Podejścia Systemowe w Badaniach Biomedycznych
zajęcia laboratoryjne 1

Statystyka w badaniach medycznych

1 Wprowadzenie - dane

R umożliwia używanie wielu wyspecjalizowanych, mniej lub bardziej popularnych funkcji matematycznych i statystycznych. Przedstawione zostaną wybrane z nich. Protokół obejmuje wprowadzenie w zagadnienia analizy statystycznej i przedstawienie wybranych metod (istotne są zagadnienia poruszane na wykładzie). Dane wykorzystywane do poniższych przykładów:

- http://www.cs.put.poznan.pl/kgutowska/PSwBB/dane/przykladoweDaneZBrakami.csv
- http://www.cs.put.poznan.pl/kgutowska/PSwBB/dane/przykladoweDaneBezBrakow.csv
- http://www.cs.put.poznan.pl/kgutowska/PSwBB/dane/przykladowe2Grupy.csv

Wymagane pakiety: *Hmisc*, *dplyr*, *ggpubr*, *car*, *dunn.test*, *FSA*.

2 Ocena danych wejściowych

Często otrzymane dane wejściowe, bez wcześniejszego przetwarzania nie nadają się do bezpośredniego użycia i przeprowadzenia analiz statystycznych. W danych mogą pojawiać się braki, wynikające z błędu popełnionego przez człowieka (zwłaszcza, jeśli dane gromadzone są przez większą grupę osób). Brakujące dane oznaczane są w R jako NA (*ang. Not Available*), funkcja pozwalająca na wyszukanie takich danych *is.na()*.

Przykład – *is.na()*

```
1 dane <- read.csv2("przykladoweDaneZBrakami.csv", sep = ";")
2
3 wektorZBrakamiDanych <- dane$HGB
4 is.na(wektorZBrakamiDanych) #zwraca wektor wartosci TRUE i FALSE
5 sum(is.na(wektorZBrakamiDanych)) #okresla ile jest wartosci NA w wektorze
6 which(is.na(wektorZBrakamiDanych)) #zwraca numery indeksow dla wartosci NA
```

R umożliwia zastosowanie różnych wariantów radzenia sobie z brakiem danych. Jednym z nich jest usunięcie brakujących danych. W przypadku usuwania takich danych z wektora należy funkcji *na.omit()*. Chcąc usunąć brakujące dane z ramki danych, należy użyć funkcji *complete.cases()*. Jednak poza usuwaniem danych zdecydowanie lepszym sposobem radzenia sobie z brakującymi danymi jest wstawienie sztucznych pomiarów.

Przykład – *na.omit()* - dla wektorów

```
1 wektorBezBrakowDanych <- na.omit(wektorZBrakamiDanych)
2 length(wektorZBrakamiDanych) #dlugosc wektora zawierajacego braki danych
3 length(wektorBezBrakowDanych) #dlugosc wektora po pominieciu brakow danych
```

Zauważ, że po wyświetleniu danych dla wektora "wektorBezBrakówDanych" uzyskujemy informacje, że pominięto konkretne dane (wskazane są także numery indeksów w wektorze).

Przykład – *complete.cases()* - dla ramek danych

```
complete.cases(dane) #zwraca FLASE dla wierszy zawierajacych NA
noweDane <- dane[complete.cases(dane), ]
```

Powyżej, poza znalezieniem wierszy zawierających niekompletne dane, przedstawiono sposób zapisania nowych danych z pominięciem wierszy z brakami.

Usuwanie brakujących danych to najprostsze rozwiązanie, jednak nie jest najlepsze, ponieważ redukcja danych prowadzi do pogorszenia właściwości procedur statystycznych. Zdecydowanie lepszym rozwiązaniem od usuwania jest wstawienie sztucznych pomiarów. Najczęściej wstawia się średnią lub medianę. Jeszcze innym, lepszym rozwiązaniem, jest dopasowanie modelu regresji na zbiorze kompletnych danych. Wstawiając w miejsca brakujące wartości wyznaczone z modelu regresji. Do wstawienia wartości można wykorzystać funkcję *impute()*, która wymaga zainstalowania pakietu *Hmisc*.

