```
 load the dataset
# importing libraries import pandas as pd import numpy as np import
matplotlib.pyplot as plt import seaborn as sns
from google.colab import drive drive.mount('/content/drive')
Drive already mounted at /content/drive; to attempt to forcibly
remount, call drive.mount("/content/drive", force_remount=True).
df=pd.read_csv('/content/drive/MyDrive/insurance.csv') df.head()
id Date number of bedrooms number of bathrooms living area \
0 6762810145 42491 5 2.50

3650
1 6762810635 42491 4 2.50

2920
2 6762810998 42491 5 2.75

2910
3 6762812605 42491 4 2.50

3310
4 6762812919 42491 3 2.00

2710
lot area number of floors waterfront present number of views \ 0 9050
2.0 0 4
1 4000 1.5 0 0
2 9480 1.5 0 0 3 42998 2.0 0 0 4 4500 1.5 0 0

condition of the house ... Built Year Renovation Year Postal
Code \
0 5 ... 1921 0

122003
1 5 ... 1909 0

122004
2 3 ... 1939 0

122004
3 3 ... 2001 0

122005
4 4 ... 1929 0
```

122006
Lattitude Longitude living_area_renov lot_area_renov \
0 52.8645 -114.557 2880 5400
1 52.8878 -114.470 2470 4000
2 52.8852 -114.468 2940 6600
3 52.9532 -114.321 3350 42847
4 52.9047 -114.485 2060 4500

Number of schools nearby Distance from the airport Price 0 2 58
2380000
1 2 51 1400000
2 1 53 1200000
3 3 76 838000
4 1 51 805000

[5 rows x 23 columns] df.tail()
id Date number of bedrooms number of bathrooms \ 14615 6762830250
42734 2 1.5
14616 6762830339 42734 3 2.0
14617 6762830618 42734 2 1.0
14618 6762830709 42734 4 1.0 14619 6762831463 42734 3 1.0

living area lot area number of floors waterfront present \ 14615 1556
20000 1.0 0
14616 1680 7000 1.5 0
14617 1070 6120 1.0 0
14618 1030 6621 1.0 0 14619 900 4770 1.0 0

number of views condition of the house ... Built Year \ 14615 0 4 ...
1957
14616 0 4 ... 1968
14617 0 3 ... 1962
14618 0 4 ... 1955
14619 0 3 ... 1969

Renovation Year Postal Code Lattitude Longitude living_area_renov \
14615 0 122066 52.6191 -114.472

2250
14616 0 122072 52.5075 -114.393

1540

14617 0 122056 52.7289 -114.507

1130
14618 0 122042 52.7157 -114.411

1420
14619 2009 122018 52.5338 -114.552

900 lot_area_renov Number of schools nearby Distance from the airport
\
14615 17286 3

76
14616 7480 3

59
14617 6120 2

64
14618 6631 3

54
14619 3480 2 55

Price
14615 221700
14616 219200
14617 209000
14618 205000
14619 146000

[5 rows x 23 columns] df.shape (14620, 23) df.info()
<class 'pandas.core.frame.DataFrame'> RangeIndex: 14620 entries, 0 to 14619
Data columns (total 23 columns):
Column Non-Null Count Dtype --- ------ -------------- -----
0 id 14620 non-null int64
1 Date 14620 non-null int64 2 number of bedrooms 14620 non-null int64 3 number of bathrooms 14620 non-null float64 4 living area 14620 non-null int64 5 lot area 14620 non-null int64 6 number of floors 14620 non-null float64 7 waterfront present 14620 non-null int64 8 number of views 14620 non-null int64
9 condition of the house 14620 non-null int64

```
10 grade of the house 14620 non-null int64
11 Area of the house(excluding basement) 14620 non-null int64 12 Area
of the basement 14620 non-null int64

13 Built Year 14620 non-null int64 14 Renovation Year 14620 non-null
int64
15 Postal Code 14620 non-null int64 16 Lattitude 14620 non-null
float64
17 Longitude 14620 non-null float64 18 living_area_renov 14620 non-
null int64 19 lot_area_renov 14620 non-null int64
20 Number of schools nearby 14620 non-null int64 21 Distance from the
airport 14620 non-null int64 22 Price 14620 non-null int64 dtypes:
float64(4), int64(19) memory usage: 2.6 MB df.isnull().any()
id False Date False number of bedrooms False number of bathrooms False
living area False lot area False number of floors False waterfront
present False number of views False condition of the house False grade
of the house False Area of the house(excluding basement) False
Area of the basement False Built Year False
Renovation Year False
Postal Code False
Lattitude False Longitude False living_area_renov False lot_area_renov
False Number of schools nearby False
Distance from the airport False Price False dtype: bool
```

Visulation

Univarient Analysis

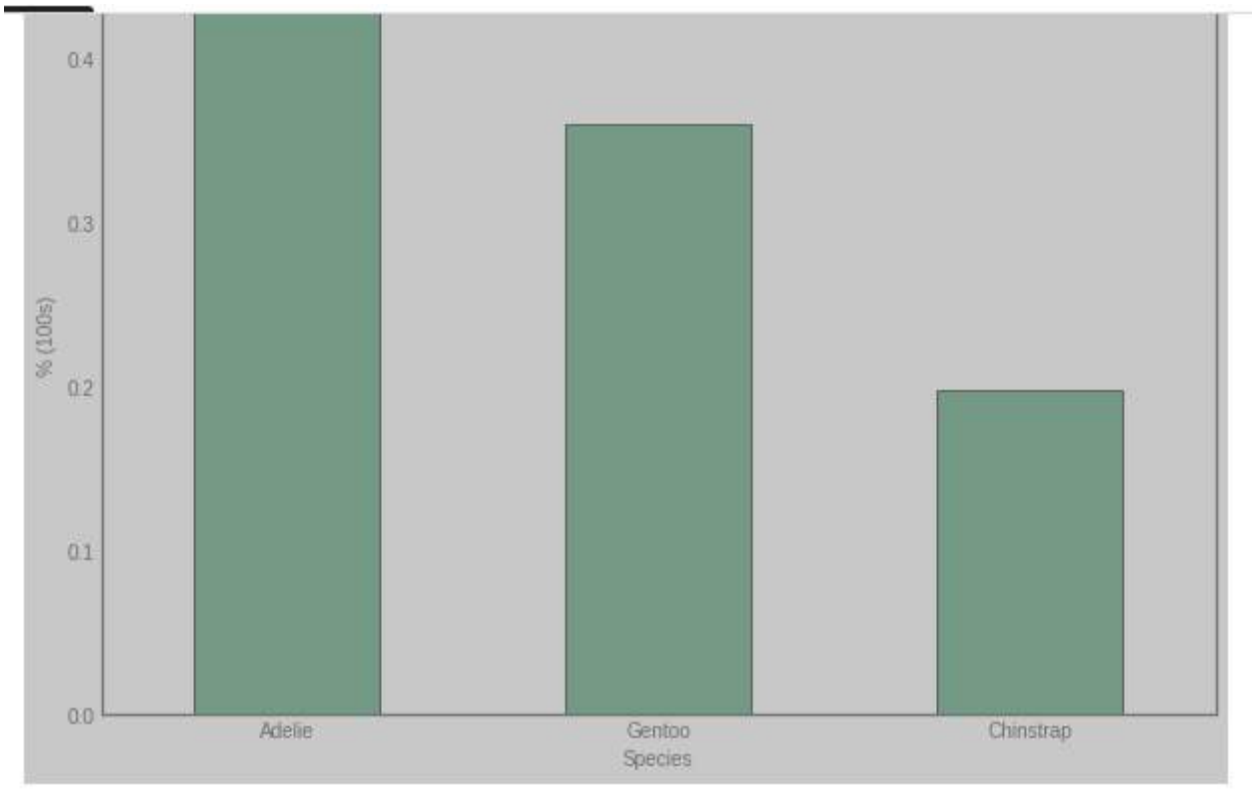
Let's try to understand how the categorical variables are distributed. I'll use the value_counts() method with an argument 'normalize' set to True to see the result i terms of percentage.

In [8]:

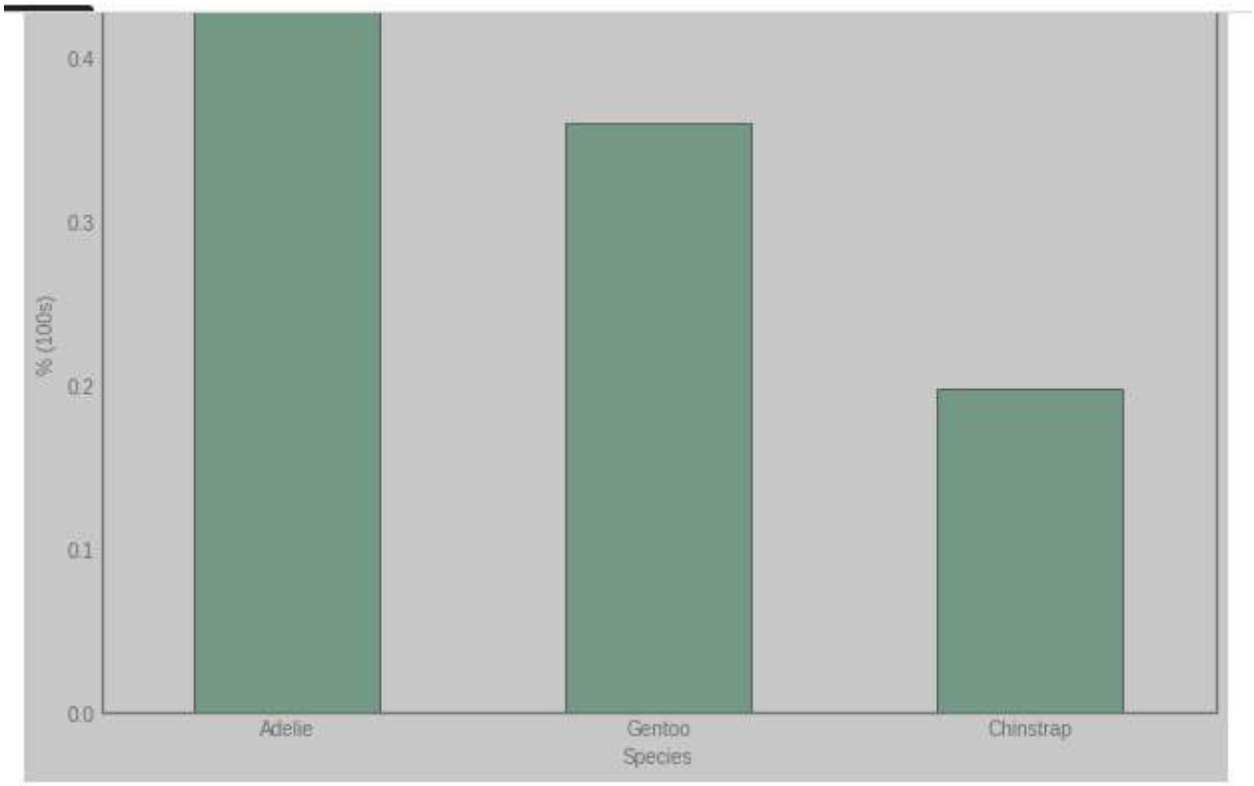
```
plt.rcParams['figure.figsize'] = (10,7)
```

In [9]:

```
df['species'].value_counts(normalize = True).plot(kind = 'bar', color = 'seagreen',
linewidth = 1, edgecolor = 'k')
plt.title('Penguin Species')
plt.xlabel('Species')
plt.ylabel('% (100s)')
plt.xticks(rotation = 360)
plt.show()
```




```
df['island'].value_counts(normalize = True).plot(kind = 'bar', color =
'seagreen', linewidth = 1, edgecolor = 'k')
plt.title('Islands where Penguins live')
plt.xlabel('Island')
plt.ylabel('% (100s)')
plt.xticks(rotation = 360)
plt.show()
```



