```
import matplotlib.pyplot as plt
import numpy as np
import pandas as pd
import seaborn as sns
import matplotlib as mpl
```

```
/usr/local/lib/python3.6/dist-packages/statsmodels/tools/_testing.py:19: FutureWarning: pandas.util.testing is deprecat
  import pandas.util.testing as tm
```

```
from google.colab import drive
drive.mount('/content/drive')
```

```
Go to this URL in a browser: https://accounts.google.com/o/oauth2/auth?client_id=947318989803-6bn6qk8qdgf4n4g3pfee6491h

Enter your authorization code:
··········
Mounted at /content/drive
```

```
insurance = pd.read_csv('/content/drive/My Drive/Python DataScience/Visualization/Seabo
insurance.head(10)
```

	age	sex	bmi	children	smoker	region	charges
0	19	female	27.900	0	yes	southwest	16884.92400
1	18	male	33.770	1	no	southeast	1725.55230
2	28	male	33.000	3	no	southeast	4449.46200
3	33	male	22.705	0	no	northwest	21984.47061

Box Plot

The box plot is a standardized way of displaying the distribution of data based on the five number summary: minimum, first quartile, median, third quartile, and maximum.

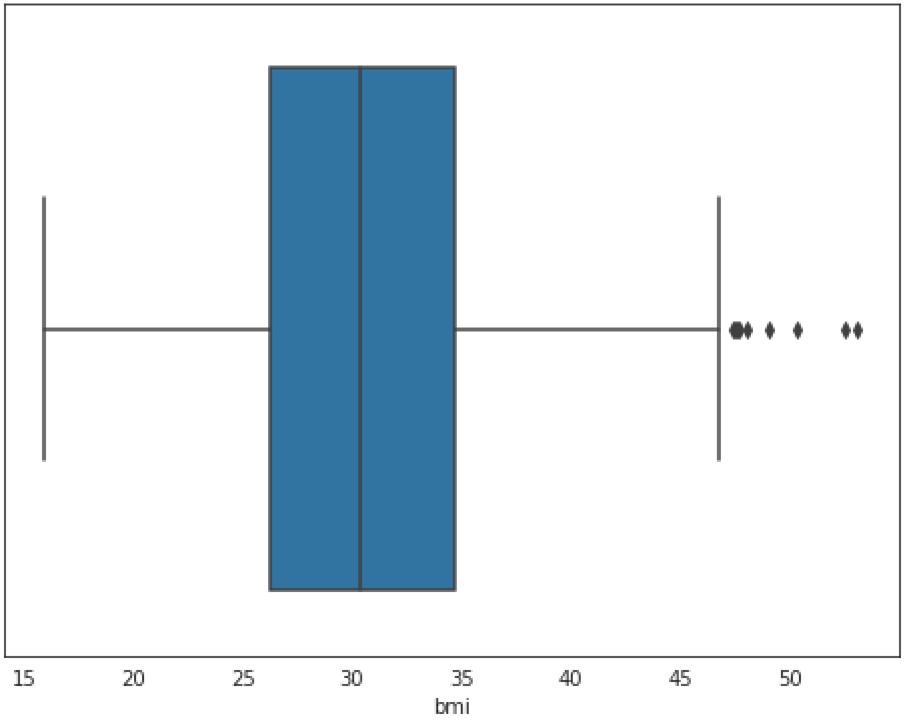
```
# Recover default matplotlib settings
mpl.rcParams.update(mpl.rcParamsDefault)
%matplotlib inline
sns.set_style("white")
```

```
insurance.head()
```

	age	sex	bmi	children	smoker	region	charges
0	19	female	27.900	0	yes	southwest	16884.92400
1	18	male	33.770	1	no	southeast	1725.55230
2	28	male	33.000	3	no	southeast	4449.46200
3	33	male	22.705	0	no	northwest	21984.47061
4	32	male	28.880	0	no	northwest	3866.85520

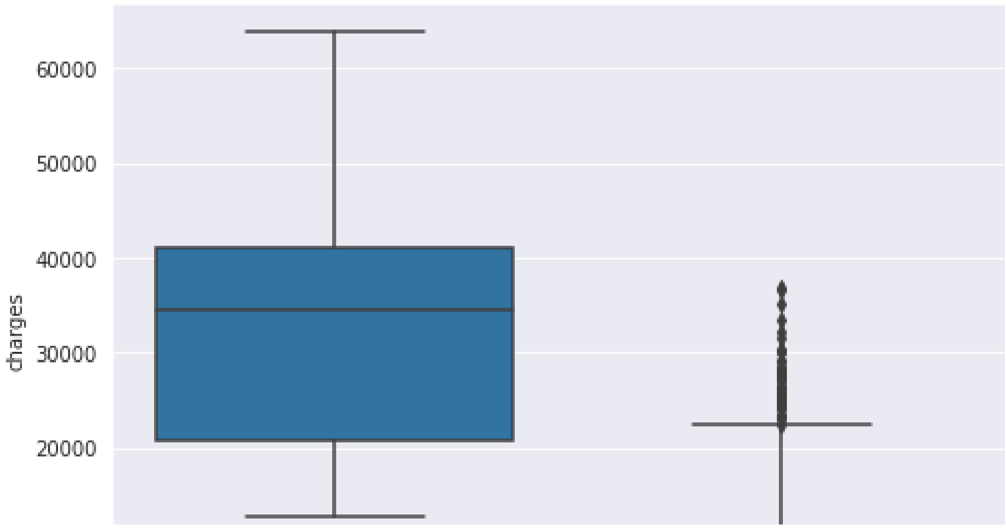
```
# Simple boxplot
# Simple boxplot
plt.figure(figsize=(8,6))
sns.boxplot(insurance.bmi)
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7f2b6f765240>
```



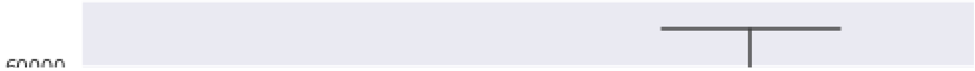
```
# Vertical boxplot
sns.set_style("darkgrid")
plt.figure(figsize=(8,6))
sns.boxplot(x= insurance.smoker , y= insurance.charges )
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7f2b6f6d8c88>
```



```
#Explicit ordering using "order" parameter
plt.figure(figsize=(8,6))
sns.boxplot(x= insurance.smoker , y= insurance.charges , order = ['no' , 'yes'])
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7f2b6f1f3fd0>
```



```
helpdesk = pd.read_csv("/content/drive/My Drive/Python DataScience/Visualization/Seabor
#helpdesk.head(10)
```

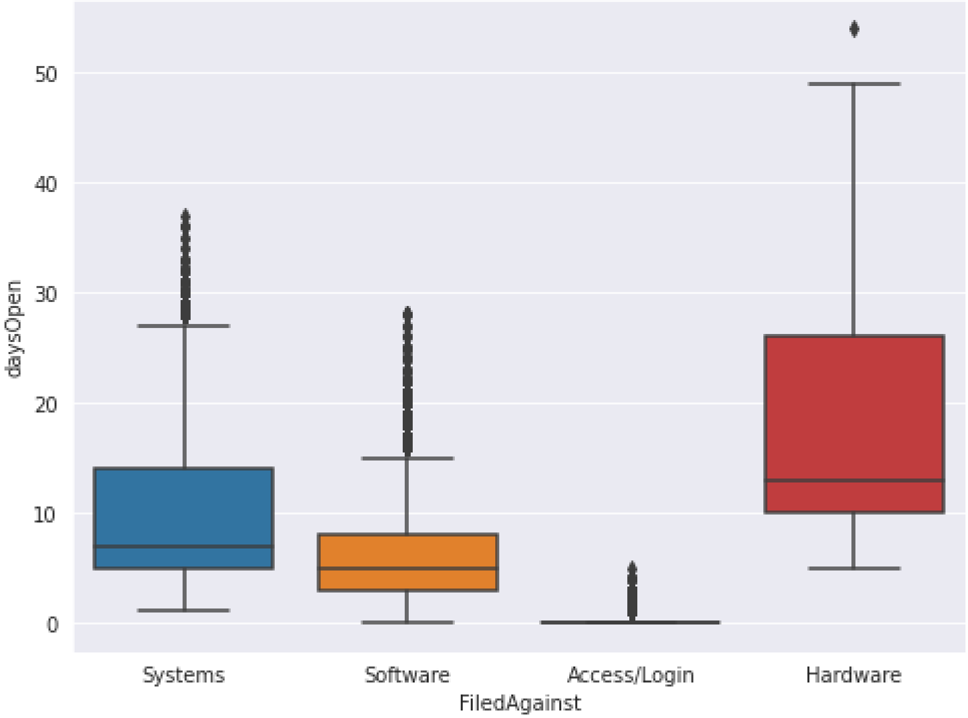
```
helpdesk.head(5)
```

	ticket	requestor	RequestorSeniority	ITOwner	FiledAgainst	TicketType	Severity	Priority	daysOpen	Satisfacti
0	1	1929	1 - Junior	50	Systems	Issue	2 - Normal	0 - Unassigned	3	1 - Unsatisfi
1	2	1587	2 - Regular	15	Software	Request	1 - Minor	1 - Low	5	1 - Unsatisfi
2	3	925	2 - Regular	15	Access/Login	Request	2 - Normal	0 - Unassigned	0	0 - Unknov
3	4	413	4 - Management	22	Systems	Request	2 - Normal	0 - Unassigned	20	0 - Unknov

```
plt.figure(figsize=(7,6))
sns.boxplot(helpdesk.daysOpen )
plt.show()
```

```
plt.figure(figsize=(8,6))
sns.boxplot(x= helpdesk.FiledAgainst , y= helpdesk.daysOpen )
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7f2b6f0e3c50>
```