Przykład – *impute()*

```
wektorZBrakamiDanych <- dane$HGB
impute(wektorZBrakamiDanych, 10) # imputuj wartosc 10 w miejsca NA
impute(wektorZBrakamiDanych, mean) # imputuj srednia
mean(wektorZBrakamiDanych, na.rm = TRUE) #oblicz srednia z pominiciem NA
```

3 Statystyki opisowe

Statystyki opisowe wykorzystuje się do przedstawienia charakterystyki badanych grup poprzez ocenę parametrów. W przypadku zmiennych o charakterze mierzalnym statystyki opisowe pozwalają na określenie jak często badana cecha występuje, jaki jest rozkład badanej cechy lub jaka wartość jest najbardziej reprezentatywna. Podczas gdy, w przypadku zmiennych o charakterze niemierzalnym można określić liczność danego parametru. W przykładzie poniżej wykonano kilka z podstawowych statystyk opisowych, więcej funkcji przedstawiono na wykładzie.

Przykład – podstawowe statystyki opisowe

```
dane <- read.csv2("przykladoweDaneBezBrakow.csv", sep = ";")
attach(dane)

range(wiek)
mean(wiek)
median(wiek)
IQR(wiek)
var(wiek)
sd(wiek)
```

Wartości uzyskane w wyniku przeprowadzenia statystyk opisowych są podstawą dla dalszej analizy i wnioskowania statystycznego. Poniżej opisano inne przydatne funkcje R, np. *summary()*, która umożliwia podsumowanie wektora obserwacji. Wynik działania tej funkcji będzie inny w zależności od charakteru danych (ilościowych lub jakościowych).

Przykład – *summary()* dla danych ilościowych i jakościowych

```
summary(wiek)
summary(plec)
```

Obok funkcji *summary()* ważna jest też funkcja *table()*, która została wspomniana we wcześniejszych protokołach. Funkcja *table()* wyznacza tablice kontyngencji jednej lub większej liczby zmiennych wyliczeniowych.

Przykład – *table()*

```
table(plec)
table(grupa, plec)
```

Kolejną funkcją służącą do podsumowania jest *summarise()*, funkcje tą wykorzystuje się w przypadku podsumowania względem grup (w przypadku korzystania z funkcji *group_by()*). Do zastosowania powyższych funkcji konieczna jest instalacja pakietu *dplyr*. W poniższym przykładzie wykonano kilka podstawowych statystyk opisowych (*count*, *mean*, *sd*, *median*), które zostały obliczone dla całej próby z zastosowaniem grupowania:

Przykład – *group_by()*, *summarise()*

```
podsumowanie_wiek <- group_by(dane, grupa) %>%
    summarise(
    count = n(),
    mean = format(round(mean(wiek, na.rm = TRUE), 2), nsmall = 2),
    sd = format(round(sd(wiek, na.rm = TRUE), 2), nsmall = 2),
    median = format(round(median(wiek, na.rm = TRUE), 2), nsmall = 2)
)
print(podsumowanie_wiek)
```

Chcąc zmienić kolejność wyświetlanych grup, należy wspomnieć o poziomach *levels()* i funkcji *ordered()*. Poziomy pokazują wszystkie możliwe warianty z danej kolumny (bez powtórzeń), podczas gdy funkcja *ordered()* pozwala zmienić ich kolejność. Funkcje te są istotne na przykład jeśli chcemy zmienić kolejność wyświetlanych grup na wykresach lub kolejność wyświetlania wyników w tabelach lub podsumowaniach.

Przykład – *levels()* i *ordered()*

```
dane$grupa
#zmiana kolejnosci w poziomie danych
dane$grupa <- ordered(dane$grupa, levels = c("KONTROLA", "CHOR1", "CHOR2"))
levels(dane$grupa)
```

4 Ocena zgodności danych z rozkładem normalnym

Dla każdego analizowanego parametru określa się jego zgodność z rozkładem normalnym. Rozkład normalny charakteryzuje się dzwonowatym kształtem krzywej symetrycznej względem wartości średniej. Spełnienie założenia o normalności z rozkładem normalnym jest często wymogiem do zastosowania odpowiednich testów statystycznych. Najczęściej stosowane testy do oceny zgodności z rozkładem normalnym: test Kołmogorowa-Smirnowa (*ks.test()*) i test Shapiro-Wilka (*shapiro.test()*).

Przykład – *shapiro.test()* dla parametru hsCRP z przykładowych danych

```
KONTROLA <- with(dane, dane[grupa == "KONTROLA", ])
CHOR1 <- with(dane, dane[grupa == "CHOR1", ])
CHOR2 <- with(dane, dane[grupa == "CHOR2", ])

shapiro.test(CHOR1$hsCRP)
shapiro.test(CHOR2$hsCRP)
shapiro.test(KONTROLA$hsCRP)
```

Przykład – *shapiro.test()* z wykorzystaniem grupowania dla parametru hsCRP

```
pvalueShapiroTestHSCRP <- group_by(dane, grupa) %>%
    summarise(
        statistic = shapiro.test(hsCRP)$statistic,
        p.value = shapiro.test(hsCRP)$p.value
    )
pvalueShapiroTestHSCRP
```

Przykład – Wydobywanie konkretnych wartości poprzez ustawione zmienne

```
#wydobycie p-value dla wszystkich grup
pvalueShapiroTestHSCRP$p.value

#wydobycie p-value dla konkretnej grupy
pvalueShapiroTestHSCRP$p.value[(pvalueShapiroTestHSCRP$grupa == "CHOR1")]
```

Jeśli wartość p-value > 0.05 oznacza to, że rozkład danych nie różni się znacząco od rozkładu normalnego (możemy założyć normalność danych).

Graficzna ocena zgodności z rozkładem normalnym (do utworzenia wykresów wymagany jest pakiet *ggpubr*). Wykres pozwalający na sprawdzenie zgodności z rozkładem normalnym to wykres gęstości. Wykres ten zapewnia wizualną ocenę tego, czy rozkład ma kształt dzwonu. Funkcja *facet_wrap()* umożliwia dodanie oddzielnych wykresów dla każdej grupy.

Przykład – *ggdensity()*

```
ggdensity(dane, x = "hsCRP",
    color = "grupa",
    fill = "grupa",
    palette = c("#99cc00", "#660099", "#0047b3"),
    ylab = "gestosc",
    xlab = "hsCRP [mg/l]"
)
```

Przykład – *ggedensity()* + *facet_wrap()*

```
ggdensity(dane, x = "hsCRP",
    color = "grupa",
    fill = "grupa",
    palette = c("#99cc00", "#660099", "#0047b3"),
    ylab = "gestosc",
    xlab = "hsCRP [mg/l]"
) + facet_wrap(~ grupa, scales = "free")
```

W przypadku zastosowania testów wymagających zgodności z rozkładem normalnych dla danych niezgodnych z rozkładem normalnym trzeba liczyć się z otrzymaniem nieprawidłowych wyników, a w konsekwencji nieprawidłowego wnioskowania.

5 Ocena homogeniczności wariancji

W przypadku niektórych testów wymagana jest zarówno zgodność z rozkładem normalnym jak i wymagane jest spełnienie założenia o homogeniczności (jednorodności) wariancji. Najczęściej stosowane testy do oceny jednorodności wariancji: test Fishera, test Levene'a, poniżej przykład użycia testu Levene'a (pakiet *car*).

Przykład – *leveneTest()*

```
leveneTest(hsCRP ~ grupa, data = dane)
leveneTest(hsCRP ~ grupa, data = dane)$"Pr(>F)"[1] #wydobycie p-value
```

Jeśli wartość p-value > 0.05 oznacza to, że dane są zgodne z założeniem o jednorodności wariancji (możemy założyć homogeniczność danych).

Podobnie jak w przypadku oceny zgodności z rozkładem normalnym, zastosowanie testów wymagających spełnienia założenia o jednorodności wariancji dla danych o niejednorodnych wariancjach prowadzi do otrzymania nieprawidłowych wyników, a w konsekwencji do wyciągnięcia nieprawidłowych wniosków. Gorsze implikacje niesie niespełnienie założenia o homogeniczności wariancji niż niespełnienie założenia o rozkładzie normalnym.

6 Porównywanie grup (porównywanie średnich)

W pierwszej kolejności należy określić charakter porównywanych grup. Czy są to grupy zależne czy niezależne. W przypadku grup zależnych, mówimy o badaniu tych samych parametrów u tej samej grupy, np. badanie pewnych parametrów w pewnych odstępach czasowych. Natomiast w przypadku grup niezależnych mowa o badaniu tych samych parametrów u różnych grup pacjentów np. badanie pewnych parametrów zarówno w grupie kontrolnej, jak i grupach chorych (np. w różnym stadium choroby). W drugim kroku należy określić liczbę porównywanych grup: 2 czy więcej niż 2. Następnie na podstawie oceny charakteru danych (spełnienia lub niespełnienia założenia o zgodności z rozkładem normalnym i o jednorodności wariancji), można określić jakie testy należy zastosować w przypadku porównywania grup. Jako, że testy parametryczne wymagają spełnienia założeń o zgodności z rozkładem normalnym i spełnienia założeń o homogeniczności wariancji, częściej stosowane są testy nieparametryczne. Jednak testy nieparametryczne są słabsze. Testy do porównywania grup pozwalają określić czy pomiędzy badanymi grupami występują istotne statystycznie różnice. W przypadku gdy mamy więcej niż 2 grupy, należy dodatkowo określić pomiędzy którymi grupami te różnice występują, do tego wykorzystuje się testy *post hoc*. W Tabeli 1 przedstawiono schemat wyboru testu statystycznego dla 2 i > 2 grup niezależnych:

Tablica 1: Wyboru testu statystycznego dla 2 i > 2 grup niezależnych.

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 Wilcoxona (Manna-Whitneya)
>2	TAK	TAK	test ANOVA (*post hoc* Tukeya)
		NIE	test Kruskala-Wallisa (*post hoc* Dunna)
	NIE	-	

6.1 Wybrane przykłady testów statystycznych dla >2 grup niezależnych

Jeśli dane nie spełniają założenia o zgodności z rozkładem normalnym (p-value < 0.05) do analizy porównawczej wykorzystuje się testy nieparametryczne, np. test Kruskala-Wallisa, funkcja *kruskal.test()*. Tak samo w przypadku, gdy dane są zgodne z rozkładem normalnym, ale nie spełniają założenia o jednorodności wariancji to również stosuje się test Kruskala-Wallisa. Zastosowanie wspomnianego testu przedstawiono na poniższym przykładzie: Przykład – *kruskal.test()*

```
kruskal.test(HGB ~ grupa, data = dane)
pvalueKWtestHGB <- kruskal.test(HGB ~ grupa, data = dane)$p.value

```

```r
if(pvalueKWtestHGB < 0.05){
    cat(pvalueKWtestHGB, "< 0.05 - sa roznice pomiedzy grupami")
}else{
    cat(pvalueKWtestHGB, "> 0.05 - brak roznic pomiedzy grupami")
}
```

Gdy wartość p-value jest mniejsza niż poziom istotności 0.05 możemy stwierdzić, że istnieją znaczące różnice między grupami. Jeśli występują istotne statystycznie różnice pomiędzy grupami, należy dokładnie określić pomiędzy którymi grupami występują te różnice. Chcąc ocenić pomiędzy jakimi grupami są różnice i jak duże są te różnice należy zastosować testy *post hoc*. Poniżej przedstawiono zastosowanie testu Dunna, umożliwia to funkcja *dunnTest*, wymagane pakiety to *dunn.test* i *FSA*:

Przykład – *dunnTest()*

```r
dunnTest(dane$HGB, dane$grupa)
```

Należy rozważyć jeszcze inny wariant danych; jeśli dane spełniają założenie o zgodności z rozkładem normalnym (p-value > 0.05) oraz spełniają założenie o jednorodności wariancji (p-value > 0.05) to stosuje się parametryczny test ANOVA, przy użyciu funkcji *aov()*. Zastosowanie parametrycznego testu ANOVA przedstawiono na poniższym przykładzie:

Przykład – *aov()*

```r
aov(MCHC ~ grupa, data = dane)
summary(aov(MCHC ~ grupa, data = dane))
pvalueAOVtestMCHC <- summary(aov(MCHC ~ grupa, data = dane))[[1]][["Pr(>F)"]][[1]]

if(pvalueAOVtestMCHC < 0.05){
    cat(pvalueAOVtestMCHC, "< 0.05 - sa roznice pomiedzy grupami")
}else{
    cat(pvalueAOVtestMCHC, "> 0.05 - brak roznic pomiedzy grupami")
}
```

Podobnie jak w przypadku testu nieparametrycznego, gdy wartość p-value jest mniejsza niż poziom istotności 0.05 możemy stwierdzić, że istnieją znaczące różnice między grupami. Chcąc określić pomiędzy którymi grupami występują te różnice należy zastosować testy *post hoc*. W przypadku testu ANOVA, często stosowany jest test Tukeya, w R umożliwia to funkcja *TukeyHSD()*:

Przykład – *TukeyHSD()*

```r
TukeyHSD(aov(MCHC ~ grupa, data = dane))
```

Wspomnieć należy też o zmiennych o innym charakterze. W przypadku zmiennych jakościowych (nominalnych) należy wykonać test χ^2, w tym celu wykorzystuje się funkcję *chisq.test()*. Przykład zastosowania poniżej + wizualizacja:

Przykład – *chisq.test()*

```r
chisq.test(dane$grupa, dane$plec)
pvalueChisqPlec <- chisq.test(dane$grupa, dane$plec)$p.value

barplot(table(dane$plec, dane$grupa),
    ylim = c(0,20),
    beside = TRUE,
```

```
7       col = c("#ffb3b3", "#b3d1ff"),
8       xlab = "grupa",
9       ylab = "plec",
10      legend = c("kobieta", "mezczyzna")
11      )
12  text(7.2, 16, paste("p-value", round(pvalueChisqPlec, digits = 3)))
```

6.2 Wybrane przykłady testów statystycznych dla 2 grup niezależnych

Jeśli dane nie spełniają założenia o zgodności z rozkładem normalnym (p-value < 0.05) do analizy porównawczej do wykonuje się testy nieparametryczne. Można wykonać test Wilcoxona (dokładniej test sumy rang Wilcoxona zwany również testem Manna–Whitneya). Funkcja umożliwiająca przeprowadzenie wspomnianego testu to *wilcox.test()*. Jeśli wartość p-value jest mniejsza niż poziom istotności 0.05 możemy stwierdzić, że istnieją znaczące różnice między grupami. Zastosowanie nieparametrycznego testu Manna–Whitneya (Wilcoxona) przedstawiono na poniższym przykładzie:

Przykład – *wilcox.test()*

```
1  dwieGrupy <- read.csv2("przykladowe2Grupy.csv", sep = ";")
2  wilcox.test(hsCRP ~ grupa, data = dwieGrupy)
```

Jeśli dane spełniają założenie o zgodności z rozkładem normalnym (p-value > 0.05) oraz spełniają założenie o jednorodności wariancji (p-value > 0.05) to stosuje się test t-Studenta. Test ten można wykonać przy pomocy funkcji *t.test()*, z argumentem o *var.equal = TRUE*. Jeśli wartość p-value dla testu t-Studenta jest mniejsza niż poziom istotności 0.05 możemy stwierdzić, że istnieją znaczące różnice między grupami. Zastosowanie parametrycznego testu t-Studenta przedstawiono na poniższym przykładzie:

Przykład – *t.test()*, *var.equal = TRUE*

```
1  t.test(LEU ~ grupa, data = dwieGrupy, var.equal = TRUE)
```

Należy rozważyć jeszcze inny wariant; jeśli dane spełniają założenie o zgodności z rozkładem normalnym (p-value > 0.05), ale nie spełniają założenia o jednorodności wariancji (p-value < 0.05) to stosuje się test Welcha. W tym celu można wykorzystać funkcję *t.test()* i dostosować ją do określonej sytuacji poprzez argument *var.equal*. Wartość argumentu powinna przyjąć FALSE (*var.equal = FALSE*). Przykład użycia:

Przykład – *t.test()*, *var.equal = FALSE*

```
1  t.test(LEU ~ grupa, data = dwieGrupy, var.equal = FALSE)
```

7 Ocena zależności pomiędzy parametrami

Ocena istnienia i siły korelacji pomiędzy wybranymi parametrami, a także określenia kształtu i kierunku tej zależności stanowią ważną część analizy statystycznej. Niezwykle istotny jest fakt, że stwierdzenie występowania zależności nie zawsze oznacza występowanie związku przyczynowo-skutkowego pomiędzy badanymi parametrami. Oznacza to tylko tyle, że techniczne wykonanie analizy bez wiedzy eksperckiej w określonej tematyce może prowadzić do wykrycia pozornych korelacji, a w konsekwencji niepoprawnej interpretacji wyników. Do przeprowadzenia testu korelacji służy funkcja *cor.test()*, wybór metody określa się w argumencie method, który przyjmuje argument: *pearson*, *kendall*, *spearman*. Najczęściej stosowaną miarą siły i kierunku zależności jest współczynnik korelacji liniowej Pearsona (parametryczny test korelacji). Nieparametrycznym odpowiednikiem współczynnika korelacji Pearsona jest współczynnik korelacji rangowej Spearmana. Przykłady użycia wspomnianych testów poniżej:

Przykład – *cor.test() spearman*

```r
CHOR1 <- dane %>% filter(grupa == "CHOR1")

wynikTestuKorelacjiSpearmana <- cor.test(CHOR1$HGB, CHOR1$ERY, method = "spearman")
wynikTestuKorelacjiSpearmana$estimate
wynikTestuKorelacjiSpearmana$p.value
```

Przykład – *cor.test() pearson*

```r
KONTROLA <- dane %>% filter(grupa == "KONTROLA")

wynikTestuKorelacjiPearsona <- cor.test(KONTROLA$HGB, KONTROLA$ERY, method = "pearson")
wynikTestuKorelacjiPearsona$estimate
wynikTestuKorelacjiPearsona$p.value
```

Na podstawie wykonanych testów korelacji można określić czy zależność pomiędzy badanymi parametrami jest istotna statystycznie. W przypadku testu korelacji Pearsona i Spearmana jeśli wartość p-value jest mniejsza od 0.05 to wskazuje to korelację pomiędzy zmiennymi. Współczynnik korelacji r może być interpretowany następująco:

- $r > 0$ korelacja dodatnia – gdy zmienna X rośnie to Y także rośnie,
- $r = 0$ brak korelacji – gdy zmienna X rośnie to Y czasem rośnie a czasem maleje,
- $r < 0$ korelacja ujemna – gdy zmienna X rośnie to Y maleje.

Natomiast siła korelacji może być interpretowana zgodnie z poniższymi przedziałami:

- $-1 < r < -0.7$ bardzo silna korelacja ujemna
- $-0.7 < r < -0.5$ silna korelacja ujemna
- $-0.5 < r < -0.3$ korelacja ujemna o średnim natężeniu
- $-0.3 < r < -0.2$ słaba korelacja ujemna
- $-0.2 < r < 0.2$ brak korelacji
- $0.2 < r < 0.3$ słaba korelacja dodatnia
- $0.3 < r < 0.5$ korelacja dodatnia o średnim natężeniu
- $0.5 < r < 0.7$ silna korelacja dodatnia
- $0.7 < r < 1$ bardzo silna korelacja dodatnia

Graficzne przedstawienie korelacji pomiędzy parametrami; na potrzeby utworzenia wykresów korelacji można wykorzystać funkcję *ggscatter()* z pakietu *dplyr*. Należy zwrócić uwagę, żeby przy tworzeniu wykresu określić stosowaną metodę, jeśli chcemy uzyskać poprawne dane co do wartości współczynnika i wartości p-value (służy do tego argument *cor.method)*, poniżej przykład dla wsp. korelacji Pearsona, z dodatkowym argumentem o dodaniu regresji liniowej.

Przykład – *cor.test() pearson*

```r
ggscatter(KONTROLA, x = "HGB", y = "ERY",
      add = "reg.line", conf.int = TRUE,
      cor.coef = TRUE, cor.method = "pearson",
      color = "grupa", fill = "grupa",
      ylab = "HGB [gl/dl]",
      xlab = "ERY [t/l]"
      )
```

Zadanie

Dla przykładowych danych (www.cs.put.poznan.pl/kgutowska/PSwBB/dane/przykladoweDane-Projekt.csv) zaproponuj i wykonaj analizę statystyczną. Rozważ poniższe etapy (pamiętaj, że istotny jest charakter danych i założenia jakie one spełniają):

- przygotowanie danych wejściowych,
- statystyka opisowa danych,
- analiza porównawcza pomiędzy grupami,
- analiza korelacji,
- wizualizacja danych na poszczególnych etapach.

Sprawozdanie powinno zawierać opis poszczególnych etapów wraz z opisem wyników. Umieść w sprawozdaniu schemat blokowy działania swojego programu. Sprawozdanie i kod R prześlij na maila (Kaja.Gutowska@cs.put.poznan.pl). Proszę nadać tytuł maila zgodnie z opisem: PSwBB_Imię_Nazwisko_lab1. Sprawozdanie należy oddać przed kolejnymi zajęciami.