

## Statystyka matematyczna i ekonometria

Projekt - Auto Sprzedam

### Autorzy

Anna Kalitka (255445)  
Antoni Bezdziety (249327)

Projekt zaliczeniowy  
Laboratorium, lato 2023/24

## I - Opis bazy danych

Wybrana została baza danych **AutoSprzedam**. Baza pochodzi z zasobów ePortalu.

### Zawartość bazy danych

Baza danych składa się z 41034 rekordów, zawierających szczegółowe dane dotyczące sprzedaży samochodów m.in.

Kolumna	Typ danych	Opis
NrOferty	int	Numer oferty sprzedaży samochodu.
CenaPLN	string	Cena samochodu wyrażona w polskich złotych (PLN).
KM	int	Liczba koni mechanicznych samochodu.
Marka	string	Marka samochodu.
Model	string	Model samochodu.
LiczbaDrzwi	string	Liczba drzwi w samochodzie.
PojemnoscSkokowa	int	Pojemność skokowa silnika wyrażona w centymetrach sześciennych (cm <sup>3</sup> ).
PrzebiegKm	int	Przebieg samochodu wyrażony w kilometrach.
RodzajPaliwa	string	Rodzaj paliwa używanego przez samochód (benzyna, diesel, hybryda, elektryczny itp.).
RokProdukcji	int	Rok produkcji samochodu.
Kolor	string	Kolor samochodu.
KrajPochodzenia	string	Kraj pochodzenia samochodu.
PojazdUszkodzony	string	Informacja czy pojazd jest uszkodzony (Tak/Nie).
SkrzyniaBiegow	string	Typ skrzyni biegów w samochodzie (manualna, automatyczna).

## Prezentacja bazy danych

```
# Wczytanie bazy danych
database = read.csv("../database/AutoSprzedam.dat", sep = "\t", dec = ',');

cat("Liczba rekordów:", nrow(database), ", liczba kolumn:", ncol(database), "\n")

## Liczba rekordów: 41034 , liczba kolumn: 14
N = 5;
cat("-", "Przykład ", N, "pierwszych rekordów.\n")
```

## - Przykład 5 pierwszych rekordów.

```
head(database, N)
```

```
##   NrOferty CenaPLN  KM      Marka      Model LiczbaDrzwi PojemnoscSkokowa
## 1         1   27900 150      Opel      Vectra         4/5             1900
## 2         2   28000 116    Toyota Corolla Verso         4/5             2000
## 3         3   25500 150      Skoda      Superb         4/5             1781
## 4         4   29900 109 Mercedes-Benz      A 180         2/3             1991
## 5         5   29800 207    Peugeot      607          4/5             2946
##   PrzebiegKm      RodzajPaliwa RokProdukcji      Kolor
## 1      80840 olej napędowy (diesel)      2005   czarny-metallic
## 2     166000 olej napędowy (diesel)      2004         biały
## 3     112000      benzyna+LPG      2002   bordowy-metallic
## 4      42000 olej napędowy (diesel)      2005      czerwony
## 5     169000      benzyna      2004   granatowy-metallic
##   KrajPochodzenia PojazdUszkodzony SkrzyniaBiegow
## 1          Niemcy          Nie      manualna
## 2          Polska          Nie      manualna
## 3          Polska          Nie      manualna
## 4          Polska          Nie      manualna
## 5         Francja          Nie      manualna
```

```
cat("-", "Przykład ", N, "ostatnich rekordów.\n")
```

## - Przykład 5 ostatnich rekordów.

```
tail(database, N)
```

```
##   NrOferty CenaPLN  KM      Marka      Model LiczbaDrzwi PojemnoscSkokowa
## 41030    41030 98000.00 220      Opel Insignia         4/5             1998
## 41031    41031 34924.50 184 Mercedes-Benz      S 400         4/5             3996
## 41032    41032 41175.09  70    Peugeot      308          2/3             1397
## 41033    41033 47900.00 115      Ford      C-MAX         4/5             1560
## 41034    41034 14200.00  90      Ford      Mondeo         4/5             1998
##   PrzebiegKm      RodzajPaliwa RokProdukcji      Kolor
## 41030     25500      benzyna      2010   grafitowy-metallic
## 41031    162000 olej napędowy (diesel)      2001   srebrny-metallic
## 41032      9289      benzyna      2010   srebrny-metallic
## 41033     45000 olej napędowy (diesel)      2010   czarny-metallic
## 41034    191024 olej napędowy (diesel)      2003   czarny-metallic
##   KrajPochodzenia PojazdUszkodzony SkrzyniaBiegow
## 41030          Polska          Nie      manualna
## 41031          Czechy          Nie      automatyczna
## 41032          Niemcy          Nie      manualna
```

## 41033	Belgia	Nie	manualna
## 41034	Niemcy	Nie	manualna

## Przygotowanie bazy danych

Przed przejściem do dalszej pracy z bazą danych postanowiono wprowadzenie zmian w jej strukturze, których celem jest uproszczenie przyszłych prac z danymi.

### Pole - *SkrzyniaBiegow*

W bazie występują 3 rodzaje typów skrzyni biegów: *półautomatyczna/sekwencyjna*, *manualna* oraz *automatyczna*. Udział typu *półautomatyczna/sekwencyjna* w całej bazie wynosi 1,5%.

W związku z niewielkim udziałem ze skrzynią *półautomatyczna/sekwencyjna*, typ ten został usunięty z bazy. Dzięki czemu możliwe jest przekształcenie kolumny *SkrzyniaBiegow* (char), na *SkrzyniaBiegowManualna* (bool).

```
gear_box_summary = database %>% group_by(database$SkrzyniaBiegow) %>% summarise(liczba = n());
gear_box_summary

## # A tibble: 3 x 2
##   `database$SkrzyniaBiegow`   liczba
##   <chr>                     <int>
## 1 automatyczna              6941
## 2 manualna                  34030
## 3 półautomatyczna/sekwencyjna    63

cat("Udział pojazdów ze skrzynią biegów typu 'półautomatyczna/sekwencyjna'", sum(gear_box_summary[3,2]), "\n")

## Udział pojazdów ze skrzynią biegów typu 'półautomatyczna/sekwencyjna' 0.1535312 %.

# Remove cars with database$SkrzyniaBiegow == 'półautomatyczna/sekwencyjna'
database = database[database$SkrzyniaBiegow != "półautomatyczna/sekwencyjna",];

# Cast to logical value
database$SkrzyniaBiegow = database$SkrzyniaBiegow == "manualna";
colnames(database)[14] = "SkrzyniaBiegowManualna";
```

### Pole *PojazdUszkodzony*

Domyślnym typem danych dla pola *PojazdUszkodzony* jest (char). Zmienna jest typem logicznym (TAK/NIE). W związku z powyższym kolumna *PojazdUszkodzony* została przekształcona to typu *bool*.

```
database$PojazdUszkodzony = database$PojazdUszkodzony == "Tak"
```

## Prezentacja bazy danych - po wprowadzonych zmianach

```
N = 5;
cat("-", "Przykład ", N, "pierwszych rekordów.\n")
```

```
## - Przykład 5 pierwszych rekordów.
```

```
head(database, N)
```

##	NrOferty	CenaPLN	KM	Marka	Model	LiczbaDrzwi	PojemnoscSkokowa
## 1	1	27900	150	Opel	Vectra	4/5	1900
## 2	2	28000	116	Toyota	Corolla Verso	4/5	2000
## 3	3	25500	150	Skoda	Superb	4/5	1781

```
## 4      4      29900 109 Mercedes-Benz      A 180      2/3      1991
## 5      5      29800 207      Peugeot      607      4/5      2946
##      PrzebiegKm      RodzajPaliwa RokProdukcji      Kolor
## 1      80840 olej napędowy (diesel)      2005      czarny-metallic
## 2      166000 olej napędowy (diesel)      2004      bialy
## 3      112000      benzyna+LPG      2002      bordowy-metallic
## 4      42000 olej napędowy (diesel)      2005      czerwony
## 5      169000      benzyna      2004 granatowy-metallic
##      KrajPochodzenia PojazdUszkodzony SkrzyniaBiegowManualna
## 1      Niemcy      FALSE      TRUE
## 2      Polska      FALSE      TRUE
## 3      Polska      FALSE      TRUE
## 4      Polska      FALSE      TRUE
## 5      Francja      FALSE      TRUE
```

```
cat("- ", "Przykład ", N, "ostatnich rekordów.\n")
```

```
## - Przykład 5 ostatnich rekordów.
```

```
tail(database, N)
```

```
##      NrOferty  CenaPLN  KM      Marka      Model LiczbaDrzwi PojemnoscSkokowa
## 41030      41030 98000.00 220      Opel Insignia      4/5      1998
## 41031      41031 34924.50 184 Mercedes-Benz      S 400      4/5      3996
## 41032      41032 41175.09 70      Peugeot      308      2/3      1397
## 41033      41033 47900.00 115      Ford      C-MAX      4/5      1560
## 41034      41034 14200.00 90      Ford      Mondeo      4/5      1998
##      PrzebiegKm      RodzajPaliwa RokProdukcji      Kolor
## 41030      25500      benzyna      2010 grafitowy-metallic
## 41031      162000 olej napędowy (diesel)      2001 srebrny-metallic
## 41032      9289      benzyna      2010 srebrny-metallic
## 41033      45000 olej napędowy (diesel)      2010 czarny-metallic
## 41034      191024 olej napędowy (diesel)      2003 czarny-metallic
##      KrajPochodzenia PojazdUszkodzony SkrzyniaBiegowManualna
## 41030      Polska      FALSE      TRUE
## 41031      Czechy      FALSE      FALSE
## 41032      Niemcy      FALSE      TRUE
## 41033      Belgia      FALSE      TRUE
## 41034      Niemcy      FALSE      TRUE
```

## Wyznaczenie podstawowych statystyk

### Pole *CenaPLN*

Kolumna zawiera informacje o cenie po jakie asmochody zostały sprzedane w PLN.

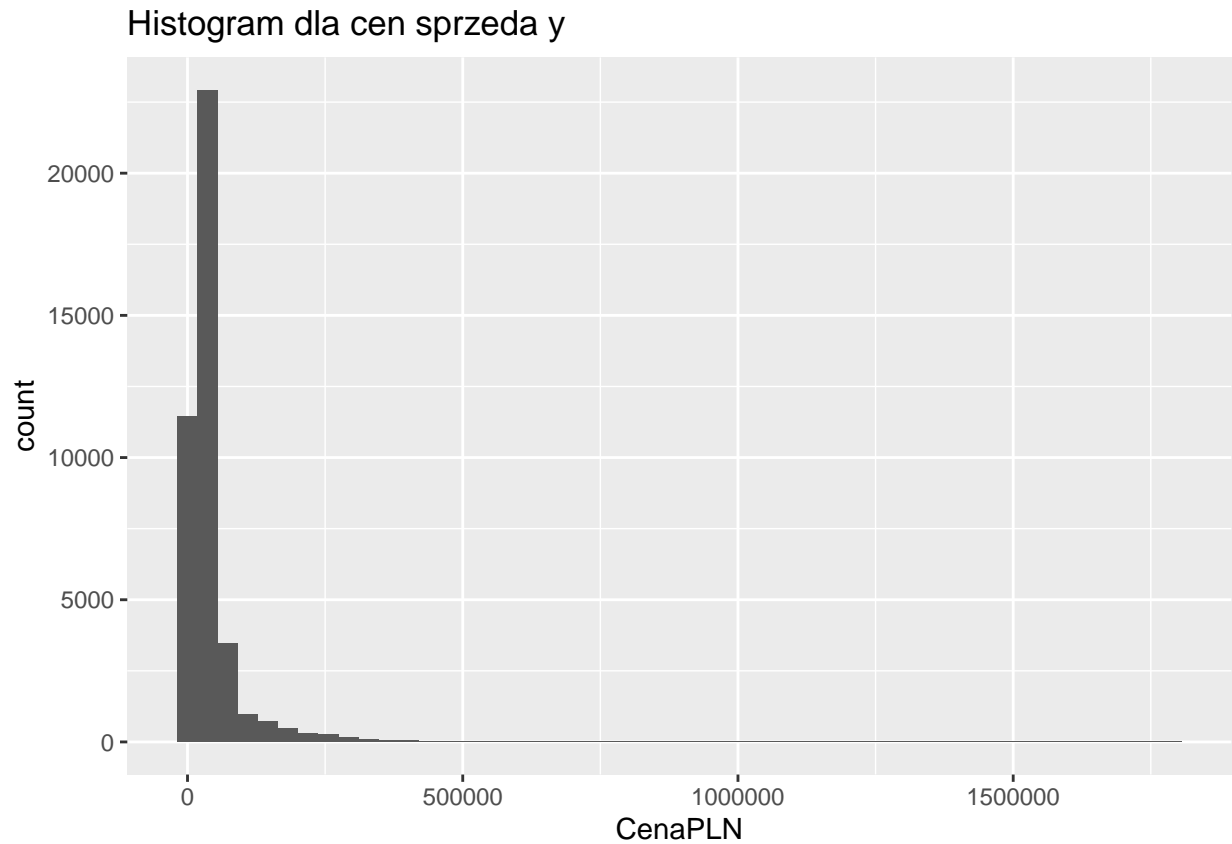
### Dane przed eliminacją danych odstających

```
summarise(database,
  srednia = mean(CenaPLN),
  mediana = median(CenaPLN),
  wariancja = var(CenaPLN),
  odchylenieStd = sd(CenaPLN),
  max = max(CenaPLN),
```

```
min = min(CenaPLN),
)
```

