Wykład 4. Podsumowania i agregacje danych oraz wybrane metody statystyczne

Michał Sochański

Plan wykładu

1. Powtórka w zakresie najważniejszych statystyk opisowych służących do opisu zmiennej.

2. Zastosowanie omówionych statystyk do podsumowań i agregacji danych w Pythonie.

3. Badanie relacji pomiędzy dwiema zmiennymi za pomocą wybranych technik statystycznych wraz z zastosowaniem w Pythonie.

1. Statystyki opisowe - najważniejsze miary tendencji centralnej i rozproszenia - przegląd

- W tej części wykładu omówimy najważniejsze metody podsumowania zbioru wartości liczbowych lub jakościowych.
- W naszym przypadku będziemy mieli na celu podsumowanie jednej zmiennej w obiekcie DataFrame.
- Będziemy rozważać podsumowania zmiennych:
  - numerycznych (typ int, float) oraz
  - kategorialnych (typ object, category)

Miary tendencji centralnej i rozproszenia

- Najczęściej stosowanymi miarami są:

1) Miary tendencji centralnej („typowa” wartość w zbiorze)
2) Miary tendencji rozproszenia (miara zróżnicowania wartości czy też ich „rozrzucenia” wokół wartości „typowej”)

Miary tendencji centralnej		**Miary rozproszenia**	
Średnia arytmetyczna	Numeryczne	Odchylenie przeciętne	Numeryczne
Mediana	Numeryczne	Odchylenie standardowe	Numeryczne
Dominanta	Kategorialne	Wariancja	Numeryczne
		Kwartyle	Numeryczne
		Liczba kategorii	Kategorialne

Inne funkcje dające informacje o zmiennej

- Można zastosować również inne funkcje działające na zmiennej oraz zwracające pojedynczą wartość liczbową będącą agregatem lub inną informacją o zmiennej. Oto najważniejsze z nich:

- Liczba wartości
- Suma
- Iloczyn
- Inne operacje arytmetyczne
- Średnia geometryczna i harmoniczna
- Maksimum
- Minimum
- Kwantyle

Średnia

Średnia arytmetyczna to suma wartości występujących w danym zbiorze podzielona przez liczbę tych elementów.

Przykład:

Oceny na koniec roku:

Polski – 4

Angielski – 3

Matematyka – 5

Biologia – 2

Geografia – 3

$$x_{śr} = \frac{suma\_ocen}{liczba\_przedmiotów} = \frac{4+3+5+2+3}{5} = 3{,}4$$

Mediana

Mediana to „środkowa” wartość w zbiorze, jeśli ustawimy wartości z tego zbioru po kolei. W przypadku, gdy mamy parzystą liczbę elementów, mediana to średnia z dwóch wartości środkowych.

Przykład:

Oceny na koniec roku:

Polski – 4

Angielski – 3

Matematyka – 5

Biologia – 2

Geografia – 3

5,4,3,3,2

Po posortowaniu ocen, „środkową” wartością jest 3.

Dominanta

Dominanta to taka wartość, która w zbiorze występuje najczęściej.

Przykład:
Oceny na koniec roku:
Polski – 4
Angielski – 3
Matematyka – 5
Biologia – 2
Geografia – 3

Najczęściej występującą oceną jest 3.

Przykład 1.

Id	Zarobki
1	2000
2	2100
3	2200
4	2300
5	2300
6	2500
7	2500
8	2500
9	7000
10	20000

Mamy następujący zbiór respondentów z podanymi zarobkami.

Średnia = 4540

Mediana = 2400

Dominanta = 2500

Miary rozproszenia

Miary rozproszenia informują nas, jak bardzo wartości jakiejś zmiennej są rozrzucone wokół średniej. Podstawowe miary rozproszenia liczy się w następujący sposób:

- **Odchylenie przeciętne** (średnia odległość wartości ze zbioru od ich średniej)

$$d = \frac{1}{N}\sum_{i=1}^{N} | x_i - x_{śr} |$$

- **Odchylenie standardowe**

$$\text{sd} = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(x_i - x_{śr})^2}$$

- **Wariancja**
- $var = sd^2$

N – ilość elementów w danym zbiorze (tu – zmiennej), $x_{śr}$ średnia wartość dla elementów tego zbioru

Przykład 2.

Id	Zarobki
1	2000
2	2100
3	2200
4	2300
5	2300
6	2500
7	2500
8	2500
9	7000
10	20000

Mamy następujący zbiór respondentów z podanymi zarobkami.

Odchylenie przeciętne = 3584

Odchylenie standardowe = 5342,883

Wariancja = 28 546 400

Kwartyle

- **Kwantyle** to najogólniej wartości badanej cechy, które dzielą ją na określone części po względem liczby jednostek.
- **Kwartyle** to szczególny typ kwantyli:

➢ Kwartyl pierwszy (dolny) – dzieli zbiorowość uporządkowaną na dwie części w ten sposób, że 25% jednostek ma wartości cechy niższe, a 75% wyższe od kwartyla pierwszego.

