Lossy Compression

In this problem you will work through a toy example of lossy compression of digital images. Suppose you have a row of 8 grayscale pixels. Let $v = [37, 155, 12, 171, 243, 56, 237, 5]$ represent the light intensities of each pixel in a sample image (see Figure 1).

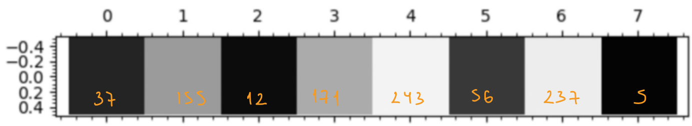


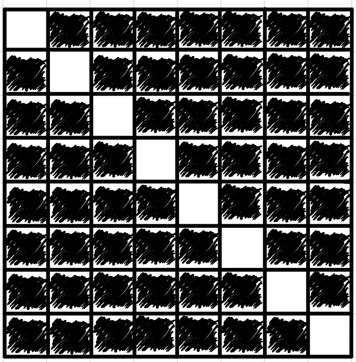
Figure 1: Row of pixels corresponding to v. This image was created using matrix_plot() in SageMath. The value range was set to $[0, 256]$ and the color map was set to 'gray' to render the corresponding image in 8-bit grayscale.

1 What do the eight standard basis vectors of represent in terms of this image? What does a linear combination of the standard vectors represent? What do the coordinates of a vector represent?

The eight standard basis vectors are eight rows of pixels that correspond to every possible standard basis vector.

Full picture:

$$\begin{array}{c|cccccccc}
R_1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\
 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\
 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\
 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\
 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\
 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\
R_8 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\
 & v_1 & & & & & & & v_8
\end{array}$$



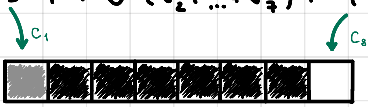
The intensity of a pixel is expressed within a range from 0 to 1, where 0 is "total absence" (black) and 1 is "total presence" (white).[1]

In this context, linear combination of the standard vectors is the sum of scalar multiples of all vectors v_1 to v_8, where scalar multiples change the range of the intensity of pixels for the corresponding pixel, or a "box". For example, let $c_1 = 0.5$, then:

$$c_1 \cdot v_1 = 0.5 \cdot \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} = \langle 0.5, 0, 0, 0, 0, 0, 0, 0 \rangle$$

Coordinates of the vector represent which pixel changes the light intensity. Example:

$$\bar{v} = c_1 \cdot v_1 + \ldots + c_8 \cdot v_8 = 0.5\, v_1 + 0\,(v_2 + \ldots + v_7) + 1 \cdot v_8$$



1. Grayscale (n.d.). Wikipedia. https://en.wikipedia.org/wiki/Grayscale

2 Consider the eight vectors

$$v_1 = \begin{bmatrix}1\\1\\1\\1\\1\\1\\1\\1\end{bmatrix}, v_2 = \begin{bmatrix}1\\1\\1\\1\\-1\\-1\\-1\\-1\end{bmatrix}, v_3 = \begin{bmatrix}1\\1\\-1\\-1\\0\\0\\0\\0\end{bmatrix}, v_4 = \begin{bmatrix}0\\0\\0\\0\\1\\1\\-1\\-1\end{bmatrix}, v_5 = \begin{bmatrix}1\\-1\\0\\0\\0\\0\\0\\0\end{bmatrix}, v_6 = \begin{bmatrix}0\\0\\1\\-1\\0\\0\\0\\0\end{bmatrix}, v_7 = \begin{bmatrix}0\\0\\0\\0\\1\\-1\\0\\0\end{bmatrix}, v_8 = \begin{bmatrix}0\\0\\0\\0\\0\\0\\1\\-1\end{bmatrix}$$

