Podstawy języka R - zadania (5)

Tomasz Owczarek, Mateusz Naramski

2024/2025, semestr letni

Zbiór danych imdb.csv

Pod zmienną `imdb` wczytaj dane z pliku *imdb.csv* (użyj funkcji `read.csv2`). W ramce `imdb` pozostaw tylko rekordy bez braków (przypomnij sobie funkcję `complete.cases()`) i pozbądź się pierwszej kolumny. Polecenie `str` powinno po tym zwrócić następujacy wynik:

```
## 'data.frame':    838 obs. of  11 variables:
## $ Title : chr "Guardians of the Galaxy" "Prometheus" "Split" "Sing" ...
## $ Genre : chr "Action,Adventure,Sci-Fi" "Adventure,Mystery,Sci-Fi"
##    "Horror,Thriller" "Animation,Comedy,Family" ...
## $ Description : chr "A group of intergalactic criminals are forced to work
##    together to stop a fanatical warrior from taking control "| __truncated__
##    "Following clues to the origin of mankind, a team finds a structure on a
##    distant moon, but they soon realize the"| __truncated__ "Three girls are
##    kidnapped by a man with a diagnosed 23 distinct personalities. They must try
##    to escape before t"| __truncated__ "In a city of humanoid animals, a
##    hustling theater impresario's attempt to save his theater with a singing
##    compe"| __truncated__ ...
## $ Director : chr "James Gunn" "Ridley Scott" "M. Night Shyamalan" "Christophe
##    Lourdelet" ...
## $ Actors : chr "Chris Pratt, Vin Diesel, Bradley Cooper, Zoe Saldana" "Noomi
##    Rapace, Logan Marshall-Green, Michael Fassbender, Charlize Theron" "James
##    McAvoy, Anya Taylor-Joy, Haley Lu Richardson, Jessica Sula" "Matthew
##    McConaughey,Reese Witherspoon, Seth MacFarlane, Scarlett Johansson" ...
## $ Year : int 2014 2012 2016 2016 2016 2016 2016 2016 2016 2016 ...
## $ Runtime.Minutes. : int 121 124 117 108 123 103 128 141 116 133 ...
## $ Rating : num 8.1 7 7.3 7.2 6.2 6.1 8.3 7.1 7 7.5 ...
## $ Votes : int 757074 485820 157606 60545 393727 56036 258682 7188 192177 232072
##    ...
## $ Revenue.Millions.: num 333 126 138 270 325 ...
## $ Metascore : int 76 65 62 59 40 42 93 78 41 66 ...
```

Funkcja nchar

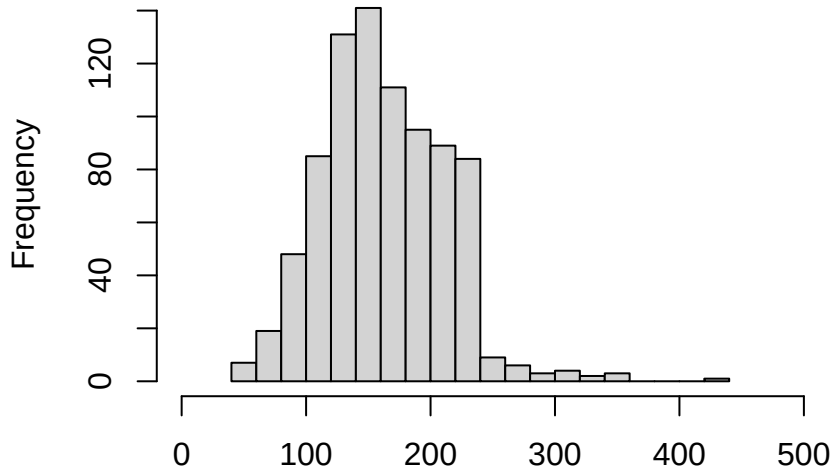
76. Funkcja `nchar` przyjmuje wektor znakowy i zwraca wektor liczbowy, którego wartości informują o liczbie znaków poszczególnych elementów wektora znakowego. Przykład działania:

```
x <- c("a", "aaa", "aa")
nchar(x)
```

```
## [1] 1 3 2
```

a) Sprawdź podstawowe statystyki długości tytułów filmów z ramki `imdb` (skorzystaj z funkcji `summary` i `nchar`)
b) Utwórz histogram długości opisów (*Description*) poszczególnych filmów w ramce `imdb` (ustal liczbę przedziałów na 20 i zakres osi **X** na 0-500)