➢ Kwartyl drugi to inaczej mediana.

➢ Kwartyl trzeci (górny) – dzieli zbiorowość uporządkowaną na dwie części w ten sposób, że 75% jednostek ma wartości cechy niższe, a 25% wyższe od kwartyla trzeciego.

- W analogiczny sposób można definiować inne **percentyle**, tzn. wielkości odpowiadające innym wartościom procentowym.

2. Statystyki opisowe w Pythonie

- Omówione statystyki możemy policzyć dla wybranych zmiennych albo dla całego zbioru (wszystkich zmiennych)
- Przykład 1.- zliczamy liczbę wszystkich wartości w zmiennej („count”)

  a. dla pojedynczej zmiennej
  - data['zmienna'].count()

  b. dla wybranych zmiennych
  - data[['zmienna1', 'zmienna2']].count()

  c. dla całego zbioru
  - data.count()

- Każdą z dalej wymienionych funkcji możemy policzyć na te trzy sposoby. Będziemy jednak wymieniać tylko wersję dla pojedynczej zmiennej.
- Uwaga – wszystkie analizowane dalej funkcje pomijają w obliczeniach braki danych!

Statystyki opisowe w Pythonie (2)

2. Średnia

- data['Nazwa_zmiennej'].mean()

3. Mediana

- data['Nazwa_zmiennej'].median()

4. Dominanta

- data['Nazwa_zmiennej'].mode()

#Uwaga – jeśli jest więcej niż jedna dominanta, Python wyświetli wszystkie z nich

#Uwaga2 – funkcja zadziała także dla zmiennych numerycznych

Statystyki opisowe w Pythonie (3)

5. Odchylenie przeciętne

- data['Nazwa_zmiennej'].mad()

6. Wariancja

- data['Nazwa_zmiennej'].var()

7. Odchylenie standardowe

- data['Nazwa_zmiennej'].std()

Statystyki opisowe w Pythonie (4)

5. Kwantyle

```
data['Nazwa_zmiennej'].quantile(0.25)    #dolny kwartyl
data['Nazwa_zmiennej'].quantile(0.75)    #górny kwartyl
data['Nazwa_zmiennej'].quantile(0.1)     #dziesiąty percentyl
```

6. Maksimum, minimum

- `data['Nazwa_zmiennej'].max()`
- `data['Nazwa_zmiennej'].min()`

Statystyki opisowe w Pythonie (5)

7. Suma

```
data['Nazwa_zmiennej'].sum()
```

- Przypomnienie – wybrane statystyki opisowe dla zmiennej numerycznej możemy wyświetlić za pomocą komendy:

```
data['Nazwa_zmiennej'].describe()
```

Statystyki opisowe w Pythonie (6)
Zmienne kategorialne

8. Wyświetlamy wszystkie odmienne kategorie.

```
data['Nazwa_zmiennej'].unique()
```

9. Wyświetlamy liczbę wszystkich wartości w zmiennej.

```
data['Nazwa_zmiennej'].count()
```

10. Dominanta.

```
data['Nazwa_zmiennej'].mode()
```

Liczymy statystyki opisowe dla podzbiorów

Jeśli chcemy policzyć jakieś statystyki dla podzbioru zmiennej, możemy po prostu najpierw zastosować filtrowanie a potem policzyć odpowiednie statystyki dla wybranego podzbioru. Można to zrobić np. w następujący sposób.:

```
data_m = data[data.gender == 'M']  #tworzymy nowy zbiór (podzbiór zbioru
                                    wyjściowego, zawierający tylko mężczyzn
data_m.wiek.mean()                 #obliczymy średni wiek mężczyzn
```

Funkcja „groupby”

- Jeśli chcemy policzyć statystyki opisowe osobno dla każdej kategorii występującej w danej zmiennej (lub zmiennych), najlepiej zastosować funkcję „groupby”.
- Funkcja „groupby” działa na zasadzie „split-apply-combine, tzn.:
  - Dzieli zbiór na podzbiory zgodnie z kategoriami występującymi we wskazanej zmiennej (lub zmiennych)
  - Wykonuje na każdym podzbiorze wskazaną operację
  - Łączy wyniki w nowym obiekcie – w tym przypadku jest to „Series” albo „DataFrame”
- Procedurą taką nazywamy **agregacją** danych (liczymy statystyki dla grup obserwacji wyznaczonych przez kategorie zmiennej grupującej)

Funkcja „groupby”

- Ogólna struktura funkcji „groupby”:

Data . groupby . zmienne_grupujące . zmienne_wynikowe . Funkcja

1) Data – nazwa zbioru .
2) Groupby – polecenie grupujące.
3) „Zmienne grupujące”– zmienna lub zmienne – dla każdej kategorii tych zmiennych (lub ich kombinacji) będziemy liczyć wartość wybranej funkcji (statystyki).
4) „Zmienne wynikowe” – zmienne lub zmienna, na podstawie których chcemy liczyć statystykę.
5) Funkcja – określona funkcja statystyczna, którą chcemy zastosować do zmiennych wynikowych.

