Palmer Archipelago Penguin Data Project

By: Milagros Cortez

2023-08-09

In this project, I will use Python to clean and analyze data from penguins in the Palmer Archipelago (Antarctica) which was collected and made available by Dr. Kristen Gorman [1] and the Palmer Station, Antarctica LTER, a member of the Long Term Ecological Research Network. The data is in two .csv files under the names penguins_size.csv and penguins_lter.csv which can be found in Kaggle

Data

penguins_size.csv:

A simplified data set from the original penguin data sets. Contains the variables:

- species: penguin species (Chinstrap, Adélie, or Gentoo)
- culmen_length_mm: culmen length (mm)
- culmen_depth_mm: culmen depth (mm)
- flipper_length_mm: flipper length (mm)
- body_mass_g: body mass (g)
- island: island name (Dream, Torgersen, or Biscoe) in the Palmer Archipelago (Antarctica)
- sex: penguin sex

penguins_lter.csv:

Original combined data for 3 penguins

We first take a look at the data and download the packages we will use for this project below:

In [199…

```
# importing packages
import numpy as np
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
import datetime
import scipy.stats as stats
import statsmodels.formula.api as smf
import statsmodels.api as sm
```

```
from pygam import LogisticGAM, s, f
import random
```

In [9]:
```
# reading the .csv files
penguin_size = pd.read_csv(r"C:\Users\Mili\Python Projects\Penguins\penguins_size.c
penguin_lter = pd.read_csv(r"C:\Users\Mili\Python Projects\Penguins\penguins_lter.c
penguin_size.head()
```

Out[9]:

	species	island	culmen_length_mm	culmen_depth_mm	flipper_length_mm	body_mas
0	Adelie	Torgersen	39.1	18.7	181.0	37
1	Adelie	Torgersen	39.5	17.4	186.0	38(
2	Adelie	Torgersen	40.3	18.0	195.0	32
3	Adelie	Torgersen	NaN	NaN	NaN	N
4	Adelie	Torgersen	36.7	19.3	193.0	34

In [10]:
```
penguin_lter.head()
```

Out[10]:

	studyName	Sample Number	Species	Region	Island	Stage	Individual ID	Clutch Completion	
0	PAL0708	1	Adelie Penguin (Pygoscelis adeliae)	Anvers	Torgersen	Adult, 1 Egg Stage	N1A1	Yes	11/
1	PAL0708	2	Adelie Penguin (Pygoscelis adeliae)	Anvers	Torgersen	Adult, 1 Egg Stage	N1A2	Yes	11/
2	PAL0708	3	Adelie Penguin (Pygoscelis adeliae)	Anvers	Torgersen	Adult, 1 Egg Stage	N2A1	Yes	11/
3	PAL0708	4	Adelie Penguin (Pygoscelis adeliae)	Anvers	Torgersen	Adult, 1 Egg Stage	N2A2	Yes	11/
4	PAL0708	5	Adelie Penguin (Pygoscelis adeliae)	Anvers	Torgersen	Adult, 1 Egg Stage	N3A1	Yes	11/

Data Cleaning

We can see that penguin_size has variables from the penguin_lter data set but both data sets need cleaning. For this project, we would like to use all the variables from the penguin_size data set, but we also want to include the Region, Stage, Clutch Completion, and Date Egg from penguin_lter. Moreover, we will create a new data set composed of penguin_size with the extra variables we need from penguin_lter.

In [11]:
```
# Extracting the variables we want from penguin_lter, renaming them to the conventi
# Adding these variables to penguin_size in a combined data frame named penguins
penguins = penguin_size.assign(region = penguin_lter.loc[:, "Region"], stage = peng
                               clutch_completion = penguin_lter.loc[:, "Clutch Comp
                               date_egg = penguin_lter.loc[:, "Date Egg"])
# Since the variable stage is composed of the life stage of the penguin and the egg
# these two and create the variables life_stage and egg_stage
penguins[['life_stage', 'egg_stage']] = penguins.stage.str.split(",", expand=True)
# removing stage variable
penguins.drop('stage', axis=1, inplace=True)
penguins.head()
```

Out[11]:

	species	island	culmen_length_mm	culmen_depth_mm	flipper_length_mm	body_mas
0	Adelie	Torgersen	39.1	18.7	181.0	37
1	Adelie	Torgersen	39.5	17.4	186.0	38(
2	Adelie	Torgersen	40.3	18.0	195.0	32
3	Adelie	Torgersen	NaN	NaN	NaN	N
4	Adelie	Torgersen	36.7	19.3	193.0	34

Now we can see that we have a data set of the variables we want to explore, but there are NaN (null) values in the data. We proceed to remove the rows with null values as we would like to explore penguins that have all these variables recorded for analysis.

In [12]:
```
# Initial number of rows in the data frame
print("Number of rows", penguins.shape[0])
print("Count of non-NaN rows per column:\n", penguins.count())
```

```
Number of rows 344
Count of non-NaN rows per column:
 species              344
island               344
culmen_length_mm     342
culmen_depth_mm      342
flipper_length_mm    342
body_mass_g          342
sex                  334
region               344
clutch_completion    344
date_egg             344
life_stage           344
egg_stage            344
dtype: int64
```

We can see that in most columns there are not many NaN values which is good. However, culmen_length_mm, cumen_depth_mm, flipper_length_mm, and body_mass_g, and sex have NaN values. In particular, sex has 10 NaN values which is the highest of all variables. This could be because beak length and depth is used to determine a penguins sex which develops in later stages of a penguins life as well as length of the penguin. However, this method, specially for king penguins has an accuracy of 79% [2], and there are more accurate but time consuming methods which involve DNA testing [3]. We look at the summary of each variable before and after deleting the rows with NaN values:

In [13]:
```
# Summary for numerical continuous variables before removing NaN values
penguins.describe()
```

Out[13]:

	culmen_length_mm	culmen_depth_mm	flipper_length_mm	body_mass_g
count	342.000000	342.000000	342.000000	342.000000
mean	43.921930	17.151170	200.915205	4201.754386
std	5.459584	1.974793	14.061714	801.954536
min	32.100000	13.100000	172.000000	2700.000000
25%	39.225000	15.600000	190.000000	3550.000000
50%	44.450000	17.300000	197.000000	4050.000000
75%	48.500000	18.700000	213.000000	4750.000000
max	59.600000	21.500000	231.000000	6300.000000

In [79]:
```
# Summary for categorical variables before removing NaN values
cat_var = ['species','island', 'sex', 'region', 'clutch_completion', 'date_egg', 'l
for cat in cat_var:
    print(penguins[cat].value_counts(dropna=False).to_string())
```

```
species
Adelie       152
Gentoo       124
Chinstrap     68
island
Biscoe       168
Dream        124
Torgersen     52
sex
MALE      168
FEMALE    165
NaN        10
.           1
region
Anvers    344
clutch_completion
Yes    308
No      36
date_egg
11/27/07    18
11/9/08     16
11/16/07    16
11/18/09    14
11/4/08     12
11/6/08     12
11/13/08    12
11/21/09    12
11/29/07    10
11/27/09    10
11/15/09    10
11/14/08    10
11/16/09    10
11/22/09    10
11/17/09    10
11/24/08     8
11/28/07     8
11/3/08      8
12/1/09      8
11/9/07      8
11/8/08      8
11/12/07     8
11/13/07     6
11/25/09     6
11/20/09     6
11/2/08      6
12/3/07      6
11/23/09     6
11/9/09      4
11/19/07     4
11/25/08     4
11/30/07     4
11/15/08     4
11/21/07     4
11/26/07     4
11/13/09     4
11/10/07     4
```

```
11/11/08     4
11/17/08     4
11/15/07     4
11/10/09     4
11/18/07     2
11/10/08     2
11/19/09     2
11/22/07     2
11/14/09     2
11/7/08      2
11/12/09     2
11/5/08      2
11/11/07     2
life_stage
Adult    344
egg_stage
 1 Egg Stage    344
```

We can see from these summary statistics that all the penguins sampled are adults and have a 1 Egg Stage. Therefore, we remove these columns and NaN rows.

In [14]:
```
# Removing unnnecessary variables
penguins.drop(['life_stage','egg_stage'], axis=1, inplace=True)
# Removing rows with NaN values
penguins.dropna(inplace=True)
# seeing the number of rows
print("Number of rows", penguins.shape[0])
print("Count of non-NaN rows per column:\n", penguins.count())
```

```
Number of rows 334
Count of non-NaN rows per column:
 species              334
island               334
culmen_length_mm     334
culmen_depth_mm      334
flipper_length_mm    334
body_mass_g          334
sex                  334
region               334
clutch_completion    334
date_egg             334
dtype: int64
```

In [15]:
```
# Getting a summary of numerical continuous variables
penguins.describe()
```

Out[15]:

	culmen_length_mm	culmen_depth_mm	flipper_length_mm	body_mass_g
count	334.000000	334.000000	334.000000	334.000000
mean	43.994311	17.160479	201.014970	4209.056886
std	5.460521	1.967909	14.022175	804.836129
min	32.100000	13.100000	172.000000	2700.000000
25%	39.500000	15.600000	190.000000	3550.000000
50%	44.500000	17.300000	197.000000	4050.000000
75%	48.575000	18.700000	213.000000	4793.750000
max	59.600000	21.500000	231.000000	6300.000000

We note that there is a slight change in the summary statistics of the numerical variables. The mean increased for every variable, while the standard deistribution increased or decreased for each variable. Other than that there was a slight increase for some values in the min, 25%, 50%, 75%, and max, while other values remained the same.

