20211012_Informationsmanagement_MV1

October 12, 2021

[2]:
```
# "for" Loops
# (Iteration über Sequenzen -- Listen, Tuples, Dict, Strings)
```

[12]:
```
fruits = ['apple', 'banana', 'cherry']

for x in fruits:
    print(x)
```

```
apple
banana
cherry
```

[13]:
```
for x in 'banana':
    print(x)
```

```
b
a
n
a
n
a
```

[15]:
```
# mit dem break sagen wir ihm er sollte stoppen, wenn die Kondition wahr ist

fruits = ['apple', 'banana', 'cherry']

for x in fruits:
    print(x)
    if x=="banana":
        break
```

```
apple
banana
```

[17]:
```
fruits = ['apple', 'banana', 'cherry']

for x in fruits:
    if x =='banana':
```

```
        break # break vor dem print
    print(x)
```

```
apple
```

[19]:
```
# range Function

for x in range(9): # Natürliche Zahlen zw. 0 und die Zahl im Klammer
    print(x)
```

```
0
1
2
3
4
5
6
7
8
```

[21]:
```
for x in range(2,6): # ich kann range auch Anfang bis vor Ende
    print(x)
```

```
2
3
4
5
```

[23]:
```
for x in range(2,30,3): # zwischen 2 und 30, printe jede dritte Zahl
    print(x)
```

```
2
5
8
11
14
17
20
23
26
29
```

[24]:
```
# "else" in einem "for" loop einbauen

for x in range(6):
    print(x)
else:
    print('Endlich sind wir fertig!')
```

```
0
1
2
3
4
5
Endlich sind wir fertig!
```

[29]:
```python
# beispiel Prüfung (Nested Loop)

# dataset wird erstellt

teams_data = [
                {"team_name": "A", "team_scores": [5, 3, None]},
                {"team_name": "B", "team_scores": [4, 2, None]},
                {"team_name": "C", "team_scores": [5, 7, 2]}
             ]

for team in teams_data:

    total_score = 0
    name = team["team_name"]

    for score in team["team_scores"]:

        if score == None:
            print(f"Team {name} Score: Incomplete Data")
            break

        total_score += score

    else:
        print(f"Team {name} Score: {total_score}")
```

```
Team A Score: Incomplete Data
Team B Score: Incomplete Data
Team C Score: 14
```

[30]:
```python
Adjektiv = ['red', 'big', 'tasty']
fruits = ['apple', 'banana', 'cherry']

for x in Adjektiv:
    for y in fruits:
        print(x,y)
```

```
red apple
red banana
red cherry
```

```
big apple
big banana
big cherry
tasty apple
tasty banana
tasty cherry
```

[31]:
```
##########
```

[32]:
```
!pip install numpy
```

```
Requirement already satisfied: numpy in
/Users/h4/opt/anaconda3/lib/python3.8/site-packages (1.19.5)
```

[33]:
```
import numpy
```

[34]:
```
arr = numpy.array([1,2,3,4,5])
print(arr)
```

```
[1 2 3 4 5]
```

[35]:
```
import numpy as np # Abkürzung der Name numpy

arr = np.array([1,2,3,4,5])
print(arr)
```

```
[1 2 3 4 5]
```

[37]:
```
arr = np.array([[1,2,3,4,5],[12,32,54,23,22]])
print(arr)
```

```
[[ 1  2  3  4  5]
 [12 32 54 23 22]]
```

[38]:
```
# Arrays werden genutzt um mehrere Werte von einer Variable zu nennen

cars = ['Ford', 'Volvo', 'BMW']
```

[39]:
```
x = cars[0]
print(x)
```

```
Ford
```

[40]:
```
x = cars[1]
print(x)
```

```
Volvo
```

[42]:
```
x = len(cars) # length oder Länge
print(x)
```

```
3
```

[44]:
```
cars.append('Honda')
print(cars)
print(len(cars))
```

```
['Ford', 'Volvo', 'BMW', 'Honda', 'Honda']
5
```

[52]:
```
# Dimensionen von Arrays

# 0-D Array ist eine Zahl

import numpy as np
arr = np.array(7.5)
print(arr)
```

```
7.5
```

