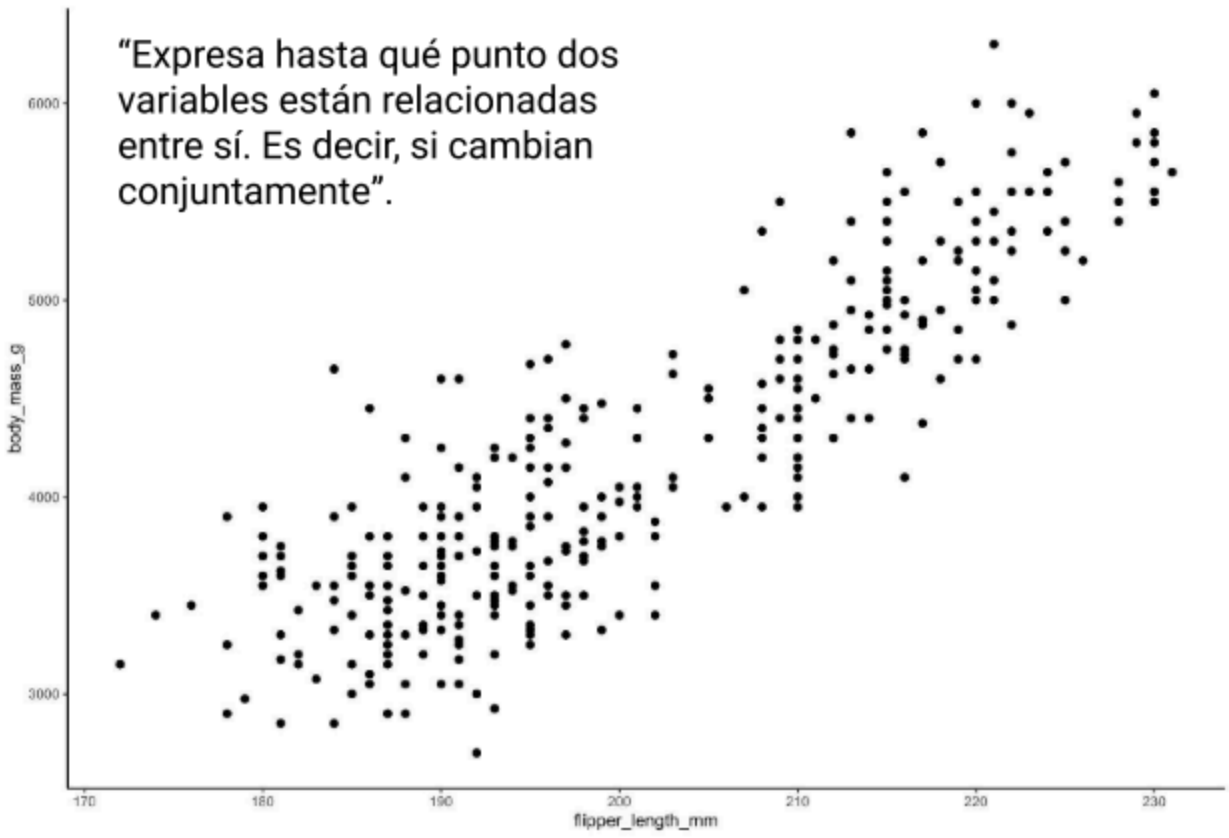
Estableciendo relaciones

En la clase pasada aprendimos a relacionar variables ya sea:

- numérica vs numérica.
- discreta vs numérica.
- numérica vs discreta.

Pero llega un punto en donde necesitamos saber si nuestras variables están correlacionadas y hasta que punto, también si cambian conjuntamente



Ejemplo tenemos las variables:

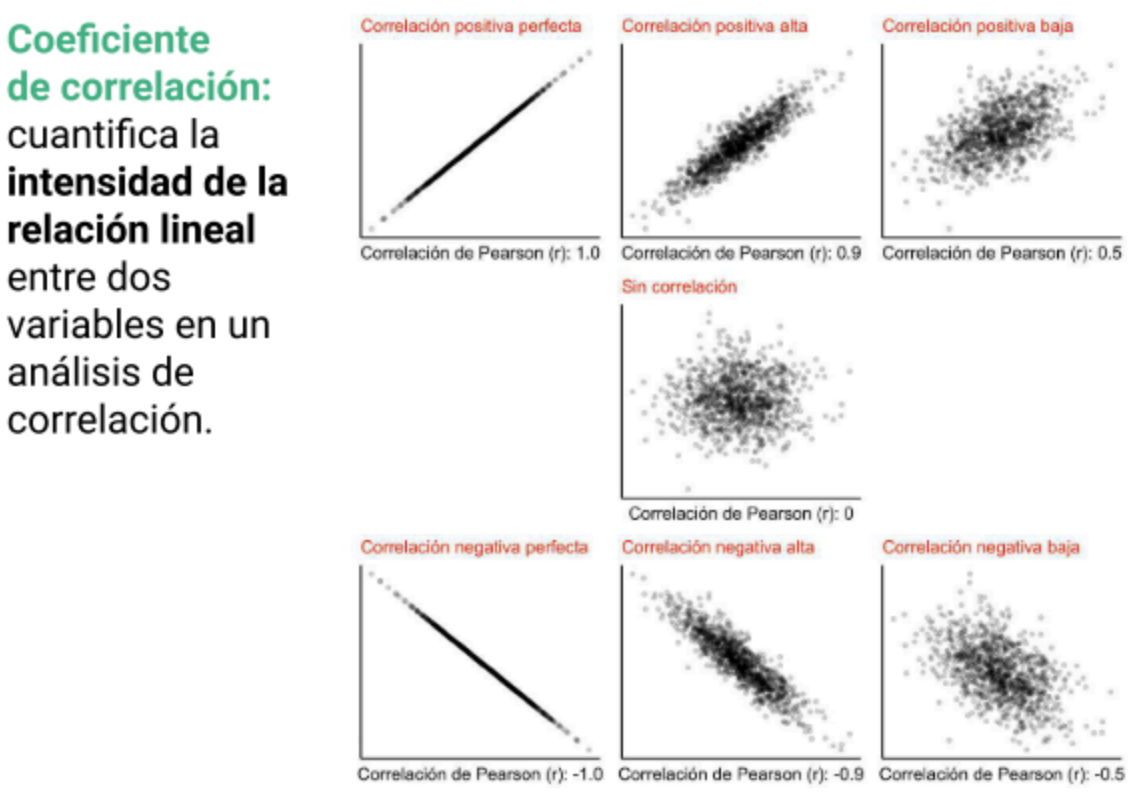
- longitud de ala
- peso de pingüino

Y la gráfica anterior parece describir que:

- Un aumento en la longitud de ala parece implicar un aumento en el peso del pingüino. Podría tener sentido de que si tienes mas masa por longitud, implica que peses más.

Realmente podemos decir que están correlacionadas, pero ¿qué tanto?

Coeficiente de correlación



Usamos un coeficiente de relación para ser exacto, el **coeficiente de Pearson**.

Este cuantifica la intensidad de una relación lineal, es muy importante hacer hincapié en que es **lineal**.

¿Cómo se interpreta el coeficiente?

Tenemos números que van de -1 a 1 y pasando por 0 (significa que no existe correlación). Para los demás casos:

- 1: Correlación positiva, significa que un cambio positivo en X, indica un cambio positivo en Y.
- -1: Correlación negativa, significa que un cambio positivo en X, indica un cambio negativo en Y.

