```
# -*- coding: utf-8 -*-
"""
Created on Sat Oct 31 20:45:05 2015


http://stanford.edu/~mwaskom/software/seaborn/tutorial/distributions.html#plotti
http://stanford.edu/~mwaskom/software/seaborn/generated/seaborn.distplot.html#se
http://stanford.edu/~mwaskom/software/seaborn/generated/seaborn.regplot.html?hig

@author: baobab
"""

import pandas;
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt


data = pandas.read_csv('../codebooks/marscrater_pds.csv', low_memory=False)

# Convert variables to numeric
data['LONGITUDE_CIRCLE_IMAGE'] = data['LONGITUDE_CIRCLE_IMAGE'].convert_objects(
data['LATITUDE_CIRCLE_IMAGE'] = data['LATITUDE_CIRCLE_IMAGE'].convert_objects(co
data['DIAM_CIRCLE_IMAGE'] = data['DIAM_CIRCLE_IMAGE'].convert_objects(convert_nu

# ---------- LONGITUDE_CIRCLE_IMAGE ---------
print('Describe LONGITUDE_CIRCLE_IMAGE:')
desc1 = data['LONGITUDE_CIRCLE_IMAGE'].describe()
print(desc1)

# Plot the longitude distribution wtih two bins - 20 and 50
sns.distplot(data['LONGITUDE_CIRCLE_IMAGE'], bins=20, kde=False, rug=False);
plt.xlabel('Crater Center Longitude')
plt.title('Longitude from the derived center of a non-linear least-squares circl

sns.distplot(data['LONGITUDE_CIRCLE_IMAGE'], bins=50, kde=False, rug=False);
plt.xlabel('Crater Center Longitude')
plt.title('Longitude from the derived center of a non-linear least-squares circl

# ---------- LATITUDE_CIRCLE_IMAGE ---------
print('Describe LATITUDE_CIRCLE_IMAGE:')
desc1 = data['LATITUDE_CIRCLE_IMAGE'].describe()
print(desc1)

print('mode   : {%f}' % data['LATITUDE_CIRCLE_IMAGE'].mode())
print('spread : {%f}' % (data['LATITUDE_CIRCLE_IMAGE'].max() - data['LATITUDE_CI
print('std dev: {%f}' % (data['LATITUDE_CIRCLE_IMAGE'].std()))

# Plot the latitude distribution wtih two bins - 20 and 50
sns.distplot(data['LATITUDE_CIRCLE_IMAGE'], bins=20, kde=False, rug=False);
plt.xlabel('Crater Center Latitude')
plt.title('Latitude from the derived center of a non-linear least-squares circle


sns.distplot(data['LATITUDE_CIRCLE_IMAGE'], bins=50, kde=False, rug=False);
plt.xlabel('Crater Center Latitude')
```

```
plt.title('Latitude from the derived center of a non-linear least-squares circle


# ---------- DIAM_CIRCLE_IMAGE ---------
print('Describe DIAM_CIRCLE_IMAGE:')
desc1 = data['DIAM_CIRCLE_IMAGE'].describe()
print(desc1)

sns.distplot(data["DIAM_CIRCLE_IMAGE"], kde=False);
plt.xlabel('Crater Diameter')
plt.title('Diameter from a non-linear least squares circle fit')


# Data Mamagement for crater diameter - filter only craters not wider than 4 km
craters4 = data[ data["DIAM_CIRCLE_IMAGE"] <= 4 ]


sns.distplot( craters4["DIAM_CIRCLE_IMAGE"],kde=False);
plt.xlabel('Crater Diameter')
plt.title('Diameter from a non-linear least squares circle fit. For craters with

# Plot the circle diameter with 100 bins
sns.distplot( craters4["DIAM_CIRCLE_IMAGE"],kde=False,bins=100);
plt.xlabel('Crater Diameter')
plt.title('Diameter from a non-linear least squares circle fit. For craters with

# Is there a dependency between the latitude and the longitude?
sns.regplot(x="LATITUDE_CIRCLE_IMAGE", y="LONGITUDE_CIRCLE_IMAGE", data=data, fi
plt.title("Dependency Between Crater Center Latitude and Crater Center Longitude
plt.xlabel("Crater Center Latitude")
plt.ylabel("Crater Center Longitude")

# Let's try a heat map.
# First we build a 2d histogram
heatmap, xedges, yedges = np.histogram2d(data["LATITUDE_CIRCLE_IMAGE"], data["LO
extent = [xedges[0], xedges[-1], yedges[0], yedges[-1]]

# Inspect heatmap
print("Inspect heatmap:")
heatmap

plt.clf()
plt.figure(figsize=(60, 60), dpi=96)
plt.imshow(heatmap, extent=extent)
plt.title("Crater Latitude-Longitude Dependency as Heat Map")
plt.xlabel("Crater Center Latitude")
plt.ylabel("Crater Center Longitude")
plt.show()
```