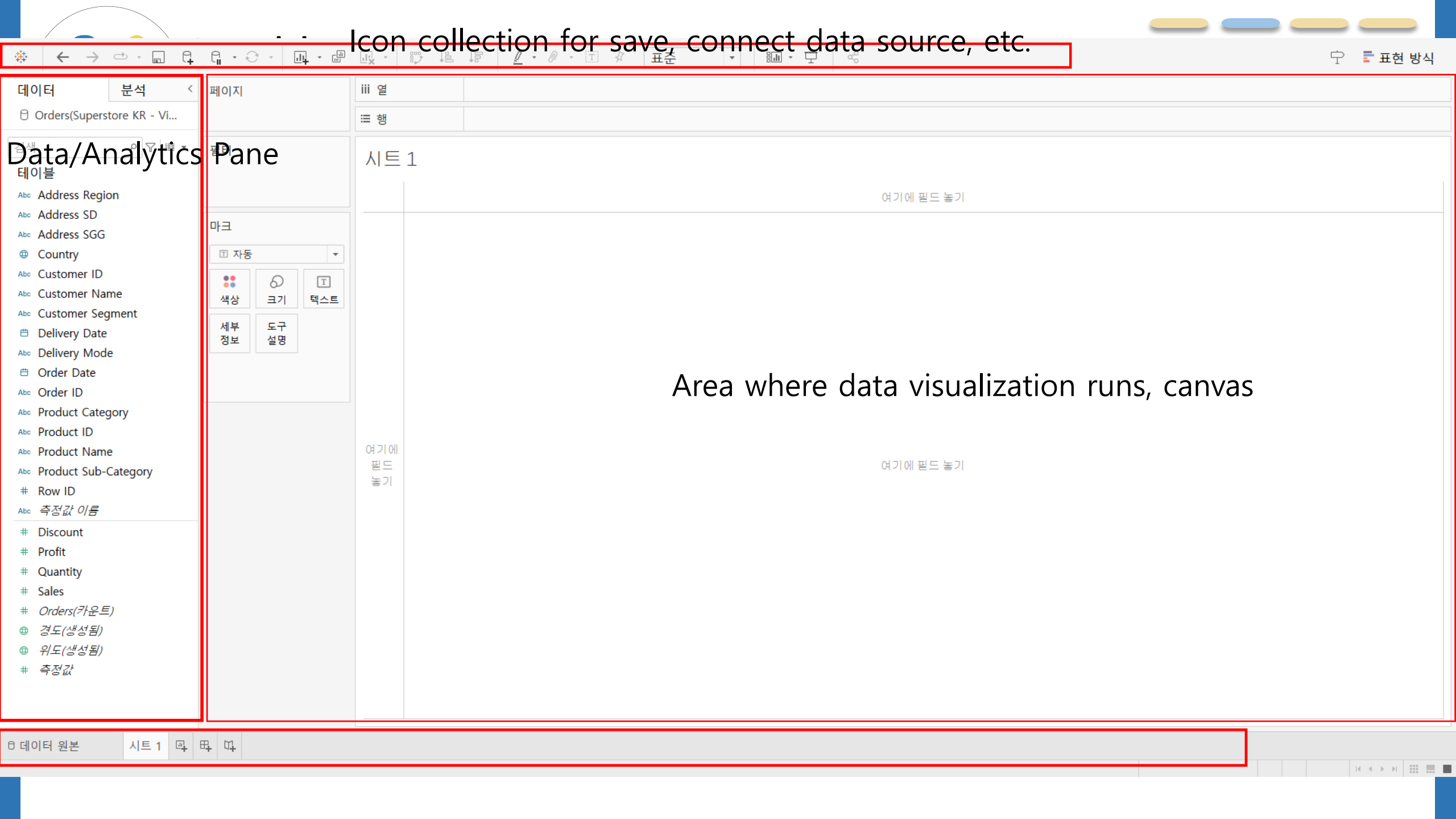
필드

유형	필드명	물리적 테이블	원격 필드명
#	Row ID	Orders	Row ID
Abc	Order ID	Orders	Order ID
📅	Order Date	Orders	Order Date

Orders Row ID	Orders Order ID	Orders Order Date	Orders Delivery Date	Orders Delivery Mode	Orders Customer ID	Orders Customer Name
1	CA-2020-152156	2020-11-08	2020-11-11	빠른배송	CG-12520	류미령
2	CA-2020-152156	2020-11-08	2020-11-11	빠른배송	CG-12520	류미령
3	CA-2020-138688	2020-06-12	2020-06-16	빠른배송	DV-13045	문윤재
4	US-2019-108966	2019-10-11	2019-10-18	일반배송	SO-20335	천경아
5	US-2019-108966	2019-10-11	2019-10-18	일반배송	SO-20335	천경아
6	CA-2018-115812	2018-06-09	2018-06-14	일반배송	BH-11710	나정진

