

```
In [ ]: #Leftover from last week
```

```
In [1]: import json
```

```
In [2]: dict = {  
    "FirstName": "Jonathan",  
    "LastName": "Freeman",  
    "LoginCount": 4,  
    "isWriter": True,  
    "WorksWith": ['Spantree Technology Group', 'InfoWorld'],  
    "Pets": [  
        {  
            "name": "Lilly",  
            "type": "Raccoon"  
        }  
    ]  
}  
dict
```

```
Out[2]: {'FirstName': 'Jonathan',  
        'LastName': 'Freeman',  
        'LoginCount': 4,  
        'isWriter': True,  
        'WorksWith': ['Spantree Technology Group', 'InfoWorld'],  
        'Pets': [{'name': 'Lilly', 'type': 'Raccoon'}]}
```

```
In [3]: person='{"name":"Bob", "languages": ["English", "French"]}'  
person_dict=json.loads(person)
```

```
In [4]: person_dict
```

```
Out[4]: {'name': 'Bob', 'languages': ['English', 'French']}
```

```
In [5]: person_dict['languages']
```

```
Out[5]: ['English', 'French']
```

```
In [ ]: #general import command  
with open('path_to_file/filename.json') as f:  
    data=json.load(f)
```

```
In [6]: with open('test.json') as f:  
    data=json.load(f)
```

```
In [7]: data
```

```
{'Names': {'0': 'John', '1': 'Sal', '2': 'Tim', '3': 'Rod'},
```

Out[7]: 'Age': {'0': 33, '1': 45, '2': 22, '3': 54}}

In [8]: dict_json=json.dumps(dict)

In [9]: dict_json

Out[9]: '{"FirstName": "Jonathan", "LastName": "Freeman", "LoginCount": 4, "isWriter": true, "WorksWith": ["Spantree Technology Group", "InfoWorld"], "Pets": [{"name": "Lilly", "type": "Raccoon"}]}'

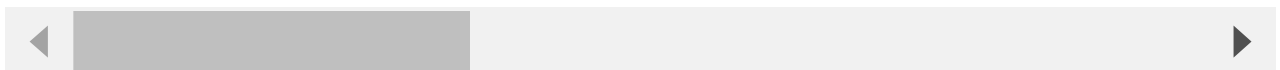
In [11]: *#new this week*
import pandas as pd
dfp = pd.read_excel(r'Absenteeism_at_work.xls')
dfp.head(5)

Out[11]:

	ID	Reason_for_absence	Month_of_absence	Day_of_the_week	Seasons	Transportation_expense	Distance
--	----	--------------------	------------------	-----------------	---------	------------------------	----------

0	11	26	7	3	1	289	
1	36	0	7	3	1	118	
2	3	23	7	4	1	179	
3	7	7	7	5	1	279	
4	11	23	7	5	1	289	

5 rows × 21 columns



In [13]: dfJson = pd.read_json(r'test.json')
dfJson.head(5)

Out[13]:

	Names	Age
--	-------	-----

0	John	33
1	Sal	45
2	Tim	22
3	Rod	54

In [14]: dfURL = pd.read_csv(r'https://archive.ics.uci.edu/ml/machine-learning-databases/abalone
'Length', 'Diameter', 'Height', 'Whole weight', 'Shucked weight',
dfURL.head(5)

Out[14]:

	Sex	Length	Diameter	Height	Whole weight	Shucked weight	Viscera weight	Shell weight	Rings
0	M	0.455	0.365	0.095	0.5140	0.2245	0.1010	0.150	15

	Sex	Length	Diameter	Height	Whole weight	Shucked weight	Viscera weight	Shell weight	Rings
1	M	0.350	0.265	0.090	0.2255	0.0995	0.0485	0.070	7
2	F	0.530	0.420	0.135	0.6770	0.2565	0.1415	0.210	9
3	M	0.440	0.365	0.125	0.5160	0.2155	0.1140	0.155	10
4	I	0.330	0.255	0.080	0.2050	0.0895	0.0395	0.055	7

In [15]:

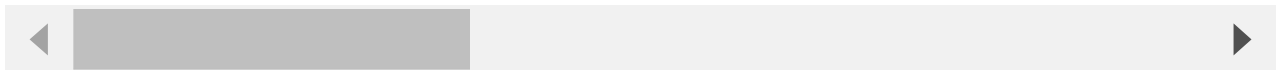
```
from pandasql import sqldf
dfpsql = pd.read_excel(r'Absenteeism_at_work.xls')

Query_string = """ select * from dfpsql limit 5 """
sqldf(Query_string, globals())
```

Out[15]:

	ID	Reason_for_absence	Month_of_absence	Day_of_the_week	Seasons	Transportation_expense	Dis
0	11		26	7	3	1	289
1	36		0	7	3	1	118
2	3		23	7	4	1	179
3	7		7	7	5	1	279
4	11		23	7	5	1	289

5 rows x 21 columns



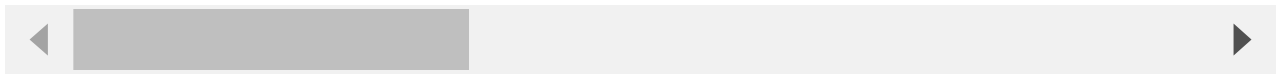
In [16]:

```
dfp[(dfp['Age'] >=30) & (dfp['Age'] <=45)]
```

Out[16]:

	ID	Reason_for_absence	Month_of_absence	Day_of_the_week	Seasons	Transportation_expense	Dis
0	11		26	7	3	1	289
2	3		23	7	4	1	179
3	7		7	7	5	1	279
4	11		23	7	5	1	289
5	3		23	7	6	1	179
...
734	13		13	7	2	1	369
735	11		14	7	3	1	289
736	1		11	7	3	1	235
737	4		0	0	3	1	118
738	8		0	0	4	2	231