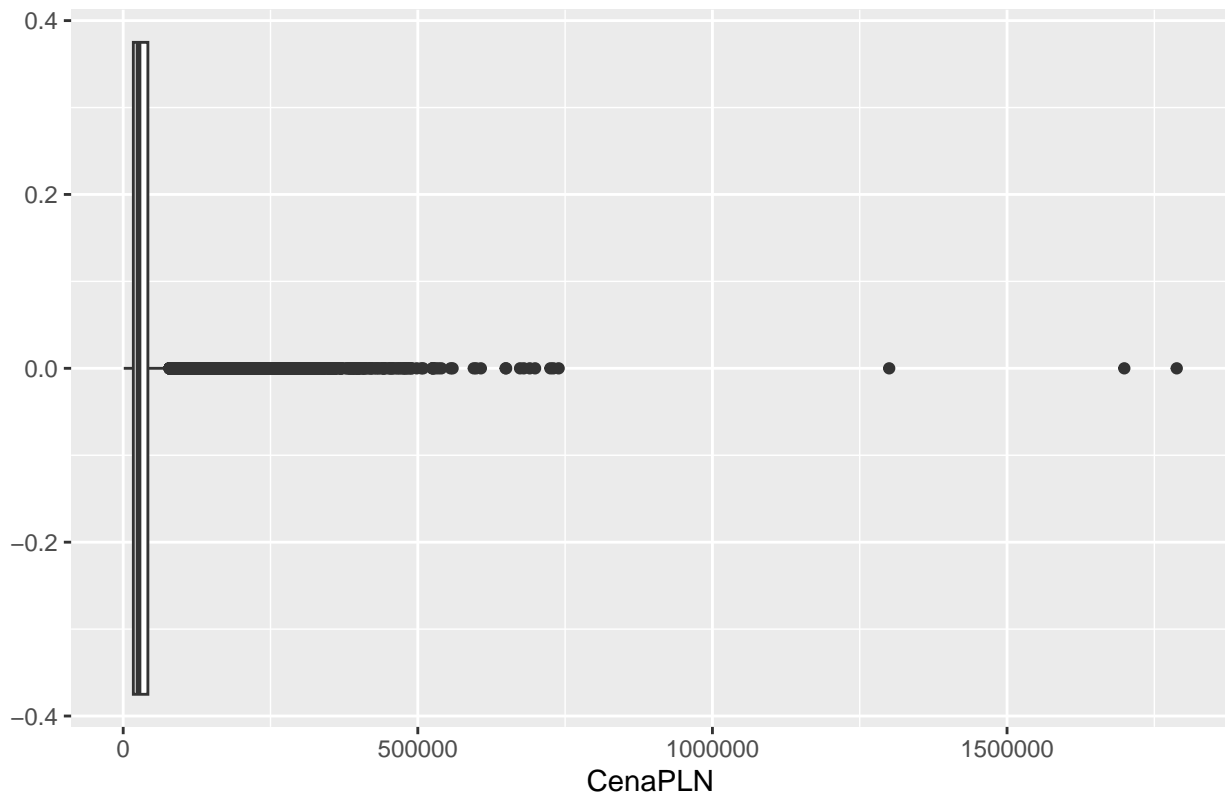
```
##      srednia mediana  wariancja odchylenieStd      max  min
## 1 40848.35   26000 2636580628      51347.64 1788000 1000
```

```
ggplot(data=database) + geom_histogram(aes(x=CenaPLN), bins = 50) + ggtitle("Histogram dla cen sprzedaży")
```



```
ggplot(data=database) + geom_boxplot(aes(x=CenaPLN)) + ggtitle("Wykres pudełko-wąsy dla cen sprzedaży")
```

Wykres pudełko-w sy dla cen sprzedaży



## IV - Obserwacje odstające

Obserwacja odstająca, element odstający – obserwacja relatywnie odległa od pozostałych elementów próby. Innymi słowy, posiadająca nietypową wartość zmiennej niezależnej (objaśniającej) lub nietypowe wartości obydwu zmiennych – zależnej (objaśnianej) i objaśniającej (objaśniających w analizie regresji wielokrotnej).

### Redukcja danych odstających zgodnie z regułą 3 sigma

Redukcja danych zgodnie z regułą 3 sigma.

Identyfikacja danych odstających za pomocą reguły trzech sigm:

1. Obliczenie średniej ( $\mu$ ): Najpierw oblicza się średnią dla danego zbioru danych.
2. Obliczenie odchylenia standardowego ( $\sigma$ ): Następnie oblicza się odchylenie standardowe, które mierzy, jak bardzo dane rozpraszają się wokół średniej.
3. Ustalenie zakresu trzech sigm: Wartości, które znajdują się poza zakresem trzech sigm ( $\mu \pm 3\sigma$ ), są uznawane za dane odstające.

```
mean_price = mean(database$CenaPLN);
sd_price    = sd(database$CenaPLN);

lower_bound <- mean_price - 3 * sd_price;
upper_bound <- mean_price + 3 * sd_price;

cat("Wyznacznowe granice metodą 3 sigma \n")
```

```
## Wyznacznowe granice metodą 3 sigma
cat("Dolna granica ", lower_bound, ", górna granica ", upper_bound, "\n");

## Dolna granica -113194.6 , górna granica 194891.3
cat("Dolna granica wykracza poza osiągnane wartości kwot sprzedaży (ujemna wartość). \n")

## Dolna granica wykracza poza osiągnane wartości kwot sprzedaży (ujemna wartość).
upper_price_example = database[which(database$CenaPLN >= upper_bound),];
head(upper_price_example[order(upper_price_example$CenaPLN, decreasing = TRUE),])

##      NrOferty CenaPLN  KM      Marka      Model LiczbaDrzwi
## 14353    14353 1788000 626 Mercedes-Benz      SLR          4/5
## 36278    36278 1699000 800 Mercedes-Benz      S 63 AMG      4/5
## 14159    14159 1300000 626 Mercedes-Benz      SLR          2/3
## 27512    27512  739000 571 Mercedes-Benz      SLS 63 AMG      2/3
## 28662    28662  730000 563 Mercedes-Benz      SLS 63 AMG      2/3
## 28663    28663  725000 563 Mercedes-Benz      SLS 63 AMG      2/3
##      PojemnoscSkokowa PrzebiegKm RodzajPaliwa RokProdukcji      Kolor
## 14353             5439           1      benzyna      2009  czarny-metallic
## 36278             6233          1500      benzyna      2011  czarny-metallic
## 14159             5439          2098      benzyna      2007  granatowy-metallic
## 27512             6208          5000      benzyna      2010  grafitowy-metallic
## 28662             6208           380      benzyna      2011  czarny-metallic
## 28663             6208           520      benzyna      2011  srebrny-metallic
##      KrajPochodzenia PojazdUszkodzony SkrzyniaBiegowManualna
## 14353             Niemcy             FALSE             FALSE
## 36278             Niemcy             FALSE             FALSE
## 14159             Niemcy             FALSE             FALSE
## 27512             Polska             FALSE             FALSE
## 28662 Stany Zjednoczone             FALSE             FALSE
## 28663 Stany Zjednoczone             FALSE             FALSE

database = database[database$CenaPLN >= lower_bound & database$CenaPLN <= upper_bound,]
```

## Redukcja danych odstających regułą odstępu międzykwartylowego

Metoda redukcji, polega na wykrywaniu i usuwaniu wartości odstających z zestawu danych. Metoda ta opiera się na kwartylach i rozstępie międzykwartylowym.

1. Wyznaczenie  $Q1$  (pierwszy kwartył) oraz  $Q3$  (trzeci kwartył)
2. Wyznaczenie odstępu międzykwartylowego  $IQR = Q3 - Q1$
3. Wyznaczenie dolnej  $Q1 - 1.5 \times IQR$  oraz górnej  $Q1 + 1.5 \times IQR$  granicy.

```
Q1 = quantile(database$PrzebiegKm, 0.25);
Q3 = quantile(database$PrzebiegKm, 0.75);
IQR = Q3-Q1;
lower_bound = Q1 - 1.5*IQR #mniejsze niż
upper_bound = Q3 + 1.5*IQR #większe niż

cat("Wyznacznowe granice metodą odstępu międzykwartylowego sigma \n")
```

```
## Wyznacznowe granice metodą odstępu międzykwartylowego sigma
```

```

cat("Dolna granica ", lower_bound, ", górna granica ", upper_bound, "\n");

## Dolna granica -47281.25 , górna granica 303568.8
cat("Dolna granica wykracza poza osiągnięte wartości kwot sprzedaży (ujemna wartość). \n")

## Dolna granica wykracza poza osiągnięte wartości kwot sprzedaży (ujemna wartość).
database = database[database$PrzebiegKm >= lower_bound & database$PrzebiegKm <= upper_bound,]

database %>% group_by(Marka) %>% summarise(liczbaSprzedanych = n(),
                                           sredniaCena = mean(CenaPLN),
                                           medianaCena = median(CenaPLN))

## # A tibble: 11 x 4
##   Marka      liczbaSprzedanych sredniaCena medianaCena
##   <chr>          <int>          <dbl>      <dbl>
## 1 Audi             3322          64178.    46900
## 2 BMW              2217          61711.    51500
## 3 Fiat             1923          19118.    14800
## 4 Ford             5393          27389.    22900
## 5 Mercedes-Benz    2319          62040.    46000
## 6 Opel             4991          24201.    22600
## 7 Peugeot          3199          25212.    22000
## 8 Renault          4714          21099.    17810.
## 9 Skoda            2928          29156.    26450
## 10 Toyota          2669          36048.    29900
## 11 Volkswagen      6120          35673.    29900

