Elixirr

파이썬으로 배우는 따릉이 데이터 분석과 시각화

7회차 수업

데이터 분석 & 시각화 및 고찰

수업 일정

전체 수업은 13회로 구성된다.

- 따릉이 이용현황 파악
- 문제 정의
- 파이썬 및 사용할 라이브러리 소개

- 비주얼 스튜디오 코드 설치
- 따릉이 데이터 수집

- 파이썬 라이브러리
- 따릉이 데이터프레임 만들기

- 따릉이 데이터프레임 관찰하기

- 시간 개념에 따른 데이터 분석을 위한 컬럼 추가

- 장소적 특징에 따른 데이터 분석을 위한 컬럼 추가

- 시간 개념에 따른 데이터 분석 및 시각화-(1)

- 시간 개념에 따른 데이터 분석 및 시각화-(2)

- 장소 특징에 따른 데이터 분석 및 시각화-(1)

- 장소 특징에 따른 데이터 분석 및 시각화-(2)

수업 일정

전체 수업은 13회로 구성된다.

- 시간 개념 X 장소 특징에 따른 데이터 분석 및 시각화

- 주말과 평일에 이용건수가 많은 대여소 데이터 분석 및 시각화

- 문제 정의에 맞춘 해결방안 도출
- 총정리

1. 문제정의
2. 데이터 수집
3. 데이터 가공

4. 데이터 분석

5. 시각화 및 탐색

데이터 분석의 5 단계

데이터 분석 단계에 맞추어 따릉이 데이터 분석을 수행한다.

데이터 분석의 5단계

- 현재 따릉이가 이용되고 있는 현황 조사
- 미래에 개선될 점에 대해서 분석주제 정의
- 파이썬과 판다스 소개

- 환경 설정 비주얼 스튜디오 코드(Visual Studio Code) 설치
- 따릉이 데이터 수집

- 확보한 따릉이 데이터 관찰 및 기초 데이터 분석
- 분석에 필요한 데이터 컬럼 추가 및 가공

- **시간 개념에 따른 데이터 분석**
- 장속적 특징에 따른 데이터 분석
- 두 개념을 동시에 고려한 데이터 분석

- **4단계 데이터 분석을 가장 잘 반영할 수 있는 시각화**
- 문제 정의에 맞춘 해결방안 도출

여기서 배울 내용은 ?

1.문제정의

2.데이터수집

3.데이터 가공

4.데이터 분석

5.시각화 및 탐색

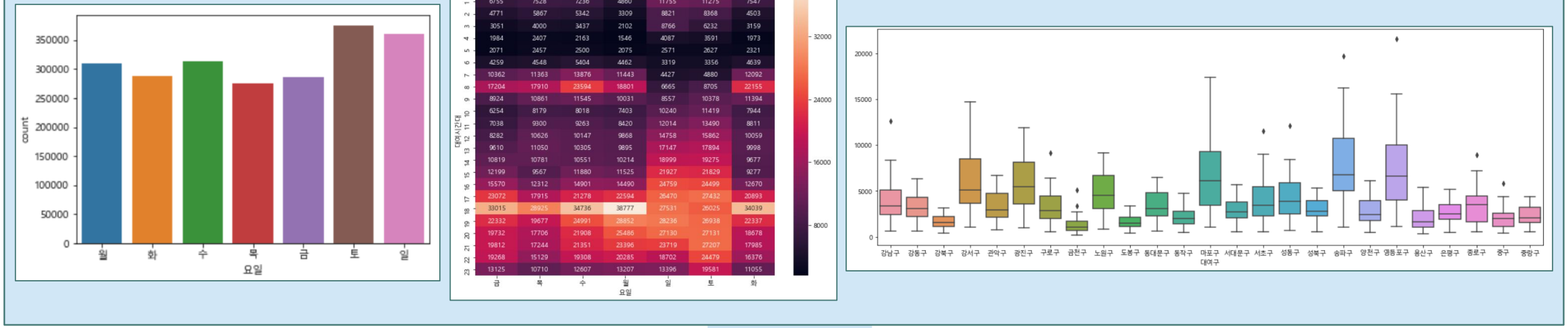
단계 1 : 시간 개념에 따른 따릉이 이용패턴 분석 및 시각화

단계 2 : 장소적 특징에 따른 따릉이 이용패턴 분석 및 지도 시각화

단계 3 : 시간 개념 x 장소적 특징 연관 분석 후 시각화

단계 3 : 주말과 평일에 인기 있는 대여소 상위 50개 지도에 표시해보기

데이터 시각화 라이브러리 : seaborn



sns.countplot() : 지정한 컬럼에 따라 count()라는 집계를 내부적으로 수행한 후, 막대 그래프로 표시해준다.

sns.heatmap() : X축과 Y축에 2개의 범주형 자료가 있을 때, 이들에 해당하는 값을 집계하고 집계한 값에 비례하여 색깔을 다르게 해서 2차원적으로 자료를 시각화 한다.

sns.boxplot() : 정돈된 데이터프레임에서 분석에 필요한 컬럼들을 뽑아내서 박스로 시각화한다.

