```
%matplotlib inline
import matplotlib.pyplot as plt
import numpy
import pandas
import seaborn
import statistics as stats


# Let's install Tensorflow 2.0:
!pip install tensorflow==2.0.0

# And verify that it is now in its latest version:
import tensorflow as tf
print(tf.__version__)
```

```
Looking in indexes: https://pypi.org/simple, https://us-python.pkg.dev/colab-wheels/
Requirement already satisfied: tensorflow==2.0.0 in /usr/local/lib/python3.7/dist-pa
Requirement already satisfied: absl-py>=0.7.0 in /usr/local/lib/python3.7/dist-packa
Requirement already satisfied: opt-einsum>=2.3.2 in /usr/local/lib/python3.7/dist-pa
Requirement already satisfied: termcolor>=1.1.0 in /usr/local/lib/python3.7/dist-pac
Requirement already satisfied: wheel>=0.26 in /usr/local/lib/python3.7/dist-packages
Requirement already satisfied: keras-preprocessing>=1.0.5 in /usr/local/lib/python3.
Requirement already satisfied: tensorboard<2.1.0,>=2.0.0 in /usr/local/lib/python3.7
Requirement already satisfied: wrapt>=1.11.1 in /usr/local/lib/python3.7/dist-packag
Requirement already satisfied: numpy<2.0,>=1.16.0 in /usr/local/lib/python3.7/dist-p
Requirement already satisfied: protobuf>=3.6.1 in /usr/local/lib/python3.7/dist-pack
Requirement already satisfied: grpcio>=1.8.6 in /usr/local/lib/python3.7/dist-packag
Requirement already satisfied: astor>=0.6.0 in /usr/local/lib/python3.7/dist-package
Requirement already satisfied: six>=1.10.0 in /usr/local/lib/python3.7/dist-packages
Requirement already satisfied: google-pasta>=0.1.6 in /usr/local/lib/python3.7/dist-
Requirement already satisfied: keras-applications>=1.0.8 in /usr/local/lib/python3.7
Requirement already satisfied: gast==0.2.2 in /usr/local/lib/python3.7/dist-packages
Requirement already satisfied: tensorflow-estimator<2.1.0,>=2.0.0 in /usr/local/lib/
Requirement already satisfied: h5py in /usr/local/lib/python3.7/dist-packages (from
Requirement already satisfied: werkzeug>=0.11.15 in /usr/local/lib/python3.7/dist-pa
Requirement already satisfied: google-auth-oauthlib<0.5,>=0.4.1 in /usr/local/lib/py
Requirement already satisfied: markdown>=2.6.8 in /usr/local/lib/python3.7/dist-pack
Requirement already satisfied: requests<3,>=2.21.0 in /usr/local/lib/python3.7/dist-
Requirement already satisfied: google-auth<2,>=1.6.3 in /usr/local/lib/python3.7/dis
Requirement already satisfied: setuptools>=41.0.0 in /usr/local/lib/python3.7/dist-p
Requirement already satisfied: cachetools<5.0,>=2.0.0 in /usr/local/lib/python3.7/di
Requirement already satisfied: pyasn1-modules>=0.2.1 in /usr/local/lib/python3.7/dis
Requirement already satisfied: rsa<5,>=3.1.4 in /usr/local/lib/python3.7/dist-packag
Requirement already satisfied: requests-oauthlib>=0.7.0 in /usr/local/lib/python3.7/
Requirement already satisfied: importlib-metadata>=4.4 in /usr/local/lib/python3.7/d
Requirement already satisfied: zipp>=0.5 in /usr/local/lib/python3.7/dist-packages (
Requirement already satisfied: typing-extensions>=3.6.4 in /usr/local/lib/python3.7/
Requirement already satisfied: pyasn1<0.5.0,>=0.4.6 in /usr/local/lib/python3.7/dist
Requirement already satisfied: chardet<4,>=3.0.2 in /usr/local/lib/python3.7/dist-pa
Requirement already satisfied: certifi>=2017.4.17 in /usr/local/lib/python3.7/dist-p
Requirement already satisfied: idna<3,>=2.5 in /usr/local/lib/python3.7/dist-package
Requirement already satisfied: urllib3!=1.25.0,!=1.25.1,<1.26,>=1.21.1 in /usr/local
Requirement already satisfied: oauthlib>=3.0.0 in /usr/local/lib/python3.7/dist-pack
Requirement already satisfied: cached-property in /usr/local/lib/python3.7/dist-pack
2.0.0
```

```
from google.colab import drive
drive.mount('/content/drive')
import os
os.chdir("/content/drive/MyDrive/pdatos")
!ls
```

```
Drive already mounted at /content/drive; to attempt to forcibly remount, call drive.
dis2.csv  dis_b.csv  disb.csv
```

```
import numpy as np
import pandas as pd

#fetal = pd.read_table("/content/drive/MyDrive/pdatos/dis_b.data", header=None)
#print(fetal.head)
data = pd.read_csv("/content/drive/MyDrive/pdatos/data_t.csv")
X=data[0:8]
y=data['class']
#print(y)
print(X)
from sklearn.model_selection import train_test_split, ShuffleSplit
from sklearn.neural_network import MLPClassifier
#model = MLPClassifier(hidden_layer_sizes=(2126,) , max_iter=1000)
#model.fit(X, y)
```

```
   age sex on thyroxine query on thyroxine on antithyroid medication sick  \
0   41   F            f                  f                          f    f
