

⋮ 문제해결을 위한 데이터 분석 및 시각화

데이터 분석을 위한 기초통계이해

한성대학교 노은희 교수

“미래로 향하는 새로운 이정표”



오늘의 학습

학습내용

- 기초통계 이해
- 실습 : 복권 당첨 번호 빈도수 구하기

[파일 다운받기]

 서울글로벌센터 상담실적.csv

서울글로벌센터 상담실적.csv - Excel

파일 홈 삽입 페이지 레이아웃 수식 데이터 검토 보기 개발 도구 파워 피벗 ? 수행할 작업을 알려 주세요

잘라내기 붙여넣기 복사 서식 복사 클립보드 글꼴 맞춤 표시 형식

맑은 고딕 11 가 가 텍스트 줄 바꿈 일반

가 가 간 가 나 가 나 병합하고 가운데 맞춤 % , .00 0.0

A1 X ✓ fx 년월

	A	B	C	D	E	F	G	H	I	J	K
1	년월	기관	전화	방문	이메일	게시판	화상상담	이동상담	합계		
2	202202	교육	128	11	1	0	0	0	140		
3	202202	교통	4	1	0	0	0	0	5		
4	202202	금융	17	3	0	0	0	0	20		
5	202202	기타	123	5	2	0	0	0	130		
6	202202	노무	348	16	1	0	0	0	365		
7	202202	문화/관광	6	1	0	0	0	0	7		
8	202202	보건/복지	462	43	9	0	0	0	514		
9	202202	비즈니스	14	2	0	0	0	0	16		
10	202202	안전	14	0	1	0	0	0	15		
11	202202	여성/가족	19	0	1	0	0	0	20		
12	202202	전년도 총	1807	195	33	0	0	0	2035		
13	202202	정보통신	11	2	0	0	0	0	13		
14	202202	주거	38	5	0	0	0	0	43		
15	202202	출입국	357	72	2	0	0	0	431		
16	202202	행정	266	34	16	0	0	0	316		
17	202201	교육	153	14	1	0	0	0	168		
18	202201	교통	4	2	0	0	0	0	6		
19	202201	금융	9	0	0	0	0	0	9		
20	202201	기타	136	12	4	0	0	0	152		

서울글로벌센터 상담실적

파일 불러오기 `read_csv()` 함수를 사용해서 csv 파일을 불러와서 DataFrame으로 저장

```
import pandas as pd
df = pd.read_csv('서울글로벌센터 상담실적.csv', encoding = 'cp949')
df
```

`encoding = 'cp949' => 한글 인코딩의 한 종류`

	년월	기관	전화	방문	이메일	게시판	화상상담	이동상담	합계
0	202202	교육	128	11	1	0	0	0	140
1	202202	교통	4	1	0	0	0	0	5
2	202202	금융	17	3	0	0	0	0	20
3	202202	기타	123	5	2	0	0	0	130
4	202202	노무	348	16	1	0	0	0	365
5	202202	문화/관광	6	1	0	0	0	0	7
6	202202	보건/복지	462	43	9	0	0	0	514
7	202202	비즈니스	14	2	0	0	0	0	16
8	202202	안전	14	0	1	0	0	0	15
9	202202	여성/가족	19	0	1	0	0	0	20
10	202202	전년도 총계	1807	195	33	0	0	0	2035

중간생략

	A	B	C	D	E	F	G	H	I	J	K
1	년월	기관	전화	방문	이메일	게시판	화상상담	이동상담	합계		
2	202202	교육	128	11	1	0	0	0	140		
3	202202	교통	4	1	0	0	0	0	5		
4	202202	금융	17	3	0	0	0	0	20		
5	202202	기타	123	5	2	0	0	0	130		
6	202202	노무	348	16	1	0	0	0	365		
7	202202	문화/관광	6	1	0	0	0	0	7		
8	202202	보건/복지	462	43	9	0	0	0	514		
9	202202	비즈니스	14	2	0	0	0	0	16		
10	202202	안전	14	0	1	0	0	0	15		
11	202202	여성/가족	19	0	1	0	0	0	20		
12	202202	전년도 총	1807	195	33	0	0	0	2035		
13	202202	정보통신	11	2	0	0	0	0	13		
14	202202	주거	38	5	0	0	0	0	43		
15	202202	출입국	357	72	2	0	0	0	431		
16	202202	행정	266	34	16	0	0	0	316		
17	202201	교육	153	14	1	0	0	0	168		
18	202201	교통	4	2	0	0	0	0	6		
19	202201	금융	9	0	0	0	0	0	9		
20	202201	기타	136	12	4	0	0	0	152		
	서울글로벌센터 상담실적										

