Solución Desafío - Variables

Desarrollo Total	Solución Desafío - Variables

Requerimiento 1

Genere una muestra de casos

Solución

```
%matplotlib inline
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import scipy.stats as stats

plt.style.use('seaborn-whitegrid') # Gráficos estilo seaborn
plt.rcParams["figure.figsize"] = (10, 6) # Tamaño gráficos
plt.rcParams["figure.dpi"] = 200 # resolución gráficos
df = pd.read_csv("qog_std_cs_jan18.csv")

df['ht_region'] = df['ht_region'].replace([1, 2, 3, 4, 5, 6, 7, 8, 9,
10],
                                          ['EastEurope', 'LatAm',
'NorthAfrica','SubSaharian',
                                            'WesternDem',
'EastAsia','SouthEastAsia','SouthAsia',
                                            'Pacific', 'Caribbean' ])

subsample_am = df.sample(frac=.5, random_state=8945)
subsample_nz = df.sample(frac=.5, random_state=8945)

subsample_am = subsample_am.loc[:, ['cname', 'ccodealp','ht_region',
'gle_cgdpc', 'undp_hdi',
                                    'imf_pop', 'ffp_hf', 'wef_qes',
```

```
'wdi_expedu',
                                    'wdi_ners']]
subsample_nz = subsample_nz.loc[:, ['cname', 'ccodealp','ht_region',
'gle_cgdpc', 'undp_hdi',
                                     'imf_pop', 'wef_imort',
'who_alc2000', 'who_tobt', 'wdi_exph'
                                    ]]
```

Requerimiento 2

Genere una función que ingrese su objeto y devuelva:

- *Por cada variable existente en su objeto, calcule las medidas descriptivas para los casos continuos.*
- *Para cada variable discreta, que calcule la frecuencia.*
- *Reporte las estadísticas descriptivas para `gle_cgdpc`, `undp_hdi`, `imf_pop`.*
  Compare las estadísticas con algún compañero. ¿Ve alguna diferencia sustancial en alguna de ellas?

Solución

```
def fetch_descriptives(dataframe):
   for key, value in dataframe.iteritems():
       print(value.describe())
```

```
fetch_descriptives(subsample_am)
fetch_descriptives(subsample_nz)
```

```
count         97
unique         97
top       Liberia
freq            1
Name: cname, dtype: object
count     97
unique     97
top       MRT
freq       1
Name: ccodealp, dtype: object
count             97
```

```
unique             10
top       SubSaharian
freq               27
Name: ht_region, dtype: object
```

```
count       96.000000
mean     16307.974571
std       20044.856128
min         488.269990
25%       2971.567500
50%       9255.265150
75%       21168.739500
max     104049.440000
Name: gle_cgdpc, dtype: float64
count   91.000000
mean     0.705846
std       0.151497
min       0.414000
25%       0.575500
50%       0.734000
75%       0.838500
max       0.948000
Name: undp_hdi, dtype: float64
count     73.000000
mean     26.047151
std       42.982955
min       0.011000
25%       3.369000
50%       8.140000
75%       29.746000
max     202.768997
Name: imf_pop, dtype: float64
count   87.000000
mean     5.241379
std       2.128738
min       1.500000
25%       3.450000
50%       5.500000
75%       7.050000
```

```
max       9.200000
Name: ffp_hf, dtype: float64
```

```
count   73.000000
mean     3.860967
std      0.928082
min      2.218669
25%      3.089141
50%      3.711759
75%      4.546944
max      5.986858
Name: wef_qes, dtype: float64
count   71.000000
mean     4.627481
std      1.665997
min      1.925660
25%      3.092540
50%      4.766260
75%      5.594055
max      8.627110
Name: wdi_expedu, dtype: float64
count   67.000000
mean    73.440645
std     21.546894
min     19.439980
25%     63.632561
50%     80.177567
75%     90.678089
max     99.573357
Name: wdi_ners, dtype: float64
count         97
unique        97
top       Liberia
freq           1
Name: cname, dtype: object
count     97
unique    97
```

```
top       MRT
freq       1
Name: ccodealp, dtype: object
count             97
unique            10
top       SubSaharian
freq              27
Name: ht_region, dtype: object


count       96.000000
mean     16307.974571
std       20044.856128
min         488.269990
25%       2971.567500
50%       9255.265150
75%       21168.739500
max     104049.440000
Name: gle_cgdpc, dtype: float64
count   91.000000
mean     0.705846
std       0.151497
min       0.414000
25%       0.575500
50%       0.734000
75%       0.838500
max       0.948000
Name: undp_hdi, dtype: float64
count     73.000000
mean     26.047151
std       42.982955
min       0.011000
25%       3.369000
50%       8.140000
75%       29.746000
max     202.768997
Name: imf_pop, dtype: float64
count     73.000000
mean     24.438830
std       25.108657
min       1.700000
25%       4.700000
50%       14.400000
75%       40.900002
```

```
max     117.400002
Name: wef_imort, dtype: float64









count   91.000000
mean     5.090440
std      3.673995
min      0.000000
25%      1.860000
50%      5.030000
75%      8.115000
max     13.940000
Name: who_alc2000, dtype: float64
count   67.000000
mean    23.310448
std      8.628637
min      7.400000
25%     18.550000
50%     22.600000
75%     26.349999
max     54.000000
Name: who_tobt, dtype: float64
count   94.000000
mean     6.919864
std      3.013875
min      2.067386
25%      4.766251
50%      6.394859
75%      8.971661
max     17.135723
Name: wdi_exph, dtype: float64
```