Entonces los valores entre este rango indican un grado de correlación. Sin embargo este parámetro puede indicar si podemos continuar con el análisis de nuestra variable. Pero siempre tenemos que observar los datos para hacer un correcto análisis.

IMPORTANTE

Lo que quiere decir es que aunque pueda existir una correlación, ambos eventos son excluyentes y uno no es la causa del otro.

Esto es muy importante, porque no siempre podemos saber si una variable está causando el efecto de otra variable. Para determinarlo se necesita más información y más pruebas o experimentos, para poder comprobar que existe una causa a cierto efecto.

In [ ]:
```
# Importando librerías
import empiricaldist
import janitor
import matplotlib.pyplot as plt
import numpy as np
import palmerpenguins
import pandas as pd
import scipy.stats
import seaborn as sns
import sklearn.metrics
import statsmodels.api as sm
import statsmodels.formula.api as smf
import statsmodels.stats as ss
import session_info
```

Establecer apariencia general de las gráficas

In [ ]:
```
%matplotlib inline
sns.set_style(style='whitegrid')
sns.set_context(context='notebook')
plt.rcParams['figure.figsize'] = (11, 9.4)

penguin_color = {
    'Adelie': '#ff6602ff',
    'Gentoo': '#0f7175ff',
    'Chinstrap': '#c65dc9ff'
}
```

Cargar los datos

Datos Preprocesados

In [ ]:
```
preprocessed_penguins_df = pd.read_csv('dataset/penguins.csv')
```

Matrices de correlación

¿Existe una correlación lineal entre alguna de nuestras variables?

Para responder analicemos nuestros datos.

In [ ]:
```
preprocessed_penguins_df.corr(numeric_only=True)
```

Out[ ]:

	bill_length_mm	bill_depth_mm	flipper_length_mm	body_mass_g	year
bill_length_mm	1.000000	-0.235053	0.656181	0.595110	0.054545
bill_depth_mm	-0.235053	1.000000	-0.583851	-0.471916	-0.060354
flipper_length_mm	0.656181	-0.583851	1.000000	0.871202	0.169675
body_mass_g	0.595110	-0.471916	0.871202	1.000000	0.042209
year	0.054545	-0.060354	0.169675	0.042209	1.000000

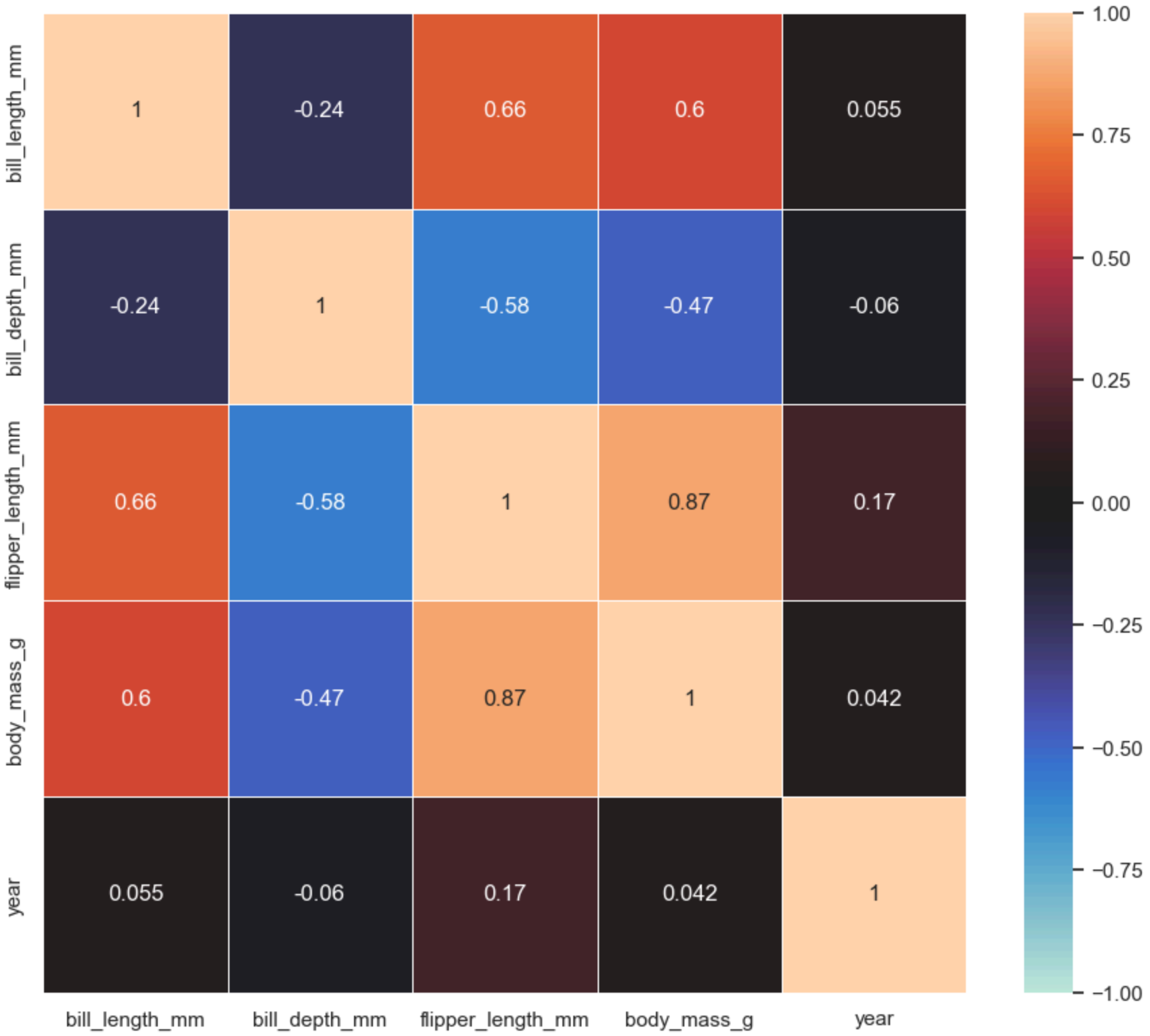
Estos son los coeficientes de correlación de Pearson de todas nuestras variables, en la diagonal vamos a ver que hay un 1, debido a que cada variable comparada consigo misma es lo mismo; por lo tanto tiene una correlación de 1.

¿Cómo puedo visualizar los coeficientes de correlación?

Seaborn nos provee una función accesible, rápida y visual.

In [ ]:
```
sns.heatmap(
    data=preprocessed_penguins_df.corr(numeric_only=True),
    center=0,
    vmin=-1,
    vmax=1,
    linewidths=0.5,
    annot=True
)
```

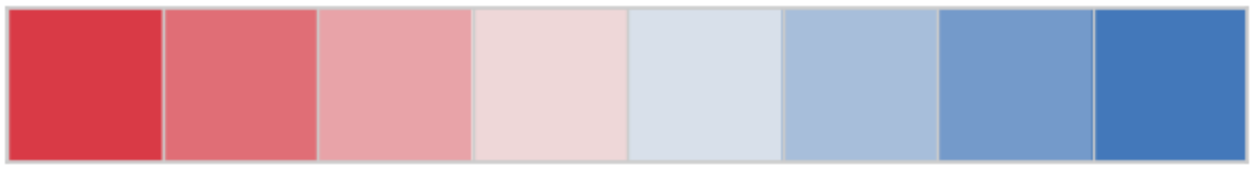
Out[ ]:
```
<AxesSubplot: >
```



Luce un poco raro por que la paleta de color puede no ser la adecuada.