- Wykonamy kilka przykładowych analiza na zbiorze (typu DataFrame) o nazwie „Pracownicy”.
- Grupujemy po wybranej zmiennej („zawód”) a następnie liczymy średnią ze *wszystkich* zmiennych numerycznych w zbiorze.

Uwaga – w wynikowym zbiorze (również typu DataFrame) kategorie ze zmiennej „zawód” umieszczone są w etykietach wierszy.

Imię	płeć	zawód	wiek	staż
Marcin	M	elektryk	36	23
Ewa	K	programista	32	2
Ola	K	programista	25	12
Tomek	M	sprzedawca	24	1
Ania	K	elektryk	29	29
Elwira	K	sprzedawca	55	48
Grzegorz	M	elektryk	33	32
Jan	M	sprzedawca	27	9
Filip	M	elektryk	51	23
Róża	K	programista	49	52

```
In [266]: Pracownicy.groupby('zawód').mean()
Out[266]:
                 wiek       staż
zawód
elektryk    37.250000  26.750000
programista 35.333333  22.000000
sprzedawca  35.333333  19.333333
```

Grupujemy po zmiennej „płeć”. Liczone jest odchylenie standardowe tylko ze zmiennej wybranej w kwadratowym nawiasie.

Uwaga – tym razem wynik jest typu „Series” ale kategorie ze zmiennej „płeć” umieszczone są jak poprzednio w etykietach wierszy.

Imię	płeć	zawód	wiek	staż
Marcin	M	elektryk	36	23
Ewa	K	programista	32	2
Ola	K	programista	25	12
Tomek	M	sprzedawca	24	1
Ania	K	elektryk	29	29
Elwira	K	sprzedawca	55	48
Grzegorz	M	elektryk	33	32
Jan	M	sprzedawca	27	9
Filip	M	elektryk	51	23
Róża	K	programista	49	52

```
In [270]: Pracownicy.groupby('płeć')['wiek'].std()
Out[270]:
płeć
K    13.190906
M    10.521407
Name: wiek, dtype: float64
```

Grupujemy po wybranej zmiennej . Liczone jest maksimum tylko ze zmiennych wybranych w podwójnym kwadratowym nawiasie.

Wynik jest typu DataFrame.

Imię	płeć	zawód	wiek	staż
Marcin	M	elektryk	36	23
Ewa	K	programista	32	2
Ola	K	programista	25	12
Tomek	M	sprzedawca	24	1
Ania	K	elektryk	29	29
Elwira	K	sprzedawca	55	48
Grzegorz	M	elektryk	33	32
Jan	M	sprzedawca	27	9
Filip	M	elektryk	51	23
Róża	K	programista	49	52

```
In [272]: Pracownicy.groupby('zawód')[['wiek','staż']].max()
Out[272]:
             wiek  staż
zawód
elektryk       51    32
programista    49    52
sprzedawca     55    48
```

Grupujemy po dwóch czynnikach podanych w liście. Liczona jest średnia z wszystkich zmiennych numerycznych w zbiorze.

Uwaga – tym razem w etykietach wierszy występują dwie zmienne, zastosowane zostało więc tzw. indeksowanie „hierarchiczne”

Imię	płeć	zawód	wiek	staż
Marcin	M	elektryk	36	23
Ewa	K	programista	32	2
Ola	K	programista	25	12
Tomek	M	sprzedawca	24	1
Ania	K	elektryk	29	29
Elwira	K	sprzedawca	55	48
Grzegorz	M	elektryk	33	32
Jan	M	sprzedawca	27	9
Filip	M	elektryk	51	23
Róża	K	programista	49	52

```
In [263]: Pracownicy.groupby(['zawód', 'płeć']).mean()
Out[263]:
                      wiek  staż
zawód       płeć
elektryk    K     29.000000  29.0
            M     40.000000  26.0
programista K     35.333333  22.0
sprzedawca  K     55.000000  48.0
            M     25.500000   5.0
```

Tym razem używamy dodatkowo metody „agg”, aby wyświetlić wyniki dla trzech statystyk. Grupujemy więc po zawodzie i szukamy osób z największym i najmniejszym stażem, jak również ilość elementów w każdej z podgrup (a więc osób wykonujących poszczególne zawody).