요일별 자전거 이용건수는 어떻게 구할까?


```
bikes.head(100)
```

✓ 0.7s Python

	자전거 번호	대여일시	대여 대여소 번호	대여 대여소명	이용시간	이용거리	일자	대여시간대	요일	주말구분	대여구	대여점 위도	대여점 경도
0	SPB-22040	2019-06-03 08:49:00	646	장한평역 1번출구 (국민은행앞)	27	1330	3	8	월	평일	동대문구	37.56	127.06
1	SPB-20387	2019-06-05 08:27:00	646	장한평역 1번출구 (국민은행앞)	12	1930	5	8	수	평일	동대문구	37.56	127.06
2	SPB-16794	2019-06-05 08:46:00	646	장한평역 1번출구 (국민은행앞)	6	1340	5	8	수	평일	동대문구	37.56	127.06
3	SPB-13926	2019-06-11 08:29:00	646	장한평역 1번출구 (국민은행앞)	7	1360	11	8	화	평일	동대문구	37.56	127.06
4	SPB-14638	2019-06-12 08:29:00	646	장한평역 1번출구 (국민은행앞)	5	1340	12	8	수	평일	동대문구	37.56	127.06
5	SPB-18588	2019-06-17 08:34:00	646	장한평역 1번출구 (국민은행앞)	8	1360	17	8	월	평일	동대문구	37.56	127.06
6	SPB-21148	2019-06-17 08:47:00	646	장한평역 1번출구 (국민은행앞)	22	1330	17	8	월	평일	동대문구	37.56	127.06
7	SPB-24533	2019-06-18 08:36:00	646	장한평역 1번출구 (국민은행앞)	6	1230	18	8	화	평일	동대문구	37.56	127.06
	SPB-	2019-06-18	646	장한평역 1번출구 (국민	11	1200	18	8	화	평일	동대문	37.56	127.06

1. bikes 데이터프레임에서 '요일' 컬럼을 살펴본다.
2. 예를 들어 bikes['요일']의 값이 '월'인 경우, 해당하는 bikes의 행을 count한다.
3. 옆의 표에서 보면 bikes['월']에 해당하는 행을 count하면 3개, bikes['수']에 해당하는 행을 count하면 3개이고 bikes['화']에 해당하는 행을 count하면 1개이다.

seaborn 시각화 명령어 : sns.countplot() - (1)

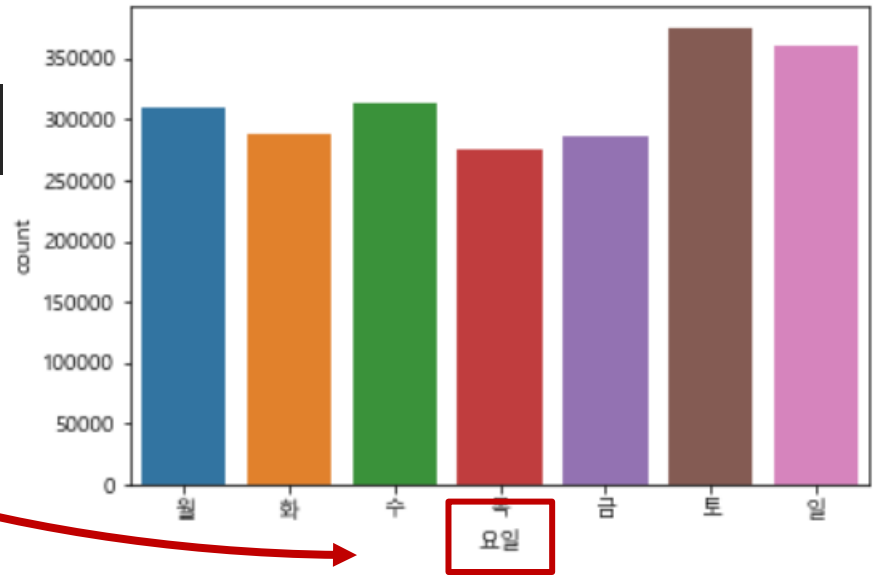
sns.countplot(data=데이터프레임, x='집계할 컬럼명', hue='집계할 컬럼명')

1 2 3 4

1. **sns.countplot() :** 정돈된 데이터프레임에서 분석에 필요한 컬럼을 지정하면, 지정된 컬럼에 따라 내부적으로 데이터를 count해서 막대 그래프로 보여준다.
2. **data=데이터프레임 :** 분석할 데이터프레임의 명칭
3. **x='집계할 컬럼명' :** 집계할 컬럼명으로 그래프의 x축에 표시된다.
4. **hue='집계할 컬럼명' :** 집계할 컬럼명으로 그래프에서 막대의 색깔로 표시된다.

```
sns.countplot(x='요일', data=bikes, order=['월','화','수','목','금','토','일']);
```

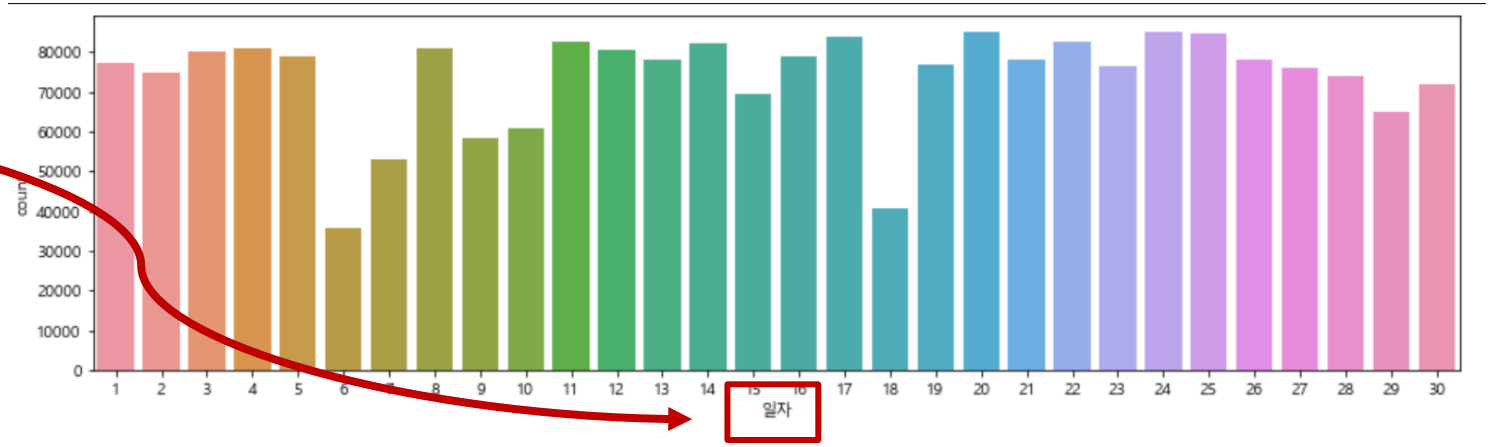
sns.countplot 내부에서 각각의 요일에 해당하는 자전거번호를 카운트해서 그래프의 x축에는 요일을 표시하고, 카운트한 자전거번호 개수를 y축으로 하여 막대그래프로 표현한다.