[47]:
```
# 1-D Array ist ein Vektor

import numpy as np
arr = np.array([1,2,3,4,5])
print(arr)
```

```
[1 2 3 4 5]
```

[48]:
```
# 2-D Array sind Matrizen

import numpy as np
arr = np.array([[1,2,3],[4,5,6]])
print(arr)
```

```
[[1 2 3]
 [4 5 6]]
```

[49]:
```
# 3-D Array sind Tensoren

import numpy as np
arr = np.array([[[1,2,3],[4,5,6]],[[7,8,9],[10,11,12]]])
print(arr)
```

```
[[[ 1  2  3]
  [ 4  5  6]]
```

```
 [[ 7  8  9]
  [10 11 12]]]
```

[50]:
```
print(arr.ndim) # die dimensionen vom Tensor werden gezeigt
```

```
3
```

[55]:
```
# Zufallszahlen Generation mit Python

from numpy import random

x = random.randint(100)
# randint generiert ein Zufallszahl zw. 0 und die Zahl im Klammer

print(x)
```

```
83
```

[57]:
```
# Zufallsreelezahl zwischen 0 und 1 generieren

from numpy import random

x = random.rand() # rand generiert eine Zufallsreele Zahl (float) zw. 0 und 1

print(x)
```

```
0.7269980626967778
```

[58]:
```
# Wir können auch den "Shape" (Dimensionen vom Array) bei
# der Zufallszahl eingeben

# Generieren Sie ein 1-D Array mit 5 random Integerzahlen zw. 0 und 100

from numpy import random

x = random.randint(100, size=(5))

print(x)
```

```
[48 73 40 24 91]
```

[61]:
```
# Generieren Sie ein 2-D Array mit 3 Zeilen und 5 Säulen mit
# Zufallsintegerzahlen zw. 0 und 80

from numpy import random

x = random.randint(80, size=(3,5))
```

```
print(x)
```

```
[[12 79 63 34 57]
 [15 59 34 63 12]
 [13 30 37 68 13]]
```

[60]:
```
# Generieren Sie ein 3-D Array mit Dimensionen 4,6,7 mit
# Zufallsintegerzahlen zw. 0 und 100

from numpy import random

x = random.randint(100, size=(5,6,7))

print(x)
```

```
[[[70 15 84 80  2 93 29]
  [13 95 14 84 10 78 82]
  [64 92 22 12  6 22  8]
  [ 9 17 19  1 42 25  8]
  [48 50  5 40 92 59 58]
  [74 73 67 72 78 61 53]]

 [[91 67 56 90 22 74 55]
  [56 97 81 78 62 95 21]
  [35 52 93 67 31 36 60]
  [83 28 17 80 38 72 88]
  [94 32 77 67 43 78 84]
  [51 16 39 86 51 15 29]]

 [[35 48 62 96 63 46 34]
  [15 18 27 54 33 79 71]
  [20 72  5 42  5 63 49]
  [31 51 68 93 35 12 87]
  [47 82 63  1 55 14 44]
  [25 16 60  9 46 14 23]]

 [[75 82 18 46  4 93  6]
  [32 83 63 71 61 32 61]
  [39 32 69  5  6 10 85]
  [ 1 54 93 52 35 65 72]
  [66  7 76 54 12  0 18]
  [90 42  8 70 29 70 52]]

 [[ 4 28 12 86 12 26 50]
  [11 39 50 19 94 73 86]
  [ 6 82 69 35 57 14 93]
  [24 78 13 84 35 19 59]
  [78 65 18 77 27 30 48]
```

```
 [48 56 13 24 61 38 47]]]
```

[62]:
```
# "Choice" erlaubt die generation von Zufallszahlen basiert auf einem␣
↪bestimmten DataArray
```

[64]:
```
from numpy import random

x = random.choice([3,5,7,9])

print(x)
```

```
9
```

