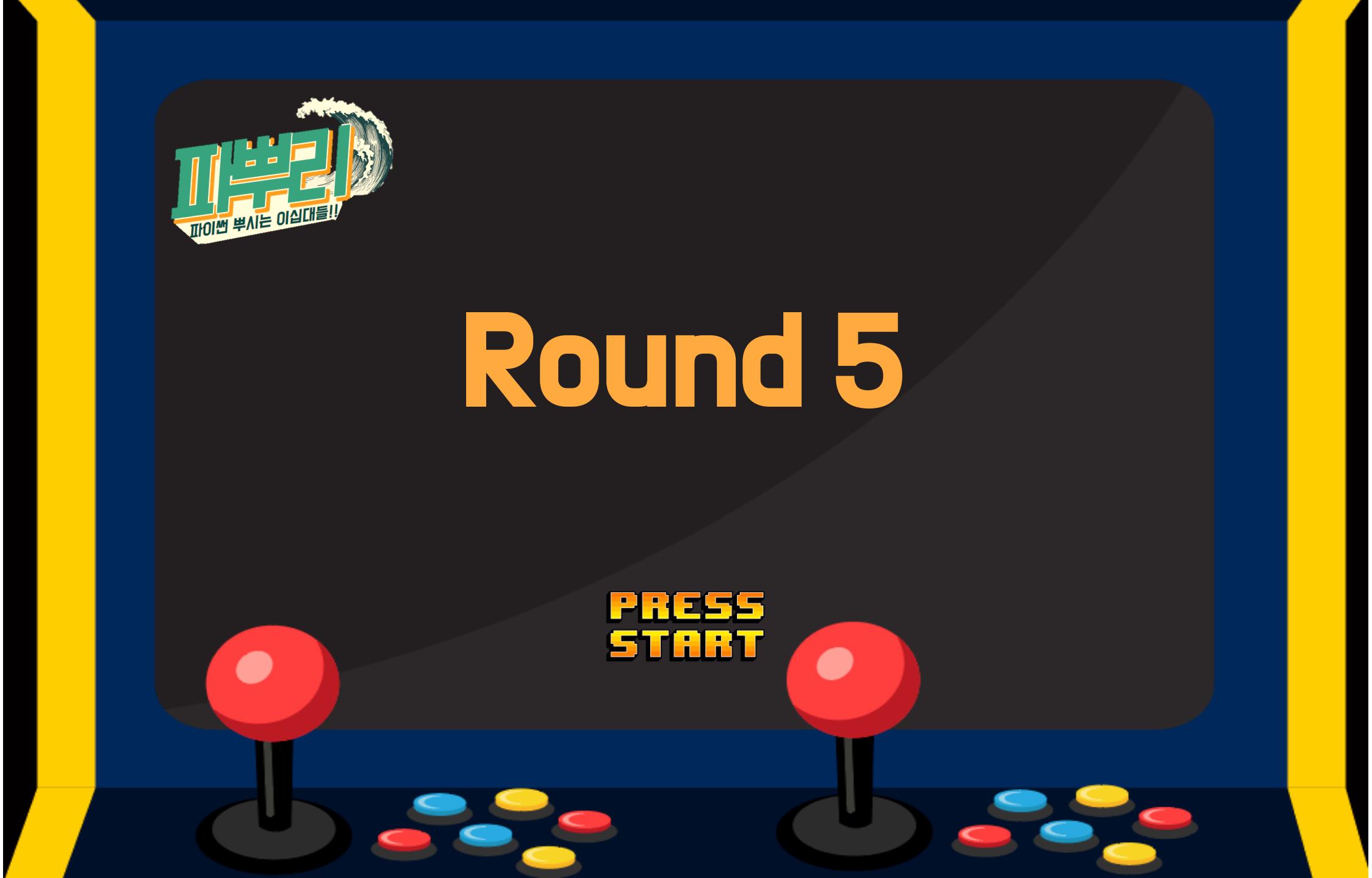

파뿌리
파이썬 뿌시는 이십대들!!
Round 5
PRESS
START

파뿌리
파이썬 뿌시는 이십대들!!

《 Round 5 》

- 데이터 사전처리 개요
- 결측치 제거 실습

New Assignment

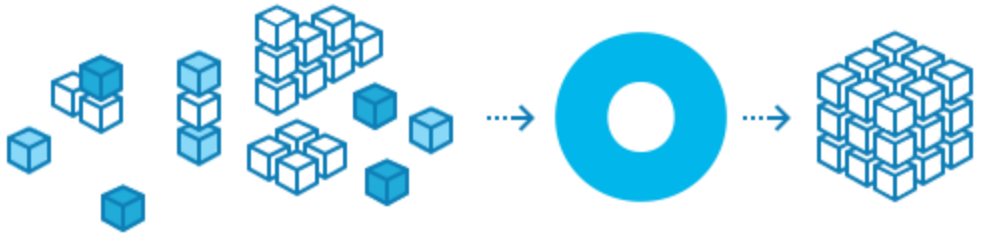
파뿌리
파이썬 뿌시는 이십대들!!

《 Round 5 》

- 데이터 사전처리 개요 《
- 결측치 제거 실습

파뿌리
파이썬 뿌시는 이십대들!!

데이터 사전처리의 과정



1. 데이터 셋 확인
2. 결측값 처리(missing value teatment)
3. 이상값 처리(outlier treatment)
4. Feature Engineering

파뿌리
파이썬 뿌시는 이십대들!!

1. 데이터셋 확인

a. 데이터셋 레이블링 & 변수 확인:

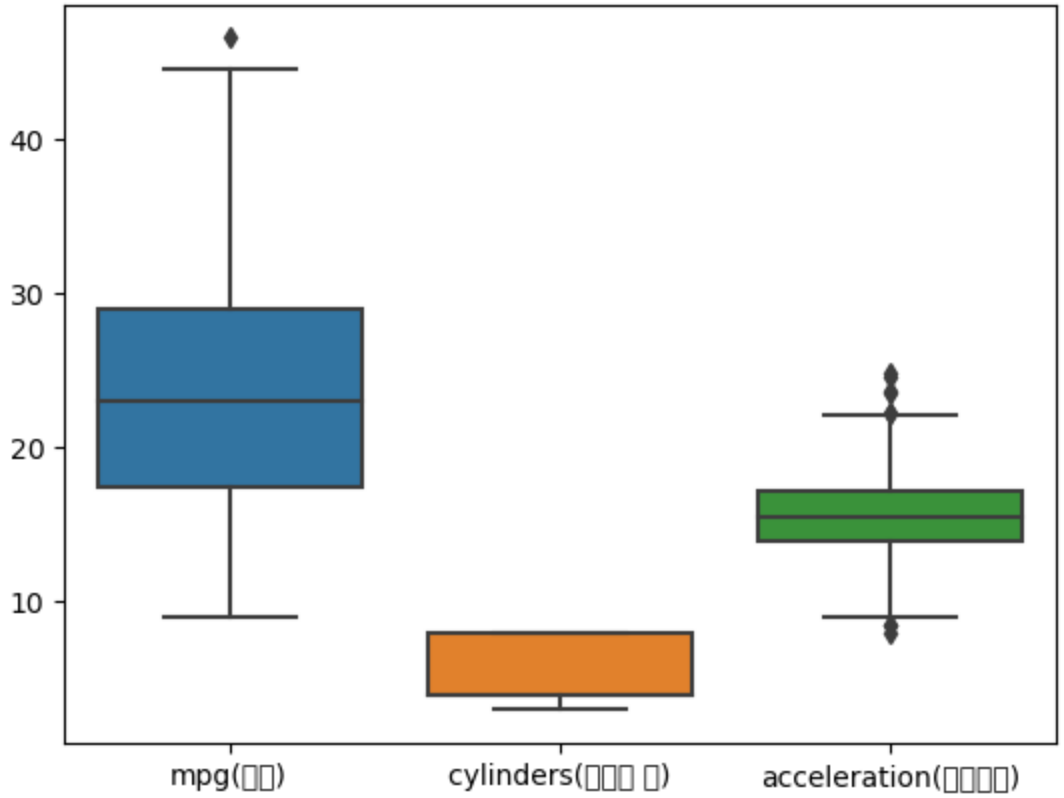
- 원본 데이터를 읽어서 head와 tail을 확인
- 유니크 식별값 인덱스 지정 및 컬럼 라벨링
- 독립/종속 변수의 정의, 변수의 유형, 변수의 데이터타입 등 확인 및 수정

```
384    38.0          4          91.0        67.00      1995          16.2          82          3                  datsun 310 gx
385    25.0          6         181.0       110.0       2945          16.4          82          1              buick century limited
386    38.0          6         262.0        85.00      3015          17.0          82          1    oldsmobile cutlass ciera (diesel)
387    26.0          4         156.0        92.00      2585          14.5          82          1          chrysler lebaron medallion
388    22.0          6         232.0       112.0       2835          14.7          82          1                ford granada l
389    32.0          4         144.0        96.00      2665          13.9          82          3                toyota celica gt
390    36.0          4         135.0        84.00      2370          13.0          82          1              dodge charger 2.2
391    27.0          4         151.0        90.00      2950          17.3          82          1              chevrolet camaro
392    27.0          4         140.0        86.00      2790          15.6          82          1              ford mustang gl
393    44.0          4          97.0        52.00      2130          24.6          82          2                  vw pickup
394    32.0          4         135.0        84.00      2295          11.6          82          1              dodge rampage
395    28.0          4         120.0        79.00      2625          18.6          82          1                ford ranger
396    31.0          4         119.0        82.00      2720          19.4          82          1                chevy s-10
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 397 entries, 0 to 396
Data columns (total 9 columns):
 #   Column                Non-Null Count  Dtype
---  ------                --------------  -----
 0   mpg(연비)             397 non-null    float64
 1   cylinders(실린더 수)  397 non-null    int64
 2   displacement(배기량)  397 non-null    float64
 3   horsepower(출력)      397 non-null    object
 4   weight(차중)          397 non-null    int64
 5   acceleration(가속능력)  397 non-null    float64
 6   model_year(출시년도)  397 non-null    int64
 7   origin_number(제조국 번호)  397 non-null    int64
 8   name(모델 명)         397 non-null    object
dtypes: float64(3), int64(4), object(2)
memory usage: 28.0+ KB
          mpg(연비)  cylinders(실린더 수)  displacement(배기량) horsepower(출력)   weight(차중)  acceleration(가속능력)  model_year(출시년도)  origin_number(제조국 번호)  name(모델 명)
count   397.000000       397.000000        397.000000          397   397.000000        397.000000        397.000000            397.000000         397
unique         NaN              NaN               NaN           94          NaN               NaN               NaN                   NaN         305
top            NaN              NaN               NaN        150.0          NaN               NaN               NaN                   NaN  ford pinto
freq           NaN              NaN               NaN           22          NaN               NaN               NaN                   NaN           6
mean     23.528463         5.448363        193.139798          NaN  2969.080605         15.577078         76.025189              1.574307         NaN
std       7.820926         1.698329        104.244898          NaN   847.485218          2.755326          3.689922              0.802549         NaN
min       9.000000         3.000000         68.000000          NaN  1613.000000          8.000000         70.000000              1.000000         NaN
25%      17.500000         4.000000        104.000000          NaN  2223.000000         13.900000         73.000000              1.000000         NaN
50%      23.000000         4.000000        146.000000          NaN  2800.000000         15.500000         76.000000              1.000000         NaN
75%      29.000000         8.000000        262.000000          NaN  3609.000000         17.200000         79.000000              2.000000         NaN
max      46.600000         8.000000        455.000000          NaN  5140.000000         24.800000         82.000000              3.000000         NaN
```

