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

```
courses = pd.read_csv('/content/courses.csv')
students = pd.read_csv('/content/students.csv')
nov = pd.read_csv('/content/reg-month1.csv')
dec = pd.read_csv('/content/reg-month2.csv')

matches = pd.read_csv('/content/matches.csv')
delivery = pd.read_csv('/content/deliveries.csv')
```

```
dec
```

	student_id	course_id
0	3	5
1	16	7
2	12	10
3	12	1
4	14	9
5	7	7
6	7	2
7	16	3
8	17	10
9	11	8
10	14	6
11	12	5
12	12	7
13	18	8
14	1	10
15	1	9
16	2	5
17	7	6
18	22	5
19	22	6
20	23	9
21	23	5
22	14	4
23	14	1
24	11	10
25	42	9
26	50	8
27	38	1

```
# pd.concat
# df.concat
# ignore_index
# df.append
# mullitindex -> fetch using iloc
# concat dataframes horizontally
```

```
regs = pd.concat([nov,dec],ignore_index=True)
regs
```

	student_id	course_id
0	23	1
1	15	5
2	18	6
3	23	4
4	16	9
5	18	1
6	1	1
7	7	8
8	22	3
9	15	1
10	19	4
11	1	6
12	7	10
13	11	7
14	13	3
15	24	4
16	21	1
17	16	5
18	23	3
19	17	7
20	23	6
21	25	1
22	19	2
23	25	10
24	3	3
25	3	5
26	16	7
27	12	10
28	12	1
29	14	9
30	7	7
31	7	2
32	16	3
33	17	10
34	11	8
35	14	6
36	12	5
37	12	7
38	18	8
39	1	10
40	1	9
41	2	5
42	7	6
43	22	5
44	22	6
45	23	9

```
nov.append(dec,ignore_index=True)
```

	student_id	course_id
0	23	1
1	15	5
2	18	6
3	23	4
4	16	9
5	18	1
6	1	1
7	7	8
8	22	3

```
multi = pd.concat([nov,dec],keys=['Nov','Dec'])
# Multiindex DataFrame
multi.loc[('Dec',4)]
```

```
student_id    14
course_id      9
Name: (Dec, 4), dtype: int64
```

13 11 7

```
pd.concat([nov,dec],axis=1)
```

	student_id	course_id	student_id	course_id
0	23.0	1.0	3	5
1	15.0	5.0	16	7
2	18.0	6.0	12	10
3	23.0	4.0	12	1
4	16.0	9.0	14	9
5	18.0	1.0	7	7
6	1.0	1.0	7	2
7	7.0	8.0	16	3
8	22.0	3.0	17	10
9	15.0	1.0	11	8
10	19.0	4.0	14	6
11	1.0	6.0	12	5
12	7.0	10.0	12	7
13	11.0	7.0	18	8
14	13.0	3.0	1	10
15	24.0	4.0	1	9
16	21.0	1.0	2	5
17	16.0	5.0	7	6
18	23.0	3.0	22	5
19	17.0	7.0	22	6
20	23.0	6.0	23	9
21	25.0	1.0	23	5
22	19.0	2.0	14	4
23	25.0	10.0	14	1
24	3.0	3.0	11	10
25	NaN	NaN	42	9
26	NaN	NaN	50	8
27	NaN	NaN	38	1

44 22 6

```
# inner join
students.merge(regs,how='inner',on='student_id')
```

	student_id	name	partner	course_id
0	1	Kailash Harjo	23	1
1	1	Kailash Harjo	23	6
2	1	Kailash Harjo	23	10
3	1	Kailash Harjo	23	9
4	2	Esha Butala	1	5

```
# left join
courses.merge(regs,how='left',on='course_id')
```

	course_id	course_name	price	student_id
0	1	python	2499	23.0
1	1	python	2499	18.0
2	1	python	2499	1.0
3	1	python	2499	15.0
4	1	python	2499	21.0
5	1	python	2499	25.0
6	1	python	2499	12.0
7	1	python	2499	14.0
8	1	python	2499	38.0
9	2	sql	3499	19.0
10	2	sql	3499	7.0
11	3	data analysis	4999	22.0
12	3	data analysis	4999	13.0
13	3	data analysis	4999	23.0
14	3	data analysis	4999	3.0
15	3	data analysis	4999	16.0
16	4	machine learning	9999	23.0

```
# right join
temp_df = pd.DataFrame({
    'student_id':[26,27,28],
    'name':['Nitish','Ankit','Rahul'],
    'partner':[28,26,17]
})

students = pd.concat([students,temp_df],ignore_index=True)
```

```
students.tail()
```

	student_id	name	partner
23	24	Radhika Suri	17
24	25	Shashank D'Alia	2
25	26	Nitish	28
26	27	Ankit	26
27	28	Rahul	17

```
students.merge(regs,how='right',on='student_id')
```

	student_id	name	partner	course_id
0	23	Chhavi Lachman	18.0	1
1	15	Preet Sha	16.0	5
2	18	Fardeen Mahabir	13.0	6
3	23	Chhavi Lachman	18.0	4
4	16	Elias Dodiya	25.0	9
5	18	Fardeen Mahabir	13.0	1
6	1	Kailash Harjo	23.0	1
7	7	Tarun Thaker	9.0	8
8	22	Yash Sethi	21.0	3
9	15	Preet Sha	16.0	1
10	19	Qabeel Raman	12.0	4
11	1	Kailash Harjo	23.0	6
12	7	Tarun Thaker	9.0	10
13	11	David Mukhopadhyay	20.0	7
14	13	Munni Varghese	24.0	3
15	24	Radhika Suri	17.0	4
16	21	Seema Kota	15.0	1
17	16	Elias Dodiya	25.0	5
18	23	Chhavi Lachman	18.0	3
19	17	Yasmin Palan	7.0	7
20	23	Chhavi Lachman	18.0	6
21	25	Shashank D’Alia	2.0	1
22	19	Qabeel Raman	12.0	2
23	25	Shashank D’Alia	2.0	10
24	3	Parveen Bhalla	3.0	3
25	3	Parveen Bhalla	3.0	5
26	16	Elias Dodiya	25.0	7
27	12	Radha Dutt	19.0	10
28	12	Radha Dutt	19.0	1
29	14	Pranab Natarajan	22.0	9
30	7	Tarun Thaker	9.0	7
31	7	Tarun Thaker	9.0	2
32	16	Elias Dodiya	25.0	3
33	17	Yasmin Palan	7.0	10
34	11	David Mukhopadhyay	20.0	8
35	14	Pranab Natarajan	22.0	6
36	12	Radha Dutt	19.0	5
37	12	Radha Dutt	19.0	7
38	18	Fardeen Mahabir	13.0	8
39	1	Kailash Harjo	23.0	10
40	1	Kailash Harjo	23.0	9

```
regs.merge(students,how='left',on='student_id')
```

	student_id	course_id	name	partner
0	23	1	Chhavi Lachman	18.0
1	15	5	Preet Sha	16.0
2	18	6	Fardeen Mahabir	13.0
3	23	4	Chhavi Lachman	18.0
4	16	9	Elias Dodiya	25.0
5	18	1	Fardeen Mahabir	13.0
6	1	1	Kailash Harjo	23.0
7	7	8	Tarun Thaker	9.0
8	22	3	Yash Sethi	21.0
9	15	1	Preet Sha	16.0
10	19	4	Qabeel Raman	12.0
11	1	6	Kailash Harjo	23.0
12	7	10	Tarun Thaker	9.0
13	11	7	David Mukhopadhyay	20.0
14	13	3	Munni Varghese	24.0
15	24	4	Radhika Suri	17.0
16	21	1	Seema Kota	15.0
17	16	5	Elias Dodiya	25.0
18	23	3	Chhavi Lachman	18.0
19	17	7	Yasmin Palan	7.0
20	23	6	Chhavi Lachman	18.0
21	25	1	Shashank D’Alia	2.0
22	19	2	Qabeel Raman	12.0
23	25	10	Shashank D’Alia	2.0
24	3	3	Parveen Bhalla	3.0
25	3	5	Parveen Bhalla	3.0
26	16	7	Elias Dodiya	25.0
27	12	10	Radha Dutt	19.0
28	12	1	Radha Dutt	19.0
29	14	9	Pranab Natarajan	22.0
30	7	7	Tarun Thaker	9.0
31	7	2	Tarun Thaker	9.0
32	16	3	Elias Dodiya	25.0
33	17	10	Yasmin Palan	7.0
34	11	8	David Mukhopadhyay	20.0
35	14	6	Pranab Natarajan	22.0
36	12	5	Radha Dutt	19.0
37	12	7	Radha Dutt	19.0
38	18	8	Fardeen Mahabir	13.0
39	1	10	Kailash Harjo	23.0
40	1	9	Kailash Harjo	23.0
41	2	5	Esha Butala	1.0
42	7	6	Tarun Thaker	9.0
43	22	5	Yash Sethi	21.0
44	22	6	Yash Sethi	21.0
45	23	9	Chhavi Lachman	18.0

```
# outer join
students.merge(regs,how='outer',on='student_id').tail(10)
```

	student_id	name	partner	course_id
53	23	Chhavi Lachman	18.0	5.0
54	24	Radhika Suri	17.0	4.0
55	25	Shashank D’Alia	2.0	1.0
56	25	Shashank D’Alia	2.0	10.0
57	26	Nitish	28.0	NaN
58	27	Ankit	26.0	NaN
59	28	Rahul	17.0	NaN
60	42	NaN	NaN	9.0
61	50	NaN	NaN	8.0
62	38	NaN	NaN	1.0

```
# 1. find total revenue generated
total = regs.merge(courses,how='inner',on='course_id')['price'].sum()
total
```

```
154247
```

```
# 2. find month by month revenue
temp_df = pd.concat([nov,dec],keys=['Nov','Dec']).reset_index()
temp_df.merge(courses,on='course_id').groupby('level_0')['price'].sum()
```

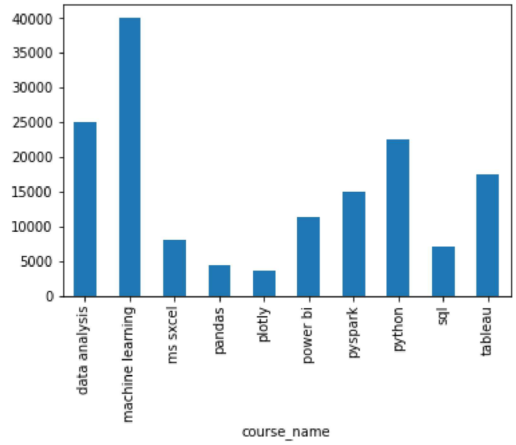
```
level_0
Dec    65072
Nov    89175
Name: price, dtype: int64
```

```
# 3. Print the registration table
# cols -> name -> course -> price
regs.merge(students,on='student_id').merge(courses,on='course_id')[['name','course_name','price']]
```

	name	course_name	price
0	Chhavi Lachman	python	2499
1	Preet Sha	python	2499
2	Fardeen Mahabir	python	2499
3	Kailash Harjo	python	2499
4	Seema Kota	python	2499
5	Shashank D’Alia	python	2499
6	Radha Dutt	python	2499
7	Pranab Natarajan	python	2499
8	Chhavi Lachman	machine learning	9999
9	Qabeel Raman	machine learning	9999
10	Radhika Suri	machine learning	9999
11	Pranab Natarajan	machine learning	9999
12	Chhavi Lachman	data analysis	4999
13	Elias Dodiya	data analysis	4999
14	Yash Sethi	data analysis	4999
15	Munni Varghese	data analysis	4999
16	Parveen Bhalla	data analysis	4999
17	Chhavi Lachman	power bi	1899
18	Fardeen Mahabir	power bi	1899
19	Kailash Harjo	power bi	1899
20	Tarun Thaker	power bi	1899
21	Yash Sethi	power bi	1899
22	Pranab Natarajan	power bi	1899
23	Chhavi Lachman	plotly	699
24	Elias Dodiya	plotly	699
25	Kailash Harjo	plotly	699
26	Pranab Natarajan	plotly	699
27	Chhavi Lachman	tableau	2499
28	Preet Sha	tableau	2499
29	Elias Dodiya	tableau	2499

```
# 4. Plot bar chart for revenue/course
regs.merge(courses,on='course_id').groupby('course_name')['price'].sum().plot(kind='bar')
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7f56b73cb2e0>
```



```
# 5. find students who enrolled in both the months
common_student_id = np.intersect1d(nov['student_id'],dec['student_id'])
common_student_id
```

```
array([ 1,  3,  7, 11, 16, 17, 18, 22, 23])
```

```
students[students['student_id'].isin(common_student_id)]
```

	student_id	name	partner
0	1	Kailash Harjo	23
2	3	Parveen Bhalla	3
6	7	Tarun Thaker	9
10	11	David Mukhopadhyay	20
15	16	Elias Dodiya	25
16	17	Yasmin Palan	7
17	18	Fardeen Mahabir	13
21	22	Yash Sethi	21
22	23	Chhavi Lachman	18

```
# 6. find course that got no enrollment
# courses['course_id']
# regs['course_id']

course_id_list = np.setdiff1d(courses['course_id'],regs['course_id'])
courses[courses['course_id'].isin(course_id_list)]
```

	course_id	course_name	price
10	11	Numpy	699
11	12	C++	1299

```
# 7. find students who did not enroll into any courses
student_id_list = np.setdiff1d(students['student_id'],regs['student_id'])
students[students['student_id'].isin(student_id_list)].shape[0]

(10/28)*100
```

```
35.714285714285715
```

```
students
```

	student_id	name	partner
0	1	Kailash Harjo	23
1	2	Esha Butala	1
2	3	Parveen Bhalla	3
3	4	Marlo Dugal	14
4	5	Kusum Bahri	6
5	6	Lakshmi Contractor	10
6	7	Tarun Thaker	9
7	8	Radheshyam Dey	5
8	9	Nitika Chatterjee	4
9	10	Aayushman Sant	8
10	11	David Mukhopadhyay	20
11	12	Radha Dutt	19

```
# 8. Print student name -> partner name for all enrolled students
# self join
students.merge(students,how='inner',left_on='partner',right_on='student_id')[['name_x','name_y']]
```

	name_x	name_y
0	Kailash Harjo	Chhavi Lachman
1	Esha Butala	Kailash Harjo
2	Parveen Bhalla	Parveen Bhalla
3	Marlo Dugal	Pranab Natarajan
4	Kusum Bahri	Lakshmi Contractor
5	Lakshmi Contractor	Aayushman Sant
6	Tarun Thaker	Nitika Chatterjee
7	Radheshyam Dey	Kusum Bahri
8	Nitika Chatterjee	Marlo Dugal
9	Aayushman Sant	Radheshyam Dey
10	David Mukhopadhyay	Hanuman Hegde
11	Radha Dutt	Qabeel Raman
12	Munni Varghese	Radhika Suri
13	Pranab Natarajan	Yash Sethi
14	Preet Sha	Elias Dodiya
15	Elias Dodiya	Shashank D’Alia
16	Yasmin Palan	Tarun Thaker
17	Fardeen Mahabir	Munni Varghese
18	Qabeel Raman	Radha Dutt
19	Hanuman Hegde	David Mukhopadhyay
20	Seema Kota	Preet Sha
21	Yash Sethi	Seema Kota
22	Chhavi Lachman	Fardeen Mahabir
23	Radhika Suri	Yasmin Palan
24	Rahul	Yasmin Palan
25	Shashank D’Alia	Esha Butala
26	Nitish	Rahul
27	Ankit	Nitish

```
# 9. find top 3 students who did most number enrollments
regs.merge(students,on='student_id').groupby(['student_id','name'])['name'].count().sort_values(ascending=False).head(3)
```

```
student_id  name
23          Chhavi Lachman    6
7           Tarun Thaker      5
1           Kailash Harjo     4
Name: name, dtype: int64
```

```
# 10. find top 3 students who spent most amount of money on courses
regs.merge(students,on='student_id').merge(courses,on='course_id').groupby(['student_id','name'])['price'].sum().sort_values(ascending=False)
```

```
student_id  name
23          Chhavi Lachman      22594
14          Pranab Natarajan    15096
19          Qabeel Raman        13498
Name: price, dtype: int64
```

```
# Alternate syntax for merge
# students.merge(regs)

pd.merge(students,regs,how='inner',on='student_id')
```

