ANALIZA CECH MORFOLOGICZNYCH PINGWINÓW NA ANTARKTYDZIE NA PODSTAWIE DATASETU SEABORN PENGUINS

PATRYCJA ŁĄCZKA,
CIĄG WI,
NR ALBUMU 35252

SPIS TREŚCI

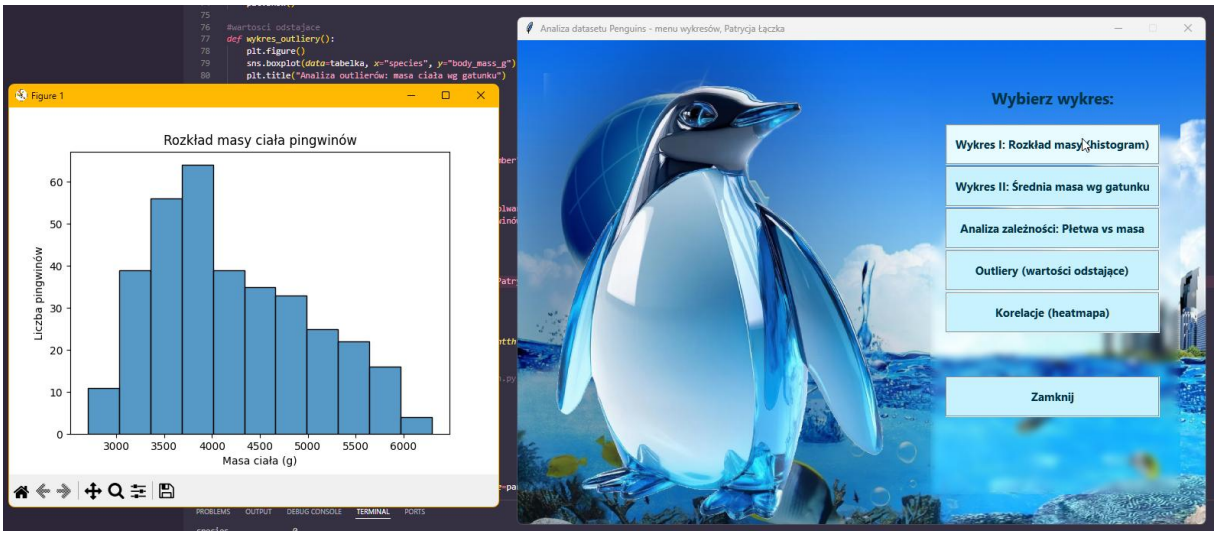
WYKORZYSTANE BIBLIOTEKI

W projekcie użyto następujących bibliotek:

- seaborn - do wczytania datasetu *penguins* oraz do tworzenia wykresów statystycznych,
- matplotlib.pyplot - do wyświetlania i formatowania wykresów,
- tkinter - do stworzenia graficznego interfejsu użytkownika,

IMPLEMENTACJA GRAFICZNEGO INTERFEJSU UŻYTKOWNIKA

Początkowo kod nie zawierał graficznego interfejsu, po uruchomieniu wykresy były wyświetlane jeden po drugim. Takie rozwiązanie było mało wygodne w użytkowaniu, dlatego zdecydowałam się na dodanie graficznego interfejsu użytkownika w celu poprawienia komfortu przeglądania wyników analizy danych. W obecnej wersji program po uruchomieniu otwiera jedno główne okno pełniące rolę menu, a następnie czeka na akcję użytkownika. Dopiero po kliknięciu odpowiedniego przycisku wyświetlany jest wybrany wykres w nowym oknie. Użytkownik może otworzyć jednocześnie kilka wykresów i swobodnie się między nimi przełączać. Interfejs został wykonany w estetyce inspirowanej stylem *frutiger aero*.



Do stworzenia interfejsu wykorzystano bibliotekę Tkinter:

```
import tkinter as tk
```

Po przebudowie programu każdy wykres został umieszczony w osobnej funkcji, dzięki czemu może być wywoływany niezależnie z poziomu menu.

Główne okno aplikacji tworzę za pomocą następującego kodu:

```
root = tk.Tk()
root.title("Analiza datasetu Penguins – menu wykresów")
root.geometry("900x633")
root.resizable(False, False)
```

Ustawia on tytuł okna, jego rozmiar oraz blokuje możliwość zmiany wymiarów, aby zachować poprawne rozmieszczenie elementów graficznych.

