

IT파이썬. 딥러닝

파이썬 모델 학습

강사 이수현

롤러코스터를 파헤치다

- 롤러코스터 데이터를 특정 속성을 기준으로 정렬 및 추출하고, 속성 간의 관계를 수치로 파악하는 상관관계 분석을 통해 새로운 정보를 탐색하는 방법을 알아본다.

문제 정의하기
클러스터 정보 탐색

데이터 불러오기
클러스터 데이터 셋

데이터 탐색 및 시각화하기

- 데이터 정리 후 추출
- 기초 그래프 시각화
- 상관관계 시각화

문제 상황 이해하기

틀러코스터를 탔을 때

짜릿함을 느끼는 가장 큰 요인은 무엇일까요?

탐색할 정보 알아보기

데이터 분석 과정에서 탐색할 정보를 미리 살펴봅시다.

탐색 정보 가장 빠른 롤러코스터 TOP 10의 테마파크 이름 (theme), 롤러코스터 유형(rollercoaster_type), 흥미도 (excitement), 최고 속도(max_speed)는 어떻게 알 수 있을까요?

롤러코스터 데이터에서 최고 속도(max_speed) 속성을 기준으로 데이터를 정렬한 후 상위 10개 데이터에서 필요한 속성을 추출하여 출력한 결과에서 확인합니다.

탐색할 정보 알아보기

데이터 분석 과정에서 탐색할 정보를 미리 살펴봅시다.

탐색 정보 2 롤러코스터 유형(rollercoaster_type) 속성에는 몇 가지 유형이 있으며 어떤 유형이 가장 많을까요?

롤러코스터 유형(rollercoaster_type) 속성의 고유한 값을 추출한 후 개수를 확인합니다. 그 후 롤러코스터 유형별 개수를 막대그래프로 시각화해보면 개수를 비교할 수 있습니다.

탐색할 정보 알아보기

데이터 분석 과정에서 탐색할 정보를 미리 살펴봅시다.

탐색 정보: 최고 속도(max_speed), 탑승 시간(ride_time), 롤러코스터 길이(ride_length), 최고 낙하 높이(highest_drop_height) 속성 중 흥미도(excitement) 속성과 가장 관련이 깊은 속성은 무엇일까요?

여러 가지 속성 간 상관관계를 분석하여 상관계수 값이 가장 큰 속성을 확인합니다.

테이터셋

- 놀이공원에 가면 롤러코스터를 타야 놀이공원에 다녀왔다고 말할 수 있을 만큼 롤러코스터는 놀이공원을 대표하는 놀이기구
- 가장 빠른 롤러코스터는 고속도로 제한 속도보다도 빠르며, 가장 높은 롤러코스터는 초고층 건물 높이

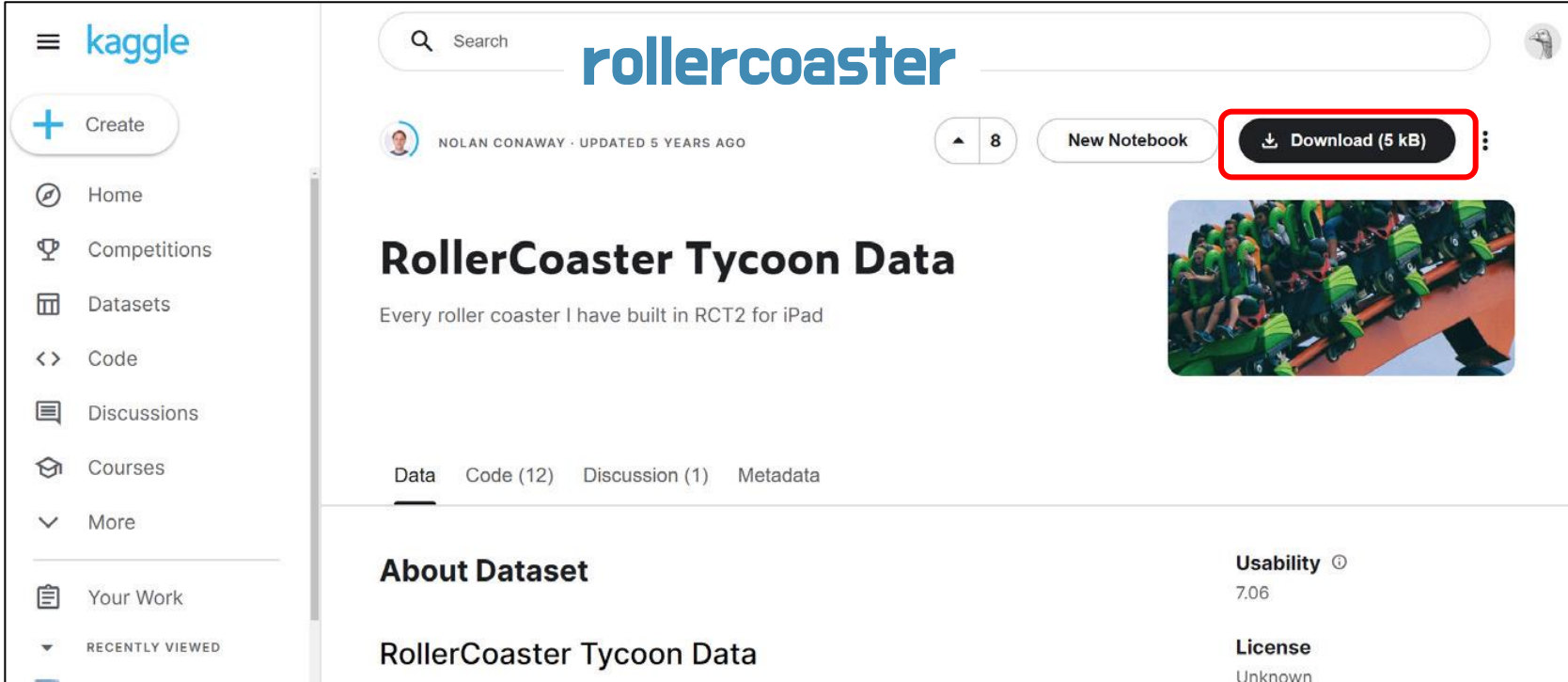
롤러코스터는 무섭기도 하지만 짜릿함에 매료되어 반복해서 타게 되는 매력적인 놀이기구라 할 수 있다.

데이터셋

- 이번 활동에서 사용할 데이터는 놀이공원에 롤러코스터를 설계하여 직접 놀이공원을 운영하는 게임에서 수집한 142개의 롤러코스터 데이터이다.
- 데이터에는 테마파크 이름, 롤러코스터 유형, 흥미도, 최고 속도, 탑승 시간 등 다양한 속성이 있다.

롤러코스터 데이터 내려받기

개글 검색창에 'rollercoaster'를 검색하여 아래 데이터를
다운로드한 후 압축을 푼다.



The screenshot shows the Kaggle interface for the 'RollerCoaster Tycoon Data' dataset. The search bar at the top contains the text 'rollercoaster'. The dataset is listed by Nolan Conway, updated 5 years ago. A red box highlights the 'Download (5 kB)' button. The page also shows a 'New Notebook' button, a 'Data' tab, and a 'Usability' score of 7.06.

<https://www.kaggle.com/datasets/nolanbconaway/rollercoaster-tycoon-rides>

**압축을 풀고 난 후 생성된 'rollercoasters.csv' 파일을
클릭하여 내용을 확인한다.**

1	park_id	theme	rollercoaster_type	custom_design	excitement	excitement_rating	intensity	intensity_ratin	nausea	nausea_rating			
2	0	Barony Bridge	Bobsleigh Coaster	0	5.49	High	5.71	High	4.86	Medium			
3	0	Barony Bridge	Dinghy Slide	0	4.83	Medium	5.86	High	3.5	Medium			
4	0	Barony Bridge	Wild Mouse	0	7.64	High	7.86	Very High	4.87	Medium			
5	0	Barony Bridge	Wooden Roller Coaste	0	7.69	High	7.92	Very High	4.75	Medium			
6	1	Forest Frontiers	Junior Roller Coaster	0	5.17	High	5.54	High	3.81	Medium			
7	2	Haunte	max_speed	avg_speed	ride_time	ride_length	max_pos_gs	max_neg_gs	max_lateral_gs	total_air_time	drops	highest_drop_height	inversions
8	2	Haunte	39	12	63	1496	2.59	-0.27	1.71	0	2	19	0
9	2	Haunte	37	11	48	1079	2.83	-0.48	2.68	0.36	4	42	0
			31	11	70	1591	3.13	-1.7	2.15	1.44	11	22	0
			47	15	79	2401	3.31	-1.15	1.71	5.73	9	62	-1
			34	13	51	1279	2.38	-0.26	2.45	0.72	4	39	0
			32	10	75	1466	2.36	-0.23	2.08	1.14	8	19	0
			45	17	63	2077	3.18	-1.28	1.83	3.21	5	62	-1
			30	12	53	1246	3.08	-1	1.96	2.16	9	22	0
			41	14	40	1255	4.02	-0.99	1.29	5.9	4	5	1

