




3. Eksploracyjna analiza danych

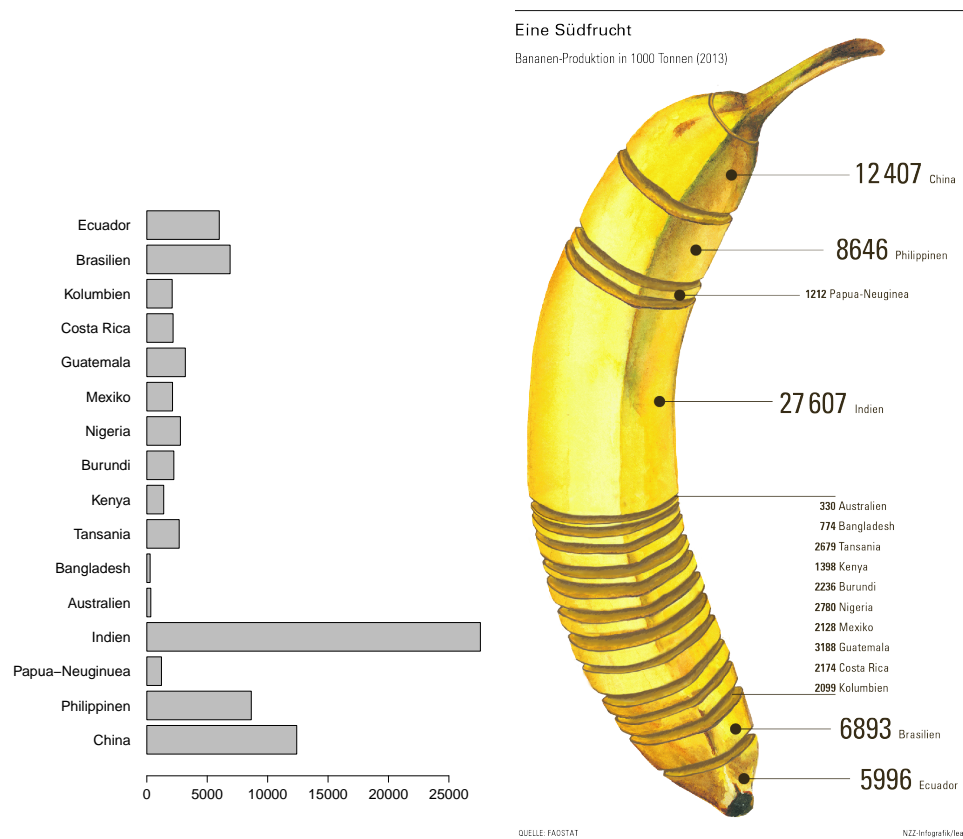
3.1 Czym jest „Eksploracyjna analiza danych”?

Eksploracyjna analiza danych (EAD, ang. *exploratory data analysis*) to proces mający na celu zrozumienie charakterystyki danych przy użyciu technik wizualizacji oraz metod statystyki opisowej. Każdy badacz danych, gdy zaczyna pracę z nowym zbiorem danych powinien spędzić trochę czasu na tego typu analizie. Polega ona na interaktywnej pracy z danymi której celem jest zaproponowanie możliwych hipotez do późniejszej weryfikacji, zdobycie intuicji dotyczący danych w stopniu umożliwiającym przyjęcie bądź odrzucenie założeń np. o rozkładzie danych. Tego typu wiedza pozwoli na wybranie właściwych metod dalszej analizy statystycznej, w szczególności metod wnioskowania statystycznego, które będziesz poznawał(a) na kolejnych laboratoriach.

3.2 Graficzna analiza danych

Rozróżniamy dwa typy wizualizacji danych: wizualizacje eksploracyjne i wizualizacje wyjaśniające. Te pierwsze wchodzą w skład EAD i są to grafiki robione zwykle standardowymi, prostymi i typowymi metodami bez przesadnej dbałości o aspekty estetyczne. Głównym celem takich grafik jest spojrzenie na dane z różnych kątów bez żadnego ukierunkowania czy tendencyjności oraz połączenie w wizualizacji różnych czynników, aby zobaczyć w danych pewne wzorce itd. Zwykle więc generuje się bardzo dużo takich rysunków pracując z danymi w sposób interaktywny (np. w pakiecie ) , sprawdzając kolejne hipotezy i zależności. Podczas tego procesu można wyrobić sobie bardzo dobre intuicje i zrozumieć co znajduje się w danych. W szczególności możemy dojść do odkrycia pewnej zaskakującej zależności i przejść do wizualizacji wyjaśniającej.

Wizualizacja wyjaśniająca to starannie stworzona wizualizacja danych, mająca za zadanie wyjaśnienie znalezionej zależności/fenomeny w sposób jak najprostszy i najbardziej oczywisty dla szerokiej publiczności. W takim procesie trzeba zadbać o odpowiednie dobranie



Rysunek 3.1: Przykład wizualizacji eksploracyjnej (lewa) i wyjaśniającej (prawa) na tych samych danych: udział poszczególnych krajów w produkcji bananów

kolorów, rozmiarów czcionek, pozycji na stronie itd. Należy także rozważyć zastosowanie zaawansowanych efektów graficznych czy elementów interaktywnych. Głównym celem takiej wizualizacji jest zakomunikowanie danych odbiorcom, a więc kluczowe jest zrozumienie publiczności (kim są?, co potrzebują wiedzieć?, co wiedzą o danych?).

Ciekawym przykładem informatycznej wizualizacji jest projekt Codeology¹ wizualizujący kod źródłowy użytkowników serwisu GitHub. Innym przykładem jest interaktywna wizualizacja zmian średniej długości życia ludzi w czasie² czy też wizualizacja analizująca różne czynniki globalnego ocieplenia³. Jednakże podczas tego laboratorium skupimy się na wizualizacjach eksploracyjnych.

Problem 3.1 Czym różni się wizualizacja eksploracyjna od wyjaśniającej?

3.2.1 Dlaczego wizualizować dane?

Wizualizacja danych to zamienianie danych w obraz, który jest bardziej zrozumiały. Możemy zobaczyć pewne zależności dużo łatwiej i szybciej niż analizować tabelki, co wynika z faktu, że nasz zmysł wzroku jest w stanie dużo szybciej przetworzyć obraz niż np. czytać. Dobra wizualizacja pozwala „odkryć historię którą opowiadają dane” – pozwala nam na zrozumienie co jest w danych czy też zaobserwować niepokojące artefakty wskazujące na problemy w procesach czyszczenia lub zbierania danych.

Słynnym przykładem motywującym graficzną analizę danych jest tzw. kwartet Anscombe’a. Jest to zbiór czterech prostych, dwuwymiarowych zbiorów danych, które mają taką samą⁴ średnią i wariancję każdej z kolumn, a dodatkowo takie same wartości innych, bardziej zaawansowanych, statystyk opisowych⁵. Jednak po zwizualizowaniu tych zbiorów można zaobserwować znaczące różnice (patrz rys. 3.2).

Innym przykładem motywującym badanie rozkładu wartości jednej zmiennej przy użyciu analizy graficznej jest przykład podany w książce [9]. Na rysunku 3.3 pokazane są trzy mocno różniące się od siebie rozkłady danych, które mają tę samą średnią i odchylenie standardowe.

