Лингвистический анализ коннотации игрового сленга

Корнилова Елизавета

Installs

```
library(tidyverse)
```

```
## ── Attaching core tidyverse packages ──────────────────────────────────────── tidyverse 2.0.0 ──
## ✔ dplyr     1.1.4     ✔ readr     2.1.5
## ✔ forcats   1.0.0     ✔ stringr   1.5.1
## ✔ ggplot2   3.5.1     ✔ tibble    3.2.1
## ✔ lubridate 1.9.4     ✔ tidyr     1.3.1
## ✔ purrr     1.0.2
## ── Conflicts ────────────────────────────────────────────────────────────── tidyverse_conflicts() ──
## ✖ dplyr::filter() masks stats::filter()
## ✖ dplyr::lag()    masks stats::lag()
## ℹ Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become errors
```

```
library(ggplot2)
library(plotly)
```

```
## Warning: пакет 'plotly' был собран под R версии 4.4.3
```

```
##
## Присоединяю пакет: 'plotly'
##
## Следующий объект скрыт от 'package:ggplot2':
##
##     last_plot
##
## Следующий объект скрыт от 'package:stats':
##
##     filter
##
## Следующий объект скрыт от 'package:graphics':
##
##     layout
```

```
library(kableExtra)
```

```
## Warning: пакет 'kableExtra' был собран под R версии 4.4.3
```

```
##
## Присоединяю пакет: 'kableExtra'
##
## Следующий объект скрыт от 'package:dplyr':
##
##     group_rows
```

```
library(factoextra)
```

```
## Warning: пакет 'factoextra' был собран под R версии 4.4.3
```

```
## Welcome! Want to learn more? See two factoextra-related books at https://goo.gl/ve3WBa
```

```
library(vegan)
```

```
## Warning: пакет 'vegan' был собран под R версии 4.4.3
```

```
## Загрузка требуемого пакета: permute
```

```
## Warning: пакет 'permute' был собран под R версии 4.4.3
```

```
## Загрузка требуемого пакета: lattice
```

```
library(FactoMineR)
```

```
## Warning: пакет 'FactoMineR' был собран под R версии 4.4.3
```

```
library(dplyr)
library(lmtest)
```

```
## Загрузка требуемого пакета: zoo
##
## Присоединяю пакет: 'zoo'
##
## Следующие объекты скрыты от 'package:base':
##
##     as.Date, as.Date.numeric
```

```
library(glmnet)
```

```
## Warning: пакет 'glmnet' был собран под R версии 4.4.3
```

```
## Загрузка требуемого пакета: Matrix
##
## Присоединяю пакет: 'Matrix'
##
## Следующие объекты скрыты от 'package:tidyr':
##
##     expand, pack, unpack
##
## Loaded glmnet 4.1-9
```

```
library(text2vec)
```

```
## Warning: пакет 'text2vec' был собран под R версии 4.4.3
```

```
library(caret)
```

```
## Warning: пакет 'caret' был собран под R версии 4.4.3
```

```
##
## Присоединяю пакет: 'caret'
##
## Следующий объект скрыт от 'package:vegan':
##
##     tolerance
##
## Следующий объект скрыт от 'package:purrr':
##
##     lift
```

```
library(Matrix)
library(stringr)
library(vcd)
```

```
## Загрузка требуемого пакета: grid
```

```
library(ggmosaic)
```

```
## Warning: пакет 'ggmosaic' был собран под R версии 4.4.3
```

```
##
## Присоединяю пакет: 'ggmosaic'
##
## Следующие объекты скрыты от 'package:vcd':
##
##     mosaic, spine
```

Введение

Цель проекта - разработать модель классификации коннотации игрового сленга.

Датасет содержит термины игрового сленга из различных видеоигр и их классификацию по словообразовательным моделям. Проект продолжает и расширяет бакалаврскую работу, фокусируясь на определении оннотации сленговых слов и её зависимости от контекста, жанра игры, словообразовательной модели.

Гипотезы:

H_0: Коннотация слова не зависит от части речи. H_1: Коннотация слова зависит от части речи.

H_0: Коннотация слова не зависит от контекста. H_1: Коннотация слова зависит от контекста.

H_0: Коннотация слова не зависит от жанра игры. H_1: Коннотация слова зависит от жанра игры.

H_0: Коннотация слова не зависит от модели словообразования. H_1: Коннотация слова зависит от модели словообразования.

H_0: Коннотация слова не зависит от длины слова. H_1: Коннотация слова зависит от длины слова

Датасет содержит 209 примеров и 6 переменных.

Переменные:

- slang word: Сленговое слово или выражение
- word formation model: Модель словообразования
- context: Контекст использования термина
- part of speech: Часть речи
- Genre of game Where it appears: Жанр игры, где встречается термин
- Connotation: Коннотация (нейтральная, положительная, отрицательная)
- Sentence: Пример предложения со сленговым словом

Целевая переменная — Connotation (коннотация), которая принимает следующие значения: neutral, positive, negative, ambivalent, irony/sarcasm.

Предварительный анализ данных

```
df <- read.csv("gaming_slang_2.csv", stringsAsFactors = FALSE)
```

Тип данных: все переменные хранятся как текстовые (chr), что может потребовать их преобразования в факторы для анализа. Пустых значений нет.

```
str(df)
```

```
## 'data.frame':    209 obs. of  7 variables:
##  $ slang.word                  : chr  "To min-max" "Speedrun" "Faceroll" "Bug user" ...
##  $ word.formation.model        : chr  "Word composition" "Word composition" "Word composition" "Word composition" ...
##  $ context                     : chr  "Managing the characteristics of the game character" "Strategies and gameplay" "Game atmosphere and state of the pla
yer" "Evaluating other users" ...
##  $ part.of.speech              : chr  "verb" "noun" "noun" "noun" ...
##  $ Genre.of.game.Where.it.appears: chr  "General" "Action" "MOBA" "General" ...
##  $ Connotation                 : chr  "ambivalent" "ambivalent" "irony/sarcasm" "negative" ...
##  $ Sentence                    : chr  "I min-maxed my warrior's gear to sacrifice defense for maximum DPS." "She set a new world record with her 22-minut
e dungeon speedrun." "Yes leveling does take a long time, but part of the reason why it's so boring is because it's a faceroll. " "Report any bug users you e
ncounter to the devs immediately." ...
```

```
summary(df)
```

```
##   slang.word        word.formation.model   context          part.of.speech
##  Length:209         Length:209           Length:209         Length:209
##  Class :character   Class :character     Class :character   Class :character
##  Mode  :character   Mode  :character     Mode  :character   Mode  :character
##  Genre.of.game.Where.it.appears Connotation          Sentence
##  Length:209                     Length:209         Length:209
##  Class :character               Class :character   Class :character
##  Mode  :character               Mode  :character   Mode  :character
```

```
sum(is.na(df))
```

```
## [1] 0
```

```
# Преобразование переменных в факторы
df$word.formation.model <- as.factor(df$word.formation.model)
df$context <- as.factor(df$context)
df$part.of.speech <- as.factor(df$part.of.speech)
df$Genre.of.game.Where.it.appears <- as.factor(df$Genre.of.game.Where.it.appears)
df$Connotation <- as.factor(df$Connotation)
```

Проверка категориальных переменных выводит все уникальные значения для каждого категориального признака и помогает оценить разнообразие категорий.

```
cat("Уникальные значения коннотации:", levels(df$Connotation), "\n")
```

```
## Уникальные значения коннотации: ambivalent irony/sarcasm negative neutral positive
```

```
cat("Уникальные модели словообразования:", levels(df$word.formation.model), "\n")
```

```
## Уникальные модели словообразования: Abbreviations Acronyms Clipping Conversion Metaphors Metonymy Reduplication Spoonerisms Suffixation Word composition
```

```
cat("Уникальный контекст:", levels(df$context), "\n")
```

```
## Уникальный контекст: Communication process Evaluating other users Expressions associated with victory and defeat Game atmosphere and state of t he player Game mechanics Games and game genres In-game creatures In-game indicators In-game items, resources, and phenomena Managing the char acteristics of the game character Player's role Strategies and gameplay Teamwork Technical terms
```

```
cat("Уникальные части речи:", levels(df$part.of.speech), "\n")
```

```
## Уникальные части речи: adjective adverb  noun verb
```

```
cat("Уникальные жанры игр:", levels(df$Genre.of.game.Where.it.appears), "\n")
```

```
## Уникальные жанры игр: Action FPS General MMORPG MMORPGB MOBA RPG Strategy
```

Так как в датасете полностью отсутсвуют числовые переменные, была добавлена новая переменная word_length, потому что это потенциально важный признак в задачах автоматического распознавания коннотации.