'서울글로벌센터 상담실적.csv',

데이터프레임 자료형 보기

```
1 df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 30 entries, 0 to 29
Data columns (total 9 columns):
 #   Column      Non-Null Count  Dtype  
---  -
 0   년월        30 non-null    int64  
 1   기관        30 non-null    object  
 2   전화        30 non-null    int64  
 3   방문        30 non-null    int64  
 4   이메일      30 non-null    int64  
 5   게시판      30 non-null    int64  
 6   화상상담    30 non-null    int64  
 7   이동상담    30 non-null    int64  
 8   합계        30 non-null    int64  
dtypes: int64(8), object(1)
memory usage: 2.2+ KB
```

Null : 데이터가 비어 있는것(결측값)

Non-Null : 비어 있지 않는 데이터

object : 문자열

Int64 : 정수형

describe()로 통계 출력하기

```
1 df.describe() # 통계 모두 출력
```

	년월	전화	방문	이메일	게시판	화상상담	이동상담	합계
count	30.000000	30.000000	30.000000	30.000000	30.0	30.0	30.0	30.00000
mean	202201.500000	240.466667	26.666667	4.466667	0.0	0.0	0.0	271.60000
std	0.508548	451.645171	52.693868	8.950952	0.0	0.0	0.0	510.48203
min	202201.000000	1.000000	0.000000	0.000000	0.0	0.0	0.0	2.00000
25%	202201.000000	11.500000	1.250000	0.000000	0.0	0.0	0.0	13.50000
50%	202201.500000	30.500000	5.000000	1.000000	0.0	0.0	0.0	36.50000
75%	202202.000000	308.000000	15.750000	3.500000	0.0	0.0	0.0	335.50000
max	202202.000000	1807.000000	205.000000	34.000000	0.0	0.0	0.0	2039.00000

Pandas 열 값을 문자열로 변환

`apply()` 메서드를 사용하여 DataFrame의 열 값 데이터 유형을 문자열로 변환

`astype()` 메서드를 사용하여 DataFrame 열 값의 데이터 유형을 string으로 변환

`applymap()` 메서드를 사용하여 모든 DataFrame 열의 데이터 유형을 string으로 변환

Pandas 열 값을 문자열로 변환

DataFrame의 열 값 데이터 유형을 문자열로 변환 apply()

```
1 import pandas as pd
2 data = pd.DataFrame({
3     'Name': ["채준호", "이사랑", "김진아", "이희진", "김동진"],
4     'Score': [31, 38, 33, 39, 35],
5     'Age': [33, 34, 38, 45, 37]
6 })
7 print("DataFrame before Conversion:")
8 print(data, "\n")
9 print("Datatype of columns before conversion:")
10 print(data.dtypes, "\n")
11
12 data["Age"] = data["Age"].apply(str)
13
14 print("DataFrame after conversion:")
15 print(data, "\n")
16 print("Datatype of columns after conversion:")
17 print(data.dtypes)
```

DataFrame before Conversion:

	Name	Score	Age
0	채준호	31	33
1	이사랑	38	34
2	김진아	33	38
3	이희진	39	45
4	김동진	35	37

Datatype of columns before conversion:

Name	object
Score	int64
Age	int64

dtype: object

DataFrame after conversion:

	Name	Score	Age
0	채준호	31	33
1	이사랑	38	34
2	김진아	33	38
3	이희진	39	45
4	김동진	35	37

Datatype of columns after conversion:

