Exploratory Data Analysis

EDA 란

- 수집한 데이터가 들어왔을 때, 이를 다양한 각도에서 관찰하고 이해하는 과정

- 데이터를 분석하기 전에 그래프나 통계적인 방법으로 자료를 직관적으로 바라보는 과정

EDA 필요 이유

- 데이터의 분포 및 값을 검토함으로써 데이터가 표현하는 현상을 더 잘 이해
- 데이터에 대한 잠재적인 문제를 발견
- 다양한 각도에서 살펴보는 과정을 통해 문제 정의 단계에서 미쳐 발생하지 못했을 다양한 패턴을 발견

EDA 과정

- 분석의 목적과 변수가 무엇이 있는지 확인, 개별 변수의 이름이나 설명을 가지는지 확인
- 데이터를 전체적으로 살펴보기
    - 데이터에 문제가 없는지 확인, head나 tail 부분을 확인
    - 추가적으로 다양한 탐색(이상치, 결측치 등을 확인하는 과정)
- 데이터의 개별 속성값을 관찰
    - 각 속성값이 예측한 범위와 분포를 갖는지 확인
    - 만약 그렇지 않다면, 이유가 무엇인지를 확인해 본다.
- 속성 간의 관계에 초점을 맞추어, 개별 속성 관찰에서 찾아내지 못했던 패턴을 발견(상관관계, 시각화 등)

(적용) EDA과정

- EDA를 위한 필요한 라이브러리를 import 함

  import pandas as pd

  import numpy as np

  import seaborn as sns # 가시화

  import matplotlib.pyplot as plt # 가시화

  %matplotlib inline

  sns.set(color_codes=True)

(적용) EDA과정

- 데이터를 data frame으로 로딩

10, 000 rows and more than 10 columns

df = pd.read_csv("data.csv")

To display the top 5 rows

df.head(5)

To display the bottom 5 rows

df.tail(5)

(적용) EDA과정

	Make	Model	Year	Engine Fuel Type	Engine HP	Engine Cylinders	Transmission Type	Driven_Wheels	Number of Doors	Market Category	Vehicle Size	Vehicle Style	highway MPG	city mpg	Popula
0	BMW	1 Series M	2011	premium unleaded (required)	335.0	6.0	MANUAL	rear wheel drive	2.0	Factory Tuner,Luxury,High-Performance	Compact	Coupe	26	19	3
1	BMW	1 Series	2011	premium unleaded (required)	300.0	6.0	MANUAL	rear wheel drive	2.0	Luxury,Performance	Compact	Convertible	28	19	3
2	BMW	1 Series	2011	premium unleaded (required)	300.0	6.0	MANUAL	rear wheel drive	2.0	Luxury,High-Performance	Compact	Coupe	28	20	3
3	BMW	1 Series	2011	premium unleaded (required)	230.0	6.0	MANUAL	rear wheel drive	2.0	Luxury,Performance	Compact	Coupe	28	18	3

	Make	Model	Year	Engine Fuel Type	Engine HP	Engine Cylinders	Transmission Type	Driven_Wheels	Number of Doors	Market Category	Vehicle Size	Vehicle Style	highwa MP(
11909	Acura	ZDX	2012	premium unleaded (required)	300.0	6.0	AUTOMATIC	all wheel drive	4.0	Crossover,Hatchback,Luxury	Midsize	4dr Hatchback	2
11910	Acura	ZDX	2012	premium unleaded (required)	300.0	6.0	AUTOMATIC	all wheel drive	4.0	Crossover,Hatchback,Luxury	Midsize	4dr Hatchback	2
11911	Acura	ZDX	2012	premium unleaded (required)	300.0	6.0	AUTOMATIC	all wheel drive	4.0	Crossover,Hatchback,Luxury	Midsize	4dr Hatchback	2
11912	Acura	ZDX	2013	premium unleaded (recommended)	300.0	6.0	AUTOMATIC	all wheel drive	4.0	Crossover,Hatchback,Luxury	Midsize	4dr Hatchback	2
11913	Lincoln	Zephyr	2006	regular unleaded	221.0	6.0	AUTOMATIC	front wheel drive	4.0	Luxury	Midsize	Sedan	2

(적용) EDA과정

- 자료형 조사

```
In [4]: # Checking the data type
        df.dtypes

Out[4]: Make                  object
        Model                 object
        Year                   int64
        Engine Fuel Type      object
        Engine HP            float64
        Engine Cylinders     float64
        Transmission Type     object
        Driven_Wheels         object
        Number of Doors      float64
        Market Category       object
        Vehicle Size          object
        Vehicle Style         object
        highway MPG            int64
        city mpg               int64
        Popularity             int64
        MSRP                   int64
        dtype: object
```

