**Sprawozdanie na przedmiot wykład specjalizujący „Klasyfikacja za pomocą k-Najbliższych sąsiadów”**

Informatyka ogólna grupa 3

Wykonał: Murawski Sebastian

Index: 155082

Sprawozdanie wersja 1.0

Uniwersytet Warmińsko-Mazurski

Wydział Matematyki i Informatyki (WMiI)

Olsztyn, 8.12.2022r.

Spis treści

[**1.** **Wstęp.** 3](#_Toc121938353)

[**2.** **Opis atrybutów w bazie danych:** 4](#_Toc121938354)

[**4.** **Wykonać badania eksploracyjne (EDA, ang. exploratory data analysis) zestawu danych.** 6](#_Toc121938355)

[1.1 Struktura danych 6](#_Toc121938356)

[**5.** **Wykonać badania eksploracyjne (EDA, ang. exploratory data analysis) zestawu danych.** 7](#_Toc121938357)

[1.1 Pozycja pozioma, licząc piksele od lewej krawędzi obrazu, środka najmniejsze prostokątne pudełko, które można narysować ze wszystkimi „włączonymi” pikselami wewnątrz pudełka. 7](#_Toc121938358)

[1.2 Pozycja pionowa powyższego pola, licząc piksele od dołu. 8](#_Toc121938359)

[1.3 Szerokość pudełka w pikselach. 9](#_Toc121938360)

[1.4 Wysokość pudełka w pikselach. 10](#_Toc121938361)

[1.5 Całkowita liczba „włączonych” pikseli na obrazie. 11](#_Toc121938362)

[1.6 Średnia pozioma pozycja wszystkich „włączonych” pikseli względem środka pudełka i podzielona po szerokości pudełka. 12](#_Toc121938363)

[1.7 Średnia pionowa pozycja wszystkich „włączonych” pikseli względem środka pudełka i podzielona po wysokości pudełka. 13](#_Toc121938364)

[1.8 Średnia kwadratowa wartość odległości pikseli w poziomie zmierzona w powyżej X6. 14](#_Toc121938365)

[1.9 Średnia wartość odległości pikseli w pionie zmierzona w powyżej X7. 15](#_Toc121938366)

[1.10 Średni iloczyn odległości poziomej i pionowej dla każdego „włączonego” piksela zgodnie z pomiarem w powyżej X6 i X7. 16](#_Toc121938367)

[1.11 Średnia wartość kwadratu odległości poziomej pomnożona przez odległości pionowe dla każdego. Mierzy to korelację wariancji poziomej z pozycją pionową. 17](#_Toc121938368)

[1.12 Średnia wartość kwadratu odległości pionowej pomnożona przez odległość poziomą dla każdego z nich „włączony” piksel. Mierzy to korelację wariancji pionowej z pozycją poziomą. 18](#_Toc121938369)

[1.13 Średnia liczba napotkanych krawędzi podczas wykonywania systematycznych skanów od lewej do prawej we wszystkich pozycjach pionowych w pudełku. 19](#_Toc121938370)

[1.14 Suma pionowych położeń napotkanych krawędzi mierzonych w powyżej X13. 20](#_Toc121938371)

[1.15 Średnia liczba krawędzi napotkanych podczas systematycznego skanowania obrazu z od dołu do góry nad wszystkimi poziomymi pozycjami w pudełku. 21](#_Toc121938372)

[1.16 Suma poziomych położeń napotkanych krawędzi mierzonych w powyżej X15. 22](#_Toc121938373)

[**6.** **Wyznaczyć cechę decyzyjną i cechy objaśniające.** 23](#_Toc121938374)

[**7.** **Stworzyć klasyfikator używający algorytmu KNN.** 23](#_Toc121938375)

[**8.** **Przeprowadzić klasyfikację dla różnych wartości K. Dla każdej wartości podać miarę accuracy (dokładności) klasyfikatora.** 24](#_Toc121938376)

[**9.** **Oszacować miary oceny klasyfikacji: Accuracy, Recall (Sensitivity), Specificity, false positive rate FPR, false discovery rate FDR, precyzję pozytywną, precyzję negatywną, F1– i Fbeta-score, współczynnik korelacji Matthews’a.** 26](#_Toc121938377)

# **Wstęp.**

Tematem sprawozdania jest praca z różnymi bazami danych w celu wykonania badania eksploracyjnego w języku R.

Podczas wykonywania zadania, zostały użyte następujące narzędzia:

* IDE – Rstudio – narzędzie ułatwiające pracę z językiem R. RStudio to zintegrowane środowisko programistyczne dla R , języka programowania do obliczeń statystycznych i grafiki.
* R – interpretowany język programowania oraz środowisko do obliczeń statystycznych i wizualizacji wyników. R rozprowadzany jest w postaci kodu źródłowego oraz w postaci binarnej wraz z wieloma dystrybucjami GNU/Linuksa. Dostępna jest także wersja dla Microsoft Windows i Mac OS.
* Baza danych „Letter Recognition”
* Dodatkowe biblioteki takie jak:
  + Tidyverse — pakiety R do analizy danych, w tym ggplot2 , dplyr , tidyr i purrr
  + Caret -(skrót od Classification And RE gression Training ) to zestaw funkcji , które mają na celu usprawnienie procesu tworzenia modeli predykcyjnych .
  + ggplot2 - to pakiet wizualizacji danych typu open source dla statystycznego języka programowania R .Służy jako zamiennik podstawowej grafiki w R i zawiera szereg ustawień domyślnych do wyświetlania.
  + Gmodels
  + Class
  + pROC

Zadania które trzeba było wykonać w ramach zajęć:

1. Wczytać zestaw danych, który odpowiada ostatniej cyfrze z numeru albumu, do systemu R.
2. Wykonać badania eksploracyjne (EDA, ang. exploratory data analysis) zestawu danych.
3. Wyznaczyć cechę decyzyjną i cechy objaśniające.
4. Stworzyć klasyfikator używający algorytmu KNN.
5. Przeprowadzić klasyfikację dla różnych wartości K. Dla każdej wartości podać miarę accuracy (dokładności) klasyfikatora.
6. Oszacować miary oceny klasyfikacji: Accuracy, Recall (Sensitivity), Specificity, false positive rate FPR, false discovery rate FDR, precyzję pozytywną, precyzję negatywną, F1– i Fbeta-score, współczynnik korelacji Matthews’a.
7. Napisać sprawozdanie i wysłać pod adres matematyka@gmx.com

Zbiór danych składa się z 20 000 punktów danych i 17 zmiennych.

# **Opis atrybutów w bazie danych:**

Poniżej podano 17 atrybutów,

T: Litery

X1: Pozycja pozioma, licząc piksele od lewej krawędzi obrazu, środka najmniejsze prostokątne pudełko, które można narysować ze wszystkimi „włączonymi” pikselami wewnątrz pudełka.

X2: Pozycja pionowa powyższego pola, licząc piksele od dołu.

X3: Szerokość pudełka w pikselach.

X4: Wysokość pudełka w pikselach.

X5: Całkowita liczba „włączonych” pikseli na obrazie.

X6: Średnia pozioma pozycja wszystkich „włączonych” pikseli względem środka pudełka i podzielona po szerokości pudełka.

X7: Średnia pionowa pozycja wszystkich „włączonych” pikseli względem środka pudełka i podzielona po wysokości pudełka.

X8: Średnia kwadratowa wartość odległości pikseli w poziomie zmierzona w powyżej X6.

X9: Średnia wartość odległości pikseli w pionie zmierzona w powyżej X7.

X10: Średni iloczyn odległości poziomej i pionowej dla każdego „włączonego” piksela zgodnie z pomiarem w powyżej X6 i X7.

X11: Średnia wartość kwadratu odległości poziomej pomnożona przez odległości pionowe dla każdego. Mierzy to korelację wariancji poziomej z pozycją pionową.

X12: Średnia wartość kwadratu odległości pionowej pomnożona przez odległość poziomą dla każdego z nich „włączony” piksel. Mierzy to korelację wariancji pionowej z pozycją poziomą.

X13: Średnia liczba napotkanych krawędzi podczas wykonywania systematycznych skanów od lewej do prawej we wszystkich pozycjach pionowych w pudełku.

X14: Suma pionowych położeń napotkanych krawędzi mierzonych w powyżej X13.

X15: Średnia liczba krawędzi napotkanych podczas systematycznego skanowania obrazu z od dołu do góry nad wszystkimi poziomymi pozycjami w pudełku.

X16: Suma poziomych położeń napotkanych krawędzi mierzonych w powyżej X15.

Ze wstępnej analizy w R stało się jasne, że dane nie mają żadnych wartości null ani brakujące wartości.