526 rows × 21 columns



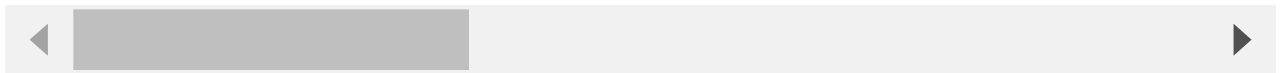
In [17]:

```
Query_string = """ select * from dfpsql where age>=30 and age<=45 """
sqldf(Query_string, globals())
```

Out[17]:

	ID	Reason_for_absence	Month_of_absence	Day_of_the_week	Seasons	Transportation_expense	Di
0	11	26	7	3	1	289	
1	3	23	7	4	1	179	
2	7	7	7	5	1	279	
3	11	23	7	5	1	289	
4	3	23	7	6	1	179	
...
521	13	13	7	2	1	369	
522	11	14	7	3	1	289	
523	1	11	7	3	1	235	
524	4	0	0	3	1	118	
525	8	0	0	4	2	231	

526 rows × 21 columns



In [18]:

```
dfp['ID'].unique()
```

Out[18]:

```
array([11, 36, 3, 7, 10, 20, 14, 1, 24, 6, 33, 18, 30, 2, 19, 27, 34,
       5, 15, 29, 28, 13, 22, 17, 31, 23, 32, 9, 26, 21, 8, 25, 12, 16,
       4, 35], dtype=int64)
```

In [19]:

```
Query_string = """ select distinct ID from dfpsql;"""
sqldf(Query_string, globals())
```

Out[19]:

	ID
0	11
1	36
2	3
3	7
4	10
5	20

	ID
6	14
7	1
8	24
9	6
10	33
11	18
12	30
13	2
14	19
15	27
16	34
17	5
18	15
19	29
20	28
21	13
22	22
23	17
24	31
25	23
26	32
27	9
28	26
29	21
30	8
31	25
32	12
33	16
34	4
35	35

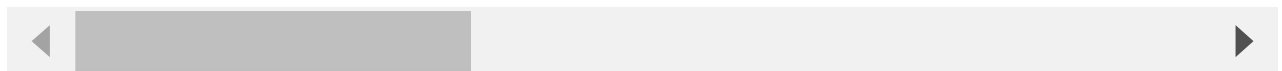
In [20]: `dfp[dfp.Age.isin([20,30,40])]`

Out[20]:

	ID	Reason_for_absence	Month_of_absence	Day_of_the_week	Seasons	Transportation_expense	Di
	47	15	23	9	5	1	291
	49	15	14	9	2	4	291
	65	22	23	10	5	4	179
	71	15	23	10	5	4	291
	75	15	14	10	3	4	291

	713	22	27	6	6	3	179
	717	22	13	6	5	3	179
	718	15	28	6	5	3	291
	719	22	13	6	2	1	179
	737	4	0	0	3	1	118

104 rows x 21 columns



In [21]:

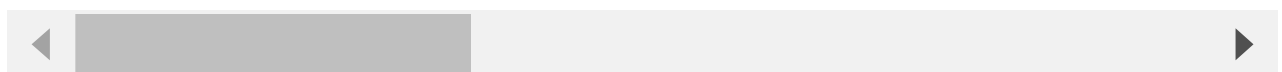
```
Query_string = """ select * from dfpsql where Age in(20,30,40);"""
sqlidf(Query_string, globals())
```

Out[21]:

	ID	Reason_for_absence	Month_of_absence	Day_of_the_week	Seasons	Transportation_expense	Di
	0	15	23	9	5	1	291
	1	15	14	9	2	4	291
	2	22	23	10	5	4	179
	3	15	23	10	5	4	291
	4	15	14	10	3	4	291

	99	22	27	6	6	3	179
	100	22	13	6	5	3	179
	101	15	28	6	5	3	291
	102	22	13	6	2	1	179
	103	4	0	0	3	1	118

104 rows x 21 columns



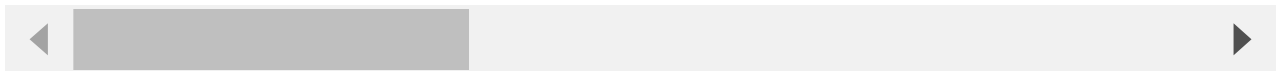
In [22]:

```
dfp[~dfp.Age.isin([20,30,40])]
```

Out[22]:

	ID	Reason_for_absence	Month_of_absence	Day_of_the_week	Seasons	Transportation_expense	Di
0	11		26	7	3	1	289
1	36		0	7	3	1	118
2	3		23	7	4	1	179
3	7		7	7	5	1	279
4	11		23	7	5	1	289
...
734	13		13	7	2	1	369
735	11		14	7	3	1	289
736	1		11	7	3	1	235
738	8		0	0	4	2	231
739	35		0	0	6	3	179

636 rows × 21 columns



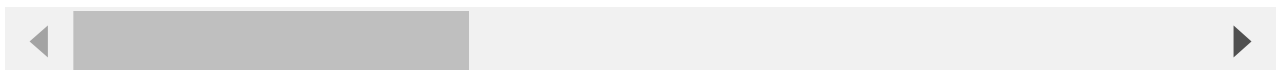
In [23]:

```
Query_string = """ select * from dfpsql where Age not in(20,30,40);"""
sqlidf(Query_string, globals())
```

Out[23]:

	ID	Reason_for_absence	Month_of_absence	Day_of_the_week	Seasons	Transportation_expense	Di
0	11		26	7	3	1	289
1	36		0	7	3	1	118
2	3		23	7	4	1	179
3	7		7	7	5	1	279
4	11		23	7	5	1	289
...
631	13		13	7	2	1	369
632	11		14	7	3	1	289
633	1		11	7	3	1	235
634	8		0	0	4	2	231
635	35		0	0	6	3	179

636 rows × 21 columns



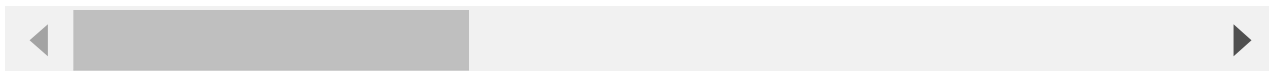
```
In [24]: dfp.sort_values(by = ['Age', 'Service_time'], ascending= True)
```

```
Out[24]:
```

