Business Analytics Programming Fall 2019 - Midterm # 1

Use the table *df* below for the questions (1-5)

	Car	Inventory	Cost	Date
a	subaru	5	200	Jan-1-2017
b	mazda	10	500	May-10-2017
c	acura	7	300	Feb-5-2017
d	toyota	3	350	Mar-7-2017
e	toyota	7	500	Apr-12-2017
f	jaguar	10	150	Nov-8-2017
g	tesla	8	50	Jul-13-2017

```
import numpy as np
import pandas as pd
```

1. What values will you get back for:

```
df.iloc[4]
```

2. What values will you get back for:

```
df.loc[’g’]
```

3. Create a table df2 where all the Cost are greater than 50 and Inventory are greater than 10. A.

```
df2=df[(df[’Cost’]>50) & (df[’Inventory’]>10)]
```

B.

```
df2=df[df[’Cost’]>50 & df[’Inventory’]>10]
```

C.

```
df2=df[df[’Cost’]>50 && df[’Inventory’]>10]
```

D.

```
df2=df[(df[’Cost’]>50 & df[’Inventory’]>10)]
```

E. None of the above

6. What is x?

```
s=pd.Series([3,5,2,1],index=['a','b','b','d'])
x=sum(s['b'])
```

7. What is x?

```
s=pd.Series([3,5,2,1],index=['a','b','c','d'])
x=sum(s[1:5])
```

8. What is x?

```
s=pd.Series([3,5,2,1],index=[0,2,3,4])
x=s[[0,3]]
```

9. What is b?

```
v=np.array([2,6,6,7])
w=np.array([2,1,5,4])
w=v+w
w[0:2]=v[[1,2]]
b=sum(w)
```

4 6 11 11

5 6

10. What is v?

```
a = np.arange(5)
a[0]=2.5
v=sum(a)
```

0 1 2 3 4

2

11. What is b?

```
a = np.array([8,10,2,6,4])
a = a*2.5
a[2]=5.5
b=sum(a)
```

20 25 5 15 10

5.5

12. What is z?

```
x=[2,4,6,8,10]
y=[4,3,2,1]
x.append(1)
z=x+y
```

13. What is z?

```
x=[4,3,2,1,0]
y=[1,2,3,4,5]
    0 1 2 3 4
z=[]
for i in x:
  z.append(y[i])
```

Business Analytics Programming Fall 2019, Midterm # 2 - A

```
import numpy as np
import pandas as pd

```

1. Create table df2, with just the columns 'text' and 'b'?

df2=df[['text','b']]. ✓

2. What is the value of x?

```
a=pd.Series([5,4,3,2,1],index=['peach','orange','melon',
'cherry','apple'])
b=pd.Series([50,40,30,20,10],index=['peach','orange','
melon','apple','cherry'])
df3=pd.DataFrame({'a':a,'b':b})
x=df3.iloc[3]['b']

```

	a	b
apple	1	20
cherry	2	10
melon	3	30
orange	4	40
peach	5	50

0	5	50
1	4	40
2	3	30
3	2	20
4	1	10

~~30~~ 20

~~40~~

orange 40

Cherry. 2.

index	a	b
0	5	50
1	4	40
2	3	30
3	2	20
4	1	10

	a	b
peach	5	50
orange	4	40
melon	3	30
cherry	2	20
apple	1	10

3. What is the value of x?

```
a=pd.Series([5,4,3,2,1],index=['peach','orange','melon
'cherry','apple'])
b=pd.Series([50,40,30,20,10],index=['peach','orange','
melon','cherry','apple'])
df3=pd.DataFrame({'a':a,'b':b})
x=df3.iloc[3]['b']

```

~~30~~

20

b

orange 40.

20

4. The API function returns a JSON object within a JSON object, that is stored in the variable *A*. The JSON object within a JSON object is named *B*. Normalize the inner JSON object into a pandas dataframe named *df*. df=Josn_normalize(A,'B')

df=Josn_normalize(A,'C/E')

A OB

A

OB B

5. What is the value of z?

```
x=np.array([0,2,4,6,8])
y=x*.5
z=y
y[3]=y[4]+0.2

```

[0., 1., 4.2, [illegible], 4.]

01.

[0,1,2,3,4].

6. Using table df, what is the sum of all the prices of each type of candy (the total sum of snickers, the total sum of mounds,etc.) Write the code to do it.

df.groupby('candy')['Price'].sum()

sum(df[

df.groupby('candy')['Price'].sum()sorted

(by=...

