Előfeldolgozás, exploratory analysis

Csima Judit

BME, VIK,
Számítástudományi és Információelméleti Tanszék

2018. február 12. és 19.

Mivel kezdődik az adatbányászat?

- majd tanulunk konkrét eljárásokat, amikkel az adatokból mindenféle érdekes infó nyerhető ki
- de ahhoz, hogy ezek menjenek szép adatok kellenek
- eredendően az adat sose szép, valamit biztos csinálni kell vele
- ez sok munka, nem egzakt feladat
- de azért a fő részeire van egy protokoll

Honnan szerzünk adatokat?

- néha úgy találjuk készen, valaki összegyűjtötte (ingyen elérhető, meg kell venni)
- szinte sose pont olyan, mint ami nekünk kell
- sokszor elosztottan van
- esetleg több táblából kell valahogy egyet csinálni (adatbázis kezelés)
- fontos, hogy dokumentáljuk, hogy honnan szereztük, honnan töltöttük le
- ha valaki már előfeldolgozta valahogy, akkor is értelmes látni a nyers adatot vagy legalább megérteni, hogy mi történt a feldolgozás során

Fő részek, ha már megvan az adat

- ismerkedés: milyen típusú attribútumok vannak, mit kódolnak, hogyan (ezt érintettük már a múltkor)
- exploratory elemzés: grafikonok, ábrák, mert így könnyebb látni mintázatokat
- preprocessing: attribútumok illetve sorok számának csökkentése

Ismerkedés az adattal

- honnan van az adat? hogyan gyűjtötték?
- elévült-e már az adat?
- attribútumok típusa, tipikus értékei, volt-e default érték a bevitelkor

Ismerkedés az iris data frame-mel

- ezt fogjuk használni demonstrációs célra
- letölthető innen:
  http://www.ics.uci.edu/∼mlearn/MLRepository.html
- benne van az R base package-ben is: ?iris
- háromféle írisz faj adatai
- négy attribútum: szirom hossza és szélessége, csészelevél hossza és szélessége

Ismerkedés az adattal R-ben

- legjobb, ha van dokumentáció, pl. R-ben ?iris elég sok infót megad:

  Format

  iris is a data frame with 150 cases (rows) and 5 variables (columns) named Sepal.Length, Sepal.Width, Petal.Length, Petal.Width, and Species.
- ebből most kiderül, hogy : Hány oszlop van? Mit kódolnak? Hány sor van?
- ha nincs ilyen dokumentáció vagy plusz infót akarunk: head(), summary() vagy str() függvények R-ben
- persze bármivel csinálhatjuk, csak derüljön ki, hogy kábé milyen számok vannak, milyen kategóriák, stb.

str()

```
> str(iris)
$ Sepal.Length:  num 5.1 4.9 4.7 4.6 5 5.4 4.6 5 4.4 4.9
...
$ Sepal.Width :  num 3.5 3 3.2 3.1 3.6 3.9 3.4 3.4 2.9 3.1
...
$ Petal.Length:  num 1.4 1.4 1.3 1.5 1.4 1.7 1.4 1.5 1.4
1.5 ...
$ Petal.Width :  num 0.2 0.2 0.2 0.2 0.2 0.4 0.3 0.2 0.2
0.1 ...
$ Species :  Factor w/ 3 levels "setosa","versicolor",..:
1 1 1 1 1 1 1 1 1 1 ...
```

summary()

```
> summary(iris)
  Sepal.Length    Sepal.Width   Petal.Length    Petal.Width
  Min.   :4.300   Min.   :2.000   Min.   :1.000   Min.   :0.100
 1st Qu.:5.100   1st Qu.:2.800   1st Qu.:1.600   1st Qu.:0.300
 Median :5.800   Median :3.000   Median :4.350   Median :1.300
   Mean :5.843     Mean :3.057     Mean :3.758     Mean :1.199
 3rd Qu.:6.400   3rd Qu.:3.300   3rd Qu.:5.100   3rd Qu.:1.800
  Max.   :7.900   Max.   :4.400   Max.   :6.900   Max.   :2.500

Species
setosa :50
versicolor:50
virginica :50
```

Exploratory elemzés: mi ez?

- ez alapján lehet eldönteni, hogy
  - milyen algoritmust használjunk
  - egy adott algoritmusban milyen attribútumok a fontosak (hol lehet érdekes, megvizsgálandó kapcsolat vagy hol van redundancia)
  - látszik-e valami nyilvánvaló hiba vagy tennivaló az adatokkal (átskálázás, hiányzó értékek, kilógó értékek)
- vannak olyan mintázatok, amiket egy jól sikerült ábrán az ember gyorsan felismer

Exploratory elemzés: fő részei

- összegző statisztikák készítése
- ábrázolás

Összegző statisztikák

- ezt már érintettük, amikor az adattal való ismerkedésről volt szó
- célja, hogy valami számszerű adattal összegezzük a változók értékeit
  - gyorsan számolható legyen
  - informatív legyen
- kábé hol vannak az értékek, mennyire szóródnak, mik a gyakoriságok
- általában vannak mindenféle hasznos parancsok erre

Összegző statisztikák, kategória típusú attribútum

- kategória típusú változónál a gyakoriságok informatívak
- erre láttuk már R-ben az str() és summary() függvényeket (ezekről mindjárt újra beszélünk)
- van egy table() függvény is:

```
> table(iris$Species)
 setosa  versicolor  virginica
     50          50         50
```

str()

```
> str(iris)
$ Sepal.Length:  num 5.1 4.9 4.7 4.6 5 5.4 4.6 5 4.4 4.9
...
$ Sepal.Width :  num 3.5 3 3.2 3.1 3.6 3.9 3.4 3.4 2.9 3.1
...
$ Petal.Length:  num 1.4 1.4 1.3 1.5 1.4 1.7 1.4 1.5 1.4
1.5 ...
$ Petal.Width :  num 0.2 0.2 0.2 0.2 0.2 0.4 0.3 0.2 0.2
0.1 ...
$ Species :  Factor w/ 3 levels "setosa","versicolor",..:
1 1 1 1 1 1 1 1 1 1 ...
```

