"https://raw.githubusercontent.com/justmarkham/DAT8/master/data/chipotle.tsv"

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

Basic DataFrame

Consider the following Python dictionary data and Python list labels:

```
data = {'birds': ['Cranes', 'Cranes', 'plovers', 'spoonbills', 'spoonbills', 'Cranes', 'plovers', 'Cranes', 'spoonbills', 'spoonbills', 'Cranes'],
        'age': [3.5, 4, 1.5, np.nan, 6, 3, 5.5, np.nan, 8, 4, 3.5], 'visits': [2, 4, 3, 4, 3, 4, 2, 2, 3, 2, 2],
        'priority': ['yes', 'yes', 'no', np.nan, 'no', 'no', 'no', 'yes', 'no', 'no','yes']}
```

```
labels = ['a', 'b', 'c', 'd', 'e', 'f', 'g', 'h', 'i', 'j', 'k']
```

Q-1:

i. Create a DataFrame birds from the above dictionary data which has the index labels.

ii. Display basic information about the dataFrame.

iii. Show Alternate rows of the dataframe.

```
# code here
data = {'birds': ['Cranes', 'Cranes', 'plovers', 'spoonbills', 'spoonbills', 'Cranes', 'plovers', 'Cranes', 'spoonbills', 'spoonbills', 'Cra
        'age': [3.5, 4, 1.5, np.nan, 6, 3, 5.5, np.nan, 8, 4, 3.5], 'visits': [2, 4, 3, 4, 3, 4, 2, 2, 3, 2, 2],
        'priority': ['yes', 'yes', 'no', np.nan, 'no', 'no', 'no', 'yes', 'no', 'no','yes']}
labels = ['a', 'b', 'c', 'd', 'e', 'f', 'g', 'h', 'i', 'j', 'k']
#1
df1 = pd.DataFrame(data = data, index = labels)
```

```
#2
df1.info()
df1.describe()
```

```
<class 'pandas.core.frame.DataFrame'>
Index: 11 entries, a to k
Data columns (total 4 columns):
 #   Column    Non-Null Count  Dtype
---  ------    --------------  -----
 0   birds     11 non-null     object
 1   age       9 non-null      float64
 2   visits    11 non-null     int64
 3   priority  10 non-null     object
dtypes: float64(1), int64(1), object(2)
memory usage: 440.0+ bytes
```

	age	visits
count	9.000000	11.000000
mean	4.333333	2.818182
std	1.903943	0.873863
min	1.500000	2.000000
25%	3.500000	2.000000
50%	4.000000	3.000000
75%	5.500000	3.500000
max	8.000000	4.000000

```
#3
df1.iloc[::2]
```

	birds	age	visits	priority
a	Cranes	3.5	2	yes
c	plovers	1.5	3	no
e	spoonbills	6.0	3	no
g	plovers	5.5	2	no
i	spoonbills	8.0	3	no
k	Cranes	3.5	2	yes

Q-2:

i. Show only rows [1st, 3rd, 7th] from columns ['bird', 'age']

ii. Select rows where the number of visits is less than 4.

```
# code here
#1
df1[['birds', 'age']].iloc[[0,2,6]]
```

	birds	age
a	Cranes	3.5
c	plovers	1.5
g	plovers	5.5

```
#2
df1[df1.visits<4]
```

	birds	age	visits	priority
a	Cranes	3.5	2	yes
c	plovers	1.5	3	no
e	spoonbills	6.0	3	no
g	plovers	5.5	2	no
h	Cranes	NaN	2	yes
i	spoonbills	8.0	3	no
j	spoonbills	4.0	2	no
k	Cranes	3.5	2	yes

Q-3:

i. Select all rows with nan values in age and visits column.

ii. Fill nan with respective series mode value.

```
# code here
#1
df1[df1.age.isna() | df1.visits.isna()]
```

	birds	age	visits	priority
d	spoonbills	NaN	4	NaN
h	Cranes	NaN	2	yes

```
#2
df1.age.fillna(df1.age.mode()[0], inplace=True)
df1.visits.fillna(df1.visits.mode()[0], inplace=True)
```

Q-4

i. Find the total number of visits of the bird Cranes

ii. Find the number of each type of birds in dataframe.

iii. Print no of duplicate rows

iv. Drop Duplicates rows and make this changes permanent. Show dataframe after changes.

```
# code here
#1
df1[df1.birds == "Cranes"].visits.sum()
```

```
     14
```

```
#2
df1.birds.value_counts()
```

```
     Cranes        5
     spoonbills    4
     plovers       2
     Name: birds, dtype: int64
```

```
#3
df1.duplicated().sum()
```

```
     2
```

```
#4
df1.drop_duplicates(inplace=True)
```

Question on IPL Data

IPL Data Link : https://drive.google.com/file/d/1yKVUuexl6lIKuFQy7uIPgDgXhJ0L4SIg/view?usp=share_link

https://www.kaggle.com/datasets/vora1011/ipl-2008-to-2021-all-match-dataset?select=IPL_Matches_2008_2022.csv

Download ipl matches 2008-2022 file.

Q-5: In IPL matches dataset some teams name has changed.

You will have to consider them as same.

```
 'Delhi Capitals' formerly as 'Delhi Daredevils'
 'Punjab Kings' formerly as 'Kings XI Punjab'
 'Rising Pune Supergiant' formerly as 'Rising Pune Supergiants'
```

You need to make changes accordingly. Consider current name for each teams.

Be careful Gujrat Titans and Gujrat Lions are different teams.

```
# code here
data = pd.read_csv('IPL_Matches_2008_2022.csv')
changed_name = {'Delhi Daredevils':'Delhi Capitals',
                'Kings XI Punjab':'Punjab Kings',
                'Rising Pune Supergiants':'Rising Pune Supergiant'}
data.replace(changed_name.keys(), changed_name.values(),inplace=True )
```

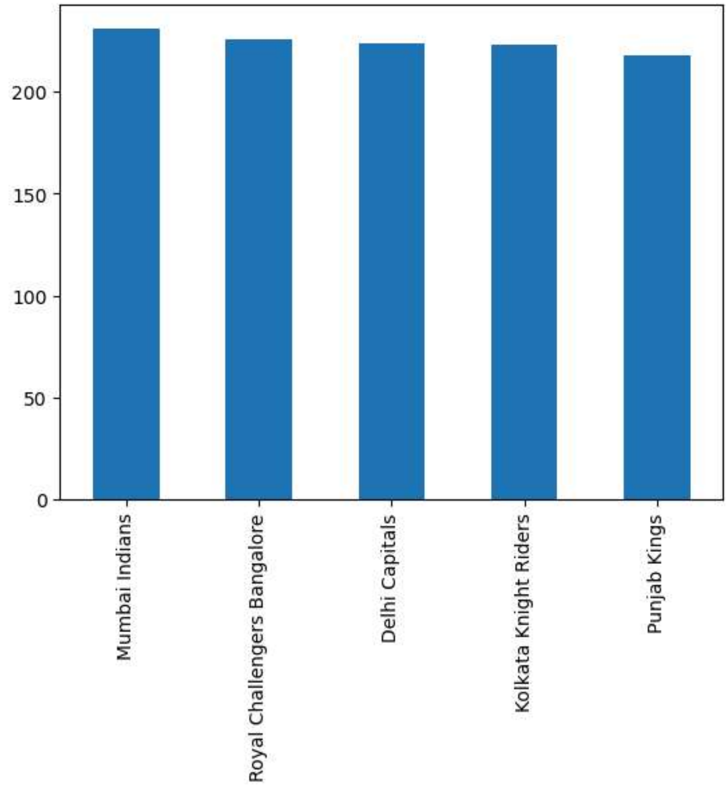
```
Start coding or generate with AI.
```

Q-6 Write a code which can display the bar chart of top 5 teams who have played maximum number of matches in the IPL.

