```
In [1]:  import test
```

__name__ value is: test

```
In [2]:  import pandas as pd
```

```
In [3]:  df = pd.read_csv('Homo_sapiens.gene_info', sep='\t')
```

```
In [5]:  df[:3]
```

Out[5]:

	#tax_id	GeneID	Symbol	LocusTag	Synonyms	
0	9606	1	A1BG	-	A1B|ABG|GAB|HYST2477	MIM:138670|HGNC:HG
1	9606	2	A2M	-	A2MD|CPAMD5|FWP007|S863-7	MIM:103950|HG
2	9606	3	A2MP1	-	A2MP	HGNC:HG

```
In [6]:  df[-3:]
```

Out[6]:

	#tax_id	GeneID	Symbol	LocusTag	Synonyms	dbXrefs	chromosome	map_locatio
61715	741158	8923217	trnA	-	-	-	MT	
61716	741158	8923218	COX1	-	-	-	MT	
61717	741158	8923219	16S rRNA	-	-	-	MT	

```
In [7]:  df.head()
```

Out[7]:

	#tax_id	GeneID	Symbol	LocusTag	Synonyms	
0	9606	1	A1BG	-	A1B|ABG|GAB|HYST2477	MIM:138670|HGNC
1	9606	2	A2M	-	A2MD|CPAMD5|FWP007|S863-7	MIM:103950|HGNC
2	9606	3	A2MP1	-	A2MP	HGNC:
3	9606	9	NAT1	-	AAC1|MNAT|NAT-1|NATI	MIM:108345|HGNC:HG
4	9606	10	NAT2	-	AAC2|NAT-2|PNAT	MIM:612182|HGNC:HG

```
In [8]:  df.tail()
```

Out[8]:

	#tax_id	GeneID	Symbol	LocusTag	Synonyms	dbXrefs	chromosome	map_locatio
61713	741158	8923215	trnD	-	-	-	MT	
61714	741158	8923216	trnP	-	-	-	MT	
61715	741158	8923217	trnA	-	-	-	MT	
61716	741158	8923218	COX1	-	-	-	MT	
61717	741158	8923219	16S rRNA	-	-	-	MT	

```
In [ ]:  !pip install pandas
         !conda -y install pandas
```

```
In [9]:  df.columns
```

```
Out[9]:  Index(['#tax_id', 'GeneID', 'Symbol', 'LocusTag', 'Synonyms', 'dbXrefs',
                'chromosome', 'map_location', 'description', 'type_of_gene',
                'Symbol_from_nomenclature_authority',
                'Full_name_from_nomenclature_authority', 'Nomenclature_status',
                'Other_designations', 'Modification_date', 'Feature_type'],
               dtype='object')
```

```
In [12]:  list(df.columns.values)
```

```
Out[12]:  ['#tax_id',
           'GeneID',
           'Symbol',
           'LocusTag',
           'Synonyms',
           'dbXrefs',
           'chromosome',
           'map_location',
           'description',
           'type_of_gene',
           'Symbol_from_nomenclature_authority',
           'Full_name_from_nomenclature_authority',
           'Nomenclature_status',
           'Other_designations',
           'Modification_date',
           'Feature_type']
```

```
In [14]:  df['Symbol']
```

```
Out[14]:  0             A1BG
          1              A2M
          2            A2MP1
          3             NAT1
          4             NAT2
                      ...
          61713         trnD
          61714         trnP
          61715         trnA
          61716         COX1
          61717     16S rRNA
          Name: Symbol, Length: 61718, dtype: object
```

```
In [15]:  df[['Symbol', 'GeneID']]
```

Out[15]:

	Symbol	GeneID
0	A1BG	1

	Symbol	GeneID
1	A2M	2
2	A2MP1	3
3	NAT1	9
4	NAT2	10
...	...	...
61713	trnD	8923215
61714	trnP	8923216
61715	trnA	8923217
61716	COX1	8923218
61717	16S rRNA	8923219

In [16]:
```python
df['chromosome']
```

Out[16]:
```
0          19
1          12
2          12
3           8
4           8
          ..
61713      MT
61714      MT
61715      MT
61716      MT
61717      MT
Name: chromosome, Length: 61718, dtype: object
```

In [21]:
```python
df['chromosome'] == '10'
```

Out[21]:
```
0          False
1          False
2          False
3          False
4          False
          ...
61713      False
61714      False
61715      False
61716      False
61717      False
Name: chromosome, Length: 61718, dtype: bool
```

In [22]:
```python
df[df['chromosome'] == '10']
```

Out[22]:

	#tax_id	GeneID	Symbol	LocusTag	Synonyms	
30	9606	36	ACADSB	-	2-MEBCAD|ACAD7|SBCAD	MIM:60030
49	9606	59	ACTA2	-	ACTSA	MIM:102620
83	9606	101	ADAM8	-	CD156|CD156a|MS2	MIM:602267
87	9606	105	ADARB2	-	ADAR3|RED2	MIM:602065|