Jako główną powierzchnię interfejsu wykorzystano komponent Canvas, na którym rysowane jest tło aplikacji oraz panel boczny. Obrazy są wczytywane z plików graficznych:

```
bg_img = tk.PhotoImage(file="tlo.png")
panel_img = tk.PhotoImage(file="panel.png")
```

Tło oraz panel są następnie renderowane na canvasie, a na panelu umieszczane są przyciski nawigacyjne.

Przyciski tworzone są za pomocą funkcji pomocniczej aero_button, która odpowiada za ich wygląd (kolory, obramowanie, efekt podświetlenia po najechaniu kursorem) oraz za przypisanie im odpowiednich akcji. Przykładowe wywołanie takiego przycisku wygląda następująco:

```
aero_button("Wykres I: Rozkład masy (histogram)", wykres_histogram,
start_x, start_y + gap*0)
```

Po kliknięciu tego przycisku uruchamiana jest funkcja wykres_histogram(), która generuje odpowiedni wykres.

Na końcu programu zastosowano pętlę główną interfejsu:

```
root.mainloop()
```

Dzięki niej aplikacja nie zamyka się po wykonaniu jednej operacji, lecz pozostaje aktywna i reaguje na kolejne kliknięcia użytkownika.

Zastosowanie graficznego menu znacząco poprawiło ergonomię programu oraz czytelność prezentowanych wyników analizy danych.

OPIS STRUKTURY DANYCH

Na początku wczytano dataset penguins z biblioteki seaborn.

```
tabelka = sns.load_dataset("penguins")
```

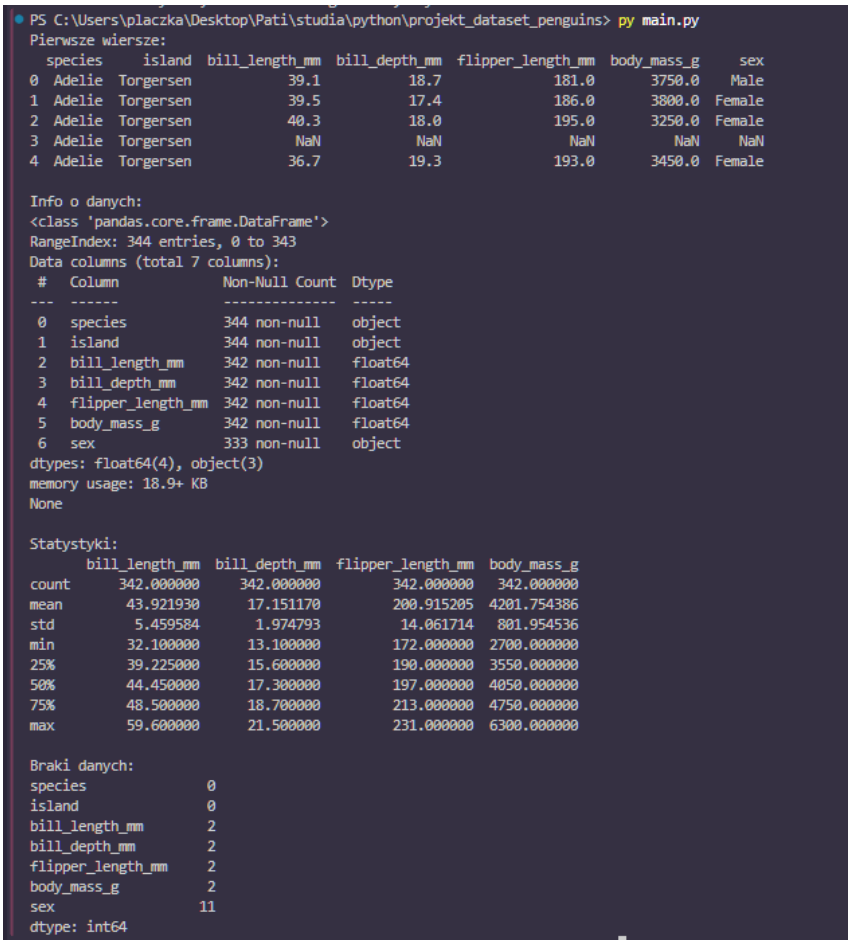
Za pomocą metod head() i info() zapoznano się ze strukturą danych.

