Python - Data cleaning and preprocessing

This assignment to exercise cleaning and preparing data for data mining. You are required to **code, run, and answer the following**:

1. Using Anaconda (or any Python compiler), install the following packages as follows:

```
import pandas as pd
import numpy as np
%matplotlib inline
import matplotlib.pyplot as plt
import matplotlib.image as mpimg
from sklearn.decomposition import PCA
```

2. Give a brief description of each package of ten packages above.
3. Load breast-cancer-wisconsin dataset as follows:

```
data = pd.read_csv('https://archive.ics.uci.edu/ml/machine-learning-databases/breast-cancer-wisconsin/breast-cancer-wisconsin.data', header=None)
data.columns = ['Sample code', 'Clump Thickness', 'Uniformity of Cell Size', 'Uniformity of Cell Shape','Marginal Adhesion', 'Single Epithelial Cell Size', 'Bare Nuclei', 'Bland Chromatin','Normal Nucleoli', 'Mitoses','Class']
data = data.drop(['Sample code'],axis=1)
```

4. Briefly describe Iris dataset.

```
print('Number of instances = %d' % (data.shape[0]))
print('Number of attributes = %d' % (data.shape[1]))
data.head()
```

5. Show the output of the above code and elaborate on the output.

```
data = data.replace('?',np.NaN)
print('Number of instances = %d' % (data.shape[0]))
print('Number of attributes = %d' % (data.shape[1]))
print('Number of missing values:')
for col in data.columns:
    print('\t%s: %d' % (col,data[col].isna().sum()))
```

6. Show the output of the above code and elaborate on the output.

```
data2 = data['Bare Nuclei']
print('Before replacing missing values:')
print(data2[20:25])
data2 = data2.fillna(data2.median())
print('\nAfter replacing missing values:')
print(data2[20:25])
print('Number of rows in original data = %d' % (data.shape[0]))
data2 = data.dropna()
```

```
print('Number of rows after discarding missing values = %d' % (data2.shape[0]))
```

7. Show the output of the above code and elaborate on the output.

```
data2 = data.drop(['Class'],axis=1)
data2['Bare Nuclei'] = pd.to_numeric(data2['Bare Nuclei'])
data2.boxplot(figsize=(20,3))
```

8. Show the output of the above code and elaborate on the output.

```
Z = (data2-data2.mean())/data2.std()
Z[20:25]
```

9. Show the output of the above code and elaborate on the output.

```
print('Number of rows before discarding outliers = %d' % (Z.shape[0]))

Z2 = Z.loc[((Z > -3).sum(axis=1)==9) & ((Z <= 3).sum(axis=1)==9),:]
print('Number of rows after discarding missing values = %d' % (Z2.shape[0]))
```

10. Show the output of the above code and elaborate on the output.

```
dups = data.duplicated()
print('Number of duplicate rows = %d' % (dups.sum()))
data.loc[[11,28]]

print('Number of rows before discarding duplicates = %d' % (data.shape[0]))
data2 = data.drop_duplicates()
print('Number of rows after discarding duplicates = %d' % (data2.shape[0]))
```

11. Show the output of the above code and elaborate on the output.

```
daily = pd.read_csv('DTW_prec.csv', header='infer')
daily.index = pd.to_datetime(daily['DATE'])
daily = daily['PRCP']
ax = daily.plot(kind='line',figsize=(15,3))
ax.set_title('Daily Precipitation (variance = %.4f)' % (daily.var()))
```

12. Show the output of the above code and elaborate on the output.

```
monthly = daily.groupby(pd.Grouper(freq='M')).sum()
ax = monthly.plot(kind='line',figsize=(15,3))
ax.set_title('Monthly Precipitation (variance = %.4f)' % (monthly.var()))

annual = daily.groupby(pd.Grouper(freq='Y')).sum()
ax = annual.plot(kind='line',figsize=(15,3))
ax.set_title('Annual Precipitation (variance = %.4f)' % (annual.var()))
```

13. Show the output of the above code and elaborate on the output.

```
data.head()
```

```
sample = data.sample(n=3)
sample

sample = data.sample(frac=0.01, random_state=1)
sample

sample = data.sample(frac=0.01, replace=True, random_state=1)
sample
```

14.Show the output of the above code and elaborate on the output.

```
data['Clump Thickness'].hist(bins=10)
data['Clump Thickness'].value_counts(sort=False)

bins = pd.cut(data['Clump Thickness'],4)
bins.value_counts(sort=False)

bins = pd.qcut(data['Clump Thickness'],4)
bins.value_counts(sort=False)
```

15.Show the output of the above code and elaborate on the output.

Dimensionality Reduction

Principal Components Analysis (PCA)

```
numImages = 16
fig = plt.figure(figsize=(7,7))
imgData = np.zeros(shape=(numImages,36963))

for i in range(1,numImages+1):
    filename = 'pics/Picture'+str(i)+'.jpg'
    img = mpimg.imread(filename)
    ax = fig.add_subplot(4,4,i)
    plt.imshow(img)
    plt.axis('off')
    ax.set_title(str(i))
    imgData[i-1] = np.array(img.flatten()).reshape(1,img.shape[0]*img.shape[1]*img.shape[
2])
```

16.Show the output of the above code and elaborate on the output.

```
numComponents = 2
pca = PCA(n_components=numComponents)
pca.fit(imgData)

projected = pca.transform(imgData)
projected = pd.DataFrame(projected,columns=['pc1','pc2'],index=range(1,numImages+1))
projected['food'] = ['burger', 'burger','burger','burger','drink','drink','drink','drink'
,
```

```
                    'pasta', 'pasta', 'pasta', 'pasta', 'chicken', 'chicken', 'chicken'
, 'chicken']
projected
```

17.Show the output of the above code and elaborate on the output.

```
colors = {'burger':'b', 'drink':'r', 'pasta':'g', 'chicken':'k'}
markerTypes = {'burger':'+', 'drink':'x', 'pasta':'o', 'chicken':'s'}

for foodType in markerTypes:
    d = projected[projected['food']==foodType]
    plt.scatter(d['pc1'],d['pc2'],c=colors[foodType],s=60,marker=markerTypes[foodType])
```

18.Show the output of the above code and elaborate on the output.