

Grafika cz. 1

Zadanie 8.1 (MG; Wykres kołowy).

Wysokopoziomowa funkcja `pie()` służy do rysowania wykresów kołowych dla danych jakościowych. Przykład:

```
dzieci <- c(krasnale = 48, zuchy = 69, wesolki = 32)
pie(dzieci)
```

Napisz swoją własną implementację `pie()` w postaci funkcji `mypie()`, wzorując się na „bazowej” implementacji. Przyjmij, że dopuszczalnymi danymi wejściowymi będą wektory liczbowe z ustawionym atrybutem `names` (liczby obserwacji przypadające na daną klasę) oraz zmienne typu `factor`.

Zadanie 8.2 (MG; Wykres skrzynkowy).

Utwórz funkcję `myboxplot()`, która zawiera Twoją własną implementację funkcji rysującej wykres skrzynkowy dla danych ilościowych, por. `?boxplot`. Pamiętaj o poprawnym wyróżnianiu obserwacji odstających (*outliers*).

Zadanie 8.3 (MG; Sin/cos).

Narysuj, korzystając tylko z funkcji niskopoziomowych, wykres funkcji sinus i kosinus na przedziale $[0, 2\pi]$. Układ współrzędnych narysuj ręcznie tak, by na osi Ox oznaczone były wartości $0, \frac{\pi}{2}, \pi, \frac{3}{2}\pi, 2\pi$ (skorzystaj z symboli opisanych na stronie podręcznika `?plotmath`), a na Oy – tylko $-1, 0, 1$. Dodaj odpowiednią legendę podobną do tej, którą wygenerowałaby funkcja `legend()` (lub ładniejszą).

Zadanie 8.4 (MG; Osie).

Napisz funkcję `myaxis()` (por. `?axis`), która narysuje układ współrzędnych (osie i ich etykiety) na Twój własny, ulubiony sposób. Skorzystaj m.in. z wartości parametru graficznego `usr`.

Zadanie 8.5 (AO; Wykres wieloskrzynkowy).

Napisz funkcję `mymultiboxplot()`, która dla danego wektora liczbowego x i czynnika g o k poziomach (wektor i czynnik są równoliczne) narysuje jeden wykres składający się z ustawionych obok siebie k „skrzynek” (por. zad. 8.2) dla wartości z wektora x odpowiadających kolejnym poziomom czynnika g . Możesz uzyskać efekt podobny do poniższego wywołania funkcji wysokopoziomowej `boxplot()`.

```
x <- rnorm(30)
g <- factor(rep(1:3, each = 10))
boxplot(split(x, g)) # u nas: mymultiboxplot(x, g)
```

Dodatkowo zaznacz na wykresach przedziały $[\bar{x}_i - cs_i, \bar{x}_i + cs_i]$, gdzie \bar{x}_i i s_i oznaczają, odpowiednio, średnią i odchylenie standardowe wartości z wektora x dla i -tego poziomu czynnika. Dla parametru $c \in \mathbb{N}$ przyjmij domyślnie wartość 2. Co więcej, wartości średnie połącz przy użyciu łamanej.

