Activity 13

Damion Huppert

Question 1-a

$$X = \sum_{i=1}^{n} \sigma_i u_i v_i^T$$

$$X_r = \sum_{i=1}^{r} \sigma_i u_i v_i^T$$

$$E_r = \sum_{i=1}^{n} \sigma_i u_i v_i^T - \sum_{i=1}^{r} \sigma_i u_i v_i^T = \sum_{i=r+1}^{n} \sigma_i u_i v_i^T$$

Question 1-b

then rank of E_r would be n - r

Question 1-c

$$E_r = \sum_{i=r+1}^{n} \sigma_i u_i v_i^T$$

$$||E_r||_{\text{op}} = \sigma_{r+1}$$

Question 1-d

X_r will be a good approimation when the error is low which means the singular values of $r+1$--n are close to 0

Question 2

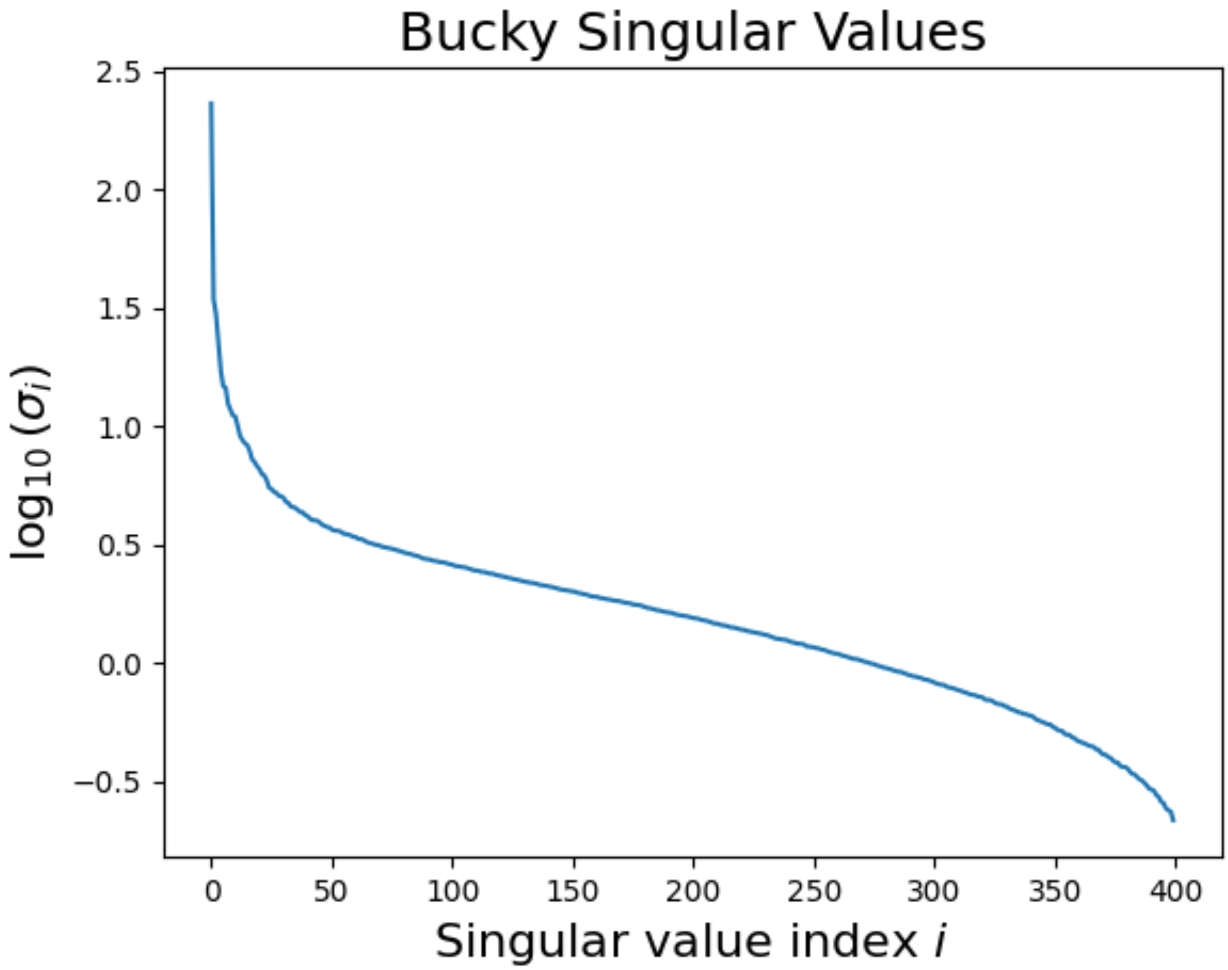
In [2]:
```
import numpy as np
from scipy.io import loadmat
import matplotlib.pyplot as plt
from mpl_toolkits.mplot3d import Axes3D

# Load data for activity
#
in_data = loadmat('bucky.mat')
```

```
A = in_data['A']

##

# Load data for activity: Another option
# A = imageio.imread("Whateveryoulike.png")
# A = np.average(A[:,:,0:3], axis=2)/256

rows, cols = np.array(A.shape)

# Display image
fig = plt.figure()
ax = fig.add_subplot(111)

ax.imshow(A,cmap='gray')
ax.set_axis_off()
plt.show()
```