Name	object
Score	int64
Age	object

dtype: object

Pandas 열 값을 문자열로 변환

```
1 import pandas as pd
2 data = pd.DataFrame({
3     'Name': ["채준호", "이사랑", "김진아", "이희진", "김동진"],
4     'Score': [31, 38, 33, 39, 35],
5     'Age': [33, 34, 38, 45, 37]
6 })
7 print("DataFrame before Conversion:")
8 print(data, "\n")
9 print("Datatype of columns before conversion:")
10 print(data.dtypes, "\n")
11
12 data["Score"] = data["Score"].astype(str)
13
14 print("DataFrame after conversion:")
15 print(data, "\n")
16 print("Datatype of columns after conversion:")
17 print(data.dtypes)
```

DataFrame before Conversion:

	Name	Score	Age
0	채준호	31	33
1	이사랑	38	34
2	김진아	33	38
3	이희진	39	45
4	김동진	35	37

Datatype of columns before conversion:

Name	object
Score	int64
Age	int64

dtype: object

DataFrame after conversion:

	Name	Score	Age
0	채준호	31	33
1	이사랑	38	34
2	김진아	33	38
3	이희진	39	45
4	김동진	35	37

Datatype of columns after conversion:

Name	object
Score	object
Age	int64

dtype: object

Pandas 열 값을 문자열로 변환

`applymap()` 메서드를 사용하여 모든 DataFrame 열의 데이터를 string으로 변환

```
1 import pandas as pd
2
3 data = pd.DataFrame({
4     'Name': ["채준호", "이사랑", "김진아", "이희진", "김동진"],
5     'Score': [31, 38, 33, 39, 35],
6     'Age': [33, 34, 38, 45, 37]
7 })
8 print("DataFrame before Conversion:")
9 print(data, "\n")
10 print("Datatype of columns before conversion:")
11 print(data.dtypes, "\n")
12
13 data = data.applymap(str)
14
15 print("DataFrame after conversion:")
16 print(data, "\n")
17 print("Datatype of columns after conversion:")
18 print(data.dtypes)
```

DataFrame before Conversion:

	Name	Score	Age
0	채준호	31	33
1	이사랑	38	34
2	김진아	33	38
3	이희진	39	45
4	김동진	35	37

Datatype of columns before conversion:

Name	object
Score	int64
Age	int64

dtype: object

DataFrame after conversion:

	Name	Score	Age
0	채준호	31	33
1	이사랑	38	34
2	김진아	33	38
3	이희진	39	45
4	김동진	35	37

Datatype of columns after conversion:

Name	object
Score	object
Age	object

dtype: object

년월을 문자열로 변환

```
1 df.info()
```

```
<class 'pandas.core.frame.DataFrame'>  
RangeIndex: 30 entries, 0 to 29  
Data columns (total 9 columns):  
#   Column  Non-Null Count  Dtype  
---  ---  
0   년월    30 non-null     int64  
1   기관    30 non-null     object  
2   전화    30 non-null     int64  
3   방문    30 non-null     int64  
4   이메일  30 non-null     int64  
5   게시판  30 non-null     int64  
6   화상상담 30 non-null     int64  
7   이동상담 30 non-null     int64  
8   합계    30 non-null     int64  
dtypes: int64(8), object(1)  
memory usage: 2.2+ KB
```

년월을 문자열로 변환

```
1 import pandas as pd  
2 df = pd.read_csv('서울글로벌센터 상담실적.csv', encoding = 'cp949')  
3 df["년월"] = df["년월"].apply(str)  
4 df.info()  
5 print('년월 문자열로 변경')  
6 df.describe()
```

```
<class 'pandas.core.frame.DataFrame'>  
RangeIndex: 30 entries, 0 to 29  
Data columns (total 9 columns):  
#   Column  Non-Null Count  Dtype  
---  ---  
0   년월    30 non-null     object  
1   기관    30 non-null     object  
2   전화    30 non-null     int64  
3   방문    30 non-null     int64  
4   이메일  30 non-null     int64  
5   게시판  30 non-null     int64  
6   화상상담 30 non-null     int64  
7   이동상담 30 non-null     int64  
8   합계    30 non-null     int64  
dtypes: int64(7), object(2)  
memory usage: 2.2+ KB  
년월 문자열로 변경
```

[엑셀파일에서 판다스로 값을 읽는 코드 작성하기]

문자열 변경(replace)

