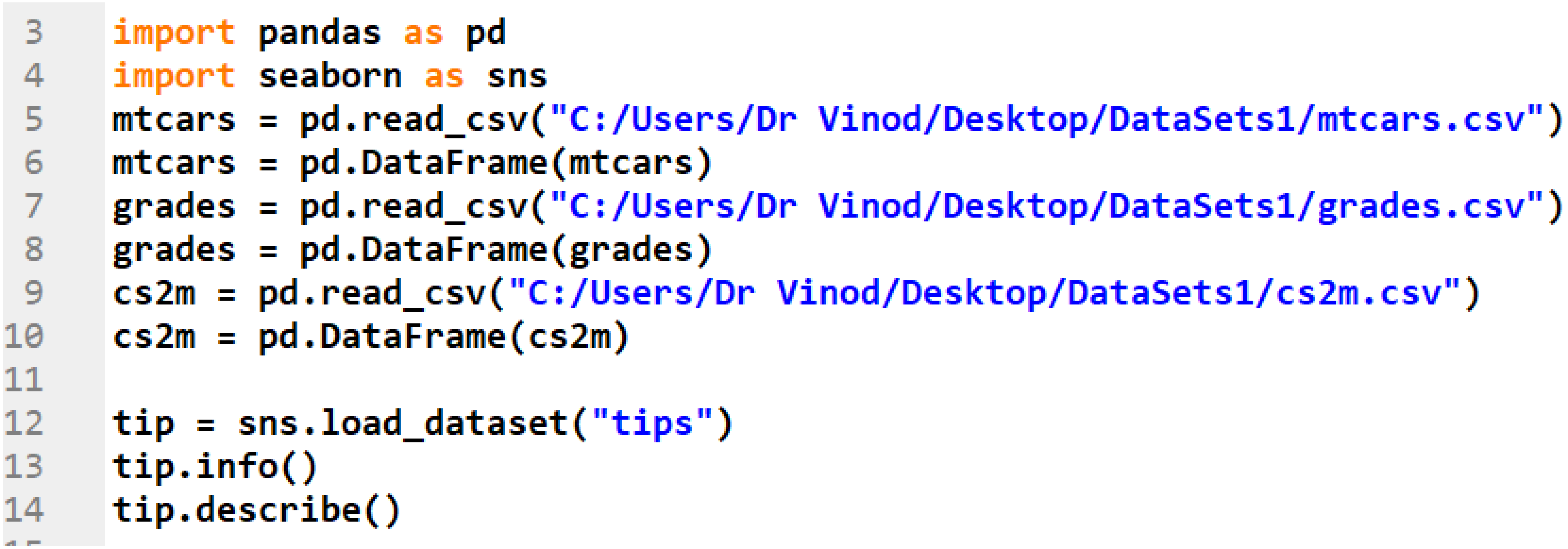
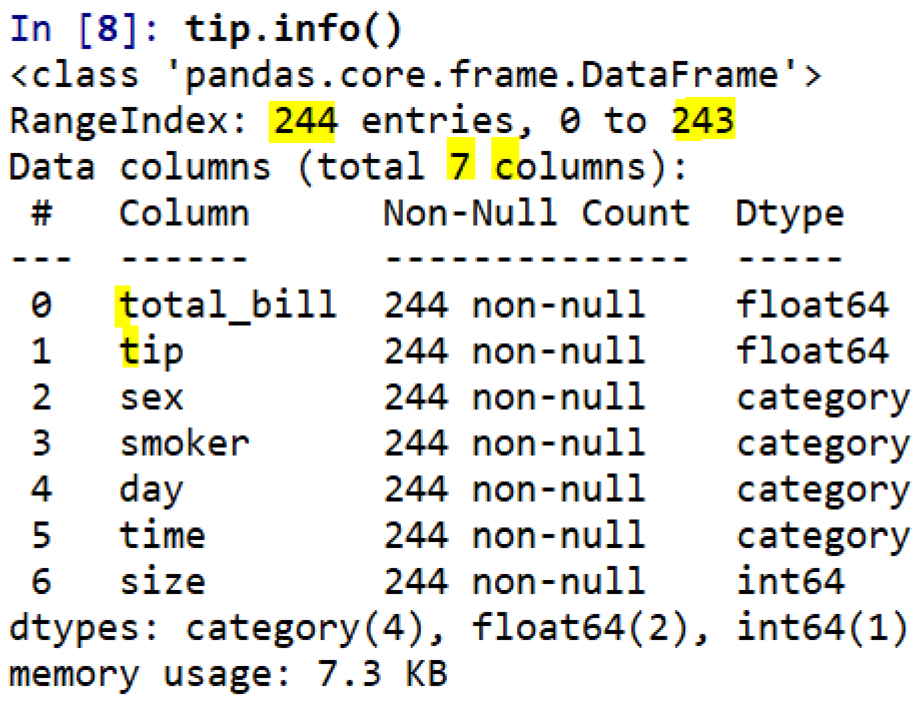
Seaborn

Libraries

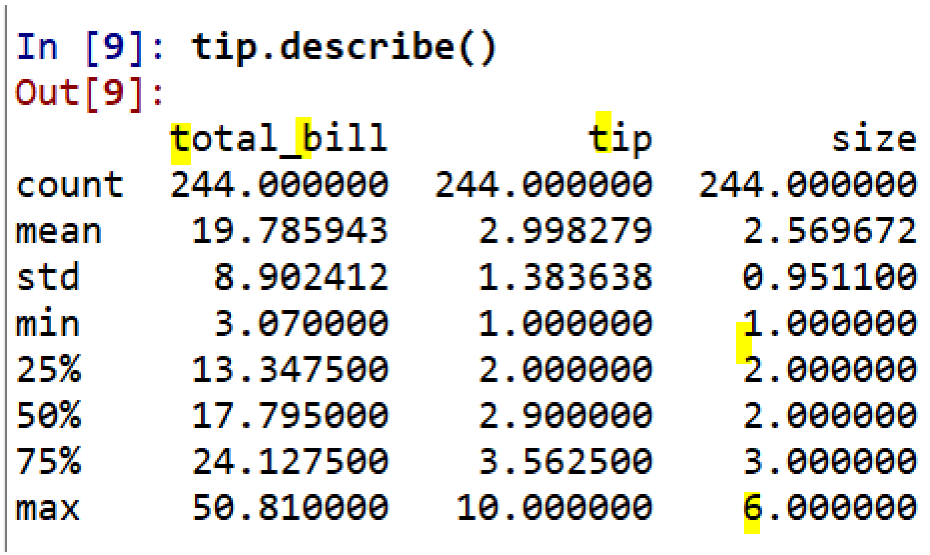


```python
import pandas as pd
import seaborn as sns
mtcars = pd.read_csv("C:/Users/Dr Vinod/Desktop/DataSets1/mtcars.csv")
mtcars = pd.DataFrame(mtcars)
grades = pd.read_csv("C:/Users/Dr Vinod/Desktop/DataSets1/grades.csv")
grades = pd.DataFrame(grades)
cs2m = pd.read_csv("C:/Users/Dr Vinod/Desktop/DataSets1/cs2m.csv")
cs2m = pd.DataFrame(cs2m)

tip = sns.load_dataset("tips")
tip.info()
tip.describe()
```

Tips data

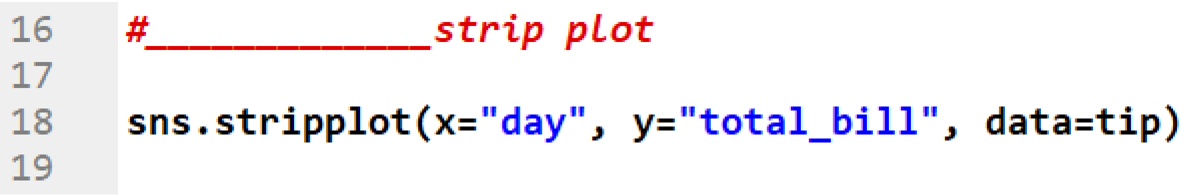


```
In [8]: tip.info()
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 244 entries, 0 to 243
Data columns (total 7 columns):
 #   Column      Non-Null Count  Dtype
---  ------      --------------  -----
 0   total_bill  244 non-null    float64
 1   tip         244 non-null    float64
 2   sex         244 non-null    category
 3   smoker      244 non-null    category
 4   day         244 non-null    category
 5   time        244 non-null    category
 6   size        244 non-null    int64
dtypes: category(4), float64(2), int64(1)
memory usage: 7.3 KB
```

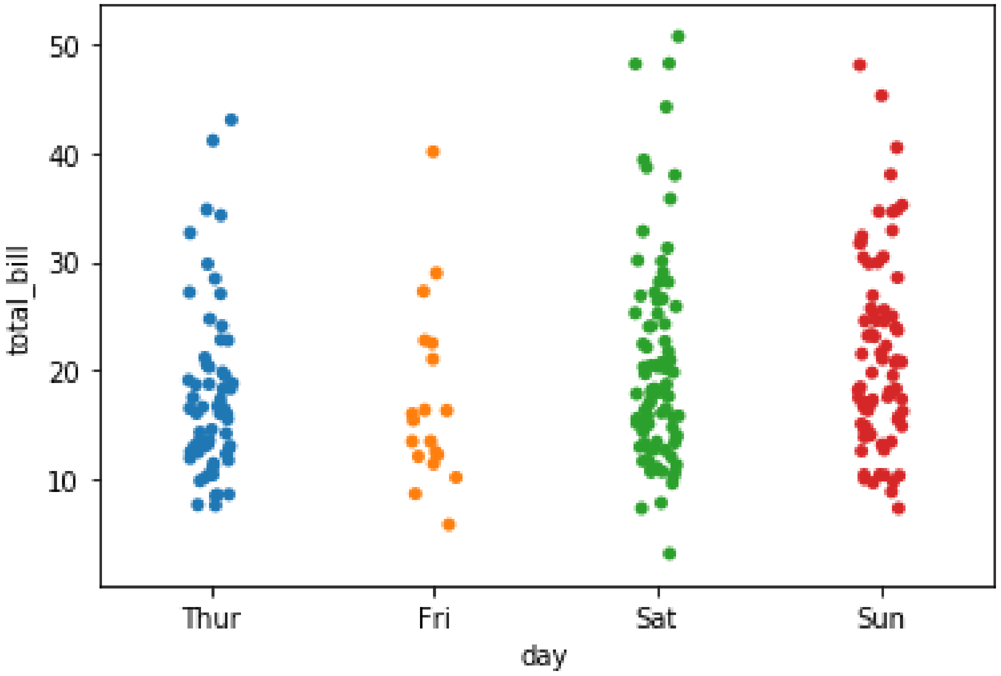


```
In [9]: tip.describe()
Out[9]:
       total_bill         tip        size
count  244.000000  244.000000  244.000000
mean    19.785943    2.998279    2.569672
std      8.902412    1.383638    0.951100
min      3.070000    1.000000    1.000000
25%     13.347500    2.000000    2.000000
50%     17.795000    2.900000    2.000000
75%     24.127500    3.562500    3.000000
max     50.810000   10.000000    6.000000
```

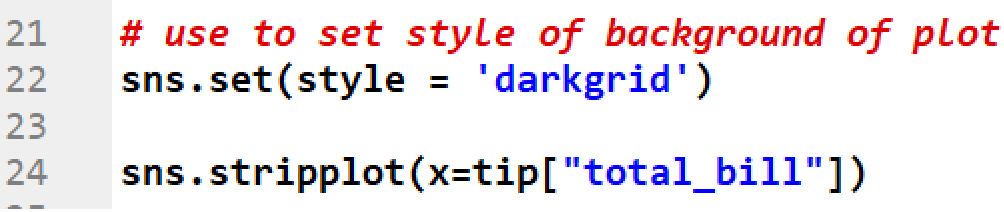
Strip Plot



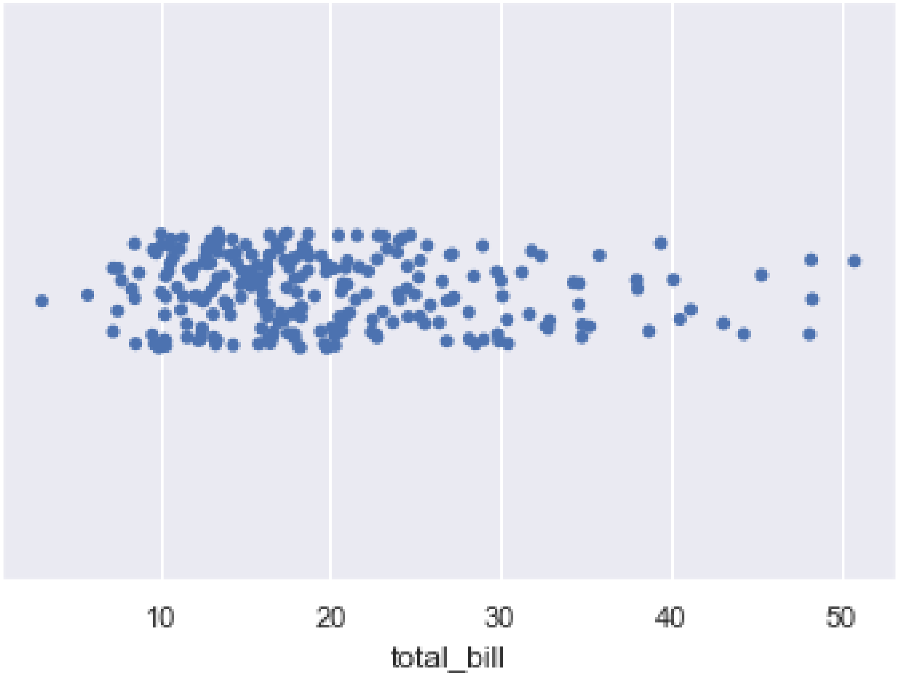
```
#____________strip plot

sns.stripplot(x="day", y="total_bill", data=tip)
```



