Special Topics Práctica 1

Leonardo Sanchez y Jose Manuel Llavona

1. Revisión inicial (0.6 puntos)

1. ¿Cuántos registros de pasajeros tiene el dataset y cuántas variables?

- El dataset tiene 712 entradas de pasajeros y 14 variables.

2. ¿Qué tipo de datos tiene cada variable?

```
RangeIndex: 712 entries, 0 to 711
Data columns (total 11 columns):
 #   Column       Non-Null Count  Dtype
---  ------       --------------  -----
 0   PassengerId  712 non-null    int64
 1   Survived     712 non-null    int64
 2   Pclass       712 non-null    int64
 3   Name         712 non-null    object
 4   Sex          712 non-null    object
 5   SibSp        712 non-null    int64
 6   Parch        712 non-null    int64
 7   Ticket       712 non-null    object
 8   Fare         712 non-null    float64
 9   Embarked     712 non-null    object
 10  BirthDate    712 non-null    object
```

- Int64 = Variable numerica Entera
- Float64 = Variable numérica con decimales
- Object = Variable no definida

3. ¿Crees que los tipos de datos son los correctos según el significado de cada columna? Justifica tu respuesta.

- La mayoría de los datos si están en su correcto tipo pero hay varios errores, por ejemplo:
    1) PassenegrId debería de ser un “String” y no una variable numérica (Aunque sí sea un número) ya que no se hacen ecuaciones matemáticas con esa variable.
    2) Birthdate debería de ser un dato tipo “DateTime”.
    3) La variable survived debería ser tipo “Bit”, ya que es una variable binaria.

2. Estadísticas descriptivas iniciales (1.2 puntos)

1. Realiza un análisis estadístico inicial ejecutando dataset.describe().

• Para las variables categóricas Sex y Embarked:

• ¿Cuántas categorías tiene cada una?

- Sex es una variable que tiene 2 categorías (Male o Female)
- Embarked es una variable que tiene 3 categorías (S, C o Q)

• *¿Cuál es la categoría más común y con qué frecuencia aparece?*
- Para la variable sex la mas comun es la categoria “Male”
- Para la variable embarked la categoría más común es “S”

• *Dime las medias de todas las variables numéricas.*

```
Median of Pclass: 2.24
Median of SibSp: 0.51
Median of Parch: 0.43
Median of Fare: 34.57
```

• *¿Todas esas medidas tienen sentido en el contexto del problema? ¿Qué variables numéricas deberían tratarse como categóricas y por qué?*
- No todas las medias de las variables tienen sentido en el problema ya que algunas de las variables tomadas no son numéricas sino categorías con un índice numérico. Las variables que deberían de ser tratadas como categorías son: “Pclass”, “SibSp” y “Parch”

**La variable index fue eliminada ya que no da información real del problema, solamente es utilizada como guía.*

3. Distribución de la variable objetivo (0.9 puntos)

1. Transforma la variable Survived para que contenga las etiquetas: 0 → "No", 1 → "Sí".

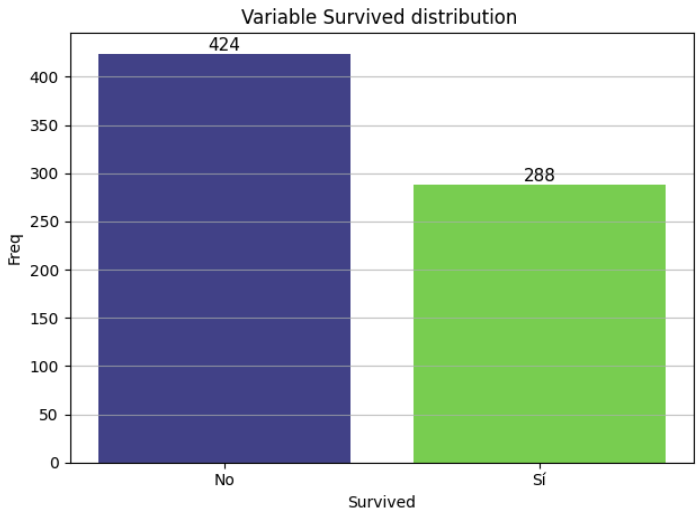
```
df["Survived"] = df["Survived"].replace({0: "No", 1: "Sí"})
```

