

2 - Çalışma Ortamı Ayarları

↳ Package management

Paketleri ve birbirine bağımlılıkları yönetir

↳ pip → requirements.txt

pipenv → pipfile

conda → environment.yml

↳ sanal ortam oluşturma

⊕
paket indirme/yönetme

↳ Virtual Environment

↳ conda create -n vername

conda activate vername

conda deactivate vername

conda install name1 name2...

conda remove package_name

conda install numpy → en güncel versiyon

conda upgrade -all → hepsi günceller

conda env rename -n vername

↳ pip : pypi (python package index)

↳ paket yönetim aracı

↳ pip install name

pip install pandas==1.2.1

↓
önceki versiyonu siler, bunu yükler

↳ conda env export > environment.yml

conda env create -f "

2 - Veri Yapıları

↳ List[]

Dictionary x = {"age": 36,
"name": "Tom"}

Tuple x = (... , ... , ...)

Set x = set([1, 2, 3])

↳ int(x) → int yapıyo → 10.5 → 10

float(x) → 10 → 10.0

↳ Çok satırlı string

x = " " " " " " " "

Üçün 2 ye
2 herici

x = "John", x[0] → J x[0:2] → Jo

↳ dir(int) → int ile kullanılabilecek methodlar

↳ fonksiyon → class içinde değil
method → class içinde

↳ str.split() → boşluklara göre bölüp

str.strip() → of of o "o" → of of

↳ list.pop(3) → 3 indextekini siler

list.append("3") → sona 3 ekle

list.insert(2, 3) → 2. indexe 3 sayısını koy, geri kalanı kaydır

↳ dic.update({"b": 11})

↳ eğer b yoksa, yeni eklenen olarak eklenir

↳ Set: Değiştirilebilir
sırasız
eşsiz
kapsayıcı

⚠ Tuple değiştirilemez

↳ discard(8) → 8 yoksa hata vermez
remove(8) → verir

3.1 - Fonksiyonlar

↳ Parametre → Fonksiyon tanımlaması esnasında ifade edilen değişkenler

Argüman \rightarrow Fonksiyon çağırıldığında
fonksiyon adına girilen
değerler

↳ Docstring :

44 11 11

- definition -

Args:

name : type

5

Returns :

type

!

Examples:

3

Notes: —

// // //

↳ Örnekte tanımlı fonksiyon

```
def abc(t, k=1) → abc(1)
                  abc(1, 3)
```

3.2 - Kosullar & Döngüler

↳ for index, student in enumerate(k(list))

print(index, student) → 0 Torq
1 will
2 John

↳ listAll = list(zip(list1, list2, list3))

$$\hookrightarrow [(1, a, \text{cor}), (2, b, \text{Kar}), (3, c, \text{Ser})]$$

değişkenler bölümünde
yaz tutmaz kullan
at fonksiyonu

↳ new-sum = lambda a,b : a+b

↑ ↑ ↑
func name girtiler çıktı

salaries = [10, 20, 30]

```
list(map(new_func, salaries))
```

```
list(filter(lambda x: x%2==0, salaries))
↳ [20]
```

reduce (lambdas a, b : eta b, sabries)

$$\hookrightarrow 60 \rightarrow ((10+20)+30)$$

3.3- Comprehensions

↳ List Comprehension

```
[funcname(salary*2) if salary < 300  
else funcname(salary) for salary in salaries]
```

[salary * 2 for salary in salaries
if salary < 3000]

↳ Dictionary Comprehensions

$$\{k: v \leq 2 \text{ for } (k, v) \text{ in dictionary. Rend}\}$$
$$\{n: n \equiv 2 \text{ for } n \text{ in numbers if } n \% 2 = 0\}$$

4.1 - Python ile Veri Analizi: Numpy

↳ Pandas, Numpy üzerine kurulmuş daha gelişmiş

↳ Matplotlib → low level

Seaborn → high level

↳ Numpy → Numerical Python

↳ Verimli veri saklama ⊕ vektörel (yüksek) işlemler

↳ Fixed type veri tutup

↳ listelere kıyasla daha iyi tek tip veri tuttuğu için

↳ döngü yazmaya ihtiyaç olmayıp array seviyesinde işlemler

↳ Numpy arrayi → ndarray

↳ np.zeros(num, dtype=int)
 → kaç tane → istersen

np.random.randint(0, 10, size=10)