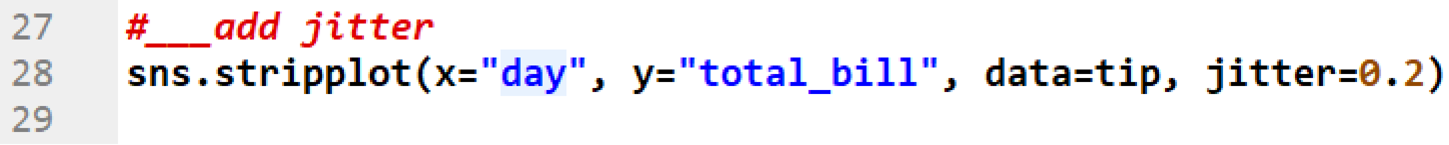
Strip Plot



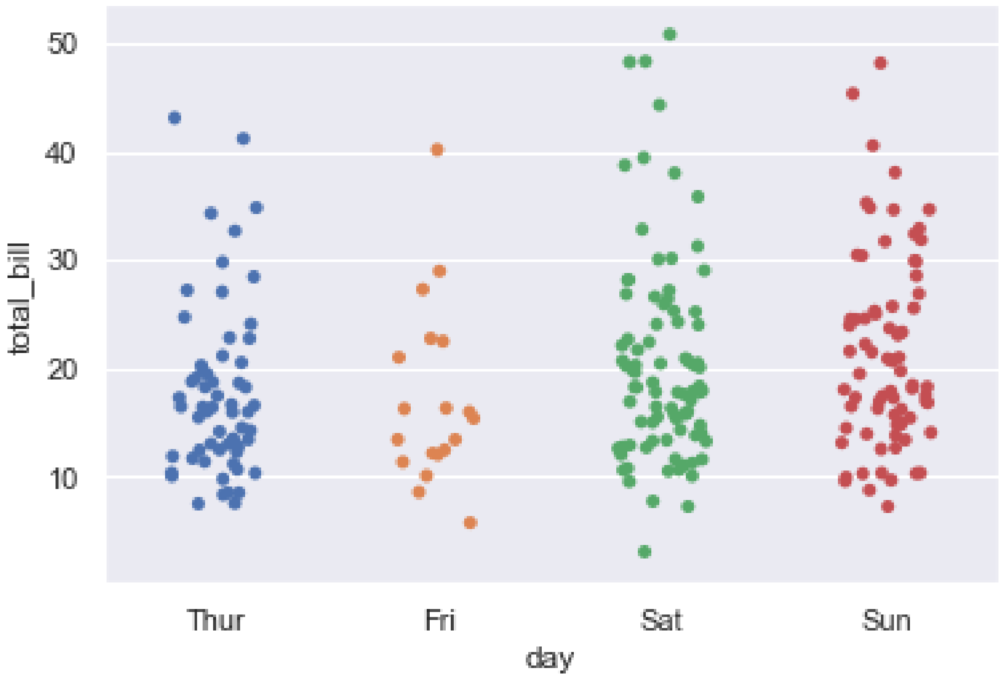
```
# use to set style of background of plot
sns.set(style = 'darkgrid')

sns.stripplot(x=tip["total_bill"])
```



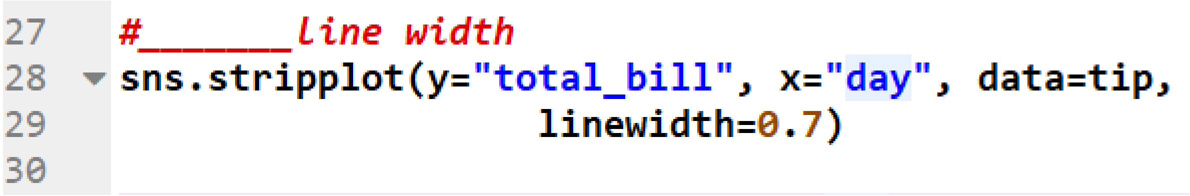
Strip Plot jitter



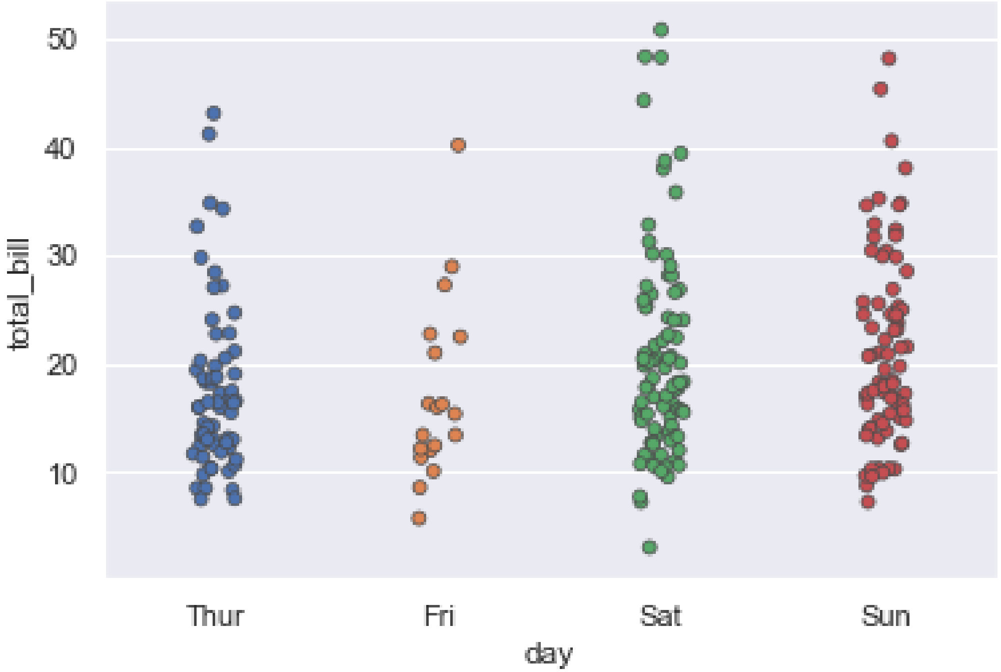
```
#___add jitter
sns.stripplot(x="day", y="total_bill", data=tip, jitter=0.2)

```



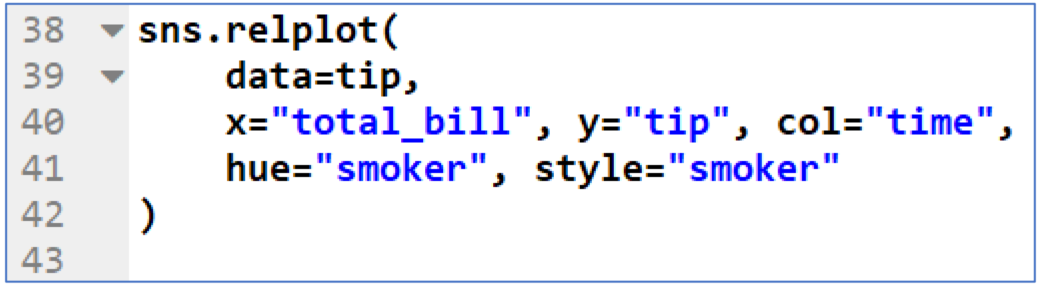
Strip Plot linewidth



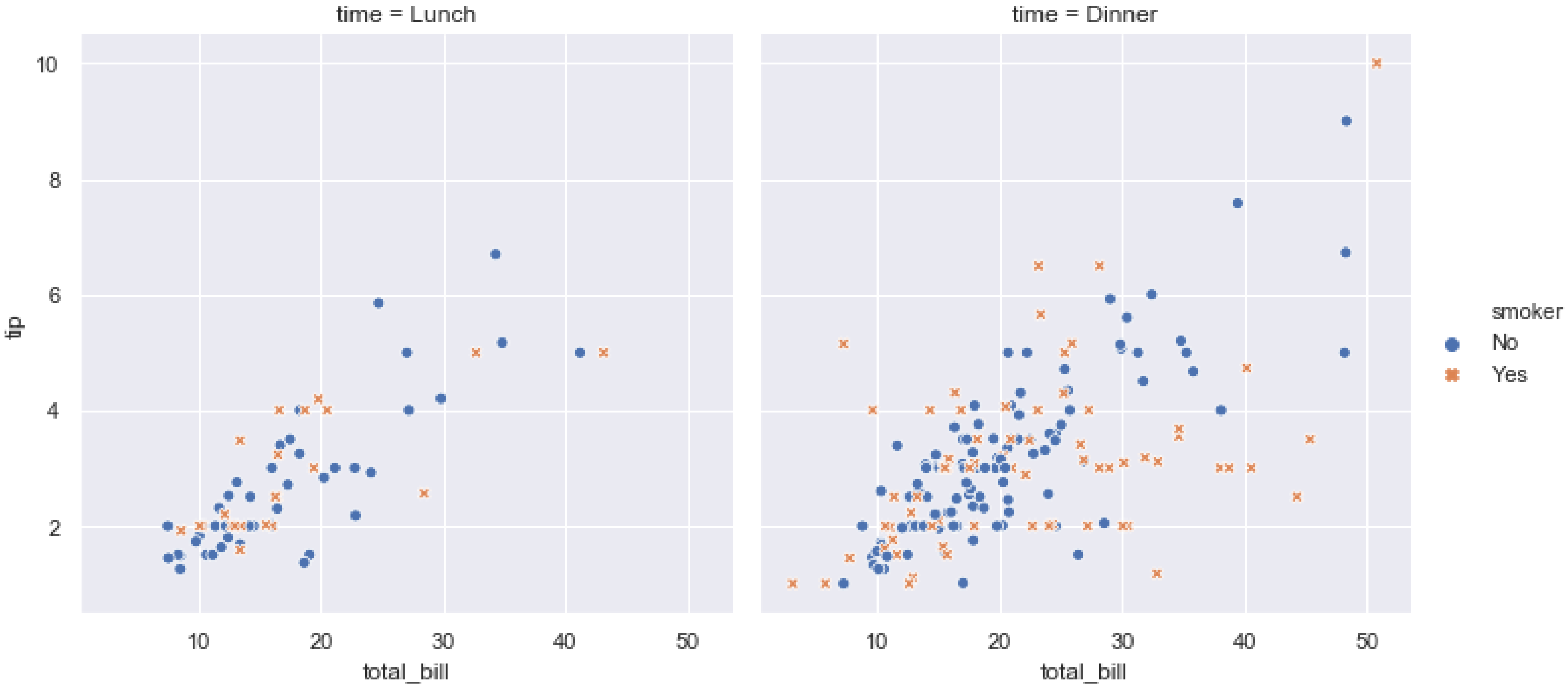
```
#______line width
sns.stripplot(y="total_bill", x="day", data=tip,
                   linewidth=0.7)

```



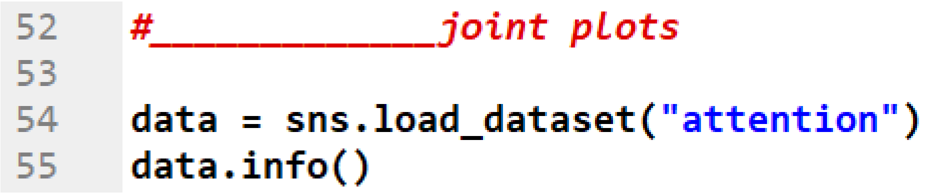
Relationship Plot



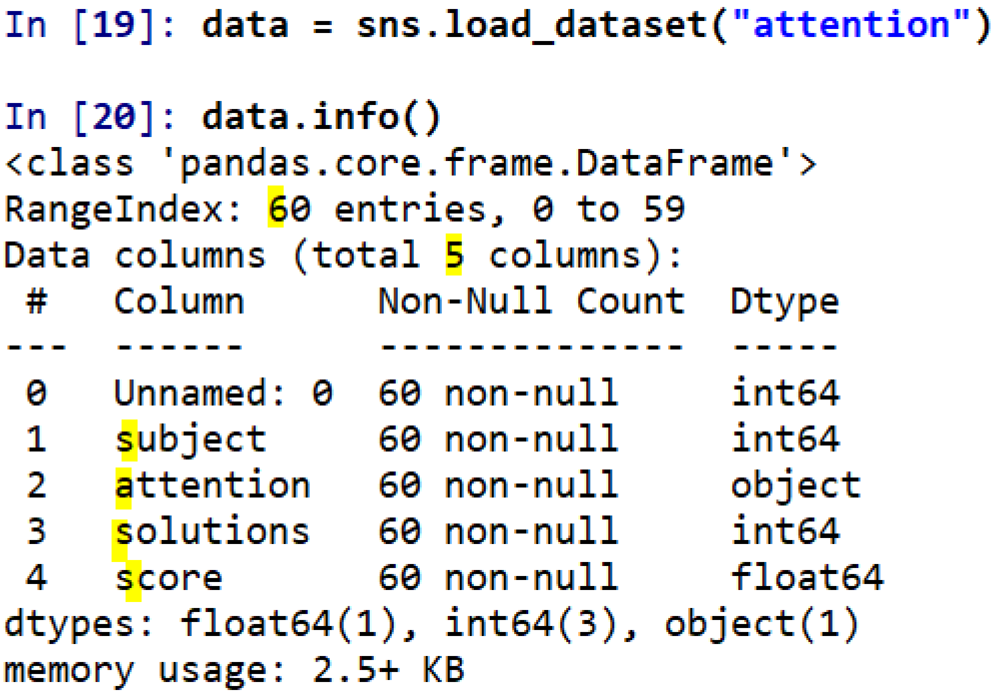
```
sns.relplot(
    data=tip,
    x="total_bill", y="tip", col="time",
    hue="smoker", style="smoker"
)
```



Joint Plots

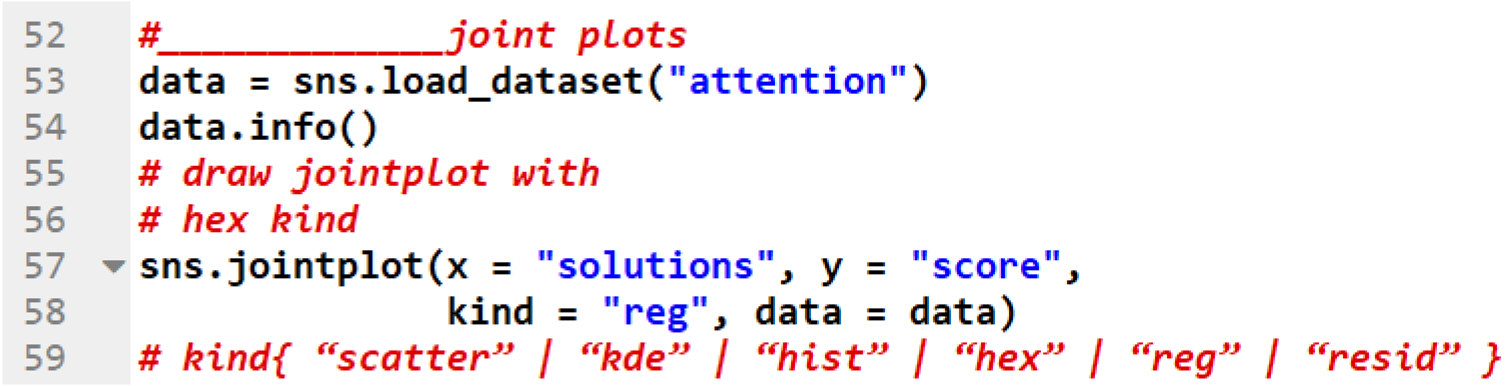


```
#____________joint plots

data = sns.load_dataset("attention")
data.info()
```