2. Cuenta cuántos pasajeros sobrevivieron y cuántos no.

NO → 424

SI → 288

3. Representa gráficamente esta distribución en un gráfico de barras.



4. Enriquecimiento de la base de datos (1.8 puntos)

1. Crea una nueva columna LastName con el apellido del pasajero.

2. Crea una nueva columna FirstName con el nombre propio del pasajero.

3. Extrae el título (Mr, Mrs, Miss, etc.) en una nueva columna Title.

```
# Assuming your DataFrame is df and the column is "Name
# Step 1 – Split surname from the rest
df[["Surname", "Rest"]] = df["Name"].str.split(",", n=1, expand=True)

# Step 2 – Extract title (Mr, Mrs, Miss, etc.)
df["Title"] = df["Rest"].str.extract(r'([A-Za-z]+)\.')

# Step 3 – Extract first and middle names (after the title)
df["FirstName"] = df["Rest"].str.extract(r'\.\s*(.*)')

# Step 4 – Clean extra spaces
df["Surname"] = df["Surname"].str.strip()
df["Title"] = df["Title"].str.strip()
df["FirstName"] = df["FirstName"].str.strip()
```

```
# Step 5 – Drop the temporary column
df.drop(columns=["Rest"], inplace=True)
```

4. Crea una columna FamilySize = SibSp + Parch + 1.

5. A partir de ella, genera la variable IsAlone (1 si FamilySize == 1, 0 en caso contrario).

```
# create family size
df["Family_Size"] = df["SibSp"] + df["Parch"] + 1
#Is alone column
df["IsAlone"] = 0
df.loc[df["Family_Size"] == 1, "IsAlone"] = 1
```

6. ¿Qué utilidad podrían tener estas variables en un análisis predictivo?

- Estás variables pudieran servir como regresor o variable predictiva en un modelo de Machine Learning, ya que puede que haya una correlación entre las variables y la variable que quisiéramos predecir, en este caso la variable "survived". Un ejemplo real puede ser el aumento de probabilidad de entrar a un barco de auxilio si estás solo/a y no con una familia con la cual debes cargar y proteger.

5. Enriquecimiento II (1.8 puntos)

1. Calcula la edad de los pasajeros en una nueva columna Age, teniendo en cuenta la fecha de nacimiento y fecha del suceso (10 de abril de 1912).

```
fecha_titanic = pd.Timestamp("1912-04-15")  # día del hundimiento
df["BirthDate"] = pd.to_datetime(df["BirthDate"], errors="coerce")
df["Age"] = (fecha_titanic - df["BirthDate"]).dt.days / 365.25
```

2. Ordena el dataset por Age de manera ascendente.

• Indica el apellido y la edad de la persona más joven y de la más mayor.

```
#sorted
df.sort_values(by="Age", ascending=True, inplace=True)
print(df["Age"])
#descending
df.sort_values(by="Age", ascending=False, inplace=True)
print(df["Age"])
```

La persona más joven es: Mr. Alegroon Henry Wilson Barkworth con 80 años
La persona más mayor es: Master. Assad Alexander con 0.418891 años, concluimos de que es un bebe.

3. ¿En qué mes se concentran más nacimientos?

El mes donde se concentran la mayoría de los nacimientos es en Abril.

4. Crea una variable categórica AgeGroup con los intervalos:

• 0–12 (niños)
• 13–18 (adolescentes)
• 19–35 (jóvenes)
• 36–60 (adultos)
• 60+ (mayores)

```
#age group
df["AgeGroup"] = ""
df.loc[(df["Age"] > 0) & (df["Age"] < 12), "AgeGroup"] = "Niño"
df.loc[(df["Age"] >= 12) & (df["Age"] < 18), "AgeGroup"] = "Adolescente"
df.loc[(df["Age"] >= 18) & (df["Age"] < 35), "AgeGroup"] = "Jovenes"
df.loc[(df["Age"] >= 35) & (df["Age"] < 60), "AgeGroup"] = "Adultos"
df.loc[df["Age"] >= 60, "AgeGroup"] = "Adulto mayor"
```

5. Cambia los valores de la variable Pclass: 1→1st, 2→ 2nd, 3→ 3rd.

```
#class replace
df["Pclass"] = df["Pclass"].replace({1: "1 St", 2: "2 St",3:"3 St"})
```

