SPRAWOZDANIE

Zajęcia: Zbiory Big Data i Eksploracja Danych

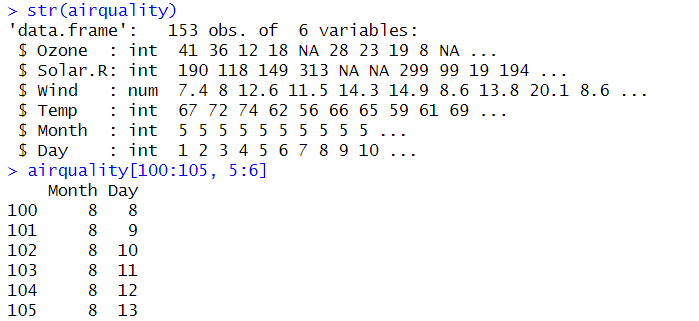
Prowadząca: dr inż. Ruslana Ziubina

|  |  |
| --- | --- |
| Laboratorium nr 2  Data rozpoczęcia: 3.11.2023  Temat: Eksploracja danych – dalsze informacje o języku R | Rafał Klinowski  Informatyka  II stopień, stacjonarne,  Semestr 2, gr. a |

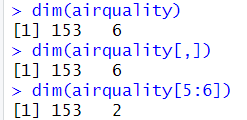
Poszczególne ćwiczenia będą wykonywane w pliku źródłowym edytowanym przy pomocy środowiska RStudio, opisanego w poprzedniej części laboratorium.

# Ćw. 1.

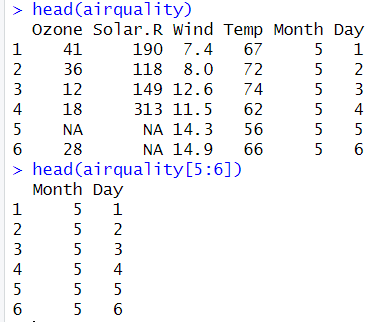
Testowanie przykładowego zbioru „airquality” – wczytanie, uzyskanie podstawowych informacji. Sposoby dostępu do danych w data.frame (na przykład niektórych kolumn czy wierszy), sprawdzenie rozmiaru danych (ile wierszy i kolumn), uzyskanie przykładowych wierszy, wybór warunkowy.



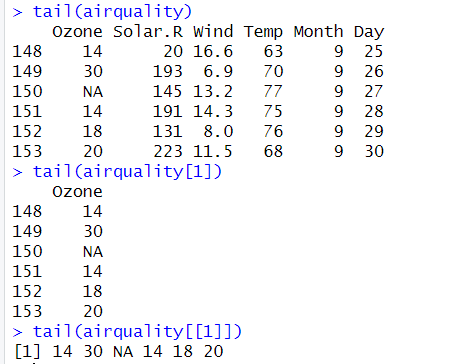
Rysunek 1. Uzyskanie podstawowych informacji o kolumnach zbioru „airquality” oraz wierszy 100-105 z kolumnami 5-6.



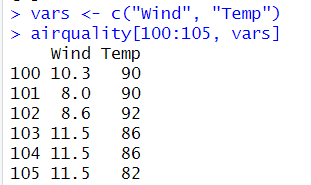
Rysunek 2. Przykłady użycia funkcji dim() do sprawdzenia rozmiaru danych (ilości kolumn i wierszy).



Rysunek 3. Przykłady użycia funkcji head() zwracającej kilka pierwszych wierszy danych.