Percentilisek

- folytonos adatokhoz jó
- 0 és 100 közötti percentilisekről beszélünk
- egy halmaz (attribútumhalmaz, adott oszlop értékei) p-percentilise az az x_p érték, aminél a halmaz értékeinek $p\%$-a kisebb egyenlő
- például $x_{50\%}$ azt az értéket adja meg, aminél az összes előforduló érték fele nem nagyobb
- szokásos nézni a 25, 50, 75 percentiliseket és a min és a max értéket
- pont erre szolgál a summary() R-ben
- de persze van quantile() függvény is, ahol beállítható, hogy milyen percentiliseket akarok, default a 0, 25, 50, 75, 100 (ahol 0 a min érték és 100 a max érték)

summary()

```
> summary(iris)
  Sepal.Length    Sepal.Width   Petal.Length    Petal.Width
  Min.   :4.300   Min.   :2.000   Min.   :1.000   Min.   :0.100
 1st Qu.:5.100   1st Qu.:2.800   1st Qu.:1.600   1st Qu.:0.300
 Median :5.800   Median :3.000   Median :4.350   Median :1.300
   Mean :5.843     Mean :3.057     Mean :3.758     Mean :1.199
 3rd Qu.:6.400   3rd Qu.:3.300   3rd Qu.:5.100   3rd Qu.:1.800
  Max.   :7.900   Max.   :4.400   Max.   :6.900   Max.   :2.500

Species
setosa :50
versicolor:50
virginica :50
```

Átlag (mean)

- az átlag (az adatok számtani közepe) az egyik leggyakoribb összegző függvény
- $mean(x) = \overline{x} = \frac{1}{m} \sum_{i=1}^{m} x_i$
- az átlag nagyon érzékeny a kilógó adatokra
- ezért sokszor a mediánt használjuk helyette

Medián (median)

- medián: hasonló az 50%-os percentilis értékéhez, de nem egészen az
- $median(x) = \begin{cases} x_{r+1} & \text{if } m = 2r+1 \\ \frac{1}{2}(x_r + x_{r+1}) & \text{if } m = 2r \end{cases}$
- ez persze enm ugyanaz, mint az átlag
- az emberek több, mint 99%-ának az átlagnál több lába van

Szórás-szerűségek

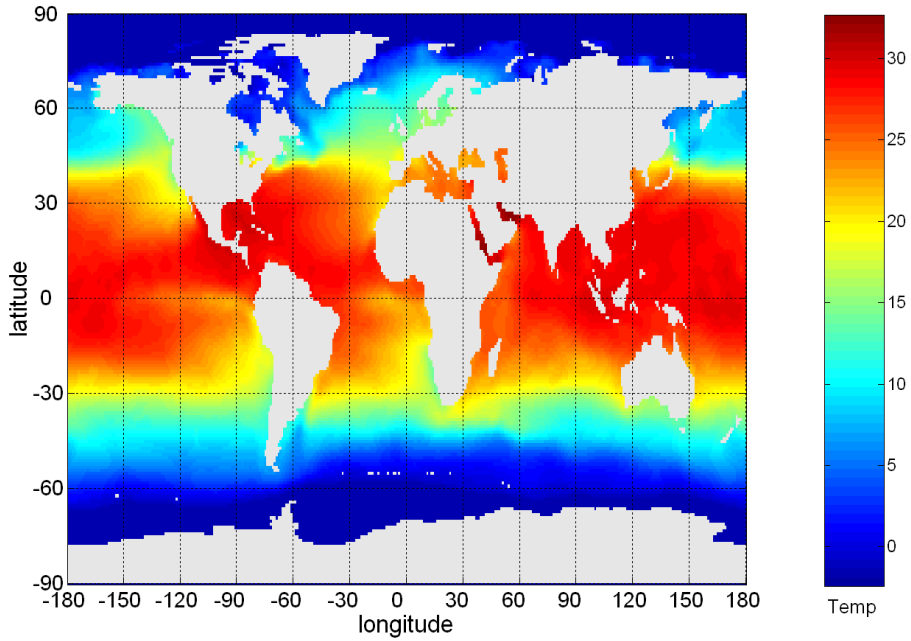
- range: milyen tartományba esnek az adatok (max - min)
- szórásnégyzet illetve szórás: $\frac{1}{m-1}\sum_{i=1}^{m}(x_i-\overline{x})^2$
- de ez is érzékeny a kilógó értékekre, ezért néha inkább $\frac{1}{m}\sum_{i=1}^{m}|x_i-\overline{x}|$
- ábrázolásnál lesznek majd olyan technikák, amikkel ezeket a mennyiségeket jól lehet látni

Ábrázolás célja az ismerkedés során

- az adatok közti kapcsolatot vagy adat tulajdonságait mutató jellemzőket ember számára feldolgozható módon megjeleníteni
- ember számára könnyebb egy grafikont értelmezni, mint egy táblázatot
- minták jobban látszanak (ember számára)
- kilógó adatok, furcsaságok is jobban kiugranak
- majd lesz szó arról, hogy az ábrázolás milyen szerepet kap az eredmények ismertetésekor
- általában sok ábra készül, gyorsan