	student_id	name	partner	course_id
0	1	Kailash Harjo	23	1
1	1	Kailash Harjo	23	6
2	1	Kailash Harjo	23	10
3	1	Kailash Harjo	23	9
4	2	Esha Butala	1	5
5	3	Parveen Bhalla	3	3
6	3	Parveen Bhalla	3	5
7	7	Tarun Thaker	9	8
8	7	Tarun Thaker	9	10
9	7	Tarun Thaker	9	7
10	7	Tarun Thaker	9	2
11	7	Tarun Thaker	9	6
12	11	David Mukhopadhyay	20	7
13	11	David Mukhopadhyay	20	8
14	11	David Mukhopadhyay	20	10
15	12	Radha Dutt	19	10
16	12	Radha Dutt	19	1
17	12	Radha Dutt	19	5

```
# IPL Problems

# find top 3 studiums with highest sixes/match ratio
# find orange cap holder of all the seasons
```

```
matches
```

	id	season	city	date	team1	team2	toss_winner	toss_decision	result	dl_appli
0	1	2017	Hyderabad	2017-04-05	Sunrisers Hyderabad	Royal Challengers Bangalore	Royal Challengers Bangalore	field	normal	

```
delivery
```

	match_id	inning	batting_team	bowling_team	over	ball	batsman	non_striker	bowler	is_supe
0	1	1	Sunrisers Hyderabad	Royal Challengers Bangalore	1	1	DA Warner	S Dhawan	TS Mills	
1	1	1	Sunrisers Hyderabad	Royal Challengers Bangalore	1	2	DA Warner	S Dhawan	TS Mills	
2	1	1	Sunrisers Hyderabad	Royal Challengers Bangalore	1	3	DA Warner	S Dhawan	TS Mills	
3	1	1	Sunrisers Hyderabad	Royal Challengers Bangalore	1	4	DA Warner	S Dhawan	TS Mills	
4	1	1	Sunrisers Hyderabad	Royal Challengers Bangalore	1	5	DA Warner	S Dhawan	TS Mills	
...	...	...	...	...	...	...	...	...	...	
150455	636	2	Royal Challengers Bangalore	Sunrisers Hyderabad	20	2	Sachin Baby	CJ Jordan	B Kumar	
150456	636	2	Royal Challengers Bangalore	Sunrisers Hyderabad	20	3	Sachin Baby	CJ Jordan	B Kumar	
150457	636	2	Royal Challengers Bangalore	Sunrisers Hyderabad	20	4	Iqbal Abdulla	Sachin Baby	B Kumar	
150458	636	2	Royal Challengers Bangalore	Sunrisers Hyderabad	20	5	Sachin Baby	Iqbal Abdulla	B Kumar	
150459	636	2	Royal Challengers Bangalore	Sunrisers Hyderabad	20	6	Iqbal Abdulla	Sachin Baby	B Kumar	

