EDA Penguins 🗺️🧭 🐧

Requerimientos:

```
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
```

```
%matplotlib inline
sns.set_style(style='whitegrid')
sns.set_context(context='notebook')
plt.rcParams['figure.figsize'] = (10,10)

penguin_color = {
    'Adelie':'#a6cee3',
    'Chinstrap':'#1f78b4',
    'Gentoo':'#b2df8a'
}
```

Ingesta de datos

Existen tres maneras de consumir la info.

1. Por el paquete palmerpenguins

2. A través de la carga de datos del dataset de Seaborn.

3. A través de la interfaz de DeepNote con pandas, read_csv

https://raw.githubusercontent.com/allisonhorst/palmerpenguins/master/inst/extdata/penguins_raw.csv

https://raw.githubusercontent.com/allisonhorst/palmerpenguins/master/inst/extdata/penguins.csv

```
#Utilizamos el paquete palmerpenguins que incluye deepnote.
raw_penguins_df = palmerpenguins.load_penguins_raw()
#raw_penguins_df.head()
raw_penguins_df
```

	studyName object	Sample Number i...	Species object	Region object	Island object	Stage object	Individual ID object	C
	PAL0910 34.9% PAL0809 33.1% PAL0708 32%	1 - 152	Adelie Pe... 44.2% Gentoo pen... 36% Chinstrap ... 19.8%	Anvers 100%	Biscoe 48.8% Dream 36% Torgersen 15.1%	Adult, 1 Eg... 100%	N6A1 0.9% N6A2 0.9% 188 others 98.3%	Y N
0	PAL0708	1	Adelie Penguin (P...	Anvers	Torgersen	Adult, 1 Egg Stage	N1A1	Y
1	PAL0708	2	Adelie Penguin (P...	Anvers	Torgersen	Adult, 1 Egg Stage	N1A2	Y
2	PAL0708	3	Adelie Penguin (P...	Anvers	Torgersen	Adult, 1 Egg Stage	N2A1	Y
3	PAL0708	4	Adelie Penguin (P...	Anvers	Torgersen	Adult, 1 Egg Stage	N2A2	Y
4	PAL0708	5	Adelie Penguin (P...	Anvers	Torgersen	Adult, 1 Egg Stage	N3A1	Y
5	PAL0708	6	Adelie Penguin (P...	Anvers	Torgersen	Adult, 1 Egg Stage	N3A2	Y
6	PAL0708	7	Adelie Penguin (P...	Anvers	Torgersen	Adult, 1 Egg Stage	N4A1	N
7	PAL0708	8	Adelie Penguin (P...	Anvers	Torgersen	Adult, 1 Egg Stage	N4A2	N
8	PAL0708	9	Adelie Penguin (P...	Anvers	Torgersen	Adult, 1 Egg Stage	N5A1	Y
9	PAL0708	10	Adelie Penguin (P...	Anvers	Torgersen	Adult, 1 Egg Stage	N5A2	Y

344 rows, 17 cols 10 / page Page 1 of 35

```
penguins_df = palmerpenguins.load_penguins() #preprocessed
penguins_df
```

	species object Adelie 44.2% Gentoo 36% Chinstrap 19.8%	island object Biscoe 48.8% Dream 36% Torgersen 15.1%	bill_length_mm fl... 32.1 - 59.6	bill_depth_mm fl... 13.1 - 21.5	flipper_length_mm 172.0 - 231.0	body_mass_g flo... 2700.0 - 6300.0	sex object male 48.8% female 48% Missing 3.2%	y 2
0	Adelie	Torgersen	39.1	18.7	181	3750	male	
1	Adelie	Torgersen	39.5	17.4	186	3800	female	
2	Adelie	Torgersen	40.3	18	195	3250	female	
3	Adelie	Torgersen	nan	nan	nan	nan	nan	
4	Adelie	Torgersen	36.7	19.3	193	3450	female	
5	Adelie	Torgersen	39.3	20.6	190	3650	male	
6	Adelie	Torgersen	38.9	17.8	181	3625	female	
7	Adelie	Torgersen	39.2	19.6	195	4675	male	
8	Adelie	Torgersen	34.1	18.1	193	3475	nan	
9	Adelie	Torgersen	42	20.2	190	4250	nan	

344 rows, 8 cols 10 / page Page 1 of 35

¿Qué tipo de dato son las variables del conjunto de datos?

```
penguins_df.dtypes
```

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

¿Cuántas variables de cada tipo de dato tenemos en el conjunto de datos?

```
(
    penguins_df
    .dtypes
    .value_counts()
)
```

```
float64    4
object     3
int64      1
dtype: int64
```

¿Cuántas variables y observaciones tenemos en el conjunto de datos?

```
penguins_df.shape
```

```
(344, 8)
```

