Actividad - Estadística básica

- **Nombre:** Sebastian Armando Flores
- **Matrícula:** 01709229

Entregar: Archivo PDF de la actividad, así como el archivo .ipynb en tu repositorio. **Nota:** Recuerda habrá una penalización de **50** puntos si la actividad fue entregada fuera de la fecha límite.

Carga el conjunto de datos `insurance.csv` (se encuentra en el repositorio de la clase) y realiza un análisis estadístico de las variables.

```
# Carga las librerías necesarias.
import pandas as pd
import numpy as np
import random
```

```
# Carga el conjunto de datos al ambiente de Google Colab y muestra los primeros

from google.colab import files

uploaded = files.upload()

for fn in uploaded.keys():
  print('User uploaded file "{name}" with length {length} bytes'.format(
      name=fn, length=len(uploaded[fn])))

# 6 renglones.
df = pd.read_csv('insurance.csv')
df.head(6)
```

El conjunto de datos contiene información demográfica sobre los asegurados en una compañía de seguros:

- **age**: Edad del asegurado principal
- **sex**: Género del asegurado. female o male
- **bmi**: Índice de masa corporal
- **children**: Número de hijos que estan cubiertos con la poliza.
- **smoke**: ¿El beneficiario fuma? (*yes*/*no*)
- **region**: ¿Dónde vive el beneficiario? Estos datos son de Estados Unidos. Regiones disponibles: northeast, southeast, southwest, northwest
- **charges**: Costo del seguro.

```
# Crea una tabla resumen con los estadísticas generales de las variables
# numéricas.
#shape nos ayuda para ver las dimensiones o el tamano de la tabla
print('Dimensiones: ',df.shape)
print('Informacion de cada dato: ',df.info())
# describe nos da resultados sobre mediociones importantes como la cantidad de valores, media
print('Mas informacion: ',df.describe())
```

```
Dimensiones:  (1338, 7)
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1338 entries, 0 to 1337
Data columns (total 7 columns):
 #   Column    Non-Null Count  Dtype  
---  ------    --------------  -----  
 0   age       1338 non-null   int64  
 1   sex       1338 non-null   object 
 2   bmi       1338 non-null   float64
 3   children  1338 non-null   int64  
 4   smoker    1338 non-null   object 
 5   region    1338 non-null   object 
 6   charges   1338 non-null   float64
dtypes: float64(2), int64(2), object(3)
memory usage: 73.3+ KB
Informacion de cada dato:  None
Mas informacion:                 age          bmi     children       charges
count  1338.000000  1338.000000  1338.000000   1338.000000
mean     39.207025    30.663397     1.094918  13270.422265
std      14.049960     6.098187     1.205493  12110.011237
min      18.000000    15.960000     0.000000   1121.873900
25%      27.000000    26.296250     0.000000   4740.287150
50%      39.000000    30.400000     1.000000   9382.033000
75%      51.000000    34.693750     2.000000  16639.912515
max      64.000000    53.130000     5.000000  63770.428010
```

```
# ¿Cómo se correlacionan las varaibles numéricas entre sí?
# aqui se esta midiendo la desviacion entre los valores de una misma columna, para ver que ta
print('Desviación estándar edad: ', df['age'].std())
print('Desviación estándar bmi: ', df['bmi'].std())
print('Desviación estándar children: ', df['children'].std())
print('Desviación estándar bmi: ', df['charges'].std())

# correlacion pearson para toda la tabla
# esta test de correlacion arroja un valor entre 0 y 1 dependiendo de que tan cercanos o popu

print('matriz de correlacion: ',df.corr())
```

```
Desviación estándar edad:  14.049960379216172
Desviación estándar bmi:  6.098186911679017
Desviación estándar children:  1.2054927397819095
Desviación estándar bmi:  12110.011236693994
matriz de correlacion:                 age       bmi  children   charges
age       1.000000  0.109272  0.042469  0.299008
bmi       0.109272  1.000000  0.012759  0.198341
children  0.042469  0.012759  1.000000  0.067998
charges   0.299008  0.198341  0.067998  1.000000
```

```
# Determina si existe o no una correlación entre el índice de masa corporal
# (bmi) y el costo del seguro.

# seleccionamos que 'categorias' queremos medir
selected = df[['bmi', 'charges']]
#selected.head(1000)

# se usaran los tres metodos para medir correlacion entre mbi y charges
print('Correlación Pearson: ', selected['bmi'].corr(selected['charges'], method='pearson'))
print('Correlación spearman: ', selected['bmi'].corr(selected['charges'], method='spearman'))
print('Correlación kendall: ', selected['bmi'].corr(selected['charges'], method='kendall'))
```

```
Correlación Pearson:  0.19834096883362895
Correlación spearman:  0.11939590358331145
Correlación kendall:  0.08252397079981415
```

```
# ¿Cuántas personas aseguradas son hombre y cuántas son mujeres?
# df.groupby(['sex']).count()

# tambien funciona
#df['sex'].value_counts()

df.groupby(['sex']).count()[['charges']]
```

```
# En este caso todas las personas estan aseguradas, esto lo podemos observa si
# vemos lo valoes unico en charge, nos damos cuenta que el valor mas pequeno es
# arriba de mil, enotnces todos estan asegurados'
```

```
# ¿Cuántos hombres y mujeres asegurados viven en cada región?

# tambien funciona este
#pd.crosstab(df['sex'], df['region'])

df.groupby(['sex', 'region']).count()[['charges']]

# En este caso todas las personas estan aseguradas, esto lo podemos observa si
# vemos lo valoes unico en charge, nos damos cuenta que el valor mas pequeno es
# arriba de mil, enotnces todos estan asegurados'
```

```
# En promedio, ¿quién paga más de cuota de seguro? ¿Los fumadores o los no
# fumadores? Muéstralo con los datos.
df.groupby(['smoker']).mean()[['charges']]

# nos interesa usar 'mean' para ver el promedio del costo respecto a si fuman o no
# como podemos
```

```
# ¿Cuáles son las cuotas mínimas y máximas que las personan pagan dependiendo
# del género y del número de hijos?
df.groupby(['sex', 'children']).agg(['min', 'max'])[['charges']]
```

```
# ¿Cuál es el índice de masa corporal promedio para hombre y mujeres dependiendo
# región en la que viven y si son fumadores? ¿Impacta eso en la tarifa del
# seguro?
#df.groupby(['bmi', 'sex', 'region', 'smoker']).mean()
df.groupby(['region', 'smoker','sex']).mean()[['bmi', 'charges']]
```

✓ 0s completed at 11:24 AM