1. **Wczytać zestaw danych, który odpowiada ostatniej cyfrze z numeru albumu, do systemu R.**

Na samym początku trzeba wczytać dane do programu oraz załadować potrzebne biblioteki.

library**(**ggplot2**)**

library**(**pROC**)**

library**(**gmodels**)**

library**(**class**)**

library**(**caret**)**

# 1. Wczytać zestaw danych, który odpowiada ostatniej cyfrze z numeru albumu, do systemu R.

mydata **<-** read.csv**(**"dane/letter-recognition.data", stringsAsFactors **=** **FALSE**, sep**=**","**)**

Dla wygody dodałem nazwy odpowiadającej każdej z kolumn.

mydata **<-** read.csv**(**file**=**"dane/letter-recognition.data", sep**=**","**)**

colnames**(**mydata**)** **<-** c**(**"T","X1","X2","X3","X4","X5","X6","X7","X8","X9","X10","X11","X12","X13","X14","X15","X16"**)**

head**(**mydata**)**

str**(**mydata**)**

dim**(**mydata**)**

head**(**mydata**)**

summary**(**mydata**)**

A następnie wyświetliłem w celu sprawdzenia

'data.frame'**:** 19999 obs. of 17 variables**:**

**$** T **:** chr "I" "D" "N" "G" ...

**$** X2 **:** int 5 4 7 2 4 4 1 2 11 3 ...

**$** X8 **:** int 12 11 11 1 11 2 1 2 15 9 ...

**$** X3 **:** int 3 6 6 3 5 5 3 4 13 5 ...

**$** X5 **:** int 7 8 6 1 8 4 2 4 9 7 ...

**$** X1 **:** int 2 6 3 1 3 4 1 2 7 4 ...

**$** X8.1**:** int 10 10 5 8 8 8 8 10 13 8 ...

**$** X13 **:** int 5 6 9 6 8 7 2 6 2 7 ...

**$** X0 **:** int 5 2 4 6 6 6 2 2 6 3 ...

**$** X6 **:** int 4 6 6 6 9 6 2 6 2 8 ...

**$** X6.1**:** int 13 10 4 6 5 7 8 12 12 5 ...

**$** X10 **:** int 3 3 4 5 6 6 2 4 1 6 ...

**$** X8.2**:** int 9 7 10 9 6 6 8 8 9 8 ...

**$** X0.1**:** int 2 3 6 1 0 2 1 1 8 2 ...

**$** X8.3**:** int 8 7 10 7 8 8 6 6 1 8 ...

**$** X0.2**:** int 4 3 2 5 9 7 2 1 1 6 ...

**$** X8.4**:** int 10 9 8 10 7 10 7 7 8 7 ...

1. **Wykonać badania eksploracyjne (EDA, ang. exploratory data analysis) zestawu danych.**

## Struktura danych

W celu sprawdzenia struktury danych oraz typów danych użyłem komendy str (dataset).

Wynikiem tej operacji jest wydruk w konsoli przydatnych informacji takie jak:

* Nazwa kolumn, wymiar danych, typ danych oraz przykładowe wartości które się w nich znajdują.

Wydruk z konsoli:

'data.frame'**:** 19999 obs. of 17 variables**:**

**$** T **:** chr "I" "D" "N" "G" ...

**$** X2 **:** int 5 4 7 2 4 4 1 2 11 3 ...

**$** X8 **:** int 12 11 11 1 11 2 1 2 15 9 ...

**$** X3 **:** int 3 6 6 3 5 5 3 4 13 5 ...

**$** X5 **:** int 7 8 6 1 8 4 2 4 9 7 ...

**$** X1 **:** int 2 6 3 1 3 4 1 2 7 4 ...

**$** X8.1**:** int 10 10 5 8 8 8 8 10 13 8 ...

**$** X13 **:** int 5 6 9 6 8 7 2 6 2 7 ...

**$** X0 **:** int 5 2 4 6 6 6 2 2 6 3 ...

**$** X6 **:** int 4 6 6 6 9 6 2 6 2 8 ...

**$** X6.1**:** int 13 10 4 6 5 7 8 12 12 5 ...

**$** X10 **:** int 3 3 4 5 6 6 2 4 1 6 ...

**$** X8.2**:** int 9 7 10 9 6 6 8 8 9 8 ...

**$** X0.1**:** int 2 3 6 1 0 2 1 1 8 2 ...

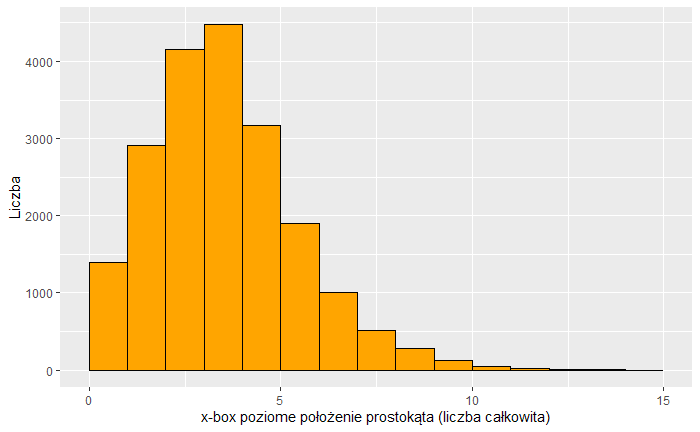
**$** X8.3**:** int 8 7 10 7 8 8 6 6 1 8 ...

**$** X0.2**:** int 4 3 2 5 9 7 2 1 1 6 ...

**$** X8.4**:** int 10 9 8 10 7 10 7 7 8 7 ...

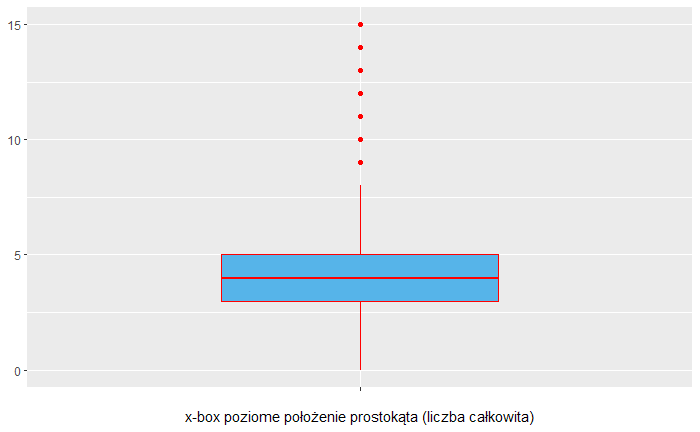
# **Wykonać badania eksploracyjne (EDA, ang. exploratory data analysis) zestawu danych.**

## Pozycja pozioma, licząc piksele od lewej krawędzi obrazu, środka najmniejsze prostokątne pudełko, które można narysować ze wszystkimi „włączonymi” pikselami wewnątrz pudełka.



**Kod:**

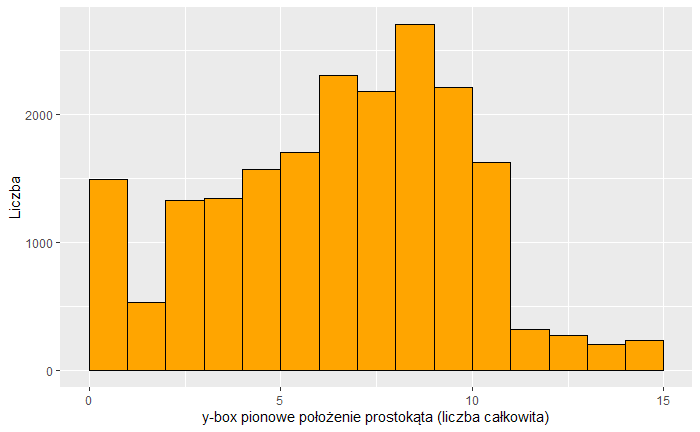
ggplot**(**data**=**mydata,aes**(**X1**))** **+** geom\_histogram**(**breaks**=**seq**(**0,15,by**=**1**)**,color**=**"black",fill**=**"orange"**)** **+** labs**(**x**=**"x-box poziome położenie prostokąta (liczba całkowita)",y**=**"Liczba"**)**



**Kod:**

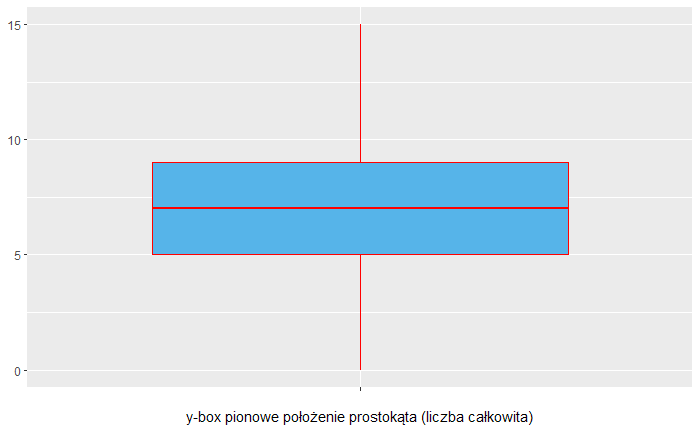
ggplot**(**mydata,aes**(**x**=**"",y**=**X1**))** **+** geom\_boxplot**(**fill **=** "#56B4E9", color **=** "red",width**=**0.5**)** **+** theme**(**axis.title.y**=**element\_blank**())** **+** labs**(**x**=**"x-box poziome położenie prostokąta (liczba całkowita)"**)**