¿Existen valores nulos explicitos en el conjunto de datos?

```
(
    penguins_df
    .isnull()
    #.any()
)
```

	species bool False 100%	island bool False 100%	bill_length_mm b... False 99.4% True 0.6%	bill_depth_mm b... False 99.4% True 0.6%	flipper_length_mm False 99.4% True 0.6%	body_mass_g bool False 99.4% True 0.6%	sex bool False 96.8% True 3.2%	y
0	False	False	False	False	False	False	False	F
1	False	False	False	False	False	False	False	F
2	False	False	False	False	False	False	False	F
3	False	False	True	True	True	True	True	F
4	False	False	False	False	False	False	False	F
5	False	False	False	False	False	False	False	F
6	False	False	False	False	False	False	False	F
7	False	False	False	False	False	False	False	F
8	False	False	False	False	False	False	True	F
9	False	False	False	False	False	False	True	F

344 rows, 8 cols 10 / page Page 1 of 35

De tener observaciones con valores nulos, ¿cuántas tenemos por cada variable?

```
(
    penguins_df
    .isna() #in this case, the same as isnull()
    .sum()
    .sort_values(ascending=True)
)
```

¿Cuántos valores nulos tenemos en total en el conjunto de datos?

```
(
    penguins_df
    .isna()
    .sum()
    .sum()
)
```

¿Cuál es la proporción de valores nulos por cada variable?

```
(
    penguins_df
    .isnull()
    .melt()
    .pipe(
        lambda df:(
            sns.displot(
                df,
                y='variable',
                hue='value',

            )
        )
    )
)
```

¿Cómo podemos visualizar los valores nulos en todo el conjunto de datos?

```
(
    penguins_df
    .isnull()
    .transpose()
)
```

```
(
    penguins_df
    .isnull()
    .transpose()
    .pipe(
        lambda df:(
            sns.heatmap(
            data=df
            )
        )
    )
)
```

¿Cuántas observaciones perdemos si eliminamos datos faltantes?

```
copy_df = penguins_df.copy()
(
    copy_df
    .dropna()
)
```

```
# Codigo añadido posteriormente al realizar todos los análisis y modelos
# Procedemos con la eliminación de los valores NaN por la facilidad al ajustar la evaluación de resultados
# de los distintos modelos. Y porque los mismos no representan gran porcentaje.
penguins_df.dropna(inplace=True)
```

Estableciendo relaciones: Gráfica de puntos

```
sns.scatterplot(
    data=penguins_df,
    x='bill_length_mm',
    y='bill_depth_mm',
    hue='species',
    alpha=1/2,
    s=100
)
```

```
sns.displot(
    data=penguins_df,
    kind='kde',
    x='bill_length_mm',
    y='bill_depth_mm',
    rug=True
)
```

```
sns.jointplot(
    data=penguins_df,
    x='bill_length_mm',
    y='bill_depth_mm',
    palette=penguin_color,
    hue='species'
)
```

Estableciendo relaciones: Gráficos de violín y boxplots

```
sns.scatterplot(
    data=penguins_df,
    x='species',
    y='flipper_length_mm',
    hue='species',
    palette=penguin_color
)
```

```
sns.stripplot(
    data=penguins_df,
    x='species',
    y='flipper_length_mm',
    palette=penguin_color
)
```

```
ax = sns.boxplot(
    data=penguins_df,
    x='species',
    y='flipper_length_mm',
    palette=penguin_color
)

ax = sns.stripplot(
    data=penguins_df,
    x='species',
    y='flipper_length_mm'
)
```

```
ax = sns.violinplot(
    data=penguins_df,
    x='species',
    y='flipper_length_mm'
)

ax = sns.stripplot(
    data=penguins_df,
    x='species',
    y='flipper_length_mm',
    palette=penguin_color
)
```

```
sns.swarmplot(
    data=penguins_df,
    x='species',
    y='flipper_length_mm',
    hue='species',
    palette=penguin_color
)
```

```
sns.scatterplot(
    data=penguins_df,
    x='island',
    y='body_mass_g',
    hue='species',
    palette=penguin_color
)
```

```
ax = sns.kdeplot(
    data=penguins_df,
    x='body_mass_g',
    hue='species',
    palette=penguin_color,
    fill=True
)
```

Matrices de Correlación

¿Existe una correlación LINEAL entre alguna de las variables?

```
penguins_correlations = penguins_df.corr().round(4)
penguins_correlations
```

```
sns.heatmap(
    data=penguins_correlations,
    cmap=sns.diverging_palette(20, 230, as_cmap=True),
    center=0,
    vmin=-1,
    vmax=1,
    linewidths=0.5,
    annot=True
)
```

```
sns.clustermap(
    data=penguins_correlations,
    cmap=sns.diverging_palette(20, 230, as_cmap=True),
    center=0,
    vmin=-1,
    vmax=1,
    linewidths=0.5,
    annot=True
)
```

¿Cómo representar una variable categórica con una variable numérica discreta?

```
penguins_df = (
    penguins_df
    .assign(
        numeric_sex=lambda s: s.sex.replace(['female','male'],[1,0])
    )
)
```

¿Cuáles son algunas limitantes de los coeficientes de correlación lineal?