(a) Demonstrate that these vectors form an orthogonal basis S of $\mathbb{R}^8$.

```
# Define the vectors v1 to v8.
v1 = vector([1,1,1,1,1,1,1,1])
v2 = vector([1,1,1,1,-1,-1,-1,-1])
v3 = vector([1,1,-1,-1,0,0,0,0])
v4 = vector([0,0,0,0,1,1,-1,-1])
v5 = vector([1,-1,0,0,0,0,0,0])
v6 = vector([0,0,1,-1,0,0,0,0])
v7 = vector([0,0,0,0,1,-1,0,0])
v8 = vector([0,0,0,0,0,0,1,-1])

'''
I. Demonstrate that these vectors form an orthogonal basis S of R^8.
'''
# Define the matrix S.
S = matrix(QQ, [v1,v2,v3,v4,v5,v6,v7,v8]).transpose()

# Check A belongs to R^8 by looking at # of pivots and vectors and LI.
show(S.rref())

# Check that all dot products of vectors == 0.
show("The dot products of all vectors are the following:",
     v1*v2,",",v1*v3,",",v1*v4,",",v1*v5,",",v1*v6,",",v1*v7,",",v1*v8,",",v2*v3,",",v2*v4,
    ",",v2*v5,",",v2*v6,",",v2*v7,",",v2*v8,",",v3*v4,",",v3*v5,",",v3*v6,",",v3*v7,",",v3*v8,",",
     v4*v5,",",v4*v6,",",v4*v7,",",v4*v8,",",v5*v6,",",v5*v7,",",v5*v8,",",v6*v7,",",v6*v8,",",v7*v8)
```

$$\begin{pmatrix}1&0&0&0&0&0&0&0\\0&1&0&0&0&0&0&0\\0&0&1&0&0&0&0&0\\0&0&0&1&0&0&0&0\\0&0&0&0&1&0&0&0\\0&0&0&0&0&1&0&0\\0&0&0&0&0&0&1&0\\0&0&0&0&0&0&0&1\end{pmatrix}$$

The dot products of all vectors are the following:0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0

1. For something to be a basis for a vector space, they must share the same dimensions. Since $\mathbb{R}^8$ has 8 dimensions and the matrix S formed by the vectors has 8 pivots in RREF => => dim(S) = 8, the vectors form a basis for $\mathbb{R}^8$.

Now, let us verify that the basis is orthogonal:

1. <u>Are vectors linearly independent?</u>

Yes! We see from the RREF above that all columns have one pivot forming an identity matrix, meaning there is no way to represent one vector in a form of the other one.

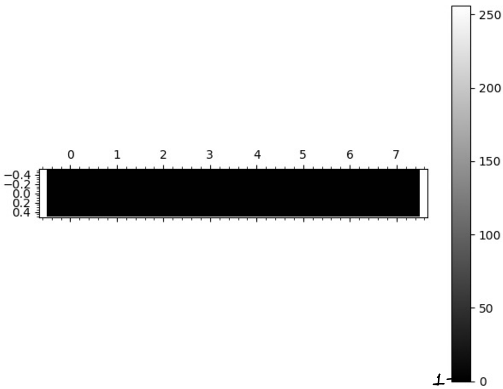
2. <u>Is dot product of each vector == 0?</u>

Yes! Check the Sage output above.

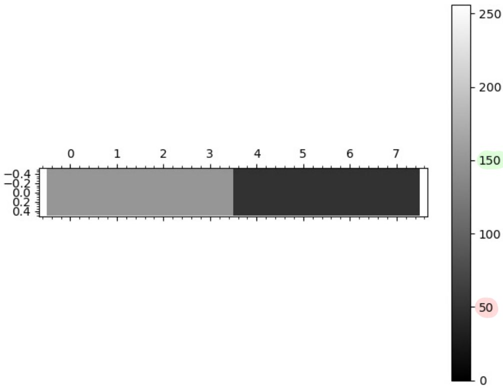
Given two conditions of orthogonality are satisfied and vectors' dimension equals to the dimension of R^8, we proved that vectors v_1 to v_8 form an orthogonal basis S of R^8.

