Elixirr

파이썬으로 배우는 따릉이 데이터 분석과 시각화

5회차
데이터 가공

수업 일정

전체 수업은 13회로 구성된다.

1
- 따릉이 이용현황 파악
- 문제 정의
- 파이썬 및 사용할 라이브러리 소개

2
- 비주얼 스튜디오 코드 설치
- 따릉이 데이터 수집

3
- 파이썬 라이브러리
- 따릉이 데이터프레임 만들기

4
- 따릉이 데이터프레임 관찰하기

5
- 시간 개념에 따른 데이터 분석을 위한 컬럼 추가

6
- 장소적 특징에 따른 데이터 분석을 위한 컬럼 추가

7
- 시간 개념에 따른 데이터 분석 및 시각화-(1)

8
- 시간 개념에 따른 데이터 분석 및 시각화-(2)

9
- 장소 특징에 따른 데이터 분석 및 시각화-(1)

10
- 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
- 인기 대여점 Top50

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

여기서 배울 내용은 ?

1.문제정의

2.데이터수집

3.데이터 가공

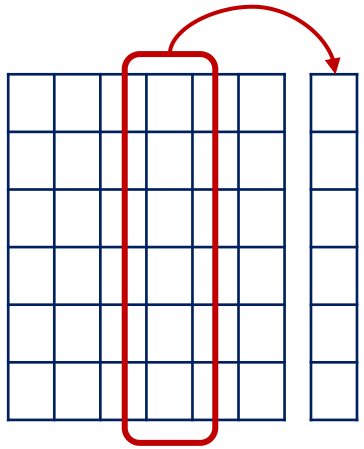
4.데이터 모델링

5.시각화 및 탐색

- 단계 1 : 분석할 데이터프레임 만들기
- 단계 2 : 데이터프레임 관찰하기
- 단계 3 : 분석주제에 맞는 새로운 컬럼 추가하기

열 단위 데이터 추출 : [ ] 인덱스 연산자

필요한 컬럼명을 대괄호 [ ] 인덱스 연산자에 기술하여 선택할 수 있다. 한 컬럼만 입력하면 시리즈, [[ ]] 이 중으로 선택하면 데이터프레임이 된다.



시리즈 : 하나의 값으로만 구성되는 데이터 구조

시리즈

```
bike_series = bikes['대여일시']
bike_series
```

✓ 0.4s

```
0          2019-06-03 08:49:00
1          2019-06-03 08:33:00
2          2019-06-05 08:27:00
3          2019-06-05 08:46:00
4          2019-06-10 08:27:00
                   ...
2215627    2019-06-24 07:40:00
2215628    2019-06-25 07:34:00
2215629    2019-06-26 08:19:00
2215630    2019-06-27 07:38:00
2215631    2019-06-28 07:37:00
Name: 대여일시, Length: 2215632, dtype: datetime64[ns]
```

데이터프레임

```
bike_df = bikes[['대여 대여소번호', '대여 대여소명']]
bike_df
```

✓ 0.1s

	대여 대여소번호	대여 대여소명
0	646	장한평역 1번출구 (국민은행앞)
1	526	용답토속공원 앞
2	646	장한평역 1번출구 (국민은행앞)
3	646	장한평역 1번출구 (국민은행앞)
4	529	장한평역 8번 출구 앞
...	...	...
2215627	240	문래역 4번출구 앞
2215628	240	문래역 4번출구 앞
2215629	240	문래역 4번출구 앞
2215630	240	문래역 4번출구 앞
2215631	240	문래역 4번출구 앞

2215632 rows × 2 columns

- 시리즈와 데이터프레임 모두 판다스에서 자주 사용되는 데이터 구조
- 시리즈는 하나의 값으로 구성 되므로 별도로 컬럼이 없음

시간 정보 추가 : 데이터프레임에 시간 컬럼 추가

시간개념에 따른 데이터 분석을 위해 '일자', '요일', '시간대', '주말/평일 여부' 등 필요한 컬럼을 추출한다.

```
bikes.head()
```