Requerimiento 3

Genere una función que liste las observaciones perdidas de una variable

Solución

```
def fetch_null_cases(dataframe, var, print_list=False):
    tmp = dataframe
    tmp['flagnull'] = tmp[var].isnull()
    count_na = 0

    for i, r in tmp.iterrows():
        if r['flagnull'] is True:
            count_na += 1
            if print_list is True:
                print( r['cname'])

    print("\nCasos perdidos para {0}:\nCantidad de Casos: {1}\nPorcentaje
de la muestra {2}".format(var, count_na, count_na/len(tmp)))
    if print_list is True:
        print("Países sin registros de {0}\n".format(var))
```

```
for i in subsample_am.columns:
    fetch_null_cases(subsample_am, i, print_list=False)
```

```
Casos perdidos para cname:
Cantidad de Casos: 0
Porcentaje de la muestra 0.0

Casos perdidos para ccodealp:
Cantidad de Casos: 0
Porcentaje de la muestra 0.0

Casos perdidos para ht_region:
Cantidad de Casos: 0
Porcentaje de la muestra 0.0

Casos perdidos para gle_cgdpc:
Cantidad de Casos: 1
Porcentaje de la muestra 0.010309278350515464

Casos perdidos para undp_hdi:
Cantidad de Casos: 6
Porcentaje de la muestra 0.061855670103092786

Casos perdidos para imf_pop:
Cantidad de Casos: 24
Porcentaje de la muestra 0.24742268041237114

Casos perdidos para ffp_hf:
Cantidad de Casos: 10
Porcentaje de la muestra 0.10309278350515463

Casos perdidos para wef_qes:
Cantidad de Casos: 24
Porcentaje de la muestra 0.24742268041237114

Casos perdidos para wdi_expedu:
Cantidad de Casos: 26
Porcentaje de la muestra 0.26804123711340205

Casos perdidos para wdi_ners:
Cantidad de Casos: 30
Porcentaje de la muestra 0.30927835051546393
```

```
for i in ['wdi_ners', 'wdi_expedu', 'wef_qes']:
    fetch_null_cases(subsample_am, i, print_list=True)
```

```
United Arab Emirates
Togo
Papua New Guinea
Azerbaijan
Kiribati
Cote dIvoire
San Marino
Congo
St Vincent and the Grenadines
Equatorial Guinea
Sudan (2012-)
Gabon
Kenya
South Africa
Botswana
Andorra
Sierra Leone
Solomon Islands
Liberia
Zambia
Maldives
Somalia
Nigeria
China
Taiwan
Turkmenistan
Vietnam
Czech Republic
Singapore
Austria

Casos perdidos para wdi_ners:
Cantidad de Casos: 30
Porcentaje de la muestra 0.30927835051546393
Países sin registros de wdi_ners
```

```
United Arab Emirates
Papua New Guinea
Kiribati
Uzbekistan
Congo
St Vincent and the Grenadines
Equatorial Guinea
Sudan (2012-)
Marshall Islands
Botswana
Philippines
Solomon Islands
Grenada
Lesotho
Kuwait
Greece
Zambia
Tuvalu
Eritrea
Somalia
Nigeria
China
Taiwan
Myanmar
Nauru
Jordan

Casos perdidos para wdi_expedu:
Cantidad de Casos: 26
Porcentaje de la muestra 0.26804123711340205
Países sin registros de wdi_expedu
```

```
Togo
Papua New Guinea
Kiribati
Uzbekistan
San Marino
Congo
St Vincent and the Grenadines