3.2.2 Podstawowe rodzaje wykresów

- wykres słupkowy – służy do prezentacji wielkości ilościowej w podziale na grupy czyli jej zależności od pewnej zmiennej jakościowej.
- histogram kolumnowy czyli w zasadzie wykres słupkowy stworzony na przedziałowym szeregu rozdzielczym w celu wizualizacji rozkładu wielkości ilościowej. W odróżnieniu od wykresu słupkowego wykres ten nie ma odstępów pomiędzy kolejnymi słupkami, aby osiągnąć wrażenie ciągłości rozkładu.
- wykres typu łodyga i liście – stosowany do wizualizacji rozkładu wielkości ilościowej przy małej liczbie obserwacji (do 200). Aby go narysować, opuszcza się jedną lub dwie cyfry w zapisie dziesiętnym, a następnie sortuje się je rosnąco. Użyte liczby zapisuje się w jednej kolumnie (łodygi), a drugą kolumnę uzupełniamy posortowanymi końcówkami dla każdego zaokrąglenia (liście). Na przykład dla

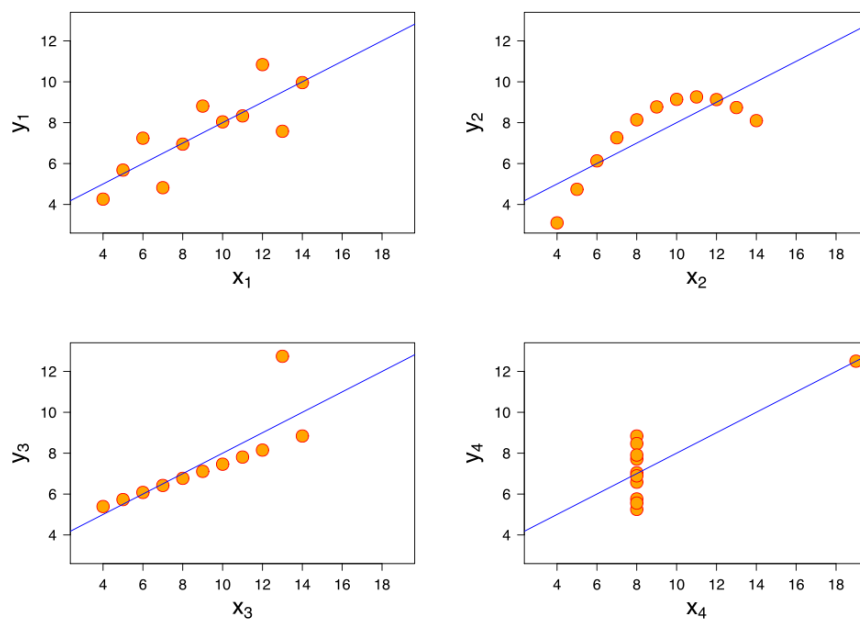
¹<http://codeology.braintreepayments.com/>

²<https://www.youtube.com/watch?v=jbkSRLYSojo>

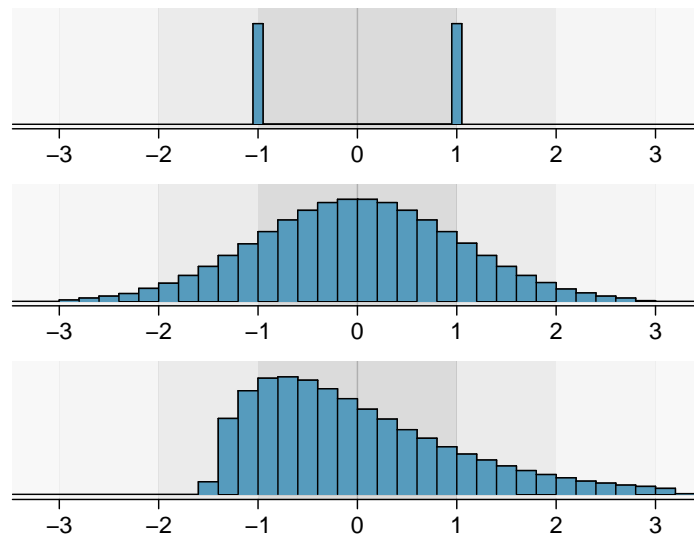
³<http://www.bloomberg.com/graphics/2015-whats-warming-the-world/>

⁴z dokładnością do 2 miejsc po przecinku

⁵Dane te mają taki sam współczynnik korelacji czy też równanie regresji liniowej („najlepszej” linii prostej opisującej zależność między zmiennymi) – wskaźniki te poznasz na kolejnych laboratoriach.



Rysunek 3.2: Kwartet Anscombe'a: Wszystkie cztery zestawy danych wydają się być identyczne, jeżeli weźmiemy pod uwagę ich charakterystykę statystyczną, ale znacznie różnią się od siebie w ujęciu graficznym [1].



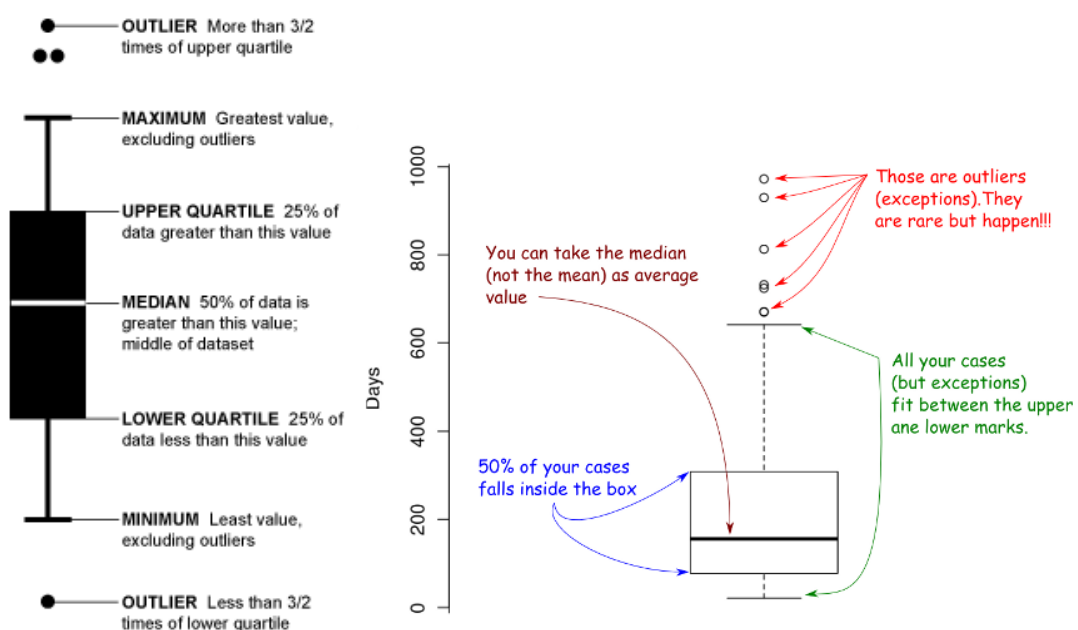
Rysunek 3.3: Trzy różne rozkłady wartości próby z tą samą średnią $\mu = 0$ i odchyleniem standardowym $\sigma = 1$. [9]

takich danych: 10, 15, 22, 25, 29, 33, 36, 36, 37, 38, 38, 47 taki wykres wyglądałby następująco:

10	05
20	259
30	366788
40	7

Jest stosowany np. na rozkładach jazdy komunikacji miejskiej w Poznaniu⁶[4].