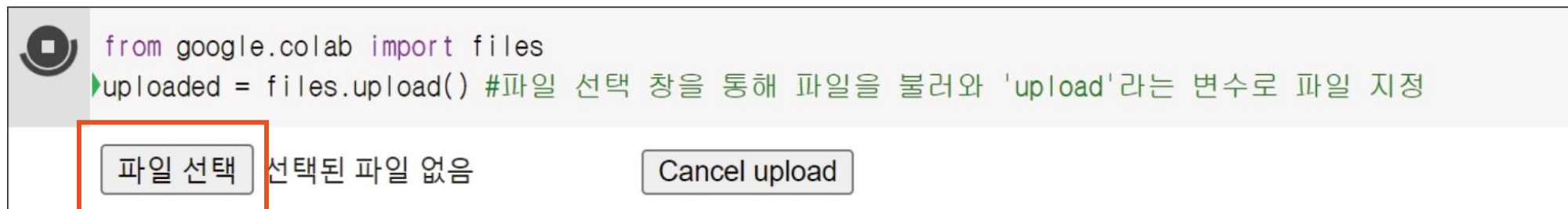
데이터셋 불러오기

숫자와 문자열이 섞인 콜러코스터 데이터를 다루기 위한
판다스(pandas) 라이브러리와 처리한 데이터를 시각화
하기 위한 맷플롯립(matplotlib) 라이브러리 사용

```
1 import pandas as pd  
2 import matplotlib.pyplot as plt
```

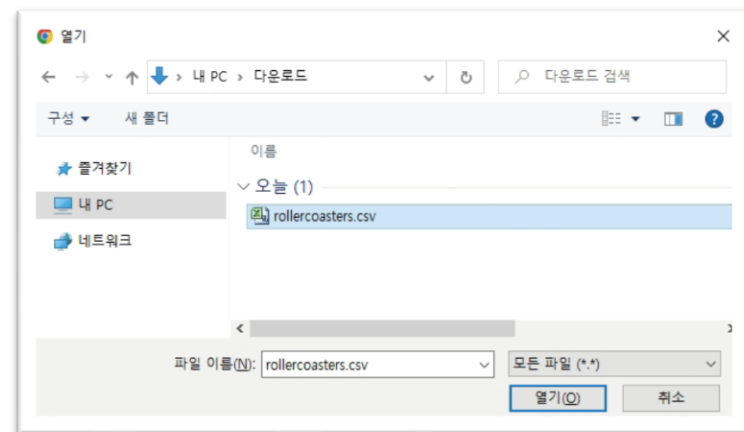
파일 업로드하기

- google.colab 라이브러리 files 모듈을 불러와 업로드하기
- upload() 함수를 사용하기



파일 업로드하기

① 파일 선택 버튼을 클릭하여 파일을 업로드한다.

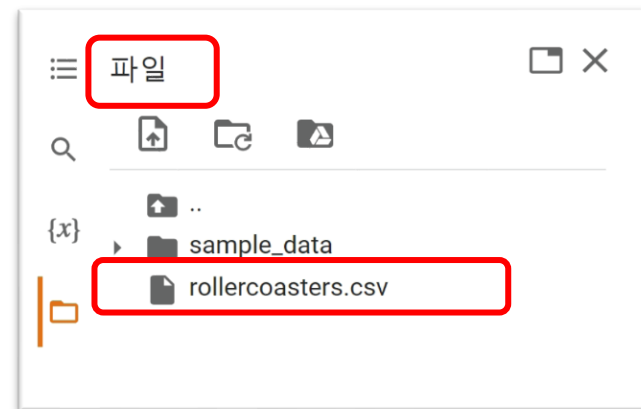


② 업로드 완료 후 안내 메시지 출력을 확인한다.

파일 선택 rollercoasters.csv

- **rollercoasters.csv**(text/csv) - 16604 bytes, last modified: 2019. 9. 20. - 100% done
Saving rollercoasters.csv to rollercoasters.csv

③ 업로드된 파일이 화면 왼쪽 '파일'에 저장된 것을
확인한다.



판다스 라이브러리로 파일 읽어 들이기

- 판다스 라이브러리의 `read_csv()` 함수를 사용하여 코랩 노트북으로 파일을 읽어 들이기

데이터프레임 객체 = 판다스 객체.`read_csv('파일명')`

판다스 라이브러리의 `read_csv()` 함수를 사용하여 파일을 코랩 노트북으로 읽어 들임.

데이터프레임으로 파일을 불러옴.

판다스
라이브러리로
파일 읽어
들이기

- 읽어 들인 파일을 데이터프레임 형태로 출력하기

```
1 df = pd.read_csv('rollercoasters.csv')  
2 df
```


데이터 불러오기



	park_id	theme	rollercoaster_type	custom_design	excitement	excitement_rating	intensity	intensity_rating	nausea	nausea_rating	...			
0	0	Barony Bridge	Bobsleigh Coaster	0	5.49	High	5.71	High	4.86	Medium	...			
1	0	Barony Bridge	Dinghy Slide	0	4.83	Medium	5.86	High	3.50	Medium	...			
2	0	Barony Bridge	Wild Mouse	0	7.64	High	7.86	Very High	4.87	Medium	...			
3	0	Barony Bridge	Wooden Roller Coaster	...	avg_speed	ride_time	ride_length	max_pos_gs	max_neg_gs	max_lateral_gs	total_air_time	drops	highest_drop_height	inversions
4	1	Forest Frontiers	Junior Roller Coaster	...	12	63	1496	2.59	-0.27	1.71	0.00	2	19	0
...	11	48	1079	2.83	-0.48	2.68	0.36	4	42	0
137	30	Botany Breakers	Spiral Coaster	...	11	70	1591	3.13	-1.70	2.15	1.44	11	22	0
138	30	Botany Breakers	Twister Coaster	...	15	79	2401	3.31	-1.15	1.71	5.73	9	62	-1
139	30	Botany Breakers	Mine Train Coaster	...	13	51	1279	2.38	-0.26	2.45	0.72	4	39	0
140	30	Botany Breakers	Wooden Roller Coaster
141	30	Botany Breakers	Hypercoaster	...	13	46	1174	2.69	-0.55	2.96	0.72	3	49	0
142	21	18	61	2221	3.86	-1.08	2.29	2.10	6	22	2
...	16	90	2972	3.33	-1.36	2.17	2.43	4	82	0
...	18	58	2001	3.01	-0.97	1.97	1.62	5	29	1
...	15	61	1791	2.69	-0.45	2.71	0.18	2	62	0

테이터프레임의 행이나 열을 모두 확인해 보고 싶다면?

모든 열을 출력할 때: `pd.set_option('display.max_columns', None)`

모든 행을 출력할 때: `pd.set_option('display.max_rows', None)`

10개 열만 출력할 때: `pd.set_option('display.max_columns', 10)`

10개 행만 출력할 때: `pd.set_option('display.max_rows', 10)`

데이터 탐색 및 시각화하기



**데이터
실패보기**

문제 해결 전에 데이터를 해결 가능한 형태로 처리한다고 해서 전처리라고 부른다.

**데이터 기초
정보 확인하기**

판다스 라이브러리의 **info()** 메소드를 사용하여
데이터의 기초 정보를 확인하기

데이터프레임 객체.info()

info() 메소드를 통해 데이터 개수, 속성 개수, 속성명, 결측치, 속성의 데이터 유형 등 확인

데이터 탐색 및 시각화하기



데이터 기초 정보 확인하기

df.info()

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 142 entries, 0 to 141
Data columns (total 21 columns):
#   Column                Non-Null Count  Dtype
---  -
0   park_id               142 non-null    int64
1   theme                 142 non-null    object
2   rollercoaster_type    142 non-null    object
3   custom_design         142 non-null    int64
4   excitement            142 non-null    float64
5   excitement_rating     142 non-null    object
6   intensity             142 non-null    float64
7   intensity_rating      142 non-null    object
8   nausea               142 non-null    float64
9   nausea_rating        142 non-null    object
10  max_speed             142 non-null    int64
11  avg_speed             142 non-null    int64
12  ride_time             142 non-null    int64
13  ride_length           142 non-null    int64
14  max_pos_gs            142 non-null    float64
15  max_neg_gs            142 non-null    float64
16  max_lateral_gs        142 non-null    float64
17  total_air_time        142 non-null    float64
18  drops                 142 non-null    int64
19  highest_drop_height   142 non-null    int64
20  inversions            142 non-null    int64
dtypes: float64(7), int64(9), object(5)
memory usage: 23.4+ KB
```

데이터 탐색 및 시각화하기



데이터 기초 정보 확인하기

속성명	설명
park_id	테마파크 식별자
theme	테마파크 이름
rollercoaster_type	롤러코스터 유형
custom_design	게임 유저가 직접 롤러코스터를 설계했는지 여부(참/거짓)
excitement	흥미도
excitement_rating	흥미도 등급
intensity	탑승 강도
intensity_rating	탑승 강도 등급
nausea	매스꺼움

데이터 탐색 및 시각화하기



데이터 기초 정보 확인하기

속성명	설명
nausea_rating	매스꺼움 등급
max_speed	최고 속도(mph)
avg_speed	평균 속도(mph)
ride_time	탑승 시간
ride_length	롤러코스터 길이
max_pos_gs, max_neg_gs, max_lateral_gs	롤러코스터를 탔을 때 탑승자가 받는 중력 가속도
total_air_time	탑승자의 무중력 경험 시간
drops	낙하 횟수
highest_drop_height	최고 낙하 높이
inversions	360° 회전하는 횟수

데이터 크기 알아보기

- `shape` 속성을 사용하여 데이터프레임의 크기를 뺀(행의 개수, 열의 개수)으로 출력하기

