

Python ve R ile Veri Analizi

HAFTA 8

Dosya İşlemleri

Dr. Öğretim Üyesi Burcu ÇARKLI YAVUZ
bcarkli@sakarya.edu.tr

Dosya işlemleri

- Python ile bir metin dosyası oluşturabiliriz, bu dosyayı okuyabiliriz, dosyaya yazabiliriz ve dosyayı silebiliriz.

Dosya nereye kaydedilecek yada okuyacağım dosya nerde?

- Python ile bir dosya oluşturduğumuzda dosya hangi dizine kaydedilecek?
- Python ile okumaya çalıştığım dosyaya erişim hatası alıyorum.
- Python ile yapacağımız dosya işlemlerinin hangi dizini kullandığını bilmemiz gerekiyor. Bunun için aşağıda verilen kod satırları bize yardımcı olacak. Kodu çalıştırdığımızda ekran çıktısı olarak dosyaların hangi dizinde olması gerektiğini göreceğiz.

```
import os                # os (operating system) modülünü içe aktarır  
print(os.getcwd())      # o an içinde bulunduğumuz klasör dizinini ekrana yazar
```

- Dizini belirterek başka bir dizinde de dosya açabiliriz.

Dosya açmak

- Yeni bir dosya oluşturmak için yada var olan bir dosyayı açmak için `open()` fonksiyonu kullanılır.

```
f = open(dosya_adı, kip)
```

- Kip, bu dosyayı hangi amaçla açtığımızı belirtir, dosya erişim yetkisi verir.

Metin Dosyaları için Dosya Erişim Kipleri

Kip	Yetki açıklaması
"r"	Dosya okuma yetkisi ile açılır. Sadece önceden var olan bir dosya bu kiple açılabilir, dosya yoksa hata mesajı verir. Dosya açılırken kip belirtilmezse varsayılan olarak bu modda açılır. (read)
"w"	Dosya yazma yetkisi ile açılır. Belirtilen isimli bir dosya mevcut değil ise boş dosya açılır, aynı isimli dosya var ise hata vermeden eski dosyanın içeriği silinir. (write)
"a"	Dosya yazma yetkisi ile açılır. Belirtilen isimli bir dosya mevcut değil ise boş dosya açılır, aynı isimli dosya var ise eski dosyanın sonuna ekleme yapar. (append)
"x"	Dosya yazma yetkisi ile açılır. Belirtilen isimli bir dosya mevcut değil ise boş dosya açılır, aynı isimli dosya var ise hata mesajı verir (w kipinden farkı eski dosyayı silmez)
"r+"	Dosya hem okuma hem yazma yetkisi ile açılır. Sadece önceden mevcut olan dosyalar bu kip ile açılabilir.
"w+"	Dosya hem okuma hem yazma yetkisi ile açılır. Dosya mevcut ise içeriği silinir, mevcut değil ise yeni oluşturulur.
"a+"	Dosya hem okuma hem yazma yetkisi ile açılır. Dosya mevcut ise var olan dosyanın içeriğine ilave yapılır, mevcut değil ise yeni oluşturulur.
"x+"	Dosya hem okuma hem yazma yetkisi ile açılır. Dosya mevcut ise hata mesajı verir, mevcut değil ise yeni dosya açar.

Dosya açmak

```
f = open("deneme.txt", "r") # okuma modunda deneme.txt dosyasını açma isteği  
                             (dosya yok ise hata mesajı verir)
```

```
f = open("deneme.txt", "w") # yazma modunda deneme.txt isimli dosya açma isteği  
                             (deneme.txt var ise içeriği siler yok ise boş dosya açar)
```

Farklı bir dizinde dosya oluşturmak istersek:

```
f = open("dosyayı oluşturmak istediğimiz dizin/dosya_adı", kip) # \ kullanma (\n), / kullan  
f = open("C:/Users/Burcu_Yavuz/Desktop/deneme2.txt", "w")    # masaüstünde oluşturma
```

Dosyaya yazmak

- `open()` fonksiyonu ile yazma modunda açtığımız dosyaya `write()` fonksiyonu ile veri yazabiliriz.

```
dosya.write(yazilacak_veri)
```

```
f = open("deneme.txt", "w")          # "w" kipi dikkatli kullanılmalı, var olan dosyayı siliyor  
f.write("Bu satırı deneme.txt dosyasına yazıyorum")
```

- Bu komutları çalıştırdıktan sonra dosya içeriğini boş görüyorsanız veriler dosyaya henüz işlenmemiştir. Bunun için dosyayı kapatmanız gerekiyor.

```
f.close()          # açtığımız dosya ile işimiz bitince mutlaka close() komutu ile kapatalım
```

Dosyaya yazmak

- Bir metin dosyasında tüm veriler aynı satıra yazılır, eğer alt satırlara veri yazmak istiyorsak \n ile alt satıra yazma isteğimizi belirtmemiz gerekiyor.

```
f = open("deneme.txt", "a") # dosyayı "a" kipinde aç, eski dosyanın içine ilave yapacak
f.write("\nSatır1\nSatır2\nSatır3")
f = open("deneme.txt", "r") # dosyayı okuma kipinde aç
print(f.read())             # dosya okuma
f.close()                   # dosyayı kapat
```

Dosya okumak

- Bir dosyayı okumak için "r" kipinde açmamız gerekiyor. Varsayılan kip "r" olduğu için bir şey belirtmesek de dosya okuma modunda açılır.

```
f = open("deneme.txt", "r")
```

yada

```
f = open("deneme.txt")
```

- Okuma modunda açtığımız dosyayı `read()`, `readline()` ve `readlines()` ile okuyabiliriz.

Dosya okumak

- `read()` ile dosyanın tamamı okunur
- `readline()` ile dosya satır satır okunur (çıktı tek bir satır olur)
- `readlines()` dosyanın her bir satırı listenin bir elemanı olacak şekilde liste yapısında çıktı verir.
- Bu komutlar kullanıldığında imleç başa dönmez, dosya okurken imleç nerde kaldıysa bir sonraki okuma işlemi oradan devam eder, imleç dosya sonuna gelmiş ise çıktı vermez.

```
f = open("deneme.txt", "r")  
print(f.read())
```

Dosya okumak

- for döngüsü ile dosya okunabilir.
- for döngüsü ile her bir satır ayrı ayrı okunur

```
f = open("deneme.txt", "r")
for i in f:      # i metindeki her bir satırı dolaşır
    print(i)     # her satır arasında bir boşluk bırakır, bunu istemiyorsak print(i,end="")
f.close()
```

Dosyayı İleri-Geri Sarmak

- Dosya okurken imleç dosya sonuna gelmişse çıktı vermez dedik.
- Peki aynı dosyayı tekrar okumak istersek ne yapacağız?

```
f = open("deneme.txt", "r")  
print(f.read())
```

`f.seek(0)` `#seek()` içindeki parametre, dosya içinde imlecin kaçınıcı byte a gideceğini belirtir, `seek(0)` sıfırıncı byte yani dosyanın en başı

- Artık `print(f.read())` ile dosyayı tekrar okuyabiliriz.

`f.tell()` `# imlecin o anda hangi byte üzerinde olduğunu verir`

Dosya Silme

- Bir dosyayı silmek için os kütüphanesinde remove() fonksiyonu kullanıyoruz.

```
import os  
os.remove("deneme.txt")
```

Kaynaklar

- <https://python-istihza.yazbel.com/>
- Yeni başlayanlar için Python, Ahmet Aksoy, Abaküs yayıncılık