```

Dane po eliminacją danych odstających

```

summarise(database,
  srednia = mean(CenaPLN),
  mediana = median(CenaPLN),
  wariancja = var(CenaPLN),
  odchylenieStd = sd(CenaPLN),
  max = max(CenaPLN),
  min = min(CenaPLN),
)

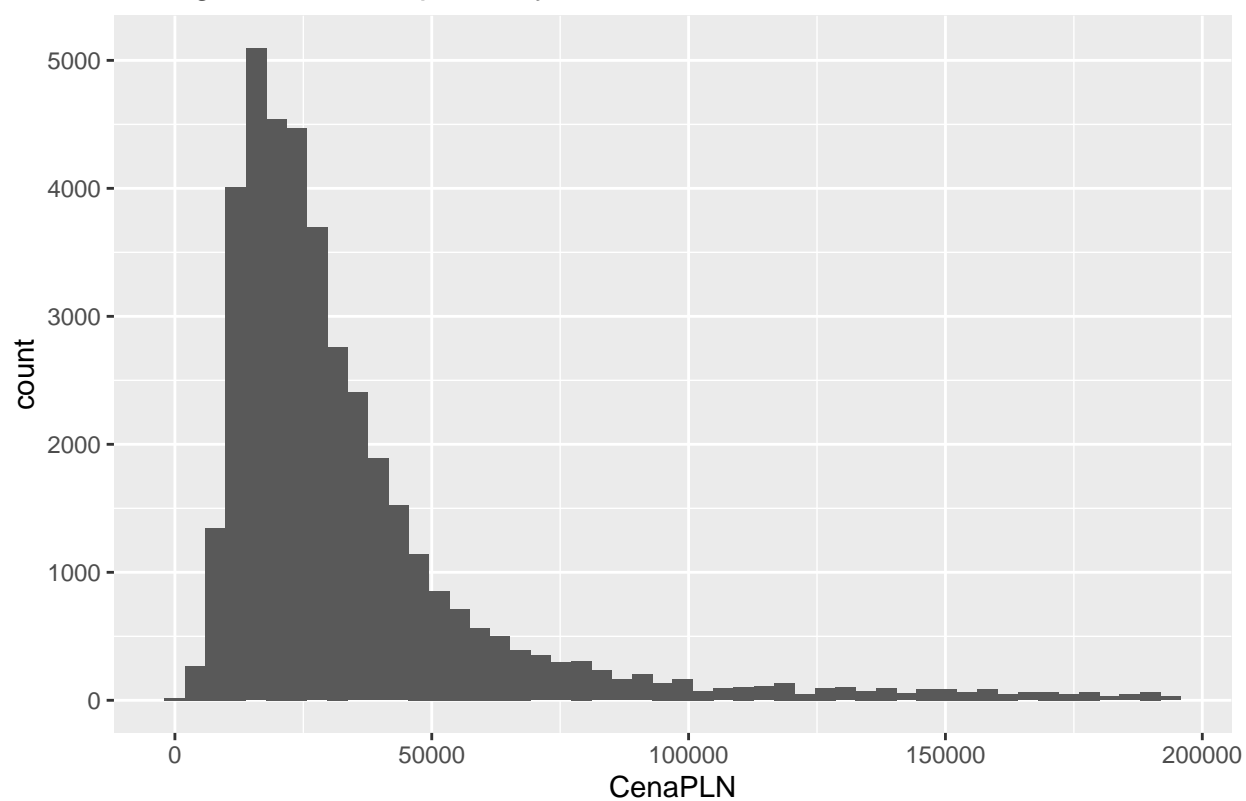
##   srednia mediana wariancja odchylenieStd   max   min
## 1 34656.67  25900 870294063    29500.75 194800 1000

ggplot(data=database) + geom_histogram(aes(x=CenaPLN), bins = 50) + ggtitle("Histogram dla cen sprzedaży")

```

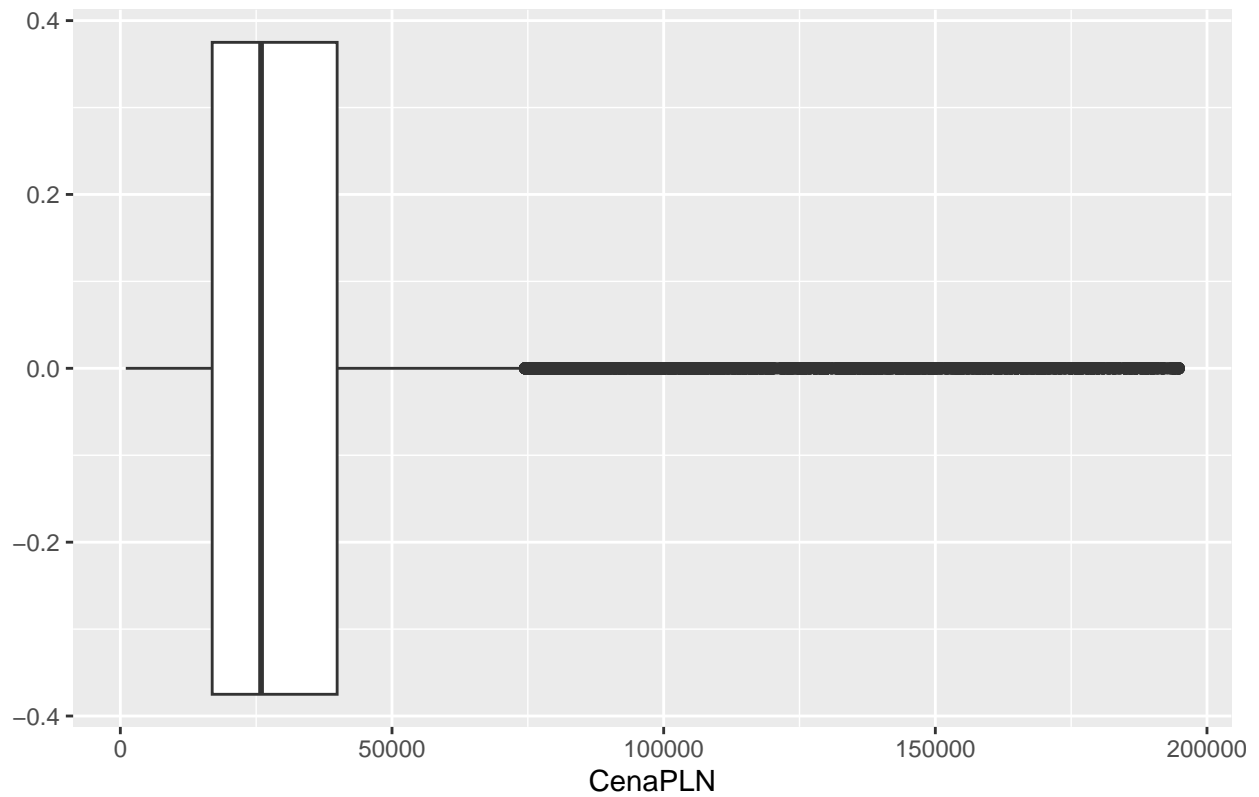


Histogram dla cen sprzedaży



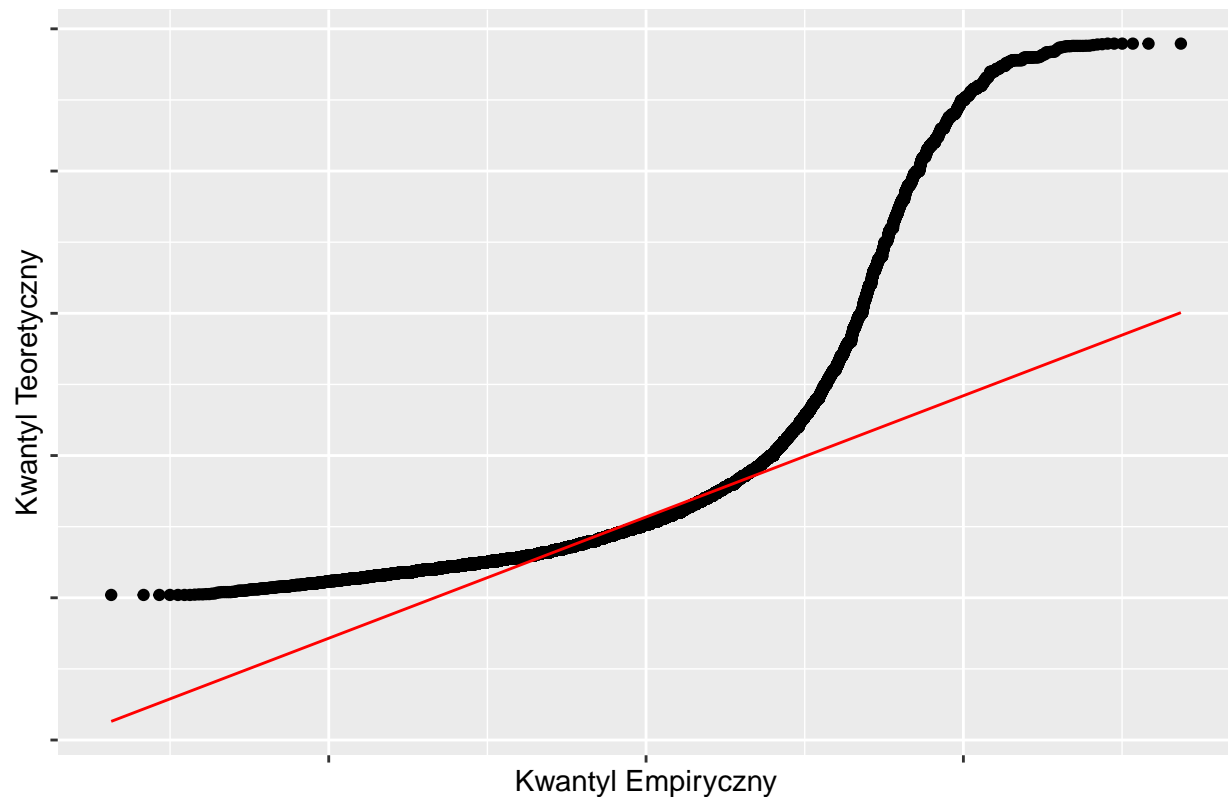
```
ggplot(data=database) + geom_boxplot(aes(x=CenaPLN)) + ggtitle("Wykres pudełko-wąsy dla cen sprzedaży")
```

Wykres pudełko-w sy dla cen sprzedaży



```
ggplot(data=database, aes(sample = CenaPLN)) + stat_qq() + stat_qq_line(colour="red") + ylab("Kwantyl T")  
ggtitle("Wykres QQ dla cen sprzedaży") + theme(axis.text.x = element_blank(), axis.text.y = element_b
```

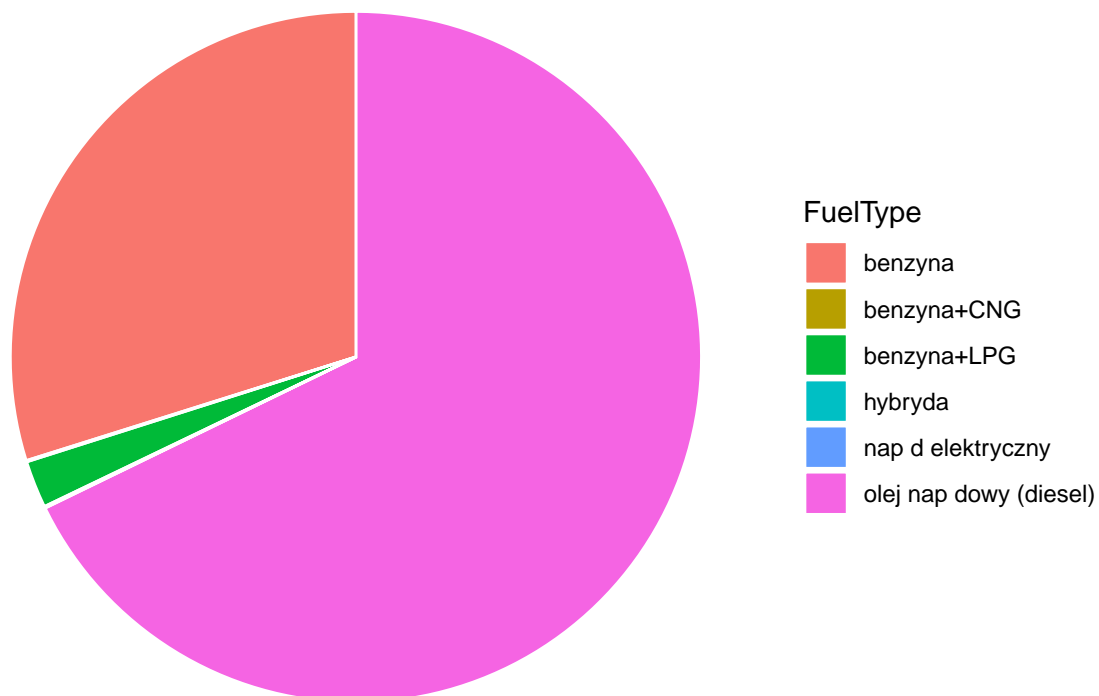
Wykres QQ dla cen sprzedaży



```
fuelType = database %>% group_by(database$RodzajPaliwa) %>% summarise(liczba = n());
colnames(fuelType) = c("FuelType", "Count");
fuelType = fuelType[order(fuelType$Count, decreasing = TRUE),]
fuelType
```

```
## # A tibble: 6 x 2
##   FuelType      Count
##   <chr>      <int>
## 1 olej napędowy (diesel) 26980
## 2 benzyna      11878
## 3 benzyna+LPG    883
## 4 benzyna+CNG    23
## 5 hybryda       21
## 6 napęd elektryczny    10
```

```
ggplot(fuelType, aes(x="", y=Count, fill=FuelType)) +
  geom_bar(stat="identity", width=1, color="white") +
  coord_polar("y", start=0) + theme_void();
```



