Exercise 10.1

1. Consider the following data:

	A	B	C	D	E	F	G	H	I	J
3		Poverty	Infant Mort	White	Crime	Doctors	Traf Deaths	University	Unemployed	Income
4	Alabama	15.7	9.0	71.0	448	218.2	1.81	22.0	5.0	42,666
5	Alaska	8.4	6.9	70.6	661	228.5	1.63	27.3	6.7	68,460
6	Arizona	14.7	6.4	86.5	483	209.7	1.69	25.1	5.5	50,958
7	Arkansas	17.3	8.5	80.8	529	203.4	1.96	18.8	5.1	38,815
8	California	13.3	5.0	76.6	523	268.7	1.21	29.6	7.2	61,021
9	Colorado	11.4	5.7	89.7	348	259.7	1.14	35.6	4.9	56,993
10	Connecticut	9.3	6.2	84.3	256	376.4	0.86	35.6	5.7	68,595
11	Delaware	10.0	8.3	74.3	689	250.9	1.23	27.5	4.8	57,989
12	Florida	13.2	7.3	79.8	723	247.9	1.56	25.8	6.2	47,778
13	Georgia	14.7	8.1	65.4	493	217.4	1.46	27.5	6.2	50,861
14	Hawaii	9.1	5.6	29.7	273	317.0	1.33	29.1	3.9	67,214
15	Idaho	12.6	6.8	94.6	239	168.8	1.60	24.0	4.9	47,576

Look over the data. By eye, which columns look correlated? Anticorrelated? Uncorrelated?

Solution. A couple of examples are:

- Poverty is anticorrelated with income.
- Income is correlated with Doctors.

■

2. Choose your two favorite columns of data from this dataset. Input these into vectors in Matlab. For each of these vectors, subtract off the mean, and then divide out the standard deviation.

Solution.

```
poverty = [15.7; 8.4; 14.7; 17.3; 13.3; 11.4; 9.3; 10.0; 13.2; 14.7; 9.1; 12.6];
income = [42666; 68460; 50958; 38815; 61021; 56993; 68595; 57989; 47778; 50861; 67214; 47576];

poverty_adj = (poverty - mean(poverty)) / std(poverty);
income_adj = (income - mean(income)) / std(income);
```

■

3. With these vectors, how would you directly compute the correlation coefficient between them?

Solution.

```
correlation = corr(poverty_adj, income_adj)
```

$$\boldsymbol{C} = -0.9005$$

■

Exercise 10.3

1. (a) What is the size of the matrix $\boldsymbol{C}$?

   Solution. 2×2 ■

   (b) What are the elements on the *diagonal* of this matrix?

   Solution. The entire diagonal should all be 1. ■

   (c) What are the elements of the off-diagonal? What is element C_{12} of this matrix? What is element C_{21}? What do you notice? Is this always going to be true?

   Solution. C_{12} is the correlation coefficient of A_1 and A_2. Because C_{21} is the correlation coefficient of A_2 and A_1, $C_{12} = C_{21}$. ■

   (d) If you create a data matrix that has completely identical columns of data, what should the correlation matrix look like?

   Solution. The correlation matrix would be:

$$\boldsymbol{C} = \begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix}$$

   ■

   (e) If you create a data matrix that has completely uncorrelated datasets, what should the correlation matrix look like?

   Solution. If the datasets were completely uncorrelated, the correlation matrix would be:

$$\boldsymbol{C} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \tag{1}$$

   ■

2. Implement this in Matlab.

Solution.

```
a = [(1:10)', rand(10, 1)]
```

$$\boldsymbol{a} = \begin{bmatrix} 1.0000 & 0.1576 \\ 2.0000 & 0.9706 \\ 3.0000 & 0.9572 \\ 4.0000 & 0.4854 \\ 5.0000 & 0.8003 \\ 6.0000 & 0.1419 \\ 7.0000 & 0.4218 \\ 8.0000 & 0.9157 \\ 9.0000 & 0.7922 \\ 10.0000 & 0.9595 \end{bmatrix}$$

```
o = ones(size(a, 1), 1)
```

$$\boldsymbol{o} = \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \end{bmatrix}$$

```
m = o * mean(a)
```

$$\boldsymbol{m} = \begin{bmatrix} 5.5000 & 0.6602 \\ 5.5000 & 0.6602 \\ 5.5000 & 0.6602 \\ 5.5000 & 0.6602 \\ 5.5000 & 0.6602 \\ 5.5000 & 0.6602 \\ 5.5000 & 0.6602 \\ 5.5000 & 0.6602 \\ 5.5000 & 0.6602 \\ 5.5000 & 0.6602 \end{bmatrix}$$

```
s = o * std(a)
```

$$s = \begin{bmatrix} 3.0277 & 0.3308 \\ 3.0277 & 0.3308 \\ 3.0277 & 0.3308 \\ 3.0277 & 0.3308 \\ 3.0277 & 0.3308 \\ 3.0277 & 0.3308 \\ 3.0277 & 0.3308 \\ 3.0277 & 0.3308 \\ 3.0277 & 0.3308 \\ 3.0277 & 0.3308 \end{bmatrix}$$

```
b = (a - m) ./ s
```

$$b = \begin{bmatrix} -1.4863 & -1.5191 \\ -1.1560 & 0.9382 \\ -0.8257 & 0.8976 \\ -0.4954 & -0.5285 \\ -0.1651 & 0.4234 \\ 0.1651 & -1.5667 \\ 0.4954 & -0.7207 \\ 0.8257 & 0.7723 \\ 1.1560 & 0.3990 \\ 1.4863 & 0.9046 \end{bmatrix}$$

```
c = (1 / (size(a,1) - 1)) * b' * b
```

