Análisis de regresión multiple

En clases pasadas aprendimos somo la regresión lineal y vimos sus limitaciones, entre ellas:

1. Es lineal y no se pueden incorporar comportamientos no lineales.

En la regresión multiple si puede hacer, ademas nos va a permitir analizar un poco más si la relación de variables afecta el comportamiento de nuestra variable objetivo.

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').dropna()
```

In [ ]:
```
preprocessed_penguins_df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 333 entries, 0 to 343
Data columns (total 8 columns):
 #   Column             Non-Null Count  Dtype  
---  ------             --------------  -----  
 0   species            333 non-null    object 
 1   island             333 non-null    object 
 2   bill_length_mm     333 non-null    float64
 3   bill_depth_mm      333 non-null    float64
 4   flipper_length_mm  333 non-null    float64
 5   body_mass_g        333 non-null    float64
 6   sex                333 non-null    object 
 7   year               333 non-null    int64  
dtypes: float64(4), int64(1), object(3)
memory usage: 23.4+ KB
```

Análisis de regresión multiple

Para ello, hagamos un experimento de tomar el rol de tomar medidas de los pingüinos y sus variables.

Digamos que en un día no llevas bascula y no puedes saber cuanto pesa un pingüino, así que tienes que tomar las demás medidas, pero nos falta saber cuanto pesa el pingüino.

¿Existe una manera que exista una asociación entre las variables y me permita determinar cuanto pesaba el pingüino? Vamos a averiguarlo usando la regresión multiple.

Creando modelos

Modelo 1

Usar una variable y compararla con otra. Anteriormente hicimos una regresión lineal simple en la cual usamos cierto modulo de `stats` y que nos servirá para la regresión multiple también.

In [ ]:
```
model_1 = (
    smf.ols(
```

```
        formula='body_mass_g ~ bill_length_mm',
        data=preprocessed_penguins_df
    )
    .fit()
)
model_1.summary()
```

Out[ ]:

OLS Regression Results

Dep. Variable:	body_mass_g	**R-squared:**	0.347
Model:	OLS	**Adj. R-squared:**	0.345
Method:	Least Squares	**F-statistic:**	176.2
Date:	Wed, 07 Aug 2024	**Prob (F-statistic):**	1.54e-32
Time:	18:48:52	**Log-Likelihood:**	-2629.1
No. Observations:	333	**AIC:**	5262.
Df Residuals:	331	**BIC:**	5270.
Df Model:	1		
Covariance Type:	nonrobust		

	coef	std err	t	P>|t|	[0.025	0.975]
Intercept	388.8452	289.817	1.342	0.181	-181.271	958.961
bill_length_mm	86.7918	6.538	13.276	0.000	73.931	99.652

Omnibus:	6.141	**Durbin-Watson:**	0.849
Prob(Omnibus):	0.046	**Jarque-Bera (JB):**	4.899
Skew:	-0.197	**Prob(JB):**	0.0864
Kurtosis:	2.555	**Cond. No.**	360.

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.

Me va a dar la descripción de la variable dependiente y las variables que vaya agregando a mi modelo.

Los parámetros que nos interesan se encuentran abajo, y nos interesa la pendiente (que se encuentra abajo de **intercept** como **bill_length_mm**).

¿Qué pasaría si quiero agregar una nueva variable o combinarla con la variable anteriormente analizada? Lo hacemos agregando el signo de $'+'$

Modelo 2

In [ ]:
```
model_2 = (
    smf.ols(
        formula='body_mass_g ~ bill_length_mm + bill_depth_mm',
        data=preprocessed_penguins_df
    )
    .fit()
)
model_2.summary()
```

Out[ ]:

OLS Regression Results

Dep. Variable:	body_mass_g	**R-squared:**	0.467
Model:	OLS	**Adj. R-squared:**	0.464
Method:	Least Squares	**F-statistic:**	144.8
Date:	Wed, 07 Aug 2024	**Prob (F-statistic):**	7.04e-46
Time:	20:15:49	**Log-Likelihood:**	-2595.2
No. Observations:	333	**AIC:**	5196.