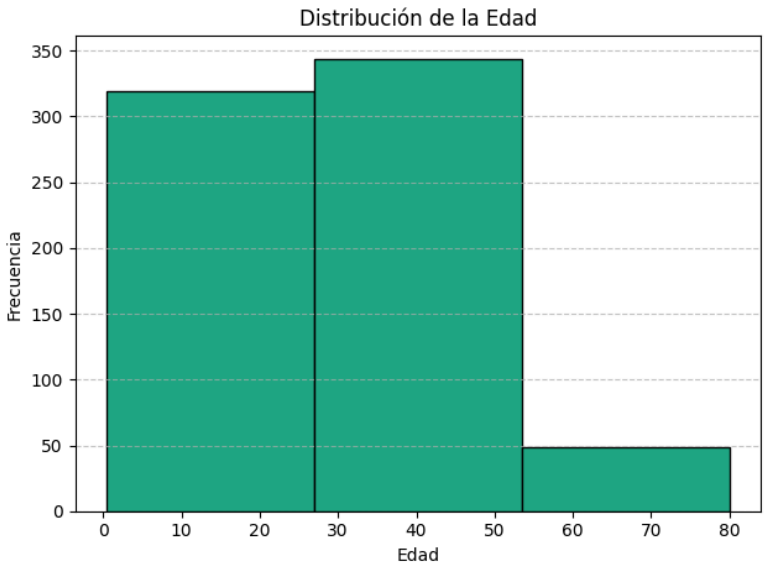
6. Análisis de las variables (2.7 puntos)

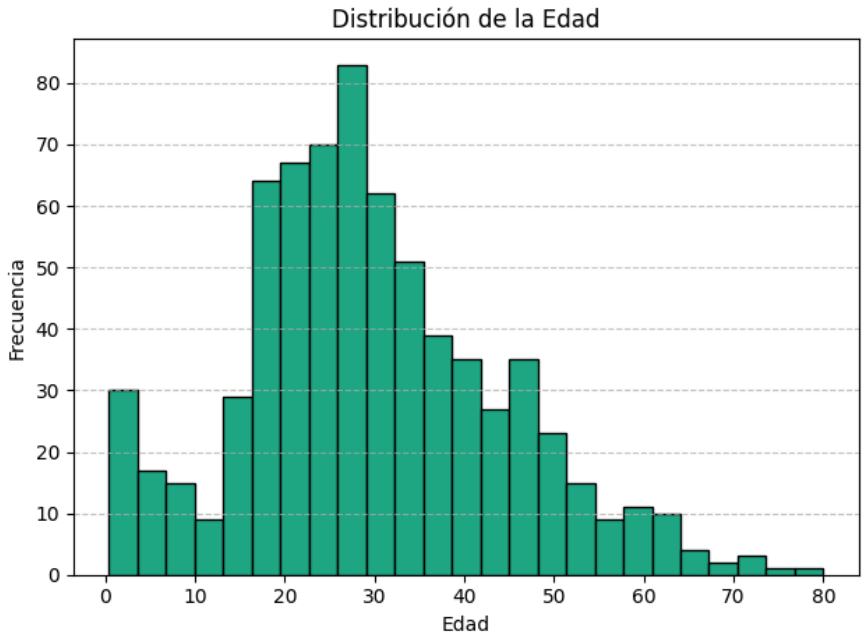
1. Haz una tabla de contingencia entre Sex y Survived. ¿Sobrevivieron más hombres o mujeres? Explica los resultados obtenidos.

Survived	No	Sí
Sex		
female	64	195
male	360	93

- Gracias a los resultados de la tabla podemos observar que más mujeres que hombres sobrevivieron el titanic. También podemos observar que la mayoría de las personas fallecieron en el accidente y que la mayoría de esas personas eran hombres ya que había muchos más que mujeres en el barco. Finalmente podemos concluir que se le dio prioridad a las mujeres en la evacuación ya que eran mayoría pero mas mujeres sobrevivieron.

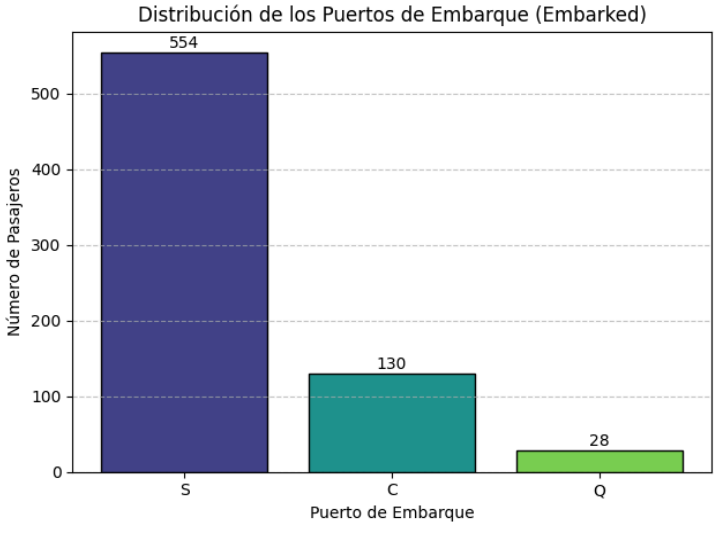
2. Haz un histograma de Age con 3 intervalos y luego 25. Describe la distribución.