In [16]:
```
cat_var = ['species','island', 'sex', 'region', 'clutch_completion', 'date_egg']
for cat in cat_var:
    print(penguins[cat].value_counts(dropna=False).to_string())
```

```
species
Adelie       146
Gentoo       120
Chinstrap     68
island
Biscoe       164
Dream        123
Torgersen     47
sex
MALE      168
FEMALE    165
.           1
region
Anvers    334
clutch_completion
Yes    299
No      35
date_egg
11/27/07    18
11/16/07    15
11/9/08     15
11/18/09    14
11/13/08    12
11/4/08     12
11/6/08     12
11/21/09    12
11/15/09    10
11/22/09    10
11/16/09    10
11/14/08    10
11/17/09    10
11/27/09    10
11/29/07     9
11/24/08     8
11/8/08      8
11/28/07     8
11/12/07     8
11/3/08      8
12/3/07      6
11/23/09     6
11/25/09     6
12/1/09      6
11/2/08      6
11/20/09     6
11/13/07     5
11/30/07     4
11/19/07     4
11/11/08     4
11/25/08     4
11/21/07     4
11/26/07     4
11/9/07      4
11/13/09     4
11/10/09     4
11/17/08     4
11/10/07     4
```

```
11/15/07     4
11/9/09      4
11/15/08     4
11/18/07     2
11/12/09     2
11/19/09     2
11/22/07     2
11/7/08      2
11/14/09     2
11/5/08      2
11/10/08     2
11/11/07     2
```

We can see from the value counts of the categorical valriables that most of these variables are clean with the exception of sex, which has "." as one of its categories. Additionally the dates are arraged in year/day/month so we have to take that in mind when formatting the variable.

In [17]:
```
# finding the row number "." is as sex
penguins.loc[penguins['sex']=="."]
```

Out[17]:

	species	island	culmen_length_mm	culmen_depth_mm	flipper_length_mm	body_mass
336	Gentoo	Biscoe	44.5	15.7	217.0	487

In [18]:
```
# deleting row with "." as sex
penguins.drop(penguins.loc[penguins['sex']=="."].index, inplace=True)
penguins['sex'].value_counts(dropna=False)
```

Out[18]:
```
sex
MALE      168
FEMALE    165
Name: count, dtype: int64
```

In total we have 333 rows, each representing a penguin in the clean data set. Now we can proceed with the analysis.

Analysis

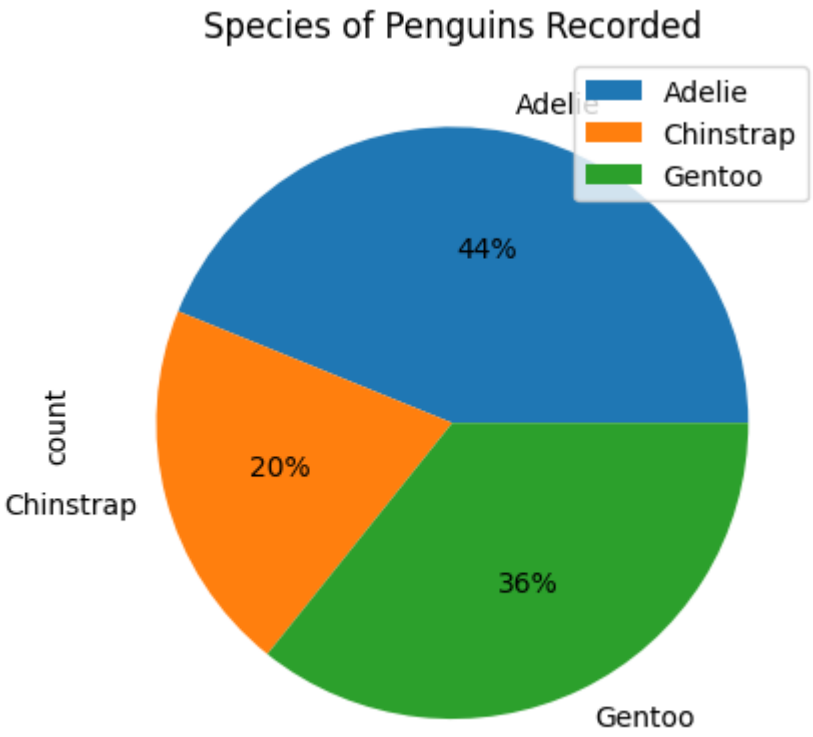
Now that we have a clean data set we proceed to take a closer look at the variables. While data cleaning, 11 out of 344 penguins did not have an identified sex, specifically for the 10 NaN values in this variable we mentioned previously it could be because of the methods to find a penguin's sex. Moreover we ask thhe following questions:

- What can we initially see from this data set?
- Are there differences in the three types of penguins? Consider island, clutch completion, sex ratios, and date of eggs.
- What is the relationship between culmen length and depth?

- Is there a difference between culmen length and culmen depth between male and female penguins? Are there differences as well with body mass and flipper length?
- Can we use a Generalized Linear Model (GLM) or a generalized additive model (GAM) to predict penguin sex using the quantitative variables as well as species in our dataset? Which of these methods is more effective?

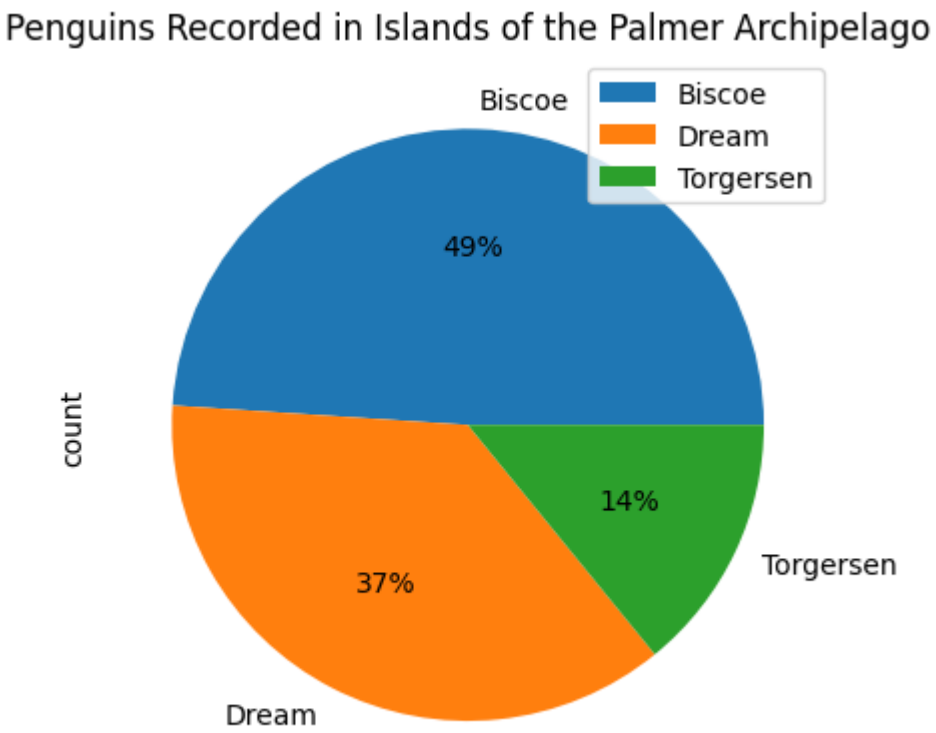
In [38]:
```
# Initial graphs of the data
# distribution of penguins on our data set
species = penguins.groupby(['species'])['species'].count().reset_index(name='count'
species.groupby(['species']).sum().plot(kind='pie', y='count', autopct='%1.0f%%', t
```

Out[38]:
```
<Axes: title={'center': 'Species of Penguins Recorded'}, ylabel='count'>
```



In [37]:
```
# distribution of penguins in each island on our data set
island = penguins.groupby(['island'])['island'].count().reset_index(name='count')
island.groupby(['island']).sum().plot(kind='pie', y='count', autopct='%1.0f%%', tit
```

Out[37]:
```
<Axes: title={'center': 'Penguins Recorded in Islands of the Palmer Archipelago'},
ylabel='count'>
```



In [94]:
```
# contingency table between species and island
species_location = pd.crosstab(penguins['species'], penguins['island'], margins = T
species_location
```

Out[94]:

species \ island	Biscoe	Dream	Torgersen	All
Adelie	44	55	47	146
Chinstrap	0	68	0	68
Gentoo	119	0	0	119
All	163	123	47	333

From the pie charts and contingency table for species and island, we note that almost half of the penguins recorded (44%) are from the Adelie species followed by Gentoo (36%) and Chinstrap (20%), about half of the penguins (49%) were recorded in Biscoe island followed by Dream (37%) and Torgersen (14%). In Biscoe island, we can find Adelie and Gentoo penguins, while in Dream island we can find Adelie and Chinstrap, and there are only Adelie Penguins in Torgersen island.

In [55]:
```
# sex of penguins
sex = penguins.groupby(['sex'])['sex'].count().reset_index(name='count')
sex['percent'] = 100*sex['count']/sex['count'].sum()
sex
```

Out[55]:

	sex	count	percent
0	FEMALE	165	49.54955
1	MALE	168	50.45045

In [95]:
```
# contingency table between species and sex
species_sex = pd.crosstab(penguins['species'], penguins['sex'], margins = True)
species_sex
```

Out[95]:

sex	FEMALE	MALE	All
species			
Adelie	73	73	146
Chinstrap	34	34	68
Gentoo	58	61	119
All	165	168	333

From the table for sex we note that there is a close equal ratio of female to male penguins, with 49.55% of penguins recorded being female, and 50.45% being male. From the contingency table between species annd sex, we also note that there is an equal number of males and females in Adelie and Chinstrap species, and a slightly larger number of males in the Gentoo species.