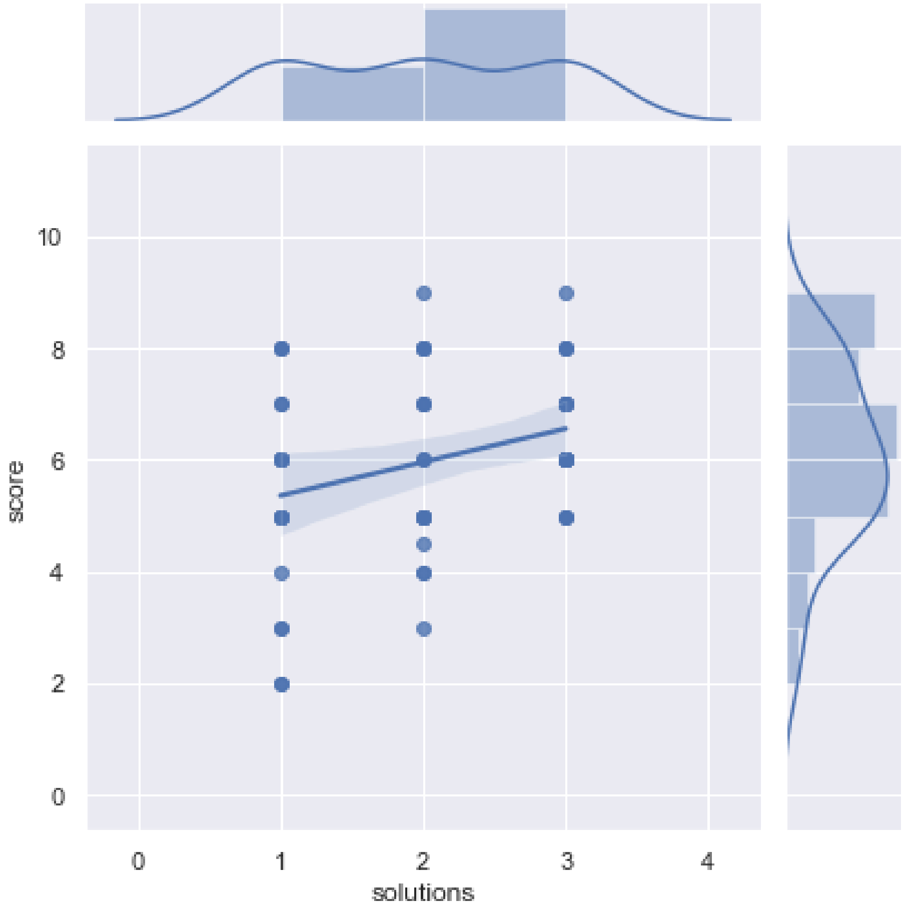


```
In [19]: data = sns.load_dataset("attention")

In [20]: data.info()
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 60 entries, 0 to 59
Data columns (total 5 columns):
 #   Column       Non-Null Count  Dtype  
---  ------       --------------  -----  
 0   Unnamed: 0   60 non-null     int64  
 1   subject      60 non-null     int64  
 2   attention    60 non-null     object 
 3   solutions    60 non-null     int64  
 4   score        60 non-null     float64
dtypes: float64(1), int64(3), object(1)
memory usage: 2.5+ KB
```

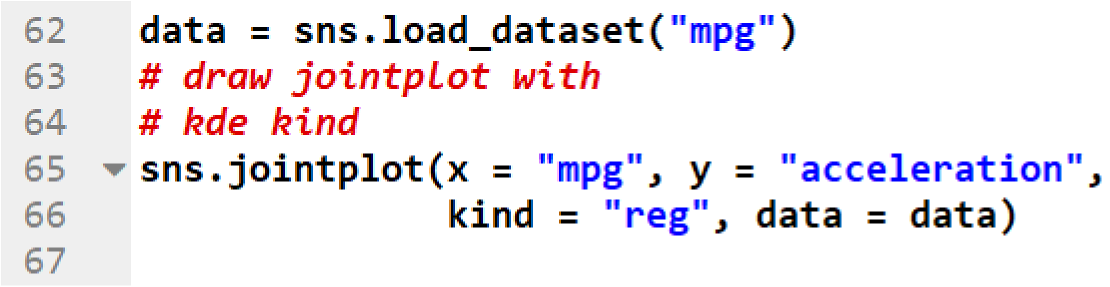
Joint Plots



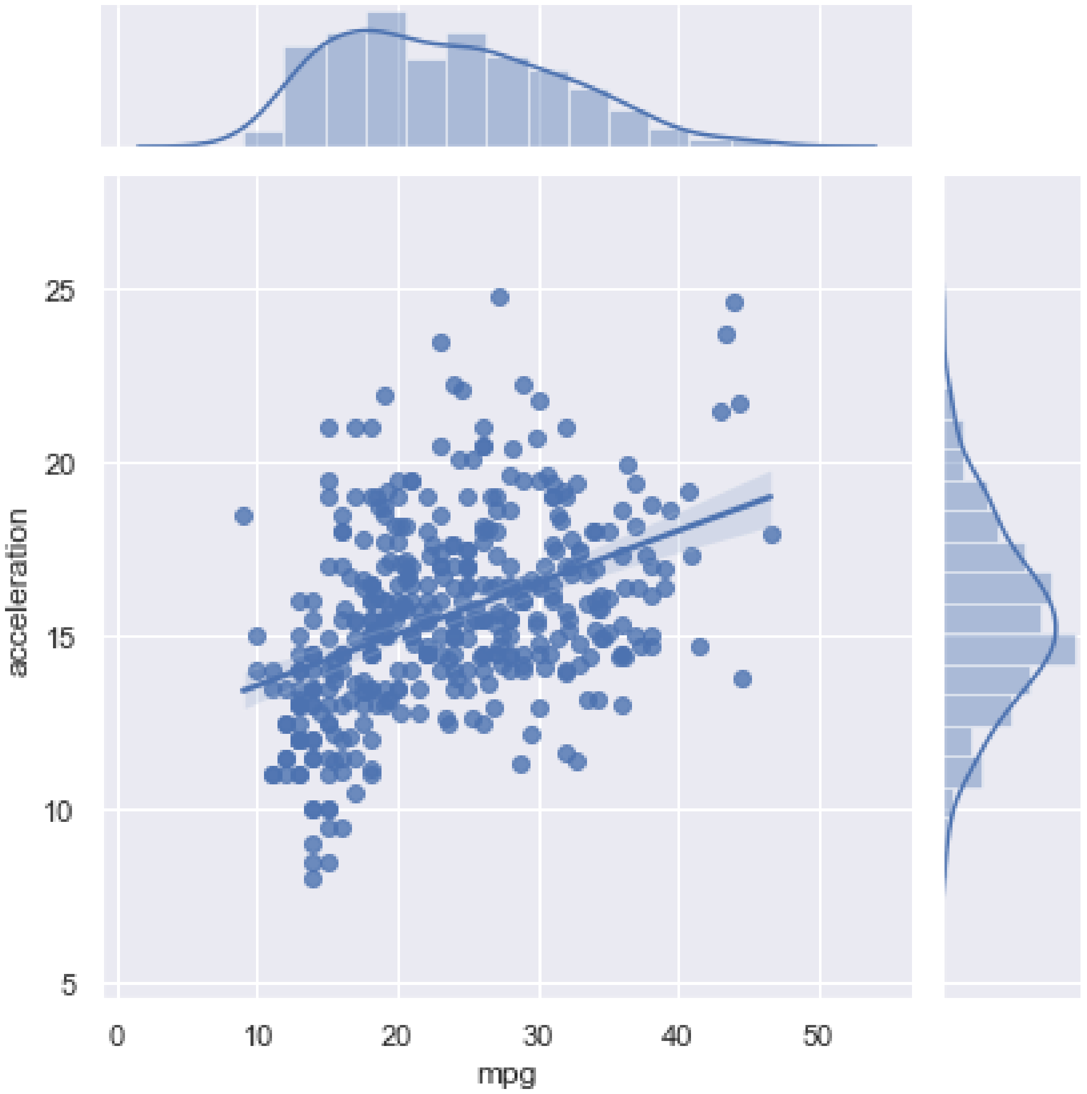
```
#_____________joint plots
data = sns.load_dataset("attention")
data.info()
# draw jointplot with
# hex kind
sns.jointplot(x = "solutions", y = "score",
              kind = "reg", data = data)
# kind{ "scatter" | "kde" | "hist" | "hex" | "reg" | "resid" }
```



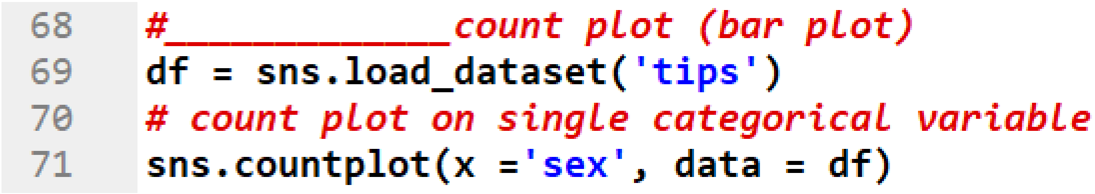
Joint Plot



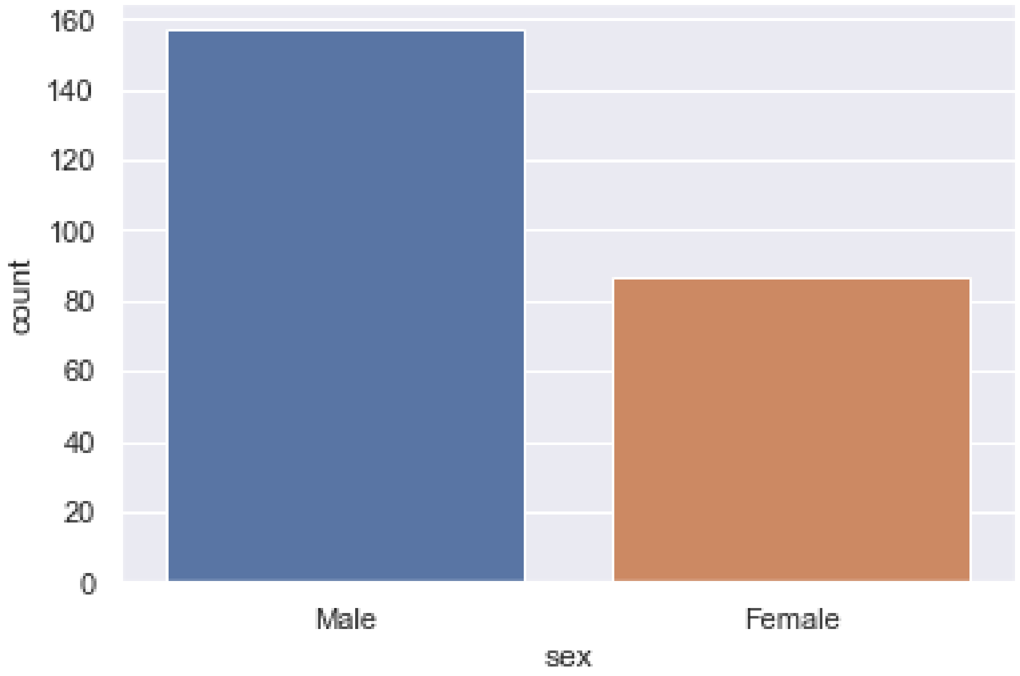
```
data = sns.load_dataset("mpg")
# draw jointplot with
# kde kind
sns.jointplot(x = "mpg", y = "acceleration",
              kind = "reg", data = data)

```



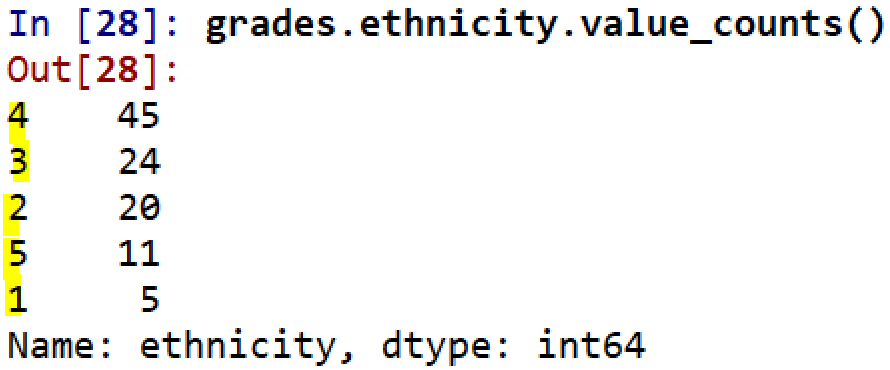
Count Plots/Bar Plots



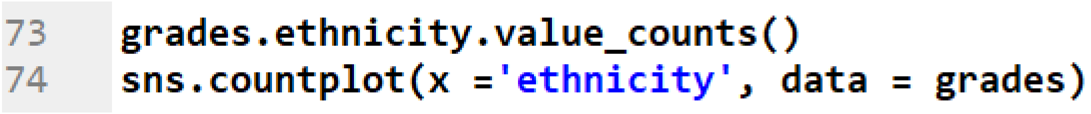
```
#_____________count plot (bar plot)
df = sns.load_dataset('tips')
# count plot on single categorical variable
sns.countplot(x ='sex', data = df)
```



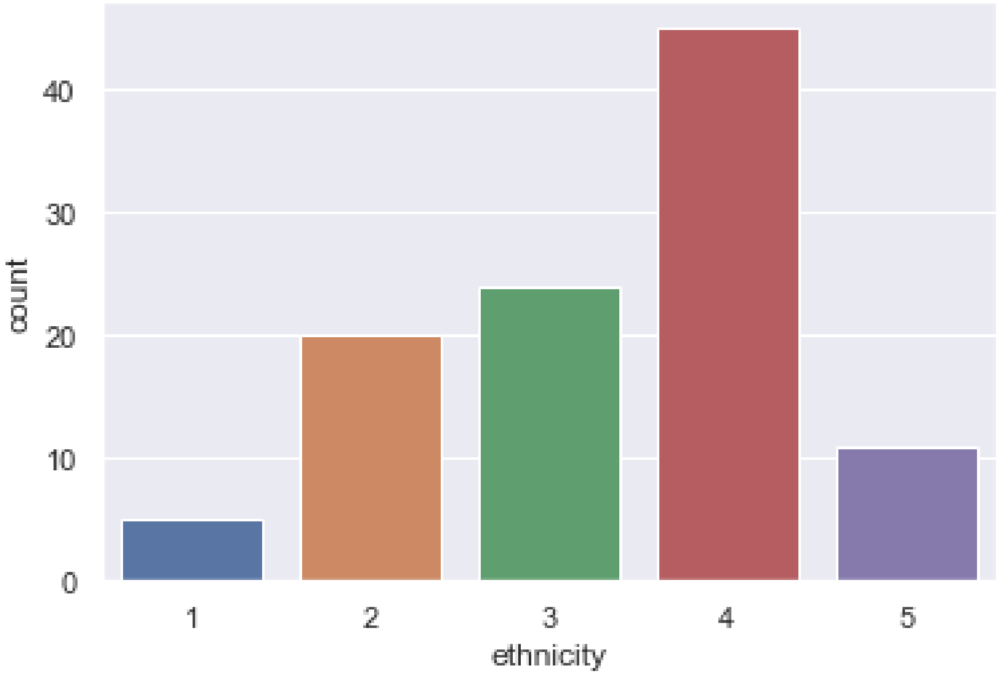
Count Plots/Bar Plots



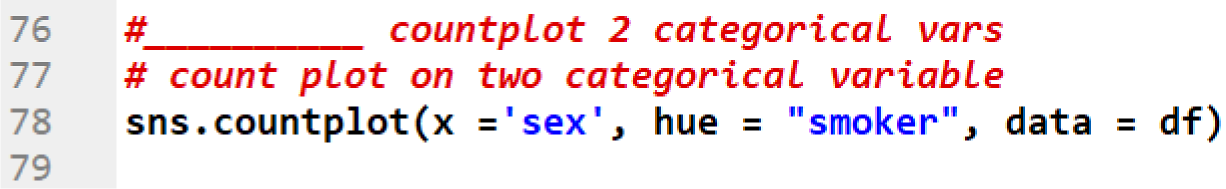
```
In [28]: grades.ethnicity.value_counts()
Out[28]:
4    45
3    24
2    20
5    11
1     5
Name: ethnicity, dtype: int64
```



