Analiza statystyczna danych dialogowych z serialu Gra o tron

Mikołaj Wielgos

21 stycznia 2022 r.

Spis treści

Wstęp

Projekt na przedmiot Rachunek prawdopodobieństwa i statystyka 2021/22.

Obejmuje on analizę statystyczną wszystkich kwestii bohaterów w serialu *Gra o tron*

Źródło danych opracowywanych *www.kaggle.com/gopinath15/gameofthrones*

Początek

Importuję potrzebne pakiety

```
library(DBI)
library(dbplot)
library(dplyr)
library(ggplot2)
```

Łączę się z bazą danych "got-dataset.db"

```
con <- dbConnect(drv=RSQLite::SQLite(), dbname="data/got-dataset.db")
```

Sprawdzam jakie tabele zawiera baza danych

```
dbListTables(con)
```

```
## [1] "got-dialogues"
```

Czyszczenie danych

Puste pola 'Speaker'

Interesują nas kwestie wypowiedziane tylko przez postacie w serialu (wpis Speaker nie może być pusty). Przykładowy błędny wiersz

```
query <- "
SELECT substr(Text,1,20)||'...'as 'Text',
Speaker,
Episode,
Season
FROM [got-dialogues]
WHERE Speaker=''
LIMIT 1"
dbGetQuery(con, query)
```

```
##                      Text Speaker             Episode    Season
## 1 [First scene opens w...         e1-Winter is Coming season-01
```

Kwerenda usuwająca wiersze

```
query <- "
DELETE
FROM [got-dialogues]
WHERE SPEAKER = ''"
res <- dbSendStatement(con, query)
```

Sprawdzam na ile wierszy miała wpływa kwerenda

```
dbGetRowsAffected(res)
```

```
## [1] 8356
```

```
dbClearResult(res)
```

Kwestie postaci w tle

Usuwamy kwestie postaci w tle, przedstawianych jako np. Woman #4 Przykładowy błędny wiersz

```
query <- "
SELECT substr(Text,1,20)||'...'as 'Text',
Speaker,
Episode,
Season
FROM [got-dialogues]
WHERE Speaker LIKE '%#%'
LIMIT 1"
dbGetQuery(con, query)
```

```
##                      Text            Speaker   Episode    Season
## 1   You earned that, S... Night's Watcher #1 e9-Baelor season-01
```

Kwerenda usuwająca wiersze

```
query <- "
DELETE
FROM [got-dialogues]
WHERE Speaker LIKE '%#%'"
res <- dbSendStatement(con, query)
```

Sprawdzam na ile wierszy miała wpływa kwerenda

```
dbGetRowsAffected(res)
```

```
## [1] 206
```

```
dbClearResult(res)
```

Wspólne kwestie

Usuwamy wspólne kwestie, dotyczy pozycji typu 'ALL TOGETHER', 'ALL THREE', 'ALL AT THE BACK'
Przykładowy błędny wiersz

```
query <- "
SELECT substr(Text,1,20)||'...'as 'Text',
Speaker,
Episode,
Season
FROM [got-dialogues]
WHERE Speaker LIKE '%ALL%'
LIMIT 1"
dbGetQuery(con, query)
```

```
##                       Text Speaker            Episode    Season
## 1   (chanting)  The Ki...     All e10-Fire and Blood season-01
```

Kwerenda usuwająca wiersz

```
query <- "
DELETE
FROM [got-dialogues]
WHERE Speaker LIKE '%ALL%'"
res <- dbSendStatement(con, query)
```

Sprawdzam na ile wierszy miała wpływa kwerenda

```
dbGetRowsAffected(res)
```

```
## [1] 177
```

```
dbClearResult(res)
```

Nieznaczące wypowiedzi

Usuwamy pojedyncze, mało znaczące wypowiedzi

Przykładowy błędny wiersz

```
query <- "
SELECT substr(Text,1,20)||'...' as 'Text',
Speaker,
Episode,
Season
FROM [got-dialogues]
GROUP BY Speaker
HAVING COUNT(*)=1
LIMIT 1"
dbGetQuery(con, query)
```

```
##                       Text
## 1  come to King's Land...
##
## 1 "Robb, I write to you with a heavy heart. Our good king Robert is dead, killed from wounds he took
##   Episode    Season
## 1      e5 season-07
```