Solo nos.

```
x = np.linspace(-100,100,100)
y = x**2
z = x**3

#print(x) # matriz original linspace
#print(y) # matriz de x elevado al cuadrado | par
#print(z) # matriz de z elevado al cubo | impar

#y += np.random.normal(0,1000, x.size) # ruido
#z += np.random.normal(0,1000, x.size)

print(np.corrcoef(x,y)) # no tienen una correlación lineal, sin embargo sabemos que existe una relación cuadratica
print(np.corrcoef(x,z)) # no tienen una relación lineal alta, sin embargo sabemos que tienen una relación cúbica

ax = sns.scatterplot(x)
#ax = sns.scatterplot(y)
ax = sns.scatterplot(z)
```

```
sns.scatterplot(
    penguins_df,
    x='bill_length_mm',
    y='body_mass_g'
)
```

```
np.random.seed(42)

x_1 = np.linspace(0,100,100)
y_1 = 0.1 * x_1 + 3 + np.random.uniform(-2,2, size=x_1.size)

sns.scatterplot(
    x=x_1,
    y=y_1
)
x_2 = np.linspace(0,100,100)
y_2 = 0.5 * x_2 + 1 + np.random.uniform(0,60,size=x_2.size)

sns.scatterplot(
    x=x_2,
    y=y_2
)

plt.legend(labels=['1','2'])

print(np.corrcoef(x_1,y_1))
print(np.corrcoef(x_2,y_2))
```

Estableciendo relaciones: Análisis de regresión simple

```
res_1 = scipy.stats.linregress(x=x_1, y=y_1)
res_2 = scipy.stats.linregress(x=x_2, y=y_2)

print(res_1, res_2, sep="\n")
```

```
sns.scatterplot(
    x=x_1,
    y=y_1
)

fx_1 = np.array([x_1.min(), x_1.max()])
fy_1 = res_1.intercept + res_1.slope * fx_1
print(fx_1, fy_1)

plt.plot(fx_1,fy_1)

sns.scatterplot(
    x=x_2,
    y=y_2
)

fx_2 = np.array([x_2.min(), x_2.max()])
fy_2 =  res_2.intercept + res_2.slope * fx_1
print(fx_2, fy_2)

plt.plot(fx_2,fy_2)

plt.legend(labels=['1','1','2','2'])
```

```
#penguins_df.bill_length_mm.fillna(0, inplace=True)
#penguins_df.bill_depth_mm.fillna(0, inplace=True)
```

```
sns.scatterplot(
    penguins_df,
    x='bill_length_mm',
    y='bill_depth_mm',
    #hue='species'
)

res_penguins = scipy.stats.linregress(
    x = penguins_df.bill_length_mm,
    y = penguins_df.bill_depth_mm
)

print(res_penguins)

# fx = np.array([penguins_df.bill_length_mm.min(), penguins_df.bill_depth_mm.max()])
# fy = res_penguins.intercept + res_penguins.slope * fx
# print(fx, fy)
# plt.plot(fx, fy)
```

```
sns.lmplot(
    penguins_df,
    x='bill_length_mm',
    y='bill_depth_mm',
    hue='species'
)
```

Limitantes de la Regresión Lineal Simple

1.

```
penguins_df.bill_length_mm.isna().value_counts()
penguins_df.sex.isna().value_counts()
```

```
penguins_df.fillna(0)
```

Análisis de regresión múltiple

Olvidé mi báscula para pesar a los pinguinos, ¿Cuál sería la mejor forma de capturar ese dato?

Creando modelos

Modelo 1

```
model_1 = (
    smf.ols(
        formula = 'body_mass_g ~ bill_length_mm', #asociacion de peso con el largor del pico
        data = penguins_df,

    )
    .fit()
)

model_1.summary()
```

Modelo 2

```
model_2 = (
    smf.ols(
        formula = 'body_mass_g ~ bill_length_mm + bill_depth_mm', # asociaciones múltiples
        data = penguins_df,

    )
    .fit()
)

model_2.summary()
```

Modelo 3

```
model_3 = (
    smf.ols(
        formula = 'body_mass_g ~ bill_length_mm + bill_depth_mm + flipper_length_mm',
        data = penguins_df,

    )
    .fit()
)

model_3.summary()
```