	#tax_id	GeneID	Symbol	LocusTag	Synonyms	
101	9606	120	ADD3	-	ADDL|CPSQ3	MIM:601568|
...	...	...	...	...	...	
61081	9606	116216122	LOC116216122	-	-	
61082	9606	116216123	LOC116216123	-	-	
61472	9606	118568810	SUV39H2-DT	-	-	
61487	9606	118568825	CSGALNACT2-DT	-	-	
61514	9606	118732302	CCNY-AS1	-	-	

In [23]:
```python
df[df['chromosome'] == '10']['Symbol']
```

Out[23]:
```
30              ACADSB
49               ACTA2
83               ADAM8
87              ADARB2
101               ADD3
             ...
61081     LOC116216122
61082     LOC116216123
61472       SUV39H2-DT
61487    CSGALNACT2-DT
61514          CCNY-AS1
Name: Symbol, Length: 2462, dtype: object
```

In [26]:
```python
df['type_of_gene'].unique()
```

Out[26]:
```
array(['protein-coding', 'pseudo', 'other', 'unknown', 'ncRNA', 'tRNA',
       'rRNA', 'scRNA', 'snoRNA', 'snRNA', 'biological-region'],
      dtype=object)
```

In [27]:
```python
df['type_of_gene'].value_counts()
```

Out[27]:
```
protein-coding     19697
ncRNA              17498
pseudo             16558
biological-region   4466
unknown             1385
other                840
tRNA                 595
snoRNA               541
snRNA                 71
rRNA                  63
scRNA                  4
Name: type_of_gene, dtype: int64
```

In [28]:
```python
df['chromosome'] == '10'
```

Out[28]:
```
0        False
```

```
1        False
2        False
3        False
4        False
         ...
61713    False
61714    False
61715    False
61716    False
61717    False
```

In [30]: `df['type_of_gene'] == 'protein-coding'`

Out[30]:
```
0        True
1        True
2        False
3        True
4        True
         ...
61713    False
61714    False
61715    False
61716    True
61717    False
Name: type_of_gene, Length: 61718, dtype: bool
```

In [32]: `df[(df['chromosome'] == '10') & (df['type_of_gene'] == 'protein-coding')]`

Out[32]:

	#tax_id	GeneID	Symbol	LocusTag	Synonyms	
30	9606	36	ACADSB	-	2-MEBCAD|ACAD7|SBCAD	MIM:6003
49	9606	59	ACTA2	-	ACTSA	MIM:10262
83	9606	101	ADAM8	-	CD156|CD156a|MS2	MIM:60226
87	9606	105	ADARB2	-	ADAR3|RED2	MIM:60206
101	9606	120	ADD3	-	ADDL|CPSQ3	MIM:601568
...	...	...	...	...	...	
53349	9606	107983989	LOC107983989	-	-	
53461	9606	107984189	LOC107984189	-	-	
53474	9606	107984203	LOC107984203	-	-	
53527	9606	107984264	LOC107984264	-	-	

	#tax_id	GeneID	Symbol	LocusTag	Synonyms
59468	9606	112577516	LOC112577516	-	-

```
In [33]: df[(df['chromosome'] == '10') & (df['type_of_gene'] == 'protein-coding')][
```

```
Out[33]: 30              ACADSB
         49               ACTA2
         83               ADAM8
         87              ADARB2
         101               ADD3
                    ...
         53349    LOC107983989
         53461    LOC107984189
         53474    LOC107984203
         53527    LOC107984264
         59468    LOC112577516
         Name: Symbol, Length: 722, dtype: object
```

```
In [36]: df[df['chromosome'] == '10']['type_of_gene'].value_counts()
```

```
Out[36]: ncRNA                814
         protein-coding       722
         pseudo               648
         biological-region    202
         unknown               46
         snoRNA                14
         other                 13
         tRNA                   3
         Name: type_of_gene, dtype: int64
```

```
In [38]: df2 = df[(df['chromosome'] == '10') & (df['type_of_gene'] == 'tRNA')]
```

```
In [39]: df2
```

Out[39]:

	#tax_id	GeneID	Symbol	LocusTag	Synonyms	dbXrefs	chromosome
31048	9606	100189083	TRN-GTT2-3	-	TRNAN11	HGNC:HGNC:34643	10
31239	9606	100189279	TRS-TGA1-1	-	TRNAS21	HGNC:HGNC:34845	10
31301	9606	100189342	TRV-TAC3-1	-	TRNAV26	HGNC:HGNC:34910	10

```
In [40]: df[(df['chromosome'] == '10') & (df['type_of_gene'] == 'protein-coding')]
```

Out[40]:

	#tax_id	GeneID	Symbol	LocusTag	Synonyms	
30	9606	36	ACADSB	-	2-MEBCAD|ACAD7|SBCAD	MIM:6003
49	9606	59	ACTA2	-	ACTSA	MIM:10262
83	9606	101	ADAM8	-	CD156|CD156a|MS2	MIM:60226