Df Residuals:	330	**BIC:**	5208.
Df Model:	2		
Covariance Type:	nonrobust		

	coef	std err	t	P>|t|	[0.025	0.975]
Intercept	3413.4519	437.911	7.795	0.000	2552.002	4274.902
bill_length_mm	74.8126	6.076	12.313	0.000	62.860	86.765
bill_depth_mm	-145.5072	16.873	-8.624	0.000	-178.699	-112.315

Omnibus:	2.839	**Durbin-Watson:**	1.798
Prob(Omnibus):	0.242	**Jarque-Bera (JB):**	2.175
Skew:	-0.000	**Prob(JB):**	0.337
Kurtosis:	2.604	**Cond. No.**	644.

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.

Tenemos la misma información, la variable dependiente (**body_mass_g**) y las variables que intentan explicarlo, ahora hay un cambio de:

- **bill_length_mm:** 74.8126
- **bill_depth_mm:** -145.5072

Hay que empezar a asociar los resultados, en este caso **length** está asociado de manera positiva y **depth** de manera negativa.

Modelo 3

In [ ]:
```
model_3 = (
    smf.ols(
        formula='body_mass_g ~ bill_length_mm + bill_depth_mm + flipper_length_mm',
        data=preprocessed_penguins_df
    )
    .fit()
)
model_3.summary()
```

Out[ ]:

OLS Regression Results

Dep. Variable:	body_mass_g	**R-squared:**	0.764
Model:	OLS	**Adj. R-squared:**	0.762
Method:	Least Squares	**F-statistic:**	354.9
Date:	Wed, 07 Aug 2024	**Prob (F-statistic):**	9.26e-103
Time:	20:21:38	**Log-Likelihood:**	-2459.8
No. Observations:	333	**AIC:**	4928.
Df Residuals:	329	**BIC:**	4943.
Df Model:	3		
Covariance Type:	nonrobust		

	coef	std err	t	P>|t|	[0.025	0.975]
Intercept	-6445.4760	566.130	-11.385	0.000	-7559.167	-5331.785
bill_length_mm	3.2929	5.366	0.614	0.540	-7.263	13.849
bill_depth_mm	17.8364	13.826	1.290	0.198	-9.362	45.035
flipper_length_mm	50.7621	2.497	20.327	0.000	45.850	55.675

Omnibus:	5.596	**Durbin-Watson:**	1.982
Prob(Omnibus):	0.061	**Jarque-Bera (JB):**	5.469
Skew:	0.312	**Prob(JB):**	0.0649
Kurtosis:	3.068	**Cond. No.**	5.44e+03

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.
[2] The condition number is large, 5.44e+03. This might indicate that there are
strong multicollinearity or other numerical problems.

Las comparamos y podemos ver las pendientes que se ajustan para cada una de las variables:

- **bill length:** 3.2929
- **bill depth:** 17.8364
- **flipper length:** 50.7624

Hay que fijarnos que la longitud de pico cuando estaba sola significaba un gran cambio, conforme se fueron agregando variables; los resultados fueron cambiando. Ahora el **depth** no es negativo y al considerar diferentes variables los coeficientes van cambiando y va a ser necesario preguntarnos sobre la relación o validez de los datos.

Tenemos que fijarnos en el parámetro $r^2 \text{ ajustado}$.

En general cuando tengas mas de 2 variables enfocate en $r^2 \text{ ajustado}$ QUE NOS VA A INDICAR QUE TANTA VARIABILIDAD DE LOS DATOS SE ESTÁN AJUSTANDO.

En este caso para **el modelo 3 tenemos: 0.762**, si lo comparamos con el **modelo 2 que tiene: 0.464**. De manera que el **modelo 3** está siendo más preciso, y finalmente para el **modelo 1** tenemos 0.345, lo que quiere decir que estamos capturando muy poca variabilidad de los datos.

Conforme hemos ido aumentando el numero de variables estamos capturando mejor el comportamiento de nuestros pingüinos para predecir el peso que debería de tener.

