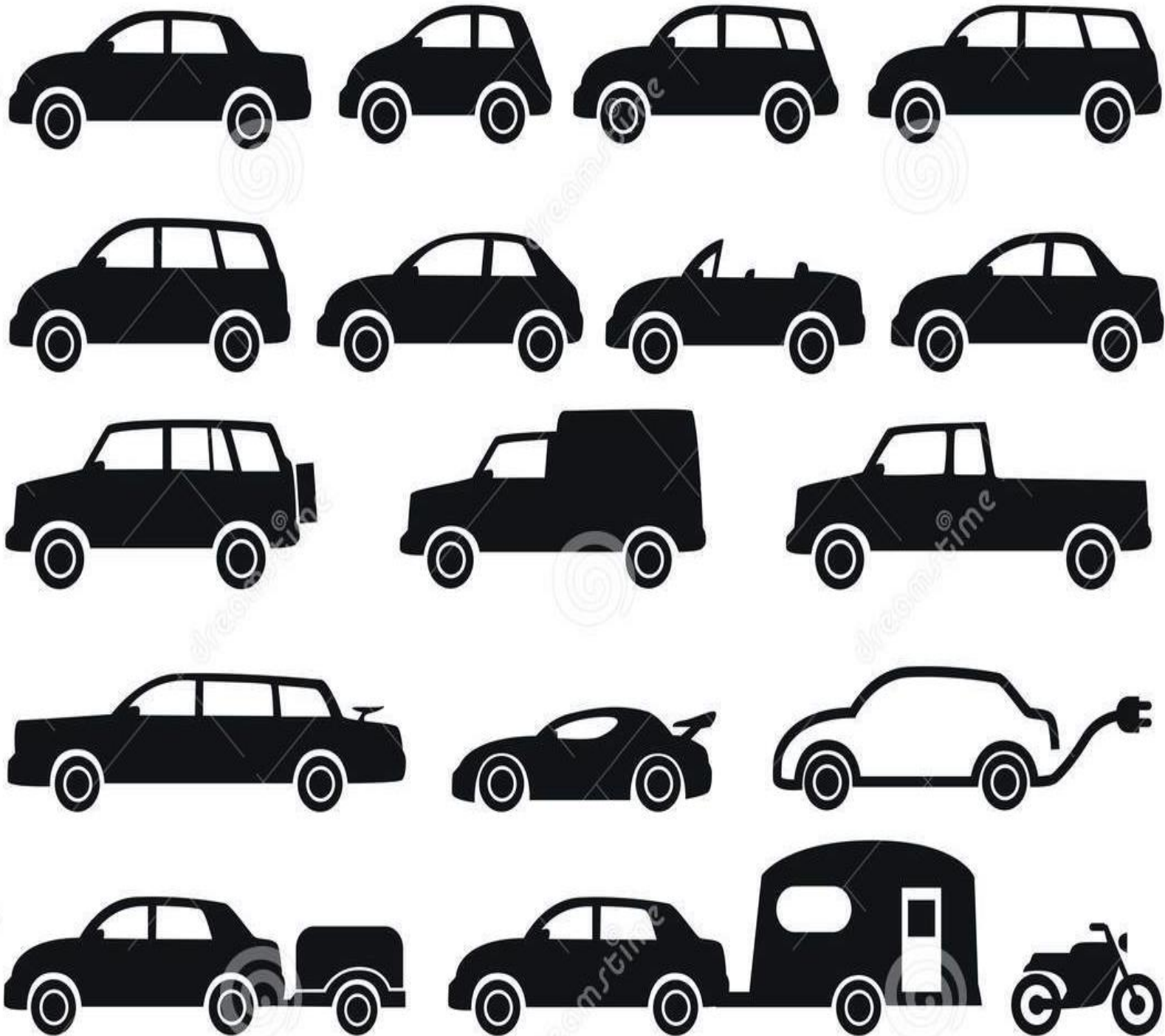


자동차 판매요인

상관관계 분석



[00] 목차

(1) 연구목적

= 자동차 판매량의 내부 요인과 외부 요인

(2) 데이터전처리

= 크롤링 오류, DataFrame 오류 해결

(3) 기술 통계

= 요인 별 상관관계분석

(4) 한계와 향후 과제

[01] 연구목적

<내부요인>:

차종 / 최대 가격 / 최소 가격
최대 연비 / 최소 연비
출시일 / 브랜드 / 국가

<외부요인>:

원-달러 환율 /
유가(WTI) /
금리 / 코스피 지수

연구목적:

차량의 판매를 결정하는 데 있어 영향을 미치는 요인 분석

[02] 데이터 전처리

<내부요인>:

다나와 자동차

- 월별 판매량, 차량 데이터 파싱

```
url = 'http://auto.danawa.com/newcar'
query = {
    'Work': 'record',
    'Brand': nations[nation],
    'Month': '{}-{}-00'.format(year, month),
    'MonthTo': ''
}
```

```
print(nation)
print(query['Month'])
```

```
header = {'user-agent': 'Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/81.0.4044.138 Safari/537.36',
          'referer': 'https://comic.naver.com/comment/comment.nhn?titleId=183559&no=485'}
resp = requests.get(url, query, headers=header)
```

```
[135] 1 df_whole_ = pd.concat([df_2018_, df_2019_, df_2020_], axis=0)
      2 df_whole_.shape
```

(5692, 11)

국산 모델 Top10

1 그랜저 13,416 1,584▼

2 신형 쏠렌토 9,297 34▲

해외 모델 Top10

1 E-Class 2,193 574▲

2 5 Series 1,953 23▼

월 선택 2020년 05월

기간 선택 2020년 05월 ~ 2020년 06월 조회

	모델	판매량	점유율	전월대비	전년대비
<input type="checkbox"/> 1	그랜저	13,416	9.2%	15,000 1,584▼	0 13,416▲
<input type="checkbox"/> 2	신형 쏠렌토	9,297	6.4%	9,263 34▲	0 9,297▲
<input type="checkbox"/> 3	뉴 아반떼	8,849	6.1%	8,249	0 8,969▲
<input type="checkbox"/> 4	뉴 아반떼	8,849	6.1%	8,249	0 8,969▲
<input type="checkbox"/> 5	뉴 아반떼	8,849	6.1%	8,249	0 8,969▲
<input type="checkbox"/> 6	뉴 아반떼	8,849	6.1%	8,249	0 8,969▲
<input type="checkbox"/> 7	뉴 아반떼	8,849	6.1%	8,249	0 8,969▲
<input type="checkbox"/> 8	뉴 아반떼	8,849	6.1%	8,249	0 8,969▲
<input type="checkbox"/> 9	뉴 아반떼	8,849	6.1%	8,249	0 8,969▲
<input type="checkbox"/> 10	뉴 아반떼	8,849	6.1%	8,249	0 8,969▲

신차

블랙 포레스트(MB9)

그랜저

2,725~4,432만 원

최저가 월 244,090원

2019.11 ~ 현재

복합연비 7.4~16.2 km/ℓ · 준대형 · 휘발유, LPG

♥ 54명 | ★★★★★ (2명)

