```
1   import numpy as np
2   import pandas as pd
```

```
1 import matplotlib.pyplot as plt
2 import seaborn as sns
```

```
1 !wget "https://api.covid19india.org/states_daily.json"
```

```
--2022-06-10 14:36:56--  https://api.covid19india.org/states_daily.jsc
Resolving api.covid19india.org (api.covid19india.org)... 74.125.23.121
Connecting to api.covid19india.org (api.covid19india.org)|74.125.23.12
HTTP request sent, awaiting response... 301 Moved Permanently
Location: http://data.covid19india.org/states_daily.json [following]
--2022-06-10 14:36:57--  http://data.covid19india.org/states_daily.jsc
Resolving data.covid19india.org (data.covid19india.org)... 185.199.108
Connecting to data.covid19india.org (data.covid19india.org)|185.199.10
HTTP request sent, awaiting response... 301 Moved Permanently
Location: https://data.covid19india.org/states_daily.json [following]
--2022-06-10 14:36:57--  https://data.covid19india.org/states_daily.js
Connecting to data.covid19india.org (data.covid19india.org)|185.199.10
HTTP request sent, awaiting response... 200 OK
Length: 1061528 (1.0M) [application/json]
Saving to: ‘states_daily.json.1’

states_daily.json.1 100%[===================>]   1.01M  --.-KB/s    in

2022-06-10 14:36:57 (17.8 MB/s) - ‘states_daily.json.1’ saved [106152
```

Alternate way

```
1 import urllib.request
```

```
1 url = "https://api.covid19india.org/states_daily.json"
2 url
```

```
'https://api.covid19india.org/states_daily.json'
```

```
1 urllib.request.urlretrieve(url,'data.json')
```

```
('data.json', <http.client.HTTPMessage at 0x7f4895082350>)
```

```
cd=pd.read_json('data.json')
cd
```

	states_daily
0	{'an': '0', 'ap': '1', 'ar': '0', 'as': '0', '...
1	{'an': '0', 'ap': '0', 'ar': '0', 'as': '0', '...
2	{'an': '0', 'ap': '0', 'ar': '0', 'as': '0', '...
3	{'an': '0', 'ap': '0', 'ar': '0', 'as': '0', '...
4	{'an': '0', 'ap': '0', 'ar': '0', 'as': '0', '...
...	...
1558	{'an': '2', 'ap': '1835', 'ar': '255', 'as': '...
1559	{'an': '0', 'ap': '16', 'ar': '0', 'as': '10',...
1560	{'an': '1', 'ap': '909', 'ar': '165', 'as': '7...
1561	{'an': '0', 'ap': '1543', 'ar': '249', 'as': '...
1562	{'an': '0', 'ap': '13', 'ar': '0', 'as': '10',...

1563 rows × 1 columns

Online csv file retreievel example

```
url ='https://www.stats.govt.nz/assets/Uploads/Annual-enterprise-survey
url
```

```
'https://www.stats.govt.nz/assets/Uploads/Annual-enterprise-survey/An
nual-enterprise-survey-2020-financial-year-provisional/Download-data/
annual-enterprise-survey-2020-financial-year-provisional-size-bands-c
```

```
urllib.request.urlretrieve(url,'sample.csv')
```

```
('sample.csv', <http.client.HTTPMessage at 0x7f489438bf90>)
```

```
scsv= pd.read_csv('sample.csv')
scsv.head()
```

	year	industry_code_ANZSIC	industry_name_ANZSIC	rme_size_grp	var
0	2011	A	Agriculture, Forestry and Fishing	a_0	Activ
1	2011	A	Agriculture, Forestry and Fishing	a_0	emp
2	2011	A	Agriculture, Forestry and Fishing	a_0	S and
3	2011	A	Agriculture, Forestry and Fishing	a_0	gove fu gra su

```
import json
with open('data.json') as f :
    data= json.load(f)
data
```

```
      'kl': '0',
      'la': '0',
      'ld': '0',
      'mh': '19',
      'ml': '0',
      'mn': '0',
      'mp': '4',
      'mz': '0',
      'nl': '0',
      'or': '0',
      'pb': '1',
      'py': '0',
      'rj': '7',
      'sk': '0',
      'status': 'Deceased',
      'tg': '0',
      'tn': '1',
      'tr': '0',
      'tt': '56',
      'un': '0',
      'up': '3',
      'ut': '0',
      'wb': '2'},
     {'an': '0',
      'ap': '80',
      'ar': '0',
```

```
      'as': '0',
      'br': '69',
      'ch': '9',
      'ct': '0',
      'date': '27-Apr-20',
      'dateymd': '2020-04-27',
      'dd': '0',
      'dl': '190',
      'dn': '0',
      'ga': '0',
      'gj': '247',
      'hp': '0',
      'hr': '5',
      'jh': '21',
      'jk': '23',
      'ka': '9',
      'kl': '13',
      'la': '0',
      'ld': '0',
      'mh': '522',
      'ml': '0',
      'mn': '0',
      'mp': '75',
      'mz': '0',
      'nl': '0',
      'or': '15',
      'pb': '8',
      'py': '0',
      'rj': '77',
      'sk': '0',
      'status': 'Confirmed',
      'tg': '2',
      'tn': '52',
      'tr': '0',
```

```
1 data = data['states_daily']
2 data
```

```
[{'an': '0',
  'ap': '1',
  'ar': '0',
  'as': '0',
  'br': '0',
  'ch': '0',
  'ct': '0',
  'date': '14-Mar-20',
  'dateymd': '2020-03-14',
  'dd': '0',
  'dl': '7',
  'dn': '0',
  'ga': '0',
  'gj': '0',
  'hp': '0',
```

```
 'hp': '0',
 'hr': '14',
 'jh': '0',
 'jk': '2',
 'ka': '6',
 'kl': '19',
 'la': '0',
 'ld': '0',
 'mh': '14',
 'ml': '0',
 'mn': '0',
 'mp': '0',
 'mz': '0',
 'nl': '0',
 'or': '0',
 'pb': '1',
 'py': '0',
 'rj': '3',
 'sk': '0',
 'status': 'Confirmed',
 'tg': '1',
 'tn': '1',
 'tr': '0',
 'tt': '81',
 'un': '0',
 'up': '12',
 'ut': '0',
 'wb': '0'},
{'an': '0',
 'ap': '0',
 'ar': '0',
 'as': '0',
 'br': '0',
 'ch': '0',
 'ct': '0',
 'date': '14-Mar-20',
 'dateymd': '2020-03-14',
 'dd': '0',
 'dl': '1',
 'dn': '0',
 'ga': '0',
 'gj': '0',
 'hp': '0',
 'hr': '0',
 'jh': '0',
 'jk': '0',
```

```
cd= pd.json_normalize(data)
cd
```

	an	ap	ar	as	br	ch	ct	date	dateymd	dd	...	sk	stat
0	0	1	0	0	0	0	0	14-Mar-20	2020-03-14	0	...	0	Confirm
1	0	0	0	0	0	0	0	14-Mar-20	2020-03-14	0	...	0	Recover
2	0	0	0	0	0	0	0	14-Mar-20	2020-03-14	0	...	0	Deceas
3	0	0	0	0	0	0	0	15-Mar-20	2020-03-15	0	...	0	Confirm
4	0	0	0	0	0	0	0	15-Mar-20	2020-03-15	0	...	0	Recover
...	...	...	...	...	...	...	...	...	...	...	...	...	
1558	2	1835	255	857	38	1	114	15-Aug-	2021-08-15	0	...	213	Recover

```
df=cd
```

1559	0	16	0	10	0	0	1	15-Aug-21	2021-08-15	0	...	0	Deceas
1560	1	909	165	758	14	2	68	16-Aug-21	2021-08-16	0	...	20	Confirm
1561	0	1543	249	1014	42	3	224	16-Aug-21	2021-08-16	0	...	147	Recover
1562	0	13	0	10	0	0	1	16-Aug-21	2021-08-16	0	...	0	Deceas

1563 rows × 42 columns

```
df.date=pd.to_datetime(df.date)
df
```

	an	ap	ar	as	br	ch	ct	date	dateymd	dd	...	sk	stat
0	0	1	0	0	0	0	0	2020-03-14	2020-03-14	0	...	0	Confirn
1	0	0	0	0	0	0	0	2020-03-14	2020-03-14	0	...	0	Recove
2	0	0	0	0	0	0	0	2020-03-14	2020-03-14	0	...	0	Deceas
3	0	0	0	0	0	0	0	2020-03-15	2020-03-15	0	...	0	Confirn
4	0	0	0	0	0	0	0	2020-03-15	2020-03-15	0	...	0	Recove
...	...	...	...	...	...	...	...	...	...	...	...	...	
1558	2	1835	255	857	38	1	114	2021-08-15	2021-08-15	0	...	213	Recove
1559	0	16	0	10	0	0	1	2021-08-15	2021-08-15	0	...	0	Deceas
1560	1	909	165	758	14	2	68	2021-08-16	2021-08-16	0	...	20	Confirn
1561	0	1543	249	1014	42	3	224	2021-08-16	2021-08-16	0	...	147	Recove
1562	0	13	0	10	0	0	1	2021-08-16	2021-08-16	0	...	0	Deceas

1563 rows × 42 columns

```
df=df[df['status']=='Confirmed']
df
```

	an	ap	ar	as	br	ch	ct	date	dateymd	dd	...	sk	status
0	0	1	0	0	0	0	0	2020-03-14	2020-03-14	0	...	0	Confirmed
3	0	0	0	0	0	0	0	2020-03-15	2020-03-15	0	...	0	Confirmed
6	0	0	0	0	0	0	0	2020-03-16	2020-03-16	0	...	0	Confirmed
9	0	0	0	0	0	0	0	2020-03-17	2020-03-17	0	...	0	Confirmed
12	0	0	0	0	0	0	0	2020-03-18	2020-03-18	0	...	0	Confirmed
...	...	...	...	...	...	...	...	...	...	...	...	...	..
1548	0	1859	180	935	43	12	98	2021-08-12	2021-08-12	0	...	100	Confirmed
1551	0	1746	166	763	47	15	77	2021-08-13	2021-08-13	0	...	150	Confirmed
1554	0	1535	161	755	39	4	83	2021-08-14	2021-08-14	0	...	129	Confirmed

```
df.drop('status',axis=1,inplace = True)
df.head()
```

```
/usr/local/lib/python3.7/dist-packages/pandas/core/frame.py:4913: Sett
A value is trying to be set on a copy of a slice from a DataFrame

See the caveats in the documentation: https://pandas.pydata.org/pandas
  errors=errors,
```

	an	ap	ar	as	br	ch	ct	date	dateymd	dd	...	rj	sk	tg	tn	tr
0	0	1	0	0	0	0	0	2020-03-14	2020-03-14	0	...	3	0	1	1	0
3	0	0	0	0	0	0	0	2020-03-15	2020-03-15	0	...	1	0	2	0	0
6	0	0	0	0	0	0	0	2020-03-16	2020-03-16	0	...	0	0	1	0	0
9	0	0	0	0	0	0	0	2020-03-17	2020-03-17	0	...	0	0	1	0	0
12	0	0	0	0	0	0	0	2020-03-18	2020-03-18	0	...	3	0	3	1	0

```
1 df.drop('dateymd',axis=1,inplace = True)
```

```
/usr/local/lib/python3.7/dist-packages/pandas/core/frame.py:4913: Sett
A value is trying to be set on a copy of a slice from a DataFrame

See the caveats in the documentation: https://pandas.pydata.org/pandas
  errors=errors,
```

```
1 df.head()
```

	an	ap	ar	as	br	ch	ct	date	dd	dl	...	rj	sk	tg	tn	tr	tt
0	0	1	0	0	0	0	0	2020-03-14	0	7	...	3	0	1	1	0	81
3	0	0	0	0	0	0	0	2020-03-15	0	0	...	1	0	2	0	0	27
6	0	0	0	0	0	0	0	2020-03-16	0	0	...	0	0	1	0	0	15
9	0	0	0	0	0	0	0	2020-03-17	0	1	...	0	0	1	0	0	11

```
1 df.columns
```

```
Index(['an', 'ap', 'ar', 'as', 'br', 'ch', 'ct', 'date', 'dd', 'dl', '
       'ga', 'gj', 'hp', 'hr', 'jh', 'jk', 'ka', 'kl', 'la', 'ld', 'mh
       'mn', 'mp', 'mz', 'nl', 'or', 'pb', 'py', 'rj', 'sk', 'tg', 'tr
       'tt', 'un', 'up', 'ut', 'wb'],
      dtype='object')
```

```
1 df.set_index('date',inplace=True)
```

```
1 df=df.apply(pd.to_numeric) #apply for every cols
```

```
1 df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
DatetimeIndex: 521 entries, 2020-03-14 to 2021-08-16
Data columns (total 39 columns):
 #   Column  Non-Null Count  Dtype
---  ------  --------------  -----
 0   an      521 non-null    int64
 1   ap      521 non-null    int64
 2   ar      521 non-null    int64
 3   as      521 non-null    int64
 4   br      521 non-null    int64
 5   ch      521 non-null    int64
 6   ct      521 non-null    int64
 7   dd      521 non-null    int64
 8   dl      521 non-null    int64
 9   dn      521 non-null    int64
 10  ga      521 non-null    int64
 11  gj      521 non-null    int64
 12  hp      521 non-null    int64
 13  hr      521 non-null    int64
 14  jh      521 non-null    int64
 15  jk      521 non-null    int64
 16  ka      521 non-null    int64
 17  kl      521 non-null    int64
 18  la      521 non-null    int64
 19  ld      521 non-null    int64
 20  mh      521 non-null    int64
 21  ml      521 non-null    int64
 22  mn      521 non-null    int64
 23  mp      521 non-null    int64
 24  mz      521 non-null    int64
 25  nl      521 non-null    int64
 26  or      521 non-null    int64
 27  pb      521 non-null    int64
 28  py      521 non-null    int64
 29  rj      521 non-null    int64
 30  sk      521 non-null    int64
 31  tg      521 non-null    int64
 32  tn      521 non-null    int64
 33  tr      521 non-null    int64
 34  tt      521 non-null    int64
 35  un      521 non-null    int64
 36  up      521 non-null    int64
 37  ut      521 non-null    int64
 38  wb      521 non-null    int64
dtypes: int64(39)
memory usage: 162.8 KB
```

```
df2=df.tail(7)
df2
```

date	an	ap	ar	as	br	ch	ct	dd	dl	dn	...	rj	sk	tg	tn
2021-08-10	2	1461	233	929	44	8	112	0	52	1	...	11	110	494	1893
2021-08-11	0	1869	188	886	47	5	83	0	37	0	...	19	157	482	1964
2021-08-12	0	1859	180	935	43	12	98	0	49	1	...	17	100	453	1942
2021-08-13	0	1746	166	763	47	15	77	0	50	0	...	24	150	427	1933
2021-08-14	0	1535	161	755	39	4	83	0	50	0	...	14	129	420	1916
2021-															

```
1 df2.style
```

date	an	ap	ar	as	br	ch	ct	dd	dl	dn	ga	gj	hp	hr	jh
2021-08-10 00:00:00	2	1461	233	929	44	8	112	0	52	1	141	21	419	23	34
2021-08-11 00:00:00	0	1869	188	886	47	5	83	0	37	0	103	16	374	16	14
2021-08-12 00:00:00	0	1859	180	935	43	12	98	0	49	1	88	17	354	16	44
2021-08-13 00:00:00	0	1746	166	763	47	15	77	0	50	0	67	23	333	26	32
2021-08-14 00:00:00	0	1535	161	755	39	4	83	0	50	0	88	25	284	14	28
2021-08-15 00:00:00	0	1506	48	411	28	1	49	0	53	0	75	16	182	22	27
2021-08-16 00:00:00	1	909	165	758	14	2	68	0	27	2	62	14	276	22	35

Data frame table styling

```
1 def color_red_negative(x) :
2     color = 'red' if x <0 else 'blue'
3     return 'color:' + color
```

```
df2.style.applymap(color_red_negative) #applymap for rows # for each cel
```

date	an	ap	ar	as	br	ch	ct	dd	dl	dn	ga	gj	hp	hr	jh
2021-08-10 00:00:00	2	1461	233	929	44	8	112	0	52	1	141	21	419	23	34
2021-08-11 00:00:00	0	1869	188	886	47	5	83	0	37	0	103	16	374	16	14
2021-08-12 00:00:00	0	1859	180	935	43	12	98	0	49	1	88	17	354	16	44
2021-08-13 00:00:00	0	1746	166	763	47	15	77	0	50	0	67	23	333	26	32
2021-08-14 00:00:00	0	1535	161	755	39	4	83	0	50	0	88	25	284	14	28
2021-08-15 00:00:00	0	1506	48	411	28	1	49	0	53	0	75	16	182	22	27
2021-08-16 00:00:00	1	909	165	758	14	2	68	0	27	2	62	14	276	22	35

```
df2.style.highlight_max(color='red') #columnwise max
```

date	an	ap	ar	as	br	ch	ct	dd	dl	dn	ga	gj	hp	hr	jh
2021-08-10 00:00:00	2	1461	233	929	44	8	112	0	52	1	141	21	419	23	34
2021-08-11 00:00:00	0	1869	188	886	47	5	83	0	37	0	103	16	374	16	14
2021-08-12 00:00:00	0	1859	180	935	43	12	98	0	49	1	88	17	354	16	44
2021-08-13 00:00:00	0	1746	166	763	47	15	77	0	50	0	67	23	333	26	32
2021-08-14 00:00:00	0	1535	161	755	39	4	83	0	50	0	88	25	284	14	28
2021-08-15 00:00:00	0	1506	48	411	28	1	49	0	53	0	75	16	182	22	27
2021-08-16 00:00:00	1	909	165	758	14	2	68	0	27	2	62	14	276	22	35

```
df2.style.highlight_max(color='red',axis=1) #rowwise max , axis =1
```

date	an	ap	ar	as	br	ch	ct	dd	dl	dn	ga	gj	hp	hr	jh
2021-08-10 00:00:00	2	1461	233	929	44	8	112	0	52	1	141	21	419	23	34
2021-08-11 00:00:00	0	1869	188	886	47	5	83	0	37	0	103	16	374	16	14
2021-08-12 00:00:00	0	1859	180	935	43	12	98	0	49	1	88	17	354	16	44
2021-08-13 00:00:00	0	1746	166	763	47	15	77	0	50	0	67	23	333	26	32
2021-08-14 00:00:00	0	1535	161	755	39	4	83	0	50	0	88	25	284	14	28
2021-08-15 00:00:00	0	1506	48	411	28	1	49	0	53	0	75	16	182	22	27
2021-08-16 00:00:00	1	909	165	758	14	2	68	0	27	2	62	14	276	22	35

```
df2.style.highlight_max(color='green').highlight_min(color='red') #both
```

date	an	ap	ar	as	br	ch	ct	dd	dl	dn	ga	gj	hp	hr	jh
2021-08-10 00:00:00	2	1461	233	929	44	8	112	0	52	1	141	21	419	23	34
2021-08-11 00:00:00	0	1869	188	886	47	5	83	0	37	0	103	16	374	16	14
2021-08-12 00:00:00	0	1859	180	935	43	12	98	0	49	1	88	17	354	16	44
2021-08-13 00:00:00	0	1746	166	763	47	15	77	0	50	0	67	23	333	26	32
2021-08-14 00:00:00	0	1535	161	755	39	4	83	0	50	0	88	25	284	14	28
2021-08-15 00:00:00	0	1506	48	411	28	1	49	0	53	0	75	16	182	22	27
2021-08-16 00:00:00	1	909	165	758	14	2	68	0	27	2	62	14	276	22	35

```
df2.drop('tt',axis=1,inplace=True)
df2
```

```
/usr/local/lib/python3.7/dist-packages/pandas/core/frame.py:4913: Sett
A value is trying to be set on a copy of a slice from a DataFrame

See the caveats in the documentation: https://pandas.pydata.org/pandas
  errors=errors,
```

date	an	ap	ar	as	br	ch	ct	dd	dl	dn	...	py	rj	sk	tg	
2021-08-10	2	1461	233	929	44	8	112	0	52	1	...	101	11	110	494	1
2021-08-11	0	1869	188	886	47	5	83	0	37	0	...	114	19	157	482	1
2021-08-12	0	1859	180	935	43	12	98	0	49	1	...	109	17	100	453	1
2021-08-13	0	1746	166	763	47	15	77	0	50	0	...	113	24	150	427	1
2021-08-14	0	1535	161	755	39	4	83	0	50	0	...	101	14	129	420	1
2021-	0	1506	48	411	28	1	49	0	53	0	...	79	18	152	245	1

```
def bold_max_value(x) :
    ismax=(x==x.max())
    return ['font-weight: bold' if y else '' for y in ismax]
```

```
df2.style.apply(bold_max_value)
```

date	an	ap	ar	as	br	ch	ct	dd	dl	dn	ga	gj	hp	hr	jh
2021-08-10 00:00:00	**2**	1461	**233**	929	44	8	**112**	**0**	52	1	**141**	21	**419**	23	34
2021-08-11 00:00:00	0	**1869**	188	886	**47**	5	83	**0**	37	0	103	16	374	16	14
2021-08-12 00:00:00	0	1859	180	**935**	43	12	98	**0**	49	1	88	17	354	16	**44**
2021-08-13 00:00:00	0	1746	166	763	**47**	**15**	77	**0**	50	0	67	23	333	**26**	32
2021-08-14 00:00:00	0	1535	161	755	39	4	83	**0**	50	0	88	**25**	284	14	28
2021-08-15 00:00:00	0	1506	48	411	28	1	49	**0**	**53**	0	75	16	182	22	27
2021-08-16 00:00:00	1	909	165	758	14	2	68	**0**	27	**2**	62	14	276	22	35

```
df2.style.apply(bold_max_value).highlight_max(color='red') #statewise
```

date	an	ap	ar	as	br	ch	ct	dd	dl	dn	ga	gj	hp	hr	jh
2021-08-10 00:00:00	2	1461	233	929	44	8	112	0	52	1	141	21	419	23	34
2021-08-11 00:00:00	0	1869	188	886	47	5	83	0	37	0	103	16	374	16	14
2021-08-12 00:00:00	0	1859	180	935	43	12	98	0	49	1	88	17	354	16	44
2021-08-13 00:00:00	0	1746	166	763	47	15	77	0	50	0	67	23	333	26	32
2021-08-14 00:00:00	0	1535	161	755	39	4	83	0	50	0	88	25	284	14	28
2021-08-15 00:00:00	0	1506	48	411	28	1	49	0	53	0	75	16	182	22	27
2021-08-16 00:00:00	1	909	165	758	14	2	68	0	27	2	62	14	276	22	35

```
df2.style.apply(bold_max_value).highlight_max(color='red',axis=1) #rowwi
```

date	an	ap	ar	as	br	ch	ct	dd	dl	dn	ga	gj	hp	hr	jh
2021-08-10 00:00:00	**2**	1461	**233**	929	44	8	**112**	**0**	52	1	**141**	21	**419**	23	34
2021-08-11 00:00:00	0	**1869**	188	886	**47**	5	83	**0**	37	0	103	16	374	16	14
2021-08-12 00:00:00	0	1859	180	**935**	43	12	98	**0**	49	1	88	17	354	16	**44**
2021-08-13 00:00:00	0	1746	166	763	**47**	**15**	77	**0**	50	0	67	23	333	**26**	32
2021-08-14 00:00:00	0	1535	161	755	39	4	83	**0**	50	0	88	**25**	284	14	28
2021-08-15 00:00:00	0	1506	48	411	28	1	49	**0**	**53**	0	75	16	182	22	27
2021-08-16 00:00:00	1	909	165	758	14	2	68	**0**	27	**2**	62	14	276	22	35

```
1 df2.style.background_gradient(cmap='Greens') #column_wise
```

