

## Es 1

Nel parco vicino alla scuola è installata una centralina per la misurazione del livello di polveri sottili PM10 nell'aria. Talvolta la centralina va in uno stato di errore per cui la misurazione delle PM10 fallisce ed il dato corrispondente riporta il valore 9999.9 all'interno dell'array delle misurazioni dati. Creare un array di 1000 campioni che generino dei valori secondo una distribuzione normale con valore medio 19 e varianza 5 e supporre che la centralina sia andata in errore 20 volte.

individuare se ci sono due dati errati in finestre di al massimo 5 misurazioni consecutive e in tal caso segnalare che la centralina deve essere mandata in riparazione.

ripulire i dati dai valori errati e calcolare media, massimo e minimo.

```
In [ ]: import numpy as np
```

```
In [ ]: n = 1000
pm10 = np.random.normal(19, 5, n) #creo array con media pari a 19 e devia
print(pm10)
```

```
In [ ]: pm10[np.random.choice(np.arange(n), size=(20,), replace=False)] = 9999.9 #
#print(pm10)
```

```
In [ ]: #per sicurezza conto il numero degli errori
cnt = 0
for element in pm10:
    if element == 9999.9:
        cnt += 1

print(cnt)
```

```
In [ ]: cord = np.argwhere(pm10 == 9999.9) #trovo le coordinate delle celle dove c
#print(cord)
```

```
In [ ]: count = 0
for element in range(pm10.size):#cerco di controllare l'elemento corrente e
    for i in range(5):
        if element + i < pm10.size:
            if pm10[element + i] == 9999.9:
                count += 1
    if count >= 2:
        print("Errore centralina")
        count = 0
    count = 0
```

```
In [ ]: pm10 = np.delete(pm10, np.argwhere(pm10 == 9999.9)) #cancello le occorrenze
print(pm10)
```

```
In [ ]: media = np.mean(pm10)
        minimo = pm10[np.argmin(pm10)]
        massimo = pm10[np.argmax(pm10)]

        print(media, massimo, minimo)
```

## Es 2

Un braccio robotico industriale libero di muoversi avanti e indietro lungo una rotaia è impazzito. Ogni secondo si muove scegliendo a caso tra due possibili movimenti: 1 cm in avanti, oppure 1 cm indietro. Non è possibile togliere corrente al robot senza bloccare tutto lo stabilimento, quindi bisogna attendere lo spegnimento che si effettua tutti i fine settimana e oggi purtroppo è soltanto lunedì! Il nostro responsabile ci chiede di simulare lo spostamento totale che il robot avrà effettuato dopo 5 interi giorni di pazzia. Il robot è partito dalla posizione centrale della rotaia avente coordinata 0 cm.

```
In [ ]: mov = np.random.choice([-1, 1], 432000) #creo l'array formato da 1 e -1
        #print(mov)
```

```
In [ ]: cm = mov.sum() #faccio la somma di ogni elemento presente nell'array
        print(cm/100) #converto in metri
```

## Es 1b

Sul territorio comunale sono installate altre due centraline come la precedente ma queste inviano un segnale errato la prima con una frequenza dello 0.2% mentre la seconda dello 0.1%

Creare una matrice avente 3 colonne e le righe contenente i dati di tutte e 3 le centraline registrati nelle 24 ore (si registra il dato ogni minuto) Eliminare tutte le righe della matrice in cui vi sia almeno una misura errata C'è stato un minuto in cui tutte e tre le centraline hanno inviato un errore?

```
In [ ]: n = 34560 #numero di minuti in 24 ore

pm10C1 = np.random.normal(19, 5, n) #creazione del array senza errori
pm10C1[np.random.choice(np.arange(n), size=(20,), replace=False)] = 9999.9

print(pm10C1) #stampa di controllo
print("\n")

pm10C2 = np.random.normal(19, 5, n)
pm10C2[np.random.choice(np.arange(n), size=(35,), replace=False)] = 9999.9

print(pm10C2)
print("\n")

pm10C3 = np.random.normal(19, 5, n)
pm10C3[np.random.choice(np.arange(n), size=(69,), replace=False)] = 9999.9

print(pm10C3)
```

```
In [ ]: mat = np.array((pm10C1, pm10C2, pm10C3)).transpose() #metto tutte e tre le
print(mat)
```

```
In [ ]: for row in mat:
        if row[0] == row[1] and row[0] == row[-1] and row[1] == row[-1]: #cont
            print("riga uguale")
```

```
In [ ]: cord = np.argwhere(mat == 9999.9) #trovo l'indice della riga dove c'è un e
cord = cord[:,1] #"taglio" l'array per ottenere solo le colonne
print(cord)
```

```
In [ ]: matEliminated = np.delete(mat, cord, axis=1) #elimino la riga contenente l
```

```
In [ ]: cord = np.argwhere(matEliminated == 9999.9) #controllo dell'indice come si
cord = cord[:,1] #"taglio" l'array per ottenere solo le colonne
print(cord)
```

### 3

Crea con Numpy le coordinate dei 4 vertici di 10 poligoni con x e y appartenenti all'intervallo [-100, 100] Dire quali sono dei rettangoli, dei rombi, dei parallelogrammi e quadrati

```
In [ ]: arr = np.random.uniform(-100, 100, size=(10,4)) #creazione di un vettore r
print(arr)
```