## Pozycja pionowa powyższego pola, licząc piksele od dołu.



**Kod:**

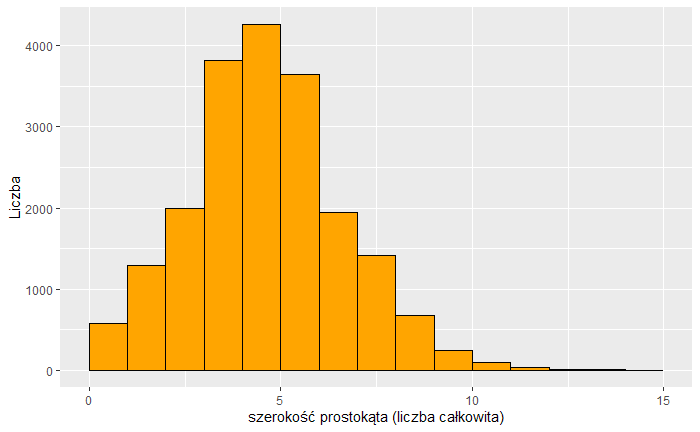
ggplot**(**data**=**mydata,aes**(**X2**))** **+** geom\_histogram**(**breaks**=**seq**(**0,15,by**=**1**)**,color**=**"black",fill**=**"orange"**)** **+** labs**(**x**=**"y-box pionowe położenie prostokąta (liczba całkowita)",y**=**"Liczba"**)**



**Kod:**

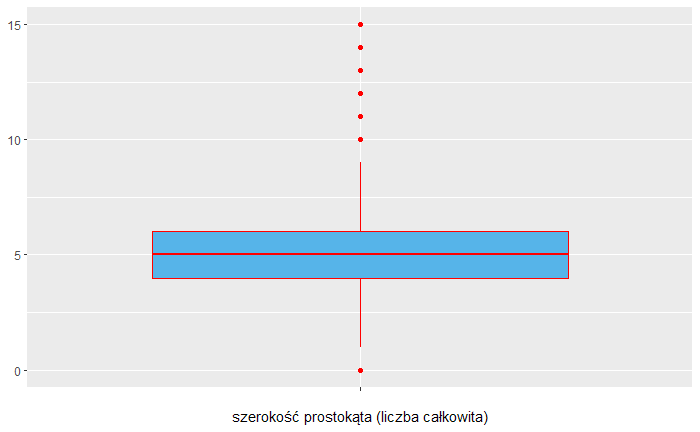
ggplot**(**mydata,aes**(**x**=**"",y**=**X2**))** **+** geom\_boxplot**(**fill **=** "#56B4E9", color **=** "red"**)+** theme**(**axis.title.y**=**element\_blank**())** **+** labs**(**x**=**"y-box pionowe położenie prostokąta (liczba całkowita)"**)**

## Szerokość pudełka w pikselach.



**Kod:**

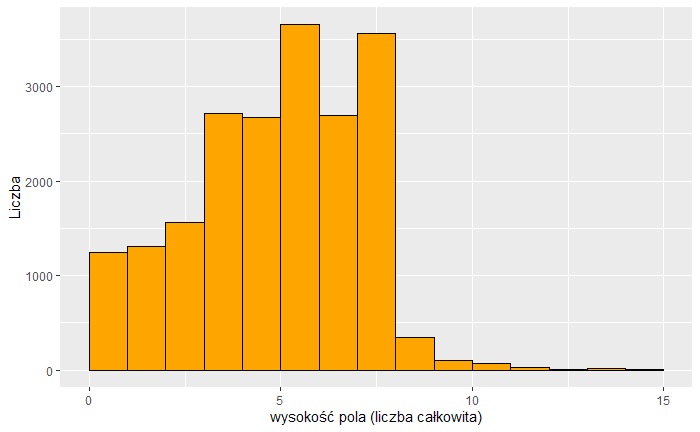
ggplot**(**data**=**mydata,aes**(**X3**))** **+** geom\_histogram**(**breaks**=**seq**(**0,15,by**=**1**)**,color**=**"black",fill**=**"orange"**)** **+** labs**(**x**=**"szerokość prostokąta (liczba całkowita)",y**=**"Liczba"**)**



**Kod:**

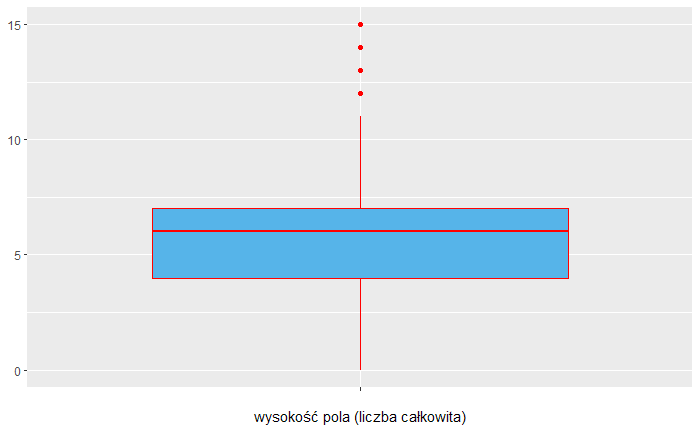
ggplot**(**mydata,aes**(**x**=**"",y**=**X3**))** **+** geom\_boxplot**(**fill **=** "#56B4E9", color **=** "red"**)+** theme**(**axis.title.y**=**element\_blank**())** **+** labs**(**x**=**"szerokość prostokąta (liczba całkowita)"**)**

## Wysokość pudełka w pikselach.



**Kod:**

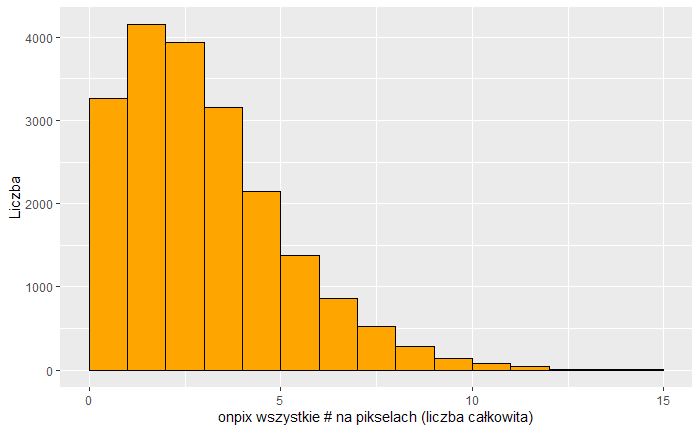
ggplot**(**data**=**mydata,aes**(**X4**))** **+** geom\_histogram**(**breaks**=**seq**(**0,15,by**=**1**)**,color**=**"black",fill**=**"orange"**)** **+** labs**(**x**=**"wysokość pola (liczba całkowita)",y**=**"Liczba"**)**



**Kod:**

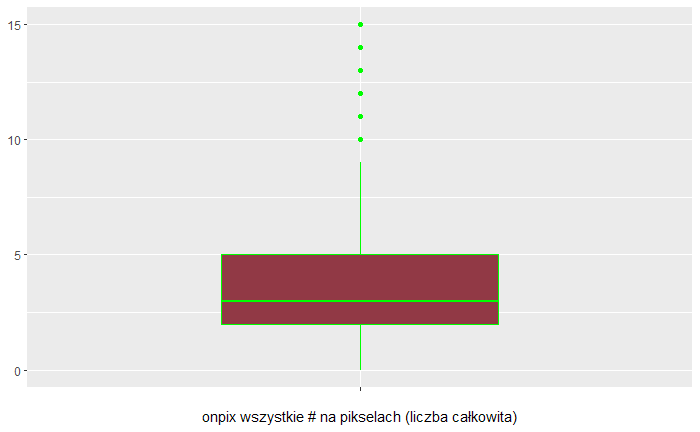
ggplot**(**mydata,aes**(**x**=**"",y**=**X4**))** **+** geom\_boxplot**(**fill **=** "#56B4E9", color **=** "red"**)+** theme**(**axis.title.y**=**element\_blank**())** **+** labs**(**x**=**"wysokość pola (liczba całkowita)"**)**