	#tax_id	GeneID	Symbol	LocusTag	Synonyms
87	9606	105	ADARB2	-	ADAR3|RED2 MIM:602065
101	9606	120	ADD3	-	ADDL|CPSQ3 MIM:601568
...	...	...	...	...	...
53349	9606	107983989	LOC107983989	-	-
53461	9606	107984189	LOC107984189	-	-
53474	9606	107984203	LOC107984203	-	-
53527	9606	107984264	LOC107984264	-	-
59468	9606	112577516	LOC112577516	-	-

```
In [43]: df['description'].str.contains('membrane')
```

```
Out[43]: False    60861
         True       857
         Name: description, dtype: int64
```

```
In [44]: df[(df['chromosome'] == '10') & (df['type_of_gene'] == 'protein-coding') &
```

Out[44]:

	#tax_id	GeneID	Symbol	LocusTag	
9703	9606	25805	BAMBI	-	
9917	9606	26103	LRIT1	-	
10303	9606	27069	GHITM	-	DERP2|HSPC282|MICS1|My
12269	9606	54708	MARCHF5	-	MARCH-V|MA
13327	9606	56889	TM9SF3	-	
15125	9606	80195	TMEM254	-	C

	#tax_id	GeneID	Symbol	LocusTag	
16177	9606	84833	ATP5MK	-	AGP|ATP5MD|DAPIT|HCVFTP2|MC5DN6
18391	9606	143098	MPP7	-	
19403	9606	170371	TMEM273	-	
19585	9606	196740	VSTM4	-	
19863	9606	219623	TMEM26	-	
19958	9606	220972	MARCHF8	-	CMIR|MARCH-VIII|MARCH
21794	9606	340745	LRIT2	-	
22098	9606	347731	LRRTM3	-	
25984	9606	643236	TMEM72	-	
27176	9606	653567	TMEM236	-	FAM23A|FAM23B|
32675	9606	100287932	TIMM23	-	
37136	9606	100652748	TIMM23B	-	
53527	9606	107984264	LOC107984264	-	

```
In [46]: df[(df['chromosome'] == '10') & (df['type_of_gene'] == 'protein-coding') &
```

Out[46]:

	#tax_id	GeneID	Symbol	LocusTag	
9703	9606	25805	BAMBI	-	
9917	9606	26103	LRIT1	-	

	#tax_id	GeneID	Symbol	LocusTag	
10303	9606	27069	GHITM	-	DERP2|HSPC282|MICS1|My
12269	9606	54708	MARCHF5	-	MARCH-V|MA
13327	9606	56889	TM9SF3	-	
15125	9606	80195	TMEM254	-	C
16177	9606	84833	ATP5MK	-	AGP|ATP5MD|DAPIT|HCVFTP2|MC5DN6
18391	9606	143098	MPP7	-	
19403	9606	170371	TMEM273	-	
19585	9606	196740	VSTM4	-	
19863	9606	219623	TMEM26	-	
19958	9606	220972	MARCHF8	-	CMIR|MARCH-VIII|MARCH
21794	9606	340745	LRIT2	-	
22098	9606	347731	LRRTM3	-	
25984	9606	643236	TMEM72	-	
27176	9606	653567	TMEM236	-	FAM23A|FAM23B|
32675	9606	100287932	TIMM23	-	
37136	9606	100652748	TIMM23B	-	

```python
df['chromosome'].value_counts().plot(kind='bar')
```

Out[48]: <AxesSubplot:>

In [50]: `df[(df['chromosome'] == '10') | (df['chromosome'] == '11') ]`

Out[50]:

	#tax_id	GeneID	Symbol	LocusTag	Synonyms	
30	9606	36	ACADSB	-	2-MEBCAD|ACAD7|SBCAD	MIM:600301|HGN
32	9606	38	ACAT1	-	ACAT|MAT|T2|THIL	MIM:607809|HGNC
44	9606	53	ACP2	-	LAP	MIM:171650|HGNC
47	9606	56	ACRV1	-	D11S4365|SP-10|SPACA2	MIM:102525|HGNC
49	9606	59	ACTA2	-	ACTSA	MIM:102620|HGNC
...	...	...	...	...	...	
61504	9606	118597841	VPS11-DT	-	-	
61514	9606	118732302	CCNY-AS1	-	-	
61519	9606	118827805	SF1-DT	-	-	
61624	9606	120356739	LRRC51	-	-	
61625	9606	120356740	TOMT	-	-	

5758 rows × 16 columns

In [51]: `df[ ~(df['chromosome'] == '10') ]`

Out[51]:

	#tax_id	GeneID	Symbol	LocusTag	Synonyms	
0	9606	1	A1BG	-	A1B|ABG|GAB|HYST2477	MIM:138670
1	9606	2	A2M	-	A2MD|CPAMD5|FWP007|S863-7	MIM:103950|
2	9606	3	A2MP1	-	A2MP	�
3	9606	9	NAT1	-	AAC1|MNAT|NAT-1|NATI	MIM:108345|HGI
4	9606	10	NAT2	-	AAC2|NAT-2|PNAT	MIM:612182|HGN
...	...	...	...	...	...	
61713	741158	8923215	trnD	-	-	
61714	741158	8923216	trnP	-	-	
61715	741158	8923217	trnA	-	-	
61716	741158	8923218	COX1	-	-	
61717	741158	8923219	16S rRNA	-	-	

59256 rows × 16 columns

```
In [53]: df[df['chromosome'].isin(['1', '3', '5'])]
```

Out[53]:

