

Excel w R

Karol Klimas Why R? 2018

Kraków 17.04.2018

Agenda

1. Problem ogólny
2. Rozwiązanie
3. Problem szczegółowy
4. Rozwiązanie problemu szczegółowego
5. Punkty 3-4 powtórzone wiele razy ...
6. Wynik
7. Wnioski

Chcemy odtworzyć zawartość arkusza kalkulacyjnego ...

| | | | | | | |
|------------------------|---------------------------|--------------------|-----------------|----------------------|-----------------------|---------------------|
| Koszty stałe budowy | 150 000 | | | | | |
| Koszty stałe na piętro | 40 000 | | | | | |
| Koszt za metr | 3 000 | | | | | |
| Zmiana standardu | 1 000 | na plus gdy Wysoki | | na minus gdy Niski | | |
| Nr planu | Cena za m2 | Działka | Liczba mieszkań | Metraż | Standard 1 | Liczba pięter |
| Nasz plan | 7 500 | 1 | 60 | 70 | Wysoki | 5 |
| | Rozpoczęcie budowy w dniu | Kredyt w dniu | Wartość kredytu | Czas spłaty w dniach | Reklama dziennie | Intensywna kampania |
| | 2 | 15 | 6 000 000 | 1 095 | 300 | 100 |
| | Czas trwania budowy | | | Max | Mnożnik intensywności | |
| | 1 200 | | | 300 | 3 | |
| Plan | | | | | | |
| A | 7 500 | 1 | 40 | 65 | Średni | |
| | 50 | 200 | 4 500 000 | 1 100 | 250 | 200 |
| B | 8 500 | 1 | 40 | 80 | Wysoki | |
| | 10 | 50 | 6 000 000 | 1300 | 300 | 600 |
| C | 7 500 | 2 | 50 | 70 | Wysoki | |
| | 40 | 100 | 5000000 | 950 | 300 | 400 |
| D | 6 000 | 2 | 70 | 40 | Średni | |
| | 100 | 95 | 3500000 | 1100 | 250 | 300 |
| E | 6 500 | 3 | 55 | 75 | Średni | |
| | 200 | 195 | 3500000 | 900 | 150 | 150 |
| F | 5 500 | 3 | 50 | 60 | Niski | |
| | 2 | 1 | 3000000 | 1600 | 100 | 100 |

| | | | | | | | | | | | |
|------------------|-------------|--------------|------------------|--------------|-----------|--------------------------------|-------|-----------------|------|----|----|
| | Czas budowy | 1 200 | Parametr | 7 | Wypadki | Duży | 0,001 | Mały | 0,01 | | |
| | | | | | Wpływ | Dużego | 5 | Małego | 30 | | |
| czy jest budowa? | Wypadki | % ukończenia | Stopa procentowa | Intensywność | | Liczba klientów, którzy kupili | | | | | |
| 0 | | 0% | 2,72% | 27,23 | | | | | | | |
| 0 0,9034031 | 0 | 0,0% | 2,73% | 43,96 | 0,2331805 | 0 | | Sprzedane | 19 | na | 60 |
| 0 0,4911290 | 0 | 0,0% | 2,74% | 43,89 | 0,2788651 | 0 | | | | | |
| 0 0,0799643 | 0 | 0,0% | 2,75% | 43,82 | 0,9929011 | 0 | | Liczba wypadków | | | |
| 0 0,9110438 | 0 | 0,0% | 2,76% | 43,75 | 0,8924617 | 0 | | Mały | 13 | | |
| 0 0,0442444 | 0 | 0,0% | 2,77% | 43,68 | 0,4236082 | 0 | | Duży | 1 | | |
| 0 0,3059265 | 0 | 0,0% | 2,78% | 43,61 | 0,9401341 | 0 | | | | | |
| 0 0,1565883 | 0 | 0,0% | 2,79% | 43,54 | 0,5663473 | 0 | | | | | |
| 0 0,1620782 | 0 | 0,0% | 2,80% | 43,47 | 0,8923096 | 0 | | | | | |
| 0 0,5100120 | 0 | 0,0% | 2,81% | 43,40 | 0,6749964 | 0 | | | | | |
| 0 0,7102972 | 0 | 0,0% | 2,82% | 43,33 | 0,2009845 | 0 | | | | | |
| 0 0,0855958 | 0 | 0,0% | 2,83% | 43,26 | 0,0850362 | 0 | | | | | |
| 0 0,7913457 | 0 | 0,0% | 2,84% | 43,19 | 0,1604088 | 0 | | | | | |
| 0 0,7669054 | 0 | 0,0% | 2,85% | 43,12 | 0,4099433 | 0 | | | | | |
| 0 0,2014569 | 0 | 0,0% | 2,86% | 43,05 | 0,2875395 | 0 | | | | | |
| 0 0,9681679 | 0 | 0,0% | 2,87% | 42,98 | 0,0271357 | 0 | | | | | |
| 0 0,3310096 | 0 | 0,0% | 2,88% | 42,91 | 0,3487723 | 0 | | | | | |
| 1 0,0624244 | 0 | 0,1% | 2,89% | 42,84 | 0,0918303 | 0 | | | | | |
| 1 0,2030576 | 0 | 0,2% | 2,90% | 42,78 | 0,3860714 | 0 | | | | | |
| 1 0,9295181 | 0 | 0,3% | 2,91% | 42,71 | 0,4115625 | 0 | | | | | |
| 1 0,7290112 | 0 | 0,3% | 2,92% | 42,65 | 0,5870676 | 0 | | | | | |
| 1 0,4725993 | 0 | 0,4% | 2,93% | 42,58 | 0,5535760 | 0 | | | | | |
| 1 0,0058729 | Mały | 0,1% | 2,94% | 42,52 | 0,3344239 | 0 | | | | | |
| 1 0,5089160 | 0 | 0,2% | 2,95% | 42,43 | 0,1723072 | 0 | | | | | |
| 1 0,3090084 | 0 | 0,3% | 2,96% | 42,36 | 0,2976280 | 0 | | | | | |
| 1 0,0000000 | 0 | 0,0% | 2,97% | 42,28 | 0,0000000 | 0 | | | | | |