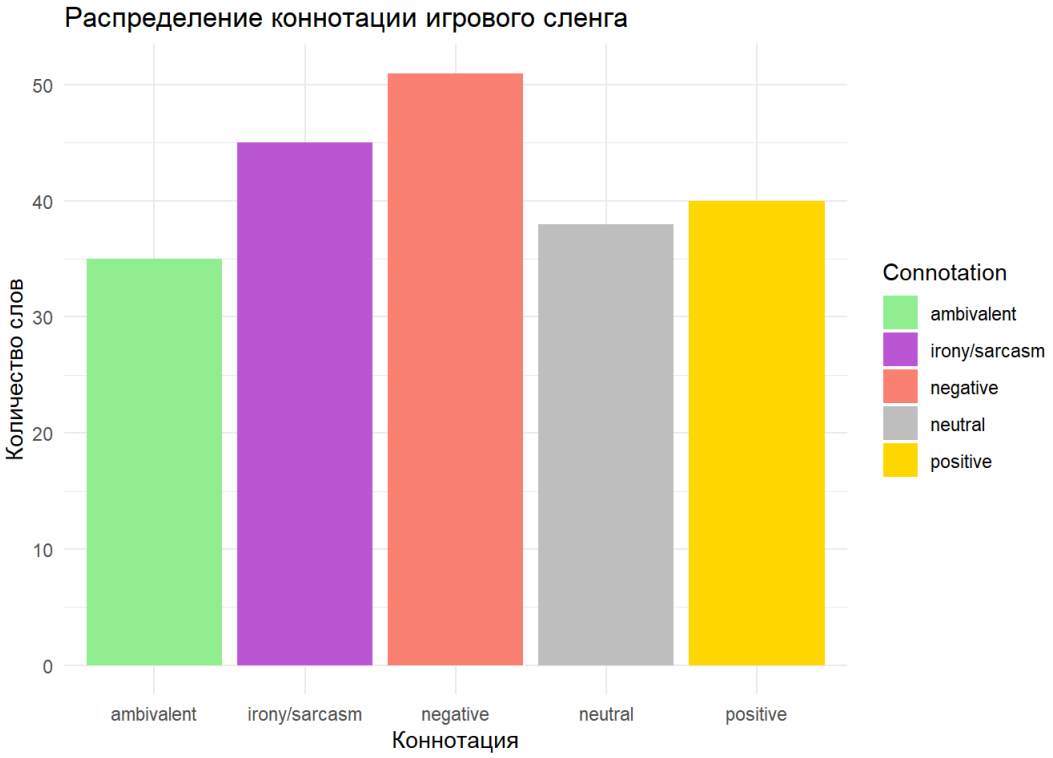
```
df$word_length <- nchar(df$slang.word)
```

Описательная статистика

Распределение коннотации

```
connotation_palette <- c(
  "negative" = "salmon",
  "neutral" = "gray",
  "positive" = "gold",
  "ambivalent" = "lightgreen",
  "irony/sarcasm" = "mediumorchid"
)
```

```
ggplot(df, aes(x = Connotation, fill = Connotation)) +
  geom_bar() +
  scale_fill_manual(values = connotation_palette) +
  labs(title = "Распределение коннотации игрового сленга",
       x = "Коннотация", y = "Количество слов") +
  theme_minimal()
```




```
table(df$Connotation)
```

```
##
##    ambivalent irony/sarcasm      negative       neutral      positive
##            35            45            51            38            40
```

Ключевые наблюдения:

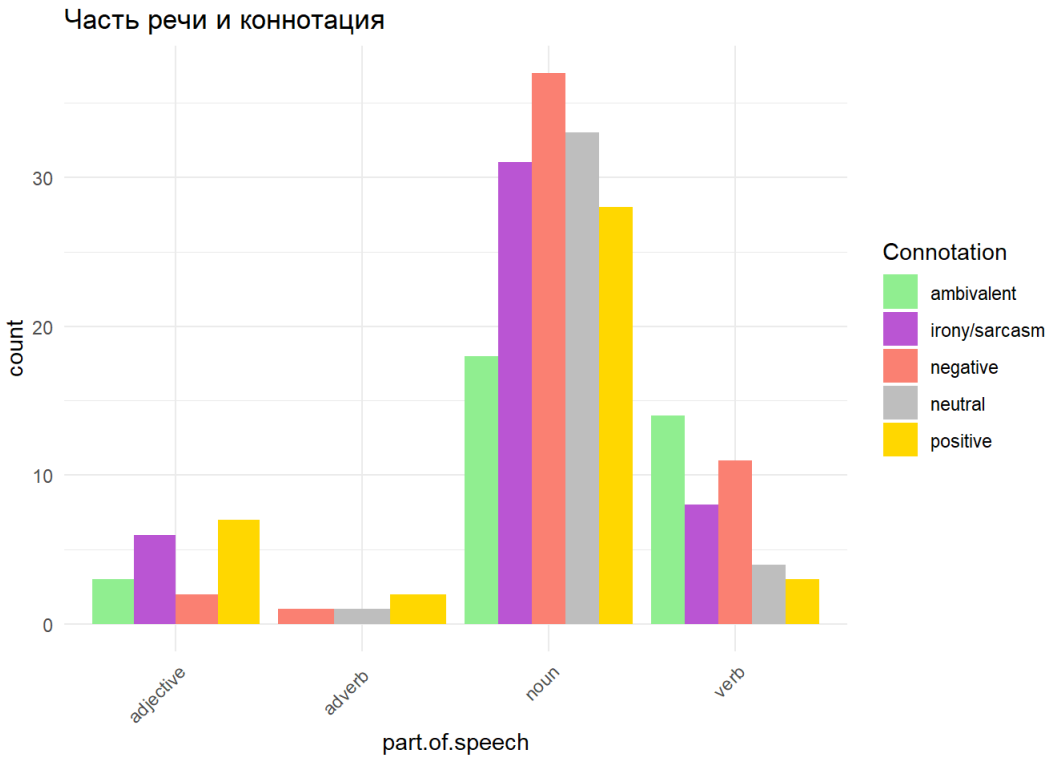
1. **Преобладание негатива**
   Токсичность и критика (`negative` + `irony/sarcasm`) составляют **45.9%** терминов:
   Это указывает на экспрессивно-оценочный характер игрового общения.
2. **Роль иронии**
   Высокая доля `irony/sarcasm` (21.5%) подчёркивает:
   - Склонность к сарказму в онлайн-среде
   - Использование юмора как защитного механизма
3. **Недооценённые категории**
   - `Ambivalent` (16.7%) — перспективны для изучения контекстной зависимости
   - `Positive` (15.8%) — редки, что характерно для оценочного сленга

Вывод: Игровой сленг выполняет преимущественно **оценочную** (а не описательную) функцию, с акцентом на негатив и иронию.

```
# Создание функции для генерации таблиц распределения
create_connotation_table <- function(data, group_var) {
  data %>%
    group_by({{group_var}}, Connotation) %>%
    summarise(Count = n(), .groups = 'drop') %>%
    mutate(Percentage = round(Count / sum(Count) * 100, 1)) %>%
    pivot_wider(names_from = Connotation,
                values_from = c(Count, Percentage),
                names_sep = "_",
                values_fill = list(Count = 0, Percentage = 0))
}
```

Распределение частей речи по коннотации

```
ggplot(df, aes(x = part.of.speech, fill = Connotation)) +
  geom_bar(position = "dodge") +
  scale_fill_manual(values = connotation_palette) +
  theme_minimal() +
  theme(axis.text.x = element_text(angle = 45, hjust = 1)) +
  ggtitle("Часть речи и коннотация")
```




```
part_of_speech_table <- df %>%
  separate_rows(part.of.speech, sep = ", ") %>%
  create_connotation_table(part.of.speech) %>%
  arrange(desc(Count_negative))

part_of_speech_table
```