[67]:
```
# Generieren Sie einen 1-d Array mit 100 Elemente
# Alle elemente müssen 3, 5, 7, oder 9 sein.
# Die Wahrscheinlichkeit von 3 ist 0.1
# Die Wahrscheinlichkeit von 5 ist 0.3
# Die Wahrscheinlichkeit von 7 ist 0.6
# Die Wahrscheinlichkeit von 9 ist 0

from numpy import random

x = random.choice([3,5 ,7 ,9 ],
                  p   = [0.1, 0.3, 0.6, 0], # p ist die Wahrscheinlichkeit
                  size= (100))

print(x)
```

```
[3 7 5 7 7 7 5 7 5 5 3 3 7 7 7 7 5 7 7 7 5 7 3 7 7 7 3 7 3 5 7 7 7 7 7 7 5
 7 7 7 7 7 5 5 3 5 3 3 5 7 7 7 7 5 5 7 7 7 5 7 7 7 7 3 7 7 7 5 7 7 7 7 5 3
 7 7 5 3 7 5 7 5 7 5 7 3 7 7 5 5 7 3 7 7 5 7 5 7 7 7]
```

[71]:
```
# Permutation (durcheinander würfeln) "shuffle"

from numpy import random
import numpy as np

arr = np.array([1,2,3,4,5])

random.shuffle(arr)

print(arr)
```

```
[2 3 5 4 1]
```

[73]:
```
# Generation einer Normalen Distribution
# Mittelwert (loc) und eine Standard Abweichung (scale)
```

```
from numpy import random

x = random.normal(loc=1, scale=2, size=(2,100))

print(x)
```

```
[[ 4.59237673e+00  7.31341341e-01  1.78011008e+00 -9.38497997e-01
  -5.34402319e-01  2.02725325e+00  2.76316818e+00 -3.55081180e-01
  -1.62507421e-01  1.07523129e+00 -2.82326827e+00  1.10509053e+00
   4.15379352e+00 -8.76953922e-01  1.26239627e+00 -3.08003821e+00
   9.28631966e-01  1.07058273e-02  2.34327704e+00 -3.87324205e-02
   2.36966051e+00  1.37178282e+00  3.47517792e+00  2.30517727e+00
   2.50260770e+00  4.57558097e+00  1.38305022e+00  1.37099255e+00
   1.53046124e+00  7.02860867e-01 -2.37397610e+00  4.98534730e+00
  -3.02540214e-03 -2.10863951e+00 -4.08002078e+00  4.68185484e+00
  -1.08006971e+00  5.80200665e+00 -1.29618341e+00  5.83148450e-01
   2.11093991e+00  1.70530364e+00  2.74319303e+00  2.21229515e-02
   7.36418726e-01 -1.37505405e-01 -6.56163395e-01 -1.18717288e+00
   1.13950423e+00 -1.98033561e+00 -2.43870012e+00 -2.56903704e+00
   5.19115780e-01  6.15937052e-02  4.55010386e+00  2.18786574e+00
   4.81332883e+00 -9.20217400e-01  9.29342408e-01 -2.59502005e+00
   3.24126022e+00  1.57708962e+00  1.87377131e+00  3.32295967e+00
   1.80523673e+00  3.28539210e+00  1.31561788e+00  2.44286305e+00
  -2.50366545e+00 -7.11185412e-02 -4.98246243e-01 -4.47622995e-01
   1.41075057e+00  1.87421530e+00  3.10807059e+00 -3.60079649e-01
  -1.59962947e+00  2.25047536e+00 -9.38071519e-01  6.97037780e-02
   1.33602079e+00  1.84639691e+00  3.01507119e+00  8.15216369e-01
   8.17272240e-01 -3.50389260e+00  2.58932676e-01 -2.37126543e+00
   3.39366392e+00  1.85921088e+00 -1.97223253e+00 -3.18786018e-02
   9.86616738e-01 -3.91675508e-01 -2.72541960e+00  2.76875271e+00
  -1.28664970e+00  1.92191817e+00  2.34732940e+00 -2.10966647e+00]
 [ 5.41240316e-01 -1.55846349e-01 -1.19959950e+00  4.79466852e-01
   9.24021060e-02  2.18793994e-03 -2.16497811e+00 -6.72468051e-01
   2.13763085e+00 -2.28177497e+00  8.68138643e-01  3.07090369e+00
   7.60691821e-01 -3.98174768e-01  1.49726601e+00  2.17002614e+00
   2.33067887e+00  3.85374456e+00  1.22241783e+00  2.65709510e+00
   5.49772328e+00 -6.78551131e-01  4.32351316e+00 -1.96694976e-02
   7.59573380e-01  1.27480038e+00  3.25572458e-01 -8.23708795e-01
   2.75916987e+00 -2.55068912e+00  6.22712357e+00  2.11551814e+00
   3.44729021e+00 -1.11987775e+00 -2.07059098e+00  6.08820214e-01
  -1.67924400e+00  9.63744554e-01  5.77471713e-01  2.41812062e+00
   2.09194087e+00  1.76666621e-02  4.35532880e+00  2.31580326e+00
  -1.58596475e+00  3.07788027e+00 -7.39824163e-01  1.12621046e+00
   3.78541568e+00  2.23152709e+00  4.30338272e-01  3.11171007e+00
  -1.68088177e+00  5.56045779e+00  1.96466098e-01 -1.49155120e+00
  -2.25380460e-02  1.46620808e+00  2.71167510e+00  3.09125626e+00
   3.11356992e+00  2.87962355e-01  5.53380372e-01  3.12685169e-01
```

```
 3.48568492e+00  1.71653811e+00 -2.45922893e+00 -2.80928138e+00
 2.18338218e+00  3.16704837e+00  1.74201432e+00  2.10236412e+00
 1.72655378e+00  4.42429503e-01  3.58754078e+00  2.34325426e-01
 1.26599470e+00 -6.97444213e-01 -3.04533222e-01  1.05939476e+00
-7.38466305e-01  2.22367920e+00  4.65862805e+00  2.81379513e+00
 1.59124012e+00  6.53834418e-01  2.83587484e+00  1.54435405e+00
 4.80620161e+00  1.46505168e+00 -3.94389307e-01 -7.63624674e-01
-2.68580761e+00  2.67880402e+00  1.60886987e+00  3.89560402e+00
-3.57353330e-01  1.18568401e+00  2.03876907e+00  6.44580524e-01]]
```

