Politechnika Świętokrzyska

Analiza i wizualizacja danych

Temat: Szacowanie pulsu na podstawie wieku pacjenta	**Wykonujący:** Anna Tutaj Kamil Tomczyk	**Grupa:** 1ID21B
		Data oddania: 06.11.2016r.

1. Warunki sprzętowe i programowe (system, środowisko, biblioteki)

Środowiskiem, w którym zaprojektowano aplikację było Microsoft Visual Studio wersja Community 2015, z użyciem Windows Forms oraz języka C#. Wykorzystano bibliotekę ZedGraph do rysowania wykresu. Podczas projektowania aplikacji pracowano na systemie operacyjnym Windows 7, a sama aplikacja działa docelowo na platformie Windows.

2. Generowanie danych i opis zmiennych

Na potrzeby projektu został napisany generator zawarty w klasie Generator. Dane są zapisywane do pliku .csv w formacie: wiek,puls. Wartości danych zostały oparte na poniższej tabeli, której wartości pochodzą ze strony z załącznika [1].

Wiek	Średnia liczba uderzeń na min	Maksymalna liczba uderzeń na min
20	100-170	200
25	98-166	195
30	95-162	190
35	93-157	185
40	90-153	180
45	88-149	175
50	85-145	170
55	83-140	165
60	80-136	160
65	78-132	155
70	75-128	150

W skład 120 danych wchodzą:

- po 10 danych dla każdego z 11 wieku (20, 25, … , 70),
- 6 danych z niedozwolonego zakresu (ujemne wartości, stanowczo za duże wartości),
- pusty rekord,
- 3 dane z maksymalną wartością pulsu.

Błędne oraz puste dane stanowią 5-6% wszystkich danych.

3. Opis kodu aplikacji

Solucja zawiera jeden projekt Windows Form, a jego trzy najważniejsze klasy zostaną omówione.

- **Calculator.cs** – odpowiada za wykonywanie obliczeń. Udostępnia wyniki klasie Form. Zawiera zadeklarowane tablice oraz zmienne na wartości takie jak minimum, mediana, czy odchylenia standardowe. Jej najważniejsze metody:
  - **void readFile() - pobiera dane** z pliku **i zapisuje** do tablicy typu double[,]. Dokonuje pierwszej części **wstępnej obróbki danych,** ponieważ gdy rekord z pliku jest niemożliwy do przekonwertowania na wartości typu double (np. pusta wartość, litera), to dana zostaje pominięta,
  - **void checkValues()** - dokonuje drugiej części **obróbki danych**. Usuwa dane, wykraczające poza przedziały – wiek [10- 100], puls [50- 100],
  - **double getMax(double[,] array2D, int targetParameter) –** zwraca wartość maksymalną z tablicy po obranym parametrze. TargetParameter= 0 odpowiada idAge, natomiast idPulse= 1,
  - **double getMin(double[,] array2D, int targetParameter –** analogicznie do getMax(),
  - **double getExpectedValue(double[,] array2D, int targetParameter) –** zwraca sumę wartości podzieloną na ich liczbę,
  - **double getMedian(double[,] array2D, int targetParameter) –** zwraca medianę. Najpierw sortuje tabelę rosnąco wg zadanego parametru. Gdy liczba wartości jest parzysta - liczy średnią arytmetyczną z dwóch środkowych wartości, a gdy nieparzysta - zwraca liczbę o środkowym indeksie,
  - **double getQ1(double[,] array2D, int targetParameter)*–** zwraca pierwszy kwantyl,
  - **dpuble getQ3(double[,] array2D, int targetParameter)*–** zwraca trzeci kwantyl,
  - **double getIQR(double[,] array2D, int targetParameter) –** zwraca rozstęp międzykwantylowy,
  - **void getindividualPoints(double[,] array2D, int targetParameter) –** oblicza punkty oddalone i zapisuje je do tabeli. Są to wartości, które są położone o przynajmniej 1.5 IQR poniżej Q1 lub przynajmniej o 1.5 IQR powyżej Q3,
  - **double getStandardDeviation(double[,] array2D, int targetParameter) –** zwraca odchylenie standardowe, czyli pierwiastek z wariancji,
  - **double getVariance(double[,] array2D, int targetParameter) –** zwraca wariancję wg wzoru (x – wartość, n – liczba wartości):

$$S^2 = \frac{\sum(x_i - \bar{x})^2}{n}$$

- **double getPearsonsR(double[,] array2D) –** zwraca współczynnika korelacji liniowej Pearsona liczony zgodnie ze wzorem [3]:

$$r_{xy} = \frac{\frac{1}{n}\sum X_i Y_i - \overline{X}\,\overline{Y}}{\sigma_X \cdot \sigma_Y}$$

gdzie

X_i, Y_i - i-te wartości obserwacji z populacji X i Y

$\overline{X}, \overline{Y}$ - średnie z populacji X i Y

σ_x , σ_y- odchylenie standardowe populacji X i Y

n- ilość obserwacji (X i Y mają tyle samo obserwacji)

- **void generateAdditionalData() -** generuje tabele z wiekiem oraz estymowaną wartością pulsu zwracaną przez linearRegression(EstimatedX[i]),
- **void calculateValues() -** wywołuje wszystkie potrzebne metody, które wykonują obliczenia odpowiadające za **analizę statystyczną** i ustawia odpowiednie pola właściwości. Wynika to z hermetyzacji danych. Pola klasy są prywatne, publiczne są natomiast właściwości dostępne przez akcesory get oraz set.