↳ 0 ile 10 arasında 10 random sayı

np.random.normal(10, 4, (3, 4))

ortalama → standart sapma → 3x4 array

↳ a.ndim → boyut

a.shape → kaç tane kaçlık

a.size → toplam eleman sayısı

a.dtype → type bilgisi

a.reshape(3, 3) → sütun
↳ satır

↳ a[0:5] → 0 dan 5 e, 5 yok

a[0, 0] → 0. satır 0. sütun

⚠ int arraye float eklemeye çalışırsan → 2.9 → 2 ekler

↳ a[:, 0] → bütün satırlar, 0. sütun

a[1, :] → 1. satır, tüm sütunlar

a[0:2, 0:3]
 satır sütun

↳ v=np.arange(0, 30, 3) → 0, 3, 6, ..., 27

catch=[1, 5, 8]

v[catch] → 1., 5., 8. indexteki eleman

v[v<3] → v arrayinin 3'ten küçük elemanları

↳ np.subtract(v, 1)

add(v, 1)

mean/sum/min/max/var(v)

↳ $5x_0 + x_1 = 12$ → a=np.array([5, 1], [1, 3])

$x_0 + 3x_1 = 8$ → b= " ([12, 10])

np.linalg.solve(a, b)

4.2 - Python ile Veri Analizi: Pandas

↳ ekonometrik ve finansal çalışmalar

↳ veri okuma, analiz, ml

↳ zamanla bağlı

↳ pandas seri → tek boyutlu

pandas dataframe → çok boyutlu

↳ s=pd.Series([...]) → index bilgisi içi özellik

s.head(x) → ilk x eleman

s.tail(x) → son x "

⚠ fonksiyon ismine ctrl+ye basarak tıklarsa, doküman bilgisi geliyor

↳ df.info() → bilgiler

df.columns → kolon isimleri

df.describe().T → verilerin transpozunu

↳ `df.isnull().values.any()`
 true/false
 true/false array

herhangibirinde true var mı?

`df.isnull().sum()` → neyde ne kadar null var?

↳ `df.drop(...)` → kalıcı hale getirmek için `df = df.drop(...)`

`df["age"]` ile `df.age` aynı şey

↳ `loc` → label based selection
`iloc` → integer based selection

`df.iloc[0:3]` → 0, 1, 2. satır

`df.loc[0:3]` → 0, 1, 2, 3. satır

↳ `df.groupby("sex").agg({"age": "mean"})`

↳ Pivot Table

`df.pivot_table("survived", "sex", "emb")`
 kesisiimdeki ortalama satır sütun

	embarked		
sex	C	Q	S
female			
male			

⚠ Pivot table
 mean alıyo
 emma sonuna
`aggfunc = "std"`
 yazabilirsın

4.3 - Python ile Veri Analizi : Veri Görselleştirme

↳ Kategorik Değişken Görselleştirme

Kategorik değişken görselleştirme → count plot bar (sütun grafik)

Sayısal değişken görselleştirme → histogram (dağılımları gösterir)
 boxplot (aykırı dağılımları da gösterir)

4.4 - Python ile Veri Analizi: Gelişmiş Fonksiyonel Keşifçi Veri Analizi (Advanced Functional EDA)

↳ Kategorik Değişken Analizi

↳ `nunique()` → number of unique

↳ Kategorik → 0, 1, 2 vs ile belirtiyö
 değışkene dikkat int olarak gözüktüyö ama kategorik

↳ Cardinalitesi yüksek değişken → mesela isim değişkeni; çok fazla ve bir anlam ifade etmiyö

↳ Korelasyon Analizi

⚠ Korelasyon -1, 1 arasında

0, -1 arası negatif → biri ↑ diğeri ↓

0, 1 arası pozitif → biri ↑ diğeri ↑

Sınırlara yaklaştıkça ilişki şiddeti kuvvetleniyö

0 civarında korelasyon yok