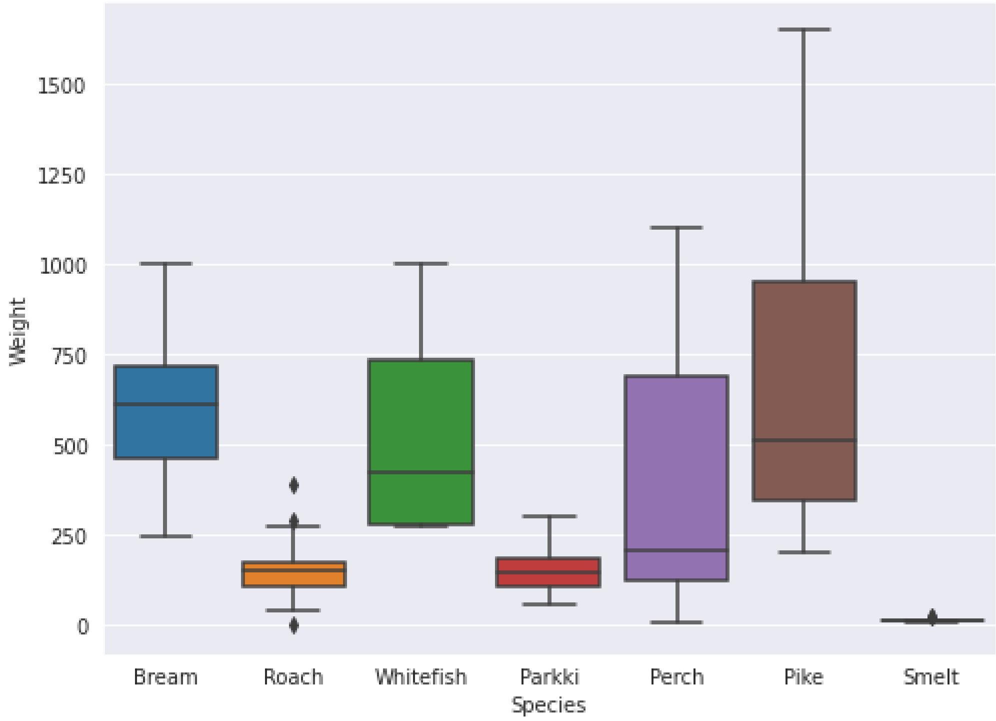


```
fish= pd.read_csv("/content/drive/My Drive/Python DataScience/Visualization/Seaborn/Dat
```

```
plt.figure(figsize=(8,6))
```

```
sns.boxplot(x= fish.Species , y= fish.Weight)
```

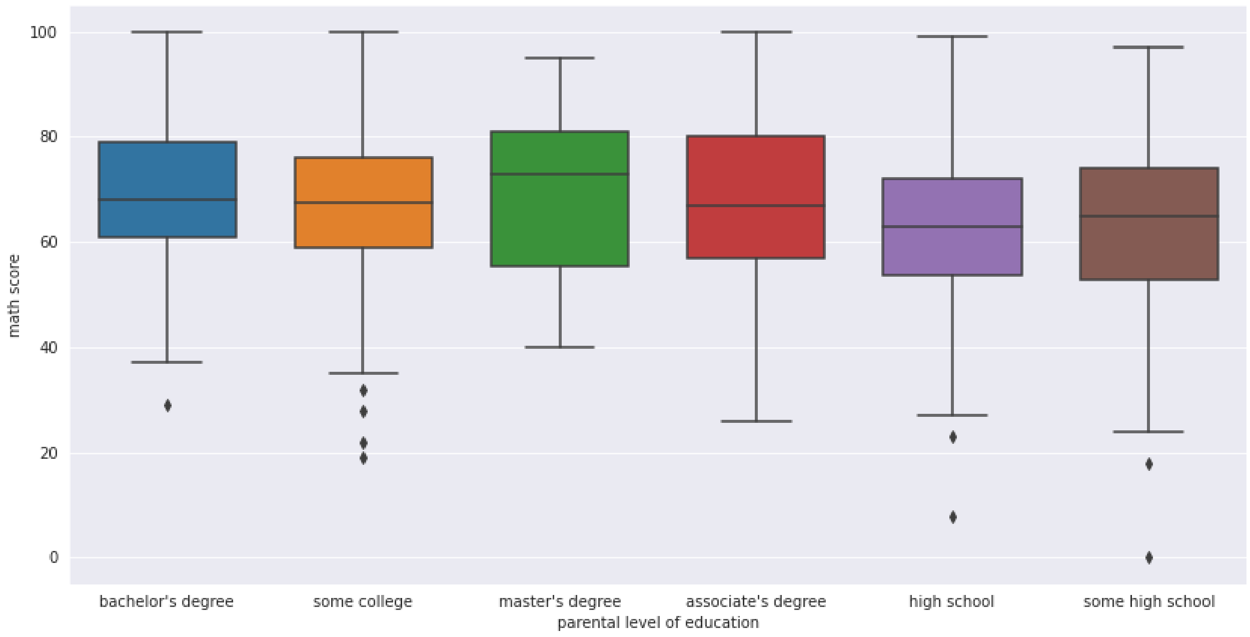
```
<matplotlib.axes._subplots.AxesSubplot at 0x7f2b6eeb9eb8>
```

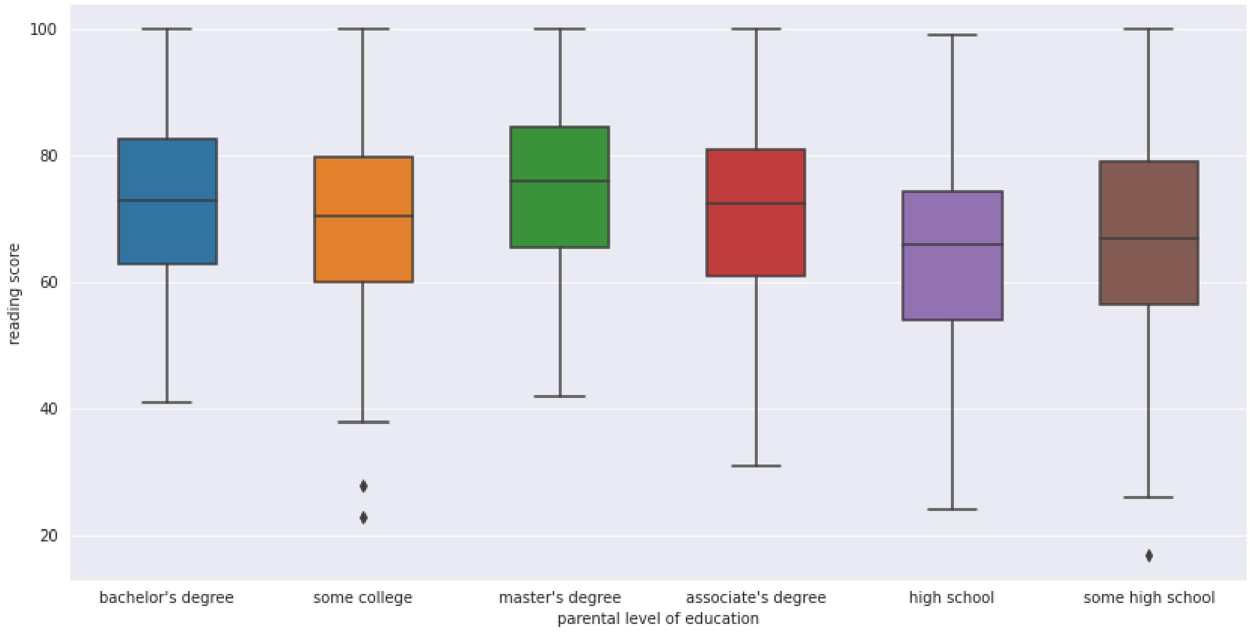


```
stdperf = pd.read_csv("/content/drive/My Drive/Python DataScience/Visualization/Seaborn

#stdperf = pd.read_csv("studentp.csv")
stdperf.head(10)
```

