PERTEMUAN 4

Visualisasi Data Menggunakan Rstudio

# SUB-CAPAIAN PEMBELAJARAN MATA KULIAH:

Mahasiswa menampilkan informasi grafis dan hubungan kompleks dalam kumpulan data menggunakan RStudio.

# POKOK BAHASAN:

1. Membuat visualisasi dasar: *scatterplot, bar plot, pie chart*
2. *Multiple plots*
3. *Pixel-oriented visualization techniques*
4. *Geometric projection visualization techniques*
5. *Icon-based visualization techniques*
6. *Hierarchical visualization techniques*
7. *Visualizing complex data*

# TUJUAN:

Setelah melakukan praktikum, mahasiswa diharapkan untuk dapat:

1. Ketepatan, kerapian dalam membuat plot
2. Kompleksitas pembuatan plot
3. Ketepatan dalam menginterpretasikan plot yang terbentuk

# DASAR TEORI:

Visualisasi data bertujuan untuk mengkomunikasikan data secara jelas dan efektif melalui representasi grafis. Visualisasi data telah digunakan secara luas di banyak aplikasi misalnya, di tempat kerja untuk pelaporan, mengelola operasi bisnis, dan melacak *progress* dari tugas yang diberikan. Lebih populer lagi, kita dapat memanfaatkan teknik visualisasi untuk menemukan hubungan data yang sebaliknya tidak mudah diamati dengan melihat data mentah. Saat ini, orang juga menggunakan visualisasi data untuk membuat grafik yang menyenangkan dan menarik. Pada bagian ini, kita secara singkat memperkenalkan konsep dasar visualisasi data. Kita mulai dengan data pembuatan visualisasi data dasar seperti *scatterplot*, *bar plot*, *pie chart* hingga multidimensi seperti yang disimpan dalam database relasional. Kita membahas beberapa pendekatan representatif, termasuk teknik berorientasi piksel, teknik proyeksi

geometris, teknik berbasis ikon, dan teknik berbasis hierarki dan grafik. Kita kemudian membahas visualisasi data yang kompleks dan hubungan.

1. **Membuat visualisasi dasar: *scatterplot, bar plot, pie chart Scatterplot***

Fungsi plot() dalam bahasa pemrograman R adalah fungsi serbaguna; fungsi ini dapat mengambil vektor, fungsi, atau objek apa pun yang memiliki metode plot() sebagai input. Dalam bentuknya yang paling sederhana, dibutuhkan dua vektor sebagai input untuk membuat *scatterplot*. *Scatterplot* memvisualisasikan hubungan antara dua variabel dengan hanya menampilkan titik data dengan koordinat 𝑥 sebagai satu variabel dan koordinat 𝑦 sebagai variabel lain. Secara konvensional, variabel pada sumbu 𝑥 adalah variabel bebas dan variabel pada sumbu 𝑦 adalah variabel terikat.

# Labeling data points

Karena *scatterplot* menampilkan semua titik data dari kumpulan data, *scatterplot* ini adalah alat yang berguna untuk memvisualisasikan tren data. Memberikan label pada beberapa titik data di *scatterplot* bisa bermanfaat untuk mengetahui poin ekstrem atau kasus menarik yang ingin kita soroti dalam visualisasi data kita. Kita dapat menggunakan fungsi identify() untuk tujuan ini. Pertama-tama kita membuat plot menggunakan fungsi plot() seperti sebelumnya. Kita dapat menemukan titik data yang ingin kita beri label dari *scatterplot* itu sendiri. Kita akan coba praktikkan menggunakan data iris.

# Points and lines

Dalam *scatterplot* di atas, kita menampilkan semua titik data secara bersamaan. Terkadang penggunaan simbol atau warna yang berbeda berguna untuk menunjukkan kelompok titik data yang berbeda. Hal ini membantu kita dalam memahami jika himpunan bagian dari data memiliki karakteristik yang berbeda dari keseluruhan kumpulan data.

## Bar plot

Diagram batang (*bar plot*) dan diagram lingkaran (*pie chart*) adalah alat yang berguna untuk memvisualisasikan nilai gabungan dari variabel kuantitatif. Diagram batang berisi sekumpulan batang atau persegi panjang, di mana tinggi batang sesuai dengan nilai variabel. Demikian pula, diagram lingkaran memvisualisasikan data agregat sebagai sektor lingkaran. Untuk Bahasa R, kita membuat diagram batang menggunakan fungsi barplot() dan diagram lingkaran menggunakan fungsi pie(). Fungsi-fungsi ini mengambil nilai agregat sebagai input.