1. 데이터셋 확인

b. 단변수 분석

- 변수에 대한 통계값 확인
  Histogram이나 Boxplot을 사용해서
  평균, 최빈값, 중간값, 산포도 등 확인



1. 데이터셋 확인

c. 이변수 분석

- 변수 2개 간의 관계를 분석하는 단계
- 오른쪽 그림 참고

	그래프	분석 방법
연속형 X 연속형	• (추세선이 있는) Scatter plot	• Correlation 분석 (두 변수 간 상관관계 여부)
범주형 X 범주형	• 누적막대그래프 • 100%기준 누적 막대 그래프	• Chi-Square분석 (두 변수가 독립적인지 여부)
범주형 X 연속형	• 누적막대그래프 • 범주 별 Histogram	범주의 종류에 따라 • 2개: T-test/Z-test • 3개 이상: ANOVA (집단 별 평균 차가 유의한지 여부)

파뿌리
파이썬 뿌시는 이십대들!!

2. 결측값 처리(Missing value treatment)

결측값이 있는 상태로 모델을 만들게 될 경우
변수간의 관계가 왜곡되어 모델의 정확성이 떨어지기 때문

결측값의 종류 :

"", "?", NaN, "-" 등등

※파이썬에서 NaN type은 numpy를 통해서 사용 가능

파뿌리
파이썬 뿌시는 이십대들!!