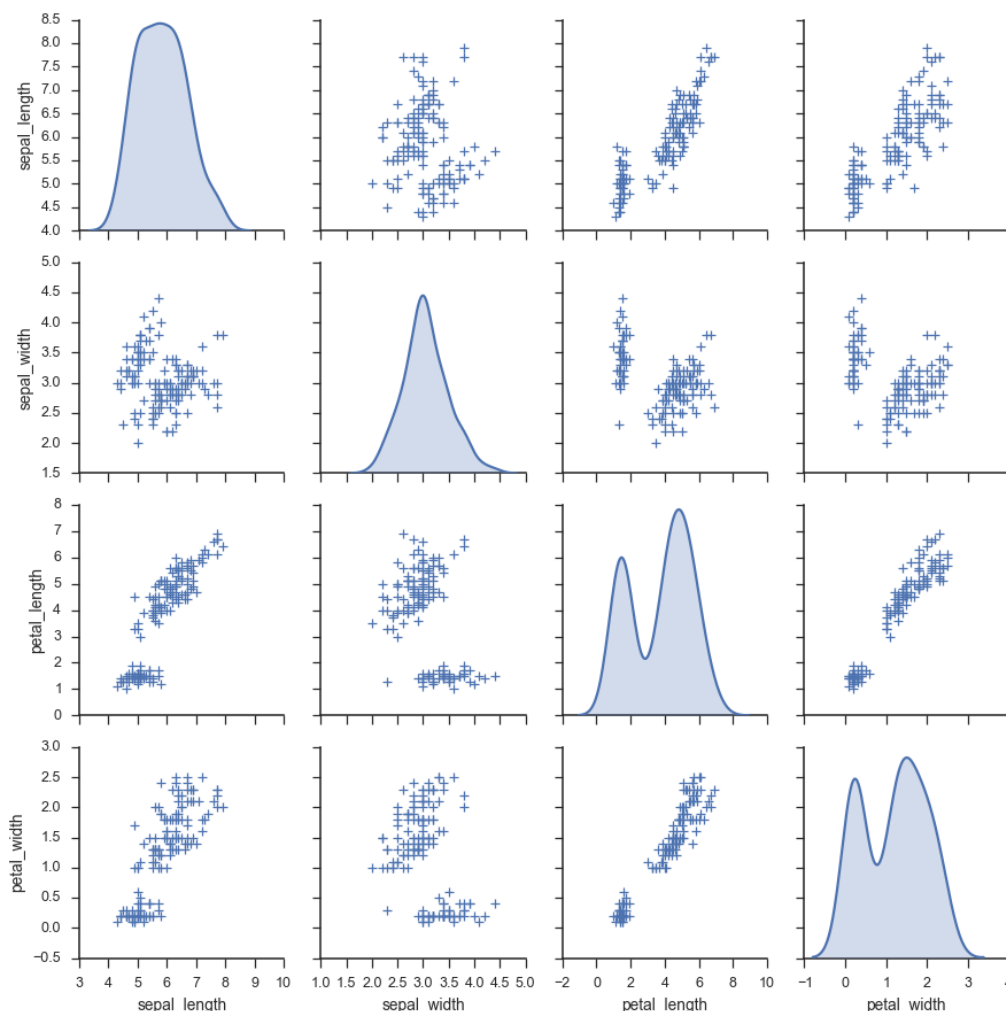
Zadanie 8.6 (MG; Wykres prostokątny).

Zaimplementuj funkcję `rectplot()`, służącą do rysowania tego, co tutaj roboczo nazwiemy wykresem prostokątnym dla danych jakościowych (por. zad. 8.1). Prostokąt wypełniający cały obszar rysowania podziel dowolnie na podprostokąty o polach proporcjonalnych do liczby obserwacji w każdej z klas. Każdy podobzdar powinien być wypełniony inną barwą. Ponadto wewnątrz niego powinna znajdować się etykieta poziomu czynnika i informacja o procentowym udziale obserwacji z tej klasy w zbiorze wejściowym.

Zadanie 8.7 (JL; Wstępna analiza danych).

Napisz funkcję `pairplot()`, która dla ramki danych zawierającej jedynie kolumny numeryczne sporządzi wykresy wszystkich par zmiennych oraz ich histogramów (por. `?pairs`).

Dokładniej, funkcja ta sporządza n^2 wykresów na siatce $n \times n$, gdzie n jest liczbą kolumn. Wykres na polu (i, j) , $i \neq j$ to wykres punktowy dla i -tej względem j -tej zmiennej w danych. Z kolei pole na przekątnej (i, i) wyznacza histogram (ew. jądrowy estymator gęstości) i -tej zmiennej. Pierwsza/ostatnia kolumna/wiersz powinny wskazywać na nazwy zmiennych. Przykładowy efekt wywołania funkcji dla danych `iris[,1:4]` jest zaprezentowany niżej.