- wykres kołowy i pierścieniowy – służy do wizualizacji procentowego podziału jakiejś całości (np. społeczeństwa) na części (np. grupy społeczne).
- wykres rozrzutu (punktowy) i liniowy – służący do pokazania zależności pomiędzy dwiema zmiennymi ilościowymi. Wykres rozrzutu stosujemy gdy chcemy pokazać zależności pomiędzy zmiennymi, z kolei wykres liniowy służy do pokazania linii trendu.
- wykres pudełkowy – najmniej znany, a bardzo przydatny. Wykres prezentujący rozkład wielkości ilościowej (czasami w podziale na grupy) poprzez narysowanie pudełka zawierającego połowę wartości zmiennej (pomiędzy 1 a 3 kwartylem). Medianę zaznaczamy poprzez narysowanie poziomej kreski w środku tego pudełka. Dodatkowo do pudełka dorysowujemy „wąsy”, które mają obejmować wszystkie wartości od minimum do maksimum. Jednak jest to najprostsza wersja: zwykle nie dopuszcza się, aby żaden z wąsów był dłuższy niż półtorej rozstępu międzykwartylowego. Obserwacje, które nie mieszczą się w przedziale oznaczonym wąsami wizualizuje się poprzez dodanie punktu nad/pod pudełkiem na odpowiedniej wysokości. Ilustracje tego wykresu wraz z wyjaśnieniem można znaleźć na rysunku 3.4, a także w książce [6].



Rysunek 3.4: Jak czytać wykres pudełkowy?

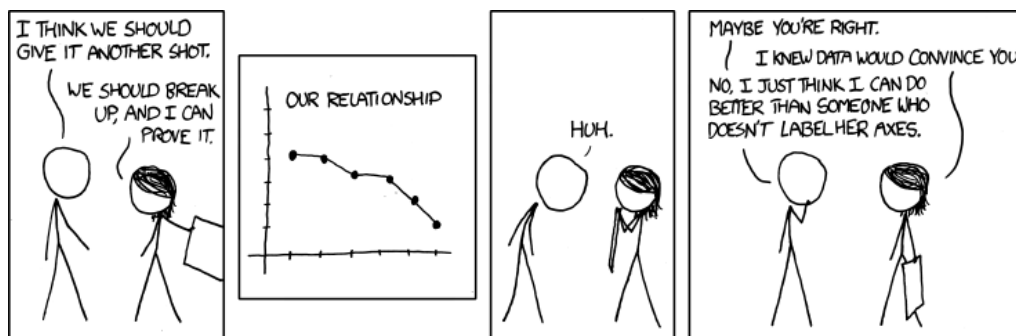
Problem 3.2 Wyjaśnij jak jest różnica pomiędzy histogramem a wykresem słupkowym.

3.2.3 Zasady dobrej wizualizacji

1. Przede wszystkim należy zrozumieć, że wykresy to porównania. Nawet oglądając wykres z jedną linią tak naprawdę niejawnie porównujemy ją z osiami wykresu. Pamiętając o tym należy dobrać dane i wizualizowane zmienne w taki sposób, aby prezentowane porównanie było jak najlepiej widoczne.

⁶<http://www.statystyczny.pl/lodyga-i-liscie-czy-naprawde-istnieje-taki-wykres/>

2. Wizualizuj odpowiednią ilość informacji (nie za dużo, nie za mało).
3. Zastosuj odpowiedni typ wykresu do twoich danych. Aby dobrze wybrać typ wykresu możesz użyć następującego schematu: <http://img.labnol.org/di/data-chart-type.png>
4. Podpisuj osie na wykresie oraz prawidłowo dobierz skale.



Rysunek 3.5: Warto podpisywać osie na wykresach...

5. Unikaj niepotrzebnych ozdóbek, a w szczególności nie wprowadzaj graficznego rozróżnienia na wykresie, które nie odzwierciedla rzeczywistych różnic w danych.
6. Dobierz kolory wykresu tak aby łatwo można było rozróżnić poszczególne serie danych – w przeciwnym wypadku sugerujesz związek, który nie istnieje w danych.
7. Jeżeli serie danych są ze sobą pogrupowane staraj się aby serie będące w jednej grupie miały jakiś element wspólny: podobne kolory, styl linii czy symbol.

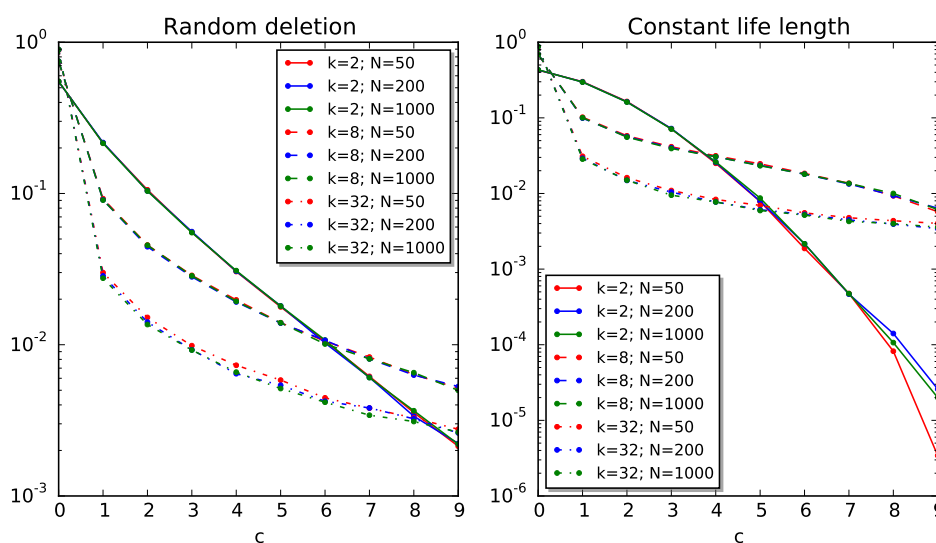
Na rysunku 3.6 możesz znaleźć przykład dość dobrej wizualizacji eksploracyjnej pokazującej działanie algorytmu genetycznego przy różnych ustawieniach parametrów: wielkość turnieju k i wielkość populacji N (dla naszych ćwiczeń ich znaczenie nie jest istotne). Zwróć uwagę w jaki sposób zostały dobrane kolory i rodzaje linii, a także jaka skala została użyta. Pomimo tego, że jest to wykres adresowany do wąskiej publiki (jest to wykres z pracy naukowej, dla szerokiej grupy odbiorców na wykresie jest np. zbyt dużo serii danych), po zapoznaniu się z legendą można bez problemu porównywać działanie algorytmu przy różnych ustawieniach parametrów np. coraz bardziej przerywana linia wskazuje na rosnące k .

Problem 3.3 Spójrz na wykresy przedstawione na rysunku 3.1. Czy są to przykłady dobrych wizualizacji danych? Co można zrobić aby je ulepszyć?

3.2.4 Najczęstsze błędy przy graficznej analizie danych

1. Użycie zbyt wielu ozdóbek, które zaciemniają odbiór. Przykładem jest tutaj wykres pokazujący wyniki wyborów parlamentarnych w 2015 roku przedstawiony na rysunku 3.7, który został szeroko skomentowany na portalu statystyczny.pl [5]:

Dlaczego 9,02% Kukiza wygląda jak ponad połowa 38,54% PiSu? (...) Czy widzicie tę jasną kresczkę idącą od lewej do prawej? To miejsce to „zero” na osi. Dopiero nad tym liczymy procenty danej partii. A kwadrat na dole po co? Żeby wszystko ładniej wyglądało. Żeby było gdzie wpisać nazwę partii. Żeby pokazać, że można fajnie graficznie pokombinować. A że większość ludzi tego nie zakuma. To przecież żaden problem. I tak nie rozumieją statystyki i wykresów więc o co chodzi...



