

```
In [1]: import pandas as pd
```

```
In [3]: customer = {'Name' : ["Terry", "Kyung", "Daniel", "Joshua"],  
                    'Location' : ["Seoul", "London", "Sydney", "Paris"],  
                    'Age' : [47, 41, 19, 15]}
```

```
In [7]: a = pd.DataFrame(customer)
```

```
In [17]: a[a.Name == "Daniel"]
```

```
Out[17]:
```

	Name	Location	Age
2	Daniel	Sydney	19

```
In [15]: a.Name == "Daniel"
```

```
Out[15]: 0    Terry  
         1    Kyung  
         2    Daniel  
         3    Joshua  
         Name: Name, dtype: object
```

```
In [23]: dates = pd.date_range('2025-04-01', periods=10)
```

```
In [25]: dates
```

```
Out[25]: DatetimeIndex(['2025-04-01', '2025-04-02', '2025-04-03', '2025-04-04',  
                        '2025-04-05', '2025-04-06', '2025-04-07', '2025-04-08',  
                        '2025-04-09', '2025-04-10'],  
                        dtype='datetime64[ns]', freq='D')
```

```
In [27]: tmp1 = pd.Series([80, 92, 100, 78, 88, 89, 94, 95, 96, 79], index=dates)  
tmp1
```

```
Out[27]: 2025-04-01    80  
         2025-04-02    92  
         2025-04-03   100  
         2025-04-04    78  
         2025-04-05    88  
         2025-04-06    89  
         2025-04-07    94  
         2025-04-08    95  
         2025-04-09    96  
         2025-04-10    79  
         Freq: D, dtype: int64
```

```
In [29]: import numpy as np
```

```
In [31]: tmp2 = pd.Series(np.random.randint(60, 100, size=10), index=dates)  
tmp2
```

```
Out[31]: 2025-04-01    73
          2025-04-02    83
          2025-04-03    73
          2025-04-04    97
          2025-04-05    96
          2025-04-06    90
          2025-04-07    86
          2025-04-08    90
          2025-04-09    83
          2025-04-10    91
          Freq: D, dtype: int64
```

```
In [33]: exam = pd.DataFrame({'Math': tmp1,
                              'English': tmp2})
          exam
```

```
Out[33]:
```

	Math	English
2025-04-01	80	73
2025-04-02	92	83
2025-04-03	100	73
2025-04-04	78	97
2025-04-05	88	96
2025-04-06	89	90
2025-04-07	94	86
2025-04-08	95	90
2025-04-09	96	83
2025-04-10	79	91

```
In [35]: exam['Math']
```

```
Out[35]: 2025-04-01    80
          2025-04-02    92
          2025-04-03   100
          2025-04-04    78
          2025-04-05    88
          2025-04-06    89
          2025-04-07    94
          2025-04-08    95
          2025-04-09    96
          2025-04-10    79
          Freq: D, Name: Math, dtype: int64
```

```
In [37]: exam.Math
```

```
Out[37]: 2025-04-01      80
         2025-04-02      92
         2025-04-03     100
         2025-04-04      78
         2025-04-05      88
         2025-04-06      89
         2025-04-07      94
         2025-04-08      95
         2025-04-09      96
         2025-04-10      79
         Freq: D, Name: Math, dtype: int64
```

```
In [45]: exam.loc['2025-04-03', 'Math']
```

```
Out[45]: 100
```

```
In [49]: exam.iloc[3,0]
```

```
Out[49]: 78
```

```
In [57]: np.mean(exam, axis=1)
```

```
Out[57]: 2025-04-01      76.5
         2025-04-02      87.5
         2025-04-03      86.5
         2025-04-04      87.5
         2025-04-05      92.0
         2025-04-06      89.5
         2025-04-07      90.0
         2025-04-08      92.5
         2025-04-09      89.5
         2025-04-10      85.0
         Freq: D, dtype: float64
```

```
In [59]: exam['Average'] = np.mean(exam, axis=1)
```

```
In [63]: exam.iloc[-1]
```

```
Out[63]: Math      79.0
         English    91.0
         Average    85.0
         Name: 2025-04-10 00:00:00, dtype: float64
```

```
In [69]: exam.columns
```

```
Out[69]: Index(['Math', 'English', 'Average'], dtype='object')
```

```
In [67]: exam.index
```

```
Out[67]: DatetimeIndex(['2025-04-01', '2025-04-02', '2025-04-03', '2025-04-04',
                        '2025-04-05', '2025-04-06', '2025-04-07', '2025-04-08',
                        '2025-04-09', '2025-04-10'],
                        dtype='datetime64[ns]', freq='D')
```

```
In [75]: exam[(exam.Average >= 80)&(exam.index >= '2025-04-05')]
```