워크시트로 이동

데이터 원본 시트 1


Icon collection for save, connect data source, etc.
표준
표현 방식
데이터
분석
Orders(Superstore KR - Vi...
Data/Analytics Pane
테이블
Abc Address Region
Abc Address SD
Abc Address SGG
Country
Abc Customer ID
Abc Customer Name
Abc Customer Segment
Delivery Date
Abc Delivery Mode
Order Date
Abc Order ID
Abc Product Category
Abc Product ID
Abc Product Name
Abc Product Sub-Category
Row ID
Abc 측정값 이름
Discount
Profit
Quantity
Sales
Orders(카운트)
경도(생성됨)
위도(생성됨)
측정값
페이지
열
행
시트 1
여기에 필드 놓기
마크
자동
색상
크기
텍스트
세부 정보
도구 설명
여기에 필드 놓기
여기에 필드 놓기
Area where data visualization runs, canvas
데이터 원본
시트 1

페이지

필터

마크

자동

색상 | 크기 | 레이블

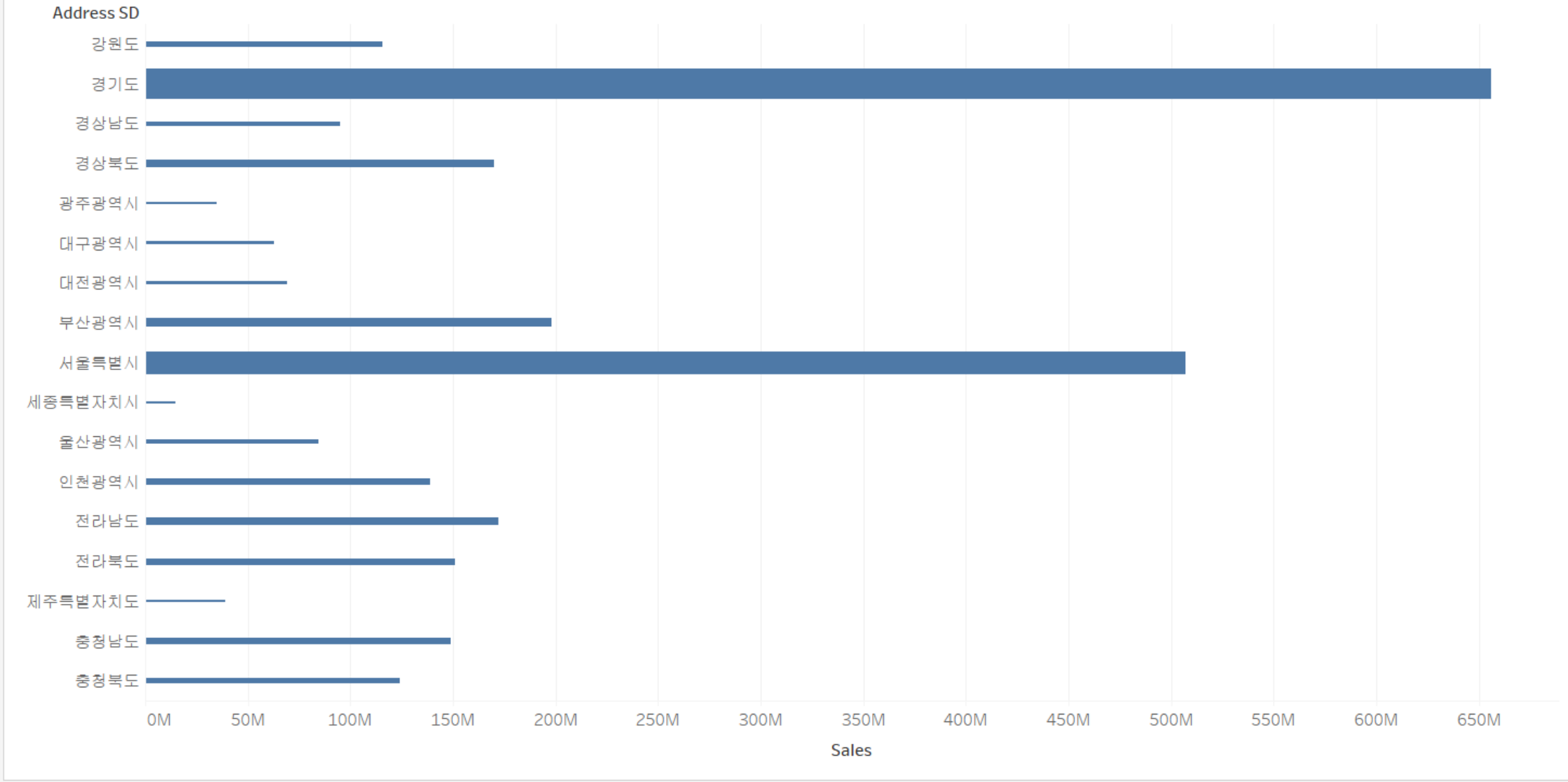
세부 정보 | 도구 설명

합계(Sales)

합계(Sales)
14,743,754
200,000,000
400,000,000
655,986,797

열 합계(Sales)

행 Address SD



페이지

필터

마크

자동

색상 | 크기 | 레이블

세부 정보 | 도구 설명

Customer Segm..