Zadanie 8.8 (MG; Europa).

Napisz funkcję `europa()`, która jako argument przyjmuje wektor napisów (być może pusty) `countries`. Funkcja rysuje mapę Europy – z wyraźnie zaznaczonymi granicami państw. Każdy obszar państwa wymienionego w `countries` (sposób podania: zgodny z ISO 3166-1 alpha-2, np. "PL", "DE" itd.) powinno być pokolorowane inną barwą. Pozostałe, nie wymienione państwa niczym wyróżniać się nie powinny.

Uwaga ogólna: To zadanie wymaga pobuszowania w internecie (na pewno znajdziesz coś, co będzie dobrym punktem wyjścia; słowo kluczowe: `spatial data`). Interesuje nas wynik, nie sposób działania.

Zadanie 8.9 (MG; Wykres słupkowy).

Wysokopoziomowa funkcja `barplot()` służy do rysowania wykresów słupkowych dla danych jakościowych. Przykład:

```
dzieci <- c(krasnale = 48, zuchy = 69, wesolki = 32)
barplot(dzieci)
```

Napisz swoją własną implementację `barplot()` w postaci funkcji `mybarplot()`.

Zadanie 8.10 (*MG; Histogram*).

Funkcja `hist()` – która służy do wyznaczania i opcjonalnie rysowania histogramu – zwraca:

```
x <- c(...) # Twoje dane
h <- hist(x, plot = FALSE) # nie rysuj, tylko wyznacz histogram
```

Napisz funkcję `myhist1()`, która dla danego jako argument wektora liczbowego \mathbf{x} wyznaczy histogram jak wyżej (obiekt klasy `histogram`, który jest listą) i narysuje go na Twój ulubiony, samodzielnie opracowany sposób.

Zadanie 8.11 (*MG; Histogram cd.*).

Utwórz funkcję `myhist2()`, która zawiera Twoją własną implementację najważniejszych działań wykonywanych przez `hist()`, włącznie z wyznaczaniem histogramu, zwracaniem go w postaci listy klasy `histogram` i ewentualnym rysowaniem. Możesz skorzystać z części kodu napisanego w ramach rozwiązania zadania 8.10. Tym razem jakiegokolwiek wywoływanie `hist()` jest niedopuszczalne.

Zadanie 8.12 (*BT; Graf prosty*).

Napisz funkcję rysującą graf prosty na podstawie podanej zerojedynkowej, symetrycznej macierzy sąsiedztwa. Wierzchołki (sposób rysowania dowolny) mają być rozlokowane równomiernie na okręgu. Krawędzie reprezentowane są przez odcinki łączące wierzchołki.

Zadanie 8.13 (*MG; Dystrybuanta empiryczna*).

Napisz funkcję `Fn()`, która dla danego wektora liczbowego $\mathbf{x} = (x_1, \dots, x_n)$ o unikatowych wartościach narysuje przy użyciu funkcji niskopoziomowych (bez użycia wbudowanych funkcji `ecdf()` oraz `approxfun()`) jego dystrybuantę empiryczną, która jest dana wzorem:

$$\hat{F}_n(y) = \begin{cases} 0 & \text{dla } y < x_{(1)}, \\ i/n & \text{dla } i < n \text{ takiego, że } x_{(i)} \leq y < x_{(i+1)}, \\ 1 & \text{dla } y \geq x_{(n)}, \end{cases}$$

gdzie $x_{(i)}$ oznacza i -tą najmniejszą wartość z \mathbf{x} .

Zadanie 8.14 (*MG; Takie tam*).