	자전거번호	대여일시	대여 대여소번호	대여 대여소명	이용시간	이용거리
0	SPB-22040	2019-06-03 08:49:00	646	장한평역 1번출구 (국민은행앞)	27	1330
1	SPB-07446	2019-06-03 08:33:00	526	용답토속공원 앞	54	1180
2	SPB-20387	2019-06-05 08:27:00	646	장한평역 1번출구 (국민은행앞)	12	1930
3	SPB-16794	2019-06-05 08:46:00	646	장한평역 1번출구 (국민은행앞)	6	1340
4	SPB-18266	2019-06-10 08:27:00	529	장한평역 8번 출구 앞	5	1230

'대여일시'는 데이터타입이 **datetime64**이다. 판다스 라이브러리에 있는 dt 액세서를 사용하여 날짜 데이터타입인 bikes['대여일시']에서 **'일자', '요일' , '주말여부', '시간대'** 를 추출할 수 있다. 이 값들을 이용하여 새로운 컬럼을 만든다.

시간 정보 추가 : 파이썬 문법 > 리스트list

정의 : 여러 데이터들을 잘 관리하기 위해서 묶어서 목록으로 관리할 수 있는 파이썬 자료구조

만들기 : 리스트의 이름 = [항목1, 항목2, ...] 예) 요일 = ['월', '화', '수', '목', '금', '토', '일']

인덱싱 : 리스트에 있는 여러 항목들은 모두 각각 그 위치가 0부터 시작하는 숫자로 매겨져 있다.

예) 요일 = ['월', '화', '수', '목', '금', '토', '일']
인덱스 -> 0 1 2 3 4 5 6

리스트내의 항목 추출하기 : 리스트의 이름[항목의 인덱스]

예) 요일 = ['월', '화', '수', '목', '금', '토', '일']
인덱스 -> 0 1 2 3 4 5 6

요일[0] = '월', 요일[1] = '화', 요일[2] = '수', 요일[3] = '목', 요일[4] = '금', 요일[5] = '토'