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##    2.00    9.00   13.00   14.83   18.75   61.00
```



Zaawansowane wyszukiwanie tekstu (funkcje grep, grepl)

77. W funkcjach `grep/l` domyślnie są stosowane wyrażenia regularne (*regular expressions*). Bardzo użyteczne są symbole `^` (początek tekstu), `$` (koniec tekstu) oraz zapis `[0-9]` oznaczający dowolną cyfrę. Dodatkowo opcja `value=TRUE` (tylko w funkcji `grep`) zwraca konkretne wartości, które spełniają podany wzorzec (a nie ich indeksy, jak to jest domyślnie). Przykłady działania:

```
x <- c("tomek", "t0mek", "tom3k", "matys", "maty5")
grep("^t", x, value = TRUE) # poczatek na "t"
```

```
## [1] "tomek" "t0mek" "tom3k"
```

```
grep("ek$", x, value = TRUE) # na koncu "ek"
```

```
## [1] "tomek" "t0mek"
```

```
grep("[0-9]", x, value = TRUE) # ciag zawiera dowolna cyfre w dowolnym miejscu
```

```
## [1] "t0mek" "tom3k" "maty5"
```

a) Wyświetl tytuły filmów z ramki `imdb`, które zaczynają się na dowolną cyfrę
b) Wyświetl tytuły filmów z ramki `imdb`, które kończą się spacją i cyfrą 2 lub większą

```
##  [1] "12 Years a Slave"       "300"                    "10 Cloverfield Lane"
##  [4] "13 Hours"               "20th Century Women"     "21 Jump Street"
```

```
##  [7] "22 Jump Street"         "300: Rise of an Empire" "3 Idiots"
## [10] "2012"                   "2307: Winter's Dream"   "31"
## [13] "1408"                   "127 Hours"              "42"
## [16] "21"                     "28 Weeks Later"         "50/50"
## [19] "17 Again"               "3 Days to Kill"
```

```
##  [1] "Harry Potter and the Deathly Hallows: Part 2"
##  [2] "Furious 6"
##  [3] "Now You See Me 2"
##  [4] "The Conjuring 2"
##  [5] "The Amazing Spider-Man 2"
##  [6] "Spider-Man 3"
##  [7] "The Twilight Saga: Breaking Dawn - Part 2"
##  [8] "The Expendables 3"
##  [9] "Grown Ups 2"
## [10] "The Hunger Games: Mockingjay - Part 2"
## [11] "Pitch Perfect 2"
## [12] "Zoolander 2"
## [13] "Kick-Ass 2"
## [14] "Big Hero 6"
## [15] "Iron Man 2"
## [16] "District 9"
## [17] "Triple 9"
## [18] "Men in Black 3"
## [19] "Super 8"
## [20] "Kung Fu Panda 3"
## [21] "Taken 3"
## [22] "Ted 2"
## [23] "Toy Story 3"
## [24] "Cars 2"
## [25] "Hotel Transylvania 2"
## [26] "Despicable Me 2"
## [27] "Horrible Bosses 2"
## [28] "How to Train Your Dragon 2"
## [29] "The Expendables 2"
## [30] "Sex and the City 2"
## [31] "Scream 4"
## [32] "My Big Fat Greek Wedding 2"
## [33] "Final Destination 5"
```

Funkcja gsub

78. Funkcja `gsub` pozwala na podmianę dowolnego ciągu znaków na inny i zwraca wektor z tymi zmianami. Przykład działania:

```
x <- c("tomek", "t0mek", "tom3k", "matys", "maty5")
gsub("t", "r", x) # wszystkie "t" zamienione na "r
```

```
## [1] "romek" "r0mek" "rom3k" "marys" "mary5"
```

```
gsub("[0-9]", "", x) # wszystkie cyfry zamienione na pusty ciag znakow (czyli usuniete)
```

```
## [1] "tomek" "tmek"  "tomk"  "matys" "maty"
```

a) Wyświetl tytuły filmów, które oryginalnie zawierają symbol “&”, ale zamień ten symbol na wyraz “and”.
b) Wyświetl tytuły, które zawierają podwójną spację, a następnie te same tyuły, ale poprawione.