In [3]:

```
U,s,VT = np.linalg.svd(A)

fig = plt.figure()
ax = fig.add_subplot(111)
ax.plot(np.log10(s))
ax.set_xlabel('Singular value index $i$', fontsize=16)
ax.set_ylabel('$\log_{10}(\sigma_i)$', fontsize=16)
ax.set_title('Bucky Singular Values', fontsize=18)
plt.show()
```

```
<>:7: SyntaxWarning: invalid escape sequence '\l'
<>:7: SyntaxWarning: invalid escape sequence '\l'
/var/folders/tn/v9tpvrrs4qgdbw0xd1q0l8qh0000gn/T/ipykernel_79225/586610892.p
y:7: SyntaxWarning: invalid escape sequence '\l'
  ax.set_ylabel('$\log_{10}(\sigma_i)$', fontsize=16)
```



Question 2-a

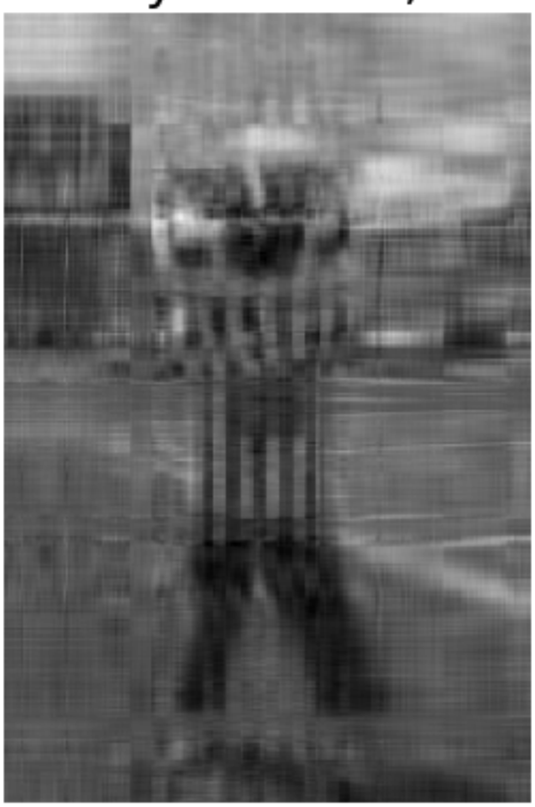
The rank of A is low.
We can see the significance of each singular value by the graph and can tell what values will be the most usefull