| | | | | |
|--------------|-----------|-------------|---------|------------------|
| | | | | |
| Koszt łączny | Przychody | Zysk | Marża | Zwrot z kapitału |
| 0 | | | | |
| 0 | | | | |
| 0 | | | | |
| 20 361 936 | 9 975 000 | -10 386 936 | -51,81% | -116,06% |

ale same dane nam nie wystarczą, bo ważne są
formuły!

```
> head(data_values[[5]])
  X1      X2      X3      X4      X5      X6      X7      X8
1 <NA> Stałe koszty reklamy <NA>      300      <NA> Koszty opreacyjne 0.02      NA
2 Okres      Wpływy Koszty reklamy Koszty operacyjne Koszty stałe budowy Koszty zmienne budowy Gotówka własna      NA
3 0      <NA>      <NA>      <NA>      <NA>      <NA>      7e+06      NA
4 1      0      900 285.833333333333      0      0 6998814.16666667 0.1524300
5 2      0      900 285.833333333333      0      0 6997628.33333333 0.0598159
6 3      0      900 285.833333333333      0      0 6996442.5 0.9689986

  X9      X10      X11      X12      X13      X14      X15      X16
1 Koszty stałe 350000 Koszt zmienny 16800000 Okres spłaty 1095      <NA> Warunki kredytu
2 Kredyt Rata kapitałowa      Odsetki Gotówka z kredytem      <NA>      NA      <NA>      <NA>
3      <NA>      <NA>      <NA>      <NA>      <NA>      NA      Wkład własny      0.25
4      0      0      0 6998814.16666667 6e+06      NA Etap zaawansowania      <NA>
5      0      0      0 6997628.33333333      <NA>      NA Wartość działki 1950000
6      0      0      0 6996442.5      <NA>      NA Próg      0.95
```

```
> head(data_formulas[[4]], 4)
  X1      X2      X3
1 MdM      79      85
2 <NA>      Parametry      <NA>
3 Okres      Standard      Lokalizacja
4 0 IF(Wybor_inwestycji!F7="Wysoki",1,IF(Wybor_inwestycji!F7="Niski",0,0.5)) (3-Wybor_inwestycji!$C$7)/2

  X4
1 6126.9
2 <NA>
3 MdM
4 IF(Wybor_inwestycji!E7<B1,1,IF(Wybor_inwestycji!E7<C1,0.8,0))*IF(Wybor_inwestycji!B7<=Sprzedaz_mieszkan!D1,1,0)

  X5
1 Średnia cena m2
2 <NA>
3 Intensywna kampania
4 IF(A4<Wybor_inwestycji!$G$9,Wybor_inwestycji!$G$11,1)

  X6      X7      X8
1 <NA>      7500      <NA>
2 <NA>      <NA>      <NA>
3 Reklama Liczba sprzedanych mieszkań      Cena m2
4 (E5*Wybor_inwestycji!$F$9)/(Wybor_inwestycji!$G$11*Wybor_inwestycji!$F$11) 0 Wybor_inwestycji!B7

  X9      X10      X11      X12      X13      X14      X15
1 <NA> Czas budowy Wybor_inwestycji!B11      <NA>      Parametr      7 Wypadki
```

| | | |
|------------|--|---------------------------|
| | | |
| | | |
| Zysk | Marża | Zwrot z kapitału własnego |
| | | |
| | | |
| | | |
| | | |
| -6 186 936 | $\cdot f_X \sim$ J7 ÷ (B7 + Finansowanie::Table 1::G4) | -69,13% |
| | | |

Rozwiązaniem problemu jest zastosowanie pakietu xlsx

```
~/Desktop/Excel_in_R/ ➜  
> install.packages("xlsx")  
also installing the dependencies 'rJava', 'xlsxjars'
```

```
excel_with_formulas <-  
  xlsx::read.xlsx("./files/ES Klimas arkusz.xlsx",  
                  sheetIndex = 1, keepFormulas = TRUE)
```

```
~/Desktop/Excel_in_R/ ➜  
> install.packages("rJava", type="source")  
trying URL 'https://cran.rstudio.com/src/contrib/rJava_0.9-9.tar.gz'  
Content type 'application/x-gzip' length 660454 bytes (644 KB)  
=====  
downloaded 644 KB  
  
* installing *source* package 'rJava' ...  
** package 'rJava' successfully unpacked and MD5 sums checked  
checking for gcc... /usr/local/clang4/bin/clang  
checking whether the C compiler works... no  
configure: error: in `/private/var/folders/01/f6ld002s4bbdn0zd_7pzmX2m0000gn/T/Rtmpb870bS/R.INSTALL55b45bef294/rJava':  
configure: error: C compiler cannot create executables  
See `config.log' for more details  
ERROR: configuration failed for package 'rJava'  
* removing '/Library/Frameworks/R.framework/Versions/3.4/Resources/library/rJava'  
Warning in install.packages :  
  installation of package 'rJava' had non-zero exit status  
  
The downloaded source packages are in  
  '/private/var/folders/01/f6ld002s4bbdn0zd_7pzmX2m0000gn/T/RtmpEYUofo/downloaded_packages'
```

Instalacja pakietu może powodować problemy!

