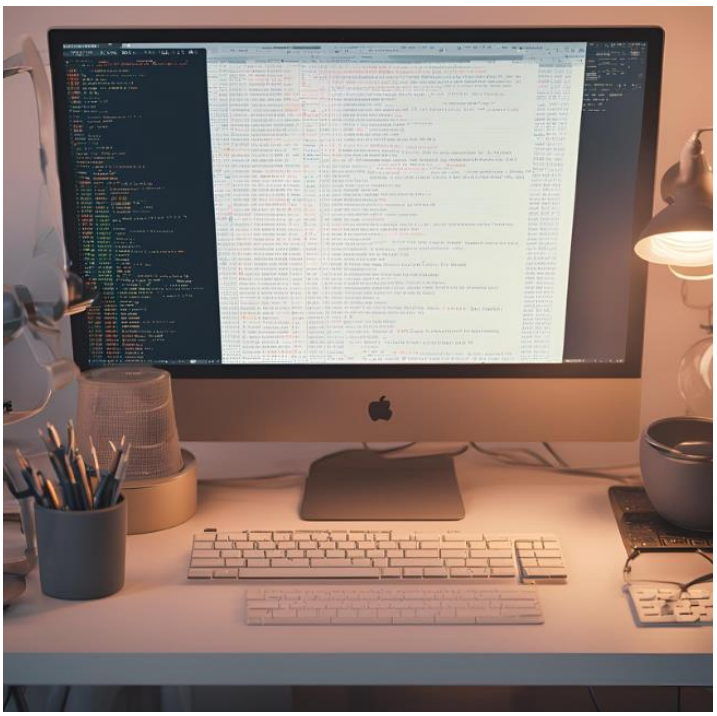
Fundusze Europejskie dla Rozwoju Społecznego
Rzeczpospolita Polska
Dofinansowane przez Unię Europejską

Ćwiczenie nr 7
Testy statystyczne w Excelu oraz języku R



POLITECHNIKA BYDGOSKA
im. Jana i Jędrzeja Śniadeckich

POLITECHNIKA BYDGOSKA
Wydział Technologii i Inżynierii Chemicznej

POLITECHNIKA BYDGOSKA
Wydział Medyczny

PRACOWNIA KOMPUTEROWA

Wstęp

Współczesne analizy statystyczne mają fundamentalne znaczenie w badaniach naukowych, diagnozowaniu problemów medycznych, a także w podejmowaniu decyzji biznesowych i społecznych. Korzystanie z narzędzi takich jak Excel oraz język R pozwala na szybkie i efektywne przetwarzanie danych, umożliwiając analizowanie dużych zbiorów danych, wykonywanie obliczeń oraz przeprowadzanie testów statystycznych, które pomagają zweryfikować hipotezy badawcze.

- Excel (z włączonym pakietem Analysis ToolPak) oferuje podstawowe funkcje statystyczne oraz wizualizacje, które są proste i intuicyjne. Dodatkowo dostępny jest do instalacji pakiet Real Statistics (Xrealstats) ze strony:
  https://real-statistics.com/free-download/real-statistics-resource-pack/
  Instalacja pakietu: Plik->Opcje->Dodatki->Przejdź->Przeglądaj (znajdź pobrany ze strony pakiet)
  Problemy przy instalacji:
  - Trzeba wyłączyć Widok chroniony w Opcjach Excela (Plik->Opcje->Centrum zaufania)
  - Trzeba odblokować plik ściągnięty z Internetu we Właściwościach tego pliku
- Język R to specjalistyczne narzędzie programistyczne, które pozwala na zaawansowane analizy i większą elastyczność, co czyni go idealnym narzędziem dla badaczy oraz analityków danych. Zaletą jest brak konieczności instalacji dodatkowych programów na komputerze i możliwość skorzystania z kompilatorów online. Np.:
  - https://posit.cloud (wymaga założenia darmowego konta, dostęp do projektów z dowolnego miejsca, możliwość przesyłania plików, możliwość instalacji pakietów)
  - https://rdrr.io/snippets/ (zaleta: brak konieczności logowania się, dużo zainstalowanych pakietów; wada: brak możliwości przesyłania plików z danymi)
  - https://www.mycompiler.io/pl/new/r (zaleta: brak konieczności logowania się; wady: mniej pakietów zainstalowanych, brak możliwości przesyłania plików z danymi)

Aby przygotować dane do analiz w języku R, pochodzące z baz danych, które często są w formacie tekstowym .csv lub .tsv, można pomóc sobie Excelem (wczytać dane za pomocą Kreatora importu tekstu), a następnie:

Sposób 1: użyć konwerter dostępny online jak np.:https://tableconvert.com/excel-to-rdataframe Konwerter ten pomoże przygotować dane w formacie wymaganym przez język R.

Sposób 2: użyć dowolnego AI wpisując odpowiedni prompt, np.: Zamień przesłaną tabelę na język R (i następnie wklejamy zaznaczone wcześniej w Excelu kolumny). Odpowiedź powinna wyglądać przykładowo tak (z użyciem ChatGPT 6):

Poniżej tabela przepisana do R jako ramka danych (`data.frame`). Pominąłem 6 pustych wierszy na końcu.