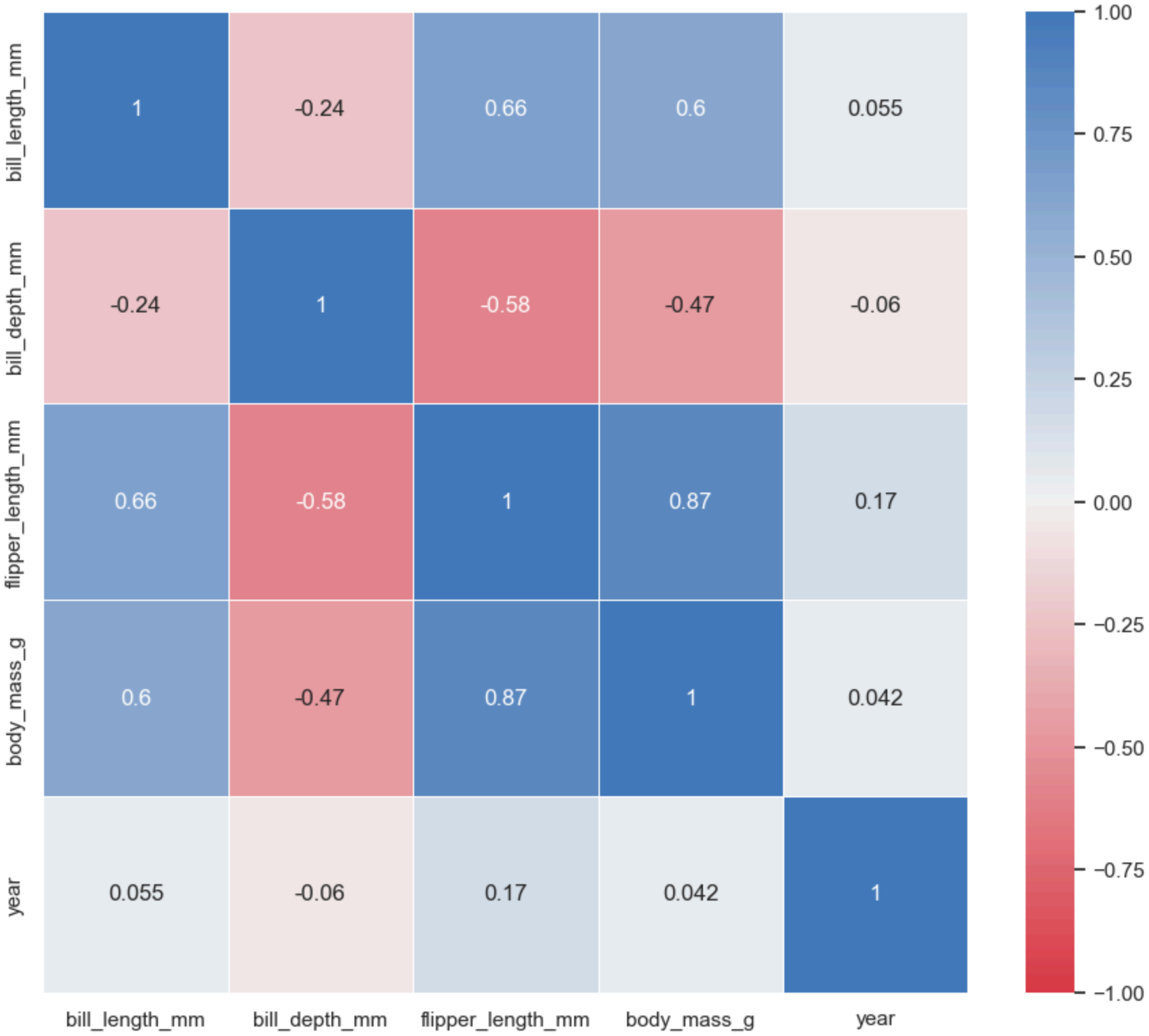
In [ ]:
```
# Create a diverging palette
palette = sns.diverging_palette(10,250, n=8)

# Display the palette
sns.palplot(palette)
plt.show()
```



In [ ]:
```
sns.heatmap(
    data=preprocessed_penguins_df.corr(numeric_only=True),
    center=0,
    vmin=-1,
    vmax=1,
    linewidths=0.5,
    annot=True,
    cmap=sns.diverging_palette(10,250, n=8, as_cmap=True)
)
```

Out[ ]: <AxesSubplot: >



La intensidad con la se muestran los colores, es la proporción de intensidad de la correlación.

Observaciones:

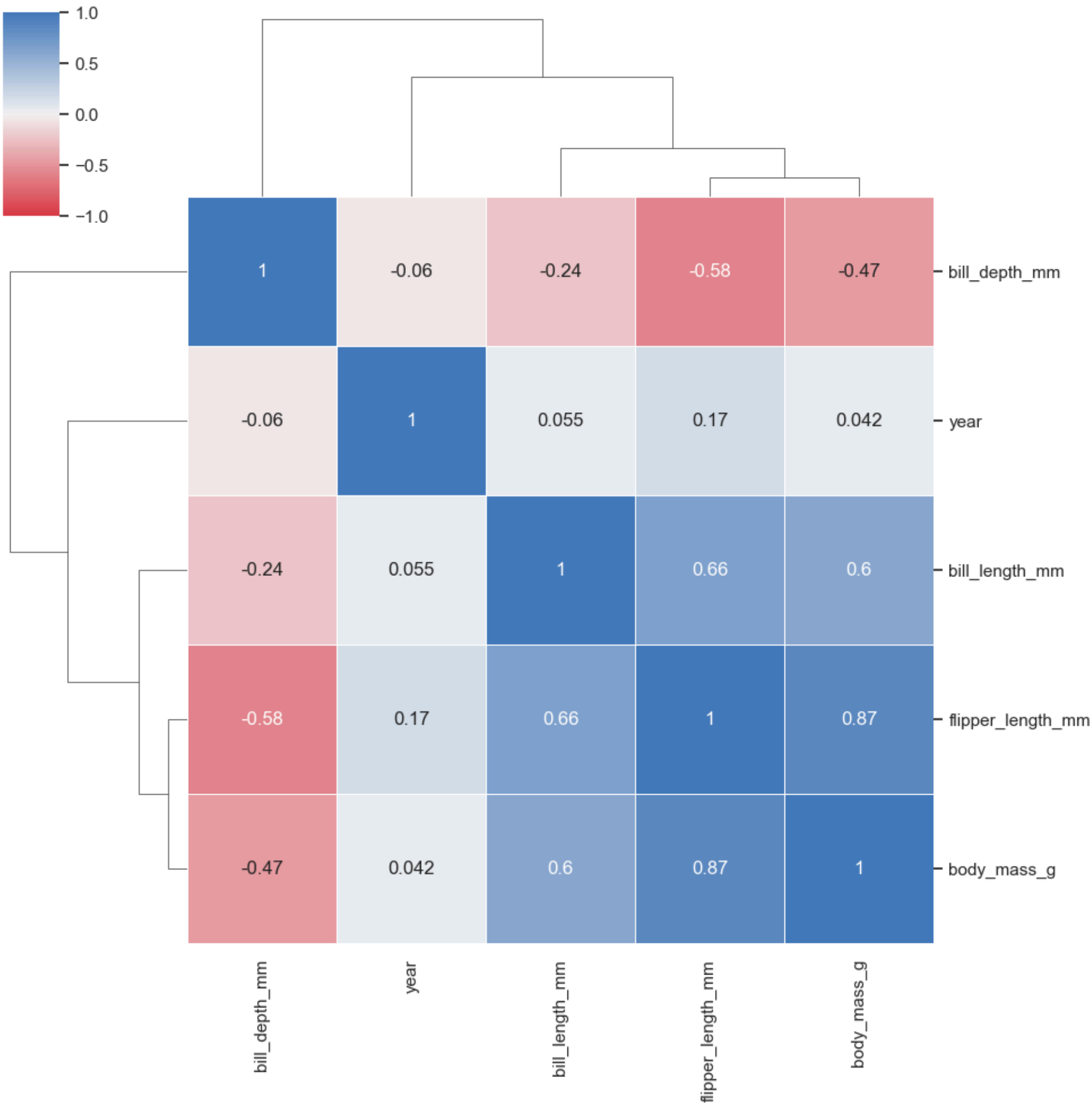
- La variable **Year** no está correlacionada con ninguna de nuestras variables, ya que tiene un valor cercano a 0.
- Las variables del peso y longitud de ala, tienen una correlación fuerte = 0.87.

