ADP 26회 실기 문제

In [2]:
```
import pandas as pd
import numpy as np
import scipy as stats
import matplotlib.pyplot as plt
import seaborn as sns
from collections import Counter
import datetime as dt
import time

from sklearn.preprocessing import StandardScaler
from sklearn.cluster import KMeans
import pystan
import warnings
warnings.filterwarnings('ignore')
```

머신러닝 - 데이터 설명

- 데이터 설명
  - InvoiceNo: Invoice number. Nominal, a 6-digit integral number uniquely assigned to each transaction. If this code starts with letter 'c', it indicates a cancellation.
  - StockCode: Product (item) code. Nominal, a 5-digit integral number uniquely assigned to each distinct product.
  - Description: Product (item) name. Nominal.
  - Quantity: The quantities of each product (item) per transaction. Numeric.
  - InvoiceDate: Invice Date and time. Numeric, the day and time when each transaction was generated.
  - UnitPrice: Unit price. Numeric, Product price per unit in sterling.
  - CustomerID: Customer number. Nominal, a 5-digit integral number uniquely assigned to each customer.
  - Country: Country name. Nominal, the name of the country where each customer resides.
- 출처 : - https://archive.ics.uci.edu/ml/index.phpm
- 데이터url : https://raw.githubusercontent.com/Datamanim/datarepo/main/adp/26/problem1.csv

In [195]:
```
import pandas as pd
df= pd.read_csv('https://raw.githubusercontent.com/Datamanim/datarepo/main/adp/26/problem1.csv')
```

문제1.1

결측치를 확인 하고, 결측치 제거 또는 대치하고 방법에 대해 설명하라

In [116]:
```
display(df.head(3))
# 결측처리 전
display(df.isna().sum()[df.isna().sum() >0])
display(df.isna().sum()[df.isna().sum() >0]*100/df.shape[0])
print(df.shape)
# UniPrice 결측처리
# 상품 코드가 같은 것들의 unit price의 최빈값으로 결측처리
# 참고할만한 상품코드의 unit price이 없다면 전체 최빈값으로 결측처리함
sc_na_up = list(df.loc[df.UnitPrice.isna(),'StockCode'].unique())
for sc in sc_na_up:
    same_sc_up = df.loc[df.StockCode == str(sc), 'UnitPrice']
    if len(same_sc_up) - sum(same_sc_up.isna()) > 0:
        mode = df.loc[df.StockCode == str(sc), 'UnitPrice'].mode()[0].copy()
        df.loc[(df.StockCode == sc)&df.UnitPrice.isna(), 'UnitPrice'] = mode
    else:
        mode = df.loc[:, 'UnitPrice'].mode()[0].copy()
        df.loc[(df.StockCode == sc)&df.UnitPrice.isna(), 'UnitPrice'] = mode

# Quantity 결측은 최빈값으로 처리
# df.loc[df.Quantity.isna(), 'Quantity'] = mode
df = df.drop(df.loc[df.Quantity.isna(),:].index, axis = 0)
print(df.shape)
```

```
Quantity     25
UnitPrice    97
dtype: int64
Quantity     0.069830
UnitPrice    0.270942
dtype: float64
(35801, 8)
(35776, 8)
```

[답안]

- 결측치가 있는 변수는 Quantity와 UnitPrice이다. 결측률은 둘 모두 1%가 되지 않는다.
- UnitPrice는 데이터 특성상 StockCode가 같으면 같은 UnitPrice를 가지기 때문에 StockCode의 UnitPrice의 최빈값으로 결측처리를 하였고 참고할만 것들이 없다면 전체 최빈값으로 처리하였다.
- Quntity의 경우는 참고할만한 데이터가 없고 결측률이 매우 낮기 때문에 25건에 대해 삭제처리하였다.