Example: Sea Surface Temperature

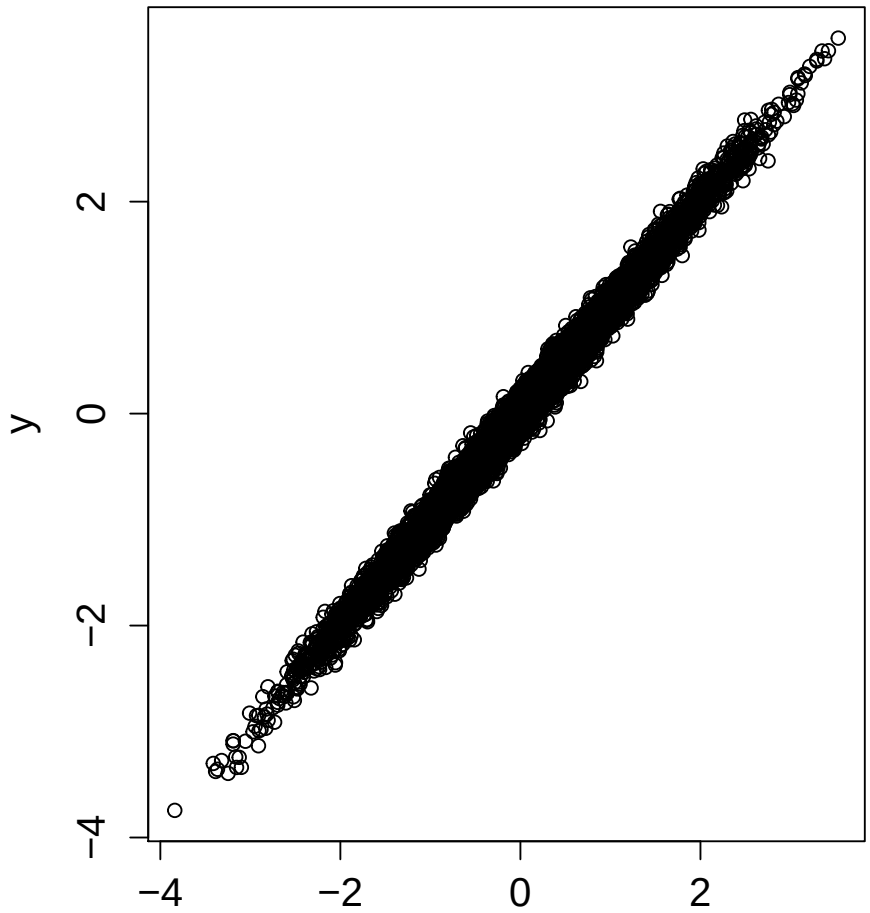
- The following shows the Sea Surface Temperature (SST) for July 1982
  - Tens of thousands of data points are summarized in a single figure

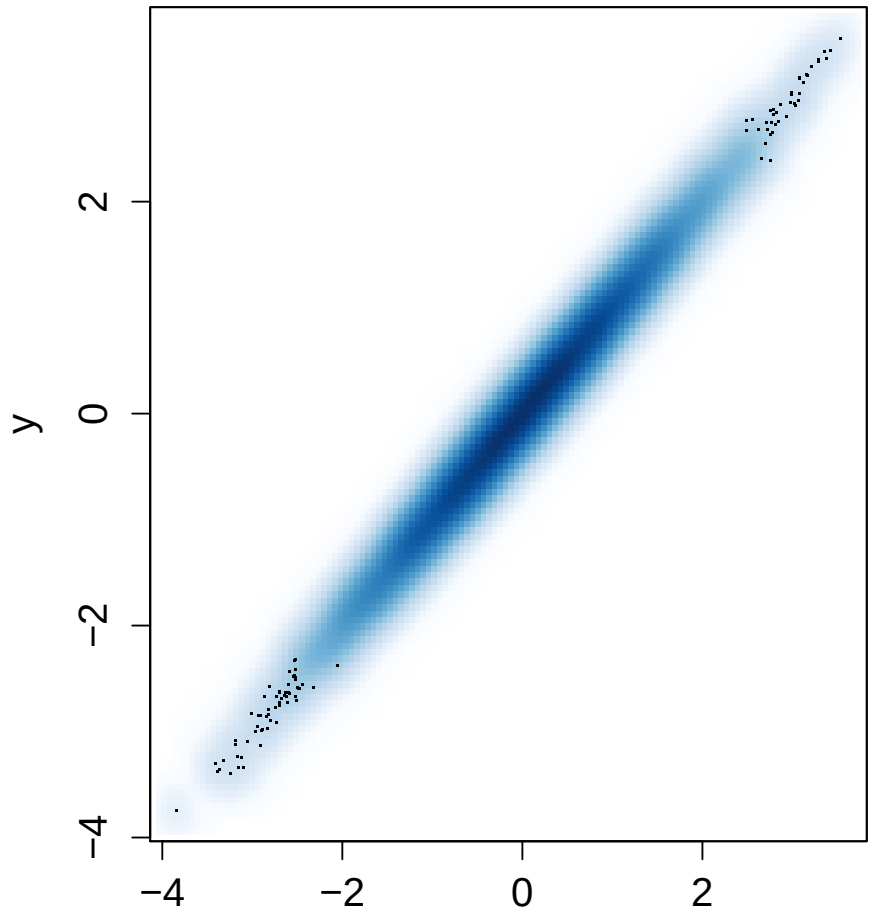


Milyen a (jó) ábrázolás?

- fontos a jó elrendezés
- cél egy jól értelmezhető ábra
- általában nem lehet mindent egy ábrában áttekinteni
- ügyesen választunk néhány attribútumot, amiket vagy amiknek a kapcsolatát megvizsgáljuk
- vannak R-ben ezt támogató klassz parancsok, erről majd laboron
- az exploratory elemzésnél elég, ha mi értjük, hogy mi van az ábrán
- sok fajtája lehet, pl. hisztogram, boxplot, scatterplot, stb.