## Pie chart

Diagram lingkaran adalah visualisasi alternatif yang menunjukkan nilai gabungan dari variabel yang dihitung pada subkelompok data yang berbeda. Nilai agregat untuk subkelompok yang berbeda ditampilkan sebagai bagian dari keseluruhan. Di sini, lingkaran mewakili nilai agregat yang dihitung dari seluruh kumpulan data, dan irisan atau sektor yang mewakili, dihitung melalui subkelompok data yang berbeda.

Diagram lingkaran menyampaikan informasi yang sama persis dengan diagram batang dimana keduanya memberikan nilai agregat untuk variabel melalui subkelompok data yang berbeda. Selain itu, ada keuntungan untuk diagram batang karena memiliki skala pada sumbu y yang memungkinkan kita untuk dengan mudah membandingkan ketinggian batang yang berbeda. Berbeda dengan diagram lingkaran karena akan lebih sulit untuk membandingkan secara visual area irisan, terutama di seluruh diagram lingkaran yang berbeda. Dalam statistik, penggunaan diagram lingkaran tidak dianjurkan karena alasan ini. Namun demikian, diagram lingkaran tetap banyak digunakan dalam bisnis dan media populer. Kita menggambar diagram lingkaran menggunakan fungsi pie(). Mirip dengan barplot(), fungsi ini juga menggunakan data agregat yang merupakan output dari fungsi by().

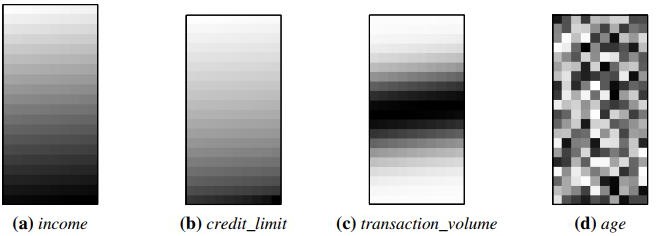
## Multiple plots

Kita sering perlu menampilkan beberapa visualisasi bersama-sama. Ini bisa menjadi jenis plot yang sama yang digambar pada subset yang berbeda dari dataset atau jenis plot yang berbeda yang digambar pada dataset yang sama. Parameter mfrow dari fungsi par() memungkinkan kita untuk menggambar banyak plot dalam satu tempat yang sama (*grid*). Parameter ini mengambil vektor dua elemen yang menunjukkan jumlah baris dan kolom plot.

## Pixel-oriented visualization techniques

Cara sederhana untuk memvisualisasikan nilai dimensi adalah dengan menggunakan piksel di mana warna piksel mencerminkan nilai dimensi. Untuk kumpulan data 𝑚 dimensi, teknik berorientasi piksel membuat 𝑚 jendela di layar, satu jendela untuk setiap dimensi. Nilai dimensi 𝑚 dari catatan dipetakan ke 𝑚 piksel pada posisi yang sesuai di jendela. Warna piksel mencerminkan nilai yang sesuai. Di dalam jendela, nilai data diatur dalam beberapa *global order* yang dibagikan oleh semua jendela. *Global order* dapat diperoleh dengan menyortir semua catatan data dengan cara yang berarti untuk tugas yang ada.

Contoh Visualisasi berorientasi piksel. AllElectronics memuat tabel informasi pelanggan, yang terdiri dari empat dimensi: pendapatan, batas kredit, volume transaksi, dan usia. Bisakah kita menganalisis korelasi antara pendapatan dan atribut lainnya dengan visualisasi?