2. 결측값 처리(Missing value treatment)

결측값 처리 방법의 종류 :

a. 삭제 : 간편하나 모델의 유효성이 낮아짐

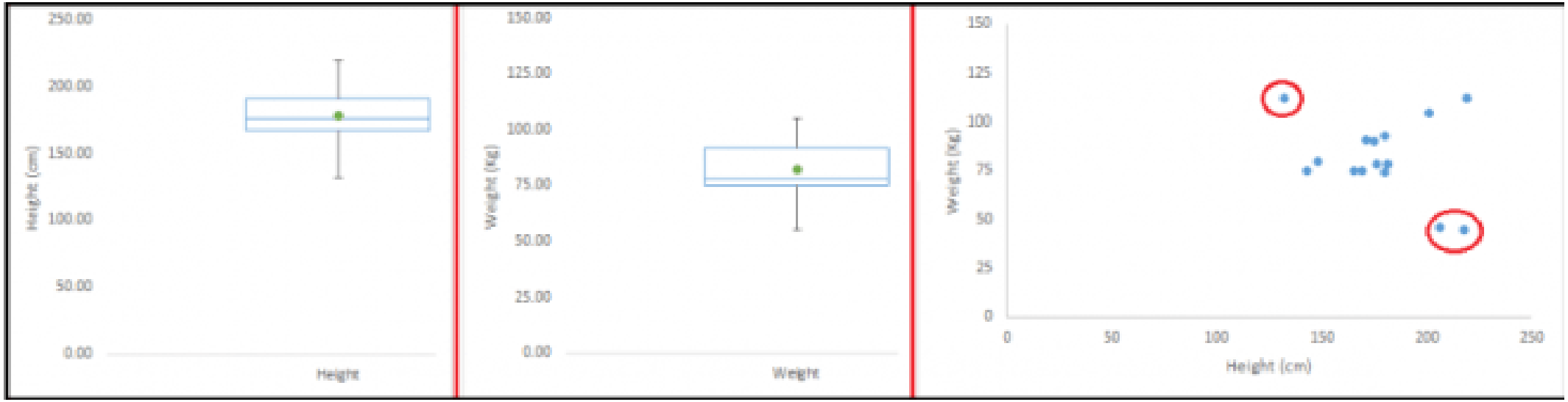
b. 대체 : 다른 관측치들의 평균, 최빈값, 중간값으로 대체하는 것

c. 예측 값 삽입 :
결측값이 없는 관측치를 트레이닝 데이터로 사용해서
결측값을 예측하는 모델을 만들어 결측값을 예측하는 방법

3. 이상값 처리(Outliar treatment)

가장 먼저 이상값을 찾아내야 함.

일반적으로 하나의 변수는 boxplot이나 Histogram
두 개의 변수 간 이상값은 Scatter plot 사용



파뿌리
파이썬 뿌시는 이십대들!!

3. 이상값 처리(Outliar treatment)

이상 값을 찾았다면…

a. 단순 삭제

- Human error에 의한 경우 해당 관측치를 삭제하면 됨.
- ex) 단순 오타, 주관식 설문 등의 비현실적 응답, 처리과정에서의 오류 등

파뿌리
파이썬 뿌시는 이십대들!!

3. 이상값 처리(Outliar treatment)

이상 값을 찾았다면…

b. 대체

- 평균, 중간값 , 중앙값 등으로 대체

- 결측값과 유사하게 다른 변수들을 사용해서 예측모델을 만들고,
- 이상값을 예측한 후 해당 값으로 대체

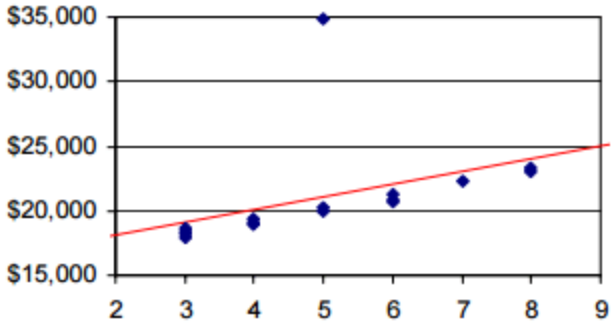
- 이상값이 자연발생한 경우 삭제/대체를 통해 모델을 만들면
현상/예측을 잘 설명할 수 없을 수도 있음.