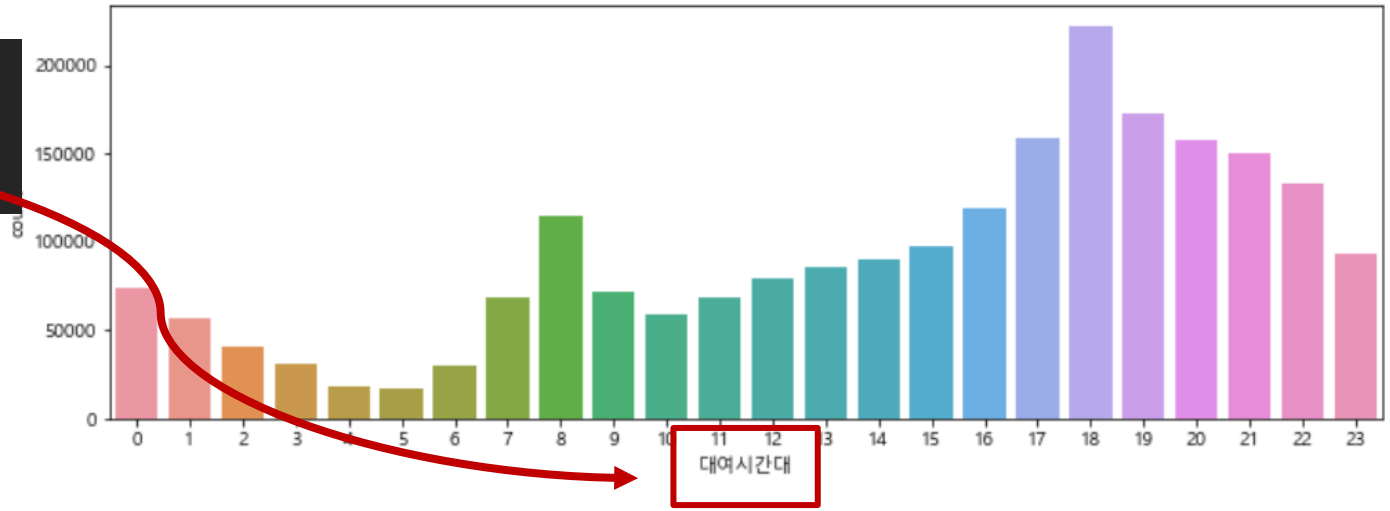
seaborn 시각화 명령어 : sns.countplot() - (2)

sns.countplot(data=데이터프레임, x='분석할 컬럼명', hue='분석할 컬럼명')

```
plt.figure(figsize=(15, 4))
sns.countplot(x='일자', data=bikes);
```




```
plt.figure(figsize=(15, 4))
sns.countplot(x='대여시간대', data=bikes);
```



- plt.figure(figsize=(12, 4)) :
  =>그래프의 사이즈 조절
- figsize=(가로폭, 세로폭)

대여시간대별 x 요일별 자전거 이용건수는?


```
bikes.head(100)
```

✓ 0.7s Python

	자전거 번호	대여일시	대여 대여소 번호	대여 대여소명	이용시간	이용거리	일자	대여시간대	요일	주말구분	대여구	대여점 위도	대여점 경도
0	SPB-22040	2019-06-03 08:49:00	646	장한평역 1번출구 (국민은행앞)	27	1330	3	8	월	평일	동대문구	37.56	127.06
1	SPB-20387	2019-06-05 08:27:00	646	장한평역 1번출구 (국민은행앞)	12	1930	5	8	수	평일	동대문구	37.56	127.06
2	SPB-16794	2019-06-05 08:46:00	646	장한평역 1번출구 (국민은행앞)	6	1340	5	8	수	평일	동대문구	37.56	127.06
3	SPB-13926	2019-06-11 08:29:00	646	장한평역 1번출구 (국민은행앞)	7	1360	11	8	화	평일	동대문구	37.56	127.06
4	SPB-14638	2019-06-12 08:29:00	646	장한평역 1번출구 (국민은행앞)	5	1340	12	8	수	평일	동대문구	37.56	127.06
5	SPB-[illegible]	2019-06-17 08:34:00	646	장한평역 1번출구 (국민은행앞)	8	1360	17	8	월	평일	동대문구	37.56	127.06
6	SPB-[illegible]	2019-06-17 08:47:00	646	장한평역 1번출구 (국민은행앞)	22	1330	17	8	월	평일	동대문구	37.56	127.06
7	SPB-24533	2019-06-18 08:36:00	646	장한평역 1번출구 (국민은행앞)	6	1230	18	8	화	평일	동대문구	37.56	127.06
	SPB-	2019-06-18	646	장한평역 1번출구 (국민	11	1200	18	8	화	평일	동대문	37.56	127.06