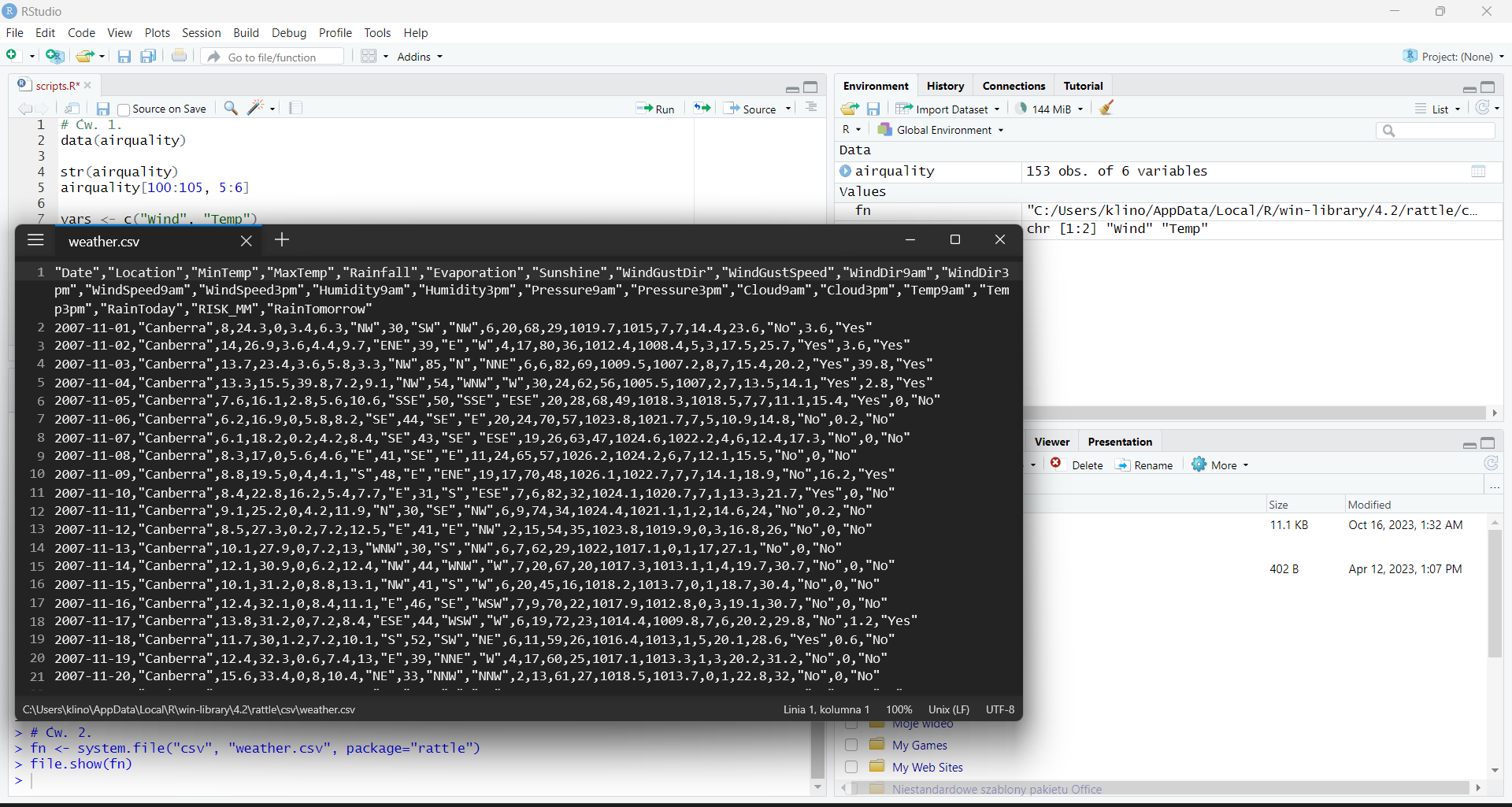
Rysunek 4. Przykłady użycia funkcji tail() zwracającej kilka ostatnich wierszy danych.