Szczegółowe problemy krok po kroku

Co mamy?

| | X13 | | | | | X14 | X15 | |
|---|------------------|--|--------|-------|-----------|-------------------|---------|---------------------|
| 1 | Parametr | | | | | 7 | Wypadki | |
| 2 | <NA> | | | | | <NA> | Wpływ | |
| 3 | Stopa procentowa | | | | | Intensywność | <NA> | |
| 4 | 0.0272 | | | | | AVERAGE(N5:N1829) | <NA> | |
| 5 | 0.0273 | (SUM(B4:F4,L4)+(3%-M4)*100+(\$G\$1-H4)/1000+2*G4/Wybor_inwestycji!\$D\$7)*\$N\$1 | | | | | RAND() | |
| 6 | 0.0274 | (SUM(B5:F5,L5)+(3%-M5)*100+(\$G\$1-H5)/1000+2*G5/Wybor_inwestycji!\$D\$7)*\$N\$1 | | | | | RAND() | |
| | | | X16 | X17 | X18 | X19 | X20 | X21 |
| 1 | | | Duży | 0.001 | Mały | 0.01 | <NA> | <NA> |
| 2 | | | Dużego | 5.000 | Małego | 30 | <NA> | <NA> |
| 3 | | Liczba klientów, którzy kupili | | NA | <NA> | <NA> | <NA> | <NA> |
| 4 | | <NA> | | NA | <NA> | <NA> | <NA> | <NA> |
| 5 | | IF(G5<Wybor_inwestycji!\$D\$7,IF(O5<N5/\$A\$1829,1,0),0) | | NA | Sprzedane | SUM(P5:P1829) | na | Wybor_inwestycji!D7 |
| 6 | | IF(G6<Wybor_inwestycji!\$D\$7,IF(O6<N6/\$A\$1829,1,0),0) | | NA | <NA> | <NA> | <NA> | <NA> |

Co chcemy mieć?

```
> data_test
  cell value formula
1   A1     1    <NA>
2   A2     2    <NA>
3   A3    NA sum(A1,A2)
```

```
> evaluate_formulas(data_test)
  cell value
1   A3     3
```


Wczytanie danych z excela - zarówno formuły jak i wartości

```
wb <- openxlsx::loadWorkbook("./files/model.xlsx")
data_formulas <- lapply(1:length(names(wb)), function(x) try(xlsx::read.xlsx(file = "./files/model.xlsx",
  sheetIndex = x, header = FALSE, keepFormulas = TRUE, encoding = "UTF-8")))
data_values <- lapply(1:length(names(wb)), function(x) try(xlsx::read.xlsx(file = "./files/model.xlsx",
  sheetIndex = x, header = FALSE, keepFormulas = FALSE, encoding = "UTF-8")))
```

```
data_formulas <- lapply(data_formulas, function(x) {names(x) <- LETTERS[1:ncol(x)]; x})
data_formulas <- lapply(data_formulas, function(x) melt(data.frame(x, rows = 1:nrow(x)),
  id.vars = c("rows"), value.name = "formula", variable.name = "columns"))
```

```
data_values <- lapply(data_values, function(x) {names(x) <- LETTERS[1:ncol(x)]; x})
data_values <- lapply(data_values, function(x) melt(data.frame(x, rows = 1:nrow(x)),
  id.vars = c("rows"), value.name = "value", variable.name = "columns"))
```

▼ # Łączymy dane =====

```
data_total <- mapply(function(u, v) full_join(u, v, by = c("rows", "columns")), SIMPLIFY = FALSE,
  data_values, data_formulas)
```

```
> head(data_total[[6]] %>% filter(value != formula))
  rows columns      value formula
1    6      B 1950000 VLOOKUP(Wybor_inwestycji!$C$7,Dane!$A$2:$E$4,5,FALSE)
2    6      C   606900 SUM(Finansowanie!C5:C1829)
3    6      D 17150000 Wybor_inwestycji!J7
4    6      E   343000 D7*Finansowanie!G1
5    6      F 20049900 SUM(B7:E7)
6    6      G 312036.245449509 SUM(Finansowanie!J5:K1829)-SUM(Finansowanie!I5:I1829)
```


Żmudny proces dostosowania formuł z Excela do środowiska R

```
# 1. Wywalamy znaczki '$'  
data_total <- lapply(data_total, function(x) x %>% mutate(formula = stri_replace_all_fixed(formula, "$", "")))
```

| | rows | columns | value | formula |
|---|------|---------|------------------|---|
| 1 | 6 | B | 1950000 | VLOOKUP(Wybor_inwestycji!C7,Dane!A2:E4,5,FALSE) |
| 2 | 6 | C | 606900 | SUM(Finansowanie!C5:C1829) |
| 3 | 6 | D | 17150000 | Wybor_inwestycji!J7 |
| 4 | 6 | E | 343000 | D7*Finansowanie!G1 |
| 5 | 6 | F | 20049900 | SUM(B7:E7) |
| 6 | 6 | G | 312036.245449509 | SUM(Finansowanie!J5:K1829)-SUM(Finansowanie!I5:I1829) |

```
# 2. Teraz dopisujemy nazwe arkusza przed kazde odniesienie do komorki  
# a) Wtedy, kiedy jest tam ':', a nazwa arkusza jest tylko przy jednym: 'Dane!A2:E4' -> 'Dane!A2:Dane!E4'  
data_total <- lapply(data_total, function(x) x %>%  
  mutate(formula = gsub("([a-zA-Z_]+[!])([A-Z]{1})([0-9]{1,4})(:)([A-Z]{1})([0-9]{1,4})",  
    "\\1\\2\\3\\4\\1\\5\\6", formula)))
```

```
> head(data_total[[6]] %>% filter(value != formula))
```

