Wprowadzenie do psychometrii z językiem R

Paweł Smoliński Uniwersytet Merito

1 Wczytywanie danych do R

Do języka R możemy jak do każdego innego programu wczytać nasze własne dane. Aby to zrobić należy najpierw zapisać nasze dane w formacie CSV (wartości oddzielone przecinkami) i wczytać używając poniższej komendy:

```
# Aby wczytac plik CSV (wartosci oddzielone przecinkami) do
    R:
data <- read.csv("sciezka_do_twojego_pliku.csv")

# Aby wyswietlic pierwsze wiersze danych:
head(data)
```

1.1 Podstawowe operacje w R

1.1.1 Tworzenie zmiennej

W R łatwo możesz przypisać wartość do zmiennej za pomocą operatora przypisania <-.

```
my_variable <- 10
print(my_variable)  # Komenda print() wyswietli wartosc 10
```

1.1.2 Wektory w R

Wektor to zbiór elementów tego samego typu. W R łatwo możemy tworzyć różne typy wektorów oraz manipulować ich strukturą. Wektory tworzymy wykorzystując połączenie symboli c(...). Aby wyświetlić dany element wekrtora, przy nazwie dodajemy [...], gdzie wewnątrz wpisujemy numer elementu wektora.

```
# Tworzenie wektora numerycznego
numeric_vector <- c(1, 2, 3, 4, 5)

# Tworzenie wektora znakowego
char_vector <- c("jablko", "banan", "wisnia")

# Dostep do elementow wektora (Pamietaj: indeksowanie w R
    zaczyna sie od 1)
```

```
first_element <- numeric_vector[1]
```

1.1.3 Podstawowe operacje matematyczne

R może też funkcjonować jako prosty kalkulator!

```
5 + 10
10 - 5
5 * 2
10 / 2
5^2
sqrt(25)
```

1.1.4 Operacje na zbiorach danych

Zbiór danych (data frame) to tabela lub struktura podobna do dwuwymiarowej tablicy, w której każda kolumna zawiera wartości jednej zmiennej, a każdy wiersz zawiera zestaw wartości z każdej kolumny (tak jak w excelu!). Zbiór danych tworzymy wykorzystując komendę data.frame(...) i wewnątrz wprowadzając wektory zmiennych, które będą kolumnami naszego zbioru. Nazwy wektorom nadajemy poprzez użycie znaku =.

```
# Tworzenie ramki danych
df <- data.frame(
  Imie = c("Jan", "Jadwiga", "Kowalski"),
  Wiek = c(23, 25, 30),
  Plec = c("Mezczyzna", "Kobieta", "Mezczyzna"))
```

Ważna i przydatna jest umiejętność wynajdywania konkretnych informacji z naszego zbioru danych. Możemy to robić za pomocą następujących funkcji:

```
# Dostep do konkretnej kolumny
df$Wiek

#Dostep do konkretnego wiersza
df[1, ]

#Dostep do wiersza i kolumny
df[1, "Wiek"]

#Wydobywanie wierszy na podstawie warunku
df[df$Plec == "Mezczyzna", ]

#Wydobywanie kilku kolumn
df[, c("Imie", "Plec")]

# Dodawanie nowej kolumny
df$Ocena <- c(85, 90, 88)
```

2 Prosta analiza danych w R

Po wczytaniu danych do R, kolejnym krokiem jest zrozumienie ich struktury oraz dokonanie wstępnych obserwacji. R dostarcza zestaw funkcji do tego celu.

2.1 Badanie struktury danych

Zanim rozpoczniemy nasze analizy warto przyjrzeć się strukturze naszych danych, a szczególnie jakiego typu są nasze zmienne. Aby uzyskać zwięzłe podsumowanie zbioru danych korzystamy z funkcji:

```
str(data)
```

2.2 Wyświetlanie zmiennych (kolumn) w zbiorze danych

W R możemy wyświetlić nasz zbiór danych, korzystając z komendy view(...). Możemy również uzyskać nazwy kolumn w zbiorze danych:

```
view(data)

#Wyswitelnaie nazw zmiennych
names(data)

#Lub alterantywnie
colnames(data)
```

2.3 Zmiana nazw zmiennych

Powiedzmy, że chcecie zmienić nazwy niektórych zmiennych dla większej jasności lub aby przestrzegać konwencji nazewnictwa. Zmiana nazw zmiennych jest dość łatwą procedurą, aczkolwiek wymaga nieco dłuższego kodu. Na początku przywołujemy funkncje **names(...)** aby R wiedział, że chcemy operować na nazwach zmiennych. Wybieramy z tego **wektora nazw** elementy, którym chcemy zmienić nazwę np.: **[c(1, 3)] (element 1 i 3)**, a następnie nadajemy tym elementom nowe nazwy. Jeżeli zmieniamy więcej niż jedną zmienną musimy również wykorzystać wektor:

```
# Zmiana nazwy pojedynczej zmiennej
names(data)[names(data) == "stara_nazwa"] <- "nowa_nazwa"

# Zmiana nazw kilku zmiennych naraz
names(data)[c(1, 3)] <- c("nowa_nazwa_dla_pierwszej_kolumny"
    , "nowa_nazwa_dla_trzeciej_kolumny")
```