Napisz funkcję `krzywa()`, która przy użyciu funkcji niskopoziomowych narysuje przybliżony kształt krzywej zamkniętej danej równaniem parametrycznym:

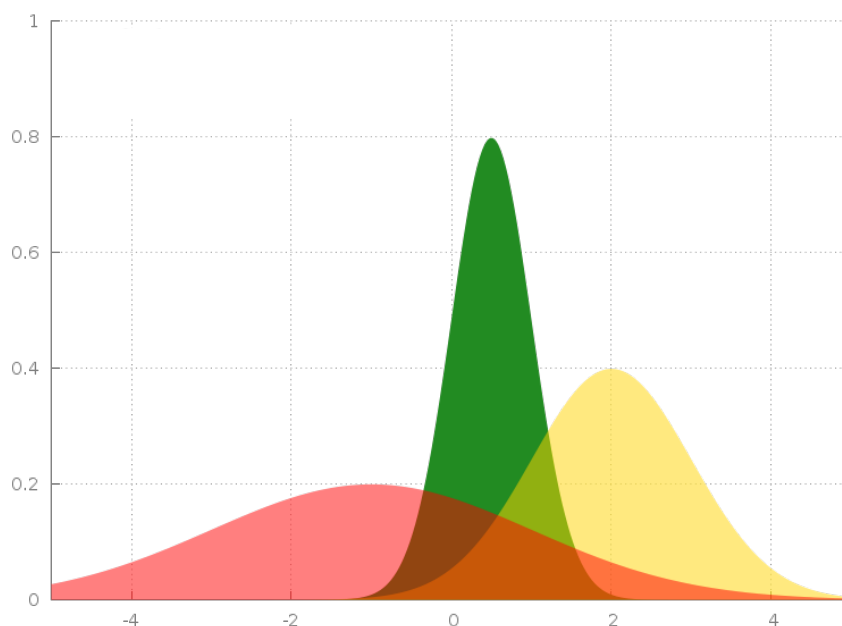
$$\begin{cases} x(t) &= 16 \sin^3 t, \\ y(t) &= 13 \cos t - 5 \cos 2t - 2 \cos 3t - \cos 4t, \end{cases}$$

dla $t \in [0, 2\pi)$. Ponadto wypełnij jej wnętrze daną barwą (domyślnie czerwoną).

Zadanie 8.15 (*MG; Przezroczysty multiplot*).

Napisz funkcję `multiplot()`, która jako argument przyjmuje n -elementowy, posortowany rosnąco wektor liczbowy \mathbf{x} oraz n -wierszową macierz rzeczywistą \mathbf{y} o k kolumnach.

Funkcja ma rysować wykresy y_i jako funkcję x , dla każdego $i = 1, \dots, k$ (interpolacja kawałkami liniowa). Pole pod krzywymi ma być wypełnione półprzezroczystą barwą, jak w poniższym przykładzie:

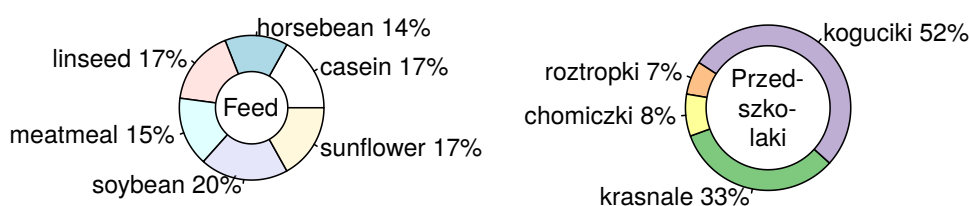


Wskazówka: „ładne” etykiety dla osi OX i OY możesz wygenerować przy użyciu funkcji `pretty()`.

Zadanie 8.16 (MG; Ciastko z dziurką).