In [57]:
```
# clutch completion of all penguins
clutch = penguins.groupby(['clutch_completion'])['clutch_completion'].count().reset
clutch['percent'] = 100*clutch['count']/clutch['count'].sum()
clutch
```

Out[57]:

	clutch_completion	count	percent
0	No	35	10.510511
1	Yes	298	89.489489

In [96]:
```
# contingency table between species and clutch completion
species_clutch = pd.crosstab(penguins['species'], penguins['clutch_completion'], ma
species_clutch
```

Out[96]:

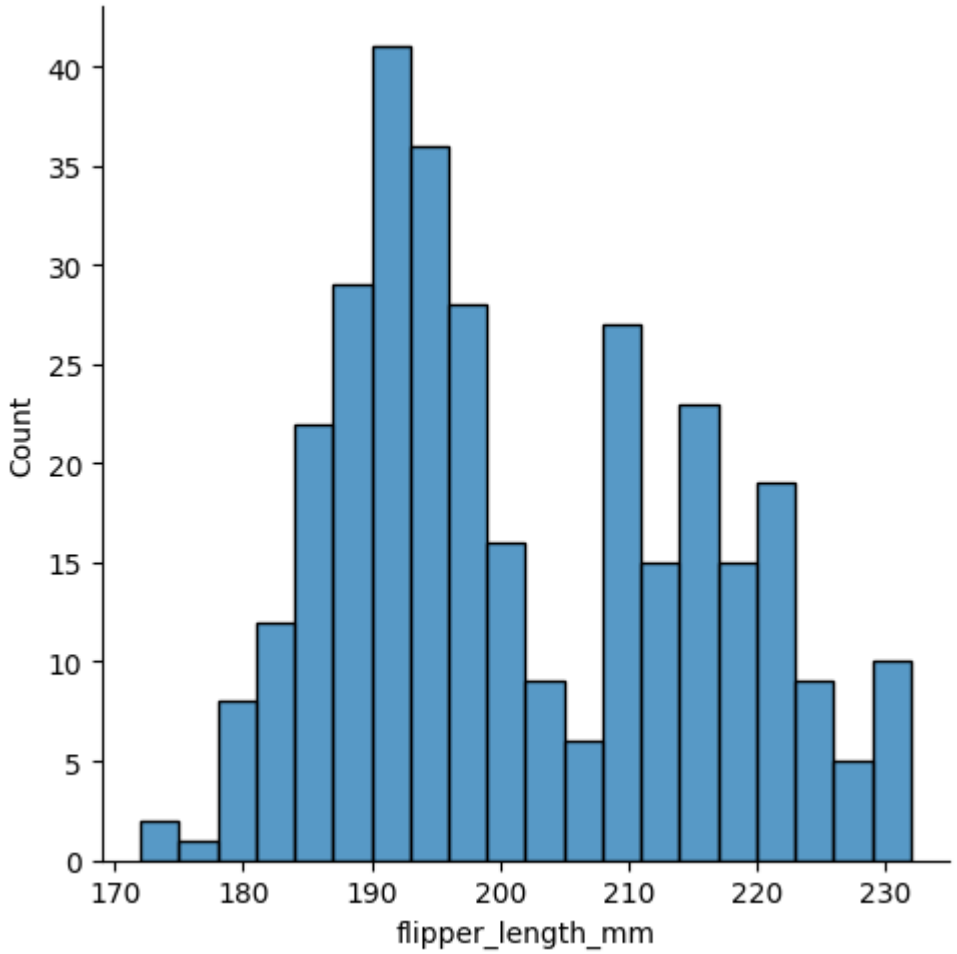
clutch_completion	No	Yes	All
species			
Adelie	14	132	146
Chinstrap	14	54	68
Gentoo	7	112	119
All	35	298	333

From the table for clutch completion, we can see that 89.49% of penguins completed their clutches, while 10.51% have not. From the contingency table between species and clutch completion, we can see that most penguins of each species have completed their clutches, but Chinstrap penguins have the highest percent in not completed clutches with 20.59% while the other species have less than 10% non completion. We look at the distributions of the continuous numerical variables:

In [73]:
```
# distribution plot of flipper length of all penguins
sns.displot(x = penguins.flipper_length_mm, binwidth=3)
```

```
C:\Users\Mili\AppData\Local\Programs\Python\Python311\Lib\site-packages\seaborn\axisgrid.py:118: UserWarning: The figure layout has changed to tight
  self._figure.tight_layout(*args, **kwargs)
```

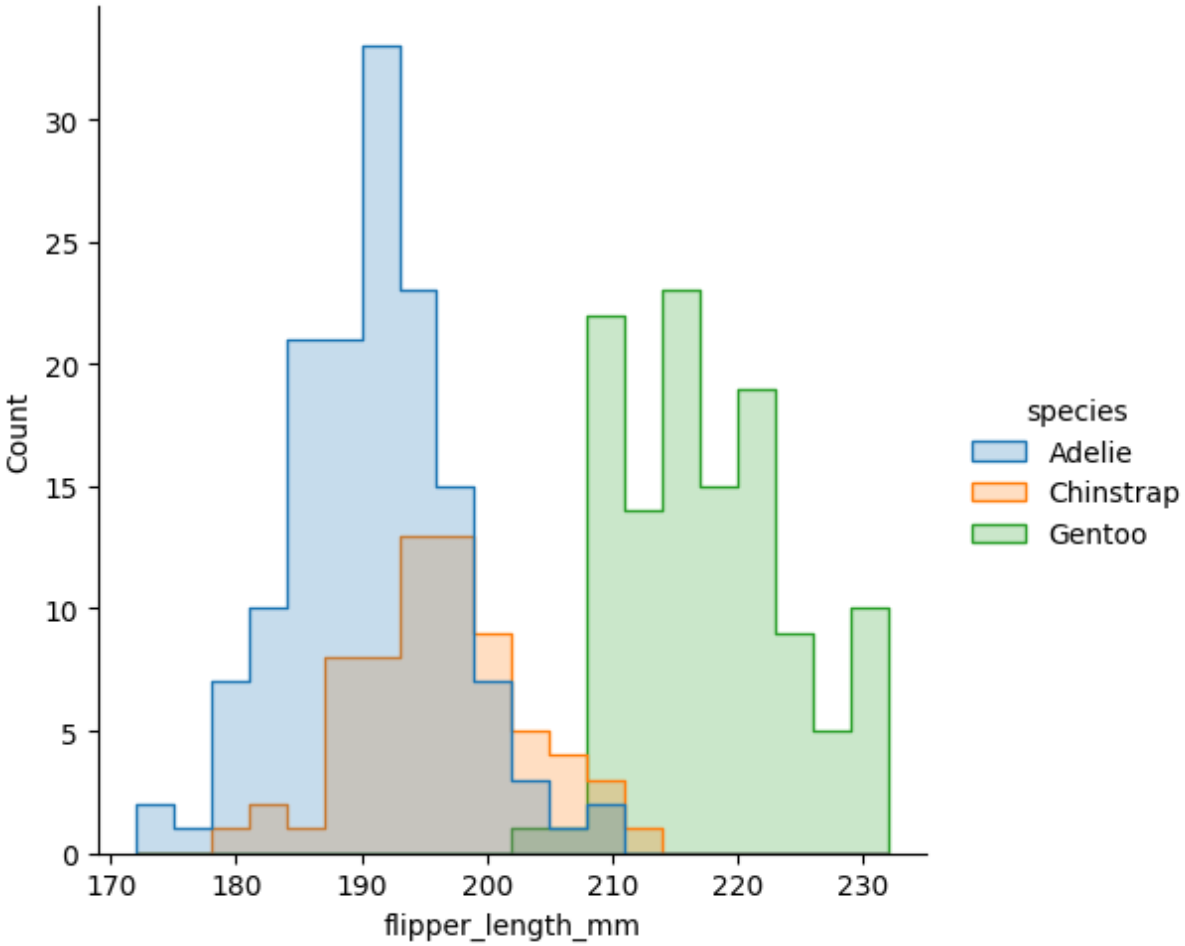
Out[73]:
```
<seaborn.axisgrid.FacetGrid at 0x1e254abf850>
```



In [93]:
```
# distribution plot of flipper length for each species of penguin
sns.displot(x = penguins.flipper_length_mm, binwidth=3, hue=penguins.species, eleme
```

```
C:\Users\Mili\AppData\Local\Programs\Python\Python311\Lib\site-packages\seaborn\axis
grid.py:118: UserWarning: The figure layout has changed to tight
  self._figure.tight_layout(*args, **kwargs)
```

Out[93]: <seaborn.axisgrid.FacetGrid at 0x1e25a9e6c90>



We can see that the distribution for flipper length of all penguins in the dataset is unimodal and right-skewed with a range from 172 mm to 231 mm. When we break this distribution by species we can see that Gentoo penguins have a flipper length distribution that is higher than Adelie and Chinstrap penguins. Is it significant? We perform a one-way ANOVA test with the following hypothesis:

$H_{0}: \mu_{Adelie}=\mu_{Chinstrap}=\mu_{Gentoo}$

$H_{a}:$ At least one population mean for flipper length is different from the rest

In [101…
```
stats.levene(penguins['flipper_length_mm'][penguins['species']=='Adelie'],penguins[
```

Out[101…
```
LeveneResult(statistic=0.44278650514651297, pvalue=0.6426253107522972)
```

At $\alpha = 0.05$ Levene's test of Homogeneity is not significant which indicates the groups have non-statistically significant difference in thier variability.