## Liczba koni mechanicznych samochodu

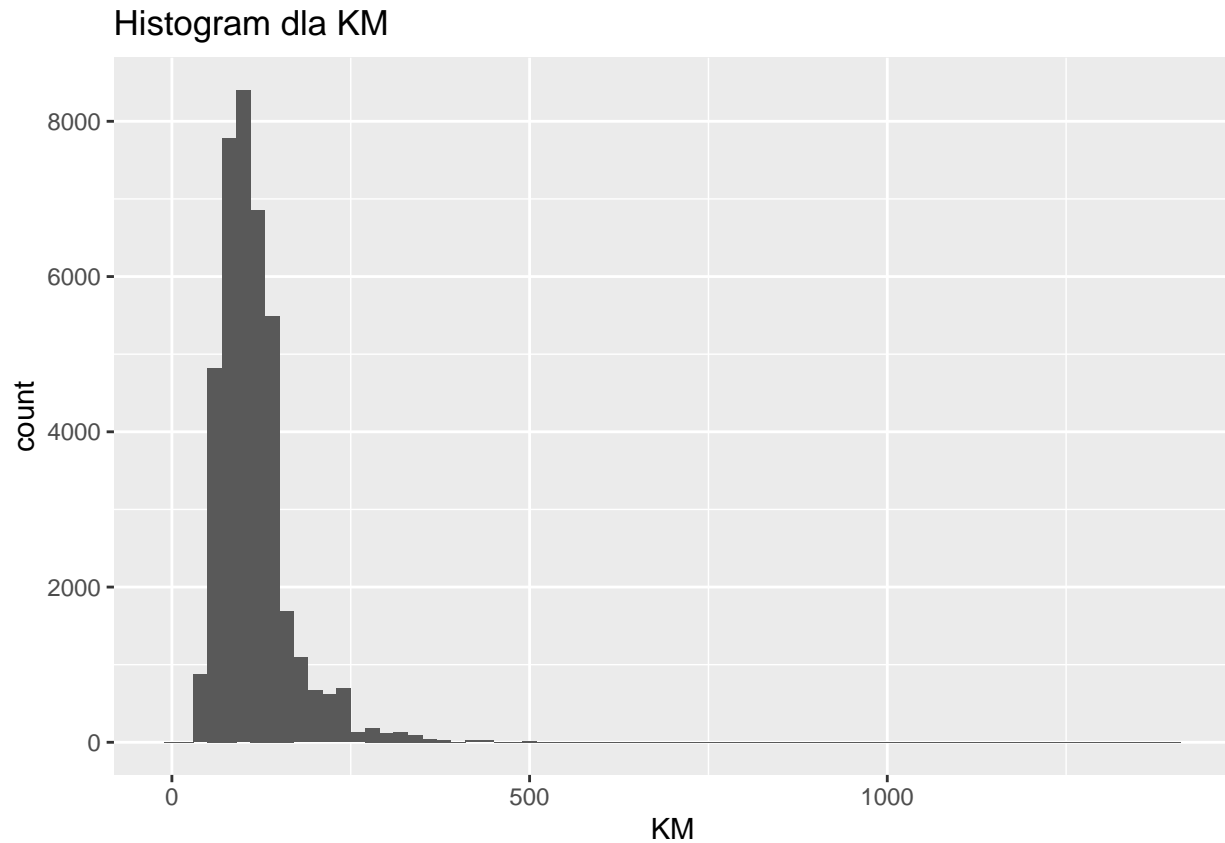
```
summarise(database,
  srednia = mean(KM),
  mediana = median(KM),
  wariancja = var(KM),
  odchylenieStd = sd(KM),
  max = max(KM),
  min = min(KM),
)
```

```
##      srednia mediana wariancja odchylenieStd max min
## 1 116.2776      106 2567.881      50.67426 1400  1
```

```
upper_price_example = database[which(database$CenaPLN >= upper_bound),];
head(upper_price_example[order(upper_price_example$CenaPLN, decreasing = TRUE),])
```

```
## [1] NrOferty          CenaPLN          KM
## [4] Marka                Model            LiczbaDrzwi
## [7] PojemnoscSkokowa     PrzebiegKm      RodzajPaliwa
## [10] RokProdukcji         Kolor           KrajPochodzenia
## [13] PojazdUszkodzony     SkrzyniaBiegowManualna
## <0 wierszy> (lub 'row.names' o zerowej długości)
```

```
ggplot(data=database) + geom_histogram(aes(x=KM), binwidth = 20) + ggtitle("Histogram dla KM")
```



## V - Wyznaczanie prawdopodobieństw dla zmiennej

### Rozkłady dyskretne

#### Dwumianowy

```
#Generowanie próbek
set.seed(42)
N <- 1000
n <- 20
p <- 0.4
X <- rbinom(N, n, p)

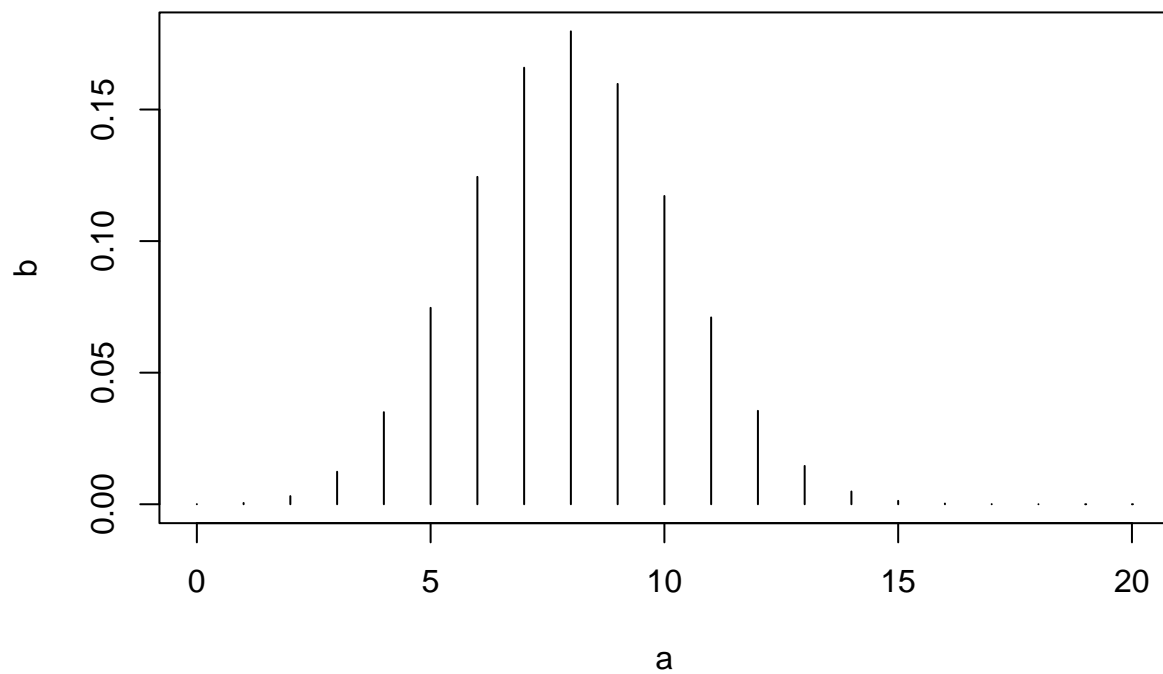
#Prawdopodobieństwa
pbinom(8, n, p) - pbinom(7, n, p) # P(X=8) p. punktowe

## [1] 0.1797058

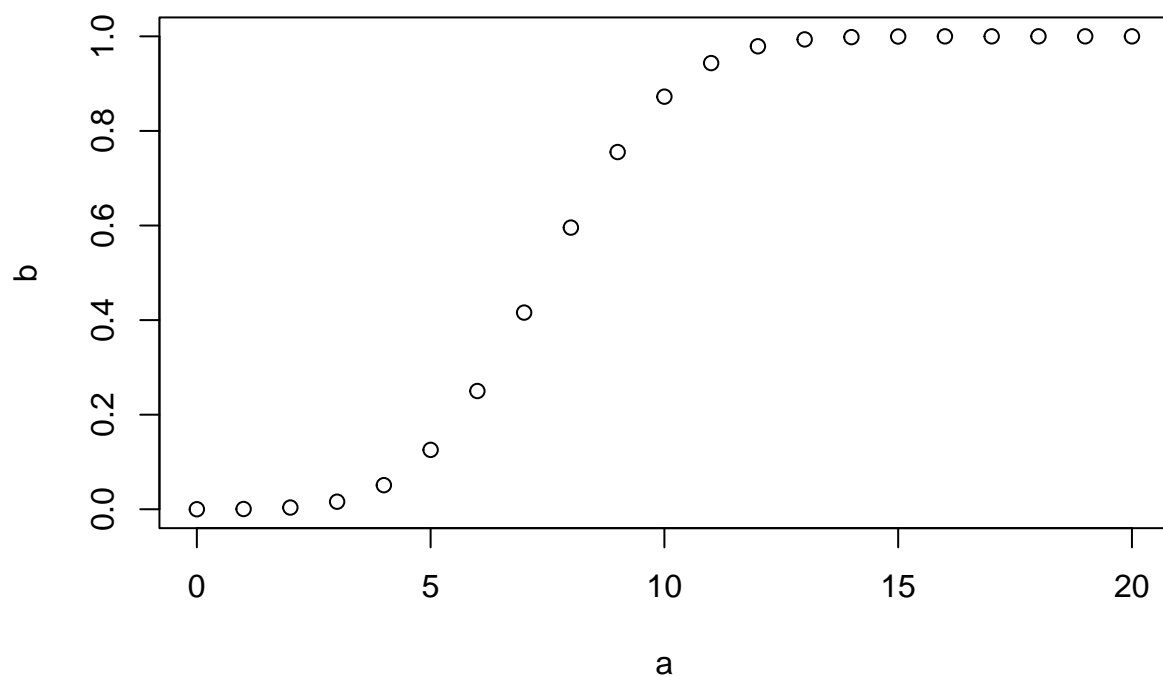
pbinom(10, n, p) # P(X<=10) p. przedziałowe

## [1] 0.8724788

#Wykres gęstości
a <- seq(0, n, by = 1)
b <- dbinom(a, n, p)
plot(a,b, type="h")
```



```
#Wykres dystrybucji  
a <- seq(0, n, by = 1)  
b <- pbinom(a, n, p)  
plot(a,b)
```



```
### Rozkład Poissona
```

```
#Generowanie próbek
```

```
N <- 1000
```

```
lambda <- 200
```

```
X <- rpois(N, lambda)
```

```
#Prawdopodobieństwa
```

```
ppois(190, lambda) - ppois(189, lambda) # P(X=190)
```

```
## [1] 0.02243432
```

```
ppois(190, lambda) #(P<=190)
```

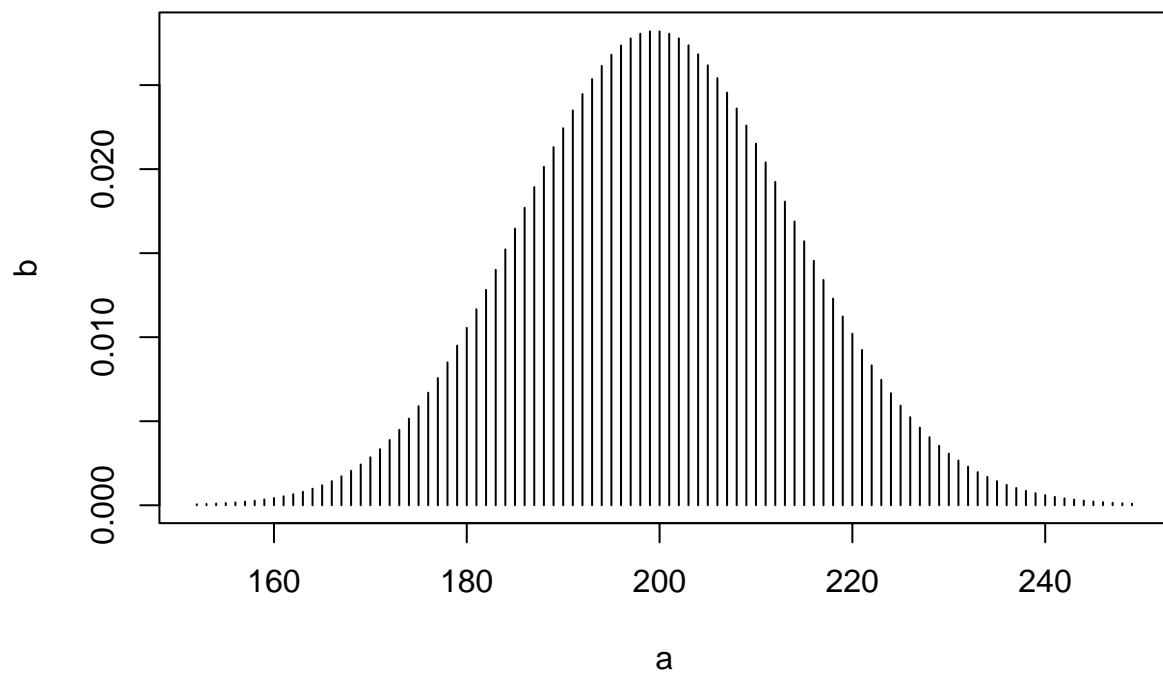
```
## [1] 0.2529326
```

```
#Wykres gęstości
```

```
a <- seq(min(X), max(X), by = 1)
```

