Desafío - Gráficos y correlación

- Para realizar este desafío debes haber estudiado previamente todo el material disponibilizado correspondiente a la unidad.
- Una vez terminado el desafío, comprime la carpeta que contiene el desarrollo de los requerimientos solicitados y sube el `.zip` en el LMS.
- Desarrollo desafío:
    - El desafío se debe desarrollar de manera Individual.
    - Para la realización del desafío necesitarás apoyarte del archivo Apoyo Desafío - Gráficos y correlación.

1. Importar librerías y `.csv`

- Importe las librerías básicas para el análisis de datos
- Descargue e importe el archivo `nations.csv`.
- ***Tip**: El formato del archivo posee una codificación de tipo `'iso-8859-1'`. Investigue cómo se puede solucionar este problema utilizando el argumento encoding de `pd.read_csv`.*

La base de datos contiene información a nivel mundial sobre demografía:

- `country:` País.
- `region:` Continente del país.
- `gdp:` Producto Interno Bruto per cápita, precios 2005.
- `school:` Promedio años de escolaridad.
- `adfert:` Fertilidad adolescente (Nacimientos 1:1000 en mujeres entre 15 y 19).
- `chldmort`: Probabilidad de muerte antes de los 5 años por cada 1000.
- `life:` Esperanza de vida al nacer.
- `pop:` Población total.
- `urban:` Porcentaje de población urbana.
- `femlab:` Tasa entre hombres y mujeres en el mercado laboral.
- `literacy:` Tasa de alfabetismo.
- `co2:` Toneladas de Co2 mitidas per cápita.
- `gini:` Coeficiente de desigualdad del ingreso.
- Apellidos desde la A hasta la N: Enfocarse en las variables `chldmort`, `adfert` y `life`.
- Apellidos desde la M hasta la Z: Enfocarse en las variables `femlab`, `literacy` y `school`.

2. Refactor gráficos matplotlib a seaborn

A continuación se presenta una serie de gráficos construidos con `matplotlib`. Se le pide refactorizarlos utilizando seaborn.

- Se presenta la función que se utilizó para construirlos. Intente llegar al resultado con mayor similitud. Comente los principales resultados de los gráficos.

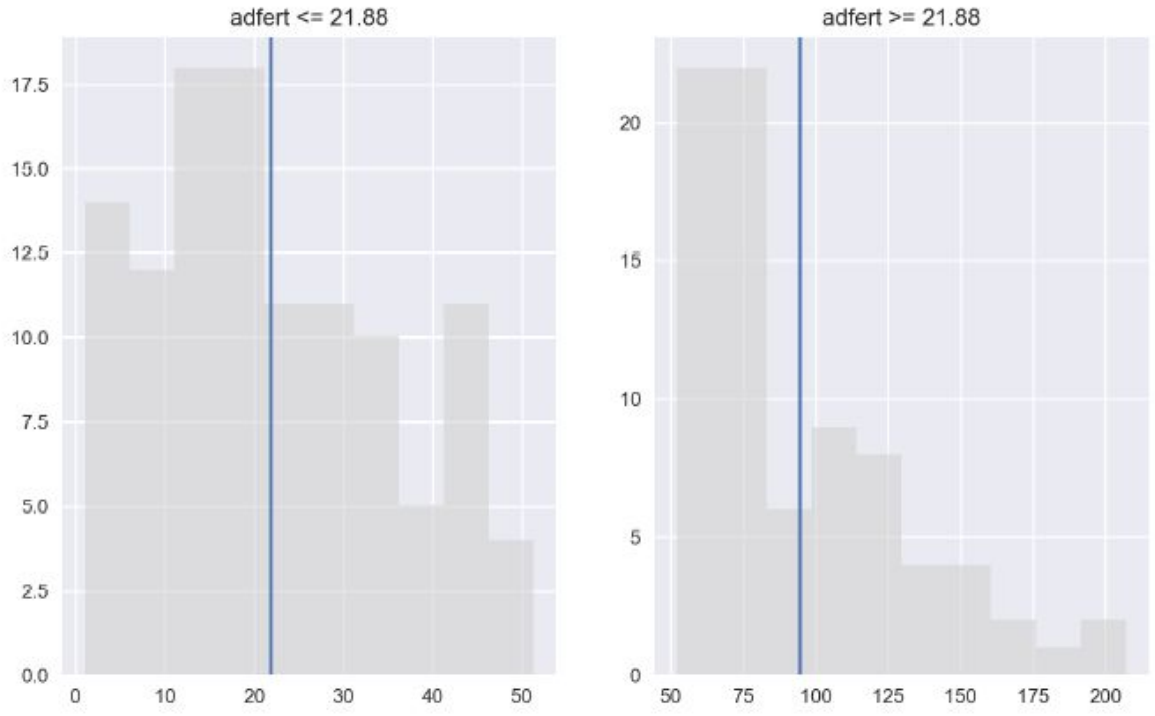
a) Histogramas

```
def binarize_histogram(dataframe, variable):
   tmp = dataframe
   tmp['binarize'] = np.where(tmp[variable] > np.mean(tmp[variable]), 1,
0)

   hist_1 = tmp[tmp['binarize'] == 1][variable].dropna()
   hist_0 = tmp[tmp['binarize'] == 0][variable].dropna()

   plt.subplot(1, 2, 1)
   plt.hist(hist_0, alpha=.6, color='lightgrey')
   plt.axvline(np.mean(hist_0))
   plt.title("{0} <= {1}".format(variable, round(np.mean(hist_0), 3)))
   plt.subplot(1, 2, 2)
   plt.hist(hist_1, alpha=.6, color='lightgrey')
   plt.axvline(np.mean(hist_1))
   plt.title("{0} >= {1}".format(variable, round(np.mean(hist_0), 3)))

binarize_histogram(df, 'adfert')
```