date	an	ap	ar	as	br	ch	ct	dd	dl	dn	ga	gj	hp	hr	jh
2021-08-10 00:00:00	2	1461	233	929	44	8	112	0	52	1	141	21	419	23	34
2021-08-11 00:00:00	0	1869	188	886	47	5	83	0	37	0	103	16	374	16	14
2021-08-12 00:00:00	0	1859	180	935	43	12	98	0	49	1	88	17	354	16	44
2021-08-13 00:00:00	0	1746	166	763	47	15	77	0	50	0	67	23	333	26	32
2021-08-14 00:00:00	0	1535	161	755	39	4	83	0	50	0	88	25	284	14	28
2021-08-15 00:00:00	0	1506	48	411	28	1	49	0	53	0	75	16	182	22	27
2021-08-16 00:00:00	1	909	165	758	14	2	68	0	27	2	62	14	276	22	35

```
df2.style.background_gradient(cmap='Reds',axis=1)
```

date	an	ap	ar	as	br	ch	ct	dd	dl	dn	ga	gj	hp	hr	jh
2021-08-10 00:00:00	2	1461	233	929	44	8	112	0	52	1	141	21	419	23	34
2021-08-11 00:00:00	0	1869	188	886	47	5	83	0	37	0	103	16	374	16	14
2021-08-12 00:00:00	0	1859	180	935	43	12	98	0	49	1	88	17	354	16	44
2021-08-13 00:00:00	0	1746	166	763	47	15	77	0	50	0	67	23	333	26	32
2021-08-14 00:00:00	0	1535	161	755	39	4	83	0	50	0	88	25	284	14	28
2021-08-15 00:00:00	0	1506	48	411	28	1	49	0	53	0	75	16	182	22	27
2021-08-16 00:00:00	1	909	165	758	14	2	68	0	27	2	62	14	276	22	35

```
df2.style.background_gradient(cmap='Reds',subset=['kl','ka','ap','dl'])#
```

date	an	ap	ar	as	br	ch	ct	dd	dl	dn	ga	gj	hp	hr	jh
2021-08-10 00:00:00	2	1461	233	929	44	8	112	0	52	1	141	21	419	23	34
2021-08-11 00:00:00	0	1869	188	886	47	5	83	0	37	0	103	16	374	16	14
2021-08-12 00:00:00	0	1859	180	935	43	12	98	0	49	1	88	17	354	16	44
2021-08-13 00:00:00	0	1746	166	763	47	15	77	0	50	0	67	23	333	26	32
2021-08-14 00:00:00	0	1535	161	755	39	4	83	0	50	0	88	25	284	14	28
2021-08-15 00:00:00	0	1506	48	411	28	1	49	0	53	0	75	16	182	22	27
2021-08-16 00:00:00	1	909	165	758	14	2	68	0	27	2	62	14	276	22	35

```
1 df2.style.bar() #columnwise
```

date	an	ap	ar	as	br	ch	ct	dd	dl	dn	ga	gj	hp	hr	jh
2021-08-10 00:00:00	2	1461	233	929	44	8	112	0	52	1	141	21	419	23	34
2021-08-11 00:00:00	0	1869	188	886	47	5	83	0	37	0	103	16	374	16	14
2021-08-12 00:00:00	0	1859	180	935	43	12	98	0	49	1	88	17	354	16	44
2021-08-13 00:00:00	0	1746	166	763	47	15	77	0	50	0	67	23	333	26	32
2021-08-14 00:00:00	0	1535	161	755	39	4	83	0	50	0	88	25	284	14	28
2021-08-15 00:00:00	0	1506	48	411	28	1	49	0	53	0	75	16	182	22	27
2021-08-16 00:00:00	1	909	165	758	14	2	68	0	27	2	62	14	276	22	35

```
1 df2.style.bar(axis=1) #rowwise
```

date	an	ap	ar	as	br	ch	ct	dd	dl	dn	ga	gj	hp	hr	jh
2021-08-10 00:00:00	2	1461	233	929	44	8	112	0	52	1	141	21	419	23	34
2021-08-11 00:00:00	0	1869	188	886	47	5	83	0	37	0	103	16	374	16	14
2021-08-12 00:00:00	0	1859	180	935	43	12	98	0	49	1	88	17	354	16	44
2021-08-13 00:00:00	0	1746	166	763	47	15	77	0	50	0	67	23	333	26	32
2021-08-14 00:00:00	0	1535	161	755	39	4	83	0	50	0	88	25	284	14	28
2021-08-15 00:00:00	0	1506	48	411	28	1	49	0	53	0	75	16	182	22	27
2021-08-16 00:00:00	1	909	165	758	14	2	68	0	27	2	62	14	276	22	35

```
df2.style.bar(subset=['kl','ka','mh'])
```

date	an	ap	ar	as	br	ch	ct	dd	dl	dn	ga	gj	hp	hr	jh
2021-08-10 00:00:00	2	1461	233	929	44	8	112	0	52	1	141	21	419	23	34
2021-08-11 00:00:00	0	1869	188	886	47	5	83	0	37	0	103	16	374	16	14
2021-08-12 00:00:00	0	1859	180	935	43	12	98	0	49	1	88	17	354	16	44
2021-08-13 00:00:00	0	1746	166	763	47	15	77	0	50	0	67	23	333	26	32
2021-08-14 00:00:00	0	1535	161	755	39	4	83	0	50	0	88	25	284	14	28
2021-08-15 00:00:00	0	1506	48	411	28	1	49	0	53	0	75	16	182	22	27
2021-08-16 00:00:00	1	909	165	758	14	2	68	0	27	2	62	14	276	22	35

```
1 df2[['kl','ka','mh']].style.bar()
```

date	kl	ka	mh
2021-08-10 00:00:00	21119	1338	5609
2021-08-11 00:00:00	23500	1826	5560
2021-08-12 00:00:00	21445	1857	6388
2021-08-13 00:00:00	20452	1669	6686
2021-08-14 00:00:00	19451	1632	5787

```
1 df2[['kl','ka','mh']].style.bar(axis=1)
```

date	kl	ka	mh
2021-08-10 00:00:00	21119	1338	5609
2021-08-11 00:00:00	23500	1826	5560
2021-08-12 00:00:00	21445	1857	6388
2021-08-13 00:00:00	20452	1669	6686
2021-08-14 00:00:00	19451	1632	5787

```
df2[['kl','hr','mh']].style.bar(subset=['kl'],color='red').bar(subset=['
```

date	kl	hr	mh
2021-08-10 00:00:00	21119	23	5609
2021-08-11 00:00:00	23500	16	5560
2021-08-12 00:00:00	21445	16	6388
2021-08-13 00:00:00	20452	26	6686
2021-08-14 00:00:00	19451	14	5787

```
x=np.random.normal(size=1000)
x
```

```
array([-1.38322217e+00, -2.46246099e-01,  1.24474920e+00, -2.24595390e
        1.30052342e-01, -8.03235432e-02,  8.59395086e-01, -1.24348778e
       -2.50069691e-01, -8.10033648e-01, -2.97807590e-01, -6.99263630e
       -5.03685207e-01, -2.09559050e+00, -3.08714230e+00,  9.93640061e
       -5.32295272e-01,  5.58544707e-01,  2.64039843e+00, -1.00989328e
       -4.97854388e-01,  1.89721976e+00, -6.17583492e-01, -9.79834694e
        2.86391322e-01,  6.36834673e-01, -2.18321720e-01, -2.83539300e
       -1.26256355e+00, -1.21413881e+00, -6.09740415e-01,  1.58720521e
       -9.02242792e-01, -4.67485689e-01,  7.56694539e-02,  3.49724611e
       -8.50341897e-01,  1.34099377e+00, -1.14039221e+00,  5.28130924e
        2.49837423e-01, -2.14629036e+00, -1.52145782e+00,  2.86707275e
        7.91444374e-01, -8.36707708e-01,  1.04711198e+00, -8.57818209e
        1.11244269e+00, -2.11796980e+00, -6.61111292e-01,  5.17357032e
        1.42448544e-01,  1.87412007e-01,  8.80707392e-01, -1.13485277e
        1.51544955e+00,  3.93089423e-01, -5.63430786e-01, -8.01267501e
       -4.08295533e-02,  6.36941340e-01, -1.67385314e-01,  8.02423013e
        3.43793723e-01, -7.49781844e-01, -5.81676009e-01,  7.48587459e
        5.88625123e-01,  4.34775061e-01,  2.25290347e+00,  8.51823326e
       -3.67542579e-01,  6.70560676e-01, -9.44672295e-02,  1.33056504e
        5.84425545e-02,  5.39200082e-01,  1.58879831e+00,  1.31066727e
        3.03346603e-01, -3.97646320e-01,  8.21319157e-02,  5.83900171e
       -1.57362669e+00,  1.32761756e+00,  3.71470857e-01, -8.91667836e
       -3.21728063e-01, -8.62704540e-01, -7.17238965e-01, -7.63628920e
       -2.31860635e-01, -2.09105854e-01,  3.72500967e-01, -1.76212769e
       -3.16541828e-01, -1.02299711e+00, -2.32499886e+00, -1.16664159e
       -7.02919394e-01, -7.63447017e-01,  2.67976287e-02,  7.26530627e
       -4.02580858e-01,  6.53880633e-01,  1.31465495e+00, -2.01441260e
       -7.18731884e-01,  2.79245162e-01,  6.94515454e-01,  6.70192613e
       -1.56043128e+00,  2.04565825e+00, -1.10809372e+00, -1.28910219e
```

```
-1.21534419e+00,  9.80842856e-02,  1.03477400e+00,  6.60193399e
 5.15758633e-01,  1.13859674e+00,  1.60267443e-01,  1.31719324e
-4.98674375e-01,  5.48765523e-01,  7.64093296e-01,  9.24973154e
 8.22545336e-01, -1.92732700e-01, -1.29756203e+00, -2.19086079e
-2.74651156e-01,  1.97568844e-01, -7.56313016e-01,  4.22033787e
 5.32942842e-01,  4.43730346e-01, -6.15907953e-01,  6.96485037e
-9.88428865e-01,  1.87830557e+00,  4.39573737e-01, -6.57048036e
 6.98227303e-01, -3.00298375e-01, -9.10082637e-01,  2.15419478e
-8.39125861e-02,  1.18068639e+00,  3.78827106e-01,  1.63994800e
 1.83450534e+00, -7.58324307e-02,  2.13438696e+00, -3.85604089e
-5.55225625e-01, -8.72179058e-01,  5.31266830e-01,  1.33387567e
 6.35970901e-01, -1.86271390e+00,  6.87213880e-01,  2.02923805e
 9.23333534e-02, -3.54213198e-02, -1.38184246e+00,  3.11398280e
 1.86062622e-02,  1.65531099e+00,  1.37756379e+00, -1.54891625e
 1.97689091e+00,  5.38525037e-01,  2.52672992e-01,  1.08677995e
 1.26511275e+00, -3.79059117e-01, -1.80495651e-01,  2.88502516e
 2.15941651e-01, -1.72774730e+00,  6.47804562e-01,  1.18701888e
 1.87938683e+00,  5.74387616e-01, -8.88974919e-01, -9.75103461e
-4.23343073e-01,  2.95528019e-01,  1.90424210e-01,  5.57764435e
-1.09796661e+00, -6.87934189e-01, -2.53639626e+00,  1.05635667e
 4.70561072e-01,  3.07988066e+00,  3.24592473e-02,  2.22908982e
-2.06186283e+00,  2.50715257e-01,  3.07328780e-01,  7.24388647e
 6.59437723e-01, -1.33151726e-01, -7.25377144e-01, -8.24386010e
-5.50999089e-01,  1.60475551e+00, -2.47934998e-01,  9.99705915e
 5.48585079e-01,  2.52641330e-01,  1.34603820e+00, -4.47396332e
 5.55362287e-01,  9.96127446e-01,  8.35639373e-01, -3.40962549e
 7.43778156e-01,  4.11647290e-01,  3.73336107e-01, -1.76224071e
 4.88802945e-01,  4.58047940e-01,  1.20104154e-01,  7.47813451e
 1.54254044e+00, -5.49480026e-01,  5.01025643e-01, -3.77305850e
 1.54489426e+00, -3.07825464e-01,  3.72918187e-02,  1.05612427e
 8.16009849e-01,  4.90971404e-01,  1.26880294e+00,  1.11456632e
```

```
1 d=sns.load_dataset('diamonds')
2 d
```

	carat	cut	color	clarity	depth	table	price	x	y	
0	0.23	Ideal	E	SI2	61.5	55.0	326	3.95	3.98	2.4
1	0.21	Premium	E	SI1	59.8	61.0	326	3.89	3.84	2.3
2	0.23	Good	E	VS1	56.9	65.0	327	4.05	4.07	2.3
3	0.29	Premium	I	VS2	62.4	58.0	334	4.20	4.23	2.6
4	0.31	Good	J	SI2	63.3	58.0	335	4.34	4.35	2.7
...	...	...	...	...	...	...	...	...	...	...
53935	0.72	Ideal	D	SI1	60.8	57.0	2757	5.75	5.76	3.5
53936	0.72	Good	D	SI1	63.1	55.0	2757	5.69	5.75	3.6
53937	0.70	Very Good	D	SI1	62.8	60.0	2757	5.66	5.68	3.5
53938	0.86	Premium	H	SI2	61.0	58.0	2757	6.15	6.12	3.7
53939	0.75	Ideal	D	SI2	62.2	55.0	2757	5.83	5.87	3.6

53940 rows × 10 columns

```
p=sns.load_dataset('penguins')
p
```

	species	island	bill_length_mm	bill_depth_mm	flipper_length_mm
0	Adelie	Torgersen	39.1	18.7	181.0
1	Adelie	Torgersen	39.5	17.4	186.0
2	Adelie	Torgersen	40.3	18.0	195.0
3	Adelie	Torgersen	NaN	NaN	NaN
4	Adelie	Torgersen	36.7	19.3	193.0
...	...	...	...	...	...
339	Gentoo	Biscoe	NaN	NaN	NaN
340	Gentoo	Biscoe	46.8	14.3	215.0
341	Gentoo	Biscoe	50.4	15.7	222.0
342	Gentoo	Biscoe	45.2	14.8	212.0
343	Gentoo	Biscoe	49.9	16.1	213.0

344 rows × 7 columns

```
1 tips=sns.load_dataset('tips')
2 tips
```

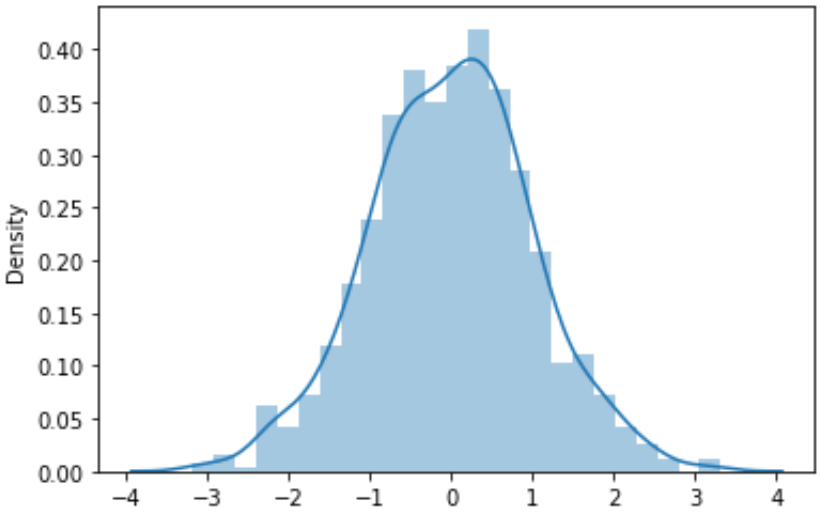
	total_bill	tip	sex	smoker	day	time	size
0	16.99	1.01	Female	No	Sun	Dinner	2
1	10.34	1.66	Male	No	Sun	Dinner	3
2	21.01	3.50	Male	No	Sun	Dinner	3
3	23.68	3.31	Male	No	Sun	Dinner	2
4	24.59	3.61	Female	No	Sun	Dinner	4
...	...	...	...	...	...	...	...
239	29.03	5.92	Male	No	Sat	Dinner	3
240	27.18	2.00	Female	Yes	Sat	Dinner	2
241	22.67	2.00	Male	Yes	Sat	Dinner	2
242	17.82	1.75	Male	No	Sat	Dinner	2
243	18.78	3.00	Female	No	Thur	Dinner	2

244 rows × 7 columns

dist plot

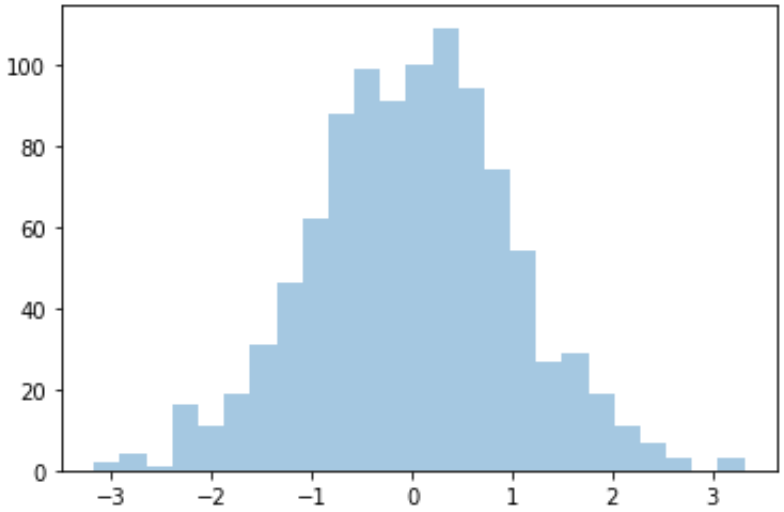
```
1 sns.distplot(x)
```

```
/usr/local/lib/python3.7/dist-packages/seaborn/distributions.py:2619:
  warnings.warn(msg, FutureWarning)
<matplotlib.axes._subplots.AxesSubplot at 0x7f4893d56890>
```



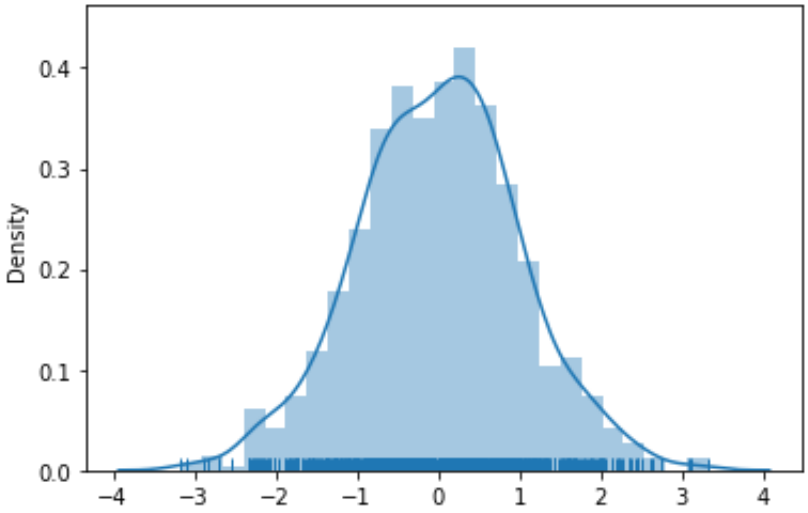

```
1 sns.distplot(x,kde=False)
```

```
/usr/local/lib/python3.7/dist-packages/seaborn/distributions.py:2619:
  warnings.warn(msg, FutureWarning)
<matplotlib.axes._subplots.AxesSubplot at 0x7f4893cde2d0>
```



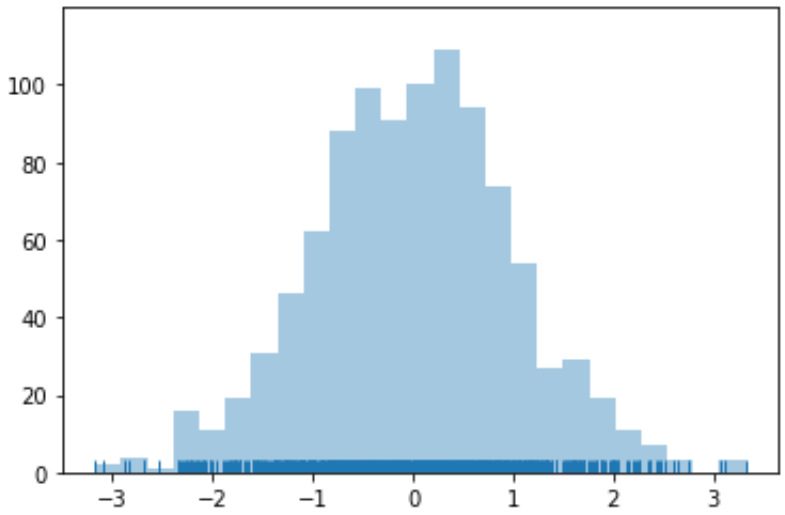
```
1 sns.distplot(x,rug=True) #observe rug below ,
```

```
/usr/local/lib/python3.7/dist-packages/seaborn/distributions.py:2619:
  warnings.warn(msg, FutureWarning)
/usr/local/lib/python3.7/dist-packages/seaborn/distributions.py:2103:
  warnings.warn(msg, FutureWarning)
<matplotlib.axes._subplots.AxesSubplot at 0x7f4893b68c90>
```



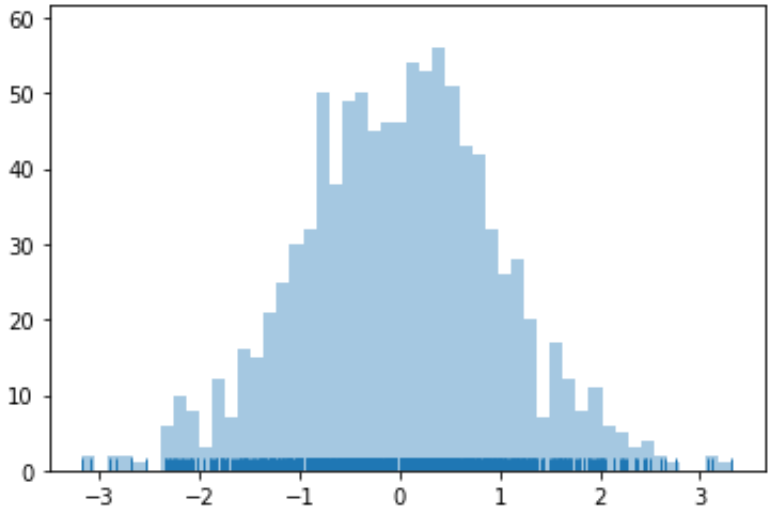

```
1 sns.distplot(x,kde=False,rug=True)
```

```
/usr/local/lib/python3.7/dist-packages/seaborn/distributions.py:2619:
  warnings.warn(msg, FutureWarning)
/usr/local/lib/python3.7/dist-packages/seaborn/distributions.py:2103:
  warnings.warn(msg, FutureWarning)
<matplotlib.axes._subplots.AxesSubplot at 0x7f4893a461d0>
```