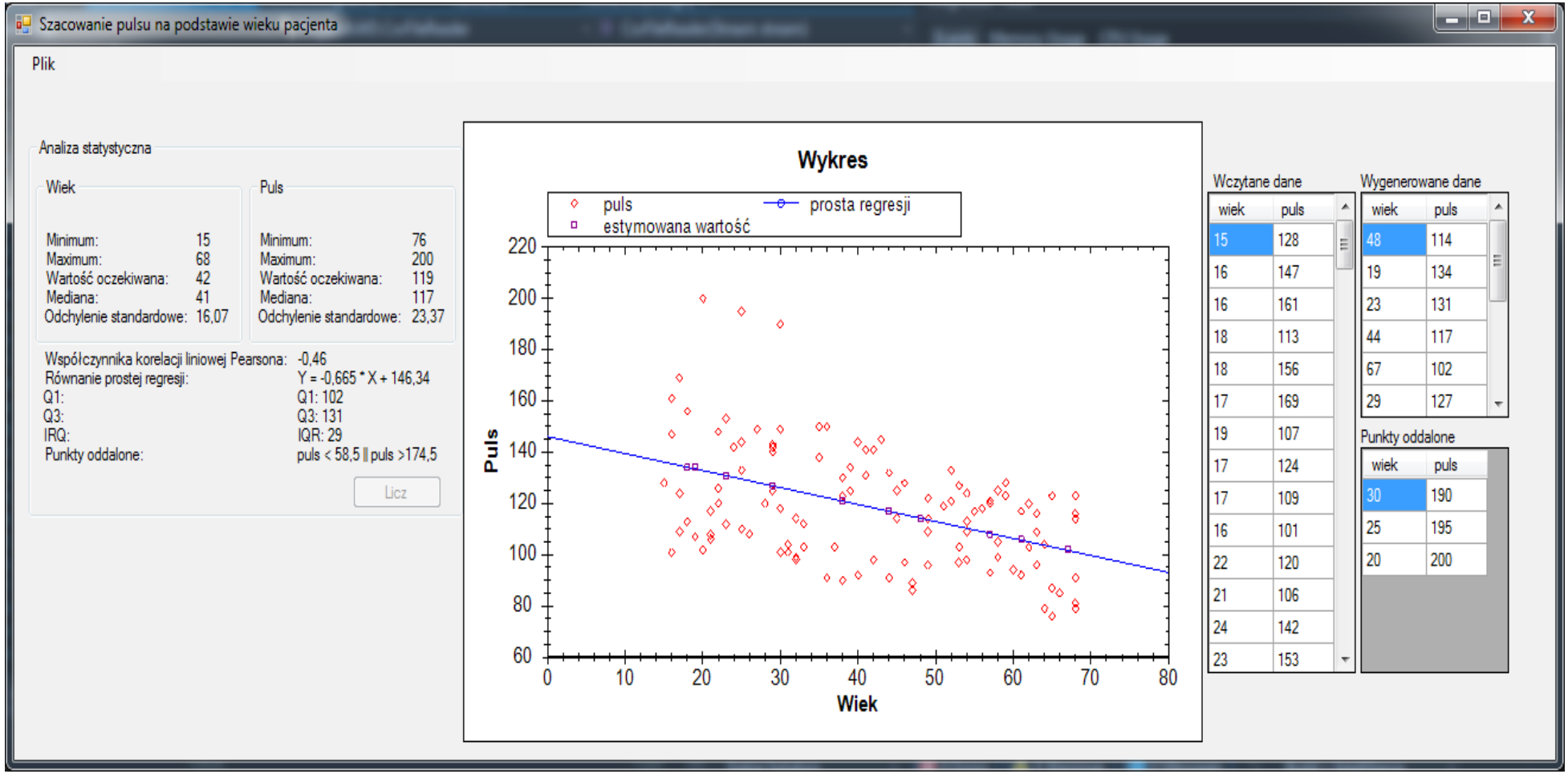
- **Form1.cs** – odpowiada za komunikację użytkownika z logiką aplikacji. Odpowiada na żądania, odświeża widok.
  - **void showValues() -** wywołuje metodę **calculateValues()** i odświeża widok,
  - **void showPoints(double[,] newArray2D) –** rysuje na grafie wszystkie punkty pobrane z tabeli,
  - **void showlinearRegression(double[,] newArray2D) –** rysuje na grafie prostą regresji w oparciu o dwa punkty z tabeli,
  - **void setDataGridAllPoints(double[,] newArray2D) –** wypełnia dataGrid danymi,

- **void loadToolStripMenuItem_Click(object sender, EventArgs e) –** odpowiada na żądanie załadowania danych,
- **void dataPreprocessingToolStripMenuItem_Click(object sender, EventArgs e) –** odpowiada na żądanie wstępnej obróbki danych,
- **void generateAdditionalDataToolStripMenuItem_Click(object sender, EventArgs e) –** odpowiada na żądanie wygenerowania dodatkowych danych w oparciu o prostą regresji,
- **void btnCalculate_Click(object sender, EventArgs e) –** odpowiada na żądanie dokonania analizy statystycznej,
- **void ageTightToolStripMenuItem_Click(object sender, EventArgs e) –** odpowiada na żądanie wygenerowania pliku z danymi, w których wartość wieku wynosi: 20, 25, 30, 35, …, 60, 65, 70,
- **void ageWidthToolStripMenuItem_Click(object sender, EventArgs e) -**
- odpowiada na żądanie wygenerowania pliku z danymi, w których wartość wieku przyjmuje losowe wartości od 15 do 70