```
1 df.shape
```

```
(142, 21)
```

데이터프레임이 142개의 행(데이터 개수)과 21개의 열(속성 개수)로 구성되어 있는 것을 알 수 있습니다.

데이터 일부 살펴보기

- `head()`와 `tail()` 메소드를 사용하여 데이터프레임 상단의 일부와 하단의 일부를 확인하기

데이터프레임 객체.`head`(확인할 데이터 개수)

`head()` 메소드를 통해 데이터프레임 상단의 일부 데이터 확인

데이터프레임 객체.`tail`(확인할 데이터 개수)

`tail()` 메소드를 통해 데이터프레임 하단의 일부 데이터 확인

데이터 탐색 및 시각화하기



데이터 일부 살펴보기

데이터프레임 상단의 일부 데이터 출력

`df.head()`



	park_id	theme	rollercoaster_type	custom_design	excitement	excitement_rating	intensity	intensity_rating	nausea	nausea_rating	...			
0	0	Barony Bridge	Bobsleigh Coaster	0	5.49		High	5.71		High	4.86	Medium	...	
1	0	Barony Bridge	Dink	vg_speed	ride_time	ride_length	max_pos_gs	max_neg_gs	max_lateral_gs	total_air_time	drops	highest_drop_height	inversions	
2	0	Barony Bridge	Wil	12	63	1496	2.59	-0.27		1.71	0.00	2	19	0
3	0	Barony Bridge	Wood	11	48	1079	2.83	-0.48		2.68	0.36	4	42	0
4	1	Forest Frontiers	Junior Rolle	11	70	1591	3.13	-1.70		2.15	1.44	11	22	0
				15	79	2401	3.31	-1.15		1.71	5.73	9	62	-1
				13	51	1279	2.38	-0.26		2.45	0.72	4	39	0

- 데이터프레임의 상위 5개 행(인덱스 0~4) 확인
- `info()` 메소드에 제시되었던 속성값 확인

데이터 탐색 및 시각화하기



데이터 일부 살펴보기

데이터프레임 하단의 일부 데이터 출력

```
df.tail()
```



park_id	theme	rollercoaster_type	custom_design	excitement	excitement_rating	intensity	intensity_rating	nausea	nausea_rating	...					
137	30	Botany Breakers	Spiral Coaster	1	5.39	High	5.94	High	3.79	Medium					
138	30	Botany Breakers	Twister Coaster	1	g_speed	ride_time	ride_length	max_pos_gs	max_neg_gs	max_lateral_gs	total_air_time	drops	highest_drop_height	inversions	
139	30	Botany Breakers	Mine Train Coaster	1	13	46	1174	2.69	-0.55	2.96	0.72	3	49	0	
140	30	Botany Breakers	Wooden Roller Coaster	1	18	61	2221	3.86	-1.08	2.29	2.10	6	22	2	
141	30	Botany Breakers	Hypercoaster	1	16	90	2972	3.33	-1.36	2.17	2.43	4	82	0	
5 rows x 21 columns					18	58	2001	3.01	-0.97	1.97	1.62	5	29	1	
					15	61	1791	2.69	-0.45	2.71	0.18	2	62	0	

실행 결과를 통해 데이터프레임의 하위 5개 행(인덱스 137~141)을 확인할 수 있습니다.

데이터 통계치 살펴보기

- 데이터의 통계량(개수, 평균, 표준편차, 최솟값, 4분위수, 최댓값)을 파악하기 위해 `describe()` 메소드를 사용
- 수치값(실수형, 정수형)을 갖는 속성에 대한 통계량 출력, 결측치는 제외

데이터프레임 객체.`describe()`

데이터 탐색 및 시각화하기



데이터 통계치 살펴보기

- 롤러코스터 데이터 통계량 확인하기

```
df.describe( )
```

	park_id	custom_design	excitement	intensity	nausea	max_speed	avg_speed	ride_time	ride_leng
count	142.000000	142.000000	142.000000	142.000000	142.000000	142.000000	142.000000	142.000000	142.0000
mean	16.985915	0.697183	6.337183	6.798028	4.35493	43.570423	14.063380	71.957746	1987.5281
std	8.727543	0.461103	1.228618	3.516611	1.40691	8.980197	4.817543	21.639822	826.8627
min	0.000000	0.000000	2.460000	1.930000	1.33000	29.000000	1.000000	12.000000	620.0000
25%	10.000000	0.000000	5.720000	5.552500	3.39500	38.000000	12.000000	58.250000	1500.7500
50%	19.000000	1.000000	6.405000	6.585000	4.25500	42.500000	14.000000	70.500000	1837.0000
75%	24.000000	1.000000	7.035000	7.627500	5.11000	47.000000	16.000000	83.000000	2354.0000
max	30.000000	1.000000	9.480000	44.940000	9.85000	89.000000	44.000000	186.000000	7497.0000

데이터 탐색 및 시각화하기



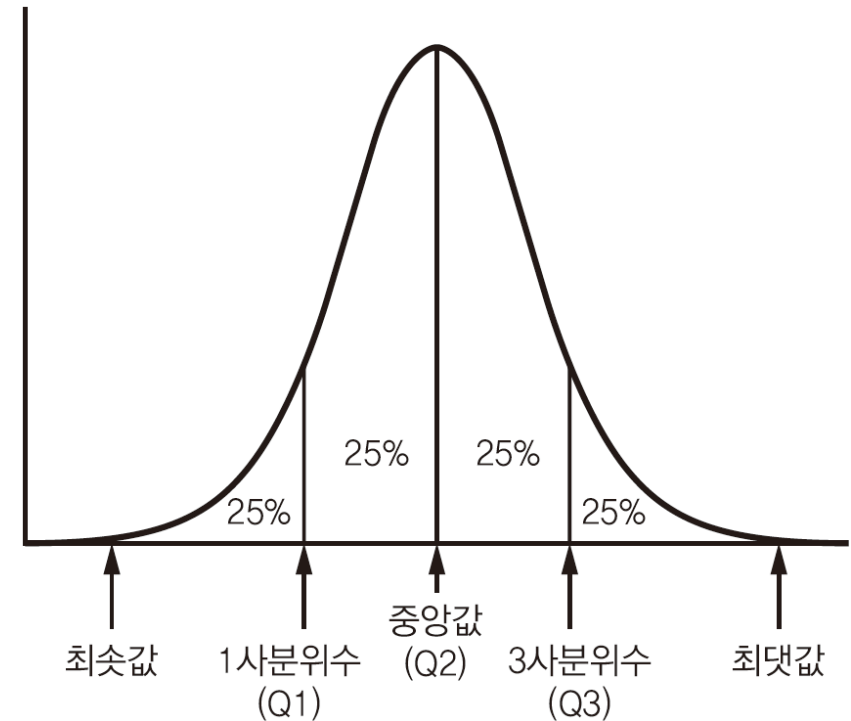
데이터 통계치 살펴보기

표준편차란 데이터가 평균을 중심으로 얼마나 떨어져 있는지를 나타낸 값으로, 표준편차가 작을수록 데이터들이 평균값에 집중

테마파크 식별자(`park_id`)의 경우 통계량이 의미가 없는 속성이지만 숫자로 구성되어 있기 때문에 통계량이 산출됩니다. 흥미도(`excitement`)의 경우 평균이 약 6.34점이며, 대부분의 값들이 평균±표준편차 범위의 6.34 ± 1.23 내에 포함되어 있다는 것을 의미합니다.

4분위수 정하기

- 4분위수는 자료의 크기에 따라 최솟값에서 최댓값 사이의 값을 4등분한 데이터
- 1사분위수는 하위 25%, 2사분위수는 50%(중앙값), 3사분위수는 75%(상위 25%)



탐색정보 일어보기

가장 빠른 롤러코스터 TOP 10의 테마파크 이름(theme),
롤러코스터 유형(rollercoaster_type), 흥미도
(excitement), 최고 속도(max_speed)는 어떻게 알 수
있을까요?

속성 기준으로 데이터 정렬하기

- 속도가 가장 빠른 롤러코스터를 찾으려면 최고 속도 (max_speed) 속성을 기준으로 데이터 정렬
- 데이터를 정렬할 때는 `sort_values()` 메소드 사용

데이터프레임 객체 `sort_values(by = '정렬 기준이 되는 속성명',
ascending = True/False)`

'정렬 기준이 되는 속성명'을 기준으로 정렬. 기본은 오름차순 정렬(ascending = True)이며, 내림차순 정렬은 (ascending = False)로 설정

속성 기준으로 데이터 정렬하기

- 앞에서 df로 설정한 롤러코스터 데이터프레임을 최고 속도(max_speed) 속성 기준
- 내림차순 정렬하여 df2 객체에 저장