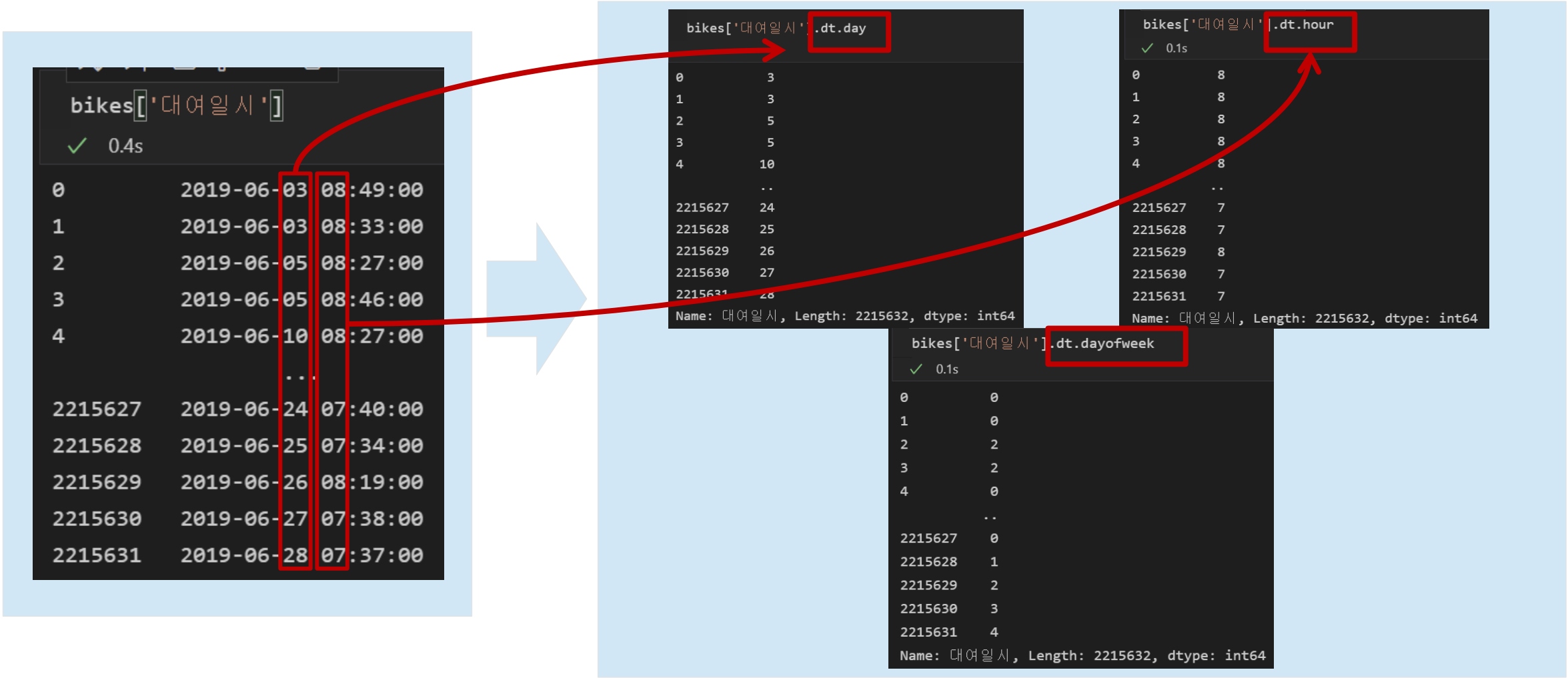
시간 정보 추가 : dt 액세서dt accessor

판다스에서 컬럼의 데이터타입이 날짜 데이터인 경우, 날짜 데이터에서 년도, 월, 일, 시간, 분, 초, 요일 등의 정보를 추출하는 방식이다.

bikes['대여일시'].**dt.year**	• bike_ride['대여일시']에서 **년도 추출**
bikes['대여일시'].**dt.month**	• bike_ride['대여일시']에서 **월 추출**
bikes['대여일시'].**dt.day**	• bike_ride['대여일시']에서 **일 추출**
bikes['대여일시'].**dt.hour**	• bike_ride['대여일시']에서 **시간 추출**
bikes['대여일시'].**dt.dayofweek**	• bike_ride['대여일시']에서 **요일 추출** • 월 : 0 화 : 1 수 : 2 목 : 3 금 : 4 토 : 5 일 : 6

시간 정보 추가 : dt 액세서dt accessor

데이터프레임의 각각의 컬럼 데이터타입이 날짜 데이터인 경우, 컬럼 값들에서 년도, 월, 일, 시간, 분, 초, 요일 등 시간단위를 추출할 수 있다.



시간 정보 추가 : 컬럼에 변환 적용하기 > df.apply ( )

데이터프레임의 컬럼 , 시리즈 또는 데이터프레임 전체에 대해 함수를 적용하게 해주는 명령어이다.

df.apply(함수)

컬럼이나 데이터프레임에 적용할 함수

```
bikes['대여일시'].dt.dayofweek.apply(lambda x: 요일[x])
✓ 0.3s
0          월
1          월
2          수
3          수
4          월
          ..
2215627    월
2215628    화
2215629    수
2215630    목
2215631    금
Name: 대여일시, Length: 2215632, dtype: object
```

```
bikes['대여일시'].dt.dayofweek.apply(lambda x : '평일' if x < 5 else '주말')
0          평일
1          평일
2          평일
3          평일
4          평일
          ..
2215627    평일
2215628    평일
2215629    평일
2215630    평일
2215631    평일
Name: 대여일시, Length: 2215632, dtype: object
```

파이썬 문법 > lambda 함수(1)

함수(Function) : 입력값을 받아서 어떤 일을 수행한 뒤 그 결과값을 돌려 주는 구문들의 모음

함수를 사용하는 이유 : 동일한 일을 반복적으로 수행해야 할 때 구문들을 매번 작성할 필요가 없다.

람다lambda 함수 : 함수를 재사용하지 않고 즉시 실행이 필용한 경우 익명 함수일회성 함수를 사용한다.
특히 판다스의 데이터프레임에 새로운 컬럼을 추가하거나 특정 컬럼값을 변형시킬 때 apply 명령어와 결합하여 사용하면 유용하다.

람다lambda 함수의 표현 : lambda 입력값 : 결과값

예) bikes['이용시간'].apply(lambda x : x / 60)

bikes['이용시간'] 컬럼값들이 처음부터 차례로 x에 입력된다.

입력된 이용시간이 60으로 나누어진 값들이 출력된다.

시간 정보 추가 : 파이썬 문법 > lambda 함수(2)

람다lambda 함수의 표현 : lambda 입력값 : 결과값

예) bikes['이용시간'].apply(lambda x : x / 60)

bikes['이용시간'] 컬럼값들이 처음부터 차례로 x에 입력된다.