합계(Sales)

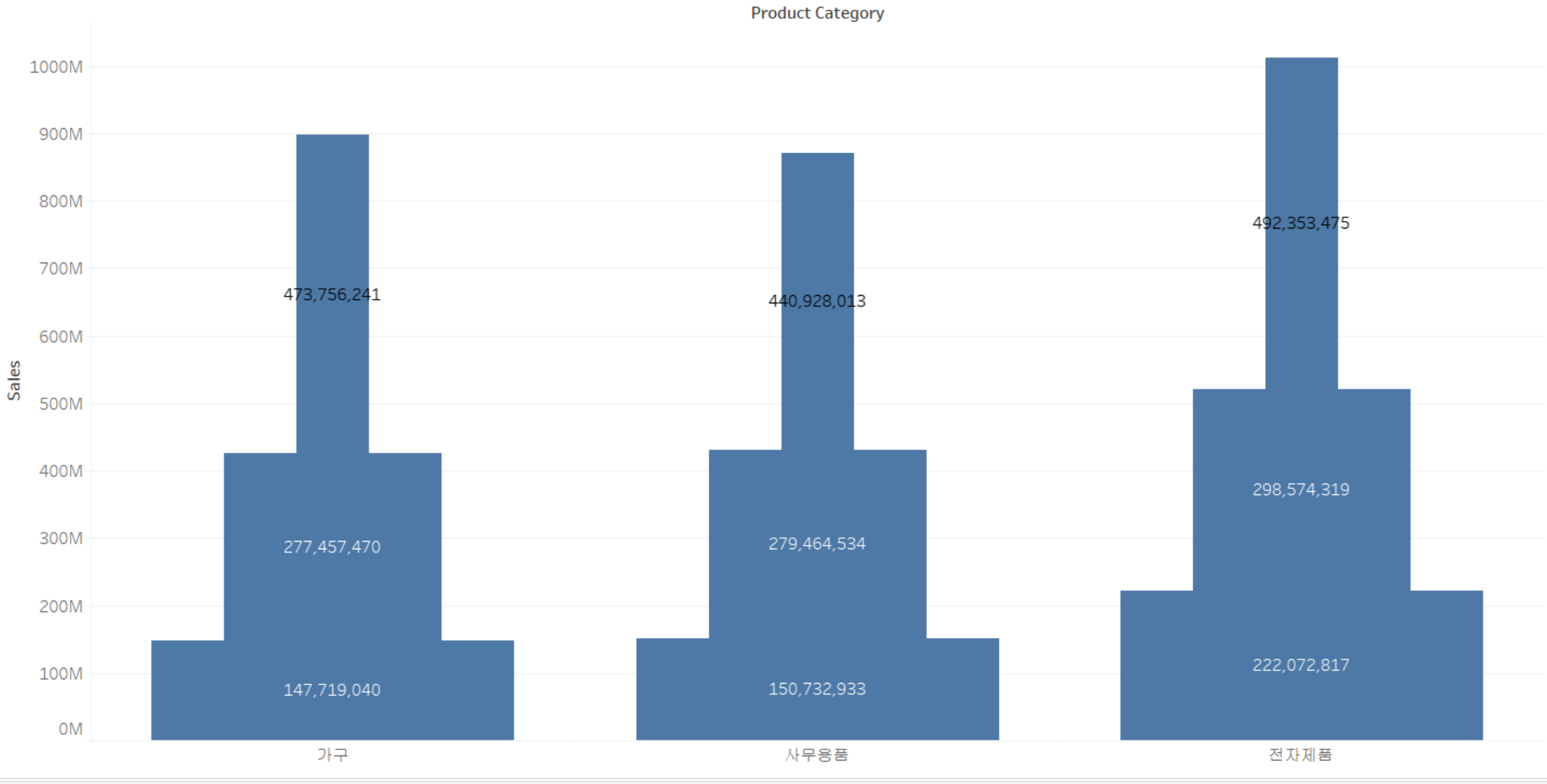
Customer Segment

- 개인고객
- 기업고객
- 홈오피스

열 Product Category

행 합계(Sales)