| | rows | columns | value | formula |
|---|------|---------|------------------|---|
| 1 | 6 | B | 1950000 | VLOOKUP(Wybor_inwestycji!C7,Dane!A2:Dane!E4,5,FALSE) |
| 2 | 6 | C | 606900 | SUM(Finansowanie!C5:Finansowanie!C1829) |
| 3 | 6 | D | 17150000 | Wybor_inwestycji!J7 |
| 4 | 6 | E | 343000 | D7*Finansowanie!G1 |
| 5 | 6 | F | 20049900 | SUM(B7:E7) |
| 6 | 6 | G | 312036.245449509 | SUM(Finansowanie!J5:Finansowanie!K1829)-SUM(Finansowanie!I5:Finansowanie!I1829) |


```
# b) W pozostałych przypadkach dodajemy nazwę arkusza tak, gdzie nie ma wcześniej odwołania do innego arkusza
data_total <- mapply(function(u, v) u %>% mutate(formula = gsub("([!])([A-Z]{1}[0-9]{1,4})",
                                                    stri_join("\\1", v, "!", "\\2"), formula)),
                    data_total, sheet_names, SIMPLIFY = FALSE)
```

| | rows | columns | value | formula |
|---|------|---------|------------------|---|
| 1 | 6 | B | 1950000 | VLOOKUP(Wybor_inwestycji!C7,Dane!A2:Dane!E4,5,FALSE) |
| 2 | 6 | C | 606900 | SUM(Finansowanie!C5:Finansowanie!C1829) |
| 3 | 6 | D | 17150000 | Wybor_inwestycji!J7 |
| 4 | 6 | E | 343000 | D7*Finansowanie!G1 |
| 5 | 6 | F | 20049900 | SUM(Wyniki_symulacji!B7:Wyniki_symulacji!E7) |
| 6 | 6 | G | 312036.245449509 | SUM(Finansowanie!J5:Finansowanie!K1829)-SUM(Finansowanie!I5:Finansowanie!I1829) |

```
# Uwzględniamy początek
data_total <- mapply(function(u, v) u %>% mutate(formula = gsub("^[A-Z]{1}[0-9]{1,4}",
                                                    stri_join(v, "!", "\\1"), formula)),
                    data_total, sheet_names, SIMPLIFY = FALSE)
```

| | rows | columns | value | formula |
|---|------|---------|------------------|---|
| 1 | 6 | B | 1950000 | VLOOKUP(Wybor_inwestycji!C7,Dane!A2:Dane!E4,5,FALSE) |
| 2 | 6 | C | 606900 | SUM(Finansowanie!C5:Finansowanie!C1829) |
| 3 | 6 | D | 17150000 | Wybor_inwestycji!J7 |
| 4 | 6 | E | 343000 | Wyniki_symulacji!D7*Finansowanie!G1 |
| 5 | 6 | F | 20049900 | SUM(Wyniki_symulacji!B7:Wyniki_symulacji!E7) |
| 6 | 6 | G | 312036.245449509 | SUM(Finansowanie!J5:Finansowanie!K1829)-SUM(Finansowanie!I5:Finansowanie!I1829) |

```
# 3. Tam, gdzie formuły są takie same, jak wartości, to nie są formułami
data_total <- lapply(data_total,
                    function(x) {x$formula[x$formula == x$value] <- NA; return(x)})
```


Finansowanie!K1829)

```
# 4. Wywalamy wszystkie spacje i od teraz odniesienia do komorek zaczynaja i koncza sie spacja " "
data_total <- lapply(data_total, function(x) x %>% mutate(formula = stri_replace_all_regex(formula, "[ ]", "")))
data_total <- lapply(data_total, function(x) x %>% mutate(formula = gsub("([a-zA-Z_]+[!][A-Z]{1}[0-9]{1,4})",
" \\1 ", formula)))
```

| rows | columns | value | formula |
|------|---------|--------------------|---|
| 1 | 6 | B 1950000 | VLOOKUP(Wybor_inwestycji!C7 , Dane!A2 : Dane!E4 ,5,FALSE) |
| 2 | 6 | C 606900 | SUM(Finansowanie!C5 : Finansowanie!C1829) |
| 3 | 6 | D 17150000 | Wybor_inwestycji!J7 |
| 4 | 6 | E 343000 | Wyniki_symulacji!D7 * Finansowanie!G1 |
| 5 | 6 | F 20049900 | SUM(Wyniki_symulacji!B7 : Wyniki_symulacji!E7) |
| 6 | 6 | G 312036.245449509 | SUM(Finansowanie!J5 : Finansowanie!K1829)-SUM(Finansowanie!I5 : Finansowanie!I1829) |

```
# 5. Dodajemy nazwy komorek zapisane jak w Excelu
data_total <- mapply(function(u, v) u %>% mutate(sheet_name = stri_trans_tolower(v),
cell = stri_join(" ", v, "!", columns, rows, " ")),
data_total, sheet_names, SIMPLIFY = FALSE)
```

```
> head(data_total[[6]] %>% filter(value != formula), n = 5)
```