Dataset składa się z 344 rekordów oraz 7 kolumn. Zawiera zarówno dane liczbowe (m.in. długość dzioba, głębokość dzioba, długość płetwy, masa ciała), jak i dane kategoryczne (gatunek, wyspa, płeć).

Za pomocą metody isnull().sum() sprawdzono liczbę brakujących wartości w poszczególnych kolumnach. Wykazano, że w zbiorze danych występują braki m.in. w kolumnach bill_length_mm, bill_depth_mm, flipper_length_mm, body_mass_g oraz sex.

Metoda describe() została użyta do obliczenia podstawowych statystyk opisowych, takich jak średnia, minimum, maksimum i odchylenie standardowe.

- count – ile jest wartości (nie licząc braków)
- mean – średnia
- std – odchylenie standardowe
- min – najmniejsza wartość
- 25% - pierwszy kwartyl
- 50% - mediana
- 75% - trzeci kwartyl
- Max – największa wartość



```
PS C:\Users\placzka\Desktop\Pati\studia\python\projekt_dataset_penguins> py main.py
Pierwsze wiersze:
  species     island  bill_length_mm  bill_depth_mm  flipper_length_mm  body_mass_g     sex
0  Adelie  Torgersen            39.1           18.7              181.0       3750.0    Male
1  Adelie  Torgersen            39.5           17.4              186.0       3800.0  Female
2  Adelie  Torgersen            40.3           18.0              195.0       3250.0  Female
3  Adelie  Torgersen             NaN            NaN                NaN          NaN     NaN
4  Adelie  Torgersen            36.7           19.3              193.0       3450.0  Female

Info o danych:
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 344 entries, 0 to 343
Data columns (total 7 columns):
 #   Column             Non-Null Count  Dtype
---  ------             --------------  -----
 0   species            344 non-null    object
 1   island             344 non-null    object
 2   bill_length_mm     342 non-null    float64
 3   bill_depth_mm      342 non-null    float64
 4   flipper_length_mm  342 non-null    float64
 5   body_mass_g        342 non-null    float64
 6   sex                333 non-null    object
dtypes: float64(4), object(3)
memory usage: 18.9+ KB
None

Statystyki:
       bill_length_mm  bill_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

Braki danych:
species              0
island               0
bill_length_mm       2
bill_depth_mm        2
flipper_length_mm    2
body_mass_g          2
sex                 11
dtype: int64
```

Na podstawie przeprowadzonej analizy wstępnej stwierdzono, że dane wymagają oczyszczenia przed dalszą analizą.

CZYSZCZENIE DANYCH

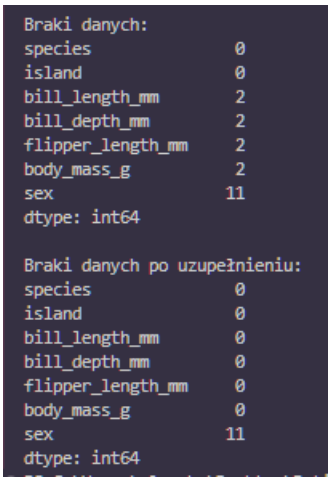
W celu zachowania wszystkich obserwacji zdecydowano się nie usuwać rekordów z brakującymi danymi, lecz uzupełnić brakujące wartości w kolumnach liczbowych za pomocą średniej arytmetycznej danej cechy

Przykład:

```
tabelka["body_mass_g"] =
tabelka["body_mass_g"].fillna(tabelka["body_mass_g"].mean())
```

„W kolumnie body_mass_g wstaw w miejsca NaN średnią z tej kolumny”

Po wykonaniu tej operacji ponownie sprawdzono liczbę brakujących danych, potwierdzając poprawność procesu czyszczenia danych.



W przypadku kolumny sex, która nie została uzupełniona średnią arytmetyczną rozważałam trzy podejścia na oczyszczenie danych:

1. Usunięcie pustych pozycji
2. Uzupełnienie „po równo”
3. Uzupełnienie najczęstszą wartością (Imputacja modą (most frequent value imputation))

Uzupełnianie brakujących wartości „po równo” (sztuczne wyrównywanie liczby obserwacji w każdej kategorii) zostałoby w praktyce wprowadzeniem do zbioru danych informacji, które nie wynikają z danych źródłowych. Takie podejście prowadziłoby do zafałszowania rozkładu cechy sex oraz mogłoby wpłynąć na dalsze wyniki analizy.