1   23   F            f                  f                          f    f
2   46   M            f                  f                          f    f
3   70   F            t                  f                          f    f
4   70   F            f                  f                          f    f
5   18   F            t                  f                          f    f
6   59   F            f                  f                          f    f
7   80   F            f                  f                          f    f

  pregnant thyroid surgery I131 treatment query hypothyroid  ... TT4 measured  \
0        f               f             f                 f  ...             t
1        f               f             f                 f  ...             t
2        f               f             f                 f  ...             t
3        f               f             f                 f  ...             t
4        f               f             f                 f  ...             t
5        f               f             f                 f  ...             t
6        f               f             f                 f  ...             t
7        f               f             f                 f  ...             t

   TT4 T4U measured   T4U FTI measured  FTI TBG measured TBG referral source  \
0  125            t  1.14            t  109            f   ?            SVHC
1  102            f     ?            f    ?            f   ?           other
2  109            t  0.91            t  120            f   ?           other
3  175            f     ?            f    ?            f   ?           other
4   61            t  0.87            t   70            f   ?             SVI
5  183            t   1.3            t  141            f   ?           other
6   72            t  0.92            t   78            f   ?           other
7   80            t   0.7            t  115            f   ?             SVI

       class
```

```
0  negative.
1  negative.
2  negative.
3  negative.
4  negative.
5  negative.
6  negative.
7  negative.

[8 rows x 30 columns]
```

```
#Informacion del dataset
data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 2800 entries, 0 to 2799
Data columns (total 30 columns):
 #   Column                     Non-Null Count  Dtype 
---  ------                     --------------  ----- 
 0   age                        2800 non-null   int64 
 1   sex                        2800 non-null   object
 2   on thyroxine               2800 non-null   object
 3   query on thyroxine         2800 non-null   object
 4   on antithyroid medication  2800 non-null   object
 5   sick                       2800 non-null   object
 6   pregnant                   2800 non-null   object
 7   thyroid surgery            2800 non-null   object
 8   I131 treatment             2800 non-null   object
 9   query hypothyroid          2800 non-null   object
 10  query hyperthyroid         2800 non-null   object
 11  lithium                    2800 non-null   object
 12  goitre                     2800 non-null   object
 13  tumor                      2800 non-null   object
 14  hypopituitary              2800 non-null   object
 15  psych                      2800 non-null   object
 16  TSH measured               2800 non-null   object
 17  TSH                        2800 non-null   object
 18  T3 measured                2800 non-null   object
 19  T3                         2800 non-null   object
 20  TT4 measured               2800 non-null   object
 21  TT4                        2800 non-null   object
 22  T4U measured               2800 non-null   object
 23  T4U                        2800 non-null   object
 24  FTI measured               2800 non-null   object
 25  FTI                        2800 non-null   object
 26  TBG measured               2800 non-null   object
 27  TBG                        2800 non-null   object
 28  referral source            2800 non-null   object
 29  class                      2800 non-null   object
dtypes: int64(1), object(29)
memory usage: 656.4+ KB
```

```
# Calcular media mediana y moda
```

```
data['age'].describe()
```

```
count    2800.000000
```

```
mean       51.686786
std        18.994013
min         1.000000
25%        36.000000
50%        54.000000
75%        67.000000
max        94.000000
Name: age, dtype: float64
```

```
data['referral source'].describe()
data['age'].dtype
```

```
dtype('int64')
```

```
data.groupby(by=['class','referral source','age']).describe()
```

