In [1]:

```
import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
import warnings
warnings.filterwarnings('ignore')
```

In [2]:

```
df = pd.read_csv('penguins_binary_classification.csv')
```

In [3]:

```
df.head()
```

Out[3]:

```
   species     island  bill_length_mm  bill_depth_mm  flipper_length_mm  body_mass_g  year
0   Adelie  Torgersen            39.1           18.7              181.0       3750.0  2007
1   Adelie  Torgersen            39.5           17.4              186.0       3800.0  2007
2   Adelie  Torgersen            40.3           18.0              195.0       3250.0  2007
3   Adelie  Torgersen            36.7           19.3              193.0       3450.0  2007
4   Adelie  Torgersen            39.3           20.6              190.0       3650.0  2007
```

In [4]:

```
df.tail()
```

Out[4]:

```
     species  island  bill_length_mm  bill_depth_mm  flipper_length_mm  body_mass_g  year
269   Gentoo  Biscoe            47.2           13.7              214.0       4925.0  2009
270   Gentoo  Biscoe            46.8           14.3              215.0       4850.0  2009
271   Gentoo  Biscoe            50.4           15.7              222.0       5750.0  2009
272   Gentoo  Biscoe            45.2           14.8              212.0       5200.0  2009
273   Gentoo  Biscoe            49.9           16.1              213.0       5400.0  2009
```

In [5]:

```
df.describe()
```

Out[5]:

	bill_length_mm	bill_depth_mm	flipper_length_mm	body_mass_g	year
count	274.000000	274.000000	274.000000	274.000000	274.000000
mean	42.702920	16.836131	202.178832	4318.065693	2008.043796
std	5.195566	2.013410	15.047938	835.933105	0.806281
min	32.100000	13.100000	172.000000	2850.000000	2007.000000
25%	38.350000	15.000000	190.000000	3600.000000	2007.000000
50%	42.000000	17.000000	198.000000	4262.500000	2008.000000
75%	46.675000	18.500000	215.000000	4950.000000	2009.000000
max	59.600000	21.500000	231.000000	6300.000000	2009.000000

In [6]:

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

Out[6]:

```
species              0
island               0
bill_length_mm       0
bill_depth_mm        0
flipper_length_mm    0
body_mass_g          0
year                 0
dtype: int64
```

In [7]:

```
df.dtypes
```

Out[7]:

```
species               object
island                object
bill_length_mm       float64
bill_depth_mm        float64
flipper_length_mm    float64
body_mass_g          float64
year                   int64
dtype: object
```

In [8]:

```
df.columns
```

Out[8]:

```
Index(['species', 'island', 'bill_length_mm', 'bill_depth_mm',
       'flipper_length_mm', 'body_mass_g', 'year'],
      dtype='object')
```

In [9]:

```
df.shape
```

Out[9]:

```
(274, 7)
```

In [11]:

```
#from sklearn.preprocessing import LabelEncoder
#object_cols = df.select_dtypes(include=['object']).columns

#le = LabelEncoder()
#for col in object_cols:
    #df[col] = le.fit_transform(df[col])
```

In [10]:

```
df.head()
```

Out[10]:

	species	island	bill_length_mm	bill_depth_mm	flipper_length_mm	body_mass_g	year
0	Adelie	Torgersen	39.1	18.7	181.0	3750.0	2007
1	Adelie	Torgersen	39.5	17.4	186.0	3800.0	2007
2	Adelie	Torgersen	40.3	18.0	195.0	3250.0	2007
3	Adelie	Torgersen	36.7	19.3	193.0	3450.0	2007
4	Adelie	Torgersen	39.3	20.6	190.0	3650.0	2007

In [11]:

```
df['species'].unique()
```

Out[11]:

```
array(['Adelie', 'Gentoo'], dtype=object)
```

In [13]:

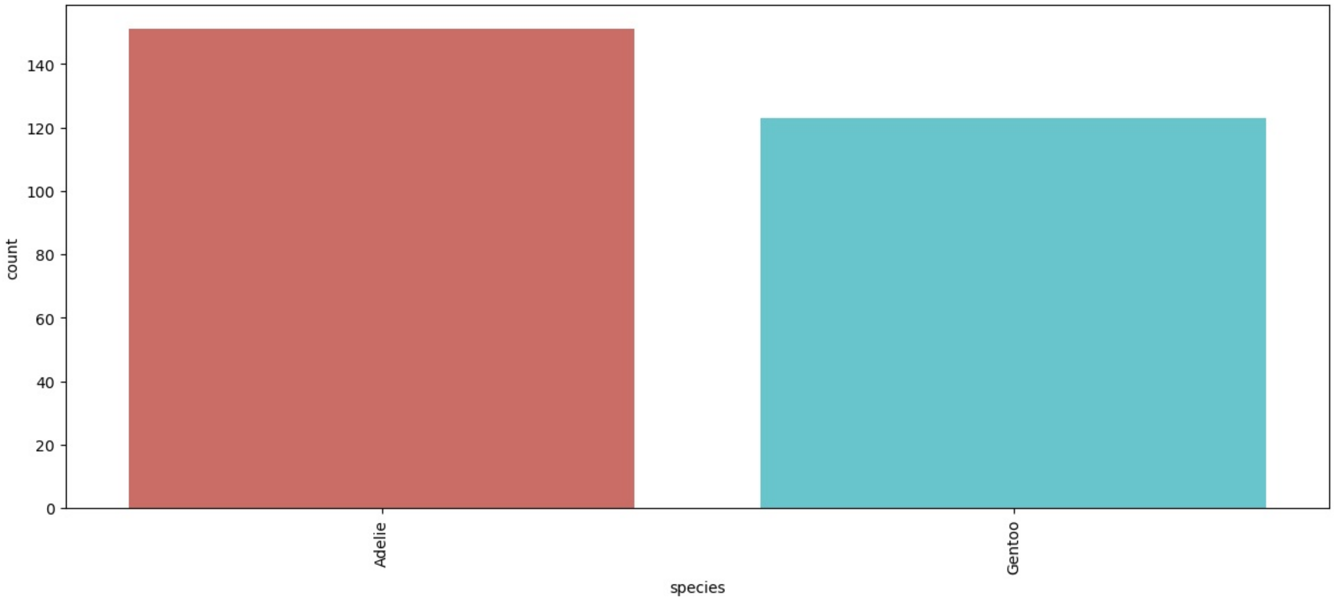
```
df['species'].value_counts()
```

Out[13]:

```
species
Adelie    151
Gentoo    123
Name: count, dtype: int64
```

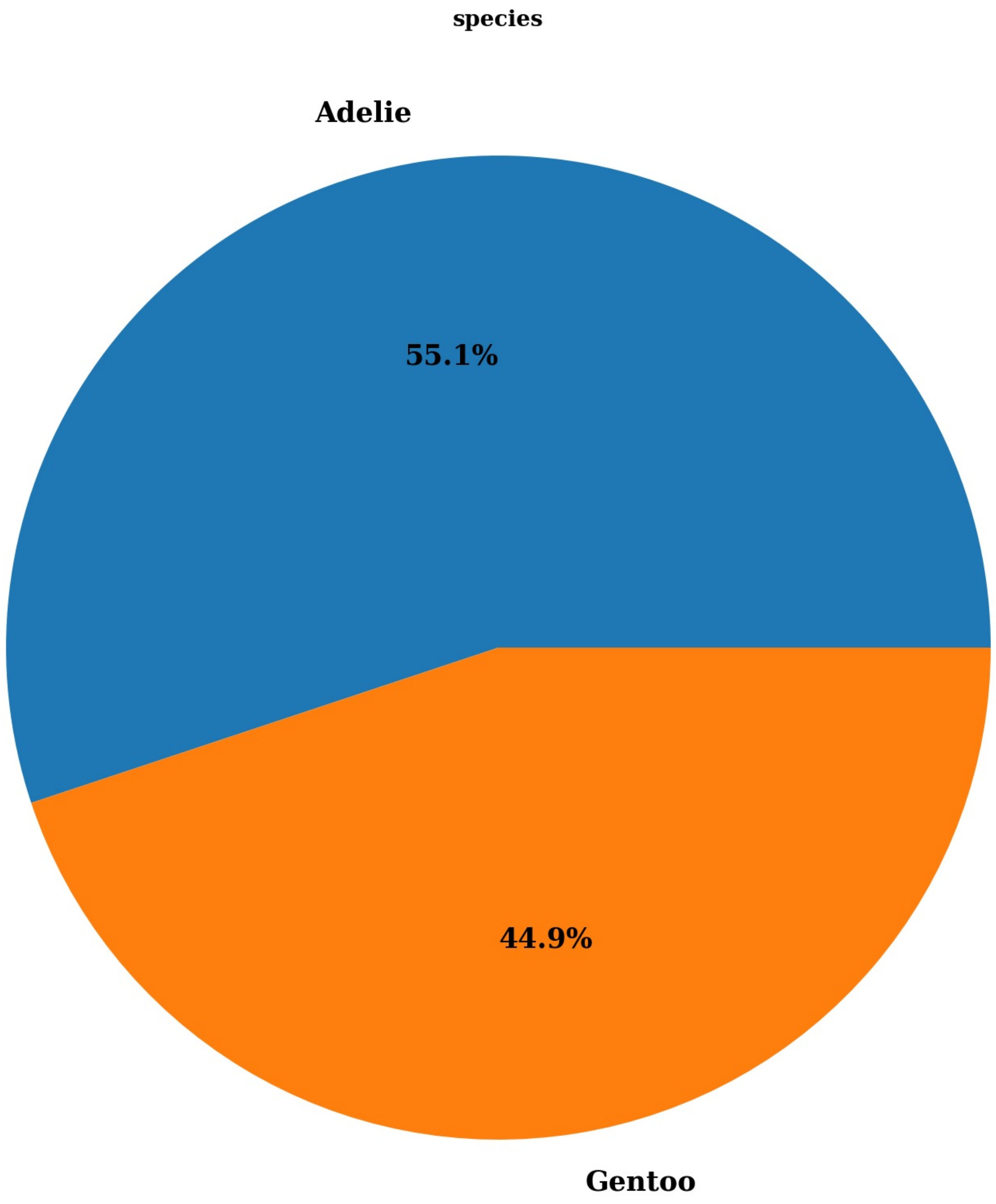
In [14]:

```
plt.figure(figsize=(15,6))
sns.countplot(x =df['species'], data = df, palette = 'hls')
plt.xticks(rotation = 90)
plt.show()
```



In [15]:

```
plt.figure(figsize=(30,20))
plt.pie(df['species'].value_counts(), labels=df['species'].value_counts().index, autopct='%1.1f%%', textprops={
                                            'color': 'black',
                                            'weight': 'bold',
                                            'family': 'serif' })
hfont = {'fontname':'serif', 'weight': 'bold'}
plt.title('species', size=20, **hfont)
plt.show()
```



In [16]:

```
df['island'].nunique()
```

Out[16]:

```
3
```

In [17]:

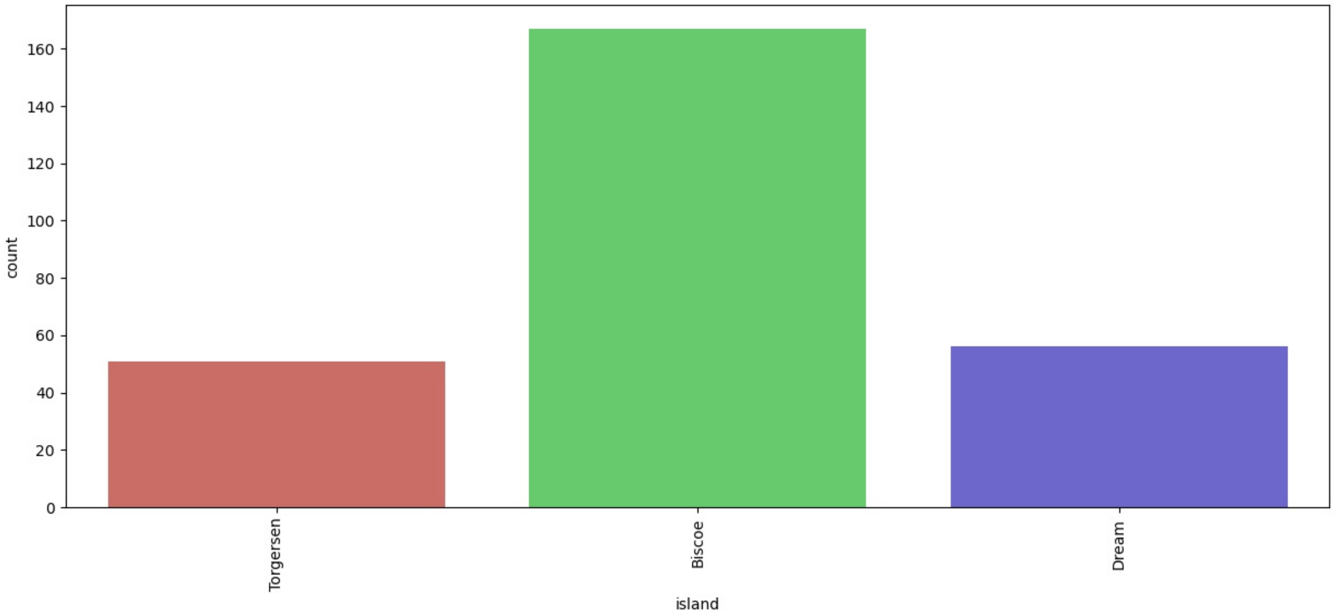
```
df['island'].value_counts()
```

Out[17]:

```
island
Biscoe       167
Dream         56
Torgersen     51
Name: count, dtype: int64
```

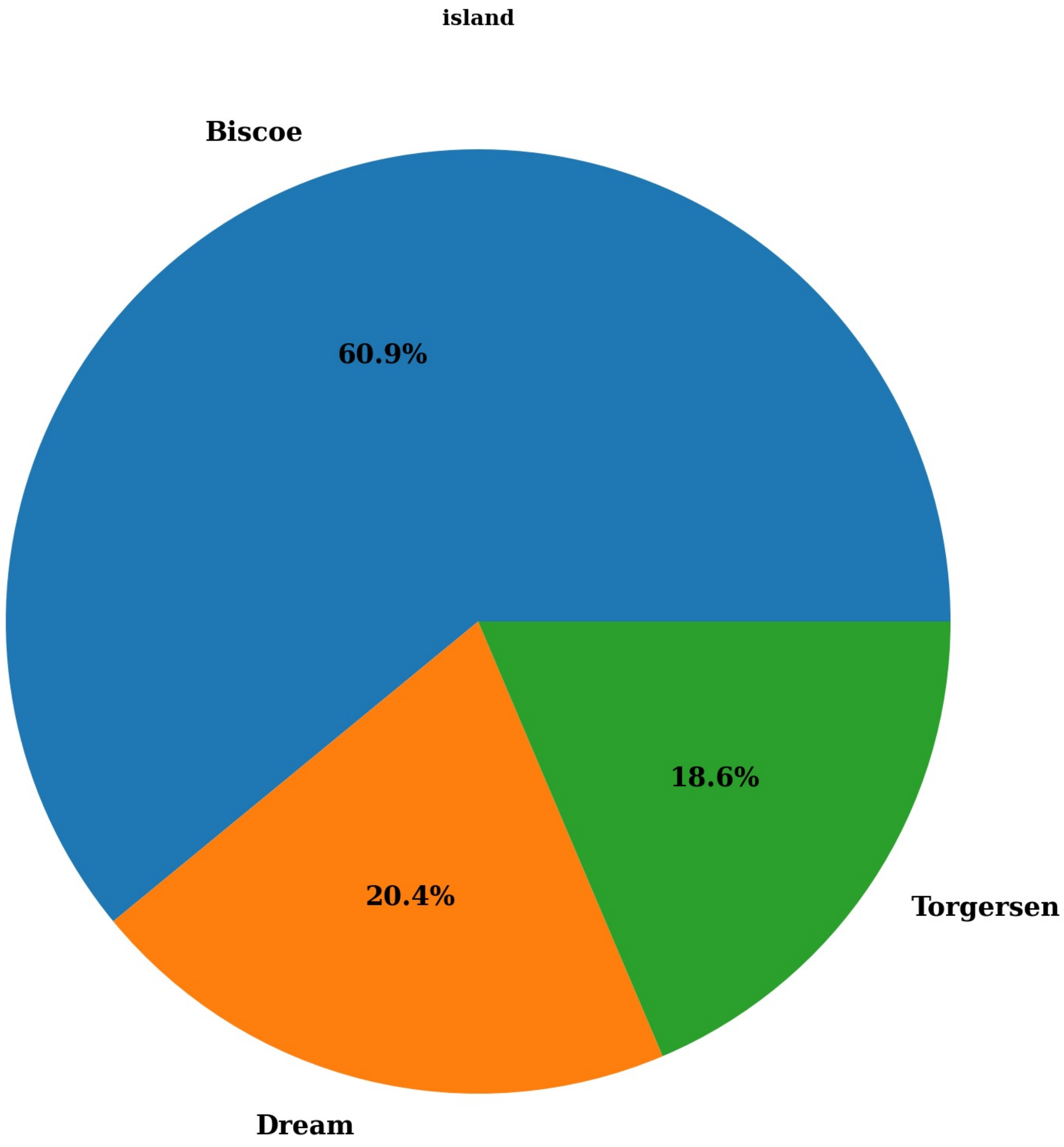
In [18]:

```
plt.figure(figsize=(15,6))
sns.countplot(x =df['island'], data = df, palette = 'hls')
plt.xticks(rotation = 90)
plt.show()
```



In [19]:

```
plt.figure(figsize=(30,20))
plt.pie(df['island'].value_counts(), labels=df['island'].value_counts().index, autopct='%1.1f%%', textprops={ '
                                           'color': 'black',
                                           'weight': 'bold',
                                           'family': 'serif' })
hfont = {'fontname':'serif', 'weight': 'bold'}
plt.title('island', size=20, **hfont)
plt.show()
```



In [22]:

```
df.rename(columns={'bill_length_mm': 'bill length', 'bill_depth_mm': 'bill depth','flipper_length_mm':'flipper
```

In [23]:

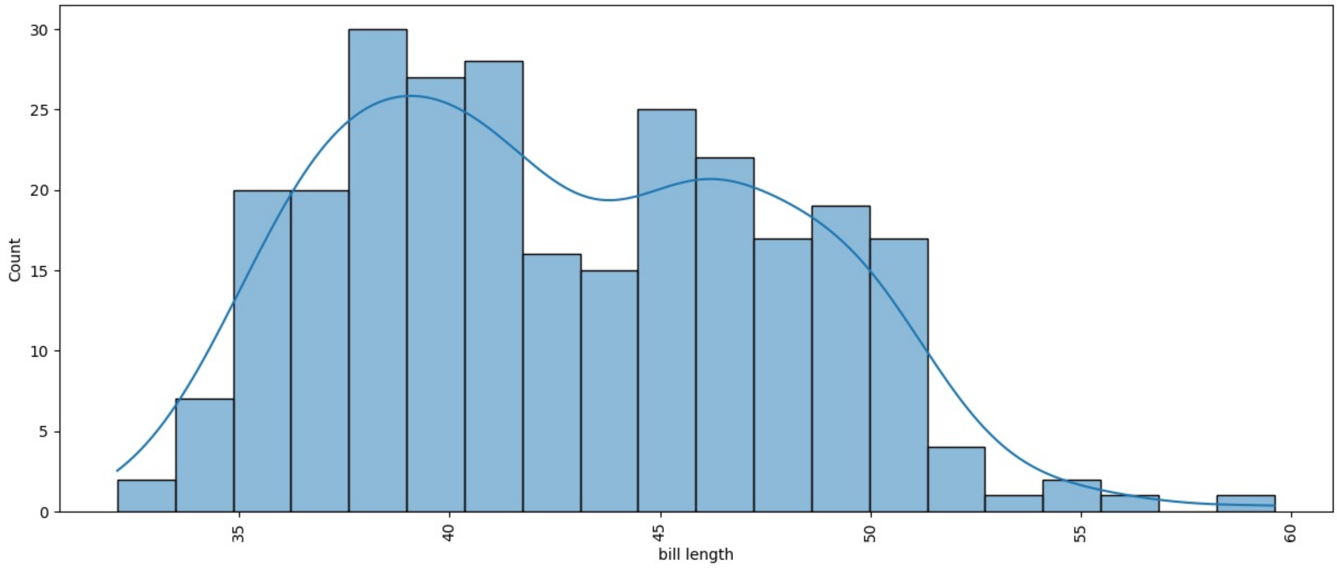
```
df.head()
```

Out[23]:

	species	island	bill length	bill depth	flipper length	body mass	year
0	Adelie	Torgersen	39.1	18.7	181.0	3750.0	2007
1	Adelie	Torgersen	39.5	17.4	186.0	3800.0	2007
2	Adelie	Torgersen	40.3	18.0	195.0	3250.0	2007
3	Adelie	Torgersen	36.7	19.3	193.0	3450.0	2007
4	Adelie	Torgersen	39.3	20.6	190.0	3650.0	2007

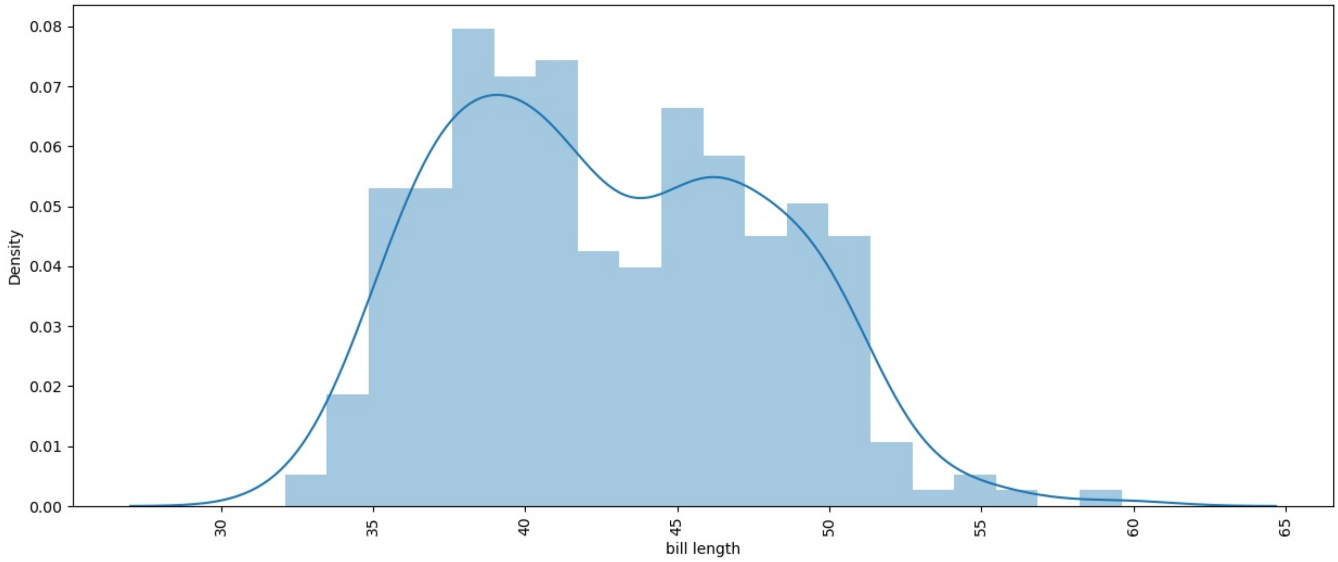
In [24]:

```
plt.figure(figsize=(15,6))
sns.histplot(df['bill length'], kde = True, bins = 20, palette = 'hls')
plt.xticks(rotation = 90)
plt.show()
```