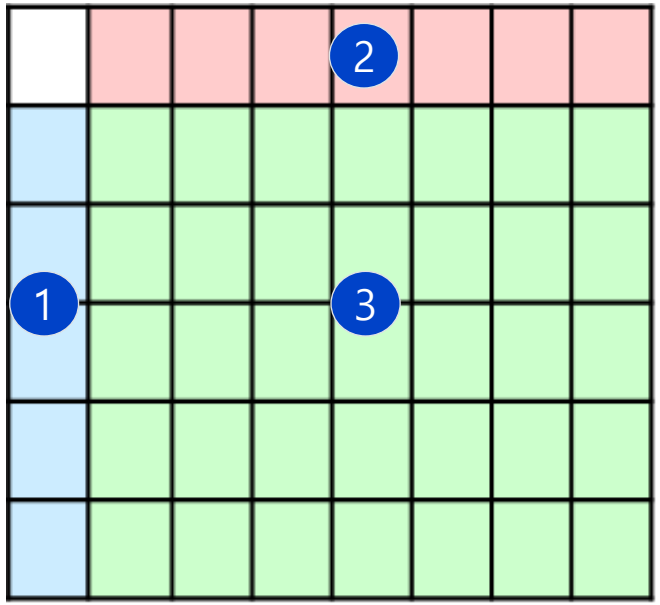
1. bikes 데이터프레임에서 '대여시간대'와 '요일' 컬럼을 살펴본다.
2. 예를 들어 bikes['대여시간대']의 값이 8이고 bikes['요일']이 '월'에 해당하는 bikes['자전거번호']를 count한다.
3. 옆의 표에서 보면 이에 해당하는 자전거번호는 3개이다.

피봇테이블pivot_table - (1)

피봇테이블은 데이터분석을 위해 특정 컬럼을 인덱스와 컬럼으로 재구조화해서 집계함수를 사용하게 한다.

df.pivot_table(index='컬럼명(들)', columns='컬럼명(들)', values='컬럼명(들)', aggfunc='집계함수')

1 2 3 4



1. pivot_table의 인덱스로 정할 컬럼명 또는 복수의 컬럼명들
2. pivot_table의 컬럼으로 정할 컬럼명 또는 복수의 컬럼명들
3. pivot_table의 값으로 정할 컬럼명 또는 복수의 컬럼명들
4. 집계함수 : count(), sum(), mean(), std(), min(), max()

피봇테이블pivot_table (2)

피봇테이블에서 인덱스와 컬럼이 모두 필요한 경우로서 피봇테이블 수행 후 결과는 데이터프레임이다.

bikes.pivot_table(index='대여시간대', columns='요일', values='자전거번호', aggfunc='count')

1 2 3 4

```
bikes.pivot_table(\
                  index = '대여시간대', \
                  columns = '요일', \
                  values = '자전거번호', \
                  aggfunc = 'count'
                 )
```

✓ 0.3s

요일	금	목	수	월	일	토	화
대여시간대							
0	8353	9568	8624	7461	17025	13438	9907
1	6748	7508	7226	4846	11731	11232	7529
2	4762	5857	5328	3305	8804	8348	4497
3	3045	3992	3430	2102	8748	6221	3157

1. pivot_table의 인덱스로 정할 컬럼명 : '대여시간대'
2. pivot_table의 컬럼으로 정할 컬럼명 : '요일'
3. pivot_table의 값으로 정할 컬럼명 : '자전거번호'
4. 집계함수 : count()

변수 할당 : = 연산자

= 연산자를 사용하여 피봇테이블로 생성된 데이터프레임을 새로운 변수에 할당할 수 있다. 분석 주제를 쉽게 표현하는 변수는 이해와 코딩에 편리하다.

새로운 변수 = 생성된 데이터프레임

```
bikes.pivot_table(\
                  index = '대여시간대', \
                  columns = '요일', \
                  values = '자전거번호', \
                  aggfunc = 'count'
                  )
```

✓ 0.3s

요일	금	목	수	월	일	토	화
대여시간대							
0	8353	9568	8624	7461	17025	13438	9907
1	6748	7508	7226	4846	11731	11232	7529
2	4762	5857	5328	3305	8804	8348	4497
3	3045	3992	3430	2102	8748	6221	3157


```
hourly_dayofweek_ride = bikes.pivot_table(\
                        index = '대여시간대', \
                        columns = '요일', \
                        values = '자전거번호', \
                        aggfunc = 'count'
                        )
hourly_dayofweek_ride
```

요일	금	목	수	월	일	토	화
대여시간대							
0	8353	9568	8624	7461	17025	13438	9907
1	6748	7508	7226	4846	11731	11232	7529
2	4762	5857	5328	3305	8804	8348	4497
3	3045	3992	3430	2102	8748	6221	3157