```
## [1] "Percy Jackson and the Olympians: The Lightning Thief"
## [2] "Fast and Furious"
## [3] "Love and Other Drugs"
## [4] "Pain and Gain"
## [5] "Love and Friendship"

## [1] "Avengers: Age  of Ultron"  "Moonrise  Kingdom"
## [3] "How to Train  Your Dragon" "Begin  Again"

## [1] "Avengers: Age of Ultron"  "Moonrise Kingdom"
## [3] "How to Train Your Dragon" "Begin Again"
```

Funkcja ifelse

79. Funkcja `ifelse` zwraca wektor, którego wartości są uzależnione od innego wektora. Przykłady działania:

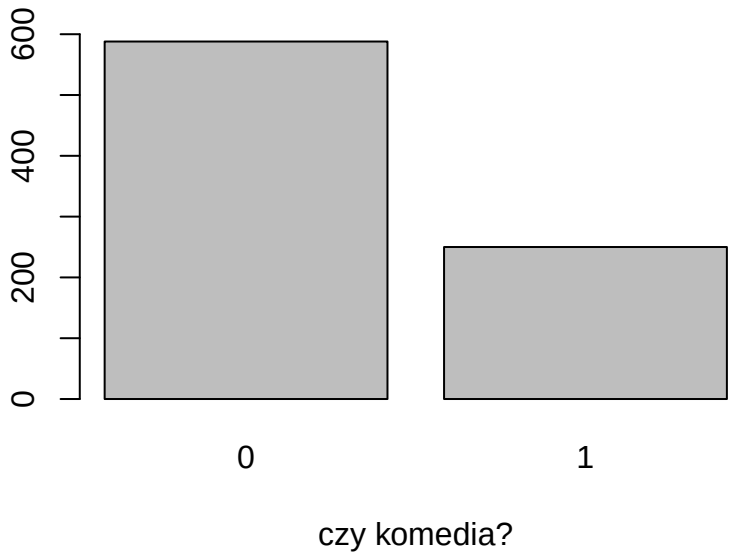
```
y <- 1:5
ifelse(y < 3, "mniejsze niż 3", "3 lub więcej")
```

```
## [1] "mniejsze niż 3" "mniejsze niż 3" "3 lub więcej"   "3 lub więcej"
## [5] "3 lub więcej"
```

```
x <- c("tomek", "t0mek", "tom3k", "matys", "maty5")
ifelse(grepl("[0-9]", x), "zawiera liczbę", "nie zawiera liczby")
```

```
## [1] "nie zawiera liczby" "zawiera liczbę"     "zawiera liczbę"
## [4] "nie zawiera liczby" "zawiera liczbę"
```

a) Dodaj do ramki `imdb` kolumnę o nazwie `dummy_comedy`, która zawiera 1, jeśli w kolumnie `Genre` jest napis “Comedy” i 0 w przeciwnym przypadku.
b) Użyj tej kolumny, żeby przedstawić na wykresie kolumnowym liczbę komedii i nie-komedii (zwiększ zakres osi **Y** i dodaj etykietę osi **X**)



Zadania różne

80. Sprawdź, ile filmów w ramce `imdb` jest filmami akcji (*Action*).