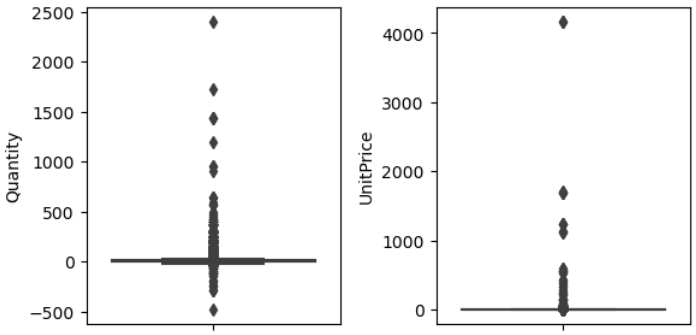
문제1.2

이상치를 제거하는 방법을 설명하고, 이상치 제거 전 후 결과에 대해 통계적인 방법을 포함하여 설명할 것

In [117]:
```
# 이상치 식별
fig, axs = plt.subplots(1,2, figsize = (2*3, 1*3))
axs = axs.reshape(1, -1)
sns.boxplot(y = 'Quantity', data = df, ax = axs[0,0])
sns.boxplot(y = 'UnitPrice', data = df, ax = axs[0,1])
plt.tight_layout()
plt.show()
print('이상치 처리 전 건수: ', df.shape)
```

```
# 이상치 처리
del_idx = []
del_idx.extend(list(df[df.Quantity > 2000].index))
del_idx.extend(list(df[df.UnitPrice > 3000].index))
df = df.drop(del_idx, axis = 0)

print('이상치 처리 후 건수: ', df.shape)
```




```
이상치 처리 전 건수:  (35776, 8)
이상치 처리 후 건수:  (35772, 8)
```

[답안]

- 먼저, 이상치를 식별부터 해야하는데 대표적인 방법으로 IQR방식이 있다. boxplot의 이상치 결정 방법을 그대로 이용하는 것인데 상황에 따라 아주 많은 데이터를 이상치로 간주할 위험이 있다. 데이터가 그러한 경우로 IQR방식이 아닌 boxplot를 확인하여 기준을 세워 크게 벗어난 데이터를 삭제한다. 그 결과 4건의 이상치가 삭제되었다.

문제1.3

각 StockCode을 기준으로 파생변수들을 만들고 제품들의 특성에 따른 군집 생성을 위한 전처리를 수행하라.

In [174]:
```
df['Price'] = df['Quantity'] + df['UnitPrice']
df['hour'] = df.InvoiceDate.str[-5:-3].astype('int')

df2 = df.groupby('StockCode').agg({'Price':['count', 'mean', 'sum'],
                                   'Quantity':['mean', 'sum'],
                                   'UnitPrice':['mean']})
df2.columns = ['_'.join(x) for x in df2.columns.to_flat_index()]
df2 = df2.reset_index()
display(df2.head(3))
df3 = df.merge(df2, on = 'StockCode', how = 'left')

df3.columns
df4 = df3[['Quantity','UnitPrice', 'Price', 'hour',
       'Price_count', 'Price_mean', 'Price_sum', 'Quantity_mean',
       'Quantity_sum', 'UnitPrice_mean']]
# 스케일링
ss = StandardScaler()
df5 = pd.DataFrame(ss.fit_transform(df4), columns = df4.columns)
```

	StockCode	Price_count	Price_mean	Price_sum	Quantity_mean	Quantity_sum	UnitPrice_mean
0	10002	12	35.933333	431.20	35.083333	421.0	0.850000
1	10120	1	10.210000	10.21	10.000000	10.0	0.210000
2	10125	13	47.673846	619.76	46.923077	610.0	0.750769

[답안]

- StockCode별로 건수, 평균판매금액, 총판매금액, 평균판매수량, 총판매수량, 평균단위금액, 구매시간을 만들었다.
- 그리고 스케일링을 진행하였다.