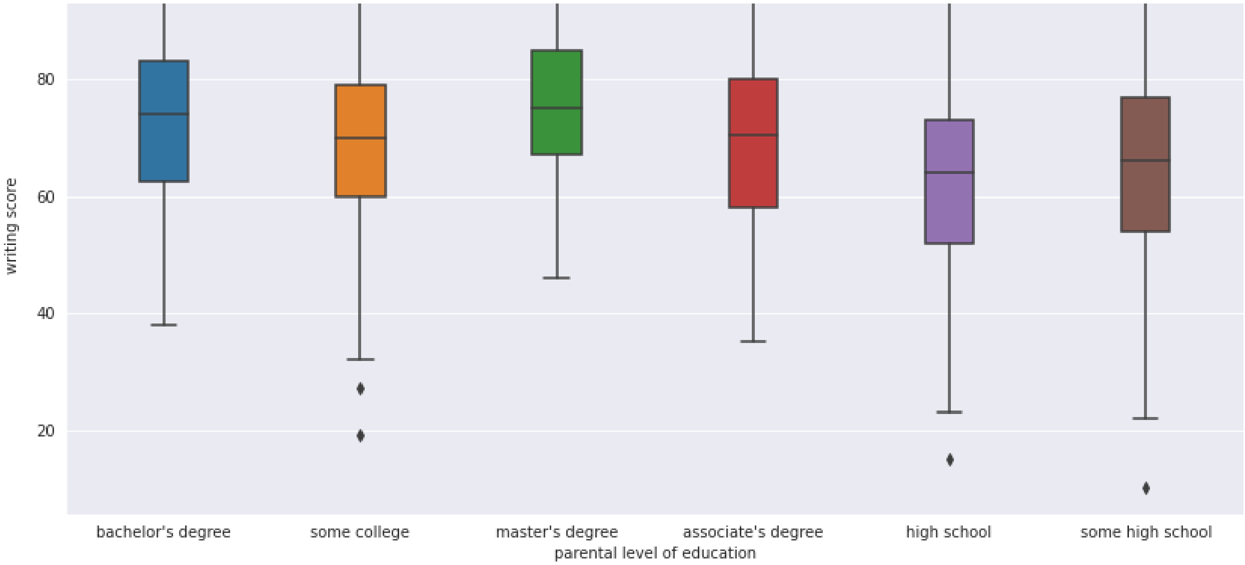
	gender	race/ethnicity	parental level of education	lunch	test preparation course	math score	reading score	writing score
0	female	group B	bachelor's degree	standard	none	72	72	74
1	female	group C	some college	standard	completed	69	90	88
2	female	group B	master's degree	standard	none	90	95	93

```
# Adjust width of boxes
plt.subplots(figsize = (14,24))
plt.subplot(3,1,1)

sns.boxplot(x= stdperf['parental level of education'] , y= stdperf['math score'] ,  wid

plt.subplot(3,1,2)
sns.boxplot(x= stdperf['parental level of education'] , y= stdperf['reading score'] ,

plt.subplot(3,1,3)
sns.boxplot(x= stdperf['parental level of education'] , y= stdperf['writing score'] ,

plt.show()
```

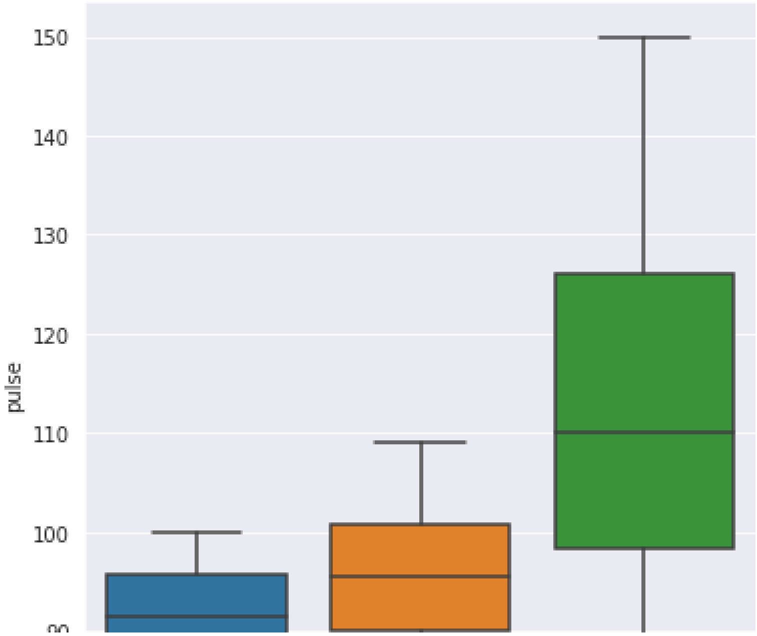

100
80
60
40
20
0
math score
bachelor's degree
some college
master's degree
associate's degree
high school
some high school
parental level of education


100
80
60
40
20
reading score
bachelor's degree
some college
master's degree
associate's degree
high school
some high school
parental level of education

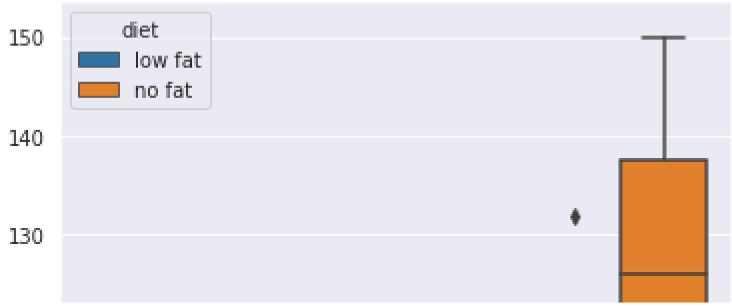
100



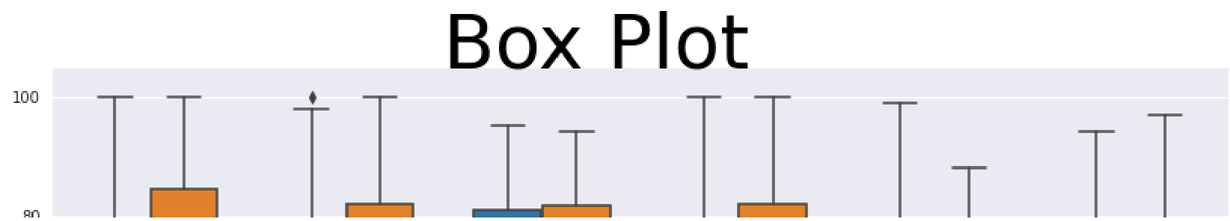
```
exercise = pd.read_csv("/content/drive/My Drive/Python DataScience/Visualization/Seabor

plt.figure(figsize = (6,7))
sns.boxplot(x= exercise.kind , y= exercise.pulse)
plt.show()
```



```
# Show groups with different colors using "hue"
plt.figure(figsize = (6,7))
sns.boxplot(x= exercise.kind , y= exercise.pulse , hue=exercise.diet)
plt.show()
```

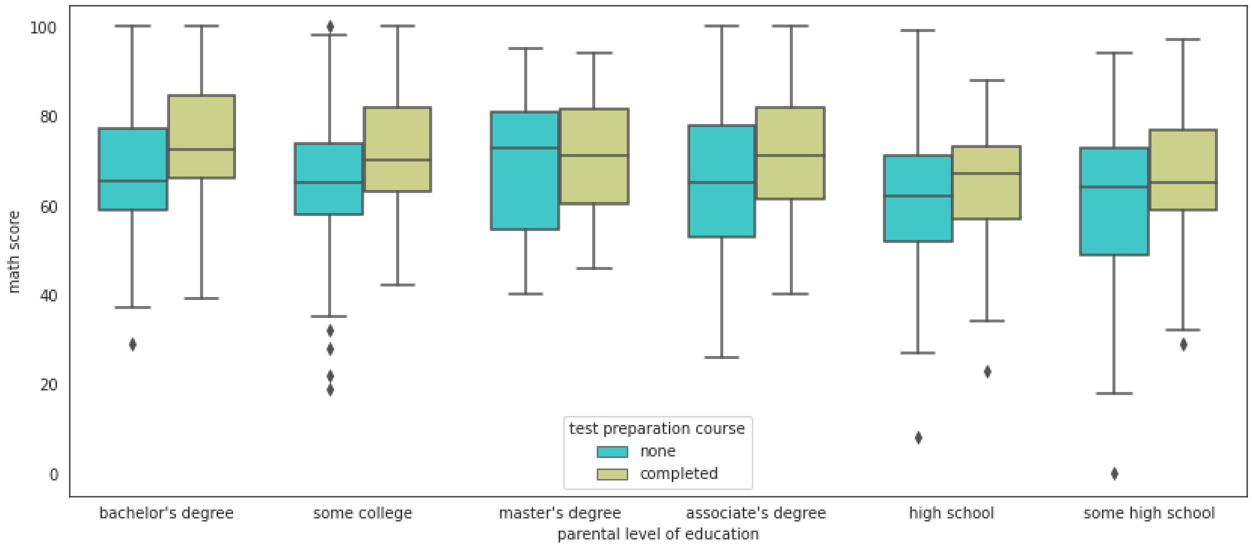


```
plt.figure(figsize = (14,8))
plt.text(1.5,105, "Box Plot", fontsize = 50, color='Black')
sns.boxplot(x= stdperf['parental level of education'] , y= stdperf['math score'] ,
            width=.7 , hue=stdperf['test preparation course'])
plt.show()
```