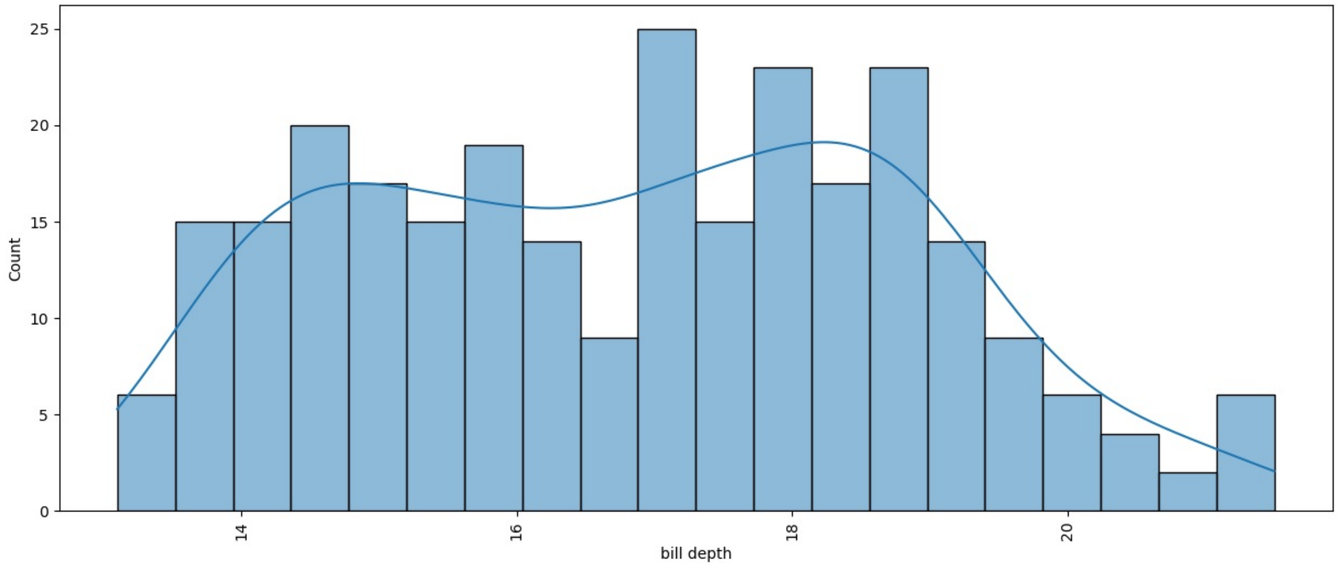
In [27]:

```
plt.figure(figsize=(15,6))
sns.distplot(df['bill length'], kde = True, bins = 20)
plt.xticks(rotation = 90)
plt.show()
```



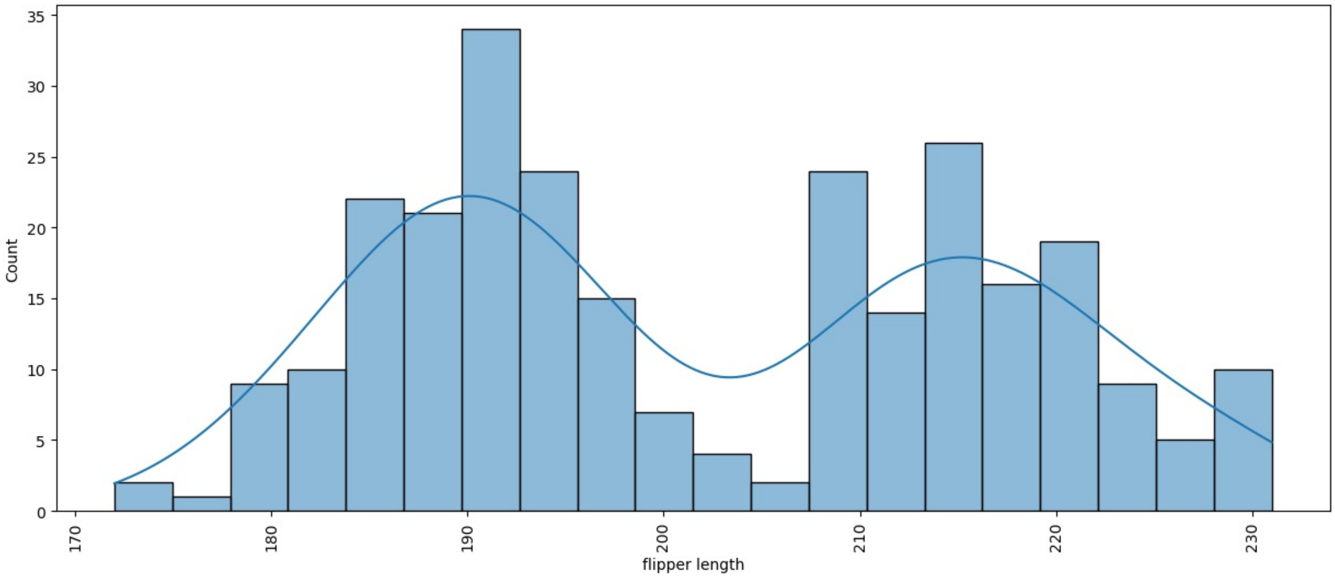
In [28]:

```
plt.figure(figsize=(15,6))
sns.histplot(df['bill depth'], kde = True, bins = 20, palette = 'hls')
plt.xticks(rotation = 90)
plt.show()
```



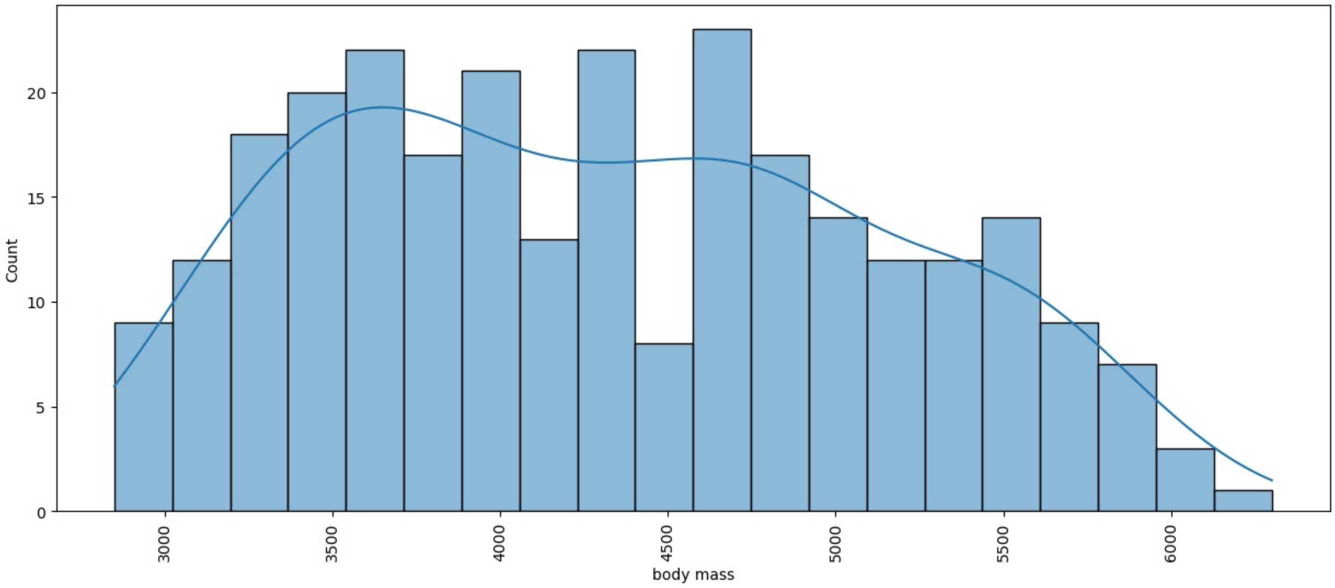
In [29]:

```
plt.figure(figsize=(15,6))
sns.histplot(df['flipper length'], kde = True, bins = 20, palette = 'hls')
plt.xticks(rotation = 90)
plt.show()
```



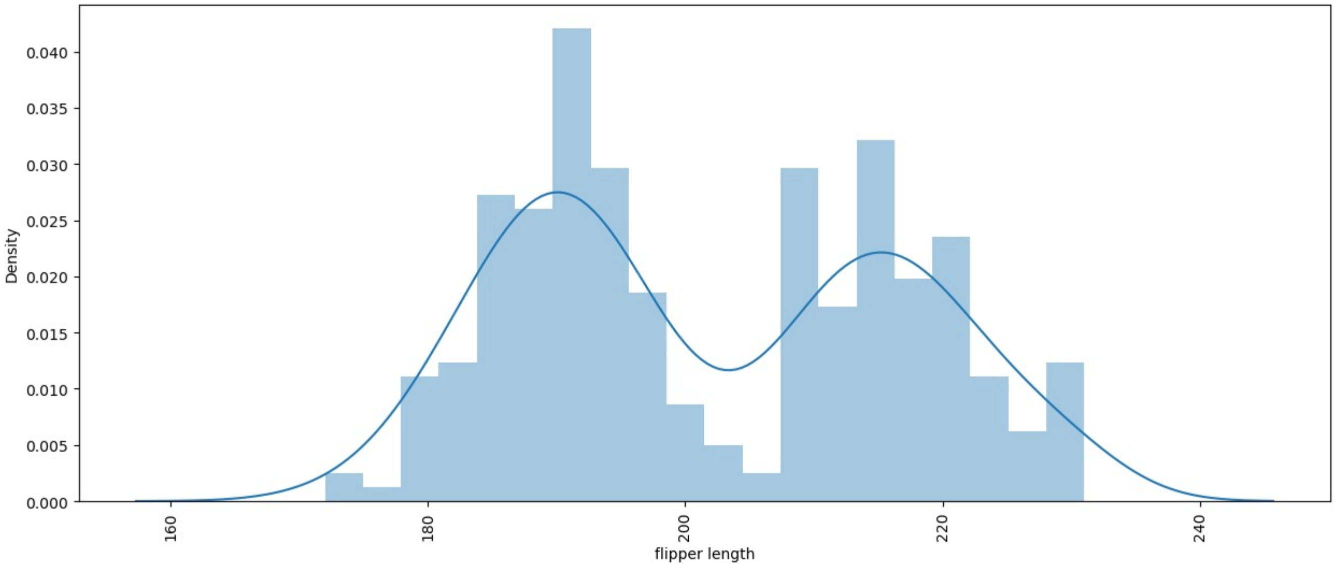
In [30]:

```
plt.figure(figsize=(15,6))
sns.histplot(df['body mass'], kde = True, bins = 20, palette = 'hls')
plt.xticks(rotation = 90)
plt.show()
```



