Wizualizacja danych w R

Bartosz Kozak

13 10 2019

Podstawy - wczytywanie danych

Dane do ćwiczeń należy pobrać ze strony Następnie dane możemy wczytać za pomocą kodu

```
filename <- "~/cw2_data.txt"
filename
```

```
## [1] "~/cw2_data.txt"
```

```
mydata <- read.csv(filename, sep = "\t", header = F)
```

Możemy obejrzeć nasze dane używając funkcji `head`.

```
head(mydata)
```

```
##     V1    V2    V3                V4 V5 V6    V7    V8          V9
## 1 chr1 10000 10600 15_Repetitive/CNV  0  . 10000 10600 245,245,245
## 2 chr1 10600 11137 13_Heterochrom/lo  0  . 10600 11137 245,245,245
## 3 chr1 11137 11737       8_Insulator  0  . 11137 11737  10,190,254
## 4 chr1 11737 11937       11_Weak_Txn  0  . 11737 11937 153,255,102
## 5 chr1 11937 12137   7_Weak_Enhancer  0  . 11937 12137   255,252,4
## 6 chr1 12137 14537       11_Weak_Txn  0  . 12137 14537 153,255,102
```

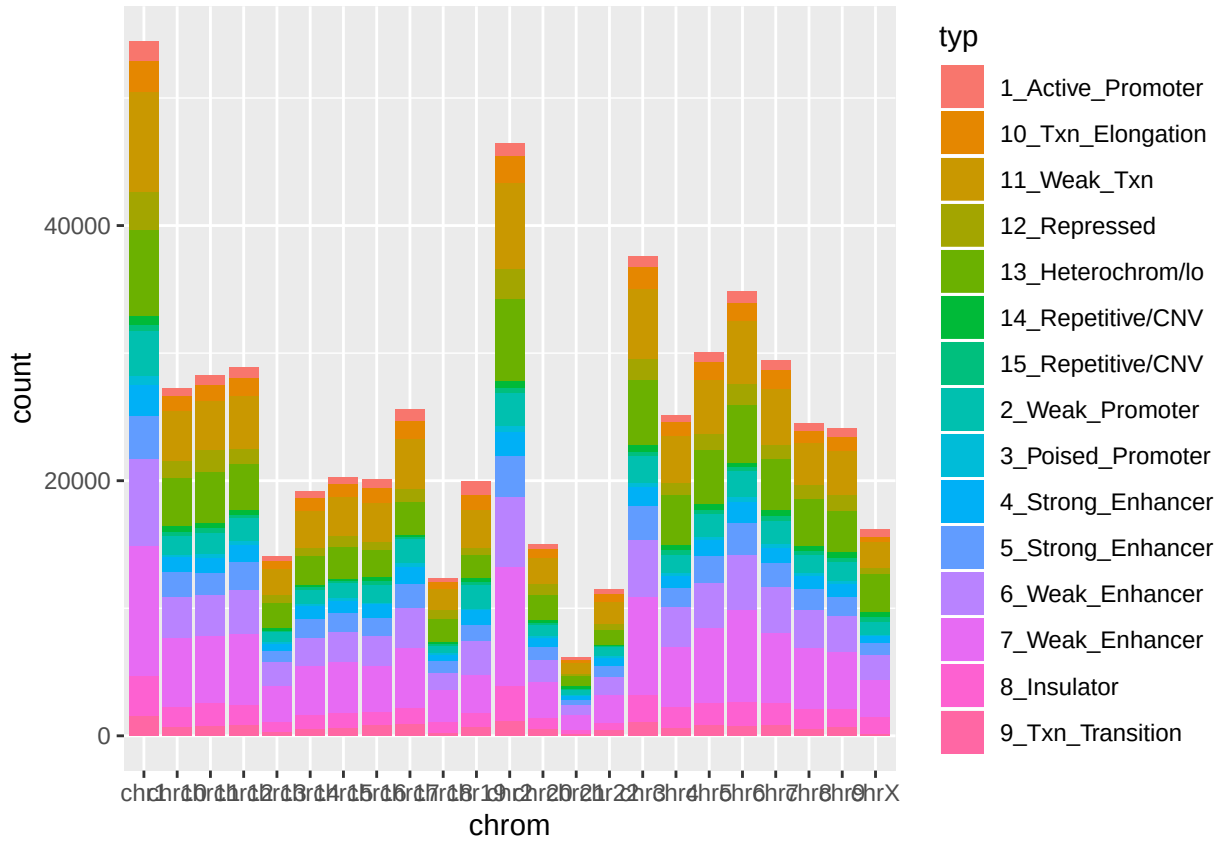
Zauważmy, że nasze dane w tabeli 'mydata' nie mają nazw nagłówków. Za pomocą funkcji `names` możemy nadać nazwy kolumną w tabeli.

```
names(mydata)[1:4] <- c('chrom', 'start', 'stop', 'typ')
head(mydata)
```

```
##   chrom start  stop               typ V5 V6    V7    V8          V9
## 1  chr1 10000 10600 15_Repetitive/CNV  0  . 10000 10600 245,245,245
## 2  chr1 10600 11137 13_Heterochrom/lo  0  . 10600 11137 245,245,245
## 3  chr1 11137 11737       8_Insulator  0  . 11137 11737  10,190,254
## 4  chr1 11737 11937       11_Weak_Txn  0  . 11737 11937 153,255,102
## 5  chr1 11937 12137   7_Weak_Enhancer  0  . 11937 12137   255,252,4
## 6  chr1 12137 14537       11_Weak_Txn  0  . 12137 14537 153,255,102
```

Do wizualizacja danych wykorzystamy bibliotekę `ggplot2`.

```
library(ggplot2)
ggplot(mydata, aes(x=chrom, fill=typ))+geom_bar()
```



Możemy też zapisać wyniki do bezpośrednio do pliku za pomocą funkcji `png`.

Możemy dowiedzieć się więcej o naszych danych korzystając z funkcji `summary` oraz `dim`.

```
dim(mydata)
```

```
## [1] 571339      9
```

```
summary(mydata)
```

```
##      chrom            start                stop          
##  chr1   : 54467   Min.   :        0   Min.   :    10200  
##  chr2   : 46499   1st Qu.: 33424623   1st Qu.: 33427336  
##  chr3   : 37617   Median : 66145965   Median : 66150096  
##  chr6   : 34846   Mean   : 77800396   Mean   : 77805350  
##  chr5   : 30071   3rd Qu.:114147254   3rd Qu.:114148704  
##  chr7   : 29420   Max.   :249229377   Max.   :249232977  
##  (Other):338419                                          
##                typ               V5    V6               V7          
##  7_Weak_Enhancer  :109468   Min.   :0   .:571339   Min.   :        0  
##  11_Weak_Txn      : 82312   1st Qu.:0              1st Qu.: 33424623  
##  13_Heterochrom/lo: 75112   Median :0              Median : 66145965  
##  6_Weak_Enhancer  : 69111   Mean   :0              Mean   : 77800396  
##  5_Strong_Enhancer: 38604   3rd Qu.:0              3rd Qu.:114147254  
##  2_Weak_Promoter  : 35065   Max.   :0              Max.   :249229377  
##  (Other)          :161667                                           
##        V8                      V9          
##  Min.   :    10200   255,252,4  :178579  
##  1st Qu.: 33427336   245,245,245: 89268
```

```
##  Median : 66150096   153,255,102: 82312
##  Mean   : 77805350   250,202,0  : 64090
##  3rd Qu.:114148704   0,176,80   : 42736
##  Max.   :249232977   255,105,105: 35065
##                      (Other)    : 79289
```

```
summary(mydata$chrom)
```

```
##  chr1 chr10 chr11 chr12 chr13 chr14 chr15 chr16 chr17 chr18 chr19  chr2
## 54467 27263 28246 28863 14064 19133 20277 20113 25570 12324 19947 46499
## chr20 chr21 chr22  chr3  chr4  chr5  chr6  chr7  chr8  chr9  chrX
## 15000  6128 11497 37617 25155 30071 34846 29420 24506 24123 16210
```

```
summary(mydata$start)
```

```
##      Min.   1st Qu.    Median      Mean   3rd Qu.      Max.
##         0  33424623  66145965  77800396 114147254 249229377
```

Dane w tabeli `mydata` możemy modyfikować. Możemy przykładowo stworzyć nową kolumnę zawierającą informacje o długości sekwencji (stop-start).

Ćwiczenie 1

Proszę utworzyć kolumnę `size` określającą długość każdej sekwencji z pliku `mydata`.

```
##   chrom start  stop               typ V5 V6    V7    V8          V9 size
## 1  chr1 10000 10600 15_Repetitive/CNV  0  . 10000 10600 245,245,245  600
## 2  chr1 10600 11137 13_Heterochrom/lo  0  . 10600 11137 245,245,245  537
## 3  chr1 11137 11737       8_Insulator  0  . 11137 11737  10,190,254  600
## 4  chr1 11737 11937       11_Weak_Txn  0  . 11737 11937 153,255,102  200
## 5  chr1 11937 12137   7_Weak_Enhancer  0  . 11937 12137   255,252,4  200
## 6  chr1 12137 14537       11_Weak_Txn  0  . 12137 14537 153,255,102 2400
```

Możemy poznać podstawowe statystyki takie jak średnia i odchylenie standardowe dla nowo utworzonej kolumny za pomocą funkcji `mean` oraz `sd`.

```
# średnia
mean(mydata$size)
```

```
## [1] 4954.585
```

```
# odchylenie standardowe
sd(mydata$size)
```

```
## [1] 21176.37
```

Przyjrzyjmy się ponownie naszemu wykresowi. Możemy zauważyć iż na osi X prefiks nie mieści się na skali, a kolejność chromosomów jest błędna.

Ćwiczenie 2

Proszę usunąć prefiks 'chr' z kolumny chr oraz zmienić typ danych na faktor.

```
summary(mydata$chrom)
```

```
##     1    10    11    12    13    14    15    16    17    18    19     2
## 54467 27263 28246 28863 14064 19133 20277 20113 25570 12324 19947 46499
##    20    21    22     3     4     5     6     7     8     9     X
## 15000  6128 11497 37617 25155 30071 34846 29420 24506 24123 16210
```

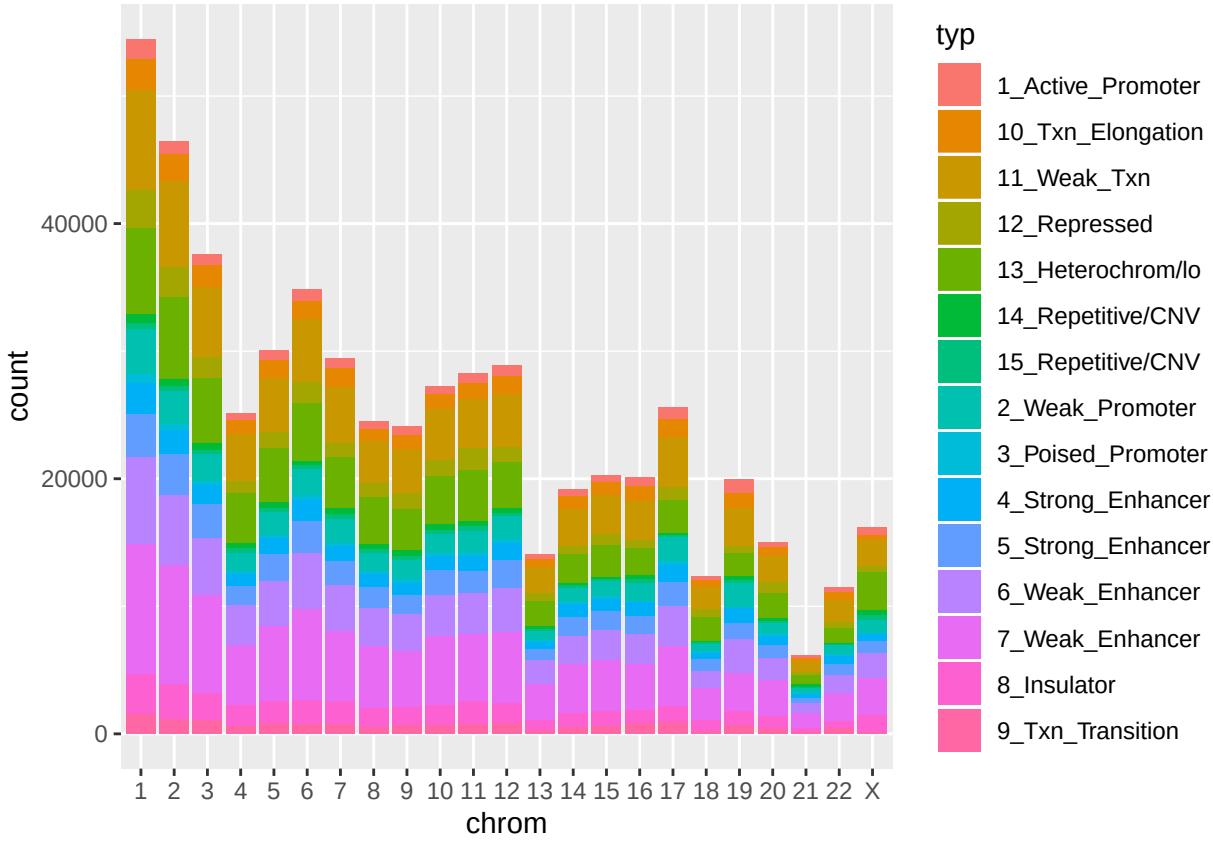
Następnie proszę 'ustawić' chromosomy w kolejności 1:22, X, Y

```
summary(mydata$chrom)
```

```
##     1     2     3     4     5     6     7     8     9    10    11    12
## 54467 46499 37617 25155 30071 34846 29420 24506 24123 27263 28246 28863
##    13    14    15    16    17    18    19    20    21    22     X     Y
## 14064 19133 20277 20113 25570 12324 19947 15000  6128 11497 16210     0
```

Po tych zmianach nasz wykres będzie wyglądał następująco:

```
ggplot(mydata, aes(x=chrom, fill=typ))+geom_bar()
```



Jeżeli chcemy ograniczyć liczbę elementów genetycznych na naszym wykresie do trzech typów: "1_Active_Promoter", "4_Strong_Enhancer", "8_Insulator" możemy przefiltrować nasze dane.