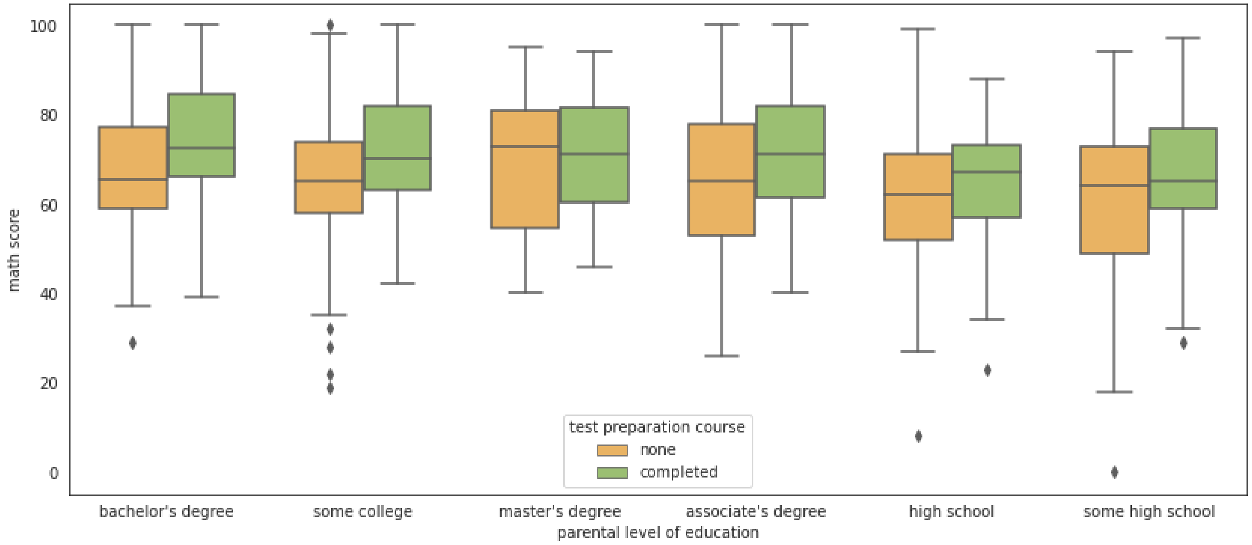


```
sns.set_style("white")
```

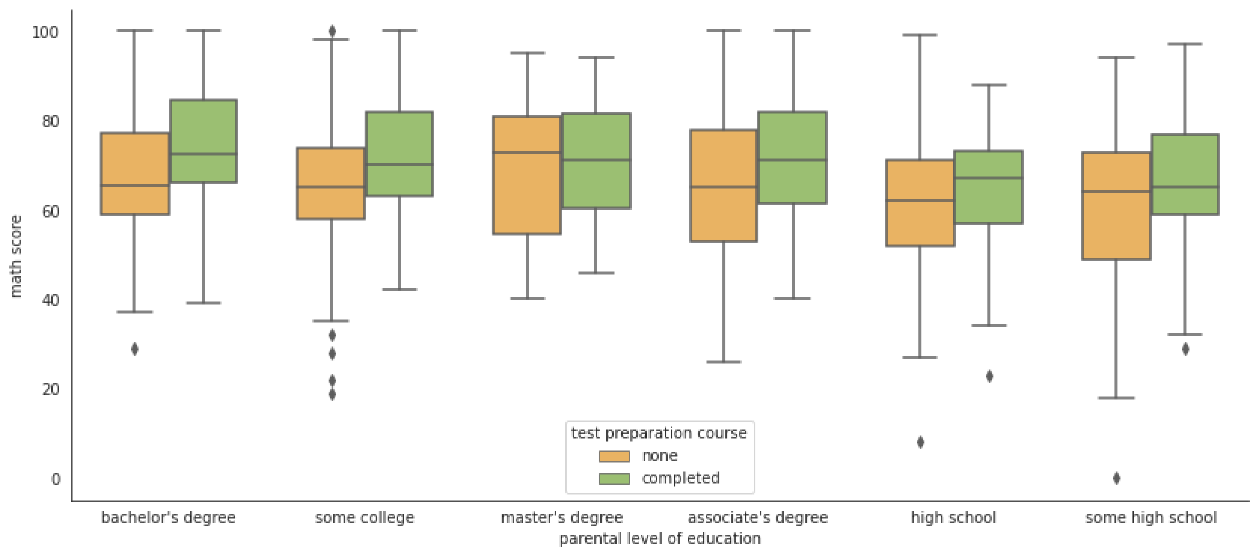
```
plt.figure(figsize = (14,6))
sns.boxplot(x= stdperf['parental level of education'] , y= stdperf['math score'] ,
            width=.7 , hue=stdperf['test preparation course'] , palette= 'rainbow')
plt.show()
```



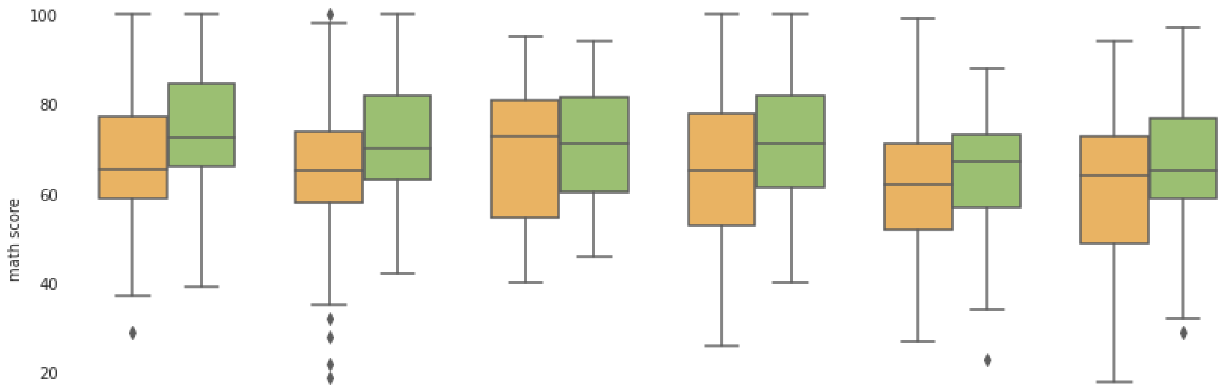
```
plt.figure(figsize = (14,6))
sns.boxplot(x= stdperf['parental level of education'] , y= stdperf['math score'] ,
sns.boxplot(x= stdperf['parental level of education'] , y= stdperf['math score'] ,
            width=.7 , hue=stdperf['test preparation course'] ,
            palette= {"none":'#FFB74D' , "completed":'#9CCC65'} )
plt.show()
```



```
# Remove the top and right axis spines
plt.figure(figsize = (14,6))
sns.boxplot(x= stdperf['parental level of education'] , y= stdperf['math score'] ,
            width=.7 , hue=stdperf['test preparation course'] ,
            palette= {"none":'#FFB74D' , "completed":'#9CCC65'} )
sns.despine()
# More about sns.despine() here - https://seaborn.pydata.org/tutorial/aesthetics.html
plt.show()
```

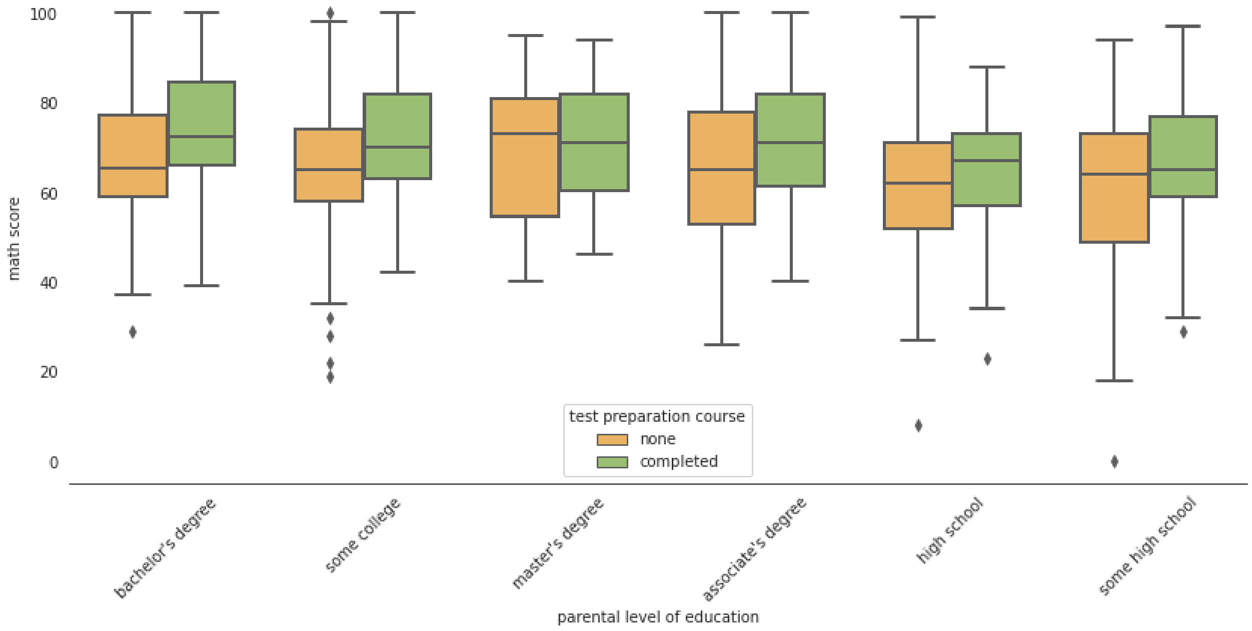


```
plt.figure(figsize = (14,6))
sns.boxplot(x= stdperf['parental level of education'] , y= stdperf['math score'] ,
            width=.7 , hue=stdperf['test preparation course'] ,
            palette= {"none":'#FFB74D' , "completed":'#9CCC65'} )
sns.despine(left=True)
plt.show()
```