| rows | columns | value | formula | sheet_name | cell |
|------|---------|------------|--|------------------|---------------------|
| 1 | 6 | B 1950000 | VLOOKUP(Wybor_inwestycji!C7 , Dane!A2 : Dane!E4 ,5,FALSE) | wyniki_symulacji | Wyniki_symulacji!B6 |
| 2 | 6 | C 606900 | SUM(Finansowanie!C5 : Finansowanie!C1829) | wyniki_symulacji | Wyniki_symulacji!C6 |
| 3 | 6 | D 17150000 | Wybor_inwestycji!J7 | wyniki_symulacji | Wyniki_symulacji!D6 |
| 4 | 6 | E 343000 | Wyniki_symulacji!D7 * Finansowanie!G1 | wyniki_symulacji | Wyniki_symulacji!E6 |
| 5 | 6 | F 20049900 | SUM(Wyniki_symulacji!B7 : Wyniki_symulacji!E7) | wyniki_symulacji | Wyniki_symulacji!F6 |


```
# 7. Arkusze beda pisane malymi literami
data_total <- data_total %>% mutate(formula = stri_replace_all_regex(formula, sheet_names,
                              stri_trans_tolower(sheet_names), vectorize_all = FALSE),
                              cell = stri_replace_all_regex(cell, sheet_names,
                              stri_trans_tolower(sheet_names), vectorize_all = FALSE))
```

| rows | columns | value | formula |
|------|------------------|---------------------|---|
| 1 | 9 | D 6e+06 | 6*10^6 |
| 2 | 9 | E 1095 | 3*365 |
| 3 | 7 | G 5 | ROUNDUP(wybor_inwestycji!D7 * wybor_inwestycji!E7 /(VLOOKUP(wybor_inwestycji!C7 , dane!A2 : dane!C4 ,3,FALSE)*0.8),0) |
| 4 | 7 | H 350000 | wybor_inwestycji!B1 + wybor_inwestycji!G7 * wybor_inwestycji!B2 |
| | sheet_name | cell | |
| 1 | wybor_inwestycji | wybor_inwestycji!D9 | |
| 2 | wybor_inwestycji | wybor_inwestycji!E9 | |
| 3 | wybor_inwestycji | wybor_inwestycji!G7 | |
| 4 | wybor_inwestycji | wybor_inwestycji!H7 | |

| rows | columns | value | formula | sheet_name | cell |
|------|---------|-----------------|---|-------------------|----------------------|
| 4 | N | 27.581901826484 | AVERAGE(sprzedaz_mieszkan!N5 : sprzedaz_mieszkan!N1829) | sprzedaz_mieszkan | sprzedaz_mieszkan!N4 |

```
formula
sprzedaz_mieszkan!N5 : sprzedaz_mieszkan!N1829 )
```

```
> replace_range_cells(" sprzedaz_mieszkan!N5 : sprzedaz_mieszkan!N8 ")
[[1]]
[1] " sprzedaz_mieszkan!N5 , sprzedaz_mieszkan!N6 , sprzedaz_mieszkan!N7 , sprzedaz_mieszkan!N8 "
```

```
> find_and_replace_range("IF(SUM( arkusz!I5 : arkusz!I10 )<= sprzedaz_mieszkan!K1 ,1,0)")
[1] "IF(SUM( arkusz!I5 , arkusz!I6 , arkusz!I7 , arkusz!I8 , arkusz!I9 , arkusz!I10 )<= sprzedaz_mieszkan!K1 ,1,0)"
```

```
> Rprof()
> data_total <- data_total %>% mutate(formula = find_and_replace_range(formula))
> Rprof(NULL)
> summaryRprof()
$by.self
```

| | self.time | self.pct | total.time | total.pct |
|--------------------------|-----------|----------|------------|-----------|
| "as.character" | 197.56 | 82.35 | 197.56 | 82.35 |
| ".Call" | 32.66 | 13.61 | 239.90 | 100.00 |
| "gsub" | 2.66 | 1.11 | 2.68 | 1.12 |
| "lapply" | 1.08 | 0.45 | 226.90 | 94.58 |
| "mapply" | 0.92 | 0.38 | 14.74 | 6.14 |
| "seq.default" | 0.84 | 0.35 | 1.28 | 0.53 |
| "FUN" | 0.78 | 0.33 | 226.74 | 94.51 |
| "stri_replace_all_regex" | 0.68 | 0.28 | 19.62 | 8.18 |
| "match.fun" | 0.40 | 0.17 | 0.42 | 0.18 |
| "<Anonymous>" | 0.38 | 0.16 | 13.74 | 5.73 |
| "pmin" | 0.28 | 0.12 | 0.42 | 0.18 |

Teraz przechodzimy do implementacji funkcji z Excela

Secure | <https://support.office.com/en-us/article/max-function-e0012414-9ac8-4b34-9a47-73e662c08098>

MAX function

Applies To: Excel 2016, Excel 2013, Excel 2010, Excel 2007, Excel 2016 for Mac, More...

This article describes the formula syntax and usage of the **MAX** function in Microsoft Excel.

Description

Returns the largest value in a set of values.

Syntax

MAX(number1, [number2], ...)

The MAX function syntax has the following arguments:

- **Number1, number2, ...** Number1 is required, subsequent numbers are optional. 1 to 255 numbers for which you want to find the maximum value.