Así de esta forma se puede observar si las variables tienen una correlación positiva o negativa.

También podemos cambiar la gráfica por un `cluster map`

In [ ]:
```
sns.clustermap(
    data=preprocessed_penguins_df.corr(numeric_only=True),
    center=0,
    vmin=-1,
    vmax=1,
    linewidths=0.5,
    annot=True,
    cmap=sns.diverging_palette(10,250, n=8, as_cmap=True)
)
```

Out[ ]:
```
<seaborn.matrix.ClusterGrid at 0x7f02bd481ba0>
```



Lo que hace `cluster_map` es agrupar nuestros datos, según el valor que tenga la matriz. De esta forma es más fácil correlacionar variables de manera positiva, negativa y por intensidad.

Sin embargo podemos añadir variables que no estamos considerando.

También nos interesaría saber si alguna de estas variables está correlacionada con el sexo.

¿Qué podríamos hacer?

Asignar o convertir una variable

In [ ]:
```
(
    preprocessed_penguins_df
    .assign(
        numeric_sex=lambda df: df.sex.replace(['female','male'],[0,1])
    )
)
```

Out[ ]:

	species	island	bill_length_mm	bill_depth_mm	flipper_length_mm	body_mass_g	sex	year	numeric_sex
0	Adelie	Torgersen	39.1	18.7	181.0	3750.0	male	2007	1.0
1	Adelie	Torgersen	39.5	17.4	186.0	3800.0	female	2007	0.0
2	Adelie	Torgersen	40.3	18.0	195.0	3250.0	female	2007	0.0
3	Adelie	Torgersen	NaN	NaN	NaN	NaN	NaN	2007	NaN
4	Adelie	Torgersen	36.7	19.3	193.0	3450.0	female	2007	0.0
...	...	...	...	...	...	...	...	...	...
339	Chinstrap	Dream	55.8	19.8	207.0	4000.0	male	2009	1.0
340	Chinstrap	Dream	43.5	18.1	202.0	3400.0	female	2009	0.0
341	Chinstrap	Dream	49.6	18.2	193.0	3775.0	male	2009	1.0
342	Chinstrap	Dream	50.8	19.0	210.0	4100.0	male	2009	1.0
343	Chinstrap	Dream	50.2	18.7	198.0	3775.0	female	2009	0.0

344 rows × 9 columns

Se nos crea una nueva variable para la cual tenemos, que la variable va a tener los valores de

- 1 macho
- 0 hembra

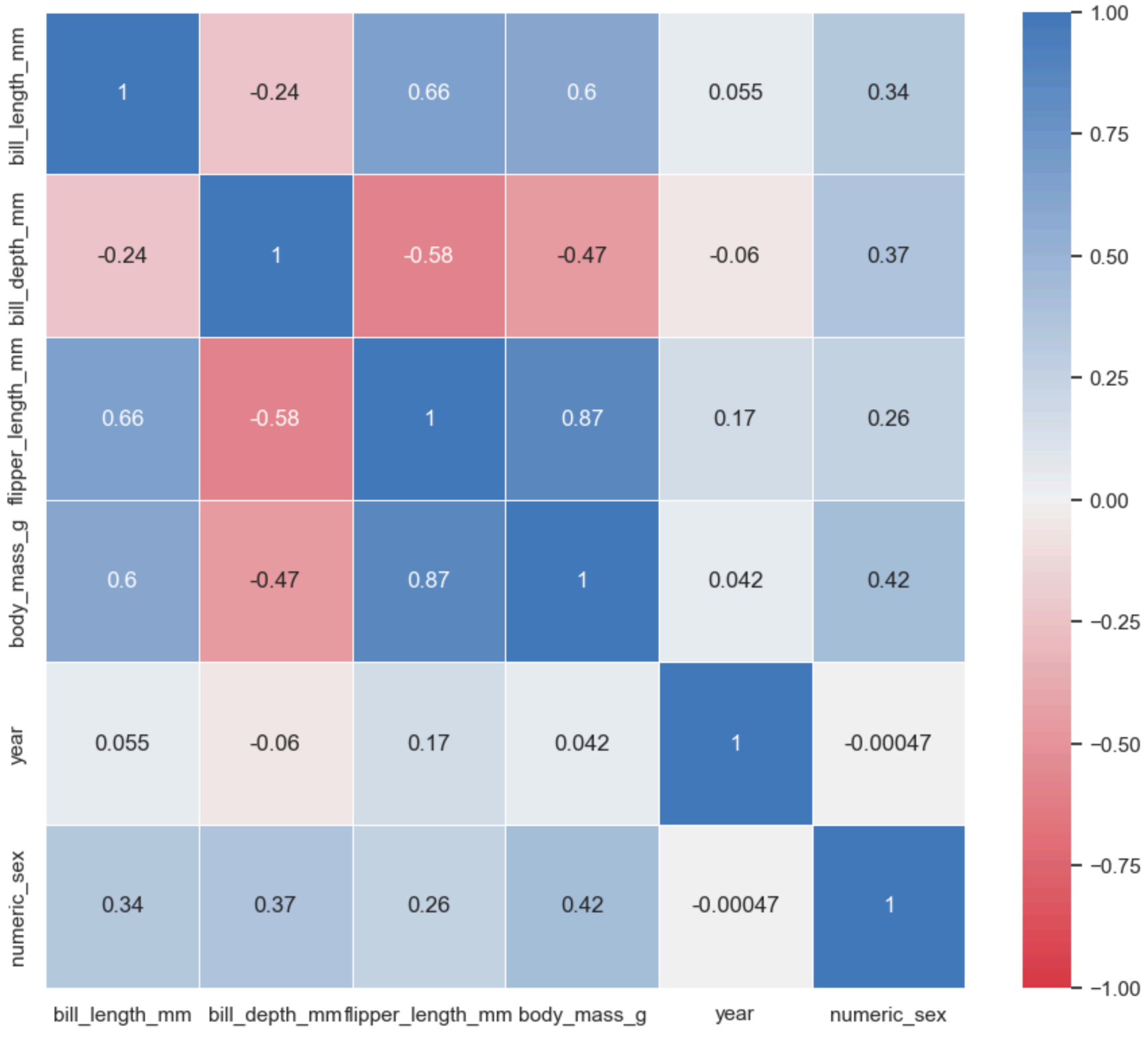
Agregando a nuestro conjunto de datos

In [ ]:
```
preprocessed_penguins_df =(
    preprocessed_penguins_df
    .assign(
        numeric_sex=lambda df: df.sex.replace(['male','female'],[0,1])
    )
)
```

Nuevamente podríamos realizar la comparación de las variables.