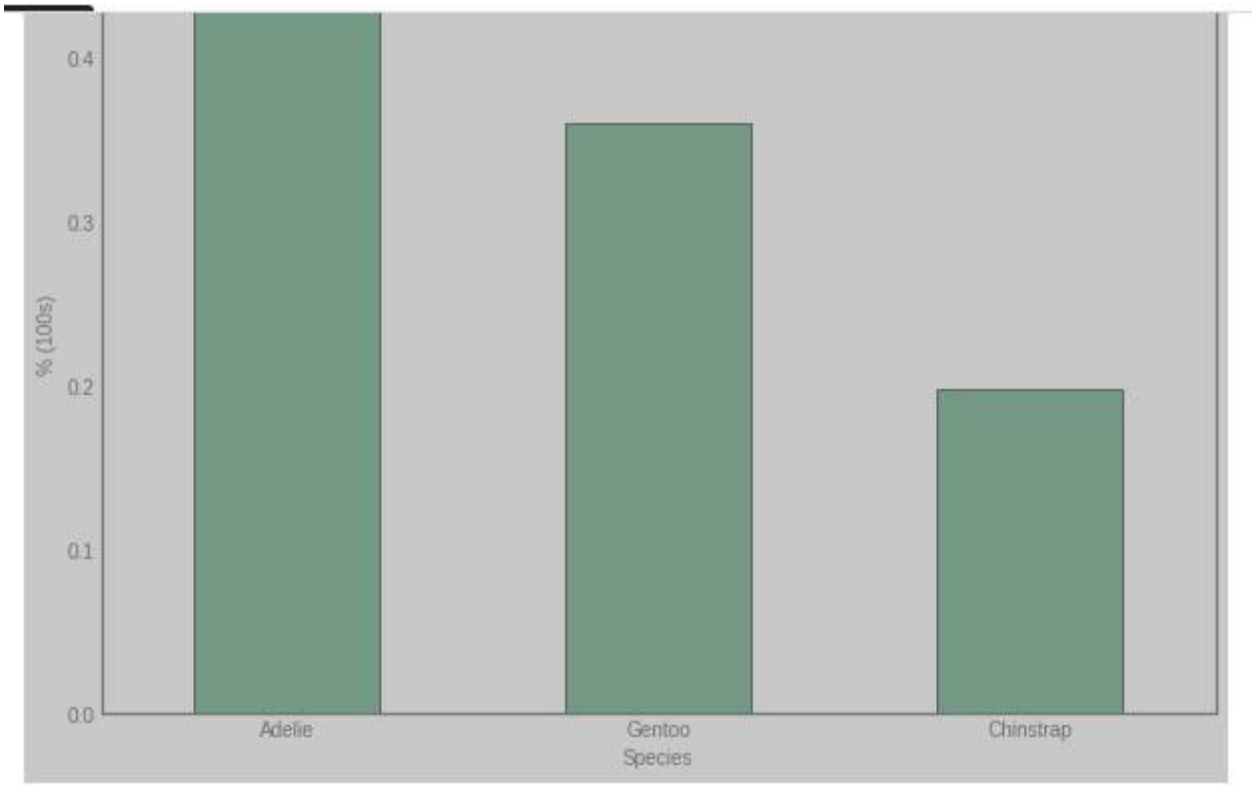

```
df['sex'].value_counts(normalize = True).plot(kind = 'bar', color = 'seagreen',
linewidth = 1, edgecolor = 'k')
plt.title('Penguins - Sex')
plt.xlabel('Sex')
plt.ylabel('% (100s)')
plt.xticks(rotation = 360)
plt.show()
```

Let me write a simple function which can plot both ECDF and PDF.

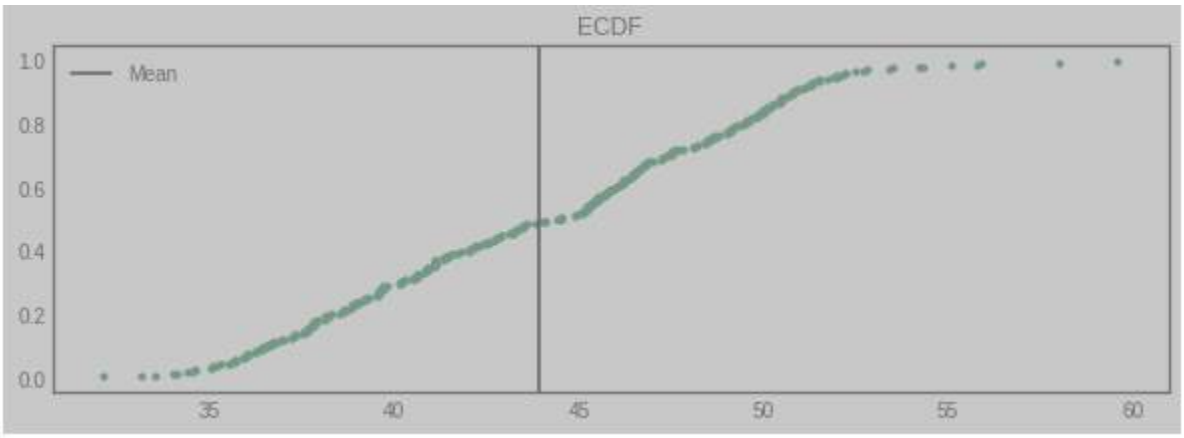
```
In [12]:
linkcode
def ecdf(x):
    n = len(x)
    a = np.sort(x)
    b = np.arange(1, 1 + n) / n
    plt.subplot(211)
    plt.plot(a, b, marker = '.', linestyle = 'None', c = 'seagreen')
    mean_x = np.mean(x)
    plt.axvline(mean_x, c = 'k', label = 'Mean')
    plt.title('ECDF')
    plt.legend()
    plt.show()
    plt.subplot(212)
    sns.distplot(x, color = 'r')
    plt.title('Probability Density Function')
    plt.show()
```

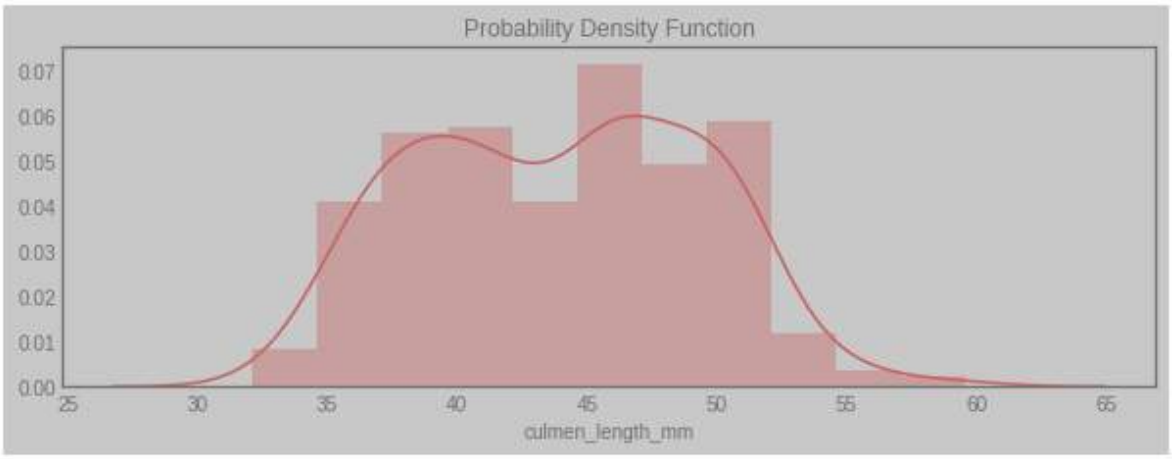
In [13]:

```
ecdf(df['culmen_length_mm'])
```