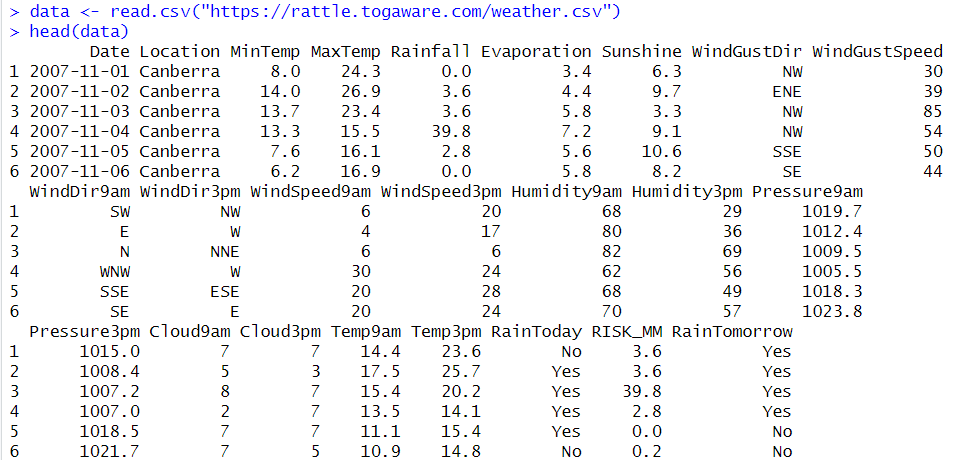
Rysunek 5. Wykorzystanie wektora nazw kolumn do wyboru ich ze zbioru danych.

# Ćw. 2.

Wczytywanie danych z pliku tekstowego (np. pliku wartości oddzielonych przecinkami – CSV) z lokacji na dysku lub w Internecie. Funkcja „read.csv()”.



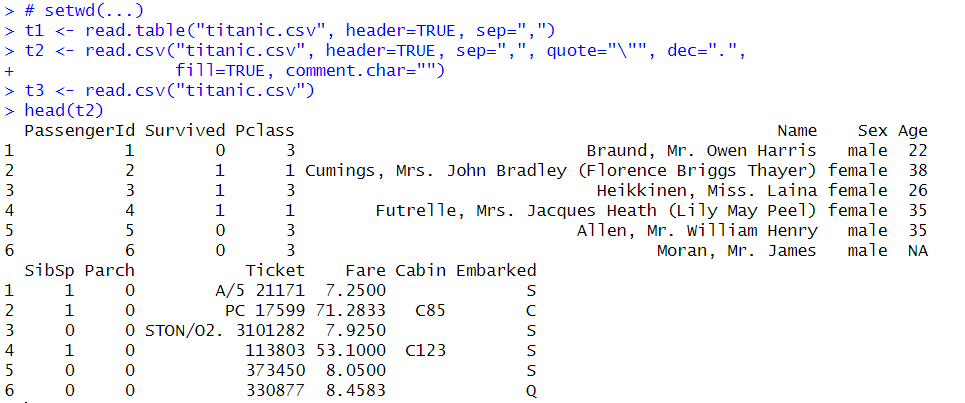
Rysunek 6. Otwarcie i pokazanie pliki tekstowego przy pomocy „file.show()”.



Rysunek 7. Otwarcie pliku tekstowego z lokalizacji w Internecie.

# Ćw. 3.

Wczytywanie danych z pliku tekstowego do ramki danych. Funkcje „read.table()” oraz „read.csv()”. Opcjonalne parametry obu funkcji dotyczące wczytywania danych.



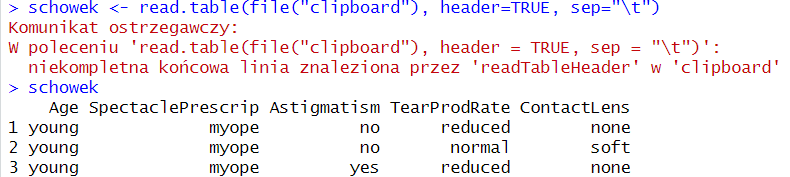
Rysunek 8. Wczytanie danych z pliku CSV na 3 sposoby. Możliwe jest sprecyzowanie wiele parametrów opisujących w jaki sposób dane zostaną wczytane i zinterpretowane. Konieczne jest najpierw ustawienie katalogu roboczego.