## Całkowita liczba „włączonych” pikseli na obrazie.



**Kod:**

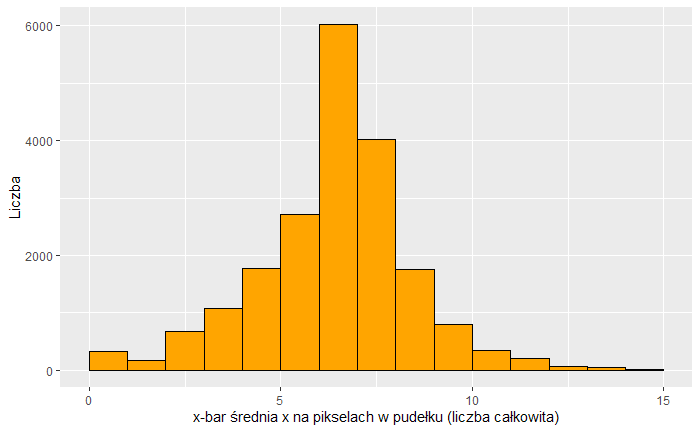
ggplot**(**data**=**mydata,aes**(**X5**))** **+** geom\_histogram**(**breaks**=**seq**(**0,15,by**=**1**)**,color**=**"black",fill**=**"orange"**)** **+** labs**(**x**=**"onpix wszystkie # na pikselach (liczba całkowita)",y**=**"Liczba"**)**



**Kod:**

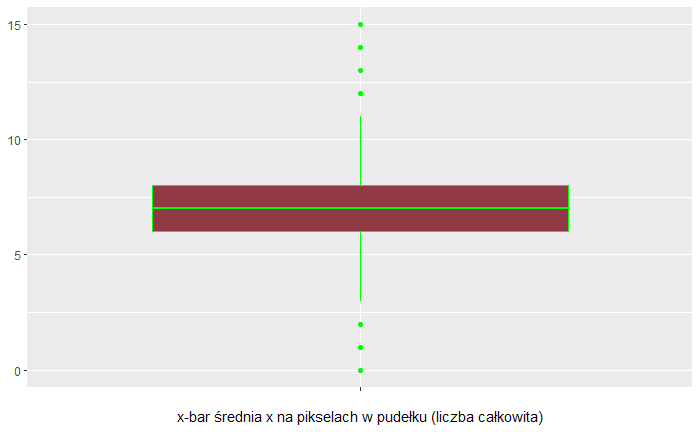
ggplot**(**mydata,aes**(**x**=**"",y**=**X5**))** **+** geom\_boxplot**(**fill **=** "#913945", color **=** "green",width**=**0.5**)** **+** theme**(**axis.title.y**=**element\_blank**())** **+** labs**(**x**=**"onpix wszystkie # na pikselach (liczba całkowita)"**)**

## Średnia pozioma pozycja wszystkich „włączonych” pikseli względem środka pudełka i podzielona po szerokości pudełka.



**Kod:**

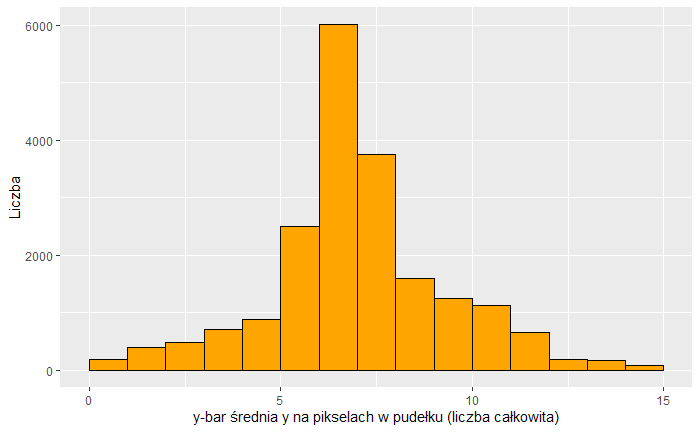
ggplot**(**data**=**mydata,aes**(**X6**))** **+** geom\_histogram**(**breaks**=**seq**(**0,15,by**=**1**)**,color**=**"black",fill**=**"orange"**)** **+** labs**(**x**=**"x-bar średnia x na pikselach w pudełku (liczba całkowita)",y**=**"Liczba"**)**



**Kod:**

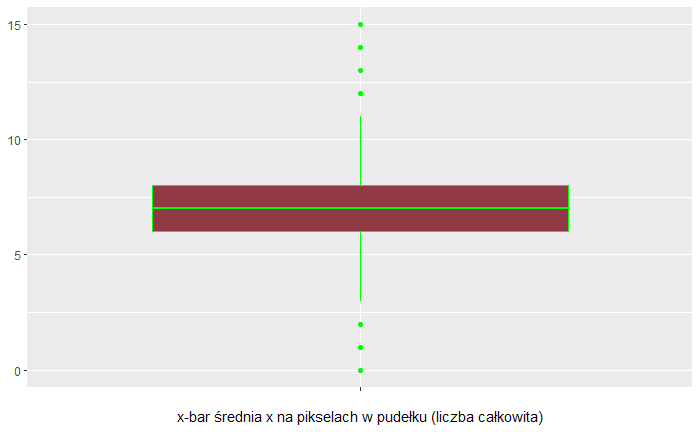
ggplot**(**mydata,aes**(**x**=**"",y**=**X6**))** **+** geom\_boxplot**(**fill **=** "#913945", color **=** "green"**)+** theme**(**axis.title.y**=**element\_blank**())** **+** labs**(**x**=**"x-bar średnia x na pikselach w pudełku (liczba całkowita)"**)**

## Średnia pionowa pozycja wszystkich „włączonych” pikseli względem środka pudełka i podzielona po wysokości pudełka.



**Kod:**

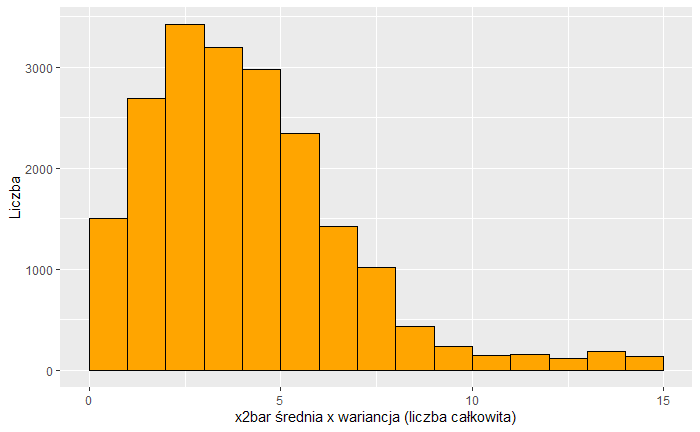
ggplot**(**data**=**mydata,aes**(**X7**))** **+** geom\_histogram**(**breaks**=**seq**(**0,15,by**=**1**)**,color**=**"black",fill**=**"orange"**)** **+** labs**(**x**=**"y-bar średnia y na pikselach w pudełku (liczba całkowita)",y**=**"Liczba"**)**



**Kod:**

ggplot**(**mydata,aes**(**x**=**"",y**=**X7**))** **+** geom\_boxplot**(**fill **=** "#913945", color **=** "green"**)+** theme**(**axis.title.y**=**element\_blank**())** **+** labs**(**x**=**"y-bar średnia y na pikselach w pudełku (liczba całkowita)"**)**

## Średnia kwadratowa wartość odległości pikseli w poziomie zmierzona w powyżej X6.



**Kod:**

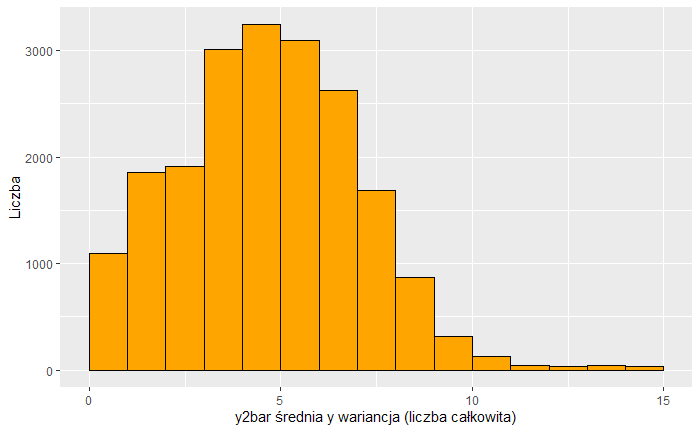
ggplot**(**data**=**mydata,aes**(**X8**))** **+** geom\_histogram**(**breaks**=**seq**(**0,15,by**=**1**)**,color**=**"black",fill**=**"orange"**)** **+** labs**(**x**=**"x2bar średnia x wariancja (liczba całkowita)",y**=**"Liczba"**)**



**Kod:**

ggplot**(**mydata,aes**(**x**=**"",y**=**X8**))** **+** geom\_boxplot**(**fill **=** "#913945", color **=** "green"**)+** theme**(**axis.title.y**=**element\_blank**())** **+** labs**(**x**=**"x2bar średnia x wariancja (liczba całkowita)"**)**