In [102…
```
stats.f_oneway(penguins['flipper_length_mm'][penguins['species']=='Adelie'],penguin
```

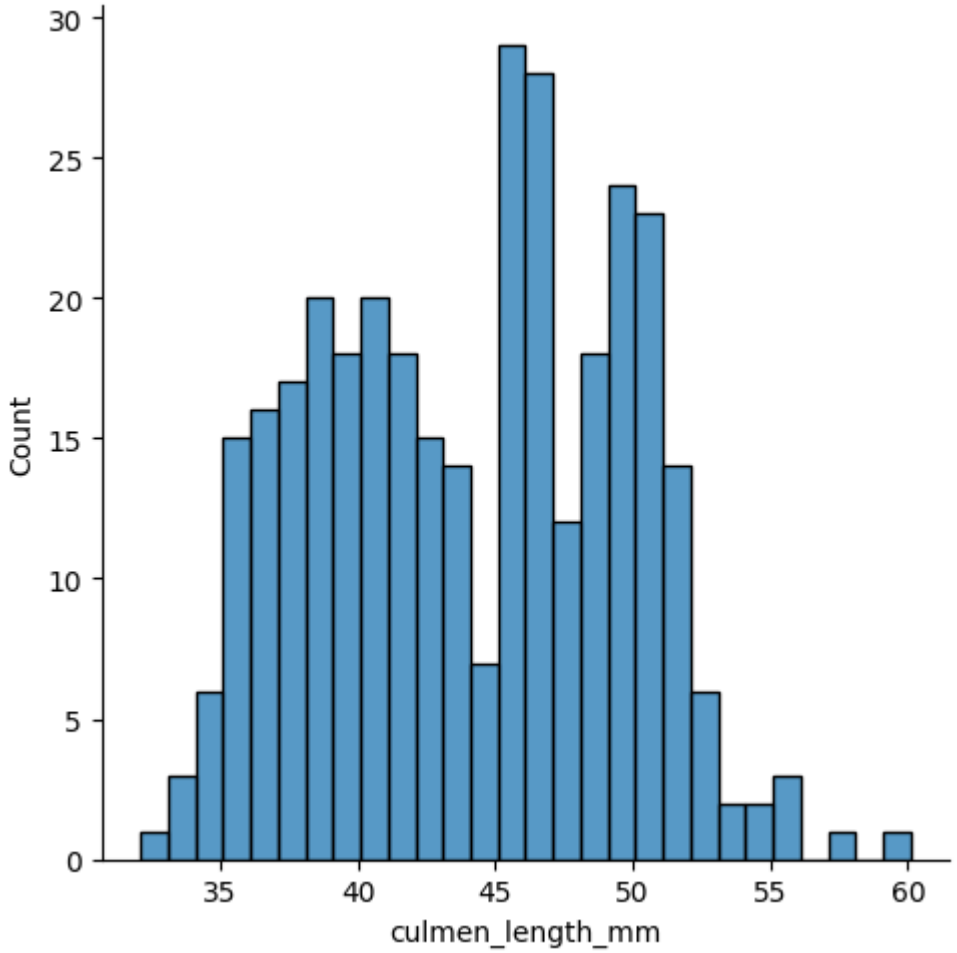
Out[102…
```
F_onewayResult(statistic=567.4069920123421, pvalue=1.5874180554406245e-107)
```

At $\alpha = 0.05$ there is a significant difference between the species flipper length.

In [77]:
```
# distribution plot of culmen length of all penguins
sns.displot(x = penguins.culmen_length_mm, binwidth=1)
```

```
C:\Users\Mili\AppData\Local\Programs\Python\Python311\Lib\site-packages\seaborn\axisgrid.py:118: UserWarning: The figure layout has changed to tight
  self._figure.tight_layout(*args, **kwargs)
```

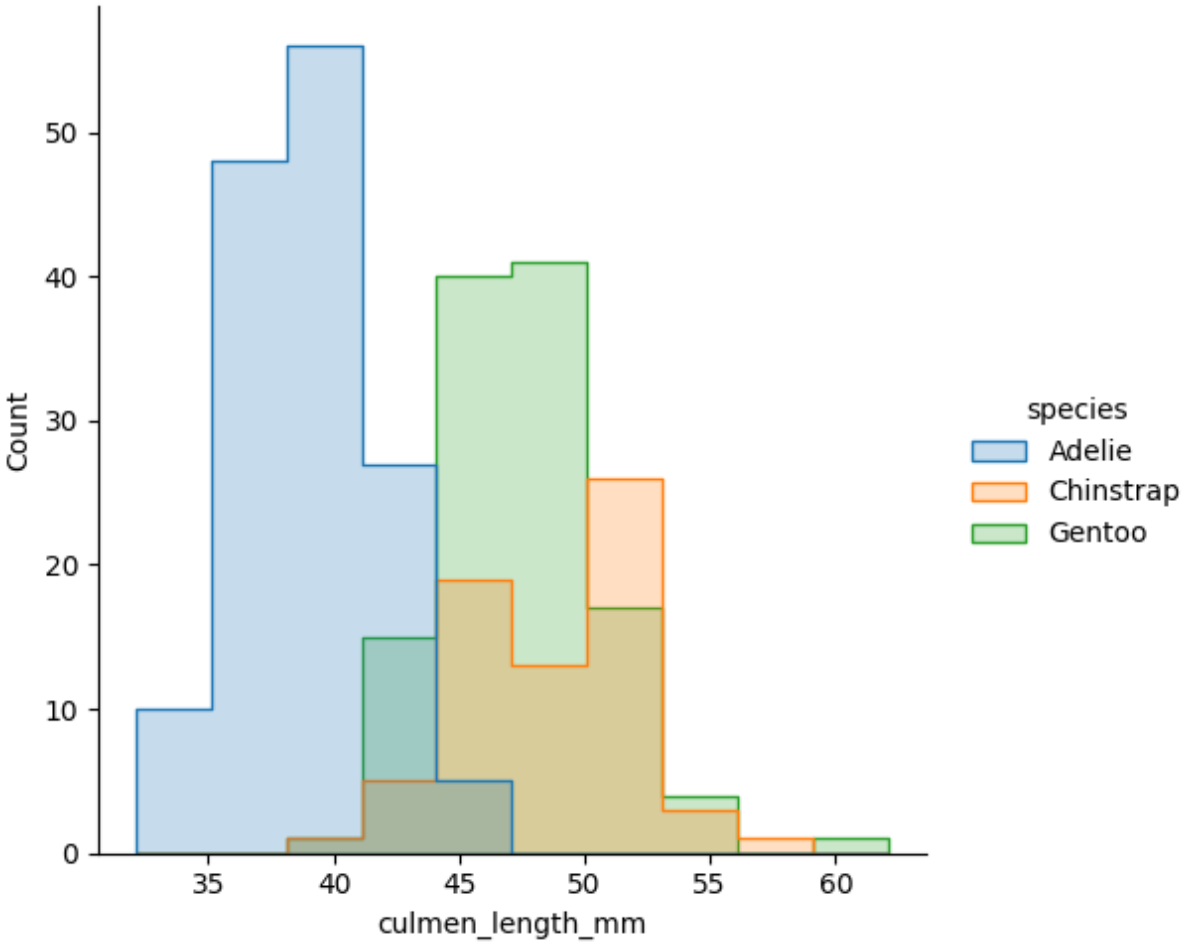
Out[77]:
```
<seaborn.axisgrid.FacetGrid at 0x1e25662bf90>
```



In [103…
```
# distribution plot of culmen length for each species of penguin
sns.displot(x = penguins.culmen_length_mm, binwidth=3, hue=penguins.species, elemen
```

```
C:\Users\Mili\AppData\Local\Programs\Python\Python311\Lib\site-packages\seaborn\axisgrid.py:118: UserWarning: The figure layout has changed to tight
  self._figure.tight_layout(*args, **kwargs)
```

Out[103…
```
<seaborn.axisgrid.FacetGrid at 0x1e261753d90>
```



We can see that the distribution for culmen length is overall unimodal and slightly left-skewed with a range from 32.1 mm to 59.6 mm. Chinstrap and Gentoo penguins appear to have a higher distribution for culmen length compared to Adelie. Is it significant? We perform a one-way ANOVA test with the following hypothesis:

$H_{0}: \mu_{Adelie}=\mu_{Chinstrap}=\mu_{Gentoo}$

$H_{a}:$ At least one population mean for culmen length is different from the rest

In [104…
```
stats.levene(penguins['culmen_length_mm'][penguins['species']=='Adelie'],penguins['
```

Out[104…
```
LeveneResult(statistic=2.285530220618524, pvalue=0.10332812696137215)
```

In [105…
```
stats.f_oneway(penguins['culmen_length_mm'][penguins['species']=='Adelie'],penguins
```

Out[105…
```
F_onewayResult(statistic=397.29943741282835, pvalue=1.3809842053150047e-88)
```

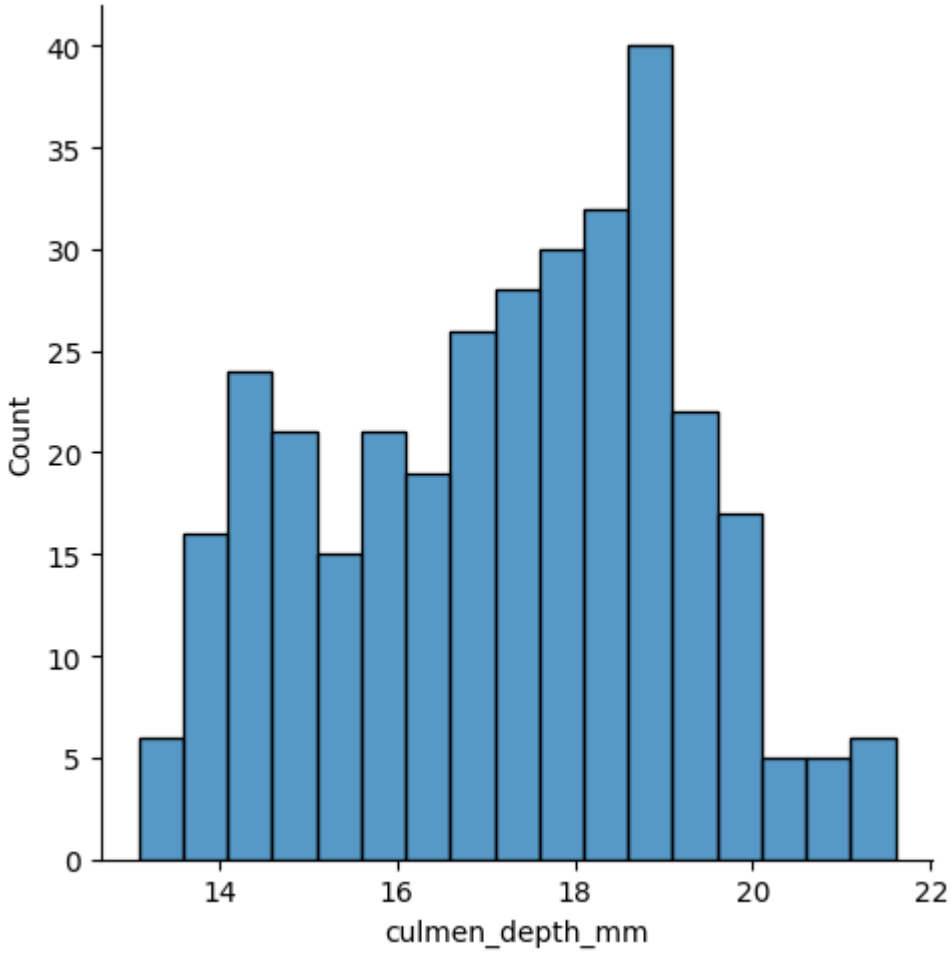
At $\alpha = 0.05$ there is a significant difference between the species culmen length. Additionally, Levene's test of Homogeneity is not significant which indicates the groups have non-statistically significant difference in thier variability.