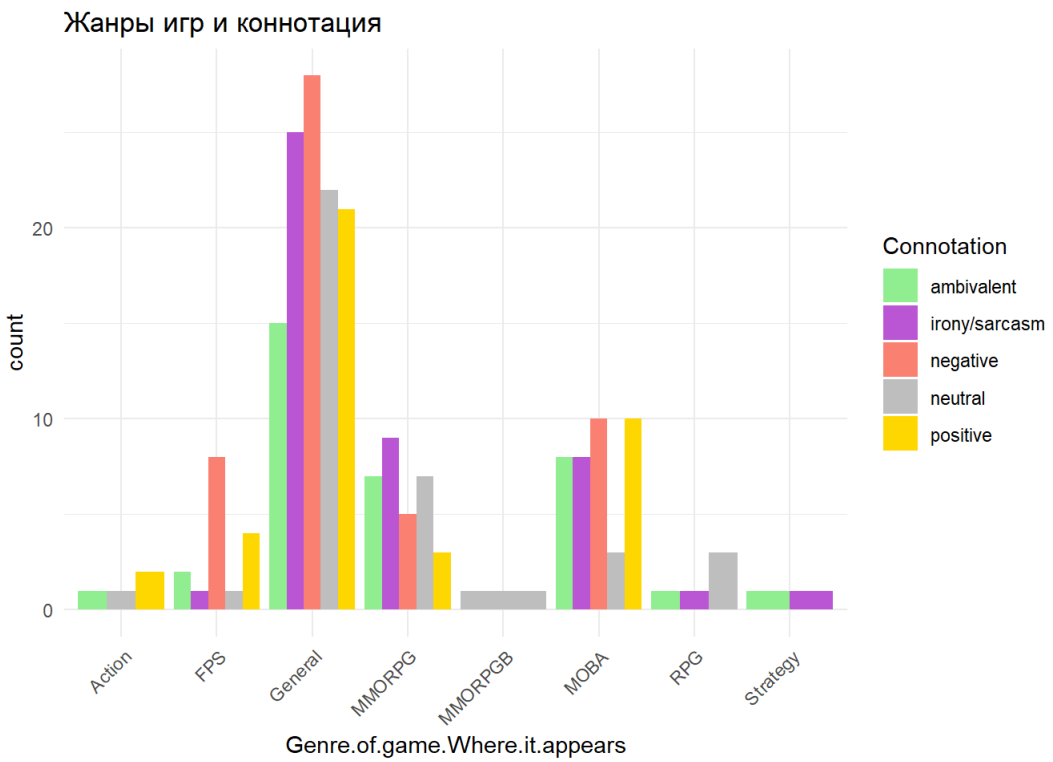
```
## # A tibble: 4 × 11
##   part.of.speech Count_ambivalent `Count_irony/sarcasm` Count_negative
##   <chr>                    <int>                 <int>          <int>
## 1 "noun"                      18                    31             37
## 2 "verb"                      14                     8             11
## 3 "adjective"                  3                     6              2
## 4 "adverb "                    0                     0              1
## # ℹ 7 more variables: Count_positive <int>, Count_neutral <int>,
## #   Percentage_ambivalent <dbl>, `Percentage_irony/sarcasm` <dbl>,
## #   Percentage_negative <dbl>, Percentage_positive <dbl>,
## #   Percentage_neutral <dbl>
```

Ключевые наблюдения:

1. **Существительные** — наиболее распространенная часть речи, при этом они чаще всего несут негативную или нейтральную окраску.
2. **Глаголы** сравнительно нейтральны или негативны. Их положительная окраска встречается редко.
3. **Прилагательные** склонны к позитивной коннотации.
4. **Наречия** встречаются редко, а если встречаются, то могут были либо положительными, либо отрицательными, либо нейтральными.

Распределение жанров игр и коннотаци

```
ggplot(df, aes(x = Genre.of.game.Where.it.appears, fill = Connotation)) +
  geom_bar(position = "dodge") +
  scale_fill_manual(values = connotation_palette) +
  theme_minimal() +
  theme(axis.text.x = element_text(angle = 45, hjust = 1)) +
  ggtitle("Жанры игр и коннотация")
```




```
genre_table <- create_connotation_table(df, Genre.of.game.Where.it.appears) %>%
  arrange(desc(Count_negative))
genre_table
```

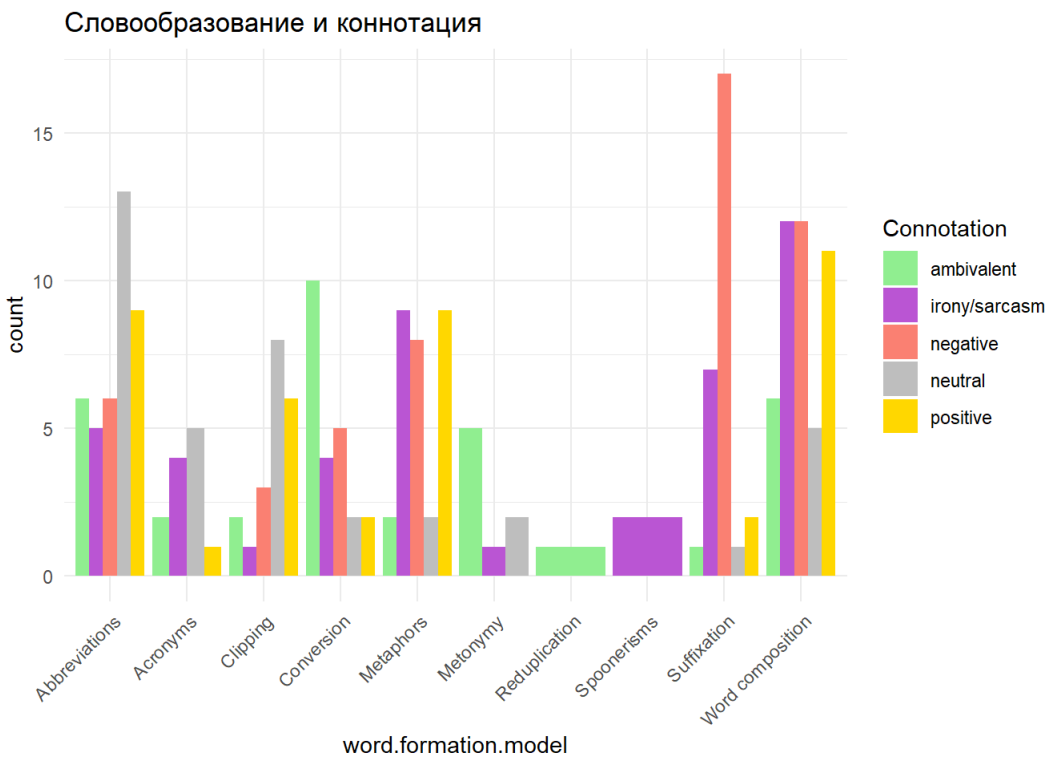
```
## # A tibble: 8 × 11
##   Genre.of.game.Where.it.appears Count_ambivalent Count_neutral Count_positive
##   <fct>                                    <int>         <int>          <int>
## 1 General                                     15            22             21
## 2 MOBA                                         8             3             10
## 3 FPS                                          2             1              4
## 4 MMORPG                                       7             7              3
## 5 Action                                       1             1              2
## 6 MMORPGB                                      0             1              0
## 7 RPG                                          1             3              0
## 8 Strategy                                     1             0              0
## # ℹ 7 more variables: `Count_irony/sarcasm` <int>, Count_negative <int>,
## #   Percentage_ambivalent <dbl>, Percentage_neutral <dbl>,
## #   Percentage_positive <dbl>, `Percentage_irony/sarcasm` <dbl>,
## #   Percentage_negative <dbl>
```

Ключевые наблюдения:

1. Больше всего наблюдений у группы **General**, сбалансированный состав, но много негатива, то есть в играх всех жанров преобладает негативная коннотация.
2. **MOBA** - второй по популярности жанр, имеет высокую долю негатива и иронии.
3. В **MMORPG** смешанные коннотации, а в FPS сильнее негатив.Strategy, RPG, Action не имеют негативную коннотацию.
4. В жанрах, требующих командного взаимодействия и быстрой реакции **(MOBA, FPS)*, негатив выражен сильнее.

Распределение моделей словообразования и коннотации

```
ggplot(df, aes(x = word.formation.model, fill = Connotation)) +
  geom_bar(position = "dodge") +
  scale_fill_manual(values = connotation_palette) +
  theme_minimal() +
  theme(axis.text.x = element_text(angle = 45, hjust = 1)) +
  ggtitle("Словообразование и коннотация")
```




```
formation_table <- create_connotation_table(df, word.formation.model) %>%
  arrange(desc(Count_negative))
formation_table
```