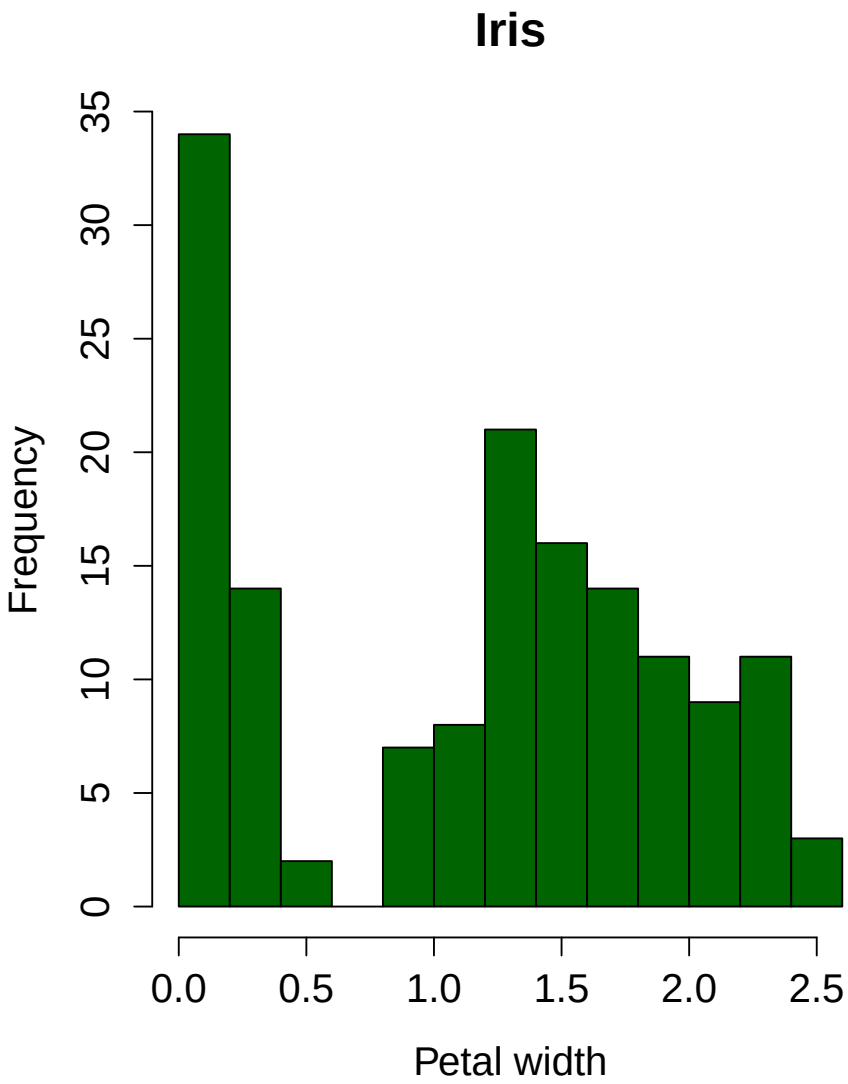
```
> x = rnorm(10000, 0, 1)
> y = x + rnorm(10000, 0, 0.1)
> plot(x,y)
> smoothScatter(x,y)
```

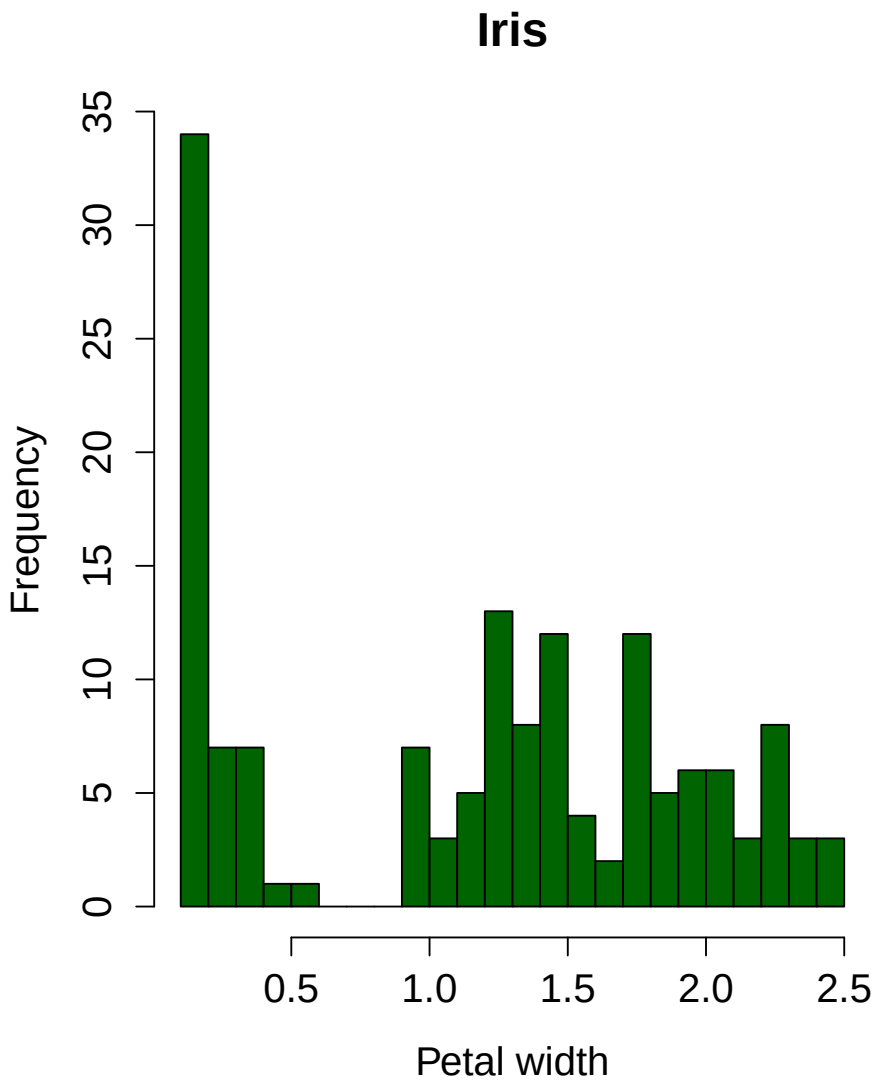




Hisztogram

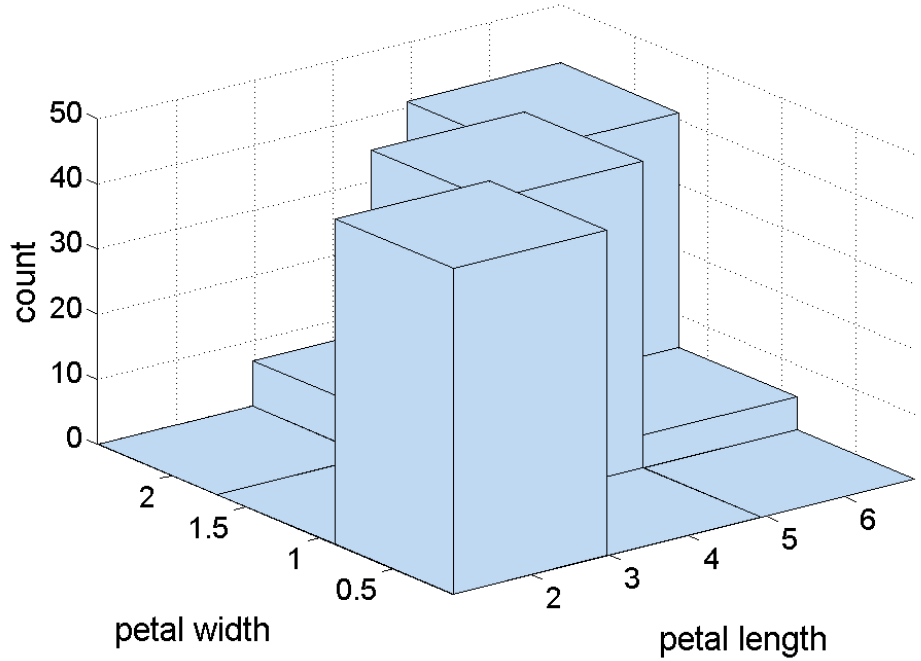
- egy változó értékeinek eloszlását mutatja
- csoportokba osztja az értékeket és az egy csoportba esők darabszámát mutatja
- az oszlopok magassága a darabszámot jelzi
- működik kategorikus és folytonos attribútumokra is