시트 1



필터

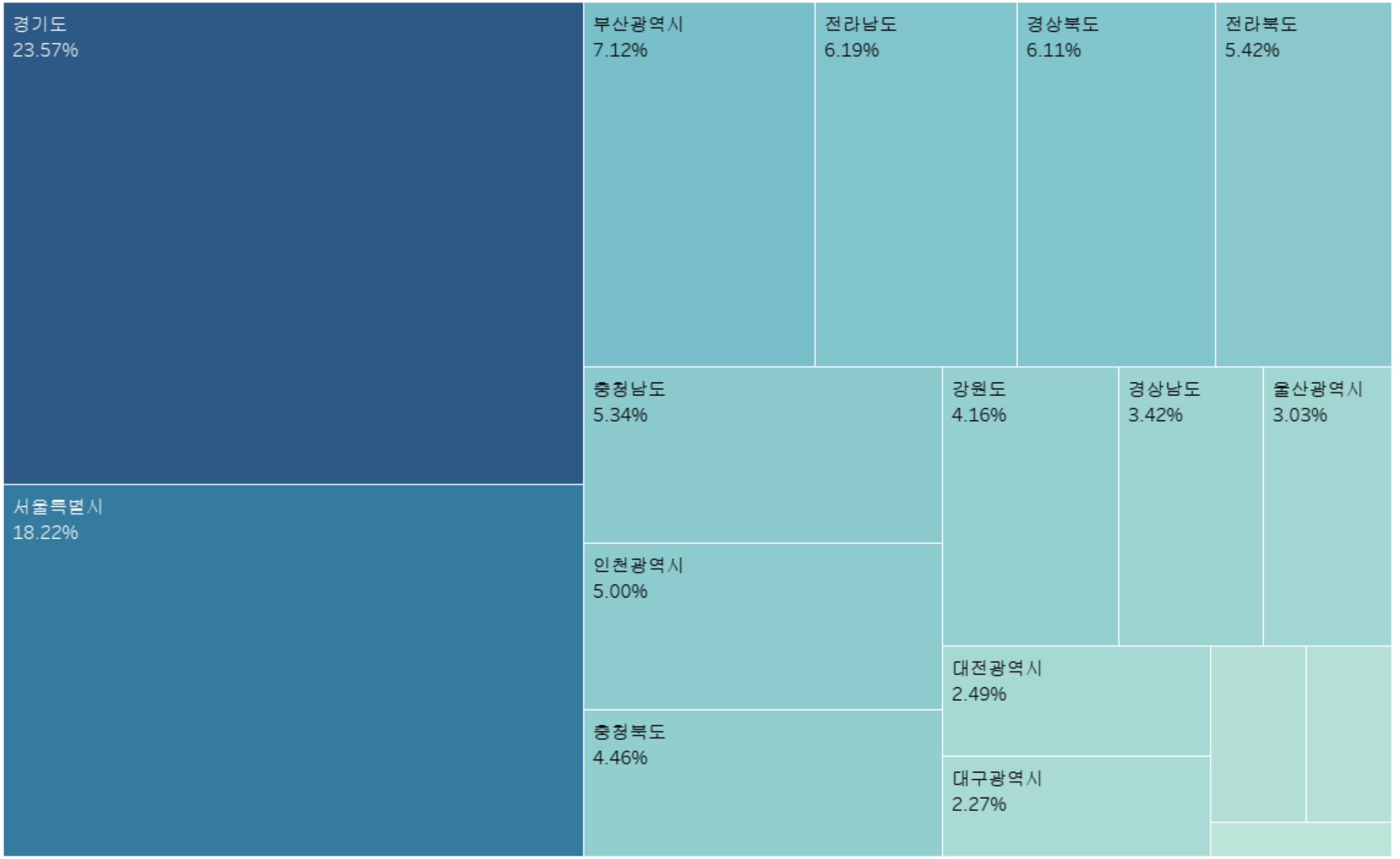
마크

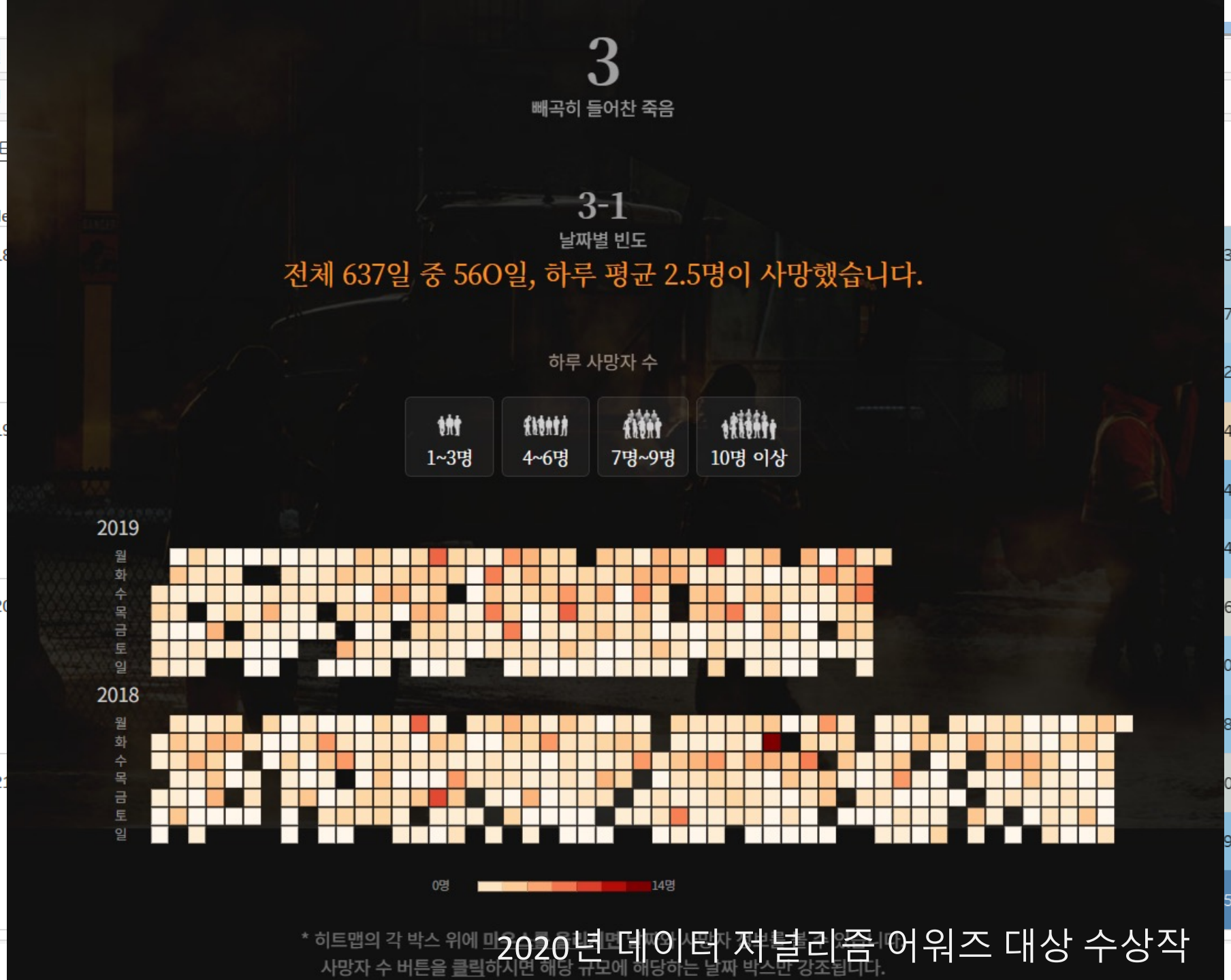
자동

색상 | 크기 | 레이블 | 세부 정보 | 도구 설명

합계(Sales)
합계(Sales)
Address SD
합계(Sales) Δ

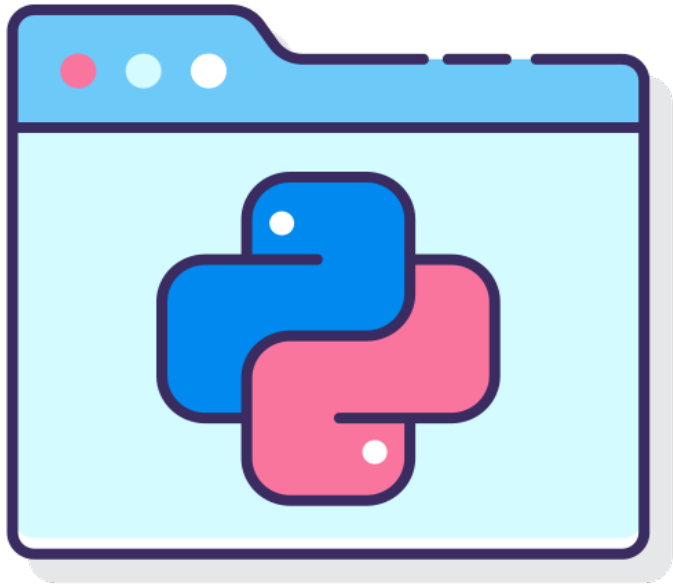
시트 1





2020년 데이터 저널리즘 어워즈 대상 수상작

Kuggle



3. Numpy

3. Numpy

Reference

파이썬 머신러닝 완벽 가이드

다양한 캐글 예제와 함께 기초 알고리즘부터 최신 기법까지 배우는 | 2 판

위키북스 데이터 사이언스 시리즈 81

권철민 저자(글)

위키북스 · 2022년 04월 21일

주간베스트 컴퓨터/IT 300위

가장 최근에 출시된 개정판입니다. 구판보기 >

10.0

(13개의 리뷰)

집중돼요

(23%의 구매자)

MD의 선택 | 무료배송 | 이벤트 | 소득공제

10% 36,000원 ~~40,000원~~

적립/혜택 2,000P

배송안내 무료배송

9월 5일(화) 도착예정

서울시 종로구 종로 1 변경 >

알림 신청하시면 원하시는 정보를 받아 보실 수 있습니다. 알림신청

매장 재고·위치

3. Numpy

Package

Package: A directory of modules (A collection of various functions)

[Packages primarily used in Machine Learning]

-Scikit-learn: Package for implementing machine learning algorithms

- **Numpy:** Package for handling matrices and linear algebra
- **Pandas:** Matrix-based data handling package
- **matplotlib / seaborn:** Visualization packages

3. Numpy

numpy

Numpy = Numerical Python (Vector, matrix operations possible)

Installation method

```
import numpy as np
```

ndarray: Numpy's base data type (can represent multi-dimensional arrays)

〈 넘파이 ndarray 배열의 차원들 〉

3. Numpy

Generating ndarray

1. Create ndarray: array() function

```
array1 = np.array([1,2,3])
array2 = np.array([[1,2,3]])
```

The difference? -> array1 is 1D data (3,), array2 is 2D data (1,3)