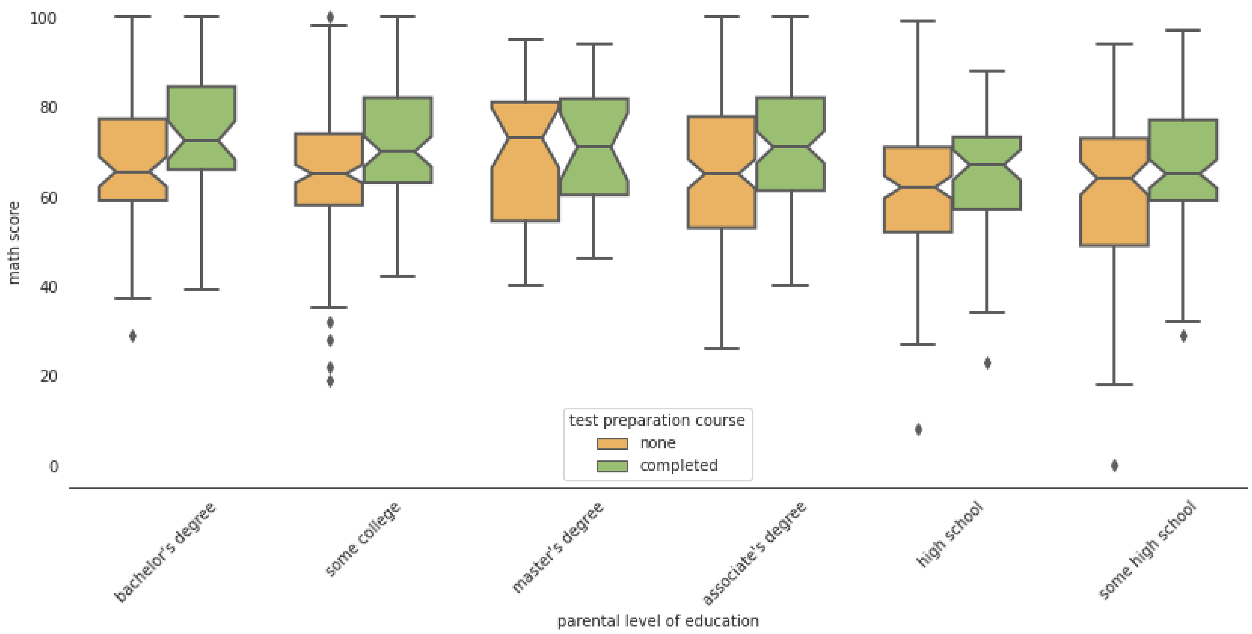


```
# Changing Linewidth
plt.figure(figsize = (14,6))
sns.boxplot(x= stdperf['parental level of education'] , y= stdperf['math score'] ,
            width=.7 , hue=stdperf['test preparation course'] ,
            palette= {"none":'#FFB74D' , "completed":'#9CCC65'} ,linewidth = 3)
sns.despine(left=True)
plt.show()
```

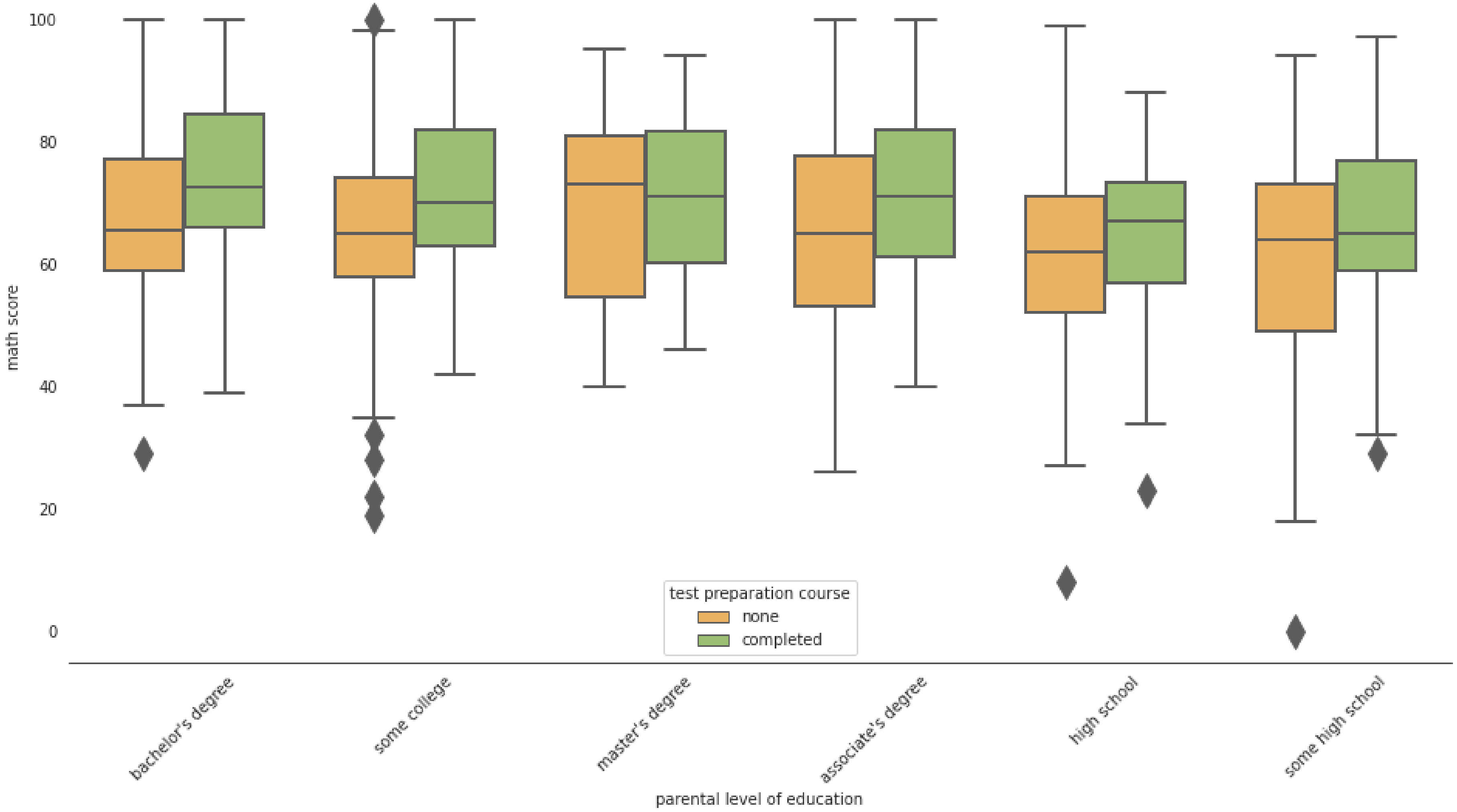
```
plt.figure(figsize = (14,6))
sns.boxplot(x= stdperf['parental level of education'] , y= stdperf['math score'] ,
            width=.7 , hue=stdperf['test preparation course'] ,
            palette= {"none":'#FFB74D' , "completed":'#9CCC65'} ,linewidth = 2)
sns.despine(left=True)
plt.xticks(rotation=45)
plt.show()
```



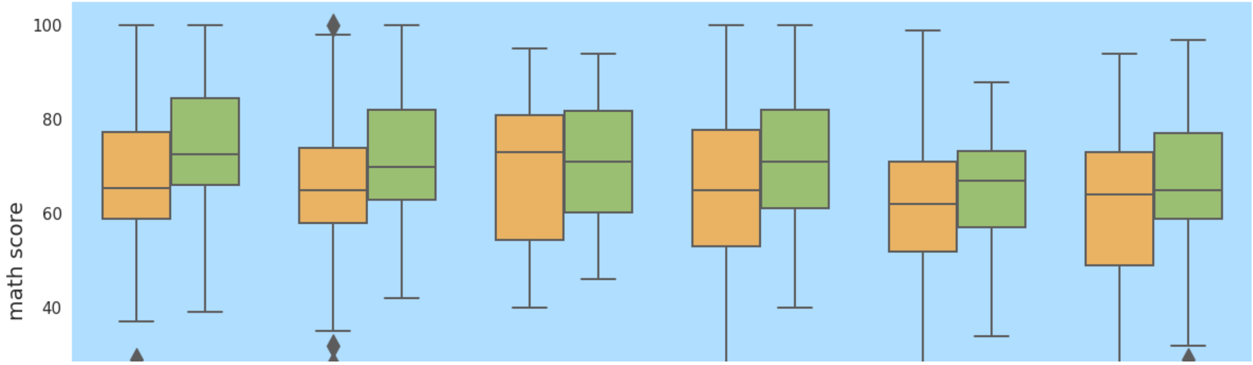
```
# Add a notch to the box
plt.figure(figsize = (14,6))
sns.boxplot(x= stdperf['parental level of education'] , y= stdperf['math score'] ,
            width=.7 , hue=stdperf['test preparation course'] ,
            palette= {"none":'#FFB74D' , "completed":'#9CCC65'} ,linewidth = 2 , notch=
sns.despine(left=True)
plt.xticks(rotation=45)
plt.show()
```



```
# Change the size of outlier markers
plt.figure(figsize = (16,8))
sns.boxplot(x= stdperf['parental level of education'] , y= stdperf['math score'] ,
sns.boxplot(x= stdperf['parental level of education'] , y= stdperf['math score'] ,
               width=.7 , hue=stdperf['test preparation course'] ,
               palette= {"none":'#FFB74D' , "completed":'#9CCC65'} ,linewidth = 2  , flier
sns.despine(left=True)
plt.xticks(rotation=45)
plt.show()
```