			sex				on thyroxine				query thyro
			count	unique	top	freq	count	unique	top	freq	count
class	referral source	age									
discordant.	**STMW**	**2**	1	1	F	1	1	1	f	1	1
		18	1	1	F	1	1	1	f	1	1
		45	1	1	?	1	1	1	t	1	1
	SVHC	**30**	1	1	F	1	1	1	f	1	1
		36	1	1	M	1	1	1	f	1	1
...	**...**	**...**	...	...	...	...	...	...	...	...	...
negative.	**other**	**88**	1	1	F	1	1	1	f	1	1
		89	2	1	F	2	2	1	f	2	2
		90	2	2	F	1	2	1	f	2	2
		92	1	1	F	1	1	1	t	1	1
		93	1	1	F	1	1	1	f	1	1

322 rows × 108 columns

```
data.groupby(by=['class','referral source','age','tumor']).describe()
```

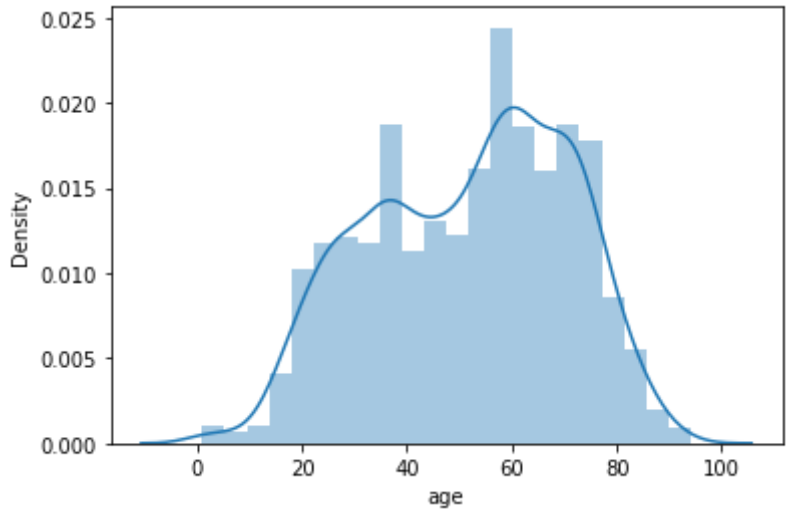
				sex				on thyroxine			
				count	unique	top	freq	count	unique	top	freq
class	referral source	age	tumor								
discordant.	**STMW**	**2**	**f**	1	1	F	1	1	1	f	1
		18	**f**	1	1	F	1	1	1	f	1
		45	**f**	1	1	?	1	1	1	t	1
	SVHC	**30**	**f**	1	1	F	1	1	1	f	1
		36	**f**	1	1	M	1	1	1	f	1
...	**...**	**...**	**...**	...	...	...	...	...	...	...	...
negative.	**other**	**88**	**f**	1	1	F	1	1	1	f	1
		89	**f**	2	1	F	2	2	1	f	2
		90	**f**	2	2	F	1	2	1	f	2
		92	**t**	1	1	F	1	1	1	t	1
		93	**f**	1	1	F	1	1	1	f	1

```
data.groupby(['age']).min()
```

sex	on thyroxine	query on thyroxine	on antithyroid	sick	pregnant	thyroid surgery	I131 treatment	hyp

```
seaborn.distplot(data.age.dropna(axis=0, how='all'))
```