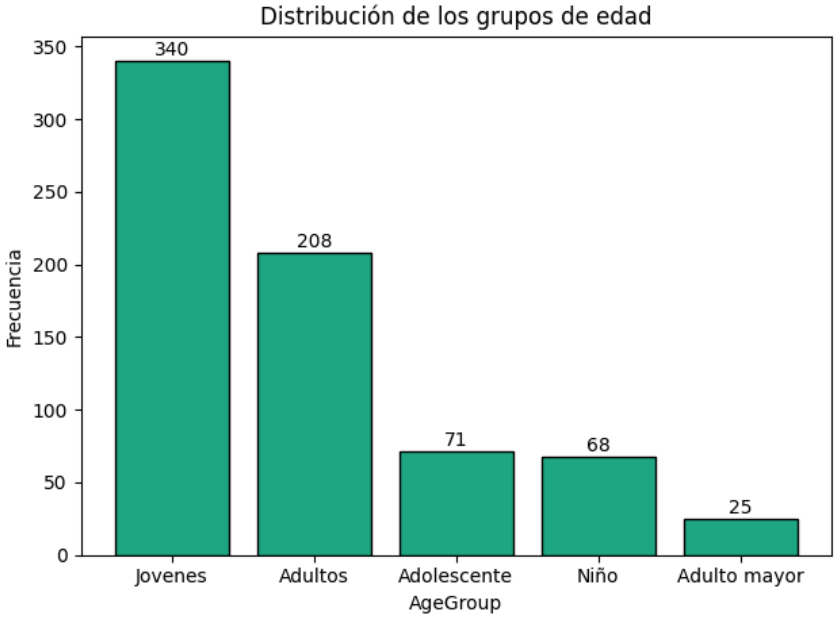
- En el primer histograma podemos observar que la mayoría de las personas que estaban a bordo del Titanic eran personas entre 25 y 55 años con una frecuencia de 325. En segundo lugar están las personas con un rango de edad entre 0 y 25 años con una frecuencia de 310. Aplicándolos a la situación real se puede conciliar que la mayoría de las personas a bordo eran familias que tenían bebés y también personas jóvenes.
- En el segundo histograma se ve una imagen más clara ya que la distribución se ve mejor. El histograma presenta una imagen donde el rango de edad donde la mayoría de las personas pertenecen es entre 20 y 45 años, con una concentración mayor en personas de 28 a 32 años. También resaltamos de que se presenta una frecuencia alta en personas con edades 0 a 5, esto apoya la hipótesis de que la mayoría de las personas a bordo eran familias jóvenes con bebés.

3. ¿Se puede hacer un histograma de la variable Embarked? Si no es adecuado, explica qué tipo de gráfico alternativo sería más apropiado y por qué, y analiza resultados.



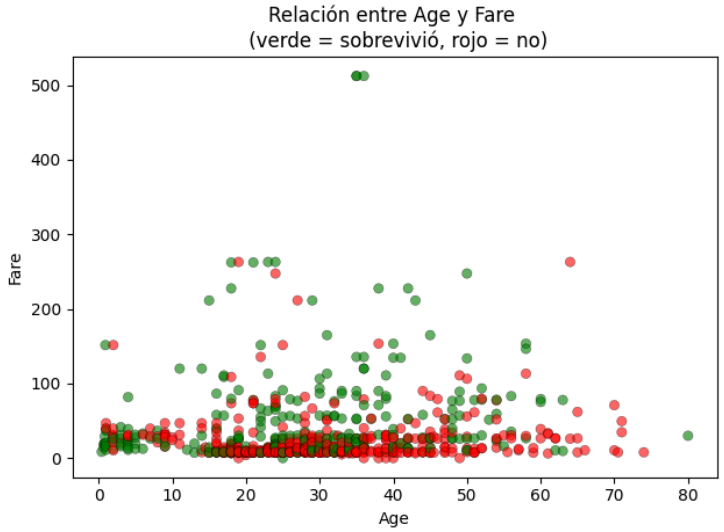
- El hecho de hacer una histograma para la variable Embarked no es posible ya que la variable contiene valores categóricos discretos y no numéricos continuos. Para esta variable el mejor gráfico alternativo fuera uno de barras, ya que se acomoda a la naturaleza de la variable. Arriba se encuentra un gráfico de barras para la variable Embarked y se puede concluir que el puerto de embarque más común es el puerto "S", y esto se puede interpretar como el puerto en donde los pasajeros con boletos normales (No Premium) embarcan.