Napisz funkcję `nicepie()`, która tworzy poniższy wykres kołowy przy użyciu wywołania funkcji `pie()` i funkcji niskopoziomowych. Funkcja powinna przyjmować następujące argumenty: `x` (zbiór danych), `main` (tytuł wykresu), `r1` (promień „dużego” koła, domyślnie 0.7), `r2` (promień wypełnionego barwą białą „małego” koła o czarnym brzegu, domyślnie `r1/2`).

```
nicepie(table(chickwts$feed), 'Feed') # wbudowana ramka danych chickwts
library('RColorBrewer')
dane <- c(krasnale=20, koguciki=32, roztropki=4, chomiczki=5)
nicepie(dane, 'Przed-\n szko-\nlaki', 0.8, 0.6,
        init.angle=200, col=brewer.pal(4, 'Accent'))
```



Zadanie 8.17 (AO; Karta kontrolna).

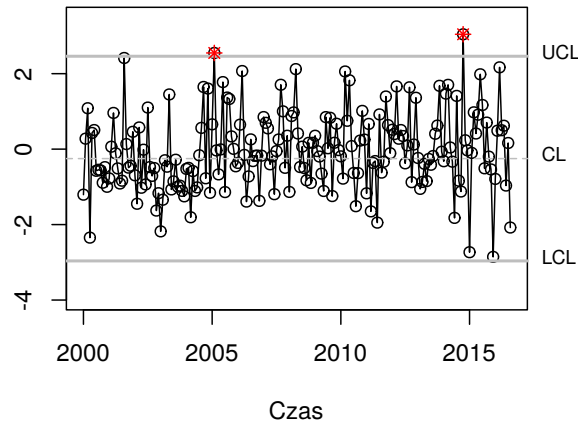
Napisz funkcję `ctrlchart()`, która przy użyciu metody `plot.ts()` narysuje przebieg wartości danego szeregu czasowego `y` składającego się z n obserwacji.

Dodatkowo zaznacz na wykresie trzy poziome linie: $CL = \bar{y}$, $UCL = \bar{y} + 3 \frac{MR}{\sqrt{1,128}}$ i $LCL = \bar{y} - 3 \frac{MR}{\sqrt{1,128}}$ (spraw, by były zawsze widoczne), gdzie \bar{y} jest średnią arytmetyczną z pierwszych m (argument wejściowy, domyślnie równy $\lfloor n/10 \rfloor$) obserwacji szeregu, zaś MR jest średnim ruchomym rozstępem z m pierwszych obserwacji szeregu wyznaczonym ze wzoru $MR = \sum_{i=1}^m |y_{i+1} - y_i| / m$. Wyróżnij te obserwacje z `y`, które znajdują się poza przedziałem $[LCL, UCL]$ innym symbolem i barwą. Pamiętaj, by poprawnie uwzględnić atrybuty `frequency`, `start` itp. szeregu czasowego, por. `?time`.

```
set.seed(1234)
y <- ts(c(rnorm(200, 0, 1)), frequency = 12, start=2000)
```

```
ctrlchart(y, 20, type = "o")
```

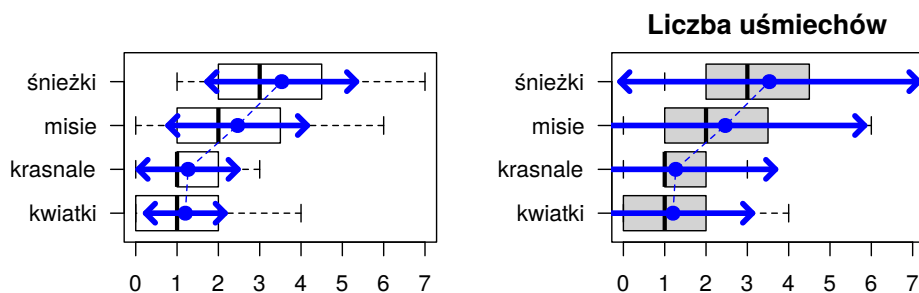
Ciekawostka. W przypadku, gdy obserwacje pochodzą z rozkładu normalnego, tak skonstruowany wykres jest tzw. kartą kontrolną pojedynczych pomiarów stosowaną często w statystycznej kontroli jakości. Obserwacje poza przedziałem [LCL, UCL] interpretuje się wówczas jako sygnały alarmowe świadczące o potencjalnym rozregulowaniu się procesu produkcji.