Ćwiczenie 3

Przefiltrować wyniki w tabeli `mydata`, tak by nowa tabela zwierała wyniki tylko dla 3 kategorii: - Active_Promoter - Strong_Enhancer -Insulato

```
dim(mydata)
```

```
## [1] 74029    10
```

Możemy zastosować funkcję

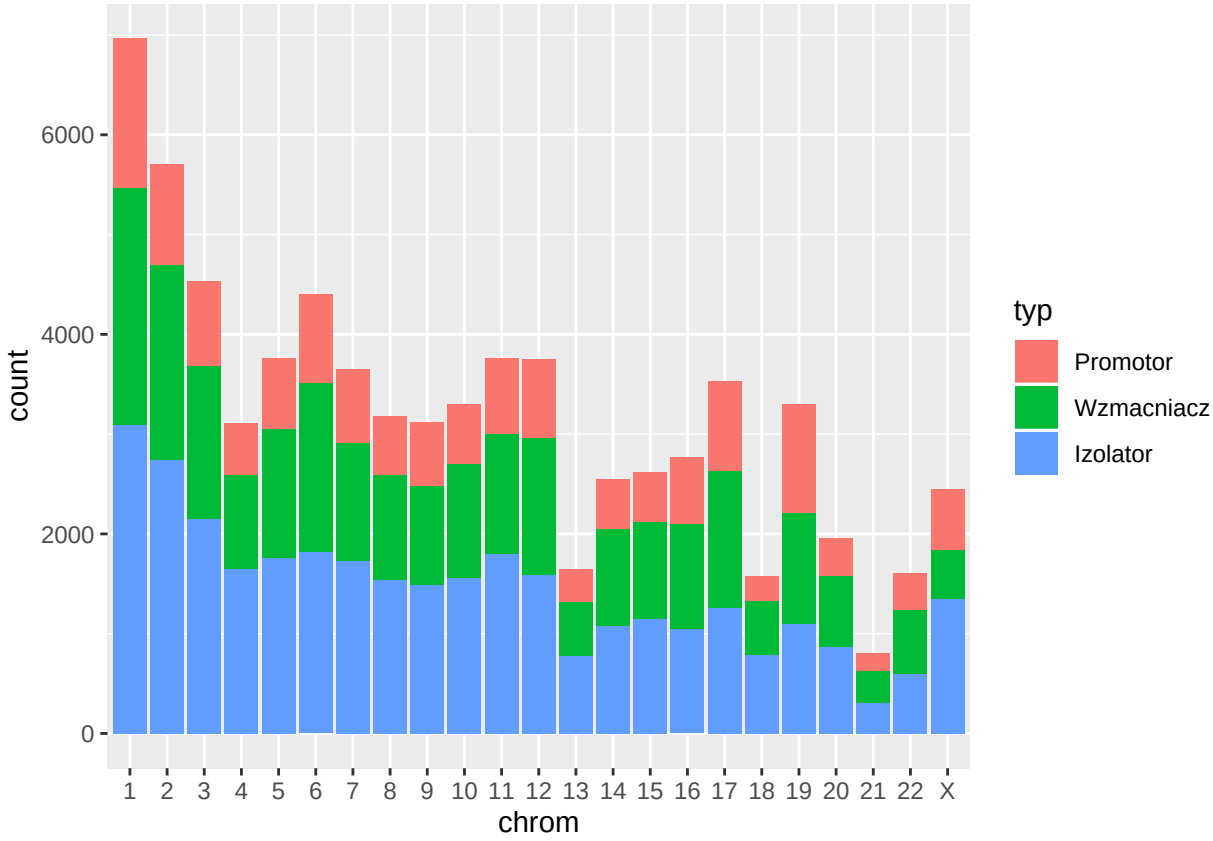
```
library(plyr) # pakiet zawiera bardzo przydatną funkcję revalue()
mydata$typ <- revalue(mydata$typ, c("1_Active_Promoter"="Promotor", "4_Strong_Enhancer"="Wzmacniacz","8_
summary(mydata$typ)
```

```
##          Promotor 10_Txn_Elongation       11_Weak_Txn      12_Repressed
##             15278                 0                 0                 0
```

```
## 13_Heterochrom/lo 14_Repetitive/CNV 15_Repetitive/CNV   2_Weak_Promoter
##                 0                 0                 0                 0
## 3_Poised_Promoter        Wzmacniacz 5_Strong_Enhancer   6_Weak_Enhancer
##                 0             25486                 0                 0
##   7_Weak_Enhancer          Izolator  9_Txn_Transition
##                 0             33265                 0
```

Możemy wykonać wykres ponownie

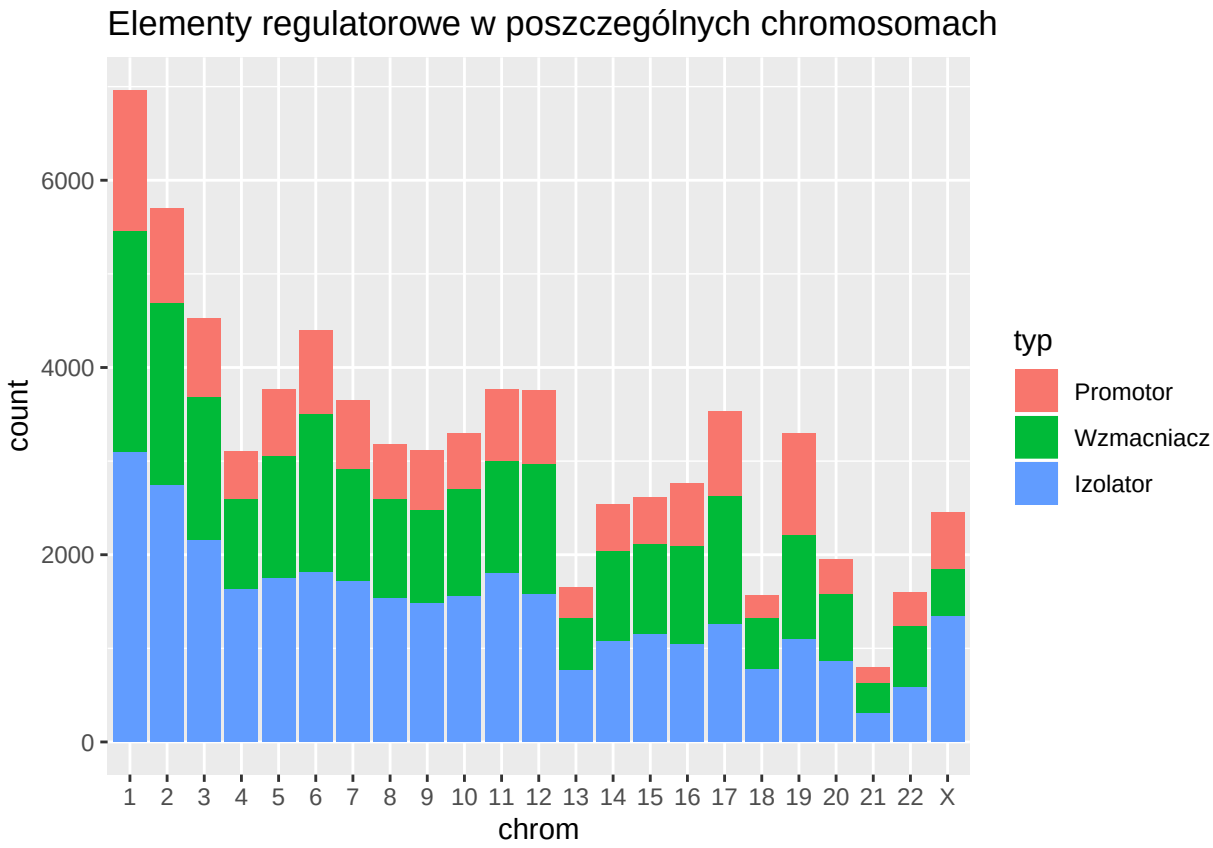
```
ggplot(mydata, aes(x=chrom, fill=typ))+geom_bar()
```



Modyfikacja wykresu ggplot

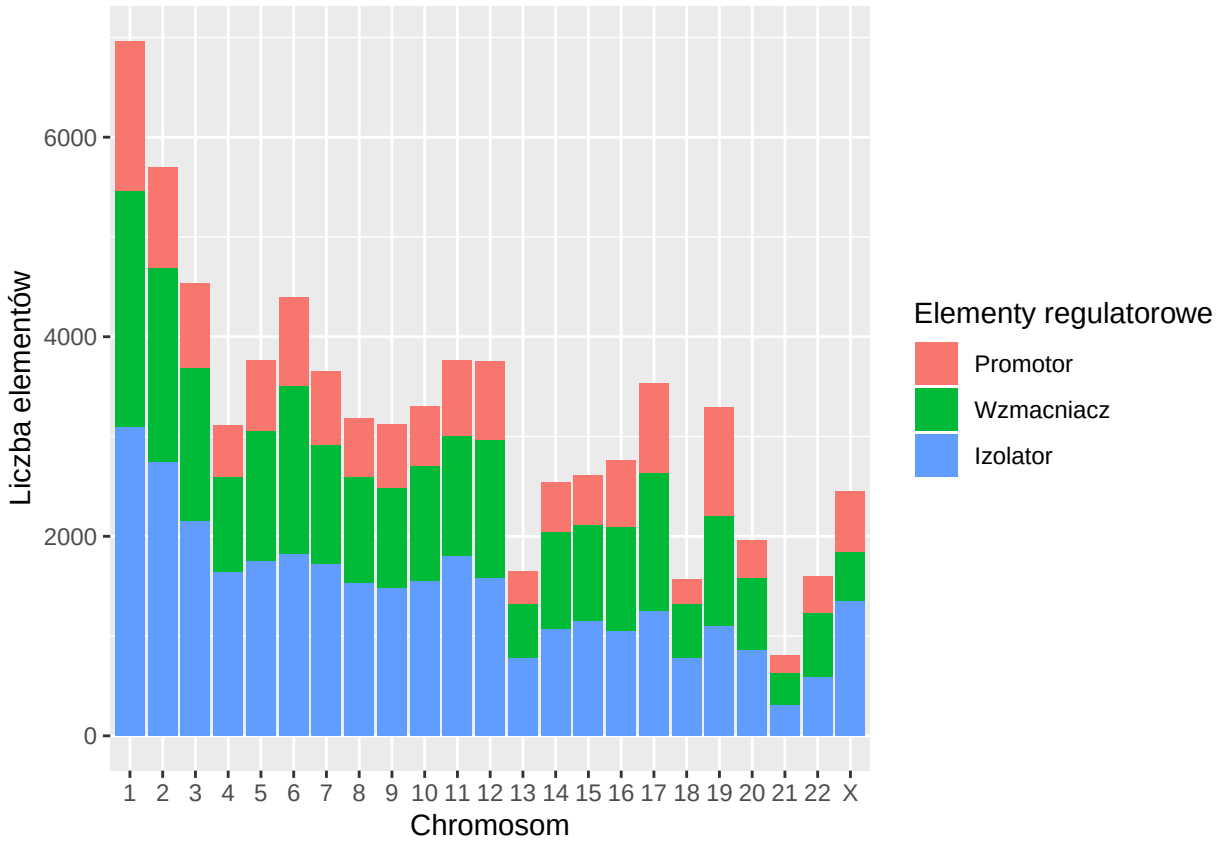
Tytuł

```
ggplot(mydata,aes(x=chrom,fill=typ)) + geom_bar() + labs(title="Elementy regulatorowe w poszczególnych
```



Podpisy osi

```
ggplot(mydata,aes(x=chrom,fill=typ)) + geom_bar() + labs(x = "Chromosom",y="Liczba elementów",fill="Ele
```



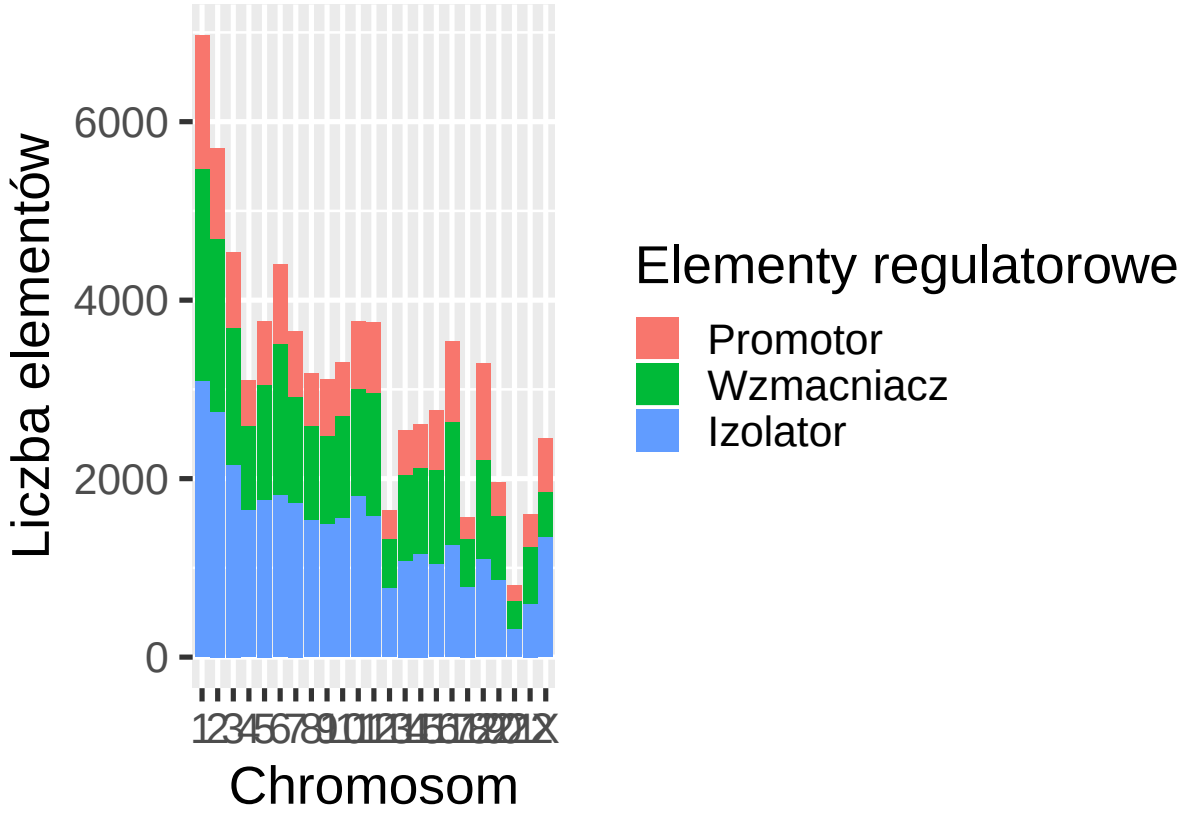
Modyfikacje

Możemy utworzyć obiekt R z naszym rysunkiem, aby przy dalszych modyfikacjach nie trzaba było powtarzać kodu.

```
podstawa <- ggplot(mydata,aes(x=chrom,fill=typ)) + geom_bar() + labs(x = "Chromosom",y="Liczba elementó
```