Podríamos considerar seguir agregando variables, por ejemplo podríamos considerar el sexo del pingüino

Modelo 4

In [ ]:
```
model_4 = (
    smf.ols(
        formula='body_mass_g ~ bill_length_mm + bill_depth_mm + flipper_length_mm + C(sex)',
        data=preprocessed_penguins_df
    )
    .fit()
)
model_4.summary()
```

Out[ ]:

OLS Regression Results

Dep. Variable:	body_mass_g	**R-squared:**	0.823
Model:	OLS	**Adj. R-squared:**	0.821
Method:	Least Squares	**F-statistic:**	381.3
Date:	Wed, 07 Aug 2024	**Prob (F-statistic):**	6.28e-122
Time:	21:58:14	**Log-Likelihood:**	-2411.8
No. Observations:	333	**AIC:**	4834.
Df Residuals:	328	**BIC:**	4853.
Df Model:	4		
Covariance Type:	nonrobust		

	coef	std err	t	P>|t|	[0.025	0.975]
Intercept	-2288.4650	631.580	-3.623	0.000	-3530.924	-1046.006
C(sex)[T.male]	541.0285	51.710	10.463	0.000	439.304	642.753
bill_length_mm	-2.3287	4.684	-0.497	0.619	-11.544	6.886
bill_depth_mm	-86.0882	15.570	-5.529	0.000	-116.718	-55.459
flipper_length_mm	38.8258	2.448	15.862	0.000	34.011	43.641

Omnibus:	2.598	**Durbin-Watson:**	1.843
Prob(Omnibus):	0.273	**Jarque-Bera (JB):**	2.125
Skew:	0.062	**Prob(JB):**	0.346
Kurtosis:	2.629	**Cond. No.**	7.01e+03

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.
[2] The condition number is large, 7.01e+03. This might indicate that there are
strong multicollinearity or other numerical problems.

Algo que no podías hacer con el modelo de regresión simple; es que no podíamos agregar variables categóricas, en este caso si podemos agregar variables categóricas con el modelo de regresión multiple. Lo único que tenemos que hacer es:

- Encerrar a nuestra variable categórica entre paréntesis y con un 'C'. `C(sex)`

Análisis

Nuevamente nos fijamos en el parámetro $\mathrm{adjust}\ r^2$: y mejoró a **0.821**, pero ahora tenemos una pendiente que parece ser un poco diferente.
C(sex)[T.male] 541.0285

Esta pendiente lo que quiere decir es que está comparando el peso de los machos vs peso de las hembras. Lo que me dice es que LOS MACHOS PESAN 541.02 gr MAS QUE LAS HEMBRAS. Esta sería su interpretación.

Así podríamos ir agregando más parámetros y haciendo más complejo el modelo, pero sucede algo muy importante;
A veces el modelo más sencillo es más funcional y también podríamos combinar de otra forma las variables para ver su comportamiento.

Modelo 5

In [ ]:
```
model_5 = (
    smf.ols(
    formula ='body_mass_g ~ flipper_length_mm',
    data = preprocessed_penguins_df
    )
.fit()
)
model_5.summary()
```

Out[ ]:

OLS Regression Results

Dep. Variable:	body_mass_g	**R-squared:**	0.762
Model:	OLS	**Adj. R-squared:**	0.761
Method:	Least Squares	**F-statistic:**	1060.
Date:	Wed, 07 Aug 2024	**Prob (F-statistic):**	3.13e-105
Time:	20:54:39	**Log-Likelihood:**	-2461.1
No. Observations:	333	**AIC:**	4926.
Df Residuals:	331	**BIC:**	4934.
Df Model:	1		
Covariance Type:	nonrobust		

	coef	std err	t	P>|t|	[0.025	0.975]
Intercept	-5872.0927	310.285	-18.925	0.000	-6482.472	-5261.713
flipper_length_mm	50.1533	1.540	32.562	0.000	47.123	53.183

Omnibus:	5.922	**Durbin-Watson:**	2.116
Prob(Omnibus):	0.052	**Jarque-Bera (JB):**	5.876
Skew:	0.325	**Prob(JB):**	0.0530
Kurtosis:	3.025	**Cond. No.**	2.90e+03

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.