Out[75]:

	Math	English	Average
2025-04-05	88	96	92.0
2025-04-06	89	90	89.5
2025-04-07	94	86	90.0
2025-04-08	95	90	92.5
2025-04-09	96	83	89.5
2025-04-10	79	91	85.0

```
In [77]: exam[(exam.Math >= 90)|(exam.English >= 95)]
```

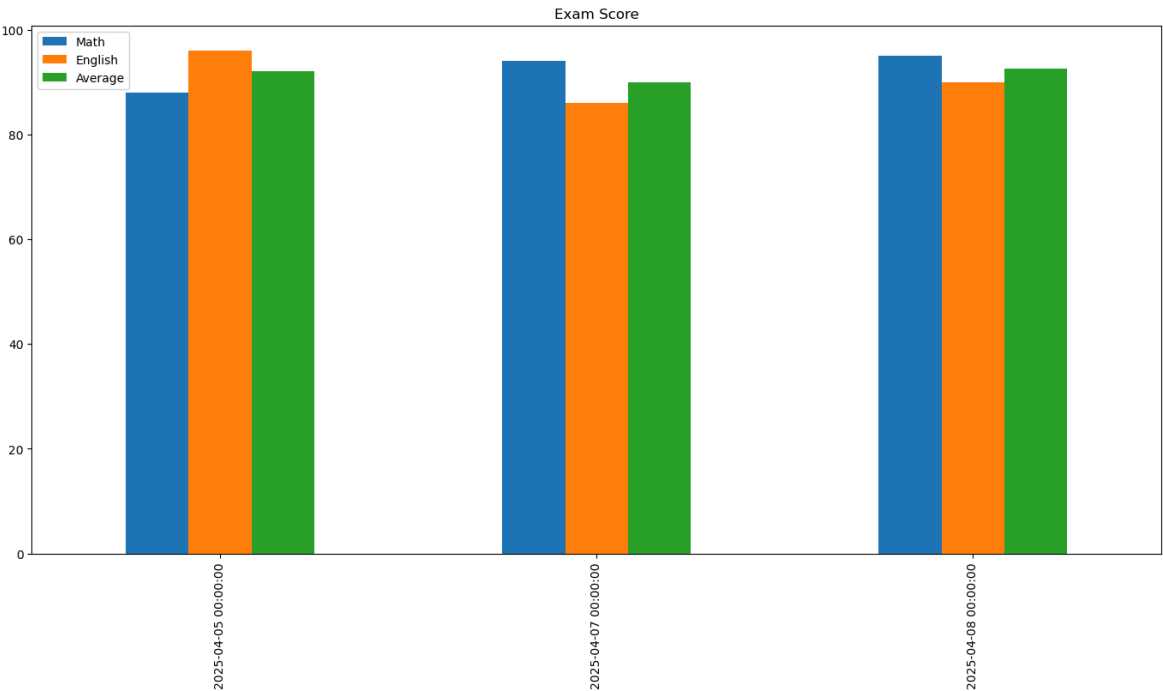
Out[77]:

	Math	English	Average
2025-04-02	92	83	87.5
2025-04-03	100	73	86.5
2025-04-04	78	97	87.5
2025-04-05	88	96	92.0
2025-04-07	94	86	90.0
2025-04-08	95	90	92.5
2025-04-09	96	83	89.5

```
In [79]: import matplotlib.pyplot as plt
```

```
In [87]: exam[exam.Average >= 90].plot(kind='bar', figsize=(17, 8), title="Exam Score")
```

Out[87]: <Axes: title={'center': 'Exam Score'}>



```
In [97]: movies = pd.read_csv('http://bit.ly/imdbratings')
print(movies.shape)
movies.tail()
```