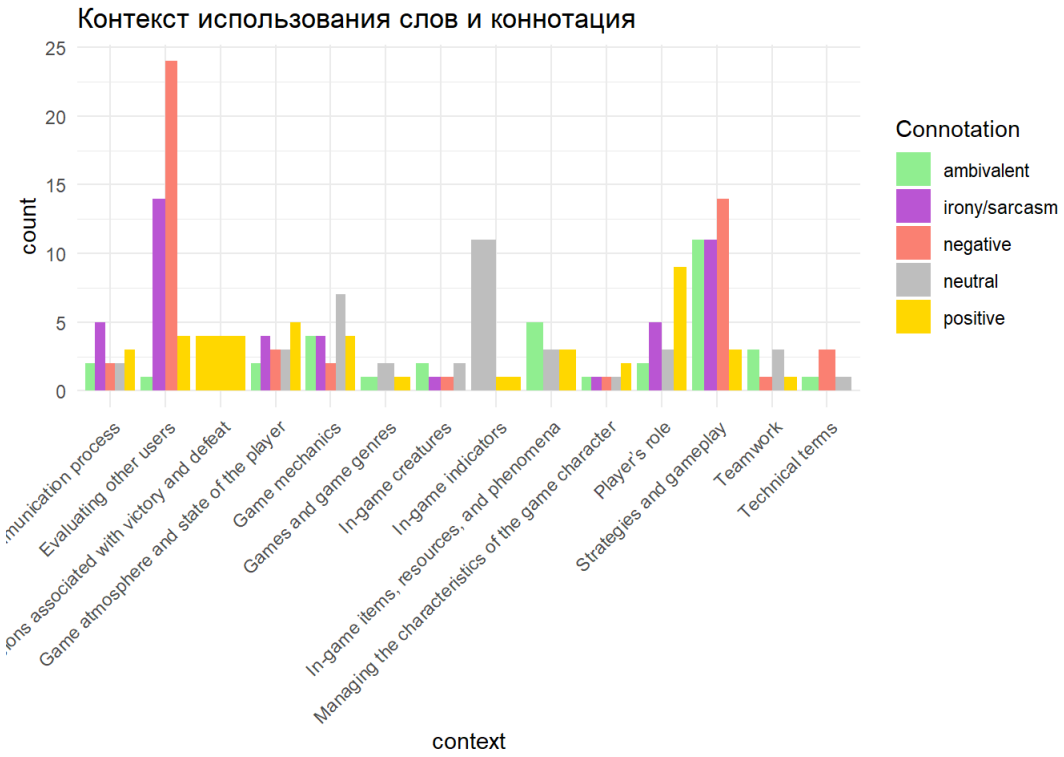
```
## # A tibble: 10 × 11
##    word.formation.model Count_ambivalent `Count_irony/sarcasm` Count_negative
##    <fct>                          <int>                 <int>          <int>
##  1 Suffixation                        1                     7             17
##  2 Word composition                   6                    12             12
##  3 Metaphors                          2                     9              8
##  4 Abbreviations                      6                     5              6
##  5 Conversion                        10                     4              5
##  6 Clipping                           2                     1              3
##  7 Acronyms                           2                     4              0
##  8 Metonymy                           5                     1              0
##  9 Reduplication                      1                     0              0
## 10 Spoonerisms                        0                     2              0
## # ℹ 7 more variables: Count_neutral <int>, Count_positive <int>,
## #   Percentage_ambivalent <dbl>, `Percentage_irony/sarcasm` <dbl>,
## #   Percentage_negative <dbl>, Percentage_neutral <dbl>,
## #   Percentage_positive <dbl>
```

Ключевые наблюдения:

1. **Суффиксация** часто используется для создания негативных ярлыков
2. **Словосложение** практически универсально: оно может обозначать как негатив с иронией, так и позитив.
3. **Метафоры** выражают эмоции, создают яркие и амбивалентные смыслы, тоже универсальны.
4. **Аббревиатуры**, **конверсия** и **акронимы** в большей степени нейтральны.

Распределение коннотации по контексту

```
ggplot(df, aes(x = context, fill = Connotation)) +
  geom_bar(position = "dodge") +
  scale_fill_manual(values = connotation_palette) +
  theme_minimal() +
  theme(axis.text.x = element_text(angle = 45, hjust = 1)) +
  ggtitle("Контекст использования слов и коннотация")
```




```
context_table <- create_connotation_table(df, context) %>%
  arrange(desc(Count_negative))
context_table
```

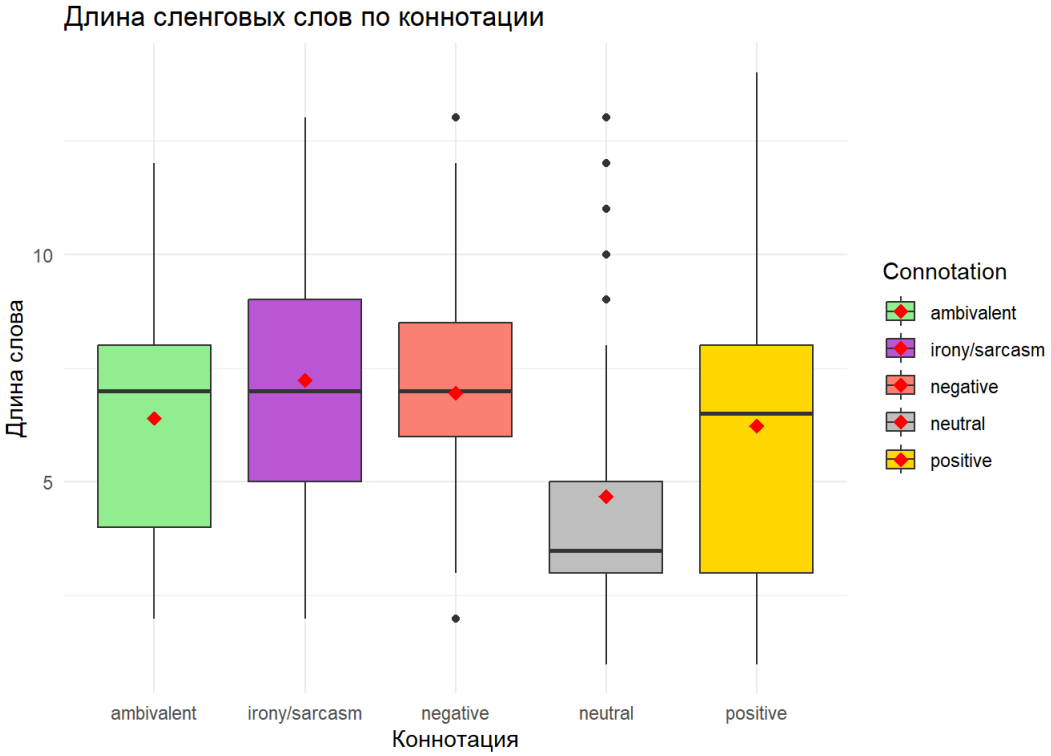
```
## # A tibble: 14 × 11
##    context   Count_ambivalent `Count_irony/sarcasm` Count_negative Count_neutral
##    <fct>              <int>               <int>          <int>         <int>
##  1 Evaluati…              1                  14             24             0
##  2 Strategi…             11                  11             14             0
##  3 Game atm…              2                   4              3             3
##  4 Technica…              1                   0              3             1
##  5 Communic…              2                   5              2             2
##  6 Game mec…              4                   4              2             7
##  7 In-game …              2                   1              1             2
##  8 Managing…              1                   1              1             1
##  9 Teamwork               3                   0              1             3
## 10 Expressi…              0                   0              0             0
## 11 Games an…              1                   0              0             2
## 12 In-game …              0                   0              0            11
## 13 In-game …              5                   0              0             3
## 14 Player's…              2                   5              0             3
## # ℹ 6 more variables: Count_positive <int>, Percentage_ambivalent <dbl>,
## #   `Percentage_irony/sarcasm` <dbl>, Percentage_negative <dbl>,
## #   Percentage_neutral <dbl>, Percentage_positive <dbl>
```

Ключевые наблюдения:

1. Контекст **“Evaluating other users”** максимально негативный — почти треть всех негативных выражений здесь.
2. **“Strategies and gameplay”** - смесь коннотаций, но преобладает негатив и нейтральность.
3. **“Game atmosphere”** / **“Communication”** умеренно сбалансированы.
4. **“Victory”** / **“Player’s role”** больше позитива.
5. Почти полностью нейтрален контекст **“In-game indicators”**.

График и таблица ниже иллюстрируют, как длина сленговых слов распределена в зависимости от коннотации.

```
ggplot(df, aes(x = Connotation, y = word_length, fill = Connotation)) +
  geom_boxplot() +
  scale_fill_manual(values = connotation_palette) +
  labs(title = "Длина сленговых слов по коннотации",
       x = "Коннотация", y = "Длина слова") +
  stat_summary(fun = mean, geom = "point", shape = 18, size = 3, color = "red") +
  theme_minimal()
```




```
df %>%
  mutate(word_length = nchar(slang.word)) %>%
  group_by(Connotation) %>%
  summarise(mean_length = mean(word_length))
```

```
## # A tibble: 5 × 2
##   Connotation   mean_length
##   <fct>               <dbl>
## 1 ambivalent           6.4
## 2 irony/sarcasm        7.24
## 3 negative             6.96
## 4 neutral              4.68
## 5 positive             6.22
```

```
# Основные статистики по длине слов
summary(df$word_length)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   1.000   4.000   6.000   6.373   8.000  14.000
```

Ключевые наблюдения:

1. Самые короткие слова у нейтральной коннотации, у них медиана длины заметно ниже (около 3).Узкий IQR показывает, что много выбросов.
2. Самые длинные в среднем у иронично-саркастической коннотации, у них медиана ближе к 8. У позитивной и иронично-саркастической коннотации распределение более широкое, встречаются как короткие, так и длинные слова.
3. У амбивалентной и негативной коннотации медиана около 7, но у амбивалентной разброс больше, чем у негативной
4. Нейтральные слова часто короче, возможно, потому что они более общие и часто употребимые.
5. Ироничные/саркастические и положительные самые длинные, что может отражать степень эмоциональной окраски или изобразительности таких слов.