꺾은선 그래프 Line plot

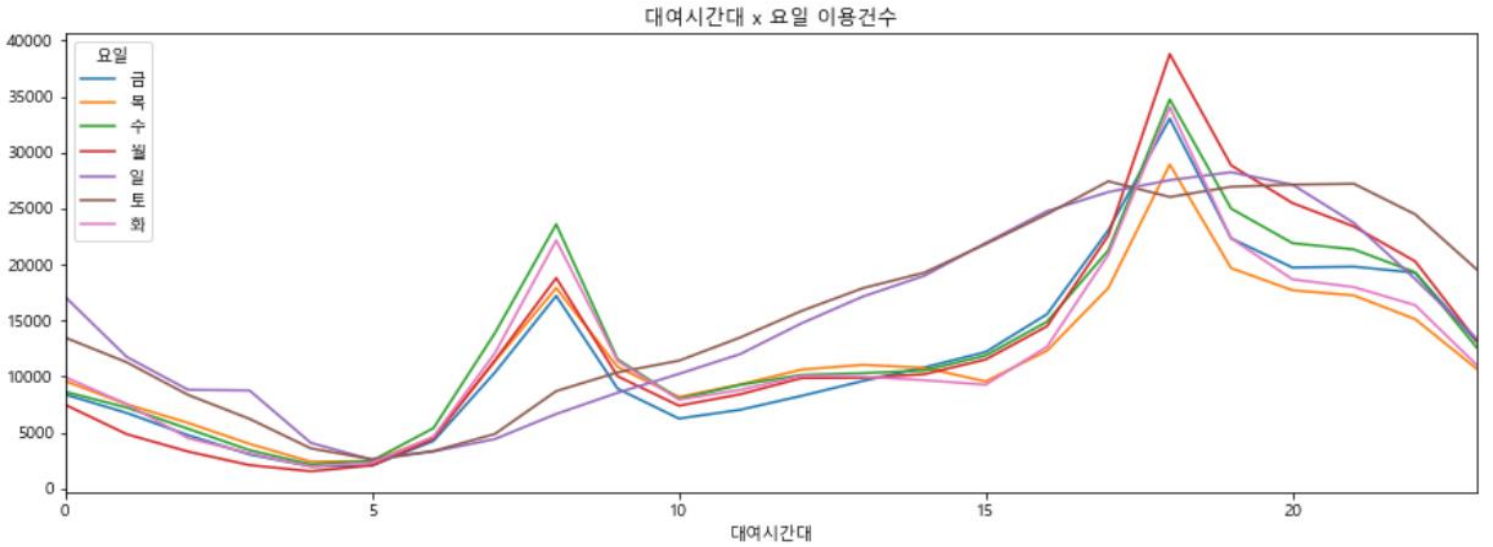
데이터프레임의 컬럼수에 따른 꺾은선의 종류가 있는 라인 그래프를 그릴 수 있다. 색상은 지정하지 않아도 구별되는 색으로 그려준다.

```
hourly_dayofweek_ride = bikes.pivot_table(\
                        index = '대여시간대', \
                        columns = '요일', \
                        values = '자전거번호', \
                        aggfunc = 'count'
                        )
hourly_dayofweek_ride
```

요일	금	목	수	월	일	토	화
대여시간대							
0	8353	9568	8624	7461	17025	13438	9907
1	6748	7508	7226	4846	11731	11232	7529
2	4762	5857	5328	3305	8804	8348	4497
3	3045	3992	3430	2102	8748	6221	3157

컬럼수 7개

```
hourly_dayofweek_ride.plot(kind='line', title = '대여시간대 x 요일 이용건수', figsize=(15, 4));
```



kind = 'line' title = 그래프의 제목 figsize = 그래프의 크기

막대 그래프 Bar plot

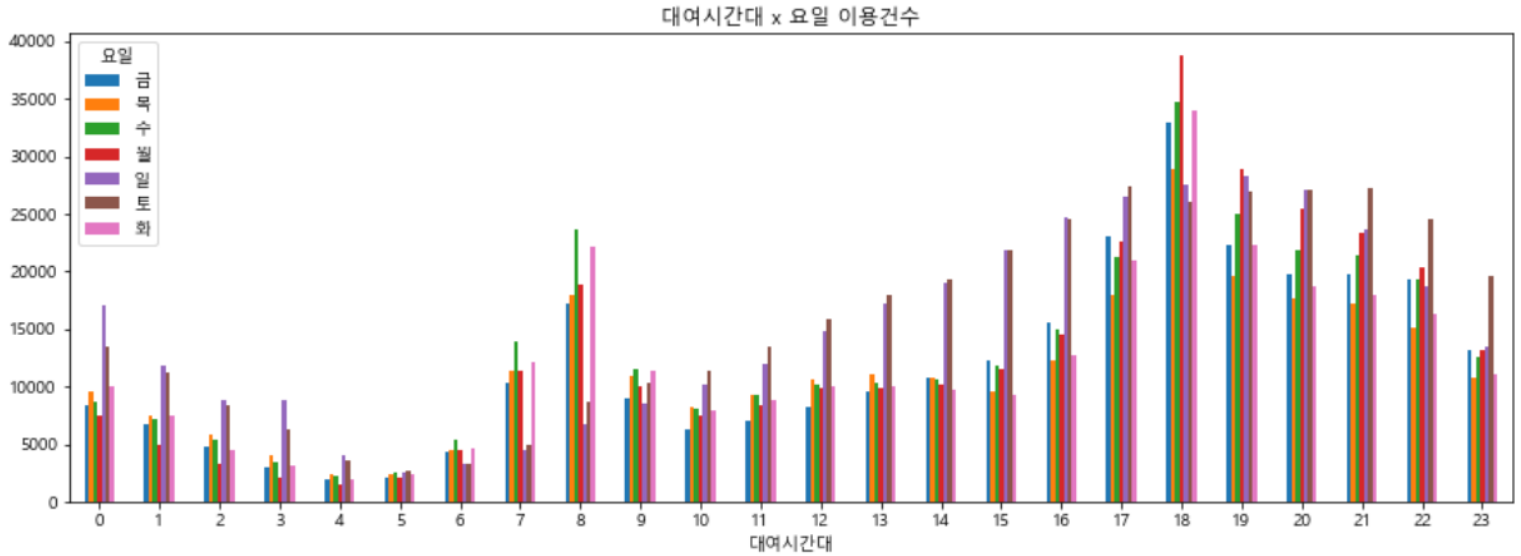
데이터프레임의 컬럼수에 따른 막대의 종류가 있는 막대 그래프로 그릴 수 있다. 색상은 지정하지 않아도 구별되는 색으로 그려준다.

```
hourly_dayofweek_ride = bikes.pivot_table(\
                        index = '대여시간대', \
                        columns = '요일', \
                        values = '자전거번호', \
                        aggfunc = 'count'
                        )
hourly_dayofweek_ride
```