```
## [1] 277
```

81. Pracując na danych z pliku *imdb.csv*:

a) wyświetl tytuły filmów, w których wystapił Ryan Gosling

```
##  [1] "La La Land"                 "The Nice Guys"
##  [3] "The Place Beyond the Pines" "The Big Short"
##  [5] "Crazy, Stupid, Love."       "Drive"
##  [7] "Blue Valentine"             "All Good Things"
##  [9] "Gangster Squad"             "Fracture"
```

b) wyświetl tytuły filmów, w których wystapili Ryan Gosling i Emma Stone (*połącz wektory logiczne uzyskane za pomocą dwóch funkcji grepl*)

```
## [1] "La La Land"           "Crazy, Stupid, Love." "Gangster Squad"
```

c) wyświetl tytuły, oceny widzów (*Rating*) i zarobek (*Revenue.Millions.*) filmów, w których wystapił Adam Sandler

```
##                             Title Rating Revenue.Millions.
## 340                       Blended    6.5             46.28
## 387                        Pixels    5.6             78.75
## 395                   Grown Ups 2    5.4            133.67
## 459               Just Go with It    6.4            103.03
## 538                   The Do-Over    5.7              0.54
## 723                     Grown Ups    6.0            162.00
## 789          Hotel Transylvania 2    6.7            169.69
## 838 You Don't Mess with the Zohan    5.5            100.02
```

82. Dołóż do ramki `imdb` nową kolumnę o nazwie *czy_scifi*, która będzie zawierała wartość "Sci-Fi", jeśli film ma wśród gatunków ciąg znaków Sci-Fi, oraz wartość "nie Sci-Fi" w przeciwnym wypadku. Polecenie `table` na tej kolumnie powinno zwrócić:

```
##
## nie Sci-Fi     Sci-Fi
##        731        107
```

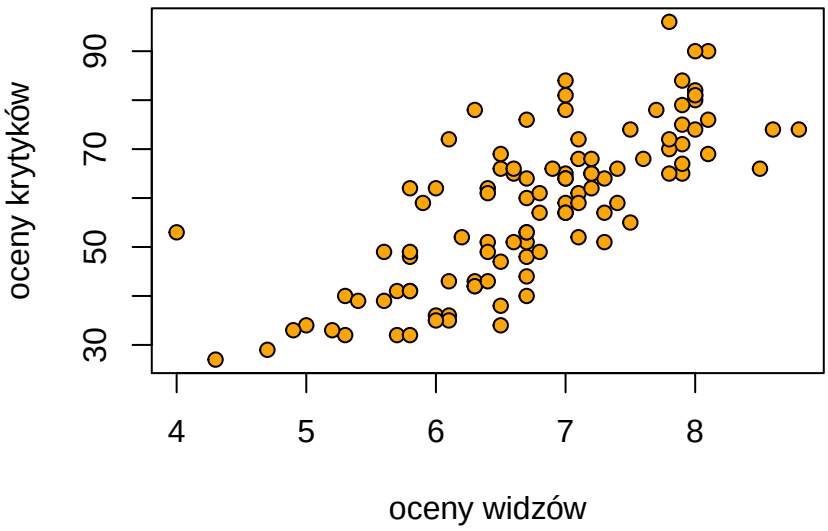
83. Korzystając z kolumny *czy_scifi* utworzonej w poprzednim zadaniu wyświetl:

a) średnią ocenę (*Rating*) filmów Sci-Fi i pozostałych
b) średni zarobek (*Revenue.Millions.*) filmów Sci-Fi i pozostałych

```
## nie Sci-Fi     Sci-Fi
##   6.821067   6.768224
```

```
## nie Sci-Fi     Sci-Fi
##   76.82275  137.45486
```

84. Przedstaw na wykresie punktowym oceny widzów (*Rating*) i oceny krytyków z serwisu Metacritics (*Metascore*) filmów Sci-Fi. Kształt punktów ustaw na koło z wypełnieniem, a kolor wypełnienia ustaw na jakiś ciekawy.



85. Wyświetl tytuły, rok, reżyserów, oceny widzów i oceny z Metacritics filmów Sci-Fi, których oceny od widzów wynoszą co najwyżej 5.

