PEiAR Laboratorium 10 **06.05.2024**

Zad.1
Chcąc zbadać zależność między stężeniem spalin w dużych aglomeracjach a szeroko rozumianą niewydolnością płuc przeprowadzono badanie według którego
- przy niskim stężeniu spalin odnotowano 1004 przypadki choroby A, 832 przypadki choroby B i 545 przypadków choroby C;
- przy średnim stężeniu spalin: 880 przypadków choroby A, 732 przypadki choroby B i 904 przypadki choroby C;
- przy wysokim stężeniu spalin: 928 przypadków choroby A, 801 przypadków choroby B i 1014 przypadków choroby C.
Zweryfikuj hipotezę o niezależności wysokości stężenia analizowanych spalin od zachorowalności na choroby płuc.

Zad.2
Wygeneruj 4 próby z rozkładu Chi-kwadrat z *k* stopniami swobody:
1020 elementową dla k = 15;
1100 elementową dla k = 5;
1180 elementową dla k = 10;
1200 elementową dla k = 15.
Zweryfikuj hipotezę, że próby te pochodzą z populacji, w których badana cecha ma identyczny rozkład. Zilustruj porównanie w/w prób przy użyciu wykresu pudełkowego.

Zad.3 (Ggplot – c.d.**)**
Narysujemy w jednym układzie współrzędnych wykresy trzech funkcji gęstości prawdopodobieństwa rozkładu normalnego, wyposażone w legendę objaśniającą parametry rozkładu.
W tym celu, zbudujemy najpierw bazę dla wykresu ggplot z zakresem argumentów np. *[-10, 10]* i kolorami odpowiadającymi zadanym parametrom rozkładu normalnego w następujący sposób:
>base = ggplot(data.frame(x = c(-10, 10)), aes(x))
Komenda
> base = base + labs(colour = '...treść nagłówka legendy...')
pozwoli nadać legendzie pożądany nagłówek.
Na bazie zdefiniowanego konstruktora, generujemy wykres funkcji gęstości rozkładu normalnego standardowego z kolorem nadanym domyślnie (na razie) przez paczkę ggplot2:
> plot1 =
base + stat_function(fun = dnorm, geom = 'line', n = 100,
aes(colour = 'mean = 0, sd = 1') // treść objaśnienia przy pierwszym kolorze w legendzie //,
size = 0.8).
Na istniejący wykres naniesiemy kolejny - gęstość rozkładu normalnego ze średnią = 1 i odchyleniem standardowym = 2 (znacznie grubszą linią):
> plot2 =
plot1 + stat_function(fun = dnorm, args = list(mean = 1, sd = 2), // wartości parametrów rozkładu // *geom = 'line', n =100, aes(colour = 'mean = 1, sd = 2'),* // treść przy drugim kolorze // *size = 2)*
Następnie, naniesiemy wykres funkcji gęstości rozkładu normalnego ze średnią = 3 oraz odchyleniem standardowym = 0.6 (cieńszą linią, np. 0.5):
> plot3 =
plot2 + stat_function(fun = dnorm, args = list(mean = 3, sd = 0.6), // precyzujemy parametry...
// geom = 'line', aes(colour = 'mean = 3, sd = 0.6'), // objaśnienie dla 3 koloru // *size = 0.5)*
Dodajmy tytuł 'Gęstości rozkładów normalnych'
> plot4 = plot3 + ggtitle('Gęstości rozkładów normalnych')

Zmieńmy kolory kolejnych wykresów np. na: czarny, niebieski i pomarańczowy:
> plot5 = plot4 + scale_colour_manual(values = c('black', 'blue', 'orange'))
Co stanie się z legendą jeżeli dodamy do uzyskanego wykresu kod
> plot6 =
plot5 + theme(legend.position = c(0.2, 0.75), legend.background = element_rect(fill = 'grey'))
Poeksperymentuj z wartościami parametru *legend.position*. Dobierz je tak, aby legenda widoczna była mniej więcej w prawym górnym rogu obszaru ograniczonego osiami układu.