3. 이상값 처리(Outliar treatment)

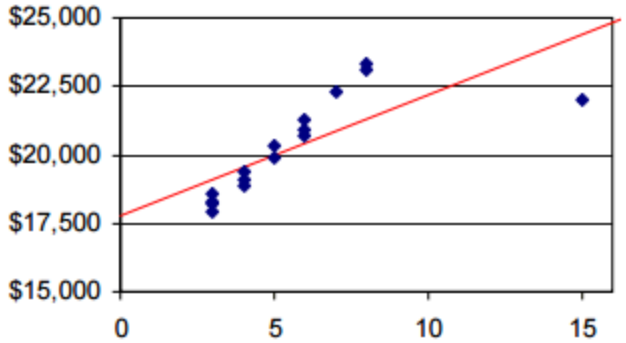
이상 값을 찾았다면…

c. 이상치가 자연발생 했을 경우의 방법

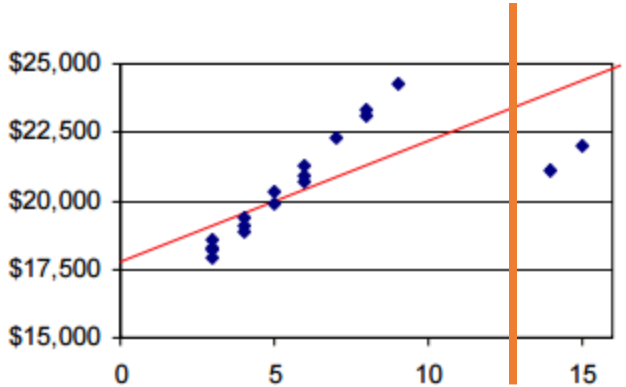
ex) 년차별 소득 수준



전문직종 종사 여부를 **변수화**
(종속변수가 outliar일 경우)



년차의 범위를 10년까지로 **리샘플링**
(종속변수, 독립변수가 outliar)

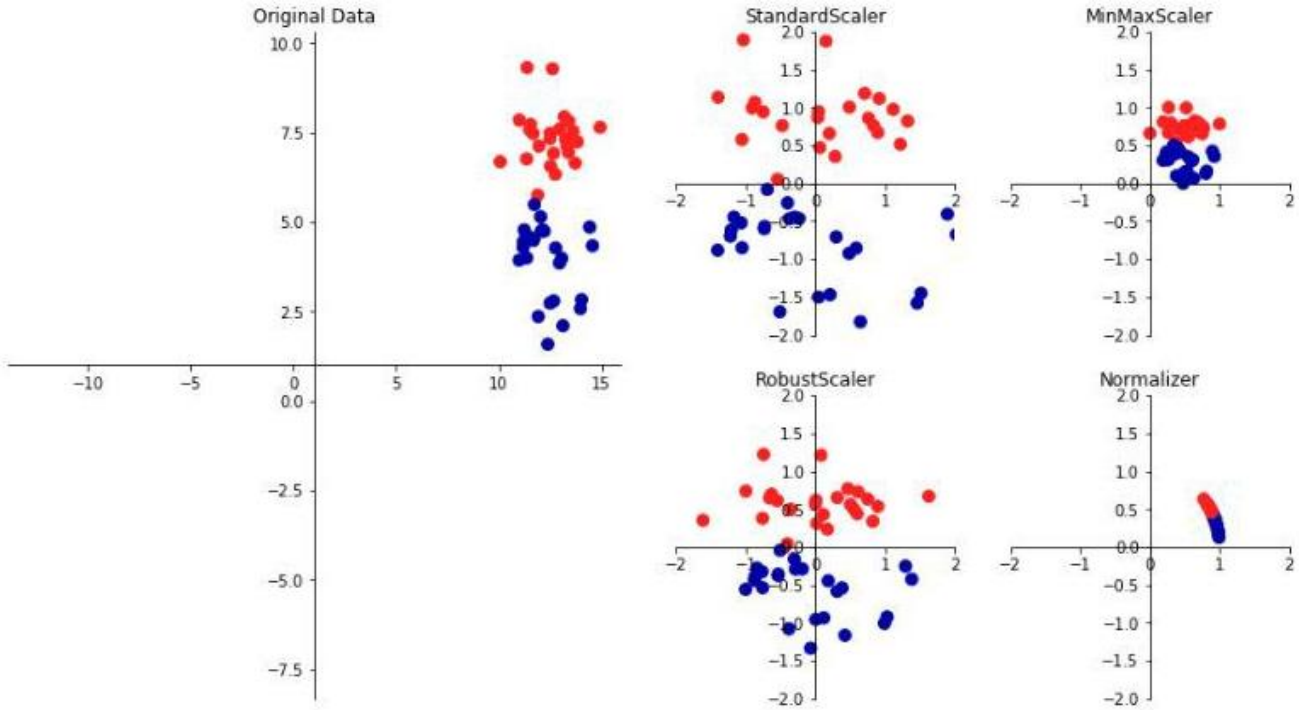


케이스 분리 해석
(특정 경향의 outliar가 여러 개일 경우)