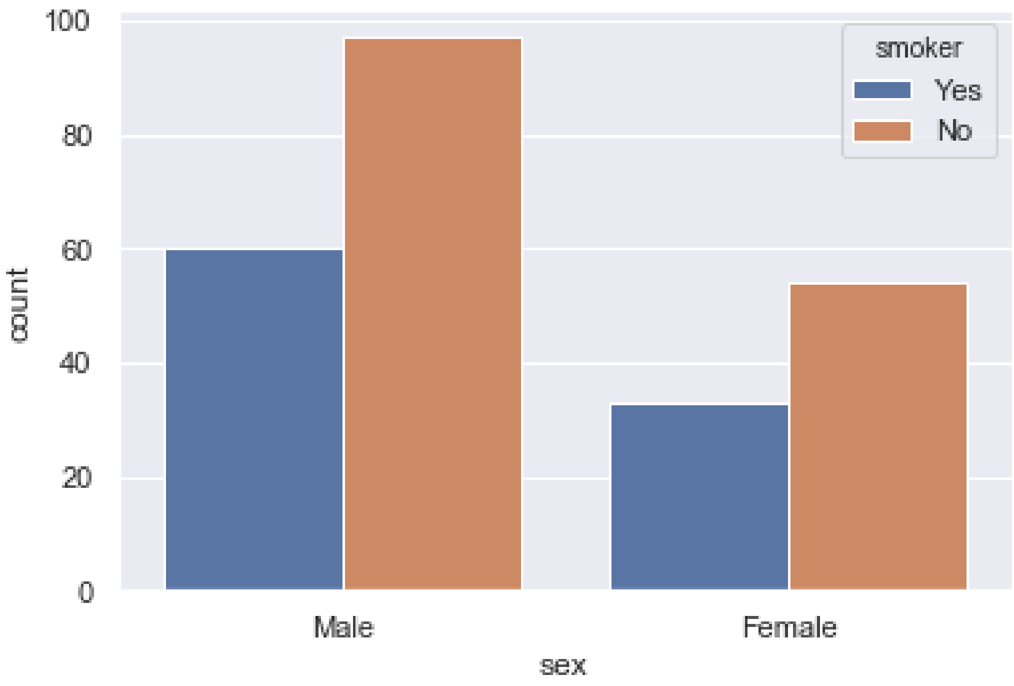
```
grades.ethnicity.value_counts()
sns.countplot(x ='ethnicity', data = grades)
```



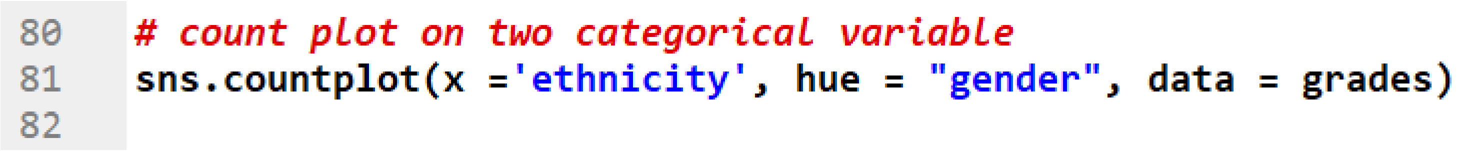
Count Plot/Grouped by



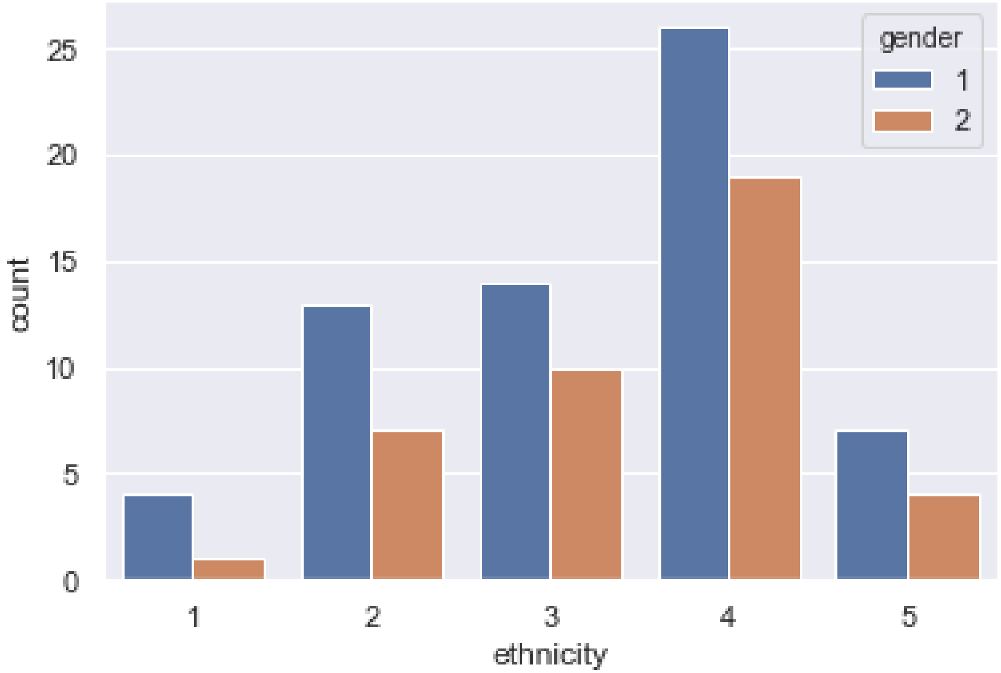
```
#__________ countplot 2 categorical vars
# count plot on two categorical variable
sns.countplot(x ='sex', hue = "smoker", data = df)

```



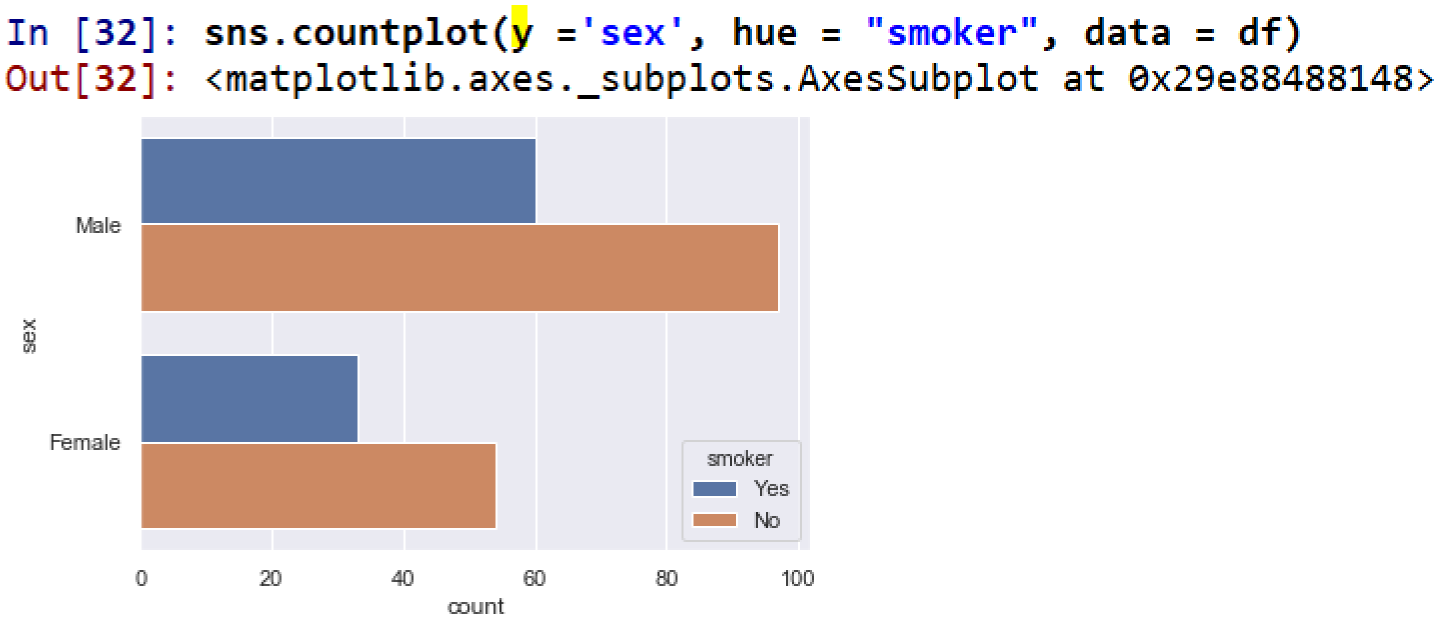
Count Plot/Grouped by



```
# count plot on two categorical variable
sns.countplot(x ='ethnicity', hue = "gender", data = grades)
```

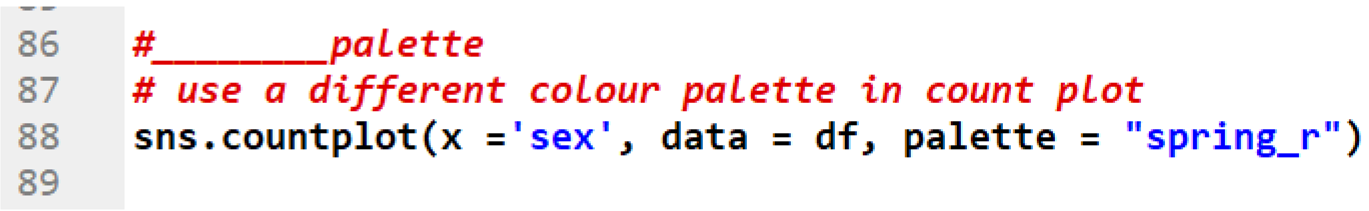


Change orienttion

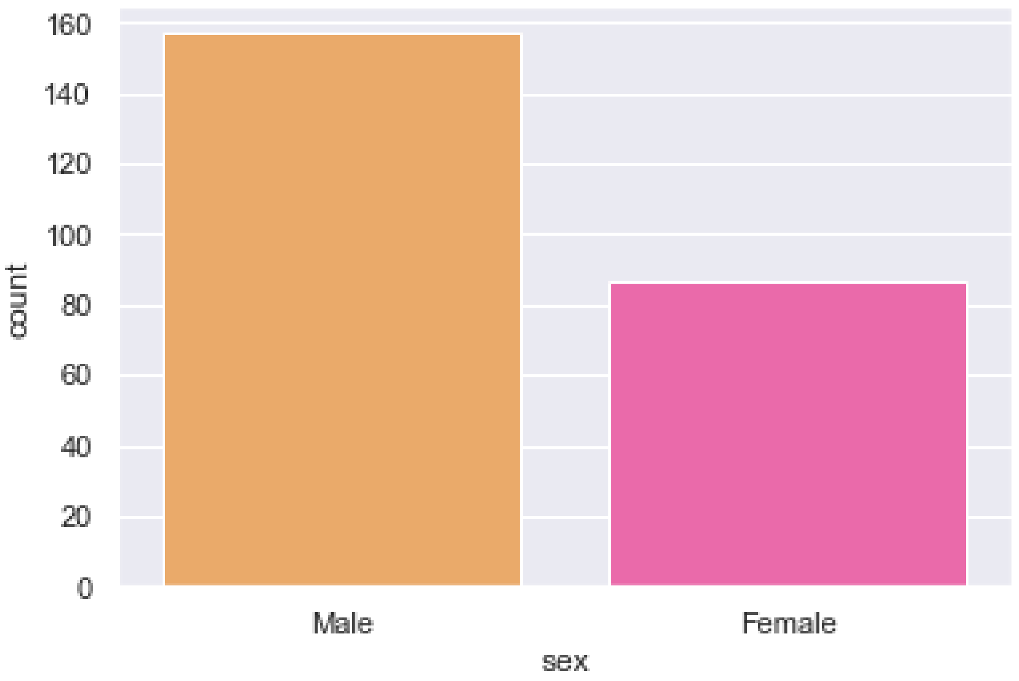


```
In [32]: sns.countplot(y ='sex', hue = "smoker", data = df)
Out[32]: <matplotlib.axes._subplots.AxesSubplot at 0x29e88488148>
```

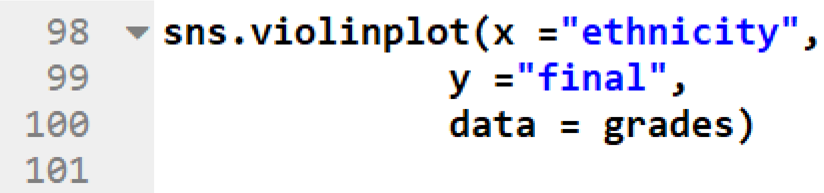
Change palette



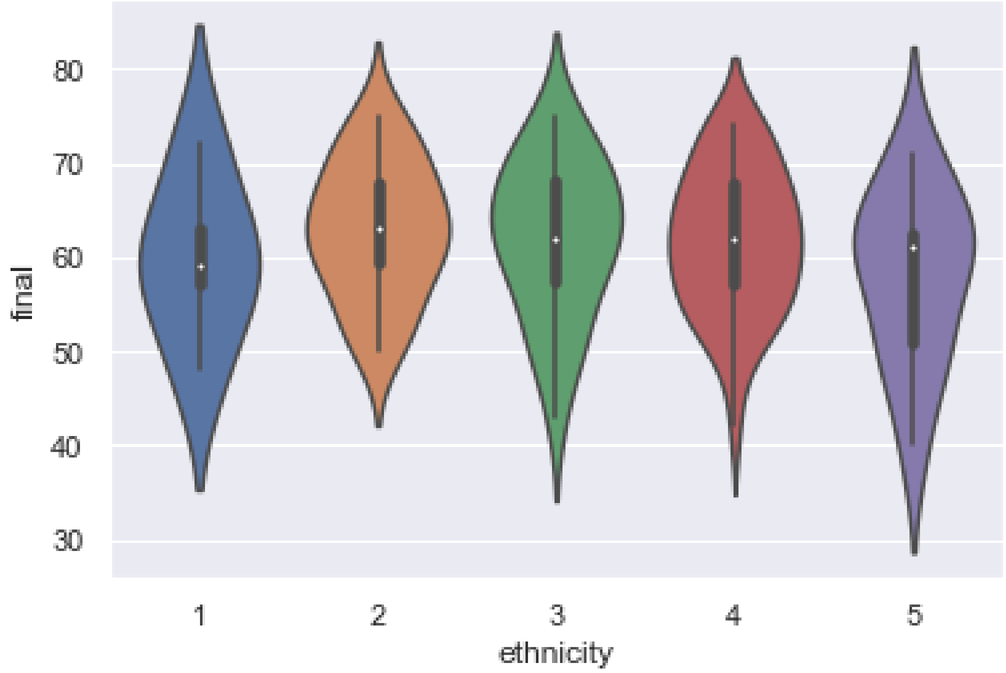
```
#________palette
# use a different colour palette in count plot
sns.countplot(x ='sex', data = df, palette = "spring_r")
```