Rysunek 9. Przykładowa operacja przeprowadzona na wczytanych danych.

# Ćw. 4.

Wczytywanie danych ze schowka systemu operacyjnego (ang. „clipboard”). Umożliwia ono skopiowanie danych w systemie, a następnie zapisanie ich jako obiekt w R.

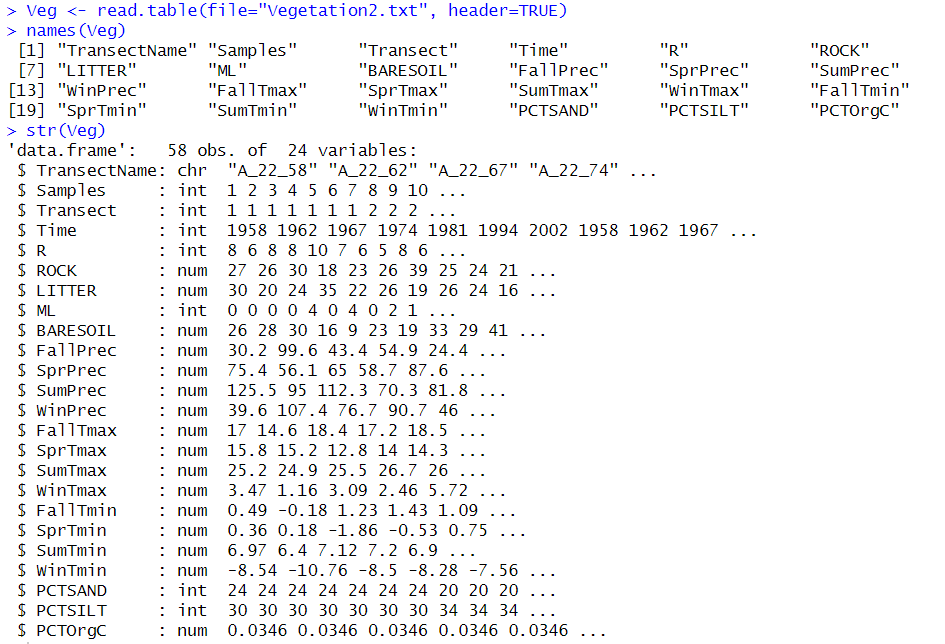


Rysunek 10. Wczytanie danych ze schowka.

Uzyskane ostrzeżenie jest spowodowane niekompletnym znakiem końca linii. Aby się go pozbyć, należy skopiować dane w taki sposób, by nie kopiować znaków końca linii – na przykład kopiując całość pliku.

# Ćw. 5.

Funkcja read.table wykorzystana do wczytania danych z pliku do tabeli. Wyświetlenie podstawowych informacji o wczytanych danych.

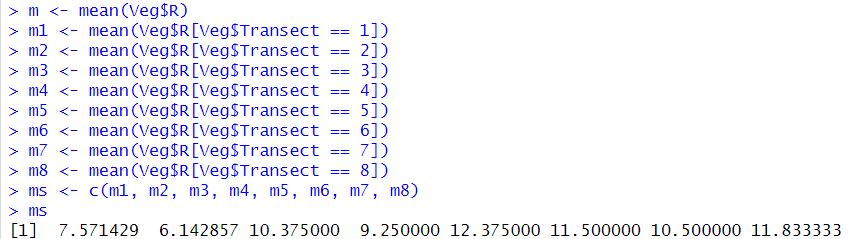


Rysunek 11. Wczytanie zbioru danych „Vegetation2” przy pomocy funkcji read.table() i wypisanie podstawowych informacji o kolumnach.

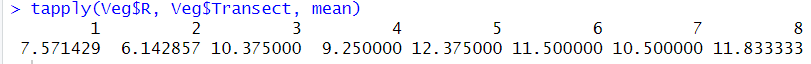
# Ćw. 6.