Фреквентистская статистика

Перед проведение статистических тестов нужно проверить данные на нормальность.

```
shapiro.test(df$word_length)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  df$word_length
## W = 0.95702, p-value = 6.122e-06
```

```
shapiro.test(df$word_length[df$Connotation == "positive"])
```

```
##
##  Shapiro-Wilk normality test
##
## data:  df$word_length[df$Connotation == "positive"]
## W = 0.92167, p-value = 0.008695
```

```
shapiro.test(df$word_length[df$Connotation == "negative"])
```

```
##
##  Shapiro-Wilk normality test
##
## data:  df$word_length[df$Connotation == "negative"]
## W = 0.96489, p-value = 0.1348
```

```
shapiro.test(df$word_length[df$Connotation == "irony/sarcasm"])
```

```
##
##  Shapiro-Wilk normality test
##
## data:  df$word_length[df$Connotation == "irony/sarcasm"]
## W = 0.9682, p-value = 0.2488
```

```
shapiro.test(df$word_length[df$Connotation == "ambivalent"])
```

```
##
##  Shapiro-Wilk normality test
##
## data:  df$word_length[df$Connotation == "ambivalent"]
## W = 0.94786, p-value = 0.09743
```

```
shapiro.test(df$word_length[df$Connotation == "neutral"])
```

```
##
##  Shapiro-Wilk normality test
##
## data:  df$word_length[df$Connotation == "neutral"]
## W = 0.81042, p-value = 1.716e-05
```

Результаты проверки нормальности:

1. Распределение длины слов не является нормальным, потому что p-value = 6.1e-06 < 0.05.
2. Распределены **нормально**: слова негативной коннотации (p-value = 0.1348 > 0.05) слова иронично-саркастичной коннотации (p-value = 0.2488 > 0.05) слова амбивалентной коннотации (p-value = 0.09743 > 0.05)
3. Распределены **ненормально**: слова позитивной коннотации (p-value = 0.0087 < 0.05) слова нейтральной коннотации (p-value = 1.716e-05 < 0.05)

Можно провести t-test между negative и irony/sarcasm или negative и ambivalent коннотацией, так как они распределены нормально.

```
t.test(word_length ~ Connotation, data = df %>% filter(Connotation %in% c("negative", "irony/sarcasm")))
```

```
##
##  Welch Two Sample t-test
##
## data:  word_length by Connotation
## t = 0.48819, df = 88.215, p-value = 0.6266
## alternative hypothesis: true difference in means between group irony/sarcasm and group negative is not equal to 0
## 95 percent confidence interval:
##  -0.8709948  1.4383150
## sample estimates:
## mean in group irony/sarcasm      mean in group negative
##                    7.244444                    6.960784
```

```
t.test(word_length ~ Connotation, data = df %>% filter(Connotation %in% c("negative", "ambivalent")))
```

```
##
##  Welch Two Sample t-test
##
## data:  word_length by Connotation
## t = -0.94533, df = 71.263, p-value = 0.3477
## alternative hypothesis: true difference in means between group ambivalent and group negative is not equal to 0
## 95 percent confidence interval:
##  -1.7435425  0.6219739
## sample estimates:
## mean in group ambivalent   mean in group negative
##                 6.400000                 6.960784
```

```
t.test(word_length ~ Connotation, data = df %>% filter(Connotation %in% c("irony/sarcasm", "ambivalent")))
```

```
##
##  Welch Two Sample t-test
##
## data:  word_length by Connotation
## t = -1.3083, df = 75.96, p-value = 0.1947
## alternative hypothesis: true difference in means between group ambivalent and group irony/sarcasm is not equal to 0
## 95 percent confidence interval:
##  -2.1300170  0.4411281
## sample estimates:
##    mean in group ambivalent mean in group irony/sarcasm
##                    6.400000                    7.244444
```

Интерпретация результатов t-теста:

1. Высокое p-value (0.6266, 0.348, 0.195) говорит о том, что **нет статистически значимого различия** между средними длинами слов в группах “irony/sarcasm” и “negative”, “negative” и “ambivalent”, “irony/sarcasm” и “ambivalent”. 2.Доверительный интервал у всех групп 95% и включает 0, что также подтверждает отсутствие значимого различия.
2. Разница в средних небольшая, но показывает, что слова с ироничной-саркастической коннотацией длиннее слов с негативной на 0.28 и длинее слов с абивалентной на 0,8, а слова с негативной коннотацией длиннее слов с амбивалентной на 0.56.

Согласно результатам Welch t-теста, гипотеза о равенстве средних длины слов в группах “irony/sarcasm” и “negative” не отвергается. Это означает, что ироничные/саркастичные, негативные и амбивалентные термины не отличаются по длине.

Для проверки других видов коннотации будет использован тест Манна-Уитни (U-тест), так как он не требует нормальности распределения, в отличие от t-теста.

```
connotations <- unique(df$Connotation)
pairs <- combn(connotations, 2, simplify = FALSE)

for (pair in pairs) {
  group1 <- pair[1]
  group2 <- pair[2]

  cat("\nТест Манна-Уитни для:", group1, "vs", group2, "\n")
  result <- wilcox.test(
    word_length ~ Connotation,
    data = df %>% filter(Connotation %in% c(group1, group2))
  )
  print(result)
}
```

```
##
## Тест Манна-Уитни для: 1 vs 2
```

```
## Warning in wilcox.test.default(x = DATA[[1L]], y = DATA[[2L]], ...): не могу
## подсчитать точное p-значение при наличии повторяющихся наблюдений
```

```
##
##  Wilcoxon rank sum test with continuity correction
##
## data:  word_length by Connotation
## W = 654.5, p-value = 0.1961
## alternative hypothesis: true location shift is not equal to 0
##
##
## Тест Манна-Уитни для: 1 vs 3
##
##  Wilcoxon rank sum test with continuity correction
##
## data:  word_length by Connotation
## W = 791, p-value = 0.3702
## alternative hypothesis: true location shift is not equal to 0
##
##
## Тест Манна-Уитни для: 1 vs 5
```

```
## Warning in wilcox.test.default(x = DATA[[1L]], y = DATA[[2L]], ...): не могу
## подсчитать точное p-значение при наличии повторяющихся наблюдений
```

```
##
##  Wilcoxon rank sum test with continuity correction
##
## data:  word_length by Connotation
## W = 743.5, p-value = 0.6448
## alternative hypothesis: true location shift is not equal to 0
##
##
## Тест Манна-Уитни для: 1 vs 4
```

```
## Warning in wilcox.test.default(x = DATA[[1L]], y = DATA[[2L]], ...): не могу
## подсчитать точное p-значение при наличии повторяющихся наблюдений
```

```
##
##  Wilcoxon rank sum test with continuity correction
##
## data:  word_length by Connotation
## W = 919.5, p-value = 0.004541
## alternative hypothesis: true location shift is not equal to 0
##
##
## Тест Манна-Уитни для: 2 vs 3
##
##  Wilcoxon rank sum test with continuity correction
##
## data:  word_length by Connotation
## W = 1224, p-value = 0.5741
## alternative hypothesis: true location shift is not equal to 0
##
##
## Тест Манна-Уитни для: 2 vs 5
```

```
## Warning in wilcox.test.default(x = DATA[[1L]], y = DATA[[2L]], ...): не могу
## подсчитать точное p-значение при наличии повторяющихся наблюдений
```

```
##
##  Wilcoxon rank sum test with continuity correction
##
## data:  word_length by Connotation
## W = 1075.5, p-value = 0.1215
## alternative hypothesis: true location shift is not equal to 0
##
##
## Тест Манна-Уитни для: 2 vs 4
```

```
## Warning in wilcox.test.default(x = DATA[[1L]], y = DATA[[2L]], ...): не могу
## подсчитать точное p-значение при наличии повторяющихся наблюдений
```

```
##
##  Wilcoxon rank sum test with continuity correction
##
## data:  word_length by Connotation
## W = 1262.5, p-value = 0.0001784
## alternative hypothesis: true location shift is not equal to 0
##
##
## Тест Манна-Уитни для: 3 vs 5
##
##  Wilcoxon rank sum test with continuity correction
##
## data:  word_length by Connotation
## W = 1180, p-value = 0.1991
## alternative hypothesis: true location shift is not equal to 0
##
##
## Тест Манна-Уитни для: 3 vs 4
##
##  Wilcoxon rank sum test with continuity correction
##
## data:  word_length by Connotation
## W = 1440.5, p-value = 8.297e-05
## alternative hypothesis: true location shift is not equal to 0
##
##
## Тест Манна-Уитни для: 5 vs 4
```

```
## Warning in wilcox.test.default(x = DATA[[1L]], y = DATA[[2L]], ...): не могу
## подсчитать точное p-значение при наличии повторяющихся наблюдений
```

```
##
##  Wilcoxon rank sum test with continuity correction
##
## data:  word_length by Connotation
## W = 562.5, p-value = 0.04567
## alternative hypothesis: true location shift is not equal to 0
```

Интерпретация результатов тестов Манна-Уитн:

Для большинства пар p-value больше 0.05, то есть различия статистически незначимы. Однако были обнаружены статистически значимые различия ($p < 0.05$):

1. Слова в нейтральной группе и амбивалентной имеют значимо разную длину (0.0045 < 0.05).
2. Значимы различия и в группе ироничные VS нейтральные (0.00018 < 0.05).
3. Отрицательные слова имеют статистически иную длину, чем нейтральные (0.0000829 < 0.05).
4. Различия между позитивными и нейтральными словами также значимы (0.0457 < 0.05).