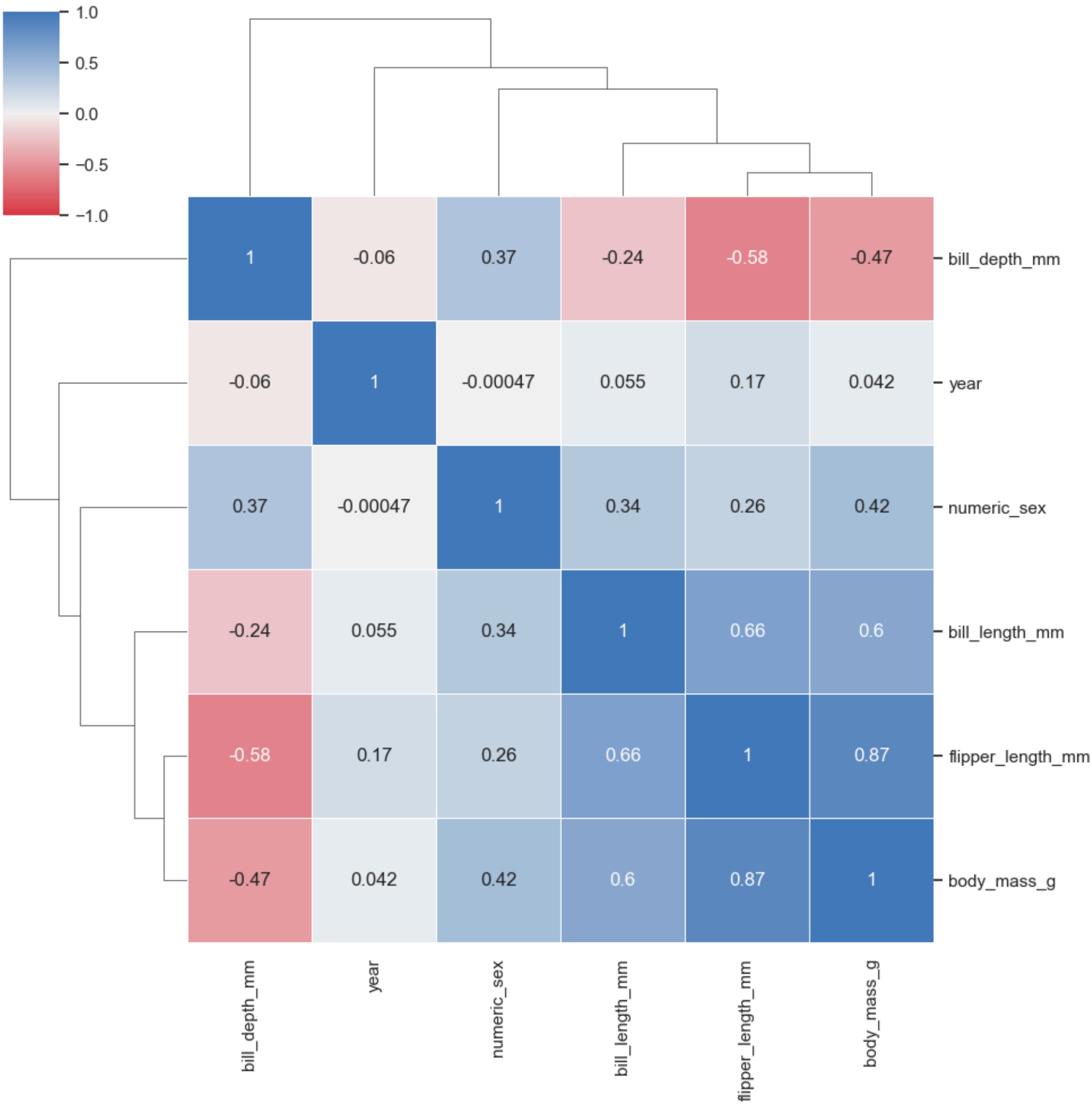
In [ ]:
```
sns.heatmap(
    data=preprocessed_penguins_df.corr(numeric_only=True),
    center=0,
    vmin=-1,
    vmax=1,
    linewidths=0.5,
    annot=True,
    cmap=sns.diverging_palette(10,250, n=8, as_cmap=True)
)
```

Out[ ]: <AxesSubplot: >



In [ ]:
```
sns.clustermap(
    data=preprocessed_penguins_df.corr(numeric_only=True),
    center=0,
    vmin=-1,
    vmax=1,
    linewidths=0.5,
    annot=True,
    cmap=sns.diverging_palette(10,250, n=8, as_cmap=True)
)
```

Out[ ]:
```
<seaborn.matrix.ClusterGrid at 0x7f02bd49a4a0>
```



Como se puede observar la variable está correlacionada de manera débil con las demás variables

¿Cuál es la limitante de los coeficientes de correlación?

Lo exploraremos en la siguiente clase

In [ ]:
```
for column in preprocessed_penguins_df.columns:
    print(column)
```

```
species
island
bill_length_mm
bill_depth_mm
flipper_length_mm
body_mass_g
sex
year
numeric_sex
```

In [ ]:
```
preprocessed_penguins_df['island'].unique()
```

Out[ ]:
```
array(['Torgersen', 'Biscoe', 'Dream'], dtype=object)
```

Vamos a añadir una variable numérica de `island`

In [ ]:
```
preprocessed_penguins_df =(
    preprocessed_penguins_df
    .assign(
        numeric_island=lambda df: df.island.replace(['Torgersen', 'Biscoe', 'Dream'],[0,1,2])
    )
)
```

In [ ]:
```
preprocessed_penguins_df
```

Out[ ]:

	species	island	bill_length_mm	bill_depth_mm	flipper_length_mm	body_mass_g	sex	year	numeric_sex	numeric_island
0	Adelie	Torgersen	39.1	18.7	181.0	3750.0	male	2007	1.0	0
1	Adelie	Torgersen	39.5	17.4	186.0	3800.0	female	2007	0.0	0
2	Adelie	Torgersen	40.3	18.0	195.0	3250.0	female	2007	0.0	0
3	Adelie	Torgersen	NaN	NaN	NaN	NaN	NaN	2007	NaN	0
4	Adelie	Torgersen	36.7	19.3	193.0	3450.0	female	2007	0.0	0
...	...	...	...	...	...	...	...	...	...	...
339	Chinstrap	Dream	55.8	19.8	207.0	4000.0	male	2009	1.0	2
340	Chinstrap	Dream	43.5	18.1	202.0	3400.0	female	2009	0.0	2
341	Chinstrap	Dream	49.6	18.2	193.0	3775.0	male	2009	1.0	2
342	Chinstrap	Dream	50.8	19.0	210.0	4100.0	male	2009	1.0	2
343	Chinstrap	Dream	50.2	18.7	198.0	3775.0	female	2009	0.0	2

344 rows × 10 columns

Vamos a añadir una variable numérica de `species`

In [ ]:
```
preprocessed_penguins_df['species'].unique()
```

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

In [ ]:
```
preprocessed_penguins_df =(
    preprocessed_penguins_df
    .assign(
        numeric_species=lambda df: df.species.replace(['Adelie', 'Gentoo', 'Chinstrap'],[0,1,2])
    )
)
```

In [ ]:
```
preprocessed_penguins_df
```

Out[ ]:

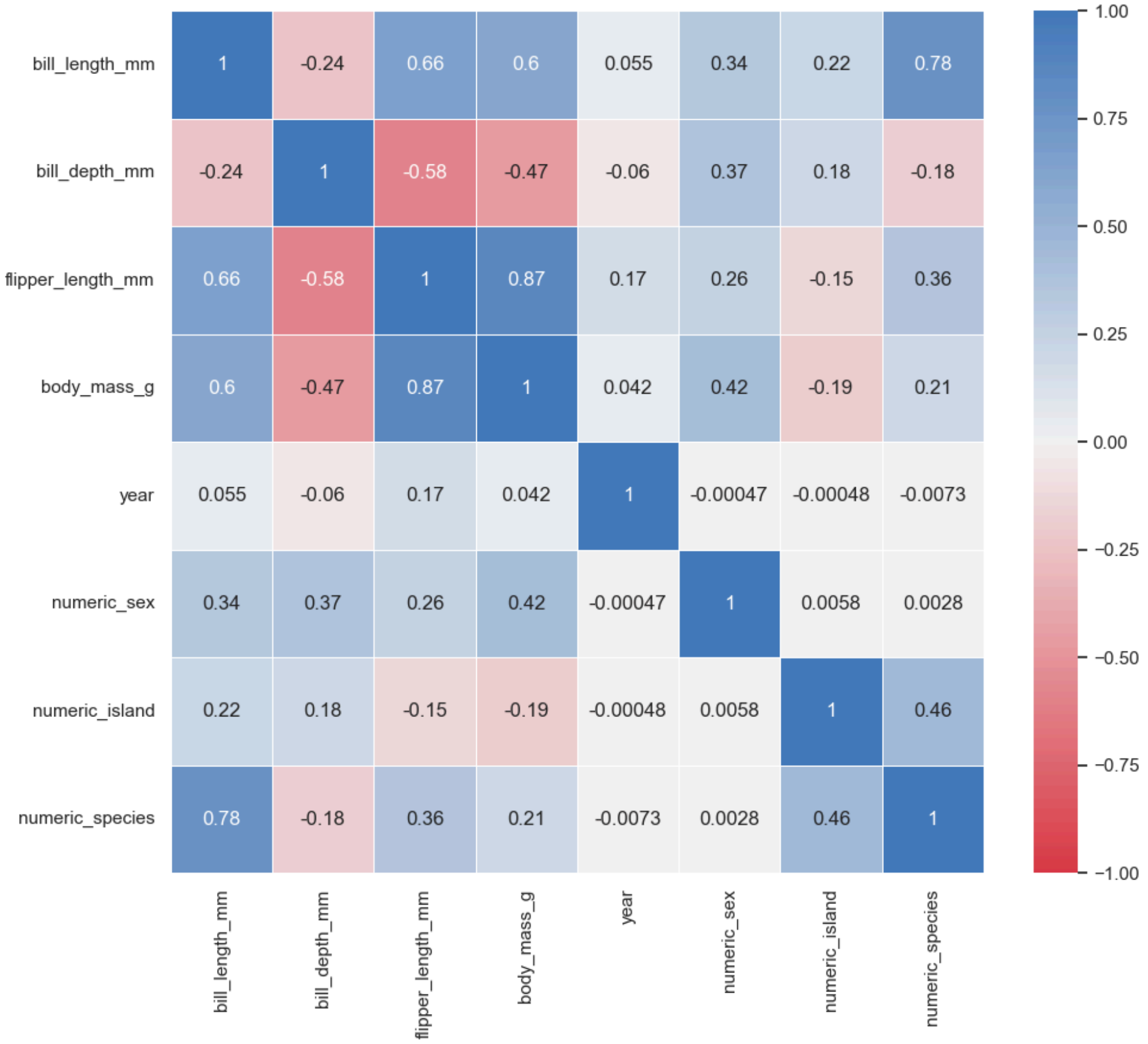
	species	island	bill_length_mm	bill_depth_mm	flipper_length_mm	body_mass_g	sex	year	numeric_sex	numeric_island	numer
0	Adelie	Torgersen	39.1	18.7	181.0	3750.0	male	2007	1.0	0	
1	Adelie	Torgersen	39.5	17.4	186.0	3800.0	female	2007	0.0	0	
2	Adelie	Torgersen	40.3	18.0	195.0	3250.0	female	2007	0.0	0	
3	Adelie	Torgersen	NaN	NaN	NaN	NaN	NaN	2007	NaN	0	
4	Adelie	Torgersen	36.7	19.3	193.0	3450.0	female	2007	0.0	0	
...	...	...	...	...	...	...	...	...	...	...	
339	Chinstrap	Dream	55.8	19.8	207.0	4000.0	male	2009	1.0	2	
340	Chinstrap	Dream	43.5	18.1	202.0	3400.0	female	2009	0.0	2	
341	Chinstrap	Dream	49.6	18.2	193.0	3775.0	male	2009	1.0	2	
342	Chinstrap	Dream	50.8	19.0	210.0	4100.0	male	2009	1.0	2	
343	Chinstrap	Dream	50.2	18.7	198.0	3775.0	female	2009	0.0	2	

344 rows × 11 columns

Graficando las relaciones

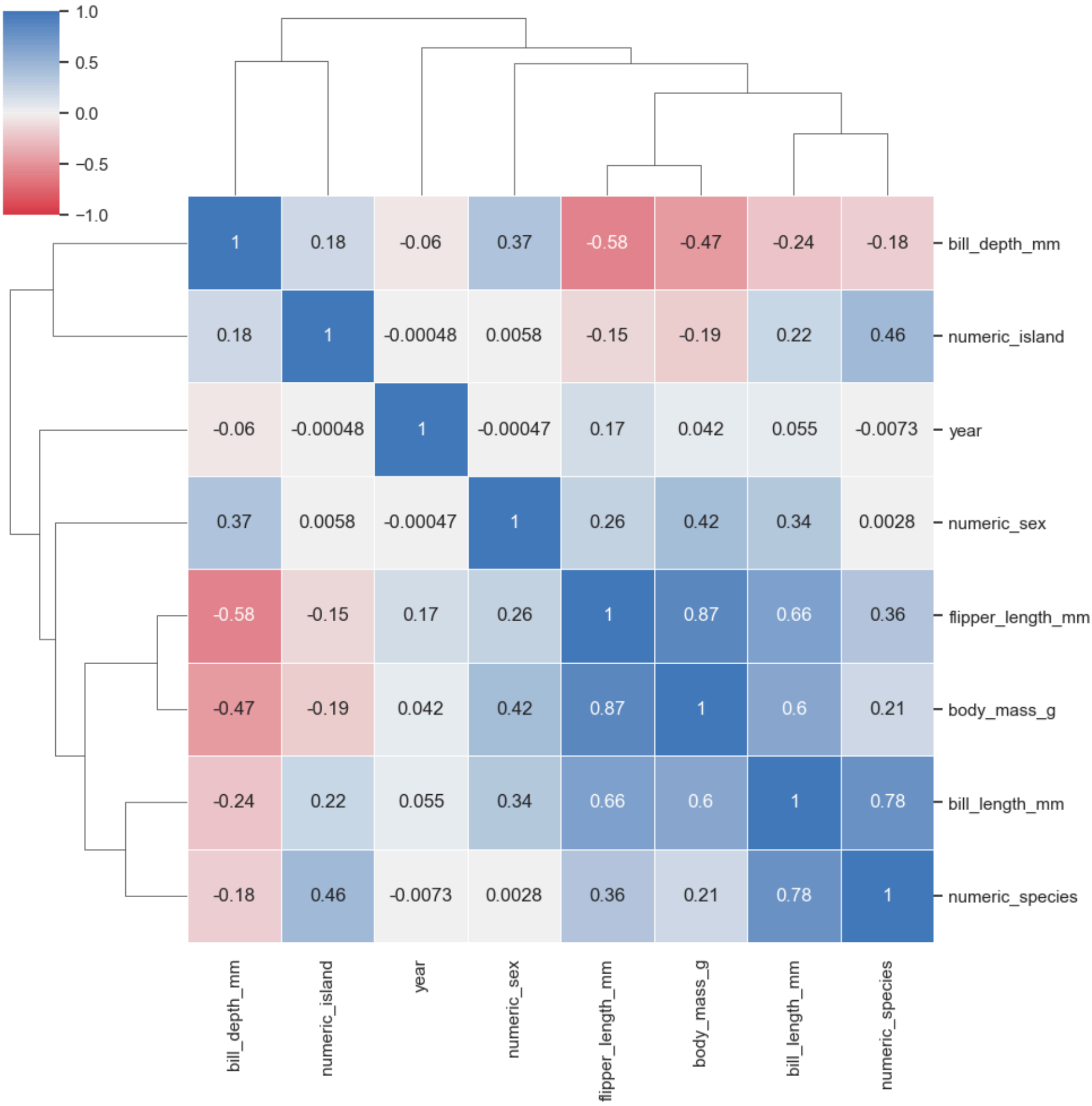
In [ ]:
```
sns.heatmap(
    data=preprocessed_penguins_df.corr(numeric_only=True),
    center=0,
    vmin=-1,
    vmax=1,
    linewidths=0.5,
    annot=True,
    cmap=sns.diverging_palette(10,250, n=8, as_cmap=True)
)
```