Iris
Frequency
0
5
10
15
20
25
30
35
0.0
0.5
1.0
1.5
2.0
2.5
Petal width


Iris
Frequency
0
5
10
15
20
25
30
35
0.5
1.0
1.5
2.0
2.5
Petal width

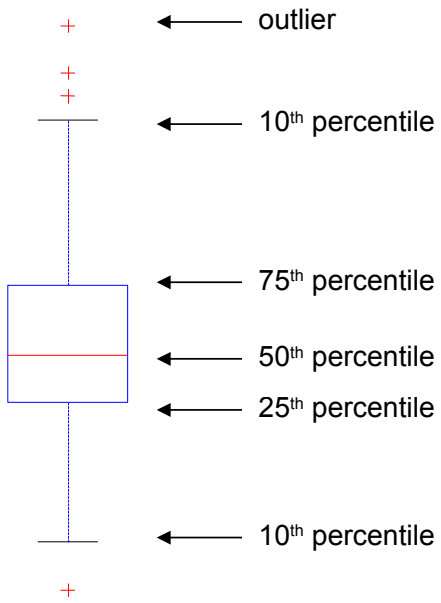
Two-Dimensional Histograms

- Show the joint distribution of the values of two attributes
- Example: petal width and petal length
  - What does this tell us?



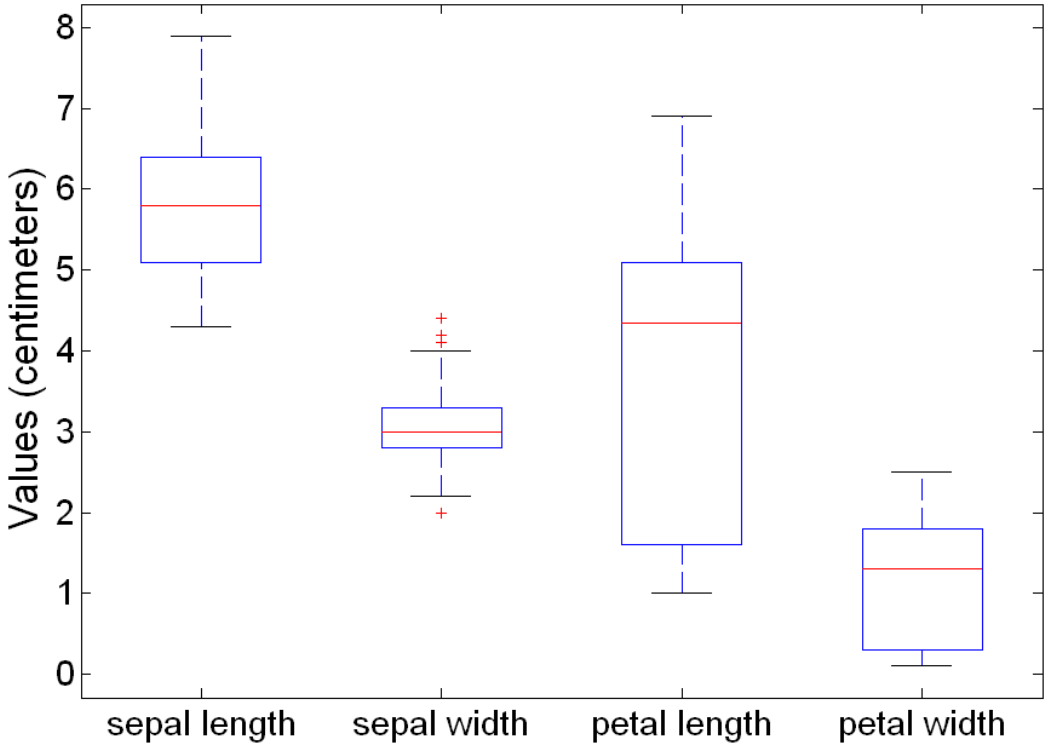
Visualization Techniques: Box Plots

- Box Plots
  - Invented by J. Tukey
  - Another way of displaying the distribution of data
  - Following figure shows the basic part of a box plot