Самый интересный вывод — нейтральная коннотация выделяется по длине слов. Это может означать, что нейтральные термины короче/длиннее по сравнению с стальными. Остальные эмоциональные категории — положительные, отрицательные, ироничные и амбивалентные — не сильно различаются между собой по длине слов.

Чтобы точно определить, короче или длиннее нейтральные слова по сравнению с другими типами коннотации, проводится сравнение медиан.

```
df %>%
  group_by(Connotation) %>%
  summarise(
    Median = median(word_length),
    Mean = mean(word_length),
    SD = sd(word_length)
  ) %>%
  arrange(Median)
```

```
## # A tibble: 5 × 4
##   Connotation   Median  Mean    SD
##   <fct>          <dbl> <dbl> <dbl>
## 1 neutral          3.5  4.68  3.01
## 2 positive         6.5  6.22  3.50
## 3 ambivalent       7    6.4   2.75
## 4 irony/sarcasm    7    7.24  3.01
## 5 negative         7    6.96  2.64
```

Ключевые закономерности:

1. Нейтральные термины существенно короче всех остальных.
2. Позитивные термины занимают промежуточное положение.
3. Негативные, ироничные и амбивалентные термины самые длинные.

Подтверждается ранее полученными тестами:

Все сравнения нейтральных слов с другими типами коннотации были статистически значимыми ($p < 0.05$). Различия между негативными, ироничными и амбивалентными терминами незначимы.

Для проверки зависимости между категориальными переменными, проводится тест хи-квадрат, так как данных достаточно много и тест Фишера будет неэффектиным.

```
# связь коннотации и жанра
chisq.test(table(df$Connotation,
                 df$Genre.of.game.Where.it.appears))
```

```
## Warning in chisq.test(table(df$Connotation,
## df$Genre.of.game.Where.it.appears)): аппроксимация на основе хи-квадрат может
## быть неправильной
```

```
##
##  Pearson's Chi-squared test
##
## data:  table(df$Connotation, df$Genre.of.game.Where.it.appears)
## X-squared = 35.294, df = 28, p-value = 0.1615
```

```
# связь коннотации и контекста
chisq.test(table(df$context, df$Connotation))
```

```
## Warning in chisq.test(table(df$context, df$Connotation)): аппроксимация на
## основе хи-квадрат может быть неправильной
```

```
##
##  Pearson's Chi-squared test
##
## data:  table(df$context, df$Connotation)
## X-squared = 163.6, df = 52, p-value = 1.814e-13
```

```
# связь коннотации и модели словообразования
chisq.test(table(df$Connotation, df$word.formation.model))
```

```
## Warning in chisq.test(table(df$Connotation, df$word.formation.model)):
## аппроксимация на основе хи-квадрат может быть неправильной
```

```
##
##  Pearson's Chi-squared test
##
## data:  table(df$Connotation, df$word.formation.model)
## X-squared = 98.35, df = 36, p-value = 1.071e-07
```

```
# связь коннотации и части речи
chisq.test(table(df$Connotation, df$part.of.speech))
```

```
## Warning in chisq.test(table(df$Connotation, df$part.of.speech)): аппроксимация
## на основе хи-квадрат может быть неправильной
```

```
##
##  Pearson's Chi-squared test
##
## data:  table(df$Connotation, df$part.of.speech)
## X-squared = 28.797, df = 12, p-value = 0.004222
```

Интерпретация результатов теста хи-квадрат:

1. **Коннотация и жанр игры** $p > 0.05$, значит, нет статистически значимой связи, жанр игры не влияет на коннотацию слова.
2. **Коннотация и контекст** $p < 0.001$, значит, очень сильная статистическая связь, контекст использования игрового сленга существенно влияет на коннотацию.
3. **Коннотация и словообразовательная модель** $p < 0.001$, значит, существенная зависимость, модель словообразования влияет на коннотацию.

4. **Коннотация и часть речи** p = 0.004222 < 0.05, значит, мы отвергаем H0, есть статистически значимая связь, коннотация слова зависит от части речи.

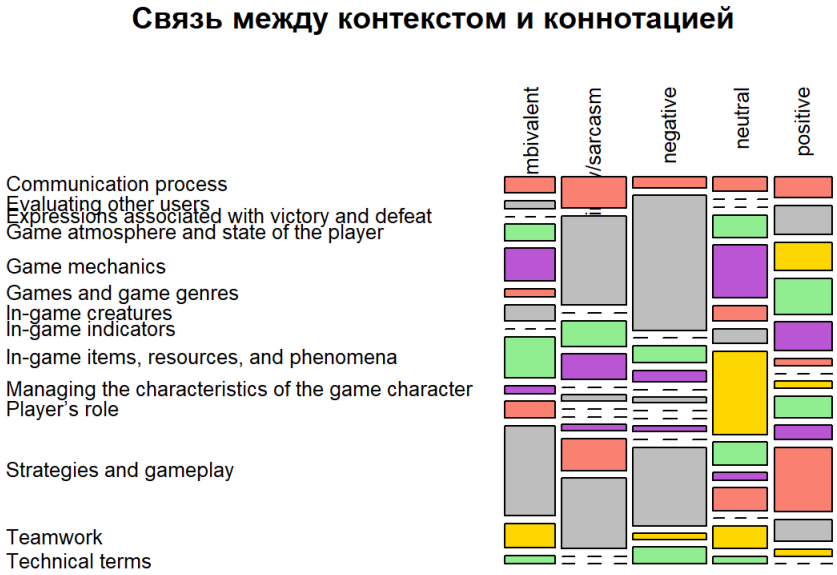
Корреляционный анализ

В дополнение к хи-квадрат тесту используется Cramér's V, показать силу зависимости переменных Connotation и context.

```
tbl <- table(df$Connotation, df$context)
assocstats(tbl)
```

```
##                     X^2 df   P(> X^2)
## Likelihood Ratio 172.82 52 6.9944e-15
## Pearson          163.60 52 1.8141e-13
##
## Phi-Coefficient   : NA
## Contingency Coeff.: 0.663
## Cramer's V        : 0.442
```

```
mosaicplot(tbl,
         main = "Связь между контекстом и коннотацией",
         color = connotation_palette,
         shade = FALSE,
         las = 2,
         cex.axis = 0.8)
```