	ID	Reason_for_absence	Month_of_absence	Day_of_the_week	Seasons	Transportation_expense	Di
	40	27	23	9	3	1	184
	118	27	23	1	5	2	184
	132	27	23	1	5	2	184
	137	27	23	2	6	2	184
	149	27	23	2	3	2	184

	620	9	25	3	3	2	228
	622	9	12	3	3	2	228
	640	9	25	3	4	2	228
	727	9	6	7	2	1	228
	729	9	6	7	3	1	228

740 rows × 21 columns



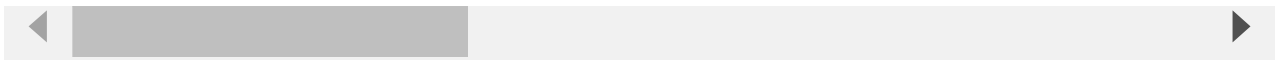
```
In [25]: Query_string = """ select * from dfpsql order by Age,Service_time;"""
         sqlidf(Query_string, globals())
```

```
Out[25]:
```

	ID	Reason_for_absence	Month_of_absence	Day_of_the_week	Seasons	Transportation_expense	Di
	0	27	23	9	3	1	184
	1	27	23	1	5	2	184
	2	27	23	1	5	2	184
	3	27	23	2	6	2	184
	4	27	23	2	3	2	184

	735	9	25	3	3	2	228
	736	9	12	3	3	2	228
	737	9	25	3	4	2	228
	738	9	6	7	2	1	228
	739	9	6	7	3	1	228

740 rows × 21 columns



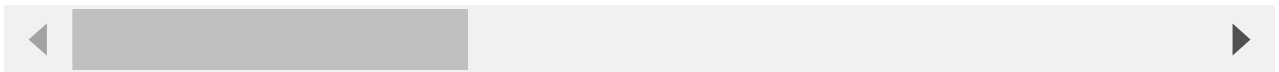
In [26]:

```
#Descending
dfp.sort_values(by = ['Age', 'Service_time'], ascending= False)
```

Out[26]:

	ID	Reason_for_absence	Month_of_absence	Day_of_the_week	Seasons	Transportation_expense	Di
255	9	18	8	3	1	228	
434	9	18	5	4	3	228	
521	9	1	10	4	4	228	
620	9	25	3	3	2	228	
622	9	12	3	3	2	228	
...
132	27	23	1	5	2	184	
137	27	23	2	6	2	184	
149	27	23	2	3	2	184	
209	27	7	5	4	3	184	
269	27	6	8	4	1	184	

740 rows × 21 columns



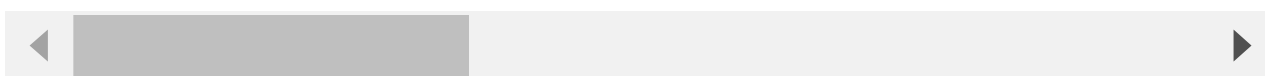
In [27]:

```
Query_string = """ select * from dfpsql order by Age Desc,Service_time Desc;"""
sqldf(Query_string, globals())
```

Out[27]:

	ID	Reason_for_absence	Month_of_absence	Day_of_the_week	Seasons	Transportation_expense	Di
0	9	18	8	3	1	228	
1	9	18	5	4	3	228	
2	9	1	10	4	4	228	
3	9	25	3	3	2	228	
4	9	12	3	3	2	228	
...
735	27	23	1	5	2	184	
736	27	23	2	6	2	184	
737	27	23	2	3	2	184	
738	27	7	5	4	3	184	
739	27	6	8	4	1	184	

740 rows × 21 columns



In [28]: `dfp.agg({'Transportation_expense': ['count', 'min', 'max', 'mean']})`

Out[28]: **Transportation_expense**

count	740.00000
min	118.00000
max	388.00000
mean	221.32973

In [29]: `Query_string = """ select count(Transportation_expense) as count, min(Transportation_ex
sqldf(Query_string, globals())`

Out[29]: **count min max mean**

0	740	118	388	221.32973
----------	-----	-----	-----	-----------

In [30]: `dfp.groupby('ID')['Service_time'].sum()`

Out[30]:

ID	
1	322
2	72
3	2034
4	13
5	247
6	104
7	84
8	28
9	128
10	72
11	520
12	7
13	180
14	406
15	444
16	48
17	340
18	64
19	36
20	462
21	24
22	414
23	88
24	480
25	80
26	65
27	49
28	684
29	69
30	42
31	27

```
32     145
33     336
34     550
35      14
36     612
Name: Service_time, dtype: int64
```

```
In [31]: Query_string = """ select ID , sum(Service_time) as Sum_Service_time from dfp
group by ID;"""
sqldf(Query_string, globals())
```

```
Out[31]:
```

	ID	Sum_Service_time
0	1	322
1	2	72
2	3	2034
3	4	13
4	5	247
5	6	104
6	7	84
7	8	28
8	9	128
9	10	72
10	11	520
11	12	7
12	13	180
13	14	406
14	15	444
15	16	48
16	17	340
17	18	64
18	19	36
19	20	462
20	21	24
21	22	414
22	23	88
23	24	480
24	25	80
25	26	65
26	27	49

	ID	Sum_Service_time
27	28	684
28	29	69
29	30	42
30	31	27
31	32	145
32	33	336
33	34	550
34	35	14
35	36	612

In [32]: `dfp.groupby('Reason_for_absence').agg({'Age': ['mean', 'min', 'max']})`

Out[32]:

	Age		
	mean	min	max

Reason_for_absence			
0	39.604651	28	53
1	37.687500	28	58
2	28.000000	28	28
3	40.000000	40	40
4	45.000000	41	49
5	41.666667	37	50
6	38.500000	27	58
7	32.866667	27	46
8	36.500000	28	40
9	36.750000	28	41
10	35.960000	28	49
11	35.115385	28	50
12	36.125000	28	58
13	36.490909	28	50
14	41.052632	28	50
15	39.500000	36	43
16	37.333333	32	40
17	40.000000	40	40
18	37.619048	28	58