Zadanie 8.18 (AO; Wariacje na temat wykresu skrzynkowego).

Napisz funkcję `myboxplots()`, która dla danego wektora liczbowego `x` i czynnika `g` o takiej samej długości narysuje przy użyciu funkcji `boxplot()` wykresy skrzynkowe dla `x` podzielonego na odpowiednie klasy. Dodatkowo na rysunku należy zaznaczyć średnią wartość obserwacji z każdej klasy oraz przedział $\pm d \times \text{odchylenie standardowe}$, gdzie `d` jest argumentem wejściowym domyślnie równym 1. Średnie należy połączyć przy użyciu łamanej. Kolejność rysowania „pudełek” – tak, by średnie były w kolejności rosnącej.

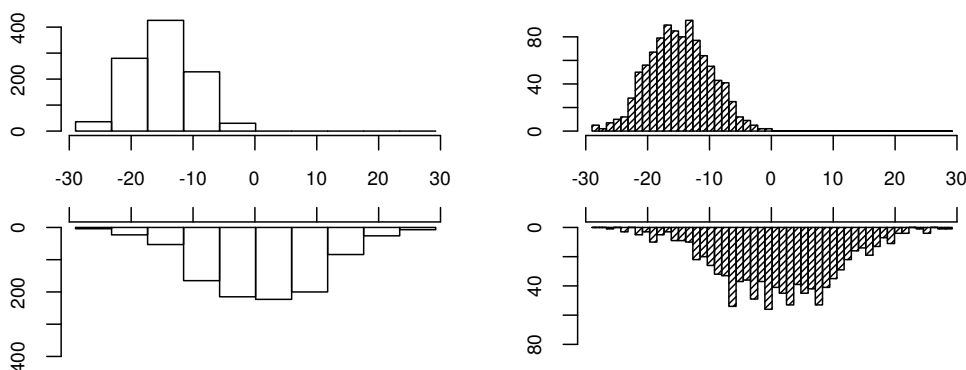
```
set.seed(123)
x <- as.integer(c(rpois(15, 1), rpois(15, 2),
  rpois(15, 1.4), rpois(15, 4)))
g <- factor(rep(c('krasnale', 'misie', 'kwiatki', 'śnieżki'), each=15))
myboxplots(x, g) # lewy rysunek
myboxplots(x, g, 2, main='Liczba uśmiechów', col='lightgray') # prawy
```



Zadanie 8.19 (AO; Lustrzany histogram).

Napisz funkcję `hist2()`, która dla danych dwóch wektorów liczbowych `x` i `y` o długości, odpowiednio, n_x i n_y narysuje dwa histogramy na jednym rysunku. Granice klas histogramów oraz zakresy rysunków na Ox i Oy powinny być w obydwu przypadkach takie same. Liczbę klas kontroluje (nie wprost) argument wejściowy `h`, domyślnie równy $\lceil \log_2(\max\{n_x, n_y\}) + 1 \rceil$.