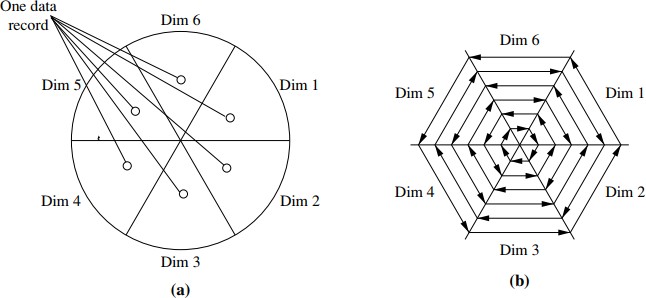
Kita dapat mengurutkan semua pelanggan dalam urutan pendapatan menaik dan menggunakan urutan ini untuk meletakkan data pelanggan di empat jendela visualisasi, seperti yang ditunjukkan pada Gambar 8.1. Warna piksel dipilih sehingga semakin kecil nilainya, semakin terang bayangannya. Dengan menggunakan visualisasi berbasis piksel, kita dapat dengan mudah mengamati hal berikut: batas kredit meningkat seiring dengan peningkatan pendapatan; pelanggan yang pendapatannya berada di kisaran menengah lebih cenderung membeli lebih banyak dari AllElectronics; tidak ada korelasi yang jelas antara pendapatan dan usia.

**Gambar 8.1** Visualisasi berorientasi piksel dari empat atribut dengan mengurutkan semua pelanggan dalam pendapatan urutan menaik

Dalam teknik berorientasi piksel, catatan data juga dapat diurutkan dengan cara *query- dependent*. Misalnya, dengan kueri titik (*point query*), kita dapat mengurutkan semua catatan dalam urutan menurun dari kesamaan dengan *point query*.

Mengisi jendela dengan meletakkan catatan data secara linier mungkin tidak bekerja dengan baik untuk jendela yang lebar. Piksel pertama berturut-turut jauh dari piksel terakhir di baris sebelumnya, meskipun mereka bersebelahan dalam tatanan global. Terlebih lagi, sebuah piksel berada di sebelah yang di atasnya di jendela, meskipun keduanya tidak bersebelahan dalam tatanan global. Untuk mengatasi masalah ini, kita dapat meletakkan catatan data dalam kurva mengisi ruang untuk mengisi jendela. Kurva pengisi ruang adalah kurva dengan rentang yang mencakup seluruh hypercube unit n-dimensi. Karena jendela visualisasi adalah 2-D, kita dapat menggunakan kurva pengisian ruang 2-D. Gambar 8.2 menunjukkan beberapa kurva pengisian ruang 2D yang sering digunakan. Perhatikan bahwa jendela tidak harus persegi

panjang. Sebagai contoh, teknik segmen lingkaran menggunakan jendela-jendela yang berbentuk segmen-segmen lingkaran, seperti diilustrasikan pada Gambar 2.12. Teknik ini dapat mempermudah perbandingan dimensi karena jendela dimensi terletak berdampingan dan membentuk lingkaran.



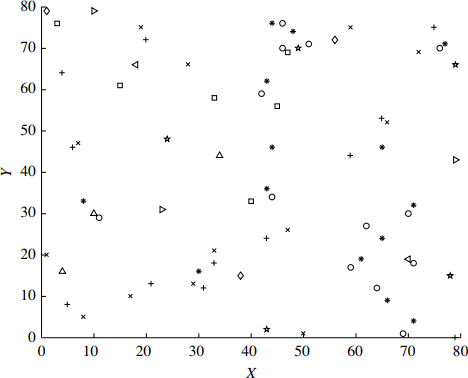
**Gambar 8.2** Circle segment technique. (a) Mewakili catatan data dalam segmen lingkaran.

(b) Meletakkan piksel dalam segmen lingkaran

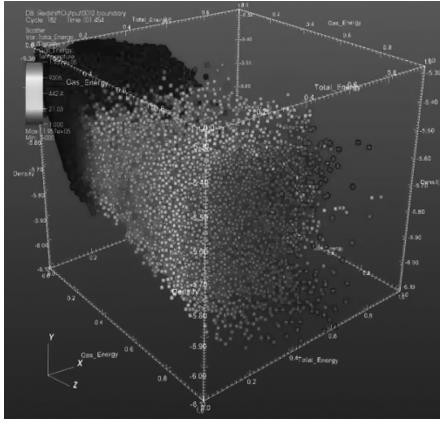
## Geometric projection visualization techniques

Kelemahan teknik visualisasi berorientasi piksel adalah bahwa teknik tersebut tidak dapat banyak membantu kita dalam memahami distribusi data dalam ruang multidimensi. Misalnya, mereka tidak menunjukkan apakah ada area padat di subruang multidimensi. Teknik proyeksi geometris membantu pengguna menemukan proyeksi menarik dari kumpulan data multidimensi. Tantangan utama yang coba diatasi oleh teknik proyeksi geometris adalah bagaimana memvisualisasikan ruang dimensi tinggi pada tampilan 2-D.