Imię	płeć	zawód	wiek	staż
Marcin	M	elektryk	36	23
Ewa	K	programista	32	2
Ola	K	programista	25	12
Tomek	M	sprzedawca	24	1
Ania	K	elektryk	29	29
Elwira	K	sprzedawca	55	48
Grzegorz	M	elektryk	33	32
Jan	M	sprzedawca	27	9
Filip	M	elektryk	51	23
Róża	K	programista	49	52

```
In [275]: Pracownicy.groupby('zawód')['staż'].agg(['max','min','count'])
Out[275]:
              max  min  count
zawód
elektryk       32   23      4
programista    52    2      3
sprzedawca     48    1      3
```

3. Badanie relacji pomiędzy dwiema zmiennymi za pomocą wybranych technik statystycznych wraz z zastosowaniem w Pythonie.

- Istnieje wiele technik badania i mierzenia siły związku pomiędzy wieloma zmiennymi. My przypomnimy tutaj trzy techniki służące do mierzenie związku pomiędzy dwoma zmiennymi:
    1. Test T
    2. Test χ^2
    3. Współczynnik korelacji liniowej Pearsona i Spearmana

- Zaczniemy od krótkiej powtórki ze statystyki. Uwaga – nie będzie to pełny wykład, a jedynie przypomnienie najważniejszych intuicji i pojęć.

Testy statystyczne: zmienne losowe (powtórka)

- Zmienna losowa
  - Intuicyjnie, zmienna losowa to pewna wielkość (liczbowa), która może charakteryzować badane przez nas obiekty (obserwacje).
  - Zmienna losowa może przyjmować różne wartości. W szczególności, zmienne losowe mogą być dyskretne (liczby całkowite lub naturalne) albo ciągłe (liczby rzeczywiste).
  - Zmienne losowa przyjmuje poszczególne wartości z różnym *prawdopodobieństwem*.
- Przykłady:
  - Różne możliwe wyniki rzutu kostką.
  - Liczba reszek uzyskanych w trzech rzutach monetą.
  - Różne możliwe wartości wzrostu badanych osób.
  - Ilość energii elektrycznej zużywanej dziennie przez zakład pracy.

Testy statystyczne: rozkłady (powtórka)

- Rozkład zmiennej losowej
  - Rozkład zmiennej losowej mówi nam, intuicyjnie, z jakim prawdopodobieństwem zmienna losowa przyjmuje poszczególne wartości.
  - Bardziej formalnie, mówimy o funkcji rozkładu prawdopodobieństwa zmiennej losowej. Wartością tej funkcji, dla każdej możliwej wartości zmiennej losowej, jest prawdopodobieństwo uzyskania tej wartości.
  - Rozkłady charakteryzują się różnymi parametrami. Najczęściej występujące to średnia i wariancja.
- Przykłady:
  - Rzut kostką: prawdopodobieństwo uzyskania każdej wartości jest takie samo i równe 1/6.
  - Potrójny rzut monetą. P(0 reszek) = 1/8, P(1 reszka)=3/8, P(2 reszki)=3/8, P(3 reszki=1/8)
  - Wzrost: intuicyjnie, najwięcej będzie osób o „średnim” wzroście, natomiast osób wysokich i niskich będzie mniej.

Rozkład jednostajny i dwumianowy

1. Rozkład jednostajny: prawdopodobieństwo uzyskania każdej wartości jest takie samo.

Rzut kostką: prawdopodobieństwo uzyskania każdej wartości jest równe 1/6.

Losując liczbę z przedziału (0,1) za pomocą funkcji „np.random.rand” również mamy do czynienia z rozkładem jednostajnym.

2. Rozkład dwumianowy: tutaj mamy do czynienia z wielokrotnie powtarzanym doświadczeniem, które może zakończyć się „sukcesem” bądź „porażką”, przy czym za każdym razem dane jest stałe prawdopodobieństwo uzyskania sukcesu. Rozkład określa, z jakim prawdopodobieństwem uzyskamy określoną liczbę sukcesów.