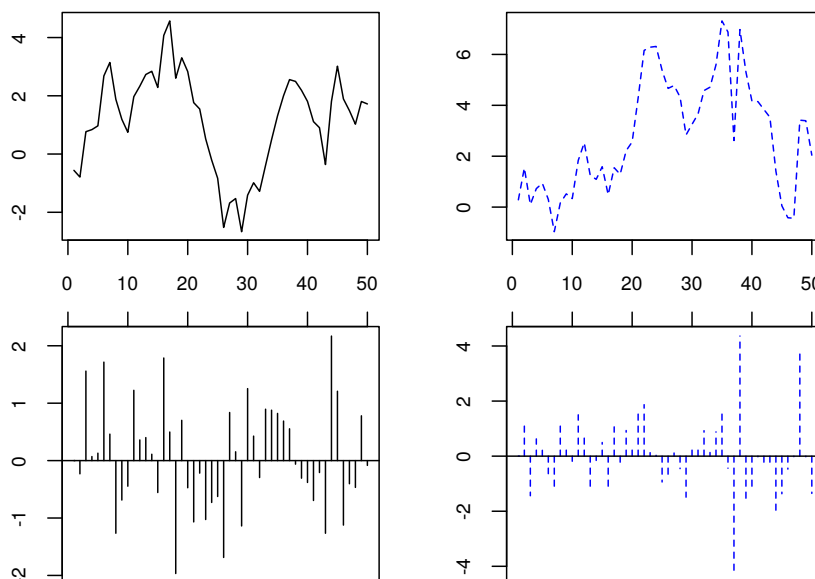
```
set.seed(321)
x <- rnorm(1000, -15, 5); y <- rnorm(1000, 1, 9)
hist2(x, y) # h domyślne, lewy rysunek
hist2(x, y, 50, density=30) # h=50, prawy rysunek
```



Zadanie 8.20 (MG; Wykres funkcji i jej przyrostów).

Napisz funkcję `plotdiff()`, która przy użyciu dwóch wywołań funkcji `plot()` z odpowiednimi parametrami oraz kilku funkcji niskopoziomowych utworzy dwa rysunki na jednej stronie. Na górnym rysunku ma znaleźć się wykres x_i w zależności od i dla danego wektora liczbowego \mathbf{x} , a na dolnym – wykres przyrostów, tj. $x_i - x_{i-1}$. Rysunki powinny mieć jedną, wspólną etykietowaną oś OX, tak jak na poniższych ilustracjach.

```
set.seed(123)
plotdiff(cumsum(rnorm(50))) # rysunek po lewej stronie
plotdiff(cumsum(rt(50,5)), col=4, lty=2) # rysunek po prawej stronie
```



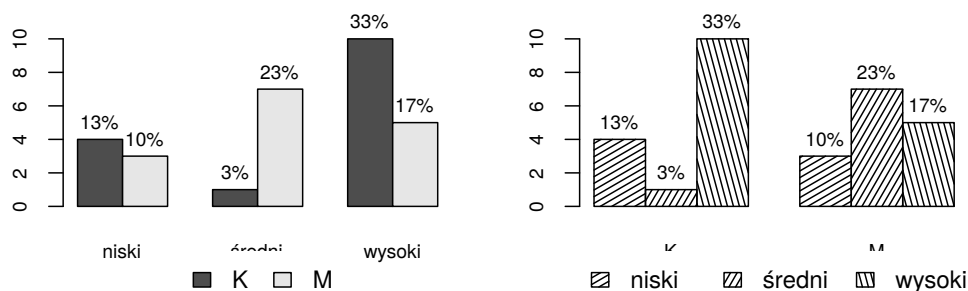
Zadanie 8.21 (AO; Wykres wielosłupkowy).