	Age		
	mean	min	max
Reason_for_absence			
19	35.450000	28	50
21	37.166667	30	40
22	32.657895	28	41
23	35.610738	27	50
24	33.000000	33	33
25	36.161290	28	58
26	37.848485	28	50
27	35.507246	28	38
28	37.508929	28	50

```
In [33]: Query_string = """ select Reason_for_absence , avg(Age) as mean, min(Age) as min, max(A
group by Reason_for_absence;"""
sqlidf(Query_string, globals())
```

```
Out[33]:
```

	Reason_for_absence	mean	min	max
0	0	39.604651	28	53
1	1	37.687500	28	58
2	2	28.000000	28	28
3	3	40.000000	40	40
4	4	45.000000	41	49
5	5	41.666667	37	50
6	6	38.500000	27	58
7	7	32.866667	27	46
8	8	36.500000	28	40
9	9	36.750000	28	41
10	10	35.960000	28	49
11	11	35.115385	28	50
12	12	36.125000	28	58
13	13	36.490909	28	50
14	14	41.052632	28	50
15	15	39.500000	36	43
16	16	37.333333	32	40
17	17	40.000000	40	40

	Reason_for_absence	mean	min	max
18	18	37.619048	28	58
19	19	35.450000	28	50
20	21	37.166667	30	40
21	22	32.657895	28	41
22	23	35.610738	27	50
23	24	33.000000	33	33
24	25	36.161290	28	58
25	26	37.848485	28	50
26	27	35.507246	28	38
27	28	37.508929	28	50

In [34]:

```
import pandas as pd
data1 = {
    'Empid': [1011, 1012, 1013, 1014, 1015],
    'Name': ['John', 'Rahul', 'Rick', 'Morty', 'Tim'],
    'Designation': ['Manager', 'Research Engineer', 'Research Engineer', 'VP', 'De
    'Date_of_joining': ['01-Jan-2000', '23-sep-2006', '11-Jan-2012', '21-Jan-1991',
Emp_df = pd.DataFrame(data1, columns = ['Empid', 'Name', 'Designation', 'Date_of_joining
Emp_df
```

Out[34]:

	Empid	Name	Designation	Date_of_joining
0	1011	John	Manager	01-Jan-2000
1	1012	Rahul	Research Engineer	23-sep-2006
2	1013	Rick	Research Engineer	11-Jan-2012
3	1014	Morty	VP	21-Jan-1991
4	1015	Tim	Delivery Manager	12-Jan-1990

In [35]:

```
data2 = {
    'Empid': [1011, 1017, 1013, 1019, 1015],
    'Department': ['Management', 'Research', 'Research', 'Management', 'Delivery']
    'Total_Experience': [18, 10, 10, 28, 22]}
Dept_df = pd.DataFrame(data2, columns = ['Empid', 'Department', 'Total_Experience'])
Dept_df
```

Out[35]:

	Empid	Department	Total_Experience
0	1011	Management	18
1	1017	Research	10
2	1013	Research	10
3	1019	Management	28

	Empid	Department	Total_Experience
4	1015	Delivery	22

In [36]:

```
# Inner Join
pd.merge(Emp_df, Dept_df, left_on='Empid',right_on='Empid', how='inner')
```

Out[36]:

	Empid	Name	Designation	Date_of_joining	Department	Total_Experience
0	1011	John	Manager	01-Jan-2000	Management	18
1	1013	Rick	Research Engineer	11-Jan-2012	Research	10
2	1015	Tim	Delivery Manager	12-Jan-1990	Delivery	22

In [37]:

```
# Inner Join
Query_string = """ select * from Emp_df a INNER JOIN Dept_df b ON a.Empid = b.Empid;"""
sqldf(Query_string, globals())
```

Out[37]:

	Empid	Name	Designation	Date_of_joining	Empid	Department	Total_Experience
0	1011	John	Manager	01-Jan-2000	1011	Management	18
1	1013	Rick	Research Engineer	11-Jan-2012	1013	Research	10
2	1015	Tim	Delivery Manager	12-Jan-1990	1015	Delivery	22

In [38]:

```
# Left Join
pd.merge(Emp_df, Dept_df, left_on='Empid',right_on='Empid', how='left')
```

Out[38]:

	Empid	Name	Designation	Date_of_joining	Department	Total_Experience
0	1011	John	Manager	01-Jan-2000	Management	18.0
1	1012	Rahul	Research Engineer	23-sep-2006	NaN	NaN
2	1013	Rick	Research Engineer	11-Jan-2012	Research	10.0
3	1014	Morty	VP	21-Jan-1991	NaN	NaN
4	1015	Tim	Delivery Manager	12-Jan-1990	Delivery	22.0

In [39]:

```
# Left Join
Query_string = """ select * from Emp_df a LEFT JOIN Dept_df b ON a.Empid = b.Empid;"""
sqldf(Query_string, globals())
```

Out[39]:

	Empid	Name	Designation	Date_of_joining	Empid	Department	Total_Experience
0	1011	John	Manager	01-Jan-2000	1011.0	Management	18.0
1	1012	Rahul	Research Engineer	23-sep-2006	NaN	None	NaN
2	1013	Rick	Research Engineer	11-Jan-2012	1013.0	Research	10.0
3	1014	Morty	VP	21-Jan-1991	NaN	None	NaN

	Empid	Name	Designation	Date_of_joining	Empid	Department	Total_Experience
4	1015	Tim	Delivery Manager	12-Jan-1990	1015.0	Delivery	22.0

In [40]:

```
# Right Join
pd.merge(Emp_df, Dept_df, left_on='Empid', right_on='Empid', how='right')
```

Out[40]:

	Empid	Name	Designation	Date_of_joining	Department	Total_Experience
0	1011	John	Manager	01-Jan-2000	Management	18
1	1017	NaN	NaN	NaN	Research	10
2	1013	Rick	Research Engineer	11-Jan-2012	Research	10
3	1019	NaN	NaN	NaN	Management	28
4	1015	Tim	Delivery Manager	12-Jan-1990	Delivery	22

In [41]:

```
# Right Join
Query_string = """ select a.Empid,Name,Designation,Date_of_joining,Department,Total_Ex
sqldf(Query_string, globals())
```

Out[41]:

	Empid	Name	Designation	Date_of_joining	Department	Total_Experience
0	1011	John	Manager	01-Jan-2000	Management	18
1	1017	None	None	None	Research	10
2	1013	Rick	Research Engineer	11-Jan-2012	Research	10
3	1019	None	None	None	Management	28
4	1015	Tim	Delivery Manager	12-Jan-1990	Delivery	22

In [42]:

```
# Outer Join
pd.merge(Emp_df, Dept_df, left_on='Empid', right_on='Empid', how='outer')
```

Out[42]:

	Empid	Name	Designation	Date_of_joining	Department	Total_Experience
0	1011	John	Manager	01-Jan-2000	Management	18.0
1	1012	Rahul	Research Engineer	23-sep-2006	NaN	NaN
2	1013	Rick	Research Engineer	11-Jan-2012	Research	10.0
3	1014	Morty	VP	21-Jan-1991	NaN	NaN
4	1015	Tim	Delivery Manager	12-Jan-1990	Delivery	22.0
5	1017	NaN	NaN	NaN	Research	10.0
6	1019	NaN	NaN	NaN	Management	28.0

In [47]:

```
# OUTER join
Query_string = """ select * from Emp_df a left OUTER JOIN Dept_df b ON a.Empid = b.Emp
```

```
sqlidf(Query_string, globals())
```

Out[47]:

	Empid	Name	Designation	Date_of_joining	Empid	Department	Total_Experience
0	1011	John	Manager	01-Jan-2000	1011.0	Management	18.0
1	1012	Rahul	Research Engineer	23-sep-2006	NaN	None	NaN
2	1013	Rick	Research Engineer	11-Jan-2012	1013.0	Research	10.0
3	1014	Morty	VP	21-Jan-1991	NaN	None	NaN
4	1015	Tim	Delivery Manager	12-Jan-1990	1015.0	Delivery	22.0

In [48]: `import sqlite3`

In [49]:

```
con = sqlite3.connect('school.sqlite')
cur = con.cursor()

q = 'select * from faculty'
cur.execute(q)

for row in cur.fetchall():
    print(row)
```

```
('98005', 'Suzanne', 'Viescas', '15127 NE 24th, #383', 'Redmond', 'WA', '98052', '425',
'555-2686', '44000', '1986-05-31 00:00:00', 'Instructor', 'Full time', 'True')
('98007', 'Gary', 'Hallmark', 'Route 2, Box 203B', 'Auburn', 'WA', '98002', '253', '555-
2676', '53000', '1985-01-21 00:00:00', 'Associate Professor', 'Full time', 'True')
('98010', 'Jeffrey', 'Smith', '30301 - 166th Ave. N.E.', 'Fremont', 'CA', '94538', '51
0', '555-2596', '52000', '1983-10-06 00:00:00', 'Associate Professor', 'On leave', 'Tru
e')
('98011', 'Ann', 'Patterson', '16 Maple Lane', 'Auburn', 'WA', '98002', '253', '555-259
1', '45000', '1983-10-16 00:00:00', 'Instructor', 'Full time', 'True')
('98012', 'Michael', 'Davolio', '672 Lamont Ave', 'Houston', 'TX', '77201', '713', '555-
2491', '49000', '1989-02-09 00:00:00', 'Instructor', 'Full time', 'True')
('98013', 'Ann', 'Fuller', '908 W. Capital Way', 'Tacoma', 'WA', '98413', '253', '555-24
96', '44000', '1986-07-05 00:00:00', 'Instructor', 'Full time', 'True')
('98014', 'James', 'Leverling', '722 Moss Bay Blvd.', 'Kirkland', 'WA', '98033', '425',
'555-2501', '60000', '1986-07-16 00:00:00', 'Professor', 'Full time', 'True')
('98019', 'Laura', 'Callahan', '901 Pine Avenue', 'Portland', 'OR', '97208', '503', '555
-2526', '45000', '1989-11-02 00:00:00', 'Instructor', 'Full time', 'True')
('98020', 'Albert', 'Buchanan', '13920 S.E. 40th Street', 'Bellevue', 'WA', '98009', '42
5', '555-2531', '45000', '1985-08-02 00:00:00', 'Instructor', 'Full time', 'True')
('98025', 'Janet', 'Leverling', '722 Moss Bay Blvd.', 'Kirkland', 'WA', '98033', '425',
'555-2576', '50000', '1984-04-12 00:00:00', 'Associate Professor', 'Full time', 'True')
('98028', 'Alaina', 'Hallmark', 'Route 2, Box 203B', 'Woodinville', 'WA', '98072', '42
5', '555-2631', '57000', '1984-01-17 00:00:00', 'Professor', 'Full time', 'True')
('98030', 'Amelia', 'Buchanan', '13920 S.E. 40th Street', 'Bellevue', 'WA', '98006', '42
5', '555-2556', '48000', '1988-05-31 00:00:00', 'Instructor', 'Full time', 'True')
('98036', 'John', 'Leverling', '611 Alpine Drive', 'Palm Springs', 'CA', '92263', '760',
'555-2611', '60000', '1982-11-20 00:00:00', 'Professor', 'Full time', 'True')
('98040', 'David', 'Callahan', '101 NE 88th', 'Salem', 'OR', '97301', '503', '555-2636',
'50000', '1987-01-13 00:00:00', 'Associate Professor', 'Full time', 'True')
('98042', 'David', 'Smith', '311 20th Ave. N.E.', 'Fremont', 'CA', '94538', '510', '555-
2646', '52000', '1991-12-17 00:00:00', 'Associate Professor', 'Full time', 'True')
('98045', 'Michael', 'Hernandez', 'PO Box 223311', 'Tacoma', 'WA', '98413', '253', '555-
2711', '60000', '1990-08-20 00:00:00', 'Professor', 'Full time', 'True')
('98048', 'Joyce', 'Bonnicksen', '2424 Thames Drive', 'Bellevue', 'WA', '98006', '425',
'555-2726', '60000', '1986-03-02 00:00:00', 'Professor', 'Full time', 'True')
```

```
('98052', 'Katherine', 'Ehrlich', '777 Fenexet Blvd', 'Redmond', 'WA', '98052', '425',
'555-0399', '45000', '1985-03-08 00:00:00', 'Instructor', 'Part time', 'False')
('98053', 'Gregory', 'Piercy', '4501 Wetland Road', 'Long Beach', 'CA', '90809', '562',
'555-0037', '45000', '1992-02-10 00:00:00', 'Instructor', 'Full time', 'True')
('98055', 'Alastair', 'Black', '3887 Easy Street', 'Seattle', 'WA', '98125', '206', '555-
-0039', '60000', '1988-12-11 00:00:00', 'Professor', 'Full time', 'True')
('98059', 'Consuelo', 'Maynez', '3445 Cheyenne Road', 'El Paso', 'TX', '79993', '915',
'555-2291', '48000', '1986-09-17 00:00:00', 'Instructor', 'Full time', 'True')
('98062', 'Caroline', 'Coie', '298 Forest Lane', 'Seattle', 'WA', '98125', '206', '555-2
306', '52000', '1983-01-28 00:00:00', 'Associate Professor', 'Full time', 'False')
('98063', 'Ryan', 'Ehrlich', '455 West Palm Ave', 'San Antonio', 'TX', '78284', '210',
'555-2311', '45000', '1988-03-02 00:00:00', 'Instructor', 'Full time', 'True')
('98064', 'Allan', 'Davis', '877 145th Ave SE', 'Portland', 'OR', '97208', '503', '555-2
316', '56000', '1989-08-20 00:00:00', 'Professor', 'Full time', 'True')
```

```
In [50]: import pandas as pd
```

```
In [51]: con = sqlite3.connect('school.sqlite')
def x(q):
    return pd.read_sql_query(q, con)
```

```
In [52]: x('select * from faculty Limit 6')
```

```
Out[52]:
```