파이썬은 문자열 변경을 할 수 있는 replace 함수를 제공

replace(old, new, [count]) -> replace("찾을값", "바꿀값", [바꿀횟수])

```
df = pd.read_excel('lotto.xlsx', index_col = 0)
```

```
a = df['1등당첨금액']
```

```
def to_int(a):
```

```
    a = a.replace('.', '')
```

```
    a = a.replace('원', '')
```

```
    a = int(a)
```

```
    return a
```

엑셀파일에서 판다스로 값을 읽는 코드 작성하기

apply()함수

파이썬은 문자열 변경을 할 수 있는 replace 함수를 제공


apply(함수, axis = 0 or 1)

apply의 첫번째 인자는 적용하고자 하는 함수

axis 는 함수를 열로 적용할지 행으로 적용할지 정해주는 것으로 0은 열, 1은 행으로 적용, 기본값은 0

```
1
2 df = pd.read_excel('lotto.xlsx', index_col = 0)
3
4 a = df['1등 당첨금액']
5
6 def to_int(a):
7     a = a.replace(',', '')
8     a = a.replace('원', '')
9     a = int(a)
10    return a
11
12 a_int = df['1등 당첨금액'].apply(to_int)
13 a_int.head(n=5)
```

회 차	
1008	2,267,377,910원
1007	2,718,786,375원
1006	2,855,602,125원
1005	2,061,199,344원
1004	2,576,251,913원



회 차	
1008	2267377910
1007	2718786375
1006	2855602125
1005	2061199344
1004	2576251913

'전화'데이터만 보기

전화 데이터만 보기

```
1 first_row = df['전화']  
2 first_row
```

```
0    128  
1      4  
2     17  
3    123  
4    348  
5      6  
6    462  
7     14  
8     14  
9     19  
10   1807  
11     11  
12     38  
13    357  
14    266  
15    153  
16      4  
17      9  
18    136  
19    322  
20      5  
21    409  
22      1  
23     13  
24     16  
25   1800  
26      7  
27     23
```

```
28    445
```

```
29    257
```

```
Name: 전화, dtype: int64
```

'전화'데이터에 대한 min()/max()/mean()

```
1 first_row.min() # min() 최소값 구하기
```

```
1
```

```
1 first_row.max() # max() 최대값 구하기
```

```
1807
```

```
1 first_row.mean() # mean() 평균값 구하기
```

```
240.46666666666667
```

연도별로 그룹화하여 평균 구하기

Python pandas의 groupby() 연산자를 사용하여 집단, 그룹별로 데이터를 집계, 요약

groupby('기준이 될 컬럼명')

```
1 print(df.groupby('년월')['전화'].mean())
```

년월별로 전화상담 평균 구하기

```
년월
202201    240.000000
202202    240.933333
```

```
1 print(df.groupby('년월')['전화'].count())
```

년월별로 전화상담 일수구하기

```
년월
202201     15
202202     15
```

```
1 print(df.groupby('년월')['전화'].sum())
```

년월별로 전화상담 총횟수구하기

```
년월
202201    3600
202202    3614
Name: 전화, dtype: int64
```



실습하기

로또 당첨 번호 빈도수 구하기

[실습] 로또 당첨 번호 빈도수

1. 다운받은 Excel(csv) 파일 읽어오기

다운로드 받은 원본파일

```
1 import pandas as pd
2 df = pd.read_excel('excel.xlsx') # 엑셀 파일
3 df.head(n=5)
4
```

	회차 별 추첨 결과	Unnamed: 1	Unnamed: 2	Unnamed: 3	Unnamed: 4	Unnamed: 5	Unnamed: 6	Unnamed: 7	Unnamed: 8	Unnamed: 9	Unnamed: 10	Unnamed: 11	Unnamed: 12	Unnamed: 13
0	년도	회차	추첨일	1등	NaN	2등	NaN	3등	NaN	4등	NaN	5등	NaN	당첨번호
1	NaN	NaN	NaN	당첨자수	당첨금액	당첨자수	당첨금액	당첨자수	당첨금액	당첨자수	당첨금액	당첨자수	당첨금액	
2	2022	1008	2022.03.26	11	2,267,377,910 원	97	42,854,222 원	2826	1,470,935 원	141072	50,000원	2311009	5,000원	
3	NaN	1007	2022.03.19	9	2,718,786,375 원	70	58,259,709 원	2844	1,433,960 원	137134	50,000원	2256573	5,000원	
4	NaN	1006	2022.03.12	9	2,855,602,125 원	65	65,898,511 원	2487	1,722,318 원	131234	50,000원	2288458	5,000원	