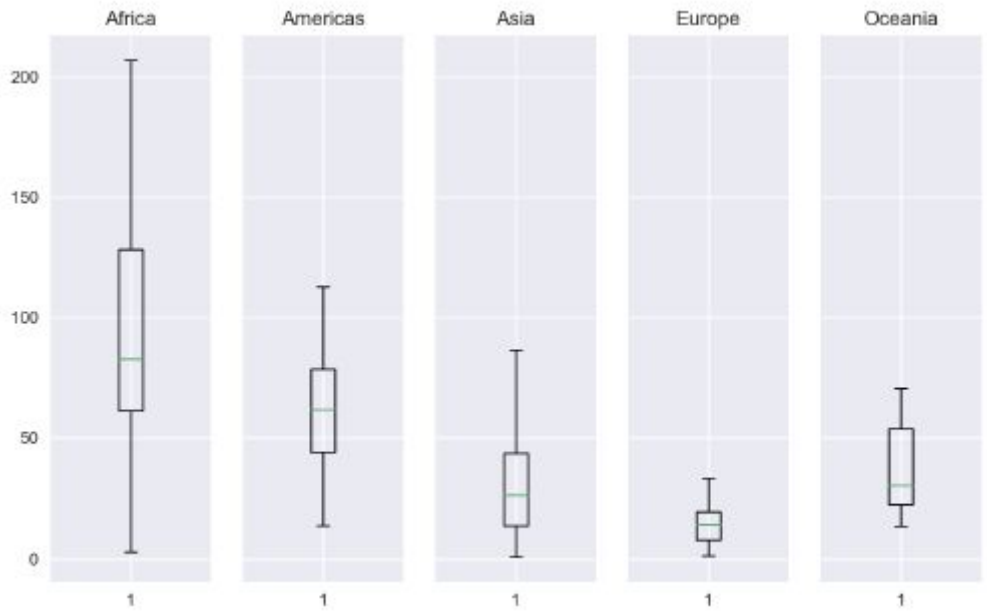
b) Boxplots

```
def grouped_boxplot(dataframe, variable, group_by):
    tmp = dataframe
    stratify_by = tmp[group_by].unique()

    if len(stratify_by) / 2 > 3:
        fig, ax = plt.subplots(2, len(stratify_by),sharey=True)
    else:
        fig, ax = plt.subplots(1, len(stratify_by),sharey=True)

    for i, n in enumerate(stratify_by):
        ax[i].boxplot(tmp[tmp[group_by] == n][variable])
        ax[i].set_title(n)

grouped_boxplot(df, 'adfert', 'region')
```