	FacultyID	FacFirstName	FacLastname	FacStreetAddress	FacCity	FacState	FacZipCode	FacAreaCoc
0	98005	Suzanne	Viescas	15127 NE 24th, #383	Redmond	WA	98052	42
1	98007	Gary	Hallmark	Route 2, Box 203B	Auburn	WA	98002	25
2	98010	Jeffrey	Smith	30301 - 166th Ave. N.E.	Fremont	CA	94538	57
3	98011	Ann	Patterson	16 Maple Lane	Auburn	WA	98002	25
4	98012	Michael	Davolio	672 Lamont Ave	Houston	TX	77201	77
5	98013	Ann	Fuller	908 W. Capital Way	Tacoma	WA	98413	25



```
In [53]: q = 'select count(title) from faculty'
x(q)
```

```
Out[53]:
```

	count(title)
0	24

```
In [54]: q = 'select count(distinct title) from faculty'
x(q)
```

```
Out[54]: count(distinct title)
0          3
```

```
In [55]: q = 'select count(distinct title) as ranks from faculty'
x(q)
```

```
Out[55]: ranks
0      3
```

```
In [56]: x('select avg(salary) from faculty')
```

```
Out[56]: avg(salary)
0 51041.666667
```

```
In [57]: x('select * from student_schedules')
```

```
Out[57]:
```

	StudentID	SectionNumber	Grade
0	1001	1560	93.28
1	1001	2071	66.41
2	1001	4055	62.85
3	1001	4760	73.21
4	1001	4812	66.91
...
84	1017	4768	85.26
85	1018	1031	87.05
86	1018	1180	69.88
87	1018	1502	67.13
88	1018	4768	81.23

89 rows × 3 columns

```
In [58]: x('select * from class_sections')
```

```
Out[58]:
```

SectionNumber	CourseID	RoomNumber	Credits	StartTime	Duration	MondaySchedule	TuesdayS
----------------------	-----------------	-------------------	----------------	------------------	-----------------	-----------------------	-----------------

	SectionNumber	CourseID	RoomNumber	Credits	StartTime	Duration	MondaySchedule	TuesdayS
0	1000	11	1231	5	1899-12-30 10:00:00	50	True	
1	1002	12	1619	4	1899-12-30 15:30:00	110	True	
2	1004	13	1627	4	1899-12-30 08:00:00	50	True	
3	1006	13	1627	4	1899-12-30 09:00:00	110	True	
4	1012	14	1627	4	1899-12-30 13:00:00	170	False	
...
71	4810	23	3319	5	1899-12-30 08:00:00	50	False	
72	4812	24	3330	3	1899-12-30 12:00:00	80	False	
73	4813	24	3406	3	1899-12-30 10:00:00	50	True	
74	4825	25	3319	5	1899-12-30 11:00:00	50	True	
75	4880	26	1231	5	1899-12-30 13:30:00	110	True	

76 rows × 12 columns



In [59]:

```
x('select * from courses')
```

Out[59]:

	CourseID	CategoryID	CourseCode	CourseName	CourseDescription
0	1	ACC	ACC 210	Financial Accounting Fundamentals I	Introduces basic accounting concepts, principl...
1	2	ACC	ACC 220	Financial Accounting Fundamentals II	Applications of basic accounting concepts, pri...
2	3	ACC	ACC 230	Fundamentals of Managerial Accounting	Analysis of accounting data as part of the man...