*Scatterplot* menampilkan titik data 2-D menggunakan koordinat Cartesian. Dimensi ketiga dapat ditambahkan menggunakan warna atau bentuk yang berbeda untuk mewakili titik data yang berbeda.

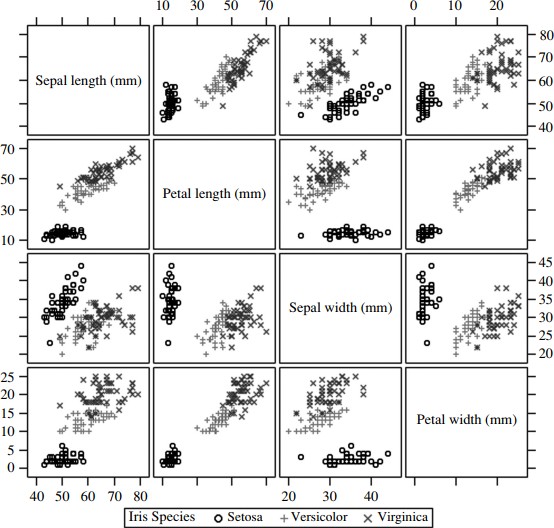


Gambar 8.3 Visualisasi kumpulan data 2-D menggunakan *scatterplot*

Gambar 8.3 menunjukkan contoh, di mana X dan Y adalah dua atribut spasial dan dimensi ketiga diwakili oleh bentuk yang berbeda. Melalui visualisasi ini, kita dapat melihat bahwa titik-titik bertipe “+” dan “×” cenderung colocated. *Scatterplot* 3-D menggunakan tiga sumbu dalam sistem koordinat Cartesian. Jika juga menggunakan warna, dapat menampilkan hingga titik data 4-D (Gambar 8.4).

Gambar 8.4 Visualisasi kumpulan data 3-D menggunakan *scatterplot*

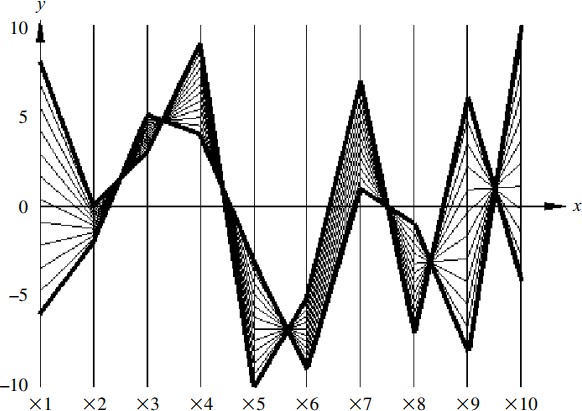
Untuk kumpulan data dengan lebih dari empat dimensi, *scatterplot* biasanya tidak efektif. Teknik *scatter-plot matrix* adalah ekstensi yang berguna untuk *scatterplot*. Untuk sekumpulan

data n-dimensi, *scatter-plot matrix* adalah grid n × n dari *scatterplot* 2-D yang memberikan visualisasi setiap dimensi dengan setiap dimensi lainnya. Gambar 8.5 menunjukkan sebuah contoh, yang memvisualisasikan kumpulan data Iris. Kumpulan data terdiri dari 450 sampel dari masing-masing tiga spesies bunga Iris. Ada lima dimensi dalam kumpulan data: panjang dan lebar dari sepal dan petal, serta spesies.

Gambar 8.5 Visualisasi kumpulan data Iris menggunakan *scatter-plot matrix*

Matriks *scatter-plot* menjadi kurang efektif dengan meningkatnya dimensi. Teknik populer lainnya, yang disebut koordinat paralel, dapat menangani dimensi yang lebih tinggi. Untuk memvisualisasikan titik data n-dimensi, teknik koordinat paralel menggambar n sumbu yang berjarak sama, satu untuk setiap dimensi, sejajar dengan salah satu sumbu tampilan. Catatan data diwakili oleh garis poligonal yang memotong setiap sumbu pada titik yang sesuai dengan nilai dimensi terkait (Gambar 8.6).