R

```r
dane <- data.frame(
  masa = c(
    18, 41, 74, 59, 62, 22, 71, 12, 79, 74,
    76, 62, 60, 80, 75, 60, 68, 61, 94, 83,
    65, 98, 73, 70, 71, 97, 82, 60, 87, 97
  ),
  wzrost = c(
    120, 156, 156, 165, 151, 134, 156, 99, 153, 160,
    162, 158, 164, 158, 151, 164, 160, 171, 184, 185,
    187, 178, 173, 176, 183, 181, 178, 190, 178, 168
  )
)
```

Cel

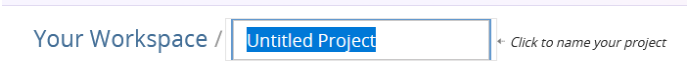
Celem ćwiczenia jest praktyczne zastosowanie wybranych testów statystycznych zarówno w Excelu, jak i w języku R. Studenci nauczą się, jak:

- Wprowadzać i zarządzać danymi w obu narzędziach.
- Wykonywać podstawowe testy statystyczne, takie jak test t-Studenta, test chi-kwadrat, aby sprawdzić zależności i różnice w zestawach danych.
- Interpretować wyniki testów statystycznych, co jest kluczowe w procesie analizy danych.
- Zrozumieć, w jaki sposób wybrane testy mogą odpowiedzieć na różne pytania badawcze, zależnie od rodzaju danych oraz stawianych hipotez.

Przebieg ćwiczenia

1. Zabawy z językiem R

Wejdź na stronę: https://posit.cloud i załóż konto. Pamiętaj aby zapisać w bezpiecznym miejscu **login** i **hasło** aby mieć dostęp do swoich projektów.
Utwórz nowy projekt RStudio (przycisk po prawej stronie ekranu) i nadaj mu nazwę, np. Cwiczenie 6:

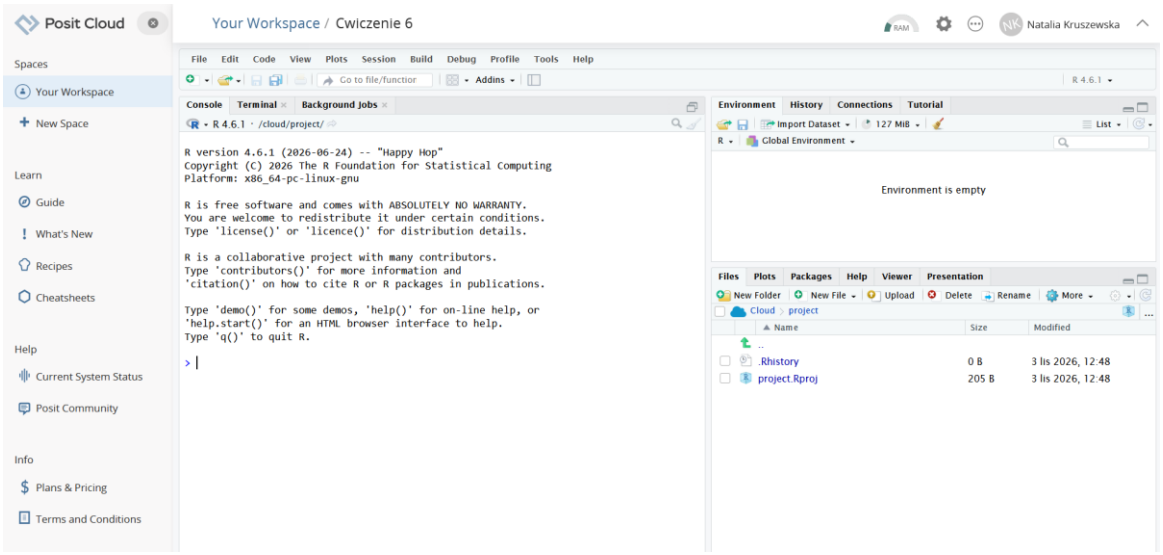


Okno projektu podzielone jest na kilka elementów jak na rysunku poniżej. Lewy panel nie jest potrzebny i możesz go zamknąć przyciskiem X. Jeśli będzie później potrzebny to otwiera się go ikonką ☰ w lewym górnym rogu.

Na ekranie pozostaje:

- okienko terminala języka R (**Console**), gdzie możemy wykonywać pojedyncze komendy i oglądać rezultat ich wykonania. Prompt konsoli to znaczek >.
- Okienko ze zmiennymi środowiskowymi (**Environment**).
- Okienko z plikami projektu. Tutaj będzie można zarówno wgrywać pliki jak i pobierać. W tym oknie jest jeszcze jedna ważna zakładka: **Packages**, gdzie będziemy instalować pakiety statystyczne wymagane aby język R rozumiał niektóre nasze polecenia.

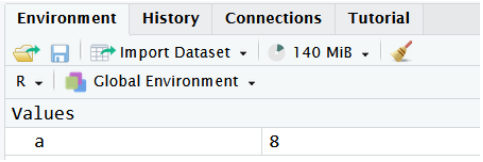
Fundusze Europejskie dla Rozwoju Społecznego
Rzeczpospolita Polska
Dofinansowane przez Unię Europejską



Rysunek 1 Wygląd okna nowego projektu

Zadanie 1

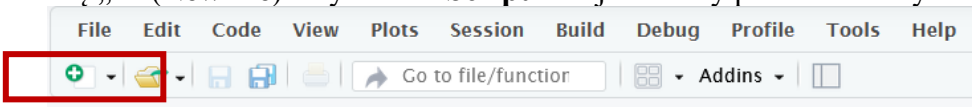
a) Ustaw kursor myszy za promptem konsoli (Console). Dla przykładu wpisz w konsole: **a=3+5** i naciśnij Enter. W konsoli nic nie zostało wypisane ale po prawej stronie w górnym panelu **Environment** pojawił się wpis, że w naszym środowisku została ustalona zmienna a i ma ona wartość 8:



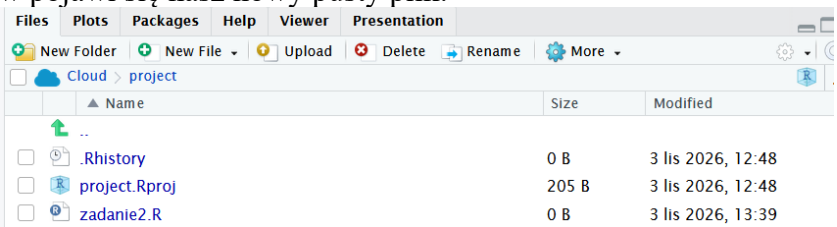
b) Napisz w konsoli: **print(a)** i naciśnij Enter. Dopiero teraz w konsoli wypisana zostanie wartość 8 ([1] – informuje, że wyświetlana wartość jest pierwszym elementem wektora).
c) Wpisz w konsole: **b=2*a** i zobacz rezultat.
d) Wpisz komendę, która pokaże jakie zmienne mamy w środowisku: **ls()**
e) Wpisz komendę, która usunie zmienną a: **rm(a)**
f) Usuń wszystkie zmienne. Najprościej wykonać to klikając ikonkę miotełki w okienku **Environment**  .

Zadanie 2

Na ogół wykonując obliczenia w języku R będziemy potrzebować skryptu, gdzie wypiszemy listę komend, które język R ma wykonać. Utworzymy więc skrypt R – plik z rozszerzeniem R. Kliknij ikonkę „+" (**New file**) i wybierz **R Script**. Miejsce ikony pokazano na rysunku poniżej:



Następnie kliknij dyskietkę aby zapisać nowy plik i nadaj mu nazwę, np. zadanie2.
Na liście plików pojawi się nasz nowy pusty plik:

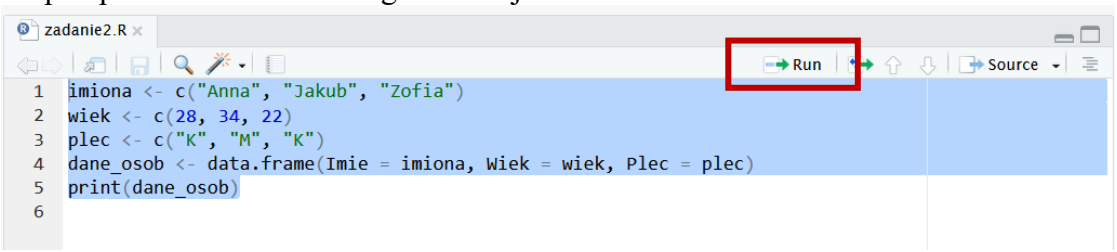


Wpisując podane niżej polecenia uzupełnij ten plik aby wykonać pięć zadań, które pomogą Ci zrozumieć i nauczyć się pracować z obiektami **data.frame**. Zadania te obejmują różne operacje, które można wykonać na ramkach danych.

a) Utwórz data.frame zawierający dane o trzech osobach, ich wieku i płci. W tym celu przepisz kod:

```
# Tworzenie data.frame
imiona <- c("Anna", "Jakub", "Zofia")
wiek <- c(28, 34, 22)
plec <- c("K", "M", "K")
dane_osob <- data.frame(Imie = imiona, Wiek = wiek,
Plec = plec)
print(dane_osob)
```

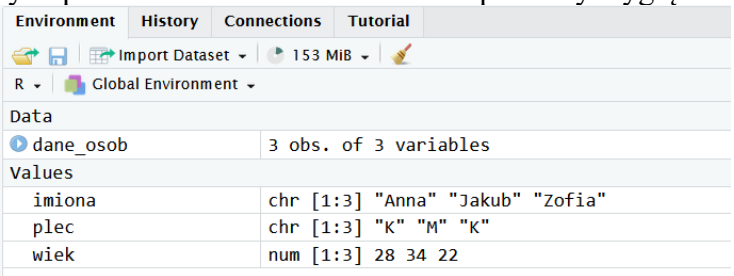
Po przepisaniu kodu zaznacz go i kliknij **Run**



Przycisk **Run** uruchamia aktualną linię lub zaznaczony fragment kodu. Poza tą możliwością przydatny jest również przycisk **Source**, który uruchamia cały skrypt od początku do końca. Po rozwinięciu listy koło przycisku Source pojawia się też opcja **Source with Echo**, która uruchamia cały skrypt i dodatkowo wyświetla wykonywane polecenia w konsoli. Do uruchamiania kodu wybieraj opcję, która bardziej Ci odpowiada. Pamiętaj, że czasem nie trzeba uruchamiać kodu od początku bo na przykład tylko dopisujesz kilka linijek więc starczy zaznaczenie fragmentu i **Run**.

Uwaga 1: w języku R ciąg znaków poprzedzony znakiem # to komentarz kodu, który nie jest interpretowany przez kompilator a pozwala na robienie sobie komentarzy tłumaczących fragmenty kodu.
Uwaga 2: Nie używaj polskich literek w nazwach zmiennych bo kompilator ich nie rozumie. Możesz do tego celu używać języka angielskiego.
Uwaga 3: c(…) pochodzi od angielskiego słowa combine (łączyć) i służy do tworzenia wektorów poprzez połączenie kilku wartości w jeden obiekt.
Uwaga 4: symboliczna strzałka <- wprowadza dane do zmiennej o nazwie podanej przed strzałką.