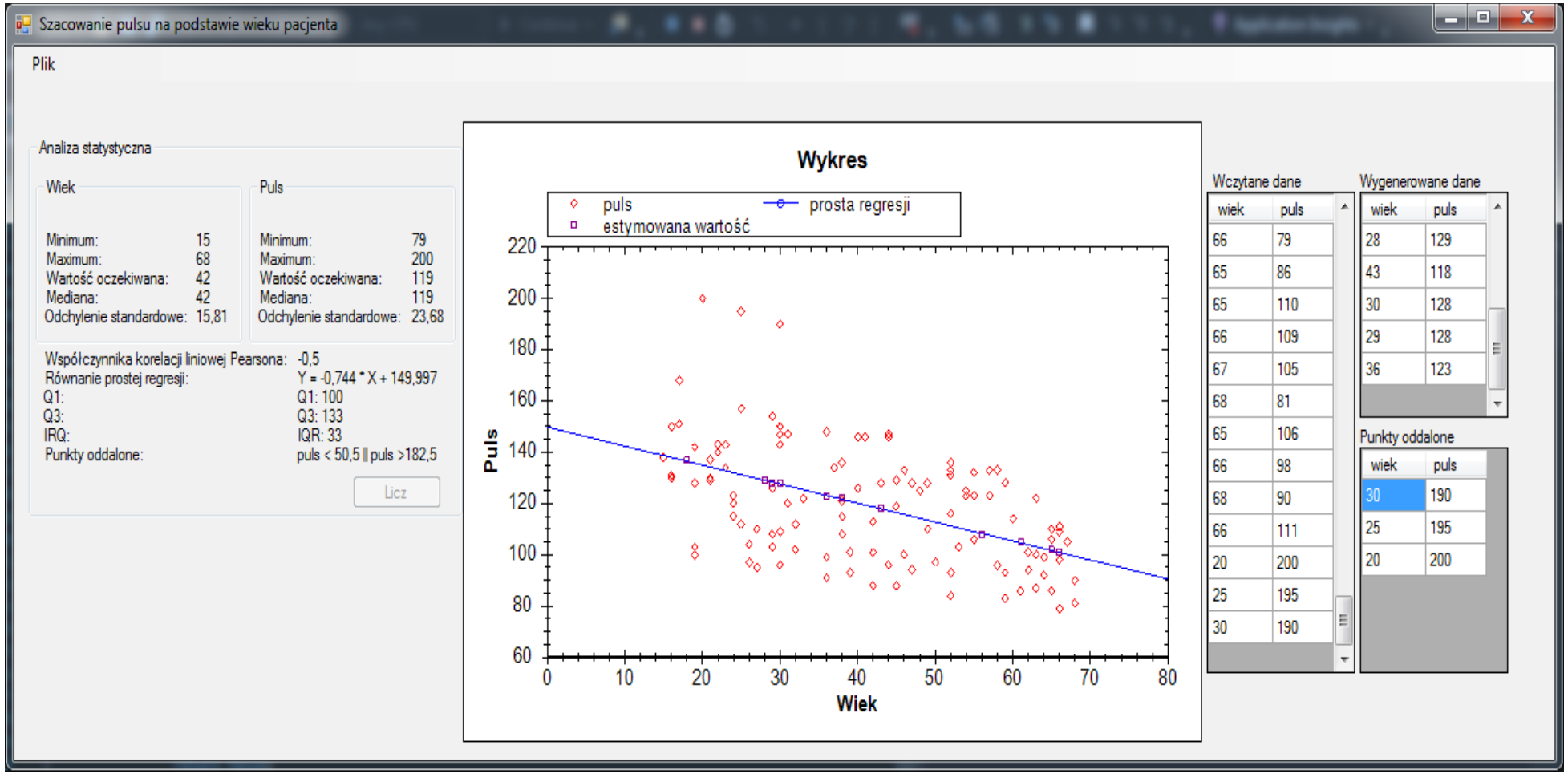
- **Generator –** odpowiada za generowanie danych w sposób omówiony w rozdziale 2.

* Istnieje kilka metod obliczania kwantyli. W projekcie wykorzystano następujący: „Wyznaczamy medianę i na jej podstawie znów dzielimy populację na dwa podzbiory. Jeśli mieliśmy do czynienia z nieparzystą liczbą obserwacji i mediana jest rzeczywistą wartością środkową (nie musieliśmy obliczać średniej arytmetycznej z dwóch sąsiednich liczb), to w takiej sytuacji uwzględniamy ją w obydwóch podzbiorach. Jeśli liczba obserwacji była parzysta, medianę liczyliśmy jako średnią z dwóch liczb środkowych, to w takiej sytuacji nie uwzględniamy mediany w żadnym podzbiorze. Podobnie jak w pierwszej metodzie, obliczamy mediany dla obu podzbiorów i w ten sposób otrzymujemy pierwszy i trzeci kwartyl.[2]”

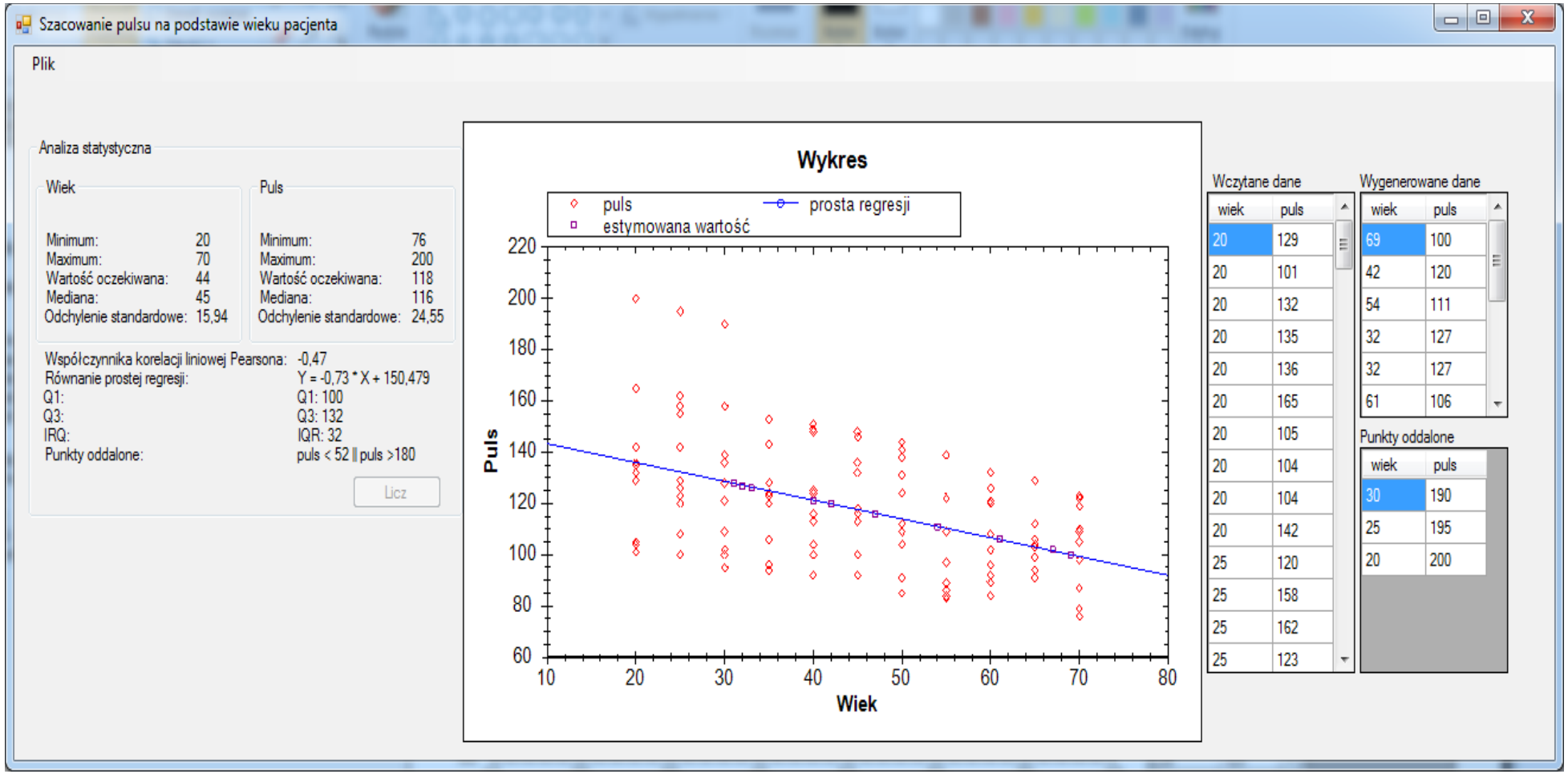
4. Przykładowe zrzuty ekranu działania programu