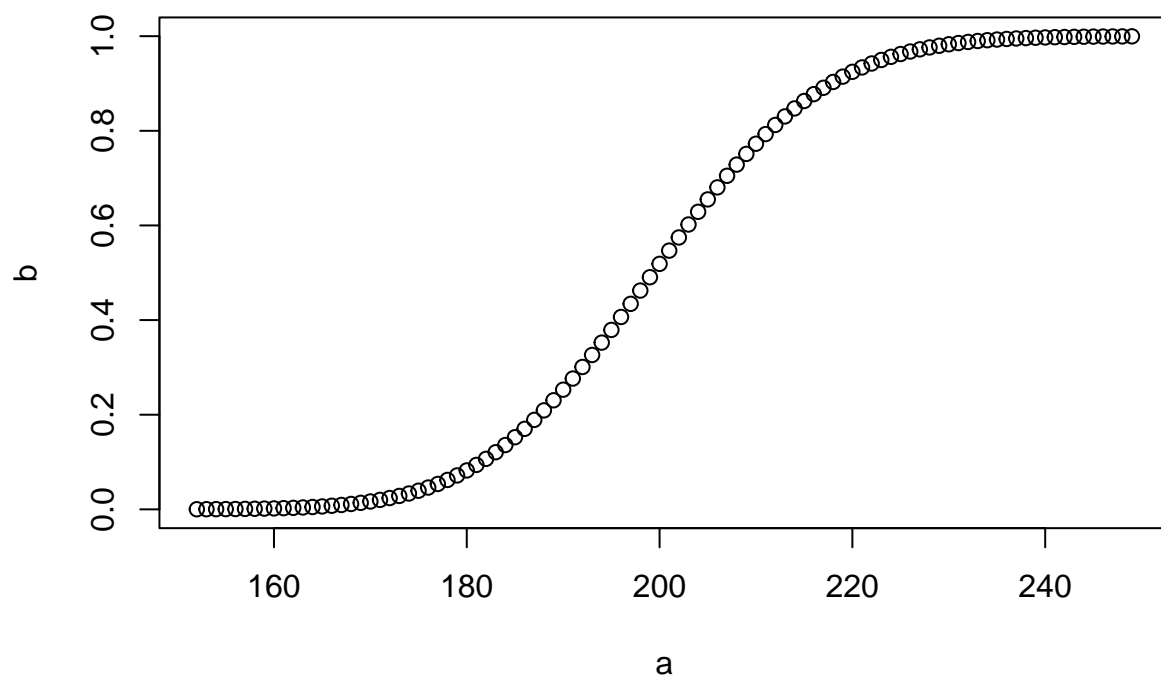
```
b <- dpois(a, lambda)
```

```
plot(a,b, type="h")
```



```
#Wykres dystrybucyjny  
a <- seq(min(X), max(X), by = 1)  
b <- dpois(a, lambda)  
plot(a,b)
```





## Rozkłady ciągłe

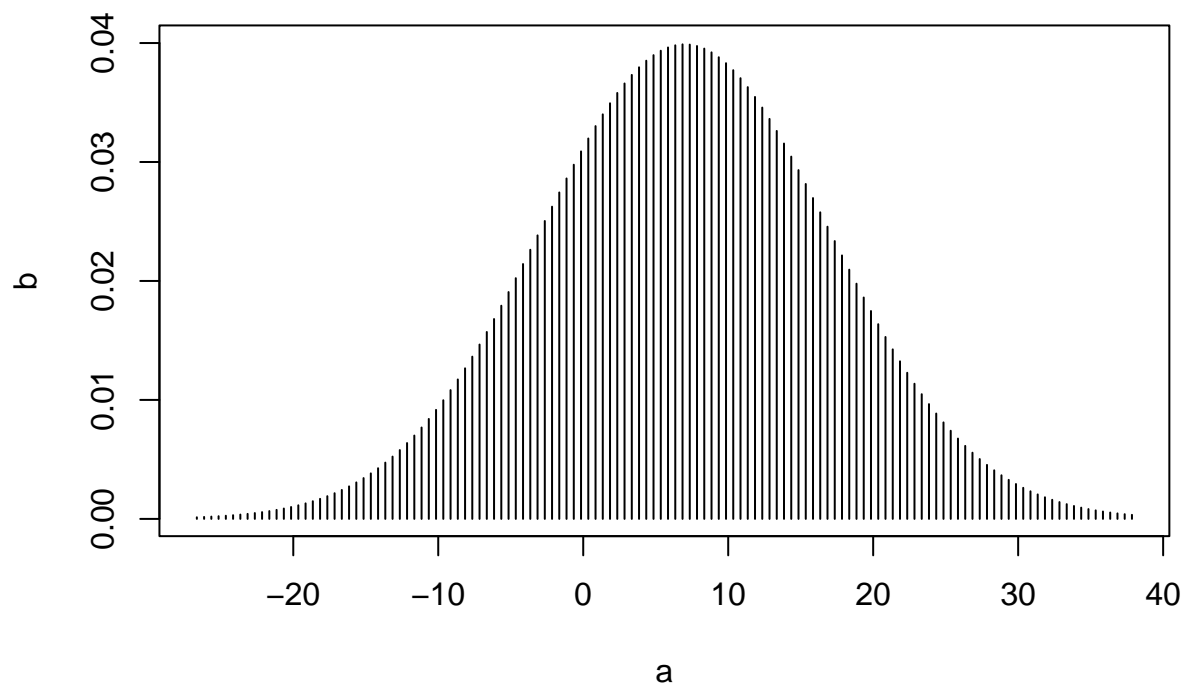
### Rozkład normalny

```
#Generowanie próbek
N <- 1000
mu <- 7 # średnia
sigma <- 10 # odchylenie standardowe
X <- rnorm(N, mu, sigma)

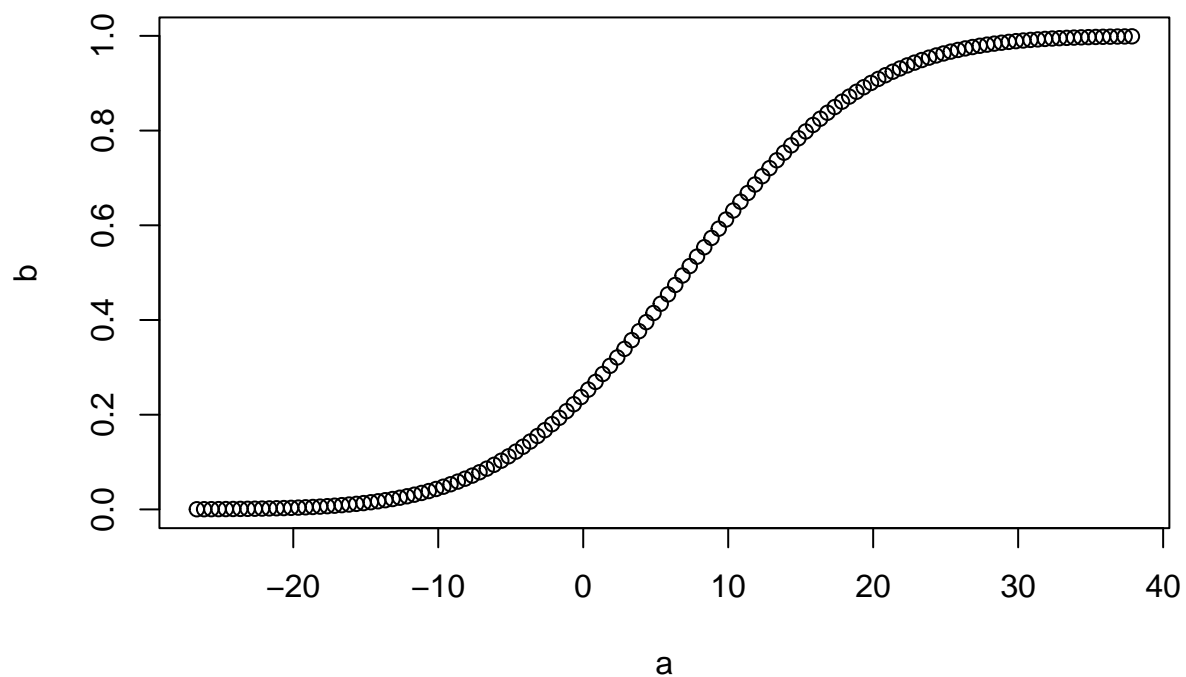
#Prawdopodobieństwa
#Prawd. punktowe w rozkładzie ciągłym = 0
pnorm(10, mu, sigma) # P(X<=10) p. przedziałowe

## [1] 0.6179114

#Wykres gęstości
a <- seq(min(X), max(X), by = 0.5)
b <- dnorm(a, mu, sigma)
plot(a,b, type="h")
```



```
#Wykres dystrybucyjny  
a <- seq(min(X), max(X), by = 0.5)  
b <- pnorm(a, mu, sigma)  
plot(a,b)
```