In [106…
```
# distribution plot of culmen depth for each species of penguin
sns.displot(x = penguins.culmen_depth_mm, binwidth=.5)
```

```
C:\Users\Mili\AppData\Local\Programs\Python\Python311\Lib\site-packages\seaborn\axisgrid.py:118: UserWarning: The figure layout has changed to tight
  self._figure.tight_layout(*args, **kwargs)
```

Out[106…

```
<seaborn.axisgrid.FacetGrid at 0x1e261c6c690>
```



We can see that the distribution for culmen depth is overall unimodal and left-skewed with a range from 13.1 mm to 21.5 mm.

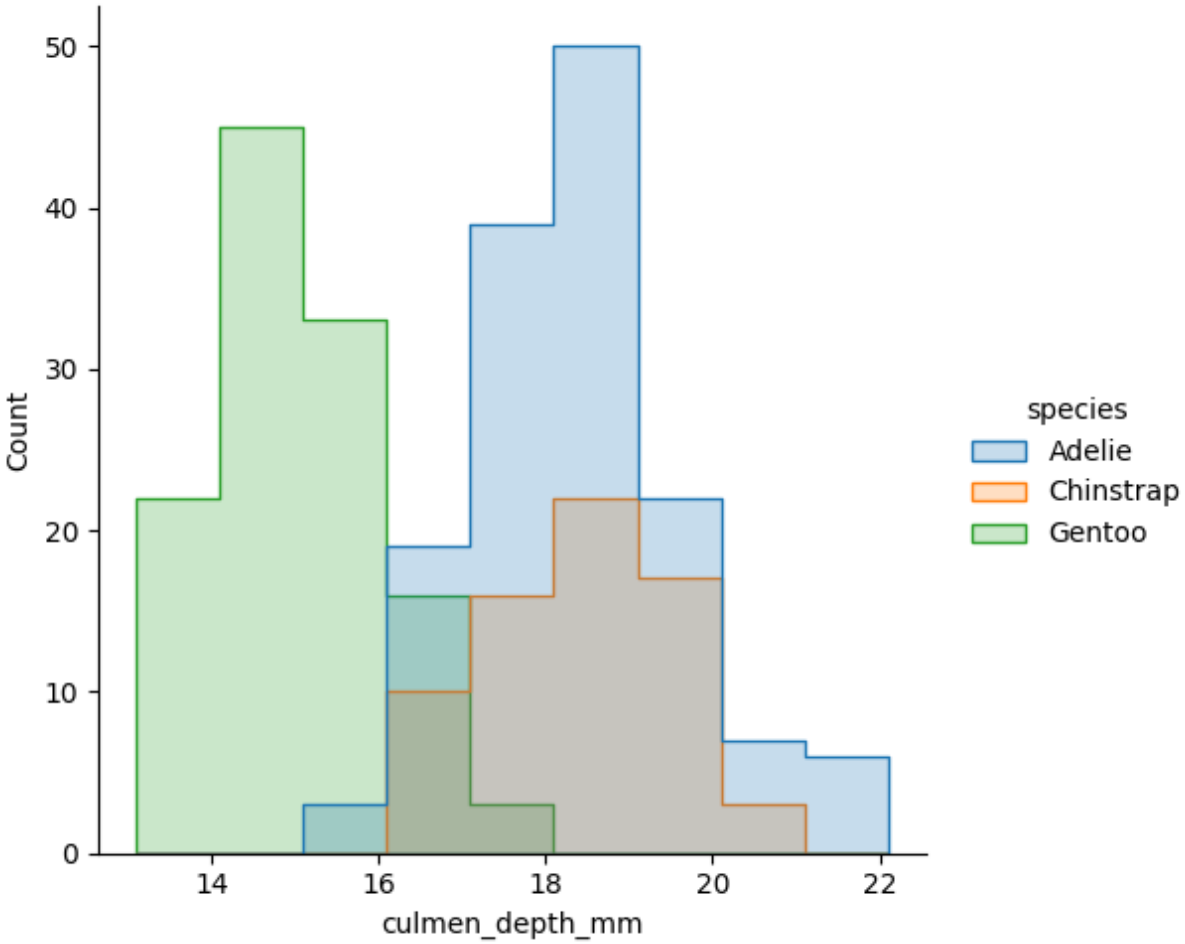
In [108…

```
# distribution plot of culmen depth for each species of penguin
sns.displot(x = penguins.culmen_depth_mm, binwidth=1, hue=penguins.species, element
```

```
C:\Users\Mili\AppData\Local\Programs\Python\Python311\Lib\site-packages\seaborn\axisgrid.py:118: UserWarning: The figure layout has changed to tight
  self._figure.tight_layout(*args, **kwargs)
```

Out[108…

```
<seaborn.axisgrid.FacetGrid at 0x1e25a2fa850>
```



We can see that the distribution for culmen depth is overall unimodal and slightly left-skewed with a range from 13.1 mm to 21.5 mm. Chinstrap and Adelie penguins appear to have a higher distribution for culmen depth compared to Gentoo. Is it significant? We perform a one-way ANOVA test with the following hypothesis:

$H_{0}: \mu_{Adelie}=\mu_{Chinstrap}=\mu_{Gentoo}$

$H_{a}:$ At least one population mean for culmen depth is different from the rest

In [109…
```
stats.levene(penguins['culmen_depth_mm'][penguins['species']=='Adelie'],penguins['c
```

Out[109…
```
LeveneResult(statistic=1.9124171393991907, pvalue=0.14935650320624513)
```

In [110…
```
stats.f_oneway(penguins['culmen_depth_mm'][penguins['species']=='Adelie'],penguins[
```

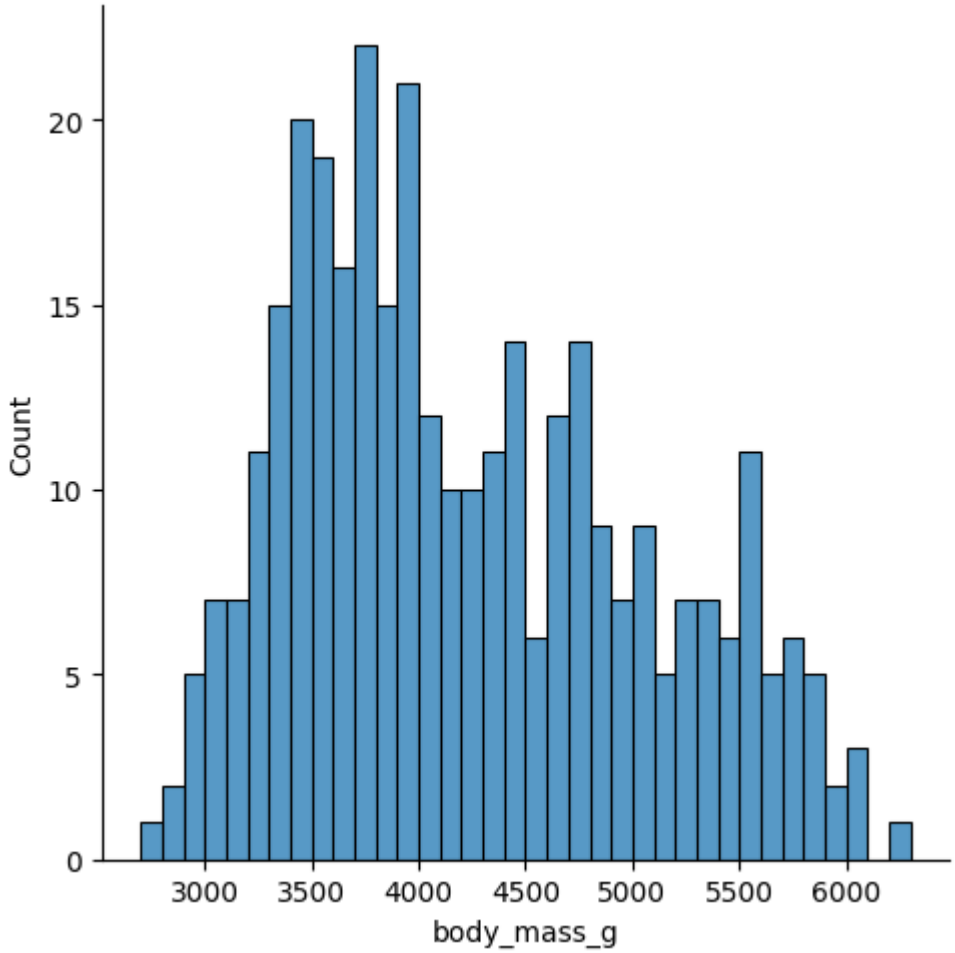
Out[110…
```
F_onewayResult(statistic=344.82508194378835, pvalue=1.4466156955780345e-81)
```

At $\alpha = 0.05$ there is a significant difference between the species culmen depth. Additionally, Levene's test of Homogeneity is not significant which indicates the groups have non-statistically significant difference in thier variability.