+) arange(n): Create an ndarray with n values starting from 0

+) zeros() / ones(): Create an ndarray filled with 0s/1s when shape is specified

3. Numpy

Data types

1.Data types of ndarray

- All data types possible

- Only one type within one ndarray! <-> Lists can have multiple types

2. Change data type: astype()

```
array_int = np.array([1, 2, 3])
array_float = array_int.astype('float64’)
print(array_float, array_float.dtype)
```

[1. 2. 3.] float64

3. Numpy

reshape

3. Change ndarray dimension, size: reshape()

*Using -1 in reshape() => automatically converts to a compatible dimension

(Mainly use reshape(-1,1) to convert the original to a 2D ndarray with 1 column)

```
array5 = array3d.reshape(-1, 1)
print('array5:\n',array5.tolist())
print('array5 shape:',array5.shape)
```

```
array3d:
 [[[0, 1], [2, 3]], [[4, 5], [6, 7]]]
array5:
 [[0], [1], [2], [3], [4], [5], [6], [7]]
array5 shape: (8, 1)
```

3. Numpy

indexing

4. Selecting parts of an ndarray – indexing

1) Extract single value: Specify index value in [] (index starts from 0)
2) Slicing: Select continuous data using ':'
3) Fancy Indexing: Return ndarray by specifying an index set
4) Boolean Indexing: Select only items matching the condition by writing a conditional statement in []

3. Numpy

etc

5. Other miscellaneous Numpy features

1) Matrix sorting

- Matrix sort: sort()

- Return original indices upon sorting: argsort()

2) Matrix calculation

- Dot product : **dot**()

- Matrix Transpose : **transpose**()

Kuggle

4. Pandas

4. Pandas

pandas

Pandas : Data handling package

```
import pandas as pd
```

[Pandas' basic data types]

- **DataFrame** : 2D data composed of multiple rows and columns
- **Series**: 1-column data

4. Pandas

Uploading files

1. Loading a file into a DataFrame

- **read_csv**(), read_table(), read_fwf() etc

2. Exploring data (EDA)

- View first/last n data: head(n) / tail(n) (default=5)

```
titanic_df.head() ; titanic_df.tail()
```

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarked
0	1	0	3	Braund, Mr. Owen Harris	male	22.0	1	0	A/5 21171	7.2500	NaN	S
1	2	1	1	Cumings, Mrs. John Bradley (Florence Briggs Th...	female	38.0	1	0	PC 17599	71.2833	C85	C
2	3	1	3	Heikkinen, Miss. Laina	female	26.0	0	0	STON/O2. 3101282	7.9250	NaN	S
3	4	1	1	Futrelle, Mrs. Jacques Heath (Lily May Peel)	female	35.0	1	0	113803	53.1000	C123	S
4	5	0	3	Allen, Mr. William Henry	male	35.0	0	0	373450	8.0500	NaN	S

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarked
886	887	0	2	Montvila, Rev. Juozas	male	27.0	0	0	211536	13.00	NaN	S
887	888	1	1	Graham, Miss. Margaret Edith	female	19.0	0	0	112053	30.00	B42	S
888	889	0	3	Johnston, Miss. Catherine Helen "Carrie"	female	NaN	1	2	W./C. 6607	23.45	NaN	S
889	890	1	1	Behr, Mr. Karl Howell	male	26.0	0	0	111369	30.00	C148	C
890	891	0	3	Dooley, Mr. Patrick	male	32.0	0	0	370376	7.75	NaN	Q

4. Pandas

EDA

- Identify data count, type, missing values: info()

```
titanic_df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 891 entries, 0 to 890
Data columns (total 12 columns):
 #   Column       Non-Null Count  Dtype
---  ------       --------------  -----
 0   PassengerId  891 non-null    int64
 1   Survived     891 non-null    int64
 2   Pclass       891 non-null    int64
 3   Name         891 non-null    object
 4   Sex          891 non-null    object
 5   Age          714 non-null    float64
 6   SibSp        891 non-null    int64
 7   Parch        891 non-null    int64
 8   Ticket       891 non-null    object
 9   Fare         891 non-null    float64
 10  Cabin        204 non-null    object
 11  Embarked     889 non-null    object
dtypes: float64(2), int64(5), object(5)
memory usage: 83.7+ KB
```

4. Pandas

EDA

- Identify summary statistics (numeric columns only): describe()

```
titanic_df.describe()
```

	PassengerId	Survived	Pclass	Age	SibSp	Parch	Fare
count	891.000000	891.000000	891.000000	714.000000	891.000000	891.000000	891.000000
mean	446.000000	0.383838	2.308642	29.699118	0.523008	0.381594	32.204208
std	257.353842	0.486592	0.836071	14.526497	1.102743	0.806057	49.693429
min	1.000000	0.000000	1.000000	0.420000	0.000000	0.000000	0.000000
25%	223.500000	0.000000	2.000000	20.125000	0.000000	0.000000	7.910400
50%	446.000000	0.000000	3.000000	28.000000	0.000000	0.000000	14.454200
75%	668.500000	1.000000	3.000000	38.000000	1.000000	0.000000	31.000000
max	891.000000	1.000000	3.000000	80.000000	8.000000	6.000000	512.329200