브랜드 정보

신차 견적내기

렌트/리스 가격비교

[02] 데이터 전처리

<외부요인>:

네이버 금융

- 일자별 KOSPI, WTI, 원/달러 환율 파싱

한국은행

- 금리 파싱



환전 고시 환율				▶ 더보기
미국USD	1,199.50	▼ 1.50		
일본JPY (100엔)	1,112.55	▲ 1.18		
유럽연합EUR	1,359.81	▲ 5.14		
중국CNY	169.39	▼ 0.10		

유가				▶ 더보기
두바이유(06.09)	40.82	▲ 0.13		
WTI(06.09)	38.94	▲ 0.75		
휘발유(06.10)	1,307.71	▲ 1.63		
고급휘발유(06.10)	1,601.21	▲ 0.75		

국제 시장 환율				▶ 더보기
일본 엔/달러(06.09)	107.7100	▼ 0.86		
달러/유로(06.09)	1.1353	▲ 0.01		
달러/파운드(06.09)	1.2722	▲ 0.00		
달러인덱스(06.09)	96.3200	▼ 0.28		

금 시세				▶ 더보기
국제금(06.09)	1,714.70	▲ 16.40		
국내금(06.10)	66,083.34	▲ 259.32		

단위 | 국제:달러/트로이온스, 국내:원/g

금리				▶ 더보기
CD(91일)(06.09)	0.79	—		
콜금리(06.08)	0.46	▼ 0.01		
국고채(3년)(06.09)	0.86	▼ 0.04		
회사채(3년)(06.09)	2.20	▼ 0.02		

원자재				▶ 더보기
구리(06.09)	5,680.50	▲ 21.50		
납(06.09)	1,726.00	▼ 12.00		
천연가스(06.09)	1.77	▼ 0.02		
옥수수(06.09)	327.50	▼ 6.25		

단위 | 구리:납/달러/톤, 천연가스: 달러/MMBtu, 옥수수: 센트/부셸

```
import requests
from bs4 import BeautifulSoup
import re
from tqdm.notebook import tqdm
import pandas as pd

# 네이버 경제 페이지에서 BeautifulSoup이동하여 KOSPI 지수와 날짜 크롤링
# 같은 방식으로 WTI(유가지수)와, 원/달러 환율 지수를 크롤링
juga={
    'day': [],
    'price': []
}

c_tag=[]
for j in tqdm(range(1,300)):
    url = 'https://finance.naver.com/sise/sise_index_day.nhn?code=KOSPI&page='+str(j)
    resp = requests.get(url)
    text = resp.text
    soup = BeautifulSoup(resp.content, 'lxml')
    a_tag = soup.select('div.box_type_m table tr td.date')

    b_tag = re.findall('number_1.+?>([0-9,\.]+)</td>',text)

    for i in range(len(b_tag)):
        if i%3==0:
            c_tag.append(b_tag[i])

    for a,b in zip(a_tag,c_tag):
        juga['day'].append(a.text.replace(' ','-'))
        juga['price'].append(b.replace(',',''))

c_tag=[]
file = pd.DataFrame(juga)
file.to_csv('juga.csv', index=False) #os파일로 저장
```

```
# 네이버 경제 페이지에서 BeautifulSoup이동하여 WTI 지수와 날짜 크롤링
yuga={
    'day': [],
    'yuga': []
}

for i in tqdm(range(1,300)):
    url = 'https://finance.naver.com/marketindex/worldDailyQuote.nhn?marketindexCd=01LCL&fdtc=2&page='+str(i)
    resp = requests.get(url)
    text = resp.text

    soup = BeautifulSoup(resp.content, 'lxml')
    a_tag = soup.select('table tr td.num')
    day_tag = soup.select('table tr td.date')

    for a in day_tag:
        yuga['day'].append(a.text.strip().replace(' ','-'))
    for j in range(len(a_tag)):
        if j%3==0:
            yuga['yuga'].append(a_tag[j].text.strip())

file = pd.DataFrame(yuga)
file.to_csv('yuga.csv', index=False)

# 네이버 경제 페이지에서 BeautifulSoup이동하여 WTI 지수와 날짜 크롤링
dala={'day': [], 'dala': []}
for i in tqdm(range(1,231)):
    url = 'https://finance.naver.com/marketindex/exchangeDailyQuote.nhn?marketindexCd=FX_USDKRW&page='+str(i)
    resp = requests.get(url)

    soup = BeautifulSoup(resp.content, 'lxml')
    a_tag = soup.select('table tr td.num')
    day_tag = soup.select('table tr td.date')

    for a in day_tag:
        dala['day'].append(a.text.strip().replace(' ','-'))

    for j in range(len(a_tag)):
        if j%2==0:
            dala['dala'].append(a_tag[j].text.strip())

file = pd.DataFrame(dala)
file.to_csv('dala.csv', index=False)
```

[02] 데이터 전처리

문제:
가격과 연비 데이터의 경우
범위형 데이터와 단일형 데이터가 혼합

해결:
len() 값의 차이를 이용해 범위형 데이터와
단일형 데이터를 구분.

범위형 데이터의 경우, 인덱스를 통해
최고, 최저를 다른 리스트에 저장

단일형 데이터의 경우, 같은 값을
다른 리스트에 저장

```
# 가격(최고, 최저)
price_tag = soup_.select('div.price_title span.num')
# print(price_tag[0].text)
price = re.findall('[0-9,]+', price_tag[0].text)
# print(price[0], price[1])
if len(price) == 2 :
    price_low = price[0]
    price_high = price[1]
elif len(price) == 1 :
    price_low = price[0]
    price_high = price[0]
```

```
# 연비
fuel_efficiency_tag = soup_.select('div.spec span.num')
if len(fuel_efficiency_tag) == 0 :
    fuel_efficiency_low = ''
    fuel_efficiency_high = ''
else :
    # print(fuel_efficiency_tag[0].text)
    fuel_efficiency = re.findall('[0-9,]+', fuel_efficiency_tag[0].text)
    if len(fuel_efficiency) == 2:
        fuel_efficiency_low = fuel_efficiency[0]
        fuel_efficiency_high = fuel_efficiency[1]
    elif len(fuel_efficiency) == 1:
        fuel_efficiency_low = fuel_efficiency[0]
        fuel_efficiency_high = fuel_efficiency[0]
```

그랜저 저공해

2,725~4,432만 원

신차 견적내기

최저가 월 244,090원

렌트/리스 가격비교

2019.11 ~ 현재

복합연비 7.4~16.2 km/ℓ · 준대형 · 휘발유, LPG

♡ 54명 | ★★★★★ (2명)

브랜드 정보

쏘나타 뉴 라이즈 저공해

1,860~2,293만 원

신차 견적내기

상품 준비 중

렌트/리스 가격비교

2017.03 ~ 현재

복합연비 9.4 km/ℓ · 중형 · LPG

♡ 37명 | ★★★★★ (0명)