Rysunek 3.6: Przykład dobrej wizualizacji eksploracyjnej [autor: mgr inż. K. Miazga]

2. Nadmierne używanie wykresu kołowego lub pierścieniowego.

Wykres kołowy w ostatnim czasie zrobił się bardzo kontrowersyjny: z jednej strony ma bardzo dużą popularność np. w prasie, z drugiej strony jest głośno krytykowany przez środowiska statystyczne [9, 10]. Nawet w pomocy pakietu `R` do funkcji `pie()` rysującej wykres kołowy znajduje się informacja, że „Pie charts are a very bad way of displaying information”, a John Turkey (statystyk, ale także twórca słowa „bit” – podstawowej jednostki pamięci w informatyce) stwierdził „There is no data that can be displayed in a pie chart, that cannot be displayed better in some other type of chart.”

Badania psychologiczne pokazały, że oko ludzkie jest dobre w ocenie miar liniowych (takich jak wysokość) czy z porównywaniem położenia w dwóch wymiarach. Natomiast nie radzi sobie z porównywaniem pola powierzchni, które jest nośnikiem informacji na wykresie kołowym. Dodatkowo eksperymenty pokazały, że np. przekręcenie wykresu kołowego wpływa na percepcyjny odbiór wielkości prezentowanej miary, chyba że prezentowane miary są wielokrotnościami 25%.

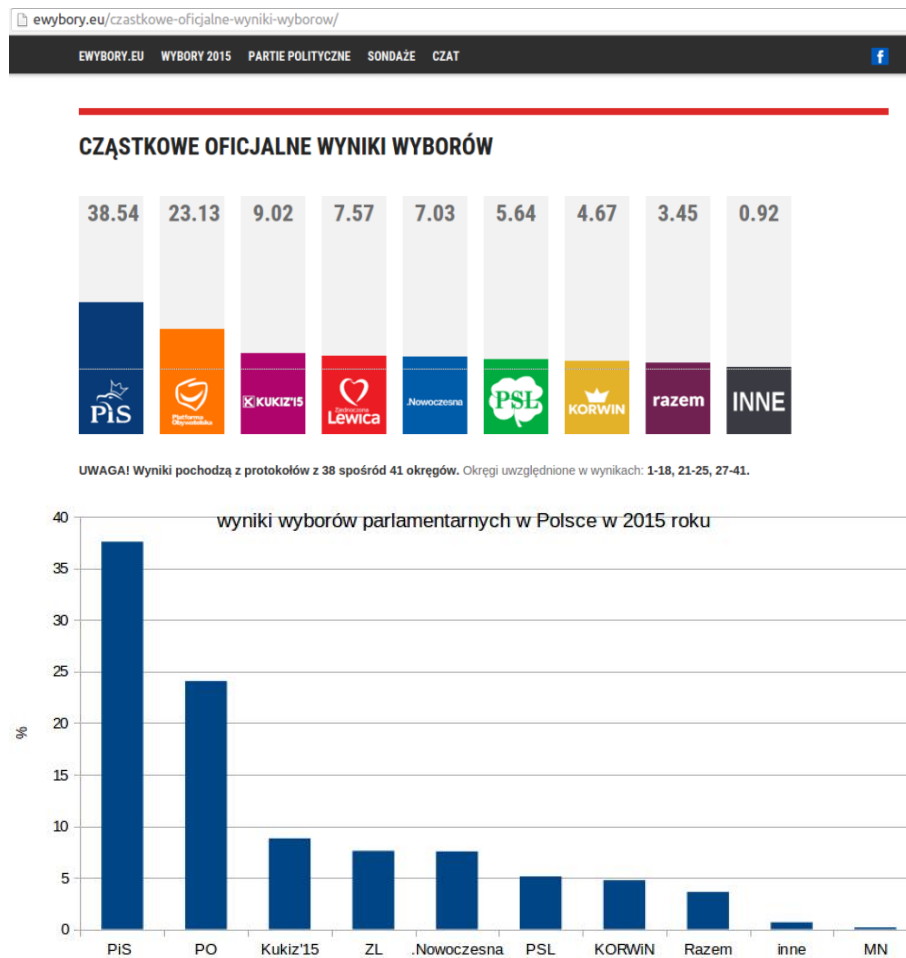
Dobrym przykładem pokazującym problemy z interpretowaniem wykresu kołowego przedstawia rysunek 3.8. Patrząc na wykres kołowy nie jesteśmy w stanie ocenić jaki procent danych należy do danej kategorii. Oczywiście rozwiązaniem jest podpisanie wszystkich kawałków wykresu wartościami, ale skoro i tak zamierzamyazać odbiorcy przeczytać wszystkie wartości – jaki jest sens wizualizacji?

3. Używanie pseudo-trójwymiarowych wykresów.

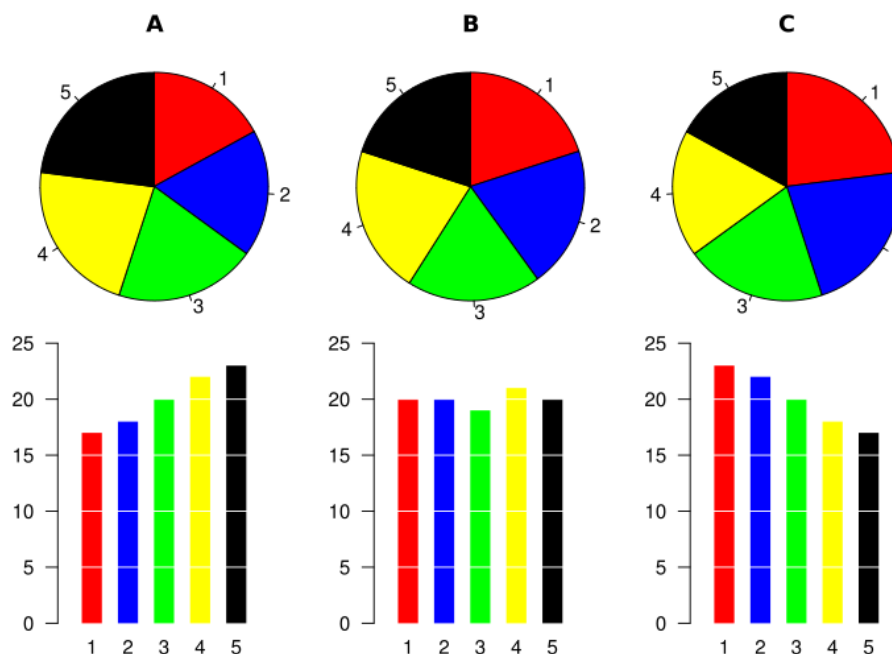
Wykresy trójwymiarowe, jak sama nazwa wskazuje służą do wizualizacji danych w trzech wymiarach. Niestety dużo osób zaczęło używać ich do wizualizacji standardowych, dwuwymiarowych danych. Użycie trzech wymiarów w takich przypadkach jedynie utrudnia porównywanie wysokości oraz odczyt wartości z osi.

4. Używanie wykresów słupkowych do prezentowania podsumowań (np. średniej).

Często spotyka się wykres słupkowy który wizualizuje dwie lub trzy... liczby. Dla przykładu chcemy podać średnią wysokość kobiet i średnią wysokość mężczyzn –



Rysunek 3.7: Estetyczny wykres słupkowy (góra) i zwykły wykres słupkowy (dół) przedstawiający cząstkowe wyniki wyborów parlamentarnych w 2015 roku [5].



