# Trabajo grupal o individual

## Identificación del trabajo

a. **Módulo:** 3
b. **Asignatura:** Procesamiento de datos
c. **RA:** Ordena conjuntos de datos aplicando técnicas de procesamiento, para prevenir errores posteriores en el proceso de modelado.
d. **Docente Online:** Gonzalo Esteban Cárdenas Rubio
e. **Fecha de entrega**:

## Identificación del estudiante

| **Nombre y apellido** | **Carrera** |
|---|---|
| Endert Alejandro Guerrero Camacho | Data Science |

## Introducción

En el análisis de datos, es común encontrarse con desafíos como errores ortográficos y formatos incorrectos en los DataFrames. Como en el caso de estudio, en este abordaremos dos DataFrames: "Ventas_Clientes_2021" y "Clientes_2021". En el primero, se identificaron errores en la representación de valores numéricos, mientras que, en el segundo, se detectaron problemas en los nombres de los clientes. El objetivo es resolver estos problemas para así realizar un análisis más preciso y coherente de los datos.

## Desarrollo

# Errores en el DataFrame

***DataFrame "Ventas_Clientes_2021"***

En el índice 2 del DataFrame "Ventas_Clientes_2021" se puede encontrar el primer error, "110,0". Este error se debe a que es una cadena de texto cuando debería ser un número entero, además de tener una coma en vez de un punto para representar la parte decimal del número. Lo mismo sucede con el índice 11, "100,00". Estos errores serán arreglados más adelante.

***DataFrame "Clientes_2021"***

En el índice 0 del DataFrame "Clientes_2021" encontramos el primer error, "JUan" cuando debería ser "Juan". Es un error simple a la hora de escribir el nombre en dicho caso hipotético. Algo similar sucede con los índices 1, "camila", que debería tener mayúscula al iniciar; índice 2, "DanIel", el cual tiene una mayúscula entremedia del nombre, aunque esto no representa problemas en este caso, ya que "Daniel" no se repite. Pero en caso de que fuera así, nos generaría un problema. El índice 8 también presenta un problema similar, "LuiS". Todos estos problemas serán solucionados más adelante.

# Solucionando problemas del DataFrame (Item1)

***DataFrame "Ventas_Clientes_2021"***

Para este dataframe, reemplazaremos la coma por un punto para que este sea representado correctamente y también eliminaremos todas las comillas haciendo uso de "replace". Por último, convertiremos todos estos números en "float".

***DataFrame "Clientes_2021"***

En este dataframe haremos uso de "str.title" con este quitaremos todos los errores, ya que esta función nos pondrá como mayúscula la primera letra de cada línea de texto y el resto de caracteres serán minúsculas.

```
Ventas_Clientes_2021 =
pd.Series([100.0,200.0,"110,0",90.0,150.0,80.0,70.0,100.0,110.0,90.0,"100,00
"])

Ventas_Clientes_2021 = Ventas_Clientes_2021.replace({',': '.', '"':
''},regex=True).astype(float)

print(Ventas_Clientes_2021)



Clientes_2021 = pd.Series
(["JUan","camila","DanIel","Luis","Juan","Ana","Camila","Ana","LuiS","Ana","
Juan"])

Clientes_2021 = Clientes_2021.str.title()

print(Clientes_2021)
```

## Item2

Ahora crearemos un nuevo DataFrame ("df4") donde agruparemos a los clientes y sumaremos sus compras. Este total lo dispondremos de forma descendente para que, al mostrarlo en pantalla, podamos ver cuánto fue el total de las compras realizadas por el cliente organizado jerárquicamente según dicho total.

También, como extra, agregué una variable alojada en "mejor_cliente" y "peor_cliente" que encontrará el nombre del cliente que más compras realizó y al que menos compras realizó durante el 2021 a través de las funciones "idxmax" e "idxmin". Por otro lado, la variable "compras" define la cantidad de compras que hicieron los clientes durante 2021 de una forma similar haciendo uso de "max" y "min".

Por último, se crearon la variable "máximo" que nos entregará el monto máximo de la suma de todas las compras y la variable "mínimo" que nos mostrará lo contrario.