CourseID	CategoryID	CourseCode	CourseName	CourseDescription
3	4	ACC ACC 251	Intermediate Accounting	In-depth review of financial accounting princi...
4	5	ACC ACC 257	Business Tax Accounting	Basic principles, practices and governmental r...
5	6	BUS BUS 101	Introduction to Business	Survey of businss practices. Covers business t...
6	7	BUS BUS 155	Developing A Feasibility Plan	With the aid of a counselor, a feasibility pla...
7	8	BUS BUS 151	Introduction to Entrepreneurship	Overview of the entrepreneurial process, exami...
8	9	BUS BUS 170	Information Technology I	Uses Word for Windows word processing skills, ...
9	10	BUS BUS 171	Information Technology II	Uses intermediate Word features including form...
10	11	ART ART 100	Introduction to Art	Historical backgrounds and design fundamentals...
11	12	ART ART 101	Design	Studio sudies in the fundamentals of two-dimen...
12	13	ART ART 111	Drawing	Study of line, value, space, perspective, and ...
13	14	ART ART 201	Painting	Beginning painting in oil or synthetic media u...
14	15	ART ART 210	Computer Art	Explore the elements of art such as line, valu...
15	16	ART ART 251	Art History	Surveys major forms of visual expression from ...
16	17	BIO BIO 100	Biological Principles	An introductory biology course with lab for th...
17	18	BIO BIO 101	General Biology	Basic biological concepts with emphasis on gene...
18	19	BIO BIO 280	Microbiology	Introduction to micro-organisms including micr...
19	20	CHE CHE 101	Chemistry	General chemistry for non-science majors. Comp...
20	21	CHE CHE 139	Fundamentals of Chemistry	Prepatory for the science major chemistry cour...
21	22	CHE CHE 231	Organic Chemistry	Structure, nomenclature, reactions, and synthe...
22	23	CIS CIS 101	Microcomputer Applications	This is a "hands-on" course. Students will lea...
23	24	CIS CIS 102	Information Systems Concepts	Provides a broad introduction to computers and...

CourseID	CategoryID	CourseCode	CourseName	CourseDescription	
24	25	CIS	CIS 114	Problem Solving and Structured Programming	Covers design, documentation, and coding of pr...
25	26	CIS	CIS 236	Database Management	Includes database concepts, data management te...
26	27	CSC	CSC 110	Programming in BASIC	Computer programming and program design using ...
27	28	CSC	CIS 142	Computer Programming	Introduction to computer science using the C p...
28	29	JRN	JRN 104	College Publications	Hands-on course in college publishing. Covers ...
29	30	ECO	ECO 100	Survey of Economics	Economics applied to various comtemporary soci...
30	31	ECO	ECO 200	Principles of Economics: Microeconomics	Covers resource allocation and income distribu...
31	32	ECO	ECO 201	Principles of Economics: Macroeconomics	Analysis of the aggregate economy: GDP, inflat...
32	33	MUS	MUS 100	Music in the Western World	An introduction to music. Features music from ...
33	34	MUS	MUS 101	First Year Theory and Ear Training	Rudiments of music - notation, scales, interva...
34	35	MUS	MUS 201	Second Year Music Theory	Continuation of MUS 101. Chromatic harmony, mo...
35	36	MUS	MUS 204	History of Jazz	Traces the roots of jazz in America from New O...
36	37	ENG	ENG 101	Composition - Fundamentals	Introduces the nature of the writing process i...
37	38	ENG	ENG 102	Composition - Intermediate	Continues instruction on the writing process, ...
38	39	ENG	ENG 104	Advanced English Grammar	Study of the grammar and rhetoric of the Engli...
39	40	GEG	GEG 100	Introduction to Geography	An introduction to the major cultures of the w...
40	41	GEG	GEG 205	Physical Geography	Study of the Earth, the materials that make it...
41	42	HIS	HIS 101	World History to 1500	Historic foundations and development of the gr...
42	43	HIS	HIS 111	U.S. History to 1877	American history from the colonial period thro...
43	44	HIS	HIS 112	U.S. History Since 1865	Includes Reconstruction, industrialization, ur...
44	45	MAT	MAT 080	Preparatory Mathematics	Individualized instruction in Arithmetic, Alge...

CourseID	CategoryID	CourseCode	CourseName	CourseDescription
45	46	MAT MAT 097	Elementary Algebra	First course in Algebra includes signed number...
46	47	MAT MAT 098	Intermediate Algebra	Sets and the real number system, polynomial an...
47	48	MAT MAT 103	Geometry and Visualization	Basic plane geometry concepts, emphasizing pro...
48	49	MAT MAT 104	Trigonometry	Elementary plane goemetry, right triangle tirtg...
49	50	PHY PHY 100	Survey Of Physics	Basic laws of physics from the laws of motion t...
50	51	PHY PHY 101	General Physics	Classical mechanics; kinematics and dynamics. ...
51	52	PHY PHY 201	Engineering Physics I	Development of the basic principles of classic...
52	53	PHY PHY 203	Engineering Physics II	Waves and oscillations. Mechanical waves and s...
53	54	POL POL 101	Introduction to Political Science	Introduction to theory, organization, politics...
54	55	POL POL 102	American Government	Origin and development of the U.S. government...
55	56	POL POL 213	Women and Politics	Introduction to concepts of power and policy i...