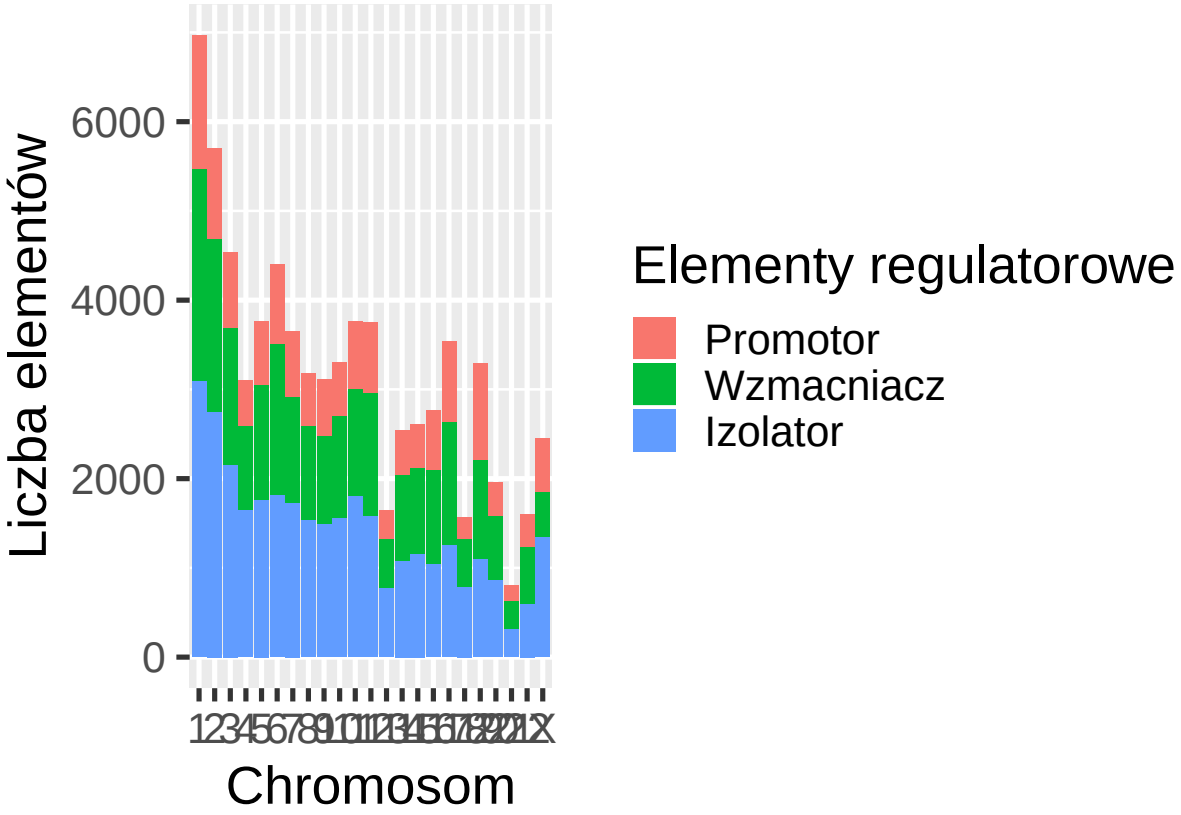
Modyfikacje wykresu

```
podstawa + theme_gray(base_size = 20)
```



Możemy też zmieniać domyślne ustawienia (zmiany będą dotyczyły wszystkich generowanych wykresów w sesji)

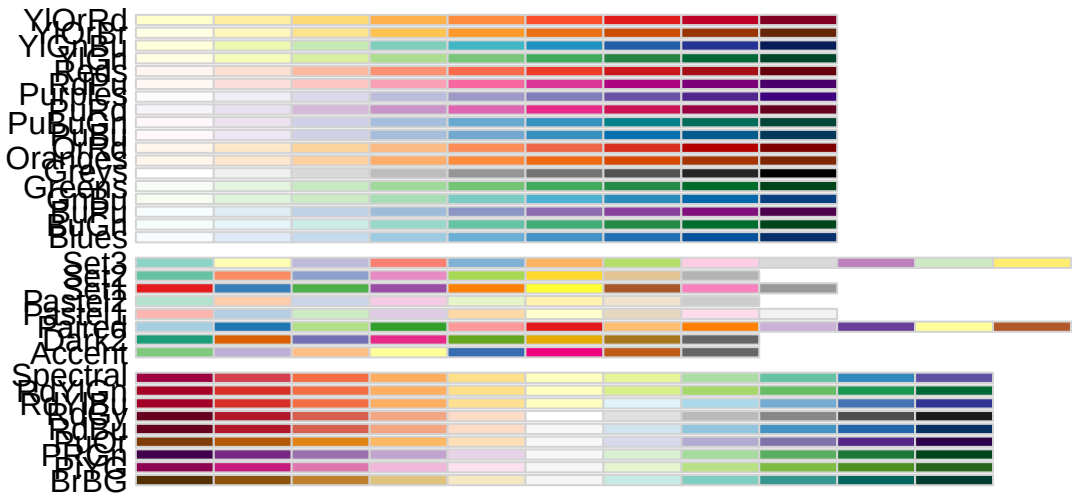
```
theme_set(theme_gray(base_size = 20))
podstawa
```



Schematy kolorów

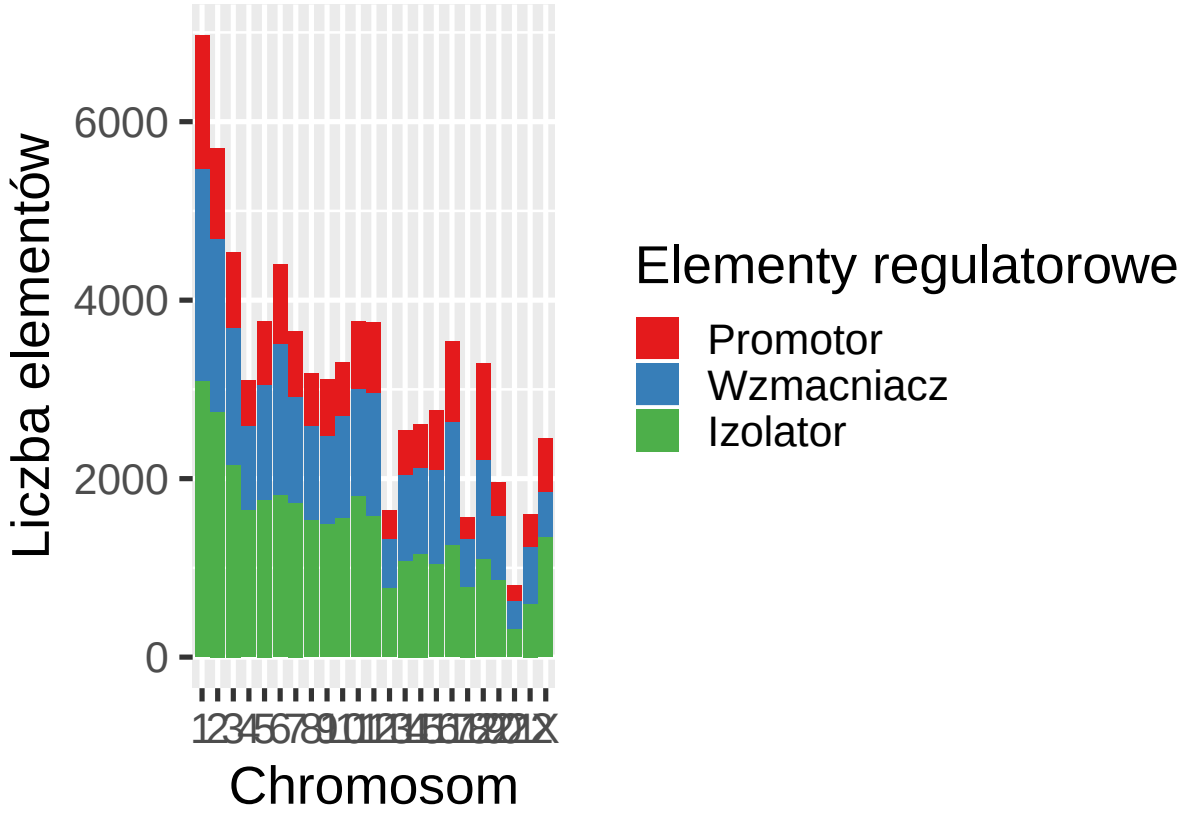
Biblioteka “ pozwala na użycie niestandarowych zestawów kolorów. Jest ona szczególnie użyteczna przy tworzeniu wykresów i heetmap.

```
library(RColorBrewer)
display.brewer.all()
```

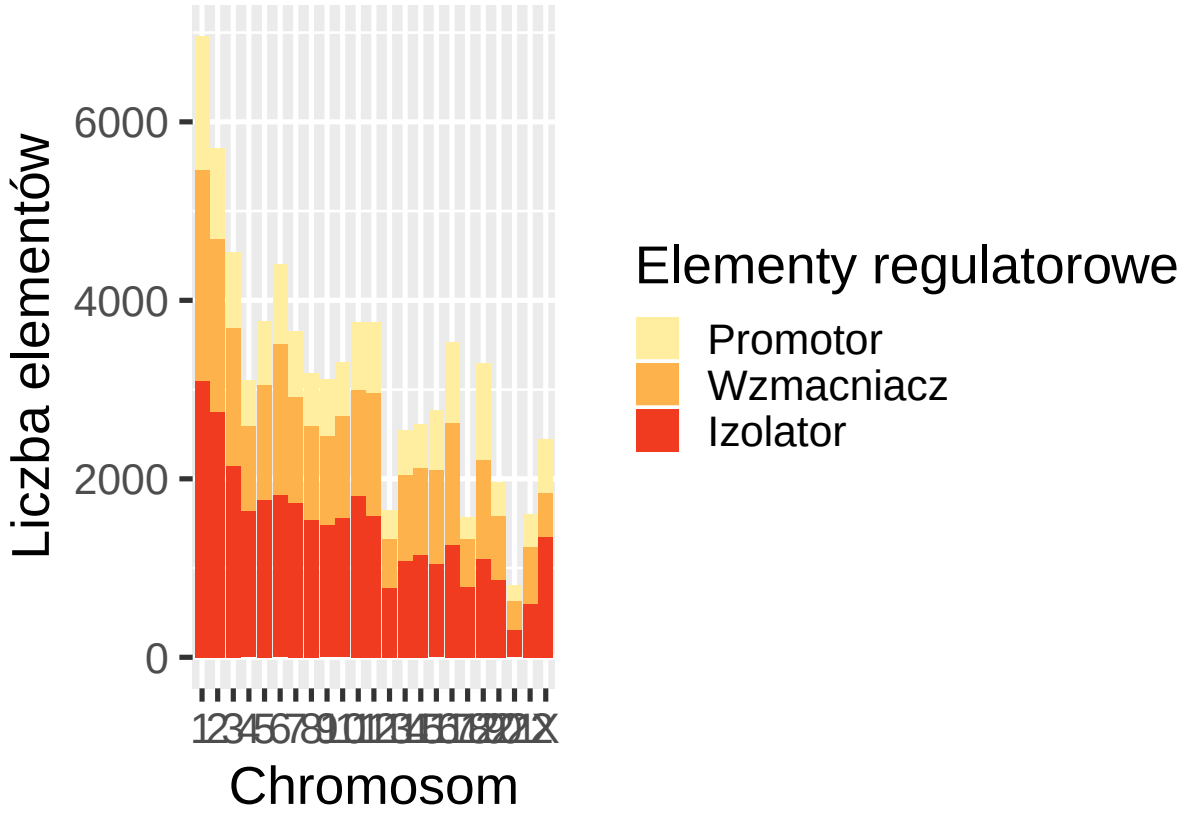


Zastosowanie do naszego przykładu:

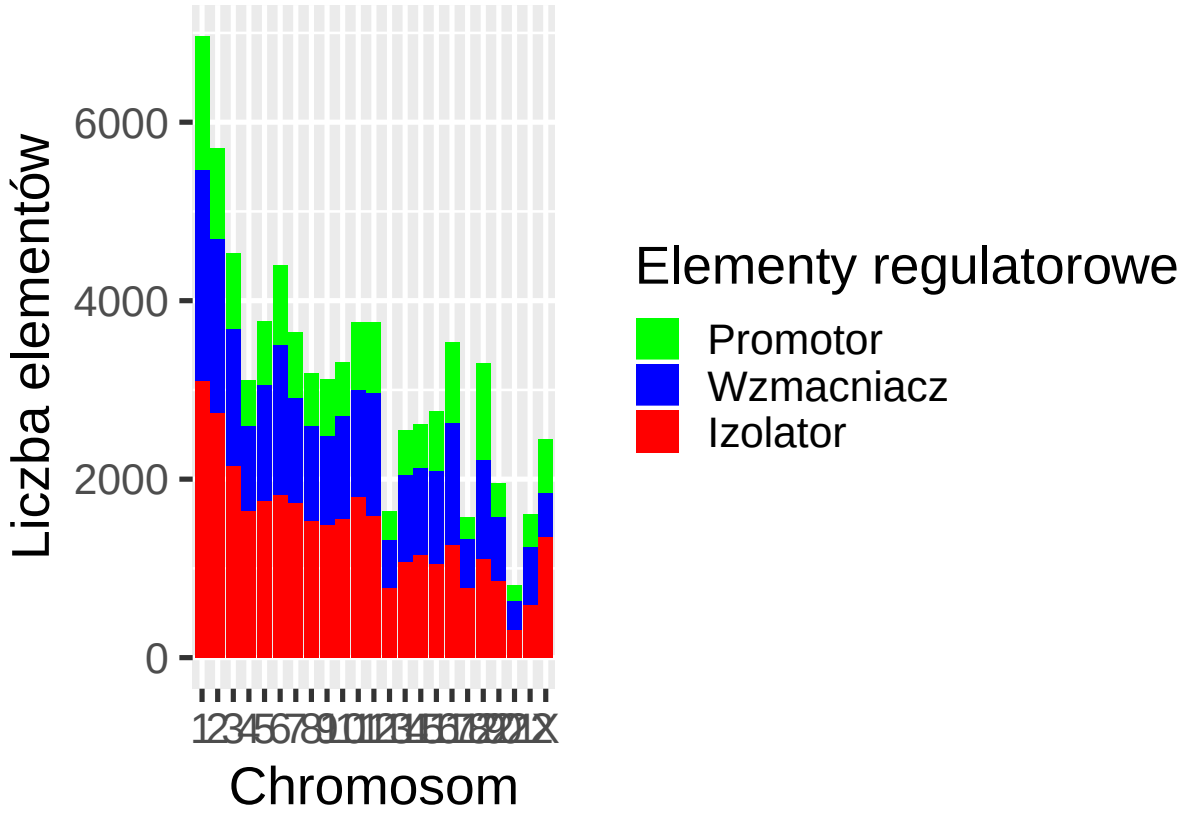
```
podstawa + scale_fill_brewer(palette="Set1")
```




```
podstawa + scale_fill_brewer(palette="YlOrRd")
```