4. ¿En qué grupo AgeGroup se concentra la mayor parte de pasajeros? ¿Y la menor? Realiza el gráfico adecuado.



- La mayoría de los pasajeros se concentran en el grupo "Jóvenes", el cual representa a personas de edades de 18 a 35 años. La menor concentración de pasajeros se manifiesta en el grupo "Adulto Mayor" el cual representa a las pasajeros con una edad mayor a 60.

5. Haz un gráfico de dispersión Age y Fare, coloreando los puntos por la variable Survived. ¿Observas algún patrón?



- Interpretando el gráfico relacionando Edad y Tarifa con la supervivencia del pasajero, no se puede ver algún patrón muy concreto conectando las tres variables. El único patrón relevante se manifiesta en la aparición de mayores puntos verdes en cuanto se aumenta la variable “Fare”, insinuando en que cuanto más se paga en tarifa el mayor es el chance de supervivencia.

6. Calcula correlación de Pearson entre Age y Fare. Explica qué significa el valor obtenido. ¿Tiene sentido observando el gráfico anterior?

```
corr = df["Age"].corr(df["Fare"], method="pearson")
print(f"Coeficiente de correlación de Pearson (Age vs Fare): {corr:.3f}")
```

```
[39]  ✓ 0.0s
```

```
Coeficiente de correlación de Pearson (Age vs Fare): 0.093
```

- El coeficiente de pearson mide la **fuerza y dirección de la relación lineal** entre dos variables numéricas. El coeficiente tiene un rango de -1 a 1, donde un coeficiente de -1 representa una relación lineal negativa perfecta y un 1 representa una relación lineal positiva perfecta. El valor del coeficiente de pearson entre la variable age y “Fare” es de 0.093, indicando que hay muy poca relación lineal entre las dos variables. Esto se ve manifestado en el gráfico de arriba ya que no hay un patrón concreto entre las dos variables.

7. Si quisieras ver si existe una relación significativa entre Sex y Survived: qué prueba estadística usarías, aplícala e interpreta.

```
from scipy import stats
from scipy.stats import chi2_contingency
contingencia = pd.crosstab(df['Survived'], df['Sex'])
print("Tabla de contingencia:")
print(contingencia)

chi2_stat, p_val, dof, expected = chi2_contingency(contingencia)
print("\nChi2:", chi2_stat)
print("p-valor:", p_val)

if p_val < 0.05:
    print("Existe dependencia significativa entre Survived y Sex")
else:
    print("No hay evidencia de dependencia entre Survived y Sex")
```

41] ✓ 22.3s

```
Tabla de contingencia:
Sex       female  male
Survived
No            64   360
Sí           195    93

Chi2: 202.86944877617123
p-valor: 4.939416685451492e-46
Existe dependencia significativa entre Survived y Sex
```

- La mejor forma de probar la relación entre dos variables categóricas es usando la estadística “Chi-Square” en conjunto con una tabla de contingencia para probar la independencia de las variables.
- La estatistica Chi-Square funciona comparando las hipótesis nula (Son independientes) y la hipótesis 1 (No son independientes). Se rechaza la hipótesis nula si el P-Valor de la estadística es menor que 0.05
- Al aplicar la estadística a las variables “Sex” y “Survived”, observamos que no se rechaza la hipótesis nula, indicando que las variables son independientes entre sí.

8. Si quieres analizar si la tarifa (Fare) varía significativamente entre los distintos puertos de Embarked (Embarked): qué prueba estadística usarías, aplícala e interpreta. ¿En este caso ¿Sería apropiado realizar un t-test? Justifica tu respuesta.

Para una prueba estadística que compara una variable cuantitativa como Fare y una variable categórica como embarked usaremos la prueba de ANOVA(One Way). En este ANOVA test veríamos si la tarifa de los pasajeros varía gracias a los puertos de embarque:

H0: La tarifa de los pasajeros es igual en todos los puertos

H1: Al menos una tarifa varía gracias al puerto

```
anova_results = stats.f_oneway(
    df[df['Embarked'] == 'C']['Fare'],
    df[df['Embarked'] == 'S']['Fare'],
    df[df['Embarked'] == 'Q']['Fare'],
)

# ¿Las medias de las temperaturas maximas son estadisticamente diferentes en estos 4 eventos?
print(f"F-statistic: {anova_results.statistic}")
print(f"P-value: {anova_results.pvalue}")
```

✓ 0.0s

```
F-statistic: 35.89219045311385
P-value: 1.4184349041131767e-15
```

Rechazamos la hipótesis nula si el P-Valor < 0.05.