```
##                             Title Year           Director Rating Metascore
## 156 Aliens vs Predator - Requiem 2007      Colin Strause    4.7        29
## 513                The Happening 2008 M. Night Shyamalan    5.0        34
## 553               Fantastic Four 2015         Josh Trank    4.3        27
## 617         2307: Winter's Dream 2016        Joey Curtis    4.0        53
## 949                  After Earth 2013 M. Night Shyamalan    4.9        33
```

Zbiór danych vgs.csv

86. Wczytaj pod zmienną `vgs` dane z pliku *vgs.csv*. Predstawiają one wielkość sprzedaży gier na poszczególne platformy na różnych rynkach (kolumna *na_sales* to sprzedaż w Ameryce Północnej, pozostałe kolumny powinny być oczywiste).

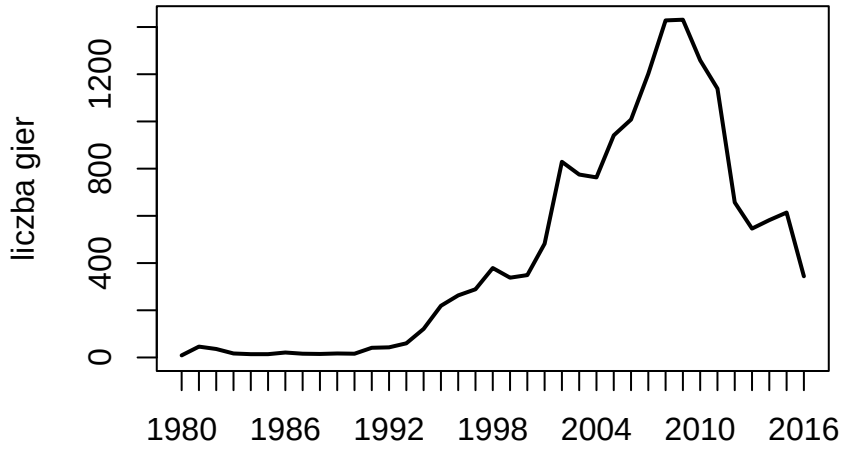
a) sprawdź strukturę ramki `vgs`
b) podsumuj ramkę poleceniem `summary`, sprawdź w której kolumnie występuja braki
c) sprawdź, ile rekordów ma kompletne dane

```
## 'data.frame':    16598 obs. of  11 variables:
## $ rank : int 1 2 3 4 5 6 7 8 9 10 ...
## $ name : chr "Wii Sports" "Super Mario Bros." "Mario Kart Wii" "Wii Sports
##    Resort" ...
## $ platform : chr "Wii" "NES" "Wii" "Wii" ...
## $ year : int 2006 1985 2008 2009 1996 1989 2006 2006 2009 1984 ...
## $ genre : chr "Sports" "Platform" "Racing" "Sports" ...
## $ publisher : chr "Nintendo" "Nintendo" "Nintendo" "Nintendo" ...
## $ na_sales : num 41.5 29.1 15.8 15.8 11.3 ...
## $ eu_sales : num 29.02 3.58 12.88 11.01 8.89 ...
## $ jp_sales : num 3.77 6.81 3.79 3.28 10.22 ...
## $ other_sales : num 8.46 0.77 3.31 2.96 1 0.58 2.9 2.85 2.26 0.47 ...
## $ global_sales: num 82.7 40.2 35.8 33 31.4 ...
```

```
##       rank           name              platform              year
##  Min.   :    1   Length:16598       Length:16598       Min.   :1980
##  1st Qu.: 4151   Class :character   Class :character   1st Qu.:2003
##  Median : 8300   Mode  :character   Mode  :character   Median :2007
##  Mean   : 8301                                         Mean   :2006
##  3rd Qu.:12450                                         3rd Qu.:2010
##  Max.   :16600                                         Max.   :2020
##                                                        NA's   :271
##     genre            publisher            na_sales          eu_sales
##  Length:16598       Length:16598       Min.   : 0.0000   Min.   : 0.0000
##  Class :character   Class :character   1st Qu.: 0.0000   1st Qu.: 0.0000
##  Mode  :character   Mode  :character   Median : 0.0800   Median : 0.0200
##                                        Mean   : 0.2647   Mean   : 0.1467
##                                        3rd Qu.: 0.2400   3rd Qu.: 0.1100
##                                        Max.   :41.4900   Max.   :29.0200
##
##     jp_sales         other_sales        global_sales
##  Min.   : 0.00000   Min.   : 0.00000   Min.   : 0.0100
##  1st Qu.: 0.00000   1st Qu.: 0.00000   1st Qu.: 0.0600
##  Median : 0.00000   Median : 0.01000   Median : 0.1700
##  Mean   : 0.07778   Mean   : 0.04806   Mean   : 0.5374
##  3rd Qu.: 0.04000   3rd Qu.: 0.04000   3rd Qu.: 0.4700
##  Max.   :10.22000   Max.   :10.57000   Max.   :82.7400
##