	#tax_id	GeneID	Symbol	LocusTag	
7	9606	13	AADAC	-	
18	9606	24	ABCA4	-	ABC10|ABCR|ARMD2|CORD3|FFM|RMP|R
21	9606	27	ABL2	-	
24	9606	30	ACAA1	-	A
28	9606	34	ACADM	-	ACA
...	...	...	...	...	
61613	9606	120017342	CZIB-DT	-	

	#tax_id	GeneID	Symbol	LocusTag
61617	9606	120285837	LOC120285837	-
61621	9606	120285841	LOC120285841	-
61643	9606	120766157	LOC120766157	-
61644	9606	120766158	LOC120766158	-

In [54]:
```python
a = [
    {'col_1': 1, 'col_2': 2, 'col_3': 10},
    {'col_1': 2, 'col_2': 2, 'col_3': 11},
    {'col_1': 3, 'col_2': 1, 'col_3': 12},
    {'col_1': 1, 'col_2': 4, 'col_3': 13},
    {'col_1': 2, 'col_2': 2, 'col_3': 14},
]
```

In [55]:
```python
pd.DataFrame(a)
```

Out[55]:

	col_1	col_2	col_3
0	1	2	10
1	2	2	11
2	3	1	12
3	1	4	13
4	2	2	14

In [56]:
```python
b = {
    'col_1': [1,2,3,2,1],
    'col_2': [2,2,1,4,2],
    'col_3': [10, 11, 12,13,14],
}
```

In [58]:
```python
d  = pd.DataFrame(b)
```

In [59]:
```python
d
```

Out[59]:

	col_1	col_2	col_3
0	1	2	10
1	2	2	11
2	3	1	12
3	2	4	13
4	1	2	14

In [61]:
```python
d.to_excel('test.xlsx')
```

In [63]:
```python
pd.read_excel('test2.xlsx')
```

```
Out[63]:
```

	Unnamed: 0	col_1	col_2	col_3
0	0	1	2	10
1	1	2	2	11
2	2	3	1	12
3	3	2	4	13
4	4	1	2	14
5	10	5	5	6
6	16	6	8	9

```
In [65]: dfs = pd.read_html('https://en.wikipedia.org/wiki/List_of_cities_and_towns_
```

```
In [66]: dfs[0]
```

```
Out[66]:
```

	Rank	City	Census 1991	Census 2001	Census 2011	Region
0	1	Athens 1 *	772072	745514	664046	Attica
1	2	Thessaloniki 2 *	383967	363987	315196	Central Macedonia
2	3	Patras 8 *	152570	160400	167446	Western Greece
3	4	Piraeus 1,3	182671	175697	163688	Attica
4	5	Larissa	112777	124394	144651	Thessaly
...	...	...	...	...	...	...
139	140	Mandra 4	10012	10947	11327	Attica
140	141	Tyrnavos	12028	11116	11069	Thessaly
141	142	Glyka Nera 1	5813	6623	11049	Attica
142	143	Ymittos 1	11671	11139	10715	Attica
143	144	Neo Psychiko 1	12023	10848	10137	Attica

144 rows × 6 columns

```
In [70]: dfs[2]
```

```
Out[70]:
```

	vteList of towns in Europe	vteList of towns in Europe.1
0	Sovereign states	Albania Andorra Armenia Austria Azerbaijan Bel...
1	States with limitedrecognition	Abkhazia Artsakh Kosovo Northern Cyprus South ...
2	Dependencies andother entities	Åland Faroe Islands Gibraltar Guernsey Isle of...

```
In [73]: def f(x):
             return 'gene_' + x['Symbol']


         df['GENE2'] = df.apply(f, axis=1)
```

```
In [74]: df
```

```
Out[74]:
```

	#tax_id	GeneID	Symbol	LocusTag	Synonyms

	#tax_id	GeneID	Symbol	LocusTag	Synonyms	
0	9606	1	A1BG	-	A1B|ABG|GAB|HYST2477	MIM:138670
1	9606	2	A2M	-	A2MD|CPAMD5|FWP007|S863-7	MIM:103950|
2	9606	3	A2MP1	-	A2MP	H
3	9606	9	NAT1	-	AAC1|MNAT|NAT-1|NATI	MIM:108345|HGI
4	9606	10	NAT2	-	AAC2|NAT-2|PNAT	MIM:612182|HGN
...	...	...	...	...	...	
61713	741158	8923215	trnD	-	-	
61714	741158	8923216	trnP	-	-	
61715	741158	8923217	trnA	-	-	
61716	741158	8923218	COX1	-	-	
61717	741158	8923219	16S rRNA	-	-	

```
In [77]:  df = df.drop('GENE2', 1)
```

```
In [78]:  df.columns
```

```
Out[78]: Index(['#tax_id', 'GeneID', 'Symbol', 'LocusTag', 'Synonyms', 'dbXrefs',
              'chromosome', 'map_location', 'description', 'type_of_gene',
              'Symbol_from_nomenclature_authority',
              'Full_name_from_nomenclature_authority', 'Nomenclature_status',
              'Other_designations', 'Modification_date', 'Feature_type'],
             dtype='object')
```

```
In [79]:  d
```