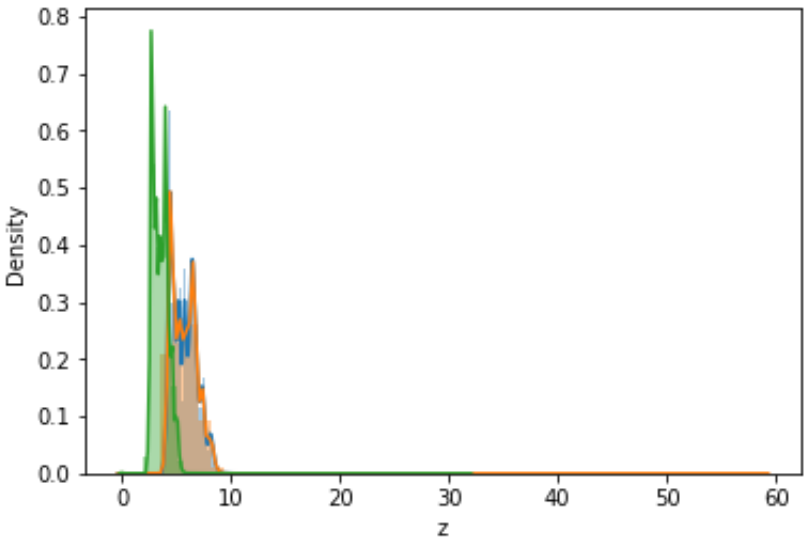
```
1 sns.distplot(x,kde=False,rug=True,bins=50) #bins
```

```
/usr/local/lib/python3.7/dist-packages/seaborn/distributions.py:2619:
  warnings.warn(msg, FutureWarning)
/usr/local/lib/python3.7/dist-packages/seaborn/distributions.py:2103:
  warnings.warn(msg, FutureWarning)
<matplotlib.axes._subplots.AxesSubplot at 0x7f48939e7c50>
```



```
1 sns.distplot(d.x,kde=True )
2 sns.distplot(d.y,kde=True )
3 sns.distplot(d.z,kde=True )
```

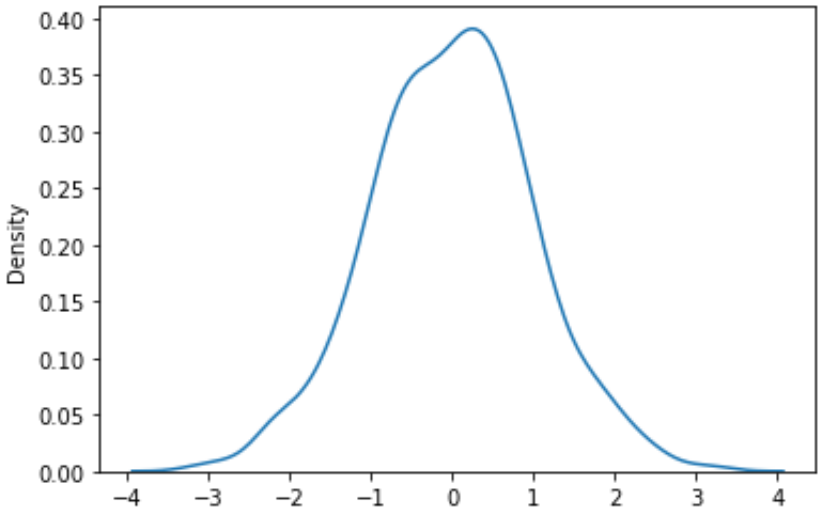
```
/usr/local/lib/python3.7/dist-packages/seaborn/distributions.py:2619:
  warnings.warn(msg, FutureWarning)
/usr/local/lib/python3.7/dist-packages/seaborn/distributions.py:2619:
  warnings.warn(msg, FutureWarning)
/usr/local/lib/python3.7/dist-packages/seaborn/distributions.py:2619:
  warnings.warn(msg, FutureWarning)
<matplotlib.axes._subplots.AxesSubplot at 0x7f489378ced0>
```



kde plot

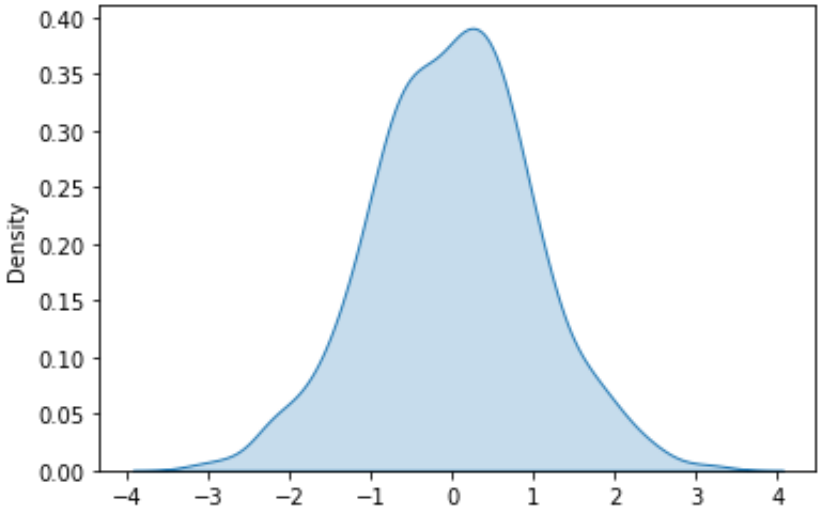
```
1 sns.kdeplot(x) #one line plot , no shade
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7f489357a650>
```




```
1 sns.kdeplot(x,shade=True) # under curve
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7f48934e7390>
```



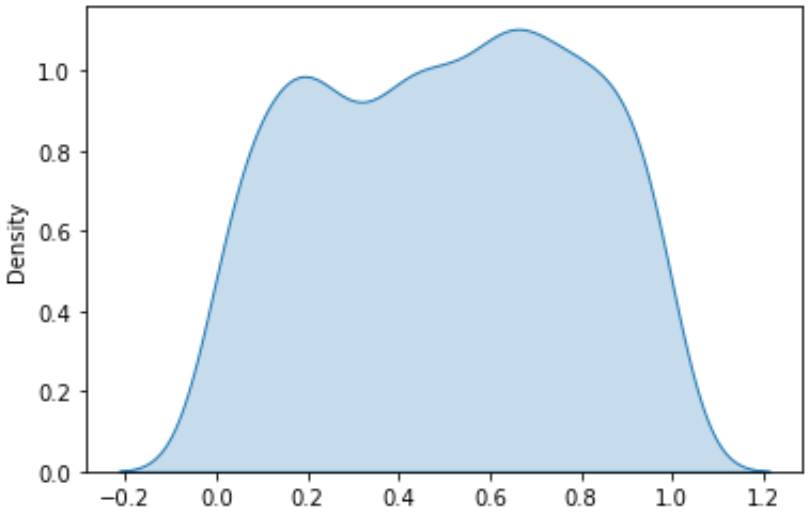

```
1 #super impose 2 plots together
2 y = np.random.uniform(size=1000)
3 y
```

```
array([0.00840268, 0.58176484, 0.10238233, 0.94945939, 0.32809189,
       0.49596742, 0.67333034, 0.96931903, 0.78235968, 0.22133178,
       0.36040894, 0.81099157, 0.93950458, 0.09039733, 0.74767172,
       0.14096118, 0.15405408, 0.67436036, 0.02613345, 0.15962105,
       0.64711397, 0.04259454, 0.8706145 , 0.2148466 , 0.10386919,
       0.8800367 , 0.43343868, 0.82882035, 0.48373386, 0.34746291,
       0.02639855, 0.44540638, 0.42984157, 0.17994989, 0.06627569,
       0.05402459, 0.70466553, 0.40864384, 0.68926721, 0.62455071,
       0.69540137, 0.3480405 , 0.24472847, 0.31754446, 0.77712553,
```

0.55965647, 0.94810329, 0.74658972, 0.183611 , 0.08733976,
0.29598083, 0.05665539, 0.14042813, 0.49993411, 0.25418364,
0.19821956, 0.01673812, 0.85032665, 0.03193886, 0.11381713,
0.54470616, 0.58067757, 0.22163509, 0.30184363, 0.81383319,
0.05255939, 0.98649265, 0.07256956, 0.61036155, 0.77449936,
0.30857029, 0.46970121, 0.89773675, 0.74708324, 0.16002394,
0.00312329, 0.73606517, 0.61164481, 0.01758414, 0.5518744 ,
0.18950027, 0.83892289, 0.75475088, 0.92763448, 0.65651529,
0.67673514, 0.16500265, 0.29078776, 0.78249804, 0.31326452,
0.8356492 , 0.17503118, 0.82045761, 0.40478116, 0.79484904,
0.42226557, 0.72135063, 0.13682856, 0.11819618, 0.98492059,
0.64287736, 0.62421466, 0.51272692, 0.680495 , 0.28829885,
0.16457608, 0.75164457, 0.20076538, 0.30782785, 0.44029025,
0.73364489, 0.03987217, 0.23150652, 0.97155785, 0.54212657,
0.51431937, 0.85437578, 0.54285335, 0.84610463, 0.59959768,
0.88380048, 0.4820623 , 0.49715978, 0.47122263, 0.79140553,
0.65385141, 0.15426023, 0.72050225, 0.43682983, 0.64959324,
0.71353027, 0.77182205, 0.07543548, 0.65511965, 0.21119982,
0.95718803, 0.81404204, 0.84772028, 0.30720839, 0.92352039,
0.29659462, 0.22750668, 0.92072928, 0.89895241, 0.37007265,
0.70197895, 0.36413746, 0.59721617, 0.36393157, 0.51599688,
0.84566894, 0.3302247 , 0.25772485, 0.60033633, 0.30182308,
0.65757186, 0.98062622, 0.69513778, 0.58725485, 0.84644984,
0.42528613, 0.84707373, 0.46826153, 0.87007444, 0.38697482,
0.14165184, 0.23870348, 0.4491918 , 0.8232365 , 0.94812056,
0.19956528, 0.69105683, 0.93154212, 0.04206353, 0.97808592,
0.87545928, 0.90232337, 0.38672908, 0.21451255, 0.40120163,
0.11488226, 0.64694549, 0.00876433, 0.51647787, 0.63676198,
0.59622484, 0.79315921, 0.4341729 , 0.26744734, 0.10766125,
0.001581 , 0.4152475 , 0.17384458, 0.27132848, 0.2025346 ,
0.06457355, 0.3438178 , 0.57191671, 0.15304913, 0.96510576,
0.0109799 , 0.6134347 , 0.50656839, 0.69405747, 0.68757674,
0.78479068, 0.50133832, 0.85812864, 0.80857515, 0.47030241,
0.22720258, 0.75062122, 0.82998996, 0.21718729, 0.65446608,
0.36810834, 0.33582894, 0.01847862, 0.6761971 , 0.8111689 ,
0.00611769, 0.8797679 , 0.94904033, 0.5606339 , 0.334467 ,
0.39596594, 0.60541563, 0.75220775, 0.04502316, 0.60663066,
0.88364851, 0.69402979, 0.5864431 , 0.46733218, 0.08731401,
0.88272379, 0.77565974, 0.93318822, 0.09025881, 0.6711778 ,
0.7395186 , 0.50224249, 0.77943095, 0.87021754, 0.88694286,
0.46397966, 0.6577817 , 0.47952794, 0.56985798, 0.00683388,
0.77012952, 0.43705059, 0.96043089, 0.24644083, 0.6046624 ,
0.28186547, 0.22827768, 0.59999344, 0.02067736, 0.51747215,
0.00758557, 0.93706007, 0.99650593, 0.48773824, 0.84304579,
0.48233798, 0.52976987, 0.8790901 , 0.69408121, 0.79858549,
0.29525808, 0.33663569, 0.10156605, 0.32759253, 0.38626297,
0.57235459, 0.61007063, 0.4899008 , 0.73988976, 0.64325424,
0.14445631, 0.62981158, 0.2822574 , 0.76504559, 0.44997154,
0.72456119, 0.41024194, 0.58957504, 0.50600518, 0.03635525,
0.87405152, 0.9495048 , 0.12663276, 0.34890542, 0.64441668,
0.18030591, 0.63823829, 0.27273834, 0.68879041, 0.82444018,

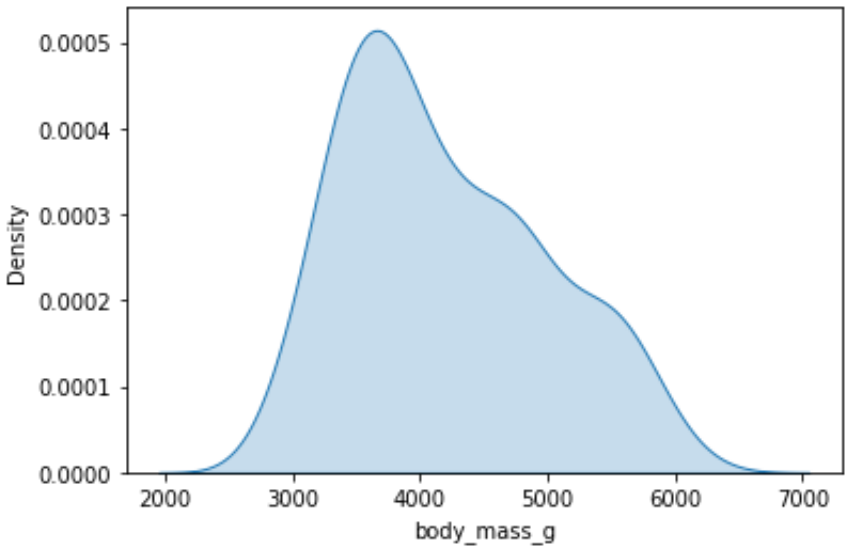
```
1 sns.kdeplot(y,shade=True)
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7f489347ef10>
```



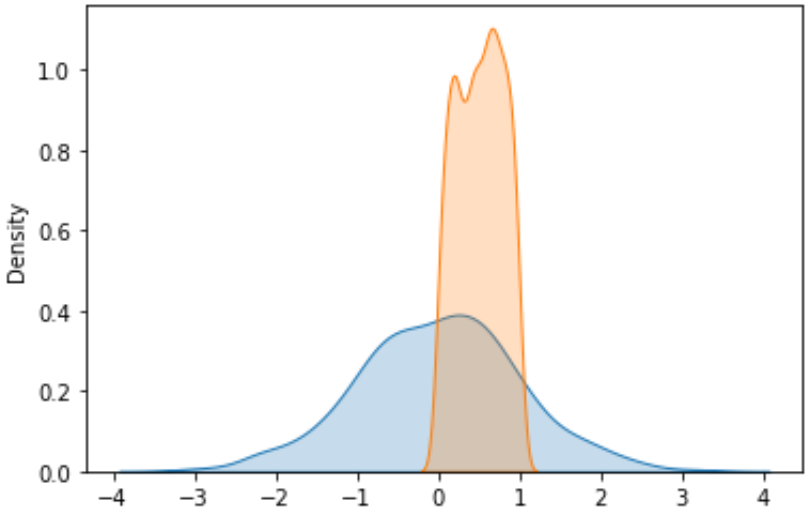

```
1 sns.kdeplot(data=p,x='body_mass_g',shade=True)
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7f4893411e50>
```



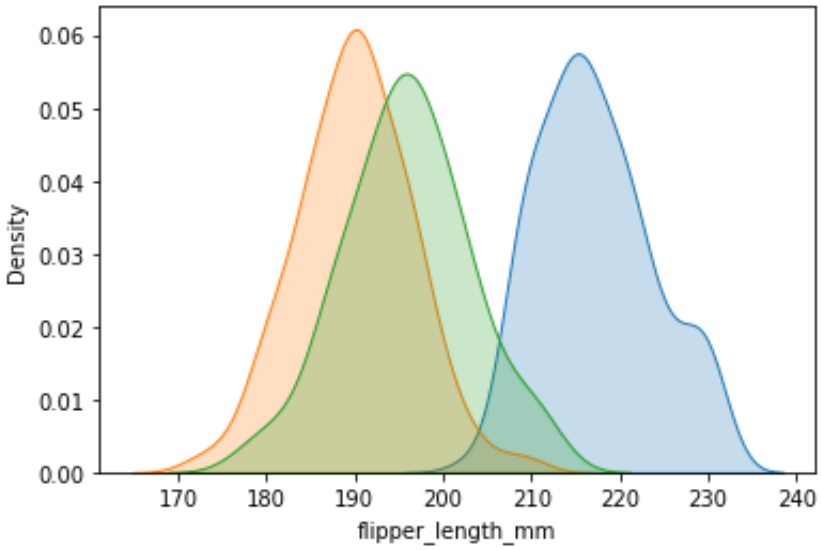
```
1 sns.kdeplot(x,shade=True)
2 sns.kdeplot(y,shade=True)
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7f4893379ad0>
```



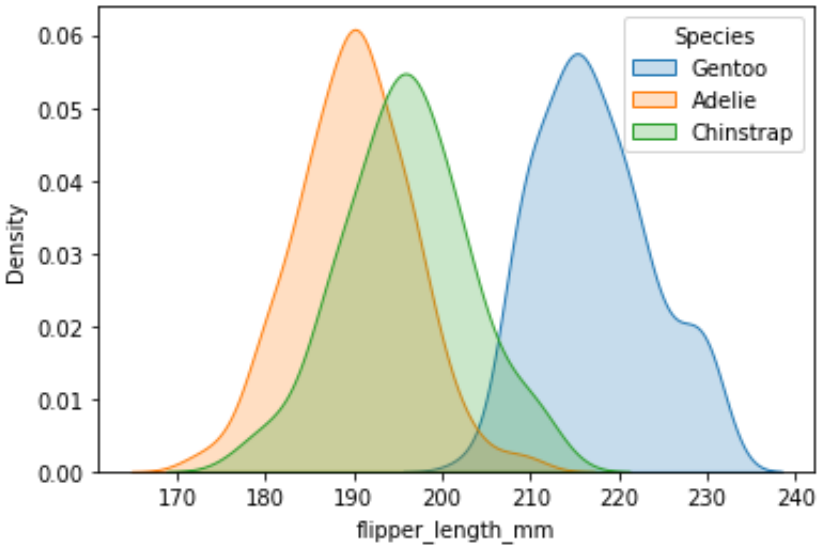

```
1 sns.kdeplot(p[p['species']=='Gentoo'].flipper_length_mm,shade=1)
2 sns.kdeplot(p[p['species']=='Adelie'].flipper_length_mm,shade=1)
3 sns.kdeplot(p[p['species']=='Chinstrap'].flipper_length_mm,shade=1)
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7f4893306290>
```



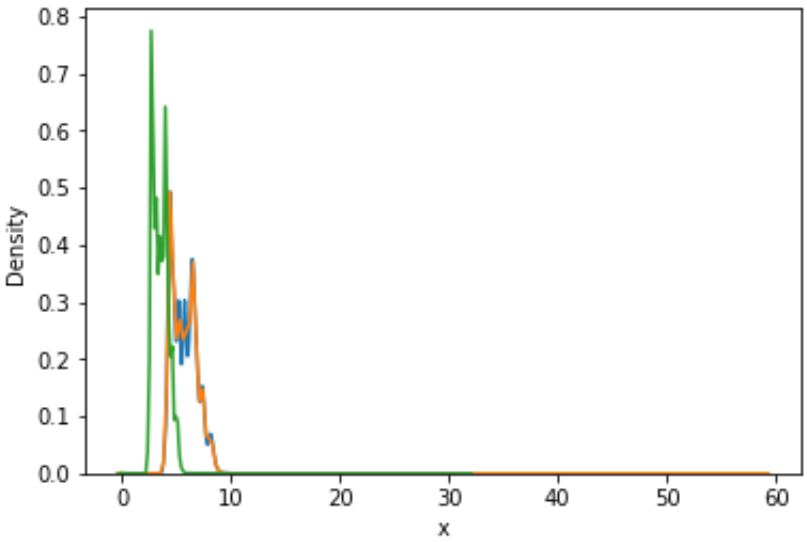
```
sns.kdeplot(p[p['species']=='Gentoo'].flipper_length_mm,shade=1)
sns.kdeplot(p[p['species']=='Adelie'].flipper_length_mm,shade=1)
sns.kdeplot(p[p['species']=='Chinstrap'].flipper_length_mm,shade=1)
plt.legend(title='Species',labels=['Gentoo','Adelie','Chinstrap'])
```

```
<matplotlib.legend.Legend at 0x7f48932d1b90>
```




```
sns.kdeplot(d.x) #blue
sns.kdeplot(d.y) #orange
sns.kdeplot(d.z) #green
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7f489345ccd0>
```




```
x
```

```
array([-1.38322217e+00, -2.46246099e-01,  1.24474920e+00, -2.24595390e
        1.30052342e-01, -8.03235432e-02,  8.59395086e-01, -1.24348778e
       -2.50069691e-01, -8.10033648e-01, -2.97807590e-01, -6.99263630e
       -5.03685207e-01, -2.09559050e+00, -3.08714230e+00,  9.93640061e
```