Africa
Americas
Asia
Europe
Oceania
200
150
100
50
0
1
1
1
1
1

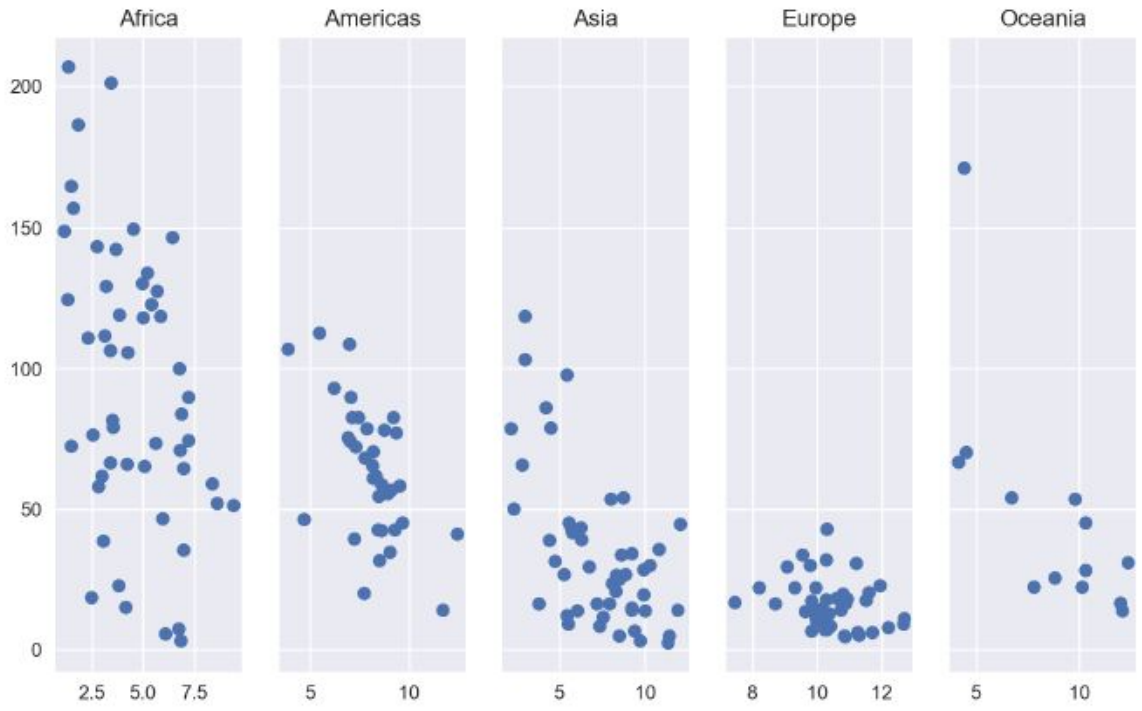
c) Scatterplots

```
def grouped_scatterplot(dataframe, x, y, group_by):
    tmp = dataframe
    stratify_by = tmp[group_by].unique()

    if len(stratify_by) / 2 > 3:
        fig, ax = plt.subplots(2, len(stratify_by),sharey=True)
    else:
        fig, ax = plt.subplots(1, len(stratify_by),sharey=True)

    for i, n in enumerate(stratify_by):
        tmp_group_plt = tmp[tmp[group_by] == n]
        ax[i].plot(tmp_group_plt[x], tmp_group_plt[y], 'o')
        ax[i].set_title(n)

grouped_scatterplot(df, 'school', 'adfert', 'region')
```



3. Genere un heatmap entre todas las variables.

- En base a las variables de interés asignadas, comente cuáles son las principales correlaciones existentes, tomando como criterio de corte aquellas superior a .6

4. En base a las principales correlaciones, sepárelas en un nuevo objeto y calcule la matriz de correlaciones para todas las regiones

- **Tip:** Genere una nueva tabla segmentando con la siguiente sintaxis: `= df.loc[:,['variables', 'a', 'agregar']]. N`. No olvide agregar la variable region.
- **Tip:** Genere un loop para recorrer cada región y generar un heatmap.
- Comente brevemente las principales correlaciones a través de las regiones.

Bonus Points: Grafique los diagramas de dispersión para los principales hallazgos.