4. Pandas

EDA

- Check DataFrame size: **shape**()

```
print('DataFrame 크기: ', titanic_df.shape)
```

```
DataFrame 크기:  (891, 12)
```

- Check data distribution: **value_counts**()

```
titanic_df['Pclass'].value_counts()
```

```
3    491
1    216
2    184
Name: Pclass, dtype: int64
```

4. Pandas

Preprocessing

3. Preprocessing

Create new column / Modify column: Use [] operator

```
titanic_df['Age_0’]=0
titanic_df['Age_by_10'] = titanic_df['Age']*10
titanic_df['Family_No'] = titanic_df['SibSp'] + titanic_df['Parch']+1
titanic_df.head(3)
```

engerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarked	Age_0	Age_by_10	Family_No
1	0	3	Braund, Mr. Owen Harris	male	22.0	1	0	A/5 21171	7.2500	NaN	S	0	220.0	2
2	1	1	Cumings, Mrs. John Bradley (Florence Briggs Thayer)	female	38.0	1	0	PC 17599	71.2833	C85	C	0	380.0	2
3	1	3	Heikkinen, Miss. Laina	female	26.0	0	0	STON/O2. 3101282	7.9250	NaN	S	0	260.0	1

4. Pandas

Preprocessing

Delete data: drop()

```
titanic_drop_df = titanic_df.drop('Age_0', axis=1 )
titanic_drop_df.head(3)
```

axis=0 : row
axis=1 : column

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarked	Age_by_10	Family_No
0	1	0	3	Braund, Mr. Owen Harris	male	22.0	1	0	A/5 21171	7.2500	NaN	S	320.0	2
1	2	1	1	Cumings, Mrs. John Bradley (Florence Briggs Thayer)	female	38.0	1	0	PC 17599	71.2833	C85	C	480.0	2
2	3	1	3	Heikkinen, Miss. Laina	female	26.0	0	0	STON/O2. 3101282	7.9250	NaN	S	360.0	1

4. Pandas

Preprocessing

Use inplace: If set to True, overwrites with changed settings

```
drop_result = titanic_df.drop(['Age_0', 'Age_by_10', 'Family_No'], axis=1, inplace=True)
print(' inplace=True 로 drop 후 반환된 값:',drop_result)
titanic_df.head(3)
```

```
inplace=True 로 drop 후 반환된 값: None
```

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarked
0	1	0	3	Braund, Mr. Owen Harris	male	22.0	1	0	A/5 21171	7.2500	NaN	S
1	2	1	1	Cumings, Mrs. John Bradley (Florence Briggs Thayer)	female	38.0	1	0	PC 17599	71.2833	C85	C
2	3	1	3	Heikkinen, Miss. Laina	female	26.0	0	0	STON/O2. 3101282	7.9250	NaN	S

4. Pandas

Preprocessing

Check missing values: isna()

```
titanic_df.isna().head(3)

titanic_df.isna( ).sum( )
```

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarked
0	False	False	False	False	False	False	False	False	False	False	True	False
1	False	False	False	False	False	False	False	False	False	False	False	False
2	False	False	False	False	False	False	False	False	False	False	True	False

```
PassengerId      0
Survived         0
Pclass           0
Name             0
Sex              0
Age            177
SibSp            0
Parch            0
Ticket           0
Fare             0
Cabin          687
Embarked         2
dtype: int64
```

4. Pandas

Preprocessing

Replace missing values: fillna()

```
titanic_df['Cabin'] = titanic_df['Cabin'].fillna('C000’)
titanic_df['Age'] = titanic_df['Age'].fillna(titanic_df['Age'].mean())
titanic_df['Embarked'] = titanic_df['Embarked'].fillna('S')
titanic_df.head(3)
titanic_df.isna().sum()
```

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarked
0	1	0	3	Braund, Mr. Owen Harris	male	22.0	1	0	A/5 21171	7.2500	C000	S
1	2	1	1	Cumings, Mrs. John Bradley (Florence Briggs Thayer)	female	38.0	1	0	PC 17599	71.2833	C85	C
2	3	1	3	Heikkinen, Miss. Laina	female	26.0	0	0	STON/O2. 3101282	7.9250	C000	S

```
PassengerId    0
Survived       0
Pclass         0
Name           0
Sex            0
Age            0
SibSp          0
Parch          0
Ticket         0
Fare           0
Cabin          0
Embarked       0
dtype: int64
```

4. Pandas

Data selection

4. Data selection

Select columns: Use [] operator

```
filtered_df = titanic_df[['Name', 'Age']]
display(filtered_df.head(3))
```

	Name	Age
0	Braund, Mr....	22.0
1	Cumings, Mr...	38.0
2	Heikkinen, ...	26.0

4. Pandas

Data selection

- Selecting row : **iloc[], loc[]**

  * What is index?

```
data = {'Name': ['Chulmin', 'Eunkyung','Jinwoong','Soobeom'],
        'Year': [2011, 2016, 2015, 2015],
        'Gender': ['Male', 'Female', 'Male', 'Male’] }
data_df = pd.DataFrame(data, index=['one','two','three','four'])
data_df
```