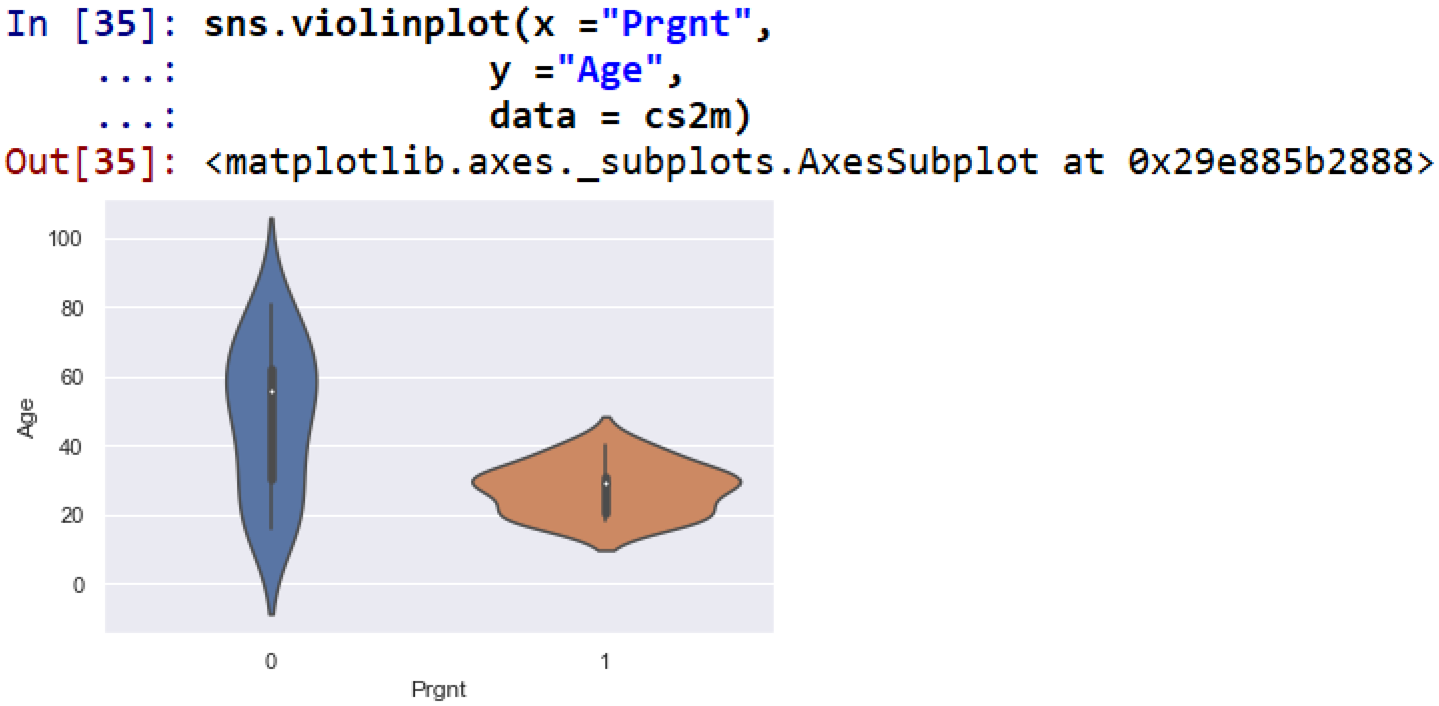
Box Plot/Violin Plot



```
sns.violinplot(x ="ethnicity",
               y ="final",
               data = grades)
```

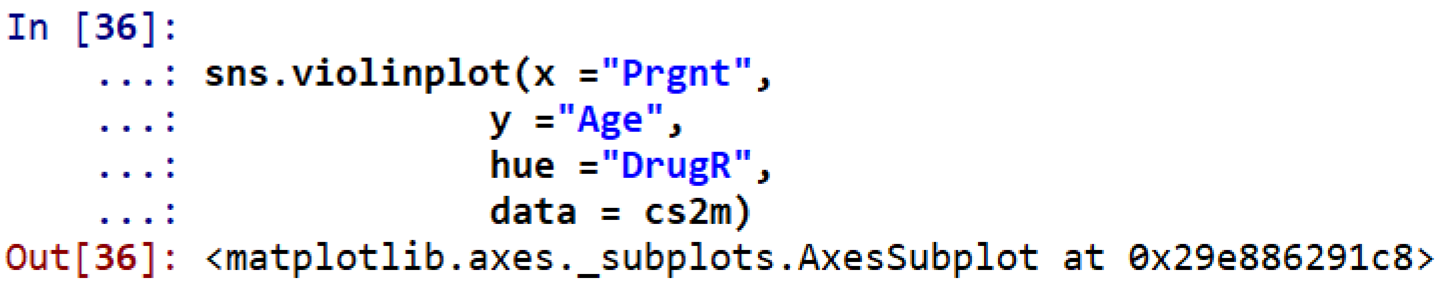


Box Plot/Violin Plot

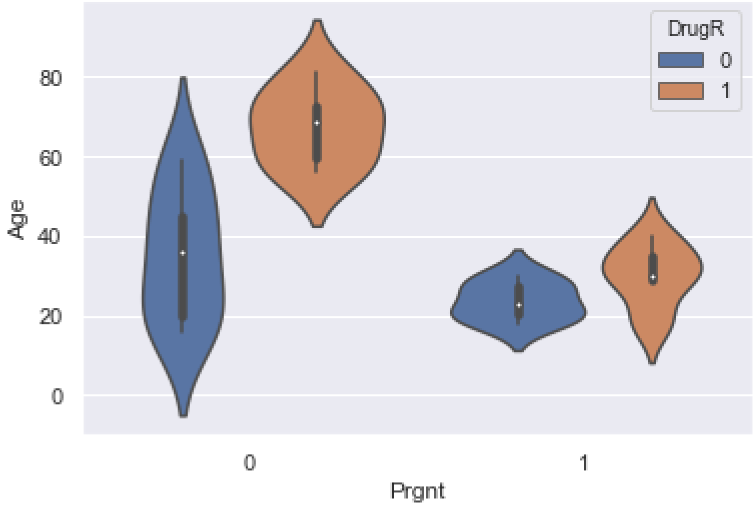


```
In [35]: sns.violinplot(x ="Prgnt",
    ...:                y ="Age",
    ...:                data = cs2m)
Out[35]: <matplotlib.axes._subplots.AxesSubplot at 0x29e885b2888>
```

Box Plot/Violin Plot

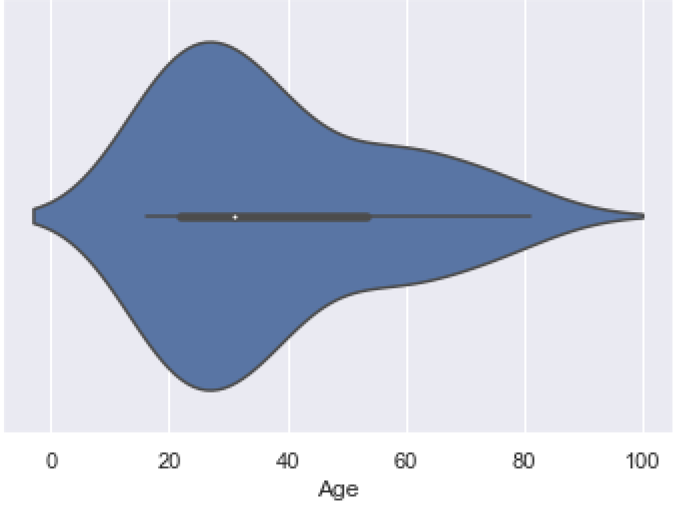


```
In [36]:
    ...: sns.violinplot(x ="Prgnt",
    ...:                y ="Age",
    ...:                hue ="DrugR",
    ...:                data = cs2m)
Out[36]: <matplotlib.axes._subplots.AxesSubplot at 0x29e886291c8>
```

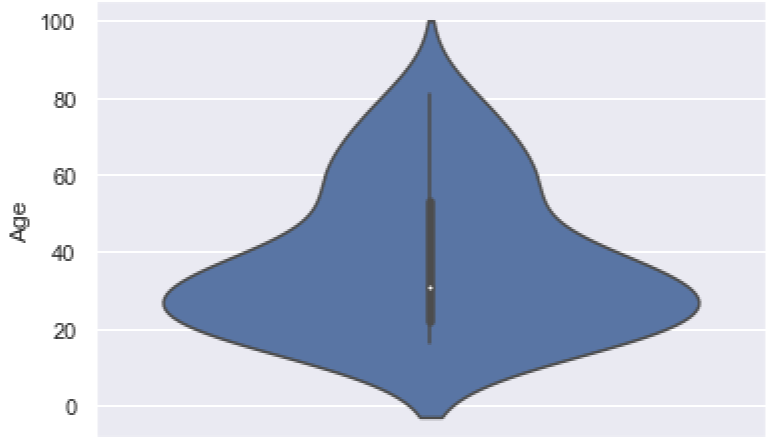


Box Plot/Violin Plot; Changing Axis

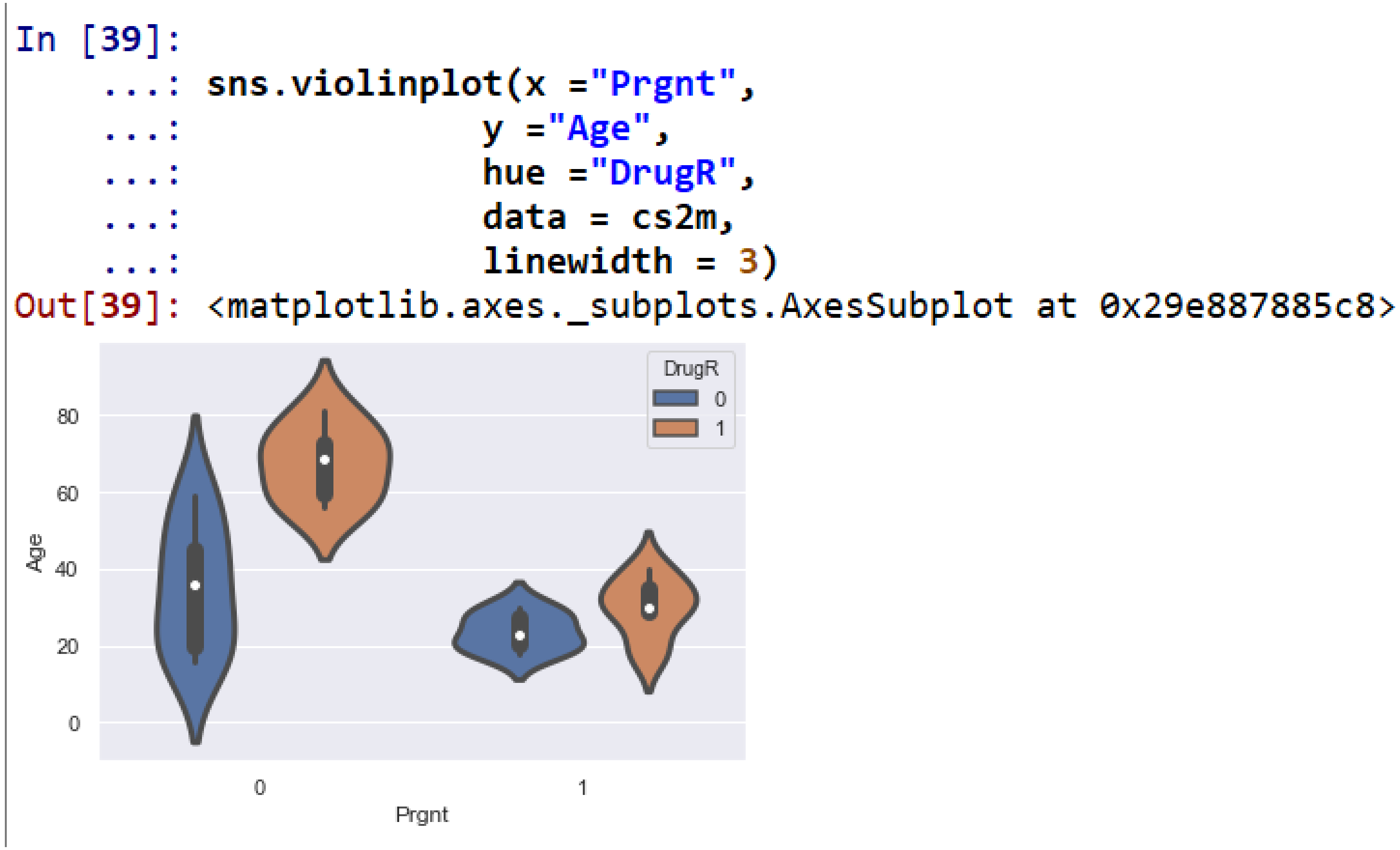
```
In [37]: sns.violinplot(x=cs2m["Age"]) # horizontal
Out[37]: <matplotlib.axes._subplots.AxesSubplot at 0x29e886ca8c8>
```



```
In [38]: sns.violinplot(y=cs2m["Age"]) # vertical
Out[38]: <matplotlib.axes._subplots.AxesSubplot at 0x29e887148c8>
```