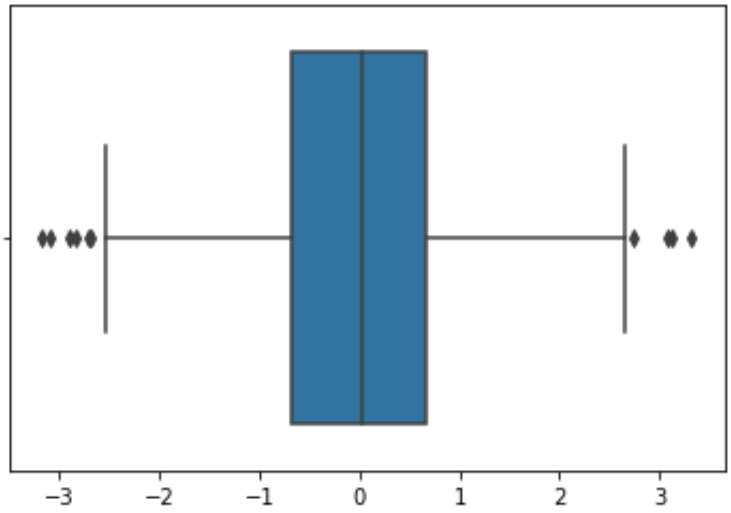
```
-5.32295272e-01,  5.58544707e-01,  2.64039843e+00, -1.00989328e
-4.97854388e-01,  1.89721976e+00, -6.17583492e-01, -9.79834694e
 2.86391322e-01,  6.36834673e-01, -2.18321720e-01, -2.83539300e
-1.26256355e+00, -1.21413881e+00, -6.09740415e-01,  1.58720521e
-9.02242792e-01, -4.67485689e-01,  7.56694539e-02,  3.49724611e
-8.50341897e-01,  1.34099377e+00, -1.14039221e+00,  5.28130924e
 2.49837423e-01, -2.14629036e+00, -1.52145782e+00,  2.86707275e
 7.91444374e-01, -8.36707708e-01,  1.04711198e+00, -8.57818209e
 1.11244269e+00, -2.11796980e+00, -6.61111292e-01,  5.17357032e
 1.42448544e-01,  1.87412007e-01,  8.80707392e-01, -1.13485277e
 1.51544955e+00,  3.93089423e-01, -5.63430786e-01, -8.01267501e
-4.08295533e-02,  6.36941340e-01, -1.67385314e-01,  8.02423013e
 3.43793723e-01, -7.49781844e-01, -5.81676009e-01,  7.48587459e
 5.88625123e-01,  4.34775061e-01,  2.25290347e+00,  8.51823326e
-3.67542579e-01,  6.70560676e-01, -9.44672295e-02,  1.33056504e
 5.84425545e-02,  5.39200082e-01,  1.58879831e+00,  1.31066727e
 3.03346603e-01, -3.97646320e-01,  8.21319157e-02,  5.83900171e
-1.57362669e+00,  1.32761756e+00,  3.71470857e-01, -8.91667836e
-3.21728063e-01, -8.62704540e-01, -7.17238965e-01, -7.63628920e
-2.31860635e-01, -2.09105854e-01,  3.72500967e-01, -1.76212769e
-3.16541828e-01, -1.02299711e+00, -2.32499886e+00, -1.16664159e
-7.02919394e-01, -7.63447017e-01,  2.67976287e-02,  7.26530627e
-4.02580858e-01,  6.53880633e-01,  1.31465495e+00, -2.01441260e
-7.18731884e-01,  2.79245162e-01,  6.94515454e-01,  6.70192613e
-1.56043128e+00,  2.04565825e+00, -1.10809372e+00, -1.28910219e
-1.21534419e+00,  9.80842856e-02,  1.03477400e+00,  6.60193399e
 5.15758633e-01,  1.13859674e+00,  1.60267443e-01,  1.31719324e
-4.98674375e-01,  5.48765523e-01,  7.64093296e-01,  9.24973154e
 8.22545336e-01, -1.92732700e-01, -1.29756203e+00, -2.19086079e
-2.74651156e-01,  1.97568844e-01, -7.56313016e-01,  4.22033787e
 5.32942842e-01,  4.43730346e-01, -6.15907953e-01,  6.96485037e
-9.88428865e-01,  1.87830557e+00,  4.39573737e-01, -6.57048036e
 6.98227303e-01, -3.00298375e-01, -9.10082637e-01,  2.15419478e
-8.39125861e-02,  1.18068639e+00,  3.78827106e-01,  1.63994800e
 1.83450534e+00, -7.58324307e-02,  2.13438696e+00, -3.85604089e
-5.55225625e-01, -8.72179058e-01,  5.31266830e-01,  1.33387567e
 6.35970901e-01, -1.86271390e+00,  6.87213880e-01,  2.02923805e
 9.23333534e-02, -3.54213198e-02, -1.38184246e+00,  3.11398280e
 1.86062622e-02,  1.65531099e+00,  1.37756379e+00, -1.54891625e
 1.97689091e+00,  5.38525037e-01,  2.52672992e-01,  1.08677995e
 1.26511275e+00, -3.79059117e-01, -1.80495651e-01,  2.88502516e
 2.15941651e-01, -1.72774730e+00,  6.47804562e-01,  1.18701888e
 1.87938683e+00,  5.74387616e-01, -8.88974919e-01, -9.75103461e
-4.23343073e-01,  2.95528019e-01,  1.90424210e-01,  5.57764435e
-1.09796661e+00, -6.87934189e-01, -2.53639626e+00,  1.05635667e
 4.70561072e-01,  3.07988066e+00,  3.24592473e-02,  2.22908982e
-2.06186283e+00,  2.50715257e-01,  3.07328780e-01,  7.24388647e
 6.59437723e-01, -1.33151726e-01, -7.25377144e-01, -8.24386010e
-5.50999089e-01,  1.60475551e+00, -2.47934998e-01,  9.99705915e
 5.48585079e-01,  2.52641330e-01,  1.34603820e+00, -4.47396332e
 5.55362287e-01,  9.96127446e-01,  8.35639373e-01, -3.40962549e
 7.43778156e-01,  4.11647290e-01,  3.73336107e-01, -1.76224071e
 4.88802945e-01,  4.58047940e-01,  1.20104154e-01,  7.47813451e
 1.54254044e+00,  5.49480026e-01,  5.01025643e-01,  2.77205850e
```

```
1.54254044e+00, -5.49480026e-01,  5.01025643e-01, -3.77305850e
1.54489426e+00, -3.07825464e-01,  3.72918187e-02,  1.05612427e
8.16009849e-01,  4.99071404e-01,  1.26880294e+00,  1.11456632e
```

box plot

```
1 sns.boxplot(x)
```

```
/usr/local/lib/python3.7/dist-packages/seaborn/_decorators.py:43: Futu
  FutureWarning
<matplotlib.axes._subplots.AxesSubplot at 0x7f48931e0990>
```



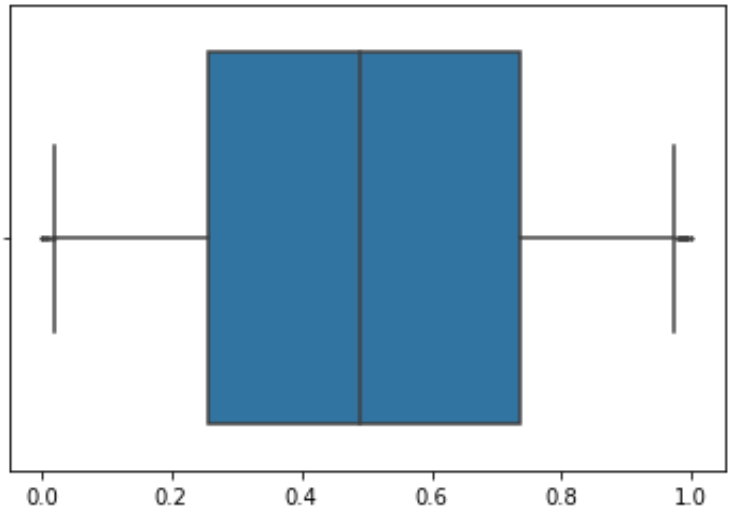

```
1 x=np.random.uniform(size=1000)
2 x
```

```
array([9.80668056e-01, 8.61888755e-01, 1.17160775e-01, 4.37125655e-01,
       8.00620243e-01, 3.04943621e-01, 7.04662125e-01, 1.99501193e-02,
       4.51544705e-01, 2.62909924e-01, 2.61713270e-02, 1.63902763e-01,
       2.31130989e-01, 9.10605587e-01, 4.58327054e-02, 4.85820503e-01,
       1.92725021e-02, 2.47979204e-01, 4.41756086e-01, 3.11357137e-01,
       3.48227629e-01, 5.66741345e-02, 5.33569478e-01, 9.29382408e-01,
       1.81261252e-02, 1.52158891e-02, 8.11125843e-03, 7.02058381e-01,
       6.49185729e-02, 2.90982923e-01, 9.87281867e-01, 7.29455691e-02,
       5.58081585e-01, 4.54787220e-01, 8.22631739e-02, 8.03154032e-01,
       1.24151770e-01, 9.12208093e-02, 3.89302623e-01, 1.09379256e-01,
       6.58372917e-01, 8.75570566e-01, 9.00952042e-01, 5.30778563e-01,
       5.76174311e-01, 5.06960383e-01, 5.04569855e-01, 2.13226097e-01,
       1.74375684e-01, 6.98105797e-01, 1.82155613e-01, 1.68838177e-01,
       9.88638428e-01, 1.95415965e-01, 7.97616028e-01, 2.55348593e-01,
       3.12033936e-01, 3.79720634e-01, 8.76681819e-01, 5.12404406e-01,
       2.53934668e-01, 6.50893537e-01, 2.73502105e-01, 9.36220485e-02,
       7.12205245e-01, 9.41063381e-01, 9.53909959e-01, 8.99620286e-01,
       6.21908911e-01, 8.30458654e-01, 3.19172310e-02, 8.23702912e-01,
       4.85967363e-01, 6.42310219e-01, 4.47347164e-01, 6.75438181e-01,
```

```
7.41680421e-01, 2.41313540e-01, 2.46471435e-01, 4.28758175e-01,
6.98601219e-02, 7.36374730e-01, 6.10655221e-01, 6.11143312e-01,
7.91413711e-01, 8.53918878e-01, 2.79666980e-01, 9.43745702e-01,
9.69807741e-01, 5.78407545e-01, 9.72812495e-01, 6.04319433e-01,
4.26540739e-01, 3.08097839e-01, 6.66614407e-01, 8.51412287e-02,
5.13485640e-01, 5.09106299e-01, 7.38277408e-01, 4.06763938e-01,
9.14029149e-02, 5.55121251e-01, 6.56834569e-01, 7.68004368e-01,
2.50354194e-01, 4.27663923e-01, 5.80538314e-01, 1.89043976e-01,
2.15788826e-01, 4.36510742e-01, 8.25333749e-01, 5.31394883e-01,
4.76335624e-01, 6.11428523e-01, 7.08438781e-01, 5.83126355e-01,
3.22220979e-01, 3.51450332e-01, 1.55033574e-01, 6.41830973e-01,
7.30458213e-01, 5.84930903e-01, 3.83848771e-02, 4.51520782e-01,
6.16264240e-01, 2.17978541e-01, 6.19354255e-01, 9.88622287e-01,
6.27007546e-03, 4.14814792e-01, 7.06205527e-01, 4.03256796e-01,
7.13499497e-01, 3.28590903e-02, 8.45227938e-02, 6.94455109e-02,
2.57906027e-01, 8.21334423e-02, 3.08419078e-01, 9.76624468e-01,
7.19197768e-01, 2.02824049e-01, 6.41376667e-01, 3.03731394e-01,
8.05274202e-01, 6.72962897e-01, 7.53496360e-01, 5.95357236e-01,
1.83990582e-01, 1.83104532e-01, 4.38473286e-01, 6.95598336e-01,
9.91438805e-01, 6.03211525e-01, 6.67594996e-01, 6.73659338e-01,
1.25102461e-01, 4.38358831e-01, 9.82466299e-01, 7.94490758e-01,
6.35056196e-01, 8.87588078e-01, 1.13771609e-01, 9.06142906e-01,
3.64397614e-02, 9.59997820e-01, 8.67641245e-01, 3.82222061e-01,
3.92420765e-01, 4.92798565e-01, 6.07304096e-01, 5.01874823e-01,
5.24696865e-01, 3.63584828e-01, 1.46186937e-03, 3.76972746e-02,
3.67333788e-01, 1.85262630e-01, 8.01993567e-02, 1.92163559e-01,
2.43552750e-01, 1.40690627e-01, 2.09833391e-01, 6.93344444e-01,
6.00136829e-01, 7.33924122e-01, 6.62556878e-01, 6.72930663e-01,
6.28451902e-01, 8.95357954e-01, 1.21724037e-01, 4.32718893e-01,
6.28071293e-02, 8.27037743e-01, 4.29907784e-01, 6.98335599e-01,
1.94332230e-01, 8.10821300e-01, 7.12681684e-01, 3.89694510e-01,
2.59537100e-01, 5.43302964e-01, 9.38805127e-04, 7.15894346e-01,
7.73674741e-01, 3.76273237e-01, 4.02391402e-01, 3.53330540e-01,
5.59433159e-02, 7.65115038e-01, 2.24585160e-01, 1.59368640e-01,
4.00689821e-01, 4.73671287e-01, 8.67609555e-01, 2.73176318e-01,
3.83521925e-02, 5.25877066e-01, 2.78484632e-01, 1.79735349e-01,
3.29011893e-02, 9.01976581e-01, 3.35577729e-01, 4.30164975e-01,
5.68868545e-01, 7.49271709e-01, 9.18561853e-01, 6.95461934e-02,
2.32521422e-01, 7.39268645e-01, 6.80103241e-01, 8.60852076e-01,
5.40665550e-01, 9.62332903e-01, 8.54896271e-01, 9.59305866e-01,
7.11041527e-01, 4.88579227e-01, 1.86423077e-01, 2.47102328e-01,
```

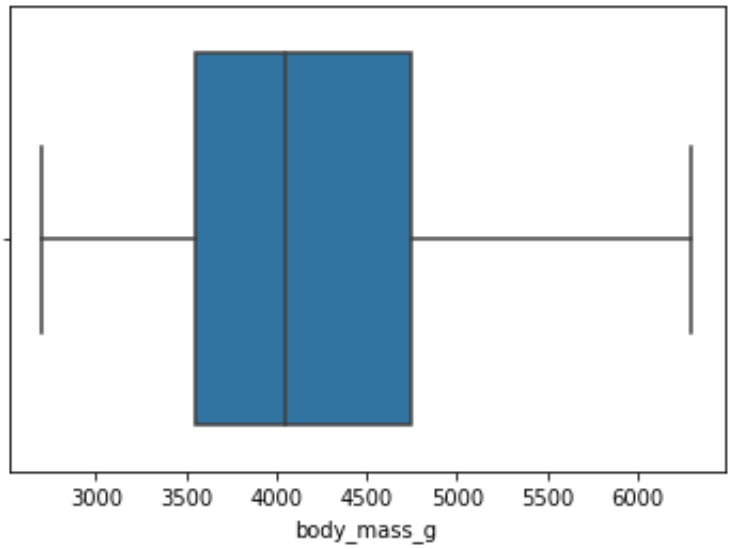
```
1 sns.boxplot(x,whis=0.5,fliersize=1,orient='v')
```

```
/usr/local/lib/python3.7/dist-packages/seaborn/_decorators.py:43: Futu
  FutureWarning
/usr/local/lib/python3.7/dist-packages/seaborn/_core.py:1326: UserWarn
  warnings.warn(single_var_warning.format("Vertical", "x"))
<matplotlib.axes._subplots.AxesSubplot at 0x7f48931d9790>
```




```
1 sns.boxplot(data=p,x='body_mass_g')
```

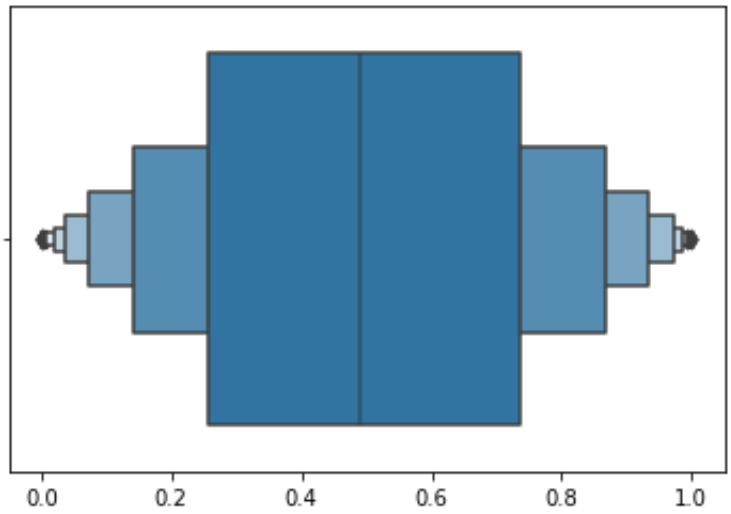
```
<matplotlib.axes._subplots.AxesSubplot at 0x7f489316bf50>
```



Boxen Plot

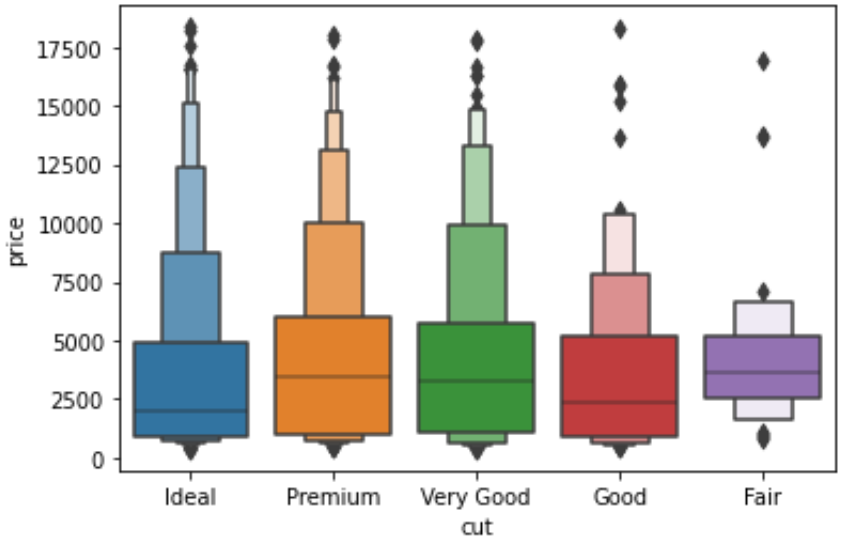
```
1 sns.boxenplot(x)
```

```
/usr/local/lib/python3.7/dist-packages/seaborn/_decorators.py:43: Futu
  FutureWarning
<matplotlib.axes._subplots.AxesSubplot at 0x7f4893256190>
```




```
1 sns.boxenplot(x = 'cut', y = 'price', data = d.sample(1000))
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7f489301af90>
```



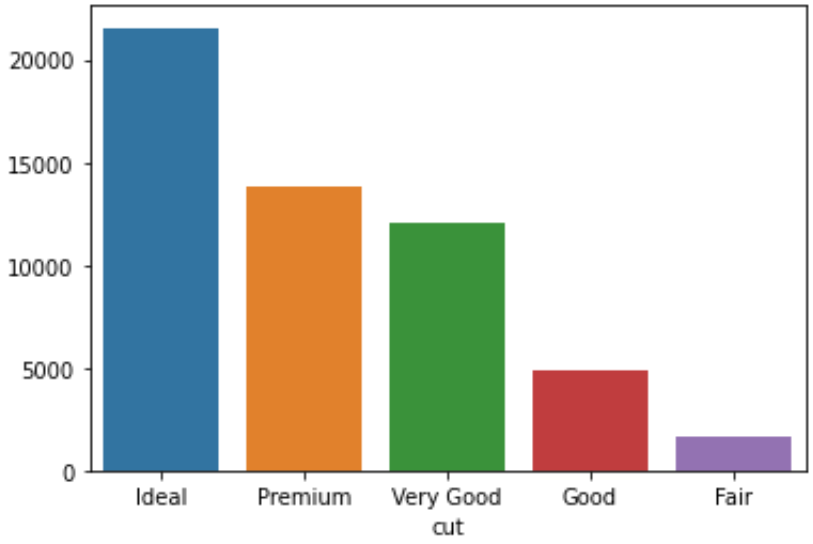
Bar Plot

```
1 c=d.groupby('cut')['cut'].count()
2 c
```

```
cut
Ideal        21551
Premium      13791
Very Good    12082
Good          4906
Fair          1610
Name: cut, dtype: int64
```

```
1 sns.barplot(x=c.index,y=c.values)
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7f4892f292d0>
```



Joint Plot

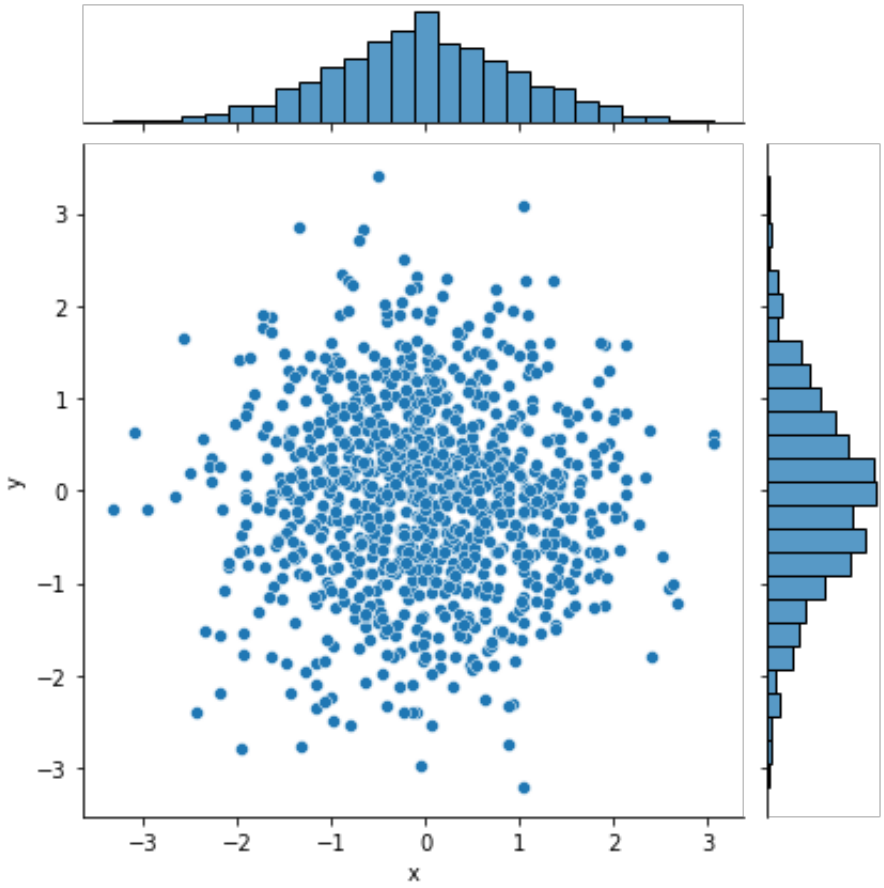
```
1 x=np.random.normal(size=1000)
2 y=np.random.normal(size=1000)
```

```
df=pd.DataFrame({'x': x, 'y':y})
df.head()
```

	x	y
0	0.060027	0.414960
1	-0.383031	-0.242303
2	0.446028	-0.547799
3	0.664419	1.026018
4	1.153170	-0.680517

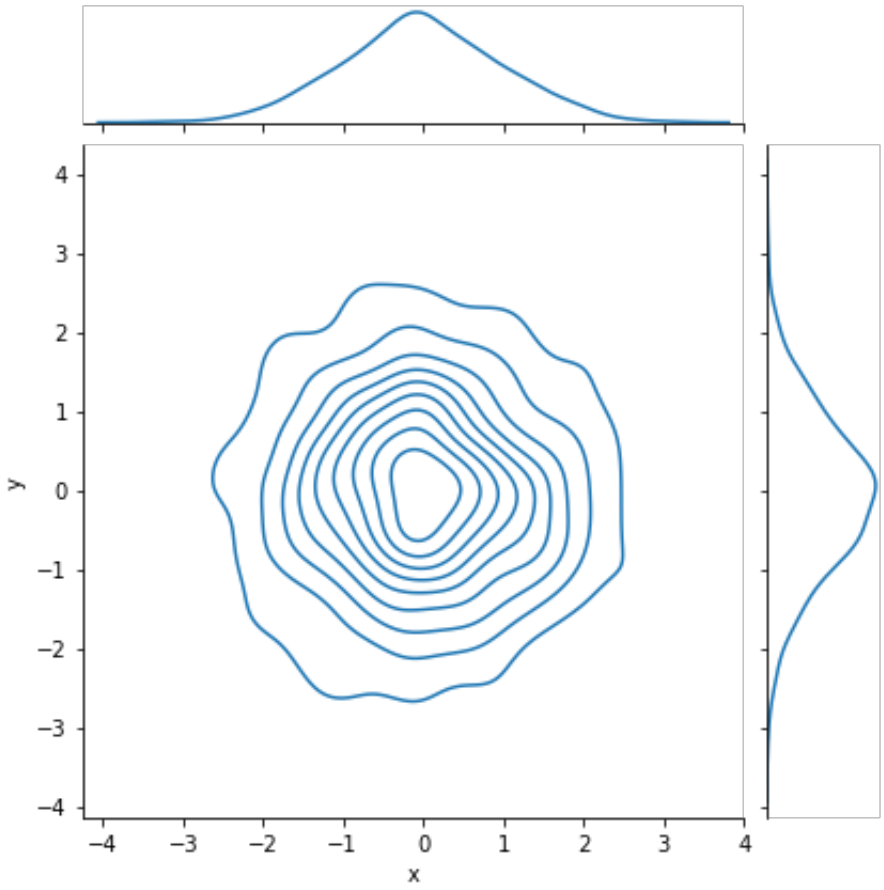
```
sns.jointplot(df.x,df.y)
```

```
/usr/local/lib/python3.7/dist-packages/seaborn/_decorators.py:43: Futu
  FutureWarning
<seaborn.axisgrid.JointGrid at 0x7f4893dba590>
```