Z kolei imputacja najczęściej występującą wartością (modą) minimalnie ingeruje w strukturę danych i opiera się na najbardziej prawdopodobnym założeniu statystycznym, że brakująca wartość należy do najliczniejszej kategorii. Z tego powodu to podejście zostało uznane za bardziej poprawne analitycznie.

MOST FREQUENT VALUE IMPUTATION

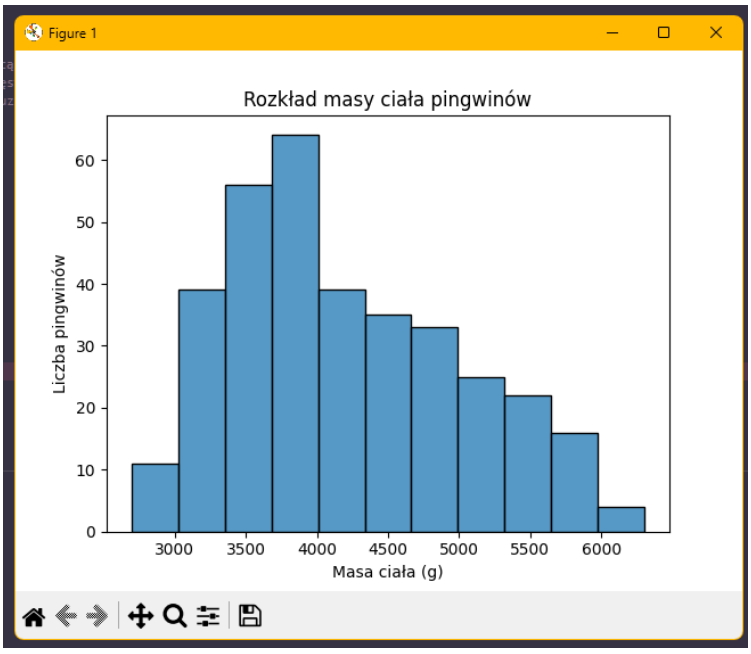
```
najczestsza_plec = tabelka["sex"].mode()[0]   # znajdź najczęstszą wartość
tabelka["sex"] = tabelka["sex"].fillna(najczestsza_plec)  # uzupełnij braki tą wartością
```

zwraca najczęściej występującą wartość w kolumnie sex, bierze pierwszą wartość i wstawia ją we wszystkie puste miejsca.

```
Braki danych:
species              0
island               0
bill_length_mm       2
bill_depth_mm        2
flipper_length_mm    2
body_mass_g          2
sex                 11
dtype: int64

Braki danych po uzupełnieniu:
species              0
island               0
bill_length_mm       0
bill_depth_mm        0
flipper_length_mm    0
body_mass_g          0
sex                 11
dtype: int64

Braki danych po uzupełnieniu kolumny 'sex':
species              0
island               0
bill_length_mm       0
bill_depth_mm        0
flipper_length_mm    0
body_mass_g          0
sex                  0
dtype: int64
```