Ya que el P-Valor de la prueba es < 0, rechazamos la hipótesis nula y concluimos de que si hay diferencia en las tarifas si hay un cambio de puerto de embarque.

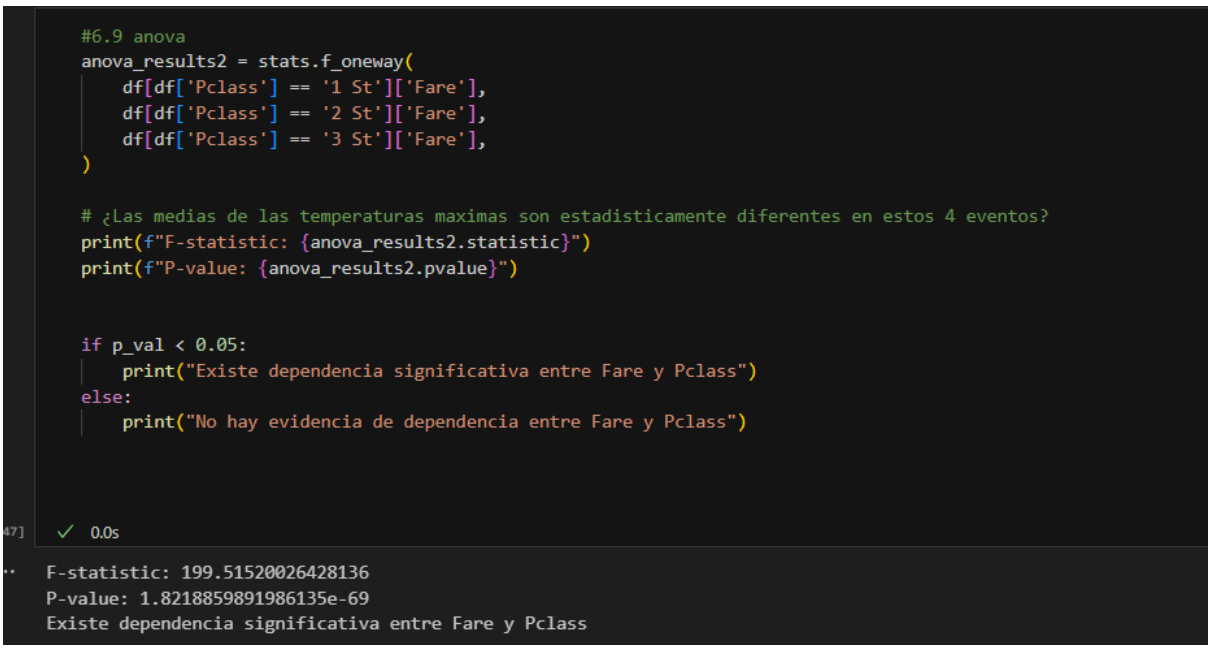
Para estas situaciones de comparaciones entre una variable categórica y una variable cuantitativa, si la variable categórica solamente tiene 2 categorías, el uso de un test si es apropiado. Ya que nuestra variable categórica tiene 3 categorías, se utiliza un ANOVA.

9. Si quieres analizar si la tarifa (Fare) varía de forma significativa entre las clases (Pclass): qué prueba estadística usarías, aplícala e interpreta. Adicionalmente, representa un boxplot de la tarifa (Fare) en función de la clase (Pclass). ¿Qué observas?

Para una prueba estadística que compara una variable cuantitativa como Fare y una variable categórica como P-class usaremos la prueba de ANOVA(One Way). En este ANOVA test veríamos si la tarifa de los pasajeros varía gracias a la clase del boleto que compraron

H0: La tarifa de los pasajeros es igual para todas las clases

H1: Al menos una tarifa varía gracias a la clase.



```python
#6.9 anova
anova_results2 = stats.f_oneway(
    df[df['Pclass'] == '1 St']['Fare'],
    df[df['Pclass'] == '2 St']['Fare'],
    df[df['Pclass'] == '3 St']['Fare'],
)

# ¿Las medias de las temperaturas maximas son estadisticamente diferentes en estos 4 eventos?
print(f"F-statistic: {anova_results2.statistic}")
print(f"P-value: {anova_results2.pvalue}")


if p_val < 0.05:
    print("Existe dependencia significativa entre Fare y Pclass")
else:
    print("No hay evidencia de dependencia entre Fare y Pclass")
```

47] ✓ 0.0s

```
F-statistic: 199.51520026428136
P-value: 1.8218859891986135e-69
Existe dependencia significativa entre Fare y Pclass
```

Rechazamos la hipótesis nula si el P-Valor < 0.05.