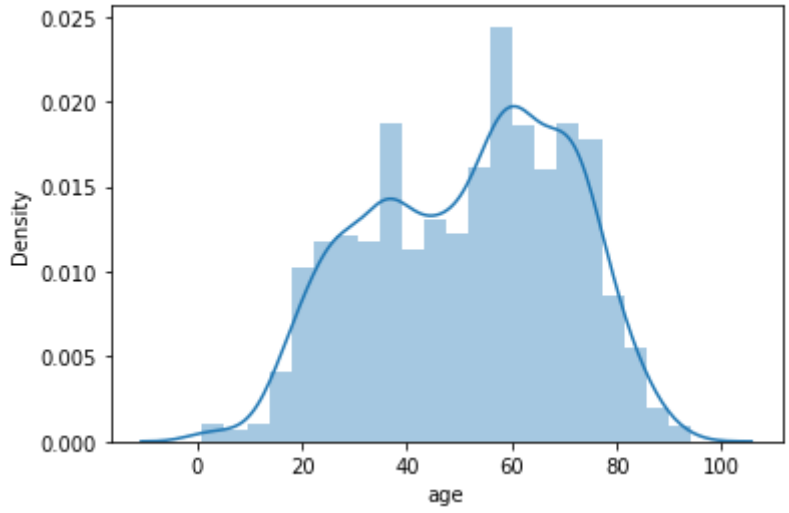
```
/usr/local/lib/python3.7/dist-packages/seaborn/distributions.py:2619: FutureWarning:
  warnings.warn(msg, FutureWarning)
<matplotlib.axes._subplots.AxesSubplot at 0x7f5f91053cd0>
```



En la figura se puede apreciar que a las personas al rededor de los 21 años son los mas afectados por la Diabetes

```
seaborn.distplot(data.age.dropna(axis=0, how='all'))
```

```
/usr/local/lib/python3.7/dist-packages/seaborn/distributions.py:2619: FutureWarning:
  warnings.warn(msg, FutureWarning)
<matplotlib.axes._subplots.AxesSubplot at 0x7f5f90f0f650>
```



Es la dispercion de datos en la que se centran mas o en la que se agrtupan mas datos

```
# calculo de la media de Años
print(stats.mean(data['age']))
```

```
51.68678571428571
```

```
#calculo de la mediana de Años
print(stats.median(data['age']))
```

```
54.0
```

```
#calculo de la moda de Años
print(stats.mode(data['age']))
```

```
59
```

Moda, Mediana y media no coinciden en la variable Longitud, con lo cual para esta variable hay asimetria positiva.

```
data['age'].describe()
```

```
count    2800.000000
mean       51.686786
std        18.994013
min         1.000000
25%        36.000000
50%        54.000000
75%        67.000000
max        94.000000
Name: age, dtype: float64
```

La forma mas facil de saber todo esto es con la funcion .describe()

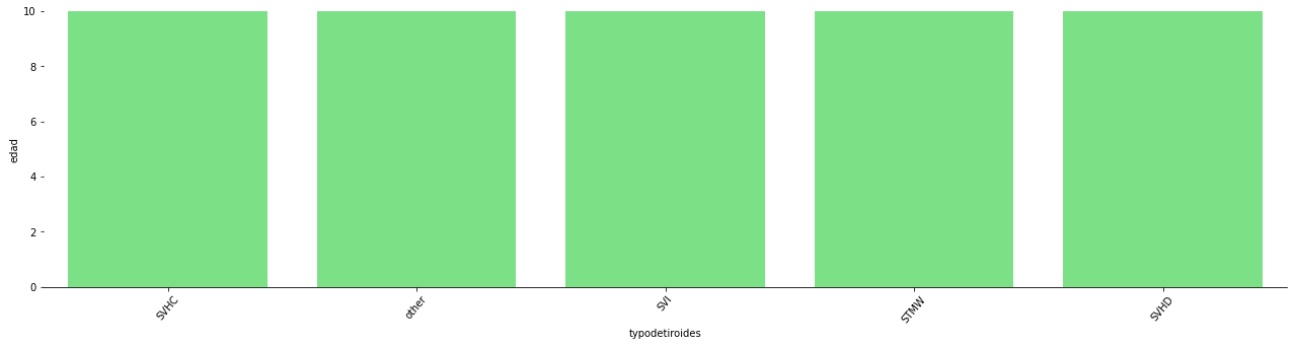
```
BLUE = '#35A7FF'
RED = '#FF5964'
GREEN = '#6BF178'
YELLOW = '#FFE74C'
```

```
data.describe()
```

age

```
plt.figure(figsize=(22,5))
seaborn.barplot(data=data, x='referral source', y='age',color=GREEN)
plt.ylabel('edad')
plt.xlabel('typodetiroides')
plt.ylim(0, 10)
seaborn.despine(left=True)
plt.xticks(rotation=50)
```