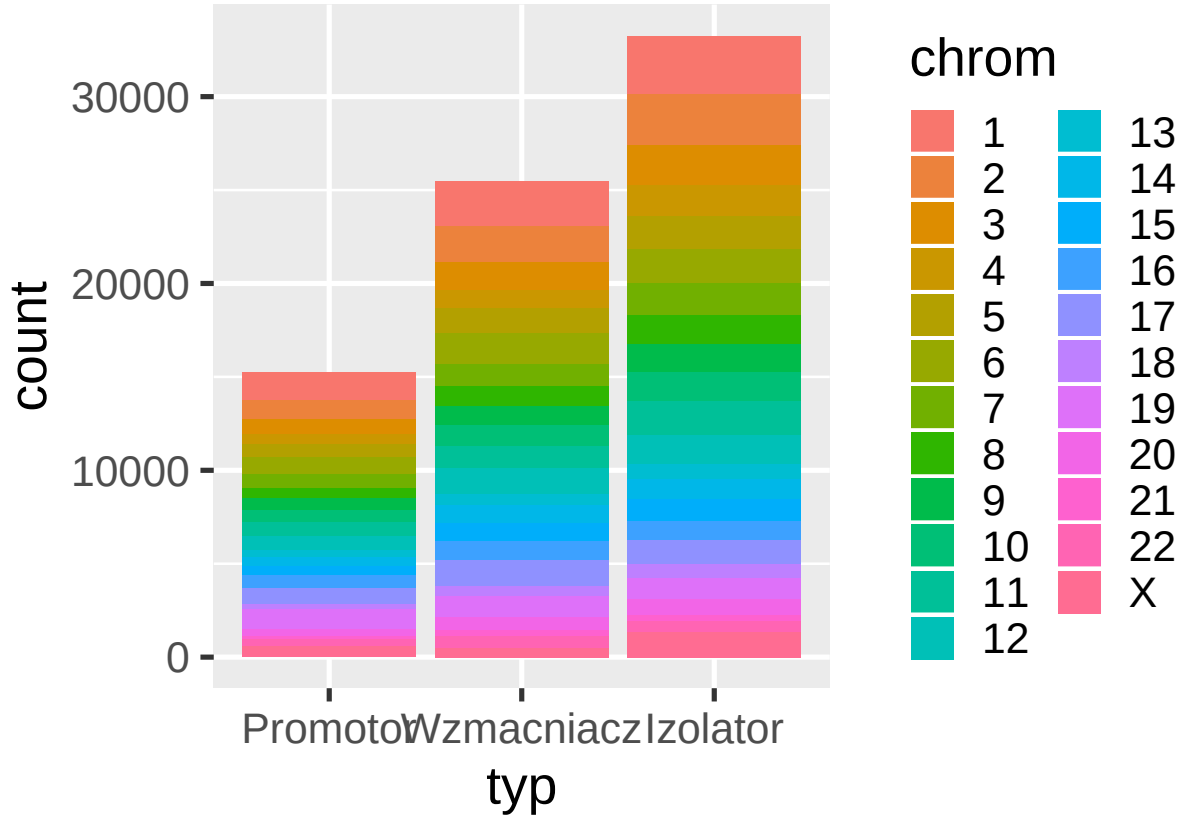


```
podstawa + scale_fill_manual(values = c("green","blue","red"))
```



Możemy też prezenotwać nasze dane w innej formie

```
chrom_plot <- ggplot(mydata,aes(x=typ,fill=chrom)) + geom_bar()
chrom_plot
```



Aby uzyskać wektor kolorów z palety możemy użyć polecenia:

```
palette1 <- brewer.pal(9,"Set1")
palette1
```

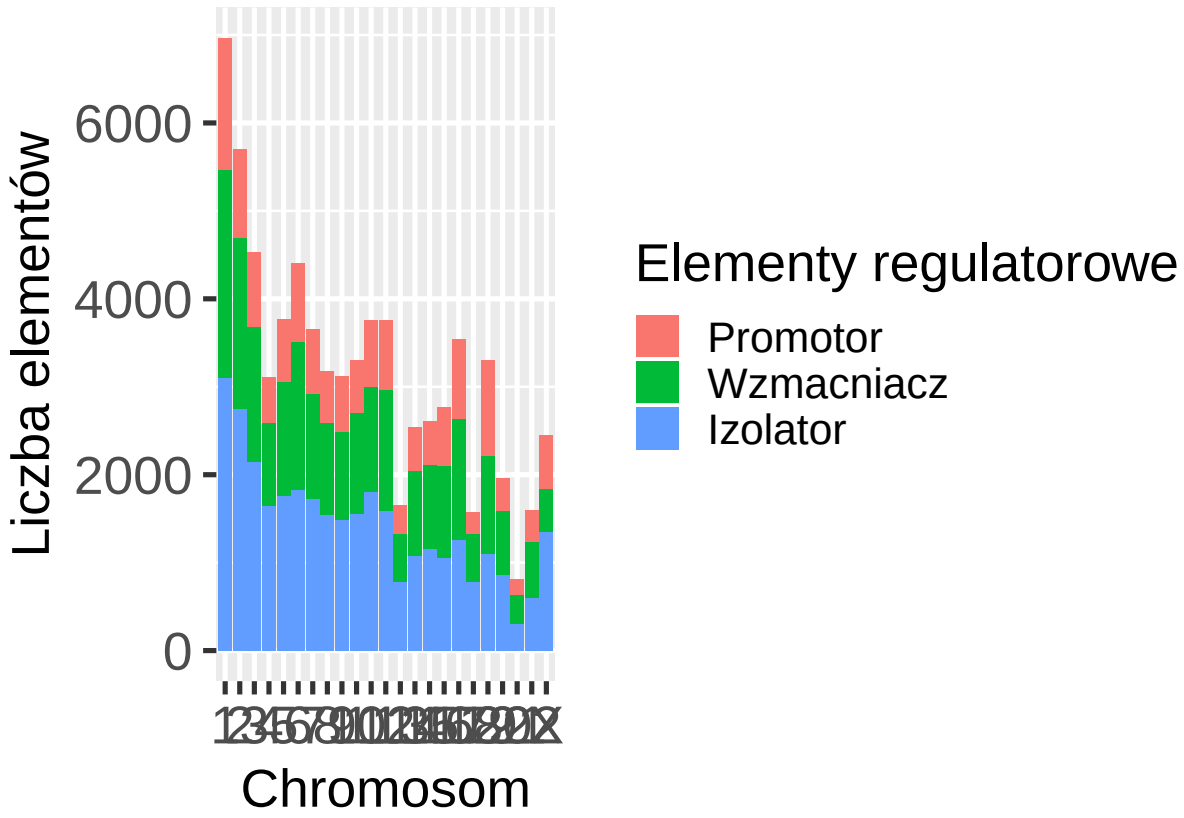
```
## [1] "#E41A1C" "#377EB8" "#4DAF4A" "#984EA3" "#FF7F00" "#FFFF33" "#A65628"
## [8] "#F781BF" "#999999"
```

Zmiana czciąki

```
#podstawa + theme_gray(base_size = 24, base_family = "Times New Roman")
```

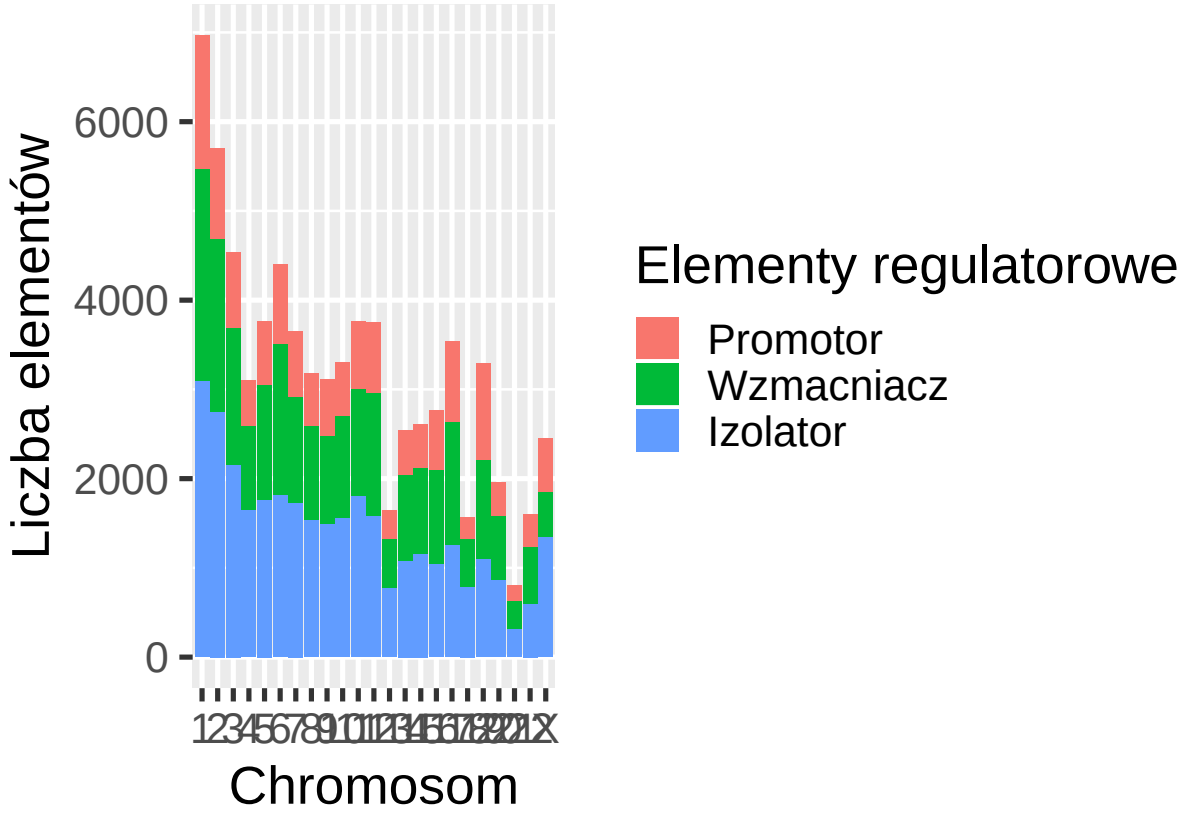
Zmiana rozmiaru czcionki na osi

```
podstawa + theme(axis.text=element_text(size=20))
```



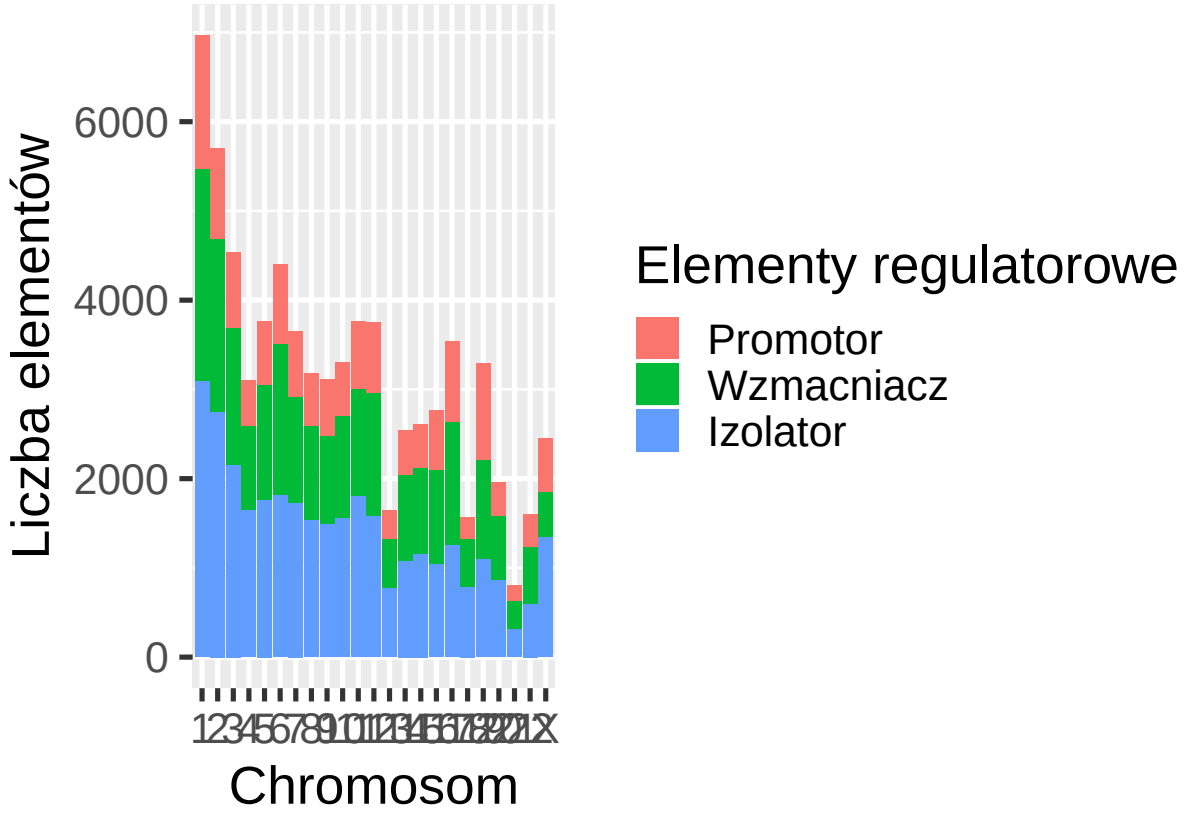
Zmiana rozmiaru czcionki podpisu osi

```
podstawa + theme(axis.title=element_text(size=20))
```