## [1] 16291
```

87. Utwórz wykres liniowy ilustrujący liczbę gier wydawanych w latach 1980-2016 (uwzględniający liczbę platform, czyli jeśli gra była wydana na 3 platformy to liczy się 3 razy).



88. Wyświetl 10 platform, na które sprzedano w sumie najwięcej gier (chodzi o globalną sprzedaż, uporządkuj wyniki malejąco) (*skorzystaj z funkcji sort i tapply z funkcją sum*).

```
##     PS2    X360     PS3     Wii      DS      PS     GBA     PSP     PS4      PC
## 1255.64  979.96  957.84  926.71  822.49  730.66  318.50  296.28  278.10  258.82
```

89. Spośród wszystkich gier z gatunku "Platform" wydanych na konsole XBox (XB, X360, XOne) wyświetl procent gier wydanych na poszczególne wersje XBoxa (*pomocne może być utworzenie ramki z grami na te platformy i z gatunku "Platform"*).

```
##
##       X360         XB       XOne
## 0.31168831 0.63636364 0.05194805
```

90. Ile wydano różnych gier z "Mario" w tytule? (uwaga - "Mario" nie ma być częścią wyrazu, nie może też mieć koło siebie żadnych znaków, które nie są spacjami).

```
## [1] 109
```

Zadania powtórkowe

91. Wczytaj dane z pliku *titanic.csv*.

a) Kolumnę *Survived* zamień na zmienną czynnikową z wartościami "not" i "yes".
b) Wyświetl średni wiek osób z klasy pierwszej z podziałem na tych, którzy przeżyli i nie przeżyli.

```
##      not      yes
## 43.69531 35.36820
```

92. Pracuj na danych o pasażerach Titanika.

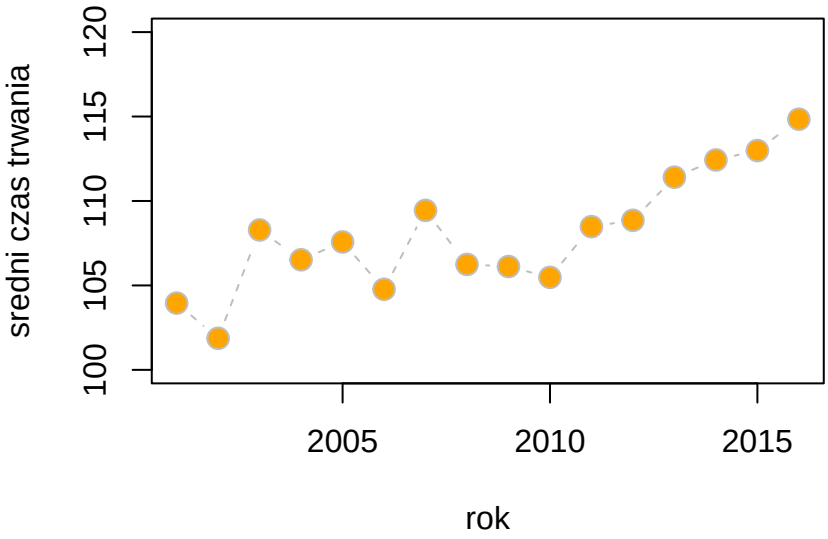
a) Utwórz nową kolumnę o nazwie *Fare_interval*, która będzie zmienną czynnikową informującą o przedziale, w którym znajduje się wartość kolumny *Fare* (ustaw przedziały na 0-10, 10-20, 20-50, 50-100, 100-500 oraz 500 i więcej)
b) Sprawdź jaki procent osób z każdej klasy zapłacił za bilet cenę z danego przedziału (wynik zaokrąglij do dwóch miejsc po przecinku)

```
##
##     (0,10] (10,20] (20,50] (50,100] (100,500] (500,Inf]
##   1   0.47    0.00   33.65    40.76     23.70      1.42
##   2   0.00   55.06   41.01     3.93      0.00      0.00
##   3  65.71   16.63   14.78     2.87      0.00      0.00
```

93. Pracuj na danych o pasażerach Titanika. Wyświetl średnią cenę biletów osób z poszczególnych portów (pomiń osoby bez informacji o porcie) (*można to zrobić na różne sposoby*).