Rysunek 3.8: Bezżyteczność wykresu kołowego przy wizualizacji trudniejszych rozkładów [2].

zamiast po prostu podać dwie liczby, konstruuje się dla nich cały wykres słupkowy, który tylko zabiera miejsce i jest mało interesujący (bo niesie ze sobą bardzo mało informacji). Jeżeli chcemy zwizualizować takie dane, powinniśmy użyć wykresu pudełkowego, który dodatkowo pokaże nam obserwacje odstające, kwartyle itd.


5. Zły dobór skali wykresu np. porównywanie wartości z bardzo dużym rozstępem bez użycia skali logarytmicznej.
6. Użycie wykresu słupkowego do wizualizacji sparowanych serii danych np. ciśnienie krwi przed i po zażyciu lekarstwa. Taka wizualizacja poprzez przemieszanie słupków dotyczących różnych pomiarów utrudnia dokonywanie porównań. Takie dane można wizualizować np. na wykresie punktowym, gdzie na osi X mamy wartości „przed”, a na osi Y wartości „po”. Dodatkowo na takim wykresie możemy zaznaczyć linię $y = x$, która pokazuje referencyjne wartości, gdyby nie było żadnej różnicy pomiędzy wartościami zmiennych.
7. Wizualizacja zbyt dużej ilości danych/informacji.


Więcej o zasadach dobrej grafiki informacyjnej oraz o badaniach psychologicznych nad percepcją w kontekście wizualizacji możesz znaleźć w [11].



3.3 Podstawy języka R




3.3.1 Oprogramowanie statystyczne

Istnieje wiele dobrych pakietów statystycznych, oferujących zarówno podstawowe jak i zaawansowane metody analizy danych. Wśród nich pakiet Statistica (darmowy dla studentów PP poprzez platformę eProgramy), SPSS czy Stata.

W ostatnich latach dużą popularność w środowiskach analityków danych zyskał język/pakiet statystyczny , który w 2015 roku stał się 6 najbardziej popularnym językiem


programowania na świecie, wyprzedzając PHP, JavaScript czy Ruby. Z kolei już w ankiecie branżowego portalu KDnuggets, badacze danych zapytani o oprogramowanie używane przez nich w ostatnim roku najczęściej wskazywali na  (46.9 %). Na drugi w rankingu RapidMiner wskazało 31,5% analityków, czyli o ponad 15% mniej.

Czemu  zawdzięcza tak dużą popularność? Niestety nie tym, że można się go szybko i łatwo nauczyć⁷. Dziwne maniere tego języka takie jak tablice indeksowane od 1 (a nie od 0) czy dziwny znak przypisania wartości do zmiennych długo daje się we znaki początkującemu programiście R. Natomiast jest to język o wolnych źródłach, który każdy może pobrać ze strony projektu r-project.org i uruchomić na swoim ulubionym systemie operacyjnym (do wyboru Windows, OS X i Linux). Dodatkowo  ma bardzo wiele pakietów w repozytorium CRAN, a więc praktycznie każda metoda analizy danych została już w nim zaimplementowana – wystarczy więc ściągnąć paczkę i używać bez żmudnej implementacji. Powoduje to, że badacz danych nie jest ograniczony poprzez wąski zestaw metod zdefiniowany w umowie licencyjnej pakietu komercyjnego. Język ten staje się też powoli standardem w publikowaniu nowych osiągnięć w analizie danych – bardzo często do artykułów naukowych dołączany jest gotowy pakiet, aby każdy statystyk na świecie mógł zweryfikować opublikowane wyniki.

Z powyższych powodów zajęcia laboratoryjne z zaawansowanych technik analizy danych (analiza regresji, testy nieparametryczne) będą przeprowadzone w . Pozostałe laboratoria, ze względu na możliwość lepszego ilustrowania przeprowadzanych przez studenta analiz, będą bazowały na arkuszach kalkulacyjnych. Studentów szczególnie zainteresowanym nauką języka  polecam stronę datacamp.com gdzie poprzez oglądanie krótkich tutoriali oraz wykonywanie różnorodnych ćwiczeń można nauczyć się nie tylko programowania w , ale także zdobyć podstawowe doświadczenie w analizie i obróbce danych.

3.3.2 Budowa szeregu rozdzielczego w R

Po uruchomieniu RStudio masz dostęp do konsoli języka R, obszaru roboczego pokazującego zmienne przechowywane w pamięci operacyjnej, historię wywoływanych komend, możliwość oglądania stworzonych wcześniej wykresów i wiele, wiele innych.

Aby poznać możliwości graficzne języka  oraz aby przetestować jak działa środowisko robocze możesz wpisać w konsolę (lewe dolne okienko w RStudio) polecenie `demo(graphics)` a następnie wciskać Enter. Za każdym razem powinien się pojawić nowy, demonstracyjny wykres.

Ćwiczenie 3.1 Konsola języka R może służyć jako kalkulator. Wypróbuj podstawowe operacje arytmetyczne: dodawanie, odejmowanie, dzielenie, mnożenie, potęgowanie (\wedge), operator modulo ($\%\%$). Sprawdź też czy dzieląc dwie liczby całkowite wynik również jest całkowity (np. w języku Python $5/2 = 2$).

⁷ „R will always be arcane to those who do not make a serious effort to learn it. It is ****not**** meant to be intuitive and easy for casual users to just plunge into. It is far too complex and powerful for that. But the rewards are great for serious data analysts who put in the effort.”, Berton Gunter

 Szeregi rozdzielcze można łatwo skonstruować w pakiecie R. W tym celu najpierw wprowadźmy do systemu nasze dane. Możemy to zrobić np. poprzez ręczne utworzenie wektora z danymi wydając polecenie:

```
dane <- c(1123, 198, 473, 784, 305, 423, 397, 298, 698, 237)
```

Operator `<-` oznacza przypisanie^a, a funkcja `c()` jest konstruktorem wektora. Wektor w R nie konieczne musi zawierać liczby – może to być dowolny typ danych np. ciąg znaków, jednakże dane w całym wektorze zawsze muszą być jednakowego typu.

Aby utworzyć szereg rozdzielczy musimy podzielić nasze dane na przedziały. Możemy zrobić to za pomocą funkcji `cut(dane, breaks=liczba_przedzialow)` w której musimy ręcznie wyspecyfikować liczbę przedziałów poprzez podanie argumentu `breaks=`. Załóżmy, że dla naszego krótkiego wektora danych wystarczą 3 przedziały i wywołajmy polecenie:

```
dane_w_przedzialach <- cut(dane, breaks = 3)
```

Utworzona właśnie zmienna `dane_w_przedzialach` nie jest już wektorem, ale zmienna typu `factor`. Stało się tak dlatego, że funkcja `cut()` po określeniu zakresu przedziałów automatycznie przekonwertowała wszystkie wartości wektora `dane` na nazwy odpowiadających im przedziałów. Dane zmieniły więc swój typ i są teraz typu jakościowego. Do przechowywania tego typu danych wykorzystywany jest właśnie typ `factor`^b (choć oczywiście, pomijając względy efektywności, mogłyby być one przechowane jako wektor ciągów znaków). Nie wierz temu co jest tu napisane na słowo – sprawdź jak wygląda zawartość zmiennej `dane_w_przedzialach` przez wpisanie jej nazwy do konsoli i wciśnięcie Enter.