[75]:
```
!pip install seaborn
!pip install matplotlib
```

```
Requirement already satisfied: seaborn in
/Users/h4/opt/anaconda3/lib/python3.8/site-packages (0.11.1)
Requirement already satisfied: numpy>=1.15 in
/Users/h4/opt/anaconda3/lib/python3.8/site-packages (from seaborn) (1.19.5)
Requirement already satisfied: pandas>=0.23 in
/Users/h4/opt/anaconda3/lib/python3.8/site-packages (from seaborn) (1.2.4)
Requirement already satisfied: matplotlib>=2.2 in
/Users/h4/opt/anaconda3/lib/python3.8/site-packages (from seaborn) (3.3.4)
Requirement already satisfied: scipy>=1.0 in
/Users/h4/opt/anaconda3/lib/python3.8/site-packages (from seaborn) (1.6.2)
Requirement already satisfied: kiwisolver>=1.0.1 in
/Users/h4/opt/anaconda3/lib/python3.8/site-packages (from
matplotlib>=2.2->seaborn) (1.3.1)
Requirement already satisfied: cycler>=0.10 in
/Users/h4/opt/anaconda3/lib/python3.8/site-packages (from
matplotlib>=2.2->seaborn) (0.10.0)
Requirement already satisfied: pillow>=6.2.0 in
/Users/h4/opt/anaconda3/lib/python3.8/site-packages (from
matplotlib>=2.2->seaborn) (8.3.2)
Requirement already satisfied: pyparsing!=2.0.4,!=2.1.2,!=2.1.6,>=2.0.3 in
/Users/h4/opt/anaconda3/lib/python3.8/site-packages (from
matplotlib>=2.2->seaborn) (2.4.7)
Requirement already satisfied: python-dateutil>=2.1 in
/Users/h4/opt/anaconda3/lib/python3.8/site-packages (from
matplotlib>=2.2->seaborn) (2.8.1)
Requirement already satisfied: six in
/Users/h4/opt/anaconda3/lib/python3.8/site-packages (from
cycler>=0.10->matplotlib>=2.2->seaborn) (1.15.0)
Requirement already satisfied: pytz>=2017.3 in
/Users/h4/opt/anaconda3/lib/python3.8/site-packages (from pandas>=0.23->seaborn)
(2021.1)
Requirement already satisfied: matplotlib in
/Users/h4/opt/anaconda3/lib/python3.8/site-packages (3.3.4)
Requirement already satisfied: numpy>=1.15 in
/Users/h4/opt/anaconda3/lib/python3.8/site-packages (from matplotlib) (1.19.5)
```