Ya que el P-Valor de la prueba es < 0, rechazamos la hipótesis nula y concluimos de que si hay diferencia en las tarifas si hay un cambio en la clase de boleto comprado

Para estas situaciones de comparaciones entre una variable categórica y una variable cuantitativa, si la variable categórica solamente tiene 2 categorías, el uso de un test si es apropiado. Ya que nuestra variable categórica tiene 3 categorías, se utiliza un ANOVA.

7. Análisis libre (1 punto) • 1 punto se reserva para que ampliemos el análisis: nueva visualización, transformación o exploración de ideas propias.

Al buscar una manera de cuantificar y entender que tanto peso pueden llegar a tener diferentes variables sobre si una persona sobrevivió o no al titanic. Nos planteamos hacer una regresión, esta nos permite entender las probabilidades de supervivencia de un individuo u observación. Un ejemplo de esto podría ser:

- Estimar qué tan significativo o determinante puede llegar a ser el sexo del observado. ¿que tanto dictamina el sexo en las probabilidades de supervivencia? , de ser una diferencia amplia encontrar una explicación de este evento.

En este caso se decide aplicar un modelo PROBIT y LOGIT por encima de un modelo de probabilidad lineal. Se eligieron estos modelos porque la variable dependiente o (Yi) *Survived* es binaria (1 = sobrevivió, 0 = no sobrevivió), por lo tanto una regresión lineal tradicional no sería apropiada ya que los valores predichos pueden exceder el rango [0,1] y los errores no cumplen con los supuestos de homocedasticidad y normalidad, lo que distorsiona la inferencia estadística.

Los modelos elegidos son capaces de transformar la probabilidad mediante combinaciones no lineales y asegurarse que el resultado se encuentre siempre dentro de un rango de 0 a 1.

```
--- LOGIT ---
                           Logit Regression Results
==============================================================================
Dep. Variable:               Survived   No. Observations:                  569
Model:                          Logit   Df Residuals:                      561
Method:                           MLE   Df Model:                            7
Date:                Sun, 12 Oct 2025   Pseudo R-squ.:                  0.3108
Time:                        17:23:47   Log-Likelihood:                -264.59
converged:                       True   LL-Null:                       -383.90
Covariance Type:            nonrobust   LLR p-value:                 7.360e-48
==============================================================================
                 coef    std err          z      P>|z|      [0.025      0.975]
------------------------------------------------------------------------------
const          3.8517      0.535      7.197      0.000       2.803       4.901
Pclass        -0.9278      0.168     -5.536      0.000      -1.256      -0.599
SibSp         -0.0956      0.130     -0.735      0.462      -0.351       0.159
Parch         -0.0374      0.135     -0.278      0.781      -0.301       0.226
Fare          -0.0008      0.003     -0.253      0.800      -0.007       0.006
Sex_male      -2.6344      0.240    -10.964      0.000      -3.105      -2.163
Embarked_Q    -0.6629      0.686     -0.966      0.334      -2.008       0.682
Embarked_S    -0.6078      0.299     -2.033      0.042      -1.194      -0.022
==============================================================================
Accuracy:  0.734  Precision: 0.667  Recall: 0.690  F1: 0.678  AUC: 0.840
Confusion matrix (rows=true, cols=pred):
 [[65 20]
 [18 40]]
```

```
--- PROBIT ---
                          Probit Regression Results
==============================================================================
Dep. Variable:               Survived   No. Observations:                  569
Model:                         Probit   Df Residuals:                      561
Method:                           MLE   Df Model:                            7
Date:                Sun, 12 Oct 2025   Pseudo R-squ.:                  0.3095
Time:                        17:23:47   Log-Likelihood:                -265.09
converged:                       True   LL-Null:                       -383.90
Covariance Type:            nonrobust   LLR p-value:                 1.202e-47
==============================================================================
                 coef    std err          z      P>|z|      [0.025      0.975]
------------------------------------------------------------------------------
const          2.2271      0.303      7.351      0.000       1.633       2.821
Pclass        -0.5221      0.095     -5.489      0.000      -0.709      -0.336
SibSp         -0.0584      0.075     -0.778      0.437      -0.206       0.089
Parch         -0.0153      0.079     -0.193      0.847      -0.171       0.140
Fare          -0.0003      0.002     -0.149      0.881      -0.004       0.004
Sex_male      -1.5603      0.135    -11.519      0.000      -1.826      -1.295
Embarked_Q    -0.4035      0.391     -1.032      0.302      -1.170       0.363
Embarked_S    -0.3513      0.174     -2.024      0.043      -0.692      -0.011
==============================================================================
Accuracy:  0.734  Precision: 0.667  Recall: 0.690  F1: 0.678  AUC: 0.839
Confusion matrix (rows=true, cols=pred):
 [[65 20]
 [18 40]]
```