Equatorial Guinea
Sudan (2012-)
Sao Tome and Principe
Marshall Islands
Liechtenstein
Andorra
Solomon Islands
Grenada
Tuvalu
Maldives
Eritrea
Somalia
Comoros
Belarus
Turkmenistan
Nauru
St Lucia

Casos perdidos para wef_qes:
Cantidad de Casos: 24
Porcentaje de la muestra 0.24742268041237114
Países sin registros de wef_qes
```

```
for i in subsample_nz.columns:
    fetch_null_cases(subsample_nz, i, print_list=False)
```

```
Casos perdidos para cname:
Cantidad de Casos: 0
Porcentaje de la muestra 0.0

Casos perdidos para ccodealp:
Cantidad de Casos: 0
Porcentaje de la muestra 0.0

Casos perdidos para ht_region:
Cantidad de Casos: 0
Porcentaje de la muestra 0.0

Casos perdidos para gle_cgdpc:
Cantidad de Casos: 1
Porcentaje de la muestra 0.010309278350515464

Casos perdidos para undp_hdi:
Cantidad de Casos: 6
Porcentaje de la muestra 0.061855670103092786

Casos perdidos para imf_pop:
Cantidad de Casos: 24
Porcentaje de la muestra 0.24742268041237114

Casos perdidos para wef_imort:
Cantidad de Casos: 24
Porcentaje de la muestra 0.24742268041237114

Casos perdidos para who_alc2000:
Cantidad de Casos: 6
Porcentaje de la muestra 0.061855670103092786

Casos perdidos para who_tobt:
Cantidad de Casos: 30
Porcentaje de la muestra 0.30927835051546393

Casos perdidos para wdi_exph:
Cantidad de Casos: 3
Porcentaje de la muestra 0.030927835051546393
```

```
for i in ['who_tobt', 'wef_imort', 'imf_pop']:
    fetch_null_cases(subsample_nz, i, print_list=True)
```

```
United Arab Emirates
Togo
Papua New Guinea
Cote d'Ivoire
Peru
San Marino
St Vincent and the Grenadines
Equatorial Guinea
Sudan (2012-)
Guatemala
Gabon
Madagascar
Sao Tome and Principe
Marshall Islands
Botswana
Liechtenstein
Qatar
Solomon Islands
Grenada
Bhutan
Kuwait
Tuvalu
Maldives
Eritrea
Somalia
Zimbabwe
Taiwan
Turkmenistan
Austria
St Lucia

Casos perdidos para who_tobt:
Cantidad de Casos: 30
Porcentaje de la muestra 0.30927835051546393
Países sin registros de who_tobt
```

```
Togo
Papua New Guinea
Kiribati
Uzbekistan
San Marino
Congo
St Vincent and the Grenadines
Equatorial Guinea
Sudan (2012-)
Sao Tome and Principe
Marshall Islands
Liechtenstein
Andorra
Solomon Islands
Grenada
Tuvalu
Maldives
Eritrea
Somalia
Comoros
Belarus
Turkmenistan
Nauru
St Lucia

Casos perdidos para wef_imort:
Cantidad de Casos: 24
Porcentaje de la muestra 0.24742268041237114
Países sin registros de wef_imort
```

```
Costa Rica
United Arab Emirates
Congo
Equatorial Guinea
Sudan (2012-)
Guatemala
Gabon
Albania
Madagascar
Sao Tome and Principe
Botswana
Liechtenstein
Andorra
Grenada
Lesotho
Liberia
Zambia
Eritrea
Somalia
Honduras
China
India
Turkmenistan
Cameroon

Casos perdidos para imf_pop:
Cantidad de Casos: 24
Porcentaje de la muestra 0.24742268041237114
Países sin registros de imf_pop
```