Po wykonaniu tych poleceń zmienne środowiskowe powinny wyglądać tak:



Kliknij dane_osób. Otworzy się tabela w zakładce obok nazwy Twojego skryptu. Przyjrzyj się sposobowi w jaki program prezentuje dane i wróć do zakładki ze skryptem.

b) Dopisz w skrypcie kolejne polecenia dotyczące nauki indeksowania i wybierania danych z data.frame: wyświetl wiek osoby o imieniu "Jakub" i przypisz go do nowej zmiennej wiek_jakub.

```
# Wyświetlanie wieku Jakuba
wiek_jakub <- dane_osob$Wiek[dane_osob$Imie == "Jakub"]
print(wiek_jakub)
```

Uwaga 5: Możesz użyć funkcję paste(), żeby elegancko opisać wyświetlony wiek: Zamiast print(wiek_jakub) użyj:

```
print(paste("Jakub ma" , wiek_jakub , "lat"))
```

c) Nauka dodawania nowych kolumn do data.frame: dodaj kolumnę, która oblicza, ile lat zostało do 40-tych urodzin każdej osoby:

```
# Dodawanie nowej kolumny
dane_osob$Lata_do_40 <- 40 - dane_osob$Wiek
print(dane_osob)
```

d) Nauka filtrowania danych w data.frame: wyświetl dane tylko dla osób, które mają mniej niż 30 lat.

```
# Filtrowanie danych
mlodsze_osoby <- dane_osob[dane_osob$Wiek < 30, ]
print(mlodsze_osoby)
```

Uwaga 6: Co oznacza przecinek powyżej? To sposób filtrowania ramek danych w R. Przecinek oddziela wybór wierszy od wyboru kolumn. W indeksowaniu ramek danych w R, wszystko przed przecinkiem odnosi się do wierszy, a wszystko po przecinku — do kolumn. Pierwsza część przed przecinkiem (dane_osob$Wiek < 30) definiuje, które **wiersze** mają zostać wybrane. Pusta wartość po przecinku oznacza, że zostaną wybrane wszystkie **kolumny** (nic nie filtrujemy). Aby sprawdzić jak odfiltrować dodatkowo kolumny (np. żeby pokazał tylko 1 i 2 kolumnę) podmień powyższą linię na:
mlodsze_osoby <- dane_osob[dane_osob$Wiek < 30, **1:2**]

e) Nauka agregacji danych w data.frame: utwórz nowy data.frame, który pokazuje średni wiek osób dla każdej płci:

```
# Obliczanie średniego wieku według płci
sredni_wiek <- aggregate(Wiek ~ Plec, data = dane_osob,
FUN = mean)
print(sredni_wiek)
```

2. Zabawy z językiem R – wykresy

Wykonaj dwa ćwiczenia, które pomogą Ci nauczyć się rysowania wykresów w R za pomocą biblioteki ggplot2. Oba ćwiczenia skupiają się na różnych typach wykresów i podstawowych estetykach, które można wykorzystać w ggplot2.

a) Nauka rysowania wykresu punktowego z użyciem ggplot2: utwórz wykres punktowy przedstawiający związek między dwoma zmiennymi: wiekiem a wzrostem osób.

```
library(ggplot2)
# Tworzenie przykładowych danych
wiek <- c(20, 25, 30, 35, 40, 45, 50, 55, 60, 65)
wzrost <- c(170, 175, 180, 178, 165, 162, 180, 182, 175,
172)
dane <- data.frame(Wiek = wiek, Wzrost = wzrost)
```

```
#Tworzenie wykres punktowego.
wykres_punktowy <- ggplot(dane, aes(x = Wiek, y =
Wzrost)) + geom_point(color = "blue", size = 3) +
  labs(title = "Zwiazek miedzy wiekiem a wzrostem",
       x = "Wiek (lata)", y = "Wzrost (cm)") +
  theme_minimal()

# Wyświetlenie wykresu
print(wykres_punktowy)
```

b) Nauka rysowania wykresu słupkowego: utwórz wykres słupkowy przedstawiający liczbę osób w różnych grupach wiekowych:

```
#Tworzenie przykładowych danych
grupa_wiekowa <- c("18-25", "26-35", "36-45", "46-55",
"56-65")
liczba_osob <- c(15, 30, 25, 20, 10)
dane_grupowe <- data.frame(Grupa_wiekowa =
grupa_wiekowa, Liczba_osob = liczba_osob)
print(dane_grupowe)

# Tworzenie wykresu słupkowego
wykres_slupkowy <- ggplot(dane_grupowe, aes(x =
Grupa_wiekowa, y = Liczba_osob, fill = Grupa_wiekowa))
+
  geom_bar(stat = "identity") +
  labs(title = "Liczba osob w grupach wiekowych",
       x = "Grupa wiekowa",
       y = "Liczba osob") +
  theme_minimal() +
  scale_fill_brewer(palette = "Pastel1") # Użycie
palety kolorów

# Wyświetlenie wykresu
print(wykres_slupkowy)
```

c) Wyeksportuj oba wykresy (zakładka Plots->Export) jako obraz PNG i zapisz w projekcie.