WYKRESY ROZKŁADU

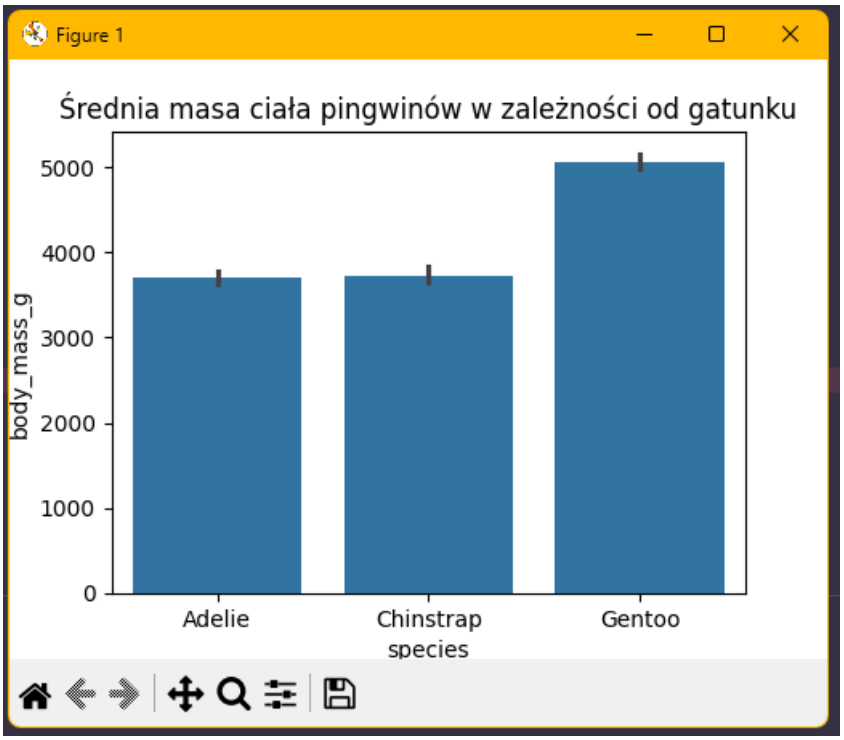
Wykres I: Rozkład masy pingwinów



Dany kod bierze kolumnę body_mass_g, następnie dzieli wartość na przedziały, oraz liczy ile pingwinów wpada do każdego przedziału

```
sns.histplot(tabelka["body_mass_g"])
plt.title("Rozkład masy ciała pingwinów")
plt.xlabel("Masa ciała (g)")
plt.ylabel("Liczba pingwinów")
plt.show()
```

