Pandas

1) what is the max temperature in new yonk in the month of janury.

2) On which dayn did it rain

3) what was avnage speed of wind during the month.

sol.

1) df['Temp'].max() How to find the date.

2) df['EST'][df['Events']=='Rain']

3) df['windspeed'].mean()

a = df['windspeed'].sum()

b = a/30

in windspeed there are some missing data.

Cleaning messy data

df.fillna(0, inplace=True)

Dataframe

it is used to represent data with rows and columns.

1) Creating dataframe ✓
2) Dealing with rows and columns
3) ope: min, max, std, describle.
4) conditional selection
5) set index.

```
weather_data = {
 'day' : ['1/1','1/2', '1/3']
'temp' : [32, 35, 28]
'winds' : [6,7,2]
'event' : ['Rain','sunny', 'snow']
df= pd.Dataframe(weather_data)
```

① import pandas as

```
df = pd.read_csv('weather_data.csv).
```

creating dataframe by using Dictionary.

wa

② rows, column = df.shape.

indexing - sliceing. 0 1 2 3 4 5

df[2:5] → Colum (2 to 4)

df.column → shown all clourmn

df.day or df['event']

type of dataframe

df[['day', 'event', 'temp']]

(3) df.describe()
↑ satiastice.

sql
Select data in dataframe.

df[df.temp >= 32]
↑ = = df.temp.max()
select

df['day'][df.temp == df['temp'].max()
, 'temp']

df.index → nepnest nange index

df.~~index~~ inplace = True)

df.set_index('day') → set the index in day format /colamn.

df.loc['1/3/17']

↳ ~~see~~ shown the

df.reset_index(inplace = True)

ways of creating Data frame

```
df = pd.read_csv('weather.csv')
```

excel → .xlsx

```
pd.DataFrame(weather)   ← Dictionary.
```

read_csv
read_excel
- hdf
- SQL
- json
- html
- stata
- sas

Read / write

```
df = pd.read_csv('stock.csv', header = 1)
                              header = None, names = ['ticker', 'eps'])
```

↑ creat the header.

```
df = pd.read_csv('stock.csv', nrows = 3)
```

↑ represent only 3 rows.

```
, na_values['not available', 'Na']
```

```
df = pd.read_csv('stock.csv', na_values=["not available", 'n.a', -1])
```

Supply a dictionary

```
df = pd.read_csv("stock.csv", na_values={
    'eps' : ["not available", "n.a."],
    'revenue' : ["not available", "n.a.", -1],
})
```

→ write csv file.

```
df.to_csv('new.csv')
```
← creat new file in directry.

```
, index=False)
```
← to remove index

```
, cloums=['ticker', 'eps'])
```
← only two cloum in new.csv

1) fill

2) inte

3) drop

df = pd

type (

1/1

df = p

201

df. se

How to handal missing Data. (panda-5)

1) fillna to fill missing value
2) interpolate to guess on missing values
3) dropna to drop missing value rows

```
df = pd.read_csv("wather.csv")
type(df.day[0]) ⟶ str
   1/1/2017
df = pd.read_csv("wather.csv", parse_dates=["day"])
   2017-01-01
```

convert type into str → date.

→ pandas.tslib.Timestamp

Index

```
df.set_index('day', inplace=True)
```

↑ modify the orginal dataframe otherwise new dataframe.

~~df.~~
new_df = df.fillna(0)

NaN → 0

different value for column

new_df = df.fillna({
    'temp': 0,
    'wind': 0,
    'event': 'no event'
})

new_df = df.fillna(method="ffill")

carry forward

1/7	32	sunny
1/8	32	sunny
1/9	32	snow

forward

backward

new_df = df.fillna(method='bfill', axis="column")

copying value Horizontaly

df.fillna(method='ffill', limit=1)

copy value only once.

[Fillna]

```
new_df = df.interpolate()
```

linear interpolation

1/1	32
4/1	30
5/1	28

$\frac{32+28}{2}$

missing 2/1 and 3/1

```
df.interpolate(method="time")
```

1/1	32
4/1	29
5/1	28

the closest time.

```
new_df = df.dropna()
```

new_df ↑ drop the row where missing any data

```
df.dropna(thresh=1)
```

↑ keep the row if there one parameter

```
dt = pd.date_range("01-01-2017", "01-11-17")
idx = pd.DatetimeIndex(dt)
df.reindex(idx)
```

df ↑ How to get the dataframe?

handal missing data (pandas-6)

replace function

```
df = pd.read_csv('weather.csv')
new_df = df.replace(-999, np.NaN)
```

-999 ↑ value; np.NaN ↑ replace value

```
df.replace([-999,-888], np.Nan)
```

```
new_df = df.replace({
    'temp' : -999,
    'wind' : -999,
    'event' : '0'
}, np.Nan)
new_df
```

thin dictionary

colamm wise

Mapping

```
df.replace({
    -999 : np.NaN
    "~~noevent~~" : "sunny"
})
```

0 → in data from

thin mapping using for replace for thin point (difine) value.

if there are given with unite.

Temp	
32f	6Mph
0 f	2mph
28 F	5mph

Now chop off the unite

```
new_df = df.replace('[^A-Za-z]', '', regex = True)
```

→ it remove event column

so provide a dictonary:

```
df.replace({
    'temp' : '[A-Za-z]',
    'wind' : [A-Za-z].
})
```

creating data fram s

```
df = pd.Dataframe({
'Score' : ['excep', 'avg', 'poon', 'avg', 'excep'],
'student' : ['rod', 'Maya', 'prut', 'tom', 'Jaliah']
})
```

df.replace(['poor', 'avg', 'good', 'exep'], [1, 2, 3, 4])

Group By. panda-7

1) find maximum temperature in each of the cities.

2) find average wind speed per city.

```
g = df.groupby('city')

for city, city_df in g:
    print(city)
    print(city_df)
print city and city info.

g.get_group('Mumbai')
```

how to work

A.1)

g.Max( ) ← represent the

A.2)

g.mean()

g.describe()

Concat dataframe

```
us-wathe
india-wath = pd.Dataframe({
        'city': ['mumbai', 'delhi', 'banglon']
        'temp': [32,45,50]
        'himidity': [80,60,78]
})

df = pd.concat([india-wath, us-wath])
```

0	mumbat	0
1	dehli	1
2	bagalon	2
0	NY	3
1	chago	4
2	onlando	5

ignore_index = True)

keys = ['India', 'US']

retrive the data

df.loc["india"]