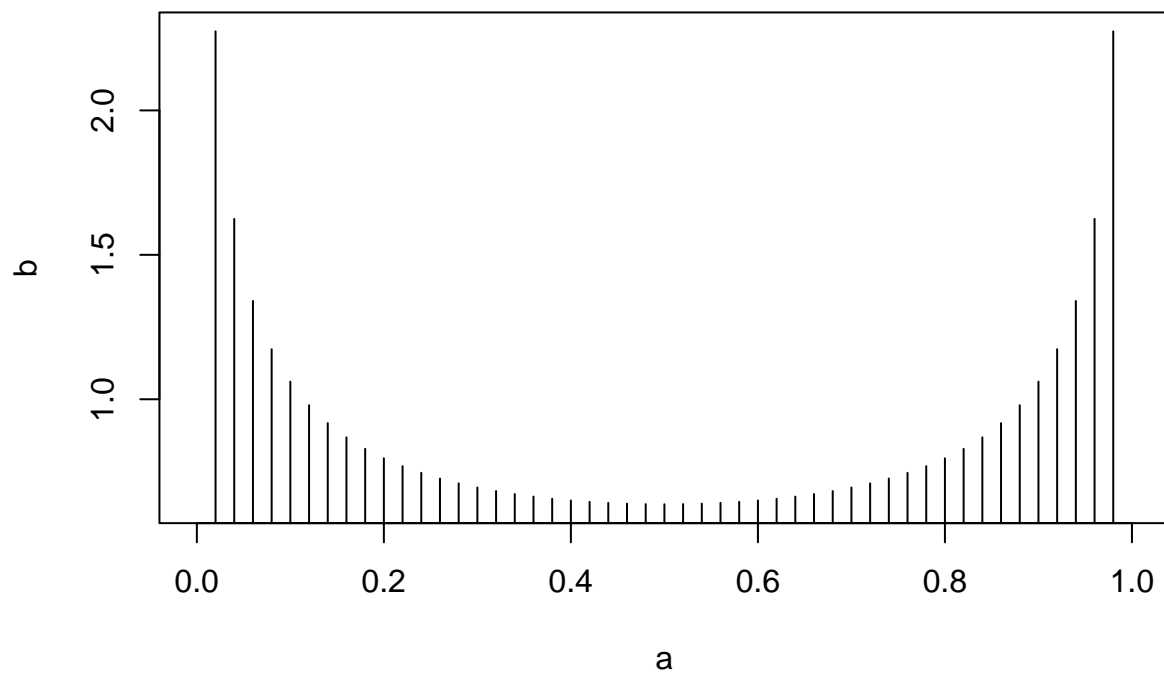
### Rozkład Beta

```
#Generowanie próbki
N <- 1000
alpha <- 0.5
beta <- 0.5
X <- rbeta(N, alpha, beta)

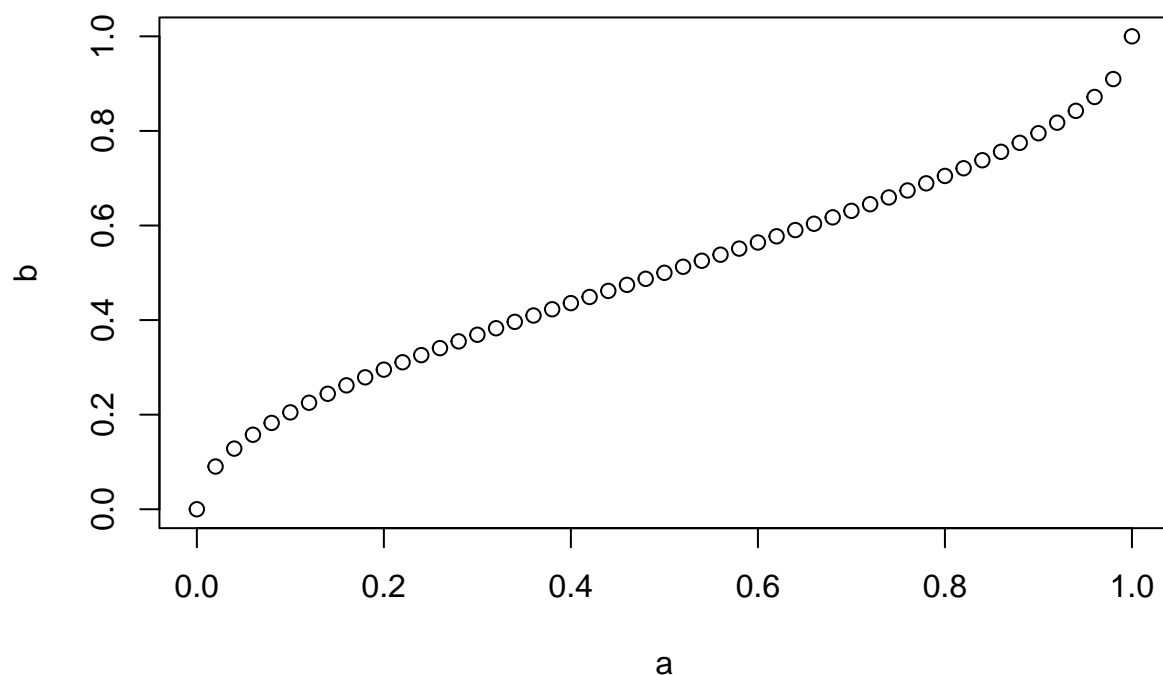
#Prawdopodobieństwa
#Prawd. punktowe w rozkładzie ciągłym = 0
pbeta(0.7, alpha, beta) # P(X<=0.7) p. przedziałowe

## [1] 0.6309899

#Wykres gęstości
a <- seq(0, 1, by = 0.02)
b <- dbeta(a, alpha, beta)
plot(a,b, type="h")
```



```
#Wykres dystrybucyj  
a <- seq(0, 1, by = 0.02)  
b <- pbeta(a, alpha, beta)  
plot(a,b)
```



## VI - Budowa macierzy z bazy danych

Do zbudowanie macierzy wykorzystano wszystkie dostępne numeryczne dane tj. *CenaPLN*, *KM*, *PrzebiegKm*, *RokProdukcji*, *PojemnoscSkokowa*. Dzięki temu możliwe jest wyznaczenie macierzy korelacji pomiędzy danymi.

```
# Select data
vi_database = select(filter(database, PojazdUszkodzony==FALSE & SkrzyniaBiegowManualna == TRUE),
                      c("CenaPLN", "KM", "PrzebiegKm", "RokProdukcji", "PojemnoscSkokowa"))
```

```
#Display head
head(vi_database)
```

```
##   CenaPLN  KM PrzebiegKm RokProdukcji PojemnoscSkokowa
## 1   27900 150    80840         2005           1900
## 2   28000 116   166000         2004           2000
## 3   25500 150   112000         2002           1781
## 4   29900 109    42000         2005           1991
## 5   29800 207   169000         2004           2946
## 6   21400 122   160000         2003           1800
```

```
# Dimension
vi_dim <- dim(vi_database);
cat("Rozmiar macierzy:", vi_dim[1], "x", vi_dim[2]);
```

```
## Rozmiar macierzy: 32646 x 5
```

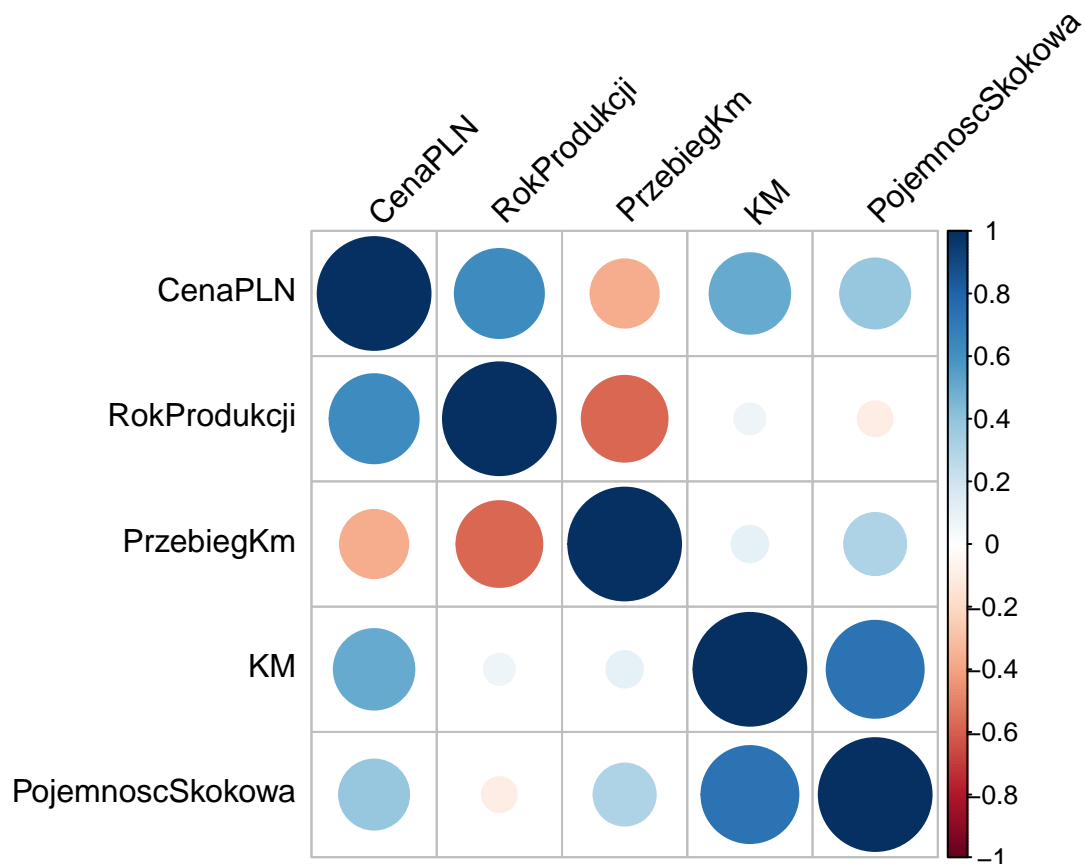
```
# Columns mean
vi_means = round(colMeans(vi_database), 3);
cat(paste(names(vi_means), vi_means, sep = " : ", collapse = ",\n"))

## CenaPLN : 29250.103,
## KM : 104.378,
## PrzebiegKm : 127080.842,
## RokProdukcji : 2005.089,
## PojemnoscSkokowa : 1739.627

# Correlation
vi_corr_matrix = cor(vi_database);
print(vi_corr_matrix)

##              CenaPLN          KM PrzebiegKm RokProdukcji PojemnoscSkokowa
## CenaPLN          1.0000000  0.5096861 -0.3659806  0.62332163  0.38550218
## KM                0.5096861  1.0000000  0.1047219  0.07592120  0.73909520
## PrzebiegKm       -0.3659806  0.1047219  1.0000000 -0.57854449  0.30351805
## RokProdukcji      0.6233216  0.0759212 -0.5785445  1.00000000 -0.09648703
## PojemnoscSkokowa  0.3855022  0.7390952  0.3035180 -0.09648703  1.00000000

corrplot(vi_corr_matrix, order = "hclust",
          tl.col = "black", tl.srt = 45)
```



Jak można zaobserwować: - *CenaPLN* skorelowana jest z *RokProdukcji* (silnie), *KM*, *PojemnoscSkokowa* oraz odwrotnie z *PrzebiegKM*,  
- *PojemnoscSkokowa* jest silnie skorelowana z ilością *KM* (większa pojemność -> więcej *KM*), - *PrzebiegKM* jest odwrotnie skorelowany z *RokProdukcji* (starszy samochód -> większy przebieg).

## VII - Przedziały ufności

W tej sekcji przedstawione zostaną badania określające przedziały ufności z różnym stopniem 'zaufania'. Oznacza to, że jeśli grupa badana była zgromadzona w sposób losowy to rzeczywisty parametr populacji z określonym stopniem 'zaufania' znajduje się w tym przedziale.

### Zmienna numeryczna

W celu określenia przedziałów ufności zmiennej numerycznej wybrano cechę *CenaPLN*. Przedział ufności dla średniej:

```
# Meta data
# Confidence level 0.9, 0.95, 0.99
confidence_level = c(0.9, 0.95, 0.99);

# Basic data
vii_price_mean = mean(database$CenaPLN);
vii_price_sd = sd(database$CenaPLN);
n = length(database$CenaPLN);

# Mean confidence function
mean_confidence <- function(mean, sd, n, confidence_level) {
  alpha = 1 - confidence_level;
  offset = qnorm(1 - alpha / 2) * sd / sqrt(n);
  lower_bound = mean - offset;
  upper_bound = mean + offset;
  return(c(lower_bound, upper_bound));
};

# Calculate confidence
mean_confidence_intervals = sapply(confidence_level, function(conf_level) {
  mean_confidence(vii_price_mean, vii_price_sd, n, conf_level)
});

# Create matrix
mean_confidence_intervals = t(mean_confidence_intervals);
rownames(mean_confidence_intervals) = paste0("Poziom ufności: ", confidence_level);
colnames(mean_confidence_intervals) = c("Dolny przedział", "Górny przedział");

# Display
print("Przedziały ufności średniej dla różnych poziomów ufności:");

## [1] "Przedziały ufności średniej dla różnych poziomów ufności:"
print(mean_confidence_intervals);

##
##          Dolny przedział Górny przedział
## Poziom ufności: 0.9      34413.42      34899.92
## Poziom ufności: 0.95     34366.82      34946.52
## Poziom ufności: 0.99     34275.75      35037.59
```

Przedział ufności dla wariancji:

```
# Meta data
# Confidence level 0.9, 0.95, 0.99
confidence_level = c(0.9, 0.95, 0.99);
```

```

# SD confidence function
sd_confidence <- function(mean, n, confidence_level) {
  alpha = 1 - confidence_level;
  offset = qnorm(1 - alpha / 2) / sqrt(2*n);
  lower_bound = mean / (1 + offset);
  upper_bound = mean / (1 - offset);
  return(c(lower_bound, upper_bound));
};

# Calculate confidence
sd_confidence_intervals = sapply(confidence_level, function(conf_level) {
  sd_confidence(vii_price_mean, n, conf_level);
});

# Create matrix
sd_confidence_intervals = t(sd_confidence_intervals);
rownames(sd_confidence_intervals) = paste0("Poziom ufności: ", confidence_level);
colnames(sd_confidence_intervals) = c("Dolny przedział", "Górny przedział");

# Display
print("Przedziały ufności odchylenia standardowego dla różnych poziomów ufności:");

## [1] "Przedziały ufności odchylenia standardowego dla różnych poziomów ufności:"
print(sd_confidence_intervals);

##
##          Dolny przedział Górny przedział
## Poziom ufności: 0.9      34455.78      34859.92
## Poziom ufności: 0.95     34417.56      34899.13
## Poziom ufności: 0.99     34343.11      34976.01