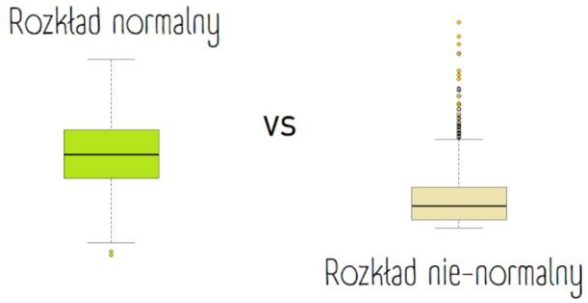
Uwaga 1: geom_bar(stat = "identity") ustawiamy, aby odnieść się do zmiennej na osi Y jako wartości liczbowej.
Uwaga 2: Kolory słupków można różnicować w zależności od poziomu współczynnika zmiennej osi X dlatego zmienna fill ustawiona jest tak: fill = Grupa_wiekowa. Każdy słupek będzie innego koloru. Zmień parametr fill na fill=„coral” i zobacz różnicę.

3. Testowanie statystyczne danych z BMI

Do poniższego zadania wykorzystaj obliczone BMI rozsegregowane według płci (K i M) z ćwiczenia 6 (plik źródłowy BMI_cw3.xlsx). Sprawdź, czy rozkłady BMI dla różnych płci są istotne statystycznie.

a) Wykorzystanie **Excela**:

i. Na podstawie obliczonych wcześniej statystyk opisowych dla BMI kobiet i mężczyzn wywnioskuj, czy te rozkłady danych są normalne (lub zbliżone do normalnych). Możesz to zrobić wizualnie na podstawie histogramu lub wykresu pudełkowego:

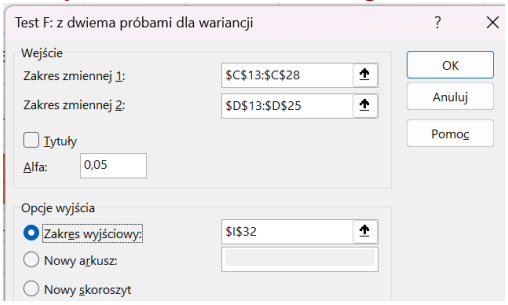


Pudełko zmiennej z rozkładu normalnego jest symetryczne. Może mieć outliery, ale nie może ich być zbyt dużo.

(Źródło: https://statystykawpsychologii.blogspot.com/2014/08/normalnie-o-normalnym-rozkadzie.html)

Dodatkowo sprawdź parametry ze statystyki opisowej:
właściwości rozkładu normalnego to jednomodalność, symetryczność (brak skośności), odpowiednia kurtoza. Sprawdź więc skośność i kurtozę: rozkład normalny ma zerową skośność i kurtozę jednakże przez losowość danych nasza skośność i kurtoza nie muszą być 0 aby uznać dane za pasujące do rozkładu normalnego. Jedną z popularniejszych reguł kciuka odnośnie skośności jest ta, która mówi, że skośność w próbie, które znajduje się między -1 a 1 to jest skośność akceptowalna. Dla kurtozy ten przedział wynosi między -2 a 2.
Po przeprowadzeniu tych obserwacji odpowiedz na pytanie: czy oba rozkłady BMI dla kobiet i mężczyzn są normalne?
Lepszy do tego celu byłby test Shapiro – Wilka lub Kołmogorowa-Smirnova ale nie ma ich standardowo w Excelu (są w Xrealstats).

ii. Sprawdź czy wariancje są homogeniczne. Możesz wykorzystać do tego „Test F z dwiema próbami dla wariancji" zawarty w dodatku Analysis ToolPak.
Kliknij Dane->Analiza danych->Test F: z dwiema próbami dla wariancji



gdzie: zakres zmiennej 1 ustaw serie danych dla kobiet, zakres zmiennej 2 ustaw zakres danych dla mężczyzn, zakres wyjściowy ustaw jedną komórkę gdzie wstawiony ma zostać wynik analizy.
Zinterpretuj wynik: Jeśli wartość statystyki F (pole: test F jednostronny) jest mniejsza niż krytyczna wartość F (pole F), oznacza to, że wariancje są

homogeniczne. Możesz też zwrócić uwagę na wartość p. Jeśli jest < 0.05 to odrzucamy hipotezę zerową o homogeniczności wariancji.
Do weryfikacji czy wariancja w badanych próbach jest równa lepszy jest test Levene'a – nie ma go standardowo w Excelu.

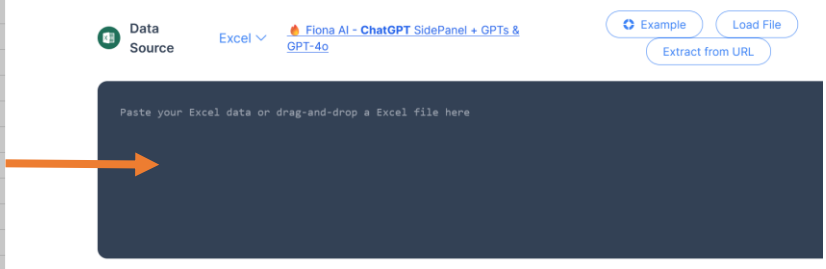
iii. Wykonaj test t-Studenta klikając Dane->Analiza danych-> Test t: z dwiema próbami zakładający równe wariancje (zakładając, że otrzymane w poprzednim podpunkcie wariancje są homogeniczne). Spójrz na otrzymane wartości p-value (P(T<=t) jednostronny oraz dwustronny). Czy analiza w takim przypadku jest prosta?
Uwaga 1:
Test jednostronny zakłada, że różnica jest istotna tylko w jednym określonym kierunku (np. średnia jednej grupy jest większa niż drugiej lub odwrotnie). Tutaj, *p*-value < 0,05 sugeruje, że istnieje statystycznie istotny dowód na to, że średnia jednej grupy jest większa od drugiej w założonym kierunku.
Test dwustronny sprawdza, czy istnieje jakakolwiek różnica między grupami, niezależnie od kierunku. Jeśli *p*-value jest większe niż 0,05 w teście dwustronnym, sugeruje to brak istotnej różnicy ogólnej między grupami na poziomie istotności 0,05.
Wnioski:

- Należy ostrożnie interpretować wynik jednostronny i stosować go tylko wtedy, gdy istnieją silne przesłanki teoretyczne lub praktyczne, uzasadniające analizę różnicy w konkretnym kierunku.
- W większości przypadków naukowych zaleca się stosowanie testów dwustronnych, ponieważ są one bardziej neutralne i pozwalają wykryć różnice w obu kierunkach.

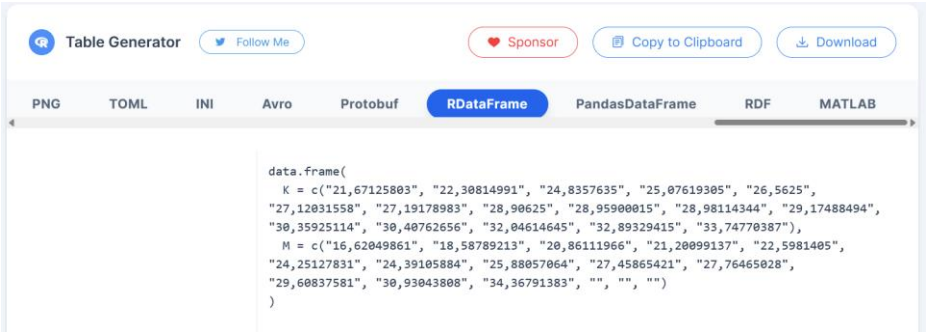
b) Wykorzystanie **języka R**:

i. Przygotuj dane z BMI kobiet i mężczyzn do użycia w języku R. Wykorzystaj do tego **stronę z AI** lub stronę https://tableconvert.com/excel-to-rdataframe gdzie za pomocą opcji Kopiuj/Wklej przekopiuj w czarne pole tabelkę z BMI kobiet i mężczyzn z Excela:

K	M
21,67125803	16,62049861
22,30814991	18,58789213
24,8357635	20,86111966
25,07619305	21,20099137
26,5625	22,5981405
27,12031558	24,25127831
27,19178983	24,39105884
28,90625	25,88057064
28,95900015	27,45865421
28,98114344	27,76465028
29,17488494	29,60837581
30,35925114	30,93043808
30,40762656	34,36791383
32,04614645	
32,89329415	
33,74770387	



Po wklejeniu poniżej wygenerowany zostanie kod w języku R:



Utwórz nowy skrypt (np. Zadanie3.R). Przekopiuj do niego wygenerowane wektory K i M (bez data.frame). Skasuj puste pola "" na końcu wektora M oraz skasuj przecinek na końcu linii wektora K (jeżeli takie są). Czyli chcemy tylko dane w takiej postaci:

K = c("21,67125803", "22,30814991", "24,8357635", "25,07619305", "26,5625", "27,12031558", "27,19178983", "28,90625", "28,95900015", "28,98114344", "29,17488494", "30,35925114", "30,40762656", "32,04614645", "32,89329415", "33,74770387")

M = c("16,62049861", "18,58789213", "20,86111966", "21,20099137", "22,5981405", "24,25127831", "24,39105884", "25,88057064", "27,45865421", "27,76465028", "29,60837581", "30,93043808", "34,36791383")

Jeśli użyto tableconverter to zauważ, że wszystkie liczby są w cudzysłowiu a więc traktowane są jako tekst. Dodatkowo w Polsce do oddzielenia cyfr dziesiętnych stosuje się przecinek a nie kropkę więc po kopiowaniu z polskiego Excela liczby oddzielone są przecinkami. Język R potrzebuje kropki. Pozbędziemy się tego problemu używając:

```
K<- as.numeric(gsub(",", ".", K))
M<- as.numeric(gsub(",", ".", M))
```

Tym sposobem przecinki zostaną zamienione na kropki a następnie całość zostanie rzutowana na zmienną numeryczną.