Przeprowadzanie operacji na zbiorze danych. Skorzystanie z funkcji tapply().

tapply() pozwala na proste zastosowanie danej funkcji na zbiorze danych wybranych ze zbioru głównego – zamiast wykonywać podobną funkcję, zmieniając zaledwie nazwy kolumn czy wierszy, możliwe jest zastosowanie tapply(), aby zrobić to szybciej uzyskując identyczny efekt.



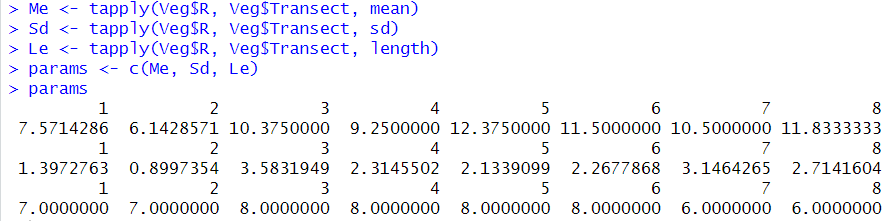
Rysunek 12. Użycie funkcji mean() do policzenia kilku średnich dla różnych wartości danej kolumny.



Rysunek 13. Kod źródłowy realizujący powyższe zadanie w jednej linii z użyciem funkcji tapply().



Rysunek 14. Alternatywny zapis funkcji z nazwami parametrów.

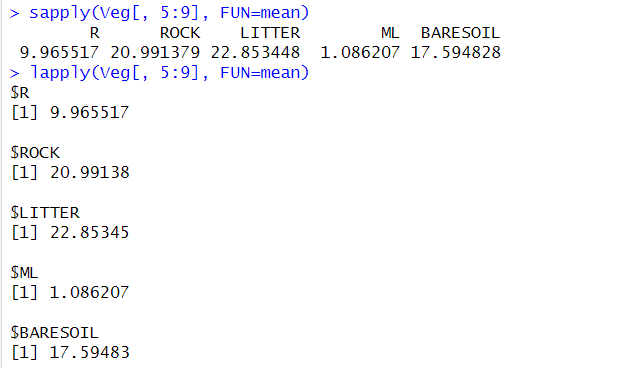


Rysunek 15. Obliczenie kilku parametrów przy pomocy tapply() i wyświetlenie ich.

# Ćw. 7.

Funkcje lapply() i sapply().

Obie funkcje są podobne do funkcji tapply() – pierwsza z nich daje w wyniku listę, druga z nich daje wektor kilku wartości. Różnicą jest również to, że obie funkcje – w przeciwieństwie to tapply() – wykonują podaną funkcję dla jednej lub więcej zmiennych dla wszystkich obserwacji, podczas gdy tapply() wykonuje ją dla podzbiorów obserwacji zmiennej.

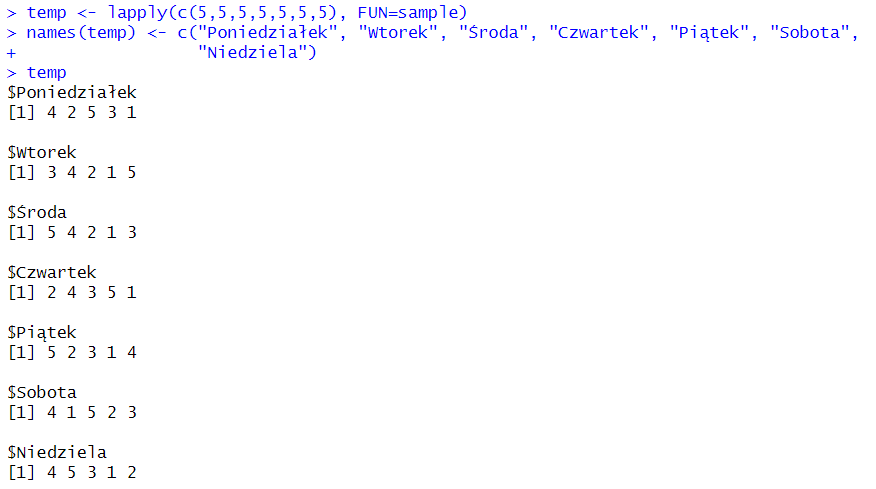


Rysunek 16. Wykorzystanie funkcji sapply() i lapply() oraz porównanie ich wyników.