150460 rows × 21 columns

```
temp_df = delivery.merge(matches,left_on='match_id',right_on='id')
```

```
six_df = temp_df[temp_df['batsman_runs'] == 6]
```

```
# stadium -> sixes
num_sixes = six_df.groupby('venue')['venue'].count()
```

```
num_matches = matches['venue'].value_counts()
```

```
(num_sixes/num_matches).sort_values(ascending=False).head(10)
```

```
Holkar Cricket Stadium                                 17.600000
M Chinnaswamy Stadium                                  13.227273
Sharjah Cricket Stadium                                12.666667
Himachal Pradesh Cricket Association Stadium           12.000000
Dr. Y.S. Rajasekhara Reddy ACA-VDCA Cricket Stadium    11.727273
Wankhede Stadium                                       11.526316
De Beers Diamond Oval                                  11.333333
Maharashtra Cricket Association Stadium                11.266667
JSCA International Stadium Complex                     10.857143
Sardar Patel Stadium, Motera                           10.833333
Name: venue, dtype: float64
```

```
matches
```

	id	season	city	date	team1	team2	toss_winner	toss_decision	result	dl_appli
0	1	2017	Hyderabad	2017-04-05	Sunrisers Hyderabad	Royal Challengers Bangalore	Royal Challengers Bangalore	field	normal	
1	2	2017	Pune	2017-04-06	Mumbai Indians	Rising Pune Supergiant	Rising Pune Supergiant	field	normal	
2	3	2017	Rajkot	2017-04-07	Gujarat Lions	Kolkata Knight Riders	Kolkata Knight Riders	field	normal	
3	4	2017	Indore	2017-04-08	Rising Pune Supergiant	Kings XI Punjab	Kings XI Punjab	field	normal	
4	5	2017	Bangalore	2017-04-08	Royal Challengers Bangalore	Delhi Daredevils	Royal Challengers Bangalore	bat	normal	
...	...	...	...	...	...	...	...	...	...	
631	632	2016	Raipur	2016-05-22	Delhi Daredevils	Royal Challengers Bangalore	Royal Challengers Bangalore	field	normal	
632	633	2016	Bangalore	2016-05-24	Gujarat Lions	Royal Challengers Bangalore	Royal Challengers Bangalore	field	normal	
633	634	2016	Delhi	2016-05-25	Sunrisers Hyderabad	Kolkata Knight Riders	Kolkata Knight Riders	field	normal	
634	635	2016	Delhi	2016-05-27	Gujarat Lions	Sunrisers Hyderabad	Sunrisers Hyderabad	field	normal	
635	636	2016	Bangalore	2016-05-29	Sunrisers Hyderabad	Royal Challengers Bangalore	Sunrisers Hyderabad	bat	normal	

636 rows × 18 columns

```
temp_df.groupby(['season','batsman'])['batsman_runs'].sum().reset_index().sort_values('batsman_runs',ascending=False).drop_duplicates(subset=
```

	season	batsman	batsman_runs
115	2008	SE Marsh	616
229	2009	ML Hayden	572
446	2010	SR Tendulkar	618
502	2011	CH Gayle	608
684	2012	CH Gayle	733
910	2013	MEK Hussey	733
1088	2014	RV Uthappa	660
1148	2015	DA Warner	562
1383	2016	V Kohli	973
1422	2017	DA Warner	641

```
temp_df.groupby(['season','batsman'])['batsman_runs'].sum().reset_index().sort_values('batsman_runs',ascending=False)
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
<ipython-input-214-5a08c989d1a3> in <module>
----> 1 temp_df.groupby(['season','batsman'])
['batsman_runs'].sum().reset_index().sort_values('batsman_runs',ascending=False).first('season')

/usr/local/lib/python3.8/dist-packages/pandas/core/generic.py in first(self, offset)
   8193         """
   8194         if not isinstance(self.index, DatetimeIndex):
-> 8195             raise TypeError("'first' only supports a DatetimeIndex index")
   8196


TypeError: 'first' only supports a DatetimeIndex index
```

SEARCH STACK OVERFLOW