Al final, mostraremos los resultados en pantalla para una mejor dicción.

```
Ventas_Clientes_2021 =
pd.Series([100.0,200.0,"110,0",90.0,150.0,80.0,70.0,100.0,110.0,90.0,"100,00
"])

Ventas_Clientes_2021 = Ventas_Clientes_2021.replace({',': '.', '"':
''},regex=True).astype(float)

Clientes_2021 = pd.Series
(["JUan","camila","DanIel","Luis","Juan","Ana","Camila","Ana","LuiS","Ana","
Juan"])

Clientes_2021 = Clientes_2021.str.title()

df1 = pd.DataFrame(Ventas_Clientes_2021, columns= ["Ventas"])

df2 = pd.DataFrame(Clientes_2021, columns= ["Clientes"])

df3 = pd.concat([df2,df1], axis=1)



df4 = df3.groupby("Clientes").sum()

df4 = df4.sort_values(by="Ventas", ascending = 0)

print(df4)



mejor_cliente = df4["Ventas"].idxmax()

peor_cliente = df4["Ventas"].idxmin()
```

```
compras = df3["Clientes"].value_counts()

max_compras = compras.max()

min_compras = compras.min()



maximo = df4["Ventas"].max()

minimo = df4["Ventas"].min()



print("El cliente que realizó más compras fue", mejor_cliente, "compro",
maximo, "$ divido en", max_compras, "compras. Mientras que", peor_cliente,
"compro", minimo,"$ realizado en", min_compras,"compra")

print("la diferencia de ventas entre", mejor_cliente, "y", peor_cliente, "es
de ", maximo - minimo, "$")
```

```
Luis      200.0
Daniel    110.0
El cliente que realizó más compras fue Juan compro 350.0 $ divido en 3 compras. Mientras que Daniel compro 110.0 $ realizado en 1 compra
la diferencia de ventas entre Juan y Daniel es de  240.0 $
```

## Item3

Ahora crearemos un nuevo DataFrame ("df5"). En este buscamos mostrar la diferencia entre el promedio de las compras realizadas durante 2021 con respecto a la última compra realizada por el cliente en ese mismo año. También encontraremos al cliente con la compra menor con respecto a su promedio a través de las variables "down_cliente" y "down_media" haciendo uso de "idxmin" y "min" respectivamente.

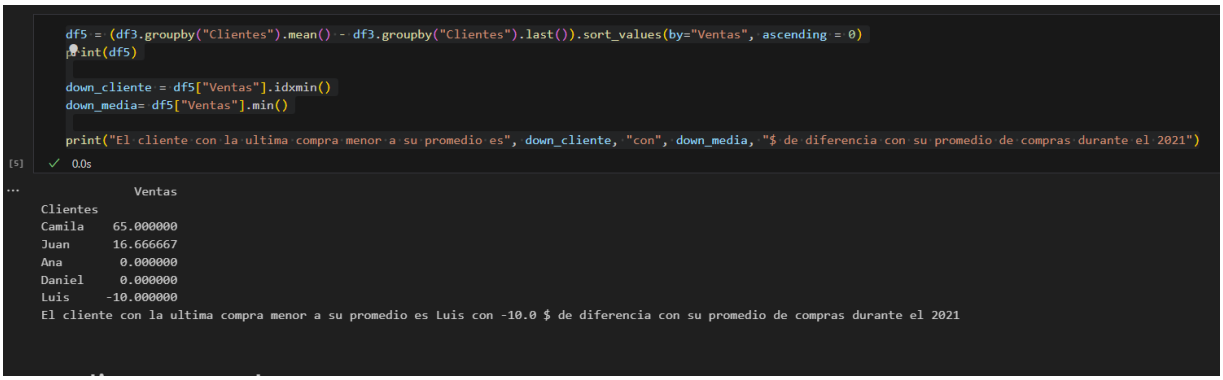
```
df5 = (df3.groupby("Clientes").mean() -
df3.groupby("Clientes").last()).sort_values(by="Ventas", ascending = 0)

print(df5)
```

```
down_cliente = df5["Ventas"].idxmin()

down_media= df5["Ventas"].min()


print("El cliente con la ultima compra menor a su promedio es",
down_cliente, "con", down_media, "$ de diferencia con su promedio de compras
durante el 2021")
```