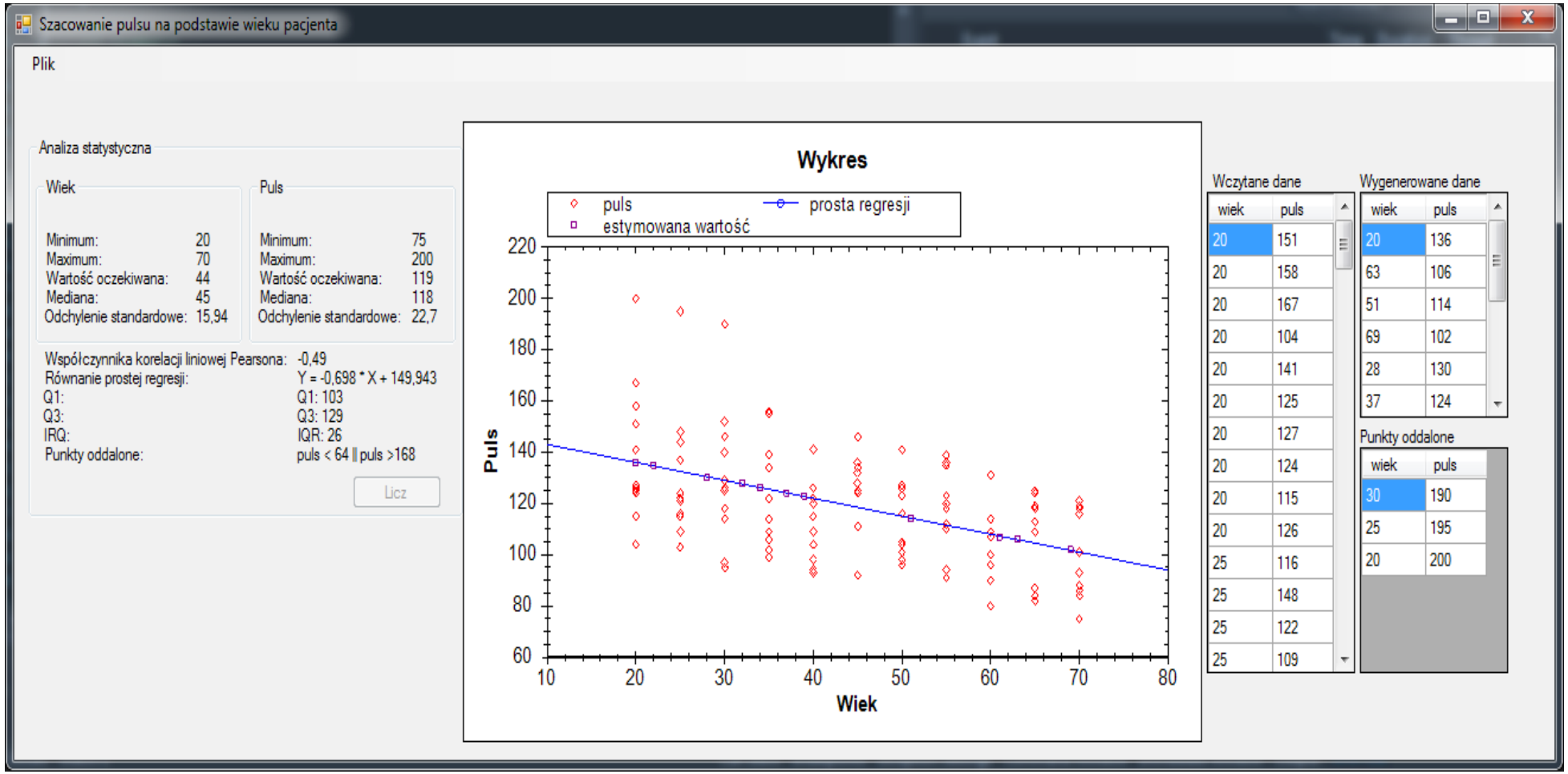
Ilustracja 1: Przykładowy wynik dla danych z szerszego przedziału wiekowego