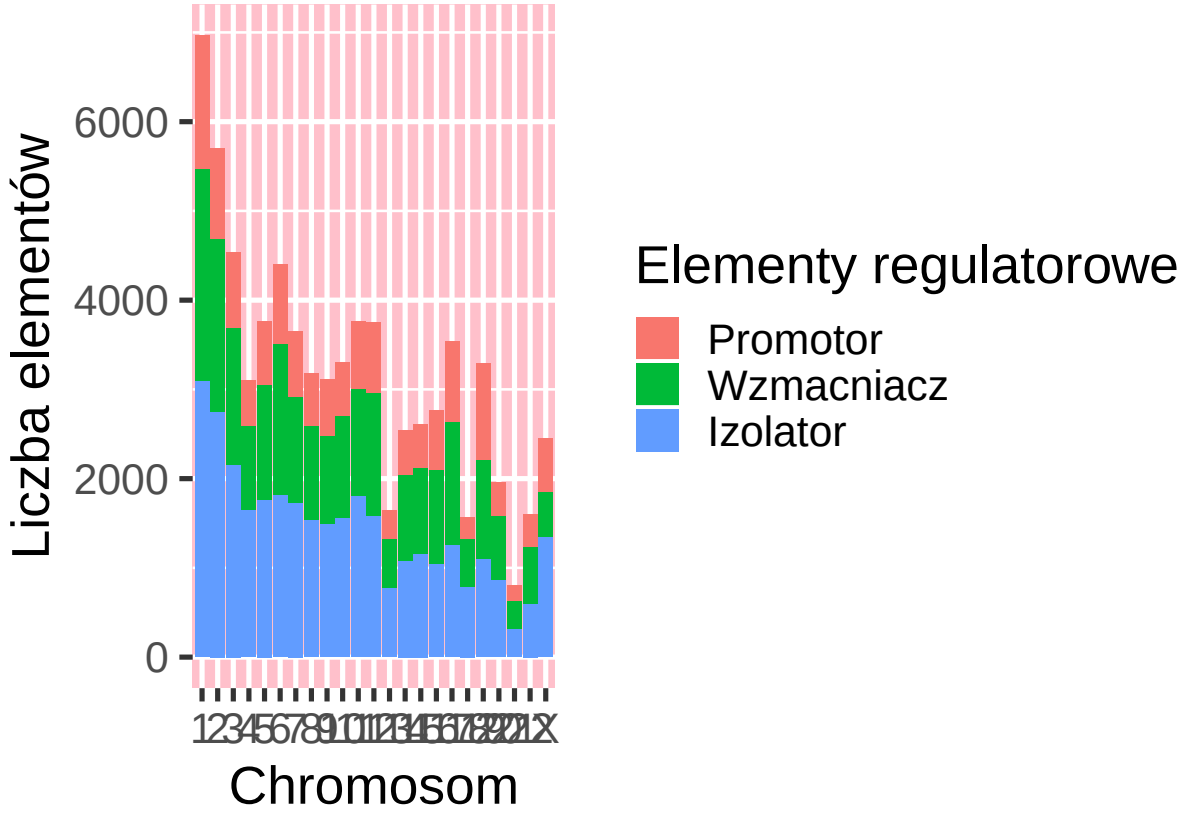
Zmiana rozmiaru czcionki tytułu legendy

```
podstawa + theme(legend.title=element_text(size=20))
```

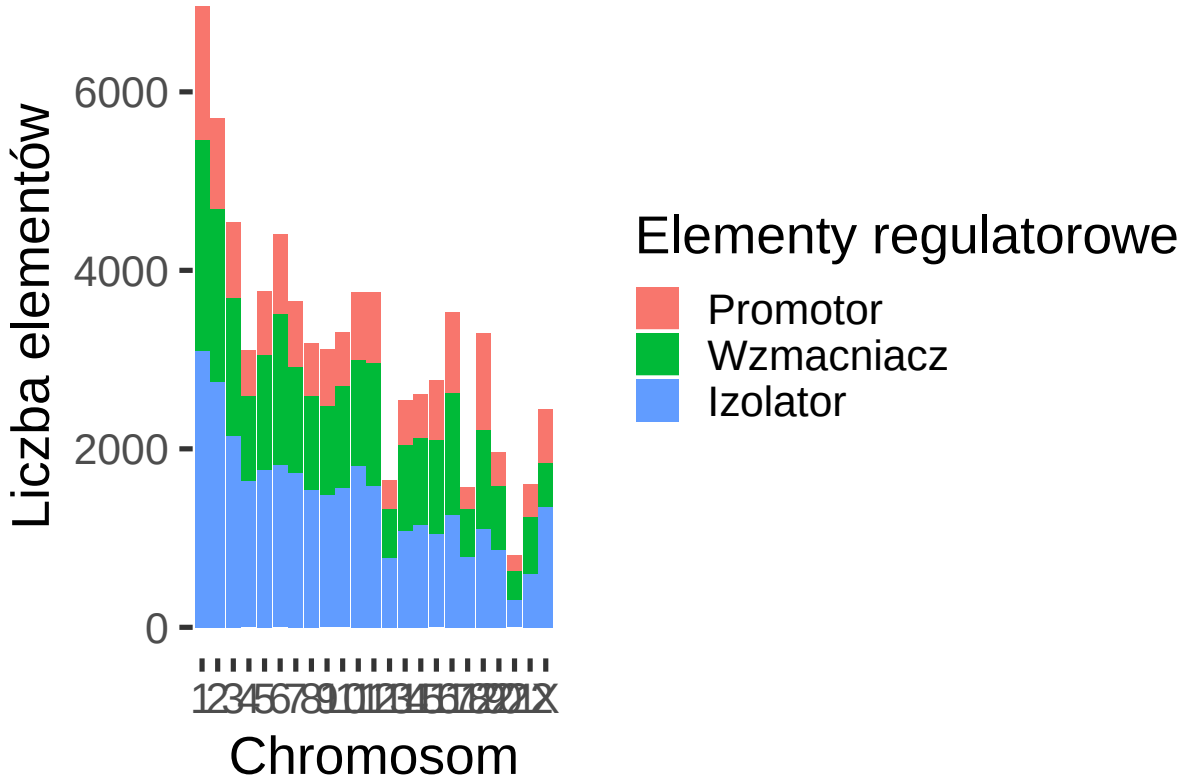


Zmiana koloru tła

```
podstawa + theme(panel.background = element_rect(fill="pink"))
```

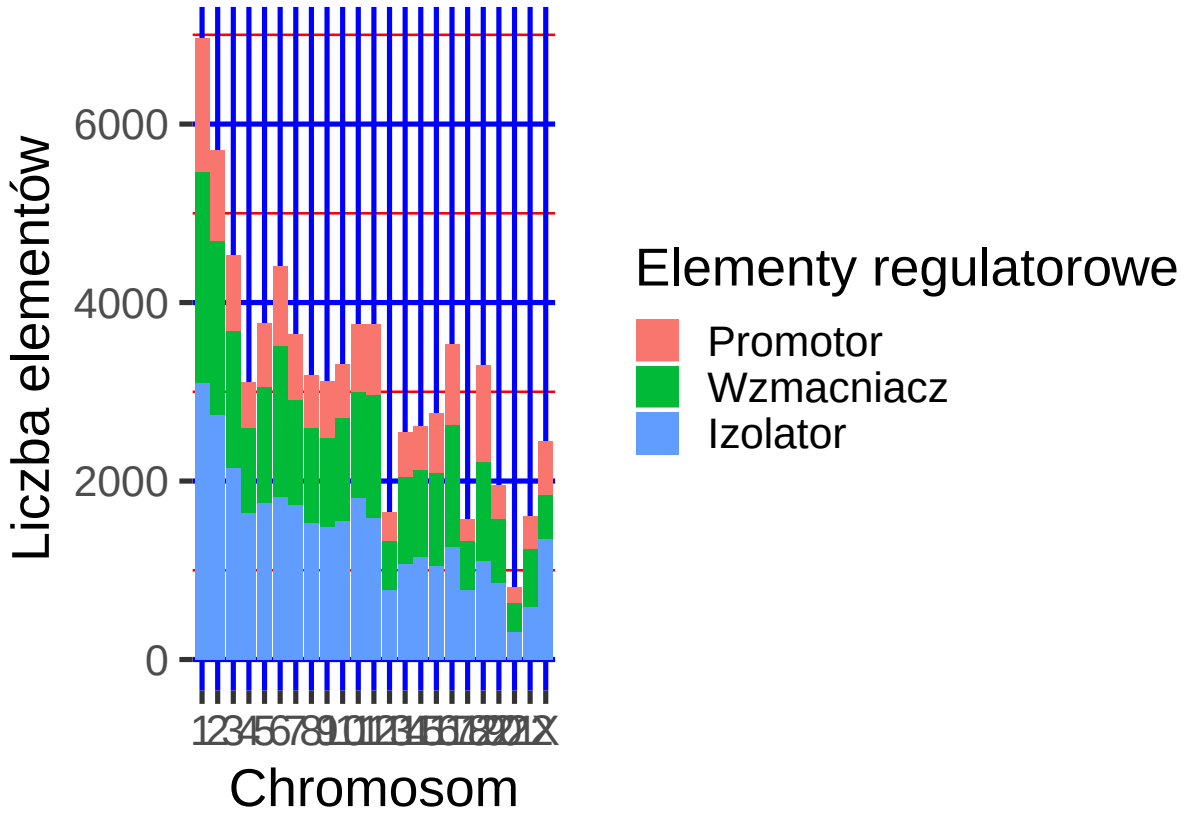



```
podstawa + theme(panel.background = element_rect(fill="white"))
```



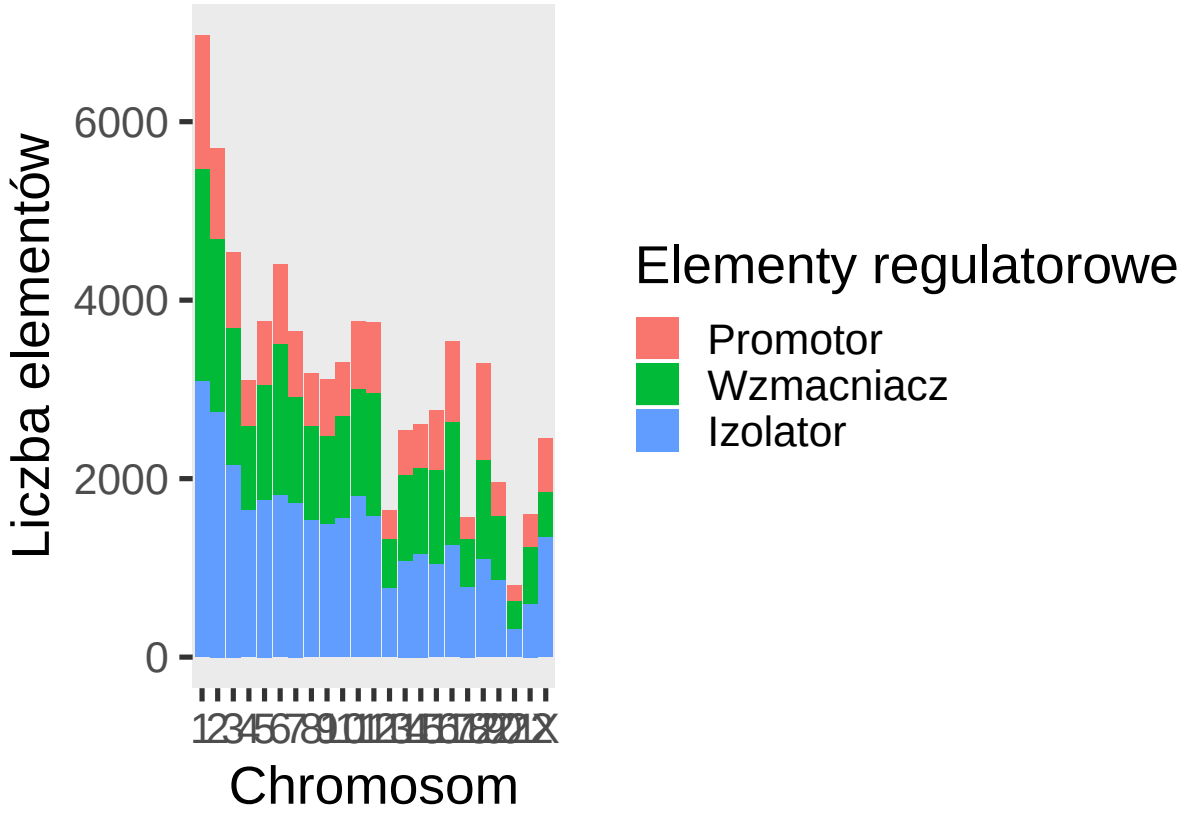
Zmiana linii siatki

```
podstawa + theme(panel.background = element_rect(fill="white"), panel.grid.major = element_line(colour =
```