```
1 df2 = df.sort_values(by = 'max_speed', ascending = False)
2 pd.set_option('display.max_columns', None)
3 df2
```

데이터 탐색 및 시각화하기



속성 기준으로 데이터 정렬하기

park_id	theme	rollercoaster_type	nausea_rating	max_speed	avg_speed	r
85	21	Vertigo Views	Hypercoaster	Medium	89	21
66	19	Razor Rocks	Air Powered Vertical Coaster	High	85	44
90	21	Vertigo Views	Reverse Freefall Coaster	High	69	39
131	29	Adrenaline Heights	Hypercoaster	High	60	15
133	29	Adrenaline Heights	Floorless Roller Coaster	High	59	15
...
16	5	Mel's World	Spinning Wild Mouse	High	30	1
7	2	Haunted Harbour	Wooden Wild Mouse	Medium	30	12
95	22	Crater Lake	Mini Roller Coaster	Medium	30	12
25	7	Gentle Glen	Dinghy Slide	Low	29	9
30	8	Jolly Jungle	Virginia Reel	High	29	9

142 rows x 21 columns

최고 속도(max_speed) 속성을 기준으로
내림차순 정렬된 속성들을 확인할 수 있습니다.

필요한 데이터 추출하기

- 현재 정렬된 결과에서 테마파크 이름(theme), 롤러코스터 유형(rollercoaster_type), 흥미도(excitement), 최고 속도(max_speed) 속성만 추출하기
- `iloc[]` 메소드와 슬라이싱(Slicing) 사용하기

`iloc[]`는 0부터 시작되는 정수형 인덱스로 행과 열 접근을 하여 데이터를 추출합니다.

데이터프레임 객체.`iloc`[추출할 행 인덱스, 추출할 열 인덱스]

`iloc`는 인덱스 번호로 데이터 추출

(예 `iloc[3:5, [1, 2, 3]]` # 3~4행, 1, 2, 3열의 데이터 추출)

필요한 데이터 추출하기

- 정렬한 데이터프레임에서 최고 속도 TOP 10을 추출하기 위해 행 인덱스를 0~9까지 슬라이싱, 0:10이라고 설정
- 열의 경우에는 테마파크 이름(theme), 롤러코스터 유형(rollercoaster_type), 흥미도(excitement), 최고 속도(max_speed)의 **인덱스 값이 각각 1, 2, 4, 10**이므로 추출할 열 인덱스를 해당 인덱스 값으로 지정

데이터 탐색 및 시각화하기



필요한 데이터 추출하기

[탐색 정보1]에 해당하는 데이터 출력하기

```
1 df2.iloc[0:10, [1, 2, 4, 10]]
```

	theme	rollercoaster_type	excitement	max_speed
85	Vertigo Views	Hypercoaster	9.48	89
66	Razor Rocks	Air Powered Vertical Coaster	8.42	85
90	Vertigo Views	Reverse Freefall Coaster	5.19	69
131	Adrenaline Heights	Hypercoaster	6.89	60
133	Adrenaline Heights	Floorless Roller Coaster	7.19	59
129	Adrenaline Heights	Lay-down Roller Coaster	7.57	59
36	Karts And Coasters	Wooden Roller Coaster	8.88	59
71	Razor Rocks	Hypercoaster	6.77	57
52	Canry Mines	Vertical Drop Coaster	7.48	57
51	Canry Mines	Vertical Drop Coaster	7.14	57

필요한 데이터 추출하기

가장 빠른 롤러코스터
TOP 10의 테마파크 이름
(theme), 롤러코스터 유
형(rollercoaster_type),
흥미도(excitement), 최
고 속도(max_speed)는
무엇인가요?

Top 10 중 가장 빠른 롤러코스터가 있는 테마파크 이름은
Vertigo Views이고 Hypercoaster 유형에 해당하며, 흥미도
점수는 9.48, 최고 속도는 89mph입니다.

Top 10 중 가장 하위에 있는 롤러코스터가 있는 테마파크 이름
은 Canry Mines이고 Vertical Drop Coaster 유형에 해당하며,
흥미도 점수는 7.14, 최고 속도는 57mph입니다.

탐색정보2 알아보기

롤러코스터 유형(rollercoaster_type) 속성에는 몇 가지 유형이 있으며 어떤 유형이 가장 많을까요?

롤러코스터 유형 탐색하기

- 롤러코스터 유형(rollercoaster_type) 속성을 탐색하기 위해서는 유형별 고유한 값 추출하기
- unique() 메소드를 사용하기

unique() 메소드는 데이터에 있는 값을 한 번씩만 출력할 때 사용하고 array 형태로 출력합니다.

```
데이터프레임 객체['속성명'].unique( )  
# 지정한 속성명의 값을 한 번씩만 출력
```


롤러코스터 유형 탐색하기

- 롤러코스터 유형(rollercoaster_type) 속성값을 한 번씩만 출력하기

```
1 x1 = df['rollercoaster_type'].unique()  
2 x1
```

```
array(['Bobsleigh Coaster ', 'Dinghy Slide', 'Wild Mouse',  
      'Wooden Roller Coaster ', 'Junior Roller Coaster',  
      'Side-Friction Roller Coaster', 'Wooden Wild Mouse',  
      'Stand Up Roller Coaster', 'Compact Inverted Coaster',  
      'Looping Roller Coaster', 'Vertical Drop Coaster',  
      'Inverted Roller Coaster', 'Spinning Wild Mouse',  
      'Suspended Swinging Coaster', 'Mini Roller Coaster',  
      'Virginia Reel', 'Heartline Twister Coaster',  
      'Lay-down Roller Coaster', 'Spiral Coaster', 'Mine Train Coaster',  
      'Corkscrew Roller Coaster', 'Stand Up Twister Roller Coaster',  
      'Hypercoaster', 'Twister Coaster', 'Air Powered Vertical Coaster',  
      'Floorless Roller Coaster', 'Giga Coaster', 'Water Coaster',  
      'Reverse Freefall Coaster', 'Inverted Hairpin Coaster',  
      'Hyper-Twister Roller Coaster'], dtype=object)
```

롤러코스터 유형 탐색하기

- 생각보다 많은 롤러코스터 유형이 총 몇 개의 유형이 있는지 살펴보기
- `nunique()` 메소드 사용하기

데이터프레임 객체['속성명'].`nunique()`

지정한 속성명의 고유한 값의 총 개수 출력

롤러코스터 유형 탐색하기

- 롤러코스터 유형(rollercoaster_type) 속성의 고유한 값의 개수를 출력하기

```
1 x2 = df['rollercoaster_type'].nunique( )  
2 x2
```

31

롤러코스터 유형(rollercoaster_type) 속성의 고유한 값이 총 31개가 있다는 것을 확인할 수 있습니다.

데이터 탐색 및 시각화하기



롤러코스터 유형별 개수 비교하기

- 롤러코스터 유형(rollercoaster_type) 속성의 유형별 개수를 파악하기 위해 `value_counts()` 메소드 사용하기

데이터프레임 객체['속성명'].value_counts()

지정한 속성명의 고유한 값별로 개수 출력

롤러코스터 유형별 개수 비교하기

- 롤러코스터 유형(rollercoaster_type) 속성의 31개 유형별 개수를 rollercoaster_type_count에 저장한 후 그 내용을 출력하기

```
1 rollercoaster_type_count = df['rollercoaster_type'].value_counts( )
```

```
2 rollercoaster_type_count
```

Wooden Roller Coaster	22	Floorless Roller Coaster	2
Looping Roller Coaster	15	Giga Coaster	2
Vertical Drop Coaster	8	Virginia Reel	2
Corkscrew Roller Coaster	8	Side-Friction Roller Coaster	2
Hypercoaster	7	Hyper-Twister Roller Coaster	2
Dinghy Slide	7	Stand Up Twister Roller Coaster	1
Mine Train Coaster	7	Heartline Twister Coaster	1
Compact Inverted Coaster	6	Spinning Wild Mouse	1
Mini Roller Coaster	6	Air Powered Vertical Coaster	1
Stand Up Roller Coaster	5	Wild Mouse	1
Inverted Roller Coaster	5	Water Coaster	1
Junior Roller Coaster	5	Reverse Freefall Coaster	1
Spiral Coaster	5	Inverted Hairpin Coaster	1
Twister Coaster	4		
Suspended Swinging Coaster	4		
Lay-down Roller Coaster	4		
Wooden Wild Mouse	3		
Bobble Coaster	3		

Name: rollercoaster_type, dtype: int64

데이터 탐색 및 시각화하기



롤러코스터 유형별 개수 비교하기

가장 많은 유형(Wooden Roller Coaster)의 롤러코스터부터
내림차순으로 정렬되어 출력된 것을 확인할 수 있습니다.

`value_counts()` 메
소드는 오른쪽 실행
결과처럼 1차원 형태
인 시리즈(series)
형태로 출력됩니다.

Wooden Roller Coaster	22	Floorless Roller Coaster	2
Looping Roller Coaster	15	Giga Coaster	2
Vertical Drop Coaster	8	Virginia Reel	2
Corkscrew Roller Coaster	8	Side-Friction Roller Coaster	2
Hypercoaster	7	Hyper-Twister Roller Coaster	2
Dinghy Slide	7	Stand Up Twister Roller Coaster	1
Mine Train Coaster	7	Heartline Twister Coaster	1
Compact Inverted Coaster	6	Spinning Wild Mouse	1
Mini Roller Coaster	6	Air Powered Vertical Coaster	1
Stand Up Roller Coaster	5	Wild Mouse	1
Inverted Roller Coaster	5	Water Coaster	1
Junior Roller Coaster	5	Reverse Freefall Coaster	1
Spiral Coaster	5	Inverted Hairpin Coaster	1
Twister Coaster	4	Name: rollercoaster_type, dtype: int64	
Suspended Swinging Coaster	4		
Lay-down Roller Coaster	4		
Wooden Wild Mouse	3		
Bobsleigh Coaster	3		

클러스터 유형별 개수 비교하기

기본 막대그래프로 시각화

- 기본 막대그래프로 시각화하기 위해 `plot.bar()`와 `plt.show()`를 사용하기