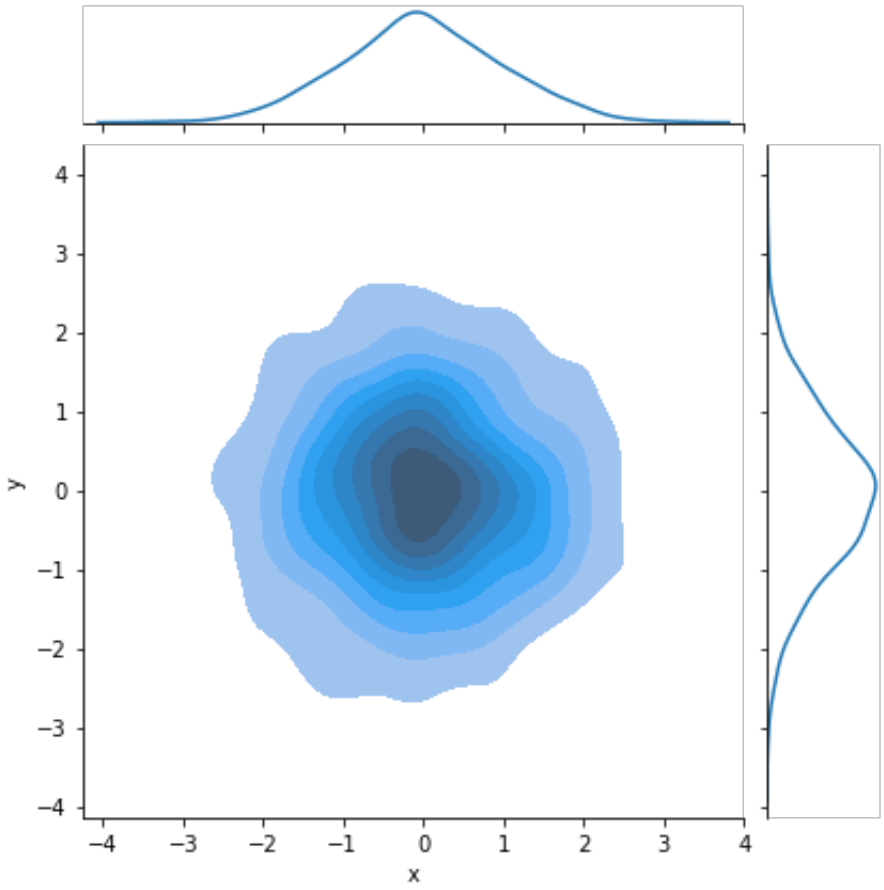
```
1 sns.jointplot('x','y',data=df,kind='kde',shade=False)
```

```
/usr/local/lib/python3.7/dist-packages/seaborn/_decorators.py:43: Futu
  FutureWarning
<seaborn.axisgrid.JointGrid at 0x7f4892d62e90>
```



```
1 sns.jointplot('x','y',data=df,kind='kde',shade=True)
```

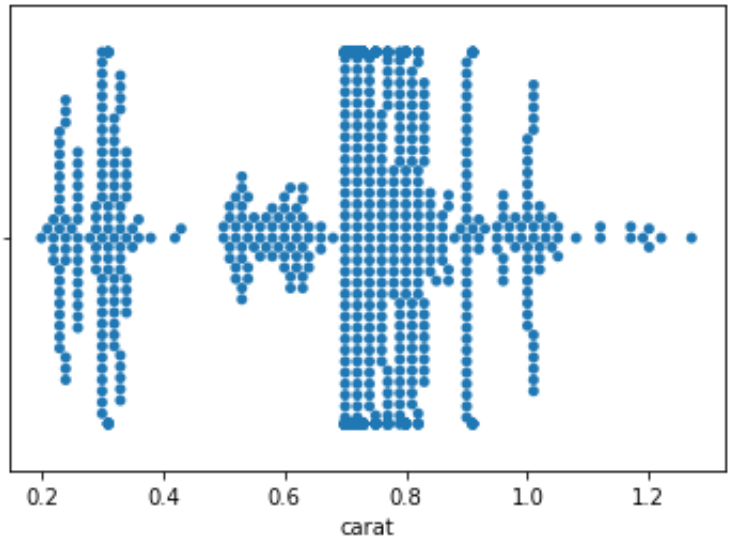
```
/usr/local/lib/python3.7/dist-packages/seaborn/_decorators.py:43: Futu
  FutureWarning
<seaborn.axisgrid.JointGrid at 0x7f4892c9e0d0>
```



Swarm Plot

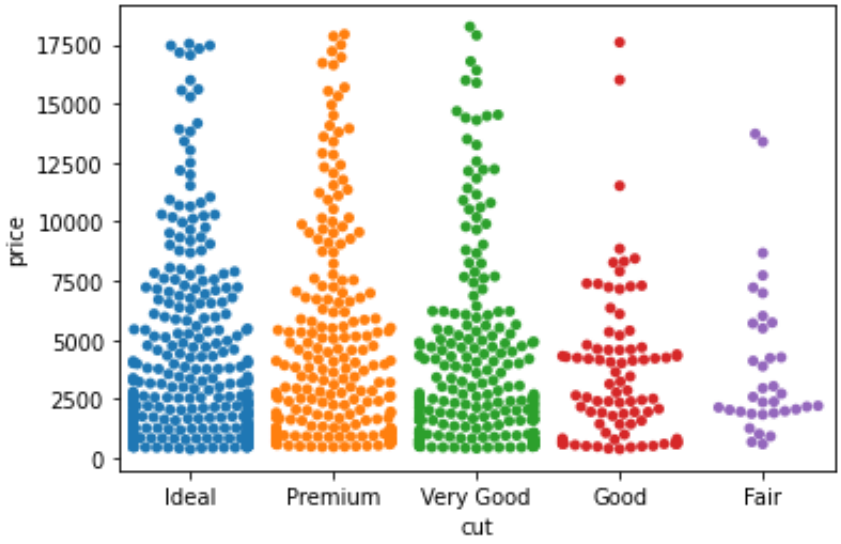
```
1 sns.swarmplot(d.head(1000).carat)
```

```
/usr/local/lib/python3.7/dist-packages/seaborn/_decorators.py:43: Futu
  FutureWarning
/usr/local/lib/python3.7/dist-packages/seaborn/categorical.py:1296: Us
  warnings.warn(msg, UserWarning)
<matplotlib.axes._subplots.AxesSubplot at 0x7f4892b15b50>
```



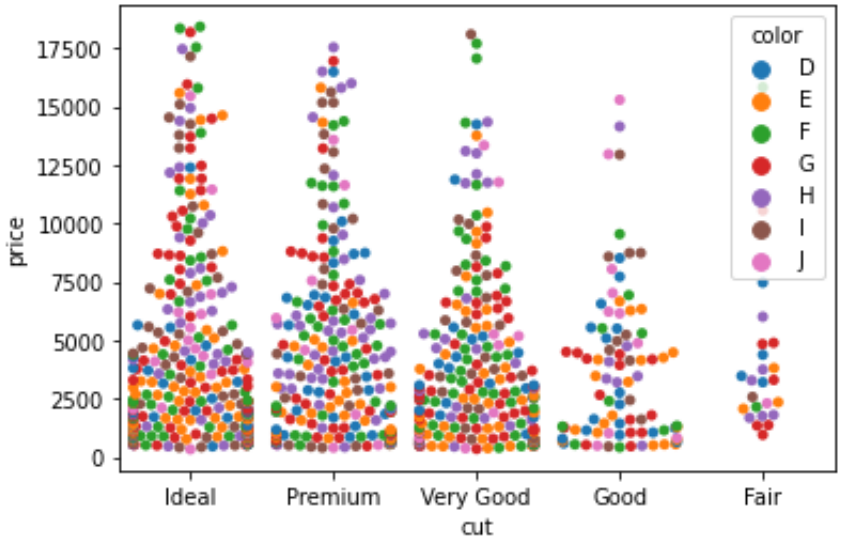
```
1 sns.swarmplot(data=d.sample(1000),x='cut',y='price')
```

```
/usr/local/lib/python3.7/dist-packages/seaborn/categorical.py:1296: Us
  warnings.warn(msg, UserWarning)
/usr/local/lib/python3.7/dist-packages/seaborn/categorical.py:1296: Us
  warnings.warn(msg, UserWarning)
/usr/local/lib/python3.7/dist-packages/seaborn/categorical.py:1296: Us
  warnings.warn(msg, UserWarning)
/usr/local/lib/python3.7/dist-packages/seaborn/categorical.py:1296: Us
  warnings.warn(msg, UserWarning)
<matplotlib.axes._subplots.AxesSubplot at 0x7f4892a3d690>
```



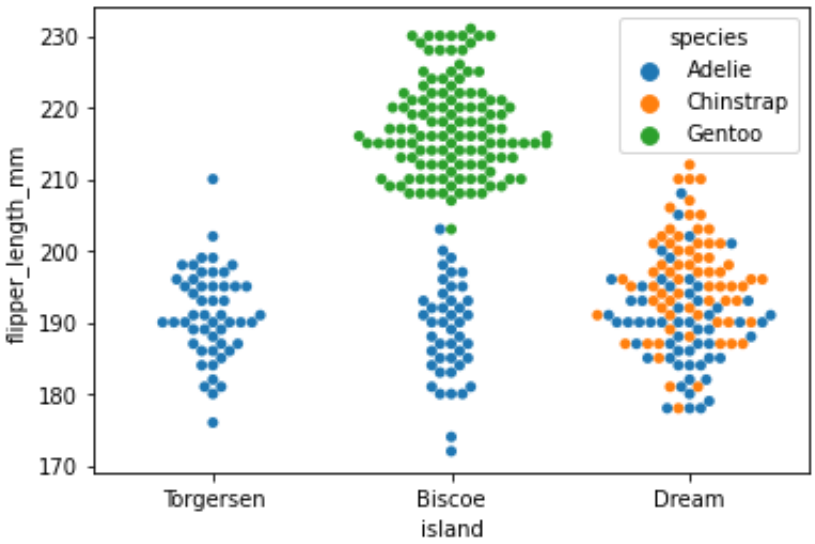
```
1 sns.swarmplot(data=d.sample(1000),x='cut',y='price',hue='color')
```

```
/usr/local/lib/python3.7/dist-packages/seaborn/categorical.py:1296: Us
  warnings.warn(msg, UserWarning)
/usr/local/lib/python3.7/dist-packages/seaborn/categorical.py:1296: Us
  warnings.warn(msg, UserWarning)
/usr/local/lib/python3.7/dist-packages/seaborn/categorical.py:1296: Us
  warnings.warn(msg, UserWarning)
/usr/local/lib/python3.7/dist-packages/seaborn/categorical.py:1296: Us
  warnings.warn(msg, UserWarning)
<matplotlib.axes._subplots.AxesSubplot at 0x7f48929b4cd0>
```



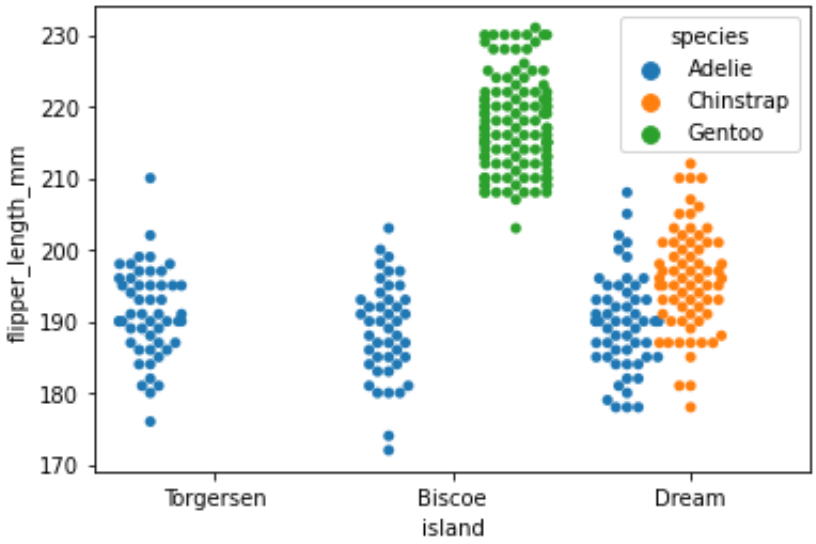

```
1 sns.swarmplot(data=p,x='island',y='flipper_length_mm',hue='species')
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7f4892951850>
```



```
1 sns.swarmplot(data=p,x='island',y='flipper_length_mm',hue='species',spli
```

```
/usr/local/lib/python3.7/dist-packages/seaborn/categorical.py:3002: Us
  warnings.warn(msg, UserWarning)
/usr/local/lib/python3.7/dist-packages/seaborn/categorical.py:1296: Us
  warnings.warn(msg, UserWarning)
/usr/local/lib/python3.7/dist-packages/seaborn/categorical.py:1296: Us
  warnings.warn(msg, UserWarning)
/usr/local/lib/python3.7/dist-packages/seaborn/categorical.py:1296: Us
  warnings.warn(msg, UserWarning)
/usr/local/lib/python3.7/dist-packages/seaborn/categorical.py:1296: Us
  warnings.warn(msg, UserWarning)
<matplotlib.axes._subplots.AxesSubplot at 0x7f48928bdb90>
```



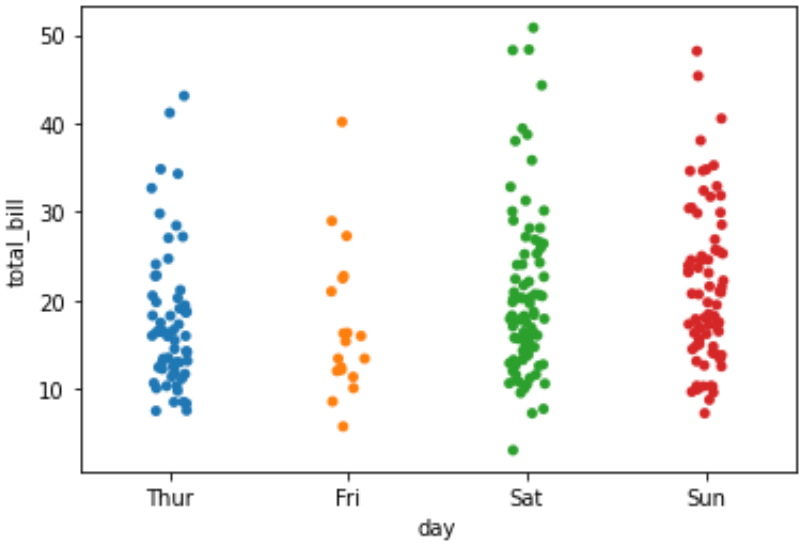
Strip Plot

```
1 tips.head()
```

	total_bill	tip	sex	smoker	day	time	size
0	16.99	1.01	Female	No	Sun	Dinner	2
1	10.34	1.66	Male	No	Sun	Dinner	3
2	21.01	3.50	Male	No	Sun	Dinner	3
3	23.68	3.31	Male	No	Sun	Dinner	2
4	24.59	3.61	Female	No	Sun	Dinner	4

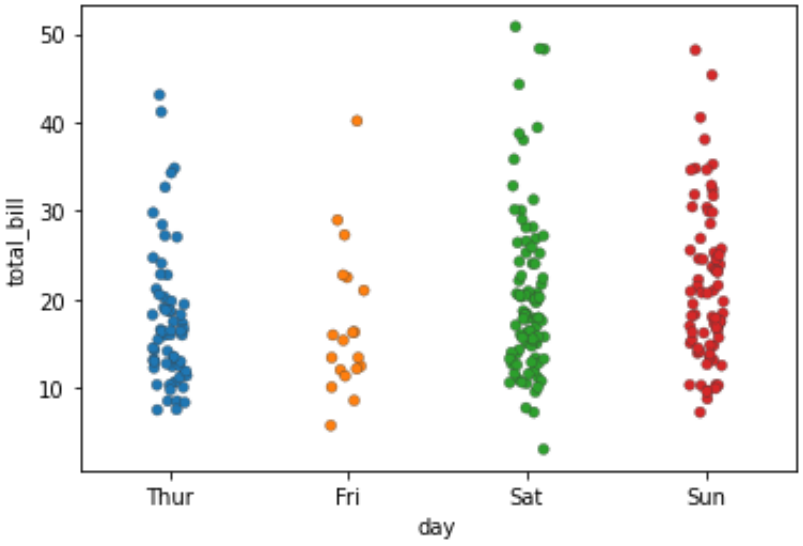
```
1 sns.stripplot(data=tips,x='day',y='total_bill')
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7f4892852210>
```



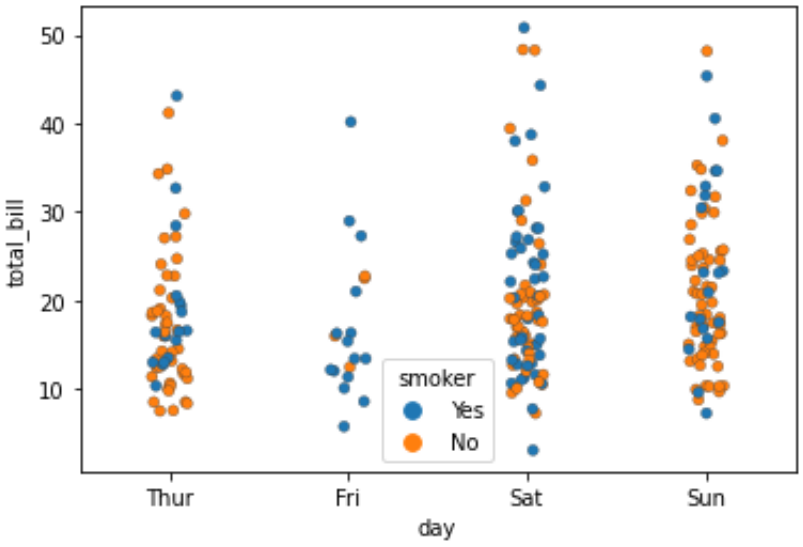

```
1 sns.stripplot(data=tips,x='day',y='total_bill',linewidth=0.3)
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7f489280c390>
```



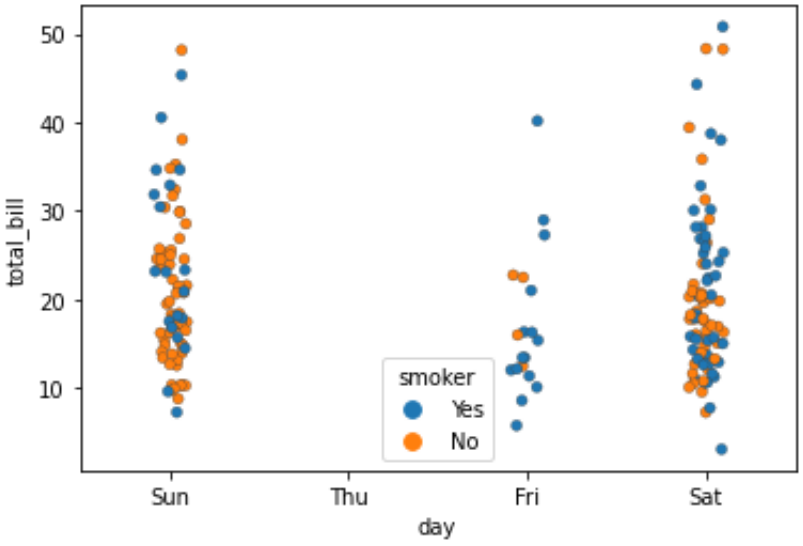
```
1 sns.stripplot(data=tips,x='day',y='total_bill',hue='smoker',linewidth=0.
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7f48927b16d0>
```




```
1 sns.stripplot(data=tips,x='day',y='total_bill',hue='smoker',linewidth=0.
```

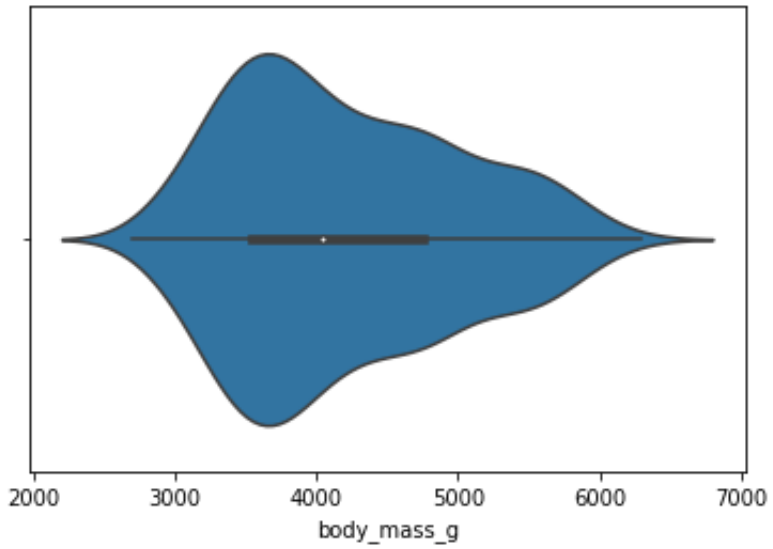
```
<matplotlib.axes._subplots.AxesSubplot at 0x7f489270cb90>
```



Violin Plot

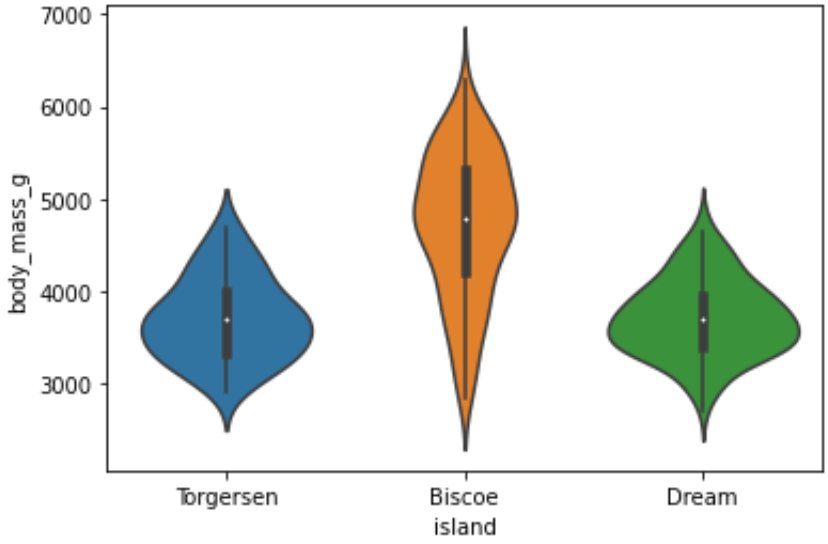
```
1 sns.violinplot(x='body_mass_g',data=p)
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7f4892705790>
```



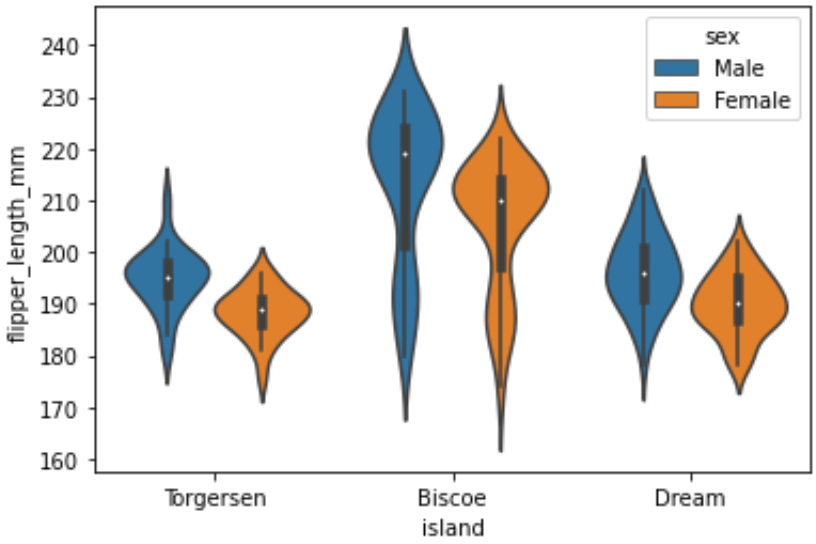

```
1 sns.violinplot(data=p,x='island',y='body_mass_g') #Can put data first
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7f48927a9890>
```