## Średnia wartość odległości pikseli w pionie zmierzona w powyżej X7.



**Kod:**

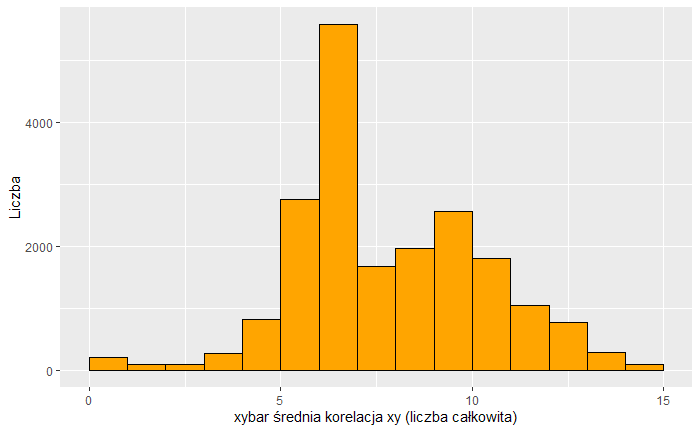
ggplot**(**data**=**mydata,aes**(**X9**))** **+** geom\_histogram**(**breaks**=**seq**(**0,15,by**=**1**)**,color**=**"black",fill**=**"orange"**)** **+** labs**(**x**=**"y2bar średnia y wariancja (liczba całkowita)",y**=**"Liczba"**)**



**Kod:**

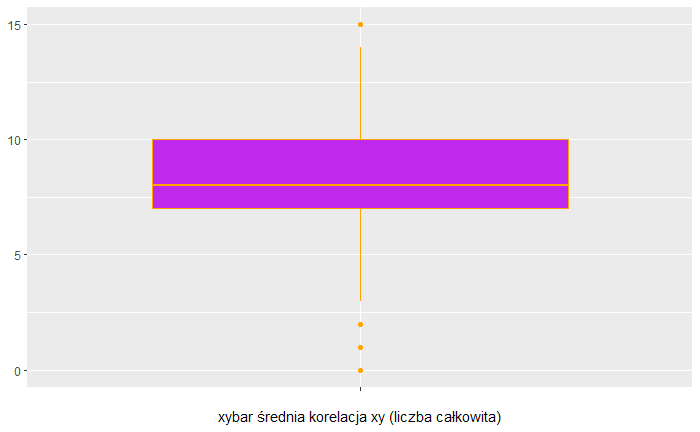
ggplot**(**mydata,aes**(**x**=**"",y**=**X9**))** **+** geom\_boxplot**(**fill **=** "#BE29EC", color **=** "orange",width**=**0.5**)** **+** theme**(**axis.title.y**=**element\_blank**())** **+** labs**(**x**=**"y2bar średnia y wariancja (liczba całkowita)"**)**

## Średni iloczyn odległości poziomej i pionowej dla każdego „włączonego” piksela zgodnie z pomiarem w powyżej X6 i X7.



**Kod:**

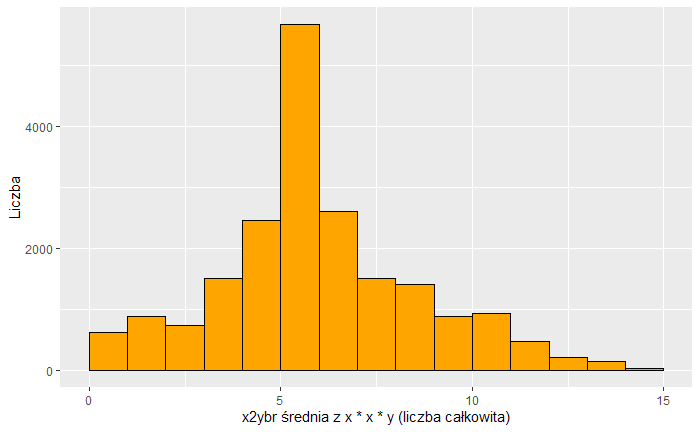
ggplot**(**data**=**mydata,aes**(**X10**))** **+** geom\_histogram**(**breaks**=**seq**(**0,15,by**=**1**)**,color**=**"black",fill**=**"orange"**)** **+** labs**(**x**=**"xybar średnia korelacja xy (liczba całkowita)",y**=**"Liczba"**)**



**Kod:**

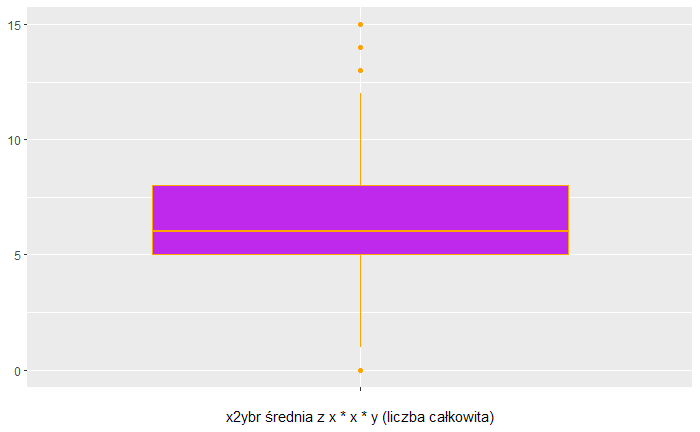
ggplot**(**mydata,aes**(**x**=**"",y**=**X10**))** **+** geom\_boxplot**(**fill **=** "#BE29EC", color **=** "orange"**)+** theme**(**axis.title.y**=**element\_blank**())** **+** labs**(**x**=**"xybar średnia korelacja xy (liczba całkowita)"**)**

## Średnia wartość kwadratu odległości poziomej pomnożona przez odległości pionowe dla każdego. Mierzy to korelację wariancji poziomej z pozycją pionową.



**Kod:**

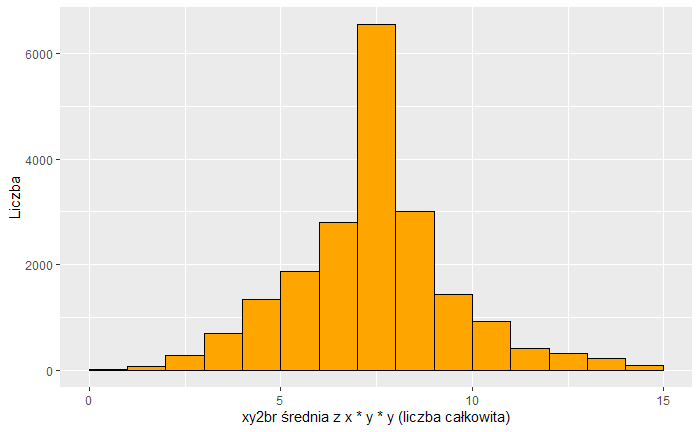
ggplot**(**data**=**mydata,aes**(**X11**))** **+** geom\_histogram**(**breaks**=**seq**(**0,15,by**=**1**)**,color**=**"black",fill**=**"orange"**)** **+** labs**(**x**=**"x2ybr średnia z x \* x \* y (liczba całkowita)",y**=**"Liczba"**)**



**Kod:**

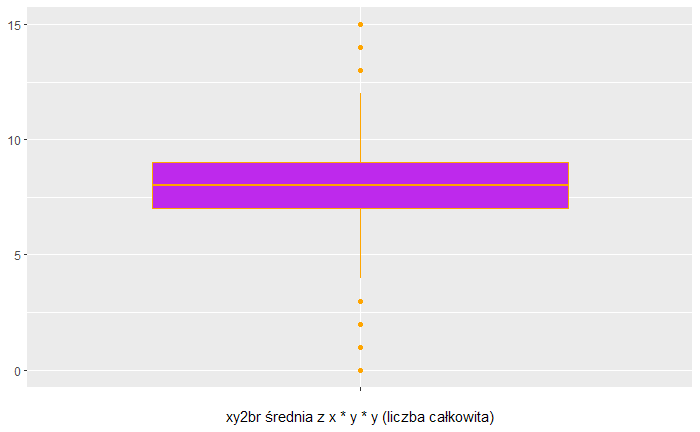
ggplot**(**mydata,aes**(**x**=**"",y**=**X11**))** **+** geom\_boxplot**(**fill **=** "#BE29EC", color **=** "orange"**)+** theme**(**axis.title.y**=**element\_blank**())** **+** labs**(**x**=**"x2ybr średnia z x \* x \* y (liczba całkowita)"**)**

## Średnia wartość kwadratu odległości pionowej pomnożona przez odległość poziomą dla każdego z nich „włączony” piksel. Mierzy to korelację wariancji pionowej z pozycją poziomą.