In [79]:
```
sns.displot(x = penguins.body_mass_g, binwidth=100)
```

```
C:\Users\Mili\AppData\Local\Programs\Python\Python311\Lib\site-packages\seaborn\axisgrid.py:118: UserWarning: The figure layout has changed to tight
  self._figure.tight_layout(*args, **kwargs)
```

Out[79]: <seaborn.axisgrid.FacetGrid at 0x1e25694bd10>



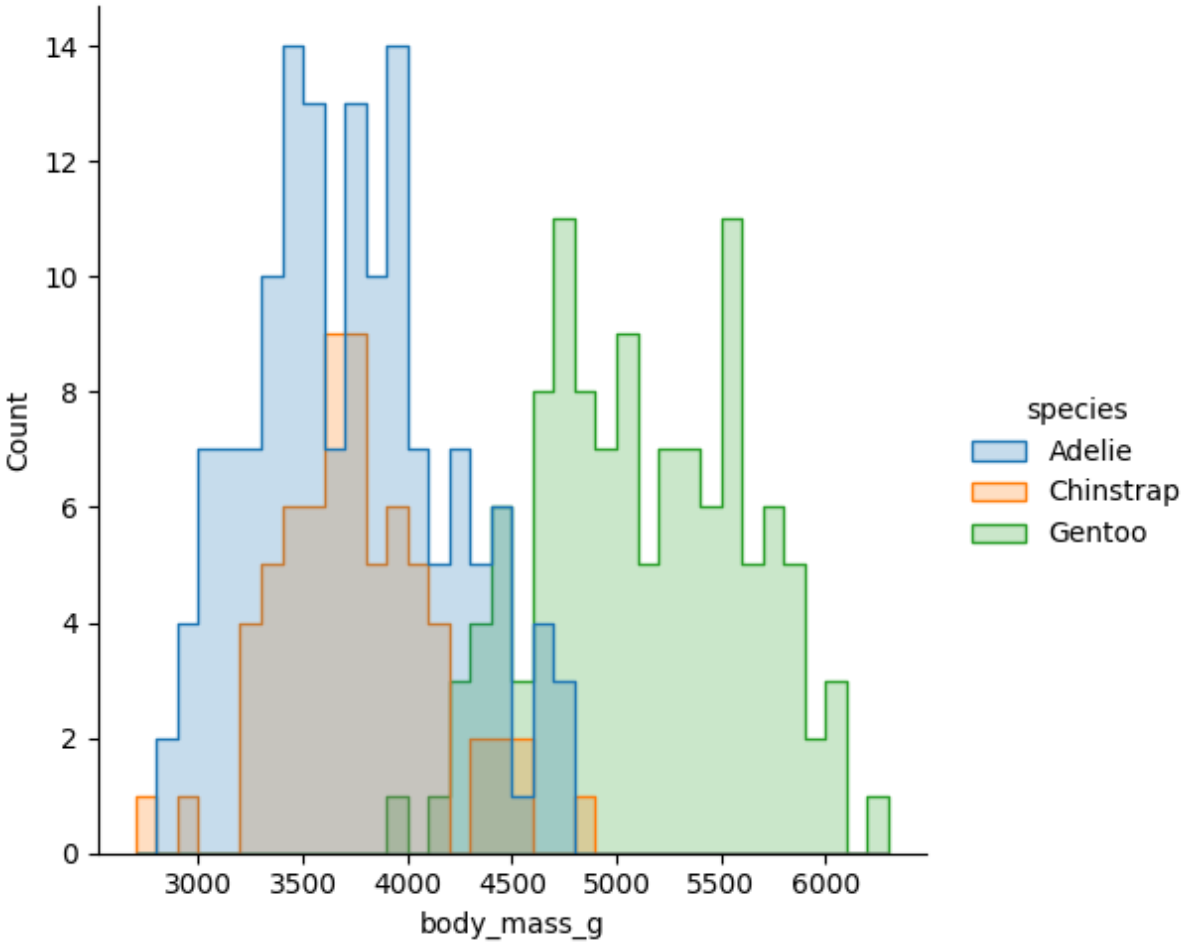
We can see that the distribution for body mass is overall unimodal and right-skewed with a range from 2700 g to 6300 g.

In [112…

```
# distribution plot of body mass for each species of penguin
sns.displot(x = penguins.body_mass_g, binwidth=100, hue=penguins.species, element='
```

```
C:\Users\Mili\AppData\Local\Programs\Python\Python311\Lib\site-packages\seaborn\axisgrid.py:118: UserWarning: The figure layout has changed to tight
  self._figure.tight_layout(*args, **kwargs)
```

Out[112… <seaborn.axisgrid.FacetGrid at 0x1e261e0dfd0>



Gentoo penguins appear to have a higher distribution for body mass compared to Adelie and Chinstrap. Is it significant? We perform a one-way ANOVA test with the following hypothesis:

$H_{0}: \mu_{Adelie}=\mu_{Chinstrap}=\mu_{Gentoo}$

$H_{a}:$ At least one population mean for body mass is different from the rest

In [113…
```
stats.levene(penguins['body_mass_g'][penguins['species']=='Adelie'],penguins['body_
```

Out[113…
```
LeveneResult(statistic=5.134899089832661, pvalue=0.0063670535459093135)
```

In [114…
```
stats.f_oneway(penguins['body_mass_g'][penguins['species']=='Adelie'],penguins['bod
```

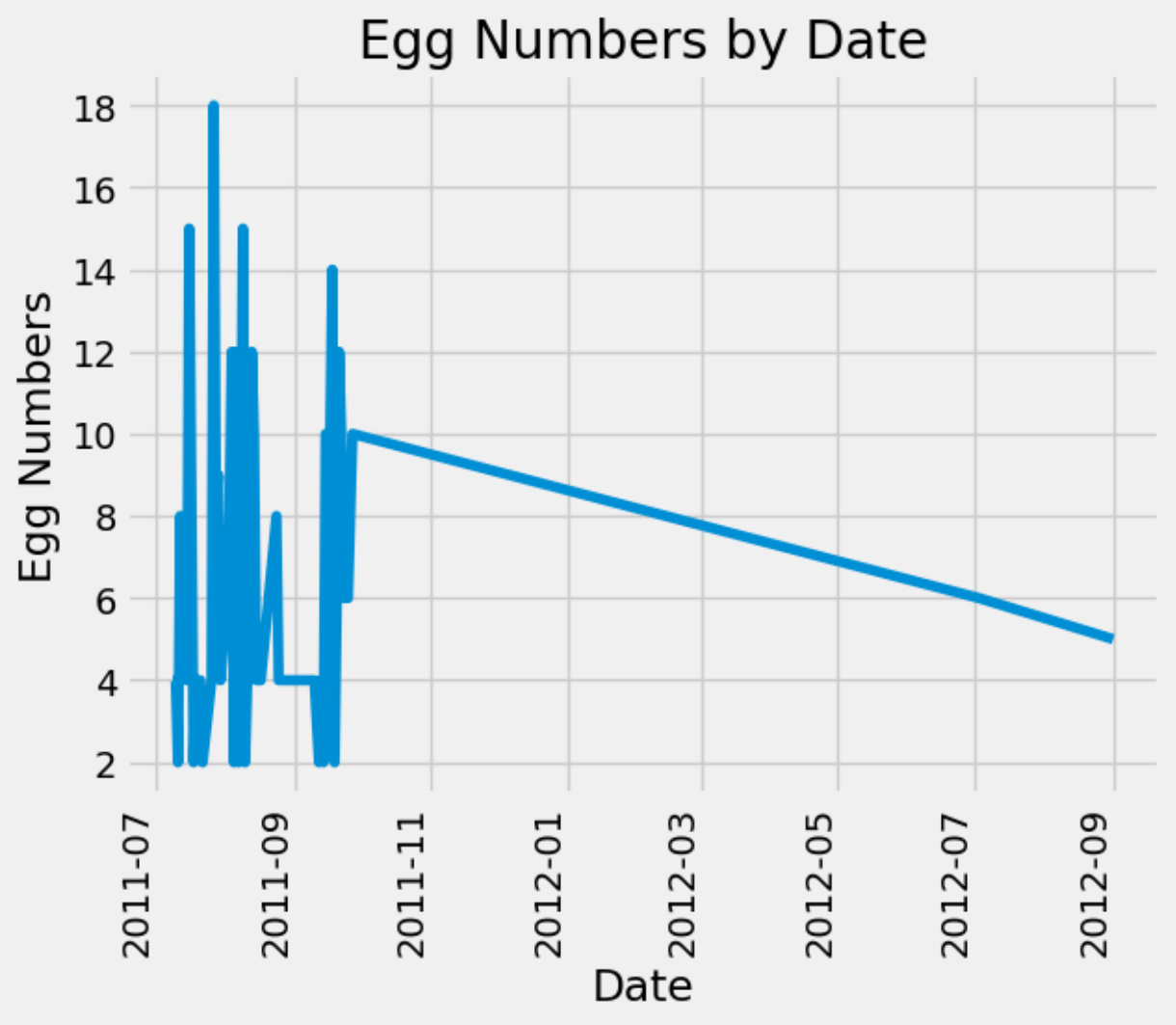
Out[114…
```
F_onewayResult(statistic=341.8948949481461, pvalue=3.744505126300443e-81)
```

At $\alpha = 0.05$ there is a significant difference between the species body mass. Additionally, Levene's test of Homogeneity is significant which indicates the groups have a statistically significant difference in thier variability. which invalidates the results.