Out[ ]:
```
<AxesSubplot: >
```



In [ ]:
```
sns.clustermap(
    data=preprocessed_penguins_df.corr(numeric_only=True),
    center=0,
    vmin=-1,
    vmax=1,
    linewidths=0.5,
    annot=True,
    cmap=sns.diverging_palette(10,250, n=8, as_cmap=True)
)
```

Out[ ]: <seaborn.matrix.ClusterGrid at 0x7f02b4edf400>



In [ ]:
```
preprocessed_penguins_df.columns
```

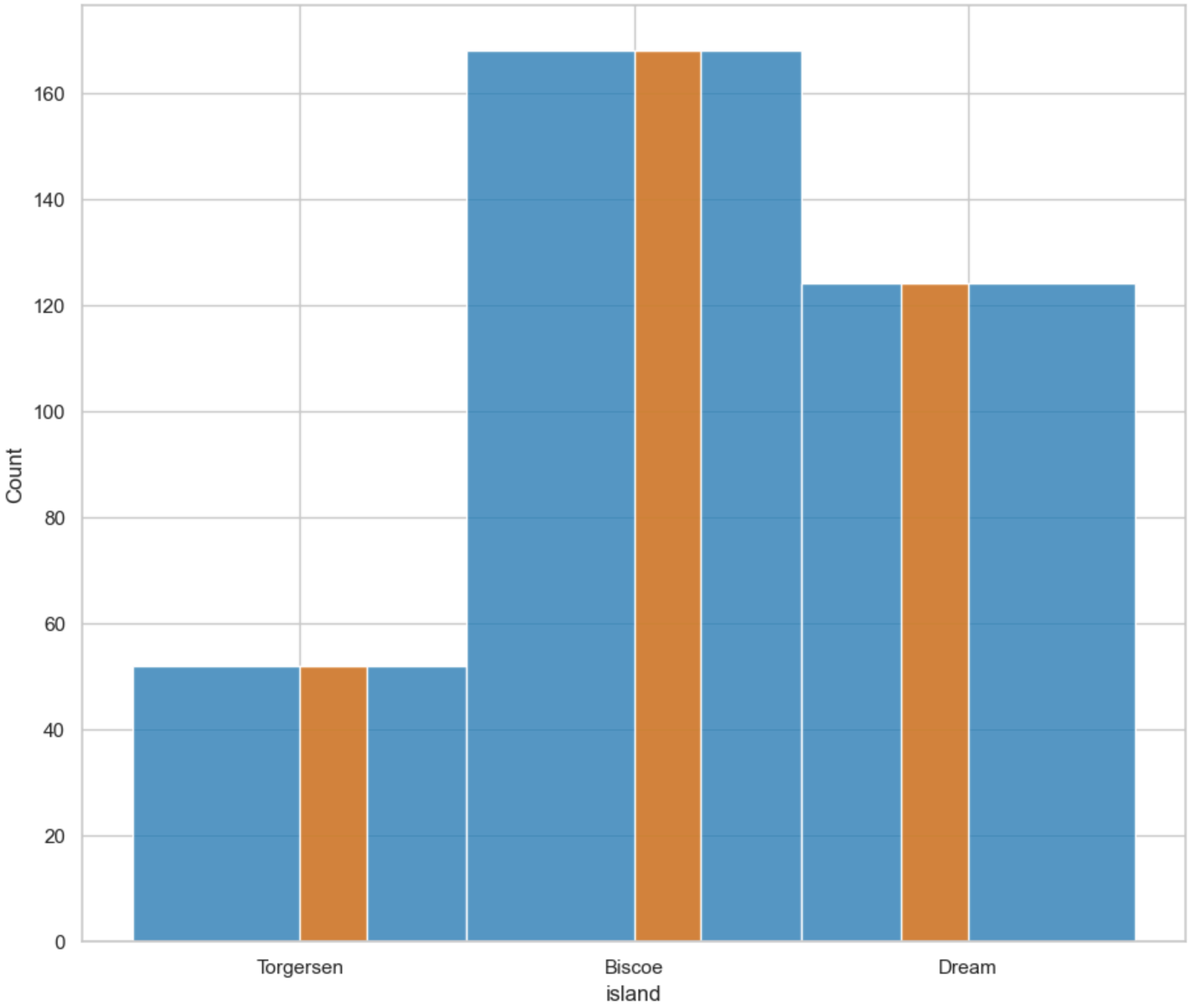
Out[ ]:
```
Index(['species', 'island', 'bill_length_mm', 'bill_depth_mm',
       'flipper_length_mm', 'body_mass_g', 'sex', 'year', 'numeric_sex',
       'numeric_island', 'numeric_species'],
      dtype='object')
```

In [ ]:
```
sns.histplot(
    data=preprocessed_penguins_df,
    x='island',

)
sns.histplot(
    data=preprocessed_penguins_df,
    x='numeric_island',

)
```