4. Feature Engineering

: 기존의 변수를 사용해서 정보를 추가하여 데이터를 더 유용하게 하는 과정

a. Scaling : 데이터가 편향되어 있거나, 너무 크거나 작은 경우, 변수들의 상관관계가 잘 보이지 않을 경우 모델의 성능을 위해 사용



4. Feature Engineering

: 기존의 변수를 사용해서 정보를 추가하여 데이터를 더 유용하게 하는 과정

b. Transform : 기존에 존재하는 변수의 성질을 이용해 다른 변수를 창조

	1990	1991	1992	1993	---	2016
'south'	1077	1186	1310	1444		5404
'north'	277	266	247	221		239

↓

	1990	1991	1992	1993	---	2016
'south'	1077	1186	1310	1444		5404
'north'	277	266	247	221		239
'president'	'14'	'14'	'14'	'14'		'18'

4. Feature Engineering

: 기존의 변수를 사용해서 정보를 추가하여 데이터를 더 유용하게 하는 과정

c. Binning : 연속형 변수를 다수의 범주형 변수로 변환

	'연봉(만)'
박수현	5100
변준현	4300
이수빈	3900

→

	'3000~4000'	'4000~5000'	'5000~6000'
박수현	'0'	'0'	'1'
변준현	'0'	'1'	'0'
이수빈	'1'	'0'	'0'

d. Dummy : 범주형 변수를 다수의 연속형 변수 형태로 변환

	'학년'
박수현	'1'
변준현	'2'
이수빈	'3'

→

	'dummy_grade_1'	'dummy_grade_2'
박수현	1	0
변준현	0	1
이수빈	0	0

파뿌리
파이썬 뿌시는 이십대들!!

4. Feature Engineering

Dummy에 대한 추가설명

ex) 만약 학생의 학년, 평소 공부시간, 1학기 평점 으로 2학기의 평점의 관계를 모델링한다면…

	'학년'	'평소 공부시간'	'1학기 평점'	'2학기 평점'
'박수현'	'1'	1.7	3.44	3.28
'변준현'	'2'	1.45	3.89	3.74
'이수빈'	'3'	2.1	3.71	3.97
…				

학년이 범주형 변수이기 때문에 관계를 추정하기 까다로워진다.

4. Feature Engineering

Dummy에 대한 추가설명

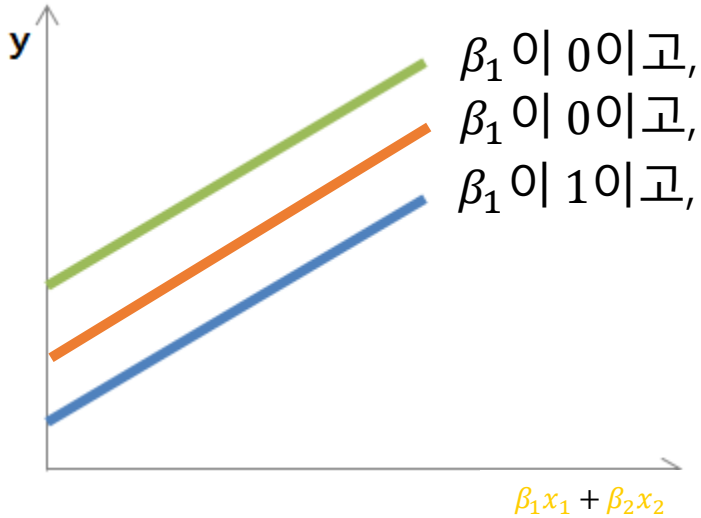
ex) 만약 학생의 학년, 평소 공부시간, 1학기 평점 으로 2학기의 평점의 관계를 모델링한다면…
범주형인 학년을 가변수 dummy로 변환하여 각종 연속형 분석에 이용할 수 있다.

	‘학년’
박수현	‘1’
변준현	‘2’
이수빈	‘3’

→

	‘dummy_grade_1’	‘dummy_grade_2’
박수현	1	0
변준현	0	1
이수빈	0	0

해당 회귀식 : $y = 1.42 + (-0.21)x_1 + (-0.12)x_2 + 0.78x_3 + 0.31x_4$



β_1이 0이고, β_2가 0일 때,
β_1이 0이고, β_2가 1일 때,
β_1이 1이고, β_2가 0일 때,

즉, 더미변수는 해당 변수의 효과를 0 or 상수로 변환

회귀선 기울기는 변화가 없고, Y절편에만 영향을 줌.