	Candy	Price	Sales	Date
a	snickers	2	100	Jan-1-2017
b	mounds	3	50	May-10-2017
c	twix	4	60	Feb-5-2017
d	twix	2	200	Mar-7-2017
e	snickers	5	400	Apr-12-2017
f	mounds	4	240	Nov-8-2017
g	dove	7	200	Jul-13-2017

df.loc[(df.Candy=='dove')&(df.Sales<160)&(df.Sales>100), 'Price']=15

7. Using the above table, change all the prices of 'dove' to 15, where Sales is less than 160 and greater than 110. df.loc[(df.Candy=='dove')&(df.Sales<160)&(df.Sales>110),'Price']

8. What is sum(s[[5,2]])?

```
s=pd.Series([1,2,3,4,5,6],index=[2,5,3,7,1,9])
```

2+1 3. ✓

9. What is sum(s['a':'c'])? 含'c'

```
s=pd.Series([1,2,3,4,5,6],index=['a','b','c','d','d','e'])
```

6. ✓

10. After executing the below code, how many rows will be in df?

```
df2 = pd.DataFrame()
df=pd.DataFrame()
mid=0
for i in range(34):
  if i==0:
    tjson=api.statuses.user_timeline(screen_name="
realDonaldTrump",tweet_mode='extended',count =60))
  else:
    tjson=api.statuses.user_timeline(screen_name="
realDonaldTrump",tweet_mode='extended',count = 60,max_id
= mid)
  if len(tjson)>0:
    df=json_normalize(tjson)
    mid=df['id'].min()
    mid=mid-1
    df2 = pd.concat([df2, df], ignore_index=True)
```

60. ✓

final 不能有

final 不能包有

11. Using the table df below, what is sum(df.loc[[3,1],1])?

can't be position in loc, iloc could. / Nub.

should be 'Price'

	Fruit	Price	Sales	Date
6	Apple	2	140	Jan-1-2017
3	Orange	2	400	May-10-2017
2	Grapes	5	180	Feb-5-2017
0	Plums	2	300	Mar-7-2017
4	Peaches	6	120	Apr-12-2017
1	Water Melon	5	180	Nov-18-2017
5	Pineapple	4	60	Jul-13-2017

4.

iloc 对应 number. loc 对应字段 attribute

Error

12. Using the table df below, what is sum(df.iloc[2:4,1])?

应改数字

iloc[3, 'Fruit']. Error

X.

扫描全能王 创建

	Fruit	Price	Sales	Date
6	Apple	2	140	Jan-1-2017
3	Orange	2	400	May-10-2017
2	Grapes ✓	5	180	Feb-5-2017
0	Plums ✓	2	300	Mar-7-2017
4	Peaches	6	120	Apr-12-2017
1	Water Melon	5	180	Nov-18-2017
5	Pineapple	4	60	Jul-13-2017

7 ✓

13. Using the table df below, create a table df2 that has the 20 lowest prices, for only rows with sales less then 250 and greater than 100:

	Fruit	Prices	Sales	Date
6	Apple	2	140	Jan-1-2017
3	Orange	2	400	May-10-2017
2	Grapes	5	180	Feb-5-2017
0	Plums	2	300	Mar-7-2017
4	Peaches	6	120	Apr-12-2017
1	Water Melon	5	180	Nov-18-2017
5	Pineapple	4	60	Jul-13-2017

C ✓

A.

```
    dft = df[df['Sales']<250 & df['Sales']>100]
    dft2 = dft.sort_values()
    df2 = dft2.head(20)
```

B.

```
    dft = df[df['Sales']<250 & df['Sales']>100]
    dft2 = dft.sort_values(by='Prices')
    df2 = dft2.head(20)
```

C.

```
    dft = df[(df['Sales']<250) & (df['Sales']>100)]
    dft2 = dft.sort_values(by='Prices')
    df2 = dft2.head(20)
```

✓ 取前排 final Take

D.

```
    dft = df[(df['Sales']<250) & (df['Sales']>100)]
    dft2 = dft.sort_values(by='Prices')
    df2 = dft2.tail(20)
```

E. None of the above

14. Using the table df below, what is the total sum of price, by Fruit, for Fruit that have sales greater than 50 and less than 200, in descending order:

	Fruit	Price	Sales	Date
6	Apple	2	140	Jan-1-2017
3	Orange	2	400	May-10-2017
2	Grapes	5	180	Feb-5-2017
0	Plums	2	300	Mar-7-2017
4	Peaches	6	120	Apr-12-2017
1	Water Melon	5	180	Nov-18-2017
5	Pineapple	4	60	Jul-13-2017

A ✓

A.

```
  df2 = df[(df['Sales']<200) & (df['Sales']>50)]
  df2.groupby('Fruit')['Price'].sum().sort_values(by='
Price',descending=True)
```

✓

B.

```
  df2 = df[(df['Sales']<200) & (df['Sales']>50)]
  df2.groupby('Fruit')['Price'].sum().sort_values(by='
Price')
```

✓

C.

```
  df2 = df[df['Sales']<200 & df['Sales']>50]
  df2.groupby('Fruit')['Price'].sum().sort_values(by='
Price',descending=False)
```

✗

D.

```
  df2 = df[(df['Sales']<200) & (df['Sales']>50)]
  df2.groupby('Fruit','Price').sum().sort_values(by='Price
',descending=True)
```

✓

E. None of the above