Example of Box Plots

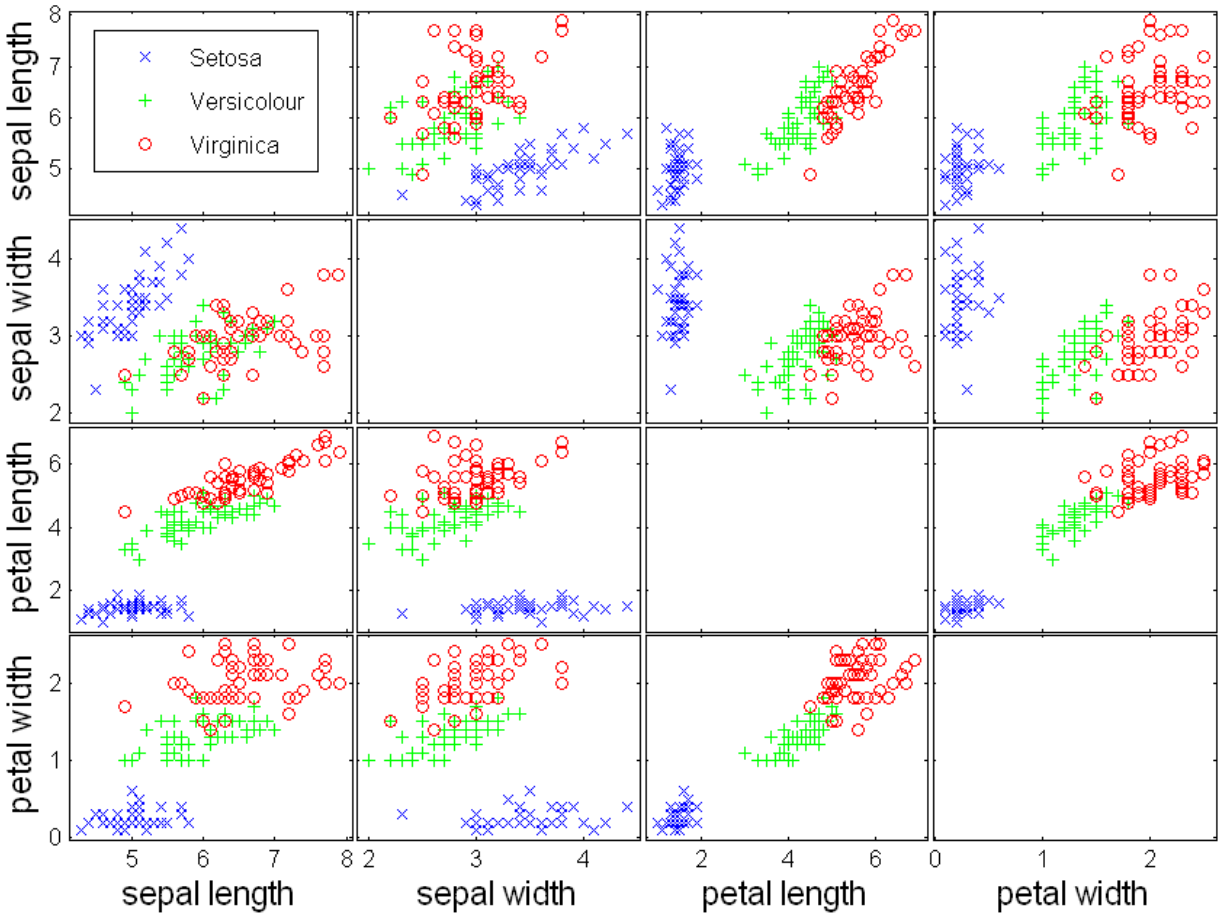
- Box plots can be used to compare attributes



Scatterplot (pontdiagramm)

- soroknak, objektumoknak pontok felelnek meg a síkon vagy esetleg térben
- a pontok helye megfelel a két vagy három kiválasztott attribútum értékeinek
- a max. három kiválasztott dimenzión felül a pontoknak lehet színe és/vagy alakja, és/vagy mérete, ezekkel együtt max. 5-6 dimenzió ábrázolható
- de azért igazából 4 dimenzió felett már nehéz értelmezni, amit látunk

Scatter Plot Array of Iris Attributes



Az eredmények prezentálása

- az ábrázolás fontos az eredmények prezentálásakor is
- részben hasonló elvek vonatkoznak rá, mint az exploratory ábrázolásra
  - fontos a jó elrendezés, cél a jól értelmezhető ábra
  - a legfontosabb eredményeket kell megmutatni, mindent nem lehet
  - sok fajtája lehet, pl. hisztogram, boxplot, scatterplot, stb.
- ami nagyon más: nem elég, ha mi értjük, hogy mi van az ábrán
- értelmes ábracím, tengelyek rendes elnevezése, skála mérete, informatív képaláírás
- laboron majd nézzük ezt R-ben

Az eredmények prezentálása

- az ábrázolás fontos az eredmények prezentálásakor is
- részben hasonló elvek vonatkoznak rá, mint az exploratory ábrázolásra
    - fontos a jó elrendezés, cél a jól értelmezhető ábra
    - a legfontosabb eredményeket kell megmutatni, mindent nem lehet
    - sok fajtája lehet, pl. hisztogram, boxplot, scatterplot, stb.
- ami nagyon más: nem elég, ha mi értjük, hogy mi van az ábrán
- értelmes ábracím, tengelyek rendes elnevezése, skála mérete, informatív képaláírás
- laboron majd nézzük ezt R-ben

Cél

- kevesebb oszlop legyen: oszlopok elhagyása, összevonása, új (jobb) feature-ök vezetetés régiek elhagyása mellet
- sorok számának csökkentése, sorok felosztása training és test (és esetleg validation) halmazra
- mindezt azért, hogy
  - gyorsabban fusson le az algoritmus
  - jobb legyen az eredmény (kifejezőbb attribútumok)

Az előfeldolgozás részei

- feature subset selection: oszlopszámot csökkent viszonylag triviális módon
- aggregáció: összevonás, célja az oszlopszám csökkentése
- mintavételezés (sampling): célja a sorok számának csökkentése
- dimenziócsökkentés: kisebb mátrix legyen, oszlopok számának csökkentése, de nem összevonással
- új attribútumok bevezetése: feature creation (de közben csökken az oszlopszám, ennek spec. esete a dimenziócsökkentés)
- diszkretizálás, binárisra átírás: az oszlop típusát változtatja meg
- attribútumok transzformálása máshogy: skálázás, standardizálás

Nem feltétlenül ez a sorrend és nem is kell mindig minden.

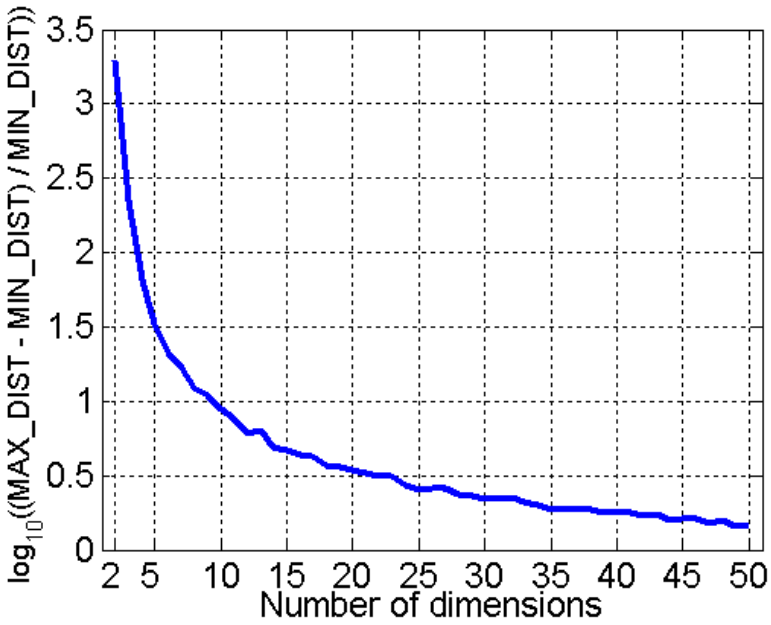
Dimenziócsökkentés: miért?