All New Explorer

5,920만원

신차 견적내기

최저가 월 774,500원

렌트/리스 가격비교

2019.11 ~ 현재

연비인증중 · 대형 SUV · 휘발유

♡ 14명 | ★★★★★ (0명)

브랜드 정보

[02] 데이터 전처리

문제:

판매량, 가격과 같은 숫자형 데이터 값에
, ' 가 추가됨으로 인해서 Dtype이 STR로 변화

해결:

Stack().str.replace(',','').unstack().astype()

해당 데이터 값의 ','를 모두 "값으로 변환 &
데이터 타입 변환

```
[ ] left_df_2018 = df_2018[['model','segment','brand', 'spec_min', 'spec_max', 'release_date', 'date', 'nation_code']]
right_df_2018 = df_2018[['sales','price_min','price_max']].stack().str.replace(',','').unstack().astype('float')
df_2018_ = pd.concat([left_df_2018, right_df_2018], axis=1)
print(df_2018_)

left_df_2019 = df_2019[['model','segment','brand', 'spec_min', 'spec_max', 'release_date', 'date', 'nation_code']]
right_df_2019 = df_2019[['sales','price_min','price_max']].stack().str.replace(',','').unstack().astype('float')
df_2019_ = pd.concat([left_df_2019, right_df_2019], axis=1)
print(df_2019_)

left_df_2020 = df_2020[['model','segment','brand', 'spec_min', 'spec_max', 'release_date', 'date', 'nation_code']]
right_df_2020 = df_2020[['sales','price_min','price_max']].stack().str.replace(',','').unstack().astype('float')
df_2020_ = pd.concat([left_df_2020, right_df_2020], axis=1)
print(df_2020_)

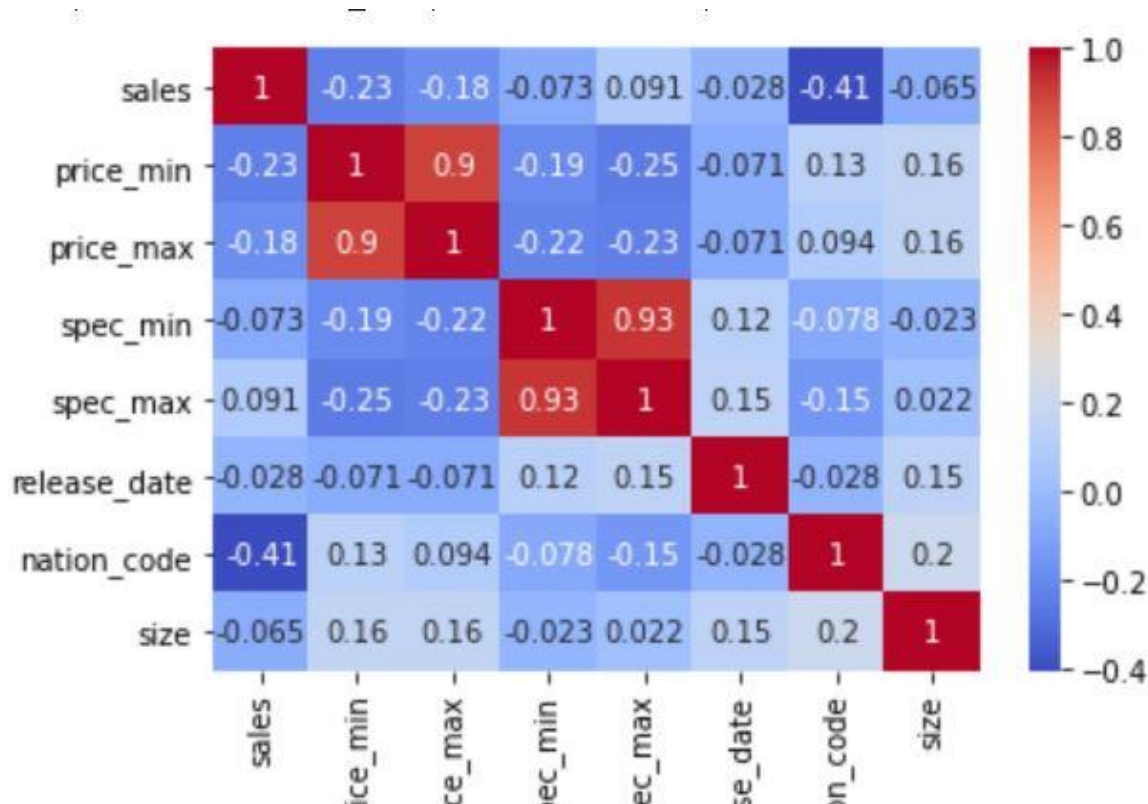
df_whole_ = pd.concat([df_2018_, df_2019_, df_2020_], axis=0)
print(df_whole_)
```

	model	segment	brand	spec_min	spec_max	release_date	date	nation_code
0	그랜저	7,662	준대형	...	2016.11	2018-01	1	
1	포터2	6,247	상용	...	2004.01	2018-01	1	
2	쏘렌토	5,906	중형SUV	...	2017.07	2018-01	1	
3	아반떼	5,677	중형	...	2015.09	2018-01	1	
4	쏘나타 뉴 라이즈	5,230	중형	...	2017.03	2018-01	1	
...	
797	Mondeo	16	중형	...	2015.03	2020-04	4	
798	Escalade	12	대형SUV	...	2017.05	2020-04	4	
799	XT5	11	대형SUV	...	2016.11	2020-04	4	
800	All new Corsair	7	소형SUV	...	2020.05	2020-04	4	
801	CTS	3	준대형	...	2014.06	2020-04	4	

	model	segment	brand	spec_min	spec_max	release_date	date	nation_code	price_min	price_max
0	그랜저	준대형	현대	7662.0	2545.0	4350.0				
1	포터2	상용	현대	6247.0	49.0	4274.0				
2	쏘렌토	중형SUV	기아	5906.0	695.0	3807.0				
3	아반떼	중형	현대	5677.0	1394.0	2415.0				
4	쏘나타 뉴 라이즈	중형	현대	5230.0	1860.0	2293.0				
...
2448	XT5	대형SUV	캐딜락	28.0	6537.0	7337.0				
2449	Escalade	대형SUV	캐딜락	12.0	12837.0	13837.0				
2450	Mondeo	중형	포드	5.0	4240.0	4240.0				
2451	Cherokee	중형SUV	지프	1.0	5070.0	5510.0				
2452	Kuga	중형SUV	포드	1.0	3930.0	4480.0				

[03] 기술통계 - 내부요인

내부 요인과 판매량 간의 상관관계 분석표



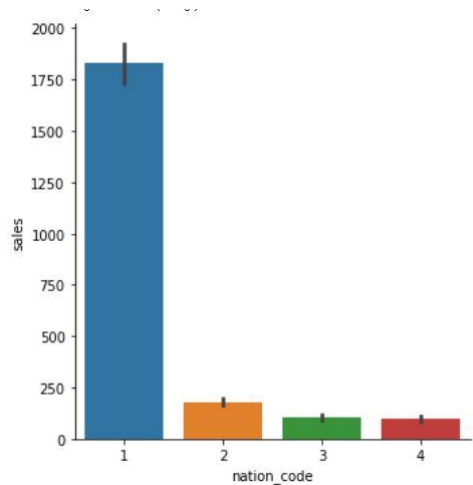
가격, 제조 국가 요인이
판매량과 상관관계를 보인다.