Question 2-b

In [4]:
```
# Find and display low-rank approximations

r_vals = np.array([10, 20, 50, 100 ])
err_fro = np.zeros(len(r_vals))

# display images of various rank approximations
for i, r in enumerate(r_vals):

    # Complete and uncomment two lines below
    Ar = U[:, :r] @ np.diag(s[:r]) @ VT[:r, :]
```

```python
    Er = A - Ar
    err_fro[i] = np.linalg.norm(Er,ord='fro')

    fig = plt.figure()
    ax = fig.add_subplot(111)
    ax.imshow(Ar,cmap='gray',interpolation='none')
    ax.set_axis_off()
    ax.set_title(['Bucky Rank =', str(r_vals[i])], fontsize=18)
    plt.show()

# plot normalized error versus rank
norm_err = err_fro/np.linalg.norm(A,ord='fro')

fig = plt.figure()
ax = fig.add_subplot(111)
ax.stem(r_vals,norm_err)
ax.set_xlabel('Rank', fontsize=16)
ax.set_ylabel('Normalized error', fontsize=16)
plt.show()
```

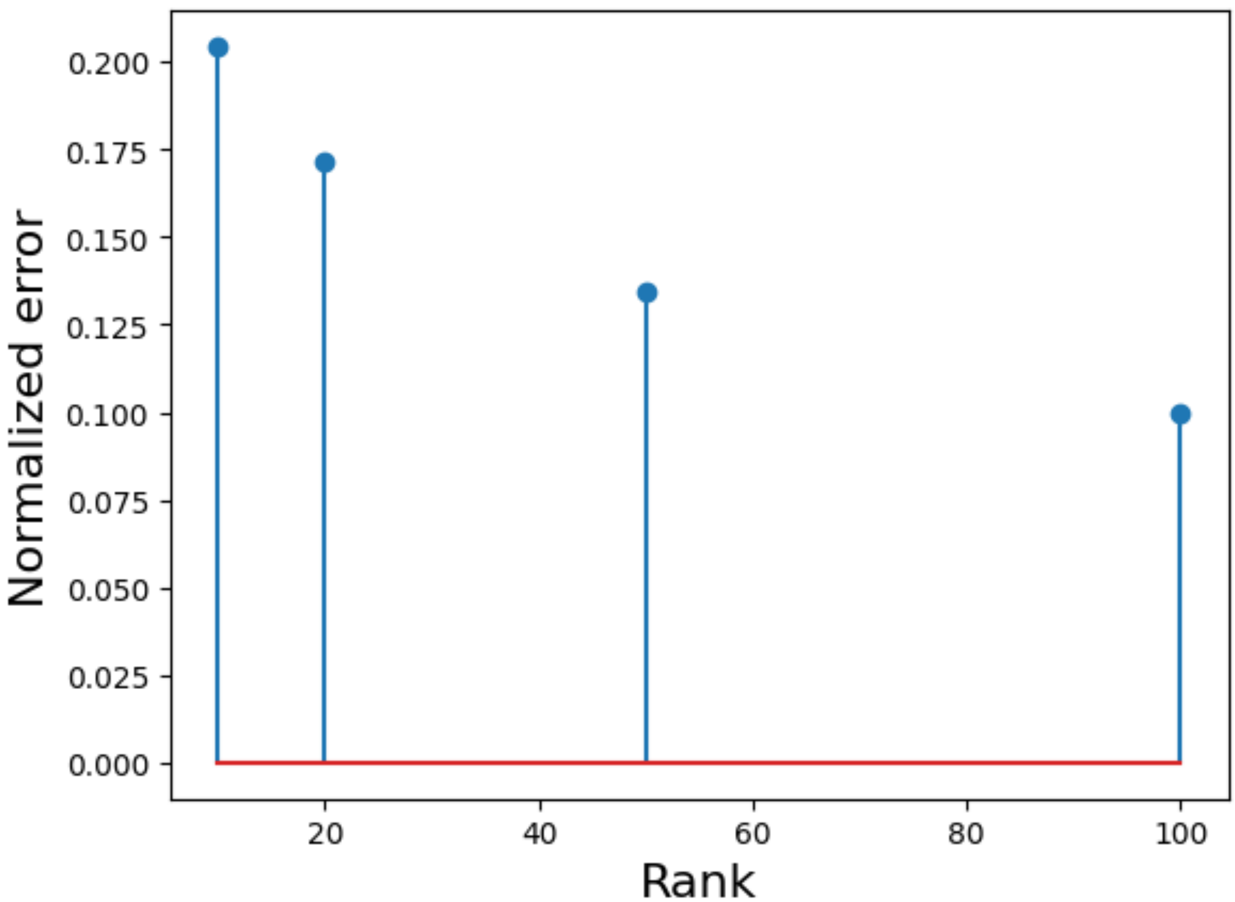
['Bucky Rank =', '10']