Napisz funkcję `mymultibarplot()`, która dla dwóch czynników g i h o tej samej długości skonstruuje wykres słupkowy zawierający liczności wystąpień obserwacji dla każdej kombinacji poziomów czynników. Argument `col` określa barwę wypełnienia słupków dla każdego poziomu czynnika g (domyślnie – poziomy szarości, zob. `?gray.colors`).

```
set.seed(1234)
g <- factor(c(rep('M', 15), rep('K', 15)))
h <- factor(sample(c('wysoki', 'średni', 'niski'), replace=TRUE, 30))
table(g, h)

##      h
## g   niski średni wysoki
## K     4      1     10
## M     3      7      5

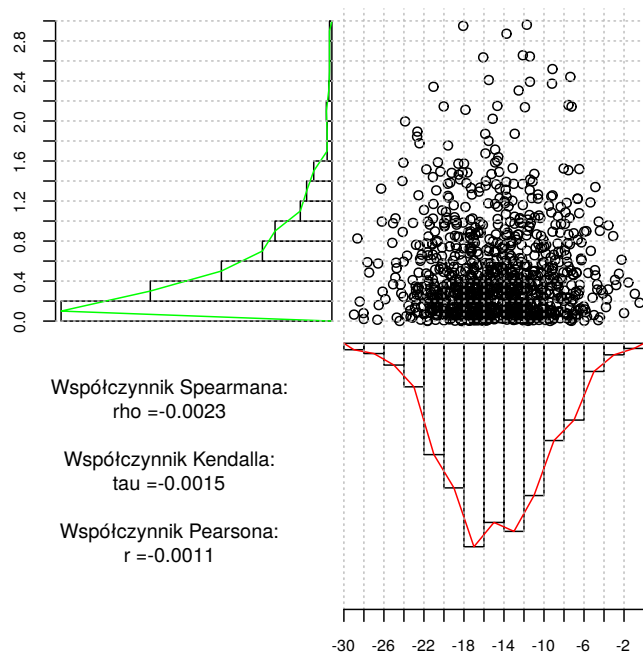
mymultibarplot(g, h) # lewy rysunek
mymultibarplot(h, g, density=25, col=1, angle=c(30, 60, 105)) # prawy
```



Zadanie 8.22 (AO; Dane dwuwymiarowe).

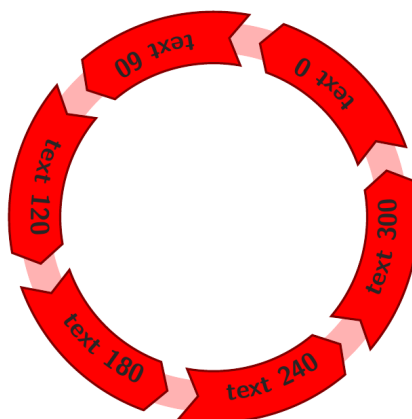
Napisz funkcję `multi2d()`, która dla dwóch wektorów liczbowych równej długości narysuje wykres rozrzutu, histogramy oraz wartości różnych współczynników korelacji (zob. `?cor`).

```
set.seed(321); x <- rnorm(1000, -15, 5); y <- rexp(1000, 2)
multi2d(x, y)
```



Zadanie 8.23 (MG; Obwarzanek).

Napisz funkcję `circularText()` która jako argument przyjmuje wektor napisów o długości $\in [2, 6]$, każdy napis składa się z maks. 10 znaków. Podane napisy narysuj na okręgu, uzyskując efekt podobny do poniższego. Zwróć uwagę na obwódkę okręgu oraz niby-strzałki wokół tekstu.



Zadanie 8.24 (MB; Otoczka wypukła).

Napisz funkcję `convexHull()`, która dla macierzy `points`, zawierającej współrzędne punktów w euklidesowej przestrzeni dwuwymiarowej, obliczy ich otoczkę wypukłą (por. `?chull`), a następnie ją narysuje na ekranie. Dokładniej, funkcja ma rysować wszystkie punkty zawarte w macierzy `points`, a także, odcinkami, otoczkę wypukłą. Niech funkcja przyjmie dodatkowo parametry `colPoints` i `colHull`, które oznaczają, odpowiednio, kolor punktów i otoczki wypukłej. Co więcej, niech funkcja zwraca współrzędne punktów (w postaci macierzy, takiej jak wejściowa), które składają się na otoczkę wypukłą, w tej kolejności, aby łącząc i -ty punkt z $(i + 1)$ -szym, a ostatni z pierwszym, otrzymać otoczkę wypukłą (użyj algorytmu Grahama, który jest bardzo dobrze opisany choćby tu: https://en.wikipedia.org/wiki/Graham_scan).

```
library(geometry)
set.seed(123)
points <- matrix(runif(100), ncol=2)
convexHull(points, colPoints=3, colHull=2)
```