Keterbatasan utama dari teknik koordinat paralel adalah tidak dapat secara efektif menunjukkan kumpulan data dari banyak catatan. Bahkan untuk kumpulan data beberapa ribu catatan, kekacauan visual dan tumpang tindih sering mengurangi keterbacaan visualisasi dan membuat pola sulit ditemukan.

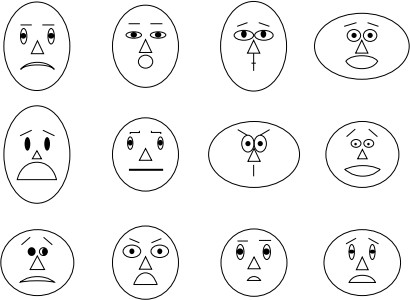


Gambar 8.6 Visualisasi yang menggunakan koordinat paralel

## Icon-based visualization techniques

Teknik visualisasi berbasis ikon menggunakan ikon kecil untuk mewakili nilai data multidimensi. Kita melihat dua teknik berbasis ikon yang populer: Chernoff faces dan stick figures.

Chernoff faces diperkenalkan pada tahun 1973 oleh ahli statistik Herman Chernoff. Mereka menampilkan data multidimensi hingga 18 variabel (atau dimensi) sebagai wajah manusia kartun (Gambar 8.7). Chernoff faces membantu mengungkap tren dalam data. Komponen wajah, seperti mata, telinga, mulut, dan hidung, mewakili nilai-nilai dimensi berdasarkan bentuk, ukuran, penempatan, dan orientasinya. Misalnya, dimensi dapat dipetakan ke karakteristik wajah berikut: ukuran mata, jarak mata, panjang hidung, lebar hidung, lengkungan mulut, lebar mulut, keterbukaan mulut, ukuran pupil, kemiringan alis, eksentrisitas mata, dan eksentrisitas kepala.



Gambar 8.7 Chernoff faces. Setiap wajah mewakili n-dimensional data point (𝑛 ≤ 18)

Chernoff faces memanfaatkan kemampuan pikiran manusia untuk mengenali perbedaan kecil dalam karakteristik wajah dan untuk mengasimilasi banyak karakteristik wajah sekaligus. Melihat tabel data yang besar bisa jadi membosankan. Dengan memadatkan data, Chernoff faces membuat data lebih mudah dipahami oleh pengguna. Dengan cara ini, mereka memfasilitasi visualisasi keteraturan dan ketidakteraturan yang ada dalam data, meskipun kekuatan mereka dalam menghubungkan banyak hubungan terbatas. Keterbatasan lain adalah bahwa nilai data tertentu tidak ditampilkan. Selain itu, fitur wajah bervariasi dalam arti penting yang dirasakan. Ini berarti bahwa kesamaan dua wajah (mewakili dua titik data multidimensi) dapat bervariasi tergantung pada urutan dimensi yang ditetapkan untuk karakteristik wajah. Oleh karena itu, pemetaan ini harus dipilih dengan cermat. Ukuran mata dan kemiringan alis dianggap penting. Chernoff faces yang asimetris diusulkan sebagai perpanjangan dari teknik aslinya. Karena wajah memiliki simetri vertikal (sepanjang sumbu y), sisi kiri dan kanan wajah adalah identik, hal ini memakan ruang. Chernoff faces yang asimetris menggandakan jumlah karakteristik wajah, sehingga memungkinkan hingga 36 dimensi untuk ditampilkan.

Teknik visualisasi *stick figure* memetakan data multidimensi ke figur stik lima potong, di mana setiap figur memiliki empat anggota badan dan satu tubuh. Dua dimensi dipetakan ke sumbu tampilan (x dan y) dan dimensi lainnya dipetakan ke sudut dan/atau panjang anggota badan. Gambar 8.8 menunjukkan data sensus, di mana usia dan pendapatan dipetakan ke sumbu layar, dan dimensi yang tersisa (jenis kelamin, pendidikan, dan sebagainya) dipetakan ke *stick figure*. Jika item data relatif padat sehubungan dengan dua dimensi tampilan, visualisasi yang dihasilkan menunjukkan pola tekstur, yang mencerminkan tren data.

