Duomenų analizės įvadas

4.1. dalis

Justas Mundeikis

VU EVAF

2019-05-25

Turinys

1. Analitinių grafikų principai
2. EDA grafikai
3. Grafikų išsaugojimas
4. ggplot2

Analitinių grafikų principai

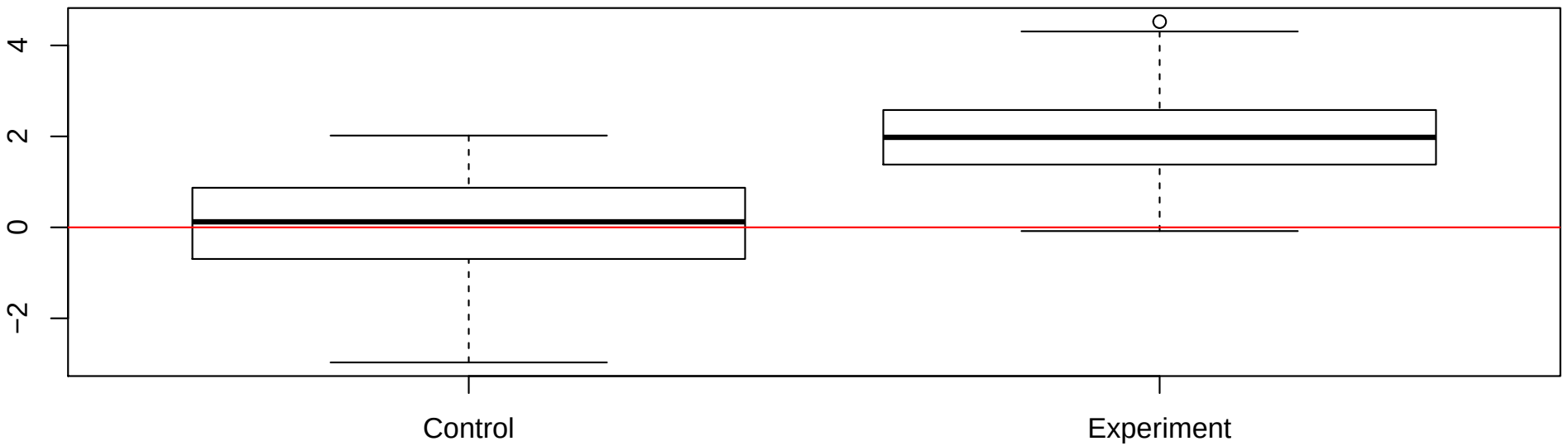
Analitinių grafikų principai

Knygoje Edward Tufte (2006), Beautiful Evidence autorius aprašo kaip reikėtų vizualiaipateikti statisiką auditorijai.

Analitinių grafikų principai

1. Pateikite palyginimus, kontrastus, skirtumus
   - Hipotezių įrodymai visada yra reliatyvūs (alternatyviai hipotezei)
   - Ar grafikas atsako į klausimą: "Palyginus su kuo?"

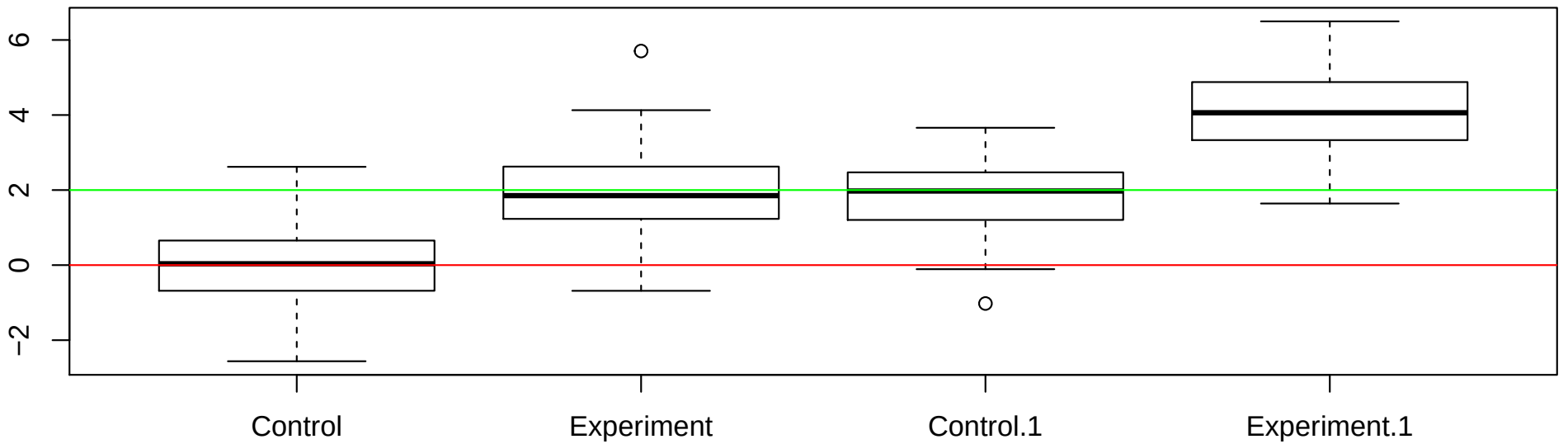
```
df <- data.frame(Control=rnorm(100,0), Experiment =rnorm(100,2))
boxplot(df)
abline(h=0, col="red")
```



Analitinių grafikų principai

2. Pateikite priežastinius-pasekminius ryšius, veikimo principus, sistemines struktūras
   - Nebūtinai tikras priežastinis ryšis, bet kaip Jūs / teorija mano

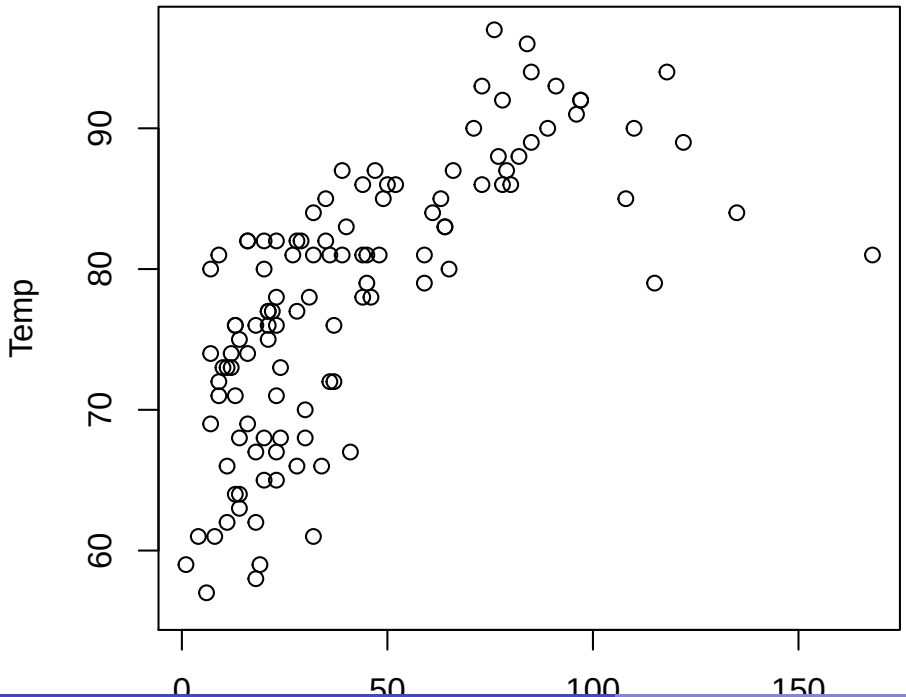
```
df <- data.frame(Control=rnorm(100,0), Experiment =rnorm(100,2), Control=rn
boxplot(df)
abline(h=0, col="red")
abline(h=2, col="green")
```

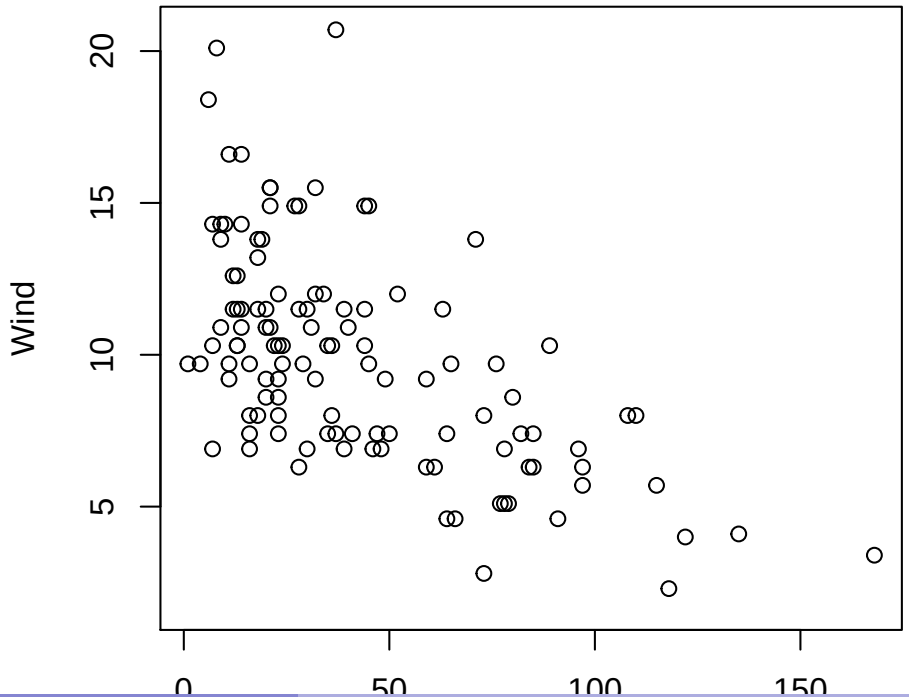


Analitinių grafikų principai

3. Pateikite *multivariate data*
   - *multivariate* = daugiau nei 2 kintamieji
   - Kartu tai perteikia galimas sąsajas

```
par(mfrow=c(1,2))
with(airquality, plot(Ozone, Temp))
with(airquality, plot(Ozone, Wind ))
```





Analitinių grafikų principai

4. Integruokite skirtingus įrodymus (žodžius, skaičius, paveiklsiukus, diagramas)
   - dažnai grafikai yra iškalbingesni
   - tačiau kartais lentelės gali būti naudingesnės
   - derinkite grafikus ir lenteles perteikdami savo *story*
5. Tvarkingai aprašykite grafikus
   - Grafiko pavadinimas
   - Grafiko ašys
   - Šaltiniai, geriausia nurodyti lentelės ID (pvz., Eurostat (nama_10_q), LSD (S3R0004_M3080242))
6. *Content is king*
   - Jeigu neturite įdomios "istorijos", joks grafikas Jūsų neišgelbės

EDA grafikai

EDA grafikai

- EDA - *exploratory data analysis*
- greitai ir paprastai sugeneruoti grafikai
- daug grafikų
- padeda suprasti sąsajas pirminiame analizės žingsnyje
- asmeniniam vartojimui
- grožis kuriamas su `ggplot2`

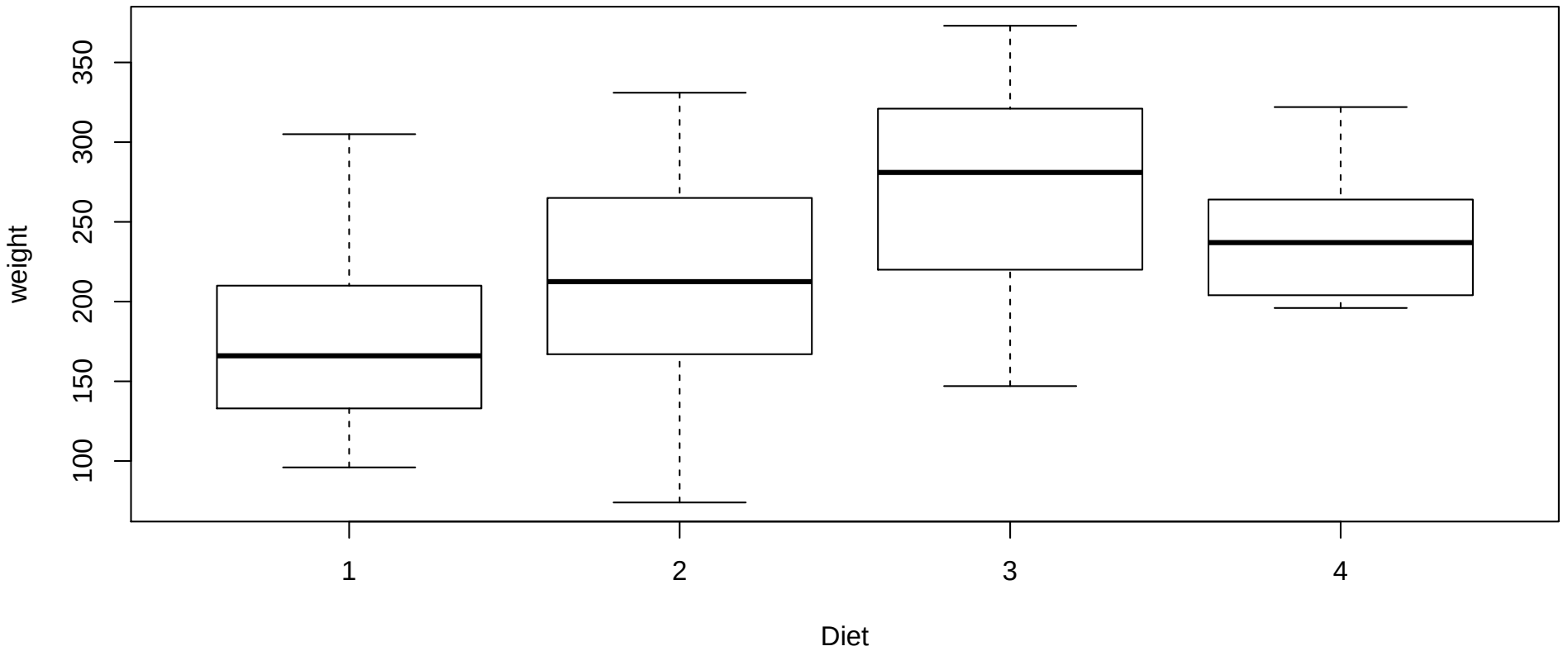
Summary

Nors ne grafikas, bet labai pravartu visad pradėti nuo `summary()`, kad pamatyti, kaip atrodo duomenis apskritai

```
summary(ChickWeight)
##      weight           Time           Chick     Diet
##  Min.   : 35.0   Min.   : 0.00   13     : 12   1:220
##  1st Qu.: 63.0   1st Qu.: 4.00   9      : 12   2:120
##  Median :103.0   Median :10.00   20     : 12   3:120
##  Mean   :121.8   Mean   :10.72   10     : 12   4:118
##  3rd Qu.:163.8   3rd Qu.:16.00   17     : 12
##  Max.   :373.0   Max.   :21.00   19     : 12
##                                  (Other):506
```

Boxplot

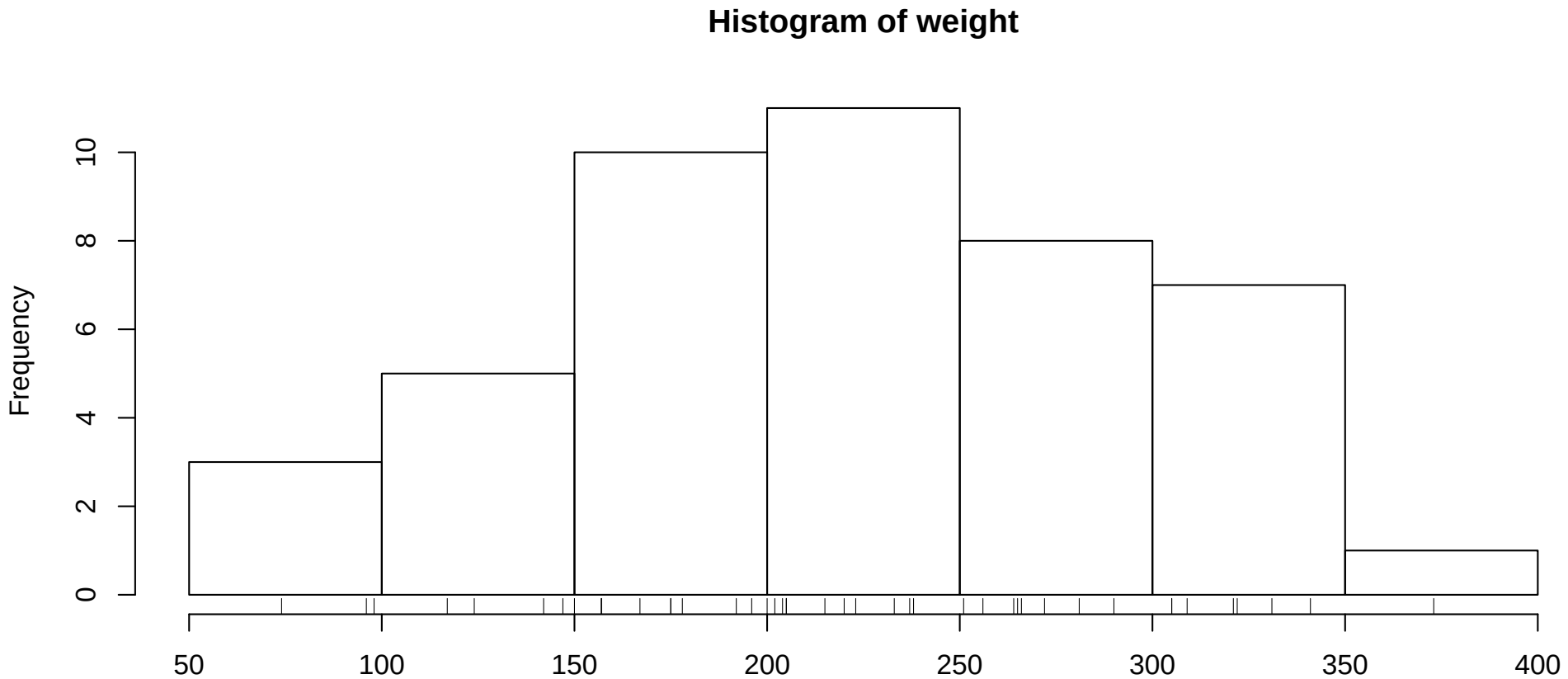
```
with(subset(ChickWeight, Time==21), boxplot(weight~Diet))
```



Histrogram

- `rug` plottina pavienius elementus kaip brūkšnelius
- ir leidžia suprasti ar deramai pasirinkti intervalai

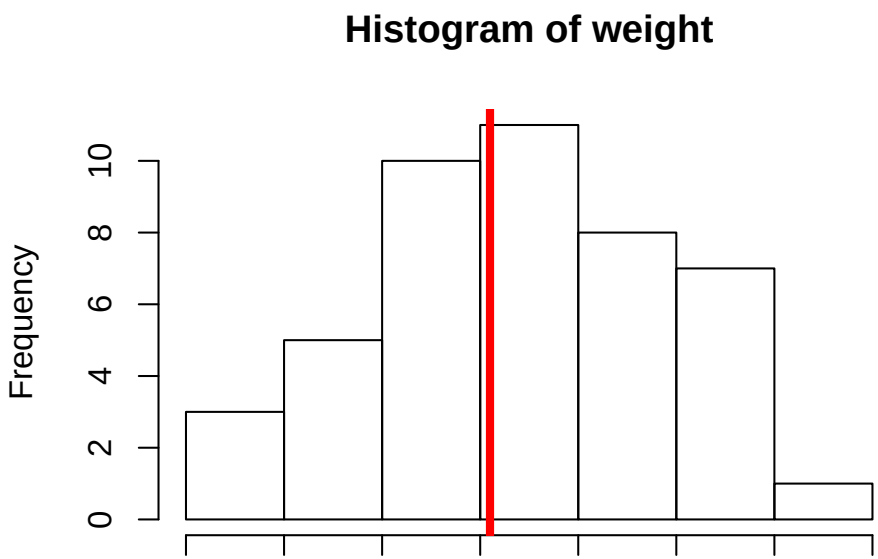
```
with(subset(ChickWeight, Time==21), hist(weight))
with(subset(ChickWeight, Time==21), rug(weight))
```

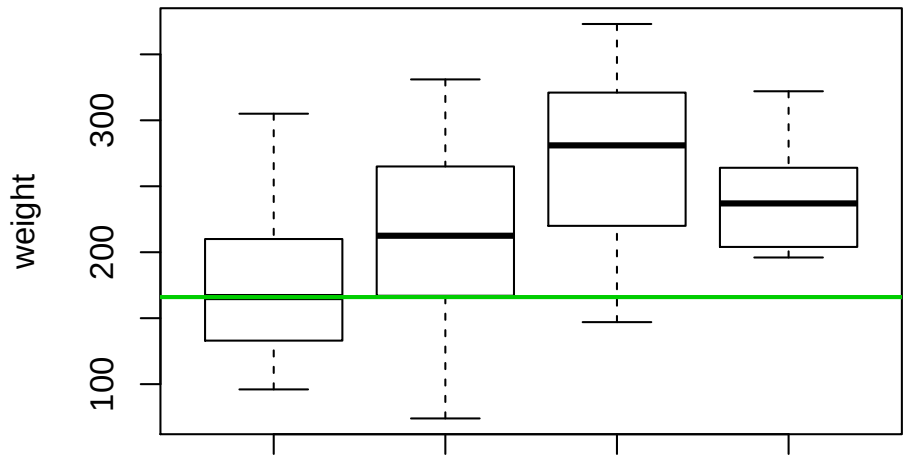


Histrogram

- abline brėžia tieses (?abline)
- v=..
- h=..

```
par(mfrow=c(1,2))
with(subset(ChickWeight, Time==21), hist(weight))
abline(v=median(ChickWeight$weight[ChickWeight$Time==21]), col=2, lwd=4)
with(subset(ChickWeight, Time==21), boxplot(weight~Diet))
abline(h=166, col=3, lwd=2)
```

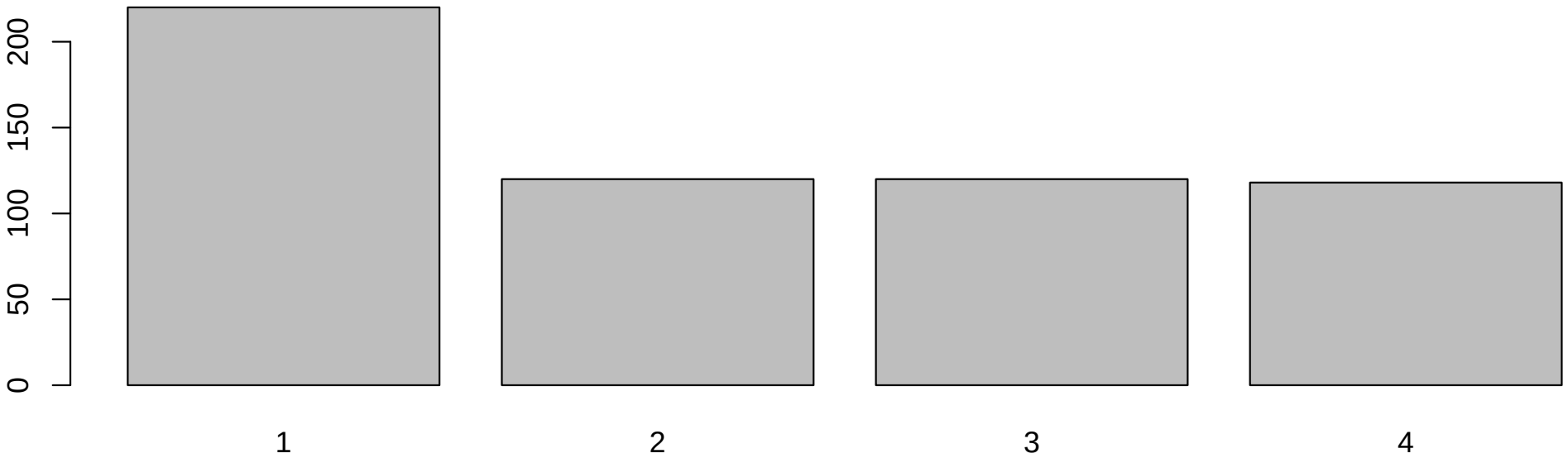




Barplot

Barplot atvaizuoja lentelinius turinius

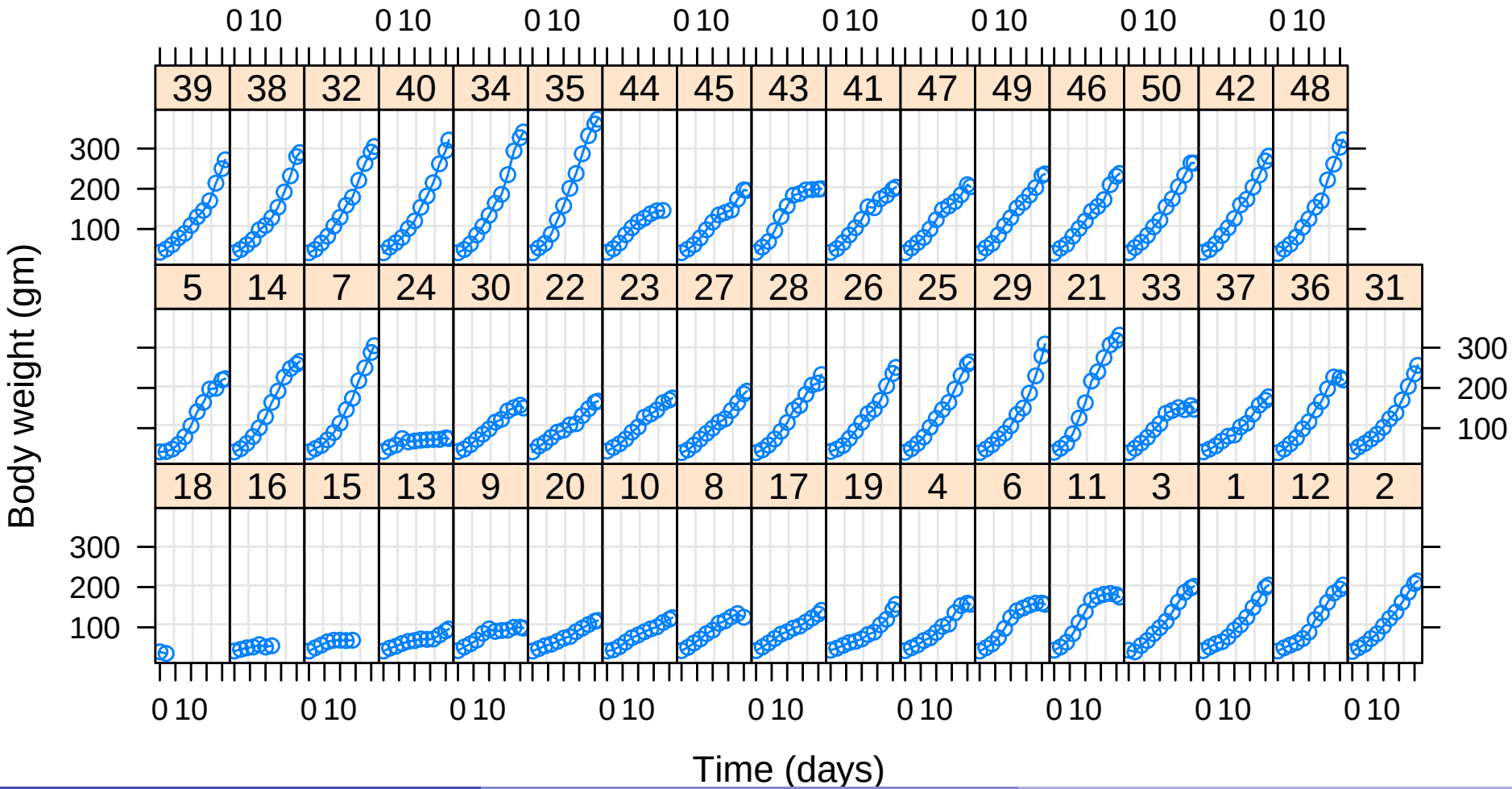
```
table(ChickWeight$Diet)
##
##   1   2   3   4
## 220 120 120 118
barplot(table(ChickWeight$Diet))
```



Scatterplot

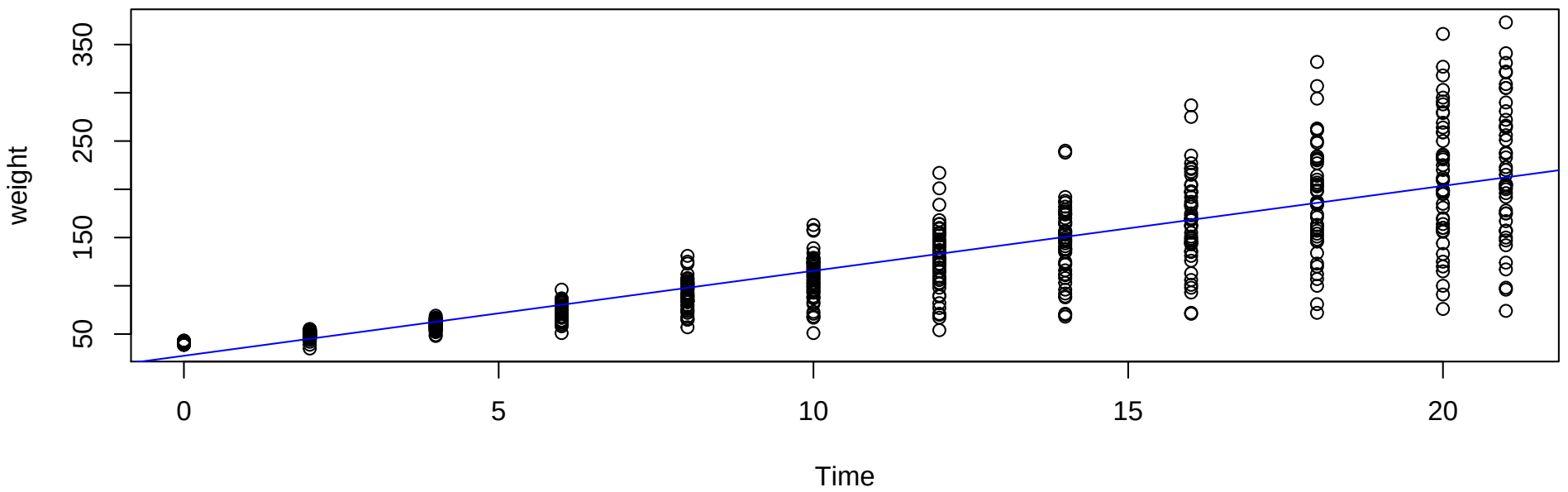
Kai nedaug kintamųjų ir norisi matyti galimas sąsajas

```
plot(ChickWeight)
```



Scatterplot

```
with(ChickWeight, plot(Time, weight))
abline(with(ChickWeight, lm(weight~Time)), col=4)
```



Base plotting funkcijos

- plot - sukuria pagrindinį grafiką
- lines - prideda linijas (vektorius)
- points - prideda taškus
- text - prideda tekstą
- title - prideda anotacijas
- axis - prideda ašių žymėjimus, tekstą

Base Graphics parametrai

- par - gloablūs parametrai
- bg - the background color
- mar - the margin size
- oma - the outer margin size
- mfrow - number of plots per row, column (filled row-wise)
- mfcol - number of plots per row, column (filled col-wise)
- pasitikrinti galima :

```
par("bg")
## [1] "transparent"
par("mar")
## [1] 5.1 4.1 4.1 2.1
```

Base Graphics parametrai

- plot(...):
- pch - the plotting symbol
- lty - the line type
- lwd - the line width
- col - color
- xlab - charackter string x-axis label
- ylab - charackter string y-axis label
- main - charackter string main label

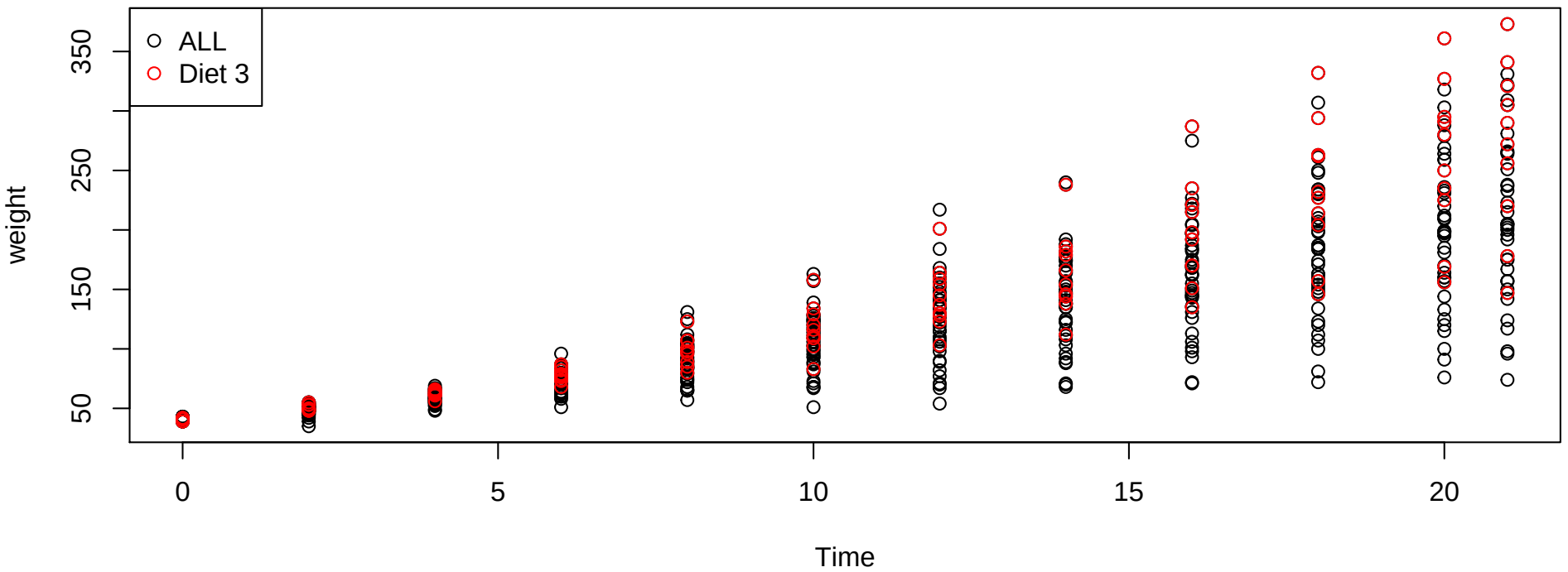
example(points)

- išbandykite: example(points)

Scatterplot

Nubraižomas grafikas, ant viršaus pasirinkti taškai, pridedama legenda

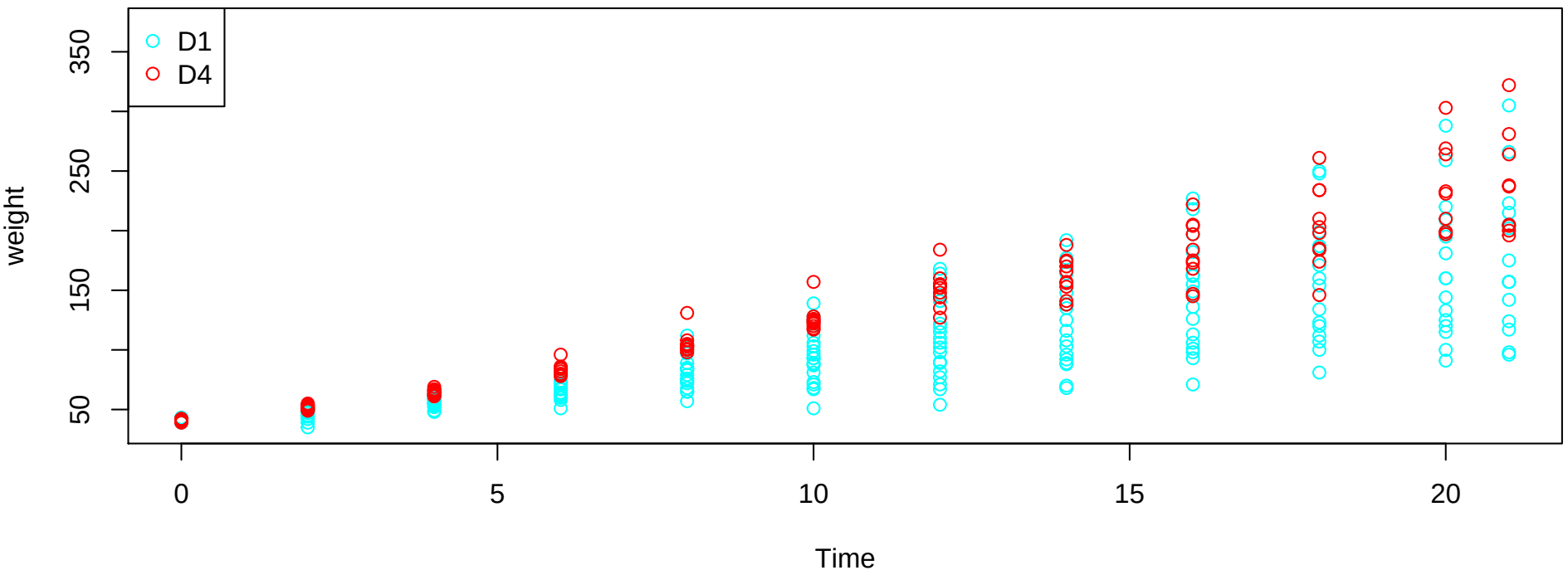
```
with(ChickWeight, plot(Time, weight))
with(subset(ChickWeight, Diet==3), points(Time, weight, col="red"))
legend("topleft", pch=1, col=c("black", "red"), legend=c("ALL", "Diet 3"))
```



Scatterplot

- `type="n"` nepiešia nieko, tik sukuria bazę, col galima nurodyti

```
with(ChickWeight, plot(Time, weight, type="n"))
with(subset(ChickWeight, Diet==1), points(Time, weight, main="Diet 1", col=
with(subset(ChickWeight, Diet==4), points(Time, weight, main="Diet 4", col=
legend("topleft", pch=1, col=c(5,2), legend=c("D1", "D4"))
```



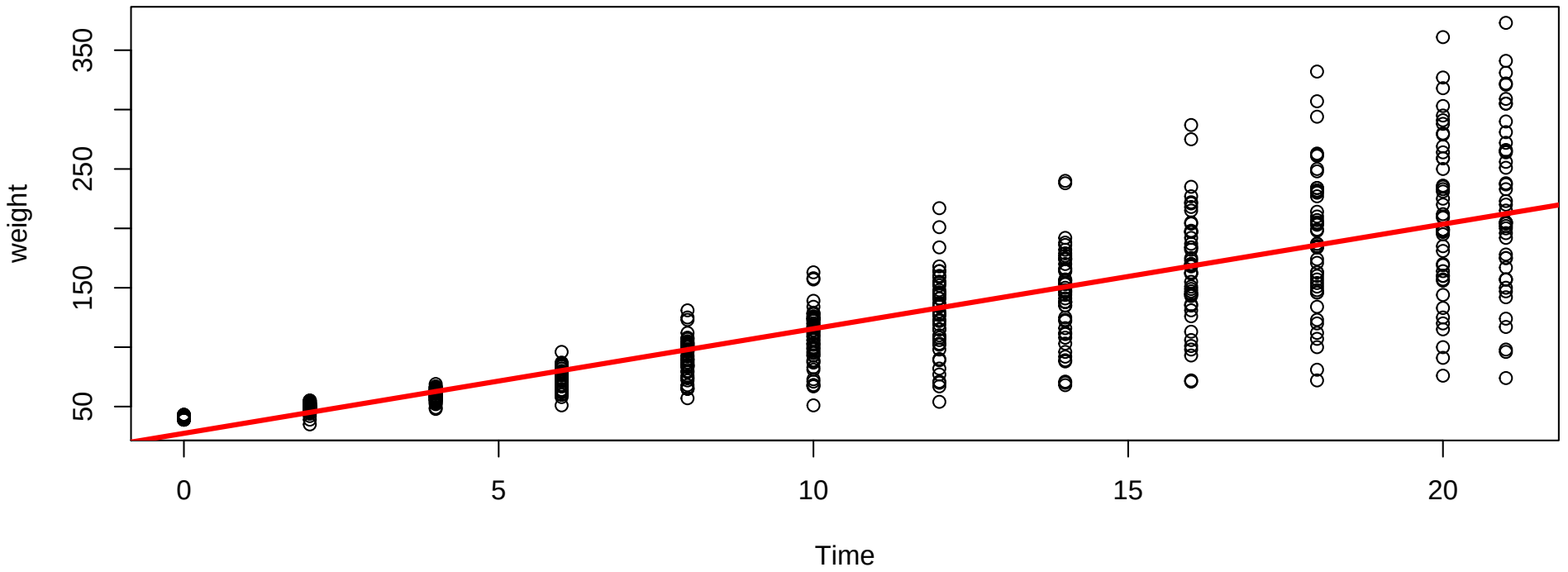
Tiesinė regresija

```
model <- lm(weight~Time, ChickWeight); summary(model)
##
## Call:
## lm(formula = weight ~ Time, data = ChickWeight)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -138.331  -14.536    0.926   13.533  160.669
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  27.4674     3.0365   9.046   <2e-16 ***
## Time          8.8030     0.2397  36.725   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 38.91 on 576 degrees of freedom
## Multiple R-squared:  0.7007, Adjusted R-squared:  0.7002
## F-statistic:  1349 on 1 and 576 DF,  p-value: < 2.2e-16
```

Tiesinė regresija

Funkcijos `lm` sugeneruojami parametrai perduodami `abline`

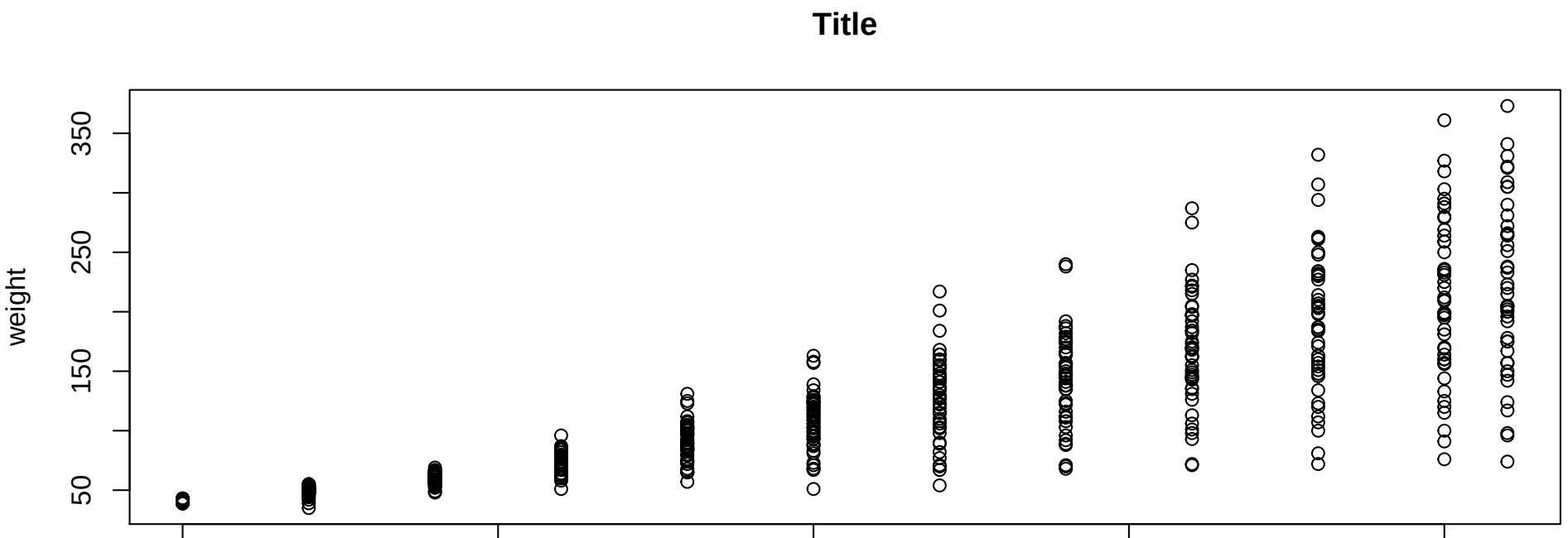
```
model <- lm(weight~Time, ChickWeight)
with(ChickWeight, plot(Time, weight), type="n")
abline(model, lwd=3, col=2)
```



mar ir oma

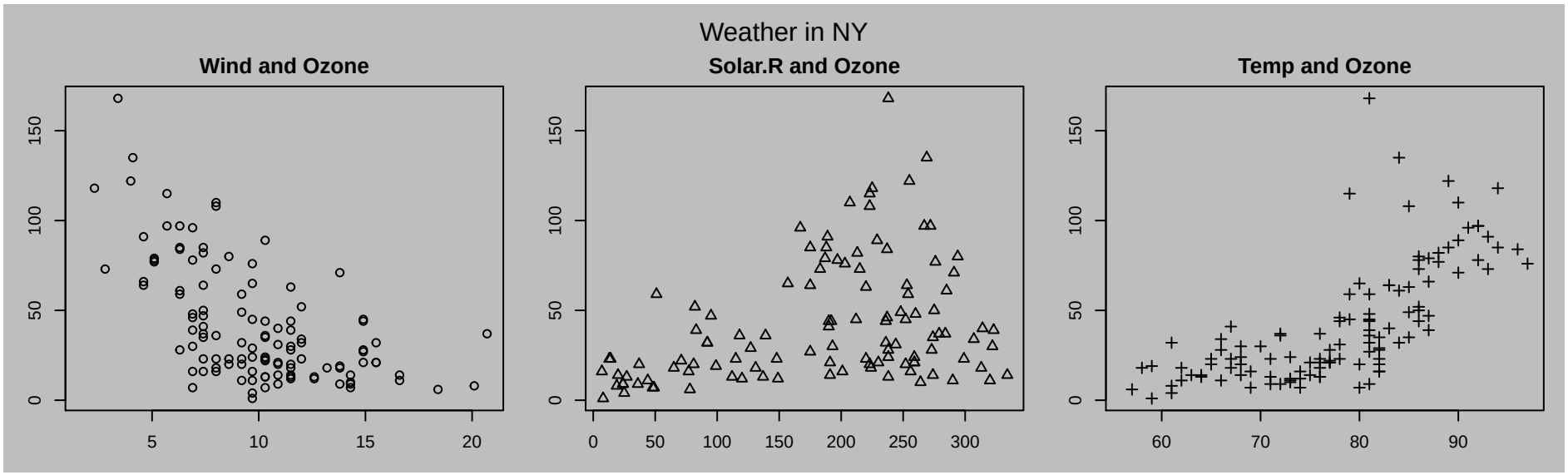
- mar - the margin size
- oma - the outer margin size

```
par("mar")
## [1] 5.1 4.1 4.1 2.1
par("oma")
## [1] 0 0 0 0
with(ChickWeight, plot(Time, weight, main="Title"))
```



outer ir mtext

```
par(mfrow=c(1,3),mar=c(3,3,2,1),oma=c(0,0,2,0),bg="grey")
with(airquality, {
    plot(Wind, Ozone, main="Wind and Ozone", pch=1)
    plot(Solar.R, Ozone, main="Solar.R and Ozone", pch=2)
    plot(Temp, Ozone, main="Temp and Ozone", pch=3)
    mtext("Weather in NY", outer = TRUE)
})
```



Grafikų išsaugojimas

Graphics devices

- `? Devices`
- Ekrane (`windows()`. `quartz()`, `x11()`)
- Failuose:
- Vektoriniai formatai
  - pdf()
  - svg()
  - ...
- Bitmap formatai
  - png()
  - jpeg()
  - tiff()
  - bmp()
- dev.copy() nukopijuoja ekrane esantį grafiką į failą
- dev.off() išjungia `device`

Graphics devices

Išsaugos grafiką faile (darbinėje direktorijoje) pavadinimu “plot”

```
png(file="plot.png") # įjungiamas device
plot(airquality$Ozone) # kas siunčiama
dev.off() # išjungiamas device
```

Graphics devices

Kartais gali būti pravarti kopijavimo funkcija

```
plot(airquality$Ozone)
dev.copy(png, file="plot.png")
dev.off() # išjungiamas device
```

Graphics devices

Skirtingus galimus nustatymus kokie išsaugomo grafiko parametrai galima pasitikrinti su pvz `?png`

ggplot2

ggplot2

- gg - Grammer of Graphics (Leland Wilkinson)
- parašyta Hadley Wickham (taip kur ir dplyr...)
- `install.packages("ggplot2")` ir suprantama (`library(ggplot2)`)
- cheatsheet ggplot2 patarčiau atsidaryti
- duomenys turi būti `dataframe` objekte, geriausia `long` formatu

ggplot2

- A data frame
- aesthetic mappings - spalva, dydis
- geoms - objektai (taškai, linijos...)
- facets - kondicionalus plotai
- stats - statistinės transormacijos
- scales - kokias skales naudojamos
- coordinate system

ggplot2

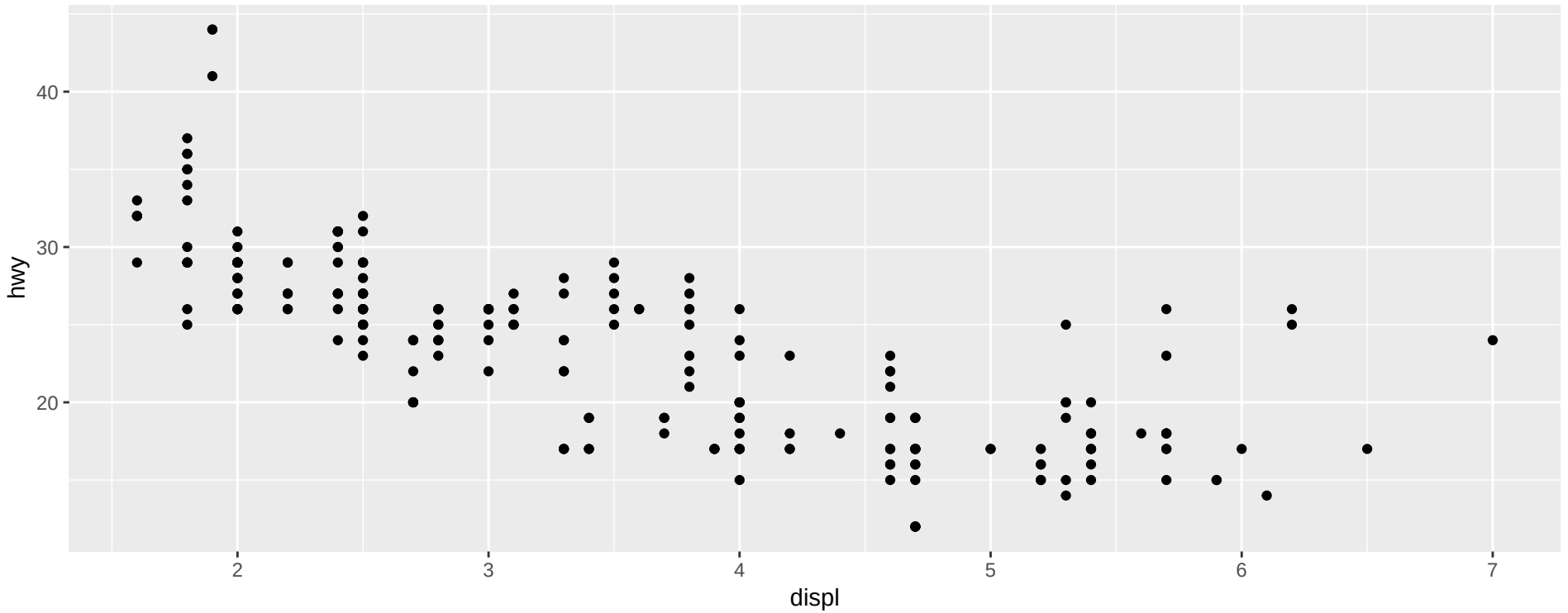
- neperdavus aesthetic mappings, nubraižytų tik `grid`

```
ggplot(mpg)
```

ggplot2

- perduodame aesthetic mappings geom_…. points

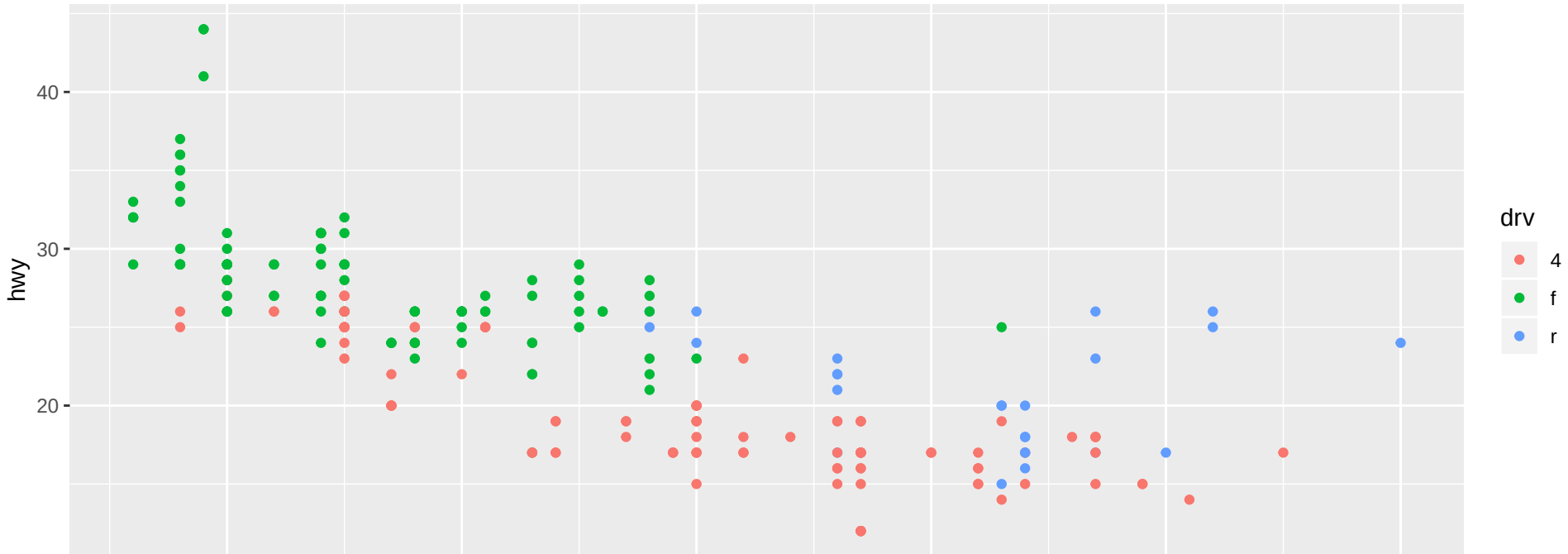
```
ggplot(mpg)+
    geom_point(aes(x=displ,y=hwy))
```



ggplot2

- `aes()` viduje esantys nustatymai veikia pačius duomenis/ jų atvaizdavimą
- `color=` faktorius , pvz `drv`, duomenys suskaidomi pagal faktorių ir nuspalvinami skirtingomis spalvomis

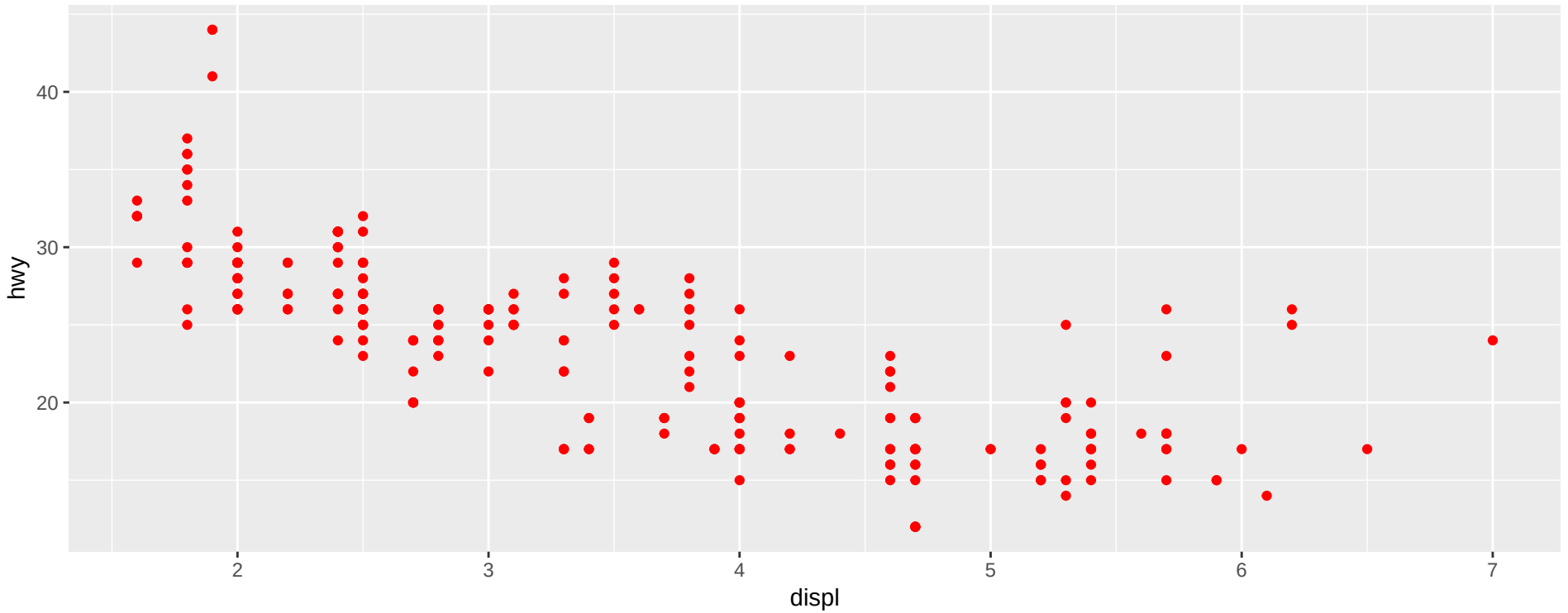
```
ggplot(mpg)+
    geom_point(aes(displ, hwy, color=drv))
```



ggplot2

- `color= "red"` perdavus ***ne*** `(aes)` viduje, viskį taškai nudažomi raudonai

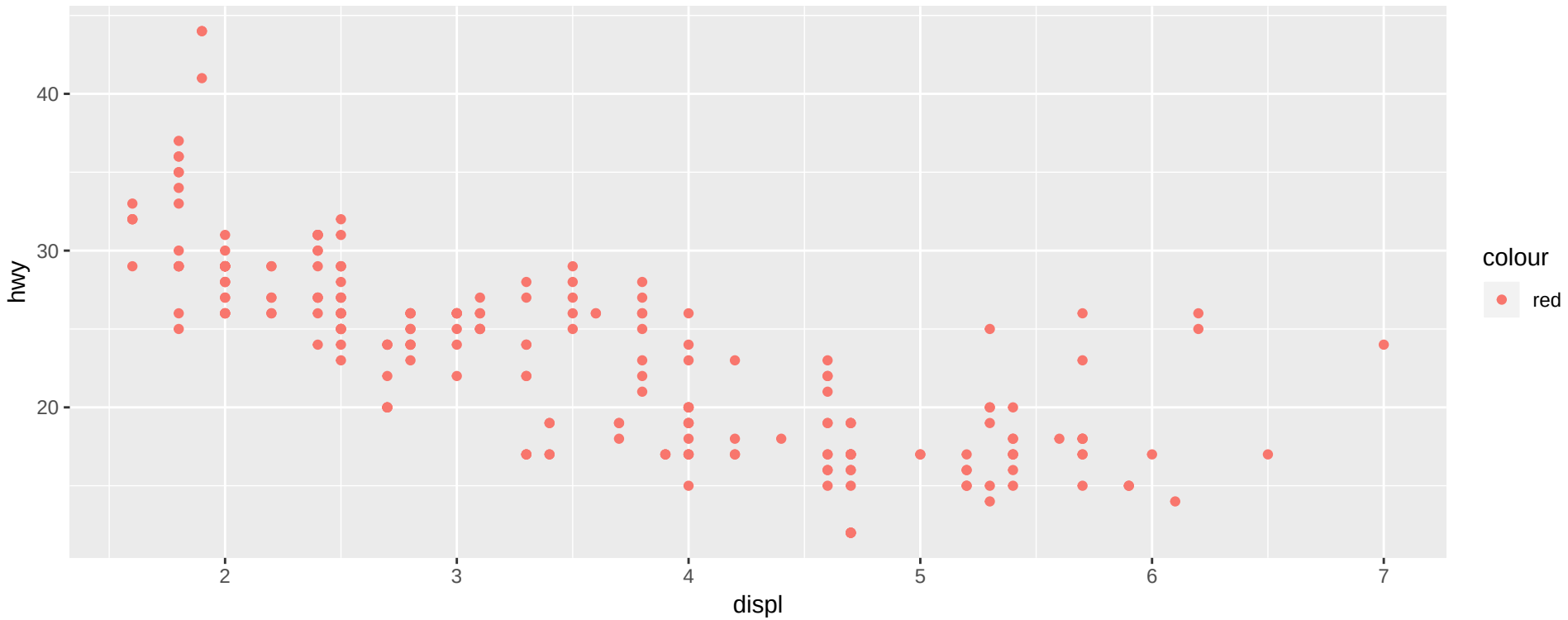
```
ggplot(mpg)+
    geom_point(aes(displ, hwy), color="red")
```



ggplot2

- `color= "red"` perdavus `(aes)` viduje, visį taškai nudažomi blyškiai raidonai, bei ggplot2 galvoja, kad tai kažkoks grupavimas, todėl sukuria legendą

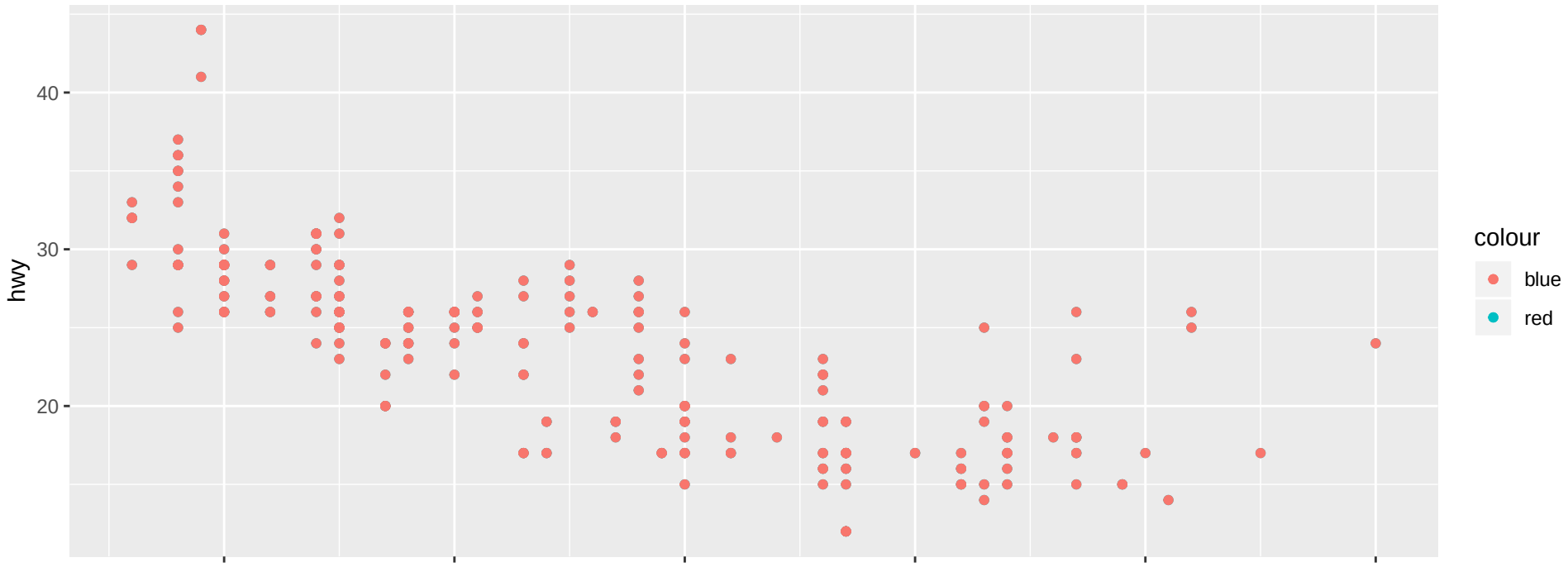
```
ggplot(mpg)+
    geom_point(aes(displ, hwy, color="red"))
```



ggplot2

- `color= "red"` perdavus `(aes)` viduje, visį taškai nudažomi blyškiai raidonai, bei ggplot2 galvoja, kad tai kažkoks grupavimas, todėl sukuria legendą

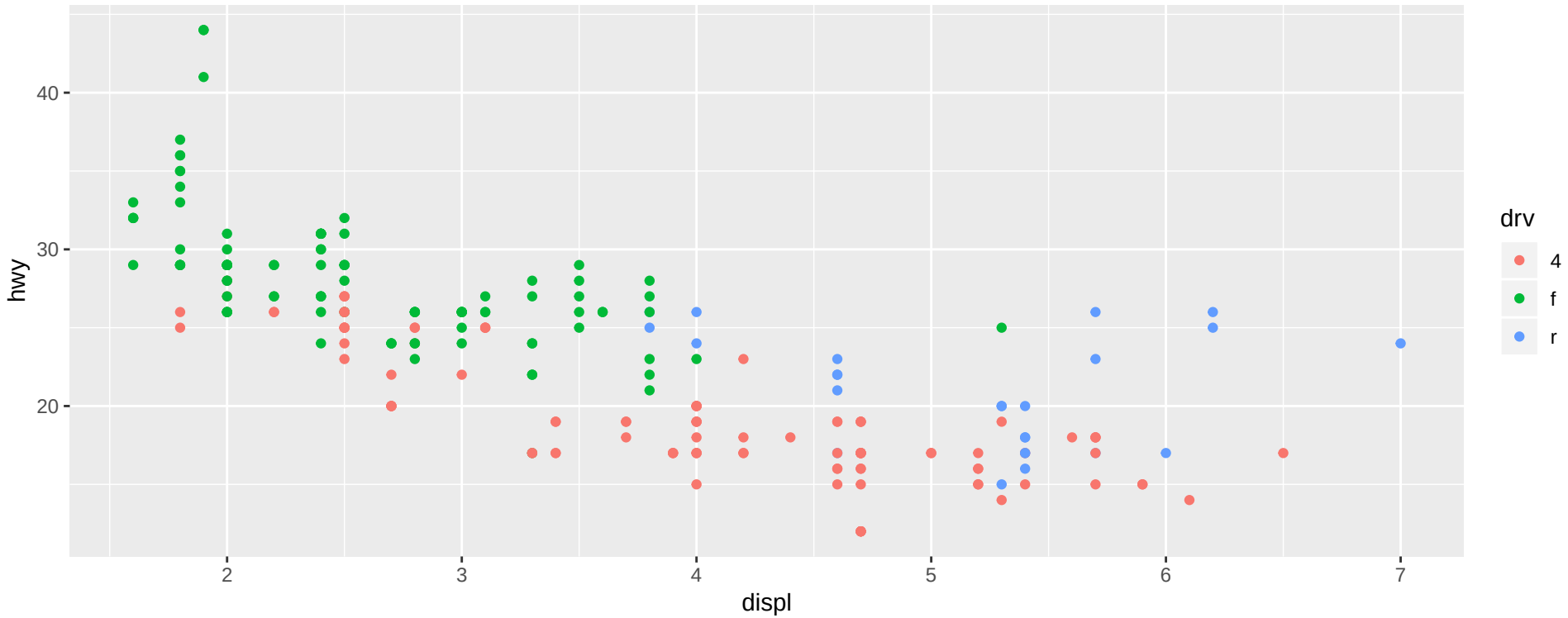
```
ggplot(mpg)+
    geom_point(aes(displ, hwy, color="red"))+
    geom_point(aes(displ, hwy, color="blue"))
```



ggplot2

- `aes()` jeigu nekinta, gali būti iškeltas į `ggplot()` dalį

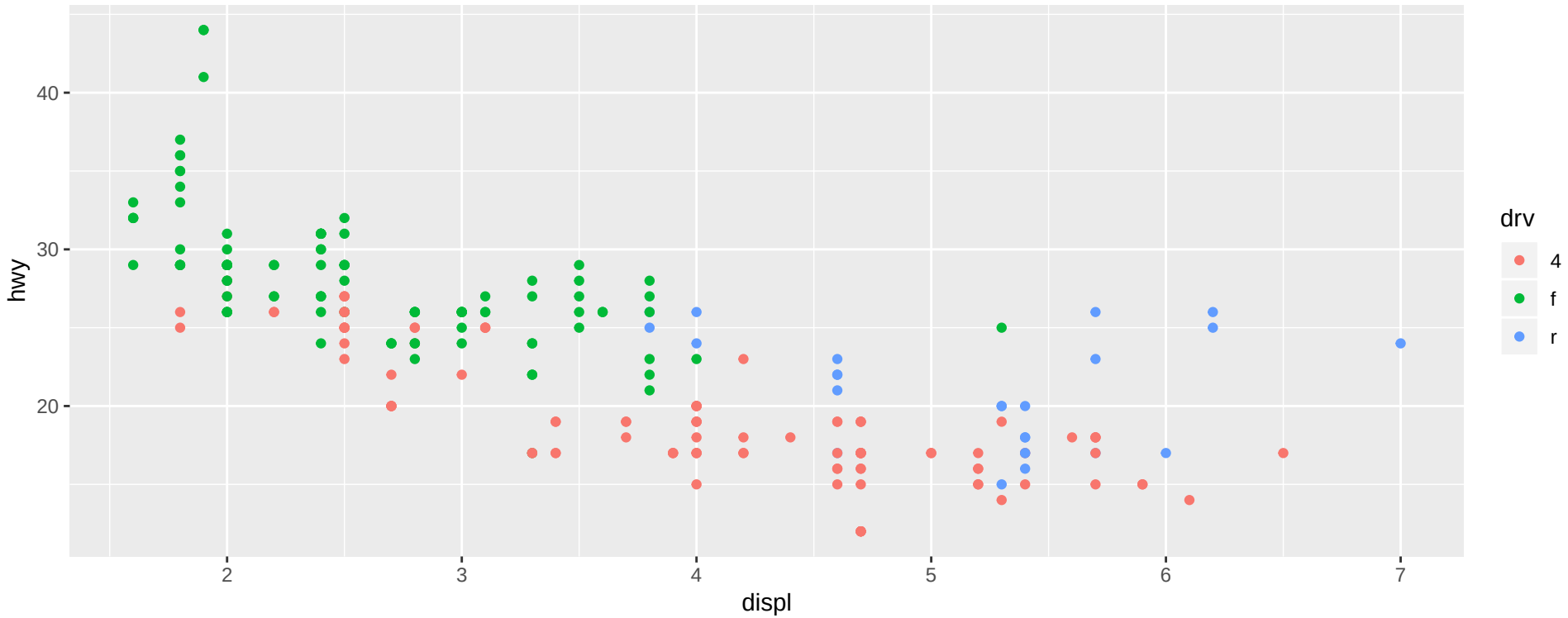
```
ggplot(mpg, aes(displ, hwy))+
    geom_point(aes(color=drv))
```



ggplot2

- `aes()` jeigu nekinta, gali būti iškeltas į `ggplot()` dalį

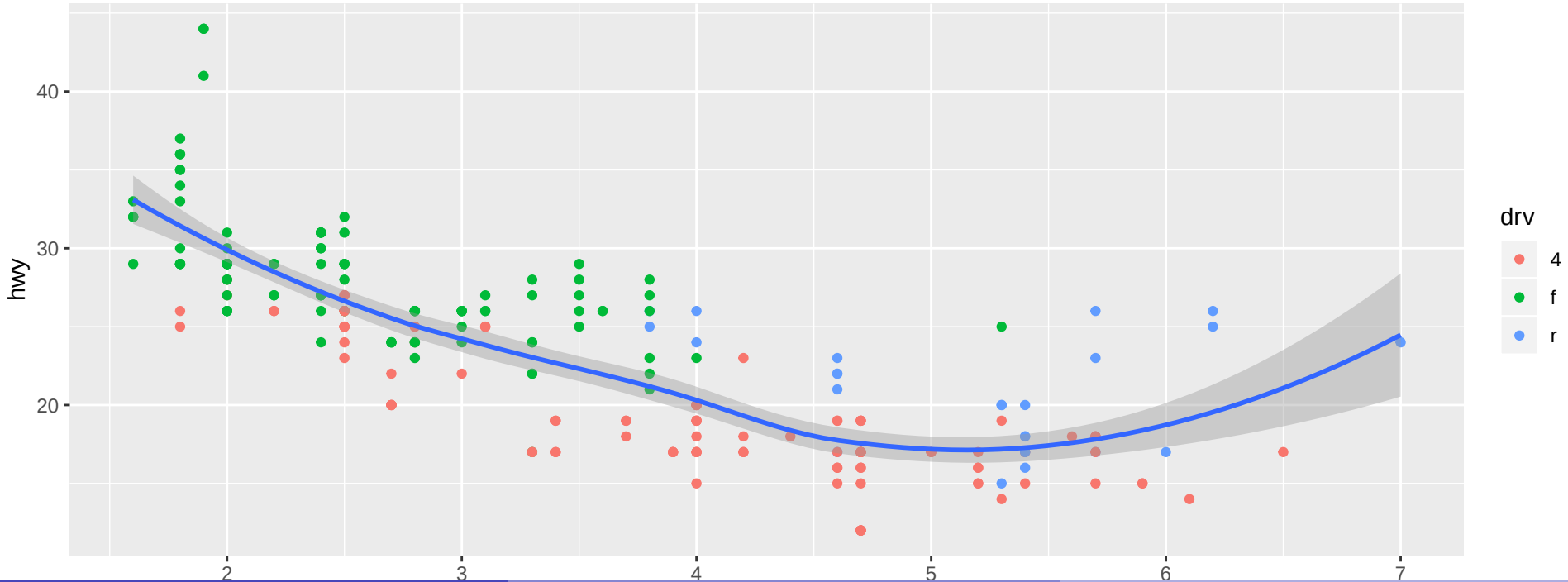
```
ggplot(mpg, aes(displ, hwy, color=drv))+
    geom_point()
```



ggplot2

- jeigu duomenų suskirstymas pačiame geom objekte, tada tai neveikia kitų geom objektų

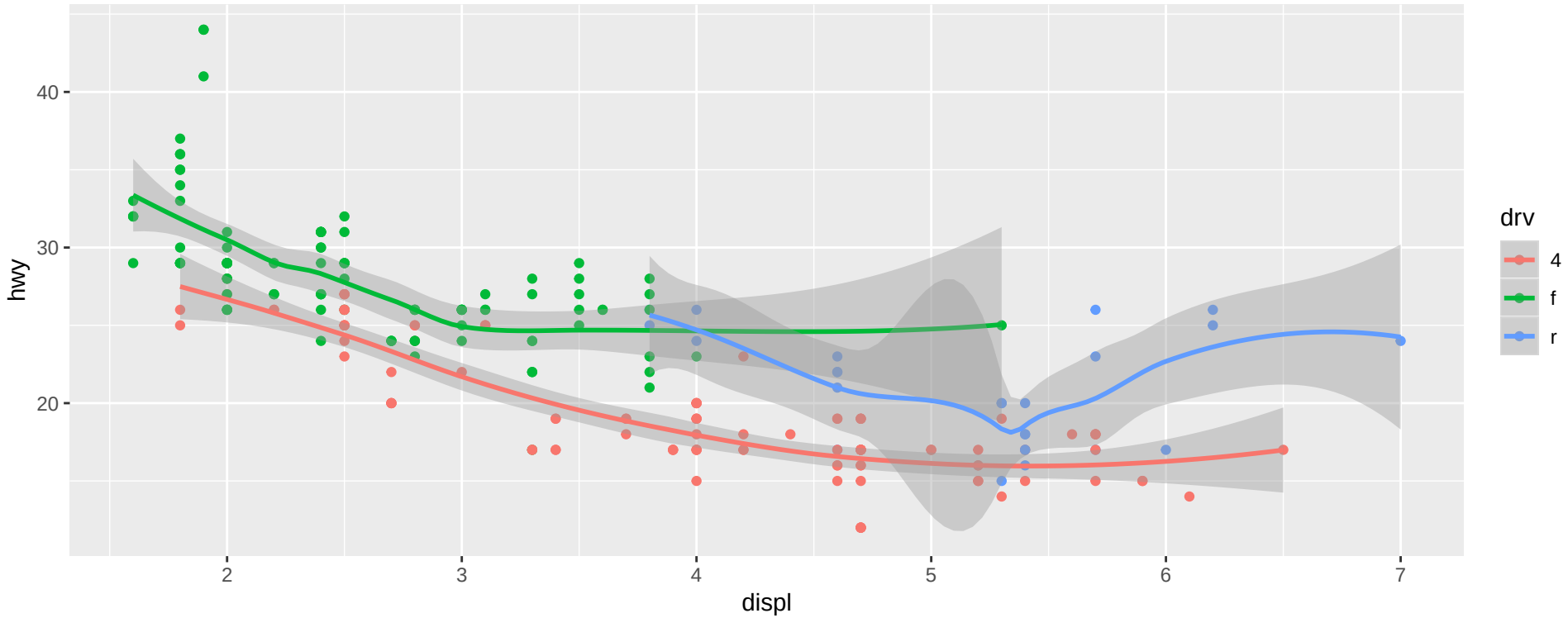
```
ggplot(mpg, aes(displ, hwy))+
    geom_point(aes(color=drv))+
    geom_smooth()
## `geom_smooth()` using method = 'loess' and formula 'y ~ x'
```



ggplot2

- jeigu duomenų suskirstymas bazinėje `ggplot()` komandoje, tada...

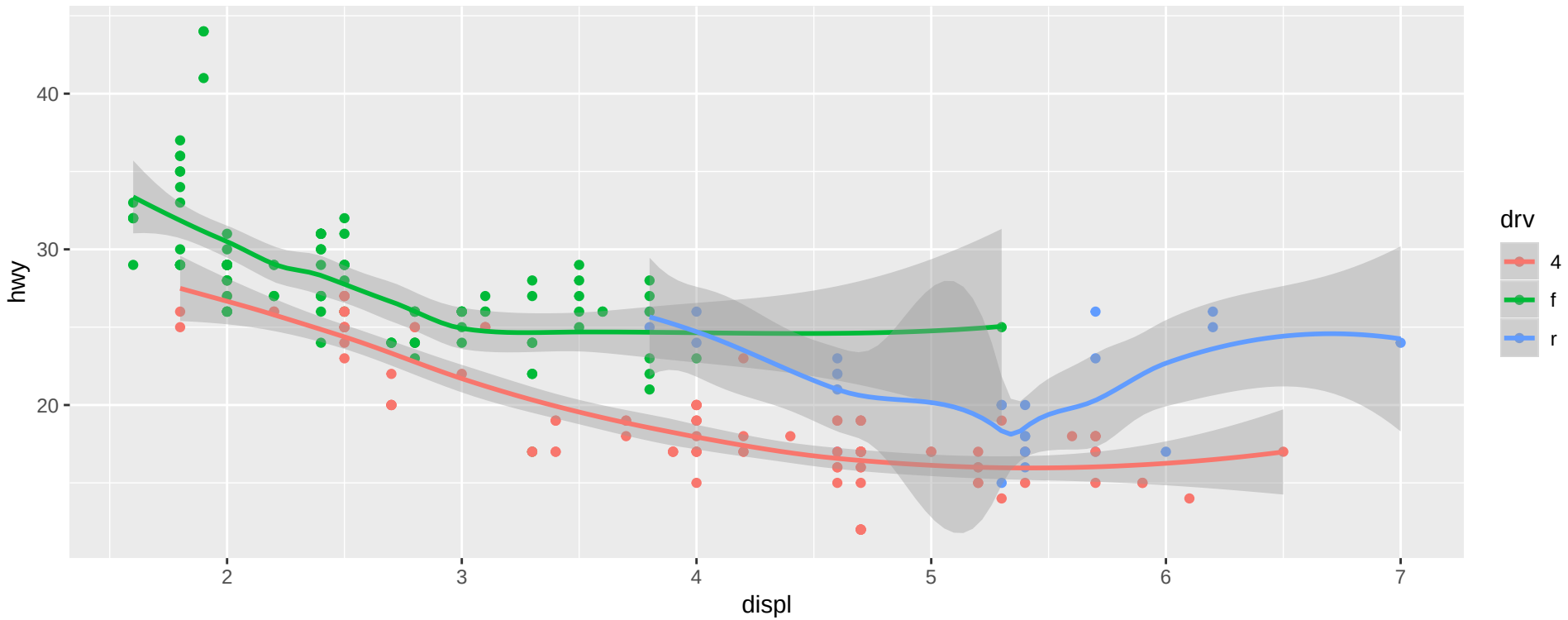
```
ggplot(mpg, aes(displ, hwy, color=drv))+
    geom_point()+
    geom_smooth()
## `geom_smooth()` using method = 'loess' and formula 'y ~ x'
```



ggplot2

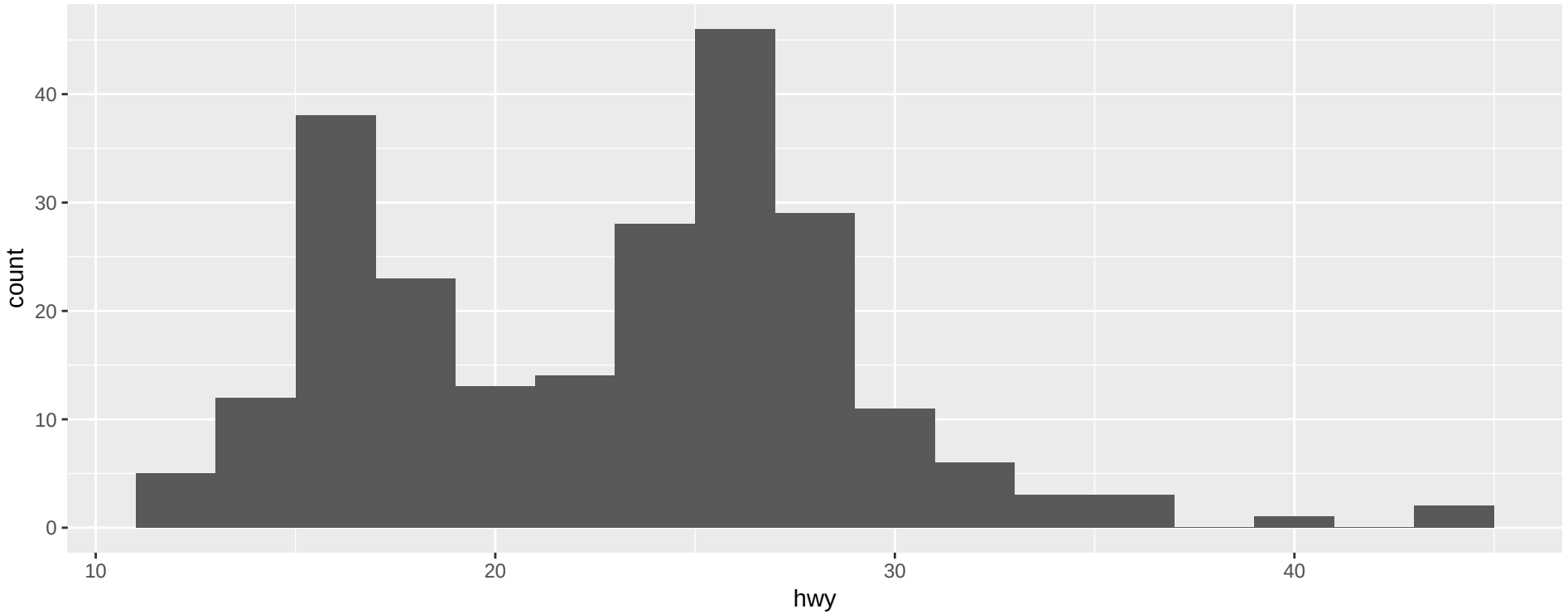
- tolygu...nes bazinė komanda perduoda parameturs geom_ objektams

```
ggplot(mpg, aes(displ, hwy, color=drv))+
    geom_point(aes(color=drv))+
    geom_smooth(aes(color=drv))
## `geom_smooth()` using method = 'loess' and formula 'y ~ x'
```



ggplot2 histograma

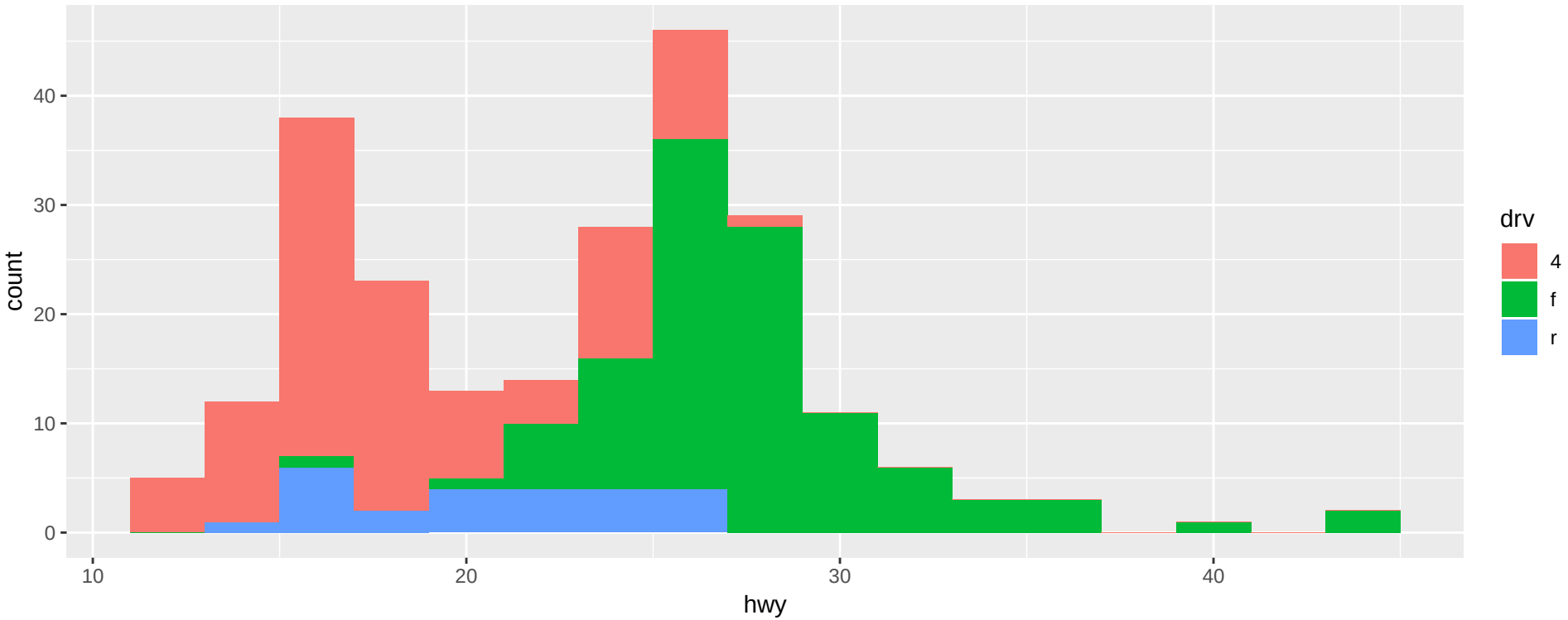
```
ggplot(mpg, aes(hwy))+
    geom_histogram(binwidth = 2)
```



ggplot2

`fill` užpildo, `color` daro linijas