판매량과 약한 상관관계 (0.1-0.3):
최대 가격 (-0.23)
최소 가격 (-0.18)

판매량과의 상관관계 (0.3-0.7):
제조 국가 (-0.41)

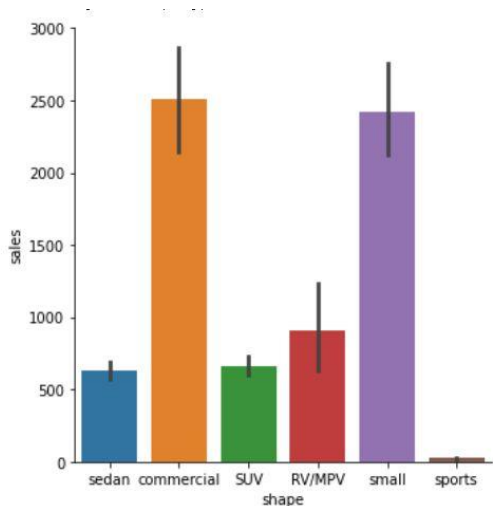
```
[ ] #수치형 자료_상관관계
corr = sns.heatmap(car[['model','sales','price_min','price_max','spec_min','spec_max','release_date','nation_code','size']].corr(),cmap='coolwarm',annot =True, annot_kws={'size': 10})
corr
```


[03] 기술통계 - 내부요인



제조 국가 (1 = 한국, 2 = 유럽, 3 = 중국/일본, 4 = 미국)
한국에서 제조된 자동차의 판매량이 압도적으로 높았고,
유럽, 일본/중국, 미국 순으로 판매량이 높았다.

```
# 제조 국가별 판매량 바플롯  
g = sns.factorplot(x='nation_code', y='sales', data=car, kind='bar')
```



차종 별 (세단, 상용차, SUV(소형, 중형, 대형), RV/MPV, 경차, 스포츠카)
상용차와 경차의 판매가 가장 많았다.
하지만 상용차의 경우, 특수 목적을 위한 차량임을 고려

```
# 모양(shape)별로 판매량 바 플롯  
g = sns.factorplot(x='shape', y='sales', data=car, kind='bar')
```

[03] 기술통계 - 내부요인

차종에 따른 선호도 비중 변화

비중을 알기 위해 float형 데이터로 치환,
이후 반복문을 통해 판매량을 % 데이터로 치환

```
car=car[['date', 'shape', 'sales']].groupby(['shape', 'date']).sum()  
car
```

sales		
shape	date	
RV/MPV	18.01	5922
	18.02	4114
	18.03	7023
	18.04	9880
	18.05	9257
...
suv	19.12	62959
	20.01	36533
	20.02	32024
	20.03	68075
	20.04	67601

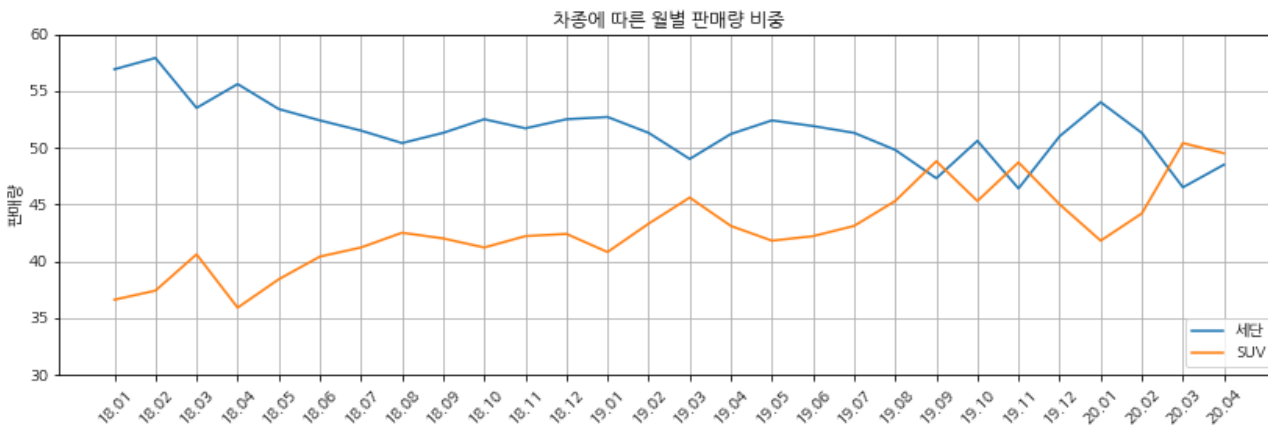
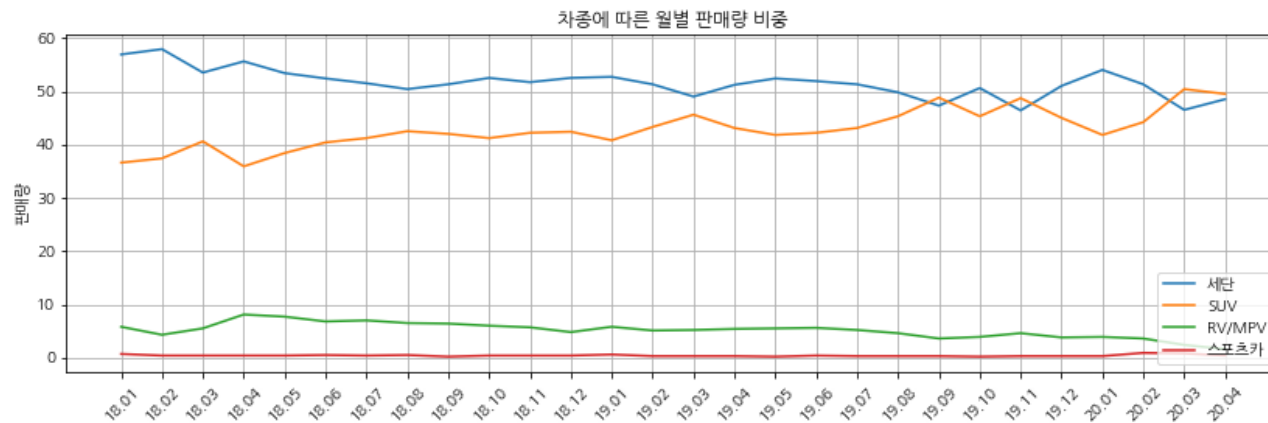
168 rows x 3 columns

```
a1=car.xs('sedan')  
a2=car.xs('suv')  
a3=car.xs('RV/MPV')  
a4=car.xs('sports')  
a=pd.merge(a1,a2, left_index=True, right_index=True)  
a=pd.merge(a,a3, left_index=True, right_index=True)  
a=pd.merge(a,a4, left_index=True, right_index=True)  
a.columns=['sedan', 'suv', 'RV/MPV', 'sports']  
a['sum']=a.sum(axis=1)  
  
#각각의 차종별로 차지하는 판매량 비중을 보기 위해 float형으로 변환  
a[['sum', 'sedan', 'suv', 'RV/MPV', 'sports']] = a[['sum', 'sedan', 'suv', 'RV/MPV', 'sports']].astype('float')  
  
#컬럼별로 반복문을 통해 월별 판매량으로 나눈 값에 *100 하여 치환  
for j in a.columns[0:-1]: #sedan, suv, RV, sports 까지 반복  
    for i in range(len(a[j])): #18.01 부터 20.04까지 반복  
        a[j][i]=round(a[j][i]/a['sum'][i]*100,1)  
  
a=a.reset_index()  
a
```

```
#차종에 따른 월별 판매량 비중 그래프 그리기  
import matplotlib.font_manager as fm  
fm._rebuild()  
fm.get_fontconfig_fonts()  
font_path = 'font/NanumGothic.ttf'  
font_name = fm.FontProperties(fname=font_path).get_name()  
print('font_name:', font_name)  
matplotlib.rc('font', family=font_name)  
  
plt.plot(a['date'], a['sedan'], label='세단')  
plt.plot(a['date'], a['suv'], label='SUV')  
# plt.plot(a['date'], a['RV/MPV'], label='RV/MPV')  
# plt.plot(a['date'], a['sports'], label='스포츠카')  
plt.grid()  
plt.xlabel('date')  
plt.ylabel('판매량')  
plt.title('차종에 따른 월별 판매량 비중')  
plt.legend(loc='lower right')  
plt.xticks(rotation=45)  
plt.rcParams["figure.figsize"] = (14,4)  
  
plt.savefig('savefig_default.png')  
  
# plt.savefig('savefig_1.png')  
  
plt.ylim(30, 130)  
plt.savefig('savefig_2.png')
```