# Ćw. 8.

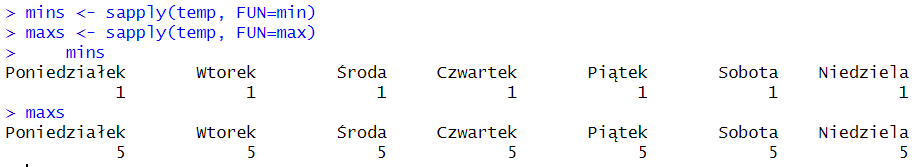
1. Utwórz listę temp…

Do utworzenia listy wykorzystano funkcję lapply() z parametrem „sample” (losowa generacja danych), a następnie dodano do jej wyniku nagłówki.



Rysunek 17. Utworzenie listy przy pomocy funkcji lapply(). Jako parametr X przekazano 7-elementowy wektor zawierający liczby 5 (argumenty dla funkcji sample()).

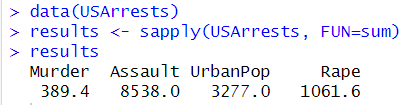
1. Wygeneruj wektor z minimalnymi temperaturami każdego dnia…
2. Wygeneruj wektor z maksymalnymi temperaturami każdego dnia…



Rysunek 18. Wykorzystanie funkcji sapply() do zwrócenia najmniejszych i największych temperatur w każdy z dni.

1. Wczytaj zbiór USArrests.csv...

Do wygenerowania sum wykorzystano funkcję sapply(), która obliczy sumy dla każdej z kolumn (nie podano w funkcji jako parametr ograniczenia). W zbiorze danych występowały cztery kolumny – „Murder”, „Assault”, „UrbanPop” i „Rape”, podczas gdy wiersze reprezentowały poszczególne stany. W wyniku uzyskano sumę poszczególnych kolumn bez rozróżnienia stanów.



Rysunek 19. Wykorzystanie funkcji sapply() z przekazanym argumentem „sum” do obliczenia osobno sum dla każdej z kolumn.

# Wnioski.

Funkcje tapply(), sapply() i lapply(), z którymi zapoznałem się w ramach tego laboratorium, znacznie ułatwiają pracę z danymi i wykonywanie tych samych operacji wielokrotnie na różnych podzbiorach danych. Za ich pomocą można znacząco uprościć wykonywanie operacji takich jak: obliczenie sumy, średniej i wiele więcej, dla wszystkich lub niektórych kolumn zbioru danych, co bez wykorzystania tych funkcji wymagałoby wielokrotnego powtarzania tego samego kodu.

Wczytywanie zbiorów w R jest wyjątkowo proste i intuicyjne. W środowisko wbudowane jest kilka funkcji, których można użyć do wczytania danych z pliku tekstowego, CSV lub ramki danych, zarówno z lokacji na komputerze, jak i w Internecie. W związku z tym praca ze zbiorami danych odbywa się znacznie prościej, niż w innych podobnych środowiskach czy językach.

Całość laboratorium została przeprowadzona w RStudio, które znacznie ułatwia nie tylko tworzenie poleceń (dzięki kolorowaniu składni oraz podpowiadaniu nazw), ale również ich powtórzenie (w przypadku uruchomienia więcej niż raz) oraz podświetlenie zarówno ich wyników, jak i danych, na podstawie których te wyniki zostały uzyskane. Środowisko posiada również podgląd danych w formie tabeli lub jako wartość, co przydaje się w celu podejrzenia ich wyglądu i rozmiaru, jak również do weryfikacji wpisywanych poleceń.