(979, 6)

```
Out[97]:
```

	star_rating	title	content_rating	genre	duration	actors_list
974	7.4	Tootsie	PG	Comedy	116	[u'Dustin Hoffman', u'Jessica Lange', u'Teri G...
975	7.4	Back to the Future Part III	PG	Adventure	118	[u'Michael J. Fox', u'Christopher Lloyd', u'Ma...
976	7.4	Master and Commander: The Far Side of the World	PG-13	Action	138	[u'Russell Crowe', u'Paul Bettany', u'Billy Bo...
977	7.4	Poltergeist	PG	Horror	114	[u'JoBeth Williams', u'Heather O'Rourke', u'Cr...
978	7.4	Wall Street	R	Crime	126	[u'Charlie Sheen', u'Michael Douglas', u'Tamar...

```
In [101... movies.columns
```

```
Out[101... Index(['star_rating', 'title', 'content_rating', 'genre', 'duration',
      'actors_list'],
      dtype='object')
```

```
In [105... movies[(movies.duration >=200)&(movies.star_rating >= 9.0)]
```

```
Out[105...
```

	star_rating	title	content_rating	genre	duration	actors_list
2	9.1	The Godfather: Part II	R	Crime	200	[u'Al Pacino', u'Robert De Niro', u'Robert Duv...

```
In [109... movies[movies.duration >=200].star_rating.mean()
```

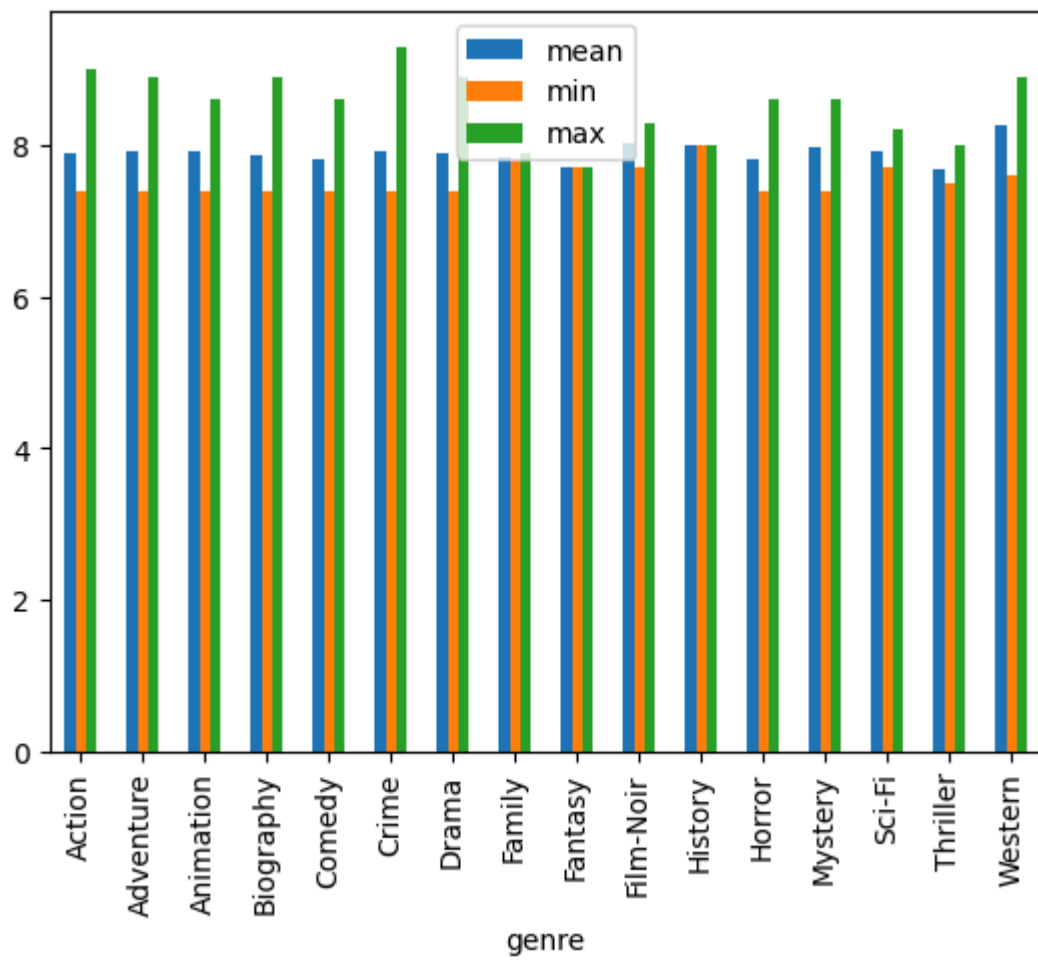
```
Out[109... 8.258333333333333
```

```
In [111... movies.loc[movies.duration >=200, 'star_rating'].mean()
```

```
Out[111... 8.258333333333333
```

```
In [119... movies.groupby('genre').star_rating.agg(['mean', 'min', 'max']).plot(kind='bar')
```

```
Out[119... <Axes: xlabel='genre'>
```



```
In [ ]:
```