$$c = \begin{bmatrix} 1.0000 & 0.2724 \\ 0.2724 & 1.0000 \end{bmatrix}$$

■

3. What would happen if you used this matrix for a instead:

```
a = [(1:10)',(1:10)'/2 -(1:10)']
```

Solution. Because the numbers are negatively correlated, the correlation would be:

$$C = \begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix}$$

■

4. Use the `corrcoef` command to compute the correlation coefficients of the original matrix.

Solution.

```
C = corrcoef([poverty_adj income_adj])
```

$$\boldsymbol{C} = \begin{bmatrix} 1 & -0.9005 \\ -0.9005 & 1 \end{bmatrix}$$

■

Exercise 10.4

1. (a) Imagine you have grayscale images that are 100×100 pixels. If you represent each image as a vector, how large is the vector space?

   Solution. The vector space would be 10000-dimensional. ■

   (b) If the intensity of pixel i, j is $a_{i,j}$, come up with an expression for the magnitude of the vector that describes the image $\boldsymbol{a}$.

   Solution.

   $$\|a\| = \sqrt{\sum a_{i,j}^2}$$

   ■

   (c) Come up with an expression that represents the distance between images $\boldsymbol{a}$ and $\boldsymbol{b}$.

   Solution.

   $$\|a\| - \|b\| = \sqrt{\sum (a_{i,j} - b_{i,j})^2}$$

   ■

   (d) What does $a^\intercal b$ tell you? What vector operation is this?

   Solution. The dot product $a^\intercal b$ tells you how similar the vectors are. ■

2. Load three face images, and calculate the distance between each pair.

Solution.

```
load('face_bases.mat');
im1 = test_images(:,:,1);
im2 = test_images(:,:,2);
im3 = test_images(:,:,3);

dist_im12 = sqrt(sum(sum(((im1-im2).^2))))
dist_im23 = sqrt(sum(sum(((im2-im3).^2))))
dist_im13 = sqrt(sum(sum(((im1-im3).^2))))
```

$$D_{12} = 0.1565$$
$$D_{23} = 0.1184$$
$$D_{13} = 0.0797$$

■

Exercise 10.5

1. Consider six grayscale pictures, each with a resolution of $m \times n$ pixels. What is the size of the data matrix containing these six pictures as the columns?

   Solution. $mn \times 6$ ■

2. What is the expression for the correlation matrix between the pictures? What size is this correlation matrix?

   Solution. The correlation matrix is given with `corrcoef(data)`. This matrix will be of size 6×6. ■

3. What is the expression for the correlation matrix between different pixels? What is the size of this correlation matrix?

   Solution. To find the correlation matrix, we would take the transpose of the data matrix. This can be found in matlab with:

   ```
   C = corrcoef(data')
   ```

   The size of the correlation matrix is $mn \times mn$. ■

4. People's faces are approximately left-right symmetric. How would you expect this to affect the entries in the correlation matrix between different pixels?

   Solution. I would expect high correlation from the pixels equidistant from the center. ■

Exercise 10.6

1. Find the correlation between six different images. Which images have the highest correlation?

Solution.

```
faces = test_images(:,:,1:6);
dfaces = imresize(faces, [25 25]);
rdfaces = reshape(dfaces, size(dfaces, 1) * size(dfaces, 2), size(
    dfaces, 3));
c = corrcoef(rdfaces)
```

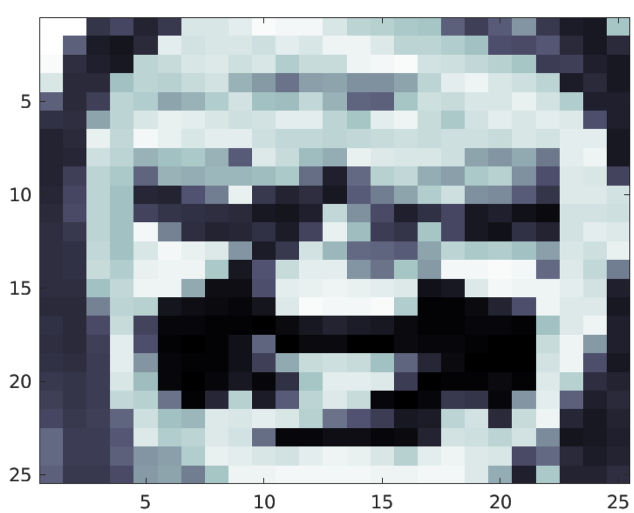
$$\boldsymbol{C} = \begin{bmatrix} 1.0000 & 0.8607 & 0.9713 & 0.9520 & 0.9048 & 0.8380 \\ 0.8607 & 1.0000 & 0.9230 & 0.9497 & 0.8818 & 0.7530 \\ 0.9713 & 0.9230 & 1.0000 & 0.9763 & 0.8958 & 0.7757 \\ 0.9520 & 0.9497 & 0.9763 & 1.0000 & 0.9131 & 0.7809 \\ 0.9048 & 0.8818 & 0.8958 & 0.9131 & 1.0000 & 0.9284 \\ 0.8380 & 0.7530 & 0.7757 & 0.7809 & 0.9284 & 1.0000 \end{bmatrix}$$

The highest correlation is between images 1 and 3. ■

2. Now find the correlation between pixels across images. Try taking a single column of this matrix and reshape that column into an image. What does that image tell you?

Solution.

```
im_corr = corrcoef(transpose(rdfaces));
im_corr_adj = reshape(im_corr(:,1), 25, 25);
imagesc(im_corr_adj)
```



■