['Bucky Rank =', '20']

['Bucky Rank =', '50']

['Bucky Rank =', '100']



As r increases the quality of the image increases

Question 2-c

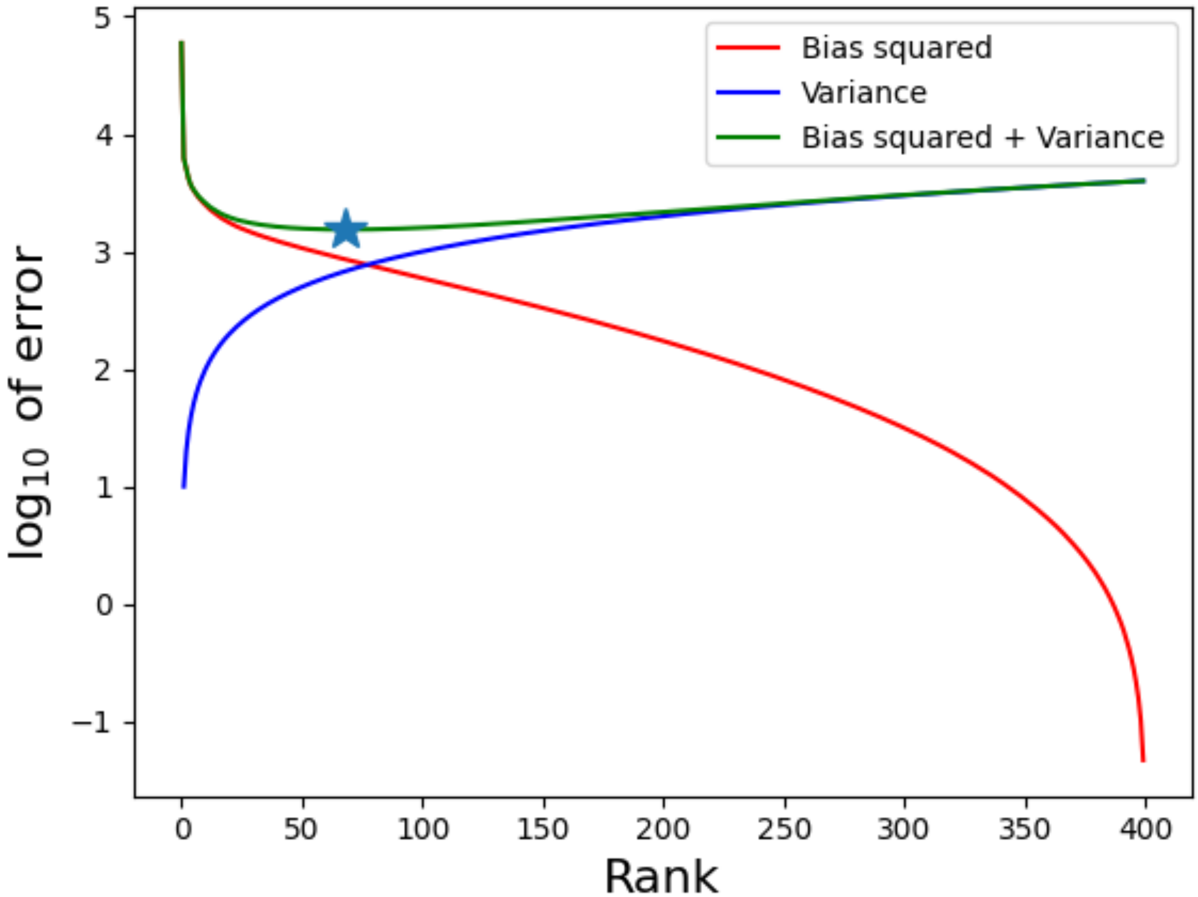
In [12]:
```
print(A.shape)

for r in r_vals:
    original_size = A.shape[0] * A.shape[1]
    compressed_size = (A.shape[0] * r) + r + (A.shape[1] * r)
    compression_factor = original_size / compressed_size
    print(f"Rank {r}: {compression_factor:.2f} times smaller storage")
```

```
(600, 400)
Rank 10: 23.98 times smaller storage
Rank 20: 11.99 times smaller storage
Rank 50: 4.80 times smaller storage
Rank 100: 2.40 times smaller storage
```