[실습] 로또 당첨 번호 빈도수

파일명 변경(excel.xlsx->lotto.xlsx)하고 엑셀 파일 수정 후 파일불러오기

```
1 import pandas as pd
2 df = pd.read_excel('lotto.xlsx')
3 df.head(n=5)
```

	회차	1등당 첨자 수	1등당첨금액	2등당 첨자 수	2등당첨금 액	3등당 첨자수	3등당첨금 액	4등당첨 자수	4등당첨 금액	5등당첨 자수	5등당첨 금액	당첨 번호 1	당첨 번호 2	당첨 번호 3	당첨 번호 4	당첨 번호 5	당첨 번호 6	보너 스 번 호
0	1008	11	2,267,377,910 원	97	42,854,222 원	2826	1,470,935 원	141072	50,000 원	2311009	5,000원	9	11	30	31	41	44	33
1	1007	9	2,718,786,375 원	70	58,259,709 원	2844	1,433,960 원	137134	50,000 원	2256573	5,000원	8	11	16	19	21	25	40
2	1006	9	2,855,602,125 원	65	65,898,511 원	2487	1,722,318 원	131234	50,000 원	2288458	5,000원	8	11	15	16	17	37	36
3	1005	12	2,061,199,344 원	84	49,076,175 원	2798	1,473,338 원	143067	50,000 원	2349017	5,000원	8	13	18	24	27	29	17
4	1004	10	2,576,251,913 원	68	63,143,430 원	2785	1,541,743 원	140047	50,000 원	2340772	5,000원	7	15	30	37	39	44	18

[실습] 로또 당첨 번호 빈도수

정보확인하기

```
: 1 df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1008 entries, 0 to 1007
Data columns (total 18 columns):
 #   Column      Non-Null Count  Dtype  
---  --
 0   회차        1008 non-null   int64  
 1   1등당첨자수 1008 non-null   int64  
 2   1등당첨금액 1008 non-null   object  
 3   2등당첨자수 1008 non-null   int64  
 4   2등당첨금액 1008 non-null   object  
 5   3등당첨자수 1008 non-null   int64  
 6   3등당첨금액 1008 non-null   object  
 7   4등당첨자수 1008 non-null   int64  
 8   4등당첨금액 1008 non-null   object  
 9   5등당첨자수 1008 non-null   int64  
10   5등당첨금액 1008 non-null   object  
11   당첨번호1   1008 non-null   int64  
12   당첨번호2   1008 non-null   int64  
13   당첨번호3   1008 non-null   int64  
14   당첨번호4   1008 non-null   int64  
15   당첨번호5   1008 non-null   int64  
16   당첨번호6   1008 non-null   int64  
17   보너스 번호 1008 non-null   int64  
dtypes: int64(13), object(5)
memory usage: 141.9+ KB
```

Counter 클래스

collections 모듈의 Counter 클래스 사용법

```
1 from collections import Counter
2
3 a_list = ['a', 's', 'd', 's', 'a', 's']
4 Counter(a_list)  # Counter() : 문자열이나, list 의 요소를 카운팅하여 많은 순으로 딕셔너리형태로 리턴한
```

```
Counter({'a': 2, 's': 3, 'd': 1})
```

데이터의 개수가 많은 순으로 정렬된 배열을 리턴하는 most_common이라는 메서드

```
1 from collections import Counter
2
3 a_list = ['a', 's', 'd', 's']
4 Counter(a_list).most_common()
```

```
[('s', 2), ('a', 1), ('d', 1)]
```

```
1 # 몇개 보여줄지 숫자를 넣어 조절
2 Counter(a_list).most_common(3)
3
```

```
[('s', 2), ('a', 1), ('d', 1)]
```