Kwerenda usuwająca wiersze

```
query <- "
DELETE
FROM [got-dialogues]
where Speaker in (SELECT Speaker
FROM [got-dialogues]
GROUP BY Speaker
HAVING COUNT(*)=1)"
res <- dbSendStatement(con, query)
```

Sprawdzam na ile wierszy miała wpływa kwerenda

```
dbGetRowsAffected(res)
```

```
## [1] 363
```

```
dbClearResult(res)
```

Ujednolicenie pola 'Speaker'

W bazie danych pojawiają się wpisy typu Speaker='Roose' oraz Speaker='ROOSE', dlatego trzymamy się jednej wersji (WIELKIE LITERY)

```
query <- "
UPDATE [got-dialogues]
SET Speaker = UPPER(Speaker)"
res <- dbSendStatement(con, query)
dbGetRowsAffected(res)
```

```
## [1] 24096
```

```
dbClearResult(res)
```

Podsumowanie i przykładowe dane

Po przeczyszczeniu przykładowe wpisy w bazie wyglądają następująco (kolumna 'Text' ograniczona do 20 znaków, by zwiększyć czytelność)

```
query <- "
SELECT substr(Text,1,20)||'...' as 'Text',
Speaker,
Episode,
```

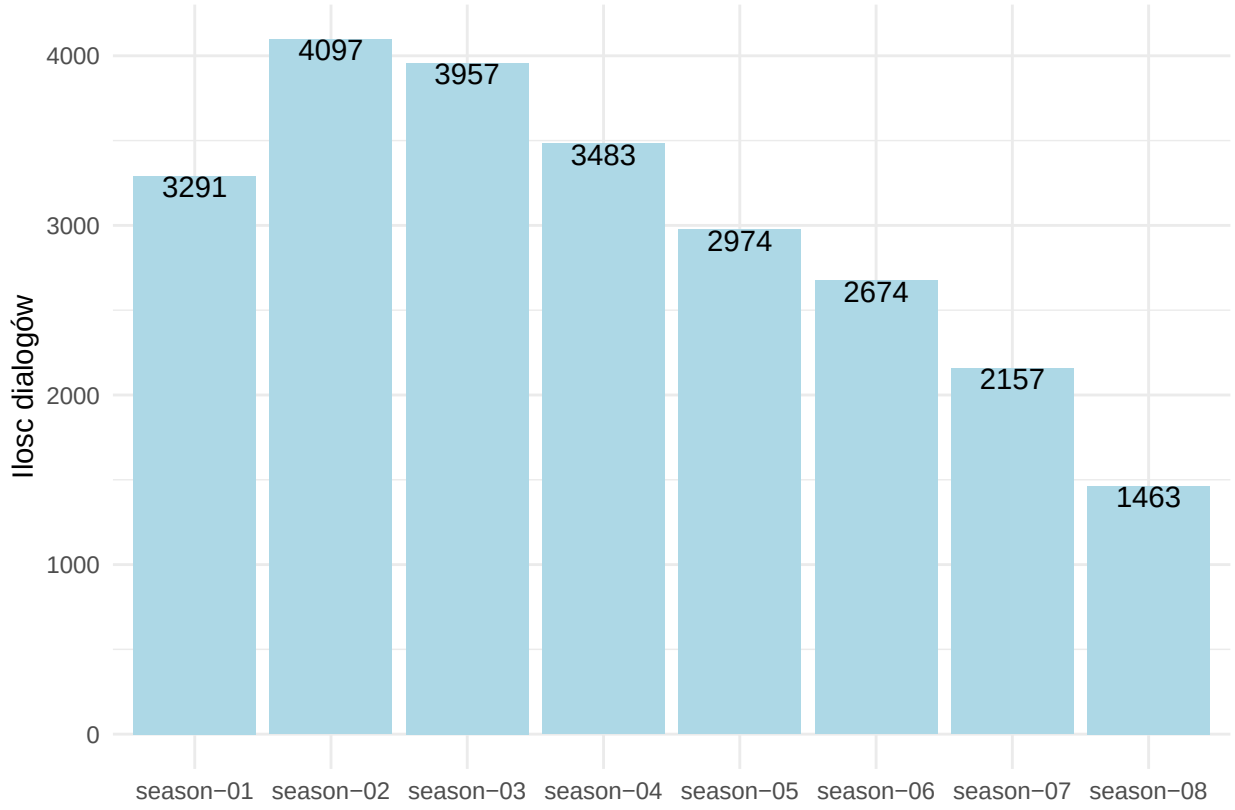
```
Season
FROM [got-dialogues]
LIMIT 5"
dbGetQuery(con, query)
```

```
##                       Text      Speaker             Episode    Season
## 1  What d'you expect? ... WAYMAR ROYCE e1-Winter is Coming season-01
## 2  I've never seen wil...         WILL e1-Winter is Coming season-01
## 3  How close did you g... WAYMAR ROYCE e1-Winter is Coming season-01
## 4  Close as any man wo...         WILL e1-Winter is Coming season-01
## 5  We should head back...        GARED e1-Winter is Coming season-01
```