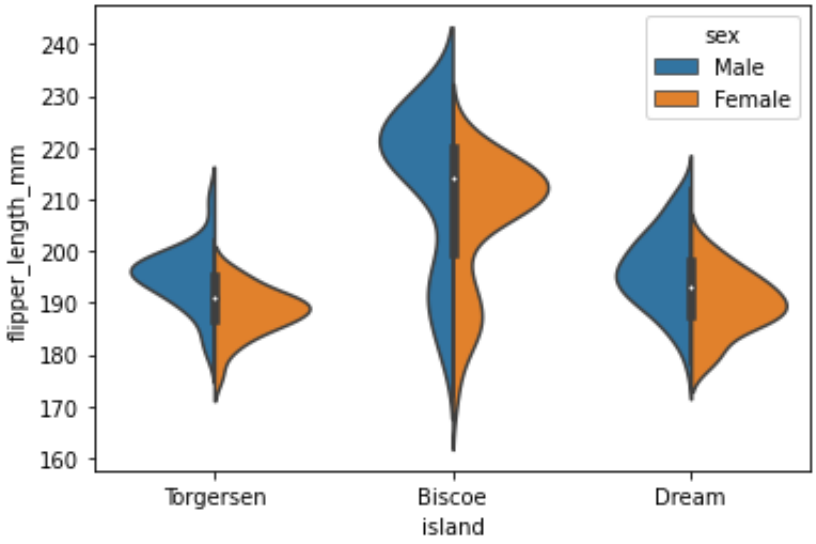
```
1 sns.violinplot(data=p,x='island',y='flipper_length_mm',hue='sex')
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7f48925e4110>
```



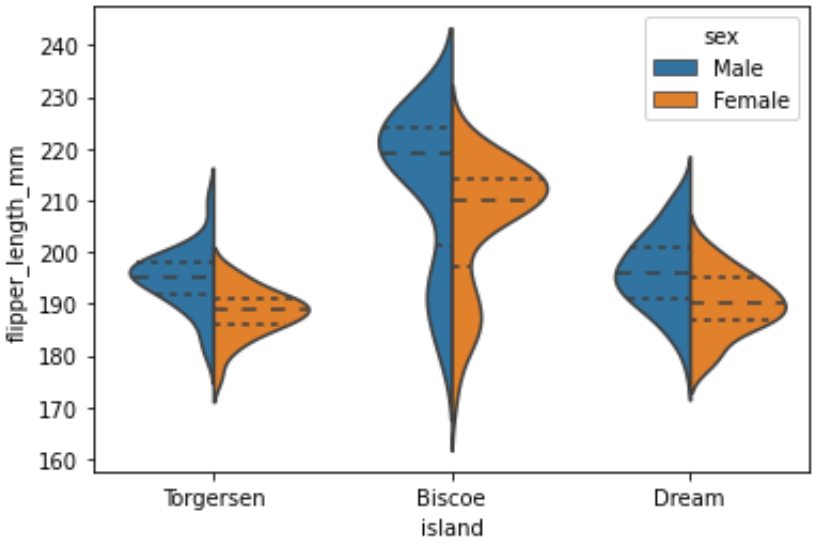

```
1 sns.violinplot(data=p,x='island',y='flipper_length_mm',hue='sex',split=1
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7f4892515250>
```



```
1 sns.violinplot(data=p,x='island',y='flipper_length_mm',hue='sex',split=1
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7f4892430410>
```




```
1 p.head()
```

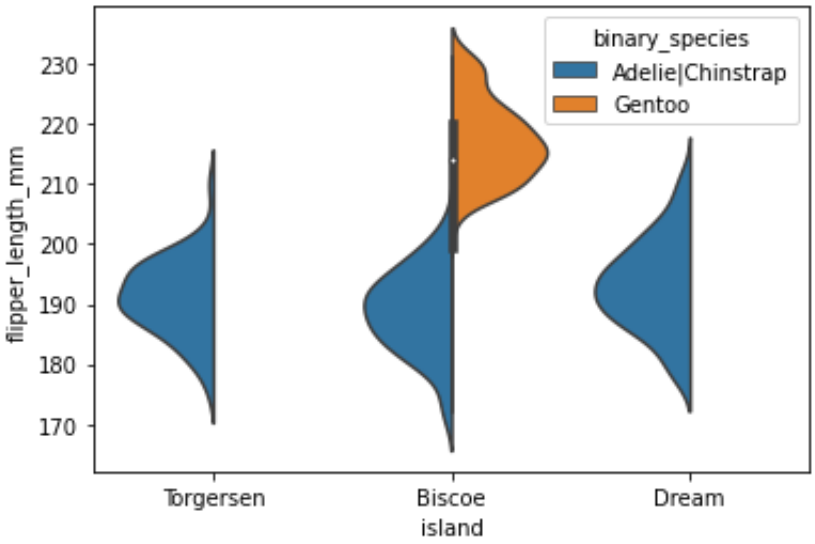
	species	island	bill_length_mm	bill_depth_mm	flipper_length_mm
0	Adelie	Torgersen	39.1	18.7	181.0
1	Adelie	Torgersen	39.5	17.4	186.0
2	Adelie	Torgersen	40.3	18.0	195.0
3	Adelie	Torgersen	NaN	NaN	NaN
4	Adelie	Torgersen	36.7	19.3	193.0

```
1 p['binary_species']=p.species.apply(lambda x : 'Gentoo' if x =='Gentoo'
2 p.head()
```

	species	island	bill_length_mm	bill_depth_mm	flipper_length_mm
0	Adelie	Torgersen	39.1	18.7	181.0
1	Adelie	Torgersen	39.5	17.4	186.0
2	Adelie	Torgersen	40.3	18.0	195.0
3	Adelie	Torgersen	NaN	NaN	NaN
4	Adelie	Torgersen	36.7	19.3	193.0

```
1 sns.violinplot(data=p,x='island',y='flipper_length_mm',hue='binary_speci
```

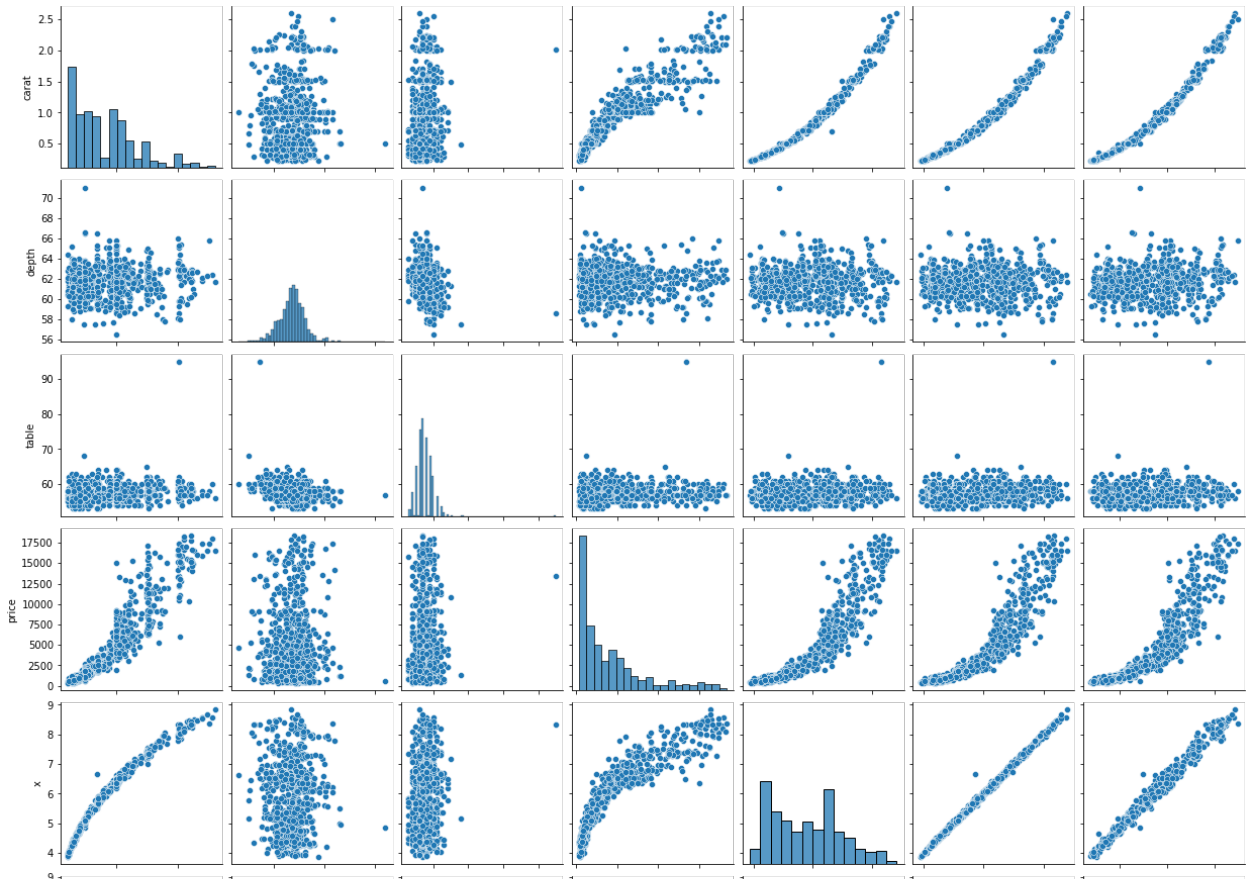
```
<matplotlib.axes._subplots.AxesSubplot at 0x7f489237a410>
```

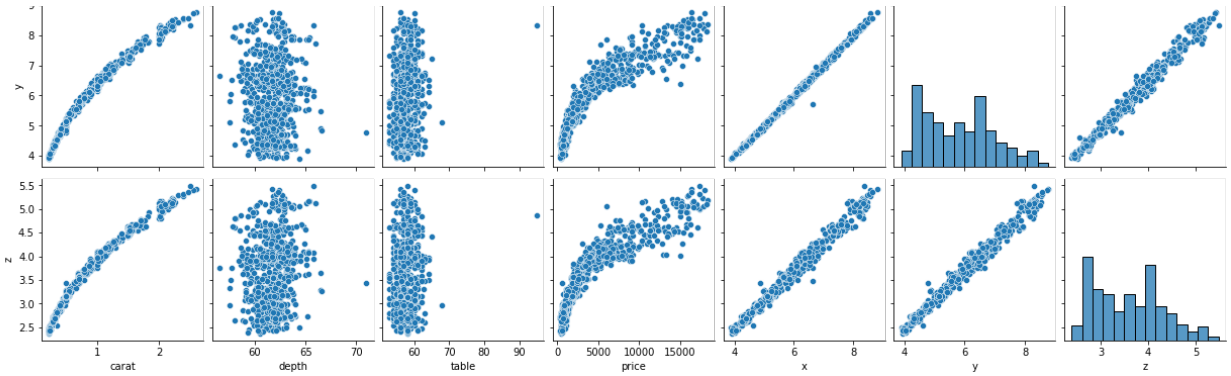


Pair Plot

```
1 sns.pairplot(d.sample(1000))
```

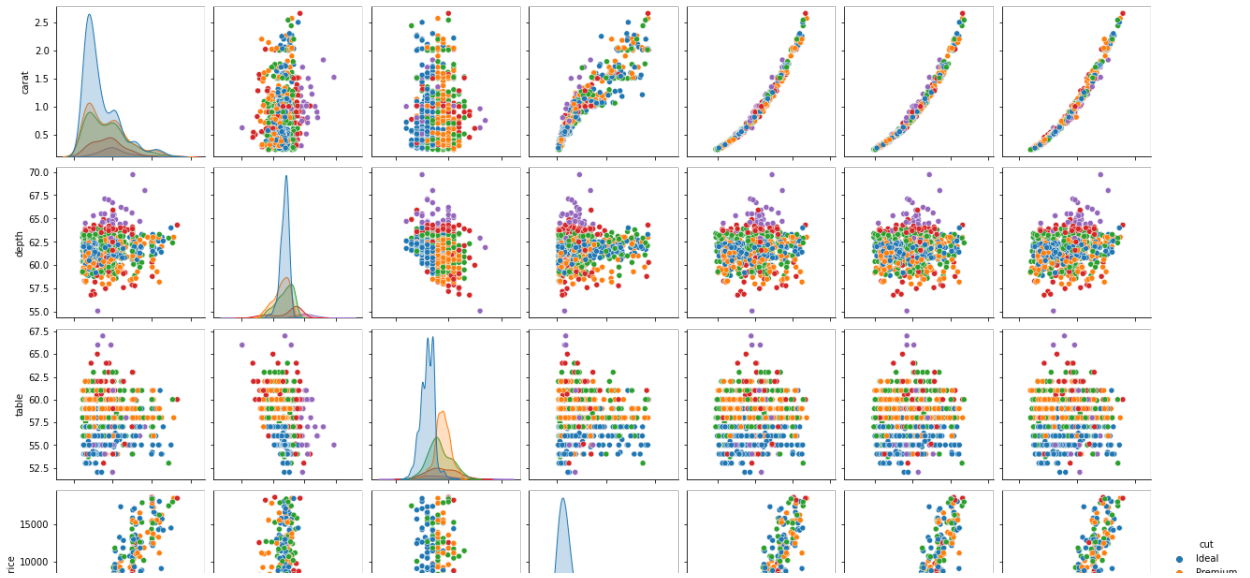
```
<seaborn.axisgrid.PairGrid at 0x7f4892adcf10>
```

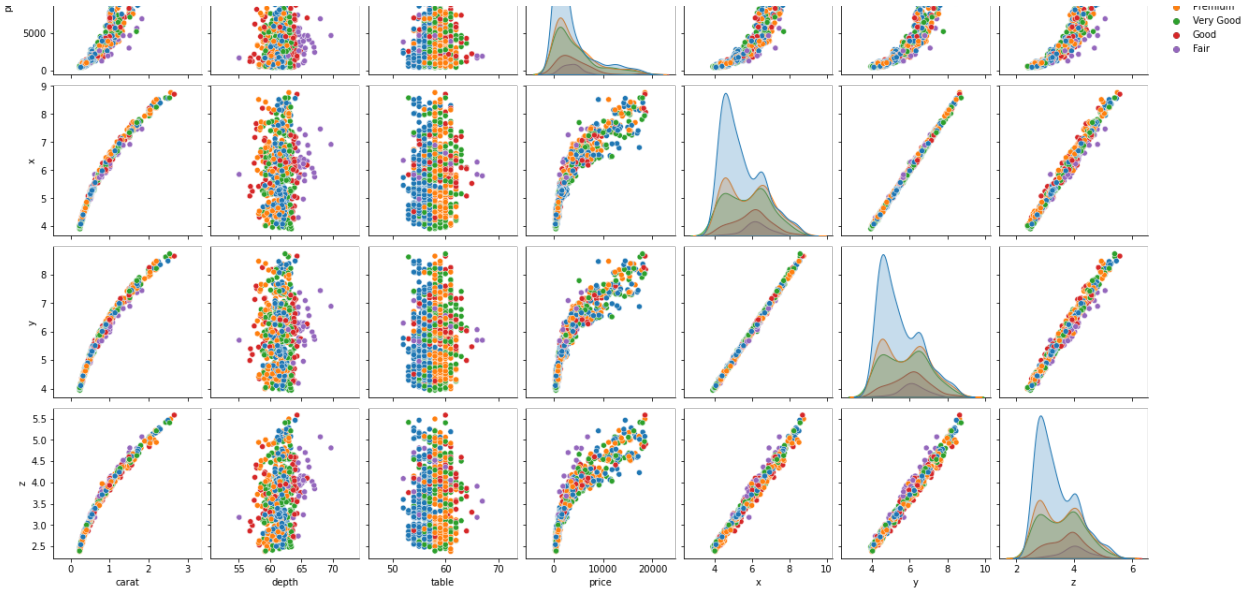





```
1 sns.pairplot(d.sample(1000),hue='cut')
```

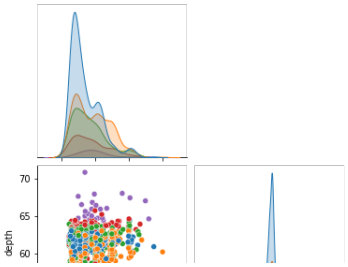
```
<seaborn.axisgrid.PairGrid at 0x7f4891003ad0>
```

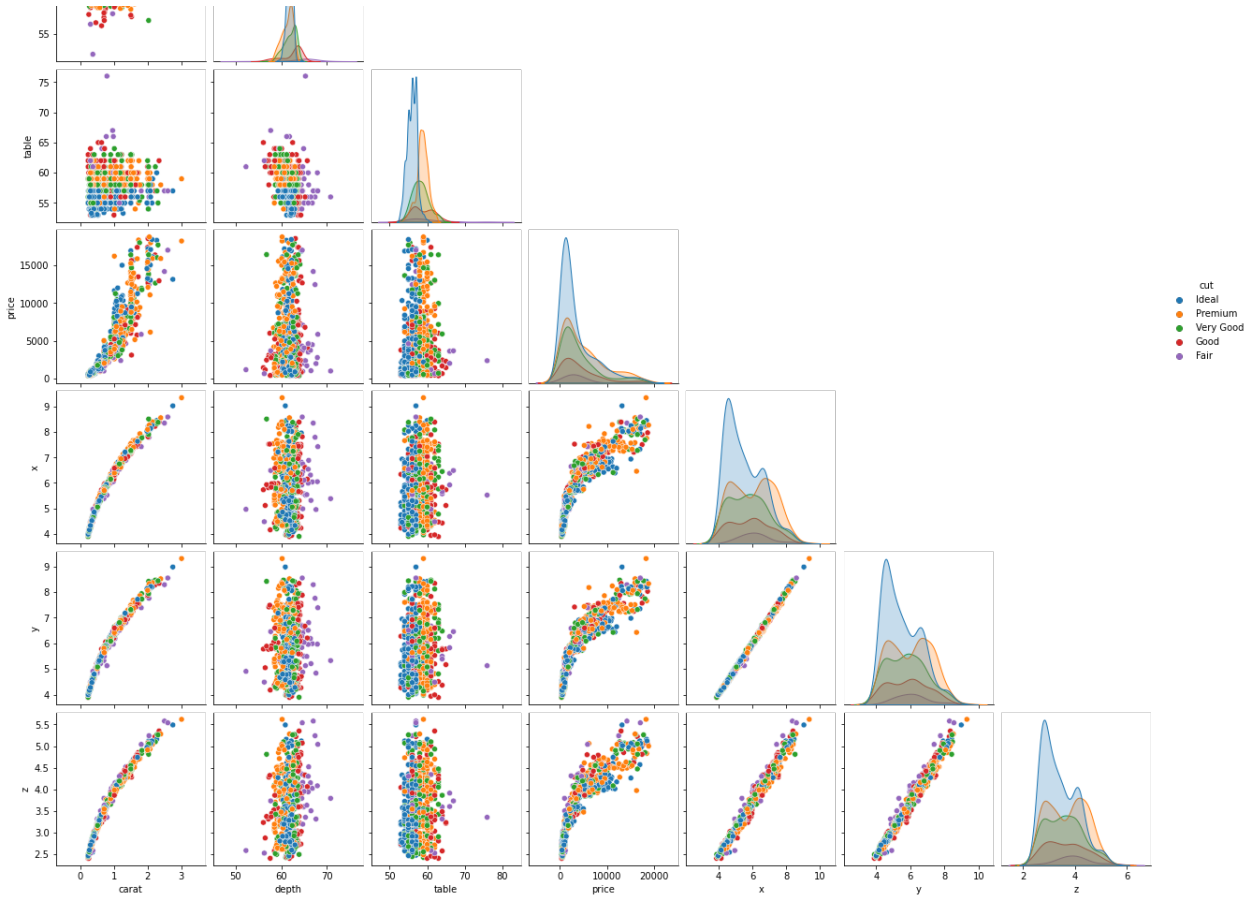





```
1 sns.pairplot(d.sample(1000),hue='cut',corner=True)
```

```
<seaborn.axisgrid.PairGrid at 0x7f488f883c10>
```






```
1 x = np.random.normal(size = 1000)
```

```
2 x
```

```
array([ 1.06104936e+00,  9.63700521e-02, -1.26151537e+00, -1.52840706e
       -1.19313926e+00, -3.41517046e-01, -1.68764617e+00, -3.66872608e
        2.11158367e-01,  1.04641208e+00,  2.12267595e+00, -5.33208610e
       -6.43494681e-01,  1.06375111e-01,  1.49661411e+00,  1.97663704e
       -6.28611001e-01,  2.20996620e-01, -1.07551303e-01,  1.36411652e
       -1.46353679e+00,  8.08858700e-01, -8.40231742e-01,  8.78917316e
        2.01058071e-01,  1.23822558e-01,  3.18398148e-01, -5.90985968e
       -8.81099808e-01,  1.23366576e+00, -1.07726197e-01, -8.45323979e
        8.93734204e-01,  8.49985425e-01,  4.82909751e-01, -1.16780249e
       -6.54714731e-01,  4.44762787e-01, -3.37430398e-01, -2.97397996e
        1.35445555e+00,  1.63609748e-01,  2.00193115e-01,  1.36596348e
       -5.93995290e-01,  1.17453235e+00, -2.67265504e-01,  4.66939345e
       -4.35157549e-01, -1.05535120e+00,  2.73841416e+00,  1.82617240e
        6.47354744e-01,  1.30566883e+00, -6.91124597e-01,  1.10788325e
        2.24543686e+00, -1.41936156e+00,  1.28966161e+00,  1.33977511e
       -5.78019914e-01,  6.48932736e-01, -7.53715755e-01, -1.47097267e
        3.77060377e-03, -1.42450018e+00, -9.53919182e-01, -9.39945954e
       -5.56561633e-01, -3.76622677e-01, -5.42837492e-01,  3.09568576e
        7.76685453e-01, -3.10741860e-02,  9.17263852e-02, -8.29015413e
       -1.76063035e-01,  1.27513068e-04, -2.40756157e-03, -5.05196682e
        9.12480389e-01, -1.62280926e+00, -1.23518460e+00,  8.99863788e
       -1.50099901e+00, -2.22096180e+00,  1.52353423e+00, -6.90360921e
        1.08959734e+00, -9.05916911e-01, -2.01079372e+00, -8.86338086e
        2.02662857e+00, -2.32910008e-01, -1.03866023e+00, -1.35532094e
        1.08268527e+00,  4.51353236e-01,  7.83980127e-01,  1.14767661e
       -8.00643698e-01,  1.36704695e+00,  5.40916268e-01, -1.09521685e
       -1.67700460e-02, -3.73585136e-02, -4.63964319e-01,  1.64813783e
        3.98257530e-01,  2.22567633e-01,  1.58406408e+00,  2.05143693e
       -1.08610525e+00,  3.38394764e-01, -4.29963696e-01,  1.04332606e
       -7.40147836e-01,  6.99655783e-01,  1.09765220e-01, -8.85232171e
        1.87828137e-01, -1.76065613e+00,  1.70202652e+00,  1.27099642e
       -1.02240197e+00,  3.46908319e-01, -2.78997864e-01, -2.64652890e
       -3.20597791e-01, -1.83697695e+00,  1.11799935e+00,  2.13643247e
        1.74487249e-01,  8.81430944e-01,  3.04910283e-01, -7.13133613e
       -2.88097049e-01,  1.53325736e-01,  4.17162405e-01,  1.67086378e
        5.99857566e-01, -1.38001945e+00, -1.00102736e+00, -5.09439265e
        7.76061565e-02, -7.67223302e-01, -5.37009566e-01,  2.45696477e
       -2.90461765e-01, -1.49686710e-01,  3.33140822e-01,  4.15291302e
       -9.44082625e-03, -9.77952359e-01,  1.00908515e-01,  1.60322167e
        1.08086947e+00,  7.59985722e-01, -1.25227632e+00,  4.46362928e
        2.87701331e-01, -4.61648981e-01, -1.43211364e-01, -1.37188841e
       -5.54812343e-01, -9.09660427e-01, -2.16707465e-01, -1.63272576e
       -4.95108087e-01, -2.47528546e+00,  9.77364396e-01,  1.46852099e
       -5.96852640e-01,  9.54972327e-01,  3.36926225e-01, -7.56062157e
       -2.72143080e-01,  5.44074513e-01,  2.82887603e+00, -1.40233483e
        1.05526887e+00,  8.51782014e-01,  1.81996718e+00, -1.47092190e
       -3.60106243e-01, -7.92200669e-01,  1.32974361e+00,  6.57084667e
       -1.36301809e+00, -7.16794542e-01,  8.34980429e-01,  2.47811965e
        1.03671453e+00,  1.05437139e-01,  2.66559056e+00, -9.52474480e
        2.09929637e-02, -6.79269257e-01,  7.93190797e-01, -2.67170425e
        2.80231355e-01, -2.49535793e-01, -9.20589485e-01,  1.67840482e
```