	Name	Year	Gender
one	Chulmin	2011	Male
two	Eunkyung	2016	Female
three	Jinwoong	2015	Male
four	Soobeom	2015	Male

4. Pandas

Data selection

Position-based indexing: iloc[]

- Specify 0-based position indexing (integer)
- End is not included

data_df.iloc[0, 0]

```
'Chulmin'
```

data_df.iloc['one', 0]

```
---------------------------------------------------------------------------
ValueError                                Traceback (most recent call last)
~₩anaconda3₩lib₩site-packages₩pandas₩core₩indexing.py in _has_valid_tuple(self, key)
    753             try:
--> 754                 self._validate_key(k, i)
    755             except ValueError as err:
```

data_df.iloc[0:2, [0,1]]

	Name	Year
one	Chulmin	2011
two	Eunkyung	2016

4. Pandas

Data selection

Label-based indexing: loc[]

Specify index value (string), column name

End is included

```
data_df.loc['one', 'Name']
```

```
'Chulmin'
```

```
data_df.loc[0, 'Name']
```

```
---------------------------------------------------------------------------
ValueError                                Traceback (most recent call last)
~₩anaconda3₩lib₩site-packages₩pandas₩core₩indexing.py in _has_valid_tuple(self, key)
    753             try:
--> 754                 self._validate_key(k, i)
    755             except ValueError as err:
```

```
data_df.loc['one':'two', ['Name', 'Year']]
```

	Name	Year
one	Chulmin	2011
two	Eunkyung	2016

4. Pandas

Data selection

Boolean indexing (conditional extraction): Use [] operator

```
titanic_df[ titanic_df['Pclass'] == 3].head(3)
```

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarked
0	1	0	3	Braund, Mr....	male	22.0	1	0	A/5 21171	7.250	NaN	S
2	3	1	3	Heikkinen, ...	female	26.0	0	0	STON/O2. 31...	7.925	NaN	S
4	5	0	3	Allen, Mr. ...	male	35.0	0	0	373450	8.050	NaN	S

```
titanic_df[titanic_df['Age'] > 60][['Name','Age']].head(3)
```

```
titanic_df.loc[titanic_df['Age'] > 60, ['Name','Age']].head(3)
```

	Name	Age
33	Wheadon, Mr. Edward H	66.0
54	Ostby, Mr. Engelhart Cornelius	65.0
96	Goldschmidt, Mr. George B	71.0

4. Pandas

etc

5. etc

- Sorting data : **sort_values**()

```
titanic_sorted = titanic_df.sort_values(by=['Name'])

titanic_sorted.head(3)
```

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarked
845	846	0	3	Abbing, Mr. Anthony	male	42.0	0	0	C.A. 5547	7.55	NaN	S
746	747	0	3	Abbott, Mr. Rossmore Edward	male	16.0	1	1	C.A. 2673	20.25	NaN	S
279	280	1	3	Abbott, Mrs. Stanton (Rosa Hunt)	female	35.0	1	1	C.A. 2673	20.25	NaN	S

```
titanic_sorted = titanic_df.sort_values(by=['Pclass', 'Name'], ascending=False)

titanic_sorted.head(3)
```

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarked
868	869	0	3	van Melkebeke, Mr. Philemon	male	NaN	0	0	345777	9.5	NaN	S
153	154	0	3	van Billiard, Mr. Austin Blyler	male	40.5	0	2	A/5. 851	14.5	NaN	S
282	283	0	3	de Pelsmaeker, Mr. Alfons	male	16.0	0	0	345778	9.5	NaN	S

T = Ascending (default) / F = Descending

4. Pandas

etc

Group operation: groupby()

```
titanic_groupby = titanic_df.groupby('Pclass').count()
titanic_groupby
```

Pclass	PassengerId	Survived	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarked
1	216	216	216	216	186	216	216	216	216	176	214
2	184	184	184	184	173	184	184	184	184	16	184
3	491	491	491	491	355	491	491	491	491	12	491

```
titanic_df.groupby('Pclass')['Age'].agg([max, min])
```

Pclass	max	min
1	80.0	0.92
2	70.0	0.67
3	74.0	0.42

Homework ^^ ❤️

1. Try data preprocessing using pandas

	지역명	규모구분	연도	월	분양가격(제곱미터)
0	서울	모든면적	2015	10	5841
1	서울	전용면적 60제곱미터이하	2015	10	5652
2	서울	전용면적 60제곱미터초과 85제곱미터이하	2015	10	5882
3	서울	전용면적 85제곱미터초과 102제곱미터이하	2015	10	5721
4	서울	전용면적 102제곱미터초과	2015	10	5879
...	...	...	...	...	...
7900	제주	모든면적	2023	6	7326
7901	제주	전용면적 60제곱미터이하	2023	6	7381
7902	제주	전용면적 60제곱미터초과 85제곱미터이하	2023	6	7084
7903	제주	전용면적 85제곱미터초과 102제곱미터이하	2023	6	6639
7904	제주	전용면적 102제곱미터초과	2023	6	7506

7905 rows × 5 columns

2. Try visualizing Anscombe's Quartet