Analiza eksploracyjna

Ilość dialogów w poszczególnych sezonach

```
df <- dbGetQuery(con,"
SELECT Season,
COUNT(*)
FROM [got-dialogues]
GROUP BY Season")
ggplot(data=df, aes(x = df[,1], y=df[,2])) +
  geom_bar(stat = "identity", fill="lightblue") +
  geom_text(aes(label = df[,2]), vjust = 1) +
  labs(x="", y="Ilość dialogów") +
  theme_minimal()
```



Wskaźniki:

- Średnia 3012
- Mediana 3132.5
- Wariancja 8.0229229×10^5
- Odchylenie standardowe 895.707701

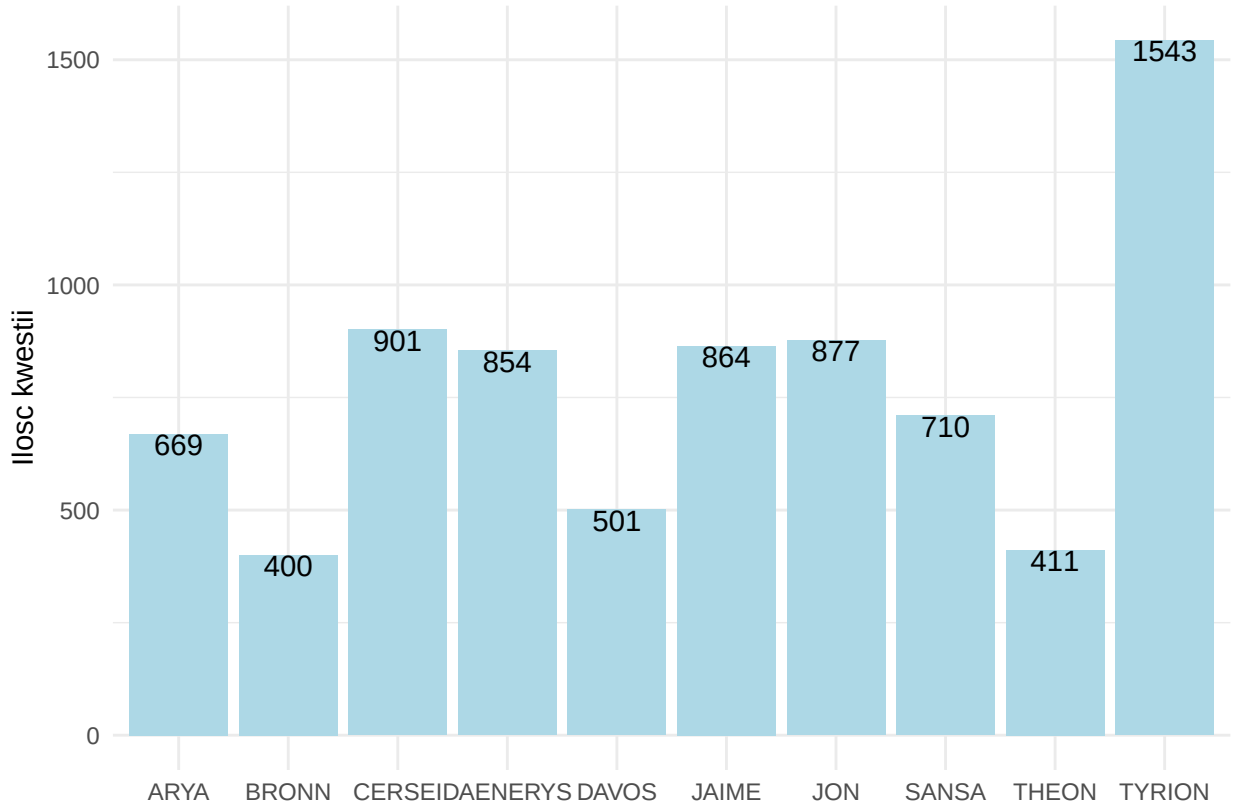
```
summary(df[,2])
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##    1463    2545    3132    3012    3602    4097
```