Ilustracja 2: Przykładowy wynik dla danych z szerszego przedziału wiekowego



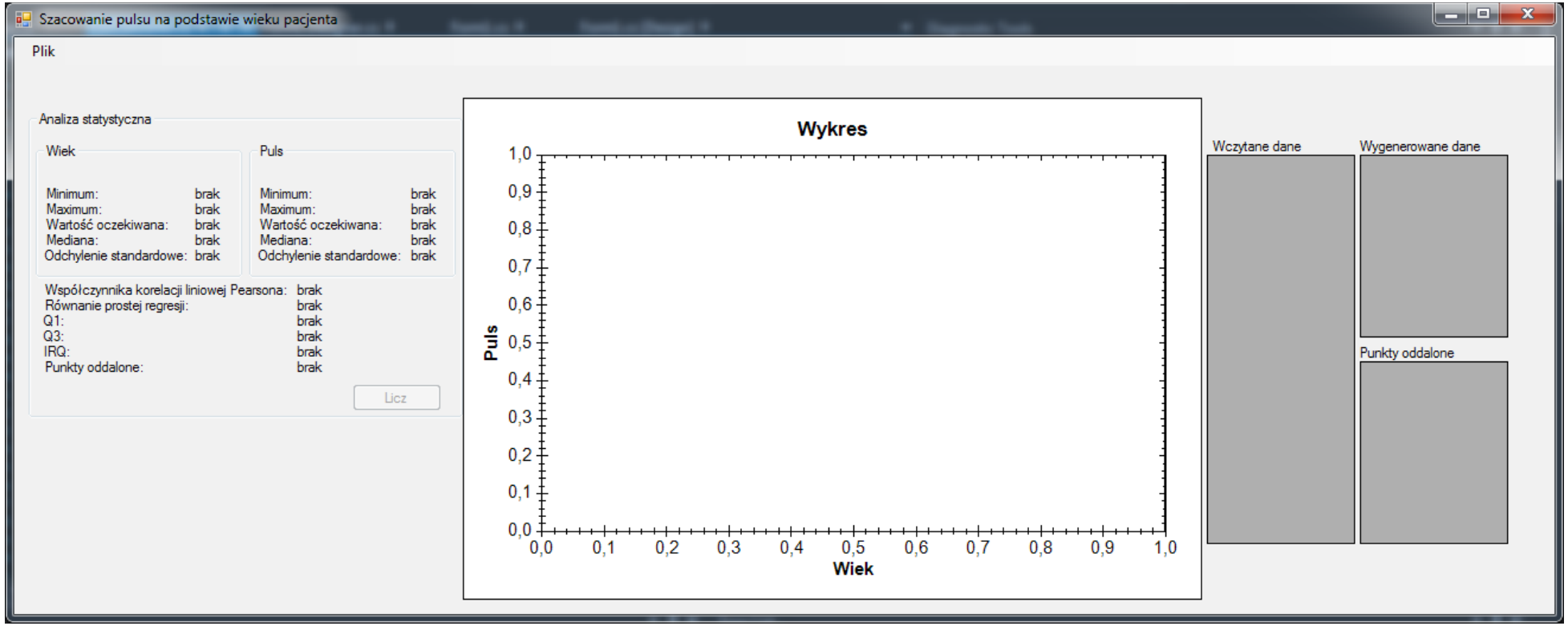
Ilustracja 3: Przykładowy wynik dla danych z węższego przedziału wiekowego



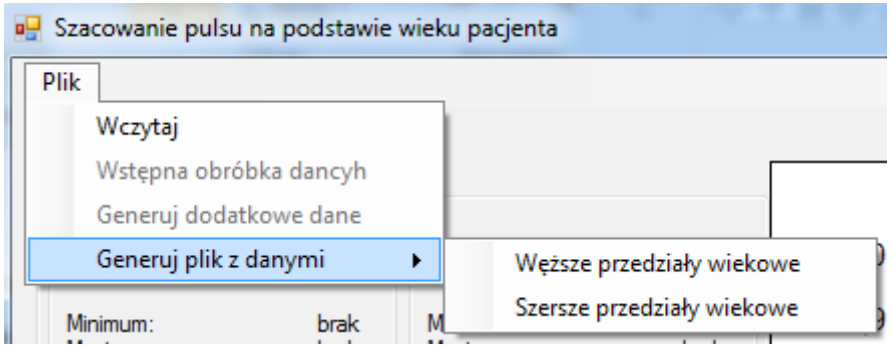
Ilustracja 4: Przykładowy wynik dla danych z węższego przedziału wiekowego

5. Instrukcja obsługi aplikacji

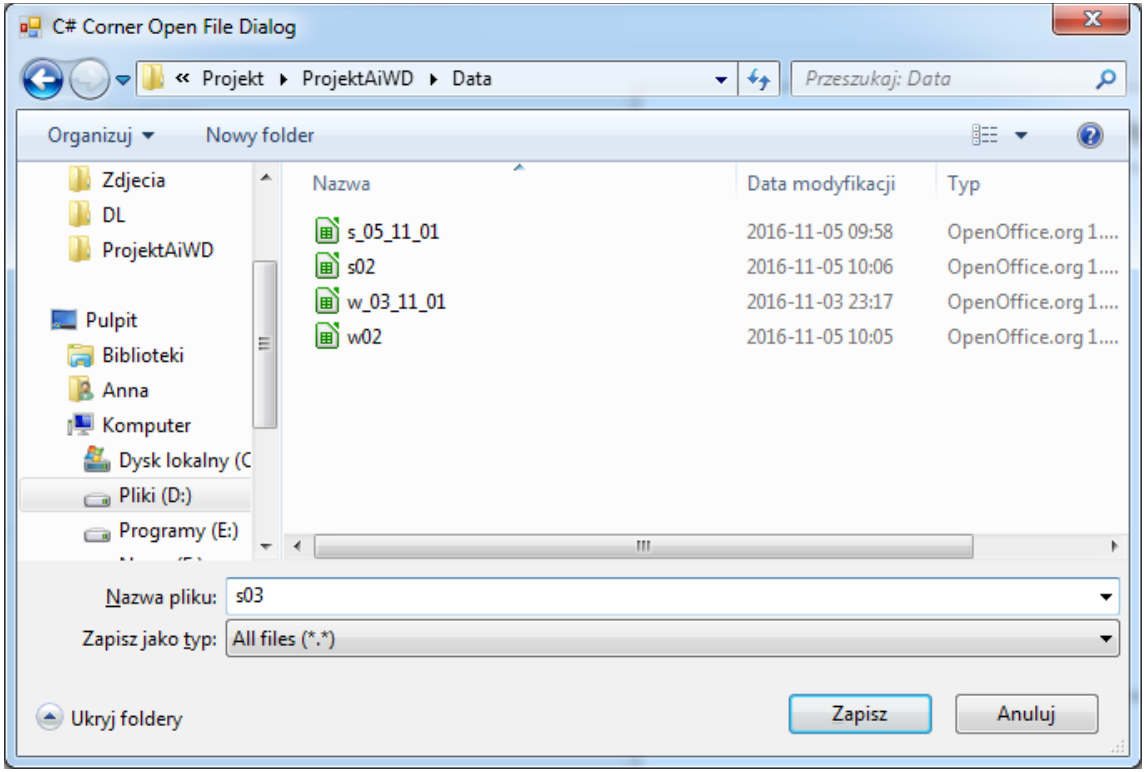
Instrukcja obsługi została przedstawiona na poniższych ilustracjach. Po otworzeniu aplikacji można zrobić dwie rzeczy: albo wygenerować dane poleceniem Plik → Generuj plik z danymi → Węższe przedziały wiekowe lub Plik → Generuj plik z danymi → Szersze przedziały wiekowe, albo wczytać dane poleceniem Plik → Wczytaj.



