

Los datos para este análisis fueron obtenidos del Catálogo de Datos Abiertos de Uruguay: <https://catalogodatos.gub.uy/dataset/miem-remuneraciones> (<https://catalogodatos.gub.uy/dataset/miem-remuneraciones>)

Introducción

Con el conjunto de datos de "Sistema de remuneraciones de los funcionarios del MIEM" (Ministerio de Industria, Energía y Minería), se intenta compartir los comandos básicos (utilizando Python | Google Colab) para realizar un análisis exploratorio sencillo, respondiendo algunas preguntas como:

- 1 - Remuneración mínima
- 2 - Remuneración máxima
- 3 - Total de remuneraciones del MIEM en período (2022)
- 4 - Total de remuneraciones por tipo de cargo
- 5 -Cuál es el tipo de cargo con MENOR remuneración acumulada del período?
- 6 -Cuál es el tipo de cargo con MAYOR remuneración acumulada del período?

Exploraremos el dataframe para saber si es necesario realizar algún proceso de limpieza y reparación de datos, ante de comenzar a responder nuestras preguntas

Librerías

```
In [ ]: import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
```

Lectura del dataset

```
In [ ]: rem = pd.read_excel('remuneraciones-miem-2022.xlsx')
```

Visualización del dataframe "rem" para chequear que quedó bien

```
In [ ]: rem.head() # 5 primeras filas
```

```
In [ ]: rem.tail() # 5 últimas filas
```

Obtener información inicial del set de datos

```
In [ ]: rem.iloc[0] # Primera fila
```

```
In [ ]: rem.iloc[1] # Segunda fila
```

```
In [ ]: rem.iloc[-1] # última fila
```

```
In [ ]: rem.iloc[[0,2,1]] # Primera, segunda y tercera filas
```

```
In [ ]: rem.iloc[:, [6,7,10]] # Sexta, séptima y décima columnas
```

```
In [ ]: # El método nunique() devuelve el número de valores únicos para cada columna.
# Al especificar el eje de la columna (axis='columns'), el método nunique() busca por columnas y devuelve el número de valores únicos para cada fila.
```

```
In [ ]: rem.nunique()
```

```
In [ ]: rem.iloc[:, 0] # Primera columna
```

```
In [ ]: rem.iloc[:, 1] # Segunda columna
```

```
In [ ]: rem.iloc[:, -1] # Última columna
```

```
In [ ]: rem.isnull() # Buscar valores faltantes
```

```
In [ ]: rem.isnull().sum() # Encontrar número de valores faltantes para cada columna
```

Búsqueda de entradas de datos incorrectas

```
In [ ]: for x in rem.columns:
    if rem[x].dtype == 'object':
        print("Nombre de columna:", x)
        print(rem[x].value_counts())
        print("-----")
```

Alguna información mas:

```
In [ ]: print(' - Cantidad de Filas y columnas:', rem.shape, '\n')

print(' - Nombre columnas:', rem.columns, '\n')

print(' - Información del dataframe rem:\n')
rem.info()

In [ ]: rem.duplicated().value_counts # Buscar duplicados

In [ ]: rem.describe() # Cálculo de datos estadísticos de las columnas numéricas (No muestra la información de las columnas categóricas)

In [ ]: rem[2022].describe() # cálculo de datos estadísticos, descripción de una sola columna
# > 1 - Remuneración mínima
# > 2 - Remuneración máxima

In [ ]: # Cambio de nombre de columnas: Nombre Organismo, DENOMINACIÓN DEL CARGO y 2022
rem.rename(columns={'Nombre Organismo': 'Organismo', 'DENOMINACIÓN DEL CARGO': 'Cargo',
                    2022: 'Montos_2022'},
           inplace=True)
rem.columns

In [ ]: # Cambio de todos los encabezados de columnas a minúsculas
rem.columns = map(str.lower, rem.columns)
rem.columns

In [ ]: # Cambio de todos los encabezados de columnas a mayúsculas
rem.columns = map(str.upper, rem.columns)
rem.columns

In [ ]: # Cantidad de registros agrupados por escalafón
rem_agrup_esc = rem.groupby('ESCALAFON').count()
rem_agrup_esc

In [ ]: # Agrupo y sumo a partir de la columna Año 2022
# > 4 - Total de remuneraciones por tipo de cargo

In [ ]: rem_agrup_cargo = rem.groupby('CARGO').agg({'MONTOS_2022': 'sum'})
rem_agrup_cargo.head()

In [ ]: rem_agrup_cargo = rem_agrup_cargo.sort_values(by='MONTOS_2022')
# podemos definir por el tipo de orden que necesitamos con ascending=False por ejemplo

In [ ]: # > 5 -Cuál es el tipo de cargo con MENOR remuneración acumulada del período?
# > 6 -Cuál es el tipo de cargo con MAYOR remuneración acumulada del período?

rem_agrup_cargo

In [ ]: # > 3 - Total de remuneraciones del período 2022
print('Suma total de remuneraciones del período 2022:')
rem_agrup_cargo['MONTOS_2022'].sum()
```

Gráfico

```
In [ ]: rem_agrup_cargo.plot(kind='barh', figsize=(14, 16), color='green')
plt.gca().set_facecolor('0.9')
plt.xticks(range(0, 600000, 50000))
plt.xlabel('SUELDO', fontsize=12).set_color('tomato')
plt.ylabel('NOMBRE DEL CARGO', fontsize=12).set_color('tomato')
plt.title('Remuneraciones por cargo en 2022', fontsize=14).set_color('C0')
plt.grid(axis='x', linestyle='dotted', linewidth=1)
print('Total de remuneraciones por cargo en 2022 \n')
```

Respuestas:

Remuneración mínima: 52823.66

Remuneración máxima: 309439.71

Total de remuneraciones del MIEM en período (2022): 5665274.44

Total de remuneraciones por tipo de cargo: detalle en cuaderno "rem_agrup_cargo"

Cuál es el tipo de cargo con MENOR remuneración acumulada del período?: Oficial IX

Cuál es el tipo de cargo con MAYOR remuneración acumulada del período?: Adscripto al Ministro

```
In [ ]: # Para poder realizar un pdf con la información completa, en otro cuaderno vamos a utilizar el comando: !jupyter nbconvert --to html /content/remuneraciones_MIEM.ipynb
# Desde el cuaderno nuevo subir el archivo .ipynb que queremos transformar en html para luego imprimirlo en pdf y ejecutar el comando
```