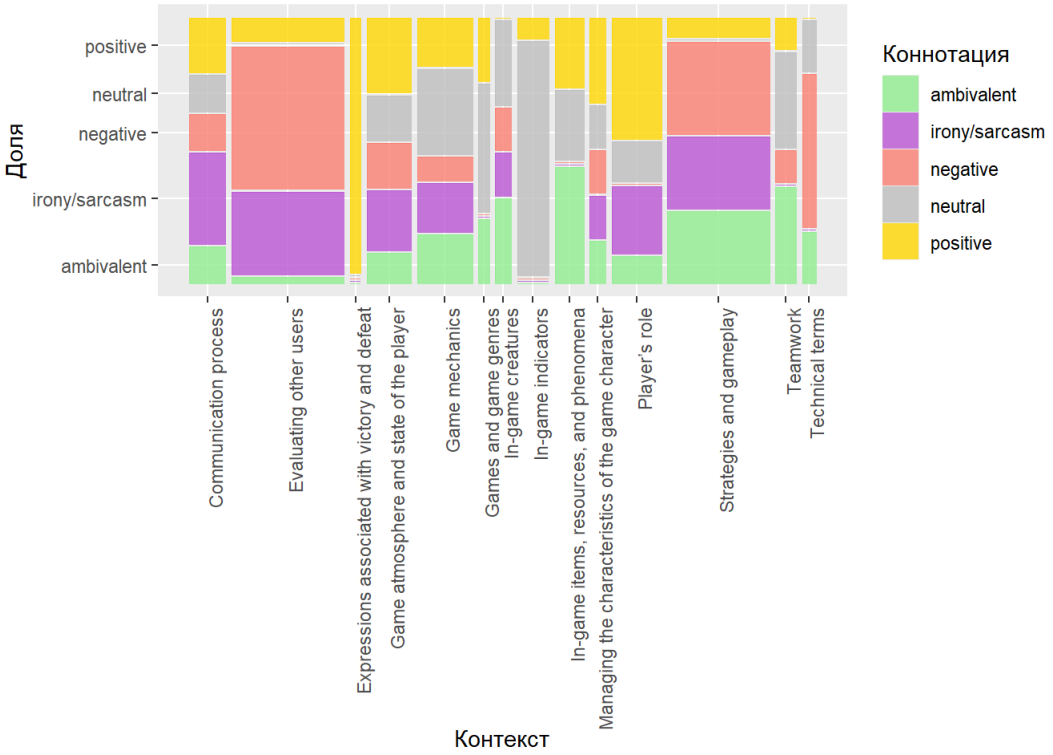



```
ggplot(data = df) +
  geom_mosaic(aes(weight = 1, x = product(context), fill = Connotation)) +
  labs(x = "Контекст", y = "Доля", fill = "Коннотация") +
  theme(axis.text.x = element_text(angle = 90, hjust = 1))+
  scale_fill_manual(values = connotation_palette)
```

```
## Warning: The `scale_name` argument of `continuous_scale()` is deprecated as of ggplot2
## 3.5.0.
## This warning is displayed once every 8 hours.
## Call `lifecycle::last_lifecycle_warnings()` to see where this warning was
## generated.
```

```
## Warning: The `trans` argument of `continuous_scale()` is deprecated as of ggplot2 3.5.0.
## ℹ Please use the `transform` argument instead.
## This warning is displayed once every 8 hours.
## Call `lifecycle::last_lifecycle_warnings()` to see where this warning was
## generated.
```

```
## Warning: `unite_()` was deprecated in tidyr 1.2.0.
## ℹ Please use `unite()` instead.
## ℹ The deprecated feature was likely used in the ggmosaic package.
##   Please report the issue at <https://github.com/haleyjeppson/ggmosaic>.
## This warning is displayed once every 8 hours.
## Call `lifecycle::last_lifecycle_warnings()` to see where this warning was
## generated.
```



Вывод:

Распределения коннотаций заметно различаются между контекстами. Например:

- **“Evaluating other users”** — явно преобладает ирония и негатив.
- **“Victory / Player’s role”** — заметно больше позитивных слов.
- **“Game mechanics” и “In-game indicators”** — много нейтральных слов.

Результат Cramér’s V составил V = 0.442, что соответствует умеренной статистически значимой связи между контекстом и эмоциональной окраской сленговых слов. Это подтверждает, что контекст влияет на коннотацию слов.

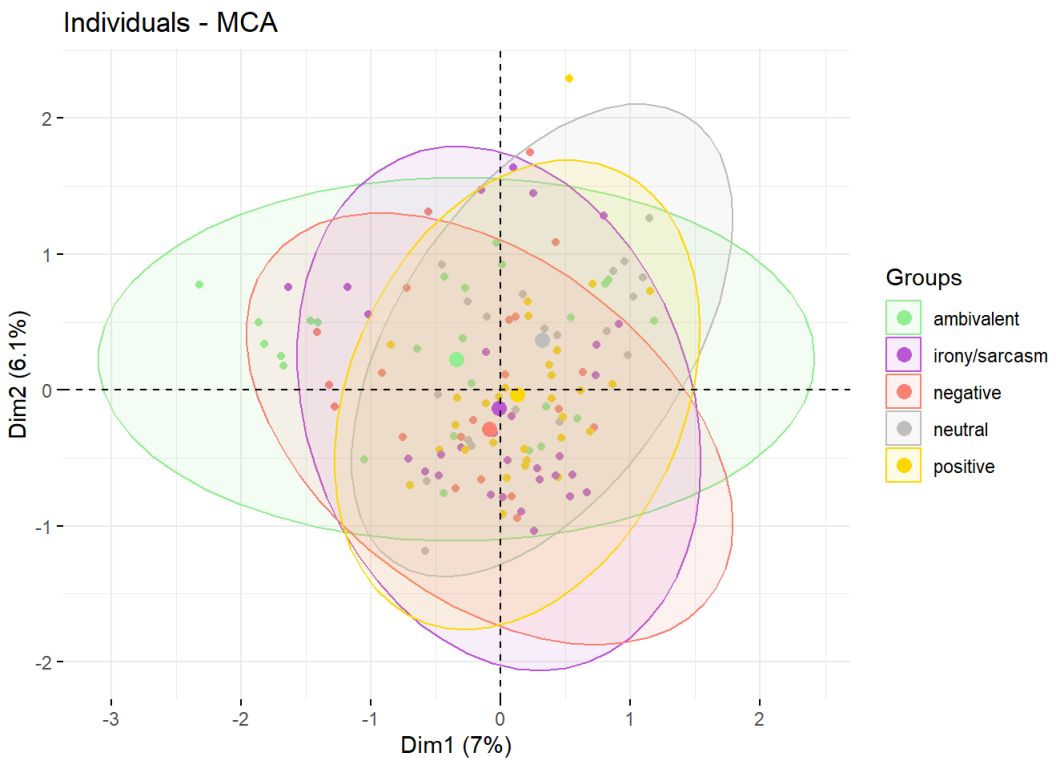
Методы сокращения размерности

Чтобы исследовать структуру категориальных признаков (жанр, контекст, часть речи, модель словообразования), был проведён множественный корреспондентный анализ (MCA). Результирующая проекция позволила визуализировать взаиморасположение категорий и выявить близость между признаками.

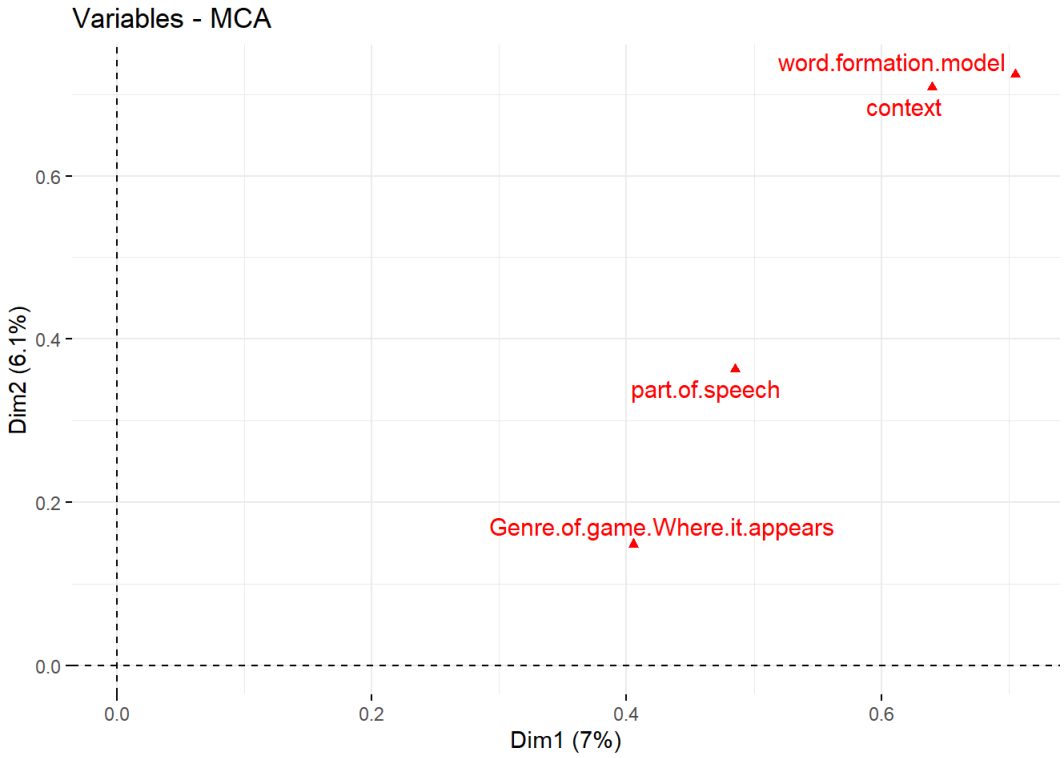
```
gaming_slang_mca <- df %>%
  select(part.of.speech, context, Genre.of.game.Where.it.appears,
         word.formation.model) %>%
  mutate(across(everything(), as.factor))

mca_result <- MCA(gaming_slang_mca, ncp = 5, graph = FALSE)

fviz_mca_ind(mca_result,
             label = "none",
             habillage = df$Connotation,
             addEllipses = TRUE,
             repel = TRUE,
             palette = connotation_palette,
             ggtheme = theme_minimal())
```




```
fviz_mca_var(mca_result, choice = "mca.cor", repel = TRUE)
```



Интерпретация анализа соответствий:

График Variables-MCA отображает вклад переменных в мультикорреспондентный анализ (MCA). Он показывает, какие переменные сильнее всего связаны с главными компонентами (Dim1 и Dim2).