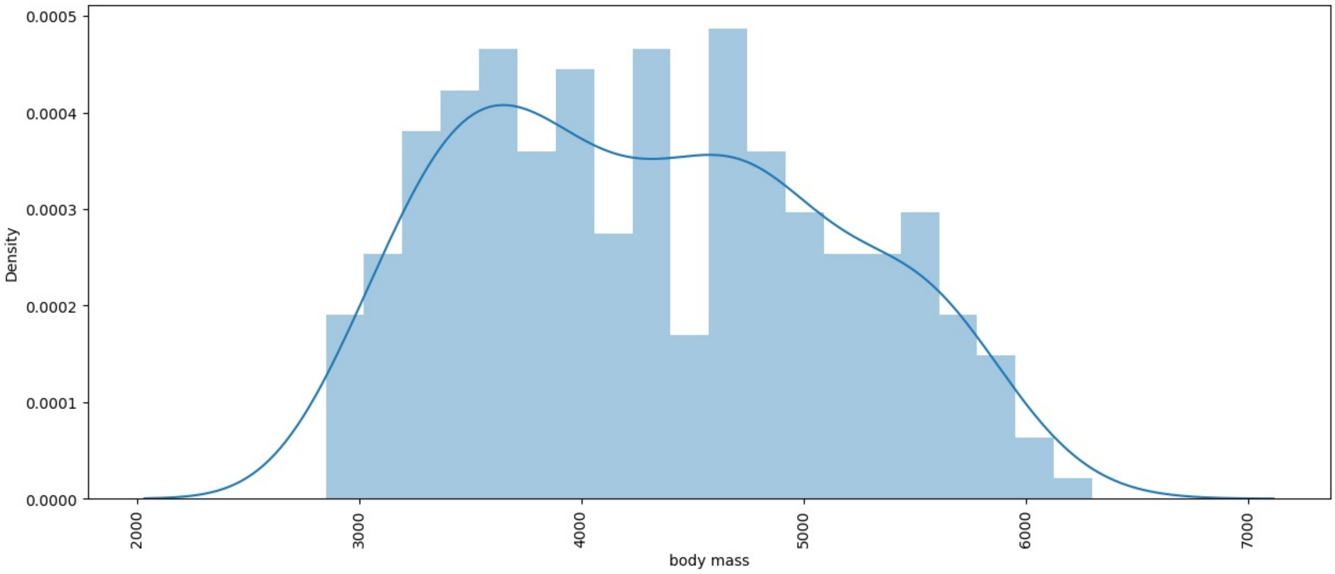
In [31]:

```
plt.figure(figsize=(15,6))
sns.distplot(df['flipper length'], kde = True, bins = 20)
plt.xticks(rotation = 90)
plt.show()
```



In [32]:

```
plt.figure(figsize=(15,6))
sns.distplot(df['body mass'], kde = True, bins = 20)
plt.xticks(rotation = 90)
plt.show()
```

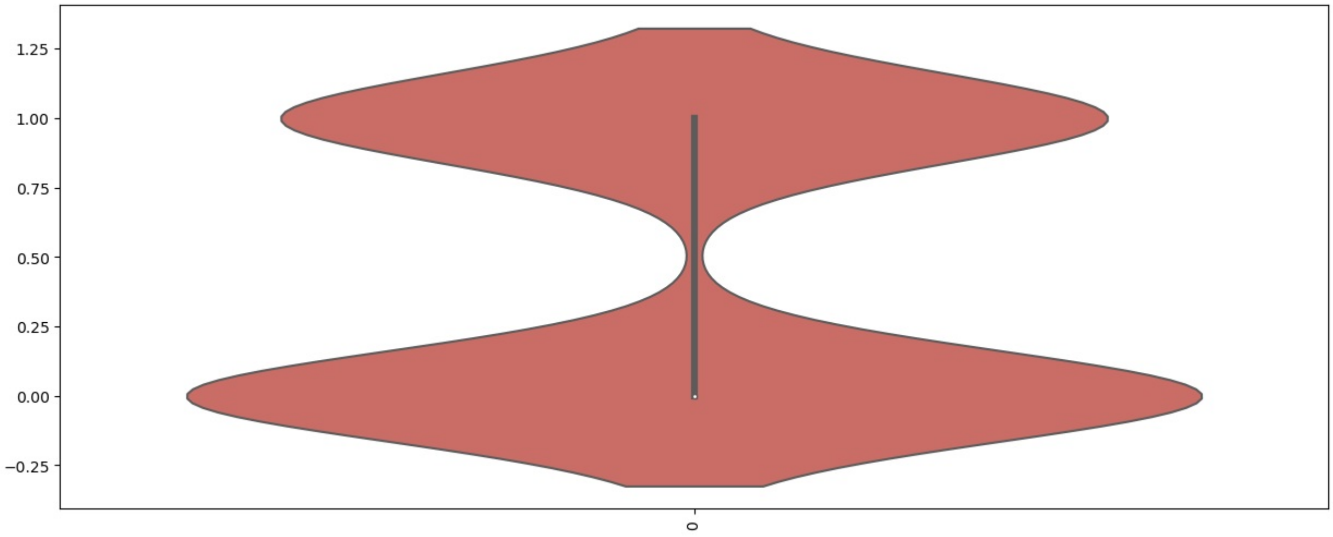


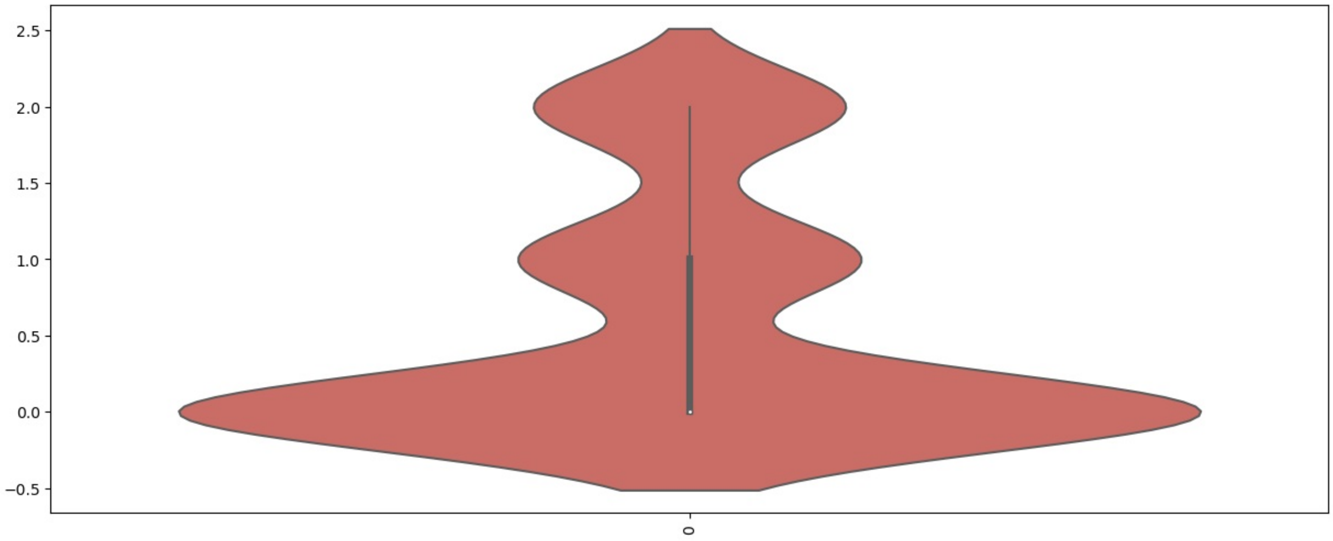
In [35]:

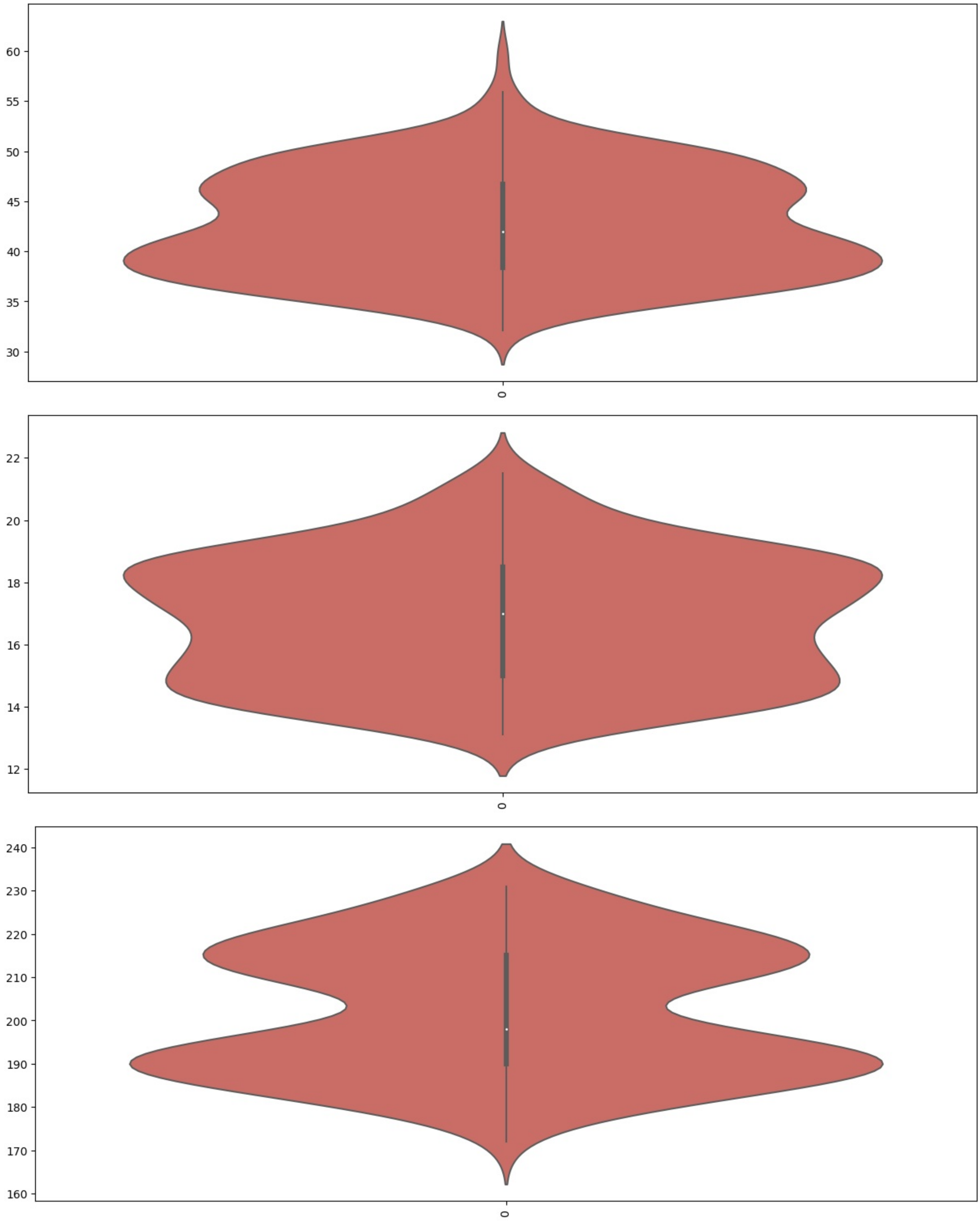
```
from sklearn.preprocessing import LabelEncoder
object_cols = df.select_dtypes(include=['object']).columns

le = LabelEncoder()
for col in object_cols:
    df[col] = le.fit_transform(df[col])
```