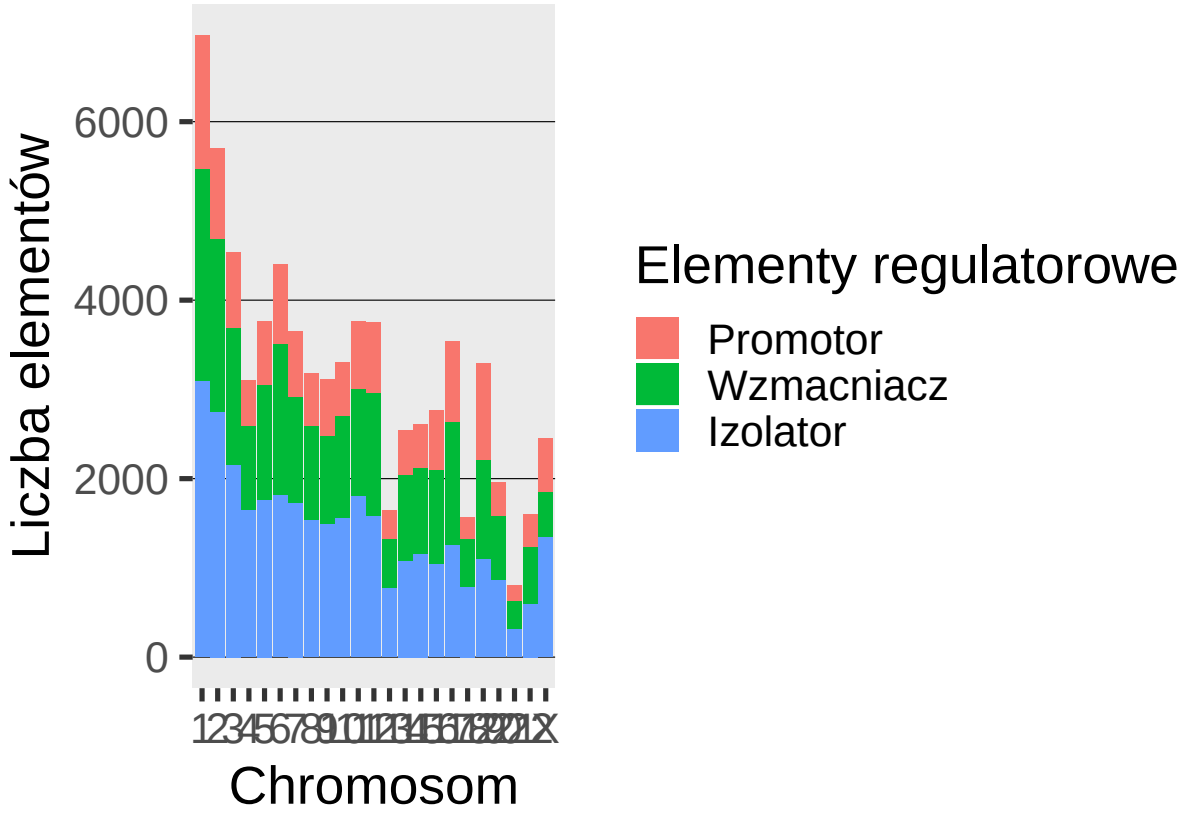


Usunięcie siatki

```
podstawa + theme(panel.grid.major = element_line(NA),
                 panel.grid.minor = element_line(NA))
```

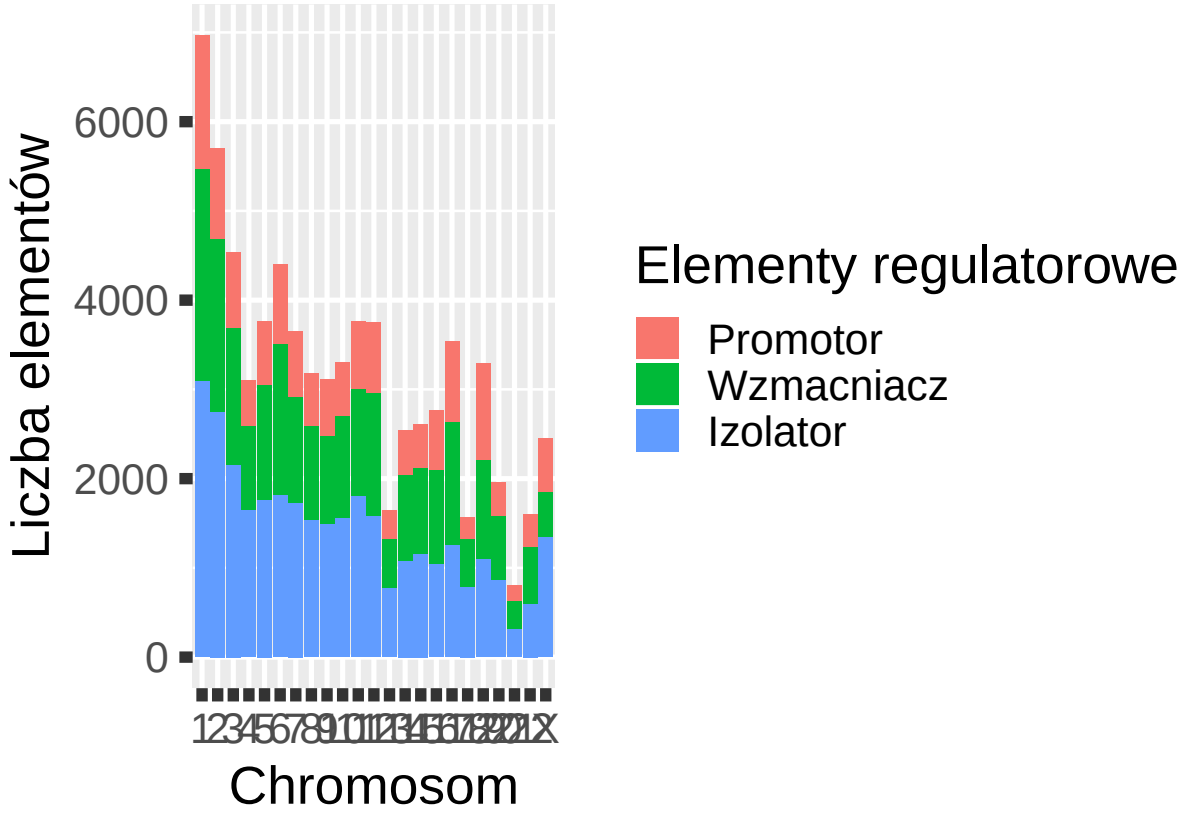



```
podstawa + theme(panel.grid.major.y = element_line(colour = "black",size=0.2),
                 panel.grid.major.x = element_line(NA),
                 panel.grid.minor = element_line(NA))
```

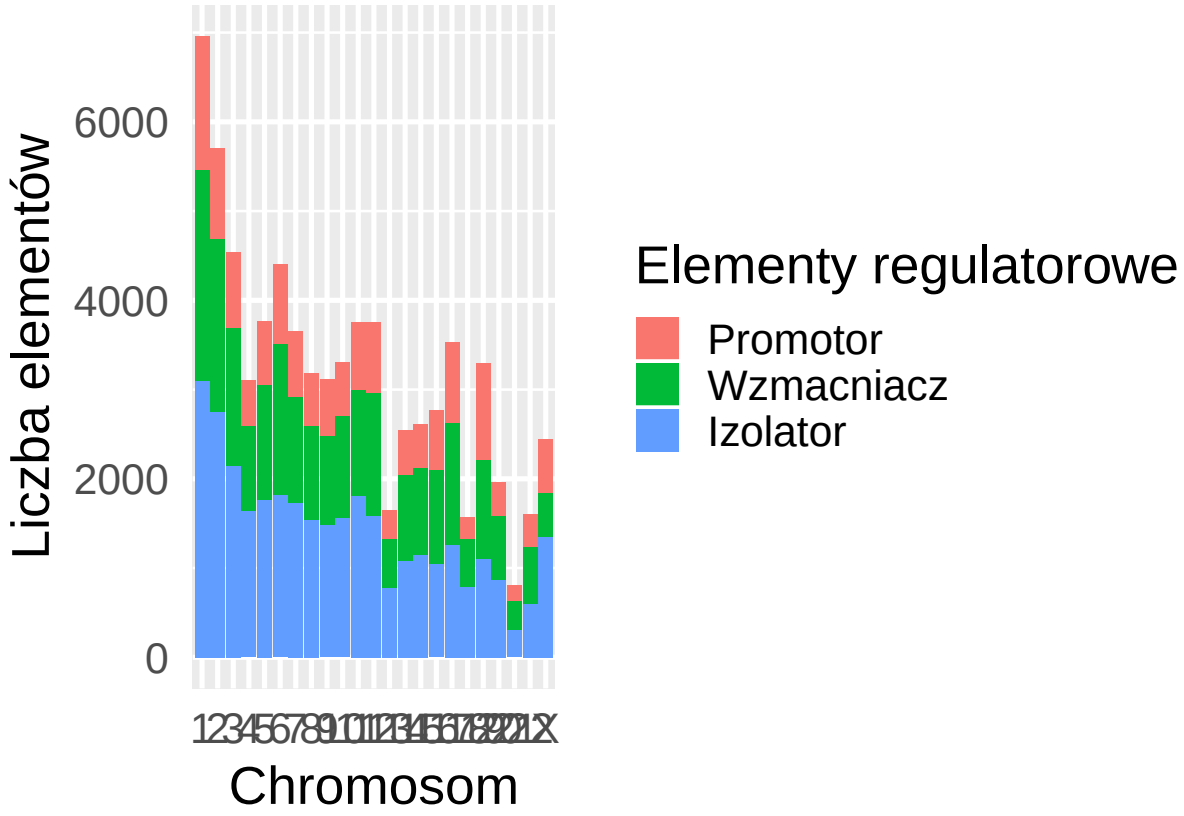


Zmiana znaczników na osiach

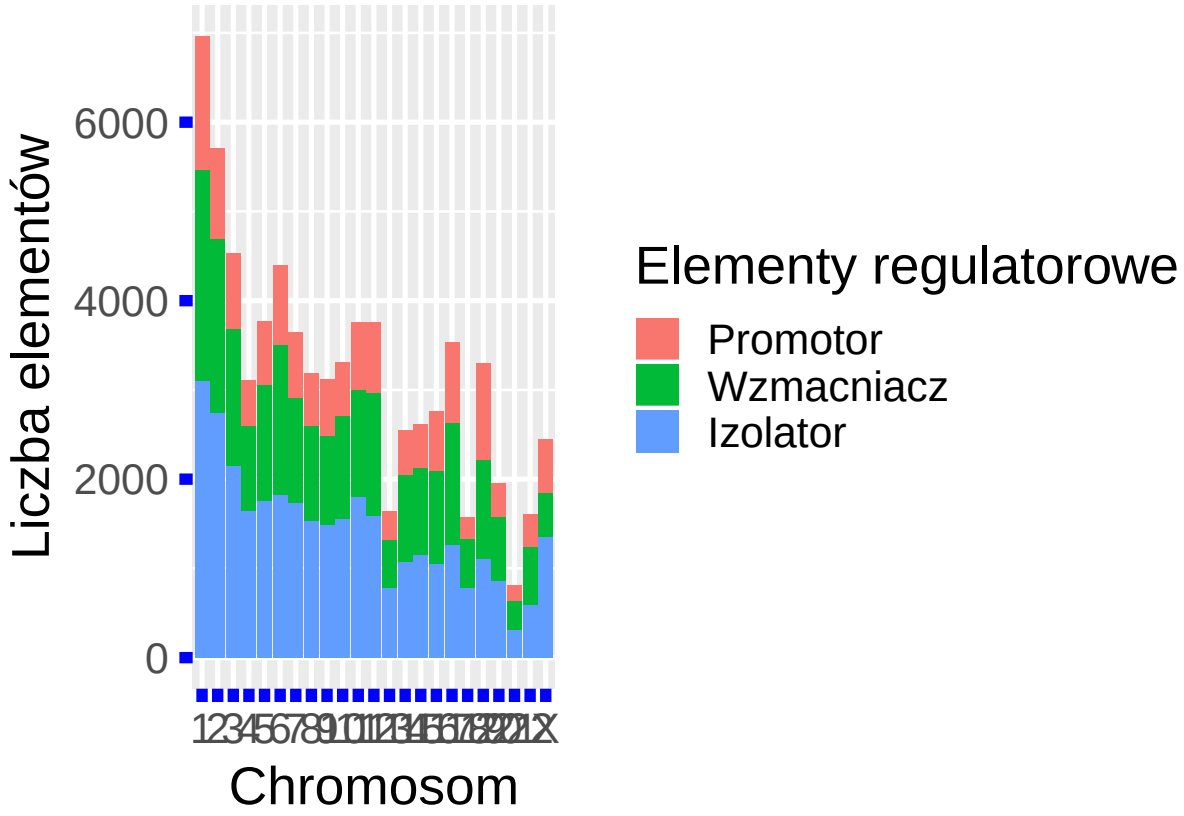
```
podstawa + theme(axis.ticks = element_line(size=2))
```



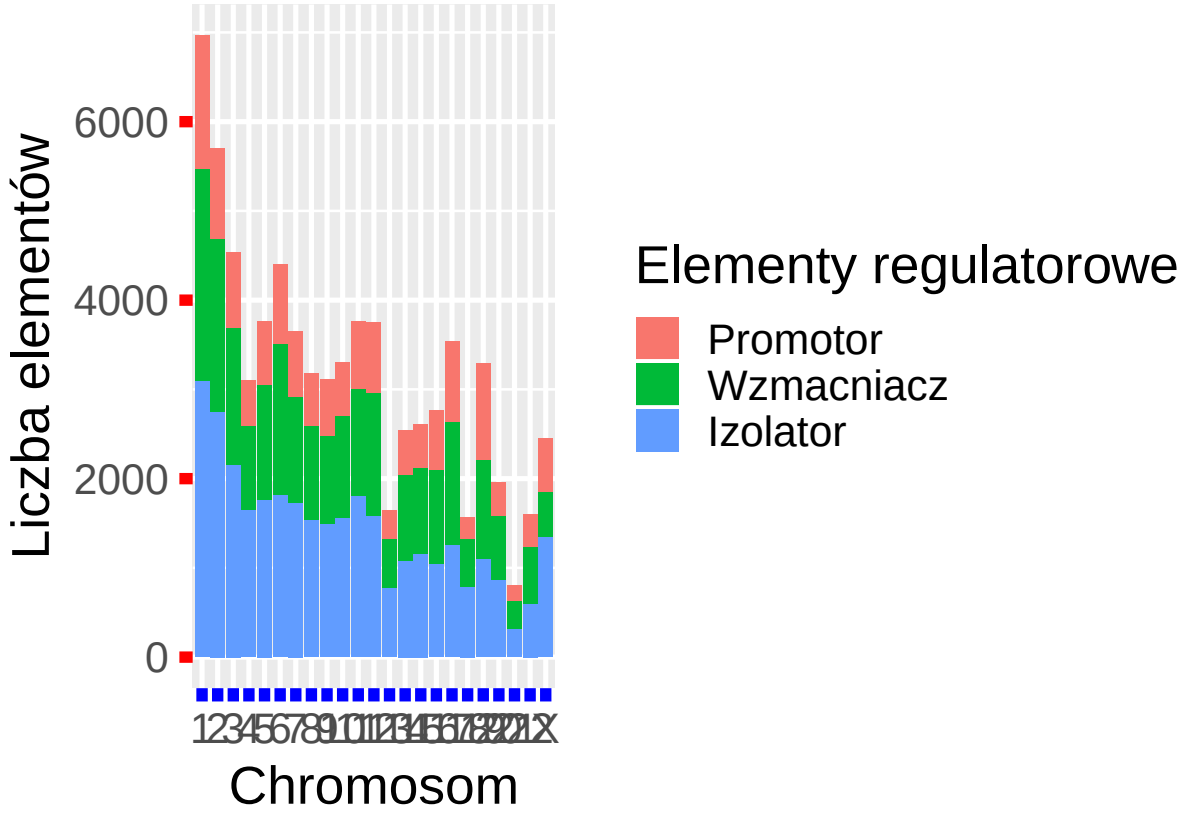

```
podstawa + theme(axis.ticks = element_line(NA))
```



```
podstawa + theme(axis.ticks = element_line(color="blue",size=2))
```