**Kod:**

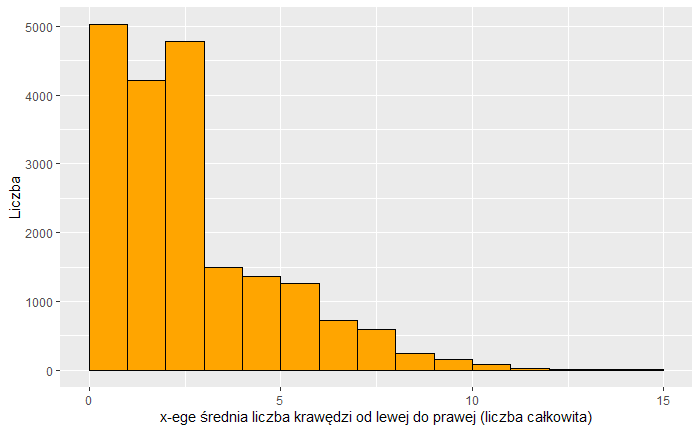
ggplot**(**data**=**mydata,aes**(**X12**))** **+** geom\_histogram**(**breaks**=**seq**(**0,15,by**=**1**)**,color**=**"black",fill**=**"orange"**)** **+** labs**(**x**=**"xy2br średnia z x \* y \* y (liczba całkowita)",y**=**"Liczba"**)**



**Kod:**

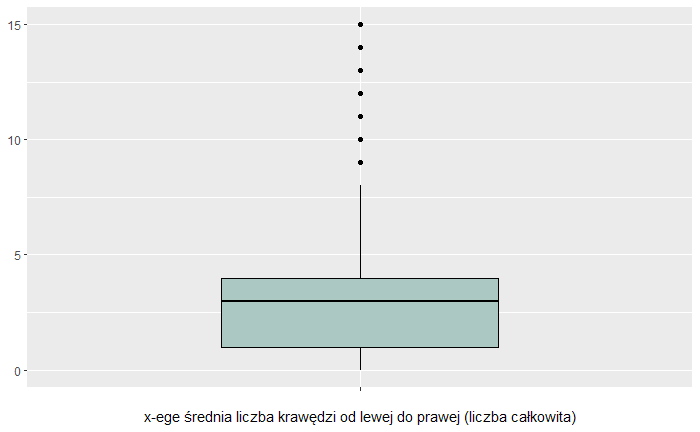
ggplot**(**mydata,aes**(**x**=**"",y**=**X12**))** **+** geom\_boxplot**(**fill **=** "#BE29EC", color **=** "orange"**)+** theme**(**axis.title.y**=**element\_blank**())** **+** labs**(**x**=**"xy2br średnia z x \* y \* y (liczba całkowita)"**)**

## Średnia liczba napotkanych krawędzi podczas wykonywania systematycznych skanów od lewej do prawej we wszystkich pozycjach pionowych w pudełku.



**Kod:**

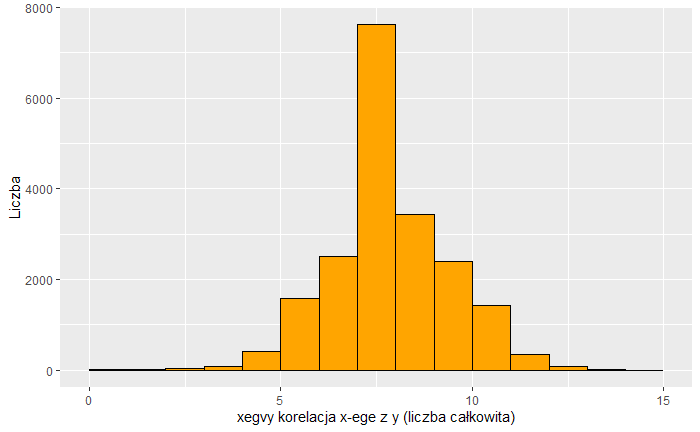
ggplot**(**data**=**mydata,aes**(**X13**))** **+** geom\_histogram**(**breaks**=**seq**(**0,15,by**=**1**)**,color**=**"black",fill**=**"orange"**)** **+** labs**(**x**=**"x-ege średnia liczba krawędzi od lewej do prawej (liczba całkowita)",y**=**"Liczba"**)**



**Kod:**

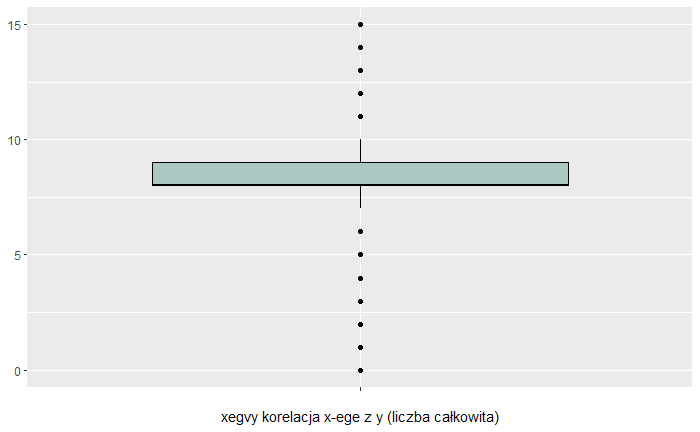
ggplot**(**mydata,aes**(**x**=**"",y**=**X13**))** **+** geom\_boxplot**(**fill **=** "#ABC8C2", color **=** "black",width**=**0.5**)** **+** theme**(**axis.title.y**=**element\_blank**())** **+** labs**(**x**=**"x-ege średnia liczba krawędzi od lewej do prawej (liczba całkowita)"**)**

## Suma pionowych położeń napotkanych krawędzi mierzonych w powyżej X13.



**Kod:**

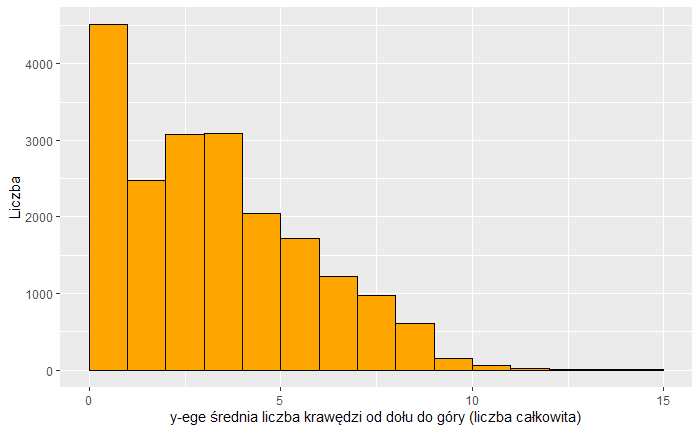
ggplot**(**data**=**mydata,aes**(**X14**))** **+** geom\_histogram**(**breaks**=**seq**(**0,15,by**=**1**)**,color**=**"black",fill**=**"orange"**)** **+** labs**(**x**=**"xegvy korelacja x-ege z y (liczba całkowita)",y**=**"Liczba"**)**



**Kod:**

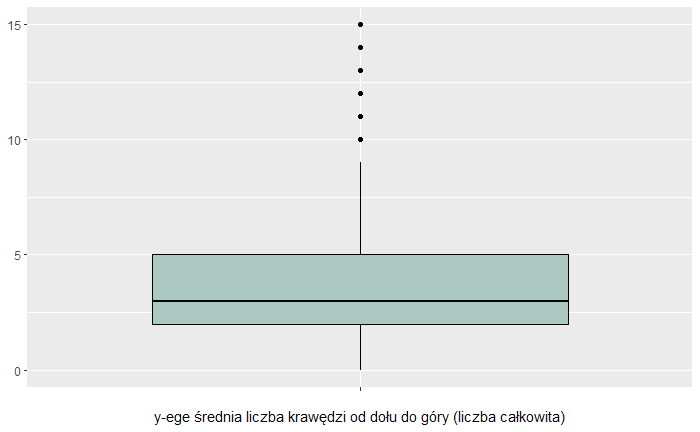
ggplot**(**mydata,aes**(**x**=**"",y**=**X14**))** **+** geom\_boxplot**(**fill **=** "#ABC8C2", color **=** "black"**)+** theme**(**axis.title.y**=**element\_blank**())** **+** labs**(**x**=**"xegvy korelacja x-ege z y (liczba całkowita)"**)**

## Średnia liczba krawędzi napotkanych podczas systematycznego skanowania obrazu z od dołu do góry nad wszystkimi poziomymi pozycjami w pudełku.



**Kod:**

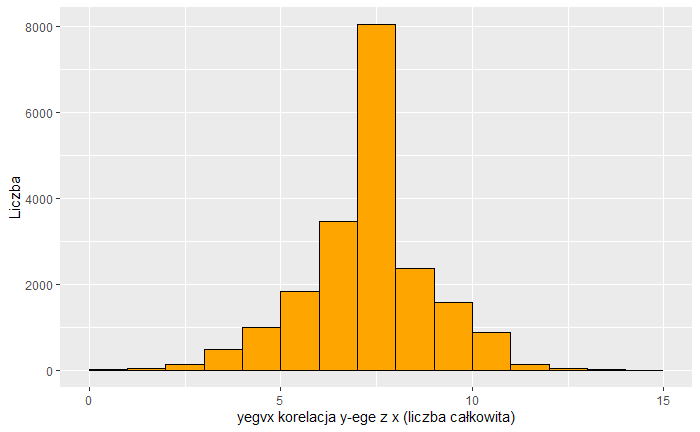
ggplot**(**data**=**mydata,aes**(**X15**))** **+** geom\_histogram**(**breaks**=**seq**(**0,15,by**=**1**)**,color**=**"black",fill**=**"orange"**)** **+** labs**(**x**=**"y-ege średnia liczba krawędzi od dołu do góry (liczba całkowita)",y**=**"Liczba"**)**



**Kod:**

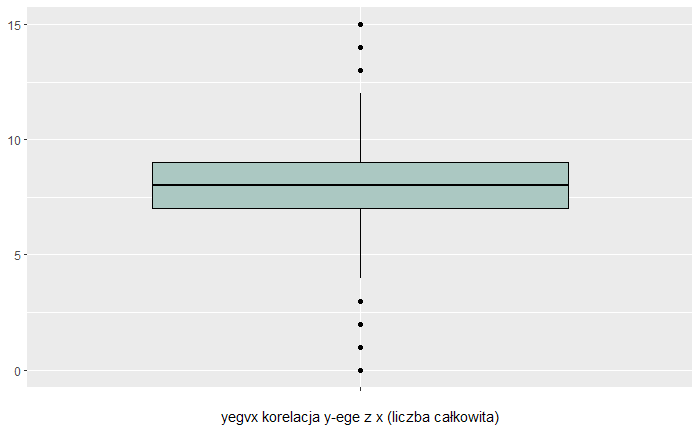
ggplot**(**mydata,aes**(**x**=**"",y**=**X15**))** **+** geom\_boxplot**(**fill **=** "#ABC8C2", color **=** "black"**)+** theme**(**axis.title.y**=**element\_blank**())** **+** labs**(**x**=**"y-ege średnia liczba krawędzi od dołu do góry (liczba całkowita)"**)**

## Suma poziomych położeń napotkanych krawędzi mierzonych w powyżej X15.



**Kod:**

ggplot**(**data**=**mydata,aes**(**X16**))** **+** geom\_histogram**(**breaks**=**seq**(**0,15,by**=**1**)**,color**=**"black",fill**=**"orange"**)** **+** labs**(**x**=**"yegvx korelacja y-ege z x (liczba całkowita)",y**=**"Liczba"**)**



**Kod:**

ggplot**(**mydata,aes**(**x**=**"",y**=**X16**))** **+** geom\_boxplot**(**fill **=** "#ABC8C2", color **=** "black"**)+** theme**(**axis.title.y**=**element\_blank**())** **+** labs**(**x**=**"yegvx korelacja y-ege z x (liczba całkowita)"**)**

# **Wyznaczyć cechę decyzyjną i cechy objaśniające.**

Cechą decyzyjną jest kolumna „T” z literami a cechami objaśniającymi są kolumny od X1 do X16 są bardzo ważne dla klasyfikacji liter występujących w zbiorze danych.

# **Stworzyć klasyfikator używający algorytmu KNN.**

table**(**mydata**$**T**)**

mydata**$**T **<-** factor**(**mydata**$**T, levels **=** c**(**"A", "B","C","D","E","F","G","H","I","J","K","L","M","N","O","P","Q","R","S","T","U","V","W","X","Y","Z"**)**,

labels **=** c**(**"1-type", "2-type","3-type","4-type","5-type","6-type","7-type","8-type","9-type","10-type","11-type","12-type","13-type","14-type","15-type","16-type","17-type","18-type","19-type","20-type","21-type","22-type","23-type","24-type","25-type","26-type"**))**

round**(**prop.table**(**table**(**mydata**$**T**))** **\*** 100, digits **=** 1**)**

summary**(**mydata**[**c**(**"X1", "X2", "X3", "X4", "X5", "X6", "X7", "X8", "X9", "X10", "X11", "X12", "X13", "X14", "X15", "X16"**)])**

normalize **<-** **function(**x**)** **{**

return **((**x **-** min**(**x**))** **/** **(**max**(**x**)** **-** min**(**x**)))** **}**

normalize**(**c**(**1, 2, 3, 4, 5**))**

normalize**(**c**(**10, 20, 30, 40, 50**))**

mydatanew **<-** as.data.frame**(**lapply**(**mydata**[**2**:**16**]**, normalize**))**

summary**(**mydatanew**$**X1**)**

# create training and test data

mydata\_train **<-** mydatanew**[**1**:**1000, **]**

mydata\_test **<-** mydatanew**[**1001**:**19999, **]**

# create labels for training and test data

mydata\_train\_labels **<-** mydata**[**1**:**1000, 1**]**

mydata\_test\_labels **<-** mydata**[**1001**:**19999, 1**]**

## Step 3: Training a model on the data ----

# load the "class" library

library**(**class**)**

mydata\_test\_pred **<-** knn**(**train **=** mydata\_train, test **=** mydata\_test,

cl **=** mydata\_train\_labels, k **=** 21**)**

## Step 4: Evaluating model performance ----

# load the "gmodels" library

library**(**gmodels**)**

# Create the cross tabulation of predicted vs. actual

CrossTable**(**x **=** mydata\_test\_labels, y **=** mydata\_test\_pred,

prop.chisq **=** **FALSE)**

## Step 5: Improving model performance ----

# use the scale() function to z-score standardize a data frame

mydata\_z **<-** as.data.frame**(**scale**(**mydata**[-**1**]))**

# confirm that the transformation was applied correctly

#summary(wbcd\_z$area\_mean)

# create training and test datasets

mydata\_train **<-** mydata\_z**[**1**:**1000, **]**

mydata\_test **<-** mydata\_z**[**1001**:**19999, **]**

# re-classify test cases

mydata\_test\_pred **<-** knn**(**train **=** mydata\_train, test **=** mydata\_test,

cl **=** mydata\_train\_labels, k **=** 21**)**

# Create the cross tabulation of predicted vs. actual

CrossTable**(**x **=** mydata\_test\_labels, y **=** mydata\_test\_pred,

prop.chisq **=** **FALSE)**

# **Przeprowadzić klasyfikację dla różnych wartości K. Dla każdej wartości podać miarę accuracy (dokładności) klasyfikatora.**

mydata\_train **<-** mydatanew**[**1**:**1000, **]**

mydata\_test **<-** mydatanew**[**1001**:**19999, **]**

mydata\_test\_pred **<-** knn**(**train **=** mydata\_train, test **=** mydata\_test, cl **=** mydata\_train\_labels, k**=**1**)**

CrossTable**(**x **=** mydata\_test\_labels, y **=** mydata\_test\_pred, prop.chisq**=FALSE)**

#Miara accuracy (dokładności) klasyfikatora.

confusion**=**table**(**mydata\_test\_pred,mydata\_test\_labels**)**

sum**(**diag**(**confusion**))/**nrow**(**mydata\_test**)**

mydata\_test\_pred **<-** knn**(**train **=** mydata\_train, test **=** mydata\_test, cl **=** mydata\_train\_labels, k**=**3**)**

CrossTable**(**x **=** mydata\_test\_labels, y **=** mydata\_test\_pred, prop.chisq**=FALSE)**

#Miara accuracy (dokładności) klasyfikatora.

confusion**=**table**(**mydata\_test\_pred,mydata\_test\_labels**)**

sum**(**diag**(**confusion**))/**nrow**(**mydata\_test**)**

mydata\_test\_pred **<-** knn**(**train **=** mydata\_train, test **=** mydata\_test, cl **=** mydata\_train\_labels, k**=**13**)**

CrossTable**(**x **=** mydata\_test\_labels, y **=** mydata\_test\_pred, prop.chisq**=FALSE)**

#Miara accuracy (dokładności) klasyfikatora.

confusion**=**table**(**mydata\_test\_pred,mydata\_test\_labels**)**

sum**(**diag**(**confusion**))/**nrow**(**mydata\_test**)**

mydata\_test\_pred **<-** knn**(**train **=** mydata\_train, test **=** mydata\_test, cl **=** mydata\_train\_labels, k**=**16**)**

CrossTable**(**x **=** mydata\_test\_labels, y **=** mydata\_test\_pred, prop.chisq**=FALSE)**

#Miara accuracy (dokładności) klasyfikatora.

confusion**=**table**(**mydata\_test\_pred,mydata\_test\_labels**)**

sum**(**diag**(**confusion**))/**nrow**(**mydata\_test**)**

mydata\_test\_pred **<-** knn**(**train **=** mydata\_train, test **=** mydata\_test, cl **=** mydata\_train\_labels, k**=**19**)**

CrossTable**(**x **=** mydata\_test\_labels, y **=** mydata\_test\_pred, prop.chisq**=FALSE)**

#Miara accuracy (dokładności) klasyfikatora.