Parametry: ***n*** – liczba prób, ***p*** – prawdopodobieństwo sukcesu

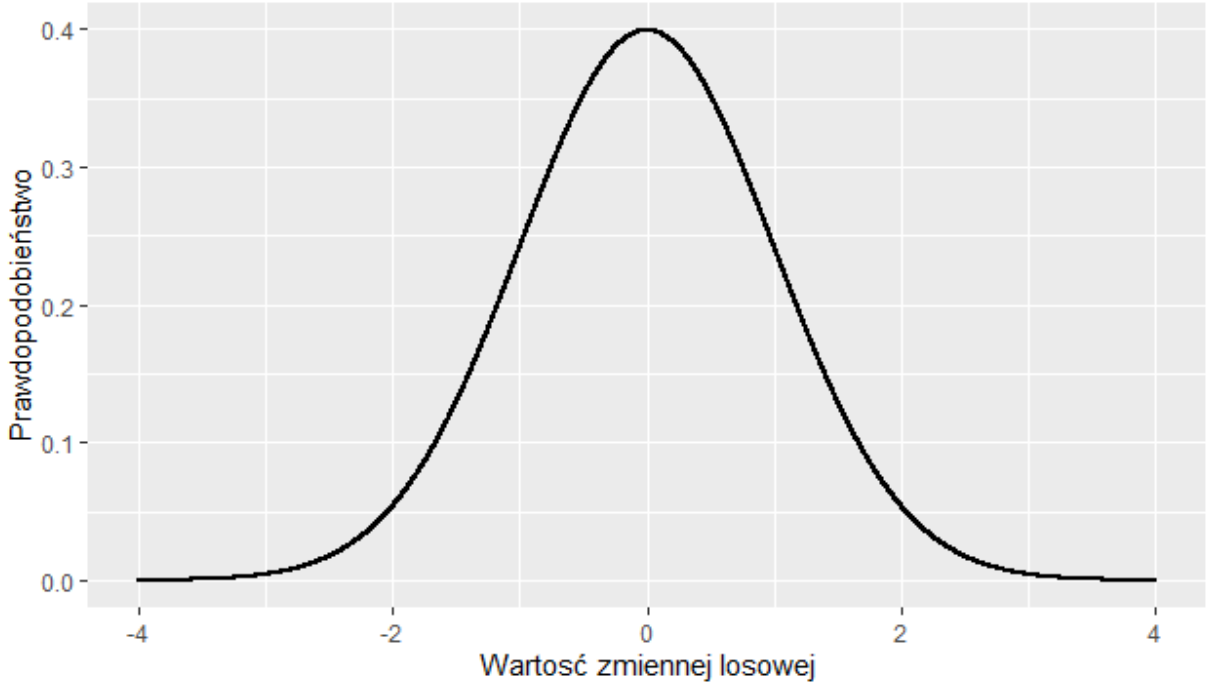
Przykład: potrójny rzut monetą. P(0 reszek) = 1/8, P(1 reszka)=3/8, P(2 reszki)=3/8, P(3 reszki=1/8).

Liczba prób =3, prawdopodobieństwo sukcesu = 1/2.

Rozkład normalny (rozkład Gaussa)

3. Rozkład normalny to najważniejszy i najczęściej spotykany rozkład prawdopodobieństwa

- Parametry: μ – średnia, σ – odchylenie standardowe (σ^2 wariancja)
- Obok widzimy wykres funkcji dla rozkładu o parametrach μ=0, σ=1.



Rozkłady i symulacje w Pythonie

- Tak możemy generować „próbki losowe” z niektórych rozkładów

```
import scipy.stats as st
st.uniform.rvs(0,5, size=10) #rozkład jednostajny, losujemy 10 liczb z
                             przedzialu (0,5)

st.binom.rvs(3, 0.5, size=8) #rozkład dwumianowy – seria trzech powtórzeń
                             eksperymentu z prawdopodobieństwem
                             sukcesu=0.5, powtórzona 8 razy

st.norm.rvs(2,7, size=1000) #rozkład normalny, średnia=2, odchylenie
                            standardowe = 7, próbka o 1000 elementach
```

Testy statystyczne: ogólna idea

- Za pomocą testów statystycznych możemy poddać analizie założenia dotyczące własności rozkładów badanych przez nas zmiennych losowych (inaczej mówiąc – własności czy cech, co do których dysponujemy próbami z większej populacji).
- Mogą to być testy parametryczne (dotyczące pewnych parametrów rozkładu) albo nieparametryczne (dotyczące innych ich własności).
- Statystyka testowa – wartość pewnej funkcji, którą stosujemy do naszej próby.

Testy statystyczne: ogólna idea

Ogólna procedura jest następująca:

- Dysponujemy próbą danej wielkości z populacji i na podstawie tej próby chcemy wnioskować o jakiejś własności rozkładu całej populacji.
- Określamy hipotezę zerową i alternatywną (hipotezą zerową może być na przykład stwierdzenie: średni wzrost mieszkańców Poznania to 1,67m).
- Określamy statystykę testową – funkcję określoną na próbie.
- Ustalamy (jako fakt), że przy założeniu hipotezy zerowej, statystyka ma określony rozkład.
- Liczymy statystykę testową dla naszej próby.
- Sprawdzamy, jakie jest prawdopodobieństwo uzyskania takiej próby przy założeniu hipotezy zerowej (na podstawie rozkładu statystyki testowej)

Testy statystyczne: ogólna idea

- Przyjmujemy poziom istotności α, np. $\alpha=0.05$.
- Korzystając ze statystyki testowej możemy popełnić błąd: hipoteza zerowa może być prawdziwa, ale my wylosujemy próbę o znacząco różnym rozkładzie niż ten z populacji i odrzucimy (błędnie) hipotezę.
- Przy poziomie istotności $\alpha=0.05$ godzimy się na to, że jedna próba na 20 doprowadzi nas to niepoprawnego wniosku.
- Określamy również obszar krytyczny – zakres wartości statystyki testowej, dla których odrzucamy hipotezę zerową.