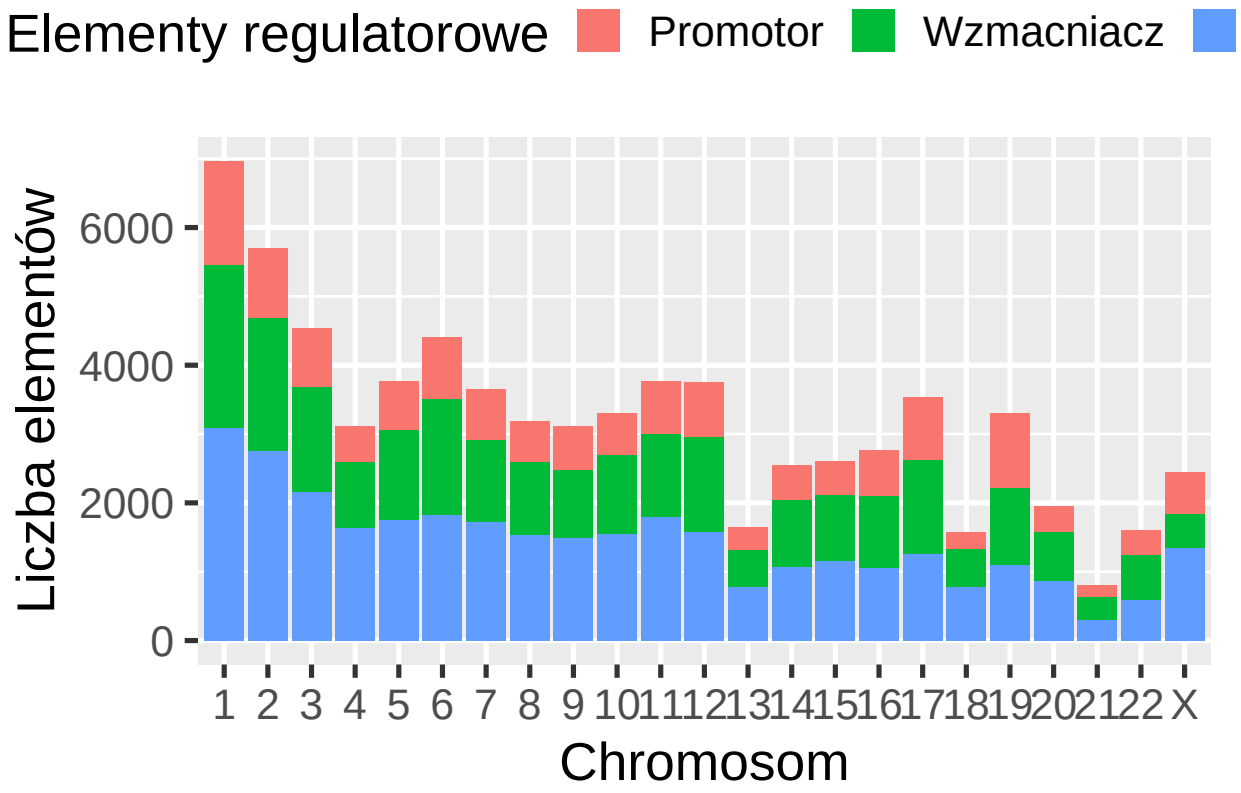



```
podstawa + theme(axis.ticks = element_line(size=2), # dotyczy x i y
                 axis.ticks.x = element_line(color="blue"), # tylko x
                 axis.ticks.y = element_line(color="red"))  # tylkko y
```

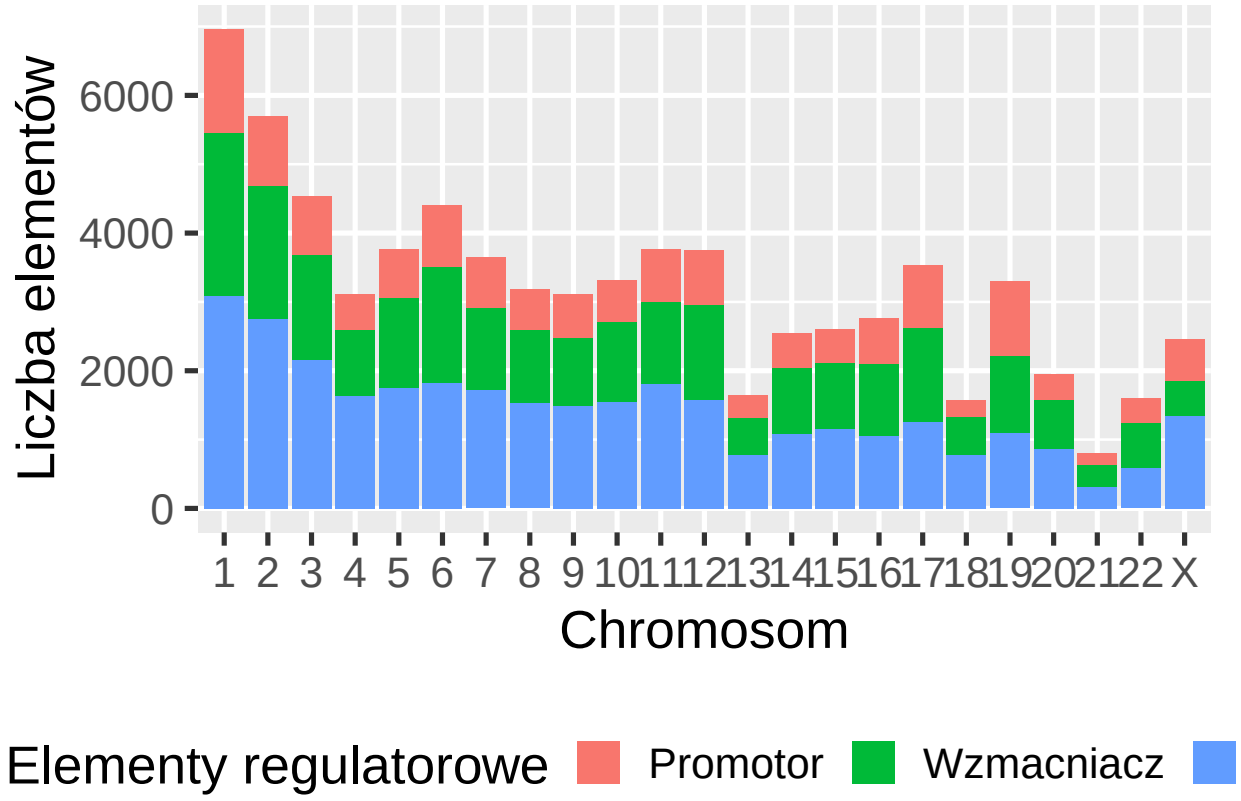


Zmiana położenia legendy

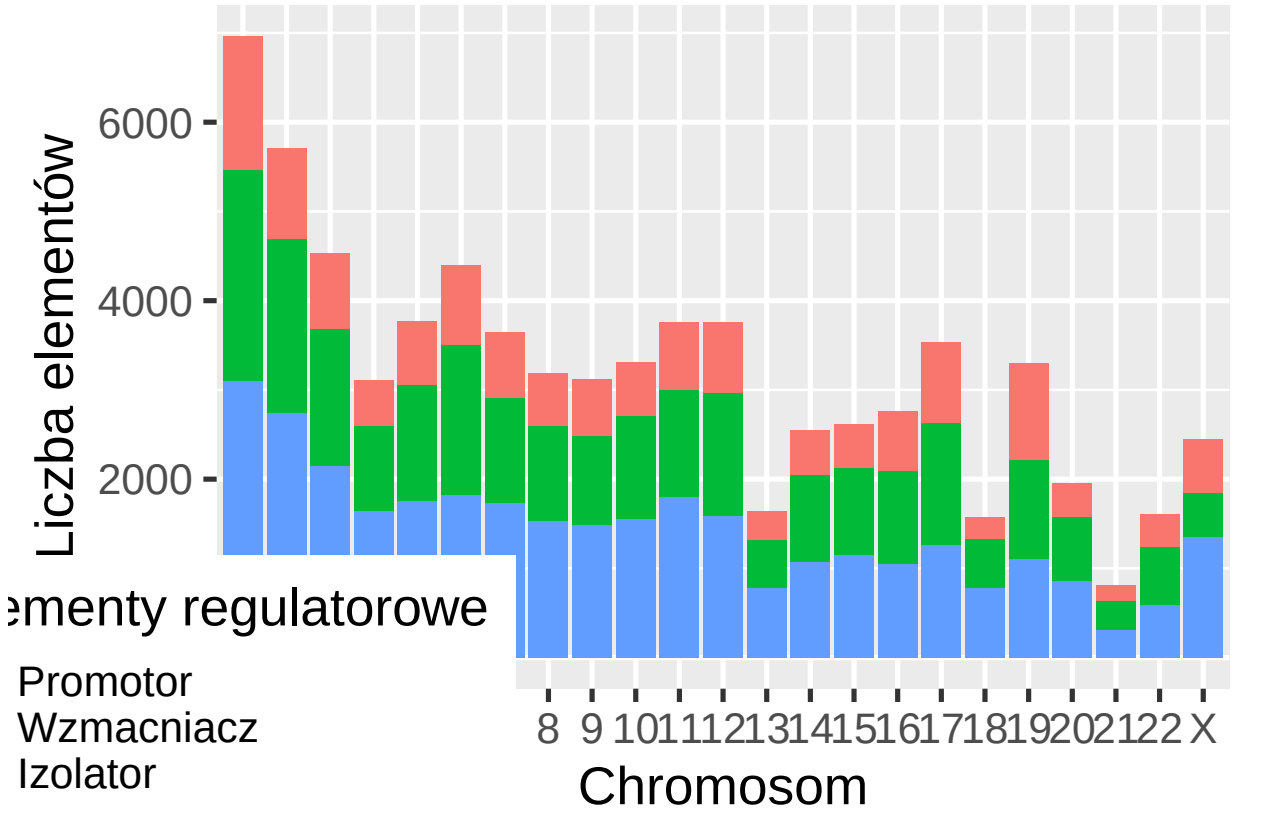
```
podstawa +  theme(legend.position="top")
```



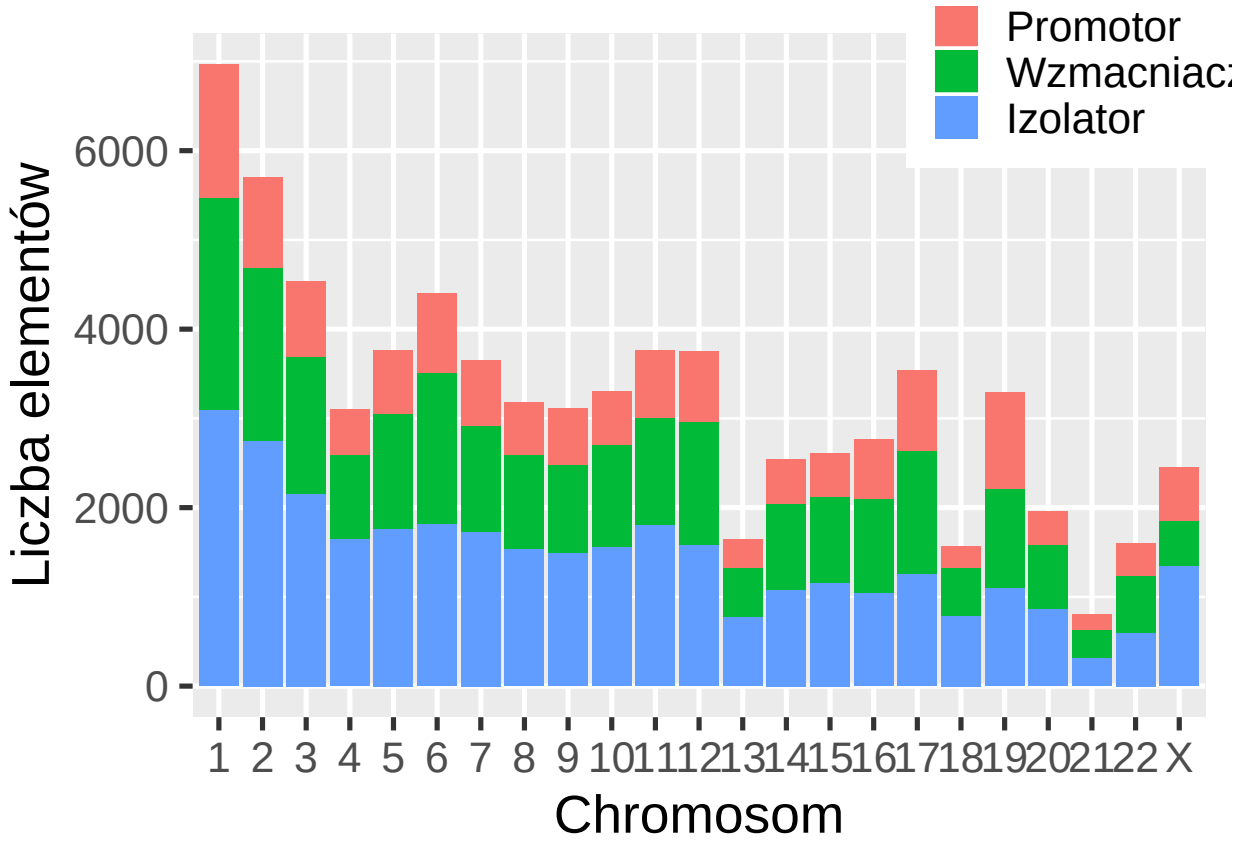

```
podstawa +  theme(legend.position="bottom")
```



```
podstawa +  theme(legend.position=c(0,0)) # lewy dół
```