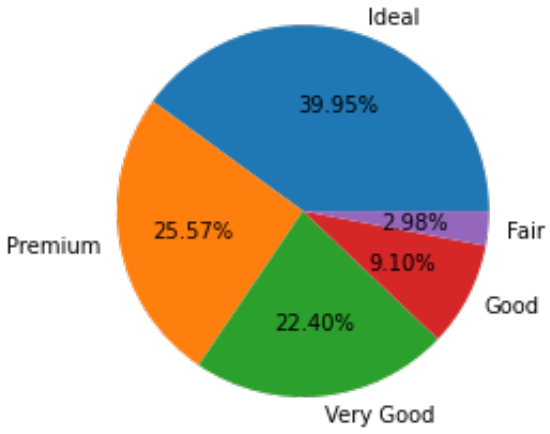
```
 8.16347099e-01, -4.15366800e-01, -1.27372843e-01,  1.83541275e
-6.95136969e-02,  1.31379684e-01, -4.77003739e-01, -9.52875794e
-1.82014403e-01,  7.58243810e-01, -4.93565190e-01,  7.48271281e
 1.10918331e-01, -5.49749685e-01,  6.90829533e-01,  1.63292032e
-6.59856708e-01,  1.20015254e+00, -2.33665957e+00,  3.56477753e
-6.56608997e-01,  7.66500441e-01,  9.07603725e-01,  9.65571234e
-2.88753975e-01,  7.61185772e-01,  1.32311117e+00,  1.40224081e
 4.29222796e-01,  1.19103647e+00, -4.58704276e-01, -1.06646011e
```

Pie Chart

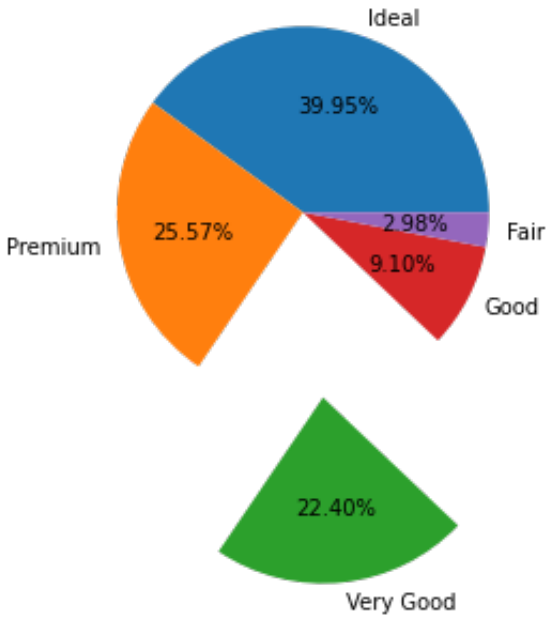
```
1 d.groupby('cut')['cut'].count()
```

```
cut
Ideal        21551
Premium      13791
Very Good    12082
Good          4906
Fair          1610
Name: cut, dtype: int64
```

```
1 z_d = d.groupby('cut')['cut'].count()
2 plt.pie(z_d, labels = z_d.index, autopct = "%.2f%%")
3 plt.show()
```



```
plt.pie(z_d, labels = z_d.index, autopct = "%.2f%%",explode=[0,0,1,0,0])
plt.show()
```




```
plt.pie(np.random.randint(0,10,5), wedgeprops = dict(width = 0.5))
plt.show()
```

```
cmap = plt.get_cmap('Set1')
cmap
```

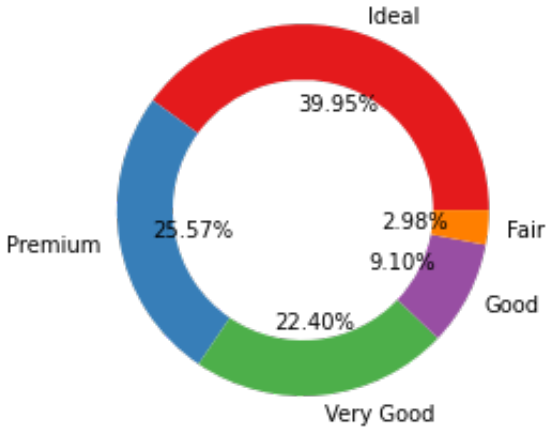
```
<matplotlib.colors.ListedColormap at 0x7f48c3d6d950>
```

```
mycolor = cmap(np.arange(10))
mycolor
```

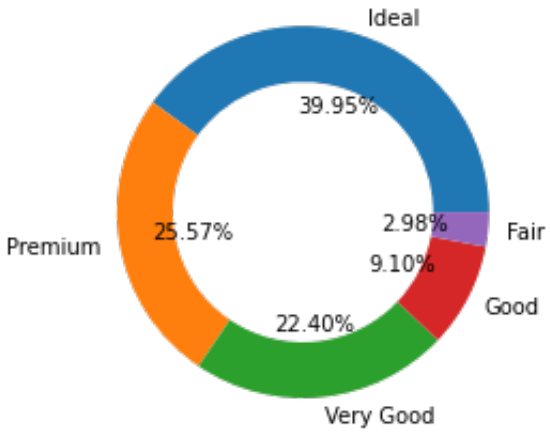
```
array([[0.89411765, 0.10196078, 0.10980392, 1.        ],
       [0.21568627, 0.49411765, 0.72156863, 1.        ],
       [0.30196078, 0.68627451, 0.29019608, 1.        ],
       [0.59607843, 0.30588235, 0.63921569, 1.        ],
       [1.        , 0.49803922, 0.        , 1.        ],
       [1.        , 1.        , 0.2       , 1.        ],
       [0.65098039, 0.3372549 , 0.15686275, 1.        ],
       [0.96862745, 0.50588235, 0.74901961, 1.        ],
       [0.6       , 0.6       , 0.6       , 1.        ],
       [0.6       , 0.6       , 0.6       , 1.        ]])
```

```
plt.pie(np.random.randint(0,10,5), wedgeprops = dict(width = 0.3), color
plt.show()
```

```
plt.pie(z_d, labels = z_d.index, autopct = "%.2f%%", wedgeprops = dict(w
plt.show()
```



```
plt.pie(z_d, labels = z_d.index, autopct = "%.2f%%", wedgeprops = dict(w
plt.show()
```




```
p1=pd.crosstab(p['species'],p['island'])
p1
```

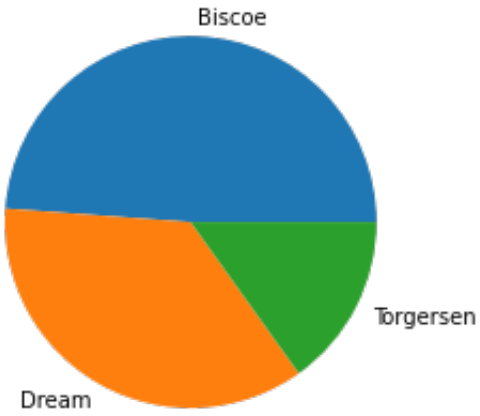
species	island Biscoe	Dream	Torgersen
Adelie	44	56	52
Chinstrap	0	68	0
Gentoo	124	0	0

```
p2=p1.T
p2
```

island	species Adelie	Chinstrap	Gentoo
Biscoe	44	0	124
Dream	56	68	0
Torgersen	52	0	0

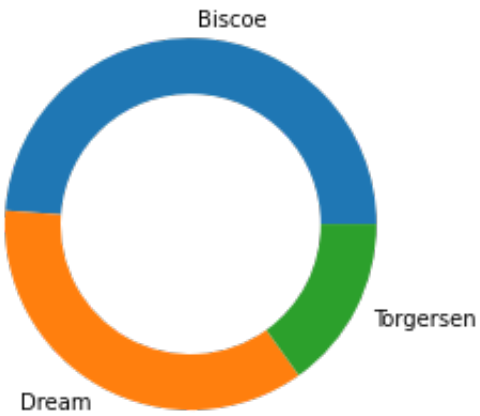
```
1 plt.pie(p2.sum(axis=1),labels=p2.index)
```

```
([<matplotlib.patches.Wedge at 0x7f488e5a5510>,
  <matplotlib.patches.Wedge at 0x7f488e5a5a10>,
  <matplotlib.patches.Wedge at 0x7f488e5a59d0>],
 [Text(0.040174244511466346, 1.099266132507471, 'Biscoe'),
  Text(-0.5383596868970475, -0.9592543184808255, 'Dream'),
  Text(0.9782763205126381, -0.502966639772713, 'Torgersen')])
```



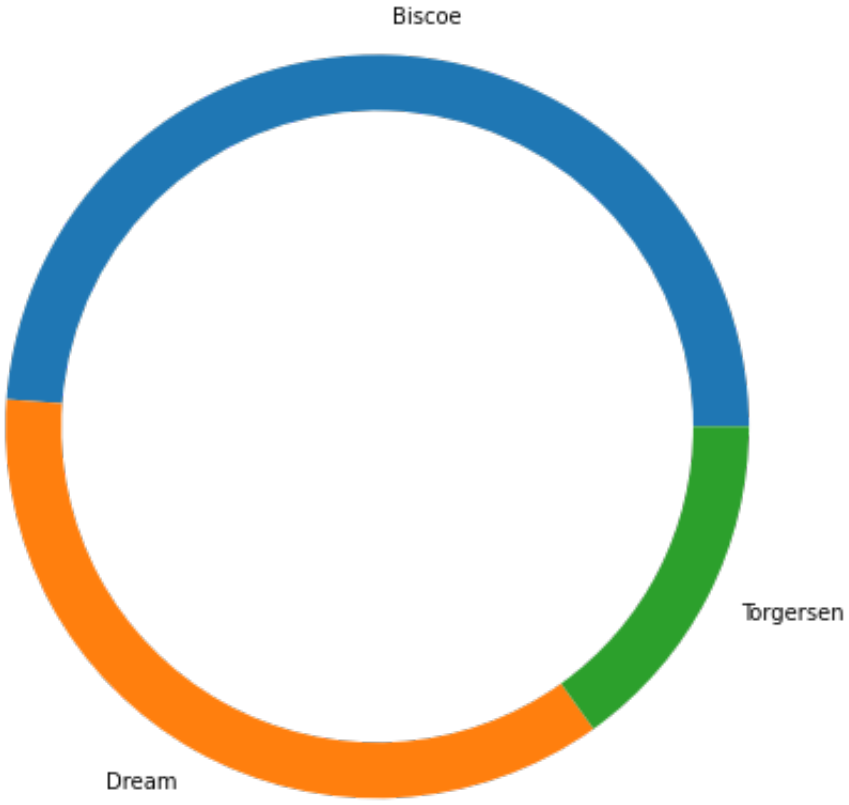

```
1 plt.pie(p2.sum(axis=1),labels=p2.index,wedgeprops=dict(width=0.3))
```

```
([<matplotlib.patches.Wedge at 0x7f488e5719d0>,
  <matplotlib.patches.Wedge at 0x7f488e571ed0>,
  <matplotlib.patches.Wedge at 0x7f488e57c490>],
 [Text(0.040174244511466346, 1.099266132507471, 'Biscoe'),
  Text(-0.5383596868970475, -0.9592543184808255, 'Dream'),
  Text(0.9782763205126381, -0.502966639772713, 'Torgersen')])
```



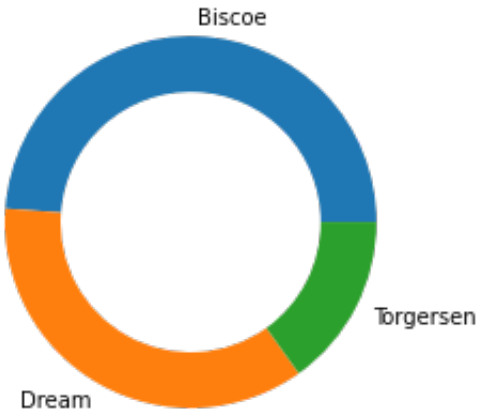
```
1 plt.pie(p2.sum(axis=1),labels=p2.index,wedgeprops=dict(width=0.3),radius
```

```
([<matplotlib.patches.Wedge at 0x7f488e53ecd0>,
  <matplotlib.patches.Wedge at 0x7f488e54a210>,
  <matplotlib.patches.Wedge at 0x7f488e54a1d0>],
 [Text(0.08034848902293269, 2.198532265014942, 'Biscoe'),
  Text(-1.076719373794095, -1.918508636961651, 'Dream'),
  Text(1.9565526410252763, -1.005933279545426, 'Torgersen')])
```



```
1 plt.pie(p2.sum(axis=1),labels=p2.index,wedgeprops=dict(width=0.3),radius
```

```
([<matplotlib.patches.Wedge at 0x7f488e510e50>,
  <matplotlib.patches.Wedge at 0x7f488e51d390>,
  <matplotlib.patches.Wedge at 0x7f488e51d350>],
 [Text(0.040174244511466346, 1.099266132507471, 'Biscoe'),
  Text(-0.5383596868970475, -0.9592543184808255, 'Dream'),
  Text(0.9782763205126381, -0.502966639772713, 'Torgersen')])
```

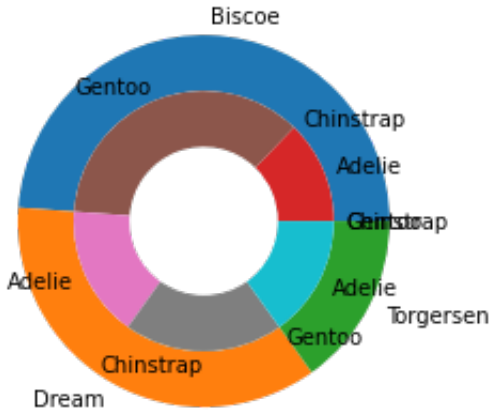



```
1 p2
```

species island	Adelie	Chinstrap	Gentoo
Biscoe	44	0	124
Dream	56	68	0
Torgersen	52	0	0

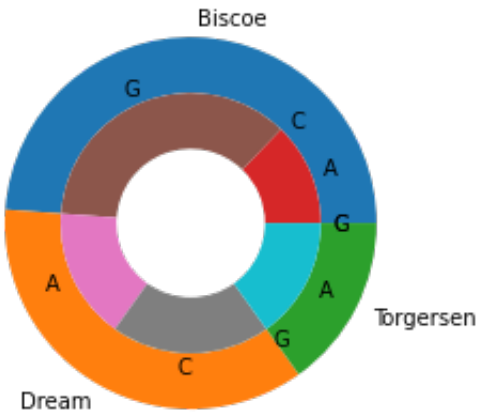
```
plt.pie(p2.sum(axis=1),labels=p2.index,wedgeprops=dict(width=0.3),radius
plt.pie(p2.values.flatten(),labels=['Adelie','Chinstrap','Gentoo','Adeli
```

```
([<matplotlib.patches.Wedge at 0x7f488e57cf90>,
  <matplotlib.patches.Wedge at 0x7f488e478450>,
  <matplotlib.patches.Wedge at 0x7f488e478790>,
  <matplotlib.patches.Wedge at 0x7f488e478750>,
  <matplotlib.patches.Wedge at 0x7f488e487250>,
  <matplotlib.patches.Wedge at 0x7f488e478110>,
  <matplotlib.patches.Wedge at 0x7f488e4783d0>,
  <matplotlib.patches.Wedge at 0x7f488e4e3c90>,
  <matplotlib.patches.Wedge at 0x7f488e493250>],
 [Text(0.7086665600368222, 0.30115063786347374, 'Adelie'),
  Text(0.5344371254920076, 0.5543256794483187, 'Chinstrap'),
  Text(-0.2750677757511998, 0.719192407317602, 'Gentoo'),
  Text(-0.697195151053494, -0.3268316406768105, 'Adelie'),
  Text(-1.414467053015193e-16, -0.77, 'Chinstrap'),
  Text(0.4480313745243962, -0.6262330935376861, 'Gentoo'),
  Text(0.6847934573226385, -0.35207658372591005, 'Adelie'),
  Text(0.77, -1.885956070686924e-16, 'Chinstrap'),
  Text(0.77, -1.885956070686924e-16, 'Gentoo')])
```



```
plt.pie(p2.sum(axis=1),labels=p2.index,wedgeprops=dict(width=0.3),radius
plt.pie(p2.values.flatten(),labels=['A','C','G','A','C','G','A','C','G']
```

```
([<matplotlib.patches.Wedge at 0x7f488e4278d0>,
  <matplotlib.patches.Wedge at 0x7f488e3ec510>,
  <matplotlib.patches.Wedge at 0x7f488e3ec850>,
  <matplotlib.patches.Wedge at 0x7f488e3ec810>,
  <matplotlib.patches.Wedge at 0x7f488e3f8310>,
  <matplotlib.patches.Wedge at 0x7f488e3ec1d0>,
  <matplotlib.patches.Wedge at 0x7f488e3ec490>,
  <matplotlib.patches.Wedge at 0x7f488e455d90>,
  <matplotlib.patches.Wedge at 0x7f488e405310>],
 [Text(0.7086665600368222, 0.30115063786347374, 'A'),
  Text(0.5344371254920076, 0.5543256794483187, 'C'),
  Text(-0.2750677757511998, 0.719192407317602, 'G'),
  Text(-0.697195151053494, -0.3268316406768105, 'A'),
  Text(-1.414467053015193e-16, -0.77, 'C'),
  Text(0.4480313745243962, -0.6262330935376861, 'G'),
  Text(0.6847934573226385, -0.35207658372591005, 'A'),
  Text(0.77, -1.885956070686924e-16, 'C'),
  Text(0.77, -1.885956070686924e-16, 'G')])
```

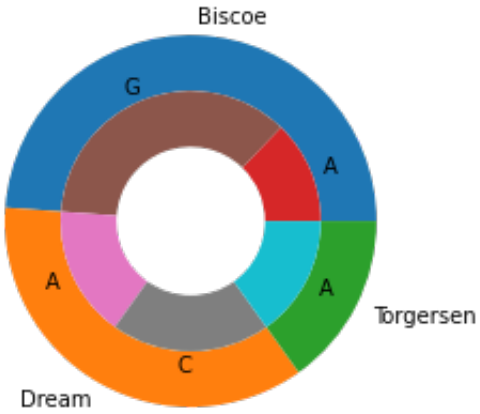



```
p2
```

species island	Adelie	Chinstrap	Gentoo
Biscoe	44	0	124
Dream	56	68	0
Torgersen	52	0	0

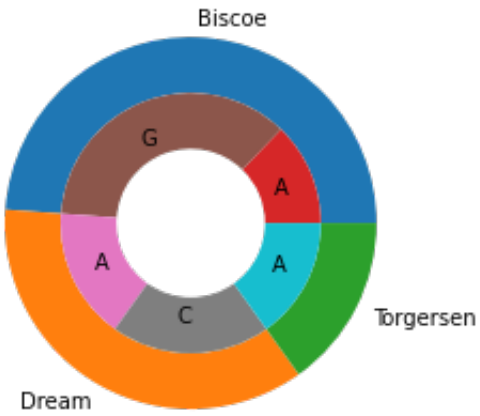
```
plt.pie(p2.sum(axis=1),labels=p2.index,wedgeprops=dict(width=0.3),radius
plt.pie(p2.values.flatten(),labels=['A','','G','A','C','','A','',''],wed
```

```
([<matplotlib.patches.Wedge at 0x7f488e51d050>,
  <matplotlib.patches.Wedge at 0x7f488e3df9d0>,
  <matplotlib.patches.Wedge at 0x7f488e3dfd10>,
  <matplotlib.patches.Wedge at 0x7f488e36c290>,
  <matplotlib.patches.Wedge at 0x7f488e36c2d0>,
  <matplotlib.patches.Wedge at 0x7f488e3df350>,
  <matplotlib.patches.Wedge at 0x7f488e36cd90>,
  <matplotlib.patches.Wedge at 0x7f488e3d6150>,
  <matplotlib.patches.Wedge at 0x7f488e3787d0>],
 [Text(0.7086665600368222, 0.30115063786347374, 'A'),
  Text(0.5344371254920076, 0.5543256794483187, ''),
  Text(-0.2750677757511998, 0.719192407317602, 'G'),
  Text(-0.697195151053494, -0.3268316406768105, 'A'),
  Text(-1.414467053015193e-16, -0.77, 'C'),
  Text(0.4480313745243962, -0.6262330935376861, ''),
  Text(0.6847934573226385, -0.35207658372591005, 'A'),
  Text(0.77, -1.885956070686924e-16, ''),
  Text(0.77, -1.885956070686924e-16, '')])
```



```
plt.pie(p2.sum(axis=1),labels=p2.index,wedgeprops=dict(width=0.3),radius
plt.pie(p2.values.flatten(),labels=['A','','G','A','C','','A','',''],wed
```