1. Самый сильный вклад имеют переменные word.formation.model и context, то есть, именно они наиболее информативны для различения значений признаков, в том числе и целевой переменной Connotation.
2. Умеренный вклад вносят переменные part.of.speech и Genre.of.game.Where.it.appears. Они могут дополнять различие между наблюдениями, но не доминируют в объяснении дисперсии.
3. Dim1 (7.2%) и Dim2 (6.3%) объясняют всего 13.5% дисперсии, и это мало, для достижения 50% объясненной дисперсии требуется 10 осей.

Для последующего анализа стоит акцентировать внимание на word.formation.model и context.

Регрессионный анализ

В рамках данного проекта была разработана модель для предсказания коннотации игрового сленга.

Цель модели: разработать классификатор, способный автоматически определять коннотацию игрового сленга на основе текстовых

признаков, включая контекст использования, жанр игры, часть речи и модель словообразования.

Модель должна помочь: автоматически анализировать тональность игровых терминов, улучшить модерацию чатов в онлайн-играх, исследовать закономерности использования сленга.

Задачи модели:

1. Предобработка данных: текстовая очистка, объединение признаков, обработка пропусков.
2. Балансировка данных: Oversampling и использование upSample() для устранения дисбаланса классов.
3. Векторизация текста (TF-IDF): токенизация, разбиение текста на слова и биграммы, построение словаря, отбор частотных терминов, преобразование в матрицу.
4. Обучение модели: логистическая регрессия (многоклассовая классификация) с регуляризацией (Elastic Net).
5. Оценка качества.

```r
# 1. Предобработка данных

# Очистка текста
clean_text <- function(text) {
  text %>%
    tolower() %>%
    stringi::stri_trans_general("Latin-ASCII") %>%
    str_replace_all("[^a-z0-9!? ]", " ") %>%
    str_squish()
}

# Объединение признаков
df$combined_text <- df %>%
  mutate(
    combined_text = paste(
      clean_text(Sentence),
      clean_text(part.of.speech),
      clean_text(Genre.of.game.Where.it.appears),
      clean_text(context),
      clean_text(word.formation.model)
    ) %>% str_squish()
  ) %>%
  pull(combined_text)

# NA
df_clean <- df %>% filter(!is.na(combined_text), !is.na(Connotation))

# 2. Балансировка данных
# Oversampling
set.seed(123)
upsampled_data <- upSample(
  x = df_clean["combined_text"],
  y = df_clean$Connotation,
  yname = "Connotation"
)

# Разделение данных
set.seed(123)
train_index <- createDataPartition(upsampled_data$Connotation, p = 0.8, list = FALSE)
train_data <- upsampled_data[train_index, ]
test_data <- upsampled_data[-train_index, ]

#3. Векторизация текста (TF-IDF)
tokens <- itoken(train_data$combined_text, tokenizer = word_tokenizer, progressbar = FALSE)
vocab <- create_vocabulary(tokens, ngram = c(1, 2)) %>%
  prune_vocabulary(term_count_min = 5)  # фильтрация редких

vectorizer <- vocab_vectorizer(vocab)

dtm_train <- create_dtm(tokens, vectorizer)
tfidf <- TfIdf$new()
dtm_train_tfidf <- fit_transform(dtm_train, tfidf)

# 4. Обучение модели
# Преобразование тестовой выборки
test_tokens <- itoken(test_data$combined_text,
                      tokenizer = word_tokenizer,
                      progressbar = FALSE)


dtm_test <- create_dtm(test_tokens, vectorizer)
dtm_test_tfidf <- transform(dtm_test, tfidf)

# Модель
model <- cv.glmnet(
  x = dtm_train_tfidf,
  y = train_data$Connotation,
  family = "multinomial",
  type.measure = "class",
  alpha = 0.5
)

# 5. Предсказания и оценка
preds <- predict(model, dtm_test_tfidf, s = "lambda.min", type = "class")
confusionMatrix(as.factor(preds), test_data$Connotation)
```

```
## Confusion Matrix and Statistics
##
##               Reference
## Prediction      ambivalent irony/sarcasm negative neutral positive
##   ambivalent            6             1        1       0        1
##   irony/sarcasm         1             8        3       2        3
##   negative              0             0        6       0        0
##   neutral               2             0        0       7        0
##   positive              1             1        0       1        6
##
## Overall Statistics
##
##                Accuracy : 0.66
##                  95% CI : (0.5123, 0.7879)
##     No Information Rate : 0.2
##     P-Value [Acc > NIR] : 2.173e-12
##
##                   Kappa : 0.575
##
##  Mcnemar's Test P-Value : NA
##
## Statistics by Class:
##
##                      Class: ambivalent Class: irony/sarcasm Class: negative
## Sensitivity                     0.6000              0.8000         0.6000
## Specificity                     0.9250              0.7750         1.0000
## Pos Pred Value                  0.6667              0.4706         1.0000
## Neg Pred Value                  0.9024              0.9394         0.9091
## Prevalence                      0.2000              0.2000         0.2000
## Detection Rate                  0.1200              0.1600         0.1200
## Detection Prevalence            0.1800              0.3400         0.1200
## Balanced Accuracy               0.7625              0.7875         0.8000
##                      Class: neutral Class: positive
## Sensitivity                  0.7000         0.6000
## Specificity                  0.9500         0.9250
## Pos Pred Value               0.7778         0.6667
## Neg Pred Value               0.9268         0.9024
## Prevalence                   0.2000         0.2000
## Detection Rate               0.1400         0.1200
## Detection Prevalence         0.1800         0.1800
## Balanced Accuracy            0.8250         0.7625
```

Несмотря на то, что MCA выделил context и word.formation.model как наиболее информативные признаки, сокращение модели до этих переменных привело к снижению качества классификации (точность упала с 66% до 60%, Kappa — с 0.575 до 0.5). Поэтому в финальной модели были сохранены все текстовые признаки (context, part.of.speech, word.formation.model, genre, sentence), что позволило достичь лучшего баланса между интерпретируемостью и точностью.

Результаты:

- Accuracy = 66% - модель показывает точность выше случайного угадывания
- Kappa = 0.575 — умеренное согласие модели с истинными метками.
- F1 ≈ 0.67

Анализ метрик:

1. **Для класса ambivalent:** модель находит 60% реальных ambivalent случаев.
2. **Для класса irony/sarcasm:** высокая Sensitivity, но слабая Precision: модель часто предсказывает иронию ошибочно.
3. **Для класса negative:** отличная Precision ((нет ложных срабатываний), но малая Sensitivity.
4. **Для класса neutral:** хороший баланс.
5. **Для класса positive:** слабее neutral, но лучше irony.

Наблюдения:

1. Классы “negative” и “neutral” имеют наименьшее количество ошибок
2. Классы “irony/sarcasm” и “positive” часто путаются между собой и с другими классами
3. Класс “negative” не имеет ложных срабатываний (все 6 предсказаний верны)
4. Модель хорошо отличает negative и neutral.

Вывод:

Модель демонстрирует хороший базовый уровень классификации, но требует доработки в области различения близких по смыслу классов (особенно irony/sarcasm и positive). Наибольшие проблемы связаны с многозначностью некоторых игровых терминов и дисбалансом в классах.

```
# Функция для предсказания конно т ации
predict_connotation <- function(sentence, model, vectorizer, tfidf) {
  token <- itoken(sentence,
                  preprocessor = tolower,
                  tokenizer = word_tokenizer,
                  progressbar = FALSE)

  dtm <- create_dtm(token, vectorizer)
  dtm_tfidf <- transform(dtm, tfidf)
  prediction <- predict(model, dtm_tfidf, s = "lambda.min", type = "class")
  return(as.character(prediction))
}
```

```
# Предсказание конно т ации предложения
predict_connotation("Press E to attack.", model, vectorizer, tfidf)
```

```
## [1] "neutral"
```

```
predict_connotation("That player is a total feeder.", model, vectorizer, tfidf)
```

```
## [1] "negative"
```

```
predict_connotation("You are a top banger.", model, vectorizer, tfidf)
```

```
## [1] "ambivalent"
```

```
predict_connotation("Our team is a total clown fiesta.", model, vectorizer, tfidf)
```

```
## [1] "negative"
```

```
predict_connotation("It is GG, guys.", model, vectorizer, tfidf)
```

```
## [1] "irony/sarcasm"
```

```
predict_connotation("My new weapon imba", model, vectorizer, tfidf)
```

```
## [1] "positive"
```