	date	sedan	suv	RV/MPV	sports	sum
0	18.01	56.9	36.6	5.8	0.7	101544.0
1	18.02	57.9	37.4	4.3	0.4	95042.0
2	18.03	53.5	40.6	5.5	0.4	127901.0
3	18.04	55.6	35.9	8.1	0.4	122442.0
4	18.05	53.4	38.4	7.7	0.4	120131.0
5	18.06	52.4	40.4	6.8	0.5	117235.0
6	18.07	51.5	41.2	7.0	0.4	117289.0
7	18.08	50.4	42.5	6.5	0.5	114887.0
8	18.09	51.3	42.0	6.4	0.2	97386.0
9	18.10	52.5	41.2	6.0	0.4	121674.0
10	18.11	51.7	42.2	5.7	0.4	123169.0
11	18.12	52.5	42.4	4.8	0.4	122286.0
12	19.01	52.7	40.8	5.8	0.6	101814.0
13	19.02	51.2	42.2	5.4	0.2	94228.0

[03] 기술통계 - 내부요인



2018년 1월부터 세단은 전반적인 감소추세
반면 SUV는 지속적으로 성장

2019년 하반기와 2020년 상반기에는 SUV
의 판매 비중이 세단을 넘어서기도 했다.

스포츠실용차(SUV) 질주가 이어지고 있다. 덩치 큰 대형차급에서 작은 체구의 소형차급까지 SUV 선호 현상은 이제 하나의 경향으로 자리를 잡았다고 해도 지나친 말이 아니다. 이 흐름을 타고 2019년 상반기 국내 완성차 업체들이 판매한 승용차 가운데 SUV 비중은 50%에 육박한다. 판매 차량 10대 중 5대는 SUV라는 이야기다. SUV 열풍이라 할 만하다.

세계적으로도 SUV는 강세다. 중국의 승용차 판매에서 SUV 비중은 40%를 넘었고, 미국에서도 픽업트럭과 함께 판매 비중이 계속 늘어나는 추세다. 차를 다목적으로 활용할 수 있고 여가를 즐기려는 사람이 느는 게 가장 큰 배경으로 꼽힌다. 국내에선 쌍용차 ‘티볼리’에 이어 현대차 ‘코나’와 ‘베뉴’, 기아차 ‘스토닉’과 ‘셀토스’ 등 소형차급부터 쌍용차의 ‘G4 렉스턴’과 현대차 ‘팔리세이드’ 등 대형차급 열기까지 가세하면서 볼륨이 더 커졌다.

[03] 기술통계 - 내부요인

차량 크기에 따른 선호도 비중 변화 차량 크기 : 차종 차량 크기별로 범주화

경차 : 경차, 소형차

소형차 : 소형 SUV, 준중형

중형차 : 중형, 중형 SUV

대형차 : 대형, 대형 SUV, 준대형

```
car=car[['date', 'size', 'sales']].groupby(['size', 'date']).sum()  
car
```

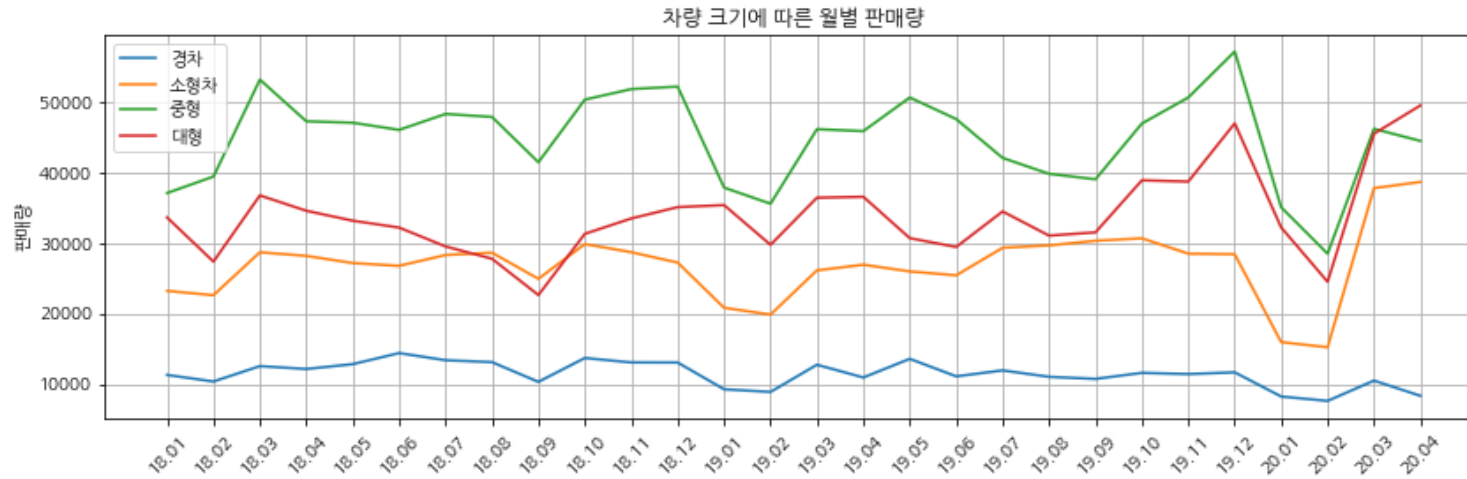
sales		
size	date	
0	18.01	24925
	18.02	22721
	18.03	30168
	18.04	33067
	18.05	32354
...
4	19.12	47061
	20.01	32270
	20.02	24555
	20.03	45578
	20.04	49606