```
##        C        Q        S
## 59.95414 13.27603 27.07981
```

94. Wczytaj dane z pliku *movies.csv*. Na ich podstawie utwórz wykres maksymalnie zbliżony do poniższego (*kolor linii nie jest czarny tylko trochę jaśniejszy, a punkty sa trochę większe niż standardowo*):



95. Na podstawie danych z pliku *movies.csv* wyświetl nazwiska trzech reżyserów z najwyższą średnią oceną filmów (tę średnią ocen też wyświetl).

```
##   Damien Chazelle Christopher Nolan       Lee Unkrich
##          8.500000          8.414286          8.300000
```

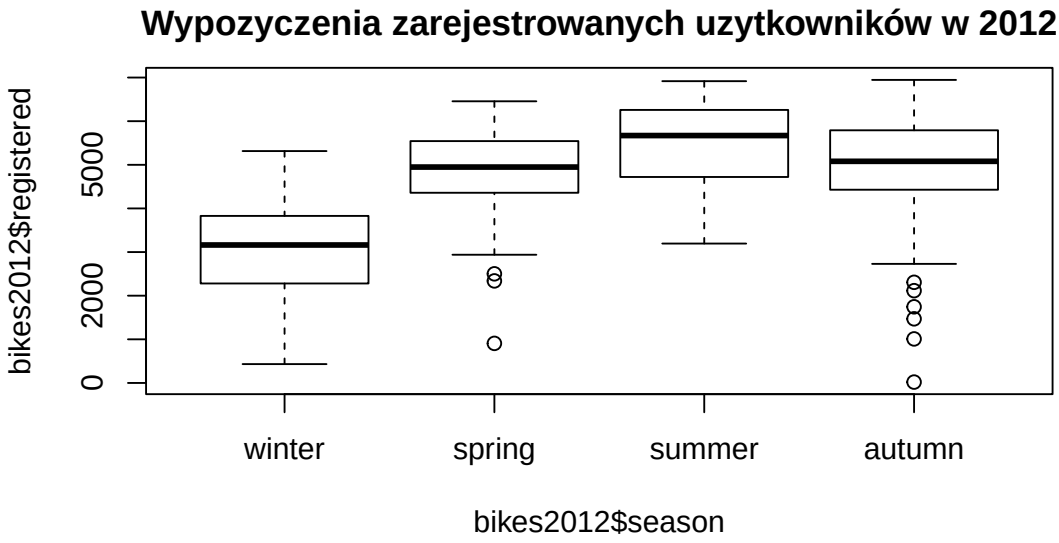
96. Wyświetl nazwiska reżyserów, tytuły i oceny filmów dla trzech reżyserów z zadania 95. (*jeśli masz rozwiązanie zadania 95., to zastosowanie funkcji* `names` *do niego zwróci nazwiska, które można użyć jako filtr do ramki*).