Gambar 8.8 Data sensus direpresentasikan menggunakan *stick figure*. Sumber: Profesor G. Grinstein, Departemen Ilmu Komputer, Universitas Massachusetts di Lowell

## Hierarchical visualization techniques

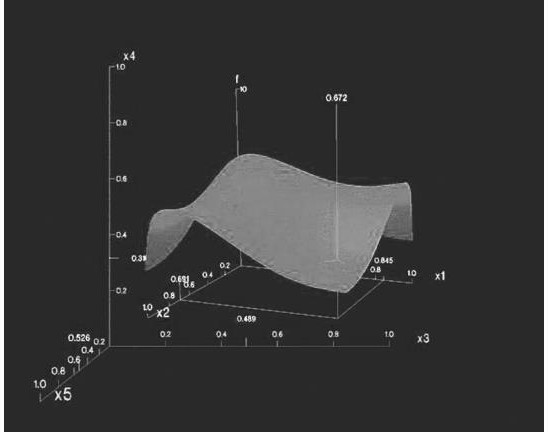
Teknik visualisasi yang dibahas sejauh ini berfokus pada memvisualisasikan beberapa dimensi secara bersamaan. Namun, untuk kumpulan data besar berdimensi tinggi, akan sulit untuk memvisualisasikan semua dimensi pada waktu yang sama. Teknik visualisasi hierarki mempartisi semua dimensi ke dalam himpunan bagian (yaitu, subruang). Subruang divisualisasikan secara hierarki.

**Worlds-within-Worlds** juga dikenal sebagai n-Vision, adalah representasi dari metode visualisasi hierarkis. Misalkan kita ingin memvisualisasikan kumpulan data 6-D, di mana dimensinya adalah 𝐹, 𝑋1, … , 𝑋5. Kita ingin mengamati bagaimana dimensi 𝐹 berubah sehubungan dengan dimensi lain. Pertama-tama kita dapat memperbaiki nilai dimensi

𝑋3, 𝑋4, 𝑋5 ke beberapa nilai yang dipilih, misalnya, 𝑐3, 𝑐4, 𝑐5. Kita kemudian dapat memvisualisasikan 𝐹, 𝑋1, 𝑋2 menggunakan plot 3-D, yang disebut *world*, seperti yang ditunjukkan pada Gambar 8.9. Posisi asal dari *inner world* terletak pada titik 𝑐3, 𝑐4, 𝑐5 di *outer world*, yang merupakan plot 3-D lainnya menggunakan dimensi 𝑋3, 𝑋4, 𝑋5. Seorang pengguna dapat secara interaktif mengubah, di *outer world*, lokasi asal *inner world*. Pengguna kemudian melihat perubahan yang dihasilkan dari *inner world*. Selain itu, pengguna dapat memvariasikan dimensi yang digunakan di *inner world* dan *outer world*. Mengingat lebih banyak dimensi, lebih banyak level *worlds* dapat digunakan, itulah sebabnya metode ini disebut Worlds-within- Worlds.

Sebagai contoh lain dari metode visualisasi hierarkis adalah ***tree-maps*** yang menampilkan data hierarkis sebagai kumpulan persegi panjang bersarang. Misalnya, Gambar

9.10 menunjukkan *tree-maps* yang memvisualisasikan berita Google. Semua berita diatur ke dalam tujuh kategori, masing-masing ditampilkan dalam persegi panjang besar dengan warna yang unik. Dalam setiap kategori (yaitu, setiap persegi panjang di tingkat atas), berita-berita tersebut selanjutnya dipartisi menjadi subkategori yang lebih kecil.



Gambar 8.9 "Worlds-within-Worlds" (juga dikenal sebagai n-Vision).



Gambar 8.10 Newsmap: Penggunaan peta pohon untuk memvisualisasikan berita utama berita Google