Question 2-d

In [15]:
```
# bias-variance tradeoff
num_sv = min(rows, cols)
bias_2 = np.zeros(num_sv)
ranks = np.arange(num_sv)

for r in range(num_sv):
    bias_2[r] = np.linalg.norm(s[r:num_sv])**2

sigma2 = 10
var = sigma2*ranks

fig = plt.figure()
ax = fig.add_subplot(111)
ax.plot(ranks,np.log10(bias_2),'r',label='Bias squared')
ax.plot(ranks[1:],np.log10(var[1:]),'b', label = 'Variance')
ax.plot(ranks,np.log10(bias_2+var),'g', label='Bias squared + Variance')
min_bias_plus_variance_index = np.argmin(np.log10(bias_2+var))
ax.plot(ranks[min_bias_plus_variance_index], np.log10(bias_2+var)[min_bias_p
ax.set_xlabel('Rank', fontsize=16)
ax.set_ylabel('$\log_{10}$ of error', fontsize=16)
ax.legend()
plt.show()
```

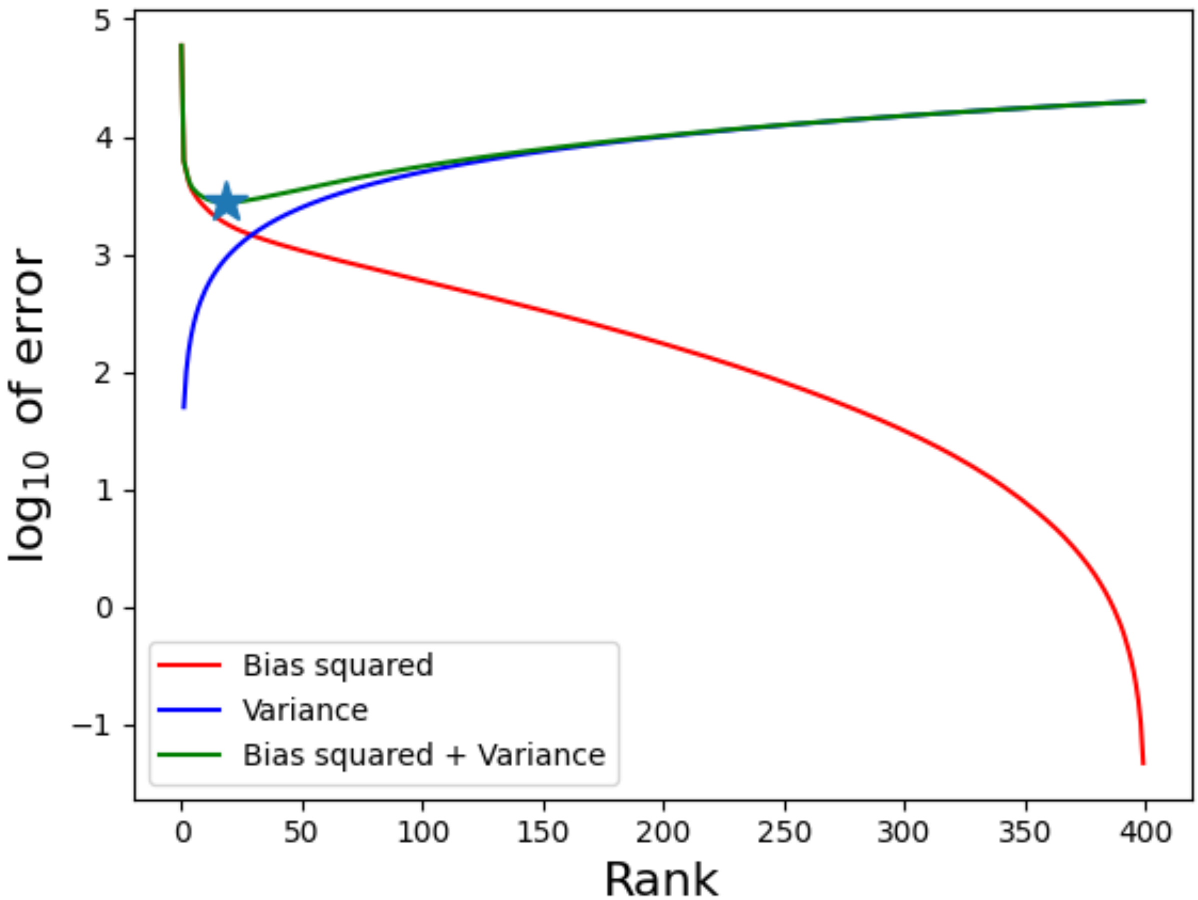
```
<>:20: SyntaxWarning: invalid escape sequence '\l'
<>:20: SyntaxWarning: invalid escape sequence '\l'
/var/folders/tn/v9tpvrrs4qgdbw0xd1q0l8qh0000gn/T/ipykernel_79225/3664024307.
py:20: SyntaxWarning: invalid escape sequence '\l'
  ax.set_ylabel('$\log_{10}$ of error', fontsize=16)
```