(적용) EDA과정

- 관련없는(irrelevant columns) 컬럼을 제외함

In [6]:
```
# Dropping irrelevant columns
df = df.drop(['Engine Fuel Type', 'Market Category', 'Vehicle Style', 'Popularity', 'Number of Doors', 'Vehicle Size
df.head(5)
```

Out[6]:

	Make	Model	Year	Engine HP	Engine Cylinders	Transmission Type	Driven_Wheels	highway MPG	city mpg	MSRP
0	BMW	1 Series M	2011	335.0	6.0	MANUAL	rear wheel drive	26	19	46135
1	BMW	1 Series	2011	300.0	6.0	MANUAL	rear wheel drive	28	19	40650
2	BMW	1 Series	2011	300.0	6.0	MANUAL	rear wheel drive	28	20	36350
3	BMW	1 Series	2011	230.0	6.0	MANUAL	rear wheel drive	28	18	29450
4	BMW	1 Series	2011	230.0	6.0	MANUAL	rear wheel drive	28	18	34500

(적용) EDA과정

- 컬럼 이름을 변경

In [7]:
```
# Renaming the column names
df = df.rename(columns={"Engine HP": "HP", "Engine Cylinders": "Cylinders", "Transmission Type": "Transmission", "Dr
df.head(5)
```

Out[7]:

	Make	Model	Year	HP	Cylinders	Transmission	Drive Mode	MPG-H	MPG-C	Price
0	BMW	1 Series M	2011	335.0	6.0	MANUAL	rear wheel drive	26	19	46135
1	BMW	1 Series	2011	300.0	6.0	MANUAL	rear wheel drive	28	19	40650
2	BMW	1 Series	2011	300.0	6.0	MANUAL	rear wheel drive	28	20	36350
3	BMW	1 Series	2011	230.0	6.0	MANUAL	rear wheel drive	28	18	29450
4	BMW	1 Series	2011	230.0	6.0	MANUAL	rear wheel drive	28	18	34500

(적용) EDA과정

- 중복 행을 제외함

In [8]:
```
# Total number of rows and columns
df.shape
```

Out[8]: (11914, 10)

In [10]:
```
# Rows containing duplicate data
duplicate_rows_df = df[df.duplicated()]
print("number of duplicate rows: ", duplicate_rows_df.shape)
```

```
number of duplicate rows:  (989, 10)
```

(적용) EDA과정

- missing or null 값을 제외함

```
In [13]: print(df.isnull().sum())
```

```
Make              0
Model             0
Year              0
HP               69
Cylinders        30
Transmission      0
Drive Mode        0
MPG-H             0
MPG-C             0
Price             0
dtype: int64
```

```
In [14]: # Dropping the missing values.
         df = df.dropna()
         df.count()
```

```
Out[14]: Make              10827
         Model             10827
         Year              10827
         HP                10827
         Cylinders         10827
         Transmission      10827
         Drive Mode        10827
         MPG-H             10827
         MPG-C             10827
         Price             10827
         dtype: int64
```

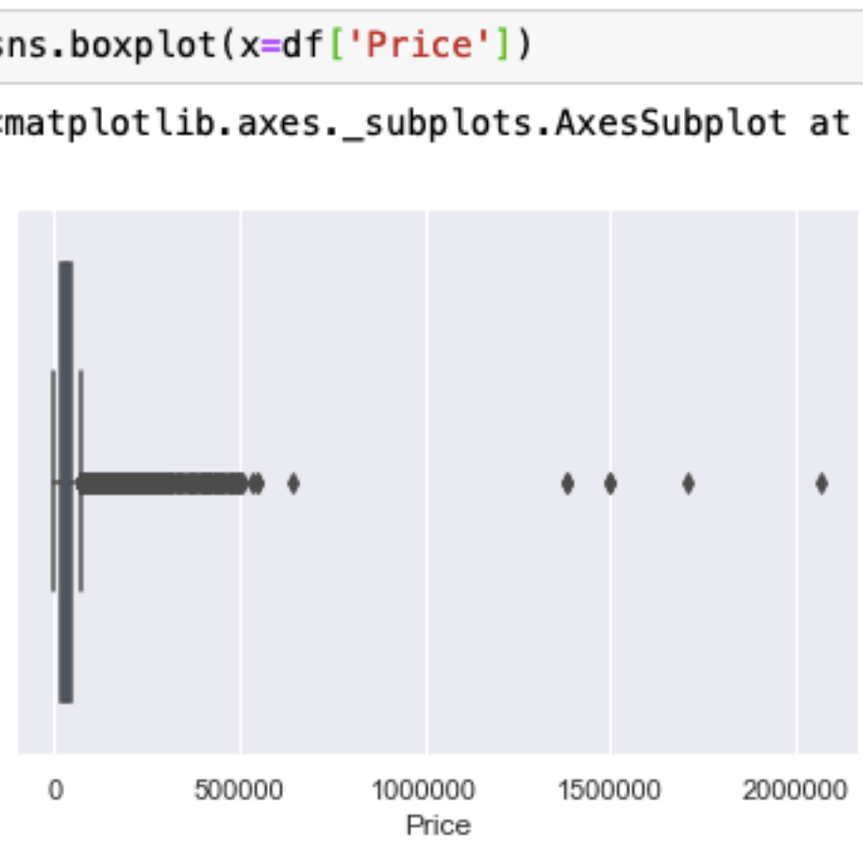
```
In [15]: # After dropping the values
         print(df.isnull().sum())
```

```
Make              0
Model             0
Year              0
HP                0
Cylinders         0
Transmission      0
Drive Mode        0
MPG-H             0
MPG-C             0
Price             0
dtype: int64
```