Mając tak przygotowane dane aby dokończyć ćwiczenie wystarczy wyświetlić tabelkę ze zliczeniem wartości poszczególnych przedziałów – możemy to uzyskać funkcją `table()`.

```
table(dane_w_przedzialach)
```

Gratulacje! W ten sposób utworzyłeś swój pierwszy szereg rozdzielczy w R. Przypomnijmy: wymagało to obliczenia zakresu przedziałów i zastąpienia wartości liczbowych wektora odpowiadającymi im przedziałami (funkcja `cut()`), a następnie zliczenia nazw poszczególnych przedziałów i wyświetlenia tego w formie tabeli (funkcja `table()`). Przy okazji dowiedzieliśmy się jak utworzyć wektor z danymi oraz poznaliśmy typ `factor`.

^aZwróć uwagę, że postawienie spacji pomiędzy znakami `<-` zmienia operator przypisania na dwa operatory „mniejszy niż” oraz „minus”. Warto też wiedzieć, że w niektórych sytuacjach jest dozwolone korzystanie z „normalnego” operatora przypisania czyli znaku równości.


^bDla zainteresowanych: wewnętrznie `factor` jest reprezentowany jako wektor kolejnych liczb całkowitych (identyfikatorów) wraz ze słownikiem mapującym te identyfikatory na nazwy. Słownik ten można odczytać wydając polecenie `levels(dane_w_przedzialach)`. Pewną magią języka R jest to, że do tej funkcji możemy także przypisać wartości np. `levels(dane_w_przedzialach)<-c('a','b','c')` spowoduje zasępienie nazw przedziałów kolejnymi literami alfabetu.

Ćwiczenie 3.2 Podczas wykonywania tutorial'a poznałeś typ `factor` służący do przechowywania wartości nominalnych. Możesz przekonwertować wektor tekstów do wektora typu `factor` poprzez wywołanie `factor(nazwa_wektora)`.

Stwórz w R wektor o wartościach `["Mężczyzna", "Kobieta", "Kobieta", "Mężczyzna", "Mężczyzna"]` i przekonwertuj do typu `factor`. Spróbuj porównać (np. znakiem większości) pierwszy i drugi element nowo utworzonego wektora. ■

Ćwiczenie 3.3 Typ wektorowy pozwala na przechowywanie danych tylko jednego typu. Co się stanie jeśli spróbujesz utworzyć wektor zawierający np. liczby i ciągi znaków? Sprawdź swoją hipotezę w R. ■

3.3.3 Operacje na wektorach i indeksowanie

 Język R, podobnie jak poznany na przedmiocie „Narzędzia informatyki” MATLAB/Octave, operuje na wektorach i macierzach. Pracując w R możesz więc wykonywać operacje na całych wektorach (tak samo jak w MATLAB).

Dla szybkiego przypomnienia, przetestuj w R następujące komendy:

- pomnożenie wszystkich danych w wektorze przez 2

```
dane * 2
```

- pokazanie 5 elementu wektora

```
dane[5]
```

- pokazanie 1 i 5 elementu wektora

```
dane[c(1, 5)]
```

- utworzenie wektora wartości logicznych z informacją o tym czy liczba na danej pozycji jest większa od 500

```
dane > 500
```

- wypisanie elementów wektora większych od 500 (indeksowanie wektorem wartości logicznych).

```
dane[dane > 500]
```

Ćwiczenie 3.4 Spróbuj wyciągnąć z wektora `dane` element o niecałkowitym indeksie np. 2.5. Co się stało? ■

Ćwiczenie 3.5 Wypisz z wektora `dane` wszystkie liczby parzyste. ■


Ćwiczenie 3.6 Oprócz funkcji `c()` możemy stworzyć wektor poprzez wywołanie funkcji `seq(od, do, co_ile)` np. `seq(1,10,1)` wypisze wszystkie liczby od 1 do 10^a.

- Wygeneruj wszystkie liczby od 1 do 20
- Wygeneruj wszystkie liczby parzyste mniejsze od 50
- Wygeneruj wszystkie wielokrotności liczby 3 mniejsze od 100, a następnie zlicz ile z nich kończy się daną cyfrą.

^aW przypadku generowania kolejnych cyfr można posłużyć się konstrukcją `od:do` np. `1:10`

Ćwiczenie 3.7 Przekonwertuj podane temperatury w Fahrenheitach na stopnie Celsjusza `fahrenheit<-c(32, 59, 86)`. Wzór na konwersję masz podany poniżej.

$$c = \frac{(f - 32) \cdot 5}{9}$$

Ćwiczenie 3.8 Jeżeli `x<-c(5,9,2,3,4,6,7,0,8,12,2,9)` to co zwróćą poniższe komendy? Odpowiedz a następnie sprawdź swoje hipotezy w .

- `x[2]`
- `x[2:4]`
- `x[c(2,3,6)]`
- `x[c(1:5,10:12)]`
- `x[-(10:12)]`

3.4 Eksploracyjna analiza danych w R

Jednym z popularnych zastosowań analizy danych w ostatnich latach jest analiza wydźwięku (ang. *sentiment analysis*). Firmy takie jak np. Samsung czy Apple chcąc dowiedzieć się co (nie) podoba się użytkownikom w ich produktach analizują wpisy pojawiające się na portalach takich jak Twitter czy Facebook. Na każdym takim wpisie („tweece”) dotyczącym np. nowego modelu smartfona jest przeprowadzana analiza statystyczna, która przydziela mu liczbę z pewnego zakresu np. od 1 do 10. Tę liczbę będziemy nazywać współczynnikiem wydźwięku. Wysokie wartości tego współczynnika, bliskie 10, oznaczają wpisy silnie pozytywne („Kocham mojego nowego Galaxy S6!!!”), a wartości niskie oznaczają wpisy niezadowolonych użytkowników („Co za bzdzień! #rozczarowanie”).

W tym ćwiczeniu będziesz pracował na właśnie takich danych, które zostały już specjalnie przetworzone i ułożone w tabelki, abyś nie musiał dokonywać żmudnych manipulacji na tekstach. Twoim zadaniem jest dokonanie eksploracyjnej analizy danych, podczas której powinieneś dowiedzieć się czego dotyczą analizowane wpisy (nie są to wpisy dot. smartfonów jak w przykładzie wyżej). Powodzenia!



1. Na początku przygotuj twoje środowisko pracy poprzez wywołanie polecenia, które ściągnie z Internetu wszystkie potrzebne dane.

```
source("http://www.cs.put.poznan.pl/mlango/siad/data/ead.R")
```

2. Dane w R najczęściej są przechowywane w tzw. ramkach danych (ang. *data frame*), która są kolekcjami wektorów (kolumn). Jedną z załadowanych przez skrypt struktur jest ramka danych `vocabulary`. Sprawdźmy jej rozmiar poprzez wywołane polecenia:

```
dim(vocabulary)
```

Zostały zwrócone dwie liczby: pierwsza z nich to liczba wierszy (obserwacji), a druga to liczba kolumn (atrybutów). Dostęp do tych danych wygląda analogicznie jak do wektorów, z tym że jest to struktura dwuwymiarowa, więc trzeba podawać 2 indeksy (numer wiersza, numer kolumny). Na szczęście jeden z indeksów można pominąć – zostanie wtedy wyświetlony cały wiersz lub cała kolumna. Na przykład aby wyświetlić piąty wiersz ramki wystarczy wpisać:

```
vocabulary[5,]
```

Zauważ, że pominęliśmy indeks kolumny w związku z czym zostały wyświetlone wszystkie kolumny piątego wiersza.

