In [ ]:

```python
# importing necessary packages
import numpy as np
import matplotlib.pyplot as plt
```

In [ ]:

```python
# generating a dataset
X = np.random.normal(0,2,size=(100,10))
```

In [ ]:

```python
# SVD
U,S,Vt = np.linalg.svd(X)
```

In [ ]:

```python
print(U.shape,S.shape,Vt.shape)
```

```
(100, 100) (10,) (10, 10)
```

In [ ]:

```python
# Function to get the principal components
def get_principal_comps(X,Vt,n):
  '''
  X: data matrix
  Vt: right singular matrix
  '''
  PC = X@(Vt.T)
  return PC[:,:n]
```

In [ ]:

```python
PC = get_principal_comps(X,Vt,2)
```

In [ ]:

```python
# Function to get the variance explained
def get_var_exp(S,n):
  '''
  S: vector of singular values
  '''
  eig = S**2
  return eig[n]/np.sum(eig)
```

In [ ]:

```python
first_comp_var_exp = get_var_exp(S,1)
print(first_comp_var_exp)
```

```
0.1452368124224459
```

In [ ]:

```
second_comp_var_exp = get_var_exp(S,2)
print(second_comp_var_exp)
```

```
0.12026535693663559
```

In [ ]:

```
M = list(range(1,10)) # components
var_exp  = [get_var_exp(S,i-1) for i in M] # Variance explained by each of the component
```

In [ ]:

```
# Scree plot
plt.plot(M,var_exp,'go--')
plt.xlabel('Components')
plt.ylabel('Variance expalined')
plt.title('Scree plot')
```

In [ ]:

```
# Function to reconstruc the data from the principal components
def reconstruct_data_mat(PC,Vt,n):
  '''
  PC: principal components
  Vt: Right singular matrix
  '''
  X_ = PC@(Vt[:n,:])
  return X_
```

In [ ]:

```
X_ = reconstruct_data_mat(PC,Vt,2)
```

In [ ]:

```
X_.shape
```

Out[22]:

```
(100, 10)
```

In [ ]:

```

```