파뿌리
파이썬 뿌시는 이십대들!!

《 Round 5 》

- 데이터 사전처리 개요 - complete
- 결측치 제거 실습 《

파뿌리
파이썬 뿌시는 이십대들!!

```python
import pandas as pd
import numpy as np
pd.set_option('display.max_rows', 400)
pd.set_option('display.max_columns', 20)
pd.set_option('display.width', 1000)

# 1. dataset 확인
mpg_df = pd.read_csv("auto-mpg.csv")
```

source code :

https://github.com/KGJsGit/Python_Breakers/blob/master/source_code/missingValues.py

```python
import pandas as pd
import numpy as np
pd.set_option('display.max_rows', 400)
pd.set_option('display.max_columns', 20)
pd.set_option('display.width', 1000)

# 1. dataset 확인
mpg_df = pd.read_csv("auto-mpg.csv")

# 1-a. data의 대략적인 모양새 파악 head는 상위 로우 5개, tail은 하위 로우 5개
print(mpg_df.head(),"\n\n")
print(mpg_df.tail(),"\n\n")

# 1-a. 컬럼 라벨링
mpg_df.columns = ['mpg(연비)', 'cylinders(실린더 수)', 'displacement(배기량)', 'horsepower(출력)', 'weight(차중)', 'acceleration(가

# 1-a. 요약통계량 및 데이터정보 확인 (출력에 noise 존재 및 출시년도, 제조국 번호가 범주형 아닌 연속형임 확인)
print(mpg_df.info())
print(mpg_df.describe(include = "all"),"\n\n")
print(mpg_df,"\n\n")
```

파뿌리
파이썬 뿌시는 이십대들!!

결측값 판단

"" : 공백은 보통 결측값으로 분류

0, '0' : 보통은 숫자 혹은 문자 0이나, 데이터에 따라 결측값으로 분류

"?", "-", … : 결측값으로 분류

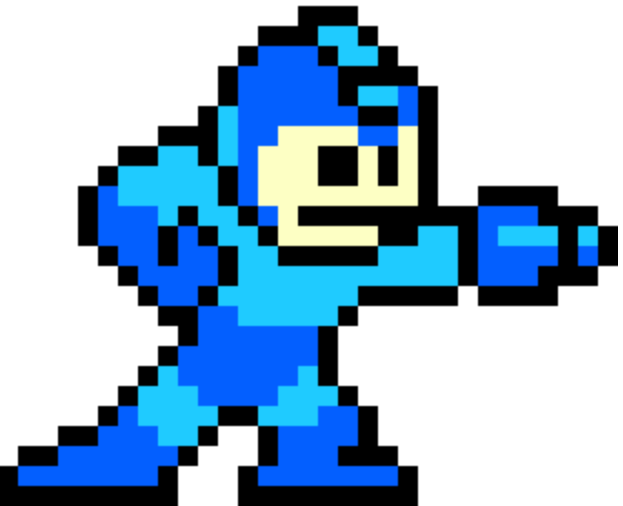
NaN(None): 코드에선 numpy.nan을 써줌. 대표적인 결측값

※ R에서는 결측값의 종류가 더 다양함.

```
# 1-a. 출시년도 및 제조국 번호를 astype() 메소드로 int -> object 변환
mpg_df[['model_year(출시년도)', 'origin_number(제조국 번호)']] = mpg_df[['model_year(출시년도)', 'origin_number(제조국 번호)']].ast
print(mpg_df.describe(include = "all"), "\n\n")

# 2. 결측값 처리
# 2-b. 결측값 대체 - 기존 DF를 복제하고 '출력 column'에 '?'가 있는 row을 제거하고 출력의 평균을 구해서 hpMean에 삽입
mpg_df_ex = mpg_df
mpg_df_ex['horsepower(출력)'].replace('?', np.nan, inplace = True)
mpg_df_ex.dropna(subset=['horsepower(출력)'], axis = 0, inplace = True)
mpg_df_ex['horsepower(출력)'] = mpg_df_ex['horsepower(출력)'].astype('float')
hpMean = round(mpg_df_ex['horsepower(출력)'].mean(),3)
print(hpMean,"\n\n")

# 2-b. 결측값 대체 - ?를 제외한 출력의 평균을 결측치들에 넣어준다.
mpg_df_ex['horsepower(출력)'].replace('?', hpMean, inplace = True)
mpg_df_ex.dropna(subset=['horsepower(출력)'], axis = 0, inplace = True)
mpg_df_ex['horsepower(출력)'] = mpg_df_ex['horsepower(출력)'].astype('float')

# 2-b. 처리된 데이터셋 확인
print(mpg_df_ex.info())
print(mpg_df_ex.describe(include = "all"))
```

파뿌리
파이썬 뿌시는 이십대들!!