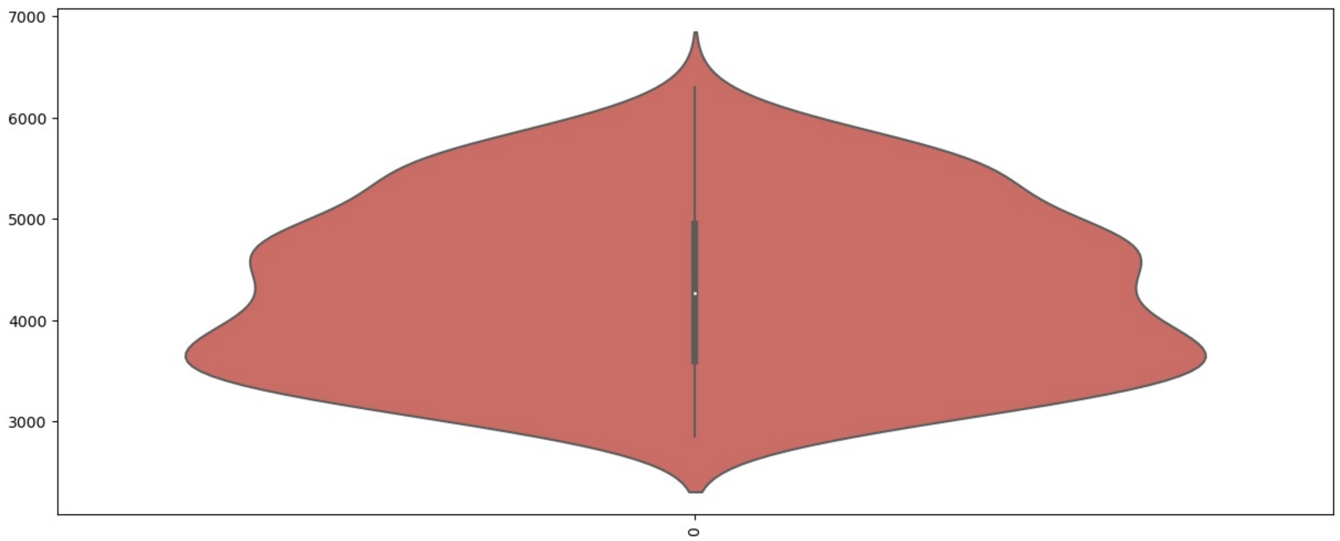
In [36]:

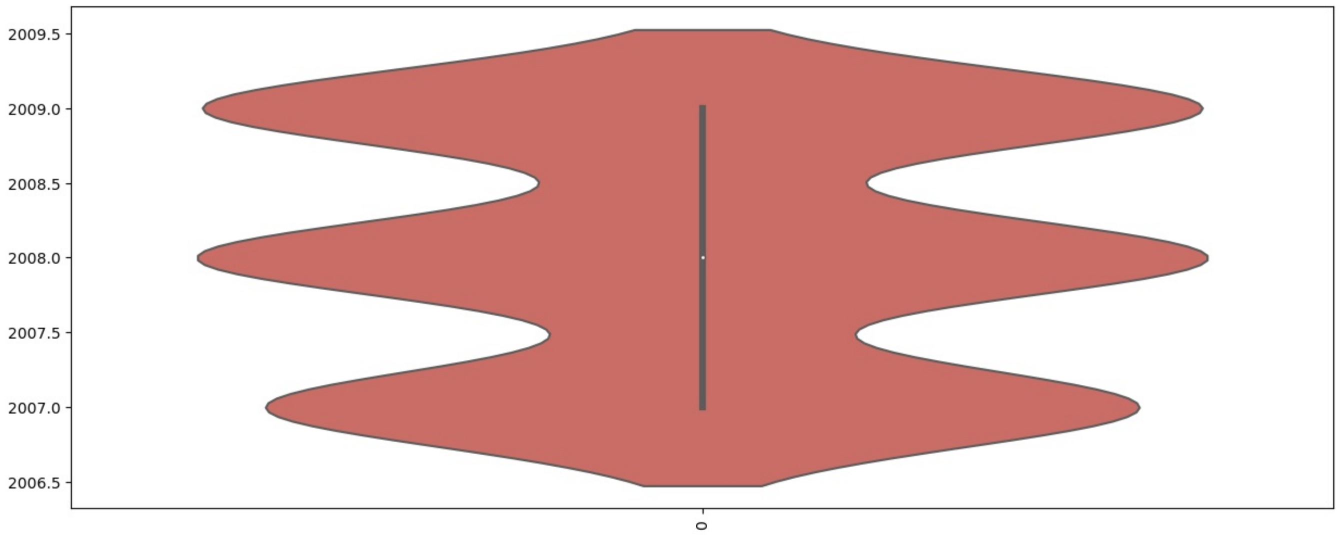
```
for i in df.columns:
    plt.figure(figsize=(15,6))
    sns.violinplot(df[i], palette = 'hls')
    plt.xticks(rotation = 90)
    plt.show()
```