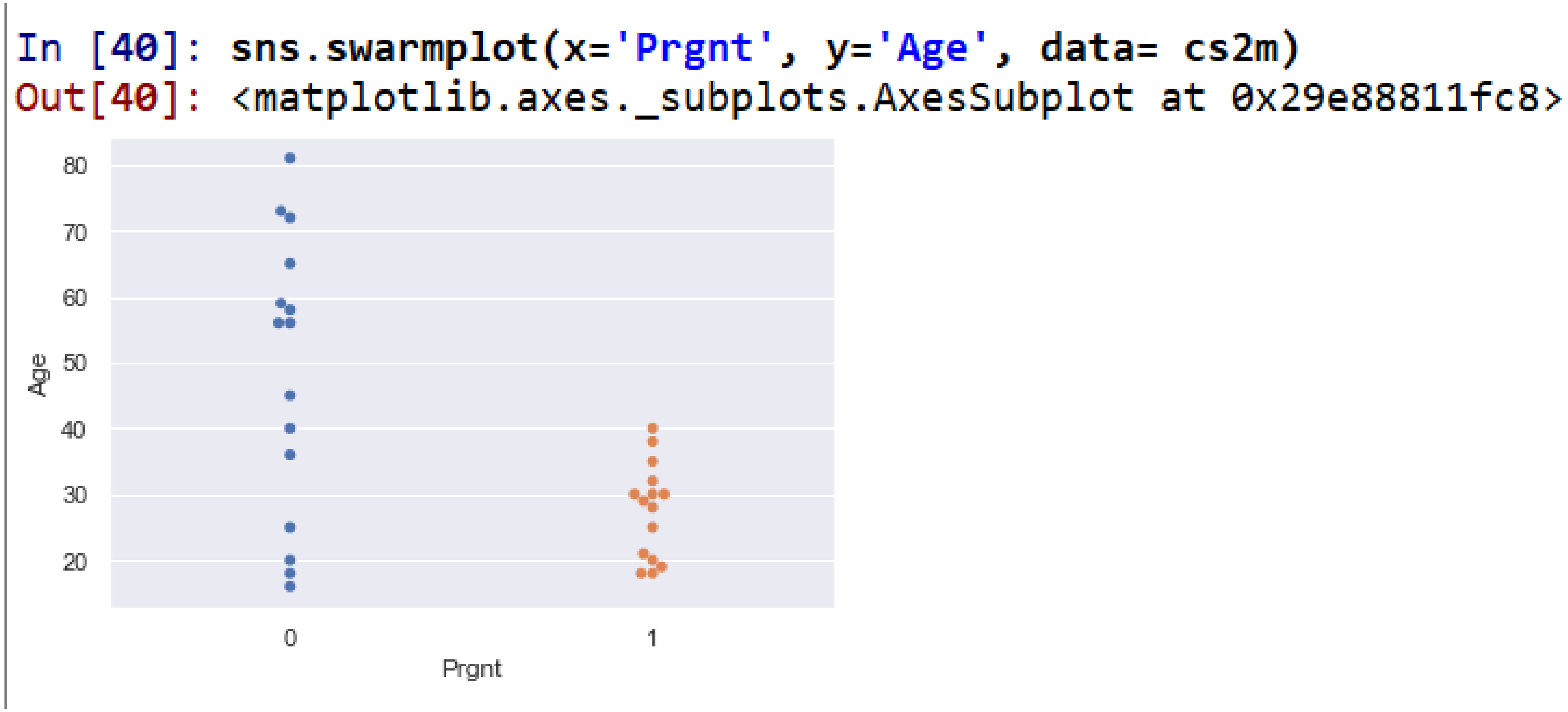


Box Plot/Violin Plot; Linewidth



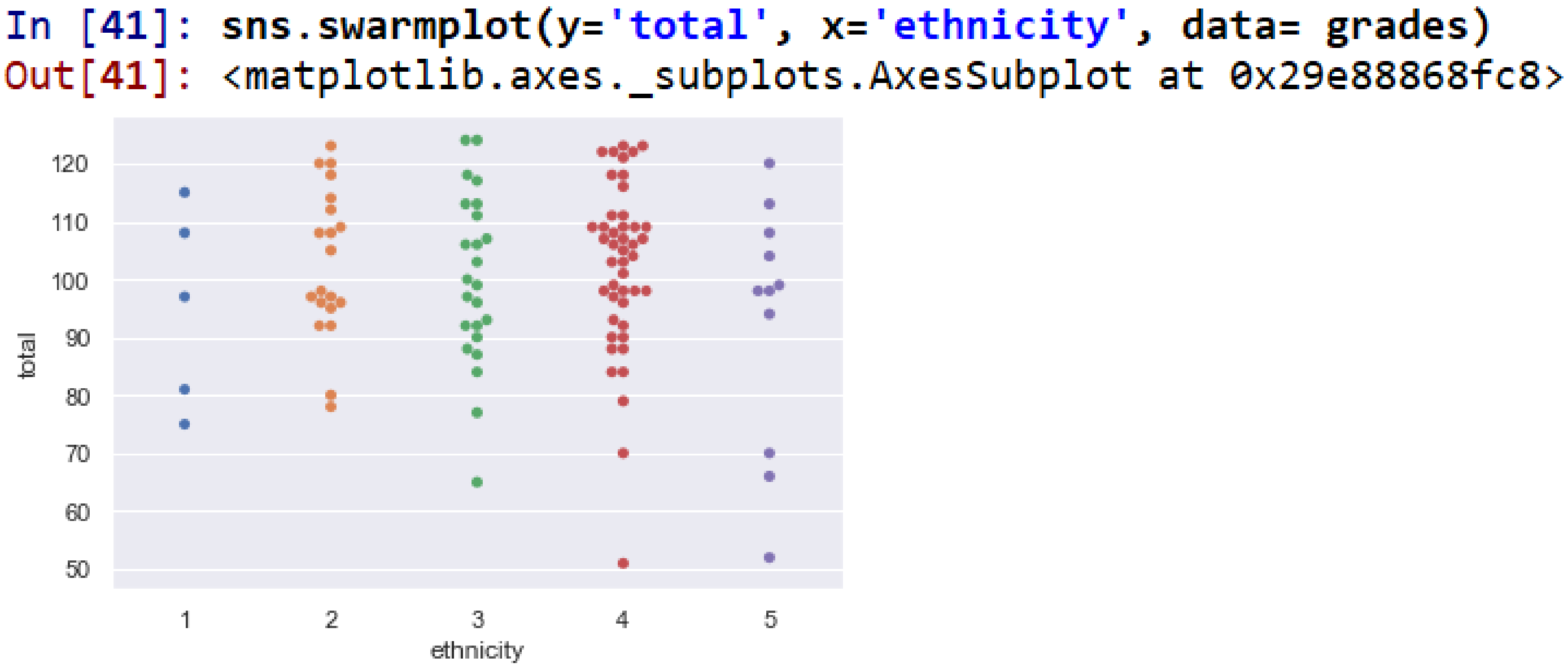
```
In [39]:
    ...: sns.violinplot(x ="Prgnt",
    ...:                y ="Age",
    ...:                hue ="DrugR",
    ...:                data = cs2m,
    ...:                linewidth = 3)
Out[39]: <matplotlib.axes._subplots.AxesSubplot at 0x29e887885c8>
```

Swarmplot



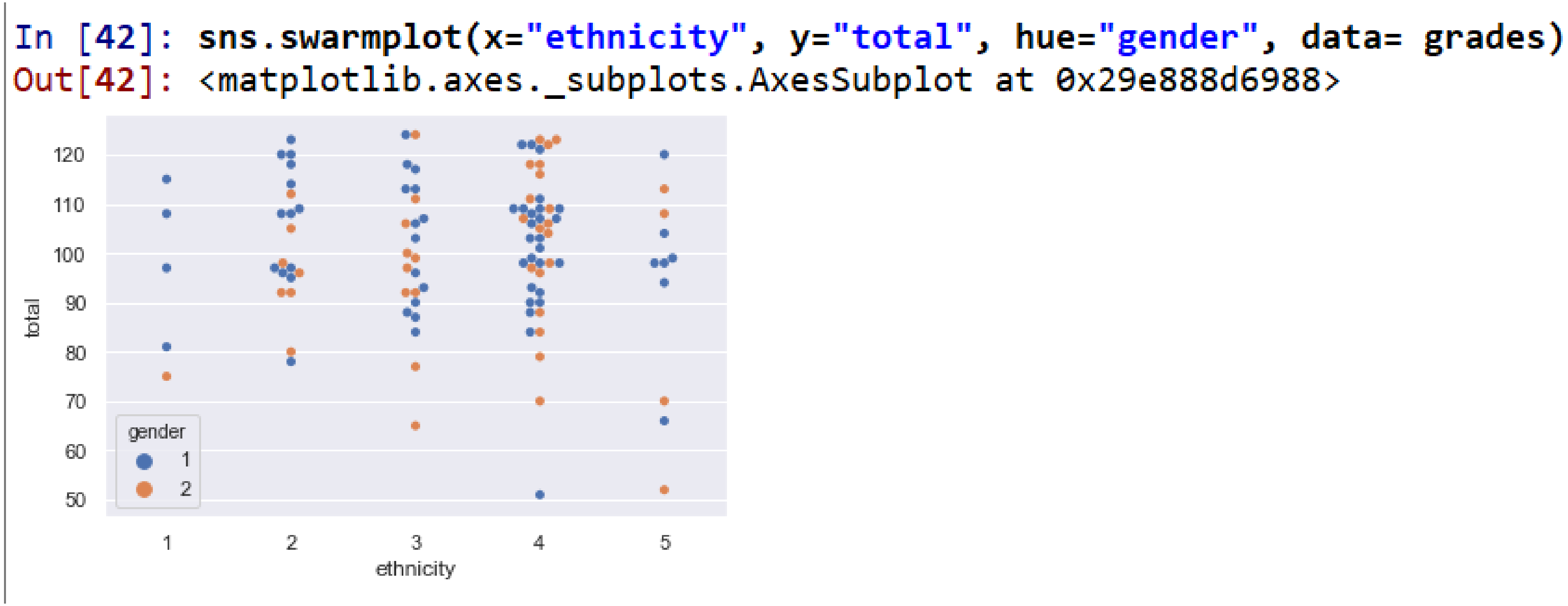
```
In [40]: sns.swarmplot(x='Prgnt', y='Age', data= cs2m)
Out[40]: <matplotlib.axes._subplots.AxesSubplot at 0x29e88811fc8>
```

Swarmplot

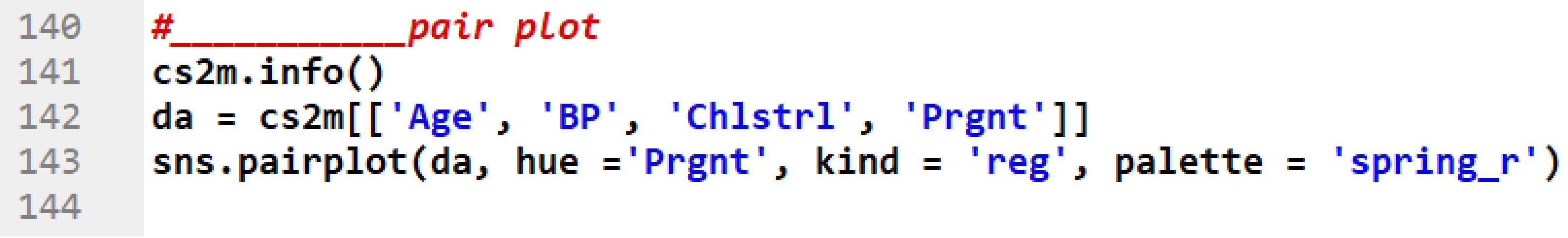


```
In [41]: sns.swarmplot(y='total', x='ethnicity', data= grades)
Out[41]: <matplotlib.axes._subplots.AxesSubplot at 0x29e88868fc8>
```