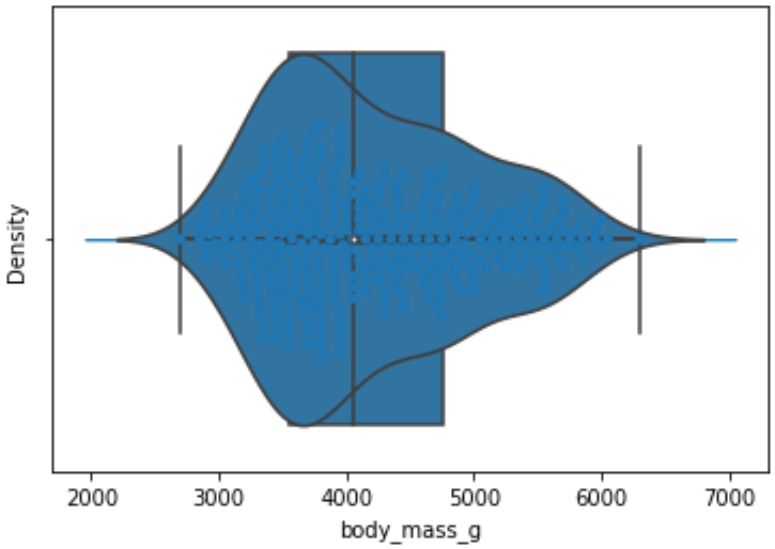
```
([<matplotlib.patches.Wedge at 0x7f488e384ed0>,
  <matplotlib.patches.Wedge at 0x7f488e353cd0>,
  <matplotlib.patches.Wedge at 0x7f488e35f050>,
  <matplotlib.patches.Wedge at 0x7f488e35f590>,
  <matplotlib.patches.Wedge at 0x7f488e35f5d0>,
  <matplotlib.patches.Wedge at 0x7f488e3536d0>,
  <matplotlib.patches.Wedge at 0x7f488e35f1d0>,
  <matplotlib.patches.Wedge at 0x7f488e348590>,
  <matplotlib.patches.Wedge at 0x7f488e2ecad0>],
 [Text(0.4509696291143413, 0.1916413150040287, 'A'),
  Text(0.34009635258582294, 0.3527527051034755, ''),
  Text(-0.17504313002349073, 0.4576678955657467, 'G'),
  Text(-0.4436696415794961, -0.20798377133978846, 'A'),
  Text(-9.001153973733045e-17, -0.48999999999999994, 'C'),
  Text(0.285110874697343, -0.3985119686148911, ''),
  Text(0.4357776546598608, -0.22404873509830636, 'A'),
  Text(0.48999999999999994, -1.200153863164406e-16, ''),
  Text(0.48999999999999994, -1.200153863164406e-16, '')])
```



Plotting 4 different plots in a figure

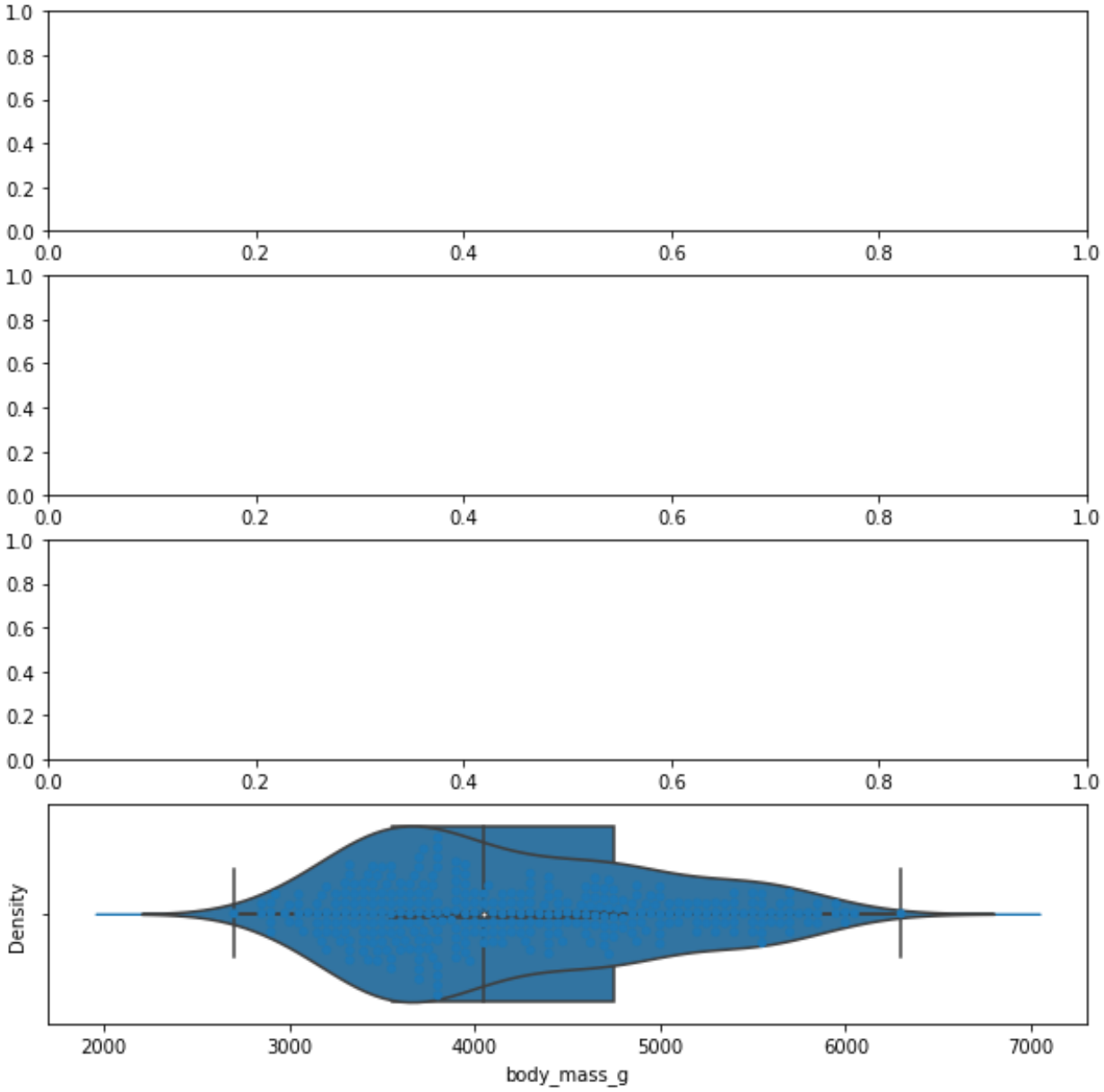
```
sns.kdeplot(data=p,x='body_mass_g',shade=1)
sns.boxplot(data=p,x='body_mass_g')
sns.swarmplot(data=p,x='body_mass_g')
sns.violinplot(data=p,x='body_mass_g')
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7f488e309350>
```



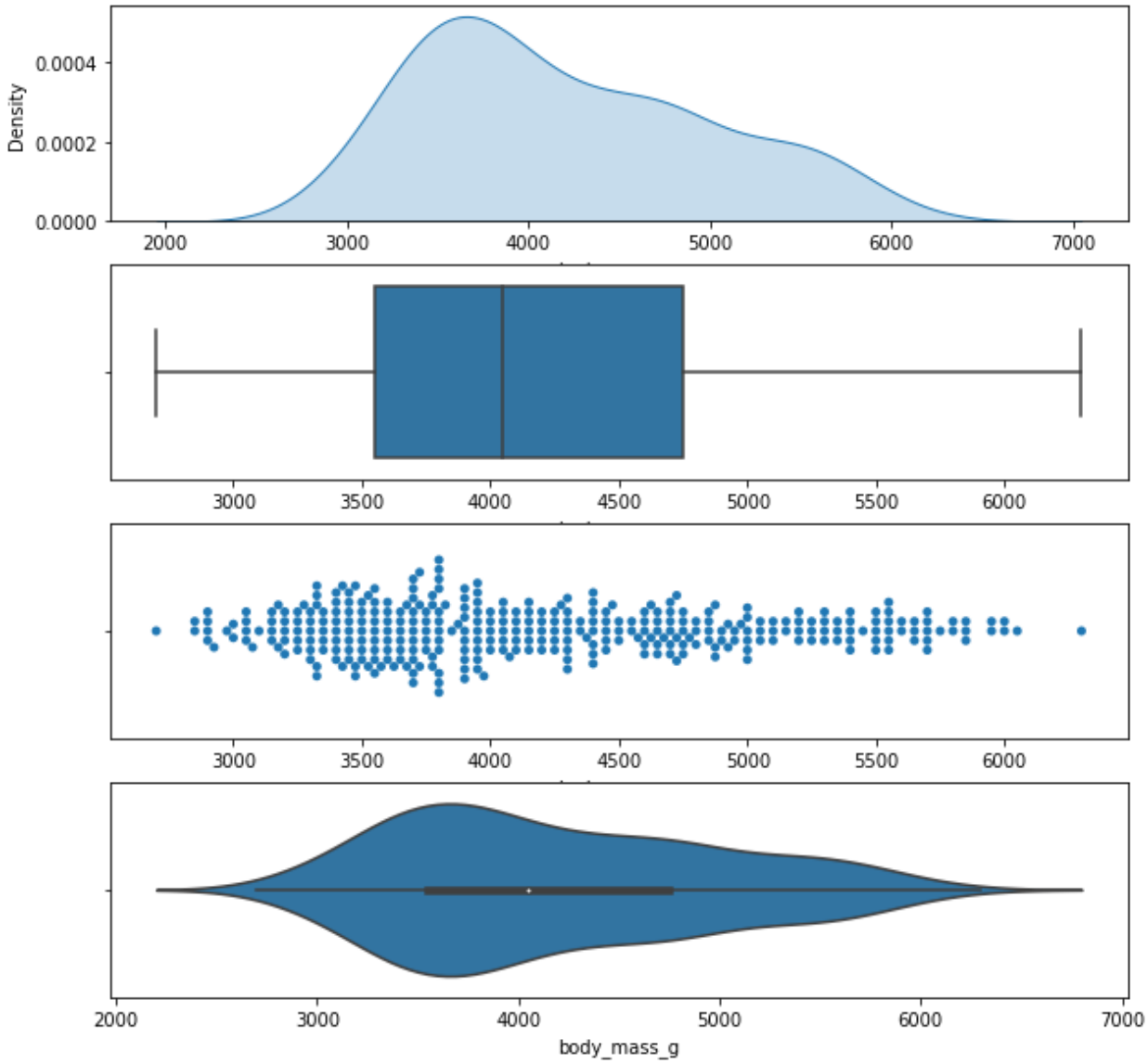
```
fig,axs = plt.subplots(nrows=4)
fig.set_size_inches(10,10)
sns.kdeplot(data=p,x='body_mass_g',shade=1)
sns.boxplot(data=p,x='body_mass_g')
sns.swarmplot(data=p,x='body_mass_g')
sns.violinplot(data=p,x='body_mass_g')
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7f488e1bee90>
```



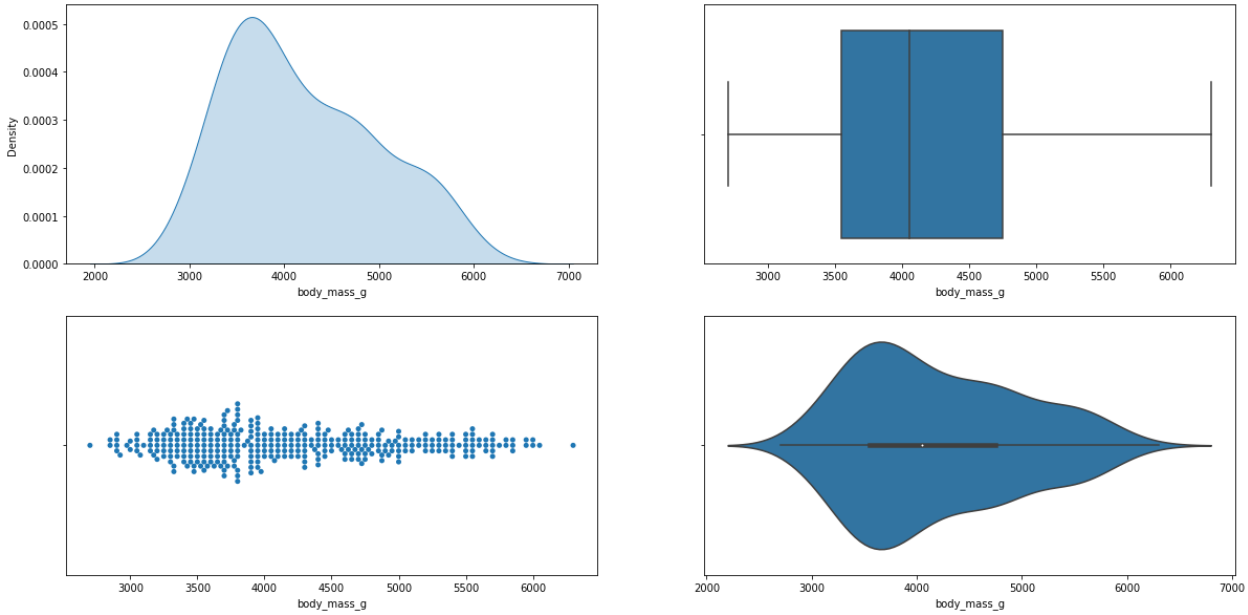
```
fig,axs = plt.subplots(nrows=4)
fig.set_size_inches(10,10)
sns.kdeplot(data=p,x='body_mass_g',shade=1,ax=axs[0])
sns.boxplot(data=p,x='body_mass_g',ax=axs[1])
sns.swarmplot(data=p,x='body_mass_g',ax=axs[2])
sns.violinplot(data=p,x='body_mass_g',ax=axs[3])
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7f488e04a710>
```



```
fig,axs = plt.subplots(nrows=2,ncols=2)
fig.set_size_inches(20,10)
sns.kdeplot(data=p,x='body_mass_g',shade=1,ax=axs[0][0])
sns.boxplot(data=p,x='body_mass_g',ax=axs[0][1])
sns.swarmplot(data=p,x='body_mass_g',ax=axs[1][0])
sns.violinplot(data=p,x='body_mass_g',ax=axs[1][1])
```

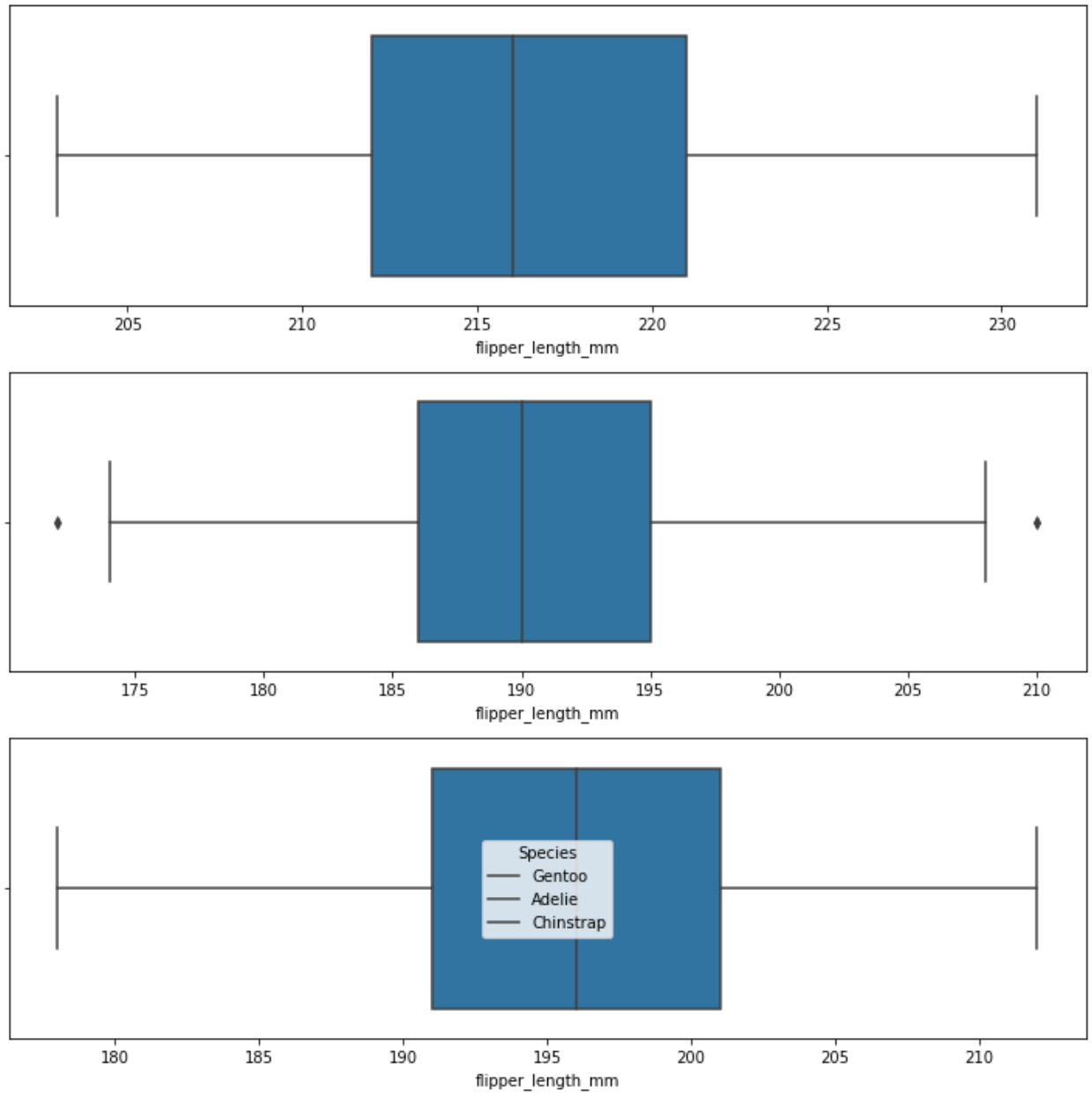
```
<matplotlib.axes._subplots.AxesSubplot at 0x7f488deb9c10>
```




```
fig,axs=plt.subplots(nrows=3)
fig.set_size_inches(10,10)
sns.boxplot(p[p['species']=='Gentoo'].flipper_length_mm,ax=axs[0])
sns.boxplot(p[p['species']=='Adelie'].flipper_length_mm,ax=axs[1])
sns.boxplot(p[p['species']=='Chinstrap'].flipper_length_mm,ax=axs[2])
plt.tight_layout()
```

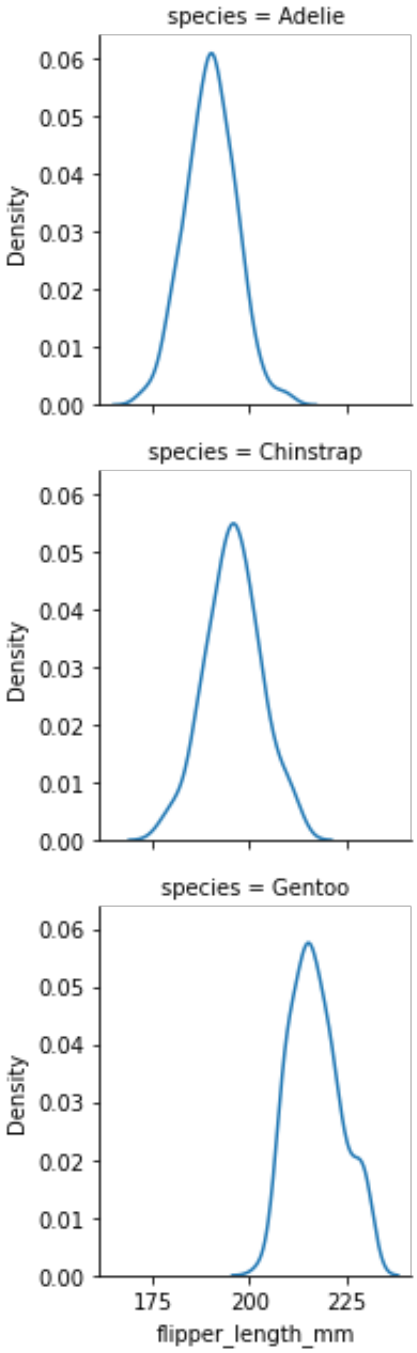
```
7 plt.legend(title='Species',labels=['Gentoo','Adelie','Chinstrap'])
```

```
/usr/local/lib/python3.7/dist-packages/seaborn/_decorators.py:43: Futu
  FutureWarning
/usr/local/lib/python3.7/dist-packages/seaborn/_decorators.py:43: Futu
  FutureWarning
/usr/local/lib/python3.7/dist-packages/seaborn/_decorators.py:43: Futu
  FutureWarning
<matplotlib.legend.Legend at 0x7f48925cd0d0>
```



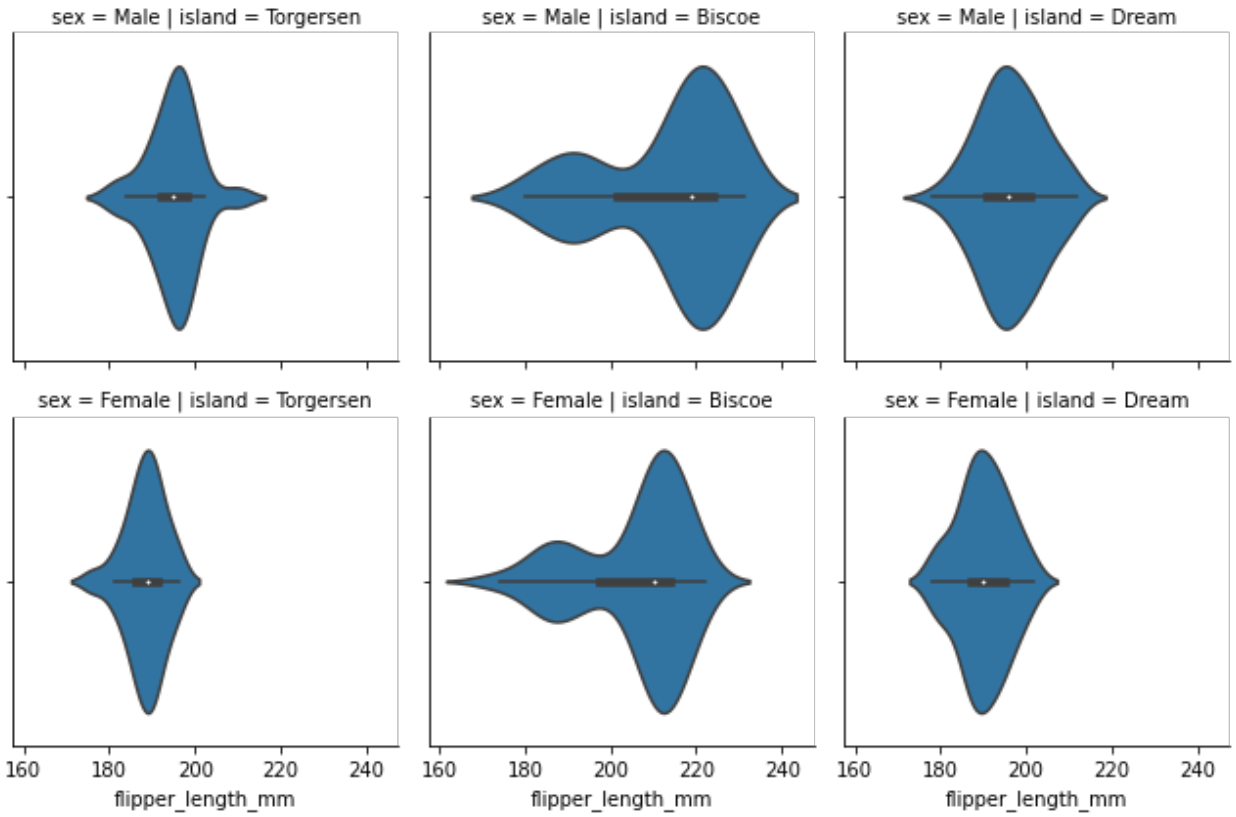
```
1 g=sns.FacetGrid(p,row='species')
2 g.map(sns.kdeplot,'flipper_length_mm')
```

```
<seaborn.axisgrid.FacetGrid at 0x7f488e2faad0>
```



```
1 g=sns.FacetGrid(p,row='sex',col='island')
2 g.map(sns.violinplot,'flipper_length_mm')
```

```
/usr/local/lib/python3.7/dist-packages/seaborn/axisgrid.py:670: UserWa
  warnings.warn(warning)
<seaborn.axisgrid.FacetGrid at 0x7f488db921d0>
```



Miscellaneous