```
데이터프레임 객체.plot.bar( ) 또는  
데이터프레임 객체.plot(kind='bar')  
plt.show( )  
# 그래프를 화면에 출력
```

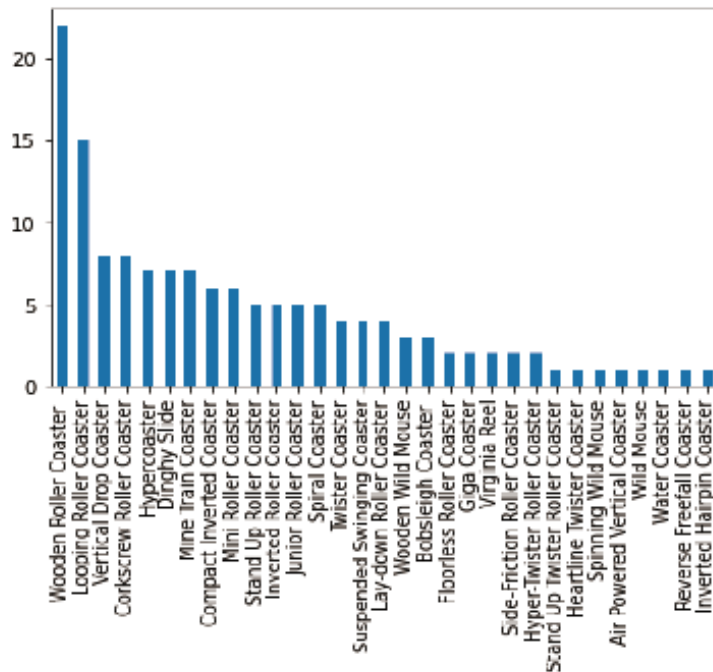
데이터 탐색 및 시각화하기



롤러코스터 유형별 개수 비교하기

• 유형 기본 막대그래프로 시각화하기

- 1 rollercoaster_type_count.plot.bar()
- 2 plt.show()

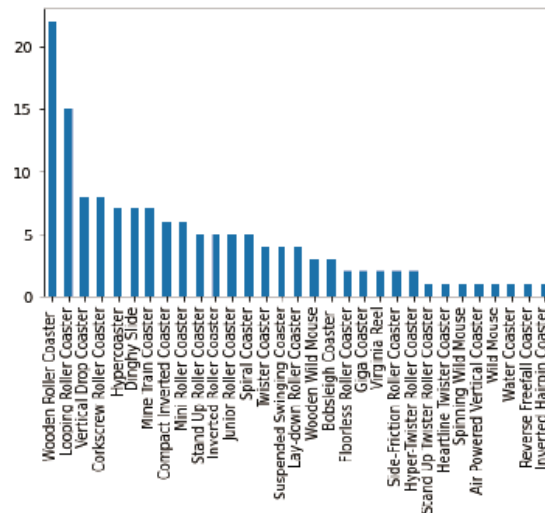


데이터 탐색 및 시각화하기



롤러코스터 유형별 개수 비교하기

롤러코스터 유형별 개수를 막대그래프로 시각화 결과, 막대그래프의 크기와 각 막대 너비가 작고, x축과 y축이 의미하는 바도 잘 드러나지 않습니다.



가로 막대그래프

bar 대신 barh를 입력하면 가로 막대그래프를 출력할 수 있습니다.

```
데이터프레임 객체.plot.barh( ) 또는 데이터프레임 객체.plot(kind = 'barh')
```

```
# 가로 막대그래프 출력
```

플랴코스터 유형별 개수 비교하기

막대그래프 크기와 막대 너비 조정 및 x축과 y축 이름 출력

- 막대그래프의 전체 크기와 막대 너비를 조절하기 위해서는 `figure()`, x축과 y축의 이름을 바꾸려면 `xlabel()`, `ylabel()`을 사용하기

그래프를 그리기 전에 크기 제시

```
plt.figure(figsize = (x축 사이즈, y축 사이즈))
```

```
데이터프레임 객체.plot.bar(width = 너비 사이즈)
```

```
plt.xlabel('x축 이름')
```

```
plt.ylabel('y축 이름')
```

그래프 크기
(`figsize`)나 막대
의 너비(`width`)
는 인치(`inch`)를
기준으로 합니다.

롤러코스터 유형별 개수 비교하기

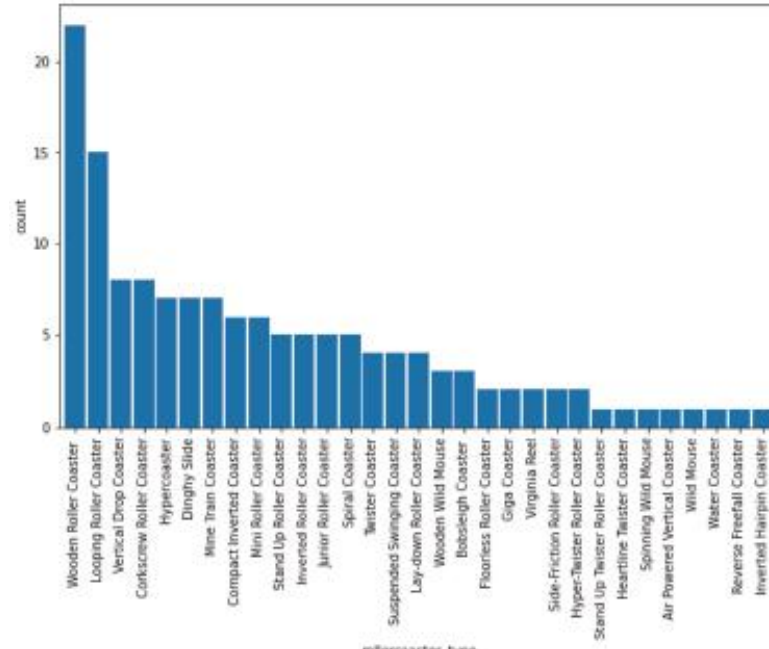
- 막대그래프의 가로는 10인치, 세로는 6인치, 막대의 너비는 0.9인치로 설정하며, x축의 이름은 'rollercoaster_type', y축의 이름은 'count'로 설정하기

```
1 plt.figure(figsize = (10, 6))
2 rollercoaster_type_count.plot.bar(width = 0.9)
3 plt.xlabel('rollercoaster_type')
4 plt.ylabel('count')
5 plt.show( )
```

데이터 탐색 및 시각화하기

롤러코스터 유형별 개수 비교하기

그래프에 제목을 입력
하려면 plt.title('그래프
제목')을 입력하면 됩
니다.



막대그래프의 크기가 커지고 x축과 y축의 정보가 표시되어 좀 더 보기 좋아졌음을 확인할 수 있으나, 각 막대가 가리키는 롤러코스터 유형의 정확한 개수를 알 수 없는 상태입니다.

롤러코스터 유형별 개수 비교하기

막대그래프의 막대에 문자열 출력

- 각 막대 위에 값을 표기하기 위해 그래프 내에 문자열을 입력할 수 있는 `text()` 사용하기
- `text()`를 사용할 때는 값을 표기할 위치 고려

```
plt.text(x축 위치, y축 위치, 출력할 내용)  
# x축 위치: 롤러코스터 유형 31개 인덱스 값  
# y축 위치: 롤러코스터 각 유형별 개수에 0.5를 더한 값  
# 출력할 내용: 그래프의 (x, y) 위치에 입력한 내용 출력
```

롤러코스터 유형별 개수 비교하기

- Wooden Roller Coaster 유형의 막대 위에 22라고 표기하기
 - x축 위치는 Wooden Roller Coaster 유형의 인덱스 값인 0 설정
 - y축 위치는 유형별 개수인 22에 0.5를 더한 22.5로 설정

```
1 plt.text(0, 22.5, 22)
```

막대의 경계 위쪽으로 텍스트를 보기 좋게 출력하기 위해 y축 위치의 값에 0.5를 더합니다.

• 데이터 탐색 및 시각화하기



불라코스터 유형별 개수 비교하기

- Dinghy Slide 유형의 막대 위에 70이라고 표기하기
 - x축을 기준으로 6번째 위치
 - 인덱스 값은 5(인덱스의 시작값이 0이므로)이고
 - 개수는 7개이므로 y축 위치는 7.5로 설정한다.