Ilość dialogów poszczególnych bohaterów (wszystkie sezony, 10 największych)

```
df <- dbGetQuery(con,"
SELECT Speaker,
COUNT(*)
FROM [got-dialogues]
GROUP BY Speaker
ORDER BY 2 DESC
LIMIT 10")
ggplot(data=df, aes(x = df[,1], y=df[,2])) +
  geom_bar(stat = "identity", fill="lightblue") +
```

```
  geom_text(aes(label = df[,2]), vjust = 1) +
  labs(x="", y="Ilość kwestii") +
  theme_minimal()
```



By policzyć wskaźniki wszystkich bohaterów, ponownie wybieram dane (tym razem bez limitu)

```
df <- dbGetQuery(con,"
SELECT Speaker,
COUNT(*)
FROM [got-dialogues]
GROUP BY Speaker")
```

Wskaźniki (wszystkich bohaterów):

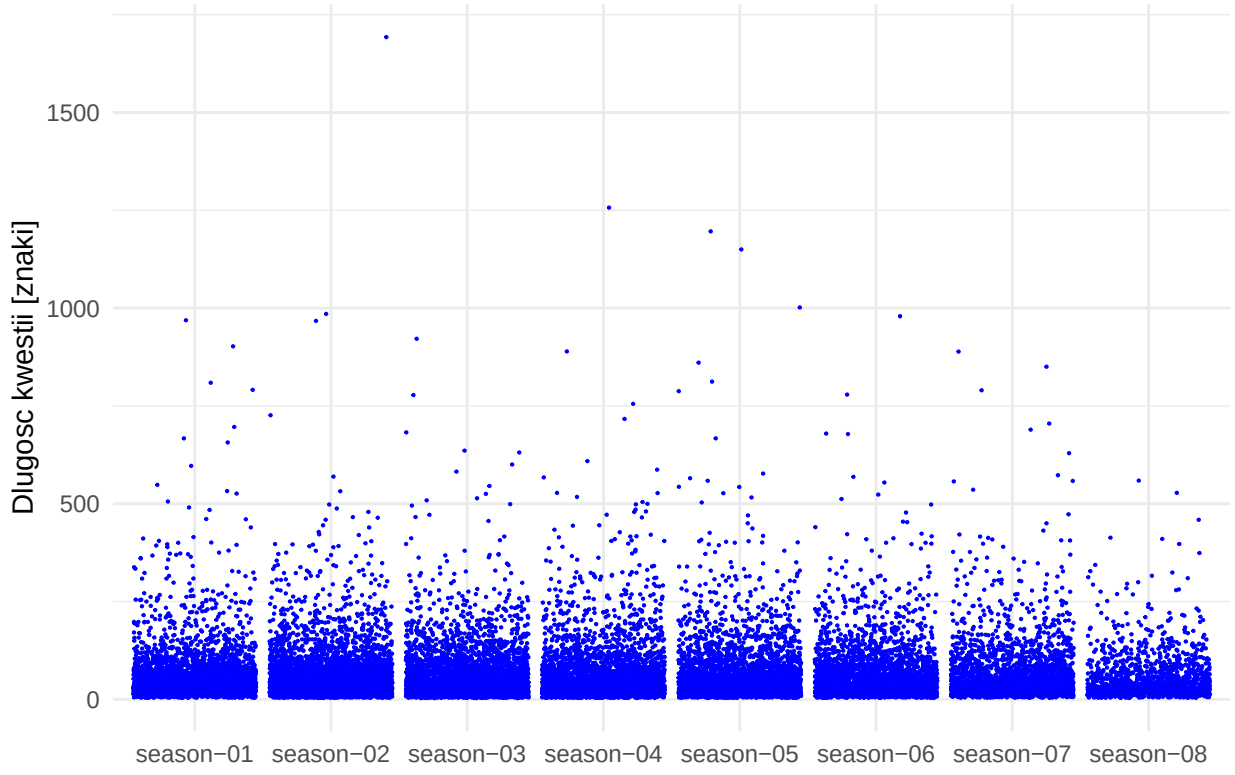
- Średnia 50.0956341
- Mediana 10
- Wariancja 1.7203058×10^4
- Odchylenie standardowe 131.1604266

```
summary(df[,2])
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##     2.0     4.0    10.0    50.1    34.0  1543.0
```

