

Lab05

Jakub Bryl

13 11 2019

Zadanie 1: Przychod per osoba

Dane zebrane z strony = <https://www.gapminder.org/data/>

Zczytanie zbioru

`dane[1]` -> sprawia ze wyciagamy caly data.frame czyli cala kolumne z jej nazwa `dane[[1]]` -> wyciagamy same elementy kolumny zapisane jako vector

```
income <- read_csv("dane.csv")
```

```
## Parsed with column specification:
## cols(
##   .default = col_double(),
##   country = col_character()
## )
```

```
## See spec(...) for full column specifications.
```

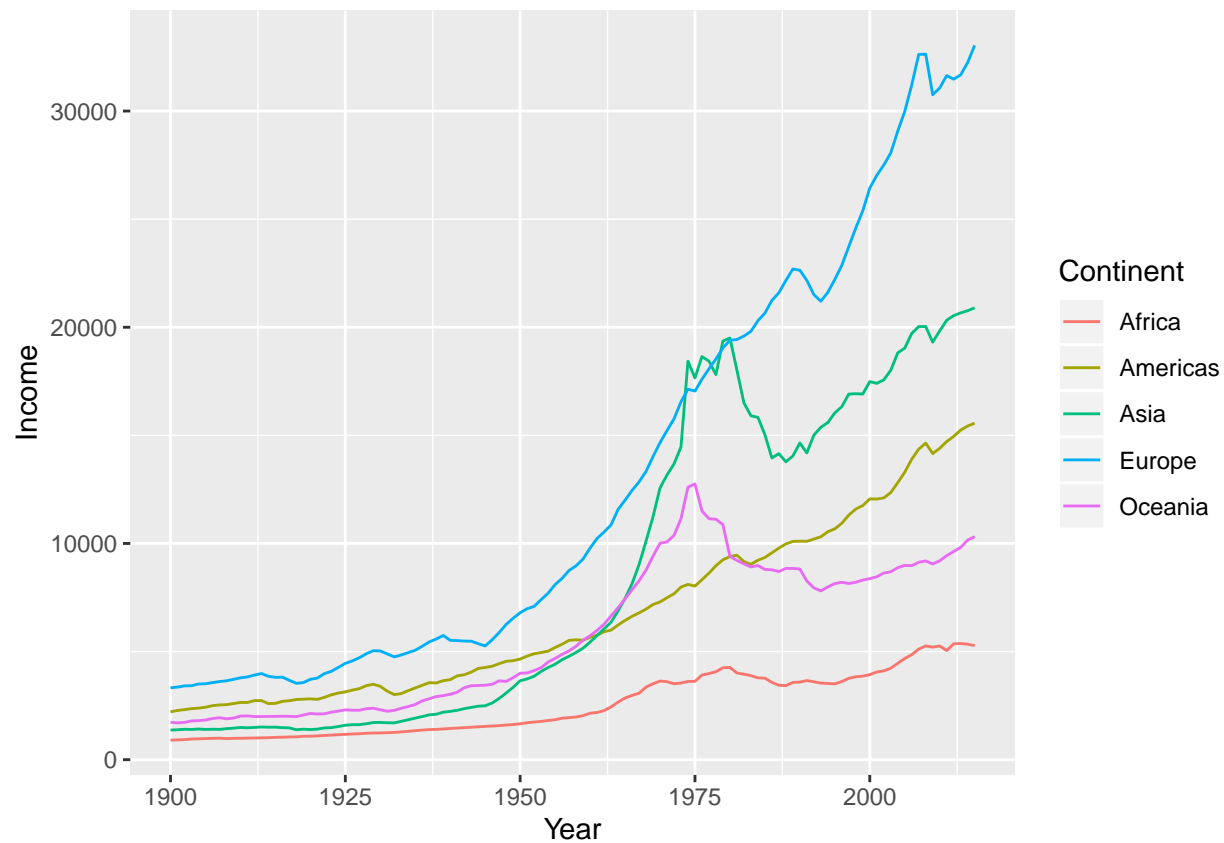
```
income$country <- countrycode(sourcevar = income[[1]], origin = 'country.name', destination = 'continent')
colnames(income)[1] <- 'Continent'
```

Segregowanie zbioru (tidy)

```
# Pomijamy kontynent ktory powinien zostac niezmienny
data <- income %>% gather( key = "Year", value="Income", -Continent)
data <- filter(data, Year >= 1900 & Year < 2016)
# Grupowanie danych z tego samego kontynentu & roku oraz obliczanie sredniej
data <- data %>% group_by(Continent, Year) %>% summarise(income = mean(Income))
colnames(data)[3] <- 'Income'
# Przekształcenie danych w odpowiednie typy danych
data$Continent <- as_factor(data$Continent)
data$Year <- as.numeric(data$Year)
```