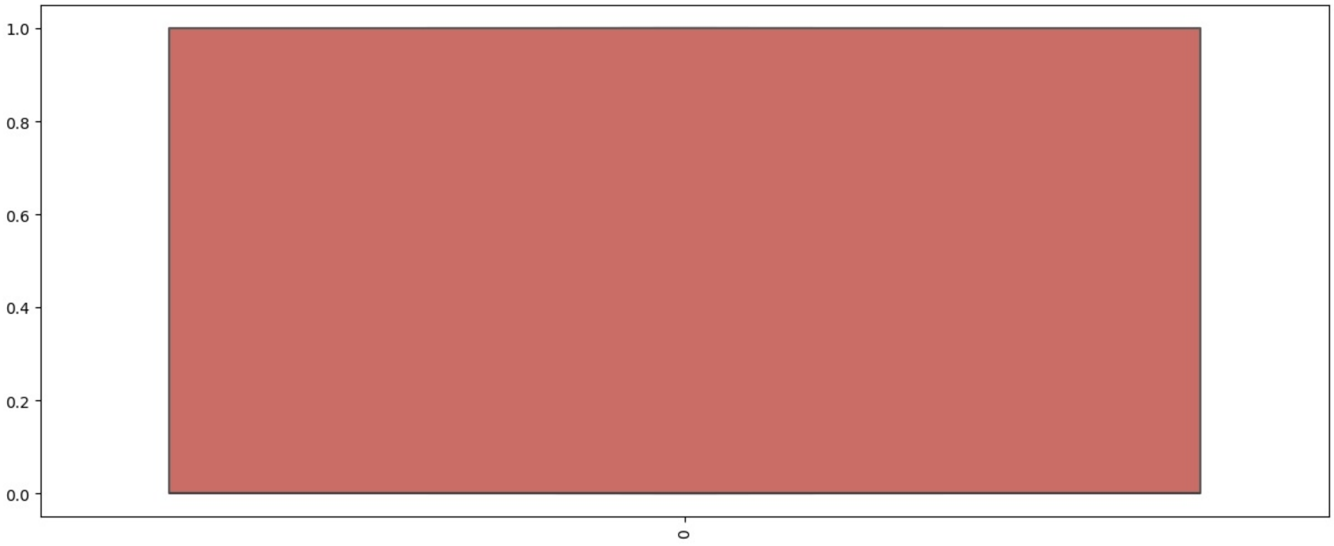


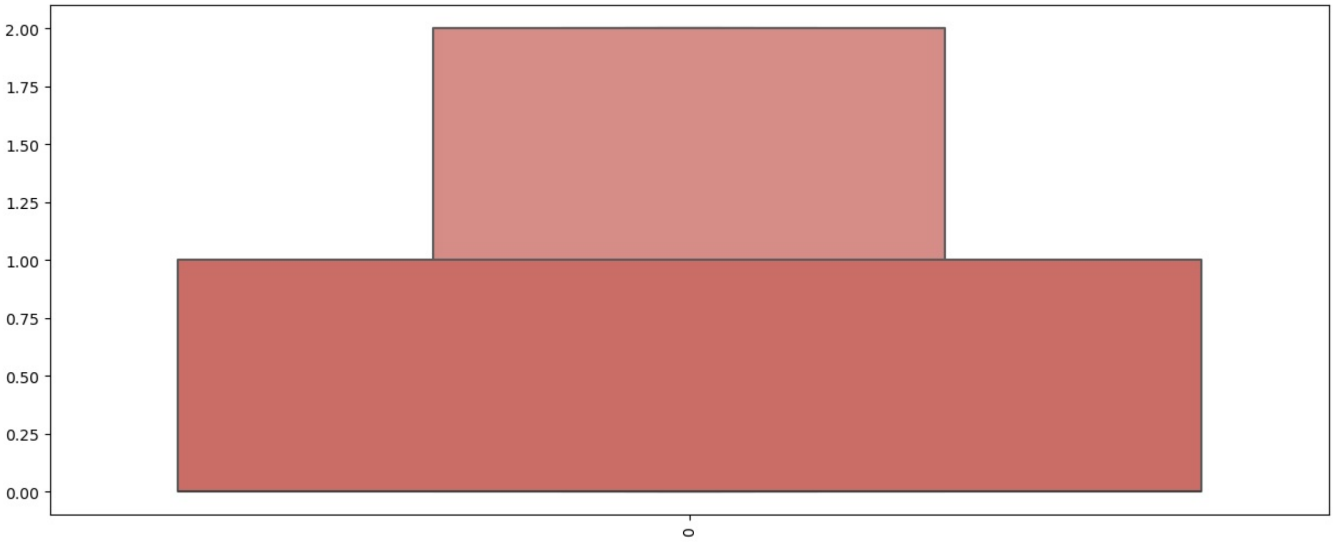

60
55
50
45
40
35
30
0
22
20
18
16
14
12
0
240
230
220
210
200
190
180
170
160
0

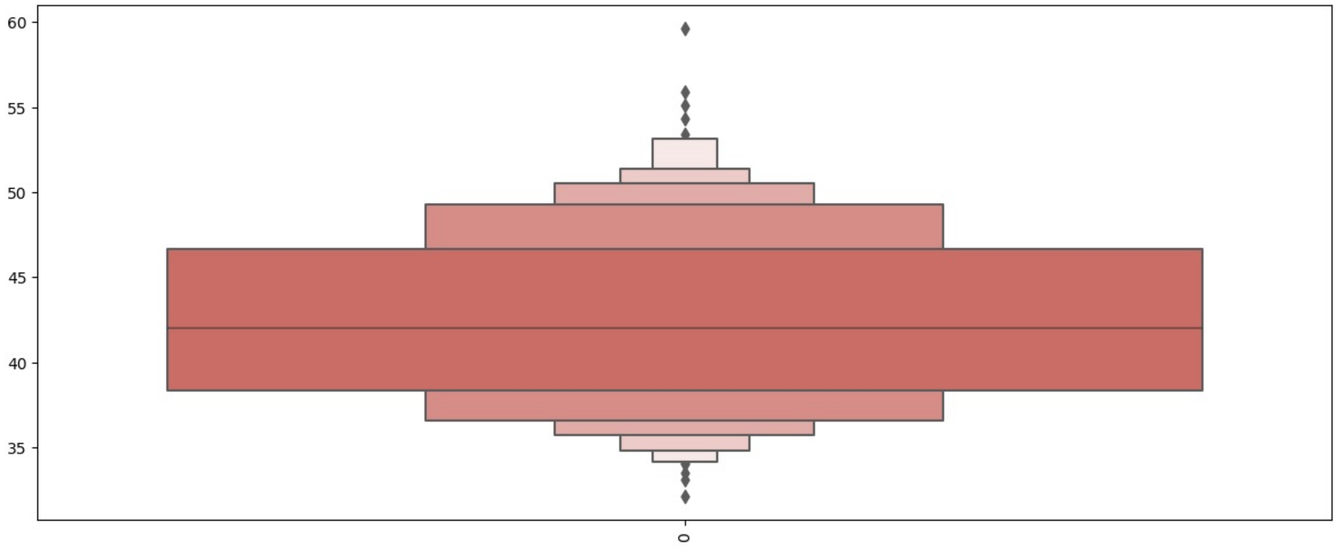


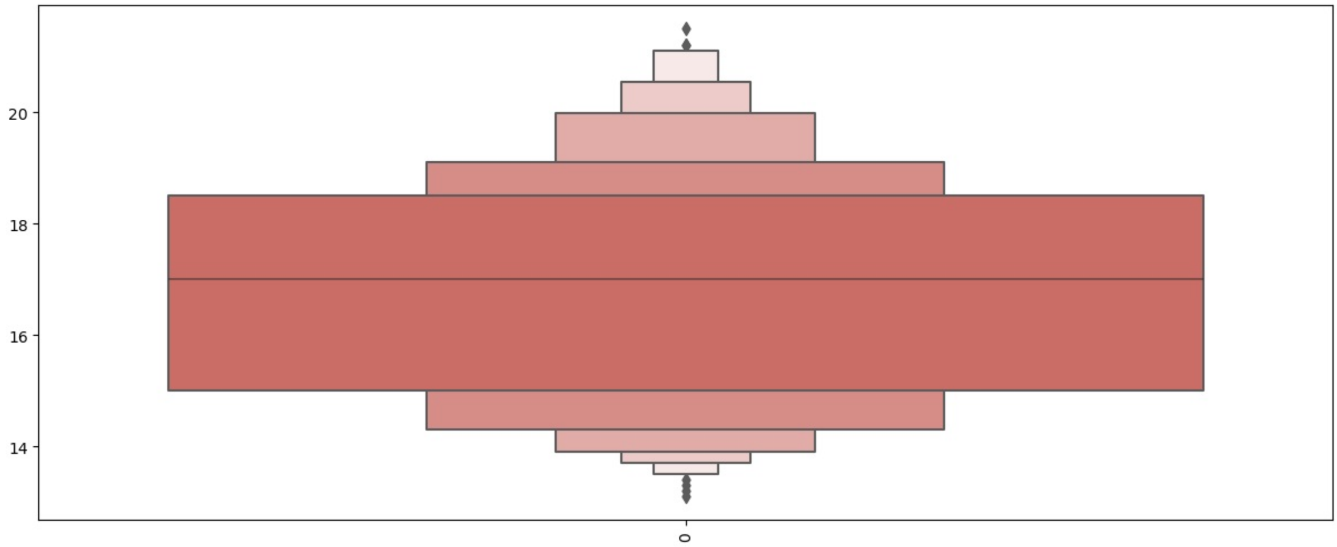


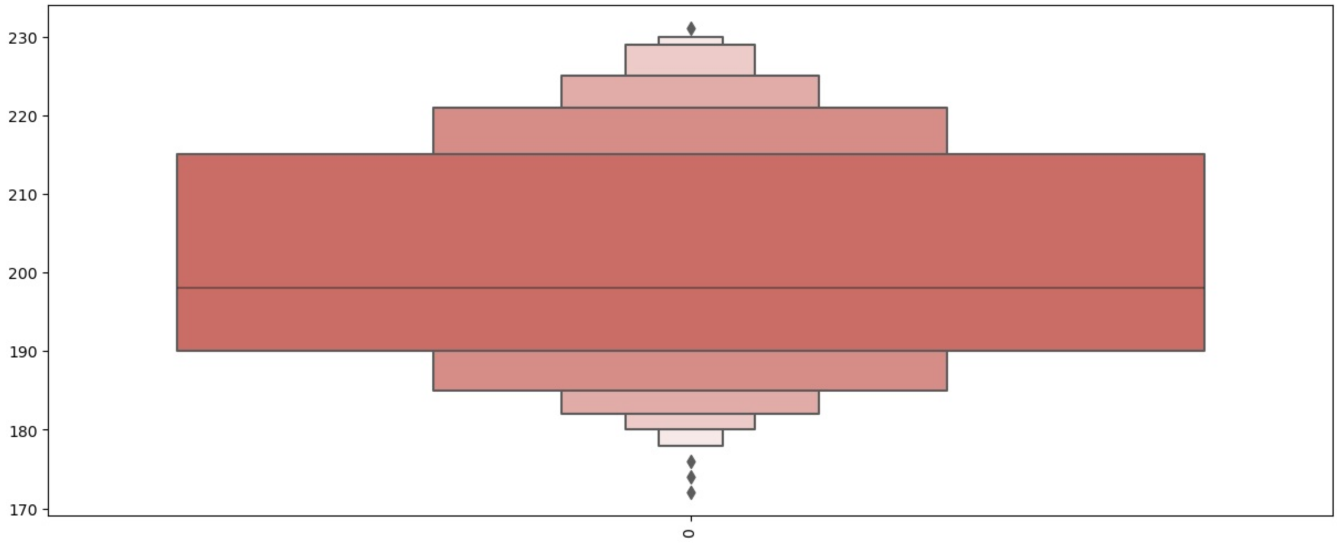
In [37]:
```
for i in df.columns:
    plt.figure(figsize=(15,6))
    sns.boxenplot(df[i], palette = 'hls')
    plt.xticks(rotation = 90)
    plt.show()
```