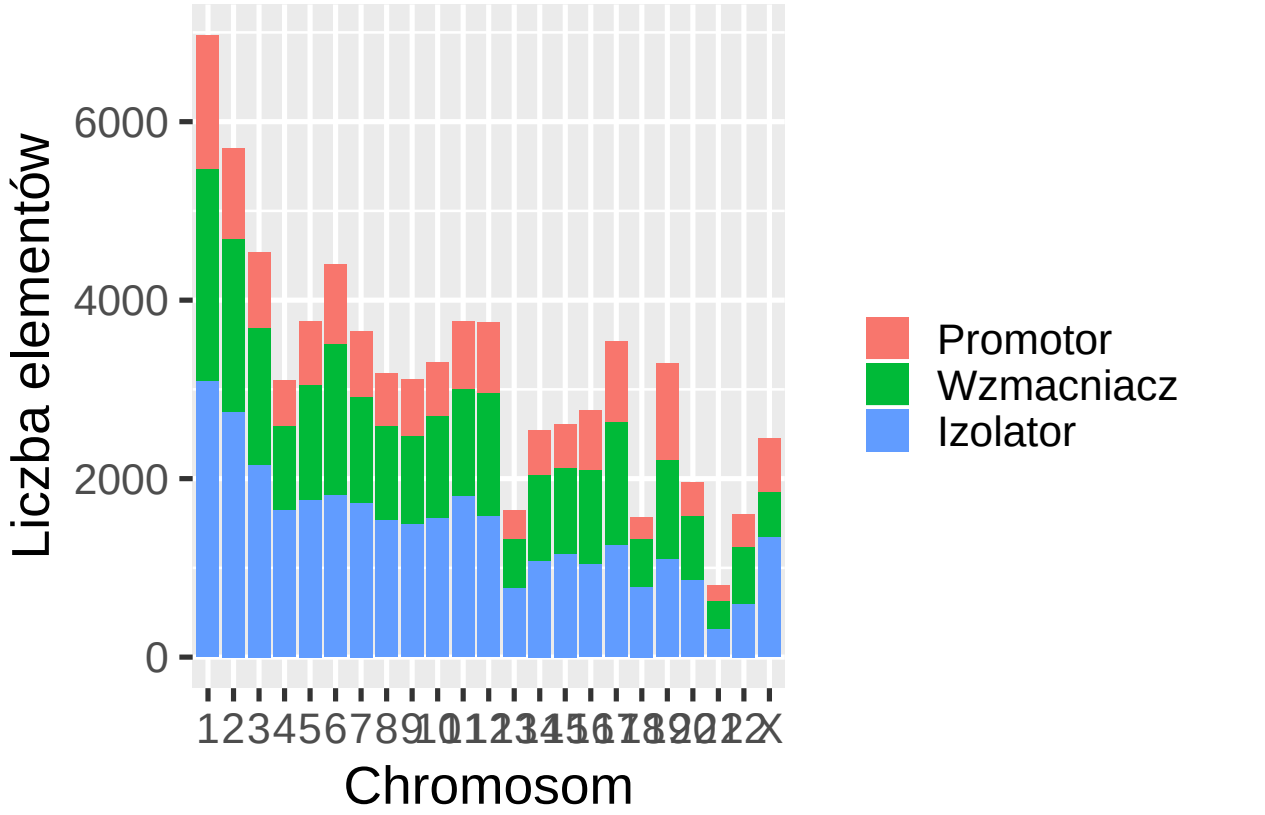



```
podstawa +  theme(legend.position=c(1,1)) # prawa góra
```

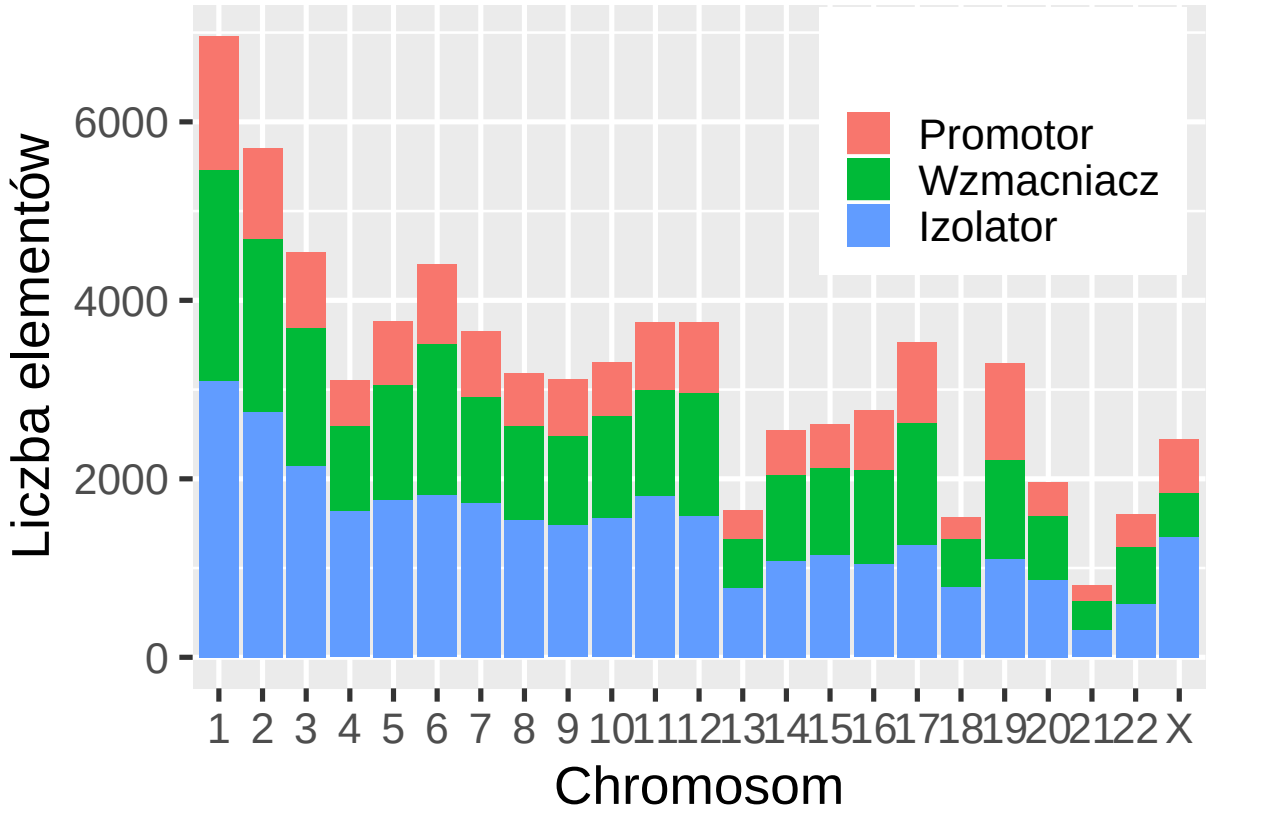


Usunięcie legendy

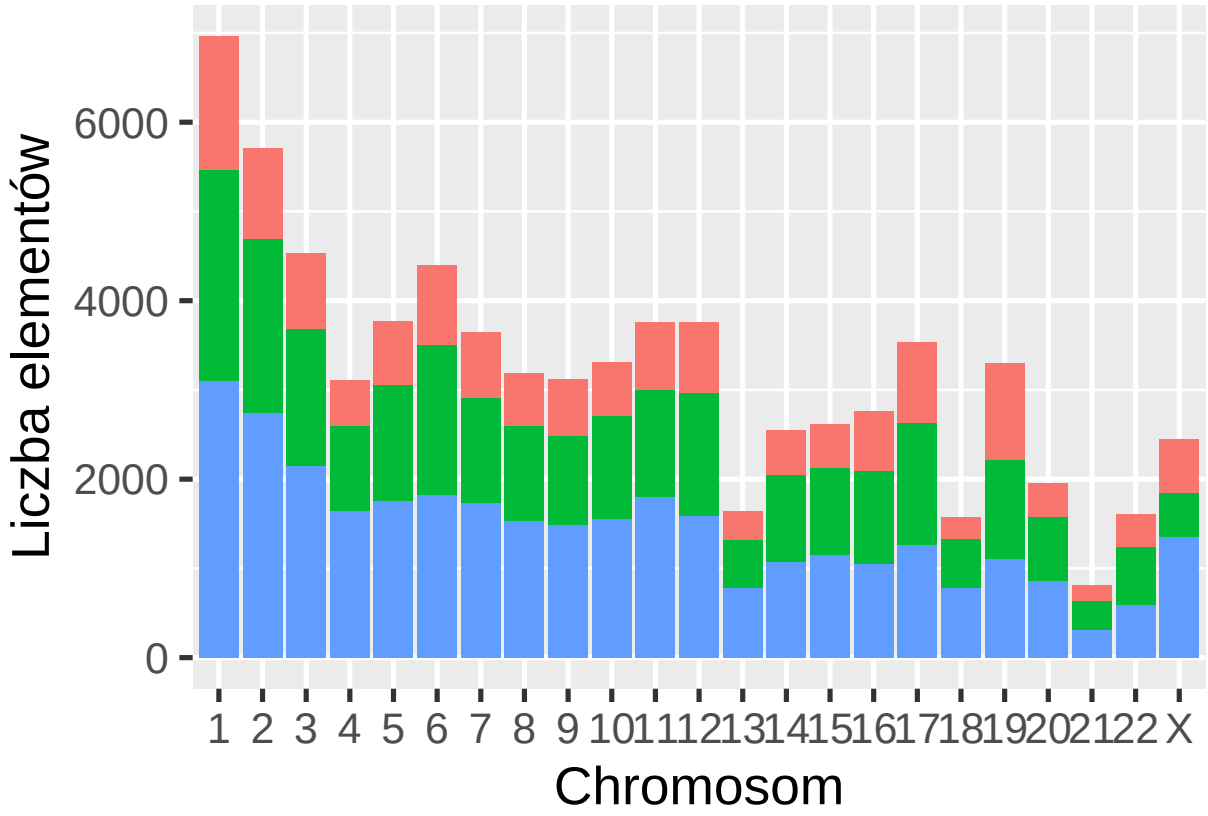
```
podstawa + labs(fill="")
```




```
podstawa + labs(fill="") + theme(legend.position=c(0.8,0.8))
```

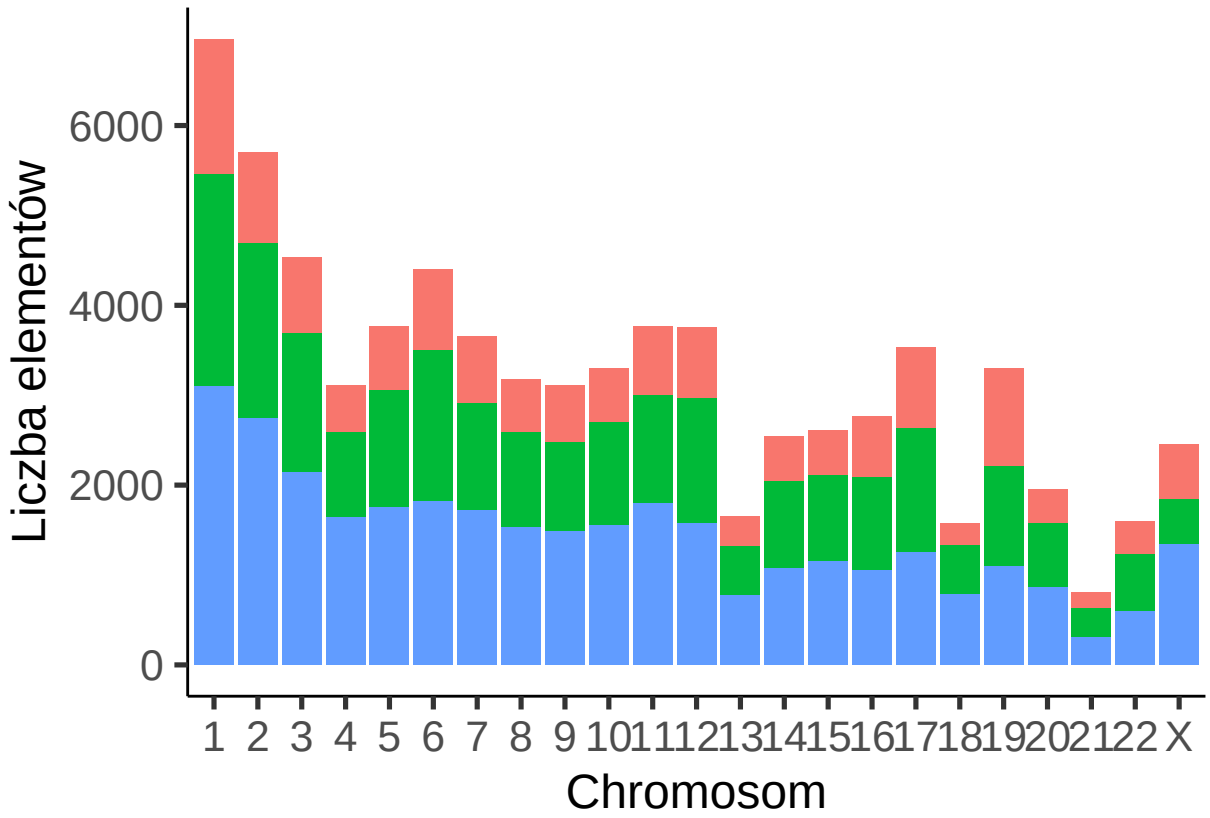


```
podstawa + guides(fill=FALSE)
```



Stworzenie własnego schematu stylu

```
publication_style <- podstawa + guides(fill=FALSE) +  theme(axis.line = element_line(size=0.5),panel.ba
```


Liczba elementów
6000
4000
2000
0
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 X
Chromosom