Out[79]:

	col_1	col_2	col_3
0	1	2	10
1	2	2	11
2	3	1	12
3	2	4	13
4	1	2	14

```
In [80]:  d.to_dict('index')
```

```
Out[80]: {0: {'col_1': 1, 'col_2': 2, 'col_3': 10},
          1: {'col_1': 2, 'col_2': 2, 'col_3': 11},
          2: {'col_1': 3, 'col_2': 1, 'col_3': 12},
          3: {'col_1': 2, 'col_2': 4, 'col_3': 13},
```

```
4. {'col 1': 1, 'col 2': 2, 'col 3': 14}]
```

In [81]:
```
d.to_dict('records')
```

Out[81]:
```
[{'col_1': 1, 'col_2': 2, 'col_3': 10},
 {'col_1': 2, 'col_2': 2, 'col_3': 11},
 {'col_1': 3, 'col_2': 1, 'col_3': 12},
 {'col_1': 2, 'col_2': 4, 'col_3': 13},
 {'col_1': 1, 'col_2': 2, 'col_3': 14}]
```

In [92]:
```python
for row in df.itertuples():
    print (row.Symbol)
    break
```

```
A1BG
```

In [87]:
```python
df.columns
```

Out[87]:
```
Index(['#tax_id', 'GeneID', 'Symbol', 'LocusTag', 'Synonyms', 'dbXrefs',
       'chromosome', 'map_location', 'description', 'type_of_gene',
       'Symbol_from_nomenclature_authority',
       'Full_name_from_nomenclature_authority', 'Nomenclature_status',
       'Other_designations', 'Modification_date', 'Feature_type'],
      dtype='object')
```

In [93]:
```python
d
```

Out[93]:

	col_1	col_2	col_3
0	1	2	10
1	2	2	11
2	3	1	12
3	2	4	13
4	1	2	14

In [97]:
```python
import random

b = {
    'col_1': [random.randint(1,4) for x in range(20)],
    'col_2': [random.randint(1,4) for x in range(20)],
    'col_3': [random.randint(1,4) for x in range(20)],
}
```

In [98]:
```python
d = pd.DataFrame(b)
```

In [99]:
```python
d
```

Out[99]:

	col_1	col_2	col_3
0	4	3	1
1	1	3	1
2	2	1	3
3	4	3	2
4	1	4	1
5	1	4	3
6	2	2	2
7	4	2	1

	col_1	col_2	col_3
8	2	1	2
9	3	1	3
10	1	3	2
11	3	2	2
12	2	2	3
13	4	3	4
14	2	4	3
15	1	4	3
16	4	2	3
17	3	1	2
18	1	4	2

In [104...
```python
d.groupby('col_1')[['col_1']].aggregate('count')
```

Out[104...

	col_1
col_1	
1	7
2	5
3	3
4	5

In [109...
```python
d.groupby('col_1')[['col_2']].aggregate('min')
```

Out[109...

	col_2
col_1	
1	3
2	1
3	1
4	2

In [147...
```python
d.groupby('col_1')[['col_2', 'col_3']].aggregate('min')
```

Out[147...

	col_2	col_3
col_1		
1	3	1
2	1	2
3	1	2
4	2	1

In [150...
```python
d.groupby('col_1')[['col_2', 'col_3']].aggregate({'col_2':'mean', 'col_3':'
```

Out[150…

col_1	col_2	col_3
1	3.571429	3
2	2.000000	3
3	1.333333	3
4	2.600000	4

In [113…
```python
d.groupby(['col_1', 'col_2'])[['col_3']].aggregate('min')
```

Out[113…

		col_3
col_1	col_2	
1	3	1
	4	1
2	1	2
	2	2
	4	3
3	1	2
	2	2
4	2	1
	3	1

In [115…
```python
df.columns
```

Out[115…
```
Index(['#tax_id', 'GeneID', 'Symbol', 'LocusTag', 'Synonyms', 'dbXrefs',
       'chromosome', 'map_location', 'description', 'type_of_gene',
       'Symbol_from_nomenclature_authority',
       'Full_name_from_nomenclature_authority', 'Nomenclature_status',
       'Other_designations', 'Modification_date', 'Feature_type'],
      dtype='object')
```

In [118…
```python
df.groupby(['chromosome', 'type_of_gene'])['chromosome', 'type_of_gene'].ag
```

```
<ipython-input-118-73a1318ec552>:1: FutureWarning: Indexing with multiple k
eys (implicitly converted to a tuple of keys) will be deprecated, use a lis
t instead.
  df.groupby(['chromosome', 'type_of_gene'])['chromosome', 'type_of_gene'].
aggregate('count')
```

Out[118…

		chromosome	type_of_gene
chromosome	type_of_gene		
-	ncRNA	1	1
	other	3	3
	protein-coding	13	13
	pseudo	1	1
	unknown	128	128
...	...	...	...
Y	ncRNA	107	107

		chromosome	type_of_gene
chromosome	type_of_gene		
	other	29	29
	protein-coding	46	46
	pseudo	389	389

In [108... `d`

Out[108...