Prezentowanie wykresow

```
ggplot(data = data, aes(x=Year, y=Income , color=Continent)) + geom_line()
```



Tibia

Przygotowanie odpowiednich bibliotek

```
require(rvest)
```

```
## Loading required package: rvest
```

```
## Loading required package: xml2
```

```
##
```

```
## Attaching package: 'rvest'
```

```
## The following object is masked from 'package:purrr':
```

```
##
```

```
##   pluck
```

```
## The following object is masked from 'package:readr':
```

```
##
```

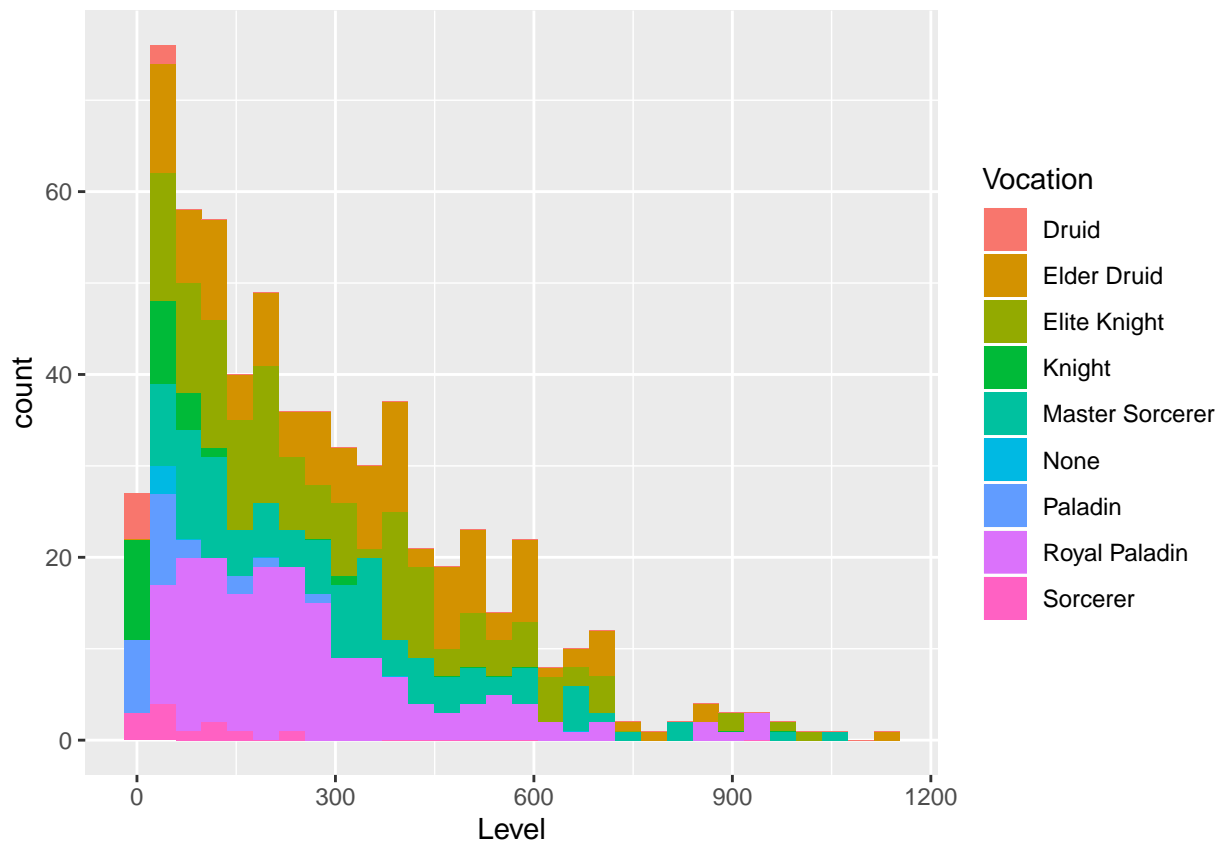
```
##   guess_encoding
```

Zaladowanie zmiennych

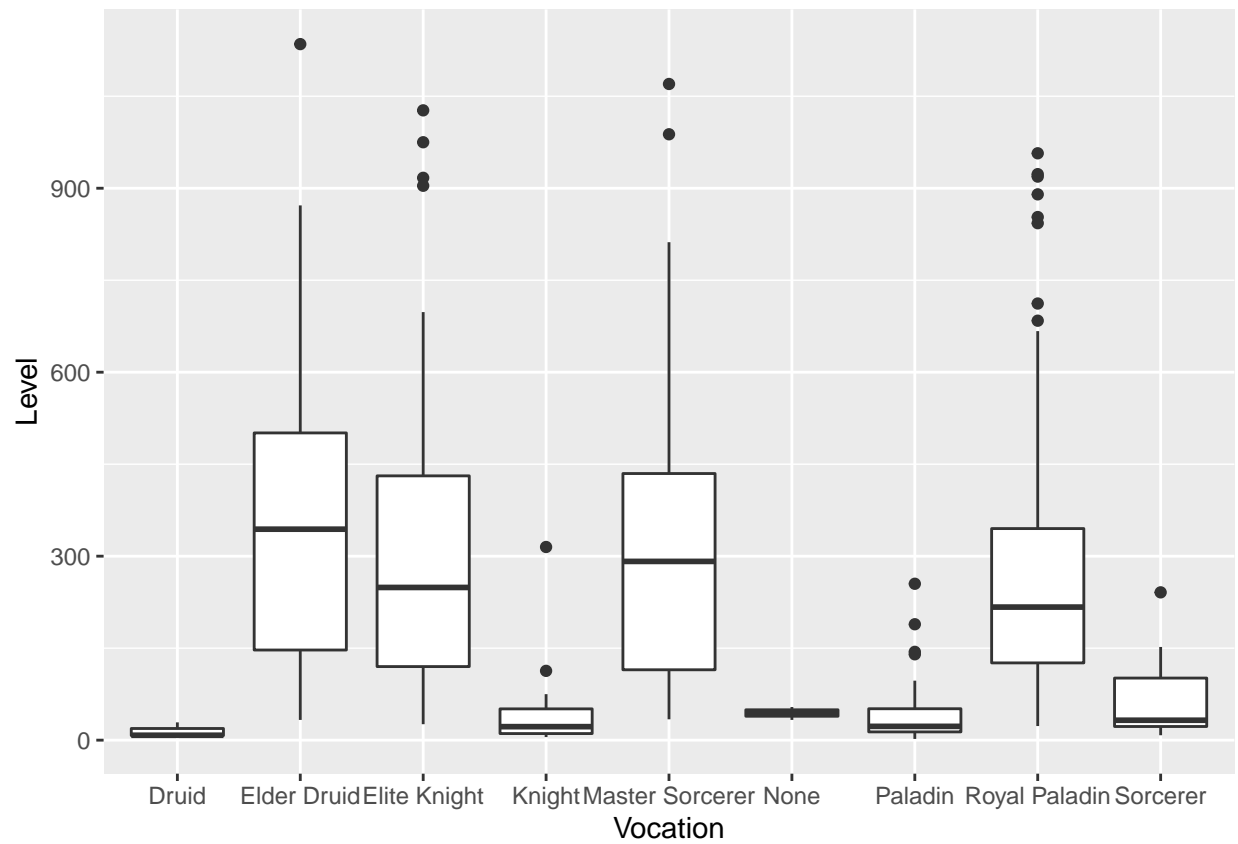
```
url_ <- 'https://www.tibia.com/community/?subtopic=worlds&world=Secura&order=level_desc'  
# Przeczytanie danych znajdujących się pod url_ -> zmiennej  
world_html <- read_html(url_)  
# Odpowiednio dobrany Xpath do pierwszej głównej tabeli  
webpage <- html_nodes(world_html, xpath = '//table')  
# Wyciągnięcie z głównej tabeli tej która nas interesuje i przekształcenie do odpowiedniego formatu  
webpage_final <- html_table(webpage[6]) %>% as.data.frame()  
# Zmiana nazw "na sztywno"  
colnames(webpage_final) <- c("Name", "Level", "Vocation")  
# Pominiecie pierwszego wiersza  
webpage_final <- webpage_final[-1,]  
# Formatowanie do odpowiedniego typu zmiennych  
webpage_final$Level <- as.numeric(webpage_final$Level)  
webpage_final$Vocation <- as.factor(webpage_final$Vocation)
```

Wizualizacja

```
ggplot(data = webpage_final, aes(x=Level, fill= Vocation)) + geom_histogram(position = "stack")  
  
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



```
ggplot(data = webpage_final, aes(x=Vocation, y = Level)) + geom_boxplot()
```



```
ggplot(data = webpage_final, aes(x=Vocation)) + geom_bar()
```