confusion**=**table**(**mydata\_test\_pred,mydata\_test\_labels**)**

sum**(**diag**(**confusion**))/**nrow**(**mydata\_test**)**

mydata\_test\_pred **<-** knn**(**train **=** mydata\_train, test **=** mydata\_test, cl **=** mydata\_train\_labels, k**=**26**)**

CrossTable**(**x **=** mydata\_test\_labels, y **=** mydata\_test\_pred, prop.chisq**=FALSE)**

#Miara accuracy (dokładności) klasyfikatora.

confusion**=**table**(**mydata\_test\_pred,mydata\_test\_labels**)**

sum**(**diag**(**confusion**))/**nrow**(**mydata\_test**)**

n **=** nrow**(**mydata**)**

train **=** sample**(**1**:**n, size **=** round**(**0.7**\***n**)**,replace **=** **FALSE)**

t1 **=** mydata**[**train,**]**

t2 **=** mydata**[-**train,**]**

n1 **=** nrow**(**t1**)**

n2 **=** nrow**(**t2**)**

labels **<-** t1**[**,1**]**

error.rate **<-** numeric**(**10**)**

**for(**i **in** 1**:**26**)**

**{**

knn.pred **<-** knn**(**t1**[**,**-**1**]**,t2**[**,**-**1**]**,labels, k **=** i**)**

error.rate**[**i**]** **<-** 1**-**mean**(**knn.pred **==** t2**[**,1**])**

**}**

plot**(**1**:**26, error.rate,"b", pch **=** 20, col **=** "red", xlab **=** "K", ylab **=** "Error Rate"**)**

pred **<-** knn**(**t1**[**,**-**1**]**,t2**[**,**-**1**]**,labels, k **=** 1**)**

table**(**pred,t2**$**T**)**

pred **<-** knn**(**t1**[**,**-**1**]**,t2**[**,**-**1**]**,labels, k **=** 3**)**

table**(**pred,t2**$**T**)**

pred **<-** knn**(**t1**[**,**-**1**]**,t2**[**,**-**1**]**,labels, k **=** 13**)**

table**(**pred,t2**$**T**)**

pred **<-** knn**(**t1**[**,**-**1**]**,t2**[**,**-**1**]**,labels, k **=** 16**)**

table**(**pred,t2**$**T**)**

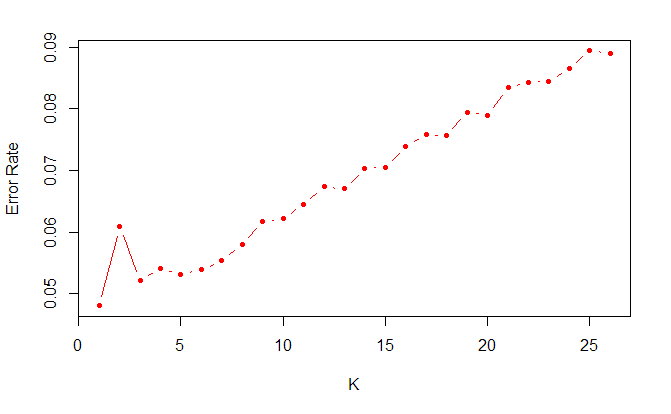
pred **<-** knn**(**t1**[**,**-**1**]**,t2**[**,**-**1**]**,labels, k **=** 19**)**

table**(**pred,t2**$**T**)**

pred **<-** knn**(**t1**[**,**-**1**]**,t2**[**,**-**1**]**,labels, k **=** 26**)**

table**(**pred,t2**$**T**)**

CrossTable**(**x**=**t2**$**T,y**=**pred, prop.chisq **=** **FALSE)**



KNN to prosty algorytm, który przechowuje wszystkie dostępne przypadki i na ich podstawie klasyfikuje nowe przypadki na miarę podobieństwa. Kiedy KNN jest używany do klasyfikacji, wynik można obliczyć jako klasę o najwyższej częstości z K-najbardziej podobnych instancji. Aby wybrać wartość dla K wybraliśmy wartości od 1 do 26 dla K, a następnie obliczyliśmy błąd. Powyżej przedstawiono wykres wskaźnika błędów.

Można zauważyć, że poziom błędu rośnie wraz ze wzrostem wartości K oprócz dla K = 3. Dla K = 26 poziom błędu jest maksymalny, a dla K = 1 poziom błędu jest minimalny.

# **Oszacować miary oceny klasyfikacji: Accuracy, Recall (Sensitivity), Specificity, false positive rate FPR, false discovery rate FDR, precyzję pozytywną, precyzję negatywną, F1– i Fbeta-score, współczynnik korelacji Matthews’a.**

library**(**caret**)**

set.seed**(**0**)**

y **<-** sample**(**c**(**T,F**)**,100,replace**=**T**)**

m1 **<-** y

m1\_err **<-** sample**(**1**:**100,sample**(**15**:**30,1**))**

m1**[**m1\_err**]=!**m1**[**m1\_err**]**

y **<-** factor**(**y**)**

m1 **<-** factor**(**m1**)**

cm1 **<-** confusionMatrix**(**m1,y,positive**=**"TRUE"**)**

#ROC curve#

m2 **<-** sample**(**c**(**T,F**)**,100,replace**=**T**)**

m2 **<-** factor**(**m2**)**

cm2 **<-** confusionMatrix**(**m2,y,positive**=**"TRUE"**)**

tablica **<-** cm1**$**table

tablica1 **<-** cm2**$**table

FPR1 **<-** tablica**[**1,2**]/**sum**(**tablica**[**1,**])**

FPR2 **<-** tablica1**[**1,2**]/**sum**(**tablica1**[**1,**])**

TPR1 **<-** tablica**[**2,2**]/**sum**(**tablica**[**2,**])**

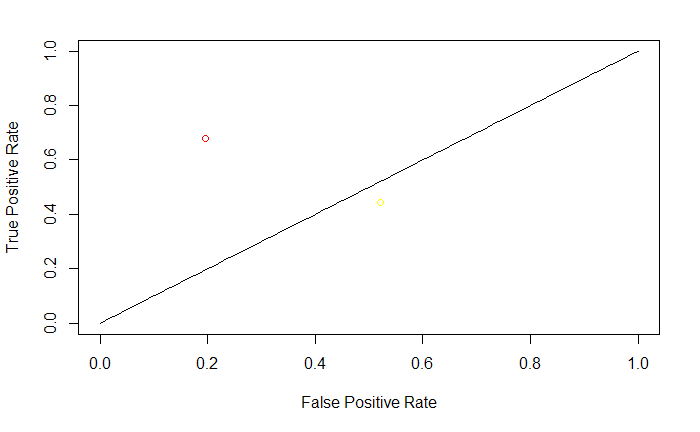
TPR2 **<-** tablica1**[**2,2**]/**sum**(**tablica1**[**2,**])**

plot**(function(**x**)**x,xlim**=**c**(**0,1**)**,ylim**=**c**(**0,1**)**,xlab**=**"False Positive Rate",

ylab**=**"True Positive Rate",col**=**"black"**)**

points**(**FPR1,TPR1,col**=**"red"**)**

points**(**FPR2,TPR2,col**=**"yellow"**)**



rm**(**list **=** ls**())**

library**(**pROC**)**

library**(**ggplot2**)**

sim\_dat **<-** read.csv**(**'dane/letter-recognition.data', header **=** T**)**

head**(**sim\_dat**)**

sim\_roc **<-** roc**(**response **=** sim\_dat**$**T,

predictor **=** sim\_dat**$**X2,

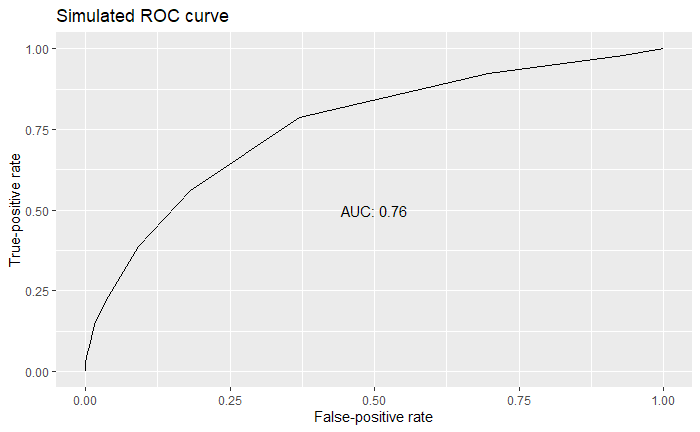
levels **=** c**(**'A','W'**))**

auc**(**sim\_roc**)**

ggroc**(**sim\_roc, legacy.axes **=** **TRUE)** **+**

labs**(**x **=** 'False-positive rate', y **=** 'True-positive rate', title **=** 'Simulated ROC curve'**)** **+**

annotate**(**'text', x **=** .5, y **=** .5, label **=** paste0**(**'AUC: ',round**(**auc**(**sim\_roc**)**, digits **=** 2**)))**



Współczynnik korelacji: **0.76**