[2] The condition number is large, 2.9e+03. This might indicate that there are
strong multicollinearity or other numerical problems.

Fijémonos que la variable por si sola, ya ajusta el 0.76 de la variabilidad de los datos y nos da la pendiente que representa que un cambio en la longitud de la aleta representa 50 gr más.

Entonces una variable por si solo ya nos está dando mayor grado de variabilidad de los datos que un conjunto de variables (3 variables).

Podríamos considerar añadir la variable categórica del **sexo**, para saber si existe una mejora en el modelo.

In [ ]:
```
model_5 = (
    smf.ols(
    formula ='body_mass_g ~ flipper_length_mm + C(sex)',
    data = preprocessed_penguins_df
    )
.fit()
)
model_5.summary()
```

Out[ ]:

OLS Regression Results

Dep. Variable:	body_mass_g	**R-squared:**	0.806
Model:	OLS	**Adj. R-squared:**	0.805
Method:	Least Squares	**F-statistic:**	684.8
Date:	Wed, 07 Aug 2024	**Prob (F-statistic):**	3.53e-118
Time:	20:58:19	**Log-Likelihood:**	-2427.2
No. Observations:	333	**AIC:**	4860.
Df Residuals:	330	**BIC:**	4872.
Df Model:	2		
Covariance Type:	nonrobust		

	coef	std err	t	P>|t|	[0.025	0.975]
Intercept	-5410.3002	285.798	-18.931	0.000	-5972.515	-4848.085
C(sex)[T.male]	347.8503	40.342	8.623	0.000	268.491	427.209
flipper_length_mm	46.9822	1.441	32.598	0.000	44.147	49.817

Omnibus:	0.262	**Durbin-Watson:**	1.710
Prob(Omnibus):	0.877	**Jarque-Bera (JB):**	0.376
Skew:	0.051	**Prob(JB):**	0.829
Kurtosis:	2.870	**Cond. No.**	2.95e+03

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.
[2] The condition number is large, 2.95e+03. This might indicate that there are
strong multicollinearity or other numerical problems.

Como se puede observar, ya llegamos a una variabilidad en los datos de **0.80** que es muy significante porque en el anterior caso llegamos a **0.821** con 3 variables.

Así pues nos acercamos mas, en lugar de combinar multiples variables.

Jugar con este concepto me ayudará a entender que modelo funciona mejor y cual es el que mejor se adecua.

Sin embargo existen estadísticos que nos ayudarán a saber cual es el mejor modelo de manera estadística. Ejemplos de ellos pueden ser;

- Modelos de varianza: ANOVA.

SOBRE LAS NOTAS

[2] El numero de la condición es largo: Esto podría indicar que existe una particularidad o un problema numérico en nuestros datos. Recordemos que el coeficiente de relación de las clases pasadas entre las 2 variables, era alto y entonces esto es lo que quiere dar a entender este texto.

Digamos que cobra significado cuando tienes una variable con la que estas intentando explicar otra, que está altamente relacionada con una segunda variable que tengamos en el conjunto de datos, y mientras que con una variable podamos observar que se modifican los valores de manera positiva, con otra podemos observar totalmente lo contrario. Es decir cuando las combinamos el comportamiento puede ser caótico o raro.

Entonces siempre es bueno que cuando tratamos de hacer un modelo, observemos que tan relacionadas esta mis variables que voy a implementar, porque de no hacerlo nos podríamos enfrentar a la situación antes mencionada y el resultado podría no ser efectivo, es decir ya no podríamos confiar en los números y el resumen que me arroja el modelo.

En este sentido no nos afecta.

In [ ]:
```
model_5.params
```

Out[ ]:
```
Intercept           -5410.300224
C(sex)[T.male]        347.850254
flipper_length_mm      46.982175
dtype: float64
```

Ahora ¿cómo puedo comparar mis modelos para saber cuales se ajustan mejor a mis datos reales? Lo veremos en la siguiente clase.

Extras:

- ¿Qué es la regresión lineal multiple y cómo analizarla?.
- Pasos para realizar una regresión lineal multiple con Python.
- REGRESIÓN LINEAL MÚLTIPLE FÁCIL