Wykres II: Średnia masa ciała dla każdego gatunku

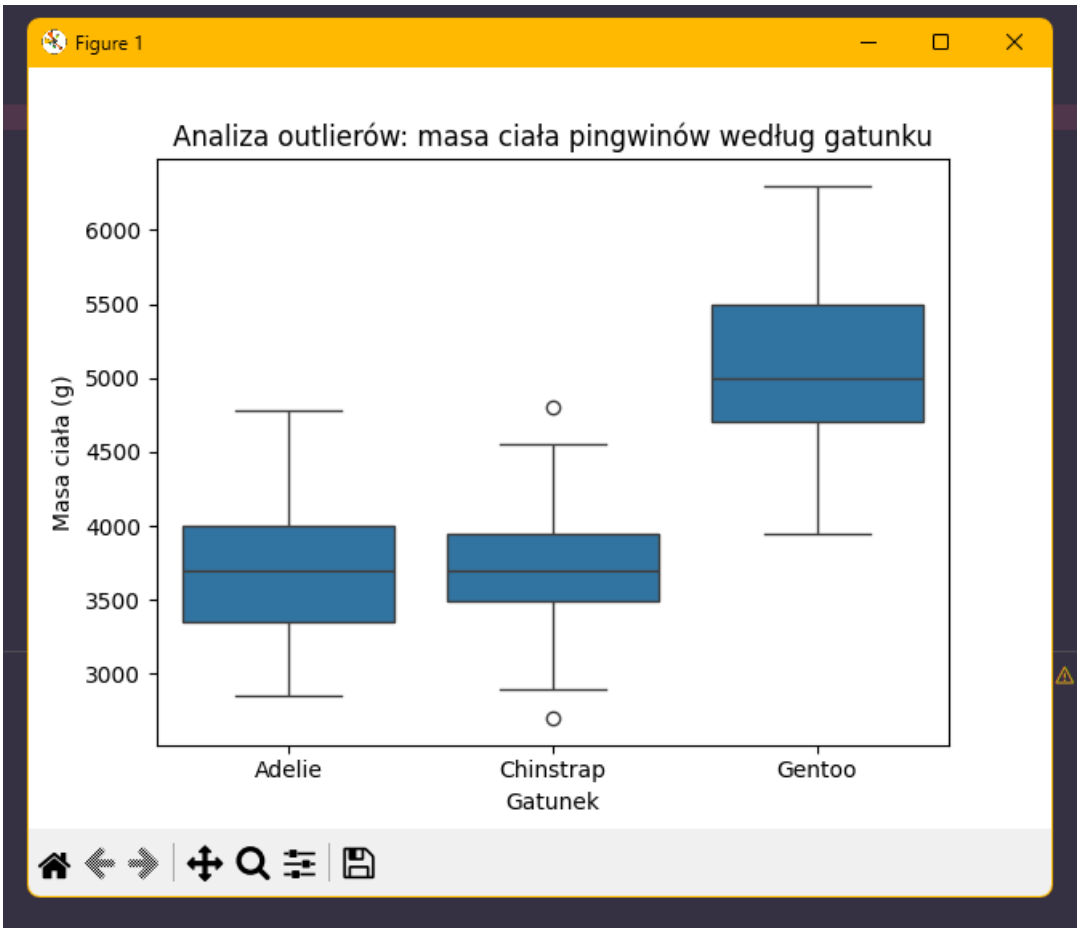


Na podstawie wykresu można stwierdzić, że gatunek ma wyraźny wpływ na masę ciała pingwinów. Pingwiny Adelie są najlżejszym gatunkiem spośród analizowanych Dany kod grupuje dane po gatunku, oraz dla każdego gatunku liczy średnią masę ciała. Następnie rysuje słupki z otrzymanych średnich.

```
sns.barplot(data=tabelka, x="species", y="body_mass_g")
plt.title("Średnia masa ciała pingwinów w zależności od gatunku")
plt.show()
```

ANALIZA OUTLIERÓW

Outliery mogą być błędem pomiaru, błędem w danych, lub rzadkim, ale prawdziwym przypadkiem. Potrafią bardzo mocno zaburzać średnie, wykresy i wnioski.



Linia w środku pudełka to mediana, czyli połowa pingwinów waży mniej, połowa więcej. Dolna krawędź pudełka to 25 percentyl, czyli 25% pingwinów wazy mniej niż ta wartość. Górna krawędź pudełka to 75 percentyl. Całe pudełko to 50% wszystkich danych. Poziome linie na dole i górze, pokazują normalny zakres danych. Kropki poza zakresem są to outliery – wartości odstające.

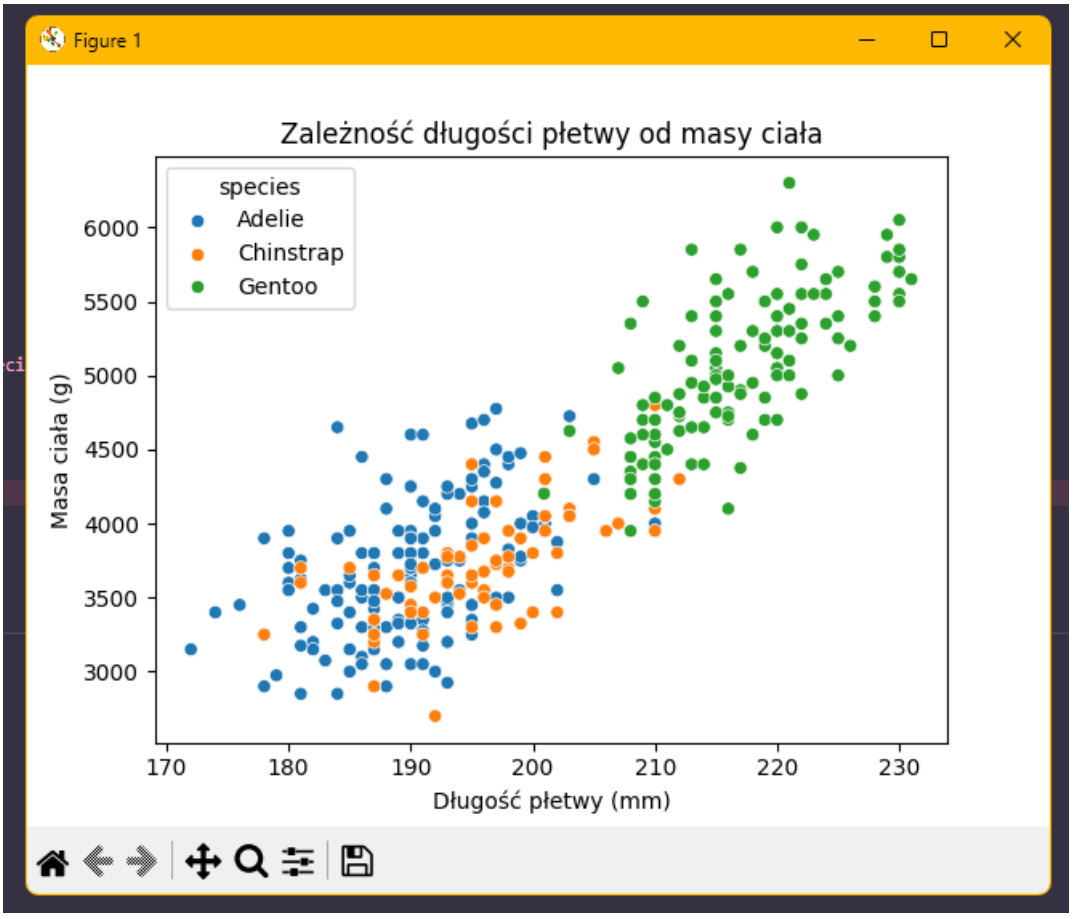
Na wykresie widać, że u pingwinów gatunku Chinstrap widać dwa bardzo wyraźne outliery jeden bardzo lekki pingwin (ok. 2700g), oraz jeden bardzo ciężki (ok. 4800g).