Requerimiento 4

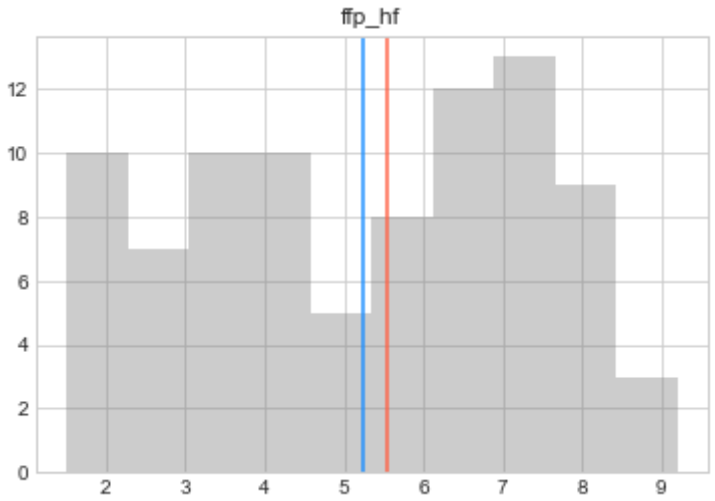
Grafique histogramas indicando medias muestral y total.

Solución

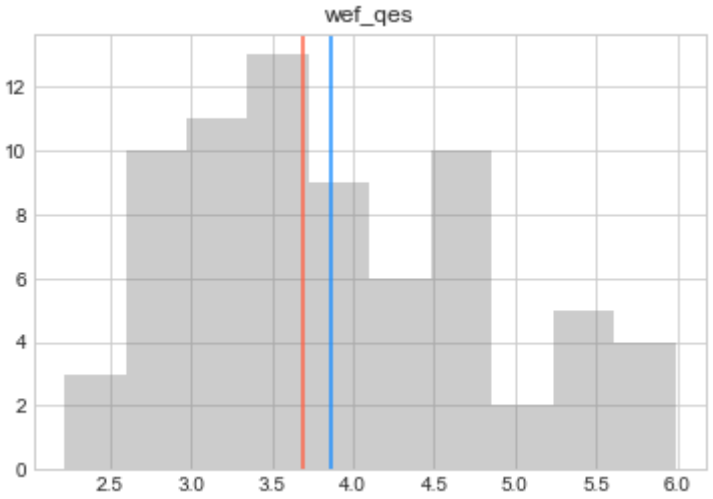
```
def plot_hist(sample_df, full_df, var, sample_mean=False,
true_mean=False):
    tmp = sample_df[var].dropna()
    plt.hist(tmp, color='grey', alpha=.4)
    plt.title(var)

    if sample_mean is True:
        plt.axvline(np.mean(tmp), color='dodgerblue')
    if true_mean is True:
        plt.axvline(np.mean(full_df[var]), color='tomato')
```

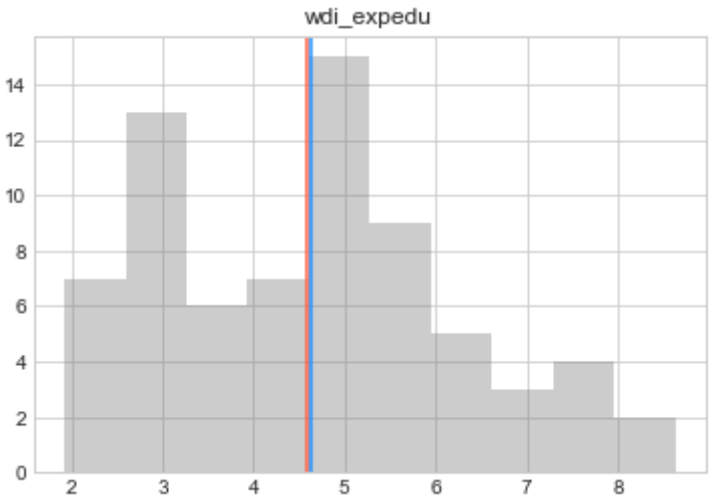
```
plot_hist(subsample_am, df, 'ffp_hf', sample_mean=True, true_mean=True)
```