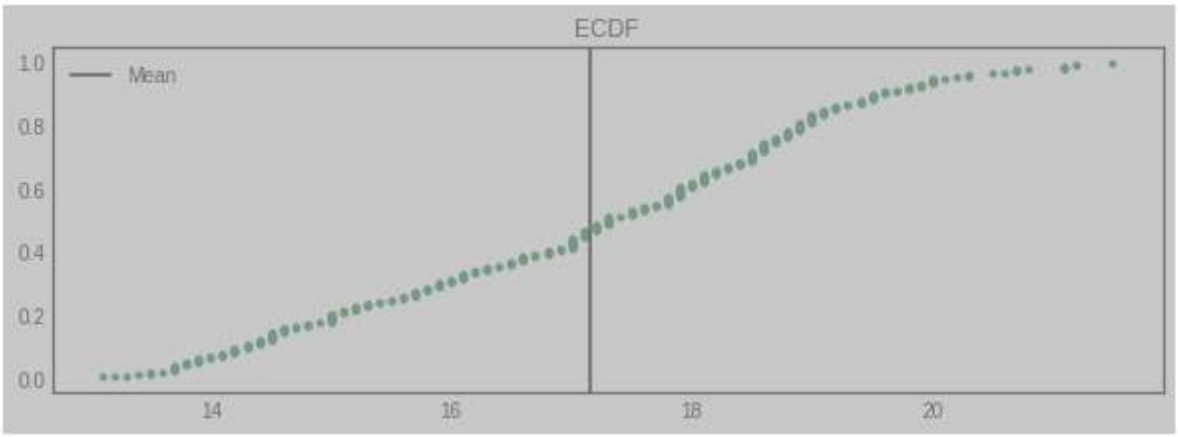


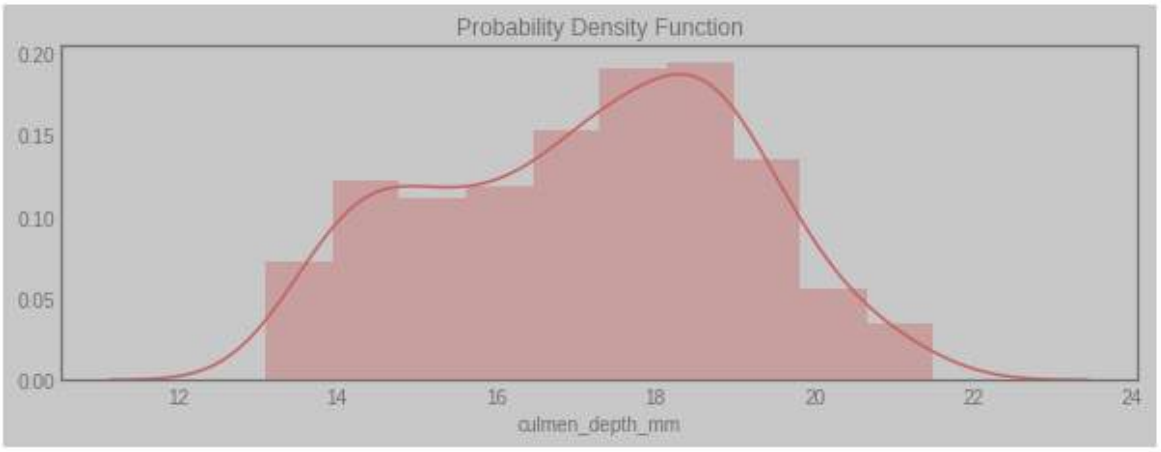

```
ecdf(df['culmen_length_mm'])
```



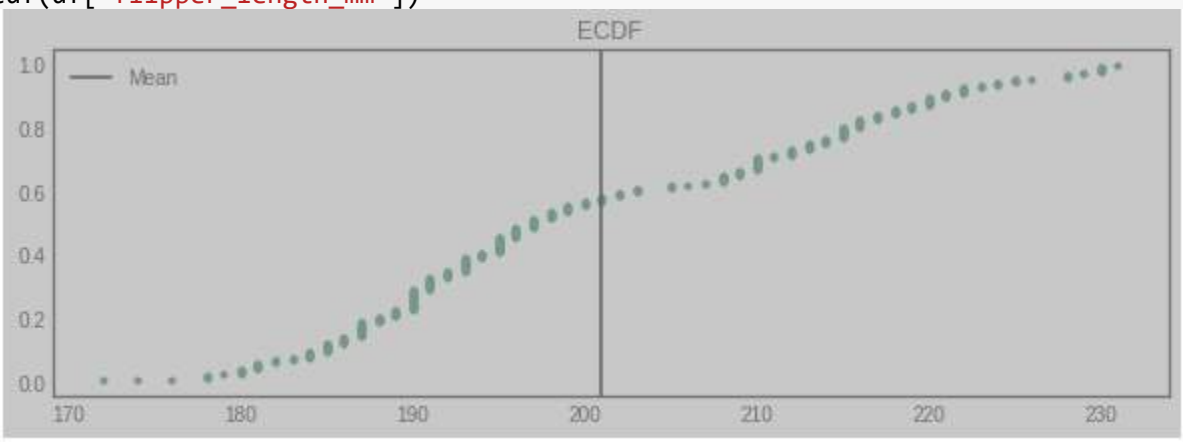


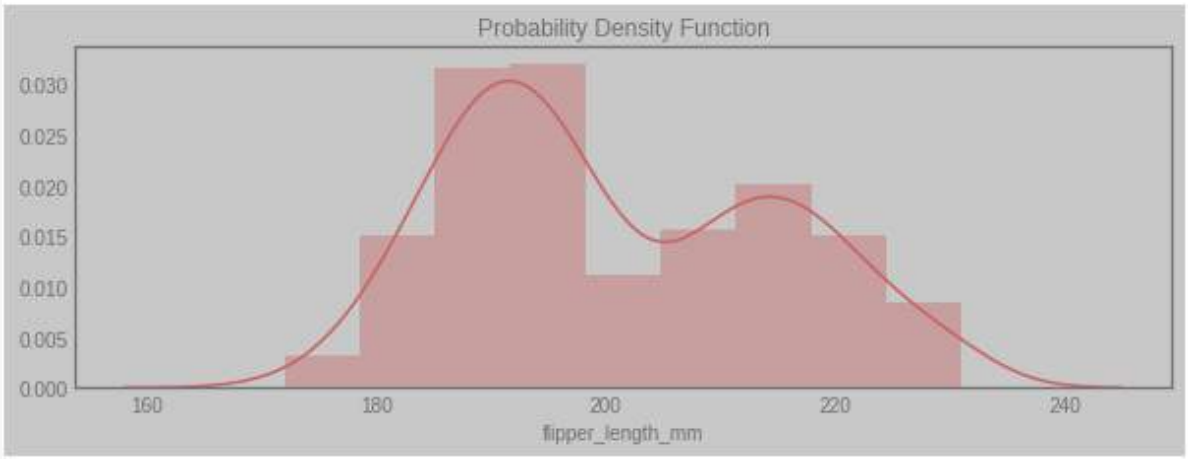

```
ecdf(df['culmen_depth_mm'])
```



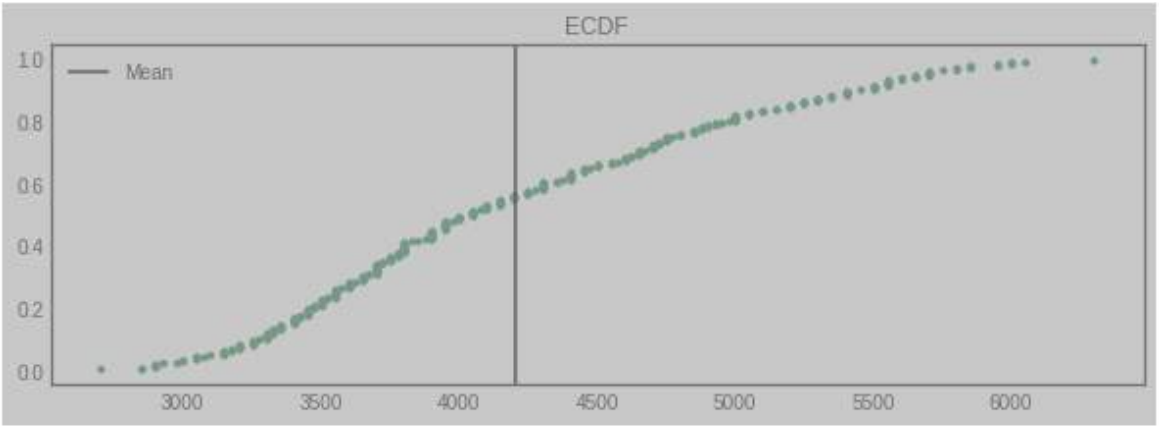


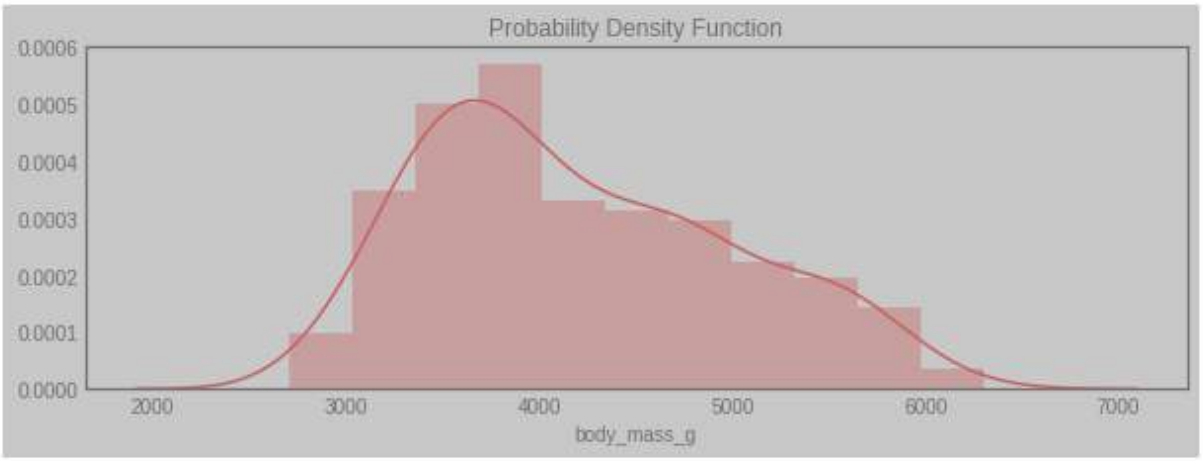

```
ecdf(df['flipper_length_mm'])
```






```
ecdf(df['body_mass_g'])
```

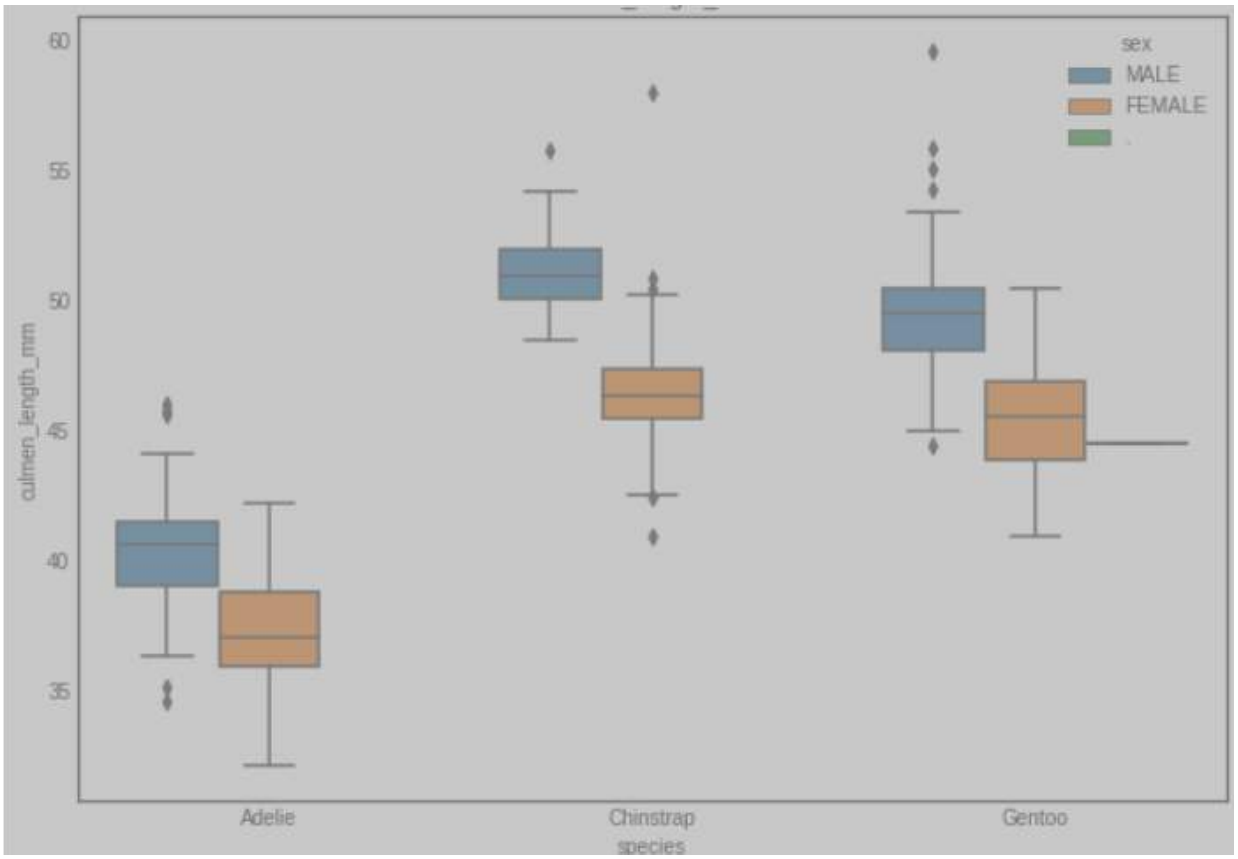




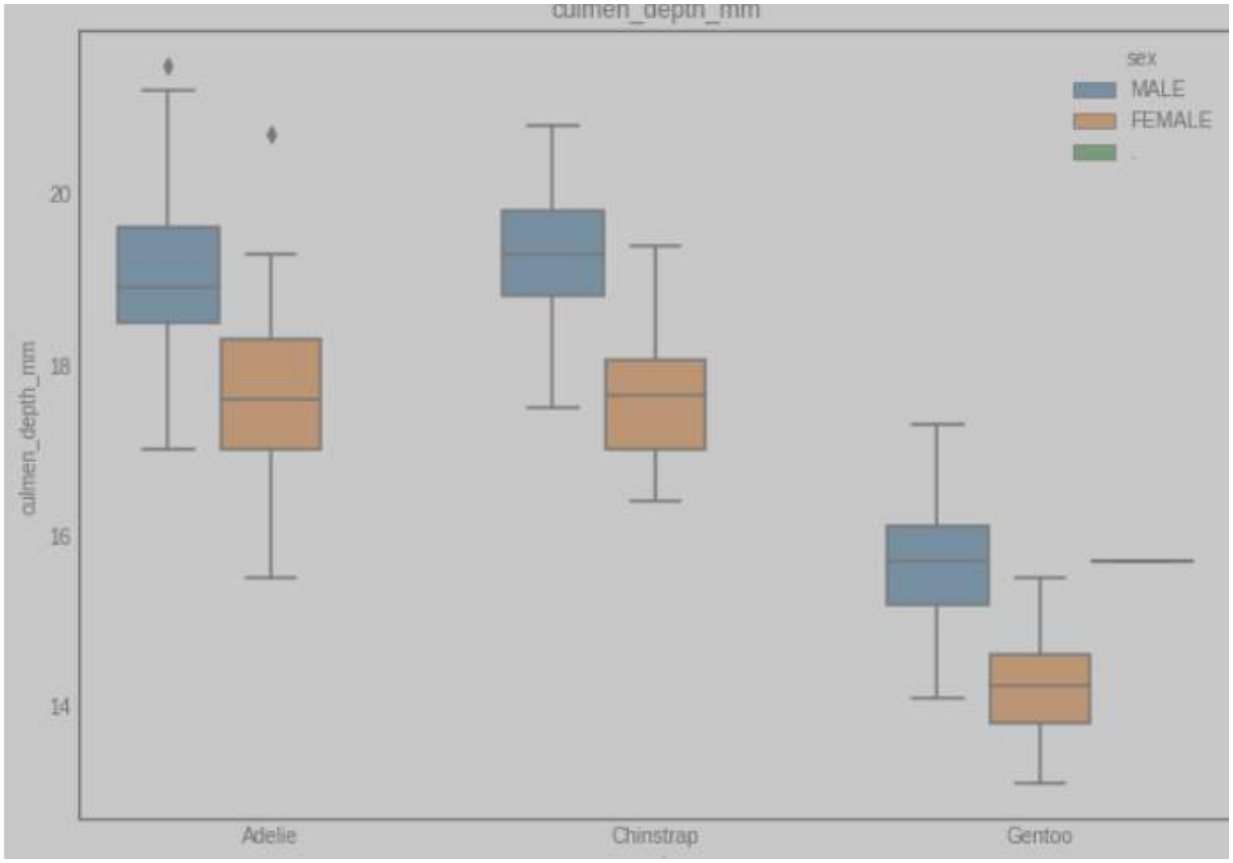
Multivariate Analysis

```
def box(f):
    sns.boxplot(y = f, x = 'species', hue = 'sex',data = df)
    plt.title(f)
    plt.show()
```