Modelo 4

```
model_4 = (
    smf.ols(
        formula = 'body_mass_g ~ bill_length_mm + bill_depth_mm + flipper_length_mm + C(sex)',
        # se pueden ingresar variables categóricas directamente
        data = penguins_df,

    )
    .fit()
)

model_4.summary()
```

Modelo 5

```
model_5 = (
    smf.ols(
        formula = 'body_mass_g ~ flipper_length_mm + C(sex)',
        # se pueden ingresar variables categóricas directamente
        data = penguins_df,

    )
    .fit()
)

model_5.summary()
```

Warning:

Siempre analizar las relaciones de las variables que tratan de predecir la variable objetivo, no debe existir demasiada correlación entre variables explicativas, ahora bien, en caso de que una variable explicativa esté muy relacionada a la variable objetivo SÍ indica un gran potencial de predictibilidad, lo cual es bueno, como el caso del flipper length y sexo para la predicción del peso de un pinguino.

Visualizando resultados

Creación de tabla de resultados

```
models_results = pd.DataFrame(
    dict(
        actual_value=penguins_df.body_mass_g,
        prediction_model_1 = model_1.predict(),
        prediction_model_2 = model_2.predict(),
        prediction_model_3 = model_3.predict(),
        prediction_model_4 = model_4.predict(),
        prediction_model_5 = model_5.predict(),
        species=penguins_df.species,
        sex=penguins_df.sex
    )
)

models_results
```

ECDFs

```
sns.ecdfplot(
    data=models_results
)
```

PDFs

```
sns.kdeplot(
    data=models_results,
    cumulative=True
)
```

¿Qué pudimos haber encontrado antes de hacer los modelos?

```
sns.lmplot(
    data=penguins_df,
    x='flipper_length_mm',
    y='body_mass_g',
    hue='sex'
)
```

Siempre explorar los datos hasta conocerlos muy bien antes de iniciar con los modelos. Comprender la información.

El EDA pudo haber ahorrado muchísimo tiempo para llegar a la conclusión debido a la comprensión de la data.

Siempre visualizar datos.

Análisis de Regresión Logística

¿Podemos crear un modelo que nos ayude a saber cuando un pinguino es macho o hembra?

```
smf.logit(
    formula='numeric_sex ~ flipper_length_mm + bill_length_mm + bill_depth_mm + C(island)',
    data=penguins_df
).fit().summary()
```

El parámetro que no aparece en una variable categórica es a la cual se está haciendo referencia

Exploración de nuestras variables categóricas

```
(
    penguins_df
    .value_counts(['island','sex'])
    .reset_index(name='count')
)
```

Los datos no están equilibrados, por ende, el modelo está sesgado y es por eso que el resultado de la probabilidad de que existan machos en la isla biscoe era la más alta. Es de suma importancia explorar y comprender los datos antes de construir los modelos propiamente.

¿Podemos definir un modelo que nos ayude a identificar si un pingüino pertenece a determinada especie?

```
# Tipos de especies
penguins_df.species.unique()
```

```
# Remplazamos las variables categóricas por valores numéricos
penguins_df = (
    penguins_df
    .assign(is_adelie=lambda df: df.species.replace(['Adelie','Gentoo','Chinstrap'], [1,0,0]))
)

penguins_df[['species','is_adelie']].value_counts()
```

```
model_is_adelie = smf.logit(
    formula='is_adelie ~ flipper_length_mm + C(sex)',
    data=penguins_df
).fit()

model_is_adelie.params
```

```
# validacion
is_adelie_df_predictions = pd.DataFrame(
    dict(
        actual_adelie = penguins_df.is_adelie,
        predicted_value = model_is_adelie.predict().round()
    )
)

is_adelie_df_predictions.head()
```

```
(
    is_adelie_df_predictions
    .value_counts(['actual_adelie', 'predicted_value'])
    .reset_index(name='count')
)
```

Matriz de confusión

```
print(
    sklearn.metrics.confusion_matrix(
        is_adelie_df_predictions.actual_adelie,
        is_adelie_df_predictions.predicted_value
    )
)

sklearn.metrics.accuracy_score(
    is_adelie_df_predictions.actual_adelie,
    is_adelie_df_predictions.predicted_value
)
```

Paradoja de Simpson

```
sns.scatterplot(
    data=penguins_df,
    x='bill_depth_mm',
    y='bill_length_mm'
)
```

```
sns.lmplot(
    data=penguins_df,
    x='bill_depth_mm',
    y='bill_length_mm'
)
```

```
sns.lmplot(
    data=penguins_df,
    x='bill_depth_mm',
    y='bill_length_mm',
    hue='sex'
)
```

```
sns.lmplot(
    data=penguins_df,
    x='bill_depth_mm',
    y='bill_length_mm',
    hue='species'
)
```

```
sns.pairplot(
    data=penguins_df,
    hue='species',
    palette=penguin_color
)
```