(적용) EDA과정

- Outlier 검출

```
In [17]: sns.boxplot(x=df['Price'])
Out[17]: <matplotlib.axes._subplots.AxesSubplot at 0x1187383c8>
```



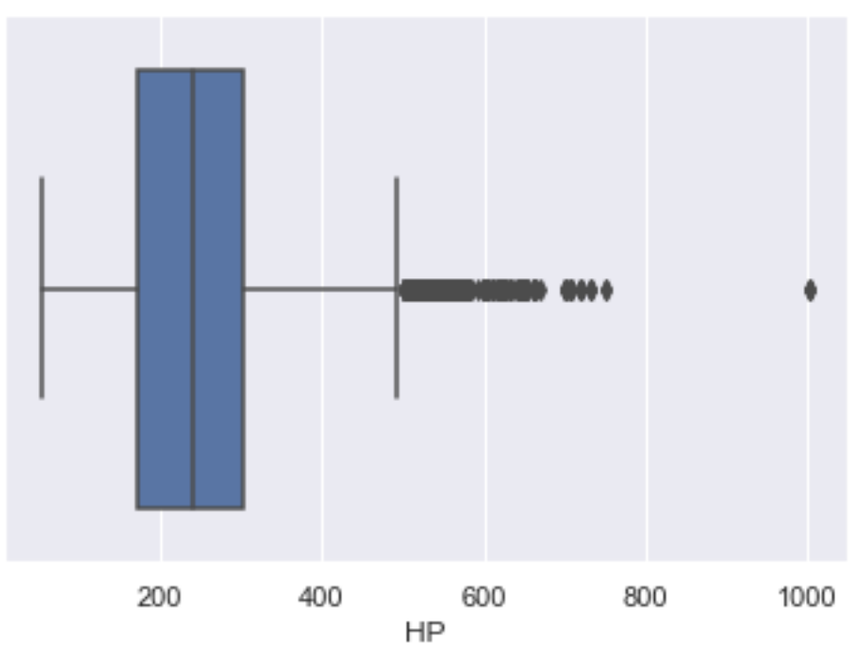
이상값 탐지

- 개별 데이터 관찰
- 통계값 활용
- 시각화 활용
- 기계학습 활용

(적용) EDA과정

- Outlier 검출

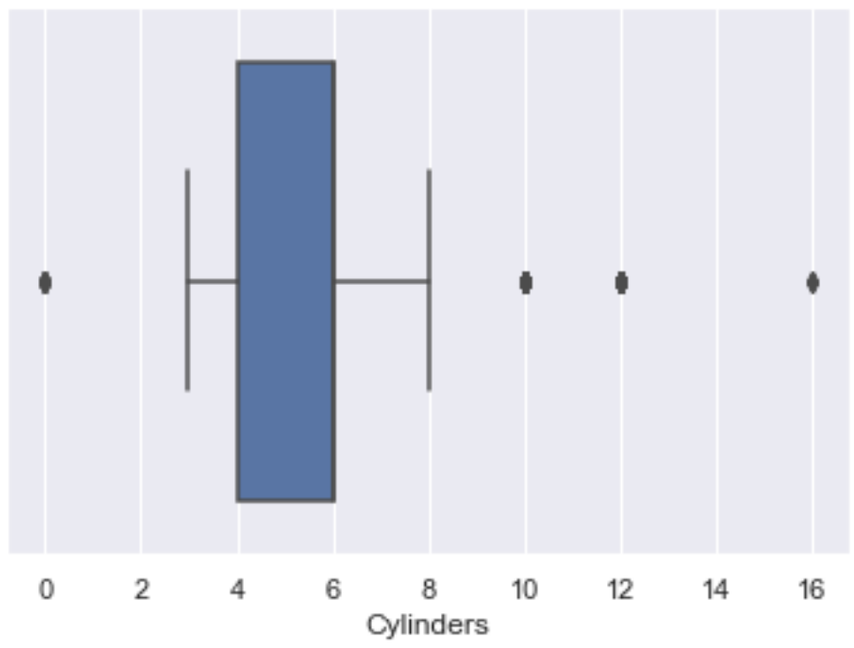
```
In [18]: sns.boxplot(x=df['HP'])
Out[18]: <matplotlib.axes._subplots.AxesSubplot at 0x118b20550>
```