In [143…
```
# Time plot between egg_date and egg numbers
penguins['date_egg']=pd.to_datetime(penguins['date_egg'], format='%y/%d/%m')
time = penguins.groupby(['date_egg'])['date_egg'].count().reset_index(name='counts'
```

```
plt.plot(time.date_egg, time.counts)
plt.title('Egg Numbers by Date')
plt.xticks(rotation=90, ha='right')
plt.xlabel('Date')
plt.ylabel('Egg Numbers')
```

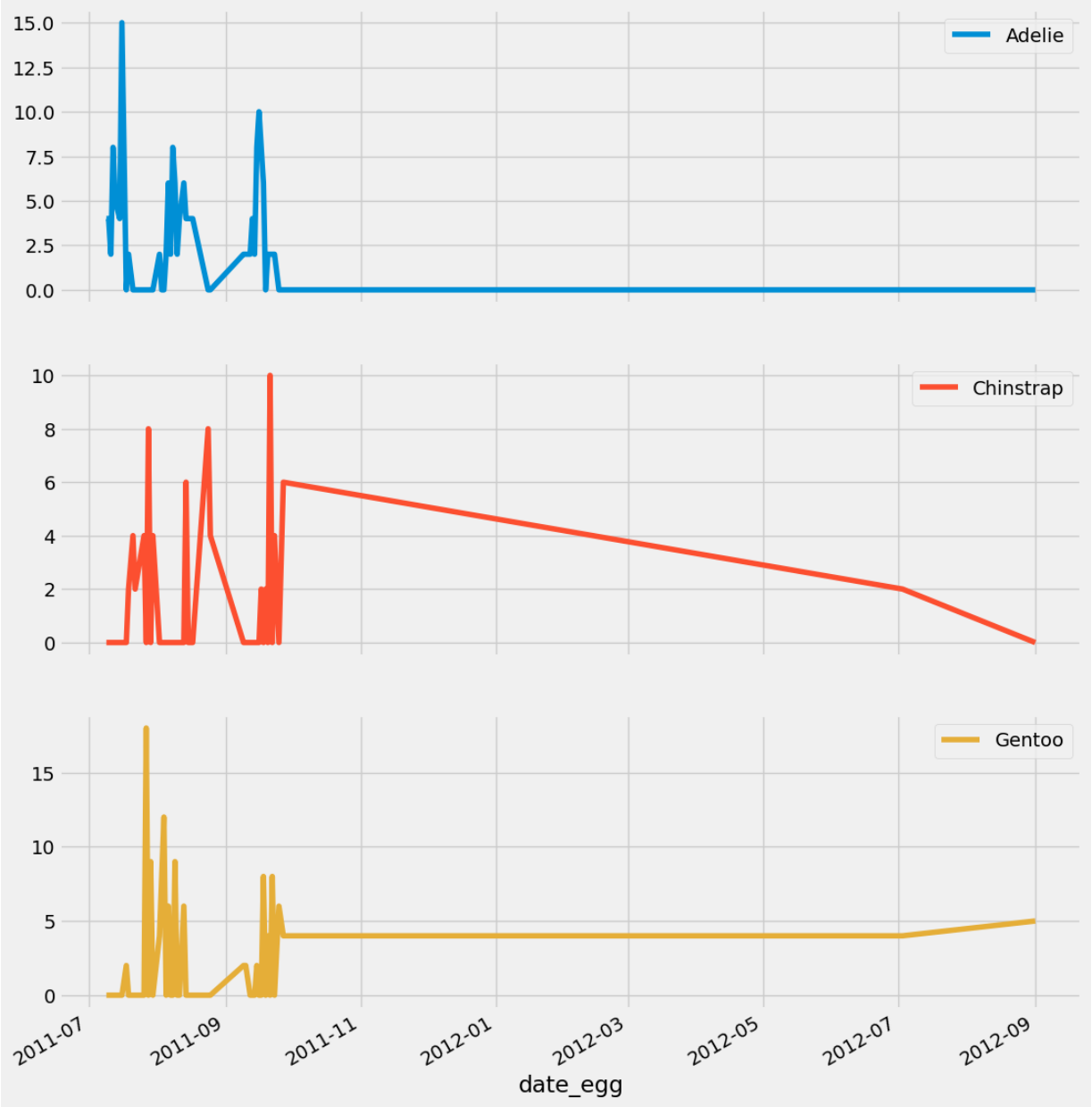
Out[143… `Text(0, 0.5, 'Egg Numbers')`



From this time plot, we can see that there was a peak of 18 of egg numbers in 11/27/09.

In [146…
```
# egg date and egg numbers by species
species_egg = pd.crosstab(penguins['date_egg'], penguins['species'], margins = Fals
plt.style.use('fivethirtyeight')
species_egg.plot(subplots = True, figsize=(12,15))
```

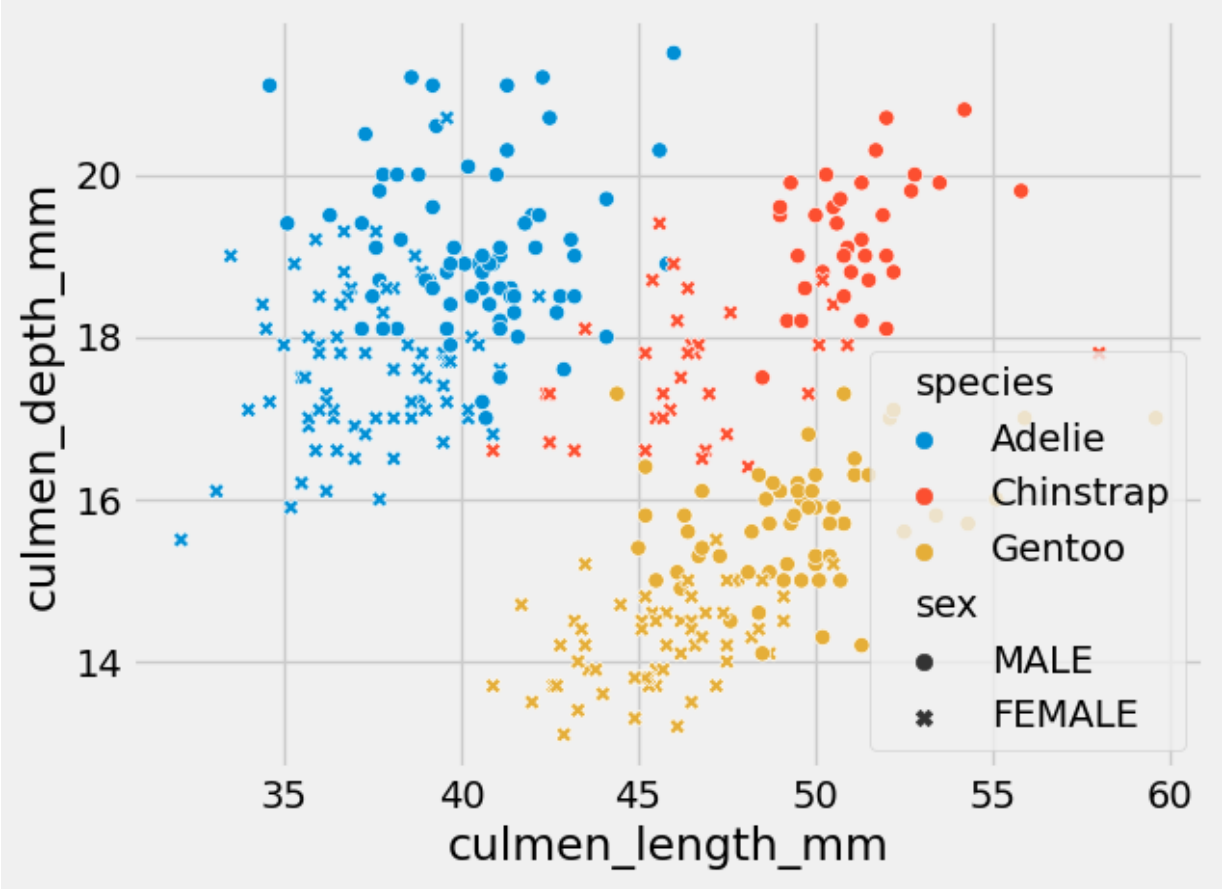
Out[146…
```
array([<Axes: xlabel='date_egg'>, <Axes: xlabel='date_egg'>,
       <Axes: xlabel='date_egg'>], dtype=object)
```



We can see that all three species of penguins have peaks for high egg counts in different times, and that there is a large gap of time in the data from October 2011 to July 2012. Now we explore culmen length and depth in more detail:

In [152…
```
# scatterplot between culmen length and depth colour-coded by species and using dif
sns.scatterplot(data=penguins, x='culmen_length_mm', y='culmen_depth_mm', hue='spec
```