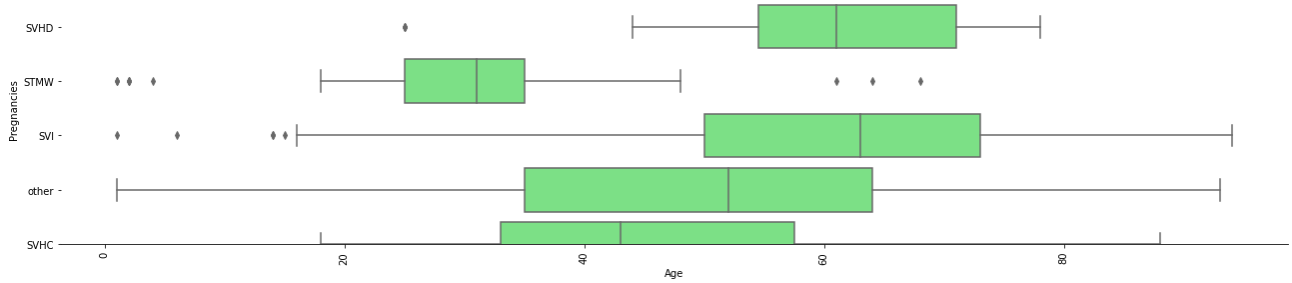
```
(array([0, 1, 2, 3, 4]), <a list of 5 Text major ticklabel objects>)
```



Aqui podemos observar la tabla de tipos y edades

```
plt.figure(figsize=(22,5))
seaborn.boxplot(data=data, x='age', y='referral source', color=GREEN)
plt.ylabel('Pregnancies')
plt.xlabel('Age')
plt.ylim(0, 5)
seaborn.despine(left=True)
plt.xticks(rotation=90)
```

```
(array([-20.,   0.,  20.,  40.,  60.,  80., 100.]),
 <a list of 7 Text major ticklabel objects>)
```



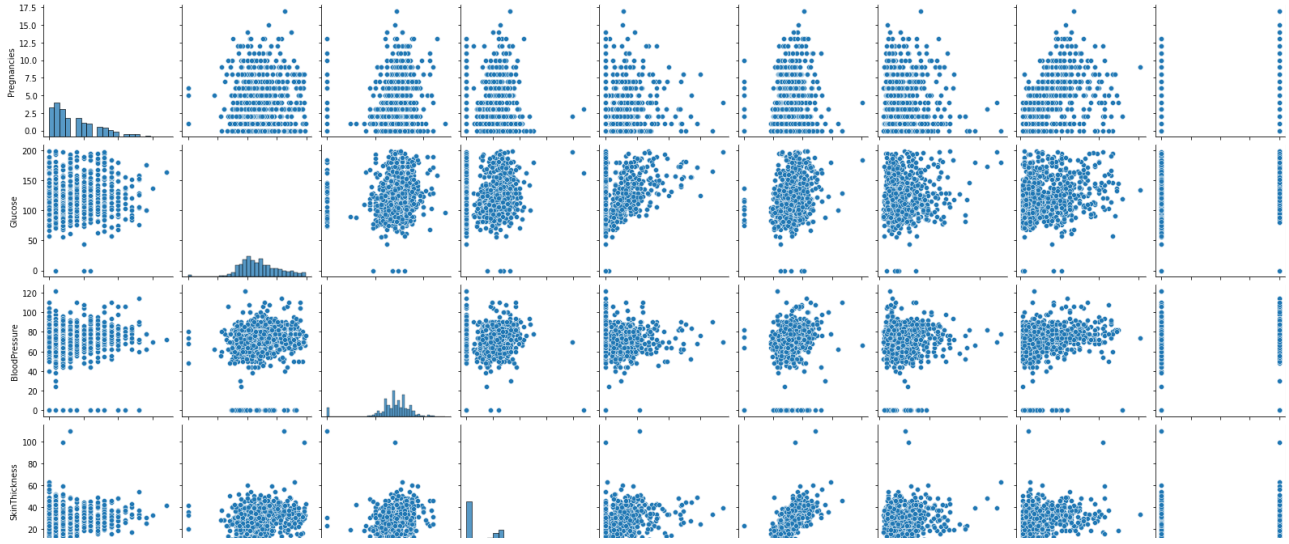

```
data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 2800 entries, 0 to 2799
Data columns (total 30 columns):
 #   Column                     Non-Null Count  Dtype
---  ------                     --------------  -----
 0   age                        2800 non-null   int64
 1   sex                        2800 non-null   object
 2   on thyroxine               2800 non-null   object
```

```
 3   query on thyroxine         2800 non-null   object
 4   on antithyroid medication  2800 non-null   object
 5   sick                       2800 non-null   object
 6   pregnant                   2800 non-null   object
 7   thyroid surgery            2800 non-null   object
 8   I131 treatment             2800 non-null   object
 9   query hypothyroid          2800 non-null   object
 10  query hyperthyroid         2800 non-null   object
 11  lithium                    2800 non-null   object
 12  goitre                     2800 non-null   object
 13  tumor                      2800 non-null   object
 14  hypopituitary              2800 non-null   object
 15  psych                      2800 non-null   object
 16  TSH measured               2800 non-null   object
 17  TSH                        2800 non-null   object
 18  T3 measured                2800 non-null   object
 19  T3                         2800 non-null   object
 20  TT4 measured               2800 non-null   object
 21  TT4                        2800 non-null   object
 22  T4U measured               2800 non-null   object
 23  T4U                        2800 non-null   object
 24  FTI measured               2800 non-null   object
 25  FTI                        2800 non-null   object
 26  TBG measured               2800 non-null   object
 27  TBG                        2800 non-null   object
 28  referral source            2800 non-null   object
 29  class                      2800 non-null   object
dtypes: int64(1), object(29)
memory usage: 656.4+ KB
```

```
important_cols =data[['class','age','sex','referral source']]
imp_cols2=data[['sex','T3','TT4','T4U','referral source','class']]


seaborn.pairplot(
    data=important_cols,
           markers='o')
```

```
<seaborn.axisgrid.PairGrid at 0x7f557bb0a750>
```

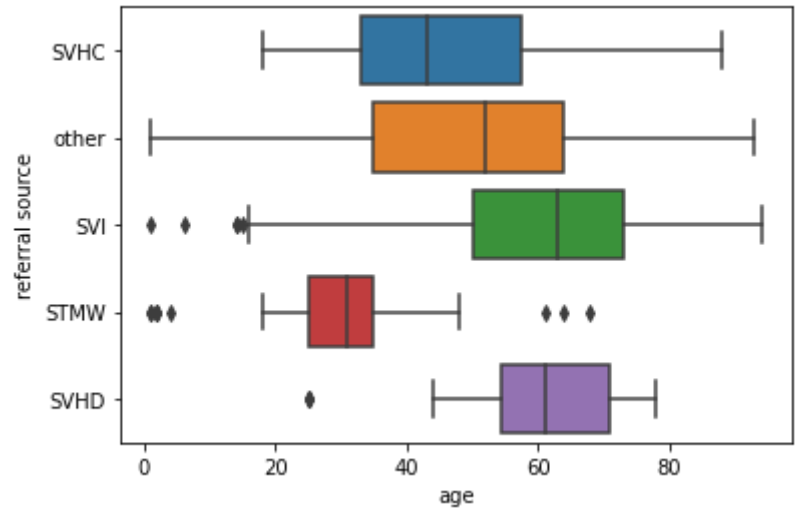



```
#a nivel numerico computar las correlaciones

import pandas as pd
import numpy as np

def get_corrs(important_cols):
    col_correlations = important_cols.corr()
    col_correlations.loc[:, :] = np.tril(col_correlations, k=-1)
    cor_pairs = col_correlations.stack()
    return cor_pairs.to_dict()

my_corrs = get_corrs(important_cols)
print(my_corrs)
```

```
{('age', 'age'): 0.0}
```

```
import seaborn as sns
sns.boxplot( x=data["age"], y=data["referral source"] )
```

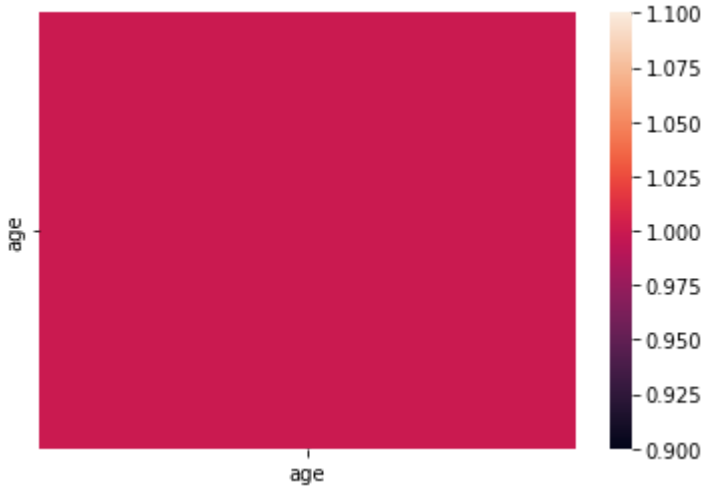
```
<matplotlib.axes._subplots.AxesSubplot at 0x7f5f8e7a0390>
```




```
import seaborn as sns
corr = important_cols.corr()
sns.heatmap(corr,
```

```
            xticklabels=corr.columns.values,
            yticklabels=corr.columns.values)
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7f5f8e69a850>
```



Correlaciones vistas en este caso con los valores de r

Correlaciones que se marcan en el heatmap: principalmente entre Age - Pregnacies y Glucose

```
## calculamos spearman's correlaciones
from scipy import stats
coef, p = stats.spearmanr(important_cols['age'].dropna(), important_cols['referra
print('Spearmans correlacion coeficiente al: %.3f' % coef)
# Interpretando el significado
alpha = 0.05
if p > alpha:
  print('Las muestras no están correlacionadas (fallo para rechazar H0) p=%.3f' %
else:
  print('Las muestras están correlacionadas (rechazar H0) p=%.3f' % p)
```

```
    Spearmans correlacion coeficiente al: -0.046
    Las muestras están correlacionadas (rechazar H0) p=0.016
```

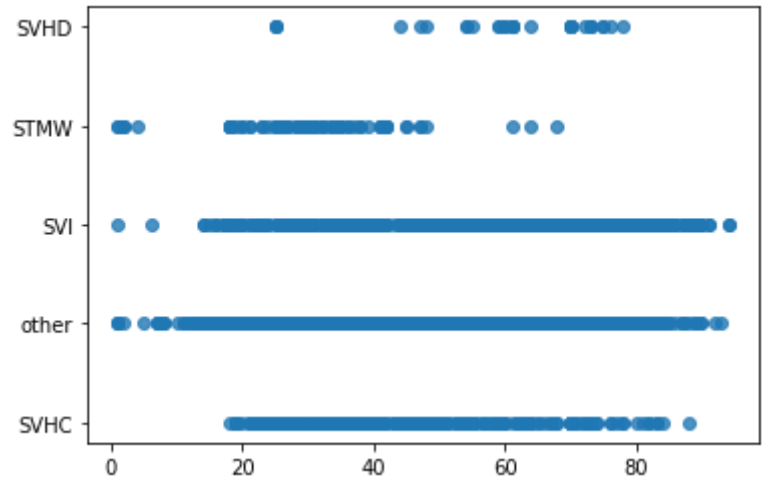
```
## calculamos kendall's correlacion
from scipy import stats
coef, p = stats.kendalltau(important_cols['age'].dropna(), important_cols['referr
print('kendall correlacion coeficiente al: %.3f' % coef)
# interpret the significance
alpha = 0.05
if p > alpha:
  print('Las muestras no están correlacionadas (fallo para rechazar H0) p=%.3f' %
else:
  print('Las muestras están correlacionadas (rechazar H0) p=%.3f' % p)
```

```
kendall correlacion coeficiente al: -0.036
Las muestras están correlacionadas (rechazar H0) p=0.015
```

```
import seaborn as sns
seaborn.regplot(data=important_cols,
                x='age', y='referral source',
                marker='o')
seaborn.despine()
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
<ipython-input-206-a049d3de3573> in <module>()
      2 seaborn.regplot(data=important_cols,
      3                 x='age', y='referral source',
----> 4                 marker='o')
      5 seaborn.despine()

                                  6 frames

/usr/local/lib/python3.7/dist-packages/seaborn/regression.py in reg_func(_x, _y)
    232         """Low-level regression and prediction using linear algebra."""
    233         def reg_func(_x, _y):
--> 234             return np.linalg.pinv(_x).dot(_y)
    235
    236         X, y = np.c_[np.ones(len(self.x)), self.x], self.y

TypeError: can't multiply sequence by non-int of type 'float'
```

SEARCH STACK OVERFLOW



Z1 sigue la dirección en la que las observaciones varían más (linea azul). La proyección de cada observación sobre esa dirección equivale al valor de la primera componente para dicha observación

0 s completado a las 15:31