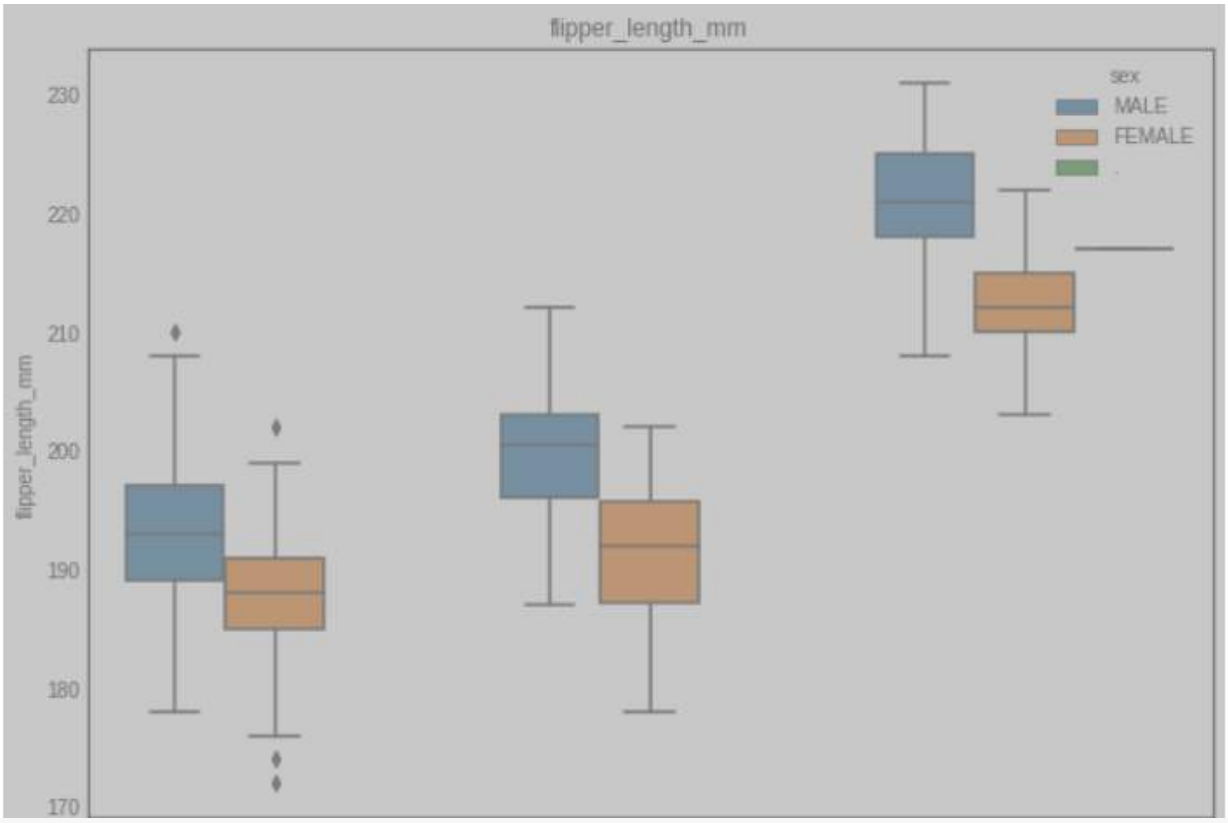
```
In [18]:
box('culmen_length_mm')
```