W celu analizy wartości odstających wykonano wykres pudełkowy (boxplot) przedstawiający rozkład masy ciała pingwinów w zależności od gatunku. Wykres został wygenerowany na podstawie danych zapisanych w strukturze tabelka, gdzie na osi X umieszczono kolumnę species, natomiast na osi Y kolumnę body_mass_g. Funkcja sns.boxplot() automatycznie oblicza medianę, kwartyle oraz zakres danych i umożliwia

wizualną identyfikację wartości odstających. Dodatkowo ustawiono tytuł wykresu oraz opisy osi w celu poprawy czytelności wizualizacji.

```
sns.boxplot(data=tabelka, x="species", y="body_mass_g")
plt.title("Analiza outlierów: masa ciała pingwinów według gatunku")
plt.xlabel("Gatunek")
plt.ylabel("Masa ciała (g)")
plt.show()
```

ANALIZA ZALEŻNOŚCI



Wykres punktowy pokazujący zależność między długością płetwy (flipper_length_mm), a masą ciała (body_mass_g). Na wykresie widoczny jest wyraźny trend rosnący, im dłuższa płetwa tym wieksza masa ciała ptaka.

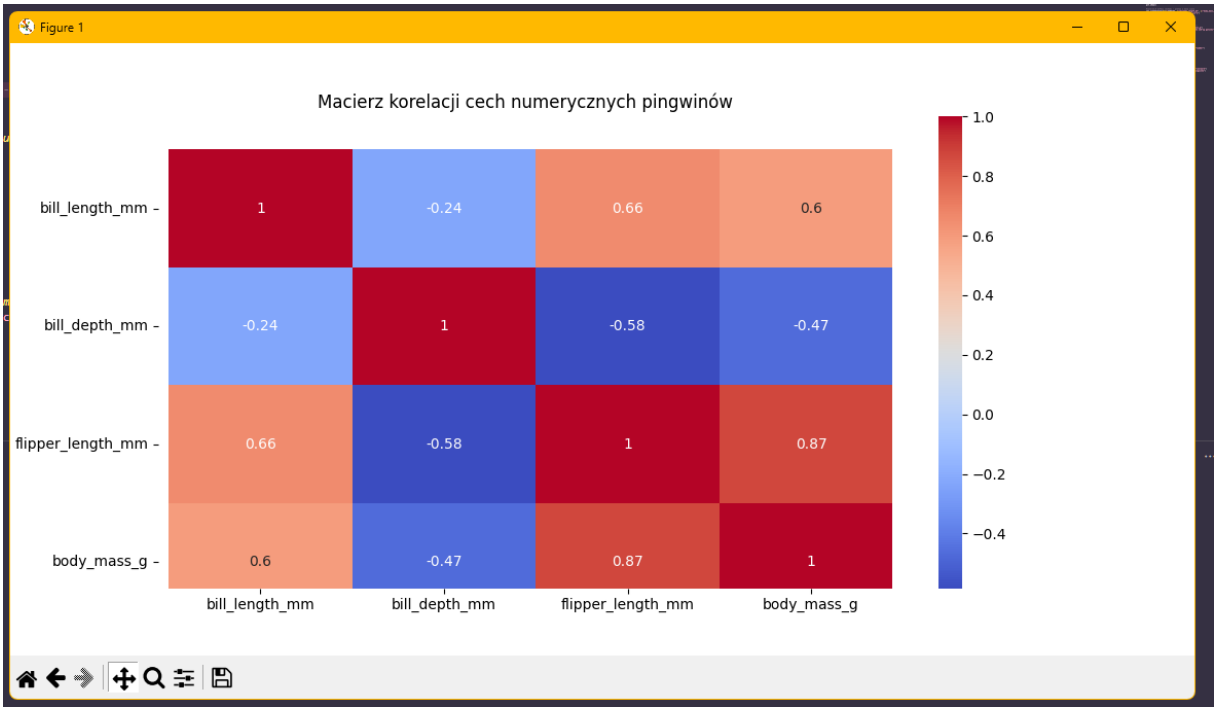
Dany kod bierze dane ze zmiennej „tabelka”, na oś X daje flipper_lengtth_mm, na oś Y daje body_mass_g, każdy punkt na wykresie odpowiada jednemu pingwinowi. Hue=”species” koloruje punkty według gatunku.

```
sns.scatterplot(data=tabelka, x="flipper_length_mm", y="body_mass_g", hue="species")

plt.title("Zależność długości płetwy od masy ciała")

plt.xlabel("Długość płetwy (mm)")

plt.ylabel("Masa ciała (g)")

plt.show()
```