[실습] 로또 당첨 번호 빈도수

로또 당첨번호 빈도수 구하기

```
1 import pandas as pd
2 from collections import Counter
3
4 df = pd.read_excel('lotto.xlsx')
5
6 num_list = list(df['당첨번호1'])
7 num_list += list(df['당첨번호2'].astype(int))      # astype(int) 정수형으로 변경하기
8 num_list += list(df['당첨번호3'].astype(int))
9 num_list += list(df['당첨번호4'].astype(int))
10 num_list += list(df['당첨번호5'].astype(int))
11 num_list += list(df['당첨번호6'].astype(int))
12
13
14 count = Counter(num_list)
15 most_num = count.most_common(45)
16 print(most_num)
17
```

```
[(34, 152), (18, 148), (27, 146), (43, 146), (17, 145), (13, 145), (39, 145), (1, 143), (12, 142), (14, 142), (37, 141), (40, 141), (20, 140), (33, 140), (45, 140), (4, 139), (10, 139), (11, 137), (2, 136), (21, 136), (15, 136), (24, 136), (31, 136), (44, 136), (8, 134), (3, 134), (19, 134), (36, 134), (7, 133), (16, 133), (38, 133), (42, 133), (26, 132), (5, 130), (25, 130), (6, 126), (35, 125), (29, 124), (28, 124), (41, 124), (30, 122), (23, 121), (22, 114), (32, 114), (9, 107)]
```

리스트 연산

리스트는 데이터들을 잘 관리하기 위해서 묶어서 관리할 수 있는 자료형

리스트 만드는 방법(2)

> 대괄호를 이용하는 방법
리스트변수이름 = [요소1, 요소2 ...]

> list()를 이용한 방법
a = list()

리스트 연산_덧셈

```
1 a = [1,2,3,4,5]
2 b = [6,7,8,9,10]
3 list = a+b
4 list
```

[1, 2, 3, 4, 5, 6, 7, 8, 9, 10]

덧셈을 하게되면 리스트가 연결이 되고, 그 연결된 하나의 리스트가 생성

리스트 연산_곱셈

```
1 a = [1,2,3,4,5]
2 list = a*3
3 list
```

[1, 2, 3, 4, 5, 1, 2, 3, 4, 5, 1, 2, 3, 4, 5]

곱셈 연산자 * 를 이용해서 리스트

3을 곱하게 되면 3번만큼 리스트를 반복해서 만들어 줍니다.

0을 곱하게 되면 빈 리스트가 됩니다.

리스트 연산

list.sort() - 리스트 정렬

리스트의 내부 요소를 정렬해주는 함수, 기본적으로는 오름차순으로 정렬((작은것 이 앞으로 오고, 큰 값들이 뒤로 가는 정렬방식)

int 타입과 str 타입은 비교가 안되기 때문에 sort() 함수를 쓸 수 없다.

리스트 정렬

```
1 a = [1,2,3,4,5]
2 list = a*3
3 list.sort()
4 list
```

```
[1, 1, 1, 2, 2, 2, 3, 3, 3, 4, 4, 4, 5, 5, 5]
```

[실습] 로또 당첨 번호 빈도수

로또 당첨번호 빈도수 구하기

```
1 import pandas as pd
2 from collections import Counter
3
4 df = pd.read_excel('lotto.xlsx')
5
6 num_list = list(df['당첨번호1'])
7 num_list += list(df['당첨번호2'].astype(int))
8 num_list += list(df['당첨번호3'].astype(int))
9 num_list += list(df['당첨번호4'].astype(int))
10 num_list += list(df['당첨번호5'].astype(int))
11 num_list += list(df['당첨번호6'].astype(int))
12
13
14 count = Counter(num_list)
15 most_num = count.most_common(45)
16 most_num = sorted(most_num) # 오름차순 정렬하기
17 print(most_num)
18
```

[(1, 143), (2, 136), (3, 134), (4, 139), (5, 130), (6, 126), (7, 133), (8, 134), (9, 107), (10, 139), (11, 137), (12, 142), (13, 145), (14, 142), (15, 136), (16, 133), (17, 145), (18, 148), (19, 134), (20, 140), (21, 136), (22, 114), (23, 121), (24, 136), (25, 130), (26, 132), (27, 146), (28, 124), (29, 124), (30, 122), (31, 136), (32, 114), (33, 140), (34, 152), (35, 125), (36, 134), (37, 141), (38, 133), (39, 145), (40, 141), (41, 124), (42, 133), (43, 146), (44, 136), (45, 140)]