(b) What do v_1, $100v_1 + 50v_2$, and $128v_1 - 64v_3 + 32v_5 - 16v_7$ represent in terms of a grayscale image? Plot the grayscale image corresponding to each linear combination using matrix_plot(). Give a general description of how linear combinations of the v_i represent a grayscale image.

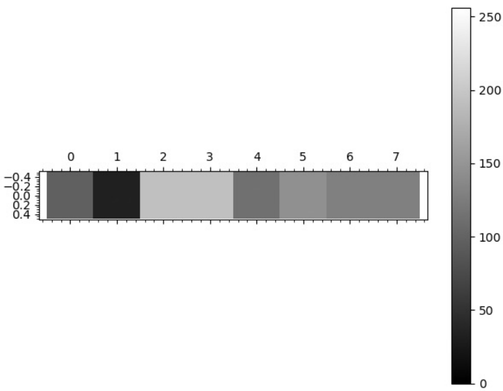
```
# Plot the grayscale image corresponding to each linear combination using matrix plot().
show(matrix_plot(matrix([v1]), vmin = 0, vmax = 256, colorbar = True, cmap = 'gray'))
show(matrix_plot(matrix([100*v1+50*v2]), vmin = 0, vmax = 256, colorbar = True, cmap = 'gray'))
show(matrix_plot(matrix([128*v1-64*v3+32*v5-16*v7]), vmin = 0, vmax = 256, colorbar = True, cmap = 'gray'))
```



This is the row of pixels representing $\bar{v}_1$ in 8-bit grayscale with values ranging from 0 to 256 and color map set to 'gray' as stated in the assingment. Given the values didn't change (all 1s), the pixels remain all black ('total absence').



This is the row of pixels representing $100v_1 + 50v_2$. Calculating the final vector, we get: $<150, 150, 150, 150, 50, 50, 50, 50>$ Those are the color intensity of the pixels we observe on the left



This is the row of pixels representing $128v_1 - 64v_3 + 32v_5 - 16v_7$

```
128*v1-64*v3+32*v5-16*v7
```

```
(96, 32, 192, 192, 112, 144, 128, 128)
```

Again, the color intensity of the pixels is represented on the left. The coordinates of v_1, v_3, v_5, v_7 put the image pixels in corresponding shades.

General description: pixels have a color intensity from 0 (black) to 256 (white). All values in between are the shades of gray which we get from linear combinations of the vectors v_i. Any coefficient (except 1) changes the color "light". In terms of a vector, it changes the magnitude, but keeps the location.

3. Research suggests that in this new basis, coordinate values close to zero are less important for rendering a human interpretable image. In order to take advantage of the fact that sparse matrices (i.e., matrices with many zeros) take less storage space, you can set a positive threshold value ϵ then set any coordinates with absolute value less than ϵ in our new basis to zero.

(a) Find the coordinates of v in terms of the new basis $S = \{v_1, v_2, \ldots, v_8\}$.

```
'''
Find the coordinates of v in terms of the new basis S = {v1, v2, . . . , v8}.
'''
# Define the given vector v.
v = vector([37, 155, 12, 171, 243, 56, 237, 5])

# Find the change of basis.
# We need to multiply the vector by the inverse of matrix S.
show(S.inverse()*v)
```

$$\left(\frac{229}{2}, -\frac{83}{4}, \frac{9}{4}, \frac{57}{4}, -59, -\frac{159}{2}, \frac{187}{2}, 116\right) \leftarrow \text{coordinates}$$

(b) Choose three thresholds ϵ_1, ϵ_2, ϵ_3. Let c_1, c_2, c_3 be the vectors you get by compressing v using thresholds ϵ_1, ϵ_2, and ϵ_3 respectively. Express c_1,c_2,c_3 in the standard basis. What do these new vectors represent? Create visual representations of v, c_1, c_2, c_3 in the standard basis vectors.

(c) Experiment with choices of ϵ. How large can you make ϵ before the compression is noticeable?