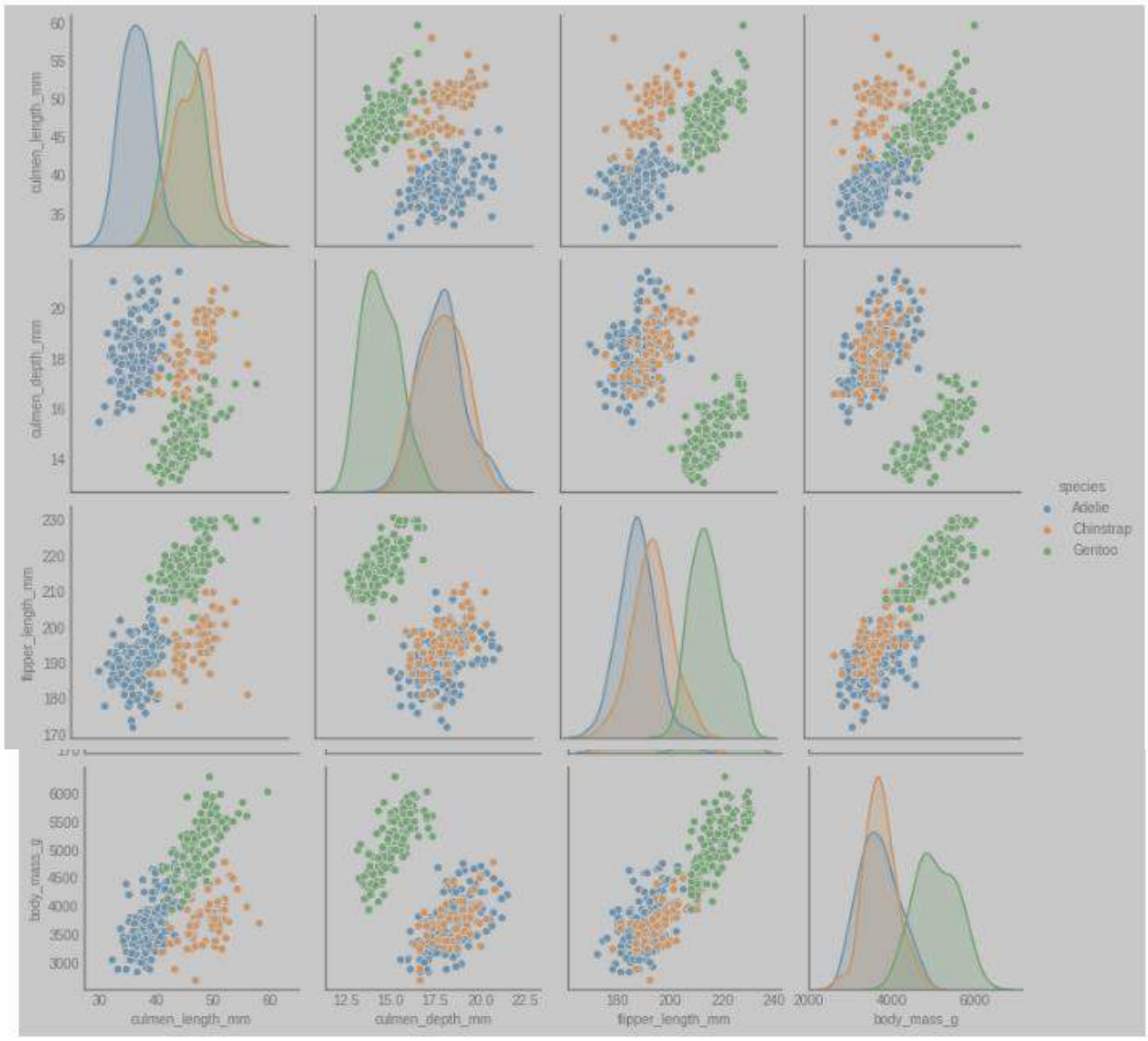

```
box('culmen_depth_mm')
```



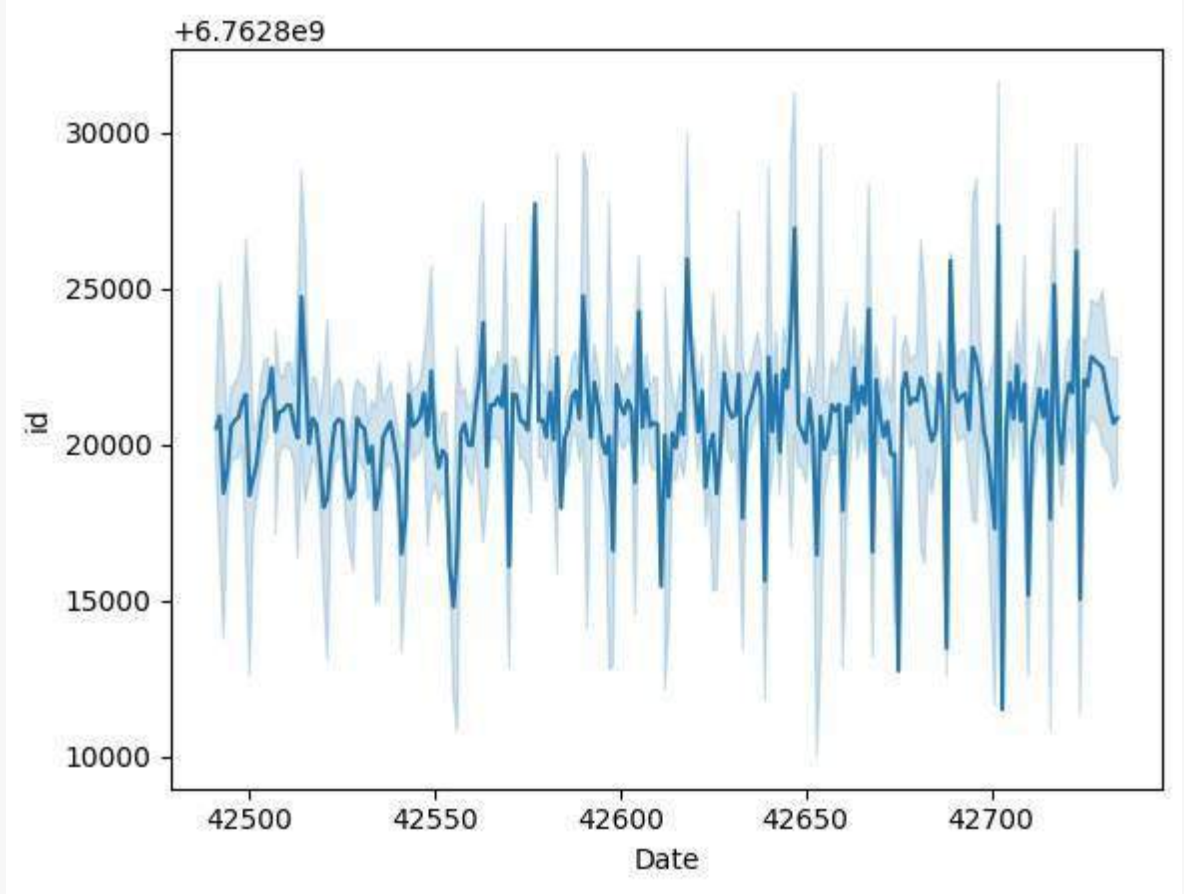

```
box('flipper_length_mm')
```



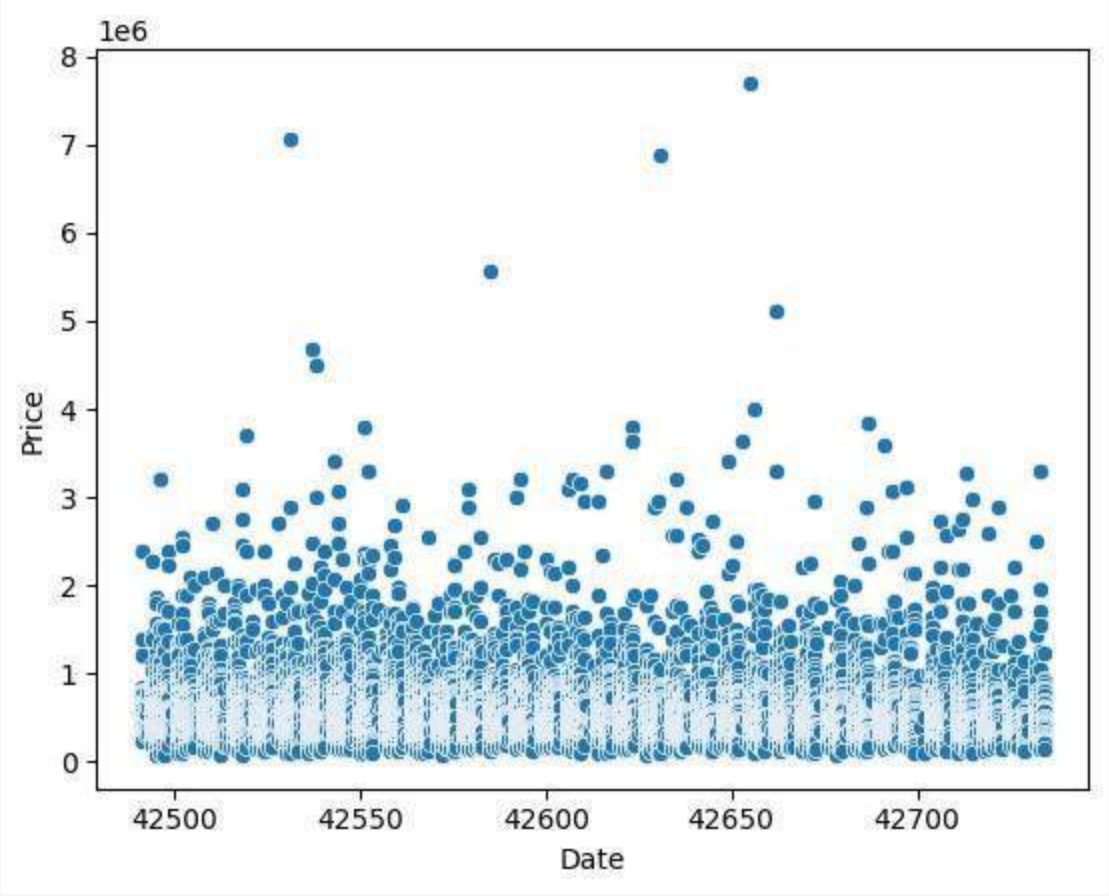

```
box('body_mass_g')
sns.pairplot(df, hue = 'species')
plt.show()
```




```
##Bivarient sns.lineplot(x=df.Date,y=df.id) <Axes: xlabel='Date',
ylabel='id'>
```

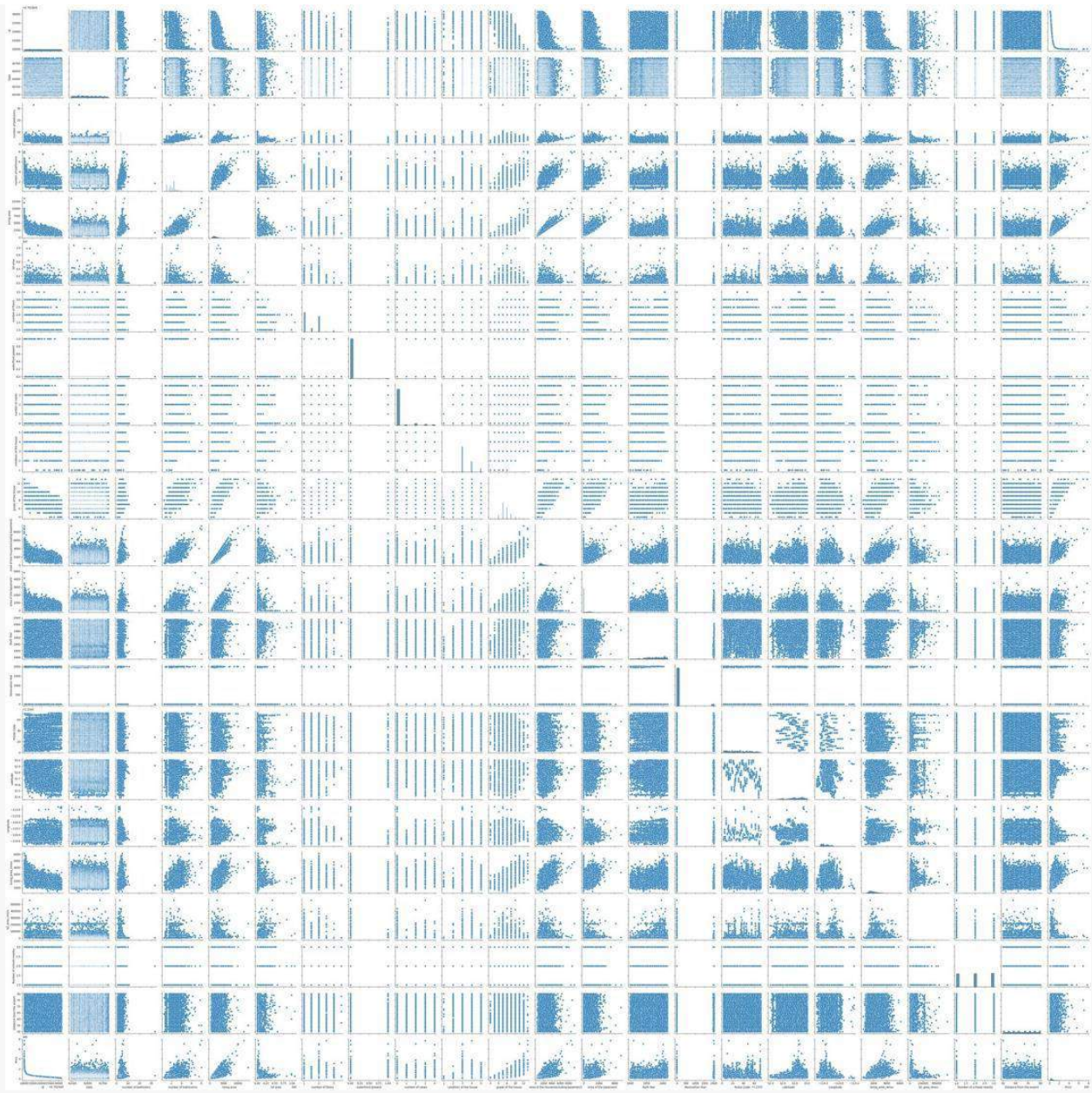



```
sns.scatterplot(x=df.Date,y=df.Price) <Axes: xlabel='Date',
ylabel='Price'>
```



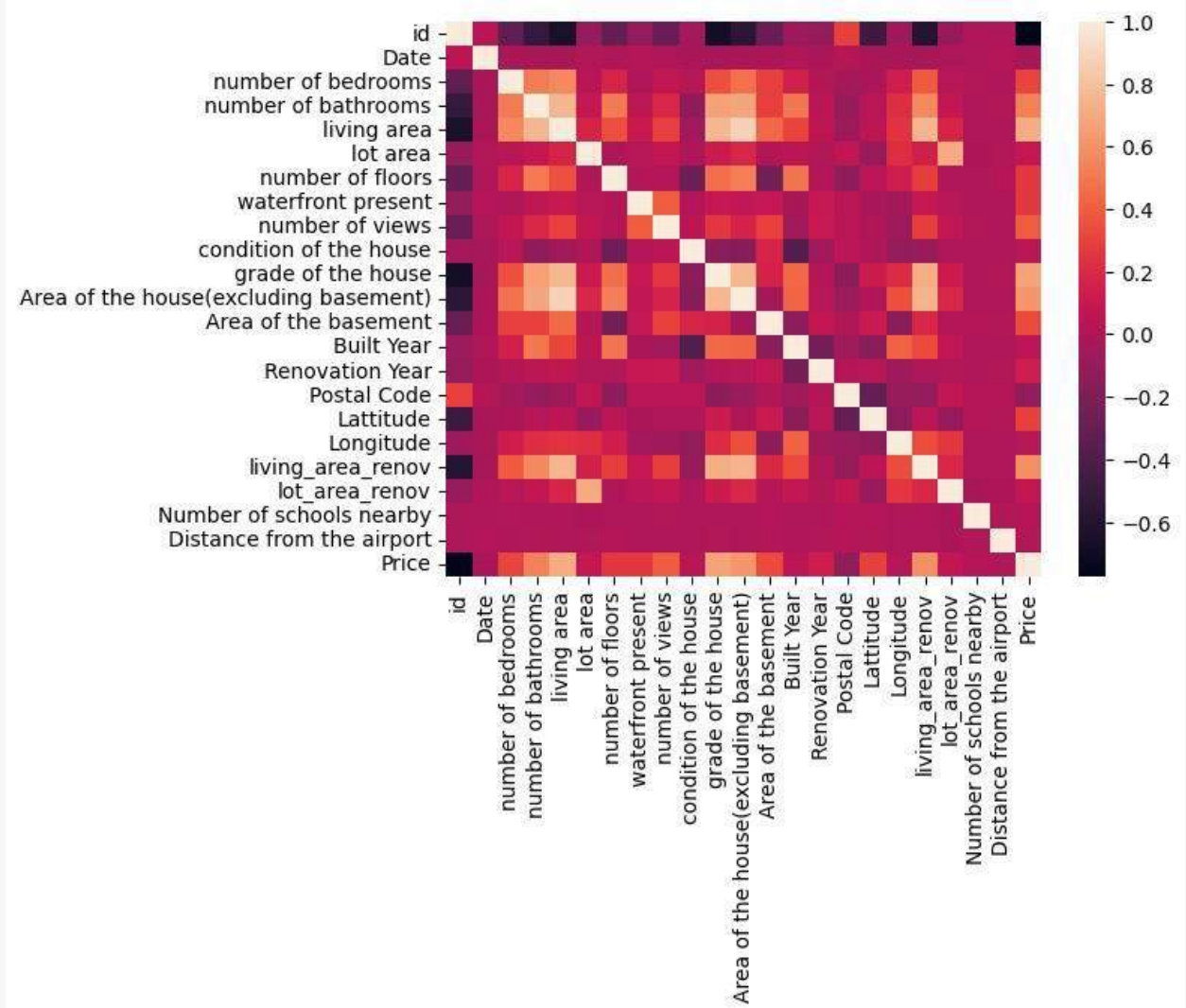
##Multivarient Analysis `sns.pairplot(df)`

```
<seaborn.axisgrid.PairGrid at 0x7ff2beaba940>
```