Długości kwestii (znaki) w poszczególnych sezonach

```
df <- dbGetQuery(con,"
SELECT Season,
length(Text)
FROM [got-dialogues]")
ggplot(data=df, aes(x = df[,1], y=df[,2])) +
  geom_jitter(size = 0.1, position = position_jitter(0.45), color="blue") +
  labs(x="",
       y="Długość kwestii [znaki]",
       caption="(Kropka odpowiada pojedynczej wypowiedzianej kwestii)") +
  theme_minimal()
```



(Kropka odpowiada pojedynczej wypowiedzianej kwestii)

Wskaźniki:

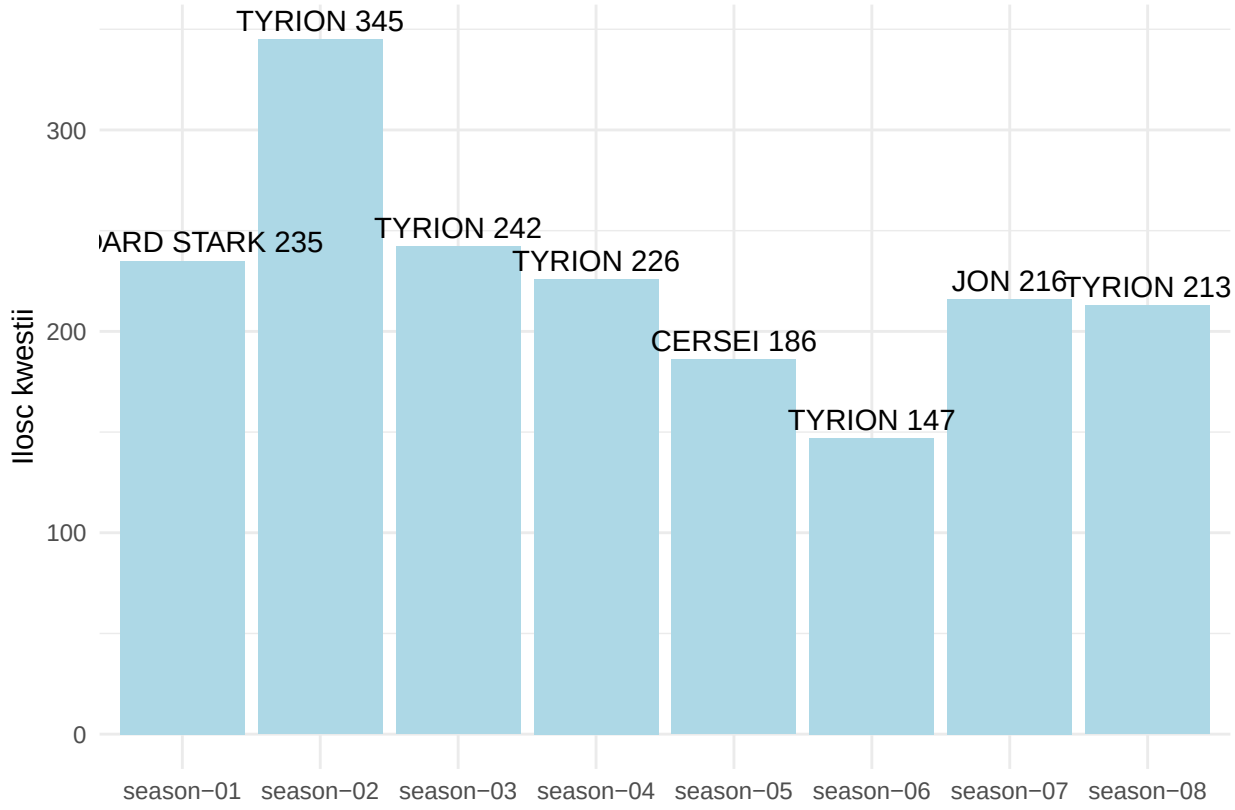
- Średnia 61.7116949
- Mediana 39
- Wariancja 5249.6609315
- Odchylenie standardowe 72.4545439

```
summary(df[,2])
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##    3.00   21.00   39.00   61.71   76.00 1693.00
```