```
1 plt.text(5, 7.5, 7)
```


롤러코스터 유형별 개수 비교하기

Twister Coaster 유형
은
어떻게 표현할까요?

Twister Coaster 유형은
14번째에 있으니까 x축 위
치는 13이고, 개수가 4개이
니까 y축 위치를 4.5로 설
정합니다.

롤러코스터 유형 속성의 유형별 개수의 plt.text() 표현

유형	유형별 개수	plt.text() 표현
Wooden Roller Coaster	22	plt.text(0, 22.5, 22)
Looping Roller Coaster	15	plt.text(1, 15.5, 15)
Vertical Drop Coaster	8	plt.text(2, 8.5, 8)
Corkscrew Roller Coaster	8	plt.text(3, 8.5, 8)
Hypercoaster	7	plt.text(4, 7.5, 7)

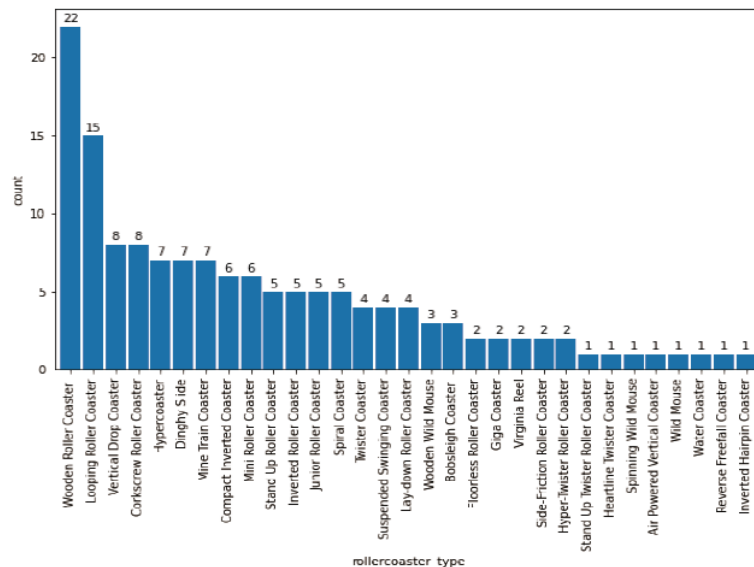
롤러코스터 유형 속성의 유형별 개수의 plt.text() 표현

유형	유형별 개수	plt.text() 표현
Dinghy Slide	7	plt.text(5, 7.5, 7)
Mine Train Coaster	7	plt.text(6, 7.5, 7)
Vertical Drop Coaster	6	plt.text(7, 6.5, 6)
...		
Hypercoaster	1	plt.text(30, 1.5, 1)

데이터 탐색 및 시각화하기



롤러코스터 유형별 개수 비교하기



Wooden Roller Coaster 유형이 22개로 가장 많고,
Looping RollerCoaster 유형이 15개로 그 다음인 것을 확인할
수 있습니다.

롤러코스터 유형별 개수 비교하기

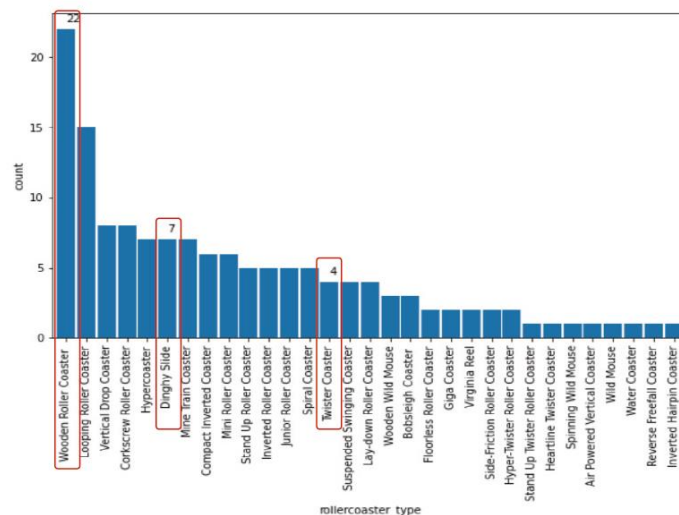
• 막대그래프에 해당 문자열을 출력하기

```
▶ 1 rollercoaster_type_count = df['rollercoaster_type'].value_counts( )  
2 plt.figure(figsize = (10, 6))  
3 rollercoaster_type_count.plot.bar(width = 0.9)  
4 plt.xlabel('rollercoaster_type')  
5 plt.ylabel('count')  
6 plt.text(0, 22.5, 22)  
7 plt.text(5, 7.5, 7)  
8 plt.text(13, 4.5, 4)  
9 plt.show()
```

데이터 탐색 및 시각화하기



롤러코스터 유형별 개수 비교하기



Wooden Roller Coaster 막대 위에 22, Dinghy Slide 막대 위에 7, Twister Coaster 막대 위에 4가 출력되어 유형별 개수를 쉽게 확인할 수 있습니다.

롤러코스터 유형별 개수 비교하기

- `plt.text()` 규칙을 찾아 반복문을 사용하면 효율적으로 코드를 작성할 수 있다.
- x축 위치는 0~30까지의 인덱스로 정하고
- y축 위치는 각 유형별 개수 (`rollercoaster_type_count`)에 0.5를 더하는 방식을 사용하면 된다.

롤러코스터 유형별 개수 비교하기

반복문을 사용하여 막대그래프의 막대에 문자열 출력

- `i`라는 변수를 사용하여 `df['rollercoaster_type']`의 고유한 값의 개수(31개)만큼 반복하면서 롤러코스터 유형별 개수의 값을 표기할 (x, y) 위치 정하기
- 표기할 위치를 보기 좋게 가운데로 정렬하기 위하여
 - 세로 정렬은 `va = 'center'`,
 - 가로 정렬은 `ha = 'center'`로 설정하기

롤러코스터 유형별 개수 비교하기

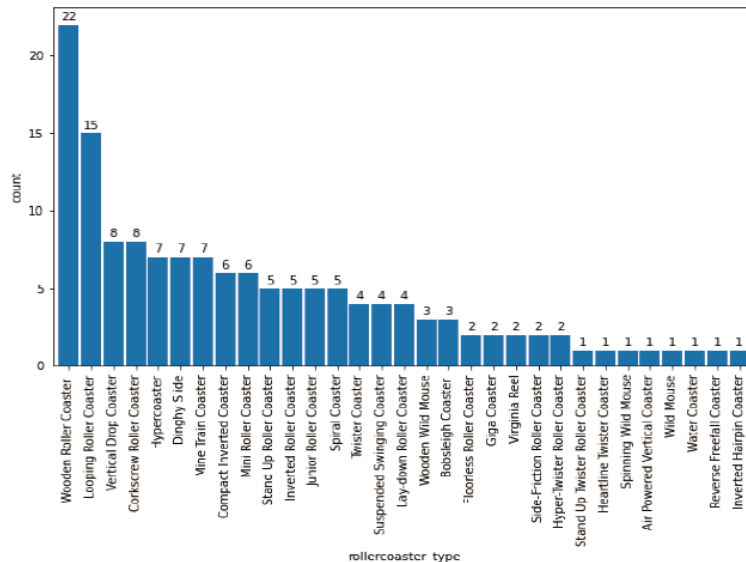
- 반복문을 사용하여 막대그래프의 막대에 문자열 출력하기

```
▶ 1 rollercoaster_type_count = df['rollercoaster_type'].value_counts( )
2 plt.figure(figsize = (10, 6))
3 rollercoaster_type_count.plot.bar(width = 0.9)
4 plt.xlabel('rollercoaster_type')
5 plt.ylabel('count')
6 for i range(len(x1)):
7     plt.text(i, rollercoaster_type_count[i] + 0.5,
8             rollercoaster_type_count[i], va = 'center', ha =
9             'center')
9 plt.show()
```

데이터 탐색 및 시각화하기



롤러코스터 유형별 개수 비교하기



Wooden Roller Coaster 유형이 22개로 가장 많고,
Looping RollerCoaster 유형이 15개로 그 다음인 것을 확인할
수 있습니다

탐색 정보3 일어보기

최고 속도(max_speed), 탑승 시간(ride_time), 롤러코스터 길이(ride_length), 최고 낙하 높이(highest_drop_height) 속성

에

흥미도(excitement) 속성과 가장 관련이 깊은 속성은 무엇일까요?

흥미도 속성과 다른 속성 간 상관관계 분석하기

- 속성 간의 관계를 수치로 파악하는 '상관관계 분석' 실시
- 상관관계 분석 결과를 내림차순으로 정렬
- 흥미도 속성과 관련이 깊은 속성을 찾기

흥미도 속성과 다른 속성 간 상관관계 분석하기

- `corr()` 는 속성 간 관계를 알 수 있도록 데이터프레임 형태의 상관관계표를 출력하기

데이터프레임 객체 `corr()`

상관관계(correlation) 분석

데이터 탐색 및 시각화하기



흥미도 속성과 다른 속성 간 상관관계 분석하기

- `corr()` 는 속성 간 관계를 알 수 있도록 데이터프레임 형태의 상관관계표를 출력하기