140 rows × 1 columns

```
#인덱스 size별로 분류 하여 date기준으로 merge  
a1=car.xs(1)  
a2=car.xs(2)  
a3=car.xs(3)  
a4=car.xs(4)  
a=pd.merge(a1,a2, left_index=True, right_index=True)  
a=pd.merge(a,a3, left_index=True, right_index=True)  
a=pd.merge(a,a4, left_index=True, right_index=True)  
a.columns=['1', '2', '3', '4']  
a=a.reset_index()
```

```
import matplotlib.font_manager as fm #한글폰트 사용하기위해 불러옴  
fm._rebuild()  
fm.get_fontconfig_fonts()  
font_path = 'font/NanumGothic.ttf'  
font_name = fm.FontProperties(fname=font_path).get_name()  
print('font_name:', font_name)  
matplotlib.rc('font', family=font_name)  
  
plt.plot(a['date'], a['1'], label='경차')  
plt.plot(a['date'], a['2'], label='소형차')  
plt.plot(a['date'], a['3'], label='중형')  
plt.plot(a['date'], a['4'], label='대형')  
plt.grid()  
plt.xlabel('date')  
plt.ylabel('판매량')  
plt.title('차량 크기에 따른 월별 판매량')  
plt.legend(loc='best')  
plt.xticks(rotation=45) #축이를 공간부족으로 45도 회전  
plt.rcParams["figure.figsize"] = (14,4) #그래프 사이즈 설정  
  
plt.savefig('savefig_default.png') #png파일로 저장
```

[03] 기술통계 - 내부요인



차량 크기에 따른 선호도:

월별 예외가 있으나, 전반적으로

중형차 > 대형차 > 소형차 > 경차 순서

2019년 1월, 2020년 1월 차종과 관계 없이 모두 판매량 급감

[03] 기술통계 - 내부요인

제조 국가별 판매 비중 변화

비중을 알기 위해 각 국가별 판매량 데이터를 float형 데이터로 치환,
이후 for문을 통해 판매량을 % 데이터로 치환

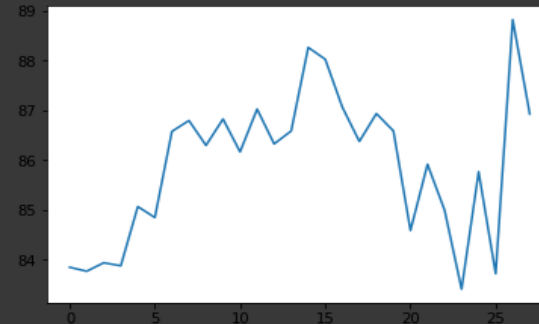
```
[ ] 1 # 전체 데이터 read
2 df_whole_tmp = df_whole[['sales', 'nation_code', 'date']].groupby(['nation_code', 'date'], as_index=False).sum()
3 df_whole_tmp
```

	nation_code	date	sales
0	1	2018-01	109276.0
1	1	2018-02	102737.0
2	1	2018-03	135597.0
3	1	2018-04	130373.0
4	1	2018-05	129957.0
...
107	4	2019-12	1998.0
108	4	2020-01	1452.0
109	4	2020-02	1077.0
110	4	2020-03	1383.0
111	4	2020-04	1486.0

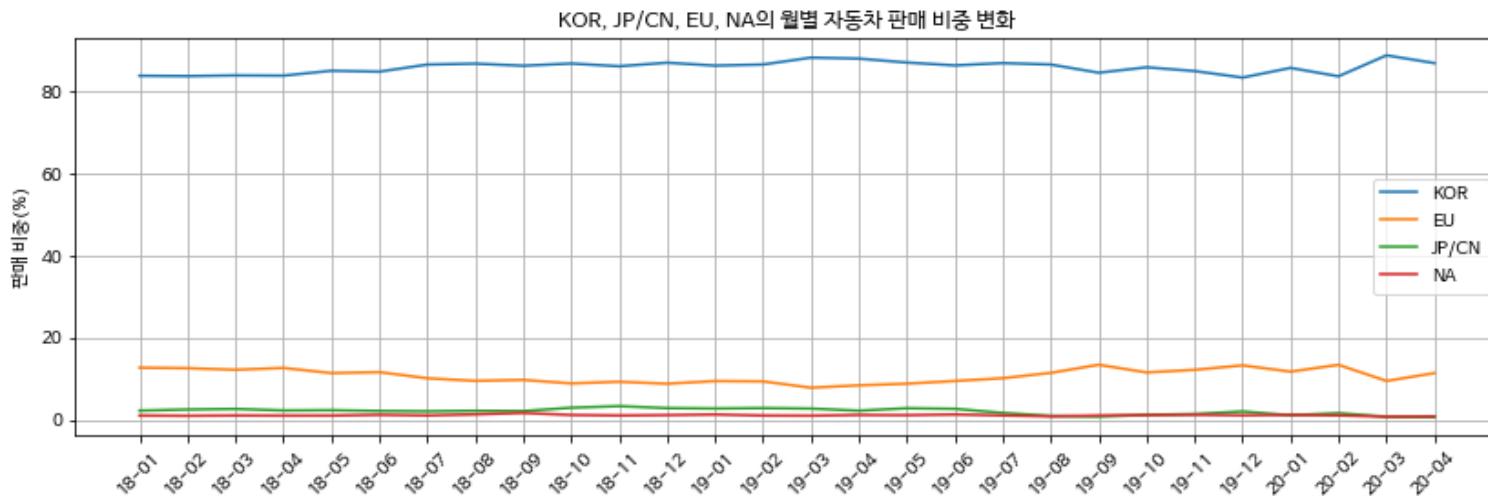
112 rows x 3 columns

```
1 df_kor = df_whole_tmp.iloc[0:28]
2 df_kor_percentage = []
3
4 # (KOR의 월 판매량) / (KOR, EU, JP/CN, NA 전체 월 판매량)
5 for i in range(df_kor.shape[0]):
6     tmp = (df_kor[['sales']].iloc[i] / df_all[['sales']].iloc[i]) * 100
7     df_kor_percentage.append(round(tmp[0], 2))
8
9
10 plt.plot(df_kor_percentage)
11 plt.show
```