Najczęstszy mówca w danym sezonie

```
df <- dbGetQuery(con,"
SELECT Season,
        (
        SELECT Speaker
        FROM [got-dialogues] as inner
        WHERE outer.Season=inner.Season
        GROUP BY Speaker
        ORDER BY COUNT(*) DESC
        LIMIT 1
        ) as TopSpeaker,
        (
        SELECT COUNT(*)
        FROM [got-dialogues] as inner
        WHERE outer.Season=inner.Season
        GROUP BY Speaker
        ORDER BY COUNT(*) DESC
        LIMIT 1
        ) as Amount
FROM (SELECT DISTINCT Season FROM [got-dialogues]) as outer")
ggplot(data=df, aes(x = df[,1], y=df[,3])) +
  geom_bar(stat = "identity", fill="lightblue") +
  geom_text(aes(label = paste(df[,2],df[,3])), vjust = -0.4) +
  labs(x="", y="Ilość kwestii") +
  theme_minimal()
```



Wskaźniki:

- Średnia 226.25
- Mediana 221
- Wariancja 3229.6428571
- Odchylenie standardowe 56.8299468

```
summary(df[,3])
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   147.0   206.2   221.0   226.2   236.8   345.0
```

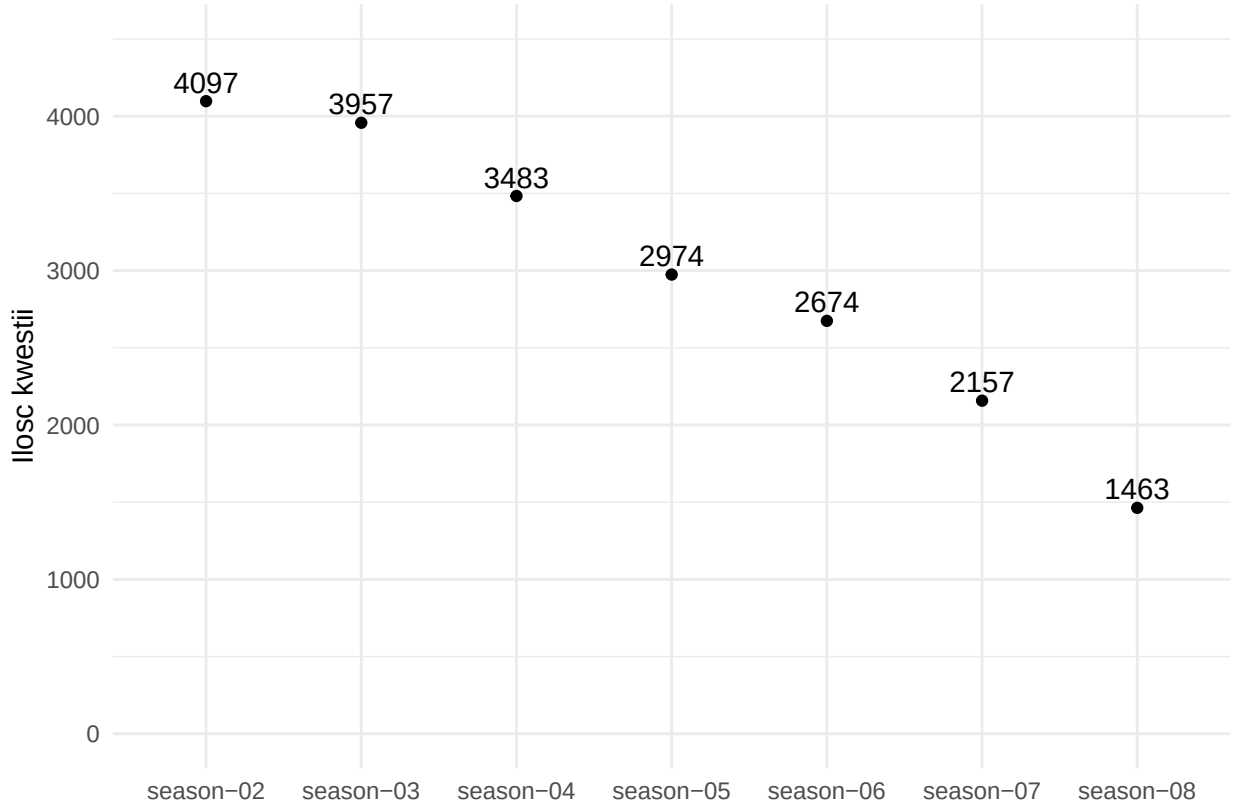
Regresja liniowa

Punkty i wartości na których oprę regresję

Do regresji nie wliczam sezonu 1, w którym wartości dialogów nie układają się w stosunku do reszty. Punkty na których obliczę wyglądają następująco:

```
df <- dbGetQuery(con,"
SELECT Season,
COUNT(*)
FROM [got-dialogues]
WHERE Season !='season-01'
```

```
GROUP BY Season")
ggplot(data=df, aes(x = df[,1], y=df[,2])) +
  geom_point(stat = "identity", fill="lightblue") +
  geom_text(aes(label = df[,2]), vjust = -0.4) +
  labs(x="", y="") +
  scale_y_continuous(name="Ilość kwestii", limits=c(0, 4500))+
  theme_minimal()
```



Model regresji liniowej

Tworzę model

```
df <- dbGetQuery(con,"
SELECT substr(Season,9,10),
COUNT(*)
FROM [got-dialogues]
WHERE Season !='season-01'
GROUP BY Season")
df2 <- data.frame(x = as.numeric(df[,1]), y=df[,2])
model_lm <- lm(y~x, data=df2)
df_lm <- data.frame(x = 2:12)
```