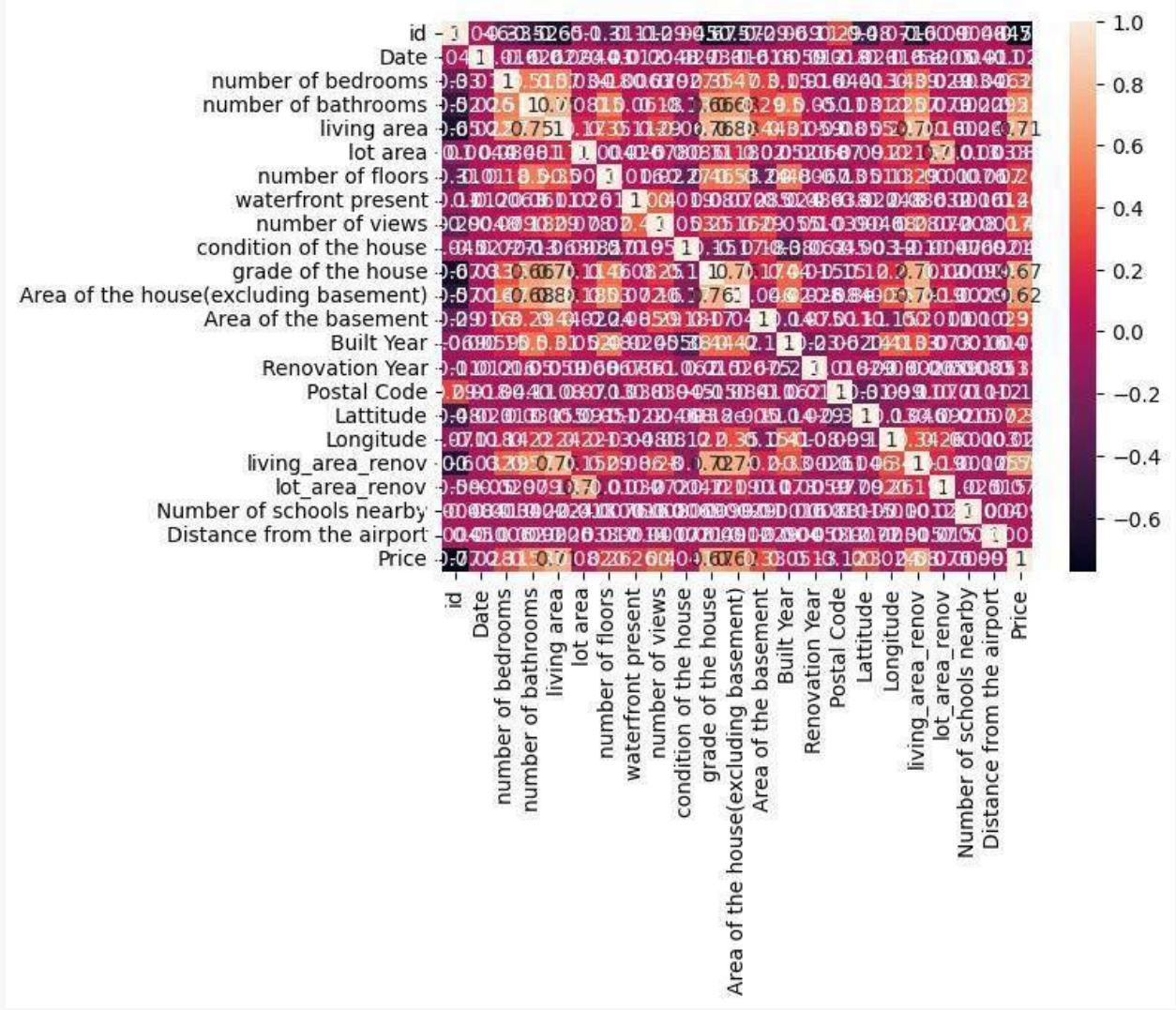
```
sns.heatmap(df.corr())
```

```
<Axes: >
```




```
sns.heatmap(df.corr(),annot=True)
```

```
<Axes: >
```



#Descriptive statistic on the dataset

```
df.describe() #descriptive statistics
id Date number of bedrooms number of bathrooms \
count 1.462000e+04 14620.000000 14620.000000
14620.000000
mean 6.762821e+09 42604.538646 3.379343
2.129583
std 6.237575e+03 67.347991 0.938719
0.769934
min 6.762810e+09 42491.000000 1.000000
0.500000
25% 6.762815e+09 42546.000000 3.000000
1.750000
50% 6.762821e+09 42600.000000 3.000000
2.250000
75% 6.762826e+09 42662.000000 4.000000
2.500000
```

```
max 6.762832e+09 42734.000000 33.000000 8.000000
living area lot area number of floors waterfront present \
count 14620.000000 1.462000e+04 14620.000000
14620.000000
mean 2098.262996 1.509328e+04 1.502360
0.007661
std 928.275721 3.791962e+04 0.540239
0.087193
min 370.000000 5.200000e+02 1.000000
0.000000
25% 1440.000000 5.010750e+03 1.000000
0.000000
50% 1930.000000 7.620000e+03 1.500000
0.000000
75% 2570.000000 1.080000e+04 2.000000
0.000000
max 13540.000000 1.074218e+06 3.500000
1.000000
number of views condition of the house ... Built Year \ count
14620.000000 14620.000000 ... 14620.000000 mean 0.233105 3.430506 ...
1970.926402 std 0.766259 0.664151 ... 29.493625 min 0.000000 1.000000
... 1900.000000 25% 0.000000 3.000000 ... 1951.000000
50% 0.000000 3.000000 ... 1975.000000 75% 0.000000 4.000000 ...
1997.000000 max 4.000000 5.000000 ... 2015.000000
Renovation Year Postal Code Lattitude Longitude \ count 14620.000000
14620.000000 14620.000000 14620.000000 mean 90.924008 122033.062244
52.792848 -114.404007 std 416.216661 19.082418 0.137522 0.141326 min
0.000000 122003.000000 52.385900 -114.709000 25% 0.000000
122017.000000 52.707600 -114.519000
50% 0.000000 122032.000000 52.806400 -114.421000 75% 0.000000
122048.000000 52.908900 -114.315000 max 2015.000000 122072.000000
53.007600 -113.505000
living_area_renov lot_area_renov Number of schools nearby \ count
14620.000000 14620.000000 14620.000000 mean 1996.702257 12753.500068
2.012244 std 691.093366 26058.414467 0.817284 min 460.000000
651.000000 1.000000 25% 1490.000000 5097.750000 1.000000
50% 1850.000000 7620.000000 2.000000 75% 2380.000000 10125.000000
3.000000
```

max 6110.000000 560617.000000 3.000000 Distance from the airport Price
count 14620.000000 1.462000e+04 mean 64.950958 5.389322e+05 std
8.936008 3.675324e+05 min 50.000000 7.800000e+04 25% 57.000000
3.200000e+05
50% 65.000000 4.500000e+05 75% 73.000000 6.450000e+05 max 80.000000
7.700000e+06
[8 rows x 23 columns] df.head()
id Date number of bedrooms number of bathrooms living area \
0 6762810145 42491 5 2.50

3650
1 6762810635 42491 4 2.50

2920
2 6762810998 42491 5 2.75

2910
3 6762812605 42491 4 2.50

3310
4 6762812919 42491 3 2.00

2710
lot area number of floors waterfront present number of views \ 0 9050
2.0 0 4
1 4000 1.5 0 0

```
2 9480 1.5 0 0 3 42998 2.0 0 0 4 4500 1.5 0 0

condition of the house ... Built Year Renovation Year Postal
Code \
0 5 ... 1921 0

122003
1 5 ... 1909 0

122004
2 3 ... 1939 0

122004
3 3 ... 2001 0

122005
4 4 ... 1929 0

122006
```
Lattitude Longitude living_area_renov lot_area_renov \

```
0 52.8645 -114.557 2880 5400
1 52.8878 -114.470 2470 4000
2 52.8852 -114.468 2940 6600
3 52.9532 -114.321 3350 42847
4 52.9047 -114.485 2060 4500

Number of schools nearby Distance from the airport Price 0 2 58
2380000
1 2 51 1400000
2 1 53 1200000
3 3 76 838000
4 1 51 805000

[5 rows x 23 columns] df.living_area_renov.unique()
array([2880, 2470, 2940, 3350, 2060, 2380, 3320, 1570, 2010, 2320,
2820,
1910, 2390, 2410, 1300, 2730, 1860, 4050, 2570, 2200, 2590,
2860,
1090, 3000, 1340, 2780, 2080, 2260, 2990, 1560, 1320, 1850,
1150,
1770, 2340, 1680, 1260, 1450, 2070, 2290, 1960, 2830, 1440,
1790,
1160, 1480, 1100, 2280, 1590, 1410, 2310, 1750, 2130, 1400,
1380,
1580, 3030, 1280, 1940, 1390, 2315, 2240, 2350, 2140, 4850,
1870,
2610, 2720, 3100, 4420, 4530, 3430, 2550, 1670, 3070, 2020,
3180,
2970, 1690, 2750, 2170, 3715, 1950, 2580, 1810, 3010, 1350,
1720,
1800, 2840, 2330, 1060, 2160, 2030, 1880, 1520, 2500, 1290,
1470,
1890, 1730, 2220, 1840, 2670, 1200, 1408, 1620, 1430, 1630,
1310,
1760, 1820, 1220, 1980, 1130, 1170, 1510, 1240, 2488, 3510,
2490,
2540, 2120, 2040, 3040, 3240, 3130, 3770, 2790, 2800, 2530,
2450,
2520, 2770, 2000, 1780, 2210, 1420, 1660, 1970, 1270, 1460,
1500,
1930, 1330, 1740, 1370, 2090, 1230, 2441, 840, 2360, 1650,
1490,
900, 820, 1700, 4100, 2960, 3470, 3820, 2430, 4130, 2190,
1990,
2250, 3200, 2850, 2560, 1640, 2870, 2510, 1180, 2600, 1540,
1250,
```

1040, 1360, 1516, 2230, 2440, 2011, 1010, 1140, 1070, 910,
1326,
3450, 2930, 2900, 3260, 2920, 2950, 3620, 1900, 1210, 3140, 2300,
1190, 2527, 2150, 2980, 1920, 1600, 1357, 1572, 4460, 3890, 3660,
3230, 3500, 3080, 3880, 2700, 2690, 2100, 2270, 1110, 1439, 998,
1714, 1610, 1550, 1020, 3220, 4760, 2890, 3530, 2400, 3600,
2480,
3170, 3640, 2370, 980, 1080, 1120, 1830, 890, 1710, 3740,
4040,
4240, 4440, 3290, 2180, 3120, 990, 2650, 3060, 1364, 2420,
3480,
4560, 3210, 3390, 3360, 2910, 950, 920, 1030, 1530, 3860,
4210,
3700, 2740, 2810, 2460, 2660, 1232, 850, 3490, 3150, 1445,
2114,
1404, 3910, 3160, 3580, 2760, 930, 3300, 5170, 4060, 3920,
3610,
2303, 1862, 1050, 3850, 3840, 1000, 2110, 2680, 2050, 2620,
3790,
2415, 3440, 2640, 3110, 2052, 2095, 3630, 2710, 3270, 5030,
3680,
970, 1571, 1307, 1658, 3540, 4290, 2358, 3370, 1665, 3494,
2434,
860, 880, 3930, 3710, 4140, 1365, 4020, 3690, 3750, 3590,
1346,
3330, 2630, 1518, 3190, 1495, 2305, 3730, 2037, 2363, 1765,
3810,
4090, 3280, 4390, 2027, 960, 2437, 770, 700, 4900, 3960,
3050,
2578, 1484, 2583, 1914, 4280, 2412, 4070, 3380, 1405, 1811,
3250,
3550, 2518, 3020, 2106, 2009, 1188, 4630, 3800, 4670, 3950,
1295,
2478, 740, 3310, 4180, 2683, 2955, 4000, 3400, 3900, 3670,
3780,
4400, 3420, 830, 460, 1256, 1494, 1098, 3720, 3560, 2028,
1459,
1584, 3340, 2496, 1934, 2456, 4470, 4170, 3980, 1798, 2376,
2594,
2214, 1768, 4550, 4010, 2554, 4950, 1277, 1156, 940, 2667,
5080,
5790, 3830, 3639, 1664, 1481, 4080, 2502, 4620, 3410, 3090,
3618,
2912, 2238, 1078, 5070, 3970, 4490, 3570, 2516, 780, 1767,
4160,