```

## Przedział ufności dla zmiennej jakościowej (frakcyjna)

Do wyznaczenia przedziału ufności dla zmiennej *PojazdUszkodzony* (rozkład Bernoulliego)

```

# Meta data
# Confidence level 0.9, 0.95, 0.99
confidence_level = c(0.9, 0.95, 0.99);

# Basic data
vii_positive = sum(database$PojazdUszkodzony == TRUE);
vii_all      = length(database$PojazdUszkodzony);

mean_bool_confidence <- function(m, n, confidence_level) {
  alpha = 1 - confidence_level;
  mn = m / n;
  offset = qnorm(1 - alpha / 2) * sqrt(mn * (1-mn)) / sqrt(n);
  lower_bound = mn - offset;
  upper_bound = mn + offset;
  return(c(lower_bound, upper_bound));
};

# Calculate confidence
confidence_intervals = sapply(confidence_level, function(conf_level) {
  mean_bool_confidence(vii_positive, vii_all, conf_level);
});

```



```
});

# Create matrix
confidence_intervals = t(confidence_intervals);
rownames(confidence_intervals) = paste0("Poziom ufności: ", confidence_level);
colnames(confidence_intervals) = c("Dolny przedział", "Górny przedział");

# Display
print("Przedziały ufności dla frakcji dla różnych poziomów ufności (Udział pojazdów uszkodzonych):");

## [1] "Przedziały ufności dla frakcji dla różnych poziomów ufności (Udział pojazdów uszkodzonych):"
print(confidence_intervals);

##              Dolny przedział Górny przedział
## Poziom ufności: 0.9      0.03528564      0.03839196
## Poziom ufności: 0.95    0.03498810      0.03868950
## Poziom ufności: 0.99    0.03440657      0.03927103
```

## VII - Testowanie hipotez

### Zmienne jakościowe

#### Test niezależności

Do testu niezależności wykorzystano dane dotyczące kraju pochodzenia oraz marki. Celem testu jest potwierdzenie niezależności wyboru marki względem kraju. H0 - zmienne są niezależne, H1 - zmienne nie są niezależne.

```
print(table(database$KrajPochodzenia, database$Marka));
```

```
##
##              Audi  BMW  Fiat  Ford  Mercedes-Benz  Opel  Peugeot  Renault
## Austria          48   38   41   75              19   67         47       73
## Belgia          204  154   51  246              86  427        171      264
## Czechy          123  125   56  254             102  191        147      226
## Dania             2    1    5   18               2   31         16        6
## Estonia           0    0    0    1               0    0          0         0
## Francja          139  102  101  402             117  322        817     1116
## Grecja            0    0    0    0               0    0          0         0
## Hiszpania         4    7    3   27               7   34          7         10
## Holandia          27   28   21  111              21  141         84        140
## Irlandia          0    0    0    0               1    2          0         0
## Islandia           0    0    0    0               0    1          0         0
## Kanada            4    5    0    3               3    0          0         0
## Luksemburg        14   10   11   14               7   27         19         30
## Niemcy           2080 1210  400 2096             1150 2119        667     1436
## Norwegia          0    0    0    0               0    1          0         0
## Polska           423  301 1133 1842             641 1459       1098     1242
## Rosja             0    0    0    0               0    0          0         0
## Rumunia           0    0    0    0               0    0          0         2
## Slowacja          1    4    0    2               1    0          0         0
## Stany Zjednoczone  90   64    0  101              72    2          0         0
## Szwajcaria        55   35   10   36              15   41         19         39
## Szwecja           5    1    0    3               3    2          1         1
```

```
##   Węgry           0    0    0    1           0    0           1    0
##   Wielka Brytania 29   23   4   15          15   17          13   19
##   Włochy          74  109   87  146          57  107          92  110
##
##           Skoda Toyota Volkswagen
##   Austria         32    31       108
##   Belgia          88   137       305
##   Czechy         953   142       243
##   Dania           2    36        22
##   Estonia         0    0         1
##   Francja         81   176       410
##   Grecja          0    1         0
##   Hiszpania        5    7        17
##   Holandia        29   44       161
##   Irlandia        0    0         1
##   Islandia        0    0         0
##   Kanada          0    2         4
##   Luksemburg       5   12        22
##   Niemcy          531   580      3271
##   Norwegia         0    2         1
##   Polska          1154  1297      1303
##   Rosja            0    0         1
##   Rumunia          0    0         0
##   Slowacja         3    1         1
##   Stany Zjednoczone 0   119       73
##   Szwajcaria       15   11        30
##   Szwecja          0    1         6
##   Węgry            0    0         0
##   Wielka Brytania   7    5        39
##   Włochy           23   65       101
```

```
vii_test = chisq.test(table(database$Marka, database$SkrzyniaBiegowManualna));
print(vii_test)
```

```
##
##   Pearson's Chi-squared test
##
## data:  table(database$Marka, database$SkrzyniaBiegowManualna)
## X-squared = 8406.9, df = 10, p-value < 2.2e-16
```

Test to potwierdza ( $p\text{-value} < 0.05$ ) mamy podstawy do odrzucenia hipotezy zerowej na korzyść hipotezy alternatywnej - istnieje zależność krajem pochodzenia, a marką.

## Test proporcji

Test proporcji w którym wykorzystano liczbę samochodów uszkodzonych oraz kraj pochodzenie.  $H_0$  - odsetek samochodów uszkodzonych jest niezależny od kraju pochodzenia.

```
vii_proporcja = database %>% group_by(database$KrajPochodzenia) %>% summarise(liczba = n());
vii_proporcja$uszkodzonych = table(database$KrajPochodzenia, database$PojazdUszkodzony)[,2];
vii_proporcja = vii_proporcja %>% filter(liczba > 500);
vii_test = prop.test(x = vii_proporcja$uszkodzonych, n = vii_proporcja$liczba);
print(vii_test);
```

```
##
##   9-sample test for equality of proportions without continuity correction
```

```
##
## data: vii_proporcja$uszkodzonych out of vii_proporcja$liczba
## X-squared = 1059.7, df = 8, p-value < 2.2e-16
## alternative hypothesis: two.sided
## sample estimates:
##      prop 1      prop 2      prop 3      prop 4      prop 5      prop 6      prop 7
## 0.06735751 0.05766526 0.00117096 0.11895321 0.02850062 0.02683398 0.02009585
##      prop 8      prop 9
## 0.06333973 0.02265705
```

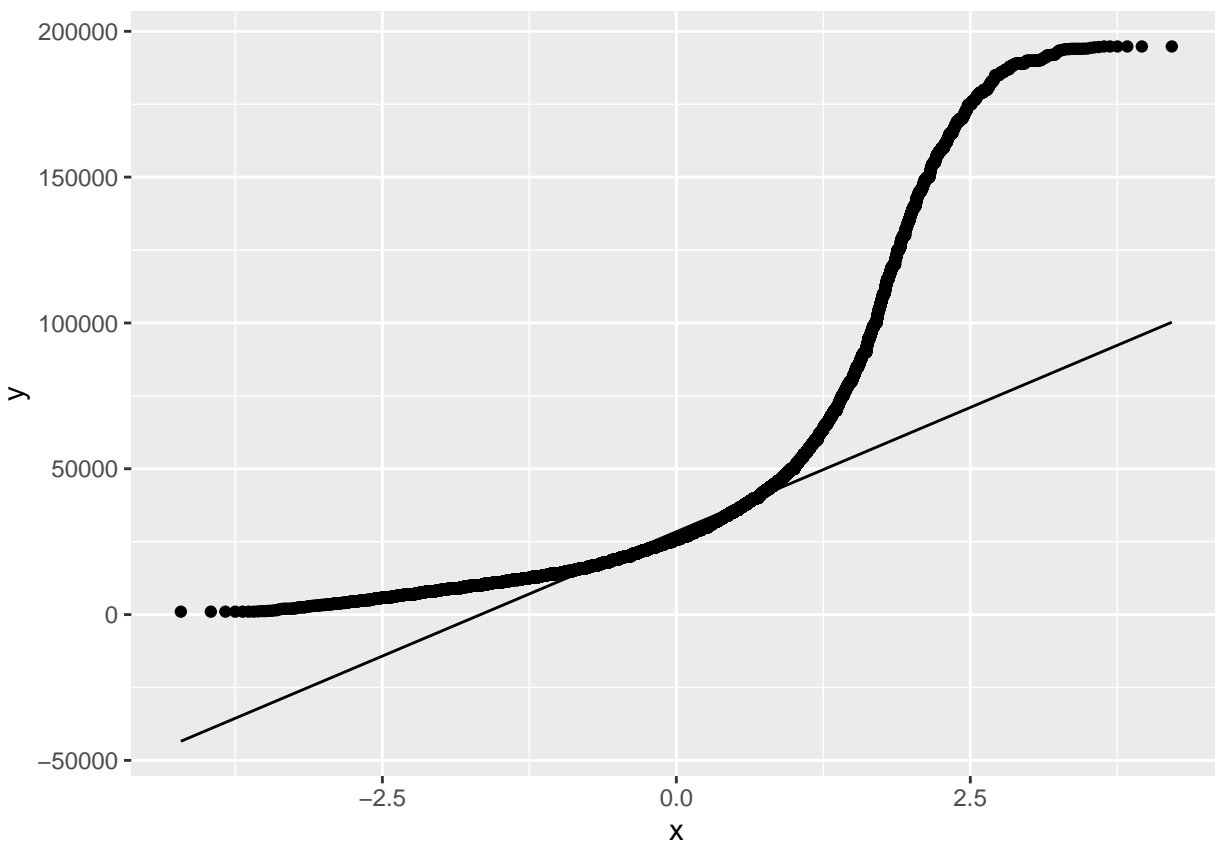
## Zmienne ilościowe

Testowanie normalności względem ceny, test pozwala na potwierdzenie tezy o normalności rozkładu.  $H_0$  - doświadczenie ma rozkład normalny.

```
vii_test = ad.test(database$CenaPLN);
print(vii_test);
```

```
##
## Anderson-Darling normality test
##
## data: database$CenaPLN
## A = 3123.5, p-value < 2.2e-16
```

```
ggplot(database, aes(sample = CenaPLN)) +
  stat_qq() +
  stat_qq_line();
```



Cena nie jest zmienną o rozkładzie normalnym.

## Test Wilcoxona

Ze względu na to, iż cena nie jest zmienną o rozkładzie normalnym. Wykorzystujemy nieparametryczny test Wilcoxona.  $H_0$  - średnia jest równa 34656.

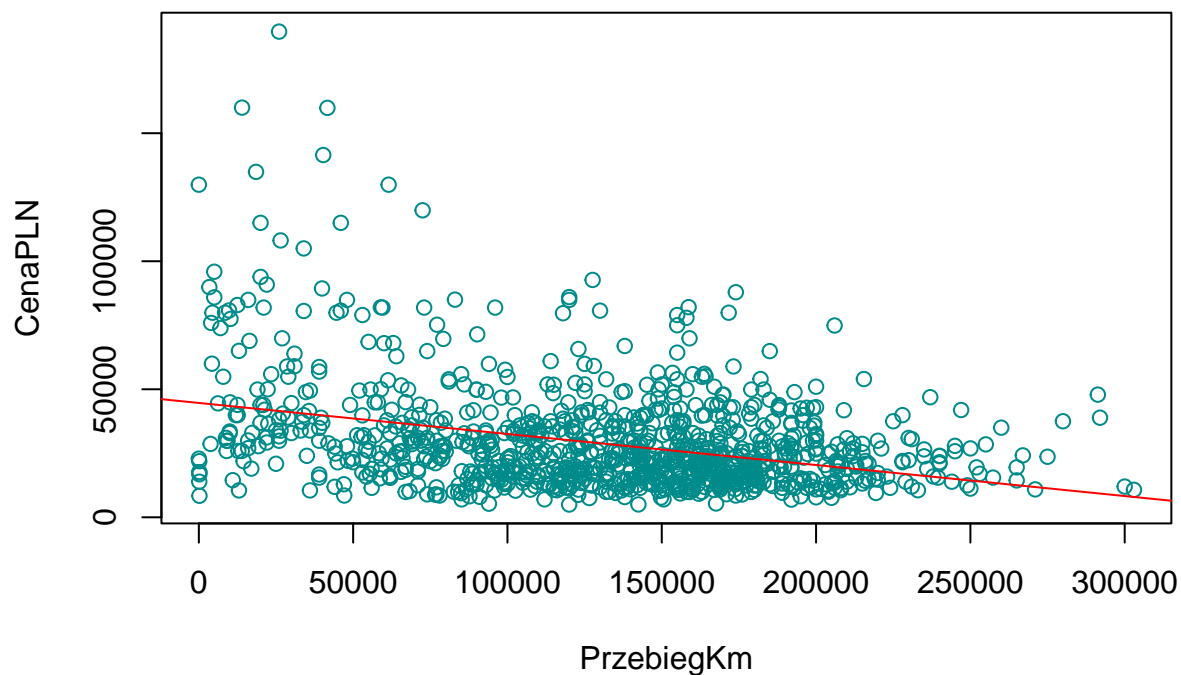
```
wilcox.test(x = database$CenaPLN, mu = 34656)

##
## Wilcoxon signed rank test with continuity correction
##
## data: database$CenaPLN
## V = 277239234, p-value < 2.2e-16
## alternative hypothesis: true location is not equal to 34656
```