- ha nagy a dimenzió, akkor
  - lassúak lehetnek az algoritmusok
  - vagy nem is működnek jól
  - meg sok hely is kell az adatok tárolására
- ha kisebb diemnzióban dolgozunk, akkor könnyebb (lehetséges egyáltalán) ábrázolni az adatokat
- tranzakciós és dokumentum mátrixoknál óriási dimenziószám van
- ez azért is baj, mert nagy dimenzióban a pontok közötti eltérések nem különülnek el nagyon

Ez a curse of dimensionality.

Curse of Dimensionality

- When dimensionality increases, data becomes increasingly sparse in the space that it occupies
- Definitions of density and distance between points, which is critical for clustering and outlier detection, become less meaningful



- **Randomly generate 500 points**
- **Compute difference between max and min distance between any pair of points**

Dimenziócsökkentés, módszerek

- lineáris algebrai módszerek, automatikus, R-ben is van
  - a régi attribútumok valami lineáris kompozíciójaként állnak elő az új attribútumok
  - főkomponens analízis: PCA (Principal Component Analysis)
  - szinguláris érték felbontás: SVD (Singular Value Decomposition)
- más módszerek: nem automatizáltak
  - supervised: emberi beavatkozással hozunk létre új változókat, háttértudás birtokában
  - nem-lineáris technikák: az új attribútumok a régiekből állnak elő, de nem lineáris kombinációval

Cél mindig az, hogy kevesebb attribútum legyen a végén.

Feature subset selection: triviális(?) rész

- redundáns oszlopok felismerése
- például eladott termék ára, befizetett ÁFA (amennyiben uaz az áfakulcs minden terméknél, akkor az egyik nem kell)
- irreleváns oszlopok felismerése
- pl. neptun kód releváns, ha következő féléves átlagot akarunk előre jelezni
- ha jó dokumnetáció van és ismerjük a környezetet, ahonnan az adat jön, akkor ez nem nehéz
- emberi feladat, nem (nagyon) lehet automatizálni

Feature subset selection: alaptechnika

- cél: a triviális szűrés utáni attribútumoknak csak egy részét tartsuk meg
- gyorsabb/jobb legyen az elemzés az új attribútum halmazzal
- futtassuk a használni kívánt adatbányászati algoritmust egy mintán az eredeti és a potenciális szűkebb oszlophalmazzal
- nézzük meg, hogy elromlott-e az eredmény illetve mi történt a sebességgel
- döntsük el, hogy megéri-e a csökkentett attribútumhalmaz

Feature subset selection: módszerek

- brute-force: nézzük meg minden részhalmazát az attribútumhalmaznak: ez nem nagyon járható, már n, az attribútumok száma is nagy, 2^n óriási
- beágyazott módszer: a használt adatbányászati algoritmus majd kiválogatja a fontosakat (döntési fák pl.)
- automatikus szűrés: az algoritmus futása előtt valahogy szűrünk, pl. ha két oszlop korrelációja valami adott értéknél nagyobb, akkor egyiket eldobjuk
- valahogyan (ember?/automatizmus) generálok esélyes részhalmazokat és ezeket tesztelem kis mintán
  - csökkentem egyesével az attibútumok számát, amíg valami STOP-feltétel miatt le nem állok ezzel
  - egy legfontosabb(nak tűnő) attribútummal kezdve egyre többwet veszek be, amíg elég jó nem lesz az elemzés

Aggregáció

- valami csoportosítás alapján összegzem a számokat
- ha az adatsorok azt tartalmazzák, hogy melyik város, melyik üzlete, mennyi bevételt produkált egy napon
  - aggregálhatok városra: adott városbeli bevétel egy napon, városok közti összefüggések
  - aggregálhatok időtartamra: boltok havi bevételei, jobban látszanak a boltok közötti sorrendek
- kérdések: mi alapján vonok össze, mit összegzek

Aggregáció haszna

- kevesebb sor lesz
- átláthatóbb, esetleg ábrázolhatóbb adatok (kevesebb dimenzió lesz, hatékonyabban lehet ábrázolni)
- stabilabb adatok, tendenciák jobban látszódnak

Mintavételezés

- lehet az adatgyűjtés része is (mikrocenzus)
- az ismerkedéskor is jól jöhet: könnyebben áttekinthető, hogy mivel van dolgunk
- a különböző módszerek tesztelésére elengedhetetlen: nem akarunk minden módszert az egész halmazon lefuttatni
- magában is érdekes lehet, ha túl sok az adat és drága vagy lassú feldolgozni

Mintavételezés alapfeltevései

- olyan minta kell, ami jól reprezentálja a teljes halmazt: reprezentatív
- honnan tudjuk, hogy ilyen-e?
  - amikor kábé ugyanaz az eredmény, következtetés, bármi, amiért az egész eljárást csináljuk hasonló a mintán és az egészen
  - ez nem valami egzakt
  - vannak ennek tesztelésére is technikák (nagy terület)

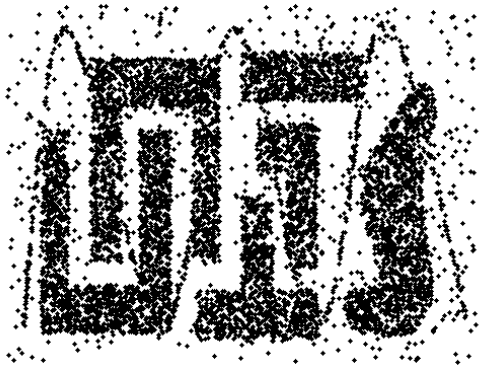
Mintavételezés típusai

- egy lehetséges felosztás:
  - egyenletes eloszlás szerinti random mintavételezés: minden elem ugyanakkora valószínűséggel kerül be, akkor jó, ha homogén az adatbázis, de ilyenkor sem árt egy permutálás a választás előtt
  - több részre osztani a mintát, minden részből választani véletlenszerűen
- visszatevéses-e?

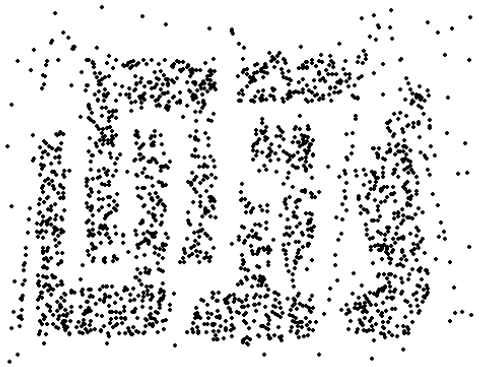
Minta mérete

- nyilván ne legyen nagyon nagy (összemérhető az eredetivel), mert akkor minek csináljuk
- de azért elég nagynak kell lennie ahhoz, hogy jól reprezentáljon
- ha van valami mintázat az adatokon, akkor az látszódjon a mintán is
- egy módszer a progresszív sampling: növelni a minta méretét, amíg az elég jó lesz, pl. predikció minősége szerint

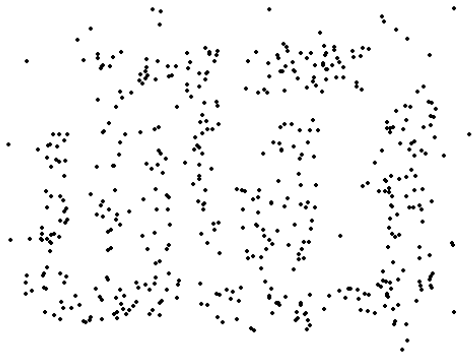
Sample Size



8000 points



2000 Points



500 Points

PCA és SVD

- mindkettő lineáris algebrai módszer
- vektorok a sorok, eredetileg egy n dimenziós térben
- az egyes oszlopok a dimenzióknak felelnek meg
- cél olyan koordinátarendszert találni valami alacsonyabb dimenzióban, amire levetítve a vektorokat (azaz sorokat) kevés az információvesztés
- ennek az alacsonyabb koordinátarendszernek a vektorai lesznek az új attribútumok
- így kisebb helyen elférnek az adatok (bár információvesztés van)
- felgyorsíthatja az algoritmusokat, ha kevesebb a paraméter

PCA

- a kovariancia mátrix sajátvektorait keressük meg (ennek mindig van oszlopszámnyi sajátvektora), ezek lesznek az új attribútumok
- az új dimenzió az lesz, hogy ezeket sajátérték alapján csökkenő sorrendbe téve hányat választok belőlük
- általában ez lassú, ha nagy a mátrix, de utána jól használható kisebb mátrix jön létre
- SVD hasonló céllal, kicsit más módszerrel talál hasonló tulajdonságú vektorokat
- R-ben svd() függvény jól használható

Új attribútumok bevezetése

- nem feltétlenül kevesebb attribútum létrehozása a cél
- általános cél: olyan új attribútumhalmazt találni, ami jobban használható
- sokszor (mindig ?) emberi feladat, háttértudás kell hozzá
- fajtái:
  - feature extraction: pl. képfeldolgozásnál a pixelek adatait tartalmazó nyers adatból: van-e rajta ember, van-e ilyen vagy olyan kontúr, stb. ehhez ember, vagy ember alkotta spéci algoritmus kell
  - attribútumok kombinálása háttértudással: tömeg és térfogat helyett sűrűséggel dolgozni

Diszkretizálás

- Célja: folytonos változót diszkrétté alakítani
- ez kellhet, ha
  - olyan algoritmust akarunk futtatni, amihez diszkrét értékű változók kellenek, pl. asszociációs szabályok kutatása, bizonyos típusú döntési fák készítése
  - nem akarunk sok értéket nyilvántartani csak a nagyobb kategóriák a fontosak: magas, közepes, alacsony értékek
- minden értéket valami kategóriába akarunk sorolni
- lehetnek diszjunkt vagy átfedő kategoriák (felhasználástól függően)
- kérdés, hogy hogyan alakítjuk ki a csoportokat

Diszkretizálás, hogyan?

Kérdés, hogy mire kell a diszkretizálás:

- ha az exploratory elemzés része (más-e a tendencia alacsony és magas értékek körében), akkor nem érdemes nagyon szofisztikált módszert használni
- ha a diszkretizálásra alapozunk valami algoritmust, akkor fontos lenne jól csinálni

Általában jól jön az adatok hátterének ismerete, valami szakértő véleménye.

Diszkretizálás, hogyan?

- egyenlő darabszámú csoportokat létrehozva (általában nem jó)
- a folytonos változó értékészletét egyenletesen felosztva csoportosítani az elemeket (ez se biztos, hogy jó)
- lehet klaszterezni és a klaszterek azonosítói lesznek a diszkrét változó lehetséges értékei (jobb, de macerás: sok idő, klaszterszámot nem ismerjük mindig)

Binárissá átírás

- a diszkretizálás után jön, előbb diszkrét értékű változót kell létrehozni
- asszociációs szabályokhoz elengedhetetlen
- módszere: minden lehetséges diszkrét értékre egy változó, ami vagy igaz vagy hamis lehet
- így egy k lehetséges értékű diszkrét változóhoz k új bináris változót kell legyártani
- az i. változó értéke pontosan akkor 1, ha az adott sorban az eredeti változó értéke i volt

Attribútumok transzformálása

Amikor már minden szép, az adatok rendben vannak, csak az a baj, hogy

- nem tudjuk jól ábrázolni, mert pl. vannak outlierek, amik miatt az ábra nagyon deformált lesz
- nem azonos skálán vannak az oszlopok: gyerekek száma vs. fizetés forintban

Valami bijektív függvényt alkalmazunk: log, kivonás, osztás (normalizálás speciális eset).