Remarks

- Arguments can either be numbers or names, arrays, or references that contain numbers.
- Logical values and text representations of numbers that you type directly into the list of arguments are counted.

```
> table(unlist(stri_extract_all_regex(data_total$formula, "[A-Z]+\\(")))
```

| AND(| AVERAGE(| COUNTIF(| IF(| MAX(| MIN(| RAND(| ROUNDUP(| SUM(| VLOOKUP(|
|------|----------|----------|-------|------|------|-------|----------|-------|----------|
| 3650 | 1 | 2 | 27368 | 1825 | 3649 | 5475 | 1 | 16419 | 3 |

```
# MAX
MAX <- max
# MIN
MIN <- min
# SUM
SUM <- sum
# IF
IF <- `if`
# AND
AND <- `&`
# RAND
RAND <- function() runif(1)
```

```
# AVERAGE
```

```
AVERAGE <- function(...) if(all(lengths(list(...)) == 1)) sum(...) / length(list(...)) else  
  stop("Length of all arguments must equal 1")
```

```
# COUNTIF - wymaga odpowiedniego przygotowania formu
```

```
COUNTIF <- function(x, y) sum(sapply(stri_join(x, y), function(u) eval(parse(text = u))))
```

```
# ROUNDUP
```

```
ROUNDUP <- function(x, rounding = 0) ceiling(x * 10 ^ (rounding)) * 10 ^ (-rounding)
```

```
# VLOOKUP
```

```
VLOOKUP <- function(x, data_range, col_number, nonsense, whole_data = data_total)
```

```
{
```

```
> COUNTIF(1:10, '>5')  
[1] 5  
> COUNTIF(1:10, '>=5')  
[1] 6
```

ce.com/en-us/article/VLOOKUP-function-0BBC8083-26FE-4963-8AB8-93A18AD188A1

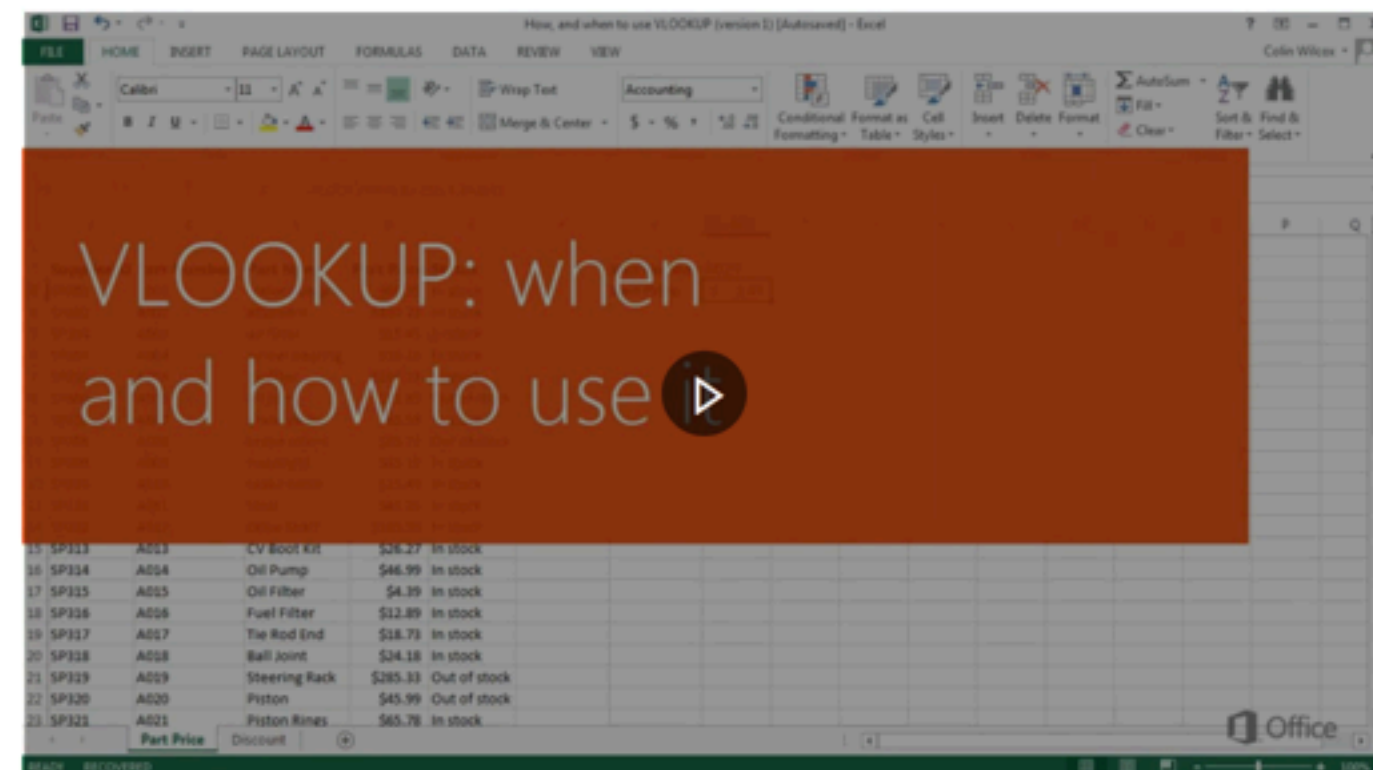
VLOOKUP function

Applies To: Excel 2016, Excel 2013, Excel 2010, Excel 2007, Excel 2016 for Mac, More...