Zad.4 (Wykresy piramidowe**)**
Wygeneruj ramkę *frame_* z kolumnami:
'Miasto' o wartościach: 'Wrocław','Poznań','Szczecin','Gdańsk','Warszawa','Kraków';
'Kobiety' o wartościach: 350, 484, 196, 320, 560, 731 oraz
'Mężczyźni' o wartościach: 370, 492, 174, 350, 530, 710.
Ramkę *frame_* poszerz o kolumnę *'Razem'* - użyj w tym celu funkcji *apply(...)*.
Zainstaluj paczkę *'plotrix'* i załaduj ją. Zechcemy narysować tzw. **wykres piramidowy** służący najczęściej do prezentacji danych demograficznych.
Załącz ramkę do bieżącej sesji w celu łatwiejszego dostępu do jej kolumn.
Wpisz polecenie:
> pyramid.plot(Kobiety, Mężczyźni, labels = Miasto, gap = 15)
Ukazały się domyślne nagłówki, które trzeba teraz właściwie ustawić. W tym celu dodaj w definicji wykresu piramidowego argument: *top.labels = c('Kobiety', 'Miasto', 'Mężczyźni').*
Czym steruje atrybut *gap*?
Dodaj kolejny argument: *main = 'Wykres piramidowy'* aby nadać wykresowi tytuł.
Aby oddzielić od siebie poszczególne paski na wykresie, użyj kolejnego argumentu w funkcji *pyramid.plot*: *space = 0.5 // na przykład //* → poeksperymentuj.
Aby ustawić żądaną podziałkę na osi x możemy użyć w definicji kolejnych atrybutów:
laxlab = c(0, 400, 800) // na lewo // oraz
raxlab = c(0, 400, 800) // na prawo //.
Dodatkowo, jeśli chcemy, by przy paskach widoczne były ich precyzyjne wartości liczbowe, możemy dodać argument: *show.values = TRUE*.

Zad.5 (Nadal wykresy…)
W wykresie piramidowym z Zad. 2, zwiększ odstęp pomiędzy skrzydłami wykresu (parametr *gap*), usuń widoczną u dołu jednostkę miary (parametr *unit*), zwiększ czcionkę etykiet (parametr *labelcex*) oraz zmień dokładność, z jaką widoczne są dane liczbowe przy paskach (parametr *ndig*).

Zad.6
Wygeneruj ramkę *frame_* z kolumnami *Planeta, Masa i Średnica* o wartościach odpowiednio:
Neptun, Uran, Saturn, Jowisz, Mars, Ziemia, Wenus, Merkury;
17.15, 14.54, 95.16, 317.80, 0.11, 1.00, 0.81, 0.06 oraz
3.88, 4.00, 9.45, 11.20, 0.53, 1.00, 0.94, 0.38 (są to dane relatywne w odniesieniu do Ziemi).
Wygeneruj wykres piramidowy zestawiający masę i średnicę dla poszczególnych planet
> attach(frame_)
> pyramid.plot(..., ..., labels = …)
Ustaw odpowiednie nagłówki oraz podziałkę na lewej osi: 0, 100, ..., 400 natomiast na prawej osi: 0, 10, ... 50. Zadbaj o detale wykresu.

Zad.7 (Błądzenie losowe na płaszczyźnie…)
Błądzenie losowe cząstki po płaszczyźnie jest pewnym stochastycznym (czyli losowym) ruchem partykuły startującej z punktu (0, 0) i idącej w jednym kroku: jednostkę w lewo lub prawo (z jednakowym pr-stwem) **i** - niezależnie od tego - jednostkę w górę lub dół (również z jednakowym pr-stwem).

Zatem z pr-stwem ¼ po jednym kroku, cząsteczka może się znaleźć w jednym z czterech punktów: (1, 1), (-1, 1), (-1, -1) lub (1, -1). Zilustrujemy to.
Chcemy najpierw aby wykres był w kwadratowej ramce (a nie prostokątnej zajmującej całą szerokość widoku) dlatego
> *par(pty = 's')* // typ wykresu (*plot type*) ustawiony na *square* (zamiast na '*m*', czyli *max*)//
Następnie używamy zwykłego polecenia *plot(…),* które jako swoje dwa pierwsze (podstawowe) argumenty przyjmuje: wektor pierwszych współrzędnych i wektor drugich współrzędnych punktów które chcemy narysować.
Ustaw:
- typ wykresu na punktowy (atrybut *type*)
- zakres obu osi na [-2, 2]
- etykiety osi: 'x coordinate' oraz 'y coordinate'
- wielkość wartości liczbowych podpisujących obie osie na 0.75 (atrybut *cex.axis*)
- wielkość punktów (kropek) wykresu na 0.9 (atrybut *cex*)
- kształt punktów (kropek) wykresu na dowolną wartość całkowitą między -1 a 25 (atrybut *pch*).

Wypoziomuj ponadto czcionkę podpisującą oś pionową.
Teraz, chcemy dodać strzałki wskazujące z początku układu współrzędnych, do jakiego punktu kolejno możemy dotrzeć. Zacznijmy od punku (1, 1).
Ustaw:
- grot strzałki sięgający punktu (0.9, 0.9)
- długość grota na wartość 0.15
- kąt prosty ramionom strzałki

Zilustruj strzałki wskazujące na pozostałe trzy punkty. Dalej, umieść napisy informujące, z jakim pr-stwem docieramy do każdego ze wskazanych punktów (np. 'prob = 0.25').
Na końcu, nanieś na wykres przerywane linie w miejscu osi pionowej i poziomej używając w tym celu polecenia *abline(...)* z atrybutem *lty* ustawionym na wartość 2.