Generowanie danych

Anna Zadka, Natalia Klepacka, Kinga Teklak

20 czerwca 2023

Spis treści

0.1 klienci

0.1.1 Imię i nazwisko

Imiona i nazwiska zostały pobrane ze strony **gov**, oddzielnie dla kobiet i mężczyzn. Następnie połączyłyśmy imiona damskie z nazwiskami oraz imiona i nazwiska męskie. Na koniec dane dla kobiet i mężczyzn zostały przemieszane. **https://dane.gov.pl/pl/dataset/1667,lista-imion-wystepujacych-w-rejestrze-pesel-osoby-zyjace**

0.1.2 e-mail

E-maile tworzyłyśmy z imion i nazwisk, oddzielonych kropką. Adres poczty zależy od podzielności indeksu nazwiska. W celu zachowania realizmu najczęstszym adresem poczty elektronicznej jest @*gmail.com*.

```
maile =[]
for idx in range(len(nazwiska)):
    if idx % 17 == 0:
        mail = imiona[idx] + "." + nazwiska[idx] + "@interia.com"
    elif idx % 10 == 0:
        mail = imiona[idx] + "." + nazwiska[idx] + "@02.com"
    elif idx % 9 == 0:
        mail = imiona[idx] + "." + nazwiska[idx] + "@poczta.onet.com"
    elif idx % 23 == 0:
        mail = imiona[idx] + "." + nazwiska[idx] + "@poczta.wp.com"
    else:
        mail = imiona[idx] + "." + nazwiska[idx] + "@gmail.com"

    maile.append(mail)
```

Rysunek 1: Tworzenie e-maili.

0.1.3 numer telefonu

Numery telofonu są tworzone przez losowanie liczby pomiędzy 555555555 a 777777777 dla każdej osoby.

```
telephon  = np.random.randint(low=555555555, high=888888888, size=(100,))
telefon = [str(tel) for tel in telephon]
telephon = pd.DataFrame([telephon])
```

Rysunek 2: Tworzenie numerów telefonu.

0.2 gry

0.2.1 nazwa, gatunek, opis, minimalny wiek i możliwa liczba graczy

Te informacje zostały skopiowane ze strony: **https://www.gracula.pl/wypozyczalnia-listagier**.

0.2.2 wydawnictwo

Wydawnictwo zostało wylosowane ze zwracaniem z podanych: *"Nasza Księgarnia","Aleksander","Trefl","Muduko","Panini","Zielona Sowa","Rebel","Albi","Fishbone Games"*.

0.2.3 cena wypożyczenia i kupna

Ceny zostały stworzone poprzez wylosowanie liczby do 10, w przypadku wypożyczeń i do 30 w przypadku kupna. Następnie dodano do nich 0.99.

```
cena_wypozyczenia = []
for i in range(50):
    cena_wypozyczenia.append(str(np.random.choice(10) + .99))


cena_kupna = []
for i in range(50):
    cena_kupna.append(str(np.random.choice(30) + .99))
```

Rysunek 3: Tworzenie cen wypożyczeń i kupna gier.

0.3 pracownicy

0.3.1 imię, nazwisko, płeć, numer telefonu

Od początku działania sklepu zostało zatrudnionych dziesięciu pracowników. Ich imiona, nazwiska i płeć zostały wylosowone jak wyżej za pomocą danych z **gov**. Numery telefonu zostały wylosowane jak powyżej.

0.3.2 data urodzenia

Data urodzenia pracowników jest wybierana jako data między rokiem 1970, a 1990.

```
urodziny = []
for _ in range(10):
    dzien =int( np.random.randint(low=1, high=28, size=(1,)))
    miesiac = int(np.random.randint(low=1, high=12, size=(1,)))
    rok = int(np.random.randint(low=1970, high=1990, size=(1,)))
    urodziny.append(str(rok)+ "-"+ str(miesiac)+ "-" + str(dzien))
```

Rysunek 4: Tworzenie daty urodzeń pracowników.

0.3.3 adres

Adresy składają się z wylosowanej ulicy we Wrocławiu lub jego okolicy(Kamień, Kiełczówek, Krzyżanowice, Łany), numeru domu od 1 do 120, miejscowości i kodu pocztowego.

```
adress = []
ulica = ["Zielona", "Motylkowa", "Nagietkowa", "Katowicka", "Cytrynowa", "Lubczykowa", "Morska", "Piastowska",
         "Graniczna","Gruszowa" ]
miasto = ["Wrocław","Wrocław","Kamień","Wrocław","łany","Wrocław","Wrocław","Krzyżanowice","Wrocław","Kiełczówek"]

for idx in range(10):
    numer = int(np.random.randint(low=1, high=120, size=(1,)))
    kod1 = str(int(np.random.randint(low=50, high=56, size=(1,)))) + "-" +
    str(int(np.random.randint(low=504, high=510, size=(1,))))
    adress.append(str(ulica[idx]) + " " + str(numer) + "; " + str(miasto[idx]) + " " + kod1)
```

Rysunek 5: Tworzenie adresów pracowników.

0.3.4 data zatrudnienia

Datę zatrudnienia pracowników tworzymy wybierając losową datę między rokiem 2013, a 2020.

```
zatrudnienie = []
for _ in range(10):
    dzien =int( np.random.randint(low=10, high=28, size=(1,)))
    miesiac = int(np.random.randint(low=1, high=12, size=(1,)))
    rok = int(np.random.randint(low=2013, high=2020, size=(1,)))
    zatrudnienie.append(str(rok)+ "-"+ str(miesiac)+ "-" + str(dzien))
```

Rysunek 6: Tworzenie daty zatrudnienia.

0.4 pensje

0.4.1 pensja

Pensje losujemy dla każdego pracownika(w przypadku zmiany kwoty pensji kolejny raz), wybierając kwote spośród następujących: 6653, 3500, 4400, 4530, 3550.

0.5 sprzedaże

Losujemy liczbę sprzedaży z rozkładu Poissona z powiększającą się zmienną λ (zaczynamy od 10 i zwiększamy o 0.001) dla każdego dnia działania sklepu. Następnie losujemy czas sprzedaży (czyli godzinę między 10 a 17 i minuty między 0 i 59), ponieważ założyłyśmy czas działania wypożyczalni od 10 do 18. Na koniec losujemy tytuł kupionej gry, który został sprzedany(używamy tutaj **random.choice**, ponieważ losujemy ze zwracaniem).

```
sprzedaze = pd.DataFrame({
    "id_gry":[],
    "data":[]
})

for i, dzien in enumerate(dni):

    if i%7==5:

        # sprzedaze

        liczba_spr = scipy.stats.poisson.rvs(mu=10+0.001*i)
        minuty = [random.randint(0, 59)  for _ in range(liczba_spr)]
        godziny_spr = [str(random.randint(10, 17))+":"+str(m) if m>=10 else str(random.randint(10, 17))+":0"+str(m) for m in minu

        for i in range(liczba_spr):
            sprzedaze.loc[len(sprzedaze)] = [random.choice(sklep_id), str(dzien)+" "+godziny_spr[i]]
```

Rysunek 7: Tworzenie sprzedaży dla każdego dnia.

0.6 turnieje, wyniki

Sposób generowania omówimy na przypadku dla roku 2023. Klasyfikacje odbyły się w kwietniu, finał natomiast się jeszcze nie odbył(dlatego w bazie nie ma id finału i wyników turnieju). Losujemy zawodników z listy zawodników i ich liczb, która jest maksymalną liczbą jaką dopuszcza gra do kwadratu. Robimy to, ponieważ w klasyfikacjach bierze udział więcej drużyn, a każdy kto wygra w swojej drużynie kwalifikuje się do finału. Jeśli zawodnik przechodzi do finału do punktów w tabeli wyniki wpisujemy 1, w przeciwnym wypadku 0. W finale za pierwsze miejsce zawodnik dostaje 2 punkty, za ostatnie 0, a każdy kto nie zajął pierwszego lub ostatniego miejsca otrzymuje 1 punkt.

0.7 wypożyczenia

Dla każdego dnia tygodnia roboczego losujemy liczbę wypożyczeń z rozkładu Poissona z rosnącym parametrem λ(zaczynając od 2 i zwiększając o 0.001). Jeśli liczba wypożyczeń jest większa niż 0, to losujemy tytuł z dostępnych

```
from random import sample
tunieje_23 = []
wyniki_23 = []
id_turnieju_23 = []
for i,num in enumerate([5,4,5,5,6,4]):
    i = i+1
#klasyfikacja
    uniqlist = list(range(1, 51))
    zawodnicy = sample(uniqlist, num**2)

    if i < 4:
        tunieje_23+=zawodnicy
        wyniki_23+=[1]*num
        wyniki_23+=[0]*(num**2 - num)
        id_turnieju_23+=[(i*2-1)+24]*num**2
#turniej

    if i<3:
        id_turnieju_23+=[(i*2)+24]*num
        tunieje_23+=zawodnicy[:num]

        wyniki_23+=[2]
        wyniki_23+=[1]*(num-2)
        wyniki_23+=[0]
```

Rysunek 8: Tworzenie tabeli turnieje i wyniki.

gier(używamy tutaj **random.sample**, ponieważ losujemy bez zwracania). Następnie wybieramy czas jej wypożyczenia (czyli godzinę między 10 a 17 i minuty między 0 i 59), ponieważ założyłyśmy czas działania wypożyczalni od 10 do 18. Data zwrotu to liczba z rozkładu Poissona z $\lambda = 2$ i średnią równą 3. Dla każdego zwrotu również losujemy czas analogicznie jak dla wypożyczeń.

```
gry_dostepne = asortyment.id.values.tolist()
gry_wypozyczone = []
for i, dzien in enumerate(dni):

    if i%7==5:

        # wypozyczenia

        liczba_wyp = min(scipy.stats.poisson.rvs(mu=2+0.001*i), len(gry_dostepne))

        if liczba_wyp > 0:
            gry_do_wyp = random.sample(gry_dostepne, liczba_wyp)
            minuty = [random.randint(0, 59)  for _ in range(liczba_wyp)]
            godziny_wyp = [str(random.randint(10, 17))+":"+str(m) if m>=10 else str(random.randint(10, 17))+":0"+str(m) for m in
            czas = scipy.stats.poisson.rvs(mu=3, loc=3, size=liczba_wyp)

            data_zwrotu = [dzien+timedelta(days=int(k)) for k in czas]
            minuty = [random.randint(0, 59)  for _ in range(liczba_wyp)]
            godziny_zwrotu= [str(random.randint(10, 17))+":"+str(m) if m>=10 else str(random.randint(10, 17))+":0"+str(m) for m i

            klienci = random.choices(klienci_id, k=liczba_wyp)

    gry_wyp_2 = []
    for gra in gry_wypozyczone:
        if gra[1]>0:
            gry_wyp_2.append((gra[0], gra[1]-1))
        else:
            gry_dostepne.append(gra[0])
    gry_wypozyczone = gry_wyp_2[:]
```

Rysunek 9: Tworzenie wypożyczeń dla kazdego dnia działania sklepu.