Optimal rank: ~60

In [13]:
```
# bias-variance tradeoff
num_sv = min(rows, cols)
bias_2 = np.zeros(num_sv)
ranks = np.arange(num_sv)

for r in range(num_sv):
    bias_2[r] = np.linalg.norm(s[r:num_sv])**2

sigma2 = 50
var = sigma2*ranks
#print(var)

fig = plt.figure()
ax = fig.add_subplot(111)
ax.plot(ranks,np.log10(bias_2),'r',label='Bias squared')
ax.plot(ranks[1:],np.log10(var[1:]),'b', label = 'Variance')
ax.plot(ranks,np.log10(bias_2+var),'g', label='Bias squared + Variance')
min_bias_plus_variance_index = np.argmin(np.log10(bias_2+var))
ax.plot(ranks[min_bias_plus_variance_index], np.log10(bias_2+var)[min_bias_p
ax.set_xlabel('Rank', fontsize=16)
ax.set_ylabel('$\log_{10}$ of error', fontsize=16)
ax.legend()
plt.show()
```

```
&lt;&gt;:21: SyntaxWarning: invalid escape sequence '\l'
&lt;&gt;:21: SyntaxWarning: invalid escape sequence '\l'
/var/folders/tn/v9tpvrrs4qgdbw0xd1q0l8qh0000gn/T/ipykernel_79225/46884132.p
y:21: SyntaxWarning: invalid escape sequence '\l'
  ax.set_ylabel('$\log_{10}$ of error', fontsize=16)
```



Optimal rank: ~20