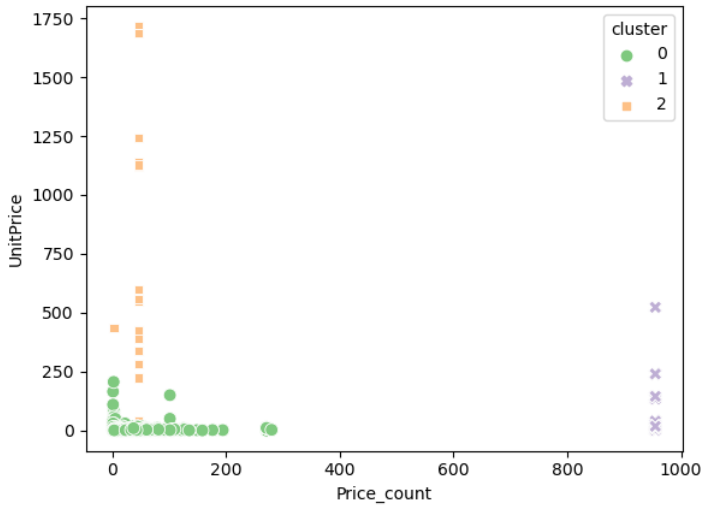
문제 2.1

Kmeans , DBSCAN 방법 중 하나를 선택하여 군집분석을 시행하고 각 군집별 특성을 분석하라

In [181]:
```
# 군집만들기
model = KMeans(n_clusters = 3)
distances = model.fit_transform(df5)
kmeans_result = pd.DataFrame(model.labels_, columns = ['cluster'])

ddf = pd.merge(df3, kmeans_result, left_index = True, right_index = True)

#군집 결과 시각화
fig, ax = plt.subplots(1,1)
sns.scatterplot(x = 'Price_count', y = 'UnitPrice', data = ddf, hue = 'cluster', style = 'cluster',
                s = 60, ax = ax, palette = 'Accent')
plt.show()
```



[답안]

- kmeans를 수행한 결과 3개의 군집을 형성하였는데 거래의 특징은 뚜렷하다.
- 0번 군집은 상품의 단위가격이 낮으면서 거래 횟수은 적은 특징이 있고
- 1번 군집은 거래 횟수는 적지만 단위가격이 높다.
- 2번 군집은 거래 횟수가 많다는 특징이 있다..

문제 2.2

각 군집 별 대표 추천 상품을 도출할 것

In [208]:
```
sc_0 = ddf.loc[ddf['cluster']==0, 'StockCode'].mode()[0]
print('0번 군집의 대표적 상품 : ({}){}'.format(sc_0, ddf.loc[ddf.StockCode == sc_0, 'Description'].unique()[0]))

sc_1 = ddf.loc[ddf['cluster']==1, 'StockCode'].mode()[0]
print('1번 군집의 대표적 상품 : ({}){}'.format(sc_1, ddf.loc[ddf.StockCode == sc_1, 'Description'].unique()[0]))

sc_2 = ddf.loc[ddf['cluster']==2, 'StockCode'].mode()[0]
print('2번 군집의 대표적 상품 : ({}){}'.format(sc_2, ddf.loc[ddf.StockCode == sc_2, 'Description'].unique()[0]))
```

```
0번 군집의 대표적 상품 : (22326)ROUND SNACK BOXES SET OF4 WOODLAND
1번 군집의 대표적 상품 : (POST)POSTAGE
2번 군집의 대표적 상품 : (M)Manual
```

문제 2.3

CustomerID가 12413인 고객을 대상으로 상품을 추천할 것

In [205]:
```
display(ddf.loc[ddf.CustomerID==12413, 'cluster'].value_counts())
```

```
cluster
0    37
1     3
Name: count, dtype: int64
```

[답안]

- 고객번호 12413의 경우 0번 군집에 해당하는 거래 37건, 1번 군집에 해당하는 거래 3건을 하였다. 해당 고객에게 상품을 추천한다면 0번 군집의 대표적 상품인 (22326)ROUND SNACK BOXES SET OF4 WOODLAND'를 추천하며 서브로 1번의 군집의 대표적 상품 (POST)POSTAGE도 추천한다.

통계

문제 3

어느 제조업체의 제품 불량률을 조사하려고 한다. 이 회사의 제품 불량률이 실제로는 90%라고 알려져 있다.
이를 표본 조사로 추정하고자 합니다. 추정된 불량률의 추정오차한계가 5% 이내가 되도록 하려면, 어느 정도의 표본 크기가 필요한지 계산하라.