KORELACJE

Korelacja między cechami liczbowymi pokazana na heatmapie. Każda liczba pokazuje jak silnie dwie cechy są ze sobą powiązane. Liczba 1 oznacza idealną zgodność, blisko 0 oznacza brak związku, blisko 1 oznacza silną dodatnią korelację, blisko -1 oznacza silną ujemną korelację.

Z wykresu możemy wywnioskować m.in. korelację między długością płetwy (flipper_length_mm), a masa ciała (body_mass_g) = 0.871. Im dłuższa płetwa, tym większa masa ciała. Jednocześnie jest to potwierdzenie wykresu punktowego.



```
Macierz korelacji:
                   bill_length_mm  bill_depth_mm  flipper_length_mm  body_mass_g
bill_length_mm           1.000000      -0.235053           0.656181     0.595110
bill_depth_mm           -0.235053       1.000000          -0.583851    -0.471916
flipper_length_mm        0.656181      -0.583851           1.000000     0.871202
body_mass_g              0.595110      -0.471916           0.871202     1.000000
```

Dany kod ze zmiennej tabelka wybiera tylko kolumny, które są liczbami. Kolejno liczy korelację każdej kolumny, z każdą inną kolumną. Powstaje macierz korelacji. Kolejno w konsoli wypisuje daną tabelkę korelacji, ustawia rozmiar wykresu, rysuje heatmapę, ustawia tytuł wykresu, oraz wyświetla wykres w oknie.

```
# Wybranie tylko kolumn liczbowych
dane_numeryczne = tabelka.select_dtypes(include="number")
# Obliczenie macierzy korelacji
macierz_korelacji = dane_numeryczne.corr()
print("\nMacierz korelacji:")
print(macierz_korelacji)
# Wizualizacja macierzy korelacji
plt.figure(figsize=(8, 6))
sns.heatmap(macierz_korelacji, annot=True, cmap="coolwarm")
plt.title("Macierz korelacji cech numerycznych pingwinów")
plt.show()
```

WNIOSKI

W ramach projektu przeprowadzono analizę danych dotyczących pingwinów. Dane zostały wstępnie oczyszczone poprzez uzupełnienie brakujących wartości, co pozwoliło na wykonanie poprawnych obliczeń statystycznych oraz wizualizacji.

Na podstawie wykonanych wykresów można zauważyć wyraźne różnice pomiędzy poszczególnymi gatunkami pingwinów. Gatunek Adelie charakteryzuje się najmniejszą średnią masą ciała, natomiast pozostałe gatunki osiągają wyraźnie większe wartości. Rozkład masy ciała pokazuje, że większość osobników mieści się w określonym przedziale, a pojedyncze obserwacje można uznać za wartości odstające.

Analiza zależności pomiędzy cechami wykazała, że istnieje silna dodatnia zależność pomiędzy długością płetwy a masą ciała - większe pingwiny posiadają zazwyczaj dłuższe płetwy. Potwierdza to również macierz korelacji, w której widoczna jest wysoka korelacja pomiędzy tymi zmiennymi.

Dodatkowo projekt pozwolił na praktyczne wykorzystanie narzędzi do analizy danych w języku Python oraz na stworzenie prostego interfejsu graficznego, który ułatwia przeglądanie wyników.