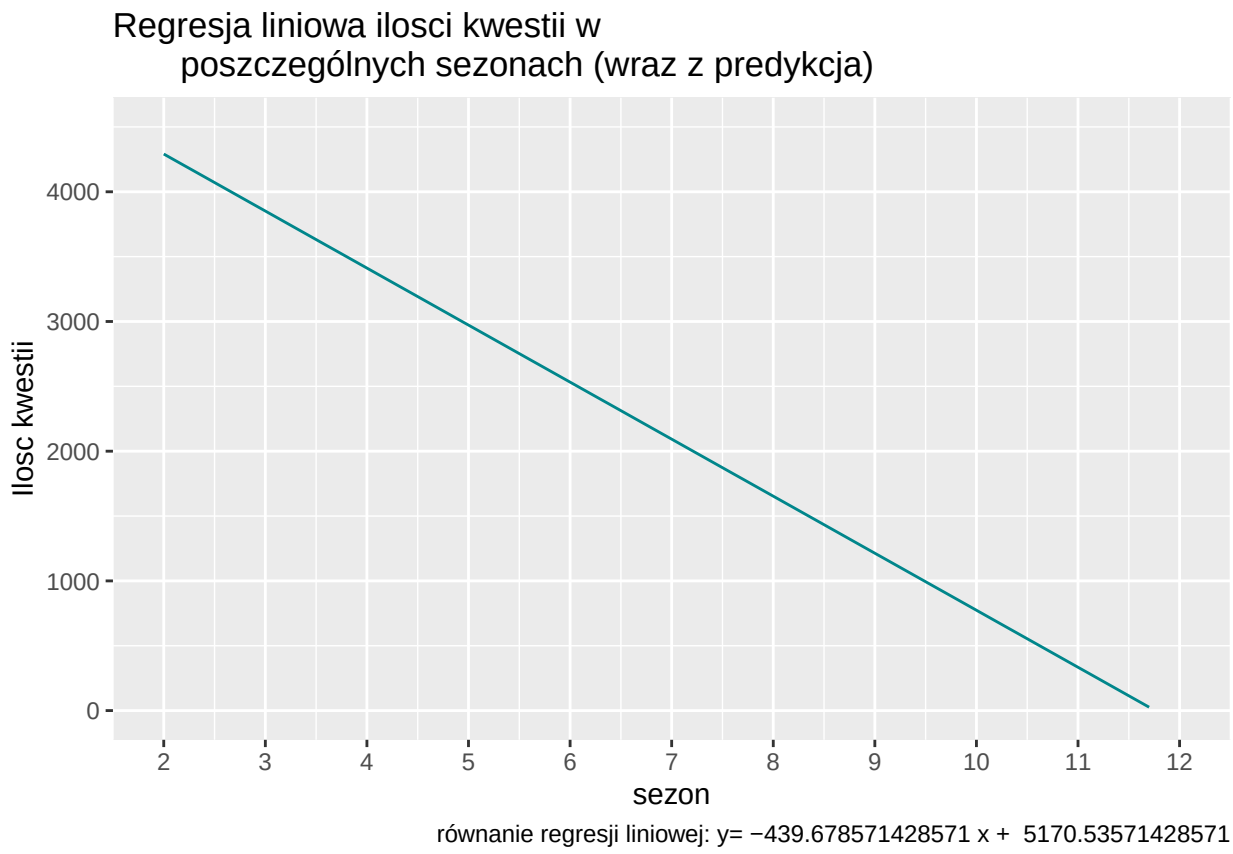
Informacje o modelu

```
summary(model_lm)
```

```
##
## Call:
## lm(formula = y ~ x, data = df2)
##
## Residuals:
##        1        2        3        4        5        6        7
## -194.179  105.500   71.179    1.857  141.536   64.214 -190.107
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  5170.54     153.81   33.62 4.38e-07 ***
## x            -439.68      28.56  -15.39 2.10e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 151.1 on 5 degrees of freedom
## Multiple R-squared:  0.9793, Adjusted R-squared:  0.9752
## F-statistic:   237 on 1 and 5 DF,  p-value: 2.099e-05
```

Wykres regresji liniowej

```
ggplot(df_lm, aes(x)) +
  stat_function(
    fun=function(x) coefficients(model_lm)[1]+x*(coefficients(model_lm)[2]),
    color='turquoise4') +
  scale_x_continuous(
    "sezon",
    labels = as.character(df_lm$x),
    breaks = df_lm$x,
    limits=c(2,12))+
  scale_y_continuous(name="Ilość kwestii", limits=c(0, 4500)) +
  labs(title="Regresja liniowa ilości kwestii w
       poszczególnych sezonach (wraz z predykcją)",
       caption=paste("równanie regresji liniowej: y=",
                     coefficients(model_lm)[2],"x + ",
                     coefficients(model_lm)[1]))
```



Obliczam punkt przecięcia z osią OX.

```
point <- -1*(coefficients(model_lm)[1])/(coefficients(model_lm)[2])
print(point)
```

```
## (Intercept)
##    11.75981
```

Czy w sezonie 12 już nikt się nie wypowie? :D

Zakończenie

Kończę pracę z bazą

```
dbDisconnect(con)
unlink("data/got-dataset.db")
```

Podsumowanie

Choć wybrałem dane proste, bo przecież to tylko kwestie bohaterów, można było dowiedzieć się z nich całkiem sporo. Wniosek po krótce jest taki, że od sezonu 2 Gry o tron, bohaterowie co raz bardziej nie lubią się wypowiadać. Dodatkowo w sezonie 2 padł rekord pod względem długości kwestii (ponad 1500 znaków w pojedynczej wypowiedzi).