Let thresholds be: $\epsilon_1 = 116$ ← the largest value in $[\bar{v}]_S$

$\epsilon_2 = 58$

$\epsilon_3 = 1$

```python
'''
(b) Choose three thresholds e1, e2, e3. Let c1, c2, c3 be the vectors you get by compressing
v using thresholds e1, e2, e3 respectively.
(c) Experiment with choices of e. How large can you make before the compression is
noticeable?
'''
# Define v in a new basis S.
vS = S.inverse()*v

# Initialize e.
e = 116

# Initialize the compressing element.
replace = 0

# Define a list that will contain new compressed vectors.
vS1 = []

# Create a for loop that replaces elements smaller than the threshold.
for i in vS:
    if abs(i) < e:
        vS1.append(replace)
    else:
        vS1.append(i)
print("The original vector is", vS, '.')
print("The compressed vector in a new basis is", vS1, '.')
```

```
The original vector is (229/2, -83/4, 9/4, 57/4, -59, -159/2, 187/2, 116) .
The compressed vector in a new basis is [0, 0, 0, 0, 0, 0, 0, 116] .
```

$[c_1]_S$ for $\varepsilon_1 = 116$

$[c_2]_S$ for $\varepsilon_2 = 58$

```
The original vector is (229/2, -83/4, 9/4, 57/4, -59, -159/2, 187/2, 116) .
The compressed vector in a new basis is [229/2, 0, 0, 0, -59, -159/2, 187/2, 116] .
```

$[c_3]_S$ for $\varepsilon_3 = 1$

```
The original vector is (229/2, -83/4, 9/4, 57/4, -59, -159/2, 187/2, 116) .
The compressed vector in a new basis is [229/2, -83/4, 9/4, 57/4, -59, -159/2, 187/2, 116] .
```

```python
'''
Express c1,c2,c3 in the standard basis.
What do these new vectors represent?
'''
# Define the compressed vector.
c = vector(vS1)

# Express c in the standard basis.
# We need to multiply the vector by matrix S.
S*c
```

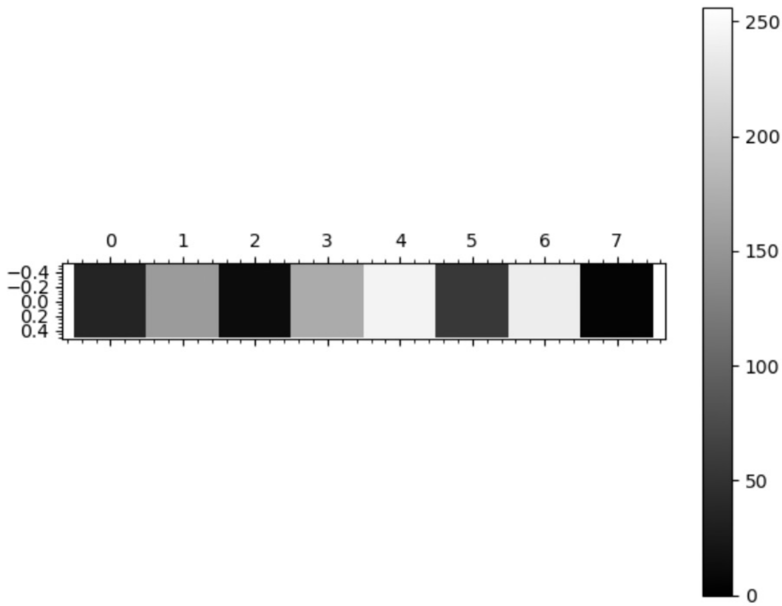
c_1 (0, 0, 0, 0, 0, 0, 116, -116)

c_2 (111/2, 347/2, 35, 194, 208, 21, 461/2, -3/2)

c_3 (37, 155, 12, 171, 243, 56, 237, 5)