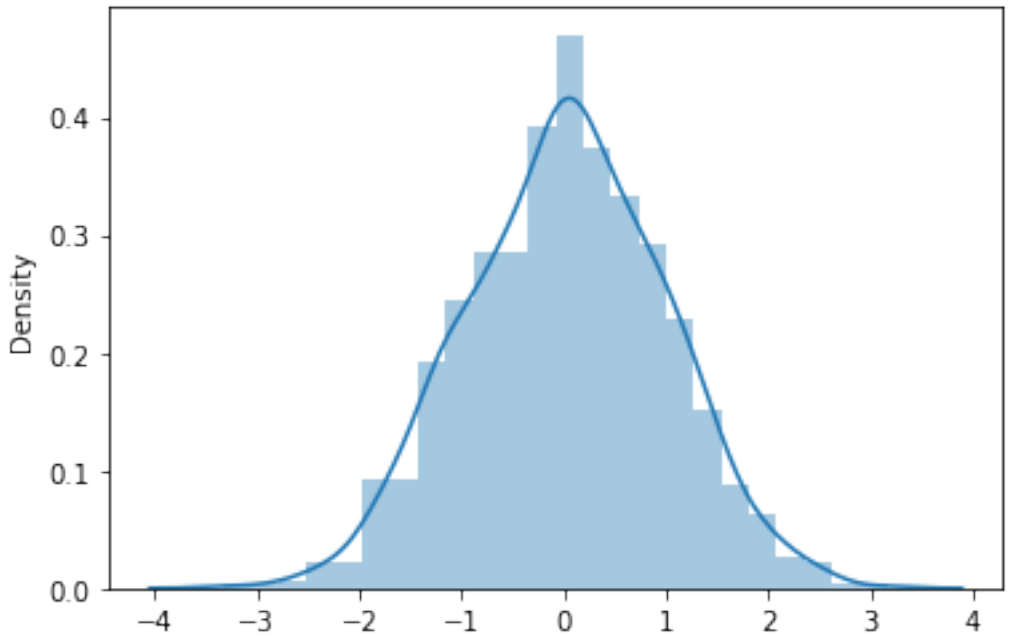
```
Requirement already satisfied: pillow>=6.2.0 in
/Users/h4/opt/anaconda3/lib/python3.8/site-packages (from matplotlib) (8.3.2)
Requirement already satisfied: python-dateutil>=2.1 in
/Users/h4/opt/anaconda3/lib/python3.8/site-packages (from matplotlib) (2.8.1)
Requirement already satisfied: pyparsing!=2.0.4,!=2.1.2,!=2.1.6,>=2.0.3 in
/Users/h4/opt/anaconda3/lib/python3.8/site-packages (from matplotlib) (2.4.7)
Requirement already satisfied: kiwisolver>=1.0.1 in
/Users/h4/opt/anaconda3/lib/python3.8/site-packages (from matplotlib) (1.3.1)
Requirement already satisfied: cycler>=0.10 in
/Users/h4/opt/anaconda3/lib/python3.8/site-packages (from matplotlib) (0.10.0)
Requirement already satisfied: six in
/Users/h4/opt/anaconda3/lib/python3.8/site-packages (from
cycler>=0.10->matplotlib) (1.15.0)
```

[76]:
```
from numpy import random

import matplotlib.pyplot as plt
import seaborn as sns

sns.distplot(random.normal(size=1000))
plt.show()
```

```
/Users/h4/opt/anaconda3/lib/python3.8/site-
packages/seaborn/distributions.py:2557: FutureWarning: `distplot` is a
deprecated function and will be removed in a future version. Please adapt your
code to use either `displot` (a figure-level function with similar flexibility)
or `histplot` (an axes-level function for histograms).
  warnings.warn(msg, FutureWarning)
```



[ ]: