

Sprawozdanie z Statystycznej Analizy Danych

Maciej Wojdyło

Bioinformatyka 2 rok 4 semestr grupa LAB3

Nr. Indeksu 157220

1. Wstęp

Celem skryptu jest przeprowadzenie analizy statystycznej danych zawartych w pliku CSV. Skrypt wykonuje następujące kroki:

1. Wczytanie danych.
2. Identyfikacja i uzupełnianie brakujących wartości.
3. Identyfikacja wartości odstających.
4. Opis statystyczny grup.
5. Testowanie hipotez statystycznych.
6. Analiza korelacji między zmiennymi.

2. Opis Działania Skryptu

2.1. Wczytanie danych

Skrypt zaczyna od wczytania danych z pliku CSV przy użyciu funkcji `read.csv2`. Plik jest wczytywany do zmiennej `table`, a kolumny są rozdzielane przy użyciu średnika (;).

```
table <- read.csv2("C:\\Users\\mjwoj\\OneDrive\\Pulpit\\ZaliczenieASD\\Przyklad.csv", sep=";")
```

2.2. Identyfikacja i uzupełnianie brakujących wartości

Skrypt identyfikuje brakujące wartości (NA) w danych i zastępuje je średnią z odpowiednich kolumn.

```
naCells <- which(is.na(table), arr.ind = TRUE)

for(i in 1:nrow(naCells)) {
  if(i==1) {
    cat("1 wartość to wiersz 2 wartość to kolumna","\n")
  }
  cat("brakuje danych w tej komórce", naCells[i,], "\n")
}
```

```

for(i in 1:nrow(table)) {
  for(j in 1:ncol(table)) {
    if(is.na(table[i,j])) {
      srednia <- mean(table[,j], na.rm = TRUE)
      table[i,j] <- srednia
    }
  }
}

```

2.3. Identyfikacja wartości odstających

Wartości odstające są identyfikowane przy użyciu kwartyłowego rozstępu międzykwartyłowego to znaczy jest brany pierwszy i trzeci kwartał ponieważ statystycznie między nimi różnica to mniej więcej 50% (IQR). Funkcja `find_outliers` zwraca wartości odstające i ich indeksy.

```

find_outliers <- function(x) {

  q1 <- quantile(x, 0.25)

  q3 <- quantile(x, 0.75)

  iqr <- q3 - q1

  lower_bound <- q1 - 1.5 * iqr

  upper_bound <- q3 + 1.5 * iqr

  outliers <- x < lower_bound | x > upper_bound

  indexes <- which(outliers)

  return(list(values = x[indexes], indexes = indexes))

}

parametry <- c()

for(i in colnames(table)) {

  if(is.numeric(table[[i]])) {

    parametry <- c(parametry, i)

  }

}

```

```

for (parametr in parametry) {

  outliers <- find_outliers(table[[parametr]])

  print(outliers$values)

  print(outliers$indexes)

}

```

2.4. Opis statystyczny grup

Dla każdej grupy zmiennych obliczane są podstawowe statystyki opisowe takie jak , średnia, mediana, wartość maksymalna, wartość minimalna

```

grupy <- unique(table$grupa)

charakterystyki_grup <- list()

charakterystyka <- function(data) {

  summary_data <- summary(data)

  return(summary_data)

}

for(i in grupy) {

  grupa <- subset(table, grupa == i)

  grupa <- grupa[, !names(grupa) %in% c("grupa", "plec")]

  charakterystyki_grup[[i]] <- charakterystyka(grupa)

}

```

2.5. Testowanie hipotez statystycznych

W zależności od liczby grup, stosowane są różne testy statystyczne:

- Dla dwóch grup: test t-Studenta, test Welcha, test Wilcoxona.
- Dla więcej niż dwóch grup: ANOVA, test Kruskala-Wallisa.

Dodatkowo w oparciu o tabele

Porównanie grup niezależnych			
Ilość porównywanych grup	Zgodność z rozkładem normalnym	Jednorodność wariancji	Wybrany test
2	TAK	TAK	test t-Studenta (dla gr. niezależnych)
		NIE	test Welcha
	NIE	-	test Wilcozona (Manna-Whitneya)
>2	TAK	TAK	test ANOVA (<i>post hoc</i> Tukeya)
		NIE	test Kruskala-Wallis (a) (<i>post hoc</i> Dunna)
	NIE	-	

Jest obliczana Zgodność z rozkładem normalnym przy pomocy `shapiroTest` następnie wykonywany jest test na Jednorodność wariancji przy pomocy `leveneTest` i w zależności od ich wyników dobierany był test

```

if(length(grupy)==2){

for(i in parametry){

  test_result <- shapiro.test(table[[i]])

  if(test_result$p.value <= 0.05){

    result <- 0

  }

}

if(result == 0){

for(nameColumn in parametry){

  wilcox_result <- wilcox.test(table[[nameColumn]] ~ grupa, data = table)

  name <- paste(nameColumn, ".pdf", sep = "")

  pdf(name)

  boxplot(table[[nameColumn]] ~ table$grupa, main = paste("Wykres dla", nameColumn), xlab = "Grupa", ylab = "Wartość")

  dev.off()

}

} else {

for(i in parametry){

  leveneTestResult <- leveneTest(table[[i]] ~ grupa, data = table)

  if(leveneTestResult$"Pr(>F)"[1] > 0.005){

    next
  }
}
}

```

```

    } else {

        resultLev <- 0

    }

}

if(resultLev == 1){

    for(nameColumn in parametry){

        t_test_result <- t.test(table[[nameColumn]] ~ grupa, data = table, var.equal = TRUE)

        name <- paste(nameColumn, ".pdf", sep = "")

        pdf(name)

        boxplot(table[[nameColumn]] ~ table$grupa, main = paste("Wykres dla", nameColumn), xlab = "Grupa", ylab = "Wartość")

        dev.off()

    }

} else {

    for(nameColumn in parametry){

        welch_result <- t.test(table[[nameColumn]] ~ grupa, data = table, var.equal = FALSE)

        name <- paste(nameColumn, ".pdf", sep = "")

        pdf(name)

        boxplot(table[[nameColumn]] ~ table$grupa, main = paste("Wykres dla", nameColumn), xlab = "Grupa", ylab = "Wartość")

        dev.off()

    }

}

} else {

    for(i in parametry){

        test_result <- shapiro.test(table[[i]])

        if(test_result$p.value <= 0.05){

            result <- 0

        }

    }

}

if(result == 0){

```

```

for(nameColumn in parametry){

  kruskal_result <- kruskal.test(table[[nameColumn]] ~ grupa, data = table)

  if (kruskal_result$p.value < 0.05){

    dunn_result <- dunn.test(table[[nameColumn]], g=table$grupa, method="bonferroni")

    print(dunn_result)

  }

  name <- paste(nameColumn, ".pdf", sep = "")

  pdf(name)

  boxplot(table[[nameColumn]] ~ table$grupa, main = paste("Wykres dla", nameColumn), xlab = "Grupa", ylab = "Wartość")

  dev.off()

}

} else {

  for(i in parametry){

    leveneTestResult <- leveneTest(table[[i]] ~ grupa, data = table)

    if(leveneTestResult$"Pr(>F)"[1] > 0.005){

      next

    } else {

      resultLev <- 0

    }

  }

  if(resultLev == 1){

    for(nameColumn in parametry){

      anova_result <- aov(table[[nameColumn]] ~ grupa, data = table)

      if (summary(anova_result)[[1]][["Pr(>F)"]][1] < 0.05){

        tukey_result <- TukeyHSD(anova_result)

        print(tukey_result)

      }

      name <- paste(nameColumn, ".pdf", sep = "")

      pdf(name)

      boxplot(table[[nameColumn]] ~ table$grupa, main = paste("Wykres dla", nameColumn), xlab = "Grupa", ylab = "Wartość")

```

```

    dev.off()

  }

}

}

}

```

2.6. Analiza korelacji

Skrypt wykonuje analizę korelacji pomiędzy zmiennymi dla każdej grupy, a następnie wizualizuje wyniki przy użyciu pakietu `ggcorrplot`.

```

for (grupa in grupy) {

  data_grupa <- subset(table, grupa == grupa)

  data_grupa <- data_grupa[, names(data_grupa) %in% parametry]

  cor_matrix <- cor(data_grupa, use = "pairwise.complete.obs")

  p_matrix <- matrix(nrow = ncol(data_grupa), ncol = ncol(data_grupa))

  for(i in 1:ncol(data_grupa)){

    for(j in 1:ncol(data_grupa)){

      p_matrix[i,j] <- cor.test(data_grupa[,i], data_grupa[,j])$p.value

    }

  }

  significant_correlations <- p_matrix < 0.05

  pdf(file = paste0("Wykres_korelacji_dla_grupy_", grupa, ".pdf"))

  print(ggcorrplot(cor_matrix, hc.order = TRUE, type = "lower",

    lab = TRUE, lab_size = 3,

    title = paste("Wykres korelacji dla grupy", grupa)))

  dev.off()

}

```

3. Opis wszystkich funkcji

3.1. Funkcja `read.csv2` wczytuje dane z pliku CSV, gdzie kolumny są rozdzielone średnikiem (;). Plik jest wczytywany do zmiennej `table`.

```
table<-read.csv2("C:\\Users\\mjwoj\\OneDrive\\Pulpit\\ZaliczenieASD\\Przyklad.csv",sep= ";")
```

3.2. Identyfikacja brakujących wartości

Funkcja `which` z argumentem `is.na(table)` i parametrem `arr.ind = TRUE` znajduje indeksy brakujących wartości (NA) w danych i zwraca je w formie macierzy współrzędnych.

```
naCells<- which (is.na(table),arr.ind = TRUE)
```

3.3. Uzupełnianie brakujących wartości

```
#1
for(i in 1:nrow(naCells)){
  if(i==1){
    cat("1 wartość to wiersz 2 wartość to kolumna","\n")
  }
  cat("brakuje danych w tej komórce",naCells[i,],"\n")
}
#Usuwanie wartości NA i zastępowanie ich średnią z kolumny
for(i in 1:nrow(table)){
  for(j in 1:ncol(table)){
    if(is.na(table[i,j])){
      cat("zmiana wartości w komórce",table[i,1],j,"z wartości ",table[i,j],"na ")
      srednia <- mean(table[,j], na.rm = TRUE)
      cat(srednia,"\n")
      table[i,j] <- srednia
    }
  }
}
```

Pierwsza pętla `for` wyświetla informacje o brakujących wartościach w danych. Druga pętla `for` iteruje przez wszystkie komórki tabeli, sprawdza, czy są brakujące wartości (NA), i zastępuje je średnią z odpowiednich kolumn.

3.4. Funkcja `find_outliers`


```

find_outliers <- function(x) {
  q1 <- quantile(x, 0.25)
  q3 <- quantile(x, 0.75)
  iqr <- q3 - q1
  lower_bound <- q1 - 1.5 * iqr
  upper_bound <- q3 + 1.5 * iqr
  outliers <- x < lower_bound | x > upper_bound
  indexes <- which(outliers)
  return(list(values = x[indexes], indexes = indexes))
}
parametry<-c()
for(i in colnames(table)){
  if(is.numeric(table[[i]])){
    parametry<- c(parametry,i)
  }else{
    next
  }
}

```

Funkcja `find_outliers` identyfikuje wartości odstające w wektorze `x`. Używa kwartyłowego rozstępu międzykwartyłowego (IQR) do obliczenia dolnego i górnego limitu. Wartości poniżej dolnego limitu lub powyżej górnego limitu są uznawane za odstające.

3.5. Identyfikacja wartości odstających

```

parametry<-c()
for(i in colnames(table)){
  if(is.numeric(table[[i]])){
    parametry<- c(parametry,i)
  }else{
    next
  }
}

#Zidentyfikowanie wartości odstających dla wszystkich parametrów
for (parametr in parametry) {
  cat("Wartości odstające dla parametru", parametr, ":\n")
  outliers <- find_outliers(table[[parametr]])
  print(outliers$values)
  cat("Indeksy wierszy, w których występują wartości odstające:\n")
  print(outliers$indexes)
  cat("\n")
}

```

Skrypt identyfikuje kolumny zawierające wartości numeryczne i zapisuje ich nazwy w wektorze `parametry`. Następnie iteruje przez te kolumny i używa funkcji `find_outliers` do identyfikacji wartości odstających, które są następnie wyświetlane.

3.6. funkcja charakterystyka

```
charakterystyka <- function(data) {  
  summary_data <- summary(data)  
  return(summary_data)  
}
```

Funkcja `charakterystyka` oblicza podstawowe statystyki opisowe (średnia, median, min, max, itd.) dla danych przekazanych jako argument `data`.

3.7. Opis statystyczny grup

```
grupy <- unique(table$grupa)  
charakterystyki_grup <- list()  
charakterystyka <- function(data) {  
  summary_data <- summary(data)  
  return(summary_data)  
}  
  
for(i in grupy){  
  grupa <- subset(table, grupa == i)  
  grupa <- grupa[, !names(grupa) %in% c("grupa", "plec")]  
  charakterystyki_grup[[i]] <- charakterystyka(grupa)  
}  
i<-1  
for(wynik_charakterystyki in charakterystyki_grup){  
  cat("Charakterystyka grupy ",grupy[i],"\n")  
  i<-i+1  
  print(wynik_charakterystyki)  
}
```

Skrypt tworzy listę unikalnych grup w kolumnie `grupa`. Dla każdej grupy wyodrębnia odpowiednie wiersze z tabeli, usuwa kolumny `grupa` i `plec`, ponieważ nie są one numeryczne co uniemożliwia przeprowadzenie charakterystyki a następnie oblicza statystyki opisowe za pomocą funkcji `charakterystyka`.

3.8. Testowanie hipotez statystycznych

a) Dla dwóch grup

```

if(length(grupy)==2){
  for(i in parametry){
    test_result <- shapiro.test(table[[i]])
    if(test_result$p.value <= 0.05){
      result <- 0
    }
  }
  if(result == 0){
    for(nameColumn in parametry){
      wilcox_result <- wilcox.test(table[[nameColumn]] ~ grupa, data = table)
      name <- paste(nameColumn, ".pdf", sep = "")
      pdf(name)
      boxplot(table[[nameColumn]] ~ table$grupa, main = paste("wykres dla", nameColumn), xlab = "Grupa", ylab = "wartość")
      dev.off()
    }
  } else {
    for(i in parametry){
      leveneTestResult <- leveneTest(table[[i]] ~ grupa, data = table)
      if(leveneTestResult$Pr(>F)"[1] > 0.005){
        next
      } else {
        resultLev <- 0
      }
    }
    if(resultLev == 1){
      for(nameColumn in parametry){
        t_test_result <- t.test(table[[nameColumn]] ~ grupa, data = table, var.equal = TRUE)
        name <- paste(nameColumn, ".pdf", sep = "")
        pdf(name)
        boxplot(table[[nameColumn]] ~ table$grupa, main = paste("wykres dla", nameColumn), xlab = "Grupa", ylab = "wartość")
        dev.off()
      }
    } else {
      for(nameColumn in parametry){
        welch_result <- t.test(table[[nameColumn]] ~ grupa, data = table, var.equal = FALSE)
        name <- paste(nameColumn, ".pdf", sep = "")
        pdf(name)
        boxplot(table[[nameColumn]] ~ table$grupa, main = paste("wykres dla", nameColumn), xlab = "Grupa", ylab = "wartość")
        dev.off()
      }
    }
  }
}

```

Dla dwóch grup skrypt najpierw sprawdza normalność rozkładu za pomocą testu Shapiro-Wilka. Jeśli dane nie mają rozkładu normalnego, stosuje test Wilcoxa. W przeciwnym razie sprawdza jednorodność wariancji za pomocą testu Levene'a i w zależności od wyniku stosuje test t-Studenta lub test Welcha. Wyniki zostają zapisane w plikach pdf o nazwach takich jak nazywają się numeryczne kolumny.

b) Dla więcej niż dwóch grup

```

else {
  for(i in parametry){
    test_result <- shapiro.test(table[[i]])
    if(test_result$p.value <= 0.05){
      result <- 0
    }
  }
  if(result == 0){
    for(nameColumn in parametry){
      kruskal_result <- kruskal.test(table[[nameColumn]] ~ grupa, data = table)
      if(kruskal_result$p.value < 0.05){
        dunn_result <- dunn.test(table[[nameColumn]], g=table$grupa, method="bonferroni")
        print(dunn_result)
      }
      name <- paste(nameColumn, ".pdf", sep = "")
      pdf(name)
      boxplot(table[[nameColumn]] ~ table$grupa, main = paste("wykres dla", nameColumn), xlab = "Grupa", ylab = "wartość")
      dev.off()
    }
  } else {
    for(i in parametry){
      leveneTestResult <- leveneTest(table[[i]] ~ grupa, data = table)
      if(leveneTestResult$Pr(>F)"[1] > 0.005){
        next
      } else {
        resultLev <- 0
      }
    }
    if(resultLev == 1){
      for(nameColumn in parametry){
        anova_result <- aov(table[[nameColumn]] ~ grupa, data = table)
        if(summary(anova_result)[1][1][1]"Pr(>F)"[1] < 0.05){
          tukey_result <- TukeyHSD(anova_result)
          print(tukey_result)
        }
        name <- paste(nameColumn, ".pdf", sep = "")
        pdf(name)
        boxplot(table[[nameColumn]] ~ table$grupa, main = paste("wykres dla", nameColumn), xlab = "Grupa", ylab = "wartość")
        dev.off()
      }
    }
  }
}

```

Dla więcej niż dwóch grup skrypt najpierw sprawdza normalność rozkładu za pomocą testu Shapiro-Wilka. Jeśli dane nie mają rozkładu normalnego, stosuje test Kruskala-Wallisa. W przeciwnym razie sprawdza jednorodność wariancji za pomocą testu Levene'a i w zależności od wyniku stosuje ANOVA lub test Tukeya. I w tym przypadku również wyniki są zapisywane do plików pdf o nazwie takiej jak nazywają się numeryczne kolumny.

3.9. Analiza korelacji

```

for (grupa in grupy) {
  data_grupa <- subset(table, grupa == grupa)
  data_grupa <- data_grupa[, names(data_grupa) %in% parametry]
  cor_matrix <- cor(data_grupa, use = "pairwise.complete.obs")
  p_matrix <- matrix(nrow = ncol(data_grupa), ncol = ncol(data_grupa))
  for(i in 1:ncol(data_grupa)){
    for(j in 1:ncol(data_grupa)){
      p_matrix[i,j] <- cor.test(data_grupa[,i], data_grupa[,j])$p.value
    }
  }

  significant_correlations <- p_matrix < 0.05
  pdf(file = paste0("wykres_korelacji_dla_grupy_", grupa, ".pdf"))
  print(ggcorrplot(cor_matrix, hc.order = TRUE, type = "lower",
                  lab = TRUE, lab_size = 3,
                  title = paste("wykres korelacji dla grupy", grupa)))
  dev.off()
}

```

Skrypt wykonuje analizę korelacji dla każdej grupy. Oblicza macierz korelacji i macierz wartości p (test korelacji) dla zmiennych numerycznych w danych. Wyniki są następnie wizualizowane przy użyciu funkcji `ggcorrplot` i zapisywane do plików pdf pod nazwą Wykres korelacji dla grupy x

Gdzie x to nazwa grup analizowanych w pliku CSV.

4. Przykładowe Wyniki dla Pliku „przykładoweDane-Projekt.csv”

4.1 Raportowanie o braku danych

```

1 wartość to wiersz 2 wartość to kolumna
brakuje danych w tej komórce 13 7
brakuje danych w tej komórce 68 7
brakuje danych w tej komórce 5 10

```

4.2 Zaraportowanie wszystkich wprowadzonych zmian

```

zmiana wartości w komórce CHOR1 10 z wartości NA na 0.857027
zmiana wartości w komórce CHOR1 7 z wartości NA na 12.16923
zmiana wartości w komórce KONTROLA 7 z wartości NA na 12.16923

```

4.3 Zaraportowanie informacji o wartościach odstających dla wybranych parametrów.

wartości odstające dla parametru wiek :

[1] 48

Indeksy wierszy, w których występują wartości odstające:

[1] 55

wartości odstające dla parametru hscRP :

[1] 20.1548 16.4069 42.6499 19.2124

Indeksy wierszy, w których występują wartości odstające:

[1] 9 13 15 38

wartości odstające dla parametru ERY :

[1] 33

Indeksy wierszy, w których występują wartości odstające:

[1] 23

wartości odstające dla parametru PLT :

[1] 456 434

Indeksy wierszy, w których występują wartości odstające:

[1] 36 51

4.4 Wykonanie charakterystyk dla badanych grup, zapisanie wyników w czytelnej formie (polecana struktura tabelaryczna).

Charakterystyka grupy CHOR1

wiek	hscRP	ERY	PLT	HGB	HCT
Min. :17.00	Min. : 0.4876	Min. : 3.530	Min. :128.0	Min. : 9.505	Min. :0.2800
1st Qu.:26.00	1st Qu.: 2.3227	1st Qu.: 4.070	1st Qu.:179.0	1st Qu.:11.921	1st Qu.:0.3500
Median :29.00	Median : 3.9665	Median : 4.200	Median :217.0	Median :12.405	Median :0.3630
Mean :29.56	Mean : 6.1030	Mean : 5.363	Mean :225.3	Mean :12.402	Mean :0.3636
3rd Qu.:32.00	3rd Qu.: 4.9935	3rd Qu.: 4.510	3rd Qu.:266.0	3rd Qu.:13.210	3rd Qu.:0.3860
Max. :43.00	Max. :42.6499	Max. :33.000	Max. :336.0	Max. :14.499	Max. :0.4050
MCHC	MON	LEU			
Min. :32.56	Min. :0.4800	Min. : 6.79			
1st Qu.:34.71	1st Qu.:0.6100	1st Qu.:10.11			
Median :35.05	Median :0.7600	Median :11.66			
Mean :35.13	Mean :0.8579	Mean :12.02			
3rd Qu.:35.60	3rd Qu.:1.0700	3rd Qu.:14.48			
Max. :36.87	Max. :1.5200	Max. :16.81			

Charakterystyka grupy CHOR2

wiek	hscRP	ERY	PLT	HGB	HCT
Min. :21.00	Min. : 0.3351	Min. :3.250	Min. : 91.0	Min. : 9.827	Min. :0.0423
1st Qu.:25.00	1st Qu.: 2.0781	1st Qu.:3.850	1st Qu.:172.0	1st Qu.:11.760	1st Qu.:0.3300
Median :30.00	Median : 3.4455	Median :4.270	Median :195.0	Median :12.566	Median :0.3600
Mean :30.04	Mean : 5.5360	Mean :4.198	Mean :209.1	Mean :12.806	Mean :0.3460
3rd Qu.:33.00	3rd Qu.: 8.6093	3rd Qu.:4.430	3rd Qu.:223.0	3rd Qu.:13.694	3rd Qu.:0.3900
Max. :42.00	Max. :19.2124	Max. :5.040	Max. :456.0	Max. :22.232	Max. :0.4120
MCHC	MON	LEU			
Min. :32.89	Min. :0.1400	Min. : 7.95			
1st Qu.:34.88	1st Qu.:0.5500	1st Qu.:10.70			
Median :35.55	Median :0.6600	Median :12.00			
Mean :35.55	Mean :0.9528	Mean :12.04			
3rd Qu.:36.04	3rd Qu.:0.8800	3rd Qu.:13.34			
Max. :38.87	Max. :7.0000	Max. :16.59			

Charakterystyka grupy KONTROLA

wiek	hsCRP	ERY	PLT	HGB	HCT
Min. :23.00	Min. : 0.7584	Min. :3.090	Min. :147.0	Min. : 9.505	Min. :0.2790
1st Qu.:29.00	1st Qu.: 2.3022	1st Qu.:3.820	1st Qu.:188.0	1st Qu.:10.472	1st Qu.:0.3200
Median :32.00	Median : 4.2204	Median :3.980	Median :214.0	Median :11.438	Median :0.3390
Mean :32.32	Mean : 5.2951	Mean :4.013	Mean :225.9	Mean :11.300	Mean :0.3376
3rd Qu.:35.00	3rd Qu.: 6.8521	3rd Qu.:4.330	3rd Qu.:254.0	3rd Qu.:12.082	3rd Qu.:0.3530
Max. :48.00	Max. :14.3951	Max. :5.050	Max. :434.0	Max. :13.210	Max. :0.3890
MCHC	MON	LEU			
Min. :32.06	Min. :0.3500	Min. : 4.83			
1st Qu.:33.72	1st Qu.:0.6500	1st Qu.: 9.22			
Median :34.55	Median :0.7600	Median :10.68			
Mean :34.40	Mean :0.7604	Mean :11.36			
3rd Qu.:35.21	3rd Qu.:0.8600	3rd Qu.:13.59			
Max. :36.04	Max. :1.2500	Max. :17.46			

4.5 Wykonanie analizy porównawczej pomiędzy grupami, określenie czy istnieją istotne statystycznie różnice. Jeśli istnieją istotne statystyczne różnice pomiędzy grupami to zaraportowanie pomiędzy którymi grupami występują i jak istotne są to różnice.

```
[1] "Więcej niż 2 grupy"
[1] "Rozkład Normalny NIE"
[1] "Test Kruskala"
[1] "Brak istotnych różnic między grupami dla wiek ."
```

```
[1] "Istnieją istotne różnice między grupami dla HGB ."
```

```
data: x and group
Kruskal-Wallis chi-squared = 13.5764, df = 2, p-value = 0
```

Comparison of x by group (Bonferroni)			
Col	Mean		
Row	Mean	CHOR1	CHOR2
	CHOR2	-0.207846	
		1.0000	
KONTROLA		3.081967	3.289813
		0.0031*	0.0015*

```
alpha = 0.05
Reject Ho if p <= alpha/2
$chi2
[1] 13.57639
```

```
$Z
[1] -0.207846 3.081967 3.289813
```

```
$P
[1] 0.4176745906 0.0010281878 0.0005012697
```

```
$P.adjusted
[1] 1.0000000000 0.003084564 0.001503809
```

```
$comparisons
[1] "CHOR1 - CHOR2" "CHOR1 - KONTROLA" "CHOR2 - KONTROLA"
```

4.6 Wykonanie analizy korelacji. Zaraportowanie pomiędzy którymi parametrami w obrębie jakich grup występują istotne statystycznie korelacje oraz określenie siły i kierunku korelacji.

Analiza korelacji dla grupy CHOR1 :

	wiek	hsCRP	ERY	PLT	HGB	HCT	MCHC	MON
wiek	1.00000000	-4.171440e-02	-6.067441e-02	0.05968452	0.13132068	-0.061797112	-0.24107554	0.10521072
hsCRP	-0.04171440	1.000000e+00	-1.168232e-05	-0.04108440	-0.01903429	0.035152126	-0.07639715	0.06193678
ERY	-0.06067441	-1.168232e-05	1.000000e+00	0.24102534	-0.10120662	-0.108694438	-0.02018154	-0.05471118
PLT	0.05968452	-4.108440e-02	2.410253e-01	1.00000000	-0.14095788	0.107475736	-0.36566298	-0.04804941
HGB	0.13132068	-1.903429e-02	-1.012066e-01	-0.14095788	1.00000000	0.424614272	0.35644686	-0.05382059
HCT	-0.06179711	3.515213e-02	-1.086944e-01	0.10747574	0.42461427	1.000000000	0.12038663	-0.04534904
MCHC	-0.24107554	-7.639715e-02	-2.018154e-02	-0.36566298	0.35644686	0.120386626	1.00000000	0.21567218
MON	0.10521072	6.193678e-02	-5.471118e-02	-0.04804941	-0.05382059	-0.045349036	0.21567218	1.00000000
LEU	-0.29857970	2.962502e-01	4.596287e-02	0.17142420	0.03528594	-0.005012048	0.09809507	-0.03054685
LEU								
wiek	-0.298579704							
hsCRP	0.296250169							
ERY	0.045962866							
PLT	0.171424197							
HGB	0.035285945							
HCT	-0.005012048							
MCHC	0.098095071							
MON	-0.030546850							
LEU	1.000000000							

Istotne statystycznie korelacje dla grupy CHOR1 :

Parametry wiek i wiek są istotnie skorelowane.

Siła korelacji: 1

Kierunek korelacji: dodatni

Parametry wiek i MCHC są istotnie skorelowane.

Siła korelacji: -0.2410755

Kierunek korelacji: ujemny

Parametry wiek i LEU są istotnie skorelowane.

Siła korelacji: -0.2985797

Kierunek korelacji: ujemny

Parametry hsCRP i hsCRP są istotnie skorelowane.

Siła korelacji: 1

Kierunek korelacji: dodatni

Parametry hsCRP i LEU są istotnie skorelowane.

Siła korelacji: 0.2962502

Kierunek korelacji: dodatni

Analiza korelacji dla grupy CHOR2 :

	wiek	hsCRP	ERY	PLT	HGB	HCT	MCHC	MON
wiek	1.00000000	-4.171440e-02	-6.067441e-02	0.05968452	0.13132068	-0.061797112	-0.24107554	0.10521072
hsCRP	-0.04171440	1.000000e+00	-1.168232e-05	-0.04108440	-0.01903429	0.035152126	-0.07639715	0.06193678
ERY	-0.06067441	-1.168232e-05	1.000000e+00	0.24102534	-0.10120662	-0.108694438	-0.02018154	-0.05471118
PLT	0.05968452	-4.108440e-02	2.410253e-01	1.00000000	-0.14095788	0.107475736	-0.36566298	-0.04804941
HGB	0.13132068	-1.903429e-02	-1.012066e-01	-0.14095788	1.00000000	0.424614272	0.35644686	-0.05382059
HCT	-0.06179711	3.515213e-02	-1.086944e-01	0.10747574	0.42461427	1.000000000	0.12038663	-0.04534904
MCHC	-0.24107554	-7.639715e-02	-2.018154e-02	-0.36566298	0.35644686	0.120386626	1.00000000	0.21567218
MON	0.10521072	6.193678e-02	-5.471118e-02	-0.04804941	-0.05382059	-0.045349036	0.21567218	1.00000000
LEU	-0.29857970	2.962502e-01	4.596287e-02	0.17142420	0.03528594	-0.005012048	0.09809507	-0.03054685
LEU								
wiek	-0.298579704							
hsCRP	0.296250169							
ERY	0.045962866							
PLT	0.171424197							
HGB	0.035285945							
HCT	-0.005012048							
MCHC	0.098095071							
MON	-0.030546850							
LEU	1.000000000							