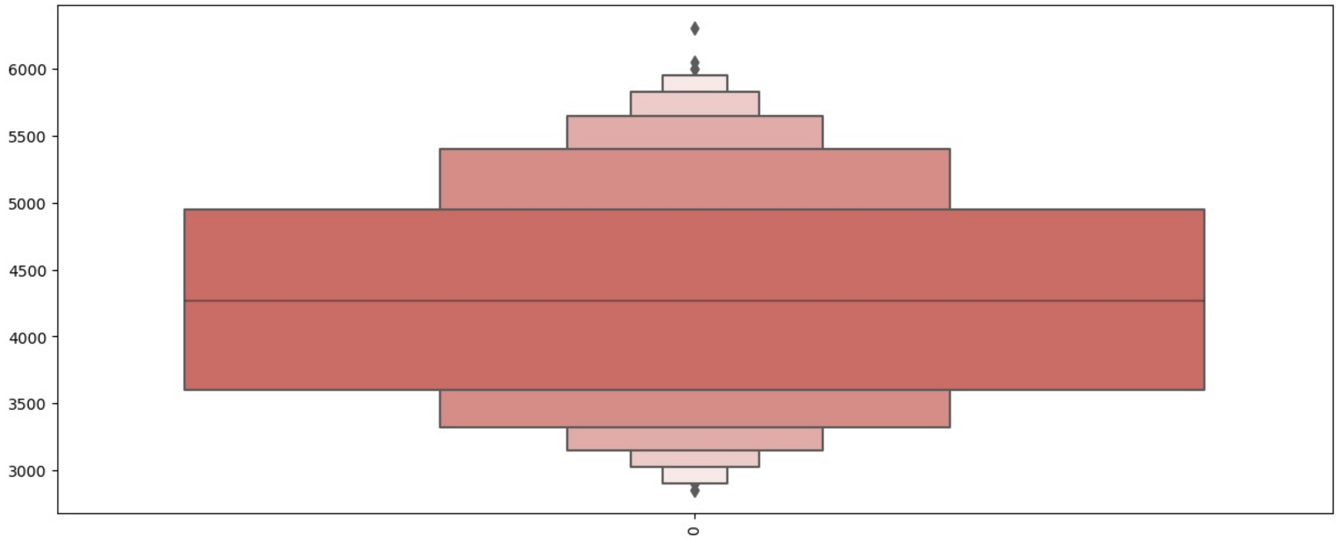

1.0
0.8
0.6
0.4
0.2
0.0
0

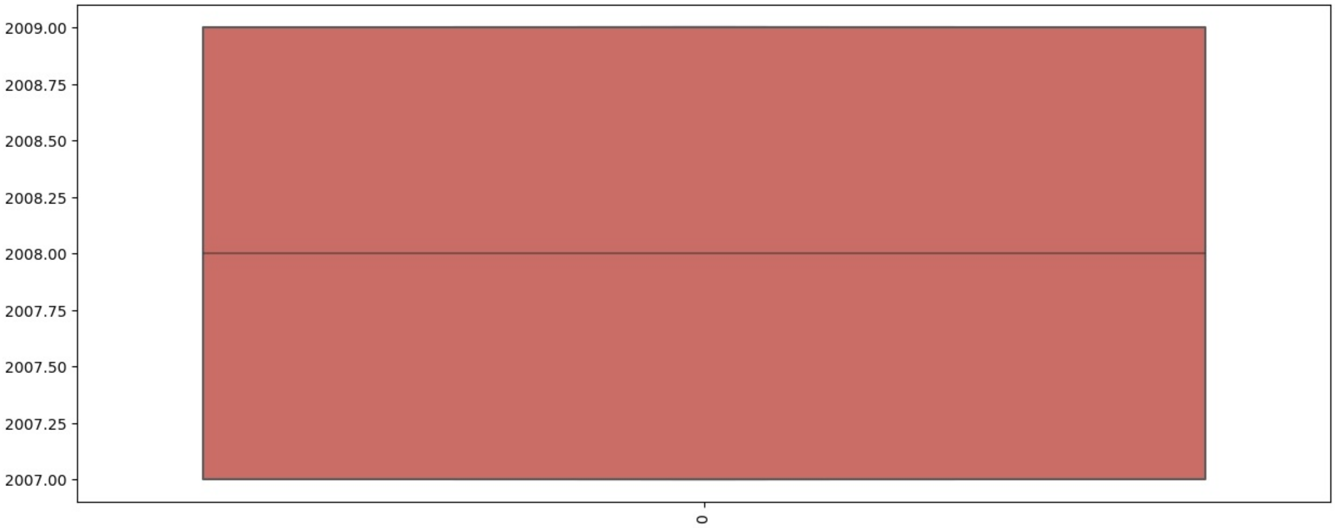

2.00
1.75
1.50
1.25
1.00
0.75
0.50
0.25
0.00
0


60
55
50
45
40
35
0

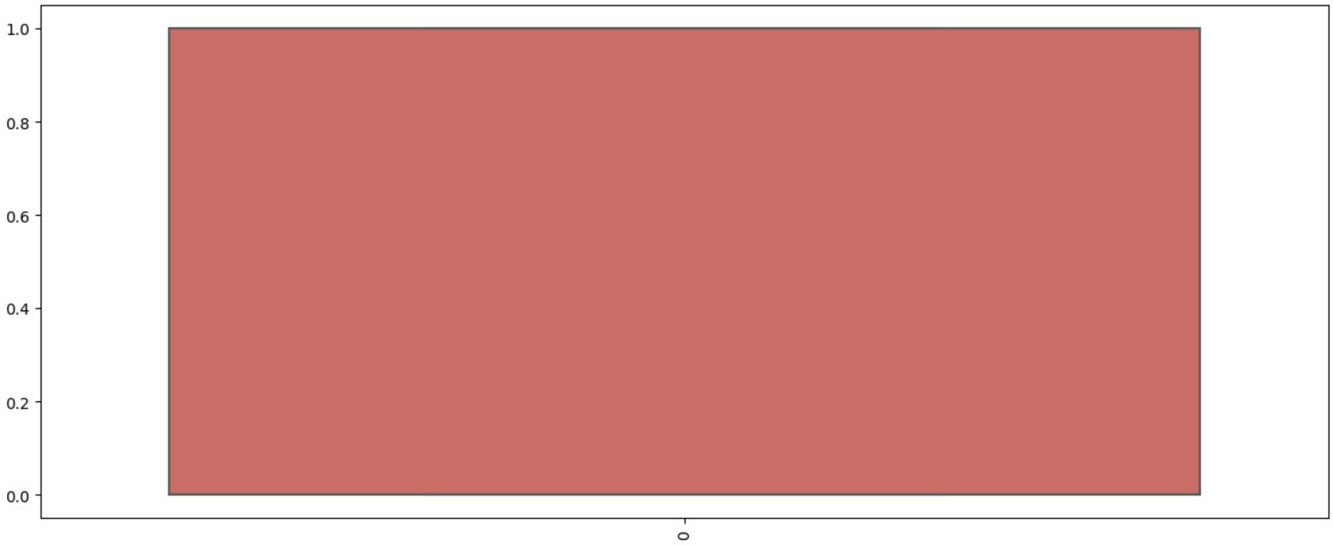

20
18
16
14
0

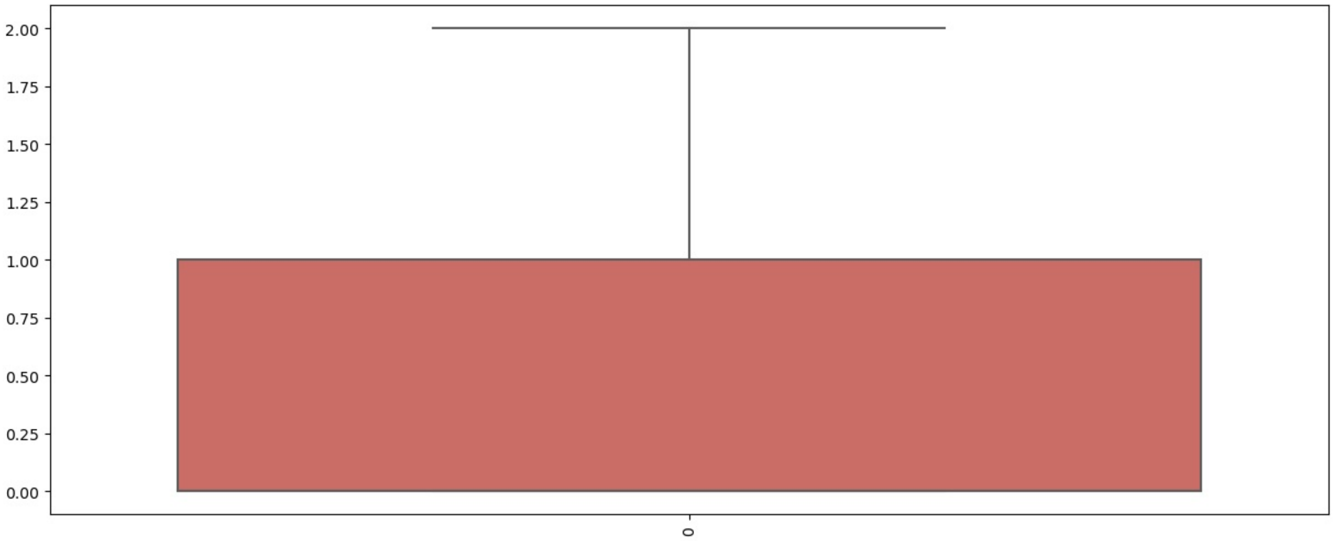

230
220
210
200
190
180
170
0

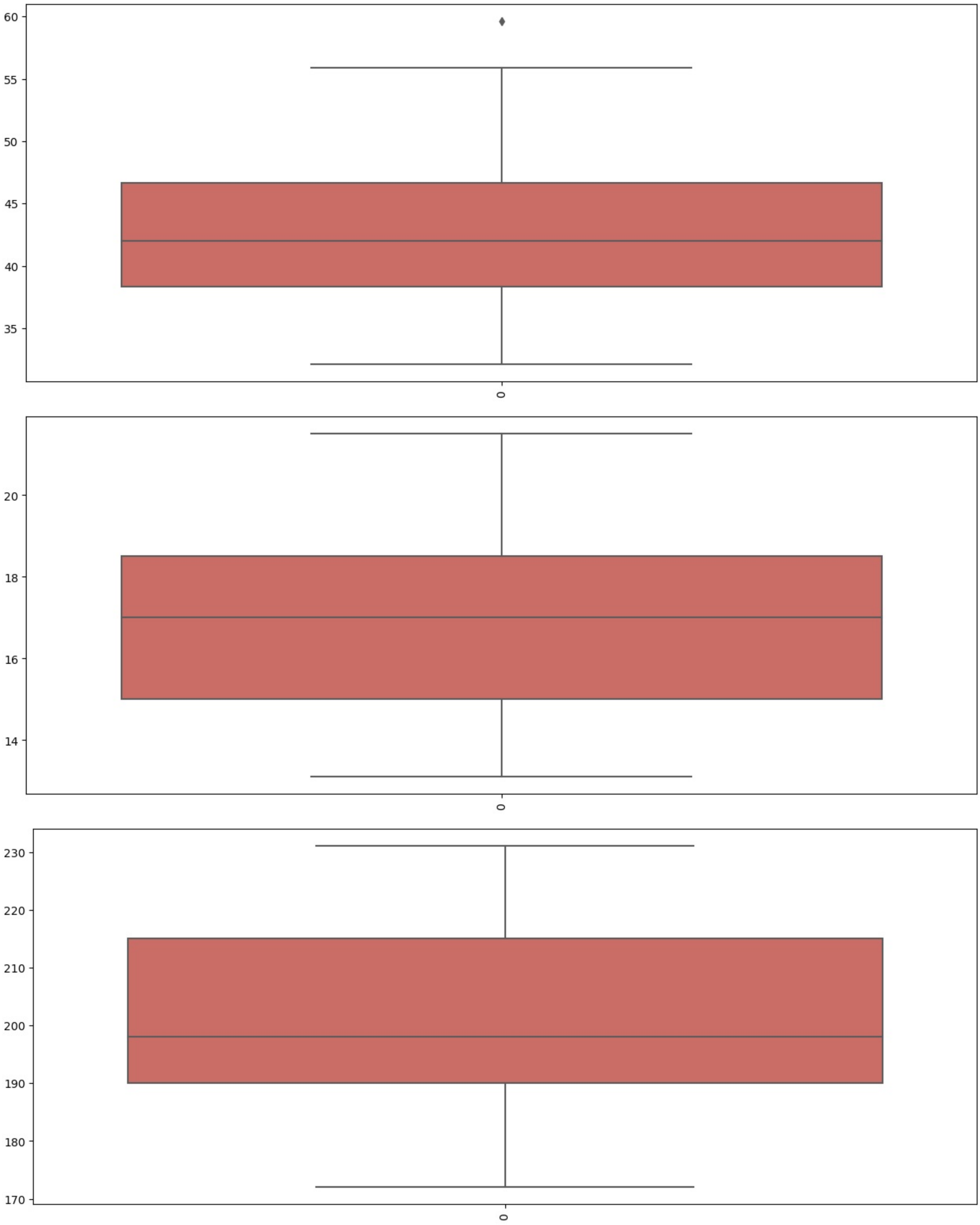

6000
5500
5000
4500
4000
3500
3000
0

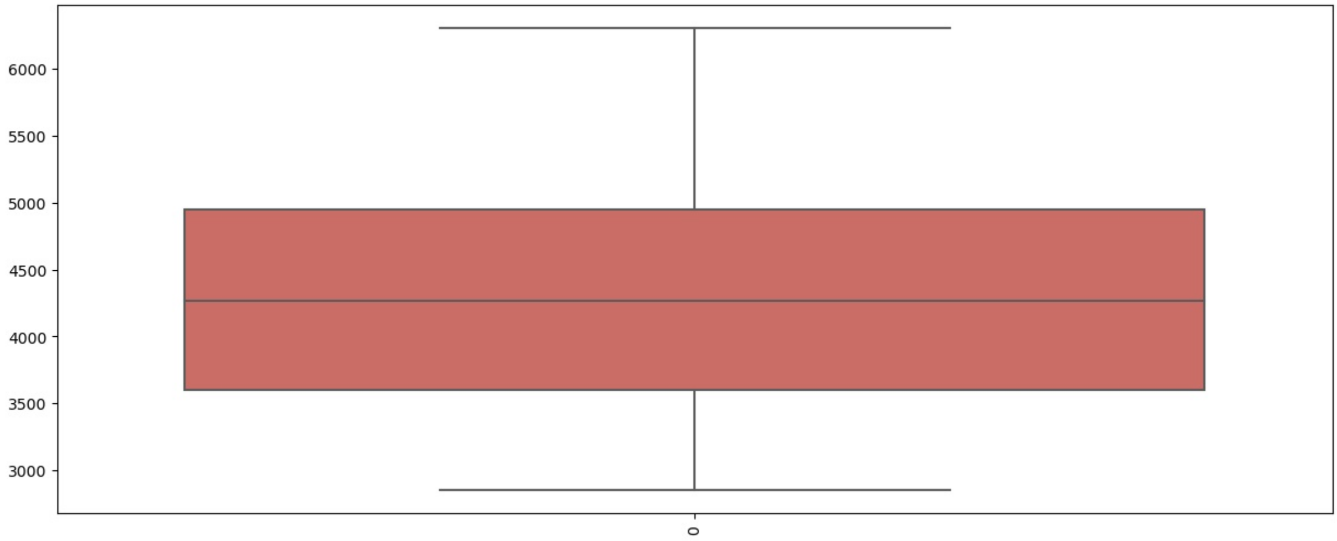


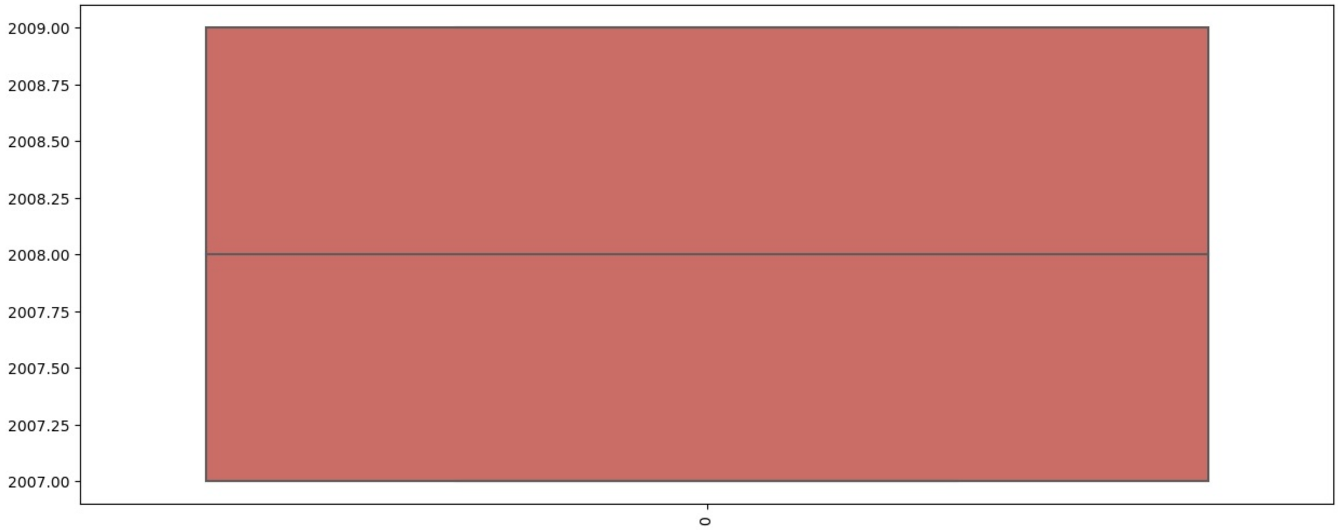
In [38]:
```
for i in df.columns:
    plt.figure(figsize=(15,6))
    sns.boxplot(df[i], palette = 'hls')
    plt.xticks(rotation = 90)
    plt.show()
```