입력된 이용시간이 60으로 나누어진 값들이 출력된다.

```
bikes['이용시간']
✓ 0.4s
0          27
1          54
2          12
3           6
4           5
          ..
2215627    13
2215628     6
2215629     7
2215630    11
2215631     6
Name: 이용시간, Length: 2215632, dtype: int64
```

```
bikes['이용시간'].apply(lambda x : x / 60)
✓ 0.3s
0          0.45
1          0.90
2          0.20
3          0.10
4          0.08
          ...
2215627    0.22
2215628    0.10
2215629    0.12
2215630    0.18
2215631    0.10
Name: 이용시간, Length: 2215632, dtype: float64
```

시간 정보 추가 : 파이썬 문법 > lambda 함수(3)

람다lambda 함수의 표현 : Lambda 입력값 : 결과값

예) bikes['대여일시'].dt.dayofweek.apply(lambda x : 요일[x])

bikes['대여일시'].dt.dayofweek 값이 차례로 x에 입력된다.

리스트 요일[x] 값들이 차례로 출력된다.

```
bikes['대여일시'].dt.dayofweek

0          0
1          0
2          2
3          2
4          0
          ..
2215627    0
2215628    1
2215629    2
2215630    3
2215631    4
Name: 대여일시, Length: 2215632, dtype: int64
```

```
요일 = ['월','화','수','목','금','토','일']

bikes['요일'] = bikes['대여일시'].dt.dayofweek.apply(lambda x: 요일[x])
bikes['요일']
✓ 0.3s

0          월
1          월
2          수
3          수
4          월
          ..
2215627    월
2215628    화
2215629    수
2215630    목
2215631    금
Name: 요일, Length: 2215632, dtype: object
```

시간 정보 추가 : 데이터프레임에 새로운 열 추가

[ ] 연산자를 사용하여 데이터프레임에 새로운 컬럼을 추가할 수 있다.

데이터프레임['새로운 열이름'] = 값 / 시리즈

```
요일 = ['월','화','수','목','금','토','일']
bikes['요일'] = bikes['대여일시'].dt.dayofweek.apply(lambda x: 요일[x])
bikes['요일']
```

✓ 0.3s

```
0    월
1    월
2    수
3    수
4    월
```

```
bikes.head()
```

	자전거번호	대여일시	대여 대여소번호	대여 대여소명	이용시간	이용거리	일자	대여시간대	요일
0	SPB-22040	2019-06-03 08:49:00	646	장한평역 1번출구 (국민은행앞)	27	1330	3	8	월
1	SPB-07446	2019-06-03 08:33:00	526	용답토속공원 앞	54	1180	3	8	월
2	SPB-20387	2019-06-05 08:27:00	646	장한평역 1번출구 (국민은행앞)	12	1930	5	8	수
3	SPB-16794	2019-06-05 08:46:00	646	장한평역 1번출구 (국민은행앞)	6	1340	5	8	수
4	SPB-18266	2019-06-10 08:27:00	529	장한평역 8번 출구 앞	5	1230	10	8	월

나 지금 어느 단계를 공부하는 거지?

1.문제정의

2.데이터수집

3.데이터 가공

4.데이터 모델링

5.시각화 및 탐색

단계 3 : 분석주제에 맞는 새로운 컬럼 추가하기

데이터프레임에 시간관련 컬럼 추가

dt 액세서

df.apply()

lambda 함수

퀴즈

퀴즈를
풀어봅시다

1. bikes['대여일시']에서 요일을 추출할 수 있는 명령어 구문을 쓰세요.

2. 여러 데이터들을 잘 관리하기 위해서 묶어서 목록으로 관리할 수 있는 파이썬 자료구조는 ?

3. 데이터프레임의 컬럼 , 시리즈 또는 데이터프레임 전체에 대해 함수를 적용하게 해주는 명령어는 ?

4. 판다스의 데이터프레임에 새로운 컬럼을 추가하거나 특정 컬럼값을 변형시킬 때 apply 명령어와 결합하여 사용하면 유용한 것은 ?

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 5회차 수업내용을 배워 보았습니다.

다음 시간에는 6회차 수업내용으로 장소적 특징에 따른 데이터 분석을 위한 컬럼들을 추가해 보겠습니다.

수고 많으셨어요. 다음 시간에 만나요.