Alternatywnie można użyć funkcji **rename(...)** z pakietu **dplyr**, która może być bardziej intuicyjna dla niektórych. Najpierw musimy zainstalować i odpowiedni pakiet i następnie uruchomić go wpisując komendę **library(...)**. Następ-

nie wewnątrz funkcji rename(...), podajemy zbiór danych i nazwy kolumn, które chcemy zmienić:

```
# Upewnij sie, ze pakiet dplyr jest zainstalowany i
    zaladowany
install.packages("dplyr")
library(dplyr)

data <- rename(data, nowa_nazwa_kolumny = stara_nazwa_
    kolumny)
```

2.4 Podsumowanie statystyk

Szybkie podsumowanie każdej zmiennej jest często bardzo pomocne. Proste statystyki możemy otrzymać z wykorzystaniem następujących funkcji:

```
summary(data)

#Lub dla konkretnej zmiennej
summary(data$nazwa_zmiennej)
```

3 Kodowanie i manipulacja pozycjami w R

Przy pracy z danymi z ankiet lub testów często spotykamy się z koniecznością kodowania lub rekodowania pozycji. Ten proces zapewnia, że dane są w odpowiednim formacie do analizy.

3.1 Konwersja pozycji na wartości liczbowe lub kategorialne

Jeśli masz dane w postaci znaków, które powinny być wartościami liczbowymi (np.: skala Likerta):

```
data$nazwa_zmiennej <- as.numeric(data$nazwa_zmiennej)
```

Natomiast jeżeli danych nie można przekształcić w wartości numeryczne lub są ze swojej natury kategorialne, wówczas wykorzystujemy:

```
data$nazwa_zmiennej <- as.factor(data$nazwa_zmiennej)
```

3.2 Rekodowanie zmiennych

Często może się zdarzyć, że wasze pozycje będę pozycjami odwróconymi. Wówczas niezbędne jest ich rekodowanie. Przyjmijmy na przykład, że mamy pozycję w skali Likerta, gdzie: 1 = Zdecydowanie nie zgadzam się, 2 = Nie zgadzam się, 3 = Neutralny, 4 = Zgadzam się, 5 = Zdecydowanie zgadzam się. Ale chcemy

to zakodować w odwrotny sposób, więc 5 staje się 1, 4 staje się 2, itd. Porsty trick aby to osiągnąć jest następujący (liczba z przodu zawsze jest o 1 większa niż liczba możliwych odpowiedzi):

```
data$pozycja_likerta <- 6 - data$pozycja_likerta
```

Dla bardziej skomplikowanego rekodowania pakiet dplyr dostarcza funkcję **recode(...)**, gdzie najpierw określamy item lub wektor itemów a następnie schemat odwracania:

```
library(dplyr)

data$pozycja_likerta <- recode(data$pozycja_likerta,
     '1' = 5, '2' = 4, '3' = 3, '4' = 2, '5' = 1)
```

3.3 Tworzenie nowej zmiennej jako średniej z innych

Często chcemy stowtorzyć nową zmienną, która jest wynikiem testu. Jest kilka sposóbów teoretycznych na obliczenie wyniku, ale najczęściej wynik jest średnią pozostałych itemów w skali. Aby obliczyć wynik dla danej osoby możemy użyć funkcji **rowMeans**. Na przykład, jeśli mamy trzy pozycje (pozycja1, pozycja2, pozycja3) i chcemy obliczyć wynik:

```
data$wynik <- rowMeans(data[, c("pozycja1", "pozycja2",
"pozycja3")], na.rm = TRUE)
```

Upewnijcie się, że ustawiliście **na.rm = TRUE**, aby obliczało średnie wartości wiersza bez brakujących wartości.

4 Prosta analiza statystyczna w R

Dane w psychometrii często wymagają podstawowych analiz statystycznych, takich jak test t czy analiza korelacji.