> Hint: Be careful the data is divided in 2 different cols(Team 1 and Team 2)

```
# code here
# Considering both team slots
(data['Team1'].value_counts()+data["Team2"].value_counts()).sort_values(ascending=False).head().plot(kind='bar')
```

```
<AxesSubplot:>
```



Q-7: Player who got Most no. of player of the match award playing against Mumbai Indians.

> Just for this question assume player of the match award is given to players from winning team. Although this is true in most of the cases.

```
# code here
m1 = (data.Team1 == "Mumbai Indians") | (data.Team2 == "Mumbai Indians")
m2 = data.WinningTeam != "Mumbai Indians"
data[m1 & m2].Player_of_Match.value_counts().head(1)
```

```
SPD Smith    4
Name: Player_of_Match, dtype: int64
```

Q-8: Team1 vs Team2 Dashbord

Create a function which will take two string(name of two teams) as input. Show win Loss record between them and player getting most player of the match award in matches between these two teams.

```
team1_vs_team2('Kolkata Knight Riders','Chennai Super Kings')
```

```
# code here
def team1_vs_team2(t1, t2):
    m1 = (data.Team1 == t1) | (data.Team2 == t1)  # Filter for matches where only t1 played
    m2 = (data.Team1 == t2) | (data.Team2 == t2)  # Filter for matches where only t1 played
    df1 = data[m1 & m2]
    print(df1.WinningTeam.value_counts())
    print(df1.Player_of_Match.value_counts().head(1))

team1_vs_team2('Kolkata Knight Riders','Chennai Super Kings')
```

```
Chennai Super Kings      17
Kolkata Knight Riders     9
```

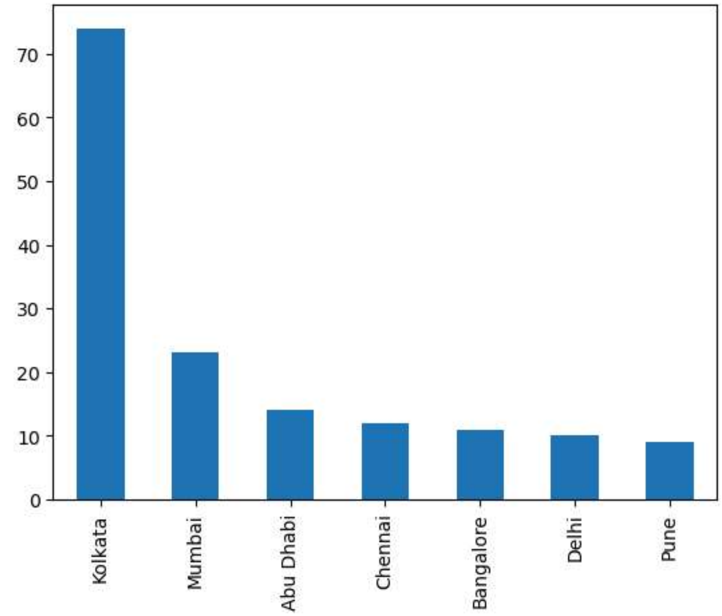
```
Name: WinningTeam, dtype: int64
RA Jadeja    3
Name: Player_of_Match, dtype: int64
```

`Q-9:` Find out the top 7 cities where the matches of Kolkata Knight Riders are played frequently and plot the result as bar chart.

`.plot(kind = "bar")` can help you to plot the bar chart. Also you can learn more about this method from [here]()

```
# code here
data[(data.Team1 == "Kolkata Knight Riders") | (data.Team2 == "Kolkata Knight Riders")].City.value_counts().head(7).plot(kind='bar')
```

```
<AxesSubplot:>
```



`Q-10:` Find out the average margin for the team Mumbai Indians for only the session 2011.

```
# code here
data[((data.Team1 == "Mumbai Indians") | (data.Team2 == "Mumbai Indians")) & (data.Season == "2011")].Margin.mean()
```

```
19.25
```

Start coding or generate with AI.