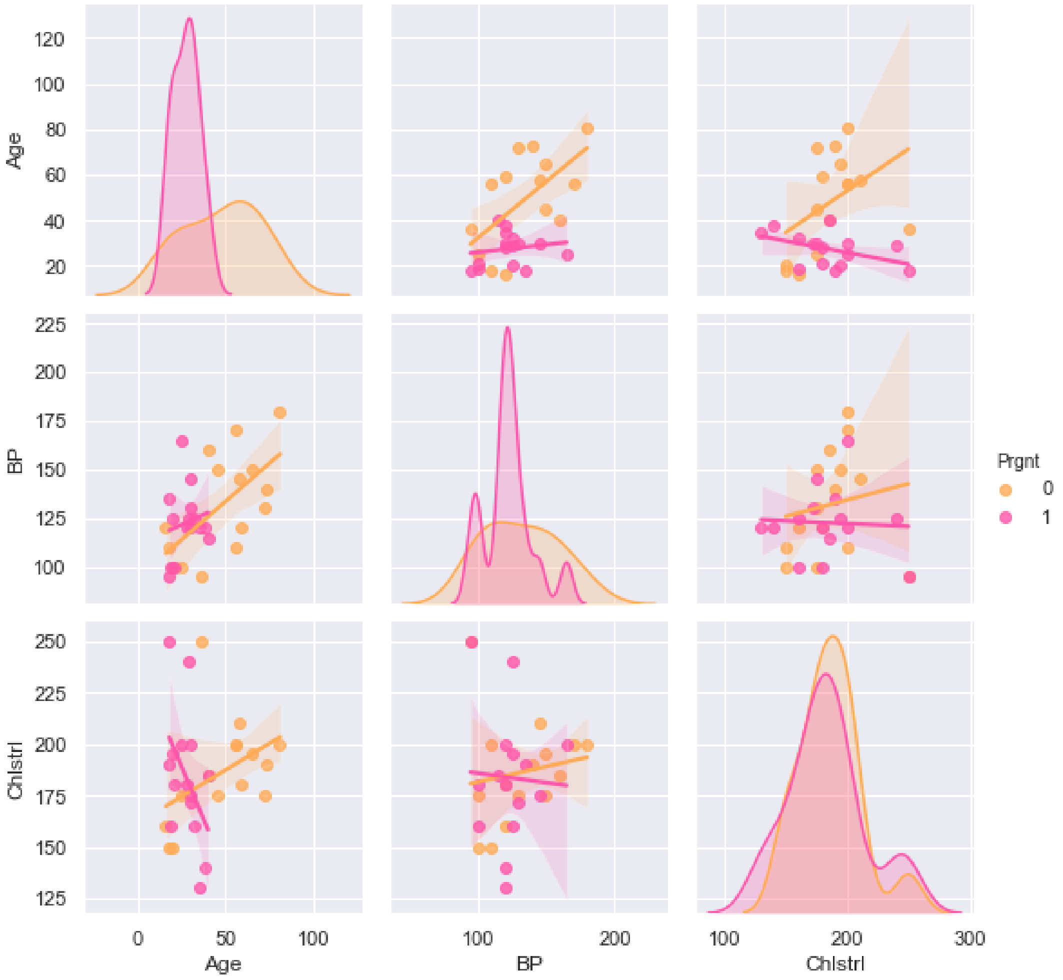
Swarmplot

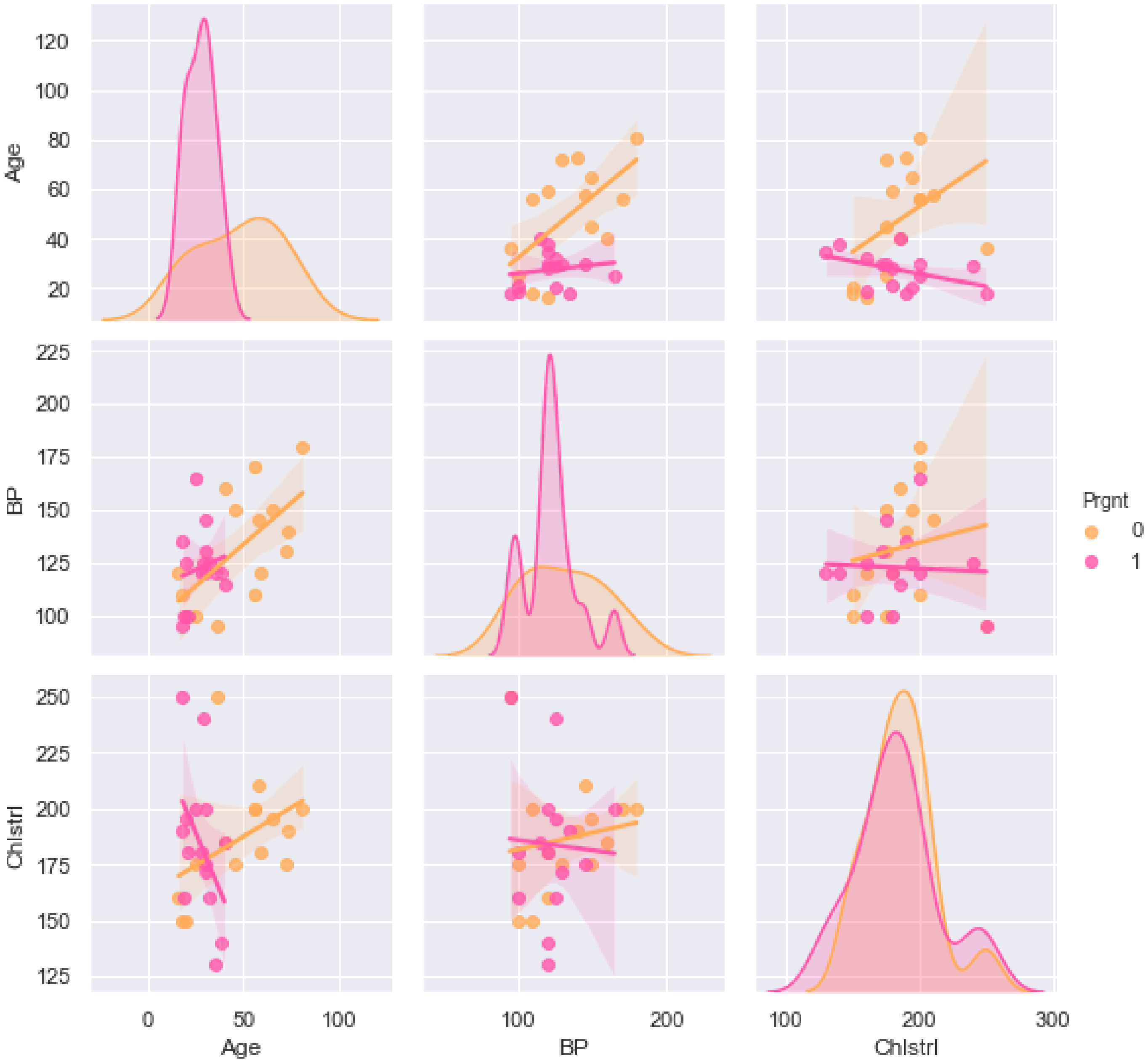


```
In [42]: sns.swarmplot(x="ethnicity", y="total", hue="gender", data= grades)
Out[42]: <matplotlib.axes._subplots.AxesSubplot at 0x29e888d6988>
```

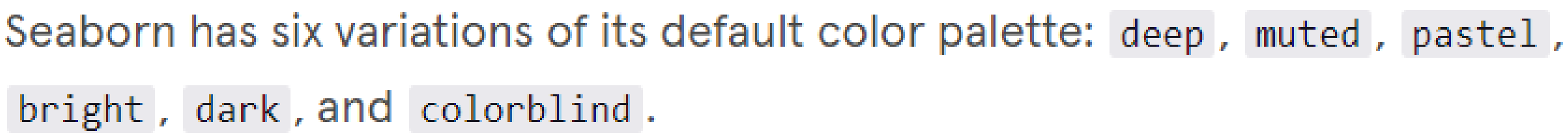


```python
#__________pair plot
cs2m.info()
da = cs2m[['Age', 'BP', 'Chlstrl', 'Prgnt']]
sns.pairplot(da, hue ='Prgnt', kind = 'reg', palette = 'spring_r')
```

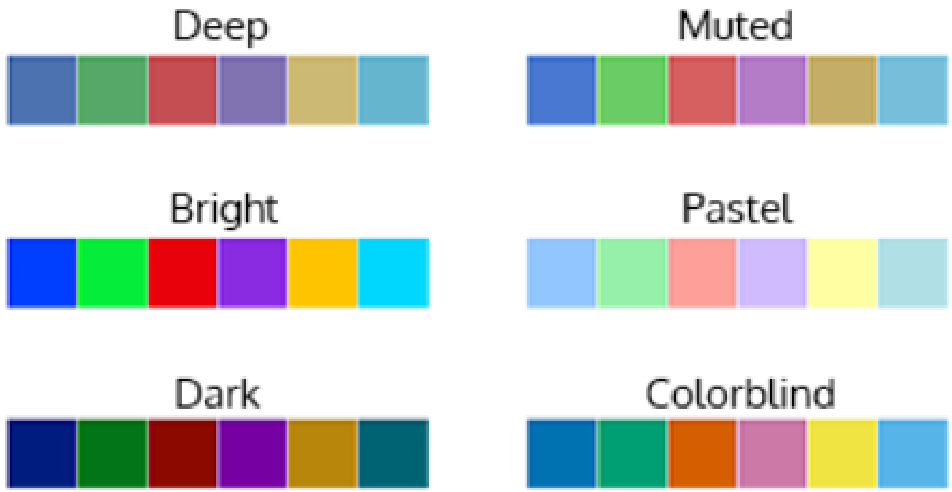



Age
120
100
80
60
40
20
BP
225
200
175
150
125
100
Chlstrl
250
225
200
175
150
125
0
50
100
Age
100
200
BP
100
200
300
Chlstrl
Prgnt
0
1

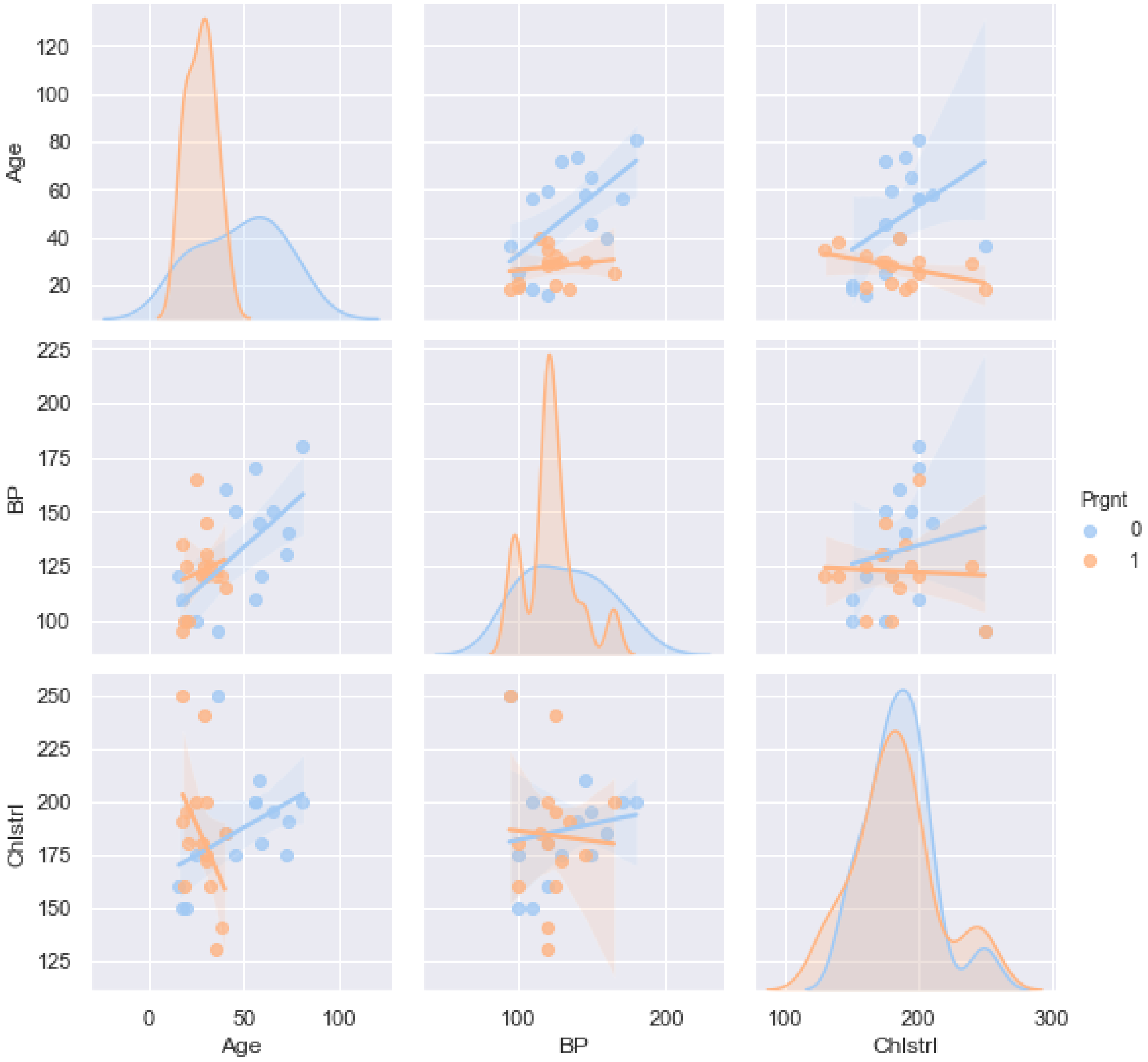
Palette



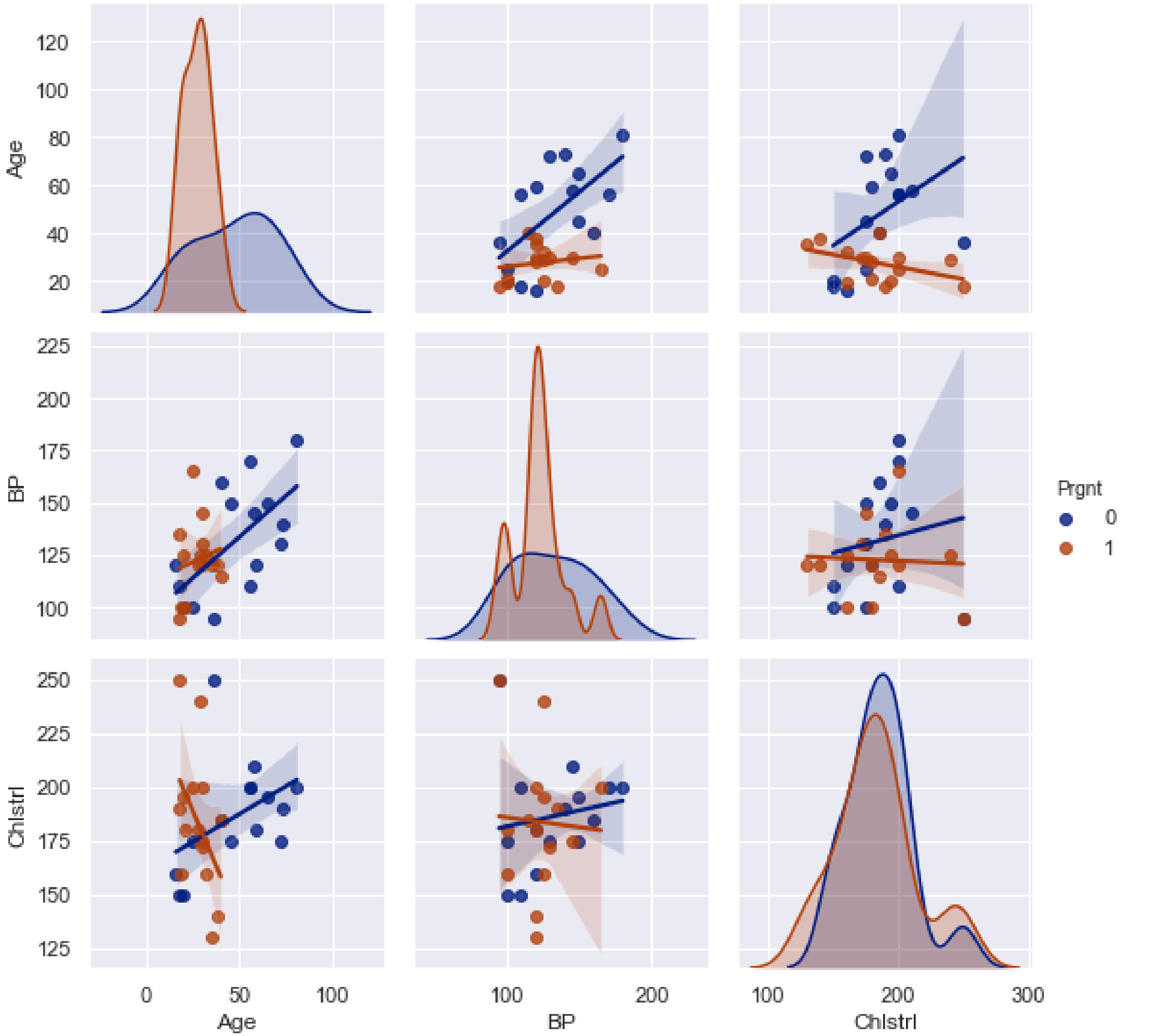
Seaborn has six variations of its default color palette: `deep`, `muted`, `pastel`, `bright`, `dark`, and `colorblind`.



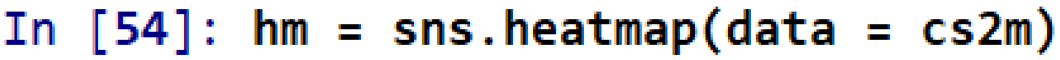
Palette ‘pastel’



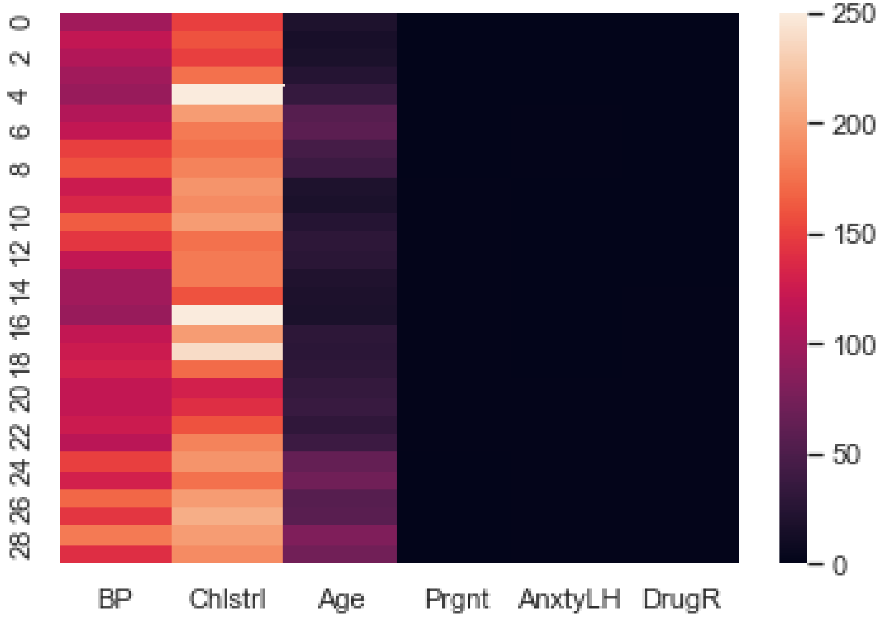
Palette ‘dark’