Out[ ]:
```
<AxesSubplot: xlabel='island', ylabel='Count'>
```

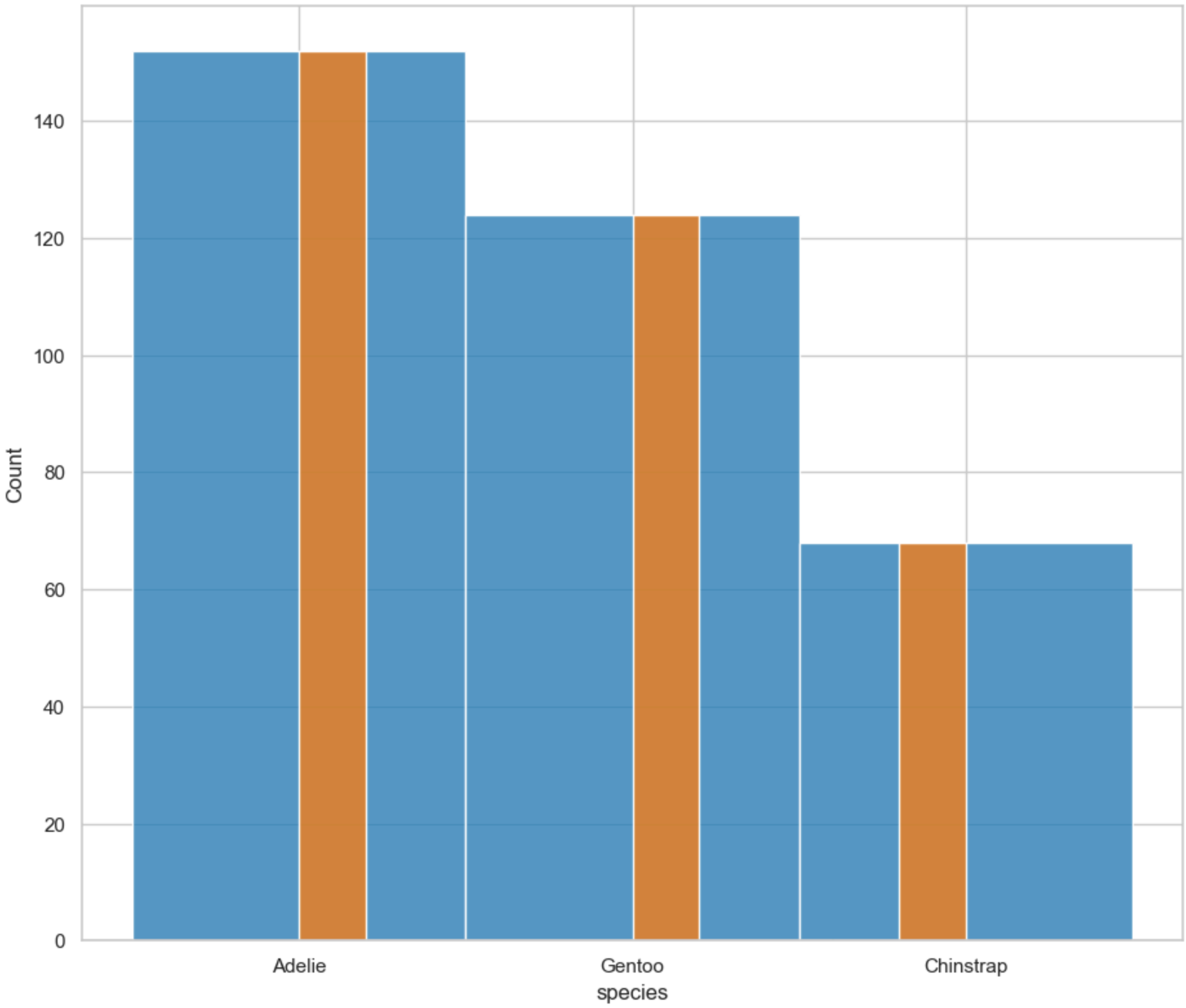


In [ ]:
```
sns.histplot(
    data=preprocessed_penguins_df,
    x='species',

)
sns.histplot(
    data=preprocessed_penguins_df,
    x='numeric_species',

)
```

Out[ ]: <AxesSubplot: xlabel='species', ylabel='Count'>

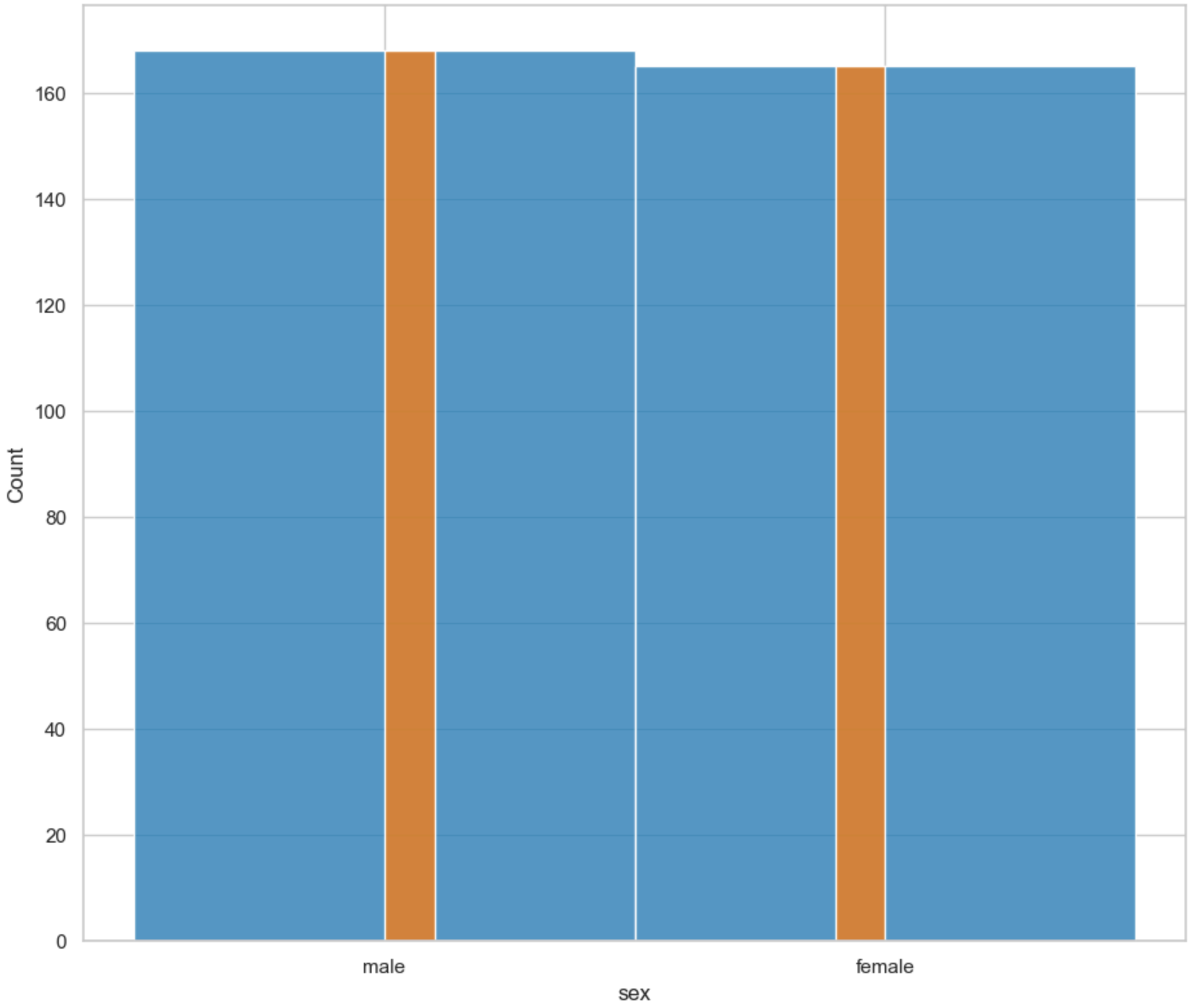


In [ ]:
```
sns.histplot(
    data=preprocessed_penguins_df,
    x='sex',

)
sns.histplot(
    data=preprocessed_penguins_df,
    x='numeric_sex',

)
```

Out[ ]: <AxesSubplot: xlabel='sex', ylabel='Count'>


160
140
120
100
80
60
40
20
0
Count
male
female
sex