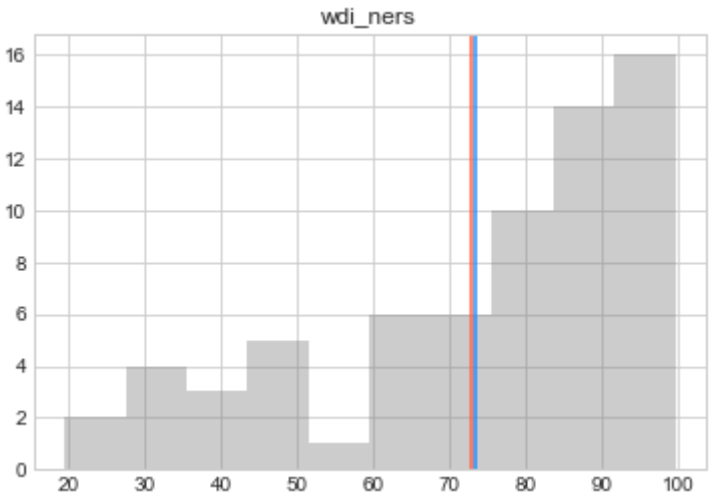

```
plot_hist(subsample_am, df, 'wef_qes', sample_mean=True, true_mean=True)
```




```
plot_hist(subsample_am, df, 'wdi_expedu', sample_mean=True,
true_mean=True)
```




```
plot_hist(subsample_am, df, 'wdi_ners', sample_mean=True,
true_mean=True)
```


wdi_ners
16
14
12
10
8
6
4
2
0
20
30
40
50
60
70
80
90
100

Requerimiento 5

Genere una función que devuelva un dotplot con las medias por región para una variable entregada.

Solución

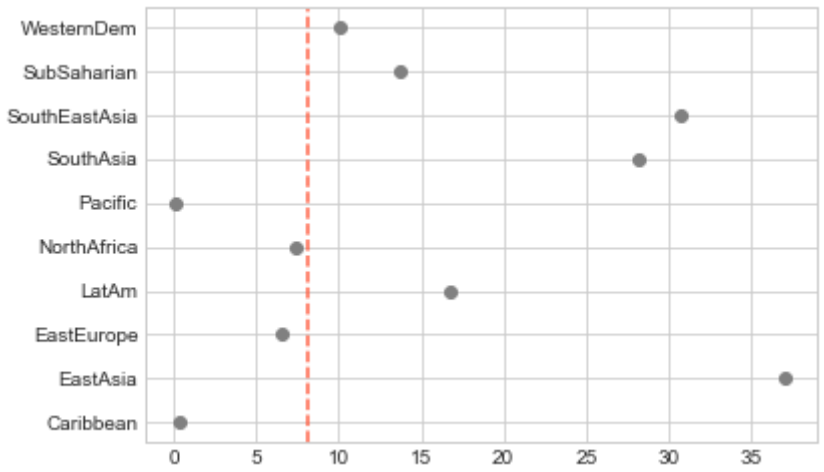
```
def dotplot(df, plot_var, plot_by, global_stat = False, statistic =
'mean'):
   tmp_df = df.loc[:, [plot_by, plot_var]]

   if statistic is 'mean':
       tmp_group_stat = tmp_df.groupby(plot_by)[plot_var].mean()
   if statistic is 'median':
       tmp_group_stat = tmp_df.groupby(plot_by)[plot_var].median()

   plt.plot(tmp_group_stat.values, tmp_group_stat.index, 'o',
color='grey')

   if global_stat is True and  statistic is 'mean':
       plt.axvline(df[plot_var].mean(), color='tomato', linestyle='--')
   if global_stat is True and statistic is 'median':
       plt.axvline(df[plot_var].median(), color='tomato',
linestyle='--')
```

```
dotplot(df, plot_var='imf_pop', plot_by='ht_region',global_stat=True,
statistic='median')
```



Requerimiento 5

Guarde la base de datos.

Solución

```
subsample_am.to_csv("subsample_am_demo.csv", na_rep='NaN')
subsample_nz.to_csv("subsample_nz_demo.csv", na_rep='NaN')
```