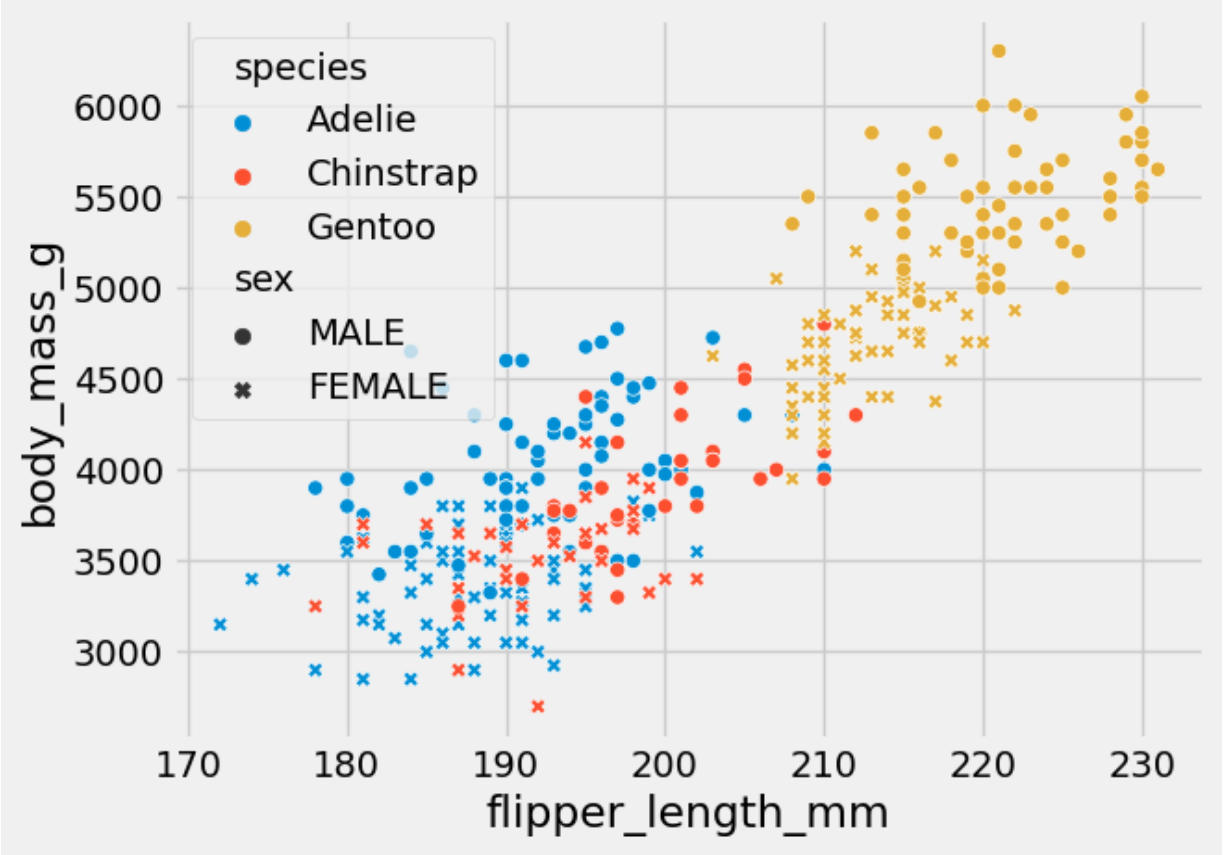
Out[152…
```
<Axes: xlabel='culmen_length_mm', ylabel='culmen_depth_mm'>
```



We can see that there are clear clusters for each species, and that the males for each species have a longer culmen length and depth from each species females. Using only culmen length and depth without considering species might not be useful to determine a penguin's sex. We now explore body mass and flipper length:

In [155…
```
# scatterplot between flipper length and body mass colour-coded by species and usin
sns.scatterplot(data=penguins, x='flipper_length_mm', y='body_mass_g', hue='species
```

Out[155…
```
<Axes: xlabel='flipper_length_mm', ylabel='body_mass_g'>
```



We can see in the scatter plot between body mass and flipper length has a positive relationship. Coded by sex and species, we see that Gentoo penguins have a larger flipper length and body mass. Adelie and Chinstrap penguins have similar body and flipper length distributions. We can also see that for each species females have a lower body mass and flipper length. Now we look at different models to predict the sex of penguins.

Generalized Linear Model (GLM)

GLM is a conventional linear regression model with a random component, a systematic component, and a link function that works with categorical and continuous variables, it includes multiple linear regression as well as ANOVA and ANCOVA. One of the most popular GLM models is the binary logistic regression model which has the form

$logit(\pi_{i})=log(\frac{\pi_{i}}{1-\pi_{i}})=\beta_{0}+\beta_{1}x_{i}$

the response variable is assumed to be binomial, in this case, male or female, x_{i} is the explanatory variable, and the link function is $log(\frac{\pi_{i}}{1-\pi_{i}})$ known as the logit function.

In [201…

```
# splitting the dataset into training and testing models
train = penguins.sample(frac = 0.75, random_state=200)
test = penguins.drop(train.index)
# Fitting, modeling and predicting GLM model
glmodel = smf.glm(formula = 'sex ~ culmen_length_mm + culmen_depth_mm + body_mass_g
glmresult = glmodel.fit()
```

```
glm_predict_sex = glmresult.predict(test)
print(glmresult.summary())
```

```
                 Generalized Linear Model Regression Results
==============================================================================
====
Dep. Variable:     ['sex[FEMALE]', 'sex[MALE]']   No. Observations:
250
Model:                                   GLM   Df Residuals:
243
Model Family:                        Binomial   Df Model:
6
Link Function:                          Logit   Scale:                          1.
0000
Method:                                  IRLS   Log-Likelihood:                -5
0.584
Date:                       Sat, 12 Aug 2023   Deviance:                       10
1.17
Time:                               02:17:53   Pearson chi2:
319.
No. Iterations:                            7   Pseudo R-squ. (CS):             0.
6253
Covariance Type:                   nonrobust
==============================================================================
====
                         coef    std err          z      P>|z|      [0.025      0.
975]
------------------------------------------------------------------------------
----
Intercept              72.8381     12.247      5.947      0.000      48.834      9
6.842
species[T.Chinstrap]    5.6970      1.579      3.609      0.000       2.603
8.791
species[T.Gentoo]       6.4754      2.638      2.455      0.014       1.305      1
1.646
culmen_length_mm       -0.5103      0.131     -3.881      0.000      -0.768      -
0.253
culmen_depth_mm        -1.6050      0.367     -4.373      0.000      -2.324      -
0.886
body_mass_g            -0.0055      0.001     -4.743      0.000      -0.008      -
0.003
flipper_length_mm      -0.0156      0.050     -0.311      0.756      -0.114
0.083
==============================================================================
====
```

we can see that the dependent variable for sex has been converted from nominal to two dummy variables ['sex['FEMALE']' , 'sex['MALE']']. We note that this encodes the variable to a binary with 1 being FEMALE and 0 being MALE. Thus, we classify the predictions higher than 0.5 to FEMALE and thanse lower than 0.5 to MALE.

In [202…

```
glm_bin_sex = ['MALE' if x < 0.5 else 'FEMALE' for x in glm_predict_sex]
glm_sex_conftab = pd.crosstab(test['sex'], glm_bin_sex, margins = True)
glm_sex_conftab
```

Out[202…

col_0	FEMALE	MALE	All
sex			
FEMALE	38	2	40
MALE	4	39	43
All	42	41	83

we can see that we have 38 + 39 = 77 correct predictions out of 83 which is about 92.77%. This means that our training error rate is 7.23%. The model is pretty accurate at predicting penguin's sex using flipper length, culmen length, culmen depth, species, and body mass.

Now let's try using a generalized additive model.

Generalized Additive Model (GAM)

Generalized additive models allow us to model non-linear data and have non-linear features making it different to GLM. The response variable follows a probability distribution which in this case we like to be binomial. The model consists in smooth functions estimated from the data using spline smoothing techniques to best fit the data while balancing the function's goodness of fit. This model has the form:

$y = \beta_{0}+f_{1}(x_{1})+f_{2}(x_{2})+...+f_{p}(x_{p})+\epsilon $ where $f_{i}(x_{i})$ are smooth non-linear functions.

In [210…

```
features = ['culmen_length_mm', 'culmen_depth_mm', 'body_mass_g', 'flipper_length_m
x_train = pd.get_dummies(train, columns =['species'], dtype = int)
x_train = x_train[features]
y_train = pd.get_dummies(train, columns=['sex'], dtype = int)
y_train = y_train['sex_FEMALE']
x_test = pd.get_dummies(test, columns =['species'], dtype = float)
x_test = x_test[features]
y_test = test['sex']
classifier = LogisticGAM().fit(x_train, y_train)
gam_predict_sex = (classifier.predict(x_test))*1
gam_bin_sex = ['MALE' if x < 0.5 else 'FEMALE' for x in gam_predict_sex]
gam_sex_conftab = pd.crosstab(y_test, gam_bin_sex, margins = True)
gam_sex_conftab
```

Out[210…

col_0	FEMALE	XALE	All
sex			
FEMALE	36	4	40
MALE	4	39	43
All	40	43	83

we can see that we have 36 + 39 = 75 correct predictions out of 83 which is about 88.24%. This means that our training error rate is 11.76%. The model is less accurate than GLM at predicting penguins' sex using flipper length, culmen length, culmen depth, species, and body mass.

Conclusion

From the penguin data of the Palmer Archipelago, we found that a great percentage of penguins were recorded in Biscoe island, a majority of all the recorded penguins were Adelie, most of the recorded penguins had complete clutches, and there were slightly more males than females. The peak number of eggs for Adelie and Gentoo penguins was recorded in June while Chinstrap penguins had a peak number of eggs in late September of 2011. There was not enough data between October 2011 to June 2012 to see other patterns in the data in relation to the dates of eggs and penguin species.

Analyzing flipper length, body mass, culmen length, and culmen depth by penguin species we found that there is a clear distinction between each species based on culmen length and depth, and there was a clear distinction between Gentoo penguins to the other species based on body mass and flipper length. As we saw that within each species there was a distinction in males and females but not between species. Moreover, we considered species, body mass, flipper length, culmen length, and culmen depth as explanatory variables to predict sex in GLM and GAM. Both GLM and GAM had a high number of correct predictions, but GLM was more effective at predicting a penguin's sex. It would be interesting to apply other classification models such as SVM or classification trees to predict penguins' sex. From this analysis, however, we can take that culmen length and depth along with species, flipper length and body mass can be used to predict the sex of a penguin.

References

[1] Gorman KB, Williams TD, Fraser WR (2014) Ecological Sexual Dimorphism and Environmental Variability within a Community of Antarctic Penguins (Genus Pygoscelis). PLoS ONE 9(3): e90081. doi:10.1371/journal.pone.0090081

[2] Wiley. "Distinguishing males from females among king penguins." ScienceDaily. ScienceDaily, 22 February 2018. <www.sciencedaily.com/releases/2018/02/180222085655.htm>.

[3] Urton, J. (2018, June 27). To tell the sex of a Galápagos Penguin, measure its beak, researchers say. UW News. https://www.washington.edu/news/2018/06/27/to-tell-the-sex-of-a-galapagos-penguin-measure-its-beak-researchers-say/