3. Innym sposobem na wyświetlenie większej liczby danych jest funkcja (analogiczna do Unix'owej) `head()`. Wywołaj ją na naszej strukturze, aby zobaczyć pierwsze pięć wierszy.



Analogie do Unix'a na tym się nie kończą: istnieje analogiczna funkcja `tail()` czy funkcja `ls()` wypisująca wszystkie zmienne w obszarze roboczym. Jednak największym błogosławieństwem początkującego programisty R jest auto-uzupełnianie poprzez wciskanie przycisku tabulacji (tak jak w konsoli Linux'a). Gorąco polecam częste korzystanie z tej funkcjonalności ;)

4. Jak pewnie się już domyślasz tabela ta zawiera słowa wraz z liczbą wystąpień we wszystkich wpisach użytkowników. Aby wyświetlić nazwy kolumn wystarczy wywołać polecenie:

```
names(vocabulary)
```

Ojej! Nazwy tych kolumn nic nam nie mówią – pora to zmienić! Skorzystajmy z magii języka R i przypiszmy wartość do wyniku funkcji (!).

```
names(vocabulary) <- ...
```

W miejsce trzech kropek należy podać wektor nazw kolumn, w naszym przypadku dwuelementowy. Aby zachować spójność z dalej przyjętą notacją nazwij pierwszą kolumnę „word”, a drugą „count”.

5. Teraz, gdy kolumny już są nazwane możemy odwoływać się do poszczególnych kolumn w sposób znacznie wygodniejszy. Na przykład aby uzyskać dostęp do całej kolumny „count” możemy użyć następującej składni: `vocabulary$count`. Na wektorze tym możemy np. policzyć średnią arytmetyczną:

```
mean( vocabulary$count )
```

Istnieją też analogiczne funkcje: `max()`, `min()`, `median()` czy `sort()`. Na szczególną uwagę zasługuje funkcja wyświetlająca podstawowe statystyki dotyczące każdej z kolumn: `summary()` (możemy ją wywołać na całej ramce danych, a nie tylko na kolumnie).

6. Przystąpmy do analizy: chcielibyśmy się dowiedzieć czego dotyczą zebrane przez nas wpisy użytkowników. W tym celu sprawdzimy jak wyglądają częstotliwości występowania poszczególnych słów. Utwórz wykres słupkowy na kolumnie „count” poprzez wpisanie komendy:

```
barplot(vocabulary$count)
```

7. Niestety, na wykresie niezbyt wiele widać... Spróbuj posortować wartości wektora zanim utworzysz wykres. Może teraz uda Ci się wyciągnąć jakieś wnioski?
8. Z wykresu wynika, że istnieje pewna mała grupa słów, które występują bardzo często w stosunku do reszty. Jeśli jakieś słowo ma dużą częstotliwość to znaczy, że wystąpiło w wielu wpisach użytkowników – może więc dzięki nim dowiemy się czego dotyczą obserwowane przez nas wpisy? Odczytaj z wykresu wartość powyżej której występuje bardzo mała liczba słów, a następnie używając filtrowania wypisz je na ekran.^a
9. Spotkało nas rozczarowanie: większość z wyświetlonych słów nie ma żadnego realnego znaczenia, a ich częste występowanie nie niesie żadnej informacji o temacie wpisów w zbiorze danych. Na szczęście jest to znany problem w analizie tekstu, a takie słowa nazywamy z angielskiego *stopwords*^b. Wśród zmiennych w środowisku roboczym jest wektor `stopwords`, zawierający powszechnie używaną listę takich słów (możesz sobie ją wyświetlić).

Utwórz nową ramkę danych `vocabulary_filtered` poprzez wyfiltrowanie wszystkich wierszy zawierających *stopwords*. Użyj operatora `%in%` sprawdzającego czy wartość zmiennej występuje w zbiorze oraz operatora negacji `!`.

10. Sprawdź poleceniem `dim()` czy liczba wierszy w nowej ramce jest trochę mniejsza niż w oryginalnej.
11. Utwórz wykres słupkowy na nowej ramce danych (pamiętaj o posortowaniu wektora).

```
barplot(sort(vocabulary_filtered$count))
```


12. Wykres szczególnie się nie zmienił, poza tym że zakres wartości jest trochę mniejszy. Jednak teraz najczęściej występujące słowa powinny być bardziej znaczące – wyświetl je. Czy już wiesz czego dotyczą wpisy użytkowników?

^aPodpowiedź: pamiętaj, że ramkę danych możesz indeksować podobnie jak wektory! Sprawdź jak filtrowaliśmy liczby większe od 500 w przykładzie z wektorami.

^bhttps://pl.wikipedia.org/wiki/Stop_lista_%28wyszukiwarki%29

Teraz, gdy już wiemy, że wpisy użytkowników dotyczą odwiecznego sporu informatyków o to który system operacyjny jest najlepszy, spróbujmy sprawdzić o którym systemie użytkownicy częściej piszą oraz który system ma więcej pozytywnych wpisów.



13. W tej części ćwiczenia będziemy analizować nową ramkę danych dostępną w zmiennej `day_stats`. Ramka ta zawiera 4 kolumny:
- (a) datę (`date`)
 - (b) nazwę systemu operacyjnego (os: „linux” lub „windows”)
 - (c) liczbę tweetów opublikowanych w danym dniu i dotyczącego danego systemu operacyjnego (`count`)
 - (d) średnią ocenę pozytywnego wydźwięku wpisów (`positive_coeff`)
- Obejrzyj kilka rekordów z tabeli (funkcja `head()`), a następnie wyświetl podsumowanie informacji o tych danych (`summary()`). Z jakiego okresu czasu pochodzą dane? Jaka jest średni współczynnik pozytywnego wydźwięku analizowanych wpisów?
14. W celu dokładniejszego zbadania wartości współczynnika wydźwięku skonstruujmy dla niego histogram:

```
hist(day_stats$positive_coeff)
```

15. Chcielibyśmy uzyskać histogram z większą liczbą słupków oraz z słupkami pokolorowanymi na czerwono (`"red"`). Korzystając z pomocy do funkcji `hist` (aby ją wyświetlić wpisz znak zapytania i nazwę funkcji) zbuduj takie histogramy dla liczby słupków 70 i 150.
16. Obraz histogramu przy 150 słupkach jest „szarpany” (raz słupek raz wolne miejsce). Dlaczego się tak stało? Aby odpowiedzieć na to pytanie zobacz do ilu miejsc po przecinku są raportowe współczynniki wydźwięku i jaki jest ich zakres wartości. Jaka jest szerokość przedziału, który obejmuje jeden słupek na wykresie? Jak sądzisz czy ten histogram jest prawidłowy?
17. Innym sposobem zobaczenia rozkładu wartości (z mniejszą szczegółowością) jest narysowanie wykresu pudełkowego. Narysuj go w R poprzez wywołanie funkcji `boxplot()`.
18. Pewną zaletą wykresu pudełkowego nad histogramem jest to, że na jednym wykresie możemy zaprezentować kilka serii wartości. Chcielibyśmy się dowiedzieć jak różni się rozkład współczynnika wydźwięku dla różnych systemów operacyjnych. Aby to zrobić narysujmy wykres pudełkowy współczynnika wydźwięku w zależności od zmiennej system operacyjny.

```
boxplot(day_stats$positive_coeff ~ day_stats$os)
```

Wyrażenia typu $y \sim x$ możesz czytać jako „y zależy od x”.

