

Title: 9.2 Exercises
 Author: Chad Wood
 Date: 12 Feb 2022
 Modified By: Chad Wood
 Description: This program demonstrates the use of python to verify/find eigen(values/vectors) along with approximate principal component calculation of data.

1. Finding the Eigenvalues and Eigenvectors of a Matrix

Let A be the matrix given below.

$$A = \begin{bmatrix} 4 & 0 & 1 \\ -1 & -6 & -2 \\ 5 & 0 & 0 \end{bmatrix}$$

- (a) Is the vector $\begin{bmatrix} 1 \\ 2 \\ 3 \end{bmatrix}$ an eigenvector of A ? Verify your answer with a calculation in Python.
- (b) Is the vector $\begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix}$ an eigenvector of A ? Again, verify your answer with a calculation in Python.
- (c) Use Python to find all of the eigenvalues and eigenvectors of A .

```
In [67]: import numpy as np
A = np.matrix(' 4  0  1;
               '-1 -6 -2;
               ' 5  0  0')

# Eigenvectors determined by testing if multiple contains common factor with V
va = np.array([1, 2, 3]).reshape(3, 1)
vb = np.array([0, 1, 0]).reshape(3, 1)

# Determines if eigenvectors by testing if elements are equal
a = (A@va)==np.nan_to_num((A@va)/va) # False
b = (A@vb)==np.nan_to_num((A@vb)/vb) # True

# Prints boolean of solution
print('Problem (a)s v an eigenvector of A? : ', np.all(a==True))
print('Problem (b)s v an eigenvector of A? : ', np.all(b==True))

# Defines eigenvalues/vectors in iterable
eigs = np.linalg.eig(A)

# c: Prints solution
print(f'Eigenvalues of A: {eigs[0]}\n'
      f'Eigenvectors of A: \n{eigs[1]}')

Problem (a)s v an eigenvector of A? :  False
Problem (b)s v an eigenvector of A? :  True
Eigenvalues of A: [-6.  5. -1.]
Eigenvectors of A:
[[ 0.          0.69431384 -0.18493168]
 [ 1.          -0.18935832 -0.33287702]
 [ 0.          0.69431384  0.9246584 ]]

C:\Users\mrcha\AppData\Local\Temp\ipykernel_8484\3899852140.py:13: RuntimeWarning: invalid value encountered in true_divide
  b = (A@vb)==np.nan_to_num((A@vb)/vb) # True
```

2. Principal Component Analysis

For this exercise, you will be working with the data set video game data.csv. This data has four columns: name of the game, critic score, user score, and total sales. Note: This data set is a modified version of the data set from <https://www.kaggle.com/ashaheedq/video-games-sales-2019>.

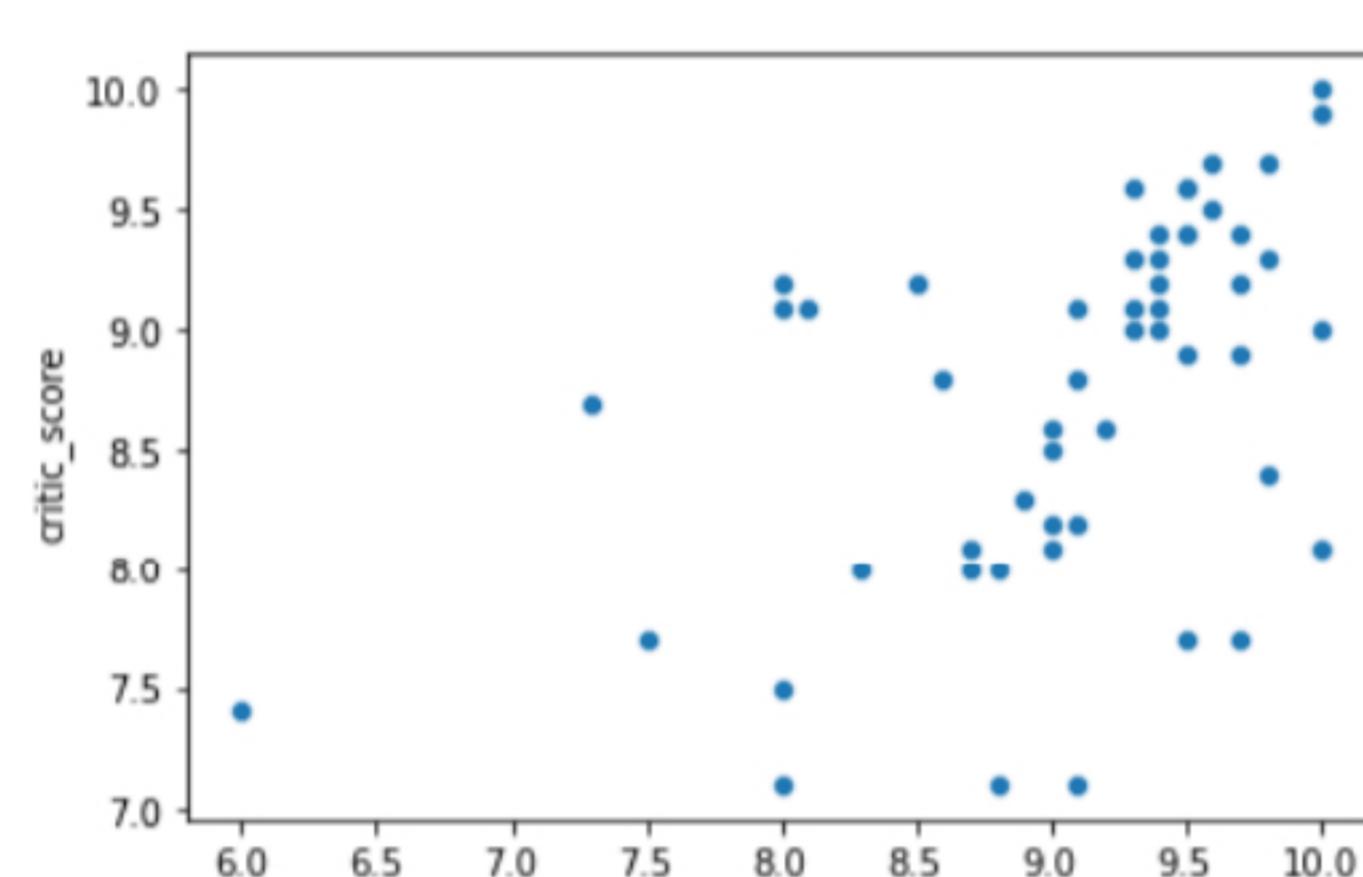
- (a) Make a scatterplot of the user scores versus critics scores.
 (b) On your scatterplot from part (a), sketch the approximate directions of the first and second components. This can be done by hand or in Python.

```
In [74]: import pandas as pd
import seaborn as sns

# Imports data
game_data = pd.read_csv(r'week9data\video_game_data.csv')

# Generates scatter
sns.scatterplot(data=game_data, x=game_data.user_score, y=game_data.critic_score)
```

Out[74]: <AxesSubplot:xlabel='user_score', ylabel='critic_score'>



The approximate direction first and second components are drawn un green / red lines