```
##              director                title rating
## 214  Christopher Nolan              Insomnia    7.2
## 542  Christopher Nolan         Batman Begins    8.3
## 646  Christopher Nolan           The Prestige    8.5
## 840  Christopher Nolan       The Dark Knight    9.0
## 1057       Lee Unkrich           Toy Story 3    8.3
## 1058 Christopher Nolan             Inception    8.8
## 1267 Christopher Nolan The Dark Knight Rises    8.5
## 1454   Damien Chazelle              Whiplash    8.5
## 1455 Christopher Nolan          Interstellar    8.6
```

(*jeśli nie znasz któregoś z tych filmów, to koniecznie to nadrób, no może poza "The Dark Knight Rises"*)

97. Wczytaj dane z pliku *bikes.csv*.

a) Kolumnę *season* zamień na factor z wartościami "winter", "spring", "summer", "autumn", a kolumnę *yr* na factor z wartościami 2011 i 2012.
b) Utwórz ramkę `bikes2012` zawierającą tylko dane z 2012 roku.
c) Utwórz następujący wykres (zwróć uwagę na kolor pudełek):



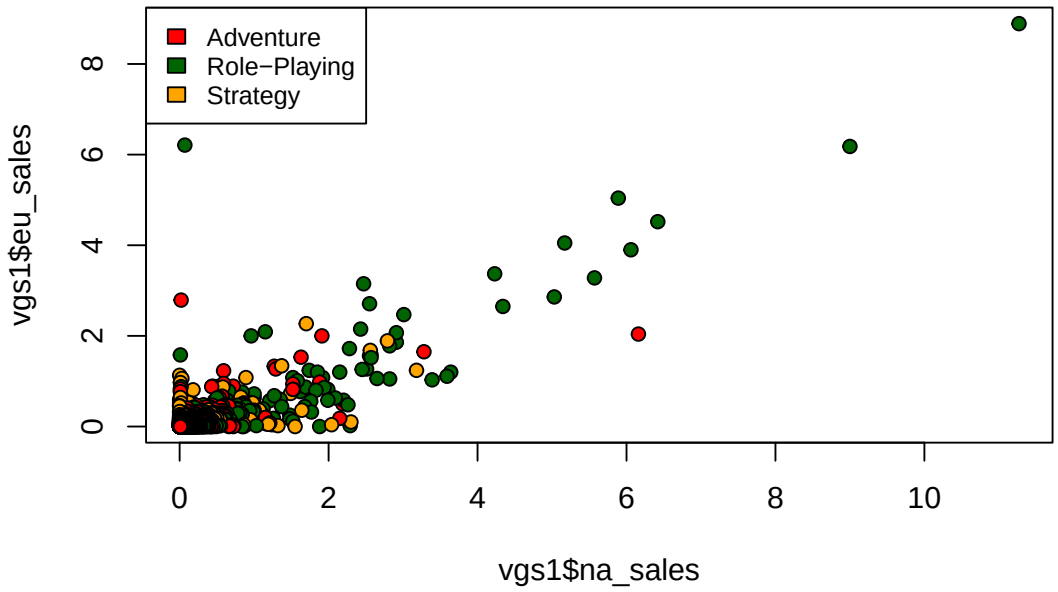
98. Pracuj na pełnych danych z pliku *bikes.csv*.

a) Zamień kolumnę *weathersit* na factor z wartościami "clear", "mist", "snow or rain".
b) Utwórz nową ramkę z wartościami z kwietnia 2011 roku.
c) Czy warunki pogodowe miały wpływ na liczbę wypożyczeń w kwietniu 2011? (*uzasadnij odpowiedź sprawdzając średnią wszystkich wypożyczeń z ramki utworzonej w poprzednim punkcie w zależności od warunków pogodowych*).

```
##        clear         mist snow or rain
##     3742.286     2778.867      795.000
```

99. Wczytaj dane z pliku *vgs.csv*.

a) Utwórz ramkę `vgs1`, która będzie zawierała tylko informacje o grach z gatunku "Adventure", "Role-Playing" lub "Strategy" (*łatwo to zrobić korzystając z operatora* `%in%`).
b) Zamień kolumnę *genre* na factor.
c) Utwórz następujący wykres (możesz wybrać inne kolory):



Zadanie dodatkowe

100! Pracuj na danych z pliku *movies.csv*. Utwórz poniższy wykres przedstawiający średni przychód filmów reżyserów, którzy wyreżyserowali przynajmniej 6 filmów (nazwiska muszą być widoczne w całości). Kod rozwiązania nie powinien korzystać z funkcji z żadnych dodatkowych pakietów.

(*hint: wszystkie konieczne funkcje już były w innych zadaniach, problemem mogą być tylko marginesy wykresu i ewentualnie pewne jego parametry*)