요일	금	목	수	월	일	토	화
대여시간대							
0	8353	9568	8624	7461	17025	13438	9907
1	6748	7508	7226	4846	11731	11232	7529
2	4762	5857	5328	3305	8804	8348	4497
3	3045	3992	3430	2102	8748	6221	3157

컬럼수 7개

```
hourly_dayofweek_ride.plot(kind='bar', title = '대여시간대 x 요일 이용건수', \
                           figsize=(15, 4), rot=0);
```



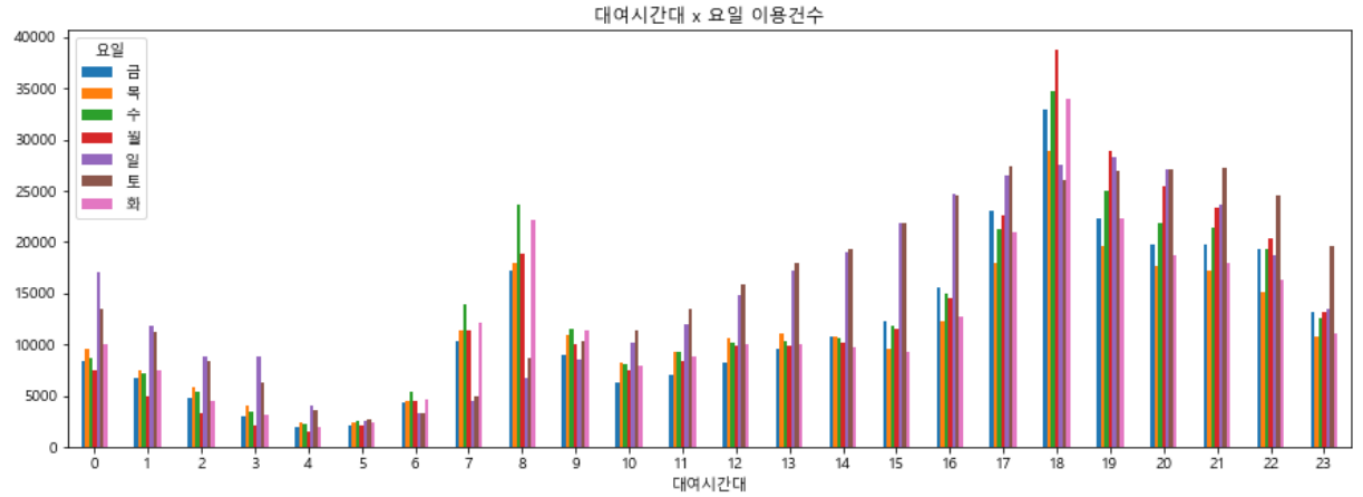
- kind = 'bar' title = 그래프의 제목
- figsize = 그래프의 크기 rot = x축 인덱스 글자 회전

sns.countplot() vs (pivot_table + plot(kind='bar))

sns.countplot(data=데이터프레임, x='분석할 컬럼명', hue='분석할 컬럼명')

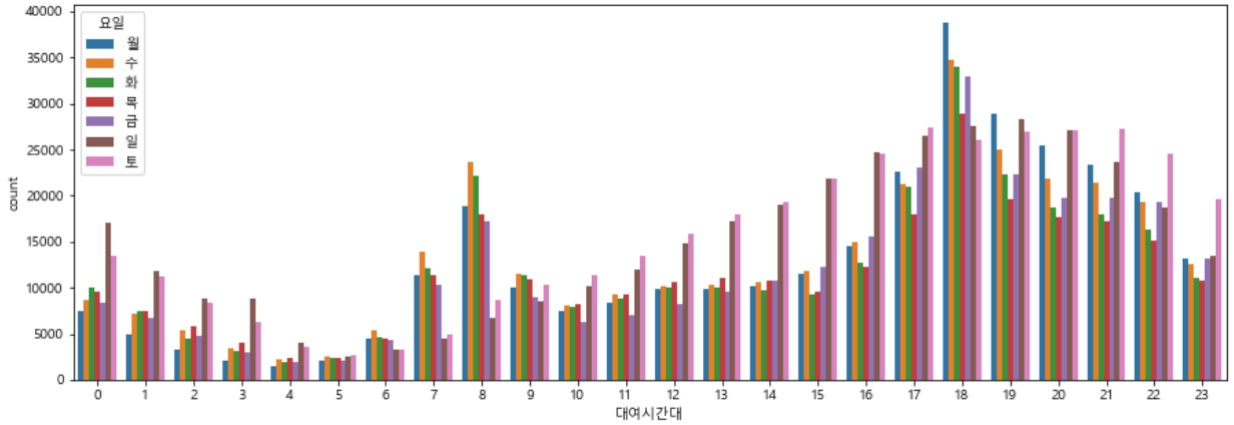
```
hourly_dayofweek_ride = bikes.pivot_table(\
                        index = '대여시간대', \
                        columns = '요일', \
                        values = '자전거번호', \
                        aggfunc = 'count'
                       )
hourly_dayofweek_ride
```