WARNING



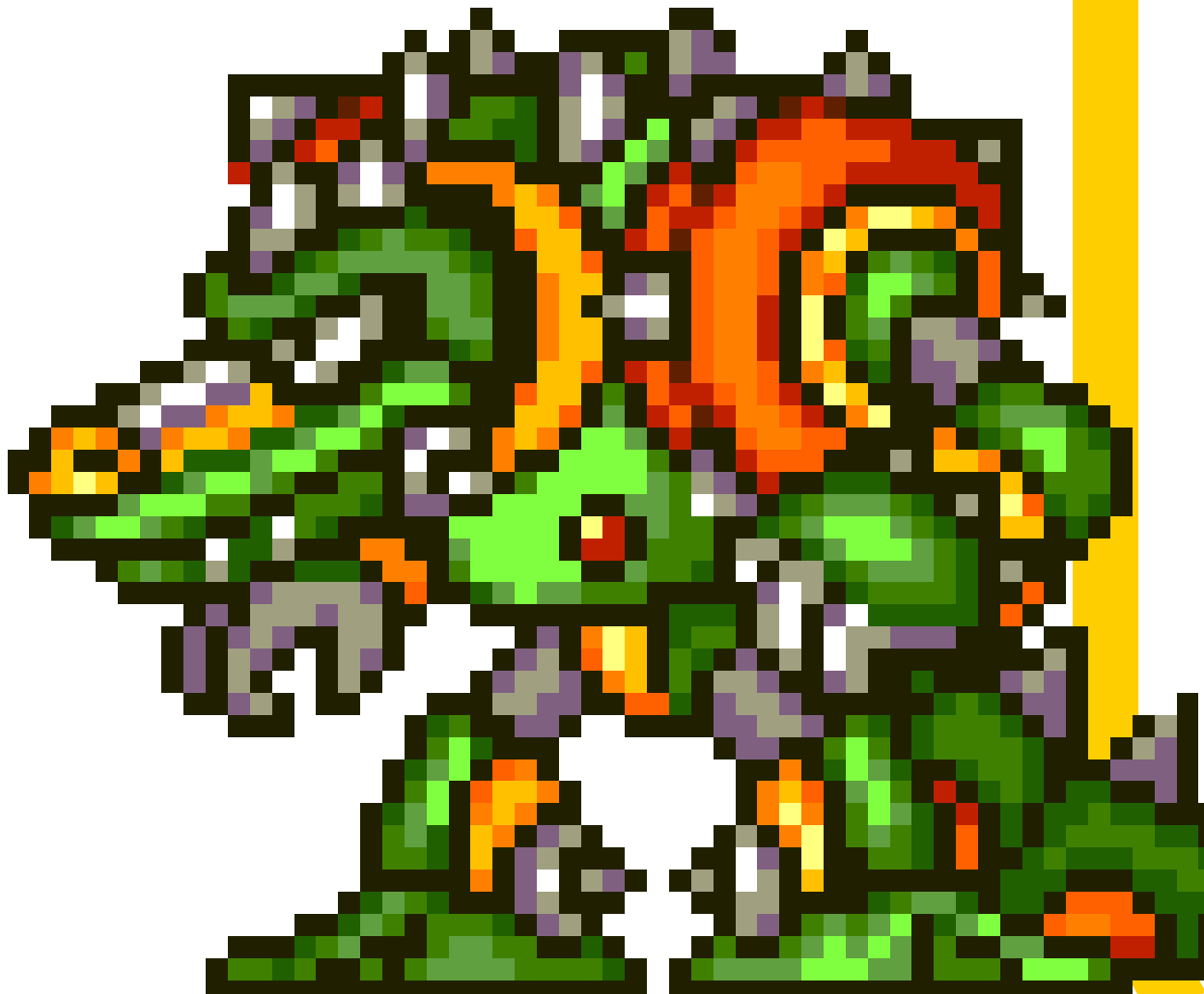
파뿌리
파이썬 뿌시는 이십대들!!

WARNING

《 QUEST 》

Titanic 데이터셋 결측치 제거

의미 있는 관계 찾아서 시각화



파뿌리
파이썬 뿌시는 이십대들!!

NEXT STAGE