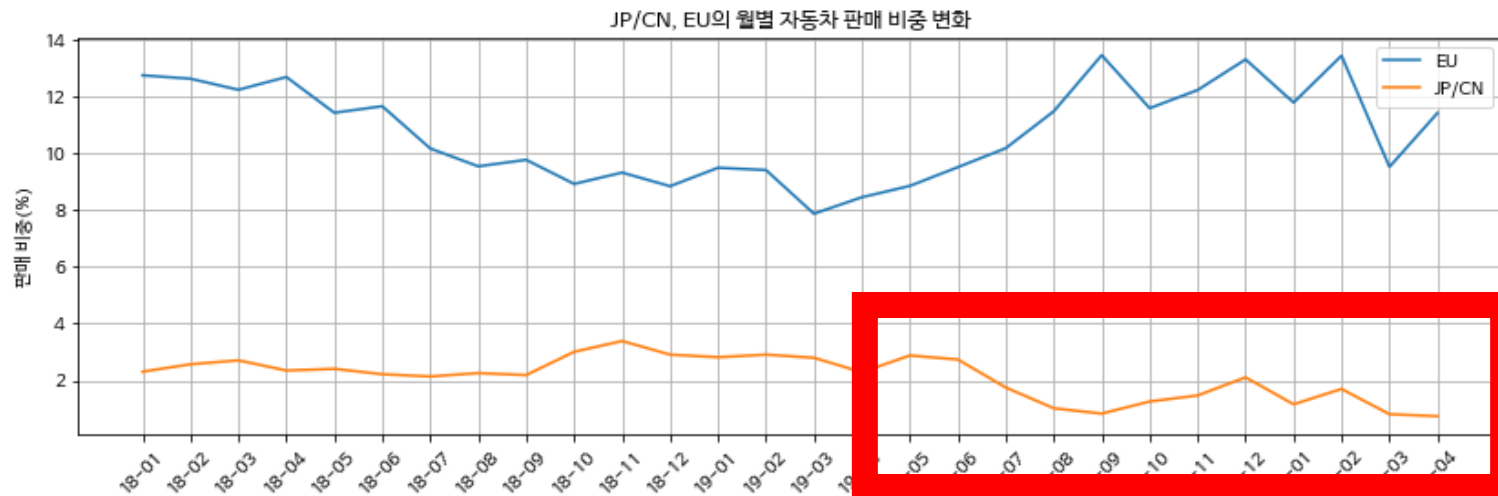
<function matplotlib.pyplot.show>



[03] 기술통계 - 내부요인



일본의 무역제재 발표 이후
JP/CN 비중 감소 시작



[03] 기술통계 - 외부요인

	new_new_date	sales_increase
0	2019 1	2.143559
1	2019 2	-3.927995
2	2019 3	-5.166961
3	2019 4	-2.619542
4	2019 5	-1.624526
5	2019 6	-6.568068
6	2019 7	-1.928198
7	2019 8	-6.037315
8	2019 9	4.1309
9	2019 10	-2.419391
10	2019 11	-1.206552
11	2019 12	9.28903
12	2020 1	-14.81245
13	2020 2	-19.34148
14	2020 3	8.537564
15	2020 4	8.524525

판매량의 전년대비 증감률

- 2019년 12월은 작년대비 9.3%의 증가를 보임
- 2020년 1,2월에는 작년대비 판매량이 급감했으나, 3월부터 시행된 개별 소득세 인하로 판매량이 회복된 것으로 추정

```
[221] #월별로 판매량 데이터프레임
percentage = car[['new_date', 'sales']].groupby(['new_date']).sum()
percentage = pd.DataFrame(percentage)

# 작년대비 판매량 증감률 표
dic = {'new_new_date': [], 'sales_increase': []}
for i in range(2019, 2021):
    for j in range(1, 13):
        if i == 2020 and j == 5:
            break
        dic['new_new_date'].append('{}-{:2}'.format(i, j))

for i in range(16):
    perc = (percentage.iloc[i+12]-percentage.iloc[i])/percentage.iloc[i]*100
    dic['sales_increase'].append(re.findall('[\w]+(\.[\w]+)?', str(perc))[0])
increase_data = pd.DataFrame(dic)
increase_data
```

■ 개소세 인하 즉시 車 판매량 상승...朴정부 경제적 효과 약 47조원

개소세 인하 조치에 따른 자동차 판매증가 효과는 즉각 나타나고 있다. 개소세 인하 첫 달인 지난 3월의 자동차 판매량이 10% 가까이 견인됐고 5월 들어서도 효과는 이어지고 있다.

한국산업연구원도 지난 4월 29일 동향분석에서 "3월 (자동차 시장) 내수는 코로나19 영향에도 불구하고 신차효과 및 정부 지원으로 금년 들어 처음으로 증가로 반전되면서 9.2% 증가했다"며 "국내 생산차의 경우 신차효과와 예전에 비해 큰 폭의 개별소비세 일시 감면으로 (판매량) 10.1% 증가했다"고 밝혔다.

[03] 기술통계 - 외부요인

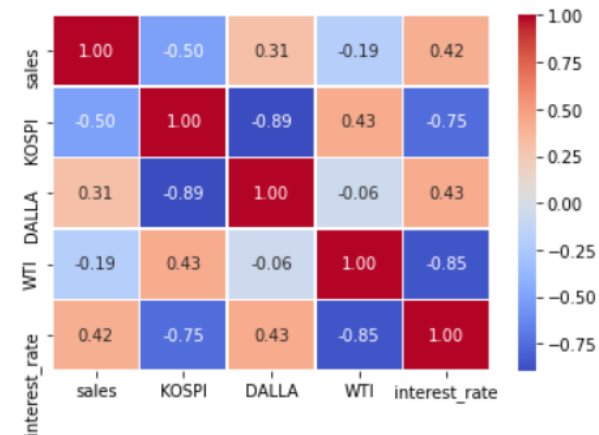
```
# 2018년 경제지표와 국산차 판매량 상관관계
df_kor_ = df_2018_tmp[['date', 'sales']][0:12]

df_index_mm_2018 = df_index_mm.iloc[0:12]

df_kor_corr = pd.merge(df_kor_, df_index_mm_2018, on='date')
print('2018년 경제지표와 국산차 판매량 상관관계')
# print(df_kor_corr)
# print()
print(df_kor_corr.corr())
sns.heatmap(df_kor_corr.corr(), cmap='coolwarm', annot=True, fmt = '.2f', annot_kws={'size' : 10}, lw=0.5)
plt.show()
```