Test T (test t-Studenta)

- Testu T używamy zazwyczaj w jednej z trzech sytuacji:

1) Mamy dwie próby danej wielkości z dwóch różnych populacji; chcemy przekonać się czy średnia z tej wielkości jest dla tych populacji taka sama.
   - przykład: porównujemy wynik testu z matematyki dla uczniów z dwóch szkół.

2) Mamy dwie próby danej wielkości z tej samej populacji, pobrane w innych punktach czasowych; chcemy przekonać się czy średnia z tej wielkości jest dla tych populacji taka sama.
   - przykład: porównujemy wynik testu z matematyki tej samej grupy uczniów – przed i po ukończeniu kursu e-learningowego

3) Mamy jedną próbę danej wielkości i chcemy przetestować hipotezę dotyczącą średniej z tej wielkości.

Test T

- Uwaga: Nie będziemy omawiać przypadku 3).

- Ogólnie rzecz biorąc, test T pomaga nam odpowiedzieć na pytanie: czy różnica pomiędzy średnimi z dwóch prób może być przypadkowa (tzn. charakterystyczna tylko dla danej próby) przy faktycznej równości średnich dla obu populacji, czy też rzeczywiście średnie dla obu populacji są różne.
- Pytamy więc: zakładając hipotezę zerową – jaka jest szansa uzyskania próby, którą dysponujemy?
- Hipoteza zerowa: średnie są równe.
  - H_0: $m_1 = m_2$
- Hipoteza alternatywna: średnie są różne.
  - H_1: $m_1 \neq m_2$

Test T - założenia

- Próby są losowe.
- Rozkład powinien być w przybliżeniu normalny (w szczególności dla małych prób)
- Wariancje obu prób powinny być podobne
- W naszym zastosowaniu w Pythonie zakłada się też, że próby są równoliczne

Test T w Pythonie

Niech *x* oraz *y* będą naszymi próbami losowymi, zapisanymi jako obiekty *ndarray* (bądź pd.Series, tzn. kolumny DataFrame'u).

```
import scipy.stats as st
st.ttest_ind(x,y)   #test dla prób niezależnych
st.ttest_rel(x,y)   #test dla prób zależnych
```

Test T w Pythonie

- Wybrane dodatkowe argumenty:
- st.ttest_rel(x,y, nan_policy = 'omit') #ignorowanie braków danych. Domyślnie braki danych w próbie skutkują brakiem danych na wyjściu

- st.ttest_ind(x,y, equal_var = False) #domyślnie mamy True, czyli test zakłada równość wariancji. W przypadku, gdy są one (istotnie) różne, należy wprowadzić ten argument – przeprowadzony wtedy będzie „w zastępstwie” inny test (test Welcha)

Test T – interpretacja wyników

- Output komendy ttest jest prosty. Składa się z:

1) Wartości statystyki testowej.
2) Wartości *p*.

Interesuje nas wartość *p*. Można ją interpretować jako prawdopodobieństwo wylosowania naszej próby przy założeniu hipotezy zerowej. Jeśli przyjmiemy np. $\alpha=0.05$, to przy $p > \alpha$ możemy stwierdzić, iż nie ma podstaw do odrzucenia hipotezy zerowej.

Test χ^2

- Za pomocą testu chi-kwadrat weryfikujemy hipotezę o niezależności dwóch zmiennych jakościowych (nominalnych lub porządkowych). Nie jest to więc test parametryczny.
- Badamy dwie zmienne jakościowe *X* i *Y* o liczbie kategorii odpowiednio *r* oraz *k*.
- Hipoteza zerowa: cechy *X* i *Y* są niezależne.
- Do weryfikacji tej hipotezy stosuje się statystykę, która – przy dużych próbach – ma rozkład χ^2 o $(r\text{-}1)(k\text{-}1)$ stopniach swobody.

Test χ^2

- Punktem wyjścia testu χ^2 jest tabela kontyngencji dla zmiennych *X* oraz *Y*, którą trzeba stworzyć i podać jako argument funkcji.
- W tabeli kontyngencji podajemy liczbę obserwacji dla każdej kombinacji różnych wartości, które przyjmują zmienne *X* oraz *Y*.
- Z założenia o niezależności zmiennych wynika, że – intuicyjnie rzecz biorąc – wartości powinny być rozłożone „równomiernie” w tabeli.
- Dokładniej, zakłada się, że wartość z *n*-tego rzędu i *k*-tej kolumny powinna być równa iloczynowi wszystkich wartości z *n*-tego rzędu i wszystkich z *k*-tej kolumny podzielonemu przez łączną liczbę wartości w tabeli.