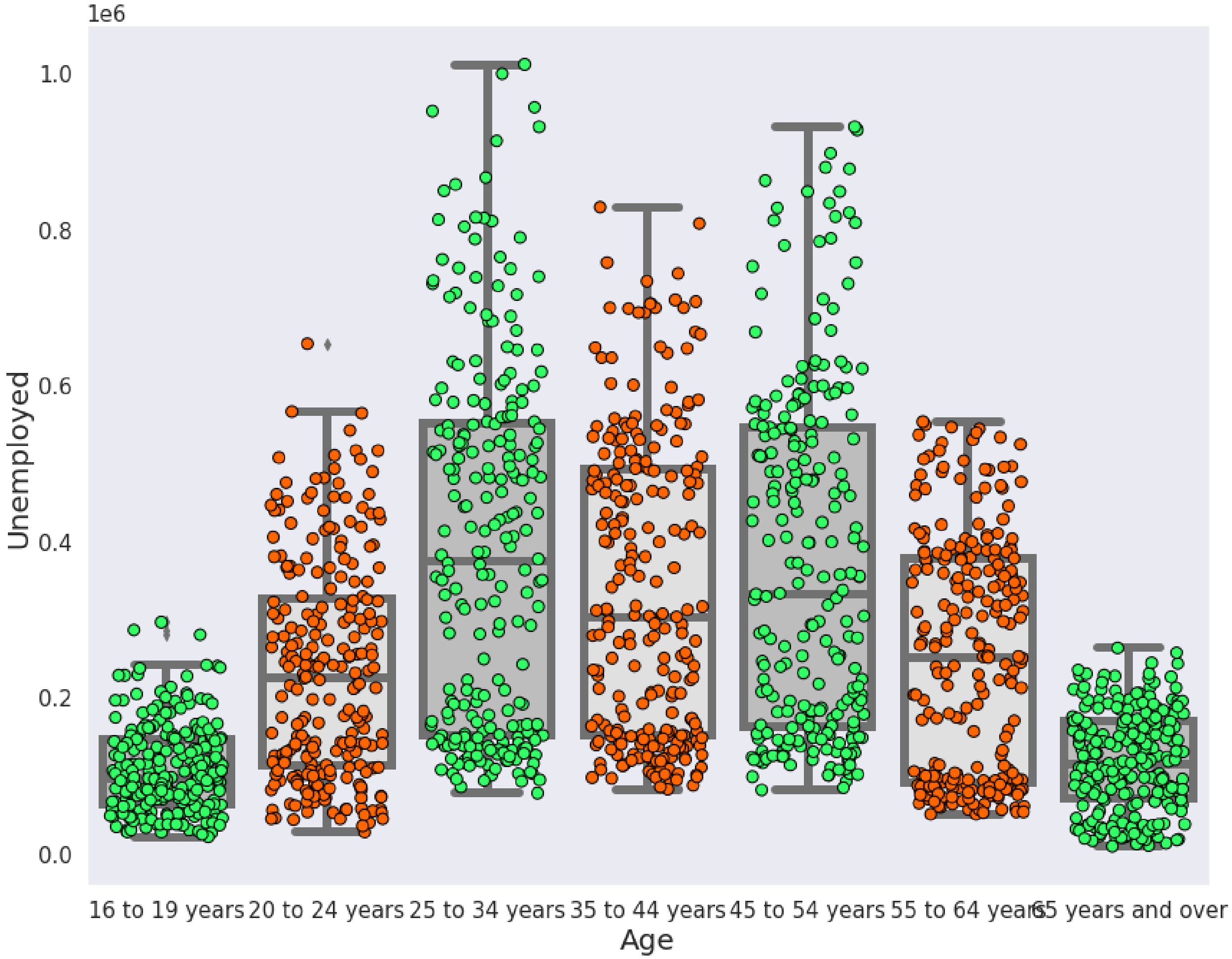
```
# Change the size of outlier markers
sns.set(rc={"axes.facecolor":"#b0deff","axes.grid":False,
            'xtick.labelsize':15,'ytick.labelsize':15,
            'axes.labelsize':20,'figure.figsize':(20.0, 9.0)})
sns.boxplot(x= stdperf['parental level of education'] , y= stdperf['math score'] ,
            width=.7 , hue=stdperf['test preparation course'] ,
            palette= {"none":'#FFB74D' , "completed":'#9CCC65'} ,linewidth = 2  , flier
sns.despine(left=True)
plt.xticks(rotation=45)
plt.show()
```



```
mpl.rcParams.update(mpl.rcParamsDefault)
%matplotlib inline
```

```
employment = pd.read_excel("/content/drive/My Drive/Python DataScience/Visualization/Se
```

```
plt.figure(figsize=(14,11))
sns.set(rc={"axes.grid":False,
            'xtick.labelsize':15,'ytick.labelsize':15,
            'axes.labelsize':20,'figure.figsize':(20.0, 9.0)})
params = dict(data=employment ,x = employment.Age ,y = employment.Unemployed)
sns.stripplot(**params , size=8,jitter=0.35,palette=['#33FF66','#FF6600'],edgecolor='bl
sns.boxplot(**params ,palette=['#BDBDBD','#E0E0E0'],linewidth=6)
plt.show()
```

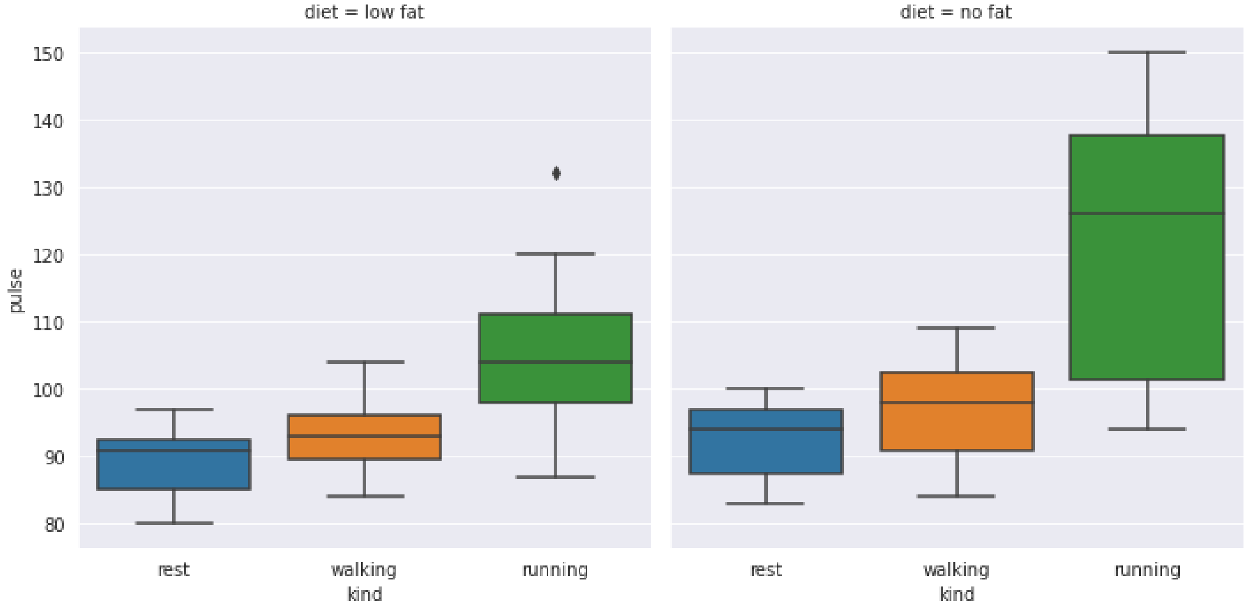


```
mpl.rcParams.update(mpl.rcParamsDefault)
%matplotlib inline
sns.set_style("darkgrid")

exercise = pd.read_csv("/content/drive/My Drive/Python DataScience/Visualization/Seabor

exercise.head()
```