요일	금	목	수	월	일	토	화
대여시간대							
0	8353	9568	8624	7461	17025	13438	9907
1	6748	7508	7226	4846	11731	11232	7529
2	4762	5857	5328	3305	8804	8348	4497
3	3045	3992	3430	2102	8748	6221	3157



VS

```
plt.figure(figsize=(15, 4))
sns.countplot(data=bikes, x='대여시간대', hue='요일');
```



꺾은선 그래프와 막대그래프의 문제점

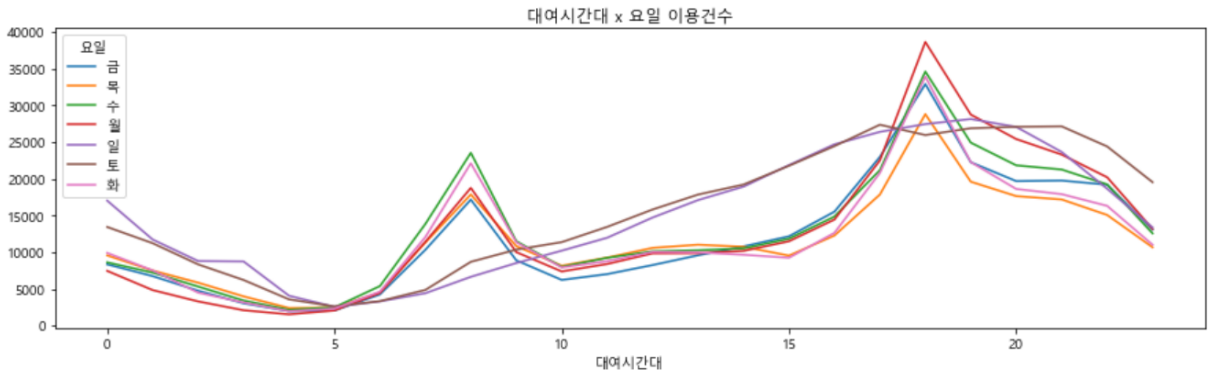
데이터프레임의 컬럼수가 너무 많으면 꺾은선이나 막대의 종류가 많아져서 그래프를 분석하기 어렵다. 다른 시각화 방안을 고려해보자.

```
hourly_dayofweek_ride
```
✓ 0.4s

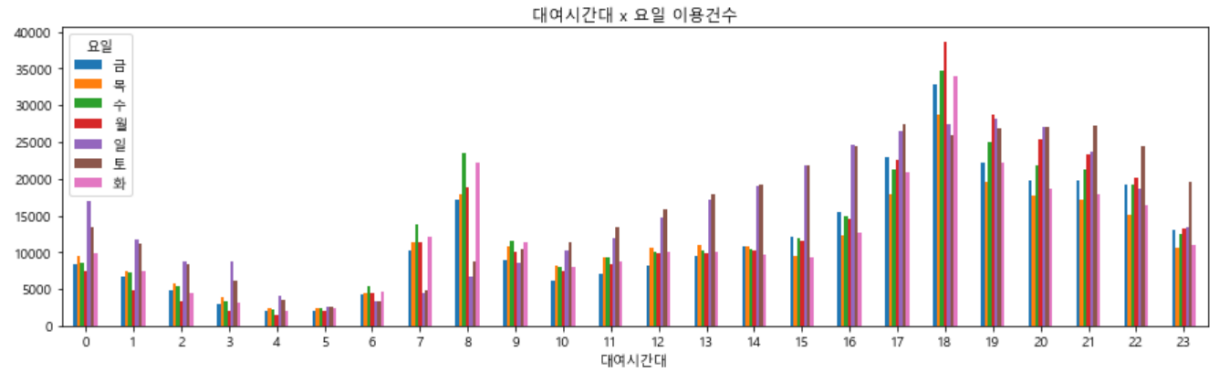
요일	금	목	수	월	일	토	화
대여시간대							
0	8353	9568	8624	7461	17025	13438	9907
1	6748	7508	7226	4846	11731	11232	7529
2	4762	5857	5328	3305	8804	8348	4497
3	3045	3992	3430	2102	8748	6221	3157
4	1979	2402	2160	1544	4076	3584	1971
5	2070	2451	2495	2069	2567	2619	2316
6	4252	4534	5393	4455	3311	3345	4633
7	10337	11324	13853	11418	4420	4869	12070
8	17164	17867	23542	18762	6651	8696	22119
9	8901	10833	11515	10017	8530	10357	11369
10	6232	8164	8006	7384	10215	11377	7924

컬럼수가 많다

```
hourly_dayofweek_ride.plot(kind='line', title = '대여시간대 x 요일 이용건수', figsize=(15, 4));
```
✓ 0.3s