```
1 corr = corr()  
2 corr
```



	park_id	custom_design	excitement	intensity	nausea	max_speed	avg_speed	ride_time
park_id	1.000000	0.275621	0.208506	0.170565	0.122935	0.286958	0.199232	0.112578
custom_design	0.275621	1.000000	-0.301970	-0.054868	-0.239727	-0.160095	-0.259485	0.044198
excitement	0.208506	-0.301970	1.000000	0.218619	0.349712	0.500800	0.393300	0.310817
intensity	0.170565	-0.054868	0.218619	1.000000	0.195823	0.191066	0.149153	-0.006945
nausea	0.122935	-0.239727	0.349712	0.195823	1.000000	0.193787	0.207210	0.122904
max_speed	0.286958	-0.160095	0.500800	0.191066	0.193787	1.000000	0.659485	0.189501
avg_speed	0.199232	-0.259485	0.393300	0.149153	0.207210	0.659485	1.000000	-0.158757
ride_time	0.112578	0.044198	0.310817	-0.006945	0.122904	0.189501	-0.158757	1.000000
ride_length	0.196867	-0.145209	0.555398	0.094163	0.204650	0.503572	0.311627	0.830813
max_pos_gs	0.235944	0.053533	0.296372	0.150158	0.199447	0.631194	0.350726	0.137502
max_neg_gs	-0.086916	-0.049630	0.004703	0.009947	-0.018263	-0.095027	-0.018478	-0.023241
max_lateral_gs	-0.019231	0.117613	0.147490	0.236579	0.268021	0.099968	-0.038441	0.075678
total_air_time	0.001091	-0.395978	0.513925	0.060871	0.164505	0.533315	0.385321	0.440037
drops	0.038230	-0.171402	0.207620	-0.004128	0.020387	0.016034	0.000896	0.236477
highest_drop_height	0.263336	-0.150410	0.474213	0.192446	0.145657	0.931333	0.546266	0.210949
inversions	0.550886	0.057233	0.143430	0.072356	0.289689	0.060904	0.117135	0.028671

흥미도 속성과 다른 속성 간 상관관계 분석하기

- 속성 간의 상관관계가 -1.0 ~ 1.0 사이의 실숫값
- 행 이름과 열 이름이 교차하는 지점의 수치는 두 속성 간의 상관관계를 나타내는 r값
- 같은 속성끼리는 완전히 일치하는 값을 가지기 때문에 상관관계 값은 1.0

롤러코스터 데이터 셋의 21개 속성 중 수치를 갖는 16개 속성만 상관관계를 출력하는 것을 기억할 것!

상관관계표의
흥미도 속성만
막대그래프로
시각화 하기

- 앞에서 출력한 상관관계표(corr)에서 흥미도 (excitement) 속성과 연관된 상관관계만 살펴보기

```
corr('속성명' )
```

```
# 데이터프레임에서 해당 속성과 연관된 상관관계 출력
```


상관관계표의 흥미도 속성만 막대그래프로 시각화하기

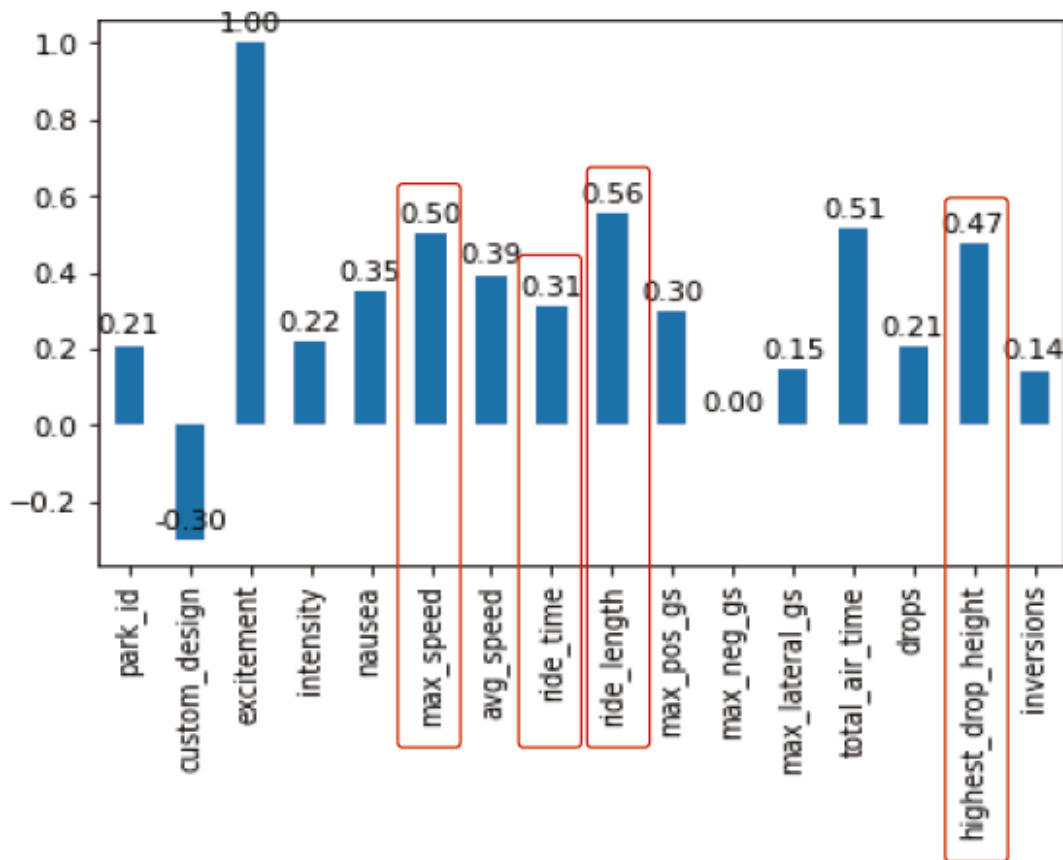
- 흥미도 속성만 추출하여 막대그래프로 시각화하고, 각 막대의 값도 출력하기

```
1 corr = df.corr( )
2 corr2 = corr['excitement']
3 corr2.plot.bar( )
4 for i in range(len(corr2)):
5     plt.text(i, corr2[i] + 0.05, '%.2f' %corr2[i], va = 'center', ha = 'center')
6 # '%.2f'는 소수점 아래 둘째 자리까지의 실수로 출력하라는 의미
```

데이터 탐색 및 시각화하기



상관관계표의
흥미도 속성만
막대그래프로
시각화 하기

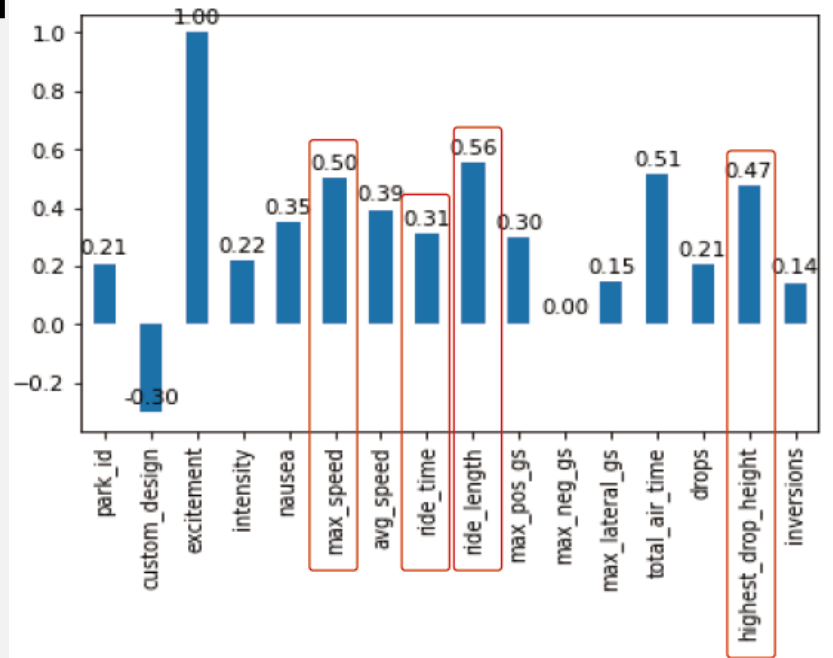


데이터 탐색 및 시각화하기



상관관계표의 흥미도 속성만 막대그래프로 시각화 하기

- 흥미도(excitement) 속성과 가장 관련이 깊은 속성은 0.56
의 롤러코스터 길이(ride_length)



흥미도 속성과 상관관계가 높은 속성부터 나열해 보기

흥미도(excitement) 속성과 높은 상관관계를 보이는 속성을
`sort_values()` 메소드를 사용하여 내림차순으로 정렬하기