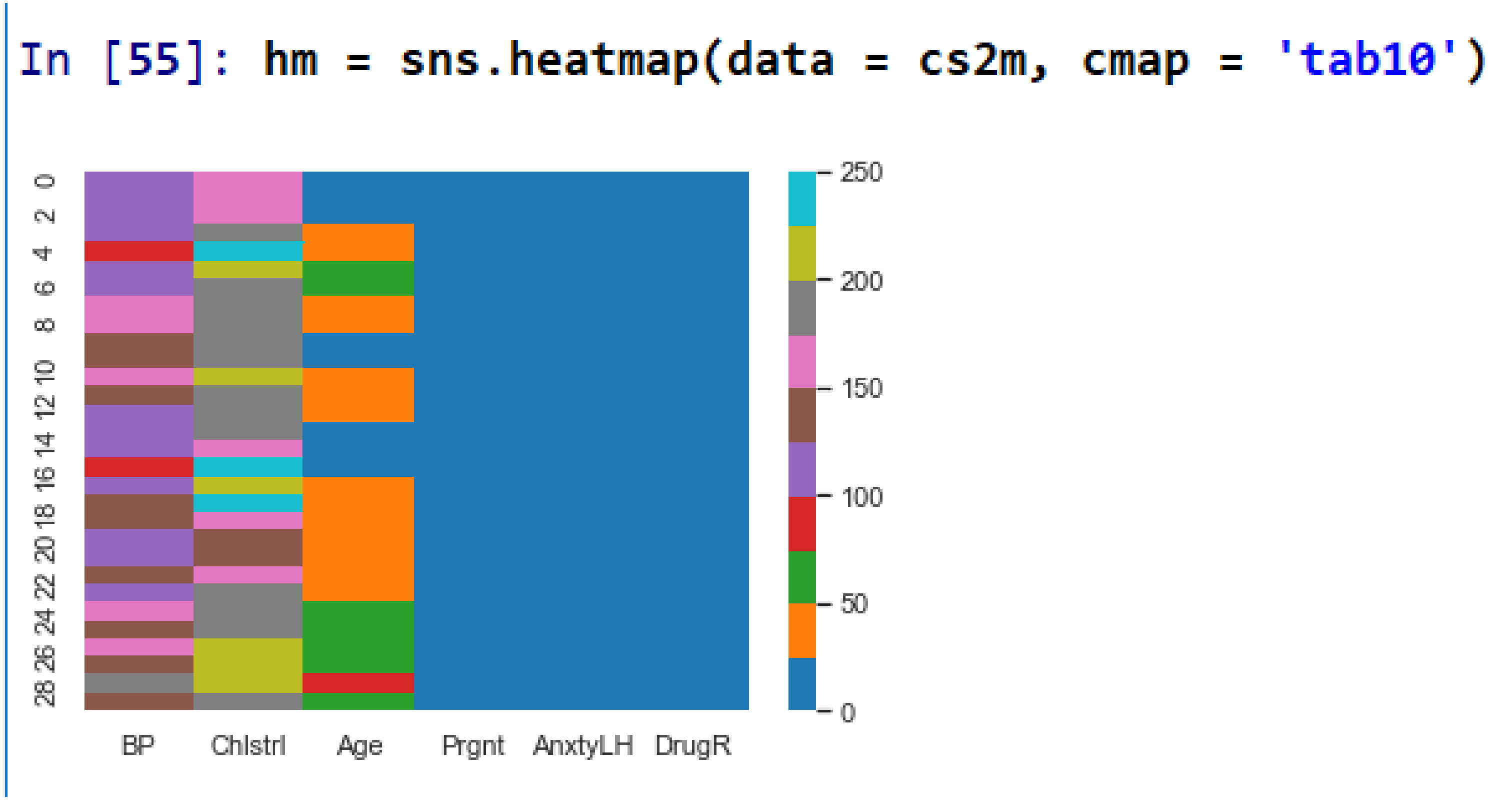
Heat Maps



```
In [54]: hm = sns.heatmap(data = cs2m)
```

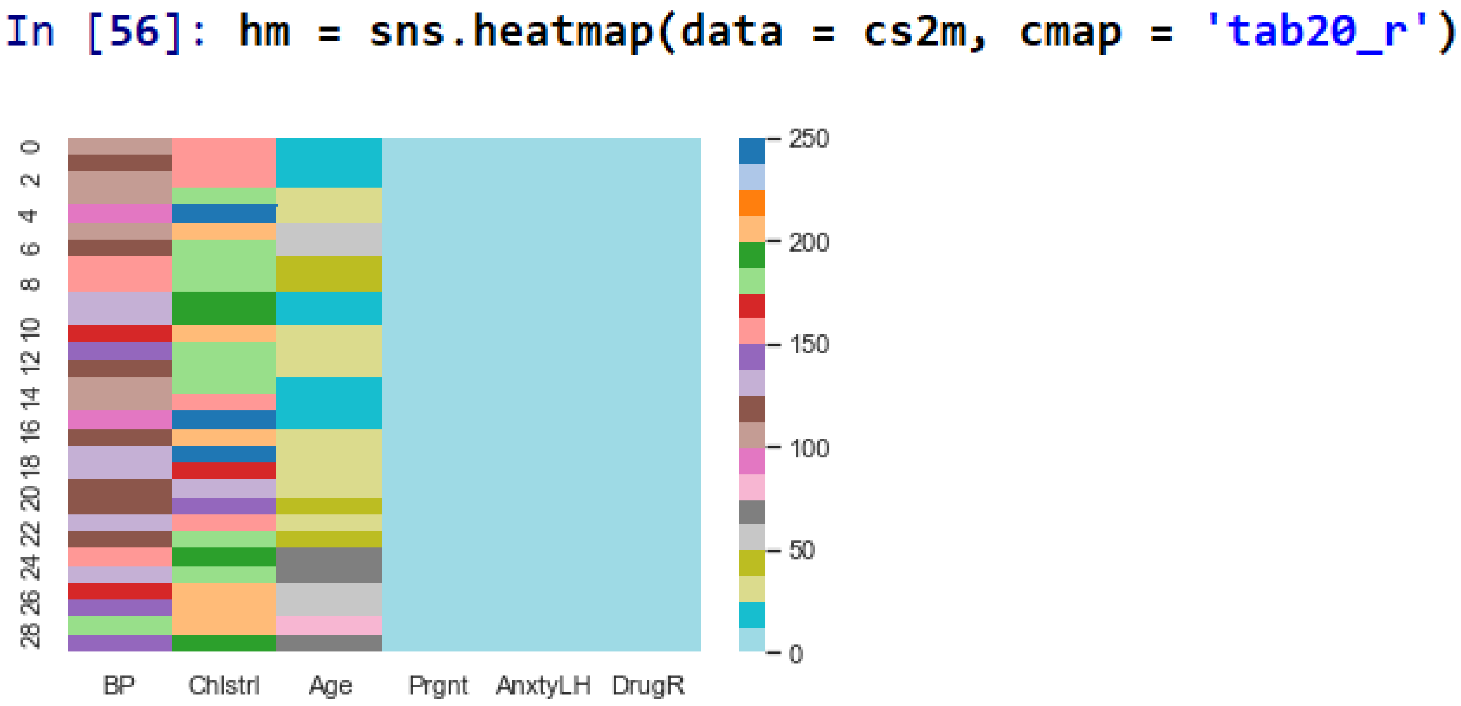


Heat Maps



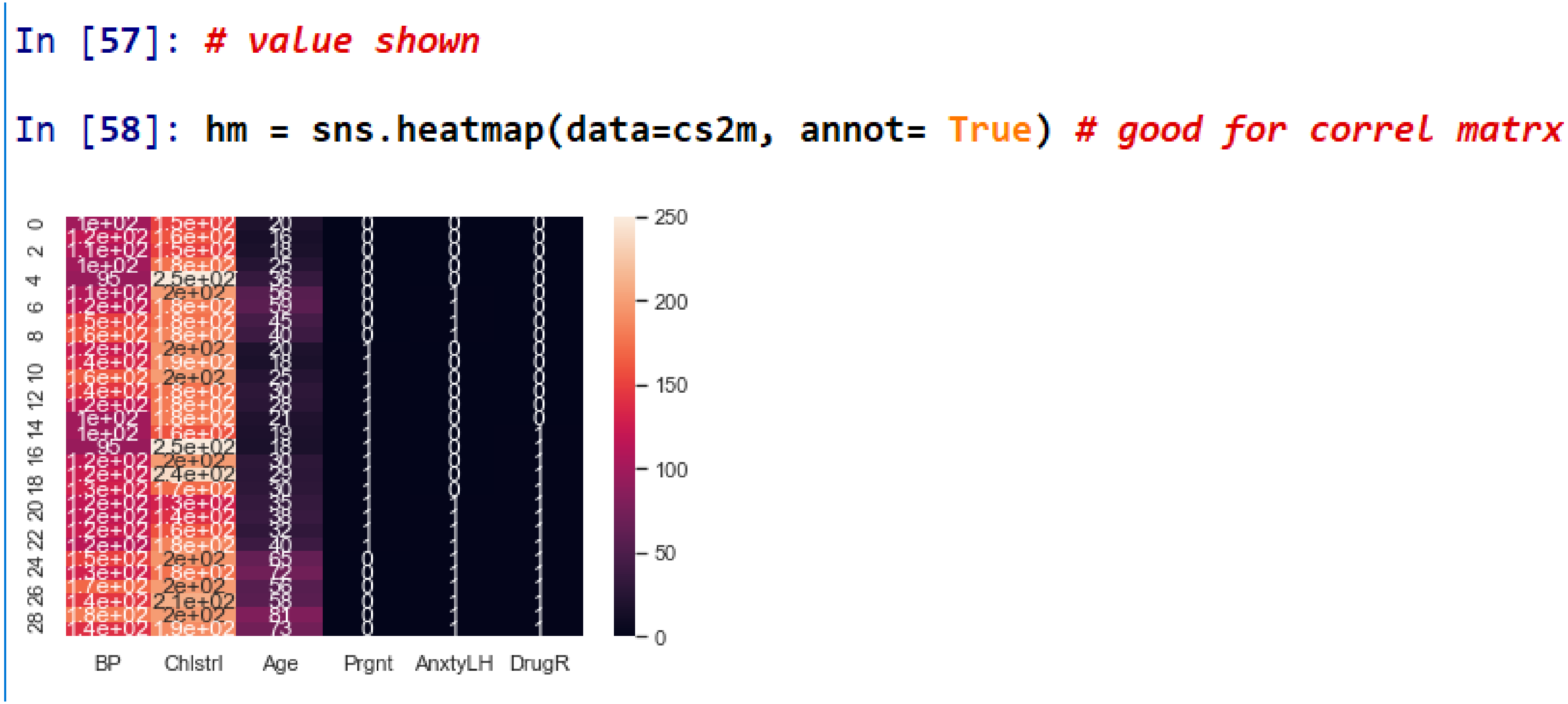
```
In [55]: hm = sns.heatmap(data = cs2m, cmap = 'tab10')
```

Heat Maps



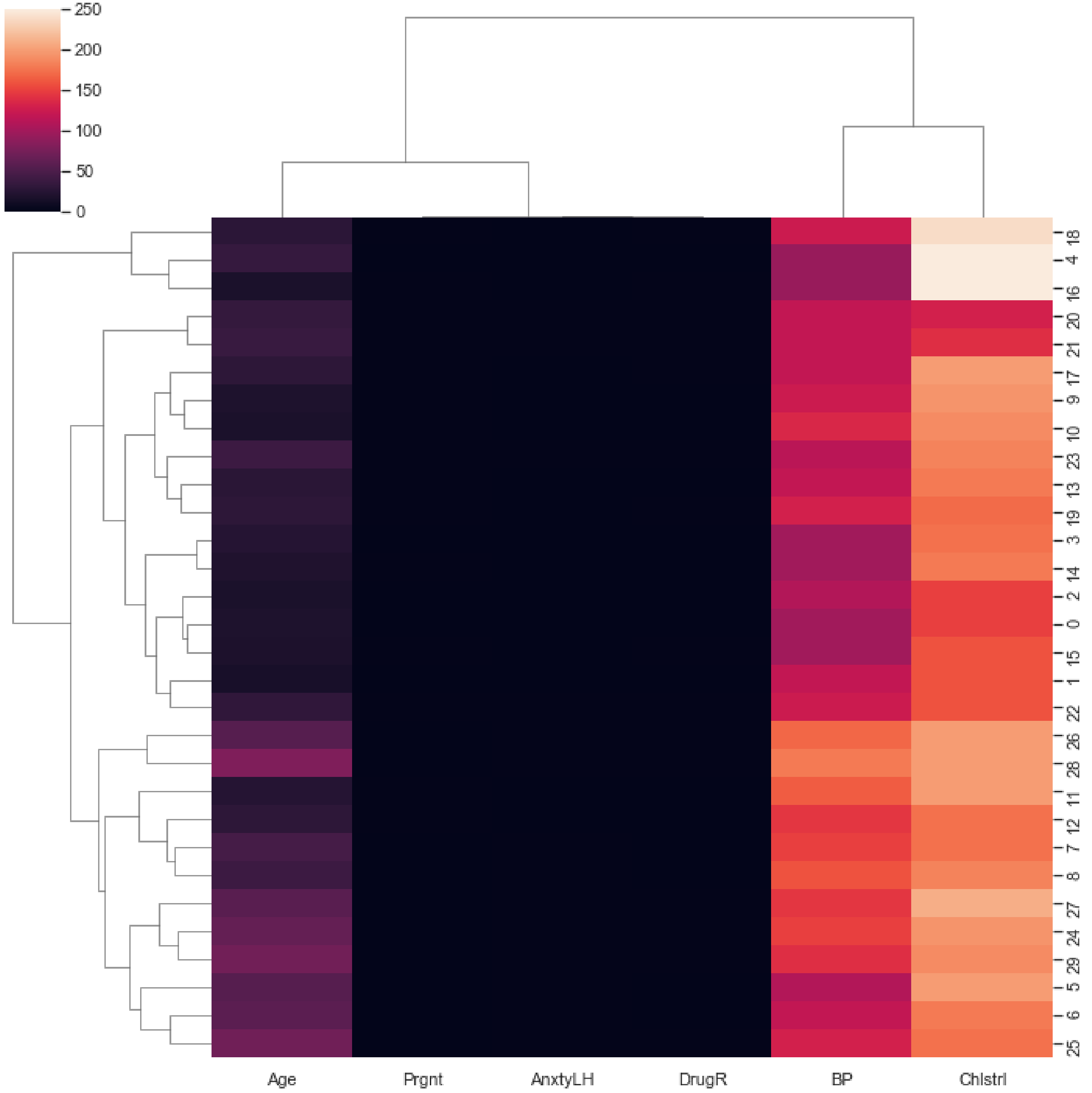
```
In [56]: hm = sns.heatmap(data = cs2m, cmap = 'tab20_r')
```

Heat Maps



```
In [57]: # value shown

In [58]: hm = sns.heatmap(data=cs2m, annot= True) # good for correl matrx
```

Heat Maps



250
200
150
100
50
0
Age
Prgnt
AnxtyLH
DrugR
BP
Chlstrl
18
4
16
20
21
17
9
10
23
13
19
3
14
2
0
15
1
22
26
28
11
12
7
8
27
24
29
5
6
25

DO
WHAT
YOU
LOVE
LOVE
WHAT
YOU
DO

Enjoy
Learning