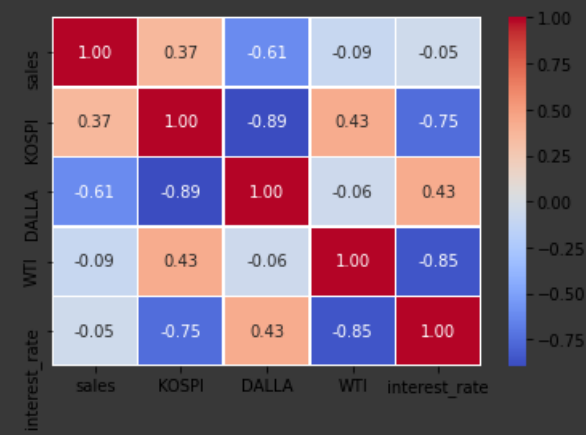
2018년 경제지표와 국산차 판매량 상관관계

	sales	KOSPI	DALLA	WTI	interest_rate
sales	1.000000	-0.502905	0.311757	-0.193097	0.419541
KOSPI	-0.502905	1.000000	-0.894971	0.432001	-0.749090
DALLA	0.311757	-0.894971	1.000000	-0.056926	0.433000
WTI	-0.193097	0.432001	-0.056926	1.000000	-0.852508
interest_rate	0.419541	-0.749090	0.433000	-0.852508	1.000000



2018년 경제지표와 외산차 판매량 상관관계

	sales	KOSPI	DALLA	WTI	interest_rate
sales	1.000000	0.369335	-0.610035	-0.092528	-0.047205
KOSPI	0.369335	1.000000	-0.894971	0.432001	-0.749090
DALLA	-0.610035	-0.894971	1.000000	-0.056926	0.433000
WTI	-0.092528	0.432001	-0.056926	1.000000	-0.852508
interest_rate	-0.047205	-0.749090	0.433000	-0.852508	1.000000



국산차의 경우:

1. KOSPI (-0.5)
2. 금리(0.42)
3. 환율(0.31)
4. 유가(0.19)

외산차의 경우:

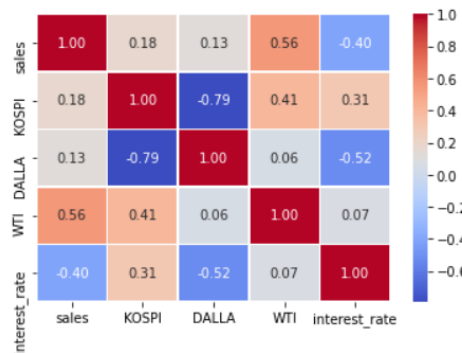
1. 환율(-0.61)
2. KOSPI (0.37)

순서로 상관관계를 보인다.

[03] 기술통계 - 외부요인

2019년 경제지표와 국산차 판매량 상관관계

	sales	KOSPI	DALLA	WTI	interest_rate
sales	1.000000	0.180309	0.126260	0.557450	-0.404783
KOSPI	0.180309	1.000000	-0.791667	0.406678	0.305096
DALLA	0.126260	-0.791667	1.000000	0.058363	-0.519029
WTI	0.557450	0.406678	0.058363	1.000000	0.069203
interest_rate	-0.404783	0.305096	-0.519029	0.069203	1.000000

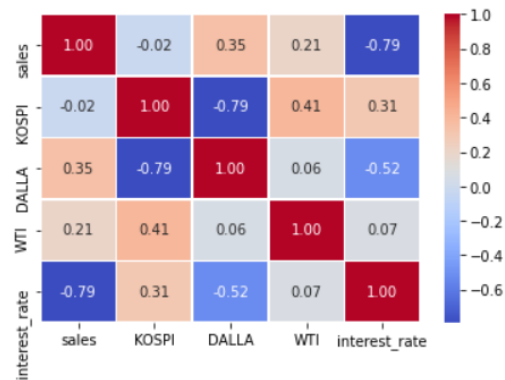


국산차의 경우:

1. 유가(0.56)
2. 금리(0.40)
3. KOSPI (0.18)
4. 환율(0.13)

2019년 경제지표와 외산차 판매량 상관관계

	sales	KOSPI	DALLA	WTI	interest_rate
sales	1.000000	-0.017229	0.353248	0.205339	-0.789433
KOSPI	-0.017229	1.000000	-0.791667	0.406678	0.305096
DALLA	0.353248	-0.791667	1.000000	0.058363	-0.519029
WTI	0.205339	0.406678	0.058363	1.000000	0.069203
interest_rate	-0.789433	0.305096	-0.519029	0.069203	1.000000

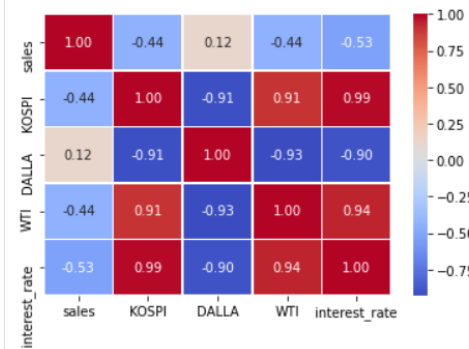


외산차의 경우:

1. 금리(-0.79)
2. 환율(0.35)
3. 유가(0.21)
3. KOSPI (0.18)

2020년 경제지표와 외산차 판매량 상관관계

	sales	KOSPI	DALLA	WTI	interest_rate
sales	1.000000	-0.436947	0.123897	-0.437612	-0.525842
KOSPI	-0.436947	1.000000	-0.912021	0.906928	0.990369
DALLA	0.123897	-0.912021	1.000000	-0.927805	-0.896823
WTI	-0.437612	0.906928	-0.927805	1.000000	0.941459
interest_rate	-0.525842	0.990369	-0.896823	0.941459	1.000000

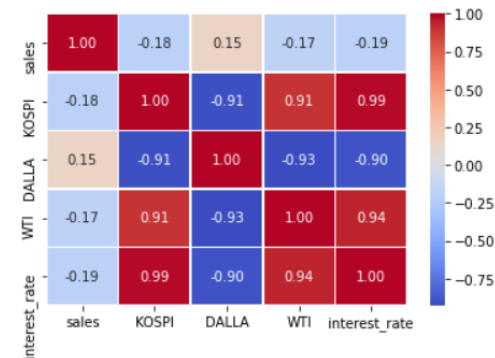


국산차의 경우:

1. 금리(0.53)
2. 유가(-0.44)
3. KOSPI (-0.44)
4. 환율(0.12)

2020년 경제지표와 국산차 판매량 상관관계

	sales	KOSPI	DALLA	WTI	interest_rate
sales	1.000000	-0.183338	0.150540	-0.168975	-0.187301
KOSPI	-0.183338	1.000000	-0.912021	0.906928	0.990369
DALLA	0.150540	-0.912021	1.000000	-0.927805	-0.896823
WTI	-0.168975	0.906928	-0.927805	1.000000	0.941459
interest_rate	-0.187301	0.990369	-0.896823	0.941459	1.000000



외산차의 경우:

1. 금리(-0.19)
2. KOSPI(-0.18)
3. 유가(-0.17)
3. 환율 (0.15)

[04]한계와 향후과제

한계:

PANDAS에 대한 숙련도 부족

가설을 세운 변수들간의 상관관계를
직접 PANDAS를 통해 구현하는데 어려움

- 내부요인과 외부요인의 복합적인 상관관계를 파악하는데 한계

향후과제:

PANDAS 전문성 강화

보다 다양한 변수들간의 복합적인 상관관계를 구현