	col_1	col_2	col_3
0	4	3	1
1	1	3	1
2	2	1	3
3	4	3	2
4	1	4	1
5	1	4	3
6	2	2	2
7	4	2	1
8	2	1	2
9	3	1	3
10	1	3	2
11	3	2	2
12	2	2	3
13	4	3	4
14	2	4	3
15	1	4	3
16	4	2	3
17	3	1	2
18	1	4	2
19	1	3	1

In [102... `d['col_1'].value_counts()`

Out[102...
```
1    7
4    5
2    5
3    3
Name: col_1, dtype: int64
```

In [120... `df['chromosome'].value_counts().plot(kind='bar')`

Out[120... `<AxesSubplot:>`

```
6000

5000

4000

3000

2000

1000
```
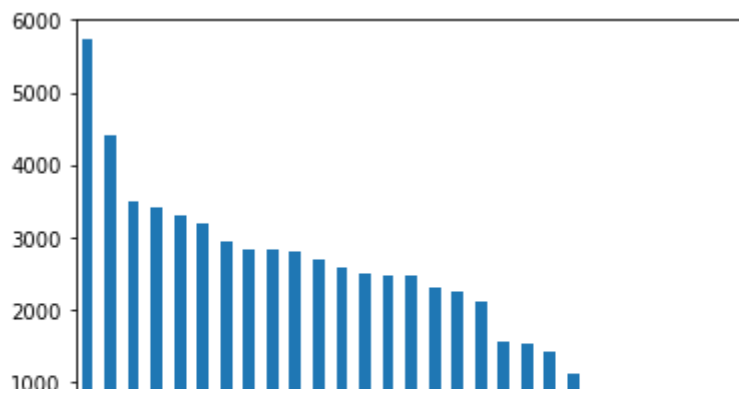
In [122… `df['chromosome'].value_counts().plot(kind='pie')`

Out[122… `<AxesSubplot:ylabel='chromosome'>`
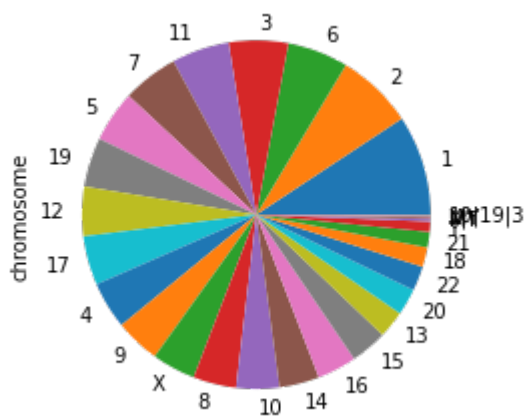
In [123… `# Why piechart are bad for data visualization`
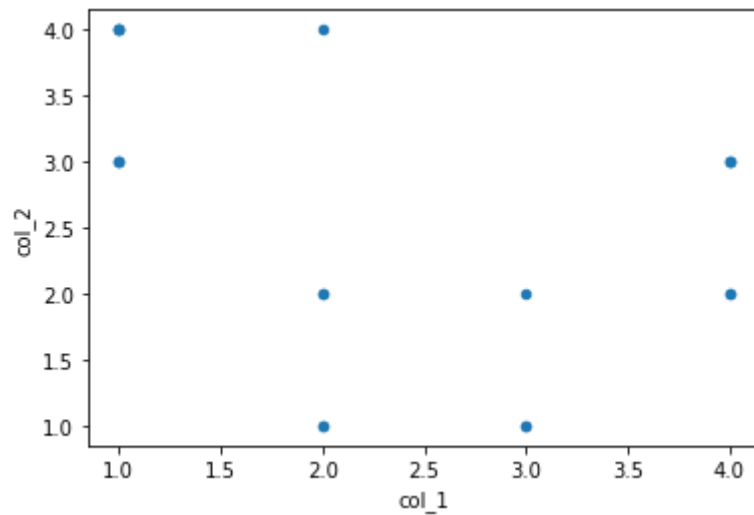
In [124… `d`

Out[124…

	col_1	col_2	col_3
0	4	3	1
1	1	3	1
2	2	1	3
3	4	3	2
4	1	4	1
5	1	4	3
6	2	2	2
7	4	2	1
8	2	1	2
9	3	1	3
10	1	3	2
11	3	2	2
12	2	2	3
13	4	3	4
14	2	4	3

	col_1	col_2	col_3
15	1	4	3
16	4	2	3
17	3	1	2
18	1	4	2

In [126...
```python
d.plot.scatter(x='col_1', y='col_2')
```
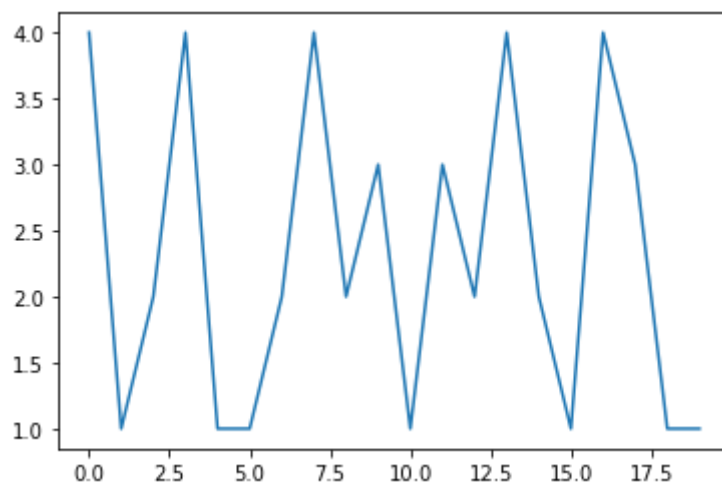
Out[126... `<AxesSubplot:xlabel='col_1', ylabel='col_2'>`