In [1]:
```
from scipy.stats import norm
ME2 = 0.05 #
p = 0.9 # 모비율
conf_a2 = 0.05 # 유의수준
conf_z2 = norm.ppf(1-conf_a2/2)
ssize = conf_z2**2 * p * (1-p) / ME2**2
print('유의수준 {:.2f} 에서 오차의 한계를 {:.2f} 이하로 하려면 표본크기 {:.1f}가 필요하다.'.format(0.05, ME2, ssize)) # 6분 소요
```

```
유의수준 0.05 에서 오차의 한계를 0.05 이하로 하려면 표본크기 138.3가 필요하다.
```

문제 설명 (4번)

month	1월	2월	3월	4월	5월	6월	7월	8월	9월
USD/oz	12.14	42.6	34.4	35.29	30.96	57.12	37.84	42.49	31.38

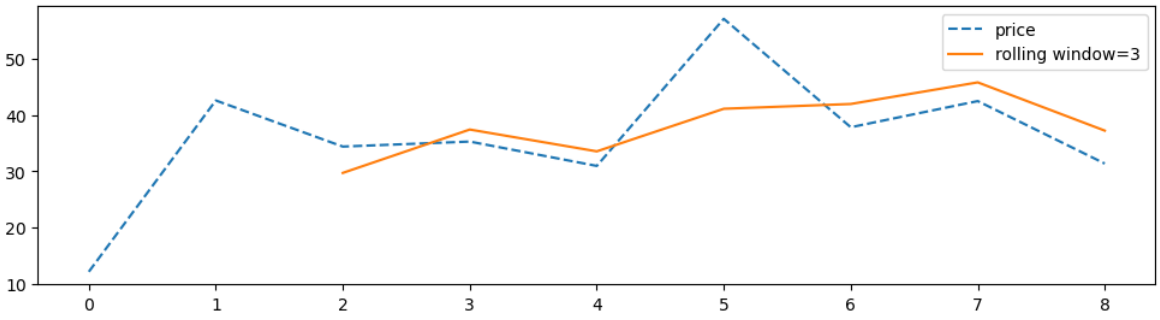
문제 4.1

은의 가격 및 이동평균값 3이 설정된 시계열 그래프를 그려라

In [9]:
```
ts = pd.DataFrame({'price':[12.14, 42.6, 34.4, 35.29, 30.96, 57.12, 37.84, 42.49, 31.38]})
rw = ts.rolling(window = 3).mean().dropna()

fig, ax = plt.subplots(1,1, figsize = (12, 3))

ax.plot(ts, label= 'price', linestyle = 'dashed')
ax.plot(rw, label= 'rolling window=3', linestyle = 'solid')
plt.legend()
plt.show() #10분 소요
```



문제 4.2

1월 대비 9월의 은의 가격은 몇 % 올랐는가? 소수점 두번째 자리에서 반올림

In [13]:
```
print('1월 대비 9월의 은의 가격은 {}% 올랐다.'.format(np.round(31.38*100/12.14,decimals=2))) # 2분 소요
```

1월 대비 9월의 은의 가격은 258.48% 올랐다.

문제설명 (5번)

	A	B	C
찬성	176	193	159
반대	124	107	141

위 표는 A,B,C 자치구별 W 의원에 대한 찬성, 반대 지지를 나타낸다. 자치구별 지지율이 같은지에 대해서 검정하라

문제 5.1

연구가설과 귀무가설을 설정하라

답안

- 귀무가설(H0): 자치구별 지지율이 같다.
- 연구가설(H1): 적어도 하나의 자치구의 지지율이 다르다.

문제 5.2

검정통계량을 구하고 결론을 내라

In [25]:
```
t = pd.DataFrame({'A':[176, 124], 'B':[193, 107], 'C':[159, 141]})
t.index = ['찬성', '반대']
```

In [24]:
```
t = pd.DataFrame({'A':[176/(176+124), 124/(176+124)], 'B':[193/(193+107), 107/(193+107)], 'C':[159/(159+141), 141/(159+141)]})
t.index = ['찬성', '반대']
stats.chi2_contingency(t, correction = False)
```

Out[24]:
```
Chi2ContingencyResult(statistic=0.026484604105571827, pvalue=0.986844991481755, dof=2, expected_freq=array([[0.58666667, 0.58666667, 0.58666667],
       [0.41333333, 0.41333333, 0.41333333]]))
```

[결론]

- 통계량 0.026, 유의확률(p-value) 0.98로 유의수준 0.05보다 크므로 귀무가설을 기각하지 못한다. 따라서 유의수준 5%하에 자치구별 지지율이 같다고 볼 수 있다.

문제설명 (6번)

A초등학교 남학생 16명과 여학생 9명의 혈압을 측정한 pressure.csv파일을 가지고 남녀학생의 평균 혈압에 차이가 없는지 확인하려한다.