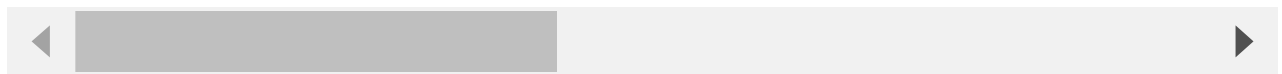
In [60]:

```
#available tables are buildings, class_rooms, class_sections, course_categories, course
#faculty, faculty_categories, faculty_courses, faculty_sections, student_schedules, stu
q = 'select * from class_sections natural join courses where categoryid = "MAT"'
x(q)
```

Out[60]:

SectionNumber	CourseID	RoomNumber	Credits	StartTime	Duration	MondaySchedule	TuesdayS
0	2889	45 2423	5	1899-12-30 09:00:00	50	True	
1	2891	45 2423	5	1899-12-30 11:00:00	50	True	
2	2895	45 2423	5	1899-12-30 13:00:00	50	True	
3	2907	46 3445	5	1899-12-30 08:00:00	50	True	
4	2911	46 3445	5	1899-12-30 12:00:00	50	True	

	SectionNumber	CourseID	RoomNumber	Credits	StartTime	Duration	MondaySchedule	TuesdayS
5	2915	46	3353	5	1899-12-30 10:00:00	50	True	
6	2917	47	3422	5	1899-12-30 14:00:00	50	True	
7	2925	47	3422	5	1899-12-30 15:00:00	50	True	
8	2933	47	3422	5	1899-12-30 09:00:00	240	False	
9	2633	48	3420	5	1899-12-30 16:00:00	50	False	
10	2639	49	3353	3	1899-12-30 09:00:00	50	True	
11	2647	49	3353	3	1899-12-30 18:00:00	50	False	



In [61]:

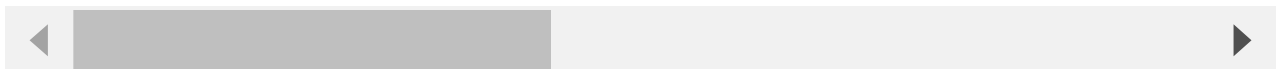
```
q = 'select * from class_sections natural join courses'
x(q)
```

Out[61]:

	SectionNumber	CourseID	RoomNumber	Credits	StartTime	Duration	MondaySchedule	TuesdayS
0	1000	11	1231	5	1899-12-30 10:00:00	50	True	
1	1002	12	1619	4	1899-12-30 15:30:00	110	True	
2	1004	13	1627	4	1899-12-30 08:00:00	50	True	
3	1006	13	1627	4	1899-12-30 09:00:00	110	True	
4	1012	14	1627	4	1899-12-30 13:00:00	170	False	
...

	SectionNumber	CourseID	RoomNumber	Credits	StartTime	Duration	MondaySchedule	TuesdayS
71	4810	23	3319	5	1899-12-30 08:00:00	50	False	
72	4812	24	3330	3	1899-12-30 12:00:00	80	False	
73	4813	24	3406	3	1899-12-30 10:00:00	50	True	
74	4825	25	3319	5	1899-12-30 11:00:00	50	True	
75	4880	26	1231	5	1899-12-30 13:30:00	110	True	

76 rows × 16 columns



In [62]: `q = 'select avg(credits), avg(duration) from class_sections natural join courses where x(q)`

Out[62]:

	avg(credits)	avg(duration)
0	4.666667	65.833333

In [63]: `q = 'select avg(credits), avg(duration) from class_sections natural join courses where x(q)`

Out[63]:

	avg(credits)	avg(duration)
0	4.375	87.5

In [66]: `q='select categoryid, avg(credits) as mean_credits from class_sections natural join cou
df=x(q)
df`

Out[66]:

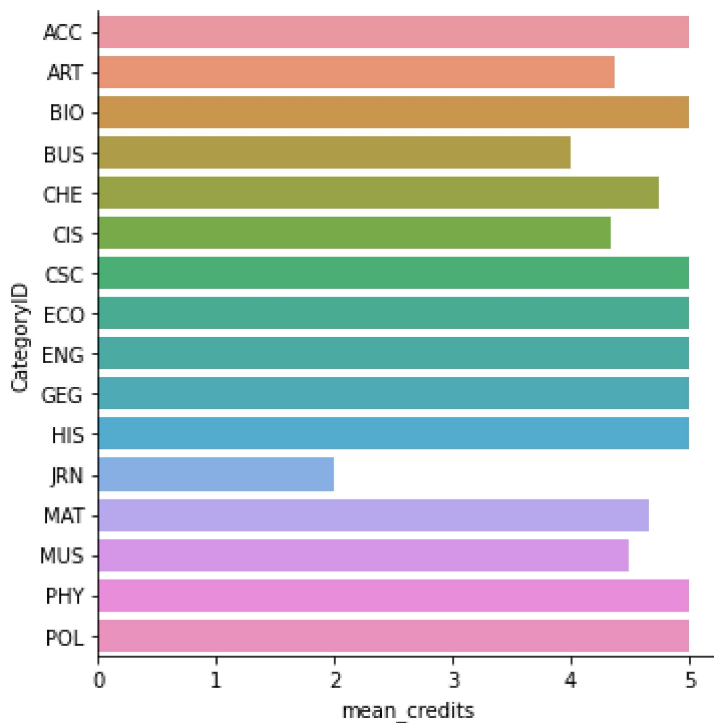
	CategoryID	mean_credits
0	ACC	5.000000
1	ART	4.375000
2	BIO	5.000000
3	BUS	4.000000
4	CHE	4.750000

	CategoryID	mean_credits
5	CIS	4.333333
6	CSC	5.000000
7	ECO	5.000000
8	ENG	5.000000
9	GEG	5.000000
10	HIS	5.000000
11	JRN	2.000000
12	MAT	4.666667
13	MUS	4.500000
14	PHY	5.000000
15	POL	5.000000

```
In [64]: import seaborn as sns
```

```
In [67]: sns.catplot(y='CategoryID', x='mean_credits', data=df, kind='bar')
```

```
Out[67]: <seaborn.axisgrid.FacetGrid at 0x2125d7f4070>
```



```
In [68]: q = 'select * from faculty_courses'  
df = x(q)  
df
```

```
Out[68]:
```

	FacultyID	CourseID	ProficiencyRating
0	98005	12	10
1	98005	16	10
2	98005	34	9
3	98005	36	8
4	98005	38	8
...
106	98063	53	8
107	98064	29	9
108	98064	37	8
109	98064	40	9
110	98064	41	8

111 rows × 3 columns

In []:

```
#save graph as an image file  
#import matplotlib.pyplot as plt  
plt.savefig('filename.png')  
#changing the file extension will change the format of the file save
```

In []:

```
#to install sqlite3 in Anaconda  
pip install sqlite3  
conda install sqlite3
```