In [127...
```python
d['col_1'].plot()
```

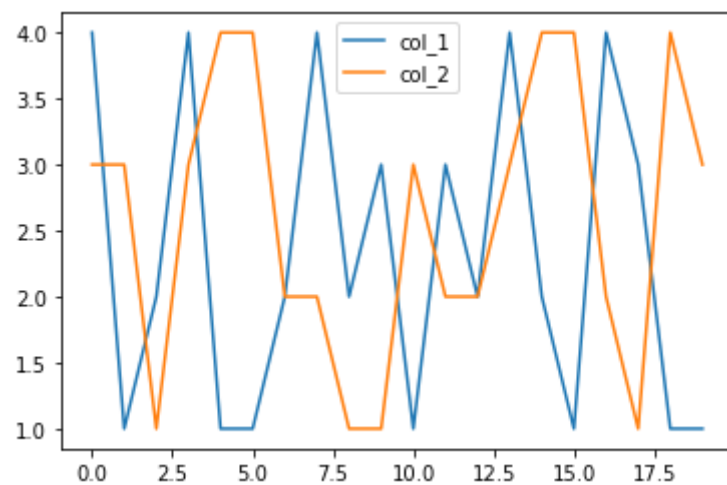Out[127... `<AxesSubplot:>`

In [128...
```python
d['col_1']
```

Out[128...
```
0     4
1     1
2     2
3     4
4     1
5     1
6     2
7     4
8     2
9     3
10    1
11    3
12    2
13    4
```

```
14    2
15    1
16    4
17    3
18    1
19    1
Name: col 1, dtype: int64
```

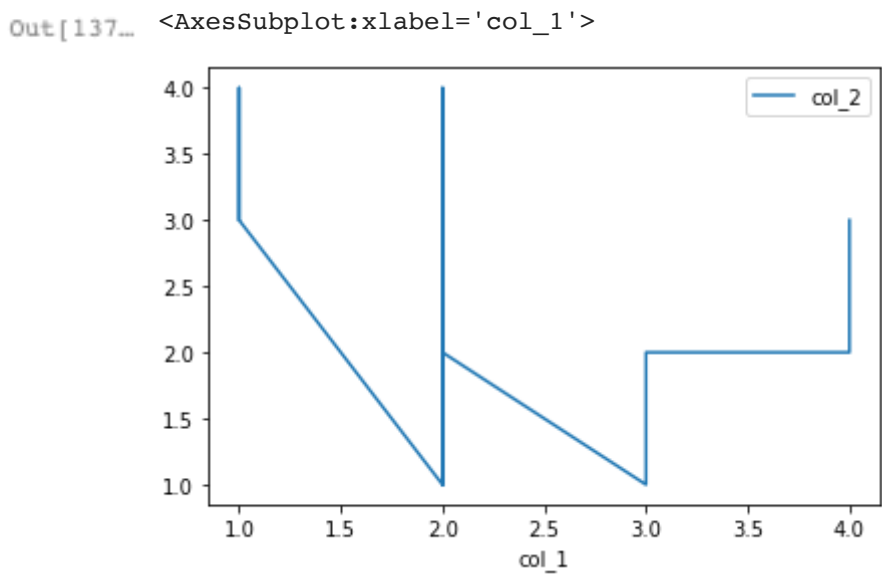In [129… `d[['col_1', 'col_2']].plot()`

Out[129… `<AxesSubplot:>`

In [130… `d`

Out[130…

	col_1	col_2	col_3
0	4	3	1
1	1	3	1
2	2	1	3
3	4	3	2
4	1	4	1
5	1	4	3
6	2	2	2
7	4	2	1
8	2	1	2
9	3	1	3
10	1	3	2
11	3	2	2
12	2	2	3
13	4	3	4
14	2	4	3
15	1	4	3
16	4	2	3
17	3	1	2
18	1	4	2
19	1	3	1

```
In [132...   d2 = d.sort_values('col_1')
```

```
In [137...   d2.plot(x='col_1', y='col_2')
```
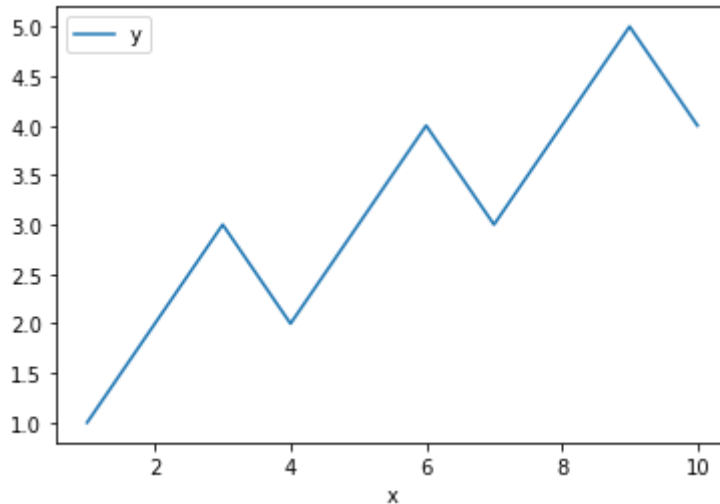
```
Out[137...   <AxesSubplot:xlabel='col_1'>
```



```
In [136...   d2
```