Jeśli użyto AI to nie powinno być powyższego problemu, ale to sprawdź. Docelowo dane mają mieć format: K = c(21.67125803, 22.30814991,…

Następnie z powyższych wektorów K i M stwórz ramkę danych odpowiedniego formatu do testu Levene'a:

```
dane <- data.frame(
  BMI = c(K, M),
  plec = factor(rep(c("kobiety", "mezczyzni"), times =
c(length(K), length(M))))
)
print(dane)
```

ii. Sprawdź normalność rozkładu testem Shapiro – Wilka:

```
# Test Shapiro-Wilka
shapiro.test(K)
shapiro.test(M)
```

iii. Sprawdź homogeniczność wariancji:

Do tego celu potrzebujemy doinstalować (tylko za pierwszym użyciem) bibliotekę **car.** Przejdź do zakładki **packages** i kliknij przycisk **Install** gdzie wpiszesz szukaną nazwę biblioteki **car**. Pojawi się ona na liście:



#Test homogenicznosci wariancji Levene

```
library(car)
wynik_levene <- leveneTest(BMI ~ plec, data = dane)
print(wynik_levene)
```

iv. W zależności od wyniku testu Levene'a wykonaj odpowiedni test t-Studenta:

- Jeśli p-value w teście Levene >= 0.05 to:

  # Równość wariancji - klasyczny test t-Studenta

  ```
  wynik_t <- t.test(K, M, var.equal = TRUE)
  ```

- Jeśli p-value w teście Levene < 0.05 to:

  # Różne wariancje - test t-Studenta z korekcją Welcha

  ```
  wynik_t <- t.test(K, M, var.equal = FALSE)
  ```

Pozostaje jeszcze wypisać wynik na ekran używając:

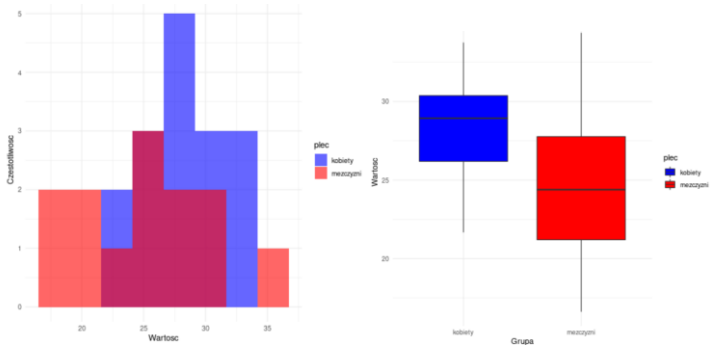
```
print(wynik_t)
```

Uwaga: linia odpowiedzi funkcji t.test: „alternative hypothesis: true difference in means is not equal to 0" znaczy, że test został przeprowadzony w wersji dwustronnej, czyli testował hipotezę alternatywną mówiącą, że średnie obu porównywanych grup różnią się od siebie.

v. Narysuj histogram K i M oraz wykres pudełkowy:

```
library(ggplot2)
#Rysowanie histogramu
ggplot(dane, aes(x = BMI, fill = plec)) +
  geom_histogram(position = "identity", alpha = 0.6,
bins = 8) +
  labs(title = "Histogram dla grup K i M", x = "Wartosc",
y = "Czestotliwosc") +
  scale_fill_manual(values = c("kobiety" = "blue",
"mezczyzni" = "red")) + theme_minimal()
  # Rysowanie wykresu pudełkowego
  ggplot(dane, aes(x = plec, y = BMI, fill = plec)) +
    geom_boxplot() +
    labs(title = "Wykres pudelkowy dla grup K i M", x =
"Grupa", y = "Wartosc")+
    scale_fill_manual(values = c("kobiety" = "blue",
"mezczyzni" = "red")) + theme_minimal()
```

Poprawne wykresy:



Jeżeli potrzebne są wykresy z większą czcionką można dodać na samym końcu za theme_minimal() w tej samej linii:

```
+  theme( plot.title = element_text(size = 20, face
= "bold"), axis.title = element_text(size = 16),
    axis.text = element_text(size = 14))
```

Wyeksportuj wykresy do plików png.

vi. Porównaj wyniki i wykresy uzyskane w Excelu oraz języku R. Czy są takie same?

vii. Sprawdź moc testu i odpowiedz na pytanie: Czy moc testu jest wystarczająca?

```
library(effsize)
coh <- cohen.d(K, M)
print(coh)
library(pwr)
# Ustalanie parametrów
wielkosc_efektu <- coh$estimate # Wielkość efektu
alpha <- 0.05            # Poziom istotności
n1 <- length(K)        # Liczebność grupy 1
n2 <- length(M)       # Liczebność grupy 2
# Obliczenie mocy testu t-Studenta z różnymi wielkościami
próbek
moc <- pwr.t2n.test(n1 = n1, n2 = n2, d = wielkosc_efektu,
sig.level = alpha)
print(moc)
```

Uwaga 3: cohen.d(K, M) oblicza wielkość efektu między danymi K i M potrzebną do wykonania testu mocy. Wielkość efektu d Cohena to miara, która opisuje siłę różnicy między średnimi dwóch grup. Jest jednym z najczęściej stosowanych wskaźników wielkości efektu w badaniach z dwiema różnymi grupami, gdy interesuje nas bezpośrednie porównanie między grupami

viii. Policz i odpowiedz na pytanie ile potrzeba by było danych (pacjentów) dla tego przypadku, aby moc była wystarczająca (80%).

```
wynik_liczebnosci <- pwr.t.test(
  d = wielkosc_efektu,
  sig.level = 0.05,
  power = 0.80,
  type = "two.sample",
  alternative = "two.sided")
print(wynik_liczebnosci)
```

4. Wykorzystanie testu chi-kwadrat

Studium przypadku: Związek między paleniem papierosów a występowaniem chorób układu oddechowego

Celem tego zadania jest zbadanie, czy istnieje związek między paleniem papierosów a występowaniem chorób układu oddechowego w grupie pacjentów. W badaniu weźmie udział 200 pacjentów, z czego część z nich pali papierosy, a część nie. Na podstawie następujących danych jakościowych przeprowadź test chi-kwadrat:

Stan zdrowia	Palący	Niepalący
Zdrowy	50	90
Astma	30	10
Przewlekła obturacyjna choroba płuc (POChP)	20	5
Inne choroby układu oddechowego	10	5
Suma	110	110

a) Stwórz nowy plik Zadanie4.R i zapisz powyższe dane w formacie tabeli, gdzie wiersze będą reprezentować stan zdrowia, a kolumny będą reprezentować pacjentów palących i niepalących:

```
#Przygotowanie danych
stan_zdrowia <- c("Zdrowy", "Astma", "POChP", "Inne")
palacy <- c(50, 30, 20, 10)
niepalacy <- c(90, 10, 5, 5)
# Utworzenie ramki danych do wizualizacji
tabela <- data.frame(
  Stan_zdrowia = stan_zdrowia,
  Palacy = palacy,
  Niepalacy = niepalacy)
print(tabela)
```

b) Użyj tabeli kontyngencji do zorganizowania danych, co pozwoli na łatwiejsze obliczenie testu chi-kwadrat.

```
# Tabela kontyngencji
tabela_kontyngencji <- matrix(c(palacy, niepalacy), nrow =
4, byrow = TRUE)
rownames(tabela_kontyngencji) <- stan_zdrowia
colnames(tabela_kontyngencji) <- c("Palacy", "Niepalacy")
print(tabela_kontyngencji)
```

c) Wykonaj test chi-kwadrat dla niezależności, aby sprawdzić, czy istnieje związek między paleniem papierosów a występowaniem chorób układu oddechowego.

```
  # Test chi-kwadrat
  test_chi_kwadrat <- chisq.test(tabela_kontyngencji)
  print(test_chi_kwadrat)
```

d) Zinterpretuj wyniki testu, zwracając uwagę na wartość p oraz statystykę chi-kwadrat. Czy istnieje istotna różnica w występowaniu chorób układu oddechowego między palaczami a niepalaczami?

Uwaga 1: Statystyka chi-kwadrat: wartość, która wskazuje, jak bardzo obserwowane wartości różnią się od wartości oczekiwanych. Jeśli p-value jest mniejsze niż 0.05,

odrzucamy hipotezę zerową, co sugeruje, że palenie papierosów ma wpływ na występowanie chorób układu oddechowego.

Uwaga 2: Dla naszego przykładu pojawia się ostrzeżenie: *Chi-squared approximation may be incorrec*t. Jest to spowodowane małą liczbą danych. Dla takich mało licznych danych można zastosować test Fishera. Dodaj linię kodu:

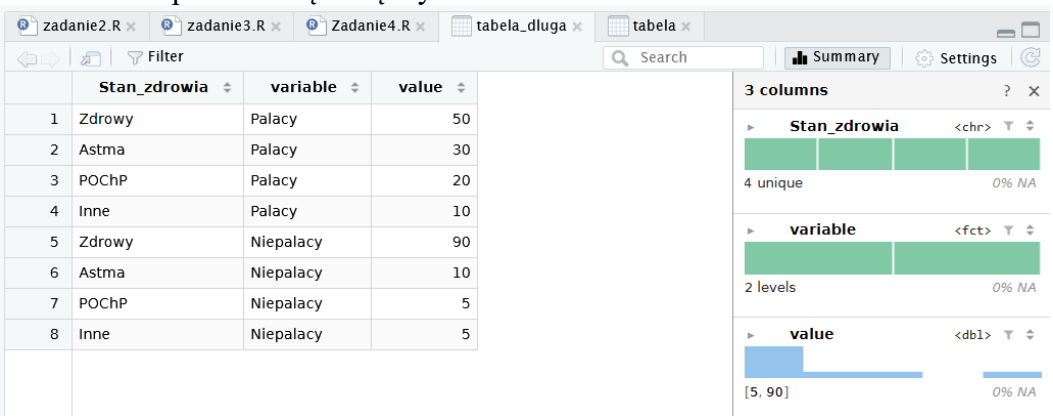
```
fisher.test(tabela_kontyngencji)
```

Zinterpretuj wynik podobnie jak powyżej. Jest jakaś różnica?

e) Narysuj wykres słupkowy, który przedstawia częstość występowania różnych stanów zdrowia w grupach palących i niepalących (wyeksportuj go później do pliku).

```
library(ggplot2)
# Przekształcenie danych do formatu długiego
tabela_dluga <- reshape2::melt(tabela)
print(tabela_dluga)
# Wykres słupkowy
ggplot(tabela_dluga, aes(x = Stan_zdrowia, y = value, fill
= variable)) +
  geom_bar(stat = "identity", position = "dodge") +
  labs(title = "Czestosc wystepowania chorob u palaczy i
niepalacych",
       x = "Stan zdrowia",
       y = "Liczba pacjentow")  +  theme_minimal()
```

Porównaj jak wygląda zmienna **tabela** oraz **tabela_dluga** klikając każdą z nich w **Environment** i przechodząc między zakładkami.



	Stan_zdrowia	variable	value
1	Zdrowy	Palacy	50
2	Astma	Palacy	30
3	POChP	Palacy	20
4	Inne	Palacy	10
5	Zdrowy	Niepalacy	90
6	Astma	Niepalacy	10
7	POChP	Niepalacy	5
8	Inne	Niepalacy	5

Zakończenie: Po wykonaniu wszystkich 4 zadań zaznacz pliki R i png (zaznacz odpowiedni checkbox) w zakładce Files i pobierz je na dysk: Files->More->Export.