## Código completo

```python
Ventas_Clientes_2021 = pd.Series([100.0,200.0,"110,0",90.0,150.0,80.0,70.0,100.0,110.0,90.0,"100,00"])
Ventas_Clientes_2021 = Ventas_Clientes_2021.replace({',': '.', '"': ''},regex=True).astype(float)
Clientes_2021 = pd.Series (["JUan","camila","DanIel","Luis","Juan","Ana","camila","Ana","Luis","Ana","Juan"])
Clientes_2021 = Clientes_2021.str.title()
df1 = pd.DataFrame(Ventas_Clientes_2021, columns= ["Ventas"])
df2 = pd.DataFrame(Clientes_2021, columns= ["Clientes"])
df3 = pd.concat([df2,df1], axis=1)

print("Ventas del año 2021")

df4 = df3.groupby("Clientes").sum()
df4 = df4.sort_values(by="Ventas", ascending = 0)
print(df4)

mejor_cliente = df4["Ventas"].idxmax()
peor_cliente = df4["Ventas"].idxmin()

compras = df3["Clientes"].value_counts()
max_compras = compras.max()
min_compras = compras.min()

maximo = df4["Ventas"].max()
minimo = df4["Ventas"].min()

print("El cliente que realizó más compras fue", mejor_cliente, "compro", maximo, "$ divido en", max_compras, "compras. Mientras que", peor_cliente, "compro", minimo,"$ realizado en", min_compras,"compra")
print("la diferencia de ventas entre", mejor_cliente, "y", peor_cliente, "es de ", maximo - minimo, "$")
print("")

print("Diferencia entre el promedio de compras durante el año 2021 con respecto a la ultima compra del mismo año")
df5 = (df3.groupby("Clientes").mean() - df3.groupby("Clientes").last()).sort_values(by="Ventas", ascending = 0)
print(df5)

down_cliente = df5["Ventas"].idxmin()
down_media= df5["Ventas"].min()

print("El cliente con la ultima compra menor a su promedio es", down_cliente, "con", down_media, "$ de diferencia con su promedio de compras durante el 2021")
```

✓ 0.0s

```
Ventas del año 2021
          Ventas
Clientes
Juan       350.0
Ana        270.0
Camila     270.0
Luis       200.0
Daniel     110.0
El cliente que realizó más compras fue Juan compro 350.0 $ divido en 3 compras. Mientras que Daniel compro 110.0 $ realizado en 1 compra
la diferencia de ventas entre Juan y Daniel es de  240.0 $

Diferencia entre el promedio de compras durante el año 2021 con respecto a la ultima compra del mismo año
             Ventas
Clientes
Camila    65.000000
Juan      16.666667
Ana        0.000000
Daniel     0.000000
Luis     -10.000000
El cliente con la ultima compra menor a su promedio es Luis con -10.0 $ de diferencia con su promedio de compras durante el 2021
```

## Conclusión

En este proceso de corrección y mejora de los DataFrames "Ventas_Clientes_2021" y "Clientes_2021", se logró solucionar los errores ortográficos y los formatos incorrectos. Gracias a ello, logramos obtener datos más precisos y confiables, preparados para poder hacer el análisis correctamente.

Además, se crearon nuevos DataFrames, como "df4" y "df5", que proporcionan información relevante sobre las compras realizadas por los clientes en el transcurso del año 2021. La inclusión de variables como "mejor_cliente" y "peor_cliente" agregó un valor adicional al análisis, al identificar los clientes más y menos rentables.

El proceso de corrección y mejora de los DataFrames es fundamental para garantizar la veracidad de los datos y brindar resultados más precisos. El uso adecuado de las funciones y métodos ayudo a resolver los problemas identificados. Este informe para el caso de estudio podría sentar las bases para un análisis más profundo y revelador de las tendencias de ventas y el comportamiento de los clientes durante el año 2021 en caso de contar con más datos.

## Bibliografía


- k3rnel. (2022, 13 julio). *Listas en Python - Somos hackers de la programación*. Somos Hackers de la Programación. Recuperado 29 de julio de 2023, de https://somoshackersdelaprogramacion.es/listas-en-python
- The pandas development team. (2020, febrero). *Pandas.Series.str.title — Pandas 2.0.3 Documentation*. Pandas. Recuperado 29 de julio de 2023, de https://pandas.pydata.org/docs/reference/api/pandas.Series.str.title.html?highlight=str%20title#pandas-series-str-title
- Babu, V. (2019, 12 noviembre). *How to use Regex in Pandas*. kanoki. Recuperado 29 de julio de 2023, de https://kanoki.org/2019/11/12/how-to-use-regex-in-pandas/
- Farooq, A. (2023, 30 enero). Eliminar comillas de cadena en Python. *Delft Stack*. Recuperado 29 de julio de 2023, de https://www.delftstack.com/es/howto/python/python-remove-quotes-from-string/#eliminar-comillas-de-cadena-en-python-usando-el-m%c3%a9todo-replace
- Malli. (2023, 7 marzo). *Pandas Convert string to integer*. Spark By {Examples}. Recuperado 29 de julio de 2023, de https://sparkbyexamples.com/pandas/pandas-convert-string-to-integer/?expand_article=1