Out[136...

	col_1	col_2	col_3
19	1	3	1
1	1	3	1
4	1	4	1
5	1	4	3
15	1	4	3
18	1	4	2
10	1	3	2
2	2	1	3
6	2	2	2
8	2	1	2
14	2	4	3
12	2	2	3
17	3	1	2
9	3	1	3
11	3	2	2
7	4	2	1
16	4	2	3
3	4	3	2
13	4	3	4
0	4	3	1

```
In [139…   e = pd.DataFrame({
               'x':[1,2,3,4,5,6,7,8,9,10],
               'y':[1,2,3,2,3,4,3,4,5,4,],
           })
```

```
In [141…   e.plot(x='x', y='y')
```

Out[141…   <AxesSubplot:xlabel='x'>


```
In [143…   gwas = pd.read_csv('https://www.ebi.ac.uk/gwas/api/search/downloads/full',
```

```
           /Users/admin/anaconda3/lib/python3.8/site-packages/IPython/core/interactive
           shell.py:3146: DtypeWarning: Columns (9,11,12,23,27) have mixed types.Speci
           fy dtype option on import or set low_memory=False.
             has_raised = await self.run_ast_nodes(code_ast.body, cell_name,
```

```
In [144…   gwas.shape
```

Out[144…   (251401, 34)

```
In [145…   gwas[ (gwas["DISEASE/TRAIT"].str.contains('Breast')) & (
               gwas["PVALUE_MLOG"]>10)].groupby("CHR_ID")['CHR_ID'].aggregate('count')
```

Out[145…   <AxesSubplot:xlabel='CHR_ID'>
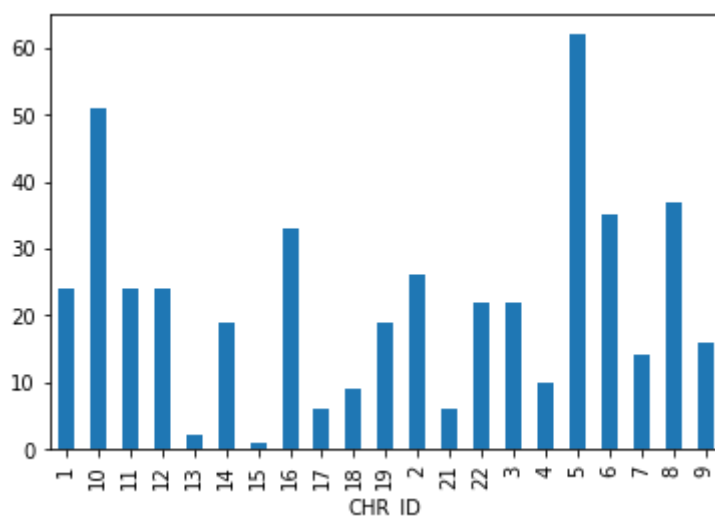

```
In [151…   df
```

Out[151…   **#tax_id   GeneID   Symbol   LocusTag   Synonyms**

	#tax_id	GeneID	Symbol	LocusTag	Synonyms	
0	9606	1	A1BG	-	A1B|ABG|GAB|HYST2477	MIM:138670
1	9606	2	A2M	-	A2MD|CPAMD5|FWP007|S863-7	MIM:103950|
2	9606	3	A2MP1	-	A2MP	I
3	9606	9	NAT1	-	AAC1|MNAT|NAT-1|NATI	MIM:108345|HGI
4	9606	10	NAT2	-	AAC2|NAT-2|PNAT	MIM:612182|HGN
...	...	...	...	...	...	
61713	741158	8923215	trnD	-	-	
61714	741158	8923216	trnP	-	-	
61715	741158	8923217	trnA	-	-	
61716	741158	8923218	COX1	-	-	
61717	741158	8923219	16S rRNA	-	-	

```
In [155...  gwas['c2'] = pd.to_datetime(gwas['DATE'])
```

```
In [156...  gwas['c2']
```

```
Out[156...  0         2020-04-15
           1         2019-10-30
           2         2019-10-30
           3         2020-04-15
           4         2019-10-30
                        ...
           251396    2020-09-02
           251397    2020-09-02
           251398    2020-09-02
           251399    2020-09-02
           251400    2020-09-02
           Name: c2, Length: 251401, dtype: datetime64[ns]
```

```
In [ ]:
```

```
In [ ]:
```

```
In [ ]:
```

```
In [ ]:
```