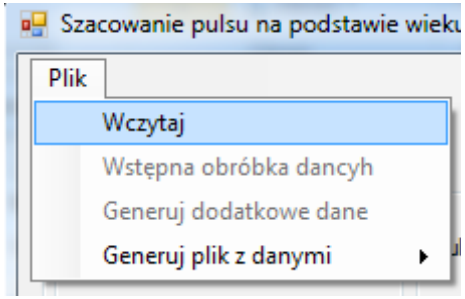
Ilustracja 5: Widok okna aplikacji po jej włączeniu



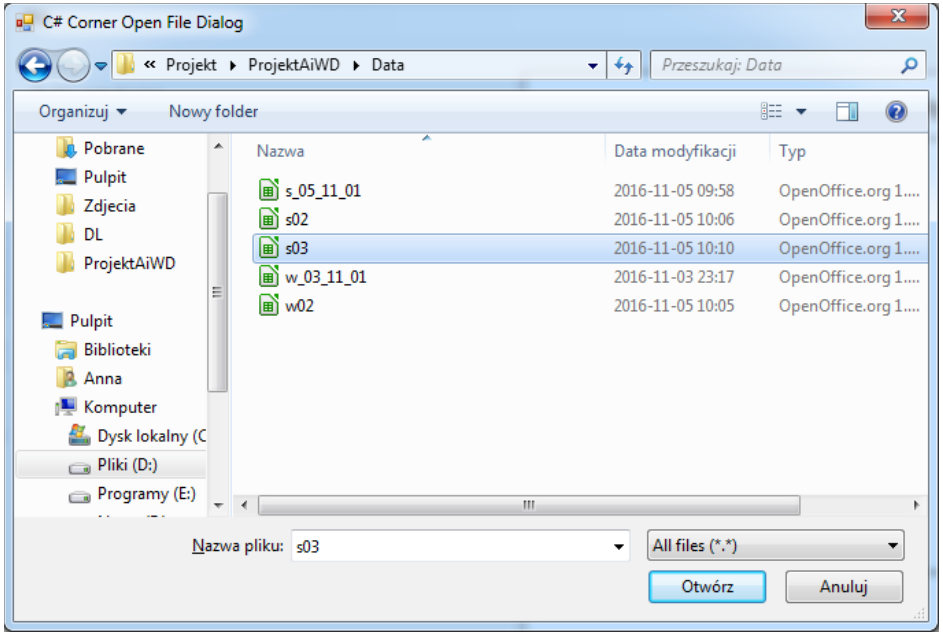
Ilustracja 6: Wybór metody generowania danych



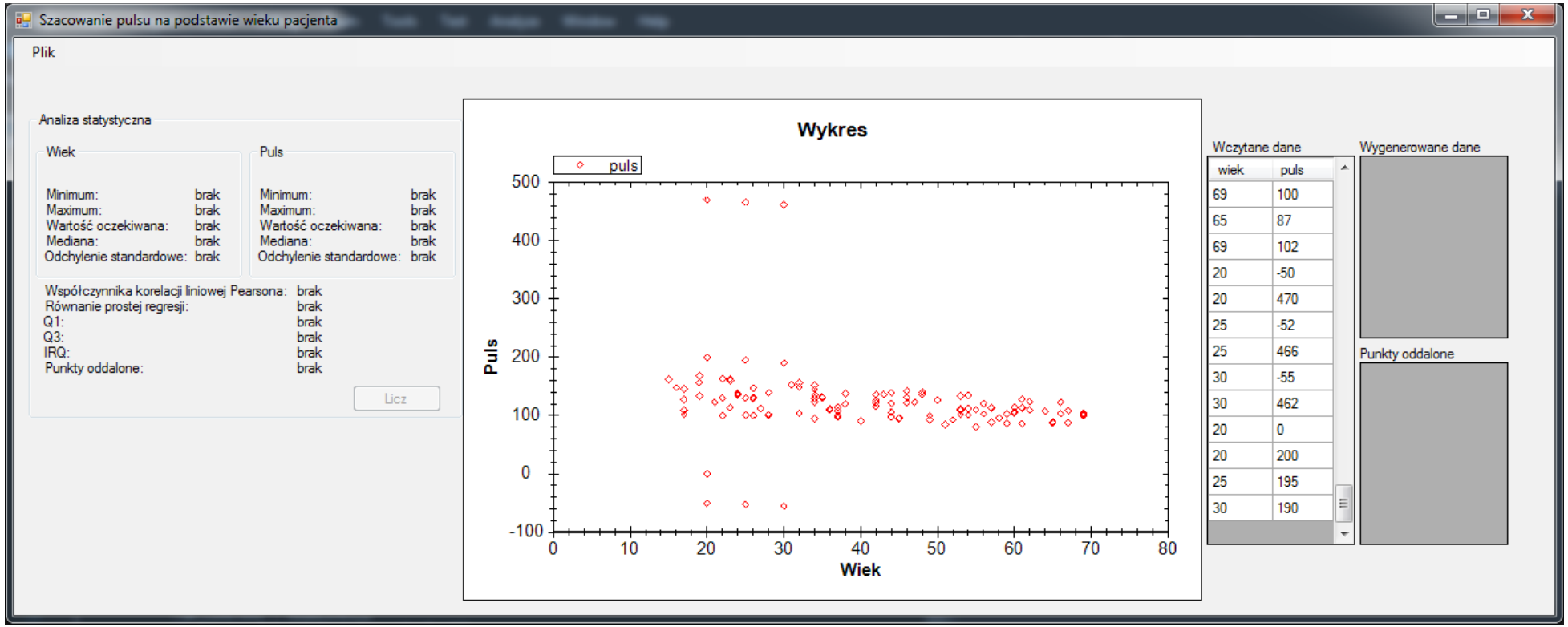
Ilustracja 7: Wybór katalogu do zapisu pliku