Al momento de analizar estos dos modelos, no es el mejor punto de referencia fijarse en los coeficientes presentados, la explicación de este fenómeno proviene de las combinaciones no lineales realizadas para la adaptación del modelo a una probabilidad entre 0 y 1. Estos pueden verse desajustados (ya sea inflados o subestimados) en comparación con el modelo de probabilidad lineal.

Los modelos PROBIT y LOGIT nos facilitan ver, gracias a los P-Valores de la variables independiente, si estas variables independientes son significativas para predecir el comportamiento de la variable dependiente. El análisis se comporta de manera similar a un análisis de hipótesis:

H0: La variable independiente no es significativa
H1: La variable independiente es significativa

Rechazamos la hipótesis nula si el P-Valor de la variable es <0.05.

La matriz de confusión por otra parte se encarga de analizar la sensibilidad y precisión del modelo, es decir: que el valor estimado haya sido igual al valor real de la observación. analizamos cuantos errores de tipo uno y dos. Así como los verdaderos negativos y verdaderos positivos.

Podemos organizarla así:

	Predicción: No (0)	**Predicción: Sí (1)**
Real: No (0)	**65** (Verdaderos Negativos)	**20** (Falsos Positivos)
Real: Sí (1)	**18** (Falsos Negativos)	**40** (Verdaderos Positivos)

la manera de analizar la bondad de ajuste del modelo podemos fijarnos en AUC, que representa la capacidad del modelo de discernir entre sobrevivientes y no sobrevivientes, que en este caso es de un 84% lo cual es una cifra más que respetable, todo esto en consideración podemos concluir que el modelo es capaz de hacer buenas predicciones

Anexo:

1)
al momento de abrir el código, ya que usamos una amplia cantidad de librerías, para facilitar la tarea del observador se incluye un archivo de texto llamado **requirements.txt** este archivo debe ser descargado y debe encontrarse en la misma carpeta local que el csv de titanic.

Para correrlo debe abrir una terminal y definir el path donde se encuentran el csv y el requirments, así como el trabajo el comando necesario para correrlo serar:

```
PS C:\Users\leodo> cd "C:\Users\leodo\OneDrive\Escritorio\special topics\proyecto_1"
>> pip install -r requirements.txt
```

1. se escribe el directorio
2. se utiliza el comando: **pip install -r requirements.txt**

en todo caso, de no querer usar este método se adjuntan las librerías usadas

C: > Users > leodo > OneDrive > Escritorio > special topics > proyecto_1 > requirements.txt

```
numpy>=1.26.0
pandas>=2.2.0
scikit-learn>=1.4.0
statsmodels>=0.14.0
matplotlib>=3.8.0

```

2)
El punto 7 de la práctica se encuentra en un archivo aparte, ya que queríamos que la regresión fuera desarrollada con el csv inicial sin variables innecesarias. Estas presentan alto nivel de correlación, ya que provienen de operaciones en otras columnas.

se desarrolló en un pipeline con la ayuda de inteligencia artificial para la estructura y aplicación correcta de test estadísticos

es necesario cambiar el path o directorio para poder observar los resultados de la regresión, en la siguiente parte del código es donde debería cambiarse el directorio del observador

```
if __name__ == "__main__":
    import argparse, os

    parser = argparse.ArgumentParser(add_help=True)
    parser.add_argument("csv", nargs="?", default=None, help="Ruta al CSV")
    args, unknown = parser.parse_known_args()

    # Manejo de argumentos tipo Jupyter (--f=kernel.json)
    if args.csv and args.csv.lower().endswith((".csv", ".csv.gz", ".gz")) and os.path.exists(args.csv):
        csv_path = args.csv
    else:
        csv_path = r"C:\Users\leodo\OneDrive\Escritorio\special topics\proyecto_1\titanic_1.csv"

    print("Usando CSV:", csv_path)
    main(csv_path)
```