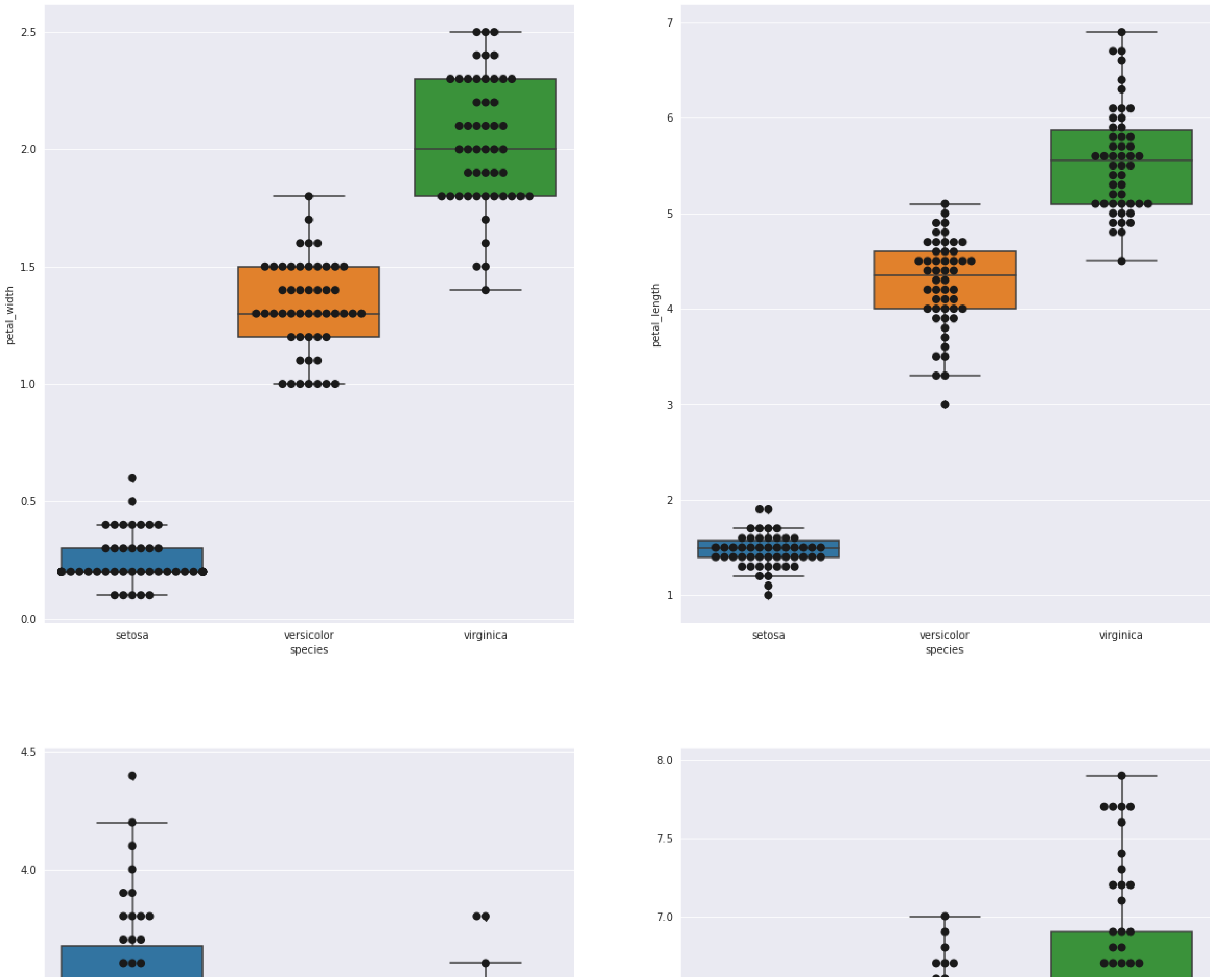
	id	diet	pulse	time	kind
0	1	low fat	85	1 min	rest
1	1	low fat	85	15 min	rest
2	1	low fat	88	30 min	rest
3	2	low fat	90	1 min	rest
4	2	low fat	92	15 min	rest

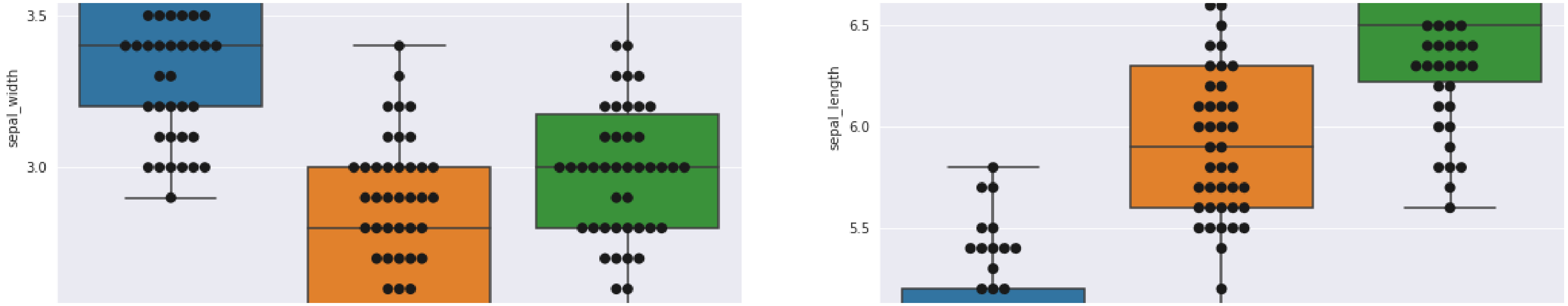
```
plt.figure(figsize=(18,16))
sns.catplot(x="kind", y="pulse",col="diet",data=exercise, kind="box");
plt.show()
```

```
<Figure size 1296x1152 with 0 Axes>
```

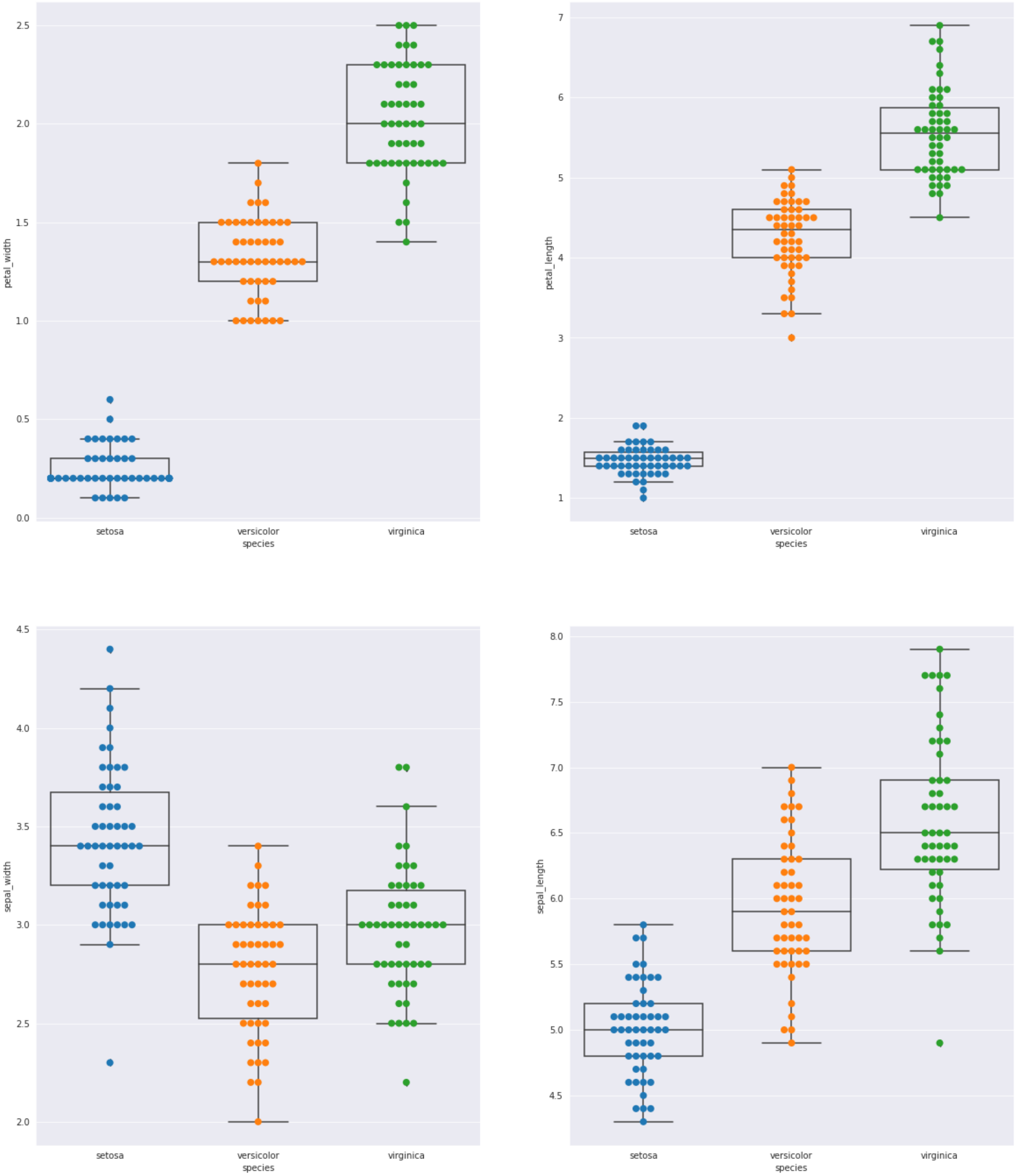


```
iris = sns.load_dataset("iris")
# Displaying multiple violin plots using subplot function
# Displaying multiple violin plots using subplot function.
fig1 , axes = plt.subplots(nrows=2,ncols=2 , figsize = (20,24))
sns.swarmplot(x="species" , y = "petal_width" , ax = axes[0,0] ,data=iris , color=".10"
sns.boxplot(x="species" , y = "petal_width" , ax = axes[0,0] ,data=iris )
sns.swarmplot(x="species" , y = "petal_length" ,ax = axes[0,1] , data=iris  ,  color=".
sns.boxplot(x="species" , y = "petal_length" ,ax = axes[0,1] , data=iris)
sns.swarmplot(x="species" , y = "sepal_width" , ax = axes[1,0] , data=iris ,  color=".1
sns.boxplot(x="species" , y = "sepal_width" , ax = axes[1,0] , data=iris)
sns.swarmplot(x="species" , y = "sepal_length" , ax = axes[1,1] , data=iris ,  color=".
sns.boxplot(x="species" , y = "sepal_length" , ax = axes[1,1] , data=iris)
plt.show()
```