## IX - Regresja liniowa i inne

### Modele dla jednej zmiennej

```
ix_database = select(filter(database, PojazdUszkodzony==FALSE & SkrzyniaBiegowManualna == TRUE),
                      c("CenaPLN", "PrzebiegKm"))
model = lm(CenaPLN ~ ., data = ix_database)
ix_sample = ix_database[sample(nrow(ix_database), size=1000),]
plot(CenaPLN ~ PrzebiegKm, data=ix_sample, col = 'darkcyan')
abline(model, col="red")
```



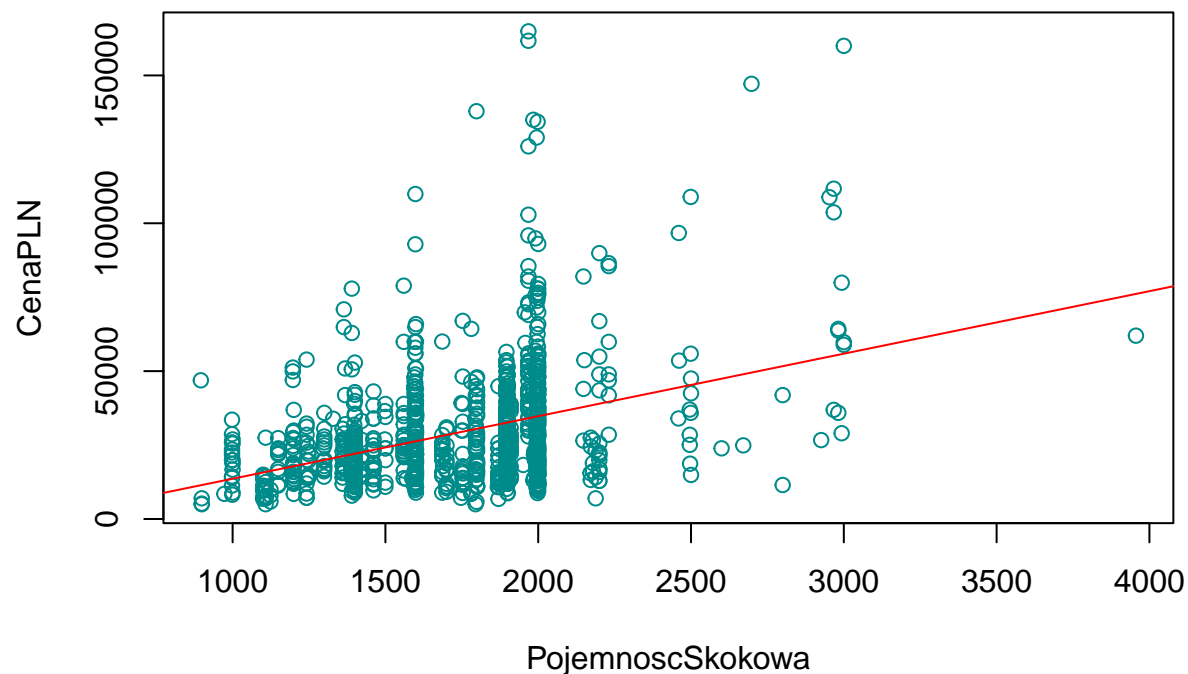
```
summary(model);
```

```
##
## Call:
## lm(formula = CenaPLN ~ ., data = ix_database)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -40733 -11549  -3678   6986 152257
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  4.465e+04  2.398e+02  186.17  <2e-16 ***
## PrzebiegKm  -1.212e-01  1.706e-03  -71.05  <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 18550 on 32644 degrees of freedom
## Multiple R-squared:  0.1339, Adjusted R-squared:  0.1339
## F-statistic: 5049 on 1 and 32644 DF, p-value: < 2.2e-16

print(model$coefficients);

##      (Intercept)      PrzebiegKm
## 44651.6424984      -0.1211948

ix_database = select(filter(database, PojazdUszkodzony==FALSE & SkrzyniaBiegowManualna == TRUE),
                      c("CenaPLN", "PojemnoscSkokowa"))
model = lm(CenaPLN ~ ., data = ix_database)
ix_sample = ix_database[sample(nrow(ix_database), size=1000),]
plot(CenaPLN ~ PojemnoscSkokowa, data=ix_sample, col = 'darkcyan')
abline(model, col="red")
```



```
summary(model);
```

```
##
## Call:
## lm(formula = CenaPLN ~ ., data = ix_database)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -106802  -11243   -3490    6365   159245
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  -7527.2394   497.7274  -15.12  <2e-16 ***
## PojemnoscSkokowa    21.1409     0.2801   75.49  <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 18390 on 32644 degrees of freedom
## Multiple R-squared:  0.1486, Adjusted R-squared:  0.1486
## F-statistic: 5698 on 1 and 32644 DF, p-value: < 2.2e-16
```

```
print(model$coefficients);
```

```
##      (Intercept) PojemnoscSkokowa
##      -7527.23942         21.14093
```

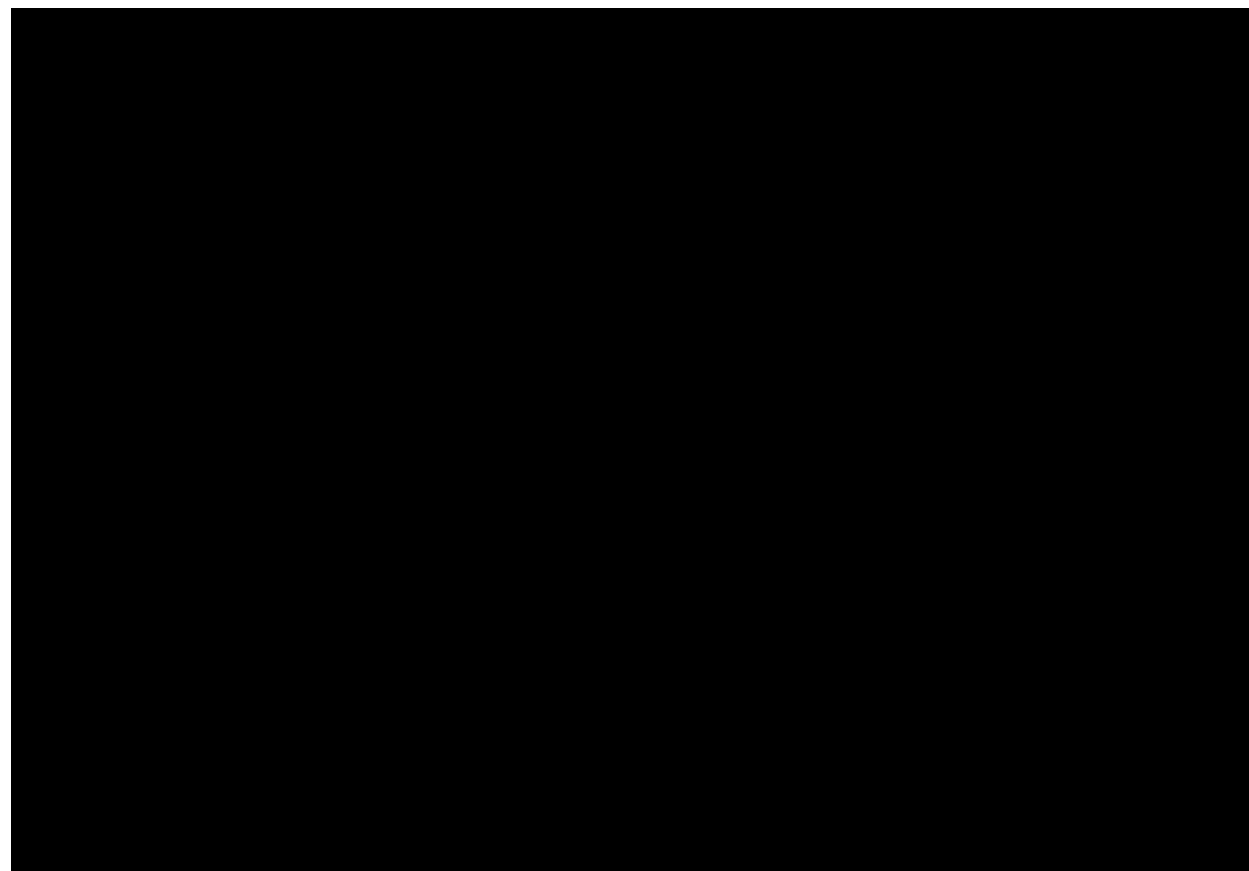
## Modele dla 2 zmiennych

```
ix_database = select(filter(database, PojazdUszkodzony==FALSE & SkrzyniaBiegowManualna == TRUE),
                      c("CenaPLN", "PrzebiegKm", "PojemnoscSkokowa"))
model = lm(CenaPLN ~ . , data = ix_database)

plot3d(model, col='darkcyan', plane.col = 'red')
rglwidget()
```

```
## Warning in snapshot3d(scene = x, width = width, height = height): webshot =
## TRUE requires the webshot2 package and Chrome browser; using rgl.snapshot()
## instead

## Warning in rgl.snapshot(filename, fmt, top): this build of rgl does not support
## snapshots
```



```
summary(model);
```

```
##
## Call:
## lm(formula = CenaPLN ~ . , data = ix_database)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -156582   -8871   -1542    6472   138780
##
## Coefficients:
```

```
##              Estimate Std. Error  t value Pr(>|t|)
## (Intercept)    -5.437e+02  4.200e+02   -1.294    0.196
## PrzebiegKm     -1.762e-01  1.483e-03 -118.785   <2e-16 ***
## PojemnoscSkokowa 3.000e+01  2.456e-01  122.129   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 15370 on 32643 degrees of freedom
## Multiple R-squared:  0.4056, Adjusted R-squared:  0.4055
## F-statistic: 1.114e+04 on 2 and 32643 DF,  p-value: < 2.2e-16
```

```
print(model$coefficients);
```

```
##      (Intercept)      PrzebiegKm PojemnoscSkokowa
##      -543.7320918      -0.1761712      29.9959737
```

## Drzewa decyzyjne

```
ix_database = select(filter(database, PojazdUszkodzony==FALSE & SkrzyniaBiegowManualna == TRUE),
                      c("CenaPLN", "RokProdukcji", "PojemnoscSkokowa"))
ix_data_split = initial_split(ix_database, prop = 0.75);
ix_train_data <- training(ix_data_split);
ix_test_data <- testing(ix_data_split);

tree_spec <- decision_tree() %>%
  set_engine("rpart") %>%
  set_mode("regression")

tree_fit <- tree_spec %>%
  fit(CenaPLN ~ ., data = ix_train_data)

predictions <- tree_fit %>%
  predict(ix_test_data) %>%
  pull(.pred)

metrics <- metric_set(rmse, rsq)
model_performance <- ix_test_data %>%
  mutate(predictions = predictions) %>%
  metrics(truth = CenaPLN, estimate = predictions)

print(model_performance)

## # A tibble: 2 x 3
##   .metric .estimator .estimate
##   <chr>   <chr>      <dbl>
## 1 rmse    standard    11106.
## 2 rsq     standard      0.677

rpart.plot(tree_fit$fit, type = 5, extra = 101, under = TRUE, cex = 0.8, box.palette = "auto")

## Warning: Cannot retrieve the data used to build the model (so cannot determine roundint and is.binary)
## To silence this warning:
##   Call rpart.plot with roundint=FALSE,
##   or rebuild the rpart model with model=TRUE.
```