Test χ^2

- Przykład:

		papierosy		ogółem
		pali	**nie pali**	
wykształcenie	**podstawowe**	28	14	42
	średnie	19	19	38
	wyższe	10	37	47
ogółem		57	70	127

- Przy założeniu hipotezy zerowej, osób, które palą i mają wykształcenie podstawowe powinno być ok. (42*57)/127 = 18.85

- Python:

st.chi2_contingency([[zmienna1],[zmienna2]])

W wyniku otrzymujemy:

  - Wartość statystyki testowej
  - „*p*-value”
  - Liczbę stopni swobody
  - Wartości „teoretyczne”, tzn. wartości, których oczekiwalibyśmy przy założeniu prawdziwości hipotezy zerowej.

- Wynik („*p*-value”) można interpretować jako prawdopodobieństwo, że uzyskalibyśmy analizowaną próbę przy założeniu, że zmienne są niezależne.

Test χ^2 - założenia

- Aby test działał dobrze, próba powinna być duża, przyjmuje się, że przynajmniej 30 obserwacji
- Liczba obserwacji dla każdej kombinacji kategorii ze zmiennych *X* oraz *Y* powinna być większa od 5. W przeciwnym przypadku należy połączyć niektóre kategorie.

Współczynnik korelacji liniowej Pearsona

- Współczynnik korelacji liniowej Pearsona stosowany jest do dwóch zmiennych **numerycznych**.
- Przyjmuje on wartości z przedziału [-1,1].
- Współczynnik ten mierzy siłę korelacji **liniowej** pomiędzy zmiennymi.
- Im większa wartość bezwzględna współczynnika, tym większa korelacja, przy czym znak wskazuje na kierunek korelacji (wartość ujemna oznacza „odwrotną” korelację; oznacza to mniej więcej, że im większe wartości przyjmuje pierwsza zmienna, tym mniejsze przyjmuje druga)

Współczynnik korelacji liniowej Pearsona

- Załóżmy, że r to wartość bezwzględna współczynnika korelacji. Przyjmuje się często, że:
  - Jeśli $r < 0.3$, korelacja jest bardzo słaba albo jej nie ma.
  - Jeśli $0.3 < r <= 0.5$, to korelacja występuje w stopniu umiarkowanym.
  - Jeśli $0.5 < r <= 0.7$, to korelacja jest silna.
  - Jeśli $0.7 < r < 1$, to korelacja jest bardzo silna.
  - Jeśli $r = 1$, to mamy do czynienia ze związkiem funkcyjnym (liniowym) pomiędzy zmiennymi.
- Uwaga – poszczególne wartości korelacji interpretuje się różnie w zależności od kontekstu czy dziedziny badań.

Współczynnik korelacji liniowej Pearsona

st.pearsonr(zmienna1,zmienna2)

Zmienne mogą być znów kolumnami DataFrame’a, obiektami typu np.array albo zwykłymi listami.

W wyniku, pierwsza podana liczba to wartość korelacji. Druga to wynik testu statystycznego, którego nie będziemy tutaj bliżej omawiać.

Współczynnik korelacji kolejnościowej (rang) Spearmana

- Stosowany jest najczęściej do dwóch zmiennych **porządkowych**.
- Współczynnik ten przyjmuje wartość z przedziału [-1,1] i jego interpretacja jest identyczna jak w przypadku współczynnika korelacji liniowej Pearsona.
- Python

```
import scipy.stats as st

st.spearmanr(zmienna1,zmienna2)
```

Polecane kursy na platformie DataCamp

- Introduction to Statistics in Python
- Statistical Thinking in Python

Ćwiczenia

Ćwiczenie 1.

Otwórz w Pythonie zbiór „USstates.txt”. Zawiera on dane dotyczące 50 stanów USA, zbierane w latach 70-tych. Bardziej szczegółowe informacje o zbiorze (w tym o znaczeniu zmiennych) można znaleźć na stronie: https://www.rdocumentation.org/packages/datasets/versions/3.6.1/topics/state (uwaga - nie wszystkie zmienne z oryginalnego zbioru zawarte są w zbiorze dostępnym w MS Teams)

1. Jaka jest średnia oczekiwana długość życia w Stanach Zjednoczonych?
2. Znajdź wszystkie trzy kwartyle dla zmiennych „Population” i „Income”.
3. Który stan ma największą powierzchnię, a który najmniejszą?
4. Wymień 5 stanów z najwyższą średnią zarobków.

Ćwiczenie 1 (cd).

5. Zmienna „Division” grupuje stany według regionu. Stanów z którego regionu jest najwięcej?
6. Grupując po zmiennej „Division” podaj:
   - Średnie zarobki dla każdego regionu
   - Odchylenie standardowe dla zmiennej „Illiteracy” w każdym regionie
   - Stan z największą populacją w każdym regionie
7. Rozważ zmienne „Income”, „Illiteracy”, „Life Exp” i „Murder”. Dla których par zmiennych występują największe korelacje? Jakie stąd wypływają wnioski?