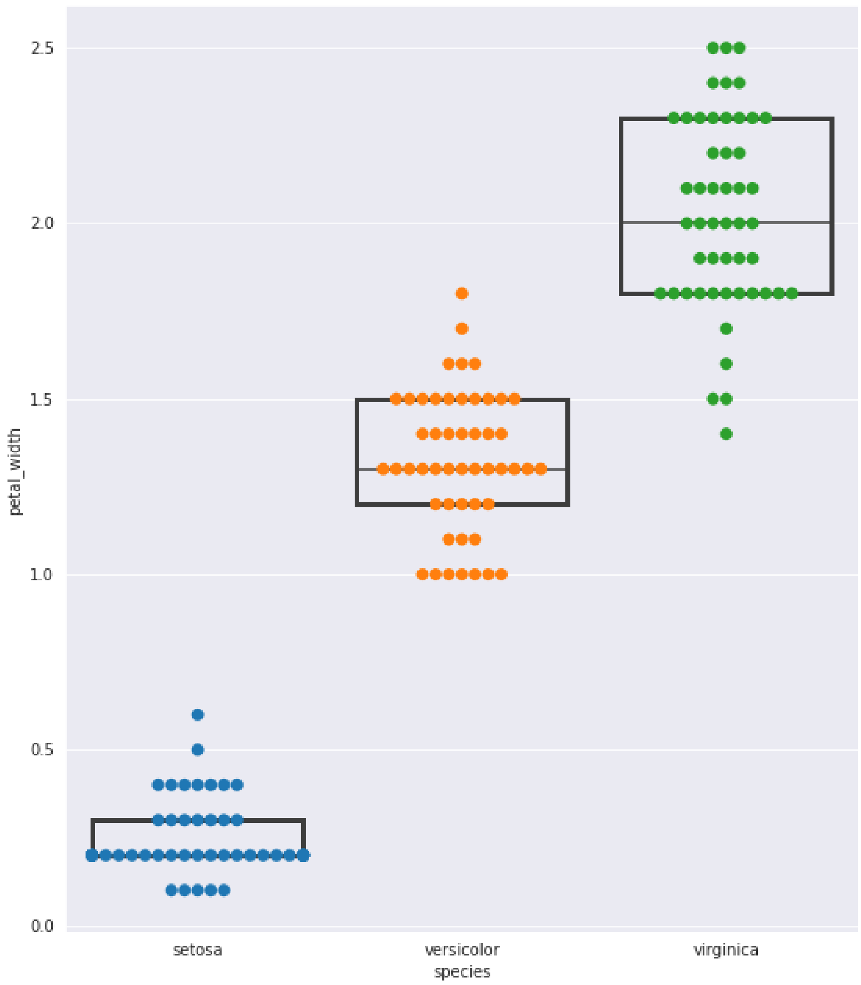

2.5
2.0
1.5
1.0
0.5
0.0
petal_width
setosa
versicolor
virginica
species
7
6
5
4
3
2
1
petal_length
setosa
versicolor
virginica
species
4.5
4.0
8.0
7.5
7.0

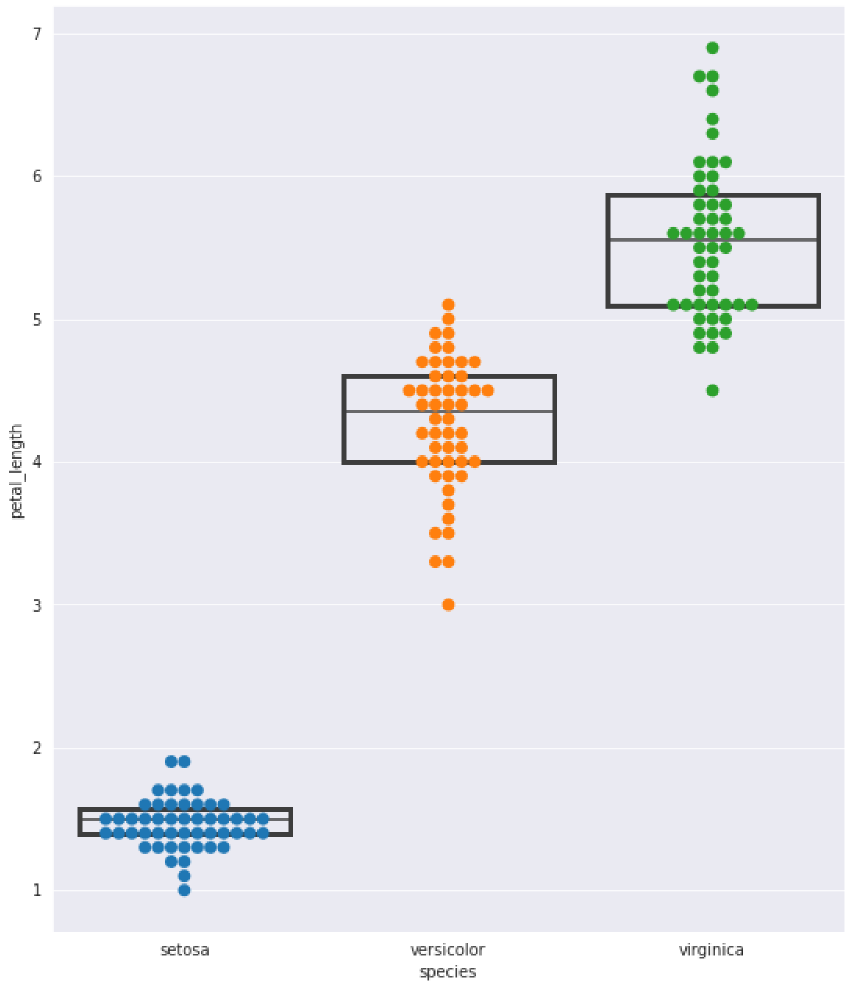


```
fig1 , axes = plt.subplots(nrows=2,ncols=2 , figsize = (20,24))
sns.swarmplot(x="species" , y = "petal_width" , ax = axes[0,0] ,data=iris, size=8,)
sns.boxplot(x="species" , y = "petal_width" , ax = axes[0,0] ,data=iris ,boxprops={'fac
sns.swarmplot(x="species" , y = "petal_length" ,ax = axes[0,1] , data=iris  , size=8)
sns.boxplot(x="species" , y = "petal_length" ,ax = axes[0,1] , data=iris ,boxprops={'fa
sns.swarmplot(x="species" , y = "sepal_width" , ax = axes[1,0] , data=iris ,  size=8)
sns.boxplot(x="species" , y = "sepal_width" , ax = axes[1,0] , data=iris ,boxprops={'fa
sns.swarmplot(x="species" , y = "sepal_length" , ax = axes[1,1] , data=iris, size=8)
sns.boxplot(x="species" , y = "sepal_length" , ax = axes[1,1] , data=iris ,boxprops={'f
plt.show()
```

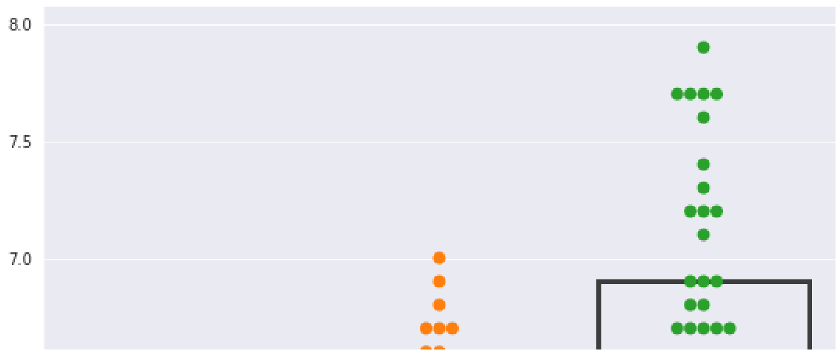

petal_width
0.0
0.5
1.0
1.5
2.0
2.5
setosa
versicolor
virginica
species
petal_length
1
2
3
4
5
6
7
setosa
versicolor
virginica
species
sepal_width
2.0
2.5
3.0
3.5
4.0
4.5
setosa
versicolor
virginica
species
sepal_length
4.5
5.0
5.5
6.0
6.5
7.0
7.5
8.0
setosa
versicolor
virginica
species

```
fig1 , axes = plt.subplots(nrows=2,ncols=2 , figsize = (20,24))
sns.swarmplot(x="species" , y = "petal_width" , ax = axes[0,0] ,data=iris, size=8)
sns.boxplot(x="species" , y = "petal_width" , ax = axes[0,0] ,data=iris , showfliers=Fa
sns.swarmplot(x="species" , y = "petal_length" ,ax = axes[0,1] , data=iris, size=8)
sns.boxplot(x="species" , y = "petal_length" ,ax = axes[0,1] , data=iris , showfliers=F
sns.swarmplot(x="species" , y = "sepal_width" , ax = axes[1,0] , data=iris, size=8)
sns.boxplot(x="species" , y = "sepal_width" , ax = axes[1,0] , data=iris , showfliers=F
sns.swarmplot(x="species" , y = "sepal_length" , ax = axes[1,1] , data=iris , size=8)
sns.boxplot(x="species" , y = "sepal_length" , ax = axes[1,1] , data=iris , showfliers=
plt.show()
```


2.5
2.0
1.5
1.0
0.5
0.0
petal_width
setosa
versicolor
virginica
species


7
6
5
4
3
2
1
petal_length
setosa
versicolor
virginica
species

4.5
4.0


8.0
7.5
7.0