4.1 Analiza testu t

Test t służy do ustalenia, czy istnieje istotna różnica między średnimi dwóch grup. Załóżmy, że mamy wyniki testu (zmienna: wynik) z dwóch grup, podzielonych według innej zmiennej (np.: zmienna płeć). W celu wykonania testu t korzystamy z funkcji **t.test(...)**, gdzie wewnątrz określamy najpierw zmienną zależną (wynik) i następnie grupę (plec), a następnie zbiór danych.

```
t_test <- t.test(wynik ~ plec, data)
print(t_test)
```

4.2 Prosta korelacja między dwoma pozycjami

Aby obliczyć współczynnik korelacji między dwoma pozycjami, korzystamy z bardzo prostej funkcji wewnątrz której wprowadzamy zmienne, które chcemy skorelować.

```
r <- cor(data$pozycja1, data$pozycja2)
print(korelacja)
```

4.3 Macierz korelacji

Jeśli mamy kilka pozycji i chcemy znaleźć korelację między wszystkimi parami:

```
cor_matrix <- cor(data[, c("pozycja1", "pozycja2",
                           "pozycja3")])
print(cor_matrix)

#Lub dla calego zbioru danych
cor_matrix <- cor(data)
```

4.4 Wizualizacja macierzy korelacji

Istnieje wiele sposobów na wizualizację macierzy korelacji, ale powszechną metodą jest mapa cieplna. Pakiet **corrplot** jest do tego celu doskonały. Najpierw instalujemy i uruchamiamy corrplot, a następnie wewnątrz funkcji **corrplot** podajemy naszą macierz korelacji oraz metodę wizualizacji.

```
# Instalacja i zaladowanie pakietu
install.packages("corrplot")
library(corrplot)

corrplot(cor_matrix, method = "circle")
```

Ten kod wygeneruje wykres, na którym rozmiar i kolor kółek reprezentują odpowiednio siłę i kierunek korelacji pomiędzy zmiennymi w zbiorze danych.

5 Analiza psychometryczna w R

Analiza psychometryczna polega na obliczeniu i ocenie statystyk dla itemów, współczynnika rzetelności oraz struktury czynnikowej. W tej sekcji zapoznacie się z kilkoma podstawowymi analizami.

5.1 Statystyki opisowe dla pozycji

Zanim zagłębimy się w właściwości testu, istotne jest, aby zdobyć ogólną wiedzę na temat wyników poszczególnych pozycji (itemów). Możemy to zrobić wykorzystując pakiet **psych**, który został specjalnie stworzony do analiz psychometrycznych. Bardzo urzyteczną funkcja jest funkcja **describe**, która wydrukuje

podstawowe statystyki dla zmiennych jak średnia czy odchylenie standardowe, ale również bardziej zaawansowane statystyki jak kurtoza rozkładu.

```
# Instalacja i zaladowanie pakietu psych
install.packages("psych")
library(psych)

# Statystyki opisowe dla pozycji
describe(data)
```

5.2 Analiza pozycji binarnych

Jeśli nasz test opiera się na pozycjach binarnych, bardzo przydatnym narzędziem jest przeprowadzenie analizy trudności oraz dyskryminacji. Można to zrealizować przy użyciu pakietu ltm. Należy jednak pamiętać, że dotyczy to wyłącznie pozycji binarnych (np.: TAK/NIE; prawidłowa/błędna odpowiedź). Dla skali Likerta konieczne jest zastosowanie bardziej zaawansowanych modeli, które wykraczają poza zakres naszego kursu.

```
install.packages("ltm")
library(ltm)
# Pozycje musza byc binarne.
# Jesli twoje pozycje nie sa binarne (np. skala Likerta),
    mozna sprobowac je przeksztalcic choc nie zawsze jest to
    niezbedne do przeporwadzenia analiz.

wynik_diff_pozycji <- item.diff(data[, c("pozycja1",
                                "pozycja2", "pozycja3")])
print(wynik_diff_pozycji)

# Otrzymasz:
# Dif: Trudnosc pozycji, ktora jest proporcja uczestnikow,
    ktorzy odpowiedzieli poprawnie.
# Wspolczynnik dyskryminacji (DI): Wspolczynnik korelacji
    punkt-biserialej.
```

5.3 Alfa Cronbacha

Alfa Cronbacha to miara rzetelności. Statystycznie ujmując jest to miara spójności wewnętrznej tzn. jak ściśle powiązany jest zestaw pozycji jako grupa. Alfa Cronbacha możemy obliczyć korzystając z funkcji **alpha** z pakiety **psych**. Ta funkcja wymaga jedynie podania zbioru danych lub jego części zawierającej itemy. Otrzymamy cały zbiór danych zawierający wiele informacji. Wiele z tych informacji jest ważnych i przydatnych i na zajęciach część z nich omówimy, ale najważniejsze informacje, czyli współczynnik rzetelności otrzymamy wpisując:

```
alfa <- alpha(data[, c("pozycja1", "pozycja2", "pozycja3")])
print(alfa$alpha)
```

Z tego samego zbioru danych **alfa** można również uzyskać szczegółowy wynik, który zawiera informacje, jak usunięcie pozycji wpłynęłoby na wartość alfa, co czasami może być przydatne do oceniania trafności itemu.