Use VLOOKUP, one of the [lookup and reference functions](#), when you need to find things in a table or a range by row. For example, look up a price of an automotive part by the part number.

In its simplest form, the VLOOKUP function says:

=VLOOKUP(Value you want to look up, range where you want to lookup the value, the column number in the range containing the return value, Exact Match or Approximate Match – indicated as 0/FALSE or 1/TRUE).



This video is part of a training course called [VLOOKUP: When and how to use it](#).

Tip: The secret to VLOOKUP is to organize your data so that the value you look up (part number) is to the left of the return value you want to find (price of the part).

```
VLOOKUP <- function(x, data_range, col_number, nonsense, whole_data = data_total)  
{  
  temp_sheet_name <- stri_extract_first_regex(data_range, "[a-z]+!")  
  
  row_range <- unlist(stri_extract_all_regex(data_range, "[0-9]"))  
  
  col_range <- unlist(stri_extract_all_regex(data_range, "[A-Z]"))  
  col_range <- LETTERS[seq(from = min(match(col_range, LETTERS)), to = max(match(col_range, LETTERS)),  
    by = 1)]  
  
  col_range <- col_range[c(1, col_number)]  
  
  part_data <- data_total %>% filter(sheet_name == stri_replace_all_fixed(temp_sheet_name, "!", ""),  
    | rows <= max(row_range), rows >= min(row_range),  
    columns %in% col_range)  
  
  row_to_choose <- part_data$rows[part_data$columns == col_range[1] & part_data$value == x]  
  return(part_data$value[part_data$rows == row_to_choose & part_data$columns == col_range[2]])  
}
```



```
# Problemem są nazwy w wartościach
data_total <- data_total %>% mutate(character_value = value)
data_total <- data_total %>% mutate(value = stri_replace_all_fixed(value,
  c("Wysoki", "Niski", "Mały", "Duży"), c(1, 2, 3, 4), vectorize_all = FALSE),
  formula = stri_replace_all_fixed(formula,
    c("Wysoki", "Niski", "Mały", "Duży"), c(1, 2, 3, 4), vectorize_all = FALSE))
data_total <- data_total %>% mutate(value = as.numeric(value))
```

```
  rows columns value
1     5         N 43.96

formula
1 (SUM( sprzedaz_mieszkanB4 , sprzedaz_mieszkanC4 , sprzedaz_mies
  iezkanL4 )+(3%- sprzedaz_mieszkanM4 )*100+( sprzedaz_mieszkanG1
  estycjiD7 )* sprzedaz_mieszkanN1
      sheet_name          cell character_value znaki
```

```
# Podmiana procentów
data_total <- data_total %>% mutate(formula = stri_replace_all_fixed(formula, "%", "/100"))
```

```
> head(data_total %>% filter(grepl("VLOOK",formula)))[2:3, ] %>% mutate(formula = stri_replace_a
  rows columns value formula sheet_name
2     5         P 1950000 VLOOKUP( wybor_inwestycjiC7 , daneA2 : daneE4 ,5,FALSE) finansowanie
3     6         B 1950000 VLOOKUP( wybor_inwestycjiC7 , daneA2 : daneE4 ,5,FALSE) wyniki_symulacji
```

```
# Problem z "VLOOKUP"
data_total <- data_total %>% mutate(formula = gsub("(VLOOKUP\[()\]([a-zA-Z0-9_ ]+)(,)([a-zA-Z0-9_ :]+)(,)",
  "\\1\\2\\3'\\4'\\5", formula))
```

Przechodzimy do przeliczenia arkusza Excel

```
evaluate_formulas <- function(x)
{
  # Instead of attaching we use within
  x_to_attach <- x$value
  names(x_to_attach) <- stri_trim(x$cell)
  x_to_attach <- as.data.frame(t(x_to_attach))

  # Use only rows, which have formula (drop NA)
  has_formulas <- !is.na(x$formula)
  Sys.time()
  x_result <- sapply(x$formula[has_formulas], function(y) try(with(x_to_attach, eval(parse(text = y))),
  Sys.time()
  names(x_result) <- x$cell[has_formulas]

  suppressWarnings(x_result <- data.frame(cell = names(x_result), value = as.numeric(x_result)))
  return(x_result)
}
```

```
> Sys.time()
[1] "2018-04-16 22:21:37 CEST"
> x_result <- sapply(x$formula[has_formulas], function(y) try(with(x_to_attach, eval(parse(text = y))), silent = TRUE))
Warning message:
In doTryCatch(return(expr), name, parentenv, handler) :
  restarting interrupted promise evaluation
> Sys.time()
[1] "2018-04-17 01:27:41 CEST"
```


Za długo się liczy? Jest na to sposób

- Usunąć odwołania łańcuchowe

```
> data_total %>% filter(cell %in% head(data_total$cell[index][next_index]))
```

| | rows | columns | value | formula | sheet_name | cell | character_value |
|---|------|---------|-------|----------------------------|-------------------|----------------------|-----------------|
| 1 | 5 | B | 1 | <u>sprzedaz_mieszkanB4</u> | sprzedaz_mieszkan | sprzedaz_mieszkanB5 | 1 |
| 2 | 6 | B | 1 | sprzedaz_mieszkanB5 | sprzedaz_mieszkan | sprzedaz_mieszkanB6 | 1 |
| 3 | 7 | B | 1 | sprzedaz_mieszkanB6 | sprzedaz_mieszkan | sprzedaz_mieszkanB7 | 1 |
| 4 | 8 | B | 1 | sprzedaz_mieszkanB7 | sprzedaz_mieszkan | sprzedaz_mieszkanB8 | 1 |
| 5 | 9 | B | 1 | sprzedaz_mieszkanB8 | sprzedaz_mieszkan | sprzedaz_mieszkanB9 | 1 |
| 6 | 10 | B | 1 | sprzedaz_mieszkanB9 | sprzedaz_mieszkan | sprzedaz_mieszkanB10 | 1 |

```
> data_total %>% filter(cell %in% " sprzedaz_mieszkanB4 ")
```