O którym systemie operacyjnym użytkownicy piszą pozytywniej? Czy możesz wyciągnąć z wykresu jakieś inne wnioski?

19. Kończąc naszą analizę sprawdźmy czy są dni w których użytkownicy piszą częściej o którymś z systemów operacyjnych np. o systemie Linux. W tym celu utwórzmy nową ramkę zawierającą tylko dane o systemie Linux:

```
day_stats_linux <- day_stats[day_stats$os == 'linux',]
```

A następnie użyj funkcji `plot()` do narysowania wykresu liczby tweetów w czasie. Argumentem tej funkcji powinno być wyrażenie „liczba wpisów zależy od daty” (konstrukcja z operatorem \sim).

20. Otrzymaliśmy wykres punktowy, a chcielibyśmy otrzymać wykres liniowy. Korzystając z pomocy pakietu R znajdź dodatkowy argument funkcji `plot()`, który spowoduje utworzenie porządanego typu wykresu.
21. Wygląda na to, że liczba wpisów o systemie Linux waha się pomiędzy kolejnymi dniami, ale nie widzimy żadnego trendu tj. liczba wpisów nie rośnie ani nie maleje. Spróbuj zbudować analogiczny wykres dla systemu Windows – może tam zobaczysz coś ciekawego?
22. Korzystając z internetu spróbuj znaleźć przyczynę, która spowodowała nagły wzrost tweetów dotyczących systemu Windows w drugiej połowie sierpnia 2015?.

Ćwiczenie 3.9 — Eksploracyjna analiza danych medycznych^a. Behavioral Risk Factor Surveillance System (BRFSS) to roczna ankieta telefoniczna przeprowadzana w Stanach Zjednoczonych mająca na celu zidentyfikowanie i kontrolowanie zagrożeń zdrowotnych w populacji ludzi dorosłych. Respondenci udzielają odpowiedzi na pytania dotyczące ich diety, cotygodniowej aktywności fizycznej, użycia papierosów, chorowania na HIV/AIDS itd.^b. Losowa próbka 20 tysięcy obserwacji z tego badania jest dostępna w zmiennej `medical_data`. Przeprowadź eksploracyjną analizę tych danych, a w szczególności:

- narysuj wykres słupkowy pokazujący liczbę osób palących i nie palących (kolumna „smoke100” zawiera odpowiednio „0” lub „1” gdy ktoś pali lub nie)^c. Podpisz osie wykresu.
- narysuj wykres pudełkowy wysokości człowieka (`height`) w zależności od płci (`gender`).
- używając wykresu pudełkowego sprawdź zależność pomiędzy ogólną oceną stanu zdrowia (`genhlth`) a wskaźnikiem masy ciała BMI^d, który (biorąc pod uwagę jednostki użyte do mierzenia wysokości i wagi w zbiorze danych) można wyrazić wzorem:

$$BMI = \frac{weight}{height^2} \cdot 703$$

Zdrowy człowiek powinien mieć BMI pomiędzy 18.5 a 25.

- narysuj histogram wieku badanych (age)
- zbadaj zależność pomiędzy płcią (gender) oraz różnicą pomiędzy pożądaną wagą (wtdesired) a wagą aktualną (weight).

^aĆwiczenie przygotowane w oparciu o „OpenIntro Statistics” [9]

^bPełen opis tego badania możesz znaleźć na stronie <http://www.cdc.gov/brfss>

^cDo konstrukcji tabeli częstości występowania wartości zmiennej można wykorzystać funkcję `table()` – patrz ćwiczenie z szeregiem rozdzielczym.

^dPatrz: https://pl.wikipedia.org/wiki/Wska%C5%BAnik_masy_cia%C5%82a

Ćwiczenie 3.10 Dane

```
y<-c(33,44,29,16,25,45,33,19,54,22,21,49,11,24,56)
```

zawierają sprzedaż mleka w litrach dla 5 dni w 3 różnych sklepach (pierwsze 3 wartości są sprzedażą mleka w poniedziałek w kolejnych sklepach). Wygeneruj podsumowanie statystyczne sprzedaży (średnia, mediana, kwartyle) dla każdego ze sklepów. [3]

Literatura

Literatura na powtórkę

Wprowadzenie po polsku do pakietu R można znaleźć np. w „Przewodniku po pakiecie R” Przemysława Biecka dostępnego pod adresem: <https://cran.r-project.org/doc/contrib/Bieck-R-basics.pdf> [8]. Opis wykresów i różnych sposobów wizualizacji danych można znaleźć w pierwszym rozdziale książki „Statystyka w zarządzaniu” [6], a ciekawe eseje o wizualizacji można znaleźć w [7].

Literatura dla chętnych

Dla chętnych polecam bardzo ciekawy przykład interaktywnej wizualizacji edukacyjnej dot. podstaw bardziej zaawansowanej dziedziny analizy danych – uczenia maszynowego <http://www.r2d3.us/visual-intro-to-machine-learning-part-1/>, a także wykłady z przedmiotu „Grafika informacyjna” [11].

Pytania sprawdzające zrozumienie

Pytanie 3.1 Czym jest eksploracyjna analiza danych? Jakie są rodzaje wizualizacji? Czym jest kwartet Anscombe’a?

Pytanie 3.2 Jak należy interpretować wykres pudełkowy?

Pytanie 3.3 Jakie są najczęstsze błędy popełniane przy wizualizacji danych?

Bibliografia

- [1] Kwartet Anscombe’a – Wikipedia, wolna encyklopedia. https://pl.wikipedia.org/wiki/Kwartet_Anscombe'a. Dostęp: 2016-02-10.

- [2] Diagram kołowy – Wikipedia, wolna encyklopedia. https://pl.wikipedia.org/wiki/Diagram_ko%C5%82owy. Dostęp: 2016-02-10.
- [3] R: A self-learn tutorial. <https://www.nceas.ucsb.edu/files/scicomp/Dloads/RProgramming/BestFirstRTutorial.pdf>. Dostęp: 2016-02-10.
- [4] Łodyga i liście – czy naprawdę istnieje taki wykres? <http://www.statystyczny.pl/lodyga-i-liscie-czy-naprawde-istnieje-taki-wykres/>, 2015. Dostęp: 2016-02-10.
- [5] Manipulacje wyborcze... <http://www.statystyczny.pl/manipulacje-wyborcze/>, 2015. Dostęp: 2016-02-10.
- [6] Amir D. Aczel. *Statystyka w zarządzaniu. Pełny wykład*. PWN, 2005.
- [7] Przemysław Biecek. *Odkrywać! Ujawniać! Objaśniać! Zbiór esejów o sztuce prezentowania danych*. Wydawnictwo Uniwersytetu Warszawskiego, 2014.
- [8] Przemysław Biecek. *Przewodnik po pakiecie R*. Oficyna Wydawnicza GiS, 2014.
- [9] D.M. Diez, C.D. Barr, i M. Çetinkaya Rundel. *OpenIntro Statistics: Third Edition*. OpenIntro, Inc., 2015. ISBN 194345003X. URL openintro.org.
- [10] R.A. Irizarry i M.I. Love. *Data Analysis for the Life Sciences*. Leanpub, 2015. URL leanpub.com/dataanalysisforthelifesciences.
- [11] Jerzy Stefanowski. Wykład z przedmiotu „Grafika Informacyjna”. <http://www.cs.put.poznan.pl/jstefanowski/infoviz.html>. Dostęp: 2016-05-03.