Ilustracja 8: Wczytanie pliku

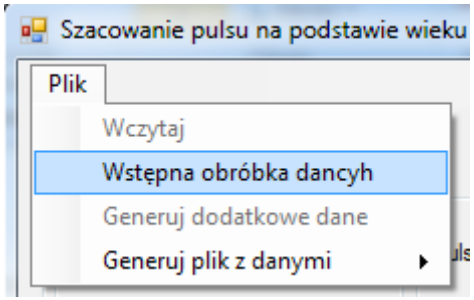


Ilustracja 9: Wybór pliku

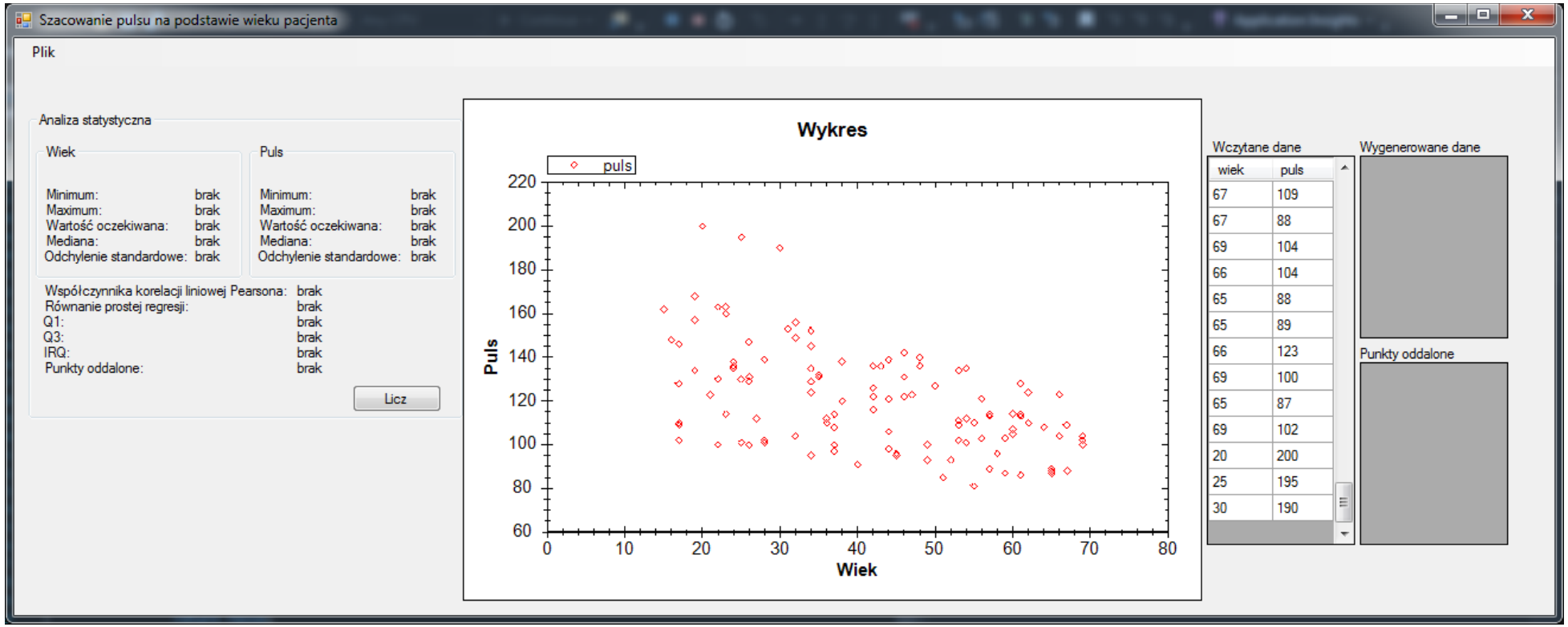


Ilustracja 10: Widok aplikacji po wczytaniu danych

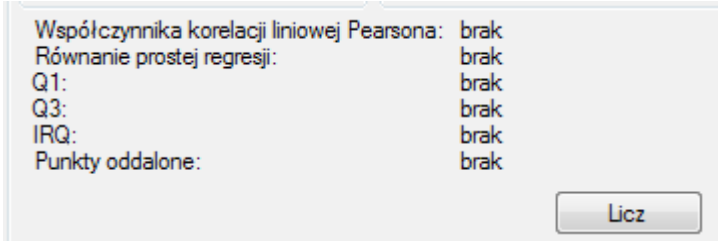
Po wczytaniu danych z pliku można dokonać wstępnej obróbki danych.