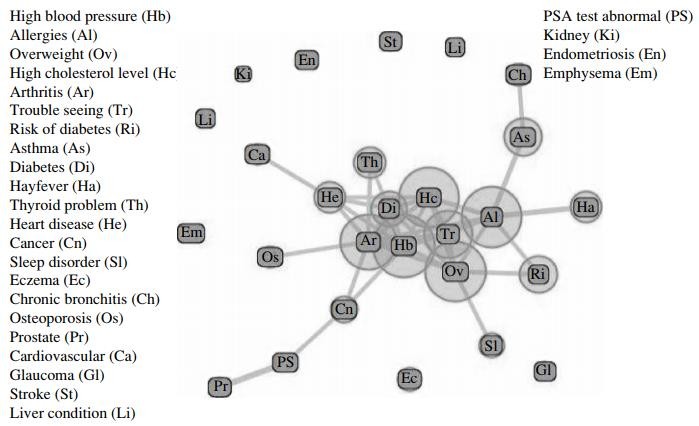
## Visualizing complex data

Pada awalnya, teknik visualisasi lebih banyak digunakan untuk data numerik. Dewasa ini, semakin banyak data non-numerik, seperti teks dan jejaring sosial (*social networks*). Memvisualisasikan dan menganalisis data semacam itu menarik banyak minat. Ada banyak teknik visualisasi baru yang didedikasikan untuk jenis data ini. Misalnya, banyak orang di Web menandai berbagai objek seperti gambar, entri blog, dan ulasan produk. ***Tag cloud*** adalah visualisasi statistik dari *tag* yang dibuat pengguna. Seringkali, dalam *tag cloud*, *tag* terdaftar menurut abjad atau dalam urutan yang disukai pengguna. Pentingnya *tag* ditunjukkan oleh ukuran font atau warna. Gambar 8.11 menunjukkan *tag cloud* untuk memvisualisasikan *tag* populer yang digunakan di situs Web.



Gambar 8.11 Menggunakan *tag cloud* untuk memvisualisasikan *Web site tags*

*Tag cloud* sering digunakan dalam dua cara. Pertama, dalam *tag cloud* untuk satu item, kita dapat menggunakan ukuran *tag* untuk menunjukkan berapa kali *tag* diterapkan ke item ini oleh pengguna yang berbeda. Kedua, saat memvisualisasikan *tag statistics* pada beberapa item, kita dapat menggunakan ukuran *tag* untuk mewakili jumlah item yang telah diterapkan *tag*, yaitu, popularitas *tag*.



Gambar 8.12 Grafik pengaruh penyakit dari orang yang berusia minimal 20 tahun dalam kumpulan data NHANES

Selain data yang kompleks, hubungan yang kompleks antara entri data juga menimbulkan tantangan untuk visualisasi. Misalnya, Gambar 8.12 menggunakan grafik pengaruh penyakit untuk memvisualisasikan korelasi antar penyakit. *Node* dalam grafik adalah penyakit, dan ukuran setiap *node* sebanding dengan prevalensi penyakit yang sesuai. Dua *node* dihubungkan oleh sebuah *edge* jika penyakit yang bersesuaian memiliki korelasi yang kuat. Lebar tepi sebanding dengan kekuatan pola korelasi dari dua penyakit yang sesuai.

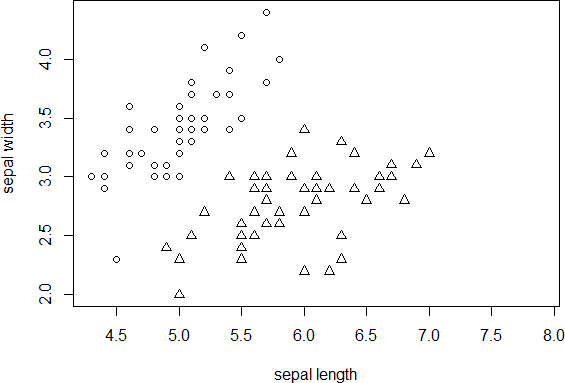
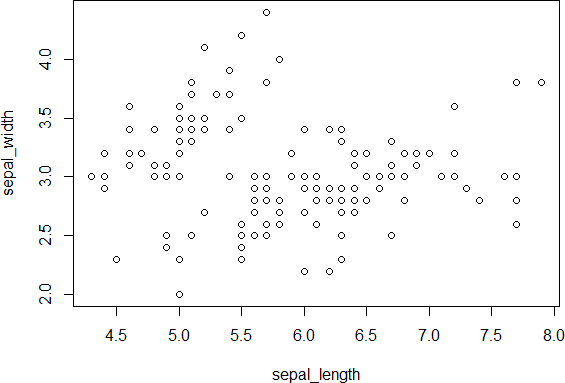
Singkatnya, visualisasi menyediakan alat yang efektif untuk mengeksplorasi data. Kita telah memperkenalkan beberapa metode populer dan ide-ide penting di baliknya. Ada banyak alat dan metode yang ada. Selain itu, visualisasi dapat digunakan dalam data mining dalam berbagai aspek. Selain untuk memvisualisasikan data, visualisasi dapat digunakan untuk merepresentasikan proses data mining, pola yang diperoleh dari metode mining, dan interaksi pengguna dengan data. Penambangan data visual (*Visual data mining*) adalah arah penelitian dan pengembangan yang penting