(적용) EDA과정

- Outlier 검출

```
In [19]: sns.boxplot(x=df['Cylinders'])
Out[19]: <matplotlib.axes._subplots.AxesSubplot at 0x118c78f60>
```

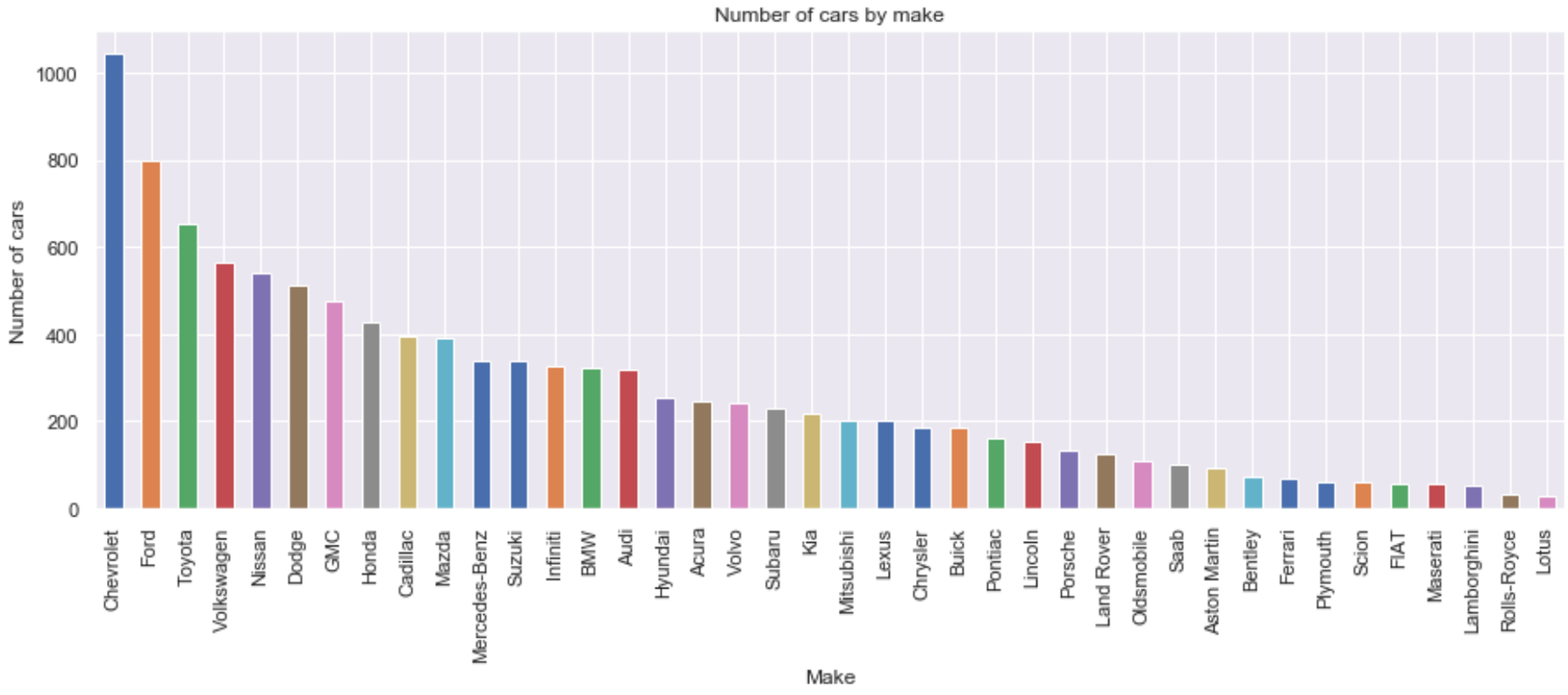


(적용) EDA과정

- 다양한 특징 플롯

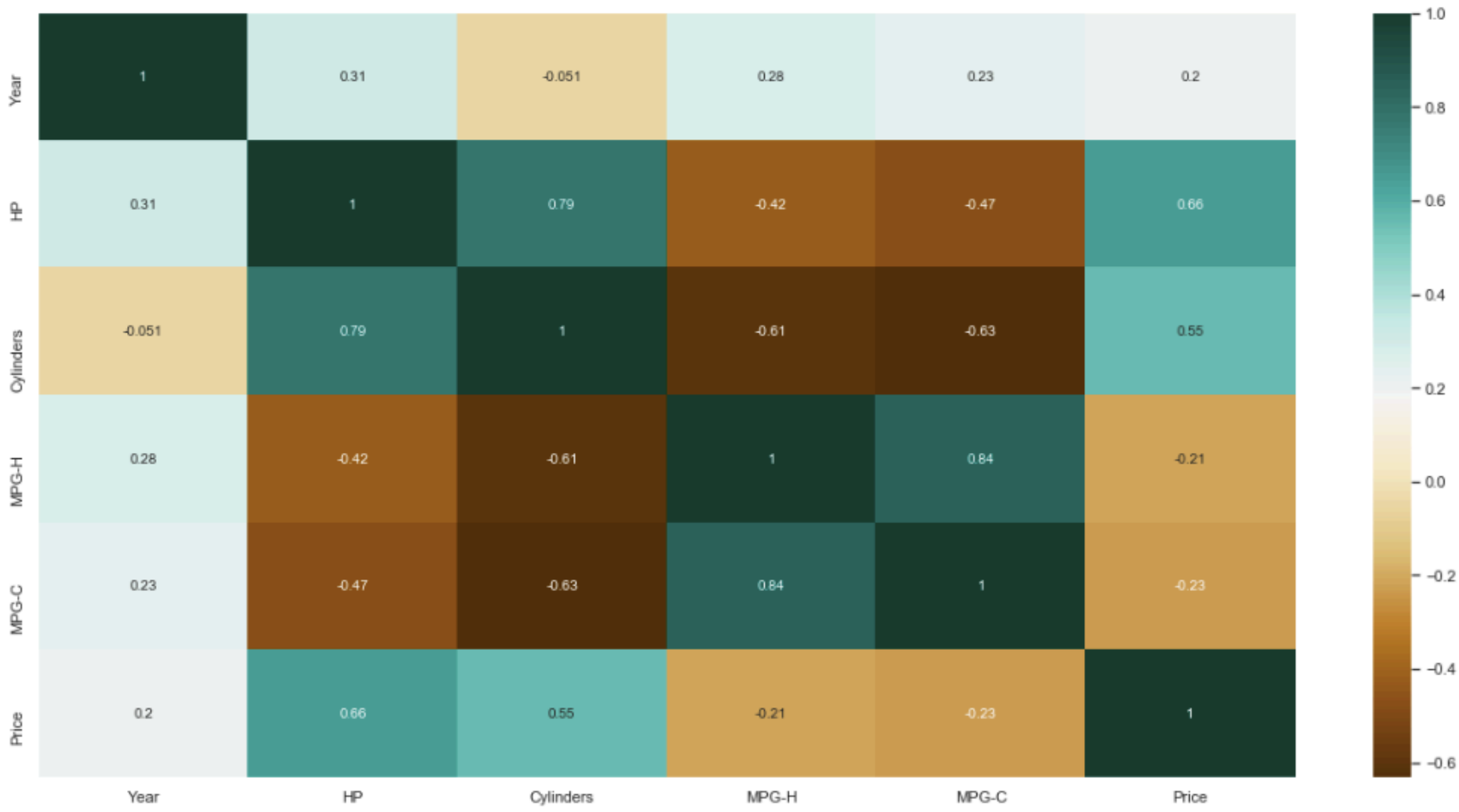
```
In [35]: df.Make.value_counts().nlargest(40).plot(kind='bar', figsize=(15,5))
         plt.title("Number of cars by make")
         plt.ylabel("Number of cars")
         plt.xlabel("Make")

Out[35]: Text(0.5, 0, 'Make')
```



(적용) EDA과정

- Heat Map



(적용) EDA과정

- 두 변수 간의 상호관계(correlation between two variables)

In [34]:
```
# Plotting a scatter plot
fig, ax = plt.subplots(figsize=(10,6))
ax.scatter(df['HP'], df['Price'])
ax.set_xlabel('HP')
ax.set_ylabel('Price')
plt.show()
```