3760, 3520, 2566, 1678, 4920, 3650, 4510, 4030, 3625, 2165,
2156,
2641, 3460, 4340, 800, 4680, 4300, 2234, 760, 3990, 4640,
1746,
1569, 1696, 2815, 1309, 870, 2458, 4750, 3045, 1894, 2648, 1802,
2598, 2154, 2029, 1616, 2738, 2634, 2166, 2673, 1137, 4270,
4310,
1979, 1537, 1847, 4150, 2996, 1546, 1813, 2704, 5380, 3721,
4190,
2475, 790, 4362, 806, 4330, 2597, 1522, 1466, 1264, 2616,
1536,
4042, 4230, 2198, 2575, 4890, 3112, 1745, 1448, 2574, 2439,
1076,
810, 4913, 2798, 2189, 1528, 3940, 2533, 2622, 5200, 2056,
1458,
1509, 2382, 1975, 4120, 4110, 4590, 4690, 2451, 1984, 2323,
1358,
5600, 2142, 3191, 1336, 4320, 4830, 4225, 2474, 3425, 2316,
2688,
2112, 3557, 5110, 1716, 2725, 2396, 1981, 4930, 3008, 1554,
1442,
1463, 4480, 1638, 3236, 1138, 2876, 3193, 750, 2424, 2901,
4540,
1303, 1919, 2049, 2077, 1381, 710, 1282, 2612, 1941, 2136,
4370,
2875, 2555, 2304, 1443, 3159, 2767, 4940, 4570, 2425, 1268,
1399,
1356, 2221, 720, 4770, 2665, 3078, 2344, 2246, 1639, 2724,
2092,
2389, 2406, 1566, 1168, 670, 2419, 2014, 2879, 2015, 3543,
2619,
1092, 1608, 1884, 1691, 2927, 4800, 2495, 1845, 1763, 4410,
2873,
2258, 1427, 690, 620, 2405, 4200, 1415, 2547, 3087, 2091,
4650,
2822, 2961, 2647, 3870, 3726, 4600, 2765, 2242, 2728, 1056,
1429,
2604, 6110, 4220, 5340, 2255, 4730, 3413, 1886, 3515, 1321,
1677,
4250, 1425, 2697, 1654, 1162]) df.corr()
id Date number of bedrooms \
id 1.000000 0.045966 -
0.329034
Date 0.045966 1.000000 -

0.015663
number of bedrooms -0.329034 -0.015663
1.000000
number of bathrooms -0.516909 -0.026485
0.509784
living area -0.648127 -0.021958 0.570526
lot area -0.100269 0.004392
0.034416
number of floors -0.312305 -0.010335
0.177294
waterfront present -0.112937 0.012006 -
0.006257
number of views -0.293004 -0.004782
0.078665
condition of the house -0.045061 -0.027402
0.026597
grade of the house -0.673448 -0.033097
0.352945
Area of the house(excluding basement) -0.565116 -0.015994
0.473599
Area of the basement -0.290806 -0.015711
0.300332
Built Year -0.068645 -0.005869
0.152954
Renovation Year -0.109155 -0.011636
0.016132
Postal Code 0.294709 0.018243 -
0.044156
Lattitude -0.479334 -0.023327 -
0.013163
Longitude -0.070841 -0.018231
0.135712
living_area_renov -0.599900 -0.032495
0.389855
lot_area_renov -0.089604 -0.000050
0.029400
Number of schools nearby -0.004821 -0.004071
0.003397
Distance from the airport -0.004542 0.011457 -
0.006157
Price -0.773114 -0.027919
0.308460
number of bathrooms living area \
id -0.516909 -
0.648127

Date -0.026485 -
0.021958
number of bedrooms 0.509784
0.570526
number of bathrooms 1.000000
0.753517
living area 0.753517
1.000000
lot area 0.080806
0.174420
number of floors 0.502924
0.354743
waterfront present 0.060104
0.105837
number of views 0.183789
0.287728
condition of the house -0.128232 -
0.063358
grade of the house 0.663054
0.761835
Area of the house(excluding basement) 0.684391
0.875793
Area of the basement 0.287190
0.441491
Built Year 0.498127
0.309602
Renovation Year 0.049669
0.059400
Postal Code -0.105546 -
0.080303
Lattitude 0.031156
0.054518
Longitude 0.223904
0.240208
living_area_renov 0.570530
0.757571
lot_area_renov 0.078627
0.180312
Number of schools nearby 0.002180
0.002370
Distance from the airport 0.009206
0.002511
Price 0.531735
0.712169
lot area number of floors \ id -0.100269 -0.312305 Date 0.004392 -
0.010335

number of bedrooms 0.034416 0.177294 number of bathrooms 0.080806
0.502924 living area 0.174420 0.354743 lot area 1.000000 -0.004138
number of floors -0.004138 1.000000 waterfront present 0.026282
0.016316 number of views 0.078308 0.020153 condition of the house -
0.008548 -0.269928 grade of the house 0.110546 0.463082 Area of the
house(excluding basement) 0.183553 0.525643
Area of the basement 0.019755 -0.242976 Built Year 0.051615 0.481565
Renovation Year 0.006848 0.006705
Postal Code 0.070131 -0.129788
Lattitude -0.090983 0.050731 Longitude 0.221432 0.127550
living_area_renov 0.149744 0.285093 lot_area_renov 0.706812 -0.010120
Number of schools nearby -0.012671 -0.007579
Distance from the airport 0.003291 0.016567 Price 0.081992 0.262732
waterfront present number of views \
id -0.112937 -
0.293004
Date 0.012006 -
0.004782
number of bedrooms -0.006257
0.078665
number of bathrooms 0.060104
0.183789
living area 0.105837
0.287728
lot area 0.026282
0.078308
number of floors 0.016316
0.020153
waterfront present 1.000000
0.400206
number of views 0.400206
1.000000
condition of the house 0.018644
0.052533
grade of the house 0.079831
0.254532
Area of the house(excluding basement) 0.071865
0.162672

Area of the basement 0.085441
0.293062
Built Year -0.024226 -
0.055357
Renovation Year 0.085865
0.102944
Postal Code 0.038318
0.039268
Lattitude -0.021795 -
0.004555
Longitude -0.047791 -
0.079706
living_area_renov 0.085743
0.281452
lot_area_renov 0.032055
0.072300
Number of schools nearby 0.001563
0.008004
Distance from the airport 0.001448 -
0.001657
Price 0.263687
0.395973
condition of the house ... \ id -0.045061 ... Date -0.027402 ...
number of bedrooms 0.026597 ... number of bathrooms -0.128232 ...
living area -0.063358 ... lot area -0.008548 ... number of floors -
0.269928 ... waterfront present 0.018644 ... number of views 0.052533
... condition of the house 1.000000 ... grade of the house -0.152530
... Area of the house(excluding basement) -0.167695 ...
Area of the basement 0.180609 ... Built Year -0.381718 ...
Renovation Year -0.062126 ...
Postal Code 0.045334 ...
Lattitude -0.002998 ... Longitude -0.121189 ... living_area_renov -
0.099743 ... lot_area_renov -0.004748 ... Number of schools nearby -
0.006939 ...
Distance from the airport -0.002136 ... Price 0.041376 ...

Built Year Renovation Year \ id -0.068645 -0.109155 Date -0.005869 -0.011636 number of bedrooms 0.152954 0.016132 number of bathrooms 0.498127 0.049669 living area 0.309602 0.059400 lot area 0.051615 0.006848 number of floors 0.481565 0.006705 waterfront present -0.024226 0.085865 number of views -0.055357 0.102944 condition of the house -0.381718 -0.062126 grade of the house 0.440358 0.014501 Area of the house(excluding basement) 0.419369 0.025727
Area of the basement -0.138843 0.075104 Built Year 1.000000 -0.233683 Renovation Year -0.233683 1.000000
Postal Code -0.062349 0.018006
Lattitude -0.143153 0.028908 Longitude 0.414591 -0.080050
living_area_renov 0.328625 -0.002601 lot_area_renov 0.072874 0.005869
Number of schools nearby -0.001631 -0.000826
Distance from the airport -0.003968 0.005342 Price 0.050307 0.133173
Postal Code Lattitude Longitude \
id 0.294709 -0.479334 -
0.070841
Date 0.018243 -0.023327 -
0.018231
number of bedrooms -0.044156 -0.013163
0.135712
number of bathrooms -0.105546 0.031156
0.223904
living area -0.080303 0.054518
0.240208
lot area 0.070131 -0.090983
0.221432
number of floors -0.129788 0.050731
0.127550
waterfront present 0.038318 -0.021795 -
0.047791
number of views 0.039268 -0.004555 -
0.079706
condition of the house 0.045334 -0.002998 -
0.121189
grade of the house -0.146342 0.115256
0.203754

```
Area of the house(excluding basement) -0.083730 -0.000088
0.345899
Area of the basement -0.010542 0.112989 -
0.145879
Built Year -0.062349 -0.143153
0.414591
Renovation Year 0.018006 0.028908 -
0.080050
Postal Code 1.000000 -0.310172 -
0.099003
Lattitude -0.310172 1.000000 -
0.131472
Longitude -0.099003 -0.131472
1.000000
living_area_renov -0.108454 0.046148 0.341221
lot_area_renov 0.077483 -0.091622
0.258066
Number of schools nearby 0.010605 0.014949 -
0.010163
Distance from the airport 0.011528 0.007193 -
0.003100
Price -0.115908 0.297490
0.024414
living_area_renov lot_area_renov \
id -0.599900 -
0.089604
Date -0.032495 -
0.000050
number of bedrooms 0.389855
0.029400
number of bathrooms 0.570530
0.078627
living area 0.757571
0.180312
lot area 0.149744
0.706812
number of floors 0.285093 -
0.010120
waterfront present 0.085743
0.032055
number of views 0.281452
0.072300
condition of the house -0.099743 -
0.004748
grade of the house 0.720019
```

0.116725
Area of the house(excluding basement) 0.737744
0.194670
Area of the basement 0.196403
0.011283
Built Year 0.328625
0.072874
Renovation Year -0.002601
0.005869
Postal Code -0.108454
0.077483
Lattitude 0.046148 -
0.091622
Longitude 0.341221
0.258066
living_area_renov 1.000000
0.189225
lot_area_renov 0.189225
1.000000
Number of schools nearby -0.001203 -
0.025014
Distance from the airport -0.005673 -
0.014587
Price 0.584924
0.075535
Number of schools nearby \ id -0.004821 Date -0.004071 number of bedrooms 0.003397 number of bathrooms 0.002180 living area 0.002370 lot area -0.012671 number of floors -0.007579 waterfront present 0.001563 number of views 0.008004 condition of the house -0.006939 grade of the house 0.000986 Area of the house(excluding basement) -0.002894
Area of the basement 0.010284 Built Year -0.001631
Renovation Year -0.000826
Postal Code 0.010605
Lattitude 0.014949 Longitude -0.010163 living_area_renov -0.001203 lot_area_renov -0.025014 Number of schools nearby 1.000000

Distance from the airport 0.004035 Price 0.009890
Distance from the airport Price
id -0.004542 -
0.773114
Date 0.011457 -
0.027919
number of bedrooms -0.006157
0.308460
number of bathrooms 0.009206
0.531735
living area 0.002511
0.712169
lot area 0.003291
0.081992
number of floors 0.016567
0.262732
waterfront present 0.001448
0.263687
number of views -0.001657
0.395973
condition of the house -0.002136
0.041376
grade of the house 0.004940
0.671814
Area of the house(excluding basement) 0.001222
0.615220
Area of the basement 0.002926
0.330202
Built Year -0.003968
0.050307
Renovation Year 0.005342
0.133173
Postal Code 0.011528 -
0.115908
Lattitude 0.007193
0.297490
Longitude -0.003100
0.024414
living_area_renov -0.005673
0.584924
lot_area_renov -0.014587
0.075535
Number of schools nearby 0.004035
0.009890
Distance from the airport 1.000000

```
0.003804
Price 0.003804
1.000000
[23 rows x 23 columns]
```

Missing Values

```
missing= missing_values_table(df)
missing
```

```
Your selected dataframe has 7 columns.
There are 5 columns that have missing values.
```

Out[9]:

	Missing Values	% of Total Values
sex	10	2.9
culmen_length_mm	2	0.6
culmen_depth_mm	2	0.6
flipper_length_mm	2	0.6
body_mass_g	2	0.6

In [10]:

```
# Handling missing values

from sklearn.impute import SimpleImputer
#setting strategy to 'most frequent' to impute by the mean
imputer = SimpleImputer(strategy='most_frequent')# strategy can also be mean or
median
df.iloc[:,:] = imputer.fit_transform(df)
```

In [11]:

```
df.isnull().sum()
```

Out[11]:

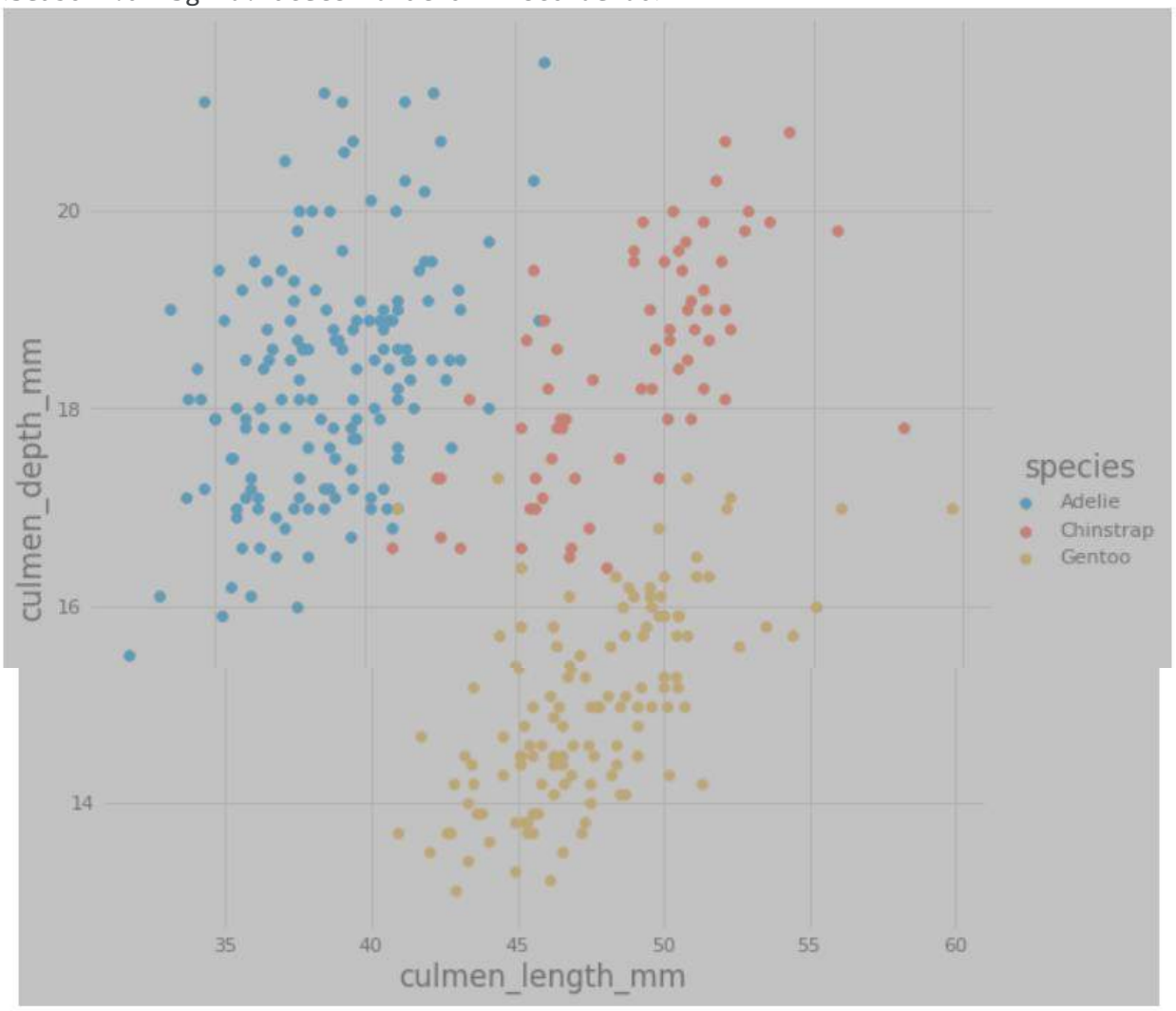
```
species              0
island               0
culmen_length_mm     0
culmen_depth_mm      0
flipper_length_mm    0
```

```
body_mass_g          0
sex                  0
dtype: int64
```

culmen_depth vs culmen_length

```
In [17]:
sns.FacetGrid(df, hue="species", size=8) \
   .map(plt.scatter, "culmen_length_mm", "culmen_depth_mm") \
   .add_legend()
```

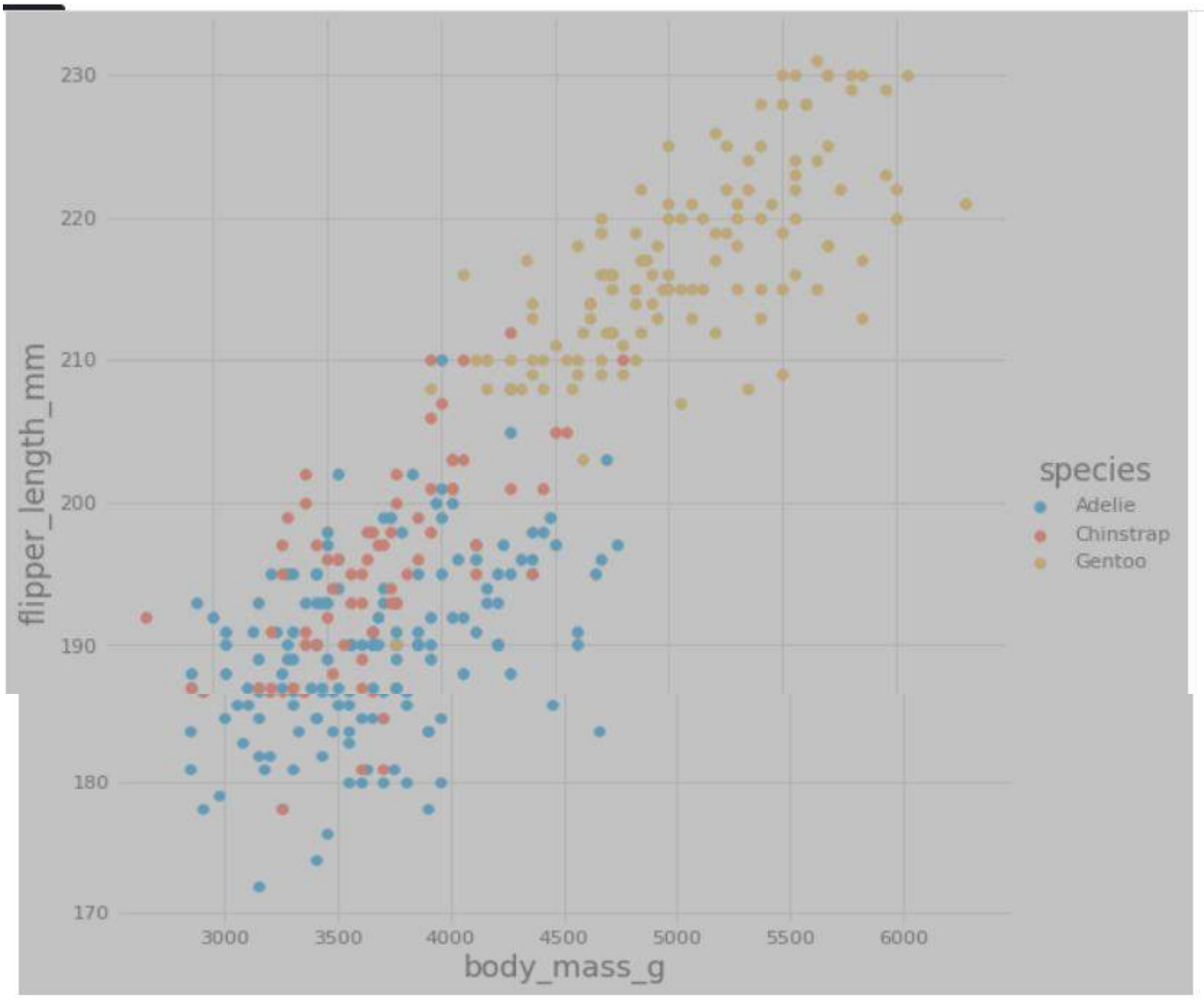
```
Out[17]:
<seaborn.axisgrid.FacetGrid at 0x7ff8c02de7d0>
```



Flipper length vs. body mass

```
In [21]:
linkcode
sns.FacetGrid(df, hue="species", size=8) \
   .map(plt.scatter, "body_mass_g", "flipper_length_mm") \
   .add_legend()
```

Out[21]:
<seaborn.axisgrid.FacetGrid at 0x7ff8c02b7ed0>



Model Building

```
from sklearn.model_selection import train_test_split, KFold, cross_val_score
from sklearn.metrics import accuracy_score, f1_score, recall_score, precision_score,
confusion_matrix

from sklearn.linear_model import LogisticRegression
from sklearn.tree import DecisionTreeClassifier
from sklearn.ensemble import RandomForestClassifier
from sklearn.neighbors import KNeighborsClassifier
from sklearn.svm import SVC
```

In [36]:

```
X = new_df_dummy.drop(columns = ['species', 'sex_FEMALE', 'sex_MALE'])
```

```
Y = new_df_dummy['species']
```
In [37]:

```
X_train, X_test, Y_train, Y_test = train_test_split(X, Y, test_size = 0.25,
random_state = 123)
```
Let's first try with a simple Logistic Regression model.

In [38]:

```
LR = LogisticRegression()
LR.fit(X_train, Y_train)

pred = LR.predict(X_test)
```
In [39]:

```
print('Accuracy : ', accuracy_score(Y_test, pred))
print('F1 Score : ', f1_score(Y_test, pred, average = 'weighted'))
Accuracy :  1.0
F1 Score :  1.0
```
This turned out to be a cool task! Let's try cross validation with different models and then pick up one.

In [40]:

```
models = []
models.append(('LR', LogisticRegression()))
models.append(('DT', DecisionTreeClassifier()))
models.append(('RF', RandomForestClassifier()))
models.append(('kNN', KNeighborsClassifier()))
models.append(('SVC', SVC()))
```
In [41]:

```
for name, model in models:
    kfold = KFold(n_splits = 5, random_state = 42)
    cv_res = cross_val_score(model, X_train, Y_train, scoring = 'accuracy', cv =
kfold)
    print(name, ' : ', cv_res.mean())
LR  :  0.9846153846153847
DT  :  0.9496229260935143
RF  :  0.9612368024132729
kNN  :  0.9846153846153847
SVC  :  0.9961538461538462
```
In [42]:

```
svc = SVC()
svc.fit(X_train, Y_train)

pred = LR.predict(X_test)
```
Model Evaluation

In [43]:

```
print('Accuracy : ', accuracy_score(Y_test, pred))
```

```
print('F1 Score : ', f1_score(Y_test, pred, average = 'weighted'))
print('Precision : ', precision_score(Y_test, pred , average = 'weighted'))
print('Recall : ', recall_score(Y_test, pred, average = 'weighted'))
```

```
Accuracy :  1.0
F1 Score :  1.0
Precision :  1.0
Recall :  1.0
```

In [44]:

```
confusion_matrix(Y_test, pred)
```

Out[44]:

```
array([[39,  0,  0],
       [ 0, 19,  0],
       [ 0,  0, 28]])
```