- 데이터 url :https://raw.githubusercontent.com/Datamanim/datarepo/main/adp/26/pressure.csv

문제 6.2

검정 통계량을 구하고 가설 검정을 수행하라

In [34]:
```
df= pd.read_csv('https://raw.githubusercontent.com/Datamanim/datarepo/main/adp/26/pressure.csv')
print(df.head())
f = df.loc[df.gender=='female', 'pressure']
```

```
m = df.loc[df.gender=='male', 'pressure']
# stats.ranksums(f, m)
print(stats.bartlett(f, m))
print(stats.ttest_ind(f,m, equal_var = True))
```

```
  gender  pressure
0   male     106.8
1   male     100.8
2   male      84.5
3   male     104.2
4   male     107.0
BartlettResult(statistic=0.23460884702104196, pvalue=0.6281272971987353)
TtestResult(statistic=-1.598335024574904, pvalue=0.12361716364314851, df=23.0)
```

[답안]

- 귀무가설: 남녀학생의 평균 혈압에 차이가 없다.
- 연구가설: 남녀학생의 평균 혈압에 차이가 있다.
- 검정통계량은 -1.59이며 유의확률은 0.12로 유의수준 0.05보다 크므로 귀무가설을 기각하지 못한다.

문제 6.3

6.2의 검정 통계량 값을 바탕으로 신뢰수준 95%하에서 신뢰구간을 설정하라

In [39]:
```
x = df.pressure.mean()
conf_a = 0.05
conf_z = norm.ppf(1-conf_a/2)
SE = df.pressure.std(ddof=1)
ME = conf_z * SE
print('신뢰구간 {}~{}'.format(x-ME, x+ME) )
```

```
신뢰구간 74.79274678225713~113.10325321774289
```

[답안]

- 신뢰수준 95%하의 신뢰구간은 74.79~113.1이다.

문제설명 (7번)

height(키), weight(몸무게), waist(허리둘레) 컬럼을 가진 problem7.csv파일을 가지고 다음을 분석하라
A시의 20대 남성 411명을 임의로 추출하여 키, 몸무게, 허리둘레를 조사하여 기록한 데이터이다.
이 데이터를 이용하여 20대 남성의 키와 허리둘레가 체중에 영향을 미치는지 알아보고자 한다.

- 데이터 url :https://raw.githubusercontent.com/Datamanim/datarepo/main/adp/26/problem7.csv

문제 7.1

아래 조건을 참고하여 회귀계수(반올림하여 소수점 두자리)를 구하시오.

- 베이지안 회귀
- 시드넘버 1234로 지정
- 1000번의 burn-in 이후 10,000의 MCMC를 수행
- 회귀계수의 사전분포는 부적절한 균일분포(improper uniform prior distribution), 오차항의 분산의 사전분포는 역감마 분포로 지정. 이때, 형상(Shape)모수와 척도(Scale)모수는 각각 0.005로 지정.

In [1]:
```
import pandas as pd
df =pd.read_csv('https://raw.githubusercontent.com/Datamanim/datarepo/main/adp/26/problem7.csv')
df
```

Out[1]:

	height	weight	waistline
0	174.396	72.102	79.3787
1	179.656	81.255	80.6649
2	175.079	76.207	80.3166
3	180.804	81.354	80.8794
4	177.448	78.768	80.3499
...	...	...	...
406	174.207	73.736	80.1779
407	174.702	74.529	80.1306
408	176.858	76.083	80.4527
409	175.566	76.459	80.2019
410	177.076	74.667	79.9108

411 rows × 3 columns

pystan 패키지 사용 추천

- 베이지안 회귀(Bayesian regression)
  - 회귀 모델의 파라미터에 대한 불확실성을 확률적으로 다루는 방법
  - 파라미터에 대한 사전 분포와 데이터에 대한 우도(likelihood)를 결합하여 사후 분포(posterior distribution)를 계산
  - 이 사후 분포를 사용하여 파라미터의 불확실성을 추정하고 예측을 수행

In [ ]:
```
!pip install pystan
```

In [ ]:
```
! jupyter notebook --NotebookApp.iopub_data_rate_limit=10000000
```

문제 7.2

위에서 만든 모델을 바탕으로 키 180cm, 허리둘레 85cm인 남성의 몸무게를 추정하라

5페이지 끝.