ex2

March 31, 2025

1 Exercise 2

Make the required imports by the notebook

[4]:
```
import numpy as np
import matplotlib.pyplot as plt
from astropy.io import fits
from astropy.table import Table
from astropy import units as u
plt.ion()
import os
```

How to define astopy Quantities from literals - single values or list of values.

[5]:
```
a = 50.0 * u.meter
b = [23, 45, 88] * u.meter
print(a)
print(b)
```

```
50.0 m
[23. 45. 88.] m
```

How to get the mean value of a list using numpy

[6]:
```
np.mean(b)
```

[6]:
52 m

How to get velocity Quantity from distance and time Quantities

15 * u.meter / (3 * u.second)

You can also define distance Quantities in astronomical units.

[7]:
```
x = 62 * u.parsec
y = 45 * u.parsec
print(x)
```

```
62.0 pc
```

When you divide Quantites in the same measurement units, you receive a scalar value - without measurement unit.

[11]:
```
x /y
```

[11]:
1.3777778

You can get the “raw” value of a Quantity using the **value** property

[15]:
```
z = x.value
z
```

[15]:
```
62.0
```

You can also divide “raw” values of Quantities.

[8]:
```
z = x.value / y.value
print(z)
```

```
1.3777777777777778
```

How to round float values.

[18]:
```
np.around(z, decimals=2)
```

[18]:
```
1.38
```

Use the method **type** to get the type of a given variable.

[19]:
```
type(x)
```

[19]:
```
astropy.units.quantity.Quantity
```

Here is a shorthand with whcih you can create a list of Quantities from the “raw” values and the measurement Unit.

[10]:
```
c = [4, 7, 10] * u.second
```

From the examples above, we have distances **b** and moments of time **c**.

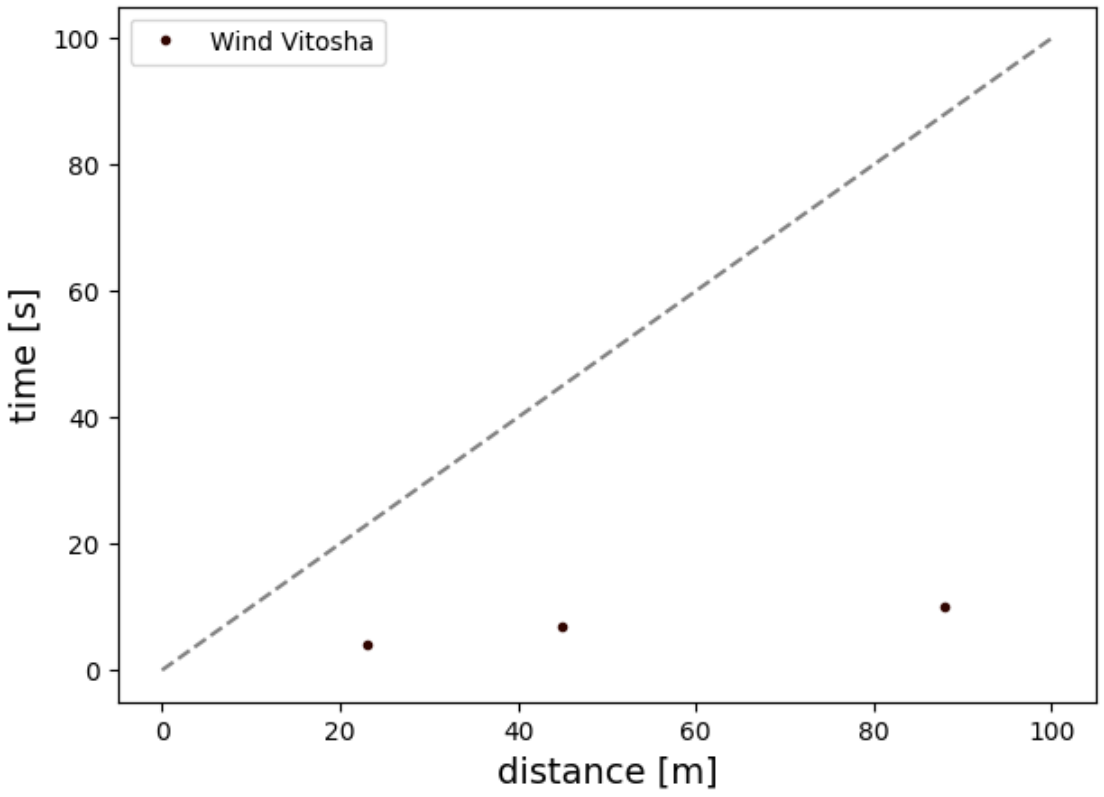
Here is how you can create a plot, in which the x-axis is **b** and y-axis is **c**. Set the appropriate lables and legends.

Then, you can use the **linspace** method from the **numpy** package to crete an array from values from given interval and step. After that simply use the same figure to plot the array.

[11]:
```
plt.figure(figsize=(7,5))
plt.plot(b, c, ls='', color='#300500', marker='.', label='Wind Vitosha')
plt.xlabel("distance [m]", fontsize=14)
plt.ylabel("time [s]", fontsize=14)
plt.legend()
```

```
l1 = np.linspace(0, 100, 2)
l2 = np.linspace(0, 100, 2)
plt.plot(l1, l2, color='gray', ls='--')
```

[11]: [<matplotlib.lines.Line2D at 0x2c3e5579290>]



[ ]: