Jazyky pre Dátovú Analytiku

Prednáška č.2

Vizualizácia dát a získaných modelov

- Dôležitou časťou procesu analýzy dát je vizualizácia
  - Samotných vstupných dát (+ tvorba jednoduchých modelov) = jednoduché vizuálne analýzy, pochopenie dát
  - Štatistických modelov o dátach
  - Aplikácie modelov v reálnom prostredí
- Z pohľadu nástrojov
  - Prostriedky pre tvorbu „grafov“ (grafických = vizuálnych výstupov, „plots“) a ich popisov, výsledkom sú grafy:
    - Statické alebo dynamické
    - Interaktívne alebo neinteraktívne

Princípy analytickej grafiky

- P1: Ukázať porovnania
  - Dôkaz pre hypotézu je vždy relatívny voči inej zvažovanej hypotéze (vždy sa pýtame ku čomu porovnávame)
- P2: Ukázať kauzalitu, mechanizmus, vysvetlenie, systematickú štruktúru
  - Pýtame sa aká je príčina toho, že sa danú otázku pýtame?
- P3: Ukázať viacrozmerné dáta (viac ako 2 premenné)
  - Reálny svet je popísaný viacerými atribútmi, potrebujeme získať nadhľad
- P4: Integrovať dôkaz
  - Prezentácia by mala integrovať slová, čísla, obrázky, diagramy; mali by sme využiť viacero možností prezentácie dát
  - Nástroj by nemal riadiť analýzu (snažiť sa vyťažiť viac z integrácie výsledkov)
- P5: Dokumentovať dôkaz so správnymi popiskami, škálami, zdrojmi, ...
  - Grafika by mala poskytnúť kompletný pohľad na dôležité aspekty
- P6: Obsah je rozhodujúci
  - Úspešnosť prezentácie analýzy závisí na kvalite, relevancii a integrite dátového obsahu

Exploratívna analýza

- Cieľom prezentácie grafov v analýze je
  - Pochopiť dátové atribúty
  - Nájsť vzory v dátach
  - Navrhnúť stratégiu ďalšieho modelovania dát
  - Vyladiť analýzy
- Charakteristiky exploratívnych grafov (~ explor. analýzy)
  - Vytvárajú sa rýchlo (princíp „quick and dirty“)
  - Zvyčajne je ich urobené veľké množstvo
  - Cieľom je pochopenie dát a ich vzťahov, zvýraznenie hlavných vlastností, explorácia základných otázok a hypotéz, odporučiť ďalšie kroky analýzy
  - Legendy/popis osí sú vo všeobecnosti očistené neskôr
  - Farba/veľkosť sú primárne pre poskytnutie informácie

Jednoduché sumarizácie dát

- Jednodimenzionálne
  - Sumarizácia 5 čísel = summary() ... Min, 1stQu, Median, 3rd Qu, Max // početnosti pre kategor. atr.
  - Boxplot
  - Histogramy
  - Stĺpcové grafy (barplot)
  - Spojité grafy (density plot)
- Dáta
  - Znečistenie vzduchu v USA (Air pollution data) – konkrétne časť dát o počte častíc (PM2.5) – údaje o ročnej strednej početnosti + z toho priemer za 3 roky
  - Podľa EPA – tento by nemal presiahnuť 12 mikrogramov na meter kubický
  - Otázka: Sú oblasti kde je táto hodnota v USA prekročená ?

Dáta

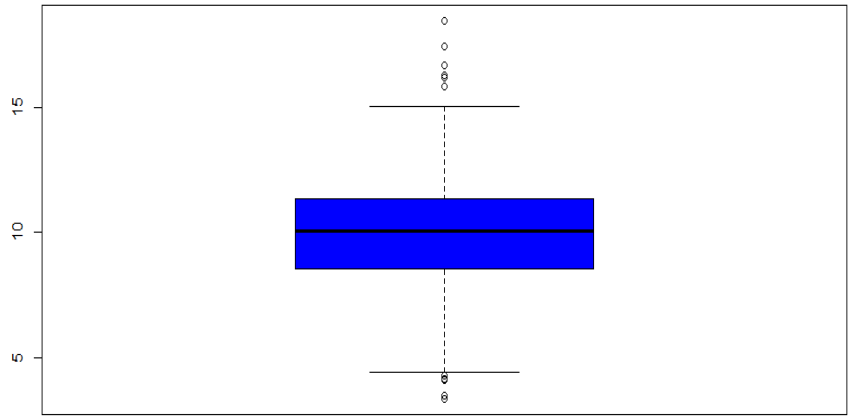
- Ročný priemer PM2.5 + cez obdobia 2008 -2010
- http://web.tuke.sk/fei-cit/butka/res/avgpm25.csv

```
> download.file("http://people.tuke.sk/peter.butka/res/avgpm25.csv",
    "avgpm25.csv")
> pmdata = read.csv("avgpm25.csv", colClasses = c("numeric",
    "character","factor", "numeric", "numeric"))
> head(pmdata)
       pm25  fips region longitude latitude
1  9.771185 01003   east  -87.74826 30.59278
2  9.993817 01027   east  -85.84286 33.26581
3 10.688618 01033   east  -87.72596 34.73148
4 11.337424 01049   east  -85.79892 34.45913
5 12.119764 01055   east  -86.03212 34.01860
6 10.827805 01069   east  -85.35039 31.18973
```

Príklady 1D explorácie dát

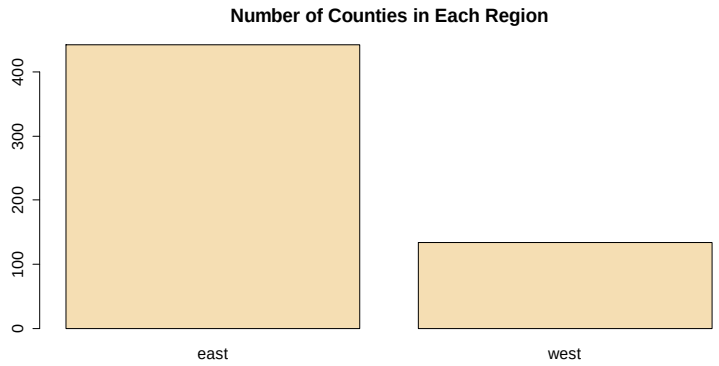
- „Sumár piatich čísel“

```
> summary(pmdata$pm25)
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
  3.383   8.549  10.047   9.836  11.356  18.441
```



- Boxplot

```
> boxplot (pmdata$pm25,
  col = "blue")
```



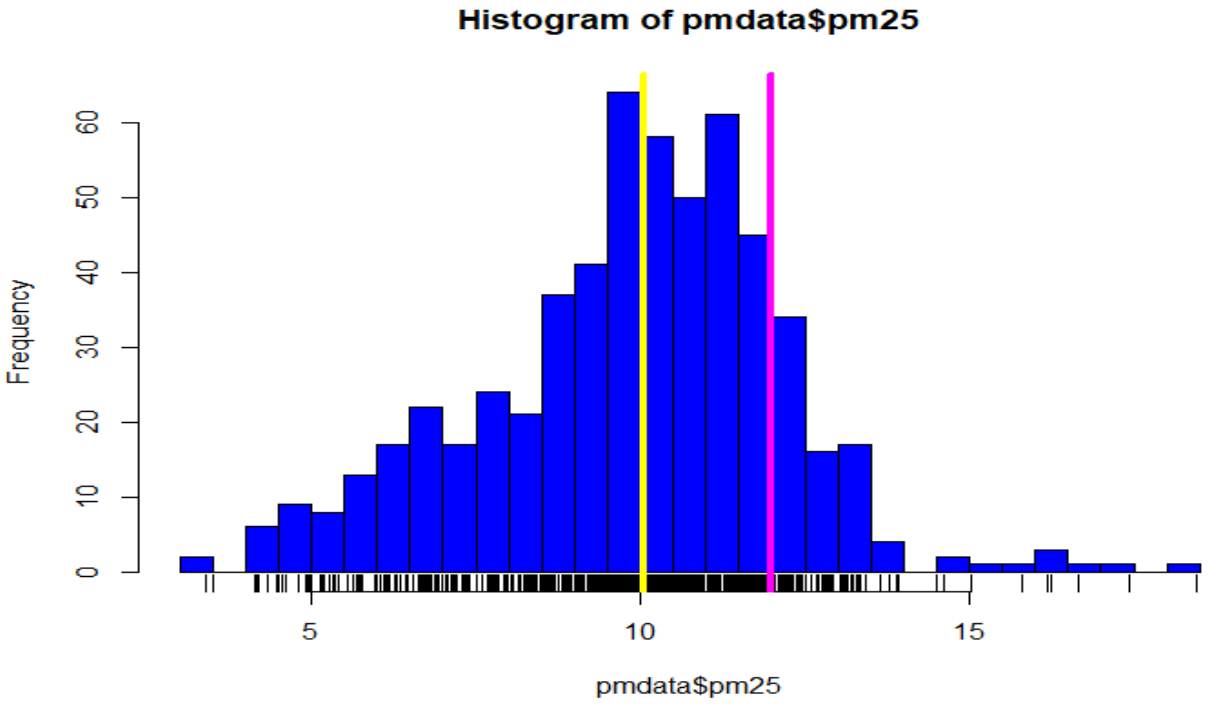
- Barplot (stĺpcový graf)

```
> barplot(table(pmdata$region), col
  = "wheat", main = "Number of
  Counties in Each Region")
```

Príklady 1D explorácie dát (2)

- **Histogram**

```
> hist(pmdata$pm25, col="blue", breaks=50)
> rug(pmdata$pm25)
> abline(v = 12, lwd = 4, col="magenta")
> abline(v = median(pmdata$pm25), col = "yellow", lwd = 4)
```



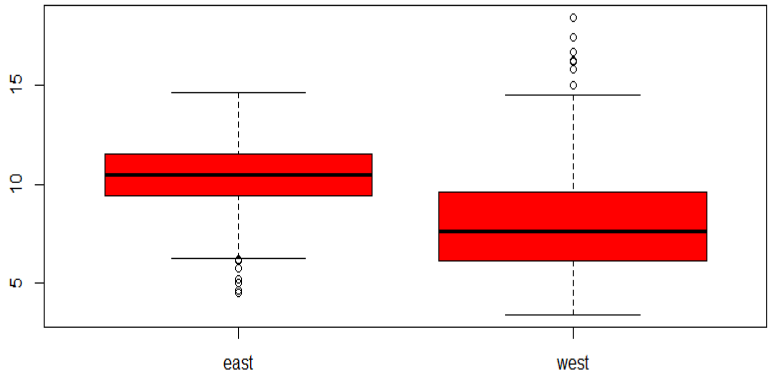
Explorácia dát pre >= 2D

- 2 dimenzie
  - Viaceré/prelínajúce sa 1-D grafy (balíky lattice/ggplot2)
  - Bodový graf (korelačný diagram ... scatterplot)
  - Spojitý graf („hladký“ bodový, smooth scatterplot)
- > 2 dimenzie
  - Prelínajúce sa /viaceré 2-D grafy; kografy
  - Použitie farieb, veľkosti, tvarov pre vizualizáciu ďalších dimenzií
  - Otáčajúce (rotujúce) grafy
  - Skutočné 3-D grafy

Príklady >=2D explorácie dát

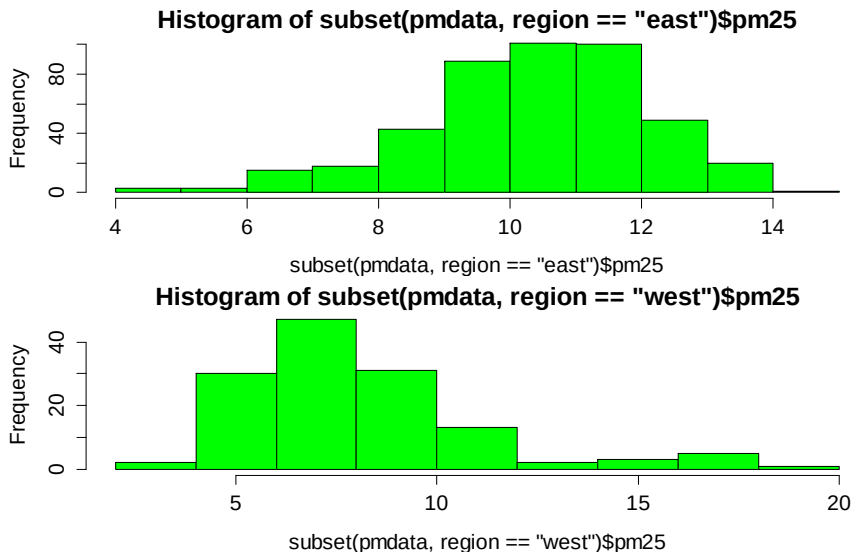
- Viacero boxplotov

```
> boxplot(pm25 ~ region,
    data = pmdata, col =
    "red")
```



- Viacero histogramov

```
> par(mfrow = c(2, 1), mar = c(4, 4, 2, 1))
> hist(subset(pmdata, region == "east")$pm25,
    col = "green")
> hist(subset(pmdata, region == "west")$pm25,
    col = "green")
```

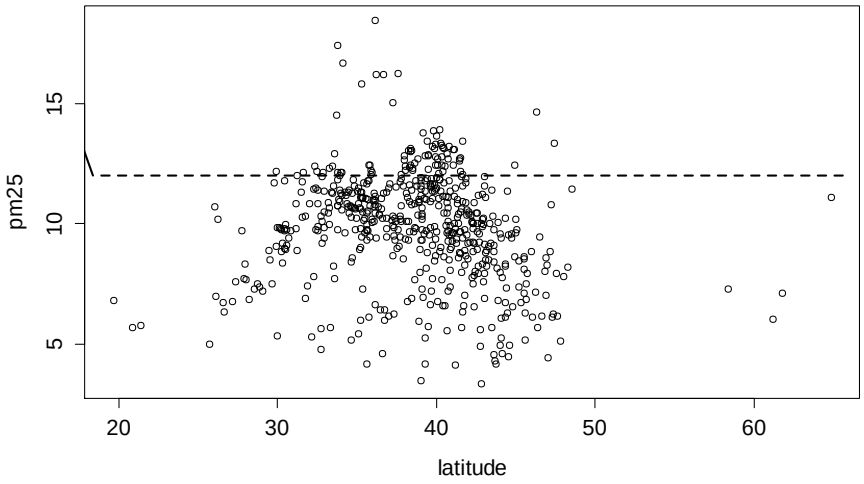


*** par() … funkcia pre nastavenie grafických parametrov ... viď. dokumentácia ***

Príklady >=2D explorácie dát (2)

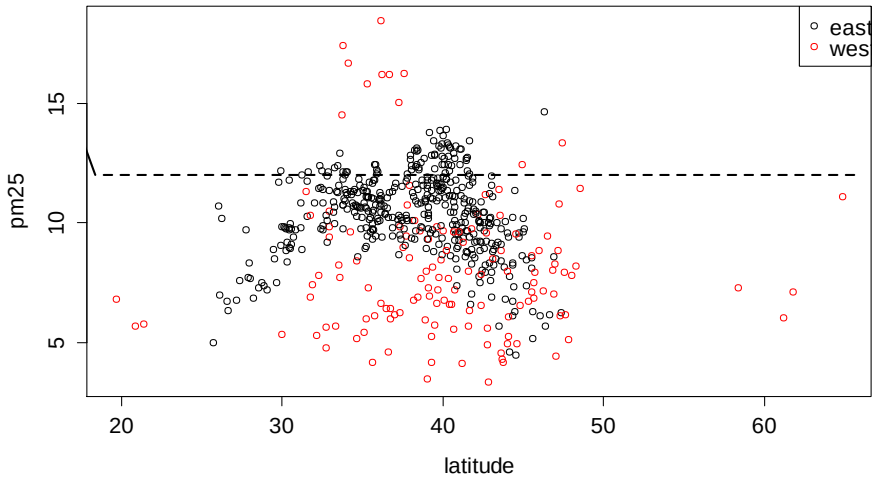
- Bodový graf (scatterplot)

```
> with(pmdata, plot(latitude,
    pm25))
> abline(h = 12, lwd = 2,
          lty = 2)
```



- Bodový graf s použitím farby (pre region atribút)

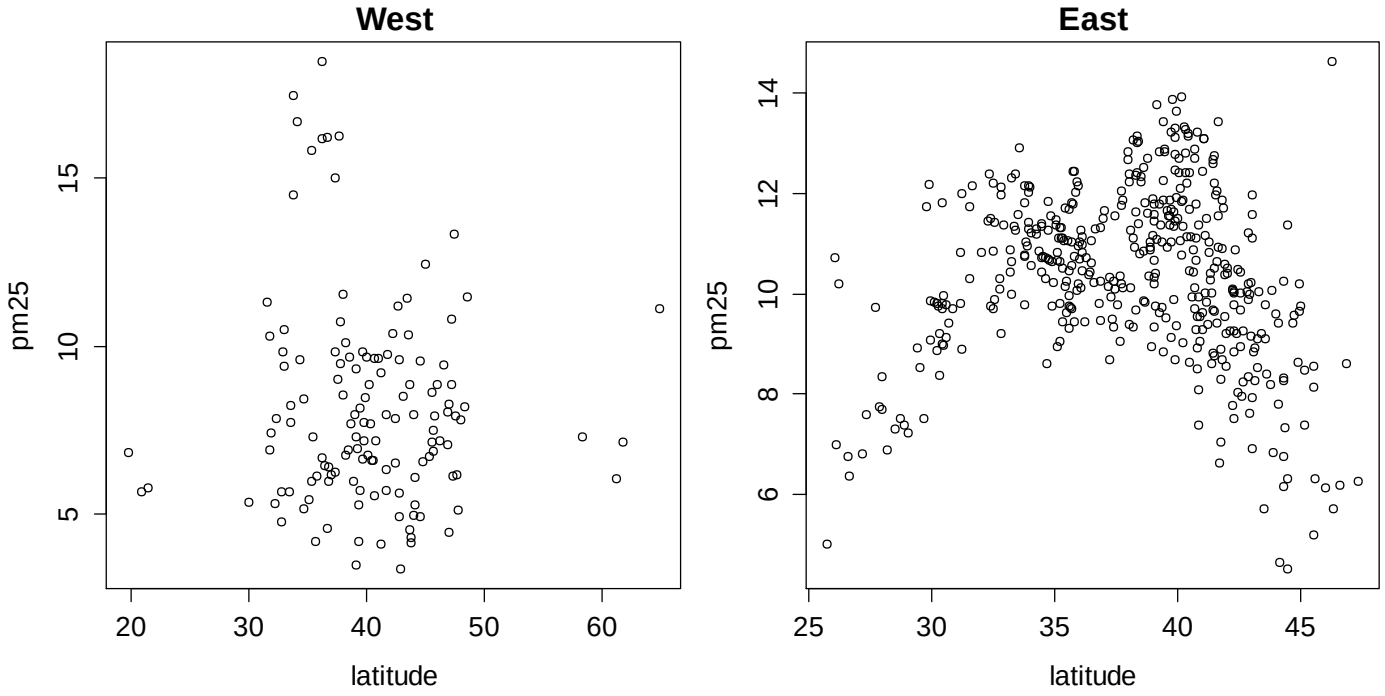
```
> with(pmdata, plot(latitude, pm25,
    col = region))
> abline(h = 12, lwd = 2, lty = 2)
> legend(x="topright", legend =
    levels(pmdata$region),
    col=c("red","black"), pch=1)
```



Príklady >=2D explorácie dát (3)

- Viacero bodových grafov (napr. namiesto použitia farby)

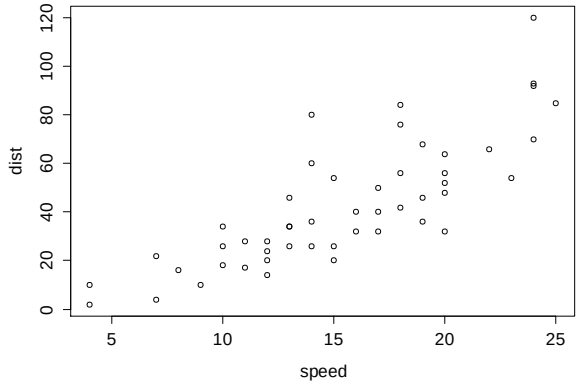
```
> par(mfrow = c(1, 2), mar = c(5, 4, 2, 1))
> with(subset(pmdata, region == "west"), plot(latitude, pm25,
    main = "West"))
> with(subset(pmdata, region == "east"), plot(latitude, pm25,
    main = "East"))
```



Systémy vykresľovania v R – Base

- Základný systém (Base)
  - Model kresliaceho plátna, začína prázdnym priestorom a v tom postupne buduje grafický výstup
  - Spúšťa sa plot funkciou (alebo podobnou), používa anotačné funkcie pre pridávanie / modifikáciu (textov, čiar, bodov, osí)
  - Odzrkadľuje spôsob ako premýšľame pri budovaní výstupu a analýze dát
  - Nie je možné sa vrátiť po vykreslení (napr. meniť ohraničenia) -> je potrebné plánovať dopredu
  - Je zložité transformovať kód na iný systém keď sa vytvorí nový výstup (nemá grafický „jazyk“)
  - Vykreslenie je vlastne séria R príkazov
- Príklad

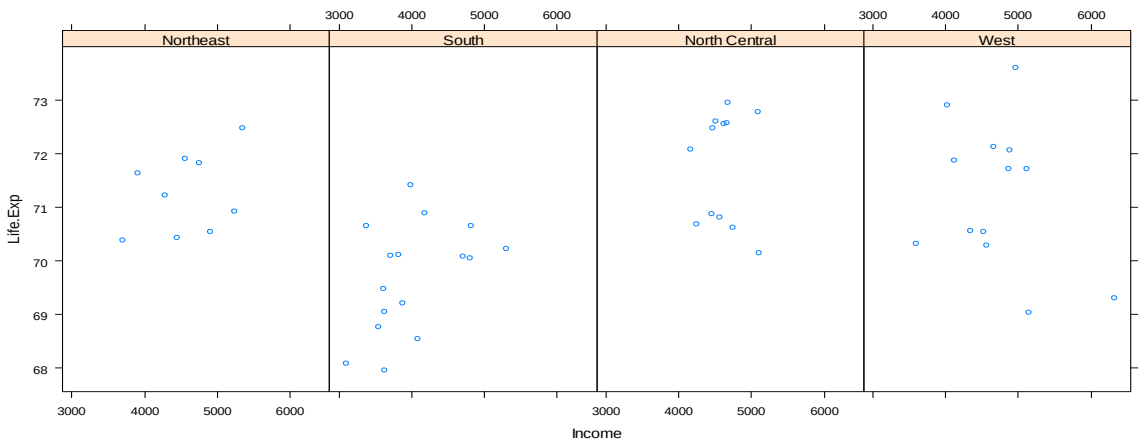
```
> library(datasets)
> data(cars)
> with(cars, plot(speed, dist))
```



Systém vykresľovania v R - lattice

- Grafy sa vytvárajú jedným volaním funkcie (xyplot, bwplot, ...)
- Užitočné najmä pre podmienené typy grafov (hľadanie ako sa mení závislosť x od y pre rôzne z); dobré pre vloženie viacerých grafov
- Hranice/vyplnenie priestoru sú automatické (keďže ide o jedno volanie), anotácia nie je veľmi intuitívna
- Niekedy je zložité špecifikovať všetko v jednom volaní; potom už však nie je možné pridávať / meniť
- Príklad:

```
> library(lattice)
> state = data.frame(state.x77,
region = state.region)
> xyplot(Life.Exp ~ Income |
region, data = state, layout =
c(4, 1))
```

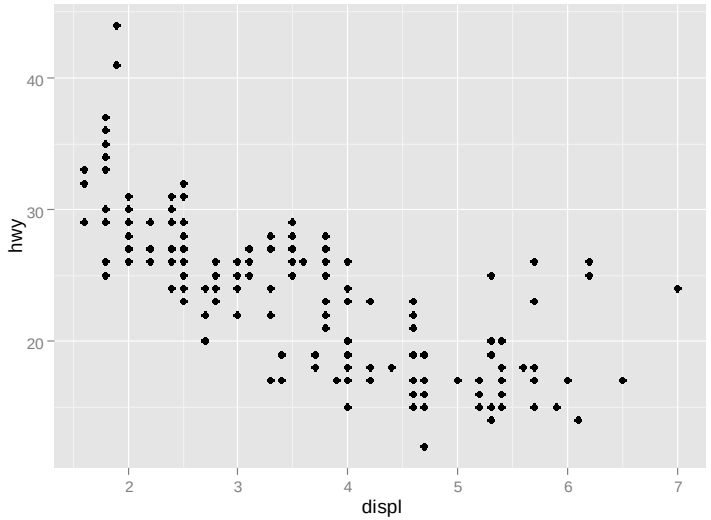


Systém vykresľovania v R – ggplot2

- Kompromis medzi base a lattice => odstraňuje viaceré rozdiely medzi nimi
- Rieši automaticky vypĺňanie priestoru, textov, názvov, ale umožňuje použiť anotácie na doplnenie grafu
- Základná podobnosť s lattice, ale vo všeobecnosti jednoduchší a intuitívnejší
- Default mód urobí viaceré rozhodnutia za používateľa (ale je možné upraviť podľa potreby)

Príklad:

```
> library(ggplot2)
> data(mpg)
> qplot(displ, hwy,
data = mpg)
```



Výstupné grafické zariadenie

- je miesto kde sa objaví / vykreslí výstupný graf
  - Zoznam možných zariadení v R – viď ?Devices
    - Okno na obrazovke (zariadenie = device je screen) ... vhodné pre exploratívnu analýzu
    - PNG, JPEG, PDF, SVG súbory – (file device) ... vhodné pre reportovanie, tlačené výstupy, prezentácie
- Najčastejšie sa používa v default prípade screen
  - Funkcie ako plot v base, xyplot v lattice, či qplot v ggplot2 majú default odoslanie na screen
- Súborové zariadenia
  - Vektorová grafika (*vector* device) – PDF, SVG, Win.metafile, PS (-> EPS)
  - Bitmapy (*bitmap* device) – PNG, JPEG, TIFF, BMP (Win bitmapa)

Základy grafického systému Base

- Najčastejšie používaná, účinná pre 2-D grafiku
- 2 fázy vytvorenia grafu
  - Inicializácia nového grafického výstupu (plot)
  - Anotácia existujúceho grafického výstupu
- Volanie plot() alebo hist() spustí grafické zariadenie (ak nejaké už nie je otvorené) a vykreslí graf na zariadení
  - Príklady: viď. predchádzajúce v explorácii dát (histogram, scatterplot, boxplot, barplot, ...)
- Ak argumenty nie sú špeciálne triedy, volá sa default verzia plot funkcie – má viacero argumetov, umožňuje nastaviť, názov, pomenovania osí, atď.
- Base systém má mnoho parametrov, ktoré môžu byť nastavené a doladené – sú zdokumentované v ?par

Parametre vykresľovania prvkov

- Mnoho vykresľovacích funkcií zdieľa množinu parametrov, niektoré z nich sú:
  - pch: symbol bodu (default - otvorený krúžok)
    - http://www.endmemo.com/program/R/pchsymbols.php
  - lty: typ čiary (default – plná čiara) - môžu byť bodkované, prerušované, ...
  - lwd: hrúbka čiary – integer – default = 1
  - col: farba – ako string, číslo, alebo hex kód – colors() dáva vektor farieb po názvoch, dá sa použiť na farebné odlíšenie bodov podľa faktorov / hodnôt – default farba = black
  - xlab: reťazec popisky pre x-ovú os
  - ylab: reťazec popisky pre y-ovú os
- Ďalšie dôležité parametre sú súčasťou par(), predstavujú globálne parametre (môžu byť prepísané špecifikáciou parametra plot funkcie)
  - las: orientácia popisov osí v grafe
  - bg: farba pozadia (default - transparent)
  - mar: nastavenie okrajov grafu (bottom, left, top, right) – def: 5.1 4.1 4.1 2.1
  - oma: veľkosť vonkajšieho okraja (default 0 pre všetky strany)
  - mfrow: počet grafov na riadok, stĺpec (grafy sa vkladajú po riadkoch)
  - mfcol: počet grafov na riadok, stĺpec (grafy sa vkladajú po stĺpcoch)

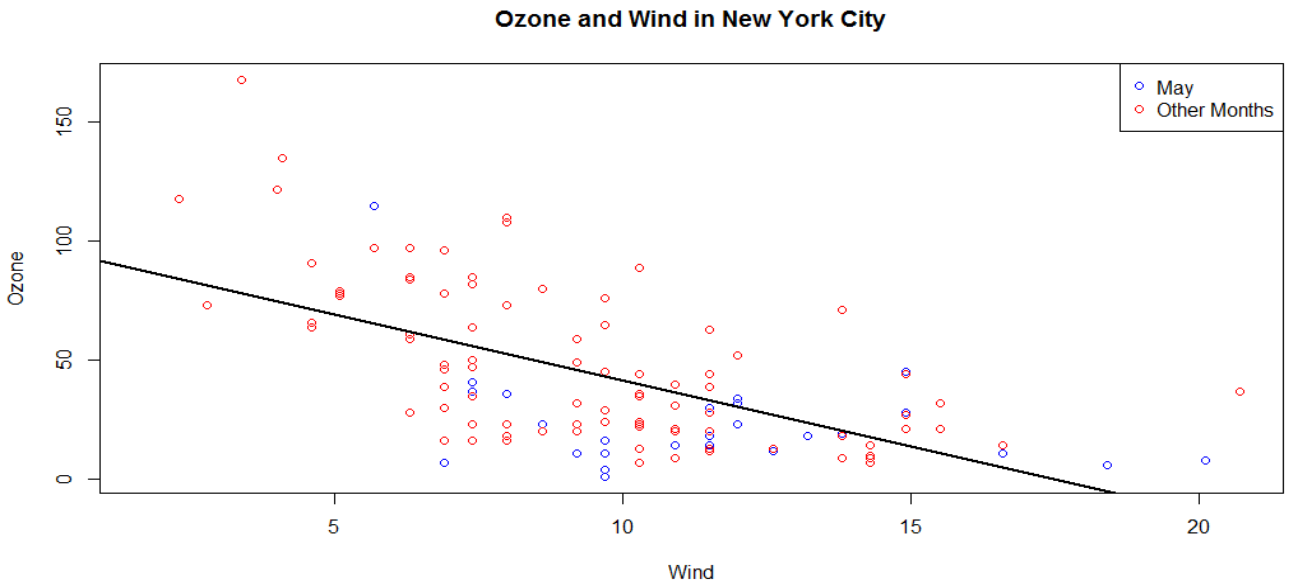
Základné vykresľovacie funkcie Base

- plot: vykresľuje scatterplot, alebo iný typ podľa triedy objektu ktorý má byť vykreslený (často sa kombinuje s dátami cez *with*)
- hist: vykresľuje histogram
- lines: pridáva čiary do grafu, k zadanému vektoru x,y hodnôt (resp. 2-stĺpcovej matice hodnôt) – funkcia spojí body
- points: pridá body do grafu
- abline: pridá rovnú čiaru do grafu
- text: pridá text ku grafu pomocou špecifikovaných x,y súradníc
- title: pridá anotácie ku osám, názvu, podnázvu, vonkajším okrajom
- legend: pridáva legendu k grafu
- mtext: pridá ľubovoľný text k okrajom grafu (vnútorným alebo vonkajším)
- axis: pridá popisy / značky k osiam

Príklady Base (1)

- Base graf s anotáciami + regresná priamka

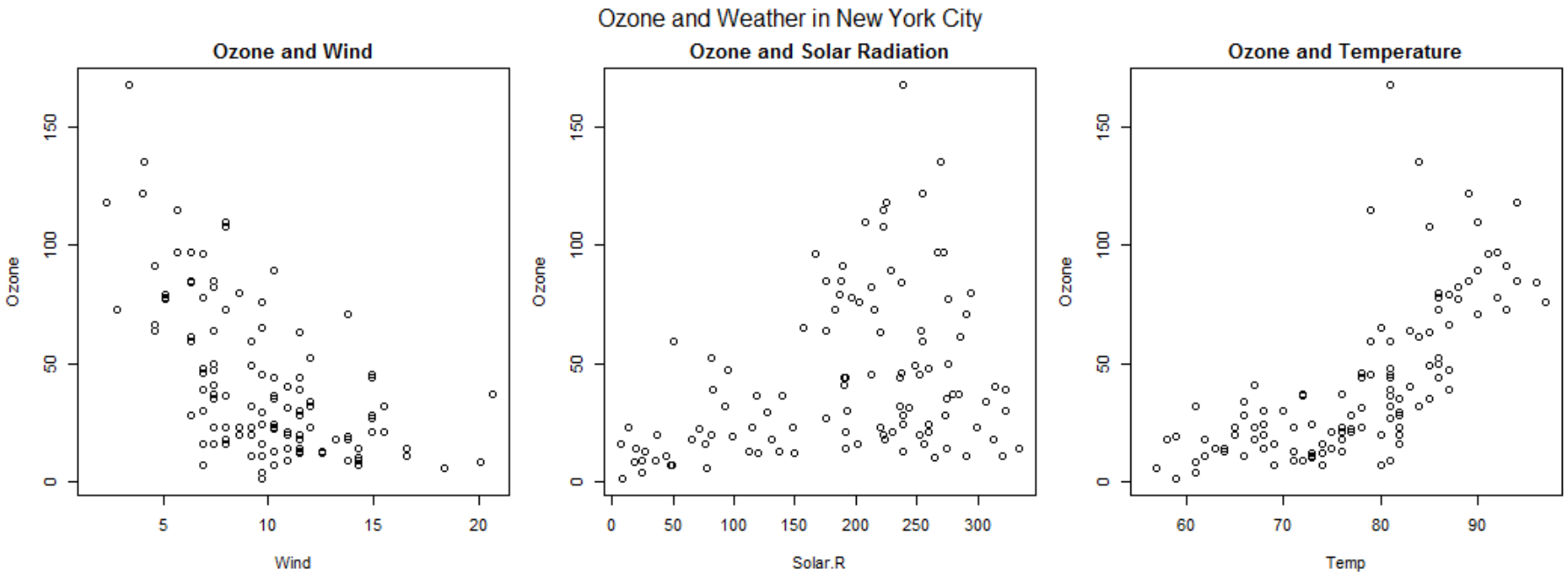
```
> with(airquality, plot(Wind, Ozone, main = "Ozone and Wind in New York City ",
    type = "n"))
> with(subset(airquality, Month == 5), points(Wind, Ozone, col = "blue"))
> with(subset(airquality, Month != 5), points(Wind, Ozone, col = "red"))
> legend("topright", pch = 1, col = c("blue", "red"), legend = c("May", "Other
    Months"))
> model = lm(Ozone ~ Wind, airquality)
> abline(model, lwd = 2)
```



Príklady Base (2)

- Kombinácia viacerých Base grafov s anotáciami

```
par(mfrow = c(1, 3), mar = c(4, 4, 2, 1), oma = c(0, 0, 2, 0))
with(airquality, {
  plot(Wind, Ozone, main = "Ozone and Wind")
  plot(Solar.R, Ozone, main = "Ozone and Solar Radiation")
  plot(Temp, Ozone, main = "Ozone and Temperature")
  mtext("Ozone and Weather in New York City", outer = TRUE)
})
```



Základy grafického systému Lattice

- Lattice systém vykresľovania je implementovaný pomocou nasledujúcich balíkov:
  - *lattice:* obsahuje kód pre produkciu Trellis grafiky, ktorá je nezávislá od grafického systému Base; zahŕňa funkcie ako xyplot, bwplot, levelplot, ...
  - *grid:* implementuje odlišný grafický systém nezávislý od base systému; lattice balík je postavený na jeho základe
- Lattice systém nemá dvojfázové vytváranie ako Base => všetky vykresľovania a anotácie sa dejú pri jednom volaní
  - Z toho vyplýva aj fakt, že aspekty ako okraje a vyplnenie priestoru sú automaticky spracované a default verzia je zvyčajne postačujúca
  - Ideálne pre podmienené grafy, kde sa skúmajú rôzne podmienky (vlastnosti) v dátach zobrazením väčšieho počtu rovnakého typu grafov
  - Panelové funkcie v Lattice systéme (ako sa zobrazujú viaceré grafy ako časť jedného súboru grafov) môžu byť špecifikované resp. kustomizované pre modifikáciu jednotlivých zobrazení na paneloch (subčastiach zobrazovanej plochy)

Základné vykresľovacie funkcie Lattice

- xyplot: hlavná funkcia pre vytváranie bodových grafov (scatterplot)
- bwplot: vytvára boxplot grafy
- histogram: vytvára histogramy
- stripplot: odpovedá boxplot-u, avšak s konkrétnymi bodmi
- dotplot: zobrazí graf s bodmi v pravidelnej štruktúre podľa hodnôt atribútov (body ležia ako keby na „strunách“ odpovedajúcich úrovniam iného atribútu)
- splom: scatterplot matica kombinácie atribútov
- levelplot, contourplot: pre zobrazenie obrázkových dát

Použitie Lattice funkcií

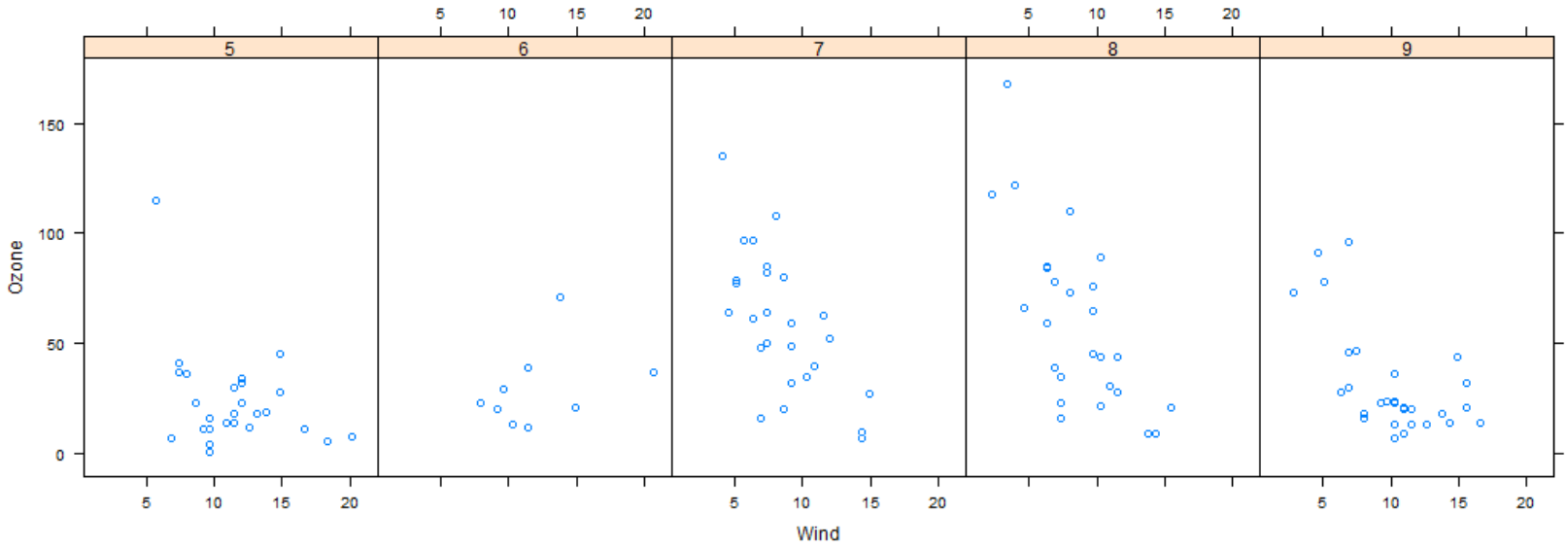
- Lattice funkcie si vo všeobecnosti berú ako prvý vstup (argument) formulu, zvyčajne napríklad v tvare:

  xyplot(y ~ x | f * g, data)
- Používa sa notácia pre *formula* objekty (s ~)
- Na ľavej strane ~ je premenná pre y-ovú os, na pravej premenná pre x-ovú os
- f a g sú podmieňujúce premenné (voliteľné nastavenie)
  - * indikuje interakciu medzi týmito premennými
- Druhý argument je *data frame* alebo *list* z ktorého sú premenné preberané
  - Ak nie je zadaný *data frame* alebo *list*, použije sa tzv. parent frame
- Ak nie sú uvedené ďalšie argumenty, sú použité default nastavenia

Príklad grafu cez Lattice

- Príklad:

```
> library(datasets)
> library(lattice)
> airqt = transform(airquality, Month = factor(Month))
> xyplot(Ozone ~ Wind | Month, data = airqt, layout = c(5, 1))
```

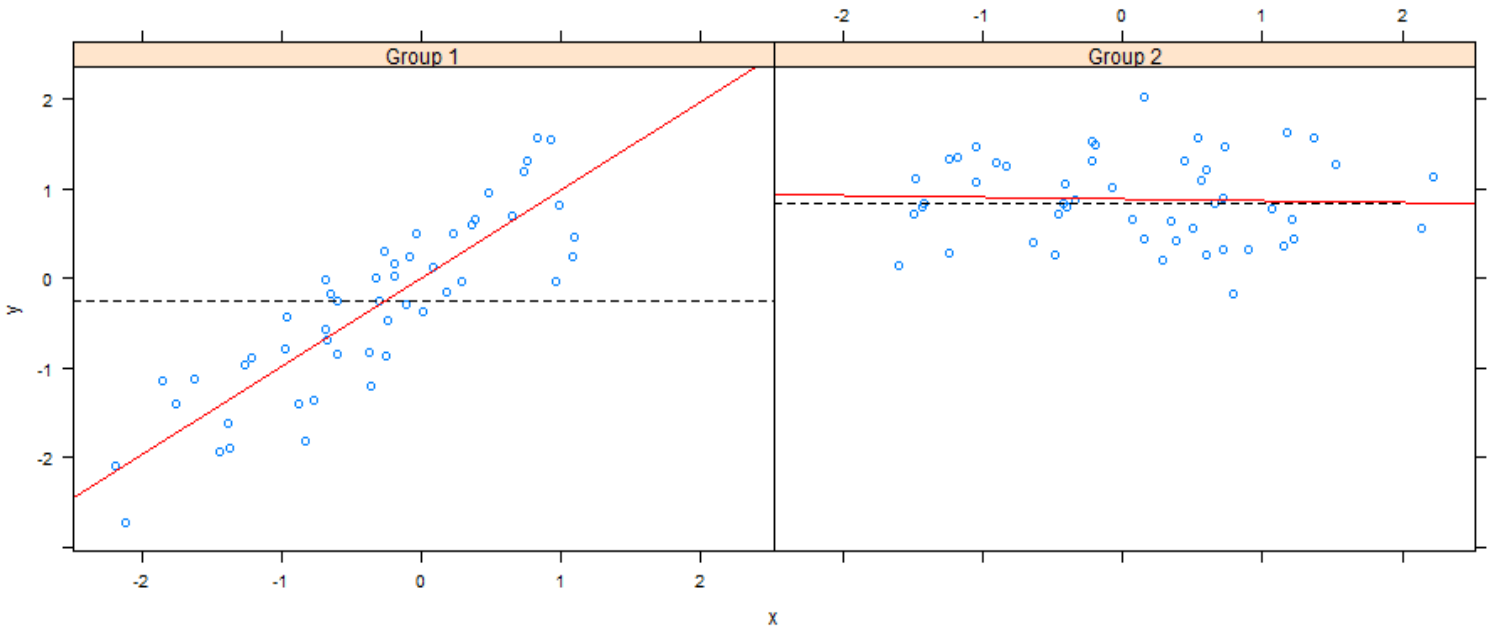


Lattice – panelové funkcie

- Lattice funkcie používajú panelovú funkciu pre kontrolu čo sa udeje v jednotlivých paneloch výstupu („subgrafoch“)
- Balík lattice má default panelové funkcie => je však možné dodať vlastné funkcie pre každý panel podľa potreby
- Panelové funkcie dostávajú x/y súradnice pre dátové body v rámci ich panelu (+ voliteľné argumenty)

Príklad – vlastná panelová funkcia

```
xyplot(y ~ x | f, panel = function(x, y, ...) {
        panel.xyplot(x, y, ...) # volanie default panel funkcie
        panel.abline(h = median(y), lty = 2)
        # rridanie horizontálnej čiary pre medián
        panel.lmline(x, y, col = 2) # pridanie regresnej priamky
})
```



Základy grafického systému ggplot2

- Niekedy ho volajú aj „tretí“ grafický systém pre R (po Base a Lattice), http://ggplot2.org
- Gramatika pre reprezentáciu a abstrakciu grafických ideií/objektov
- Má premyslený spôsob ako definovať grafické objekty a operácie s nimi (resp. nad nimi)
  - „Skracuje vzdialenosť medzi mysľou a výstupom“
- „Skrátene, gramatika nám hovorí že štatistická grafika je mapovanie z dát do esteticky vhodných atribútov (farba, tvar, veľkosť) geometrických objektov (body, čiary, stĺpec). Graf (či grafický výstup) môže takisto obsahovať štatistické transformácie dát a je vykreslený v špecifickom systéme súradníc“
    - z ggplot2 knihy

Základné funkcie ggplot2

- Grafy sa skladajú z „objektov“ *aesthetic* (obsah osí x,y, veľkosť, tvar, farba) and *geom* (body, čiary,... – napríklad vykreslenie regresnej krivky)
- Okrem toho vieme upravovať postupne upravovať ďalšie aspekty ako napr. súradnicový systém, škály, ...
- Veľmi dôležité sú (pomenované) faktory pre indikáciu podmnožín dát
- Okrem klasického volania ggplot() existuje jednoduchšia verzia qplot(), ktorá skrýva viaceré kroky
- Syntax je niekde medzi Base a Lattice
- Vytvorí veľmi dobré výstupy vhodné pre publikovanie

Základné komponenty zobrazovania

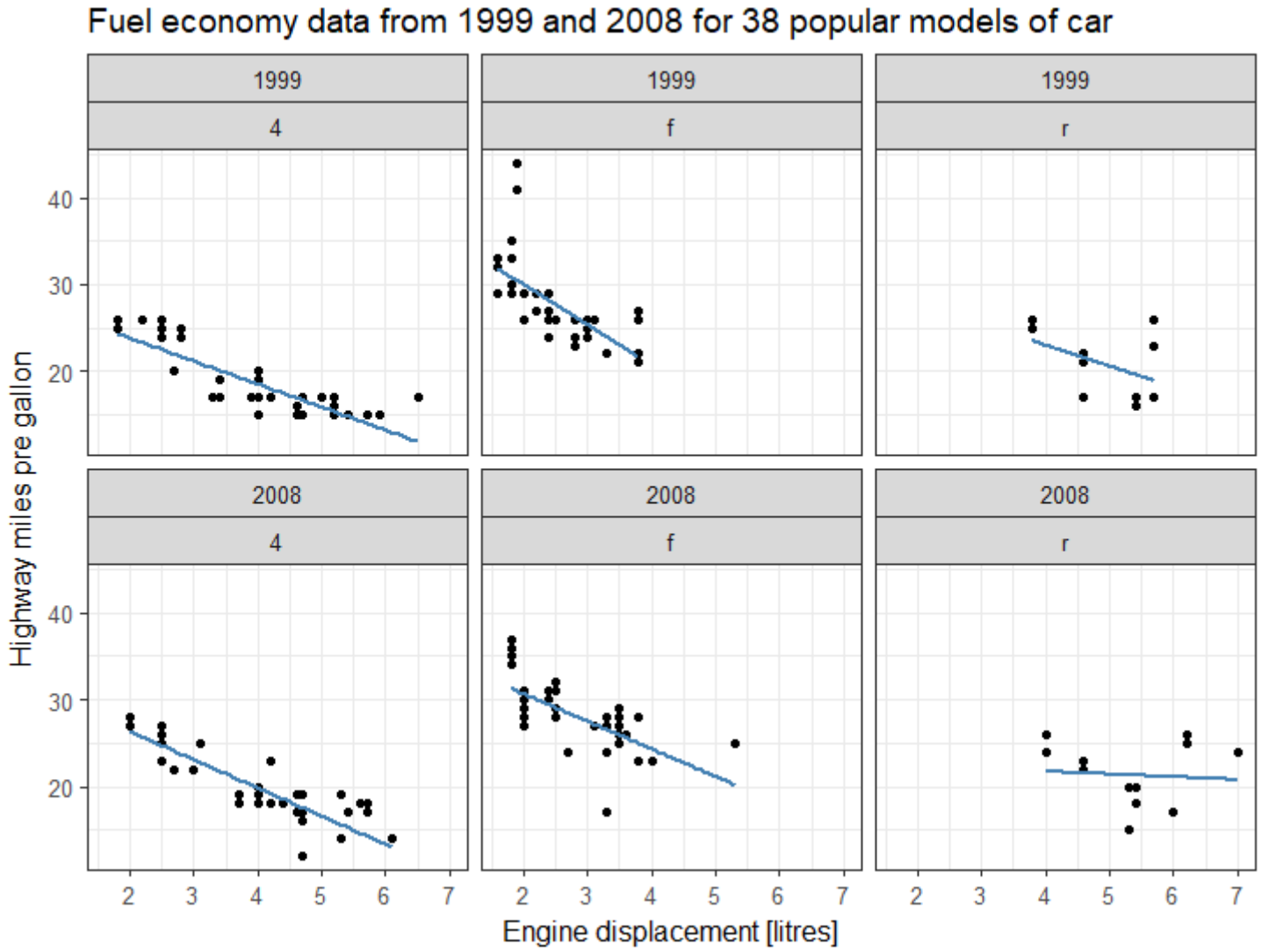
- *data* (dáta resp. dátový rámec)
- mapovanie *aesthetic*: ako sú dáta sú mapované na jednotlivé osi x, y, na farbu, veľkosť, tvar
- Mapovanie *geoms*: geometrické objekty ako čiary, krivky, body, ...
- *facets*: pre podmienené vykresľovanie (podgrafy)
- *stats*: štatistické transformácie ako binning, kvantily, vyhladzovanie, spriemerovanie, štatistické algoritmy, ...
- *scales*: aká škála sa má použiť pre mapovanie *aesthetic*
- súradnicový systém (*coordinate system*)

Budovanie grafu cez ggplot()

- Graf cez ggplot() sa tvorí pridávaním vrstiev (layers) odpovedajúcim rôznym komponentom
- Majme dáta mpg - postupne pridávame vrstvy ku grafu

```
> mydata = mpg
> mydata$year = factor(mpg$year)
> g = ggplot(mydata, aes(displ, hwy))
> g + geom_point() +
     facet_wrap(year ~ drv, nrow = 2, ncol = 3)  +
    geom_smooth(method="lm", se=FALSE, col="steelblue") +
    theme_bw(base_size = 12) +
    labs(x = "Engine displacement [litres]") +
    labs(y = "Highway miles pre gallon") +
    labs(title = "Fuel economy data from 1999 and 2008 for 38 popular models of car")
```

Príklad – výsledný graf

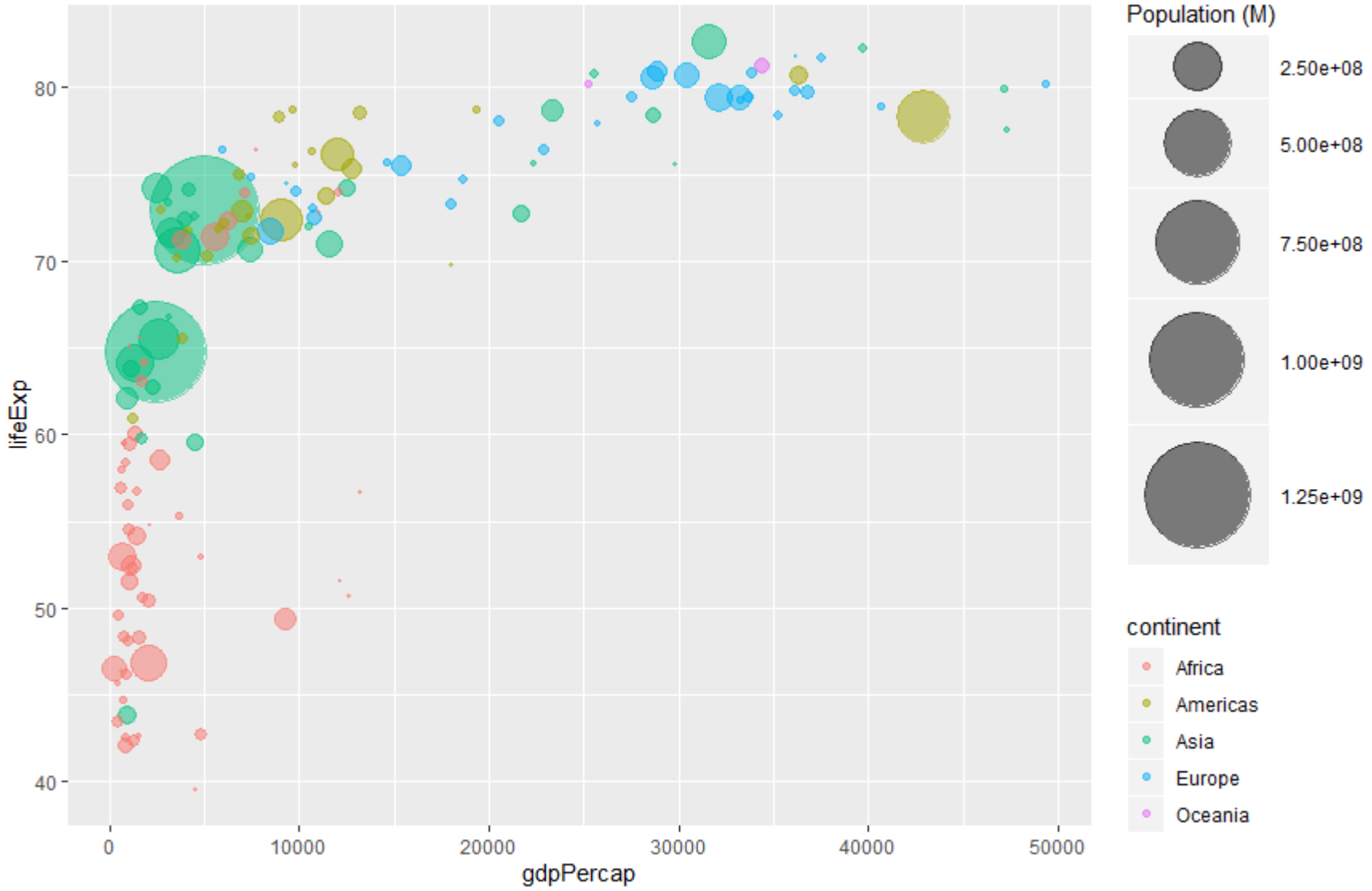


Príklad – gapminder dáta

- dplyr + ggplot
- Krajiny – graf dĺžka života vs HDP na obyvateľa + krajiny sú ešte rozlíšené podľa populácie (veľkosť bubliny) a kontinentu (farba bubliny)

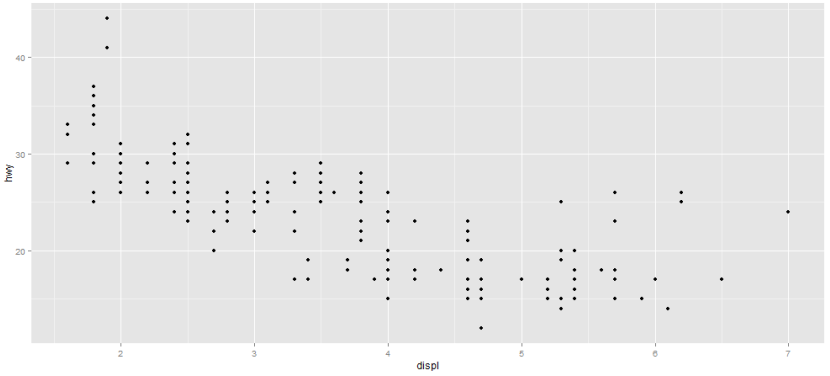
```
library(ggplot2)
library(dplyr)
library(gapminder)
data <- gapminder %>% filter(year=="2007") %>% dplyr::select(-year)

data %>%
   arrange(desc(pop)) %>%
   mutate(country = factor(country, country)) %>%
   ggplot(aes(x=gdpPercap, y=lifeExp, size=pop, color=continent)) +
      geom_point(alpha=0.5) +
      scale_size(range = c(.1, 24), name="Population (M)")
```

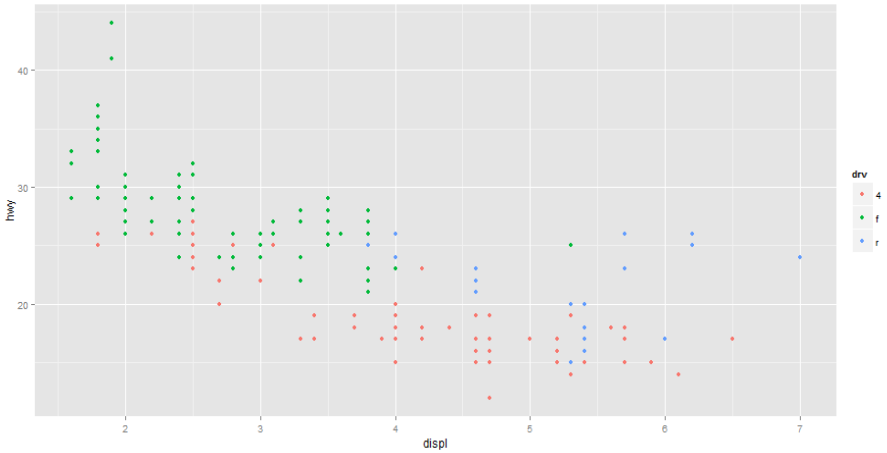
Príklad – gapminder dáta - graf



qplot (zjednodušené volanie ggplot)

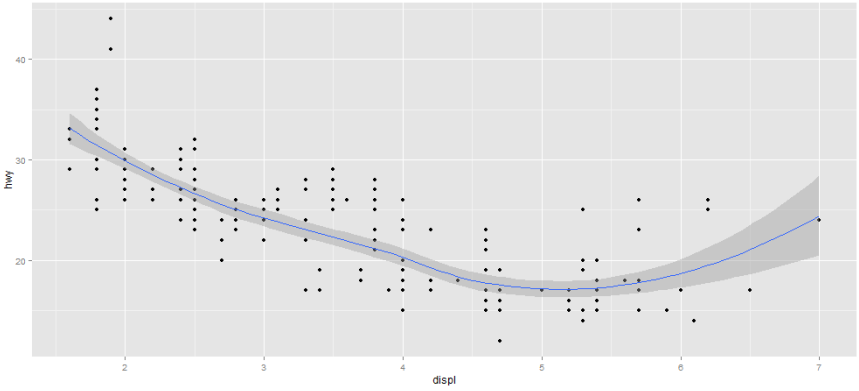
Jednoduché vypísanie x (displ) ku y (hwy) pre dané *data* (mpg)
[atr. *hwy*, *displ*, faktorový atr. *drv*]
qplot(displ, hwy, data = mpg)





Pridanie **aesthetic** prvku ke jednoduchej verzii => použitie col pre nastavenie rozličnej farby pri faktore *drv* (hodnoty 4,f,r – generuje legendu)
qplot(displ, hwy, data = mpg, col=drv)

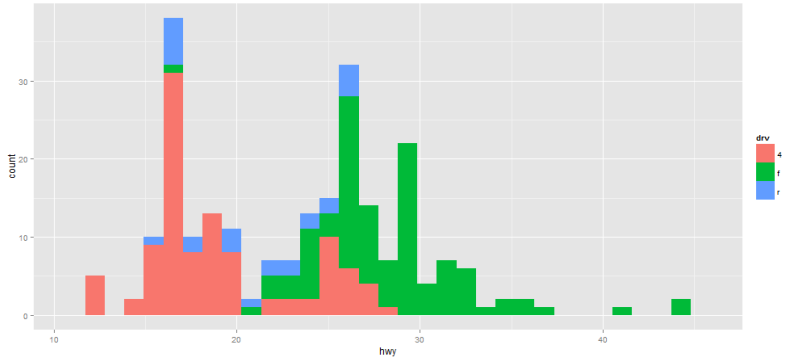
Pridanie **geom** prvku k jednoduchej verzii => vloženie hladkej krivky modelujúcej dáta cez geom „smooth“
qplot(displ, hwy, data = mpg, geom = c("point", "smooth"))
# pridaním method = lm ... dostaneme cez geom regresnú priamku

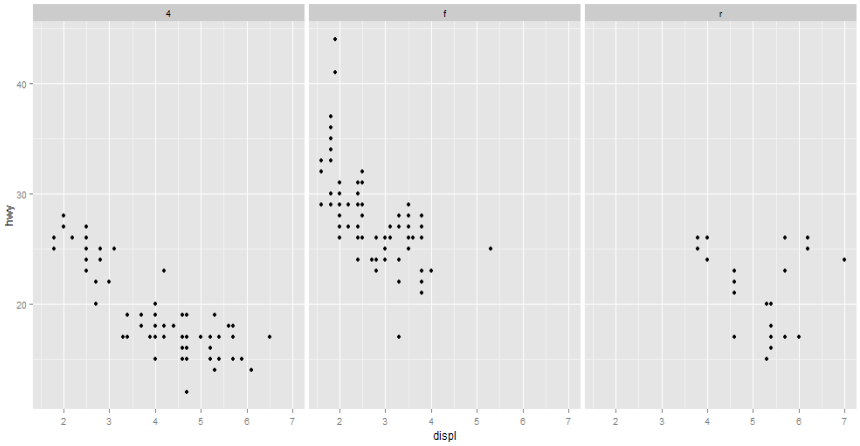


Príklady – qplot (2)

Histogram pre hwy cez fakor drv

```
qplot(hwy, data = mpg, fill =
drv)
```



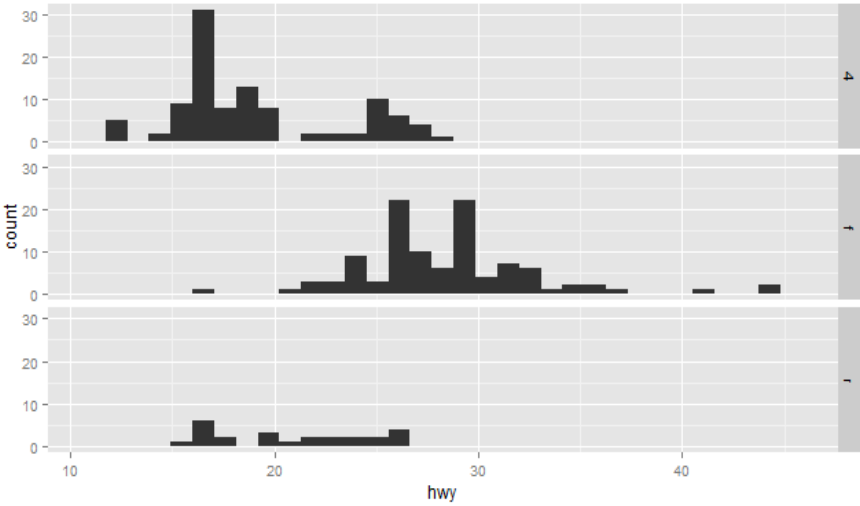


Použitie **facets** na zobrazenie viacero grafov cez faktor drv

```
qplot(displ, hwy, data = mpg,
facets = . ~ drv)
```

Použitie **facets** pre vytvorenie histogramov pre hwy po jednotlivých faktoroch drv

```
qplot(hwy, data = mpg, facets = drv ~ .)
# pre spojitú verziu grafu hustoty je
# možné nastaviť geom="density"
```



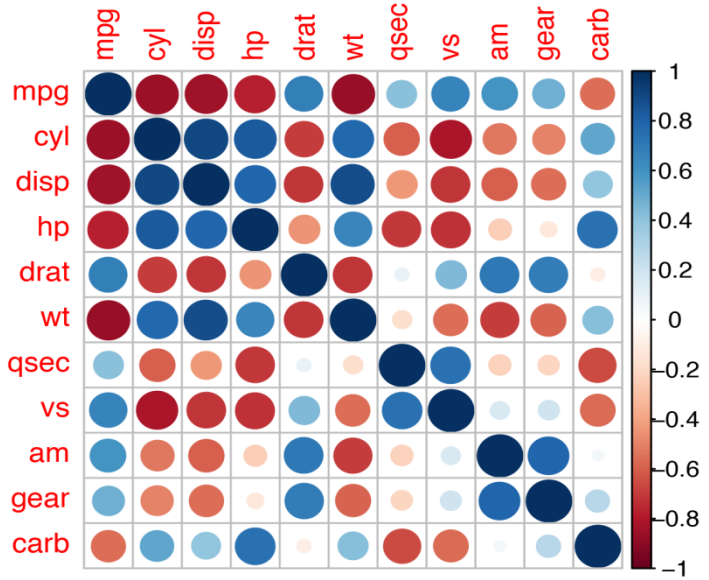
Ďalšie typy grafov a vizualizácií v R

Okrem klasických grafov ako

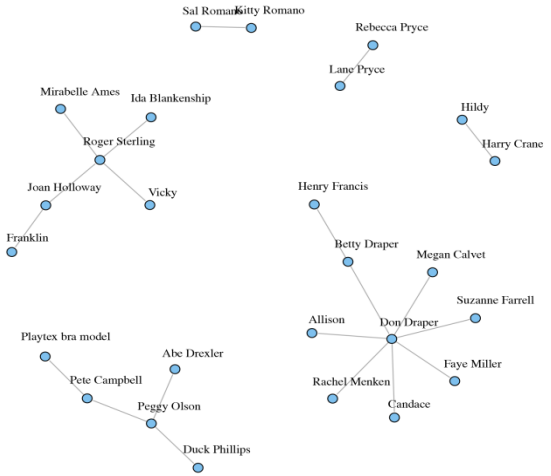
- Stĺpcové, bodové, čiarové grafy, histogramy, hustotné grafy, boxploty, grafy funkcií, ...
- a ich úprav – zmena súradnicového systému (coord_polar), pomenovania bodov v grafe, veľkosti bodov podľa vlastností, zmena škál, spojité zobrazovanie škál, „heat“ mapy...

existujú viaceré ďalšie možnosti

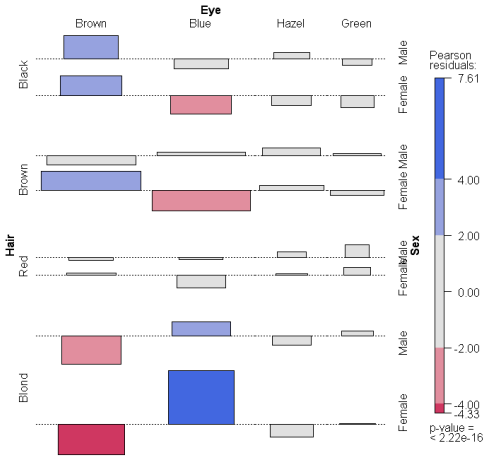
- Geografické mapovanie výsledkov na mapu (balík **maps**)
- Koláčové grafy (cez pie() + 3D verzie v balíku **plotrix**)
- 3D scatterplot grafy (balík **3Dscatterplot**) + otáčajúce sa 3D verzie (balík **rgl**)
- Mozaikové a asociačné grafy (pre rozšírenú vizualizáciu kategoriálnych dát - balík **vcd**)
- Špeciálne korelačné diagramy (tzv. „correlograms“, balík **corrgram**)
- Interaktívne (real-time) grafy – balíky **rggobi**, **iplots**, ...
- Sieťové grafy (balík **igraph**)
- Mnoho ďalších knižníc okrem base, lattice, ggplot, najmä javascriptových knižníc – napr. plotly, D3, Leaflet, ...
- ....



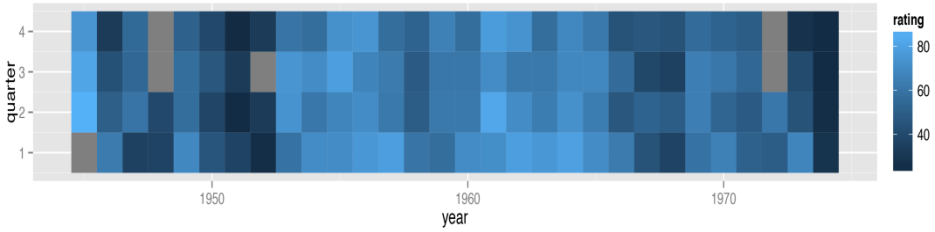
Korelačný diagram



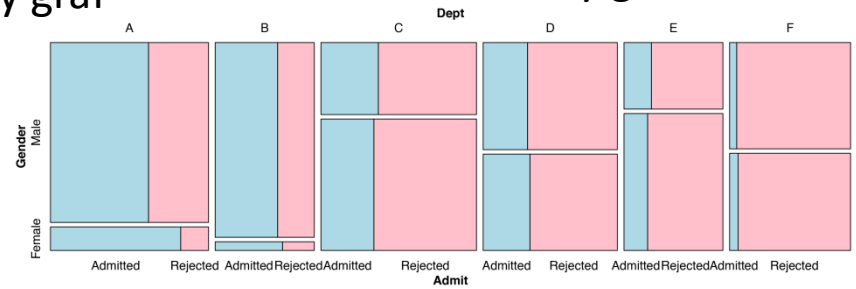
Sieťový graf



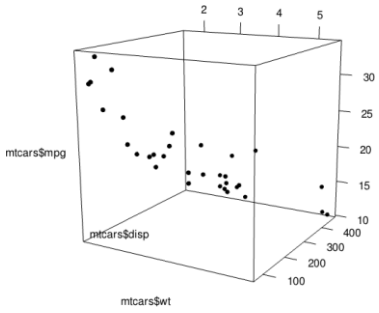
Asociačný graf

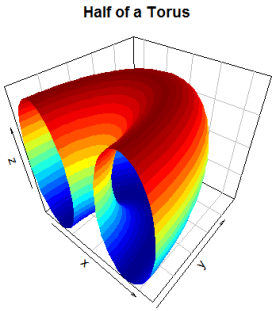


Heat mapa

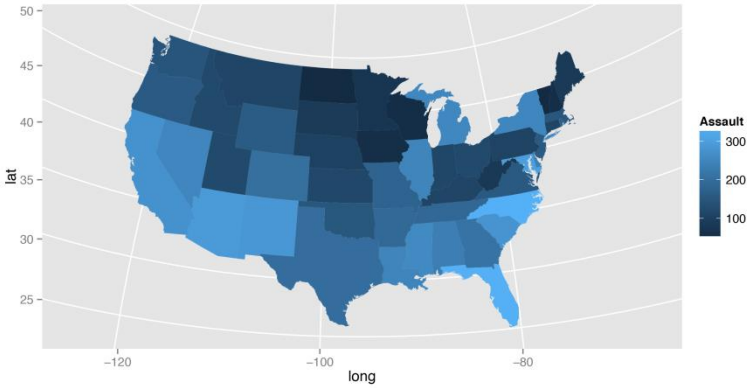


Mozaikový graf





3D grafy(bodový a spojitý)



Použitie geograf. mapovania