```
1 corr = df.corr()  
2 corr2 = corr.sort_values(by='excitement', ascending=False)  
3 corr3
```

데이터 탐색 및 시각화하기



	park_id	custom_design	excitement	intensity	nausea
excitement	0.208506	-0.301970	1.000000	0.218619	0.349712
ride_length	0.196867	-0.145209	0.555398	0.094163	0.204650
total_air_time	0.001091	-0.395978	0.513925	0.060871	0.164508
max_speed	0.286958	-0.160095	0.500800	0.191066	0.193787
highest_drop_height	0.263336	-0.150410	0.474213	0.192446	0.145657
avg_speed	0.199232	-0.259485	0.393300	0.149153	0.207216
nausea	0.122935	-0.239727	0.349712	0.195823	1.000000
ride_time	0.112578	0.044198	0.310817	-0.006945	0.122904
max_pos_gs	0.235944	0.053533	0.296372	0.150158	0.199447
intensity	0.170565	-0.054868	0.218619	1.000000	0.195823
park_id	1.000000	0.275621	0.208506	0.170565	0.122935
drops	0.038230	-0.171402	0.207620	-0.004128	0.020387
max_lateral_gs	-0.019231	0.117613	0.147490	0.238579	0.268021
inversions	0.550886	0.057233	0.143430	0.072356	0.289689
max_neg_gs	-0.086916	-0.049630	0.004703	0.009947	-0.018263
custom_design	0.275621	1.000000	-0.301970	-0.054868	-0.239727

데이터 탐색 및 시각화하기



	park_id	custom_design	excitement	intensity	nausea
excitement	0.208506	-0.301970	1.000000	0.218619	0.349712
ride_length	0.196867	-0.145209	0.555398	0.094163	0.204650
total_air_time	0.001091	-0.395978	0.513925	0.060871	0.164505
max_speed	0.286958	-0.160095	0.500800	0.191066	0.193787
highest_drop_height	0.263336	-0.150410	0.474213	0.192446	0.145657
avg_speed	0.199232	-0.259485	0.393300	0.149153	0.207210
nausea	0.122935	-0.239727	0.349712	0.195823	1.000000
ride_time	0.112578	0.044198	0.310817	-0.006945	0.122904
max_pos_gs	0.235944	0.053533	0.296372	0.150158	0.199447
intensity	0.170565	-0.054868	0.218619	1.000000	0.195823
park_id	1.000000	0.275621	0.208506	0.170565	0.122935
drops	0.038230	-0.171402	0.207620	-0.004128	0.020387
max_lateral_gs	-0.019231	0.117613	0.147490	0.236579	0.268021
inversions	0.550886	0.057233	0.143430	0.072356	0.289689
max_neg_gs	-0.086916	-0.049630	0.004703	0.009947	-0.018263
custom_design	0.275621	1.000000	-0.301970	-0.054868	-0.239727

[탐색 정보3]에서 제시되었던 롤러코스터 길이 (ride_length), 최고 속도(max_speed), 최고 낙하 높이(highest_drop_height), 탑승 시간 (ride_time) 속성 외에도 흥미도 속성과 상관관계가 높은 속성에는 탑승자의 무중력 경험 시간 (total_air_time), 평균 속도(avg_speed), 메스꺼움(nausea) 등이 있습니다.

상관관계 분석

- 두 속성 간에 관련성을 직선으로 표현. 이 직선에 얼마나 밀집되어 있는가 (관련성이 높은가)를 $-1.0 \sim 1.0$ 사이의 실숫값 r 로 설명하는 분석 방법
- 양수는 양의 상관관계, 음수는 음의 상관관계를 의미하며, 절댓값 $1(1)$ 에 가까울수록 두 속성 간의 관련성이 높고
- 0 에 가까울수록 두 속성 간의 관련성이 낮음.

상관관계 분석

r 값의 범위에 따른 상관관계 정도

상관관계 r 값의 절댓값	상관관계 정도
0.0 ~ 0.2	상관관계가 거의 없다.
0.2 ~ 0.4	상관관계가 낮다.
0.4 ~ 0.6	상관관계가 있다.
0.6 ~ 0.8	상관관계가 높다.
0.8 ~ 1.0	상관관계가 매우 높다.

상관관계 분석



$$r = -1$$

음의 상관관계가
강하다.



$$-1 < r < 0$$

음의 상관관계가
있기는 하다.



$$r = 0$$

상관관계가
없다.



$$0 < r < 1$$

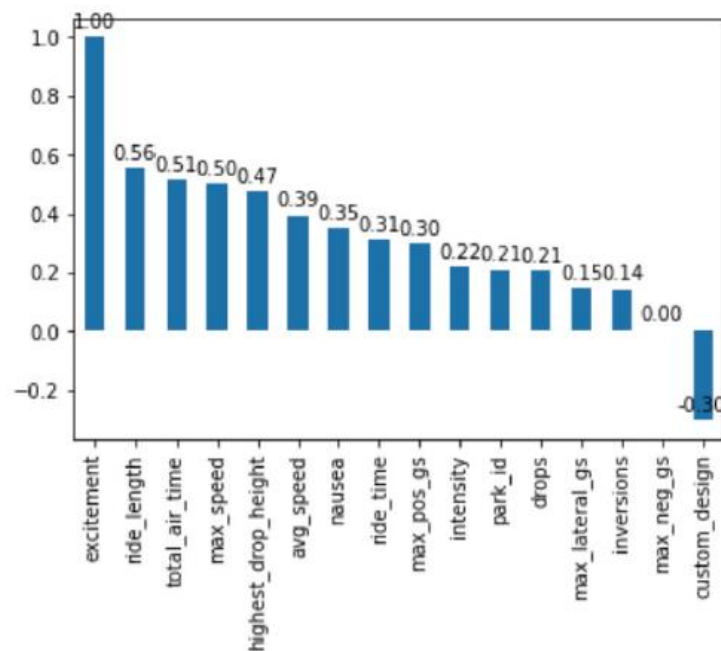
양의 상관관계가
있기는 하다.



$$r = 1$$

양의 상관관계가
강하다.

내림차순으로 정렬한 상관관계표(corr)에서 흥미도 속성만 추출하여 다음 결과와 같은 막대그래프로 출력하기



내림차순으로 정렬한 상관관계표(corr)에서 흥미도 속성만 추출하여 다음 결과와 같은 막대그래프로 출력하기

```
1 corr=df.corr()
2 corr2 = corr['excitement']
3 corr3=corr2.sort_values(ascending=False)
4 corr3
5 corr3.plot.bar()
6 for i in range(len(corr3)):
7     plt.text(i, corr3[i]+0.05, '%.2f' %corr3[i], va='center', ha='center')
```

문제 정의하기



데이터 불러오기



데이터 탐색 및 시각화하기

문제 상황 이해하기
탐색할 정보 알아보기

롤러코스터 데이터 셋 소개
롤러코스터 데이터 셋 불러오기

데이터 셋의 속성 살펴보기
탐색 정보 알아보기(정렬, 추출, 상관관계 등)
상관관계 분석하여 막대그래프로 시각화하기
실행 결과 해석하기

1. 가장 빠른 롤러코스터 TOP 10의 테마파크 이름(theme), 롤러코스터 유형(rollercoaster_type), 흥미도(excitement), 최고 속도(max_speed)는 어떻게 알게 되었나요?

- 최고 속도(max_speed) 속성을 내림차순으로 정렬한 결과, 가장 빠른 롤러코스터는 Vertigo Views라는 테마파크에 있고 Hypercoaster 유형이며, 흥미도 점수는 9.48, 최고 속도는 89mph이었습니다.
- 최고 속도 10위의 롤러코스터는 Canry Mines라는 테마파크에 있고, Vertical Drop Coaster 유형이며, 흥미도 점수는 7.14, 최고 속도는 57mph이었습니다.

2. 롤러코스터 유형(rollercoaster_type) 속성에는 몇 가지 유형이 있으며 어떤 유형이 가장 많았나요?

총 31개의 유형이 있으며, Wooden Roller Coaster 유형이 22개로 가장 많았습니다.

3. 최고 속도(max_speed), 탑승 시간(ride_time), 롤러코스터 길이(ride_length), 최고 낙하 높이(highest_drop_height) 속성 중 흥미도(excitement) 속성과 가장 관련이 깊은 속성은 무엇이었나요?

최고 속도, 탑승 시간, 롤러코스터 길이, 최고 낙하 높이 속성과 흥미도 속성과의 상관관계 값은 각각 0.50, 0.31, 0.56, 0.47로 이 중 흥미도 속성과 가장 관련이 깊은 속성은 롤러코스터 길이였습니다.

데이터로부터 원하는 정보를 탐색하기 위해 데이터의 기초 정보와 크기, 데이터프레임의 상위와 하위 일부, 데이터의 통계량을 출력하여 살펴보았다.

[탐색 정보1]에서는 속성별로 데이터를 정렬한 후, 이 중 우리에게 필요한 속성만을 추출하여 필요한 정보를 확인하는 과정을 살펴보았다.

[탐색 정보2]에서는 속성의 고유한 값과 그 개수를 추출하였고, 이를 막대그래프로 시각화하여 정보를 한눈에 살펴볼 수 있었다.

[탐색 정보3]에서는 속성 간의 상관관계를 분석하고 이를 막대그래프로 시각화하여 속성 간의 관련 정도를 쉽게 확인할 수 있었다.