# PERCOBAAN PRAKTIKUM:

Untuk percobaan praktikum kali ini, mahasiswa mencoba menuliskan kode dalam Bahasa pemrograman R untuk beberapa jenis plot.

setwd('...')

data = read.csv('iris.csv') attach(data)



# scatterplot

plot(data$sepal\_length, data$sepal\_width)

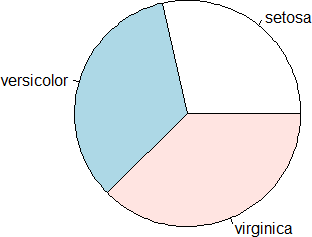
# scatterplot version 2

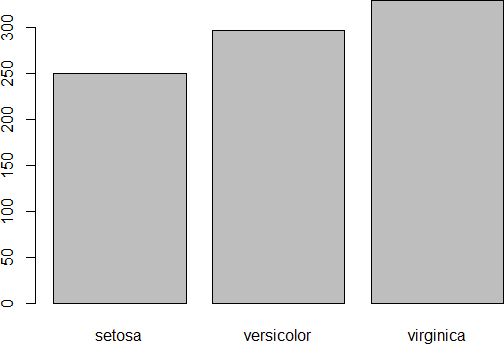
spec1 = which(species == 'setosa') spec2 = which(species == 'versicolor')

plot(sepal\_length[spec1], sepal\_width[spec1], xlim=range(sepal\_length), ylim=range(sepal\_width), xlab='sepal length', ylab='sepal width')

points(sepal\_length[spec2], sepal\_width[spec2], pch=2)

# bar plot barplot(by(sepal\_length,species,sum))





# pie chart

pie(by(as.numeric(sepal\_length), species, sum))

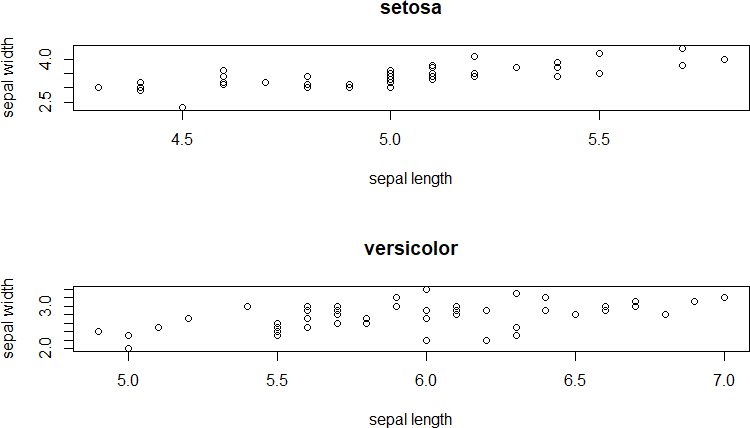
# multiple plots par(mfrow=c(2,1))

spec1 = which(species == 'setosa') spec2 = which(species == 'versicolor')

plot(sepal\_length[spec1], sepal\_width[spec1], main = 'setosa', xlab

= 'sepal length', ylab = 'sepal width')

plot(sepal\_length[spec2], sepal\_width[spec2], main = 'versicolor', xlab = 'sepal length', ylab = 'sepal width')



# TUGAS PRAKTIKUM:

Gunakan Bahasa pemrograman R

Pilih salah satu dataset yang menurut anda menarik untuk di eksplorasi, sumber dari Kaggle / UCI ML/ Lainnya

Buatlah infografik menggunakan RStudio/Rmarkdown dengan plot minimal 5 plot (dengan berbagai tipe plot) dan interpretasikan apa maksud dari plot yang telah dibuat.

* Tugas dikerjakan berkelompok.
* Tugas dikumpulkan sesuai deadline Hebat
* Bentuk pengumpulan tugas adalah file R / Rmd **dan** File JPG Poster Infografik
* Minggu depan presentasi temuan di Kelas teori & praktikum