| | rows | columns | value | formula | sheet_name |
|---|------|---------|-------|--|-------------------|
| 1 | 4 | B | 1 | <u>IF(wybor_inwestycjiF7 == "1",1,IF(wybor_inwestycjiF7 == "2",0,0.5))</u> | sprzedaz_mieszkan |

| | cell | character_value |
|---|----------------------------|-----------------|
| 1 | <u>sprzedaz_mieszkanB4</u> | 1 |

```
> data_total %>% filter(grepl(" sprzedaz_mieszkanB5 ", formula))
```

| | rows | columns | value |
|---|------|---------|-------|
| 1 | 6 | B | 1.00 |
| 2 | 6 | N | 43.89 |

| | formula | sheet_name | cell | character_value |
|---|---|-------------------|---------------------|-----------------|
| 1 | <u>sprzedaz_mieszkanB5</u> | sprzedaz_mieszkan | sprzedaz_mieszkanB6 | 1 |
| 2 | (SUM(<u>sprzedaz_mieszkanB5</u> , sprzedaz_mieszkanC5 , sprzedaz_mieszkanD5 , sprzedaz_mieszkanE5 , sprzedaz_mieszkanF5 , sprzedaz_mieszkanL5)+(3%- sprzedaz_mieszkanM5)*100+(sprzedaz_mieszkanG1 - sprzedaz_mieszkanH5)/100+2* sprzedaz_mieszkanG5 / wybor_inwestycjiD7)* sprzedaz_mieszkanN1 | sprzedaz_mieszkan | sprzedaz_mieszkanN6 | 43.89 |

Za długo się liczy? Jest na to sposób

- Wybierać tylko komórki potrzebne do wyliczenia wyników

```
> data_total %>% filter(cell %in% c(" wyniki_symulacjiK6 ", " wyniki_symulacjiL6 "))
  rows columns      value                                formula      sheet_name      cell
1     6         K -0.5442389      wyniki_symulacjiJ7 / wyniki_symulacjiF7 wyniki_symulacji wyniki_symulacjiK6
2     6         L -1.2192108      wyniki_symulacjiJ7 /( wyniki_symulacjiB7 + finansowanieG4 ) wyniki_symulacji wyniki_symulacjiL6
  character_value
```

```
> cells_to_get
[1] " wyniki_symulacjiJ7 " " wyniki_symulacjiF7 " " wyniki_symulacjiB7 " " finansowanieG4 "
```

```
> data_total %>% filter(cell %in% stri_replace_all_fixed(cells_to_get, "7", "6"))
  rows columns      value
1     4         G  6998814
2     6         B  1950000
3     6         F  20049900
4     6         J -10911936

                                formula
1      finansowanieG4 + finansowanieB5 -SUM( finansowanieC5 , finansowanieD5 , finansowanieE5 , finansowanieF5 )
2                                VLOOKUP( wybor_inwestycjiC7 , daneA2 : daneE4 ',5,FALSE)
3                                SUM( wyniki_symulacjiB7 , wyniki_symulacjiC7 , wyniki_symulacjiD7 , wyniki_symulacjiE7 )
4                                wyniki_symulacjiI7 - wyniki_symulacjiH7

  sheet_name      cell      character_value
1   finansowanie      finansowanieG4      6998814.16666667
2 wyniki_symulacji      wyniki_symulacjiB6      1950000
3 wyniki_symulacji      wyniki_symulacjiF6      20049900
4 wyniki_symulacji      wyniki_symulacjiJ6      -10911936.2454495
```


Za długo się liczy? Jest na to sposób

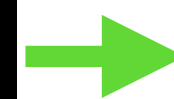
- Ustalić kolejność obliczania

```
> data_simple
  cell  value  formula
1  A1  0.10000    <NA>
2  A2 -0.15000    <NA>
3  B3  0.30000    A1 + A2
4  C7 19.95262 10^(1+ B3 )
```

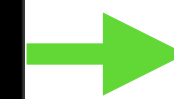


```
  cell  value  formula
1  A1  0.10000    <NA>
2  A2 -0.15000    <NA>
3  B3 -0.05000    A1 + A2
4  C7 19.95262 10^(1+ B3 )
```

```
  cell  value  formula c_order
1  A1  0.10000    <NA>        0
2  A2 -0.15000    <NA>        0
3  B3  0.30000    A1 + A2      1
4  C7 19.95262 10^(1+ B3 )    2
```



```
  cell  value  formula c_order
1  A1  0.10000    <NA>        0
2  A2 -0.15000    <NA>        0
3  B3 -0.05000    A1 + A2      1
4  C7 19.95262 10^(1+ B3 )    2
```



```
  cell  value  formula c_order
1  A1  0.10000    <NA>        0
2  A2 -0.15000    <NA>        0
3  B3 -0.05000    A1 + A2      1
4  C7  8.912509 10^(1+ B3 )    2
```

Dziękuję za uwagę

kontakt: karolklimas@gmail.com

Pre-meetings before Why R? 2018 Conference in Wroclaw