6 Analiza czynnikowa w R

Analiza czynnikowa pozwala badaczom na identyfikowanie podstawowych struktur (czynników) w zbiorze danych. Jest powszechnie stosowana w psychometrii do zrozumienia struktury konstruktów psychologicznych.

6.1 Eksploracyjna analiza czynnikowa (EFA)

EFA pomaga zidentyfikować liczbę i charakter konstruktów w zestawie danych. Obliczenie EFA jest bardzo proste i wymaga jedynie pakiety **psych** oraz funkcji **fa(...)** od angielskiego factor analysis. W funkcji fa(...) określamy, które pozycje podlegają analizie oraz zakładaną liczbę czynników.

```
# Korzystajac z pakietu psych
library(psych)

# Wykonaj EFA zakladajac 2 czynniki
efa <- fa(data[, c("pozycja1", "pozycja2",
          "pozycja3", "pozycja4")], nfactors = 2)

# Wydrukuj wyniki
print(efa)
```

Określenie optymalnej liczby czynników w analizie czynnikowej bywa wyzwaniem. Idealnie, wybór liczby czynników powinien być oparty na solidnych podstawach teoretycznych, ale w takim przypadku często wybiera się CFA zamiast EFA. Niemniej jednak, gdy podstawy teoretyczne nie dostarczają jednoznacznych wskazówek co do liczby czynników, można zastosować różne metody empiryczne. Jedną z popularnych metod jest wykorzystanie wykresu scree, nazywanego również wykresem osypiska.

```
scree(dane[, c("pozycja1", "pozycja2",
               "pozycja3", "pozycja4")])
```

Wykres scree pomaga zidentyfikować "łokieć", po którym dodawanie kolejnych czynników nie wyjaśnia znacznie więcej wariancji. Czynniki przed tym łokciem są zwykle zachowywane.

6.2 Konfirmacyjna analiza czynnikowa (CFA)

Podczas gdy EFA dotyczy eksploracji, CFA dotyczy potwierdzenia zakładanej struktury konstruktu psychologicznego. Określamy w niej model w oparciu o teorię lub wcześniejsze analizy i testujemy, jak dobrze pasuje on do naszych

danych. Do CFA powszechnie stosowany jest pakiet **lavaan**. Na początku musimy określić model naszej struktury czynnikowej. Robi się to nadając nazwy czynnikom oraz przypisując im itemy z wykorzystaniem odpowiednich symboli. Następnie wykorzystujemy funkcje **cfa** aby obliczyć statystyki dla modelu.

```
# Instalacja i zaladowanie pakietu lavaan
install.packages("lavaan")
library(lavaan)

# Dla demonstracji zakladamy model dwuczynnikowy:
# Czynnik1 obejmuje pozycja1 i pozycja2
# Czynnik2 obejmuje pozycja3 i pozycja4

# Specyfikacja modelu:
model <- '
Czynnik1=~pozycja1+pozycja2
Czynnik2=~pozycja3+pozycja4
'

cfa <- cfa(model, data)

# Wydrukuj podsumowanie
summary(cfa)
```

CFA testuje, jak dobrze określony model pasuje do danych. Podsumowanie zawiera różne wskaźniki dopasowania i oszacowania parametrów, które będziemy omawiać pobieżnie na zajęciach.

7 Korzystanie z narzędzi AI w analizie danych

Technologia sztucznej inteligencji (AI) stała się niezwykle pomocna w wielu dziedzinach nauki, w tym w analizie danych. Narzędzia takie jak ChatGPT lub Google Bard oferują zaawansowane możliwości analizy i odpowiedzi na pytania dotyczące różnych zagadnień.

7.1 Zastosowanie ChatGPT w analizie danych

Możecie skorzystać z ChatGPT, aby uzyskać wsparcie w rozwiązywaniu problemów analizy danych. Kluczem jest sformułowanie pytania w sposób precyzyjny i jasny. Im bardziej konkretnie zapytacie, tym bardziej precyzyjna i użyteczna odpowiedź

Przykład dobrego pytania:

> *Mam zbiór danych, w którym jest 6 zmiennych. Chcę zmienić nazwę dwóch pierwszych i ostatnich zmiennych. Jak mogę to zrobić korzystając z języka R?Chcę skorzystać z pakietu dplyr. Wyjaśnij krok po kroku co mam zrobić i co kod oznacza.*

Korzystając z narzędzi AI, takich jak ChatGPT, możecie znacząco przyspieszyć proces uczenia się i analizy. Ale pamiętajcie! Intuicja i głębokie zrozumienie tematu są niezbędne do skutecznej analizy danych i interpretacji wyników. Dlatego zalecam traktowanie AI jako dodatkowego narzędzia wspierającego proces nauki, a nie jako jedynego źródła informacji. Współpracujcie z technologią, ale ufajcie też swoim zdolnościom i intuicji!