Ilustracja 11: Wstępna obróbka danych

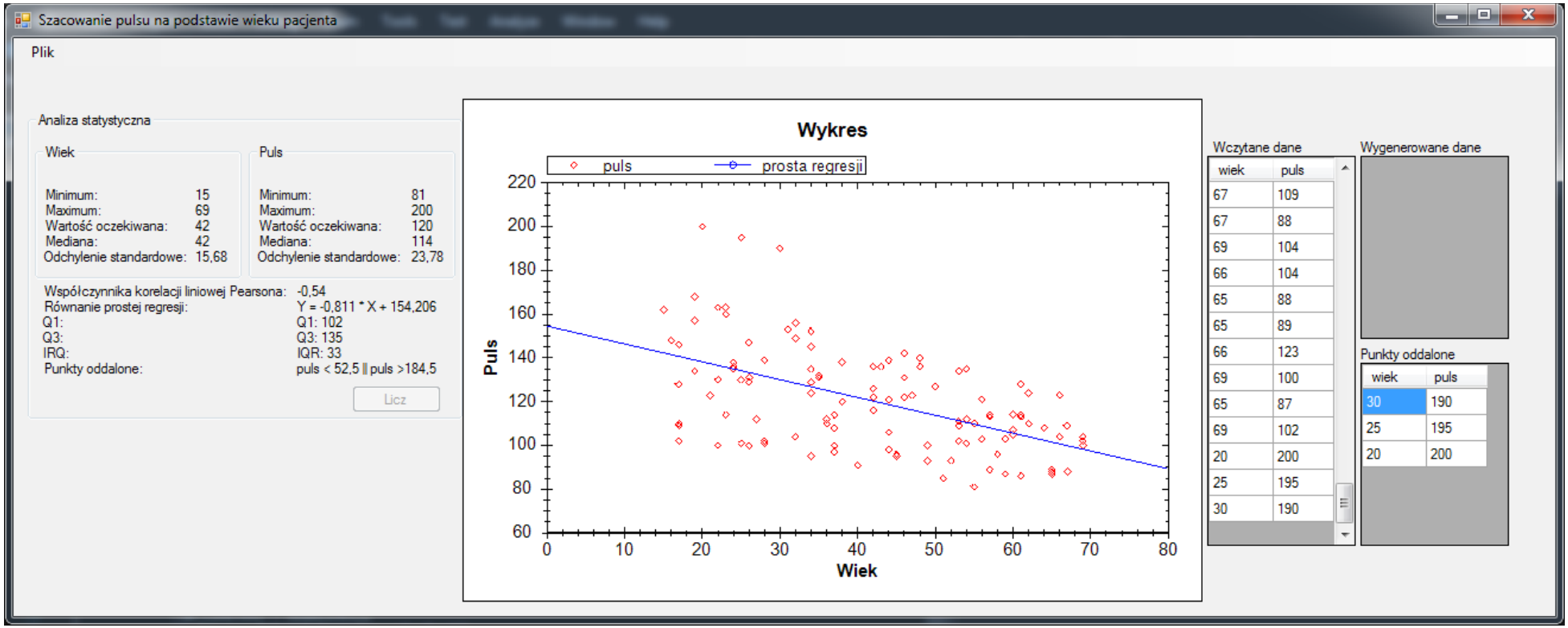


Ilustracja 12: Widok aplikacji po wstępnej obróbce danych



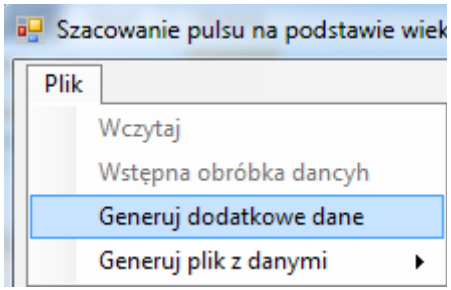
Ilustracja 13: Przycisk "Licz" odpowiedzialny za dokonanie analizy statystycznej został odblokowany

Po wstępnej obróbce danych można dokonać analizy statystycznej klikając na przycisk „Licz”.

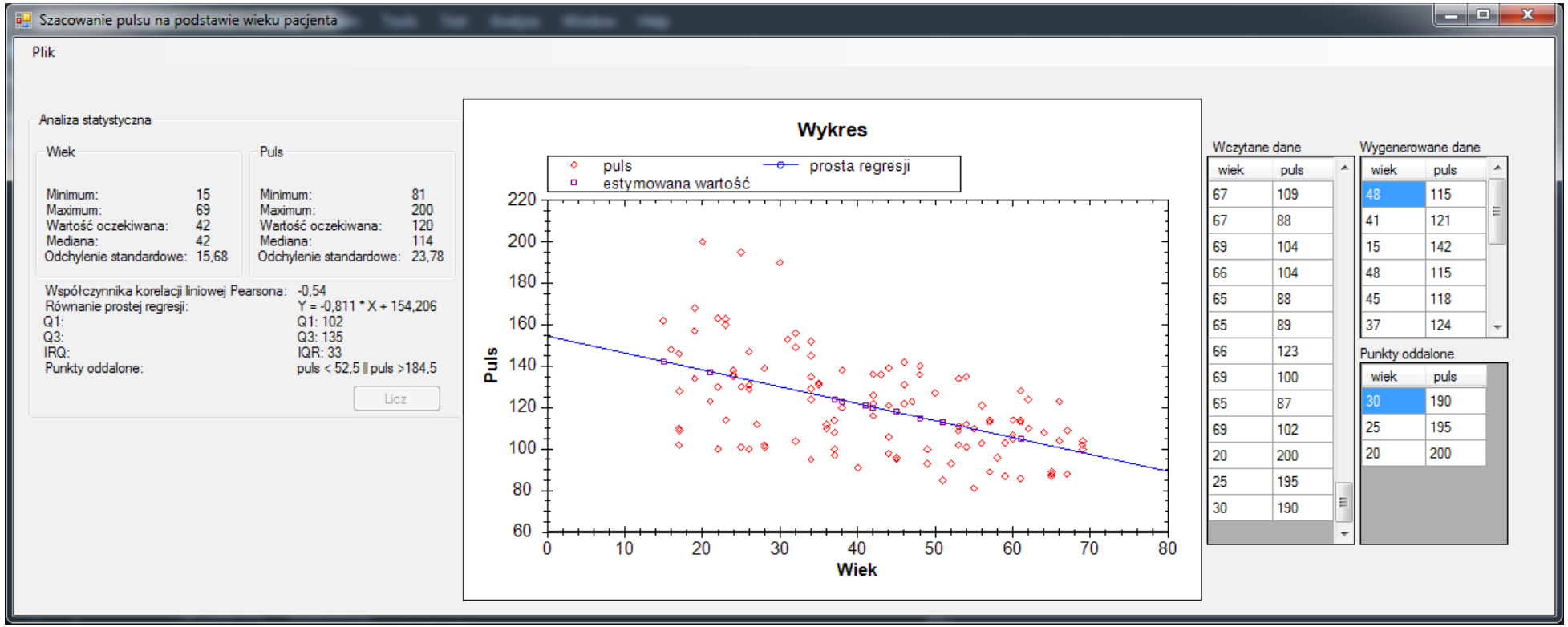


Ilustracja 14: Widok aplikacji po dokonaniu analizy statystycznej

Teraz można generować dodatkowe dane i estymować ich wartości oraz nanieść je na wykres. Odpowiada za to polecenie Plik → Generuj dodatkowe dane.



Ilustracja 15: Estymacja



Ilustracja 16: Widok aplikacji po estymowaniu danych

5. Bibliografia

[1]www.heart.org/HEARTORG/HealthyLiving/PhysicalActivity/FitnessBasics/Target-Heart-Rates_UCM_434341_Article.jsp
[2] https://www.statystyczny.pl/jak-obliczamy-kwantyle/
[3] http://www.statystyka-zadania.pl/wspolczynnik-korelacji-liniowej-pearsona/