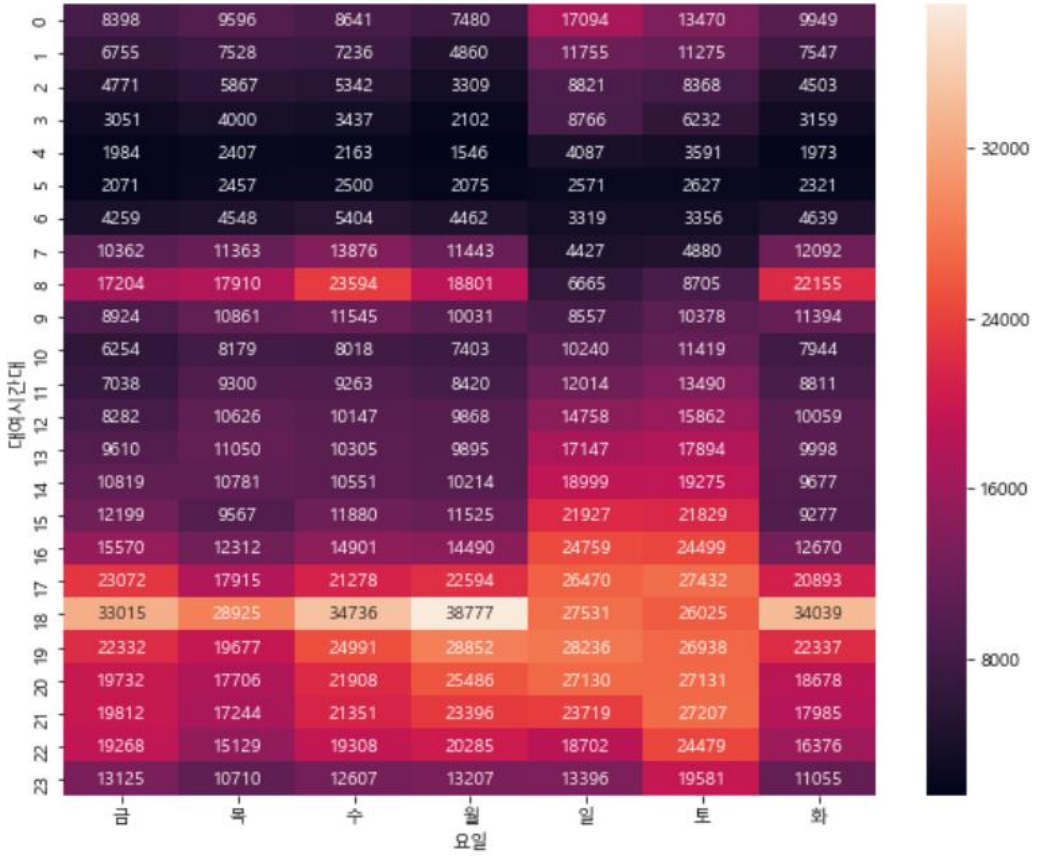

```
hourly_dayofweek_ride.plot(kind='bar', title = '대여시간대 x 요일 이용건수', \
                           figsize=(15, 4), rot=0);
```
✓ 0.6s



데이터 시각화 : 히트맵 Heatmap > sns.heatmap

X축과 Y축에 2개의 범주형 자료가 있을 때, 이들에 해당하는 값을 집계하고 집계한 값에 비례하여 색깔을 다르게 해서 2차원적으로 자료를 시각화 한다.

1 plt.figure(figsize=(10, 8)) 2
sns.heatmap(data=데이터프레임, annot=True, fmt='d');
3 4 5 6




```
plt.figure(figsize=(10, 8))
sns.heatmap(data=hourly_dayofweek_ride, annot=True, fmt='d');
```

1. plt.figure : 그래프 사이즈를 조절할 수 있는 명령어
2. figsize=(가로길이, 세로길이)
3. sns.heatmap : 집계한 수량을 색깔로 표시하는 그래프
4. data=데이터프레임 : 표시할 데이터프레임
5. annot=True : 색깔을 나타내는 칸에 해당 집계 수량도 표시
6. fmt='d' : 정수로 나타낸다는 표시

나 지금 어느 단계를 공부하는 거지?

1.문제정의

2.데이터수집

3.데이터 가공

4.데이터 분석

5.시각화 및 탐색

단계 1 : 다양한 분석 명령어를 사용해서 시간 개념에 따른 따릉이 이용패턴 분석

내부적으로 집계해서 시각화 -> sns.countplot()

특정 컬럼들을 재구조화 -> bike_ride.pivot_table()

꺾은선 그래프, 막대 그래프, 히트맵 시각화

퀴즈

퀴즈를
풀어봅시다

1. 집계를 포함하는 다양한 시각화 명령어를 가지고 있고 high-level interface를 제공하는 라이브러리는?

2. 정돈된 데이터프레임에서 분석에 필요한 컬럼을 지정하면, 지정된 컬럼에 따라 내부적으로 데이터를 count해서 막대 그래프로 보여주는 명령어는 ?

3. 데이터분석을 위해 특정 컬럼을 인덱스와 컬럼으로 재구조화해서 집계함수를사용하는 명령어는 ?

4. 2개의 범주형 자료가 있을 때, 이들에 해당하는 값을 집계하고 집계한 값에 비례하여 색깔을 다르게 해서 2차원적으로 자료를 시각화 하는 명령어는 ?

실습 순서

시간 개념에 따른 데이터 분석

1. 일자별 따릉이 이용건수
2. 요일별 따릉이 이용건수
3. 대여시간대별 따릉이 이용건수
4. 대여시간대 x 요일 따릉이 이용건수
5. 대여시간대 x 주말구분 따릉이 이용건수

실습 시작

GD쌤

이제부터 Visual Studio Code 실습 환경에서 지금까지 배운 내용을 실습해 보겠습니다.

앞에서 배웠던 내용을 Visual Studio Code에서 직접 실습해보면 더욱 이해하기 편리할 것입니다.

수업 마무리

GD쌤

지금까지 7회차 수업내용을 배워 보았습니다.

다음 시간에는 8회차 수업내용으로 시간 개념에 따른 데이터 분석 및 시각화에서 실습 위주로 진행해 보겠습니다. 실습 순서를 기억해 주세요.

수고 많으셨어요. 다음 시간에 만나요.