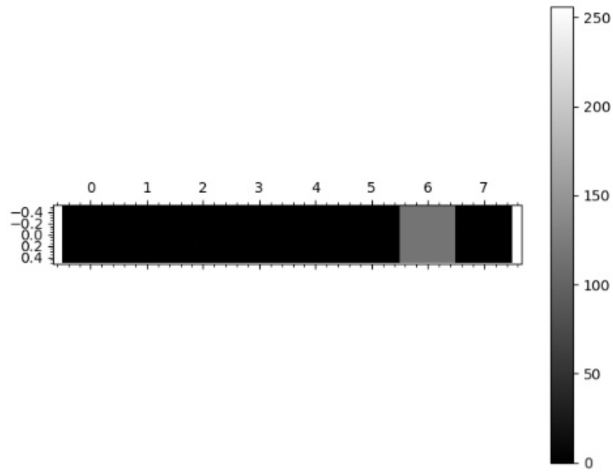
c_1, c_2, c_3 show the same grayscale as in 2, but with lower resolution; thus, lower memory space needed to store them because ε values compressed the pixels.

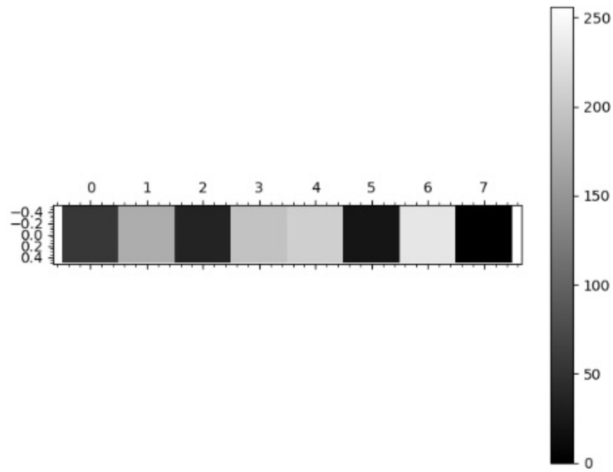
```
'''
Create visual representations of v, c1, c2, c3
in the standard basis vectors.
'''
show(matrix_plot(matrix([v]), vmin = 0, vmax = 256, colorbar = True, cmap = 'gray'))
```

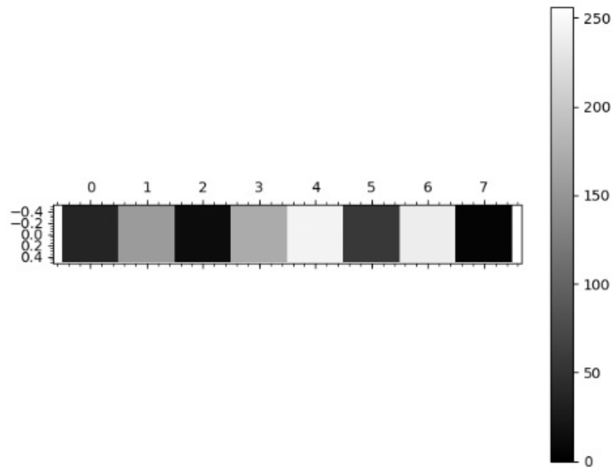


From the visualizations, we see that the larger threshold value makes more colors look the same as the larger ϵ results in more pixels equal to zero; thus, the image isn't differentiable to human eyes.

```
c1 = vector([0, 0, 0, 0, 0, 0, 116, -116])
c2 = vector([111/2, 347/2, 35, 194, 208, 21, 461/2, -3/2])
c3 = vector([37, 155, 12, 171, 243, 56, 237, 5])
show(matrix_plot(matrix([c1]), vmin = 0, vmax = 256, colorbar = True, cmap = 'gray'))
show(matrix_plot(matrix([c2]), vmin = 0, vmax = 256, colorbar = True, cmap = 'gray'))
show(matrix_plot(matrix([c3]), vmin = 0, vmax = 256, colorbar = True, cmap = 'gray'))
```







We see the largest difference between $\epsilon = 116$ and $\epsilon = 1$; the former one is unrecognizable to original $\bar{v}$. $\epsilon = 58$ had three zero values in a new basis, but looks the same to human eye as $\bar{v}$ does. So, we can compress at least twice before its noticeable.