Istotne statystycznie korelacje dla grupy CHOR2 :

Parametry wiek i wiek są istotnie skorelowane.

Siła korelacji: 1

Kierunek korelacji: dodatni

Parametry wiek i MCHC są istotnie skorelowane.

Siła korelacji: -0.2410755

Kierunek korelacji: ujemny

Parametry wiek i LEU są istotnie skorelowane.

Siła korelacji: -0.2985797

Kierunek korelacji: ujemny

Parametry hsCRP i hsCRP są istotnie skorelowane.

Siła korelacji: 1

Kierunek korelacji: dodatni

Parametry hsCRP i LEU są istotnie skorelowane.

Siła korelacji: 0.2962502

Kierunek korelacji: dodatni

Analiza korelacji dla grupy KONTROLA :

	wiek	hsCRP	ERY	PLT	HGB	HCT	MCHC	MON
wiek	1.00000000	-4.171440e-02	-6.067441e-02	0.05968452	0.13132068	-0.061797112	-0.24107554	0.10521072
hsCRP	-0.04171440	1.000000e+00	-1.168232e-05	-0.04108440	-0.01903429	0.035152126	-0.07639715	0.06193678
ERY	-0.06067441	-1.168232e-05	1.000000e+00	0.24102534	-0.10120662	-0.108694438	-0.02018154	-0.05471118
PLT	0.05968452	-4.108440e-02	2.410253e-01	1.00000000	-0.14095788	0.107475736	-0.36566298	-0.04804941
HGB	0.13132068	-1.903429e-02	-1.012066e-01	-0.14095788	1.00000000	0.424614272	0.35644686	-0.05382059
HCT	-0.06179711	3.515213e-02	-1.086944e-01	0.10747574	0.42461427	1.000000000	0.12038663	-0.04534904
MCHC	-0.24107554	-7.639715e-02	-2.018154e-02	-0.36566298	0.35644686	0.120386626	1.00000000	0.21567218
MON	0.10521072	6.193678e-02	-5.471118e-02	-0.04804941	-0.05382059	-0.045349036	0.21567218	1.00000000
LEU	-0.29857970	2.962502e-01	4.596287e-02	0.17142420	0.03528594	-0.005012048	0.09809507	-0.03054685

	LEU
wiek	-0.298579704
hsCRP	0.296250169
ERY	0.045962866
PLT	0.171424197
HGB	0.035285945
HCT	-0.005012048
MCHC	0.098095071
MON	-0.030546850
LEU	1.000000000

Istotne statystycznie korelacje dla grupy KONTROLA :

Parametry wiek i wiek są istotnie skorelowane.

Siła korelacji: 1

Kierunek korelacji: dodatni

Parametry wiek i MCHC są istotnie skorelowane.

Siła korelacji: -0.2410755

Kierunek korelacji: ujemny

Parametry wiek i LEU są istotnie skorelowane.

Siła korelacji: -0.2985797

Kierunek korelacji: ujemny

Parametry hsCRP i hsCRP są istotnie skorelowane.

Siła korelacji: 1

Kierunek korelacji: dodatni

Parametry hsCRP i LEU są istotnie skorelowane.

Siła korelacji: 0.2962502

Kierunek korelacji: dodatni

5. Podsumowanie

Skrypt przeprowadza kompleksową analizę danych, obejmującą wczytanie i wstępne przetwarzanie danych, identyfikację i uzupełnianie brakujących wartości, identyfikację wartości odstających, opis statystyczny grup, testowanie hipotez statystycznych oraz analizę korelacji. Każda funkcja w skrypcie została zaprojektowana w celu realizacji określonego etapu analizy, co pozwala na dokładne i systematyczne badanie danych.