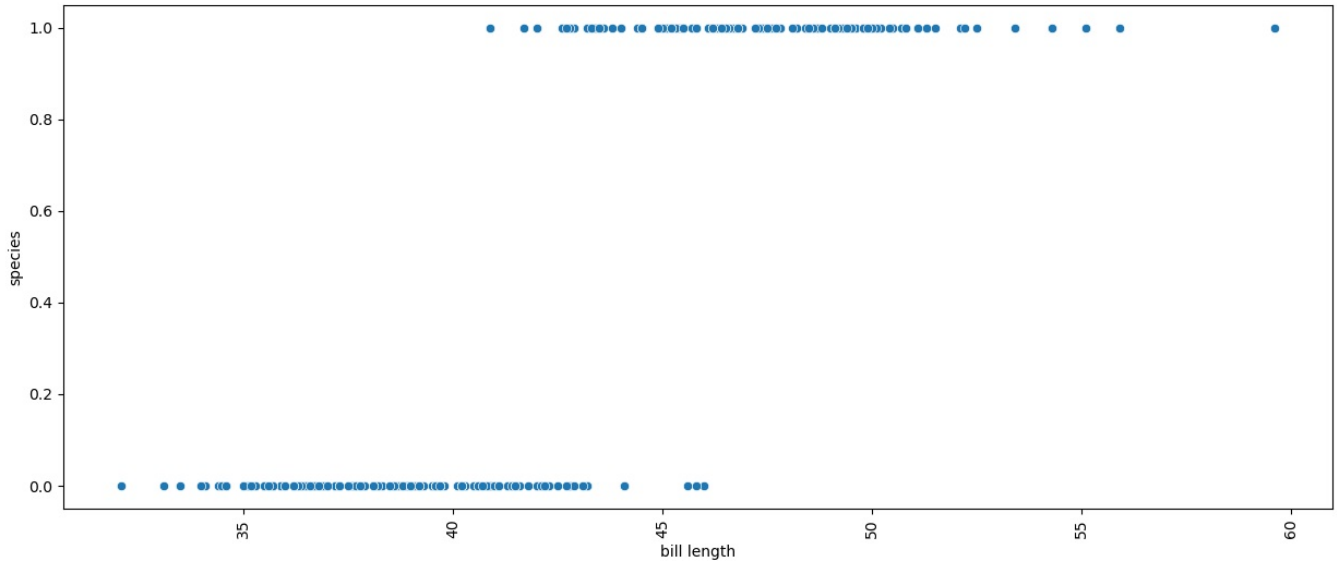



60
55
50
45
40
35
0
20
18
16
14
0
230
220
210
200
190
180
170
0





In [40]:
```
plt.figure(figsize=(15,6))
sns.scatterplot(x = df['bill length'], y = df['species'], palette = 'hls')
plt.xticks(rotation = 90)
plt.show()
```



In [41]:
```
Q1 = df.quantile(0.25)
Q3 = df.quantile(0.75)
IQR = Q3 - Q1
print(IQR)
```

```
species              1.000
island               1.000
bill length          8.325
bill depth           3.500
flipper length      25.000
body mass         1350.000
year                 2.000
dtype: float64
```

In [42]:

```
df_new = df[~((df < (Q1 - 1.5 * IQR)) | (df > (Q3 + 1.5 * IQR))).any(axis = 1)]
```

In [43]:

```
df_new.shape
```

Out[43]:

```
(273, 7)
```

In [44]:

```
df_corr = df.corr()
```

In [45]:

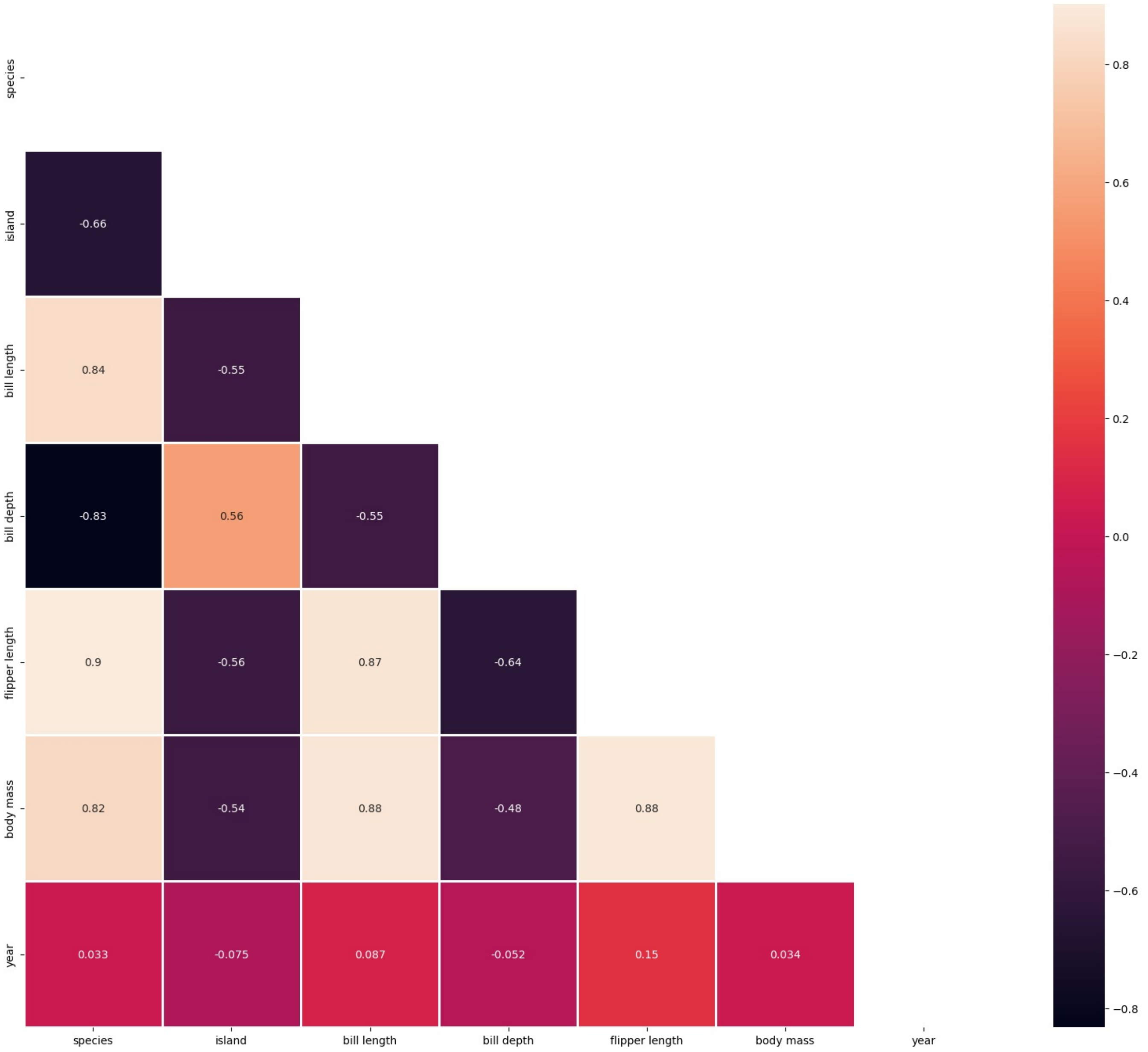
```
df_corr
```

Out[45]:

	species	island	bill length	bill depth	flipper length	body mass	year
species	1.000000	-0.662893	0.835687	-0.832606	0.901796	0.819837	0.032942
island	-0.662893	1.000000	-0.553917	0.559274	-0.562413	-0.544047	-0.074625
bill length	0.835687	-0.553917	1.000000	-0.546050	0.869026	0.876905	0.086799
bill depth	-0.832606	0.559274	-0.546050	1.000000	-0.640141	-0.483223	-0.051973
flipper length	0.901796	-0.562413	0.869026	-0.640141	1.000000	0.882262	0.150004
body mass	0.819837	-0.544047	0.876905	-0.483223	0.882262	1.000000	0.034148
year	0.032942	-0.074625	0.086799	-0.051973	0.150004	0.034148	1.000000

In [46]:

```
plt.figure(figsize=(20, 17))
matrix = np.triu(df_corr)
sns.heatmap(df_corr, annot=True, linewidth=.8, mask=matrix, cmap="rocket");
plt.show()
```



```
In [48]: from sklearn.model_selection import train_test_split
         X_train,X_test,y_train,y_test = train_test_split(X,y)
```

```
In [63]: X_train.shape
```

```
Out[63]: (205, 6)
```

```
In [64]: X_test.shape
```

```
Out[64]: (69, 6)
```

```
In [47]: X = df.drop('species', axis=1)
         y = df['species']
```

```
In [50]: from sklearn.linear_model import LogisticRegression
```

```
In [60]: regressor = LogisticRegression()
         regressor.fit(X_train,y_train)
```

```
Out[60]: ▾ LogisticRegression
         LogisticRegression()
```

```
In [51]: model = LogisticRegression()
```

```
In [53]: model.fit(X_train,y_train)
```

```
Out[53]: ▾ LogisticRegression
         LogisticRegression()
```

```
In [54]: y_predict = model.predict(X_test)
```

```
y_predict
```

Out[54]:

```
array([0, 0, 0, 1, 0, 0, 0, 0, 0, 1, 1, 1, 0, 0, 1, 0, 0, 1, 1, 0, 1, 0,
       0, 0, 0, 1, 1, 1, 0, 0, 1, 1, 0, 1, 0, 1, 0, 0, 1, 1, 0, 1, 0, 0,
       0, 1, 0, 1, 1, 1, 1, 1, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0,
       0, 1, 0])
```

In [55]:

```
from sklearn.metrics import accuracy_score, classification_report, confusion_matrix
```

In [56]:

```
accuracy_score(y_test,y_predict)
```

Out[56]:

```
1.0
```

In [57]:

```
print(classification_report(y_test,y_predict))
```

```
              precision    recall  f1-score   support

           0       1.00      1.00      1.00        42
           1       1.00      1.00      1.00        27

    accuracy                           1.00        69
   macro avg       1.00      1.00      1.00        69
weighted avg       1.00      1.00      1.00        69
```

In [65]:

```
df.head()
```

Out[65]:

	species	island	bill length	bill depth	flipper length	body mass	year
0	0	2	39.1	18.7	181.0	3750.0	2007
1	0	2	39.5	17.4	186.0	3800.0	2007
2	0	2	40.3	18.0	195.0	3250.0	2007
3	0	2	36.7	19.3	193.0	3450.0	2007
4	0	2	39.3	20.6	190.0	3650.0	2007

In [73]:

```
df.tail()
```

Out[73]:

	species	island	bill length	bill depth	flipper length	body mass	year
269	1	0	47.2	13.7	214.0	4925.0	2009
270	1	0	46.8	14.3	215.0	4850.0	2009
271	1	0	50.4	15.7	222.0	5750.0	2009
272	1	0	45.2	14.8	212.0	5200.0	2009
273	1	0	49.9	16.1	213.0	5400.0	2009

In [66]:

```
input_data = (2,39.1,18.7,181.0,3750.0,2007)

input_data_as_numpy_array = np.asarray(input_data)

input_data_reshaped = input_data_as_numpy_array.reshape(1,-1)

prediction = regressor.predict(input_data_reshaped)
print(prediction)
```

```
[0]
```

In [67]:

```
print('The penguins species',prediction[0])
```

```
The penguins species 0
```

In [68]:

```
if (prediction[0])==0:
    print('The penguin species is Adelie')
else:
    print('The penguin species is Gentoo')
```

```
The penguin species is Adelie
```

In [69]:

```
import pickle
```

In [70]:

```
filename = 'trained_model.sav'
pickle.dump(regressor,open(filename,'wb'))
```

In [71]:

```
loaded_model = pickle.load(open('trained_model.sav','rb'))
```

In [74]:

```
input_data = (1,39.1,18.7,181.0,3750.0,2007)

input_data_as_numpy_array = np.asarray(input_data)

input_data_reshaped = input_data_as_numpy_array.reshape(1,-1)

prediction = regressor.predict(input_data_reshaped)
```

```
print(prediction)


if (prediction[0])==0:
    print('The penguin species is Adelie')
else:
    print('The penguin species is Gentoo')
```

```
[0]
The penguin species is Adelie
```

In [1]:

```
import ultralytics
```

In [2]:

```
ultralytics.__version__
```

Out[2]:

```
'8.0.232'
```

In [3]:

```
import supervision
print("supervision.__version__:", supervision.__version__)
```

```
supervision.__version__: 0.17.1
```

In [ ]:

Loading [MathJax]/jax/output/CommonHTML/fonts/TeX/fontdata.js