Ćwiczenie 2.

- Wygeneruj dwa wektory o 100 elementach, tak aby wartość każdej liczby była wylosowana z rozkładu normalnego o średniej równej 0 i odchyleniu standardowym równym 1. Przypisz je do zmiennych o wybranych nazwach.
- Zbadaj korelację pomiędzy zmiennymi i użyj testu T aby sprawdzić czy średnie w tych próbach istotnie się różnią.
- Powtórz powyższe obliczenia dla następujących par wektorów:
  - Rozkład normalny o śr.=0 i odch.st.=1 oraz rozkład normalny o śr.=0 i odch.st.=4
  - Rozkład jednostajny w przedziale (3,5) i rozkład normalny o śr.=4 i odch.st.=2
  - Wektor1: liczby naturalne od 1 do 100; wektor2: liczby naturalne od 101 do 200

Ćwiczenie 3.

W folderze zawarty jest zbiór „countries_of_the_world.csv” z wybranymi danymi demograficznymi i geograficznymi dotyczącymi wszystkich państw świata (państwa stanowią więc obserwacje, a każdy wiersz odpowiada jednym państwu). Zbiór został pobrany ze strony www.kaggle.com, informacje o nim można uzyskać na stronie https://www.kaggle.com/search?q=countries+of+the+world.

Uwaga1: większość zmiennych numerycznych zostanie wczytanych jako typ „object”. Aby je zmienić na typ numeryczny należy najpierw zamienić w liczbach przecinki na kropki (data.zmienna=data.zmienna. str.replace(',','.')), a następnie zamienić je na numeryczne (funkcja pd.to_numeric – patrz materiały z wykładu 3).

Uwaga2: Tworząc nowe zmienne musimy korzystać z notacji nawiasowej , np.

```
data['nowa_zmienna'] = data ['istniejaca_zmienna']>2
```

Korzystając ze zbioru odpowiedz na zamieszczone poniżej pytania.

1. Jaka jest średnia populacja w każdym z regionów (zmienna „Region”)?
2. Dla ilu krajów występują braki danych w zmiennej „Literacy (%)”

Ćwiczenie 3 (cd)

3. W którym kraju występuje największa gęstość zaludnienia („Pop.density..”)?

4. Ile jest krajów z populacją powyżej 20 mln w każdym z regionów?

5. Jaka jest średnia i odchylenie standardowe ze zmiennej „Agriculture” (odsetek osób zatrudnionych w rolnictwie)? Jak kształtują się te wielkości dla poszczególnych regionów?

6a. Stwórz nową zmienną logiczną, która przyjmuje wartość „True”, kiedy wartość zmiennej „Birthrate” jest większa lub równa od wartości zmiennej „Deathrate”, a wartość „False” w przeciwnym przypadku. Wartość „True” w nowej zmiennej oznacza – najprościej ujmując – że w roku badania w danym kraju więcej osób się urodziło niż umarło.

Ćwiczenie 3 (cd)

6b. Pogrupuj dane po regionie a następnie po nowo utworzonej zmiennej. Sprawdź, ile jest krajów w każdej z podgrup (przykładem podgrupy będzie Region=BALTICS oraz nowa_zmienna=True). Sprawdź również jaka jest średnia z „GDP ($ per capita)” w każdej z podgrup.

7. Sprawdź w jakim stopniu skorelowane są następujące pary zmiennych:

- „GDP ($ per capita)” i „ Infant mortality (per 1000 births)”
- „GDP ($ per capita)” i „ Net migration”
- „Population” i „Pop. Density (per sq. mi.)”
- „ Literacy (%)” i „Agriculture”

Pamiętaj o usunięciu braków danych!

Ćwiczenie 4.

Wczytaj do Pythona plik „Titanic.csv”, z którego korzystaliśmy już przy ćwiczeniach z poprzedniego wykładu. Przeprowadź test χ^2 dla zmiennych „Class” oraz „Survived” aby przekonać się czy szanse przetrwania uczestników katastrofy powiązane były z tym, w jakiej klasie podróżowali. Aby to zrobić musisz stworzyć tabelę kontyngencji dla tych dwóch zmiennych. Aby to zrobić, możesz np. pogrupować dane po tych dwóch zmiennych i policzyć ilość osób z poszczególnych przecięć kategorii (np. Class=1st, Survived = Yes) przez obliczenie sumy po wartościach ze zmiennej „Freq” dla każdego przecięcia. Tabelę kontyngencji można stworzyć „ręcznie”, przy użyciu funkcji pd.crosstab albo „pivot”.

Dokonaj podobnej analizy dla par zmiennych „Age”, „Survived” oraz „Sex”, „Survived”. Która ze zmiennych „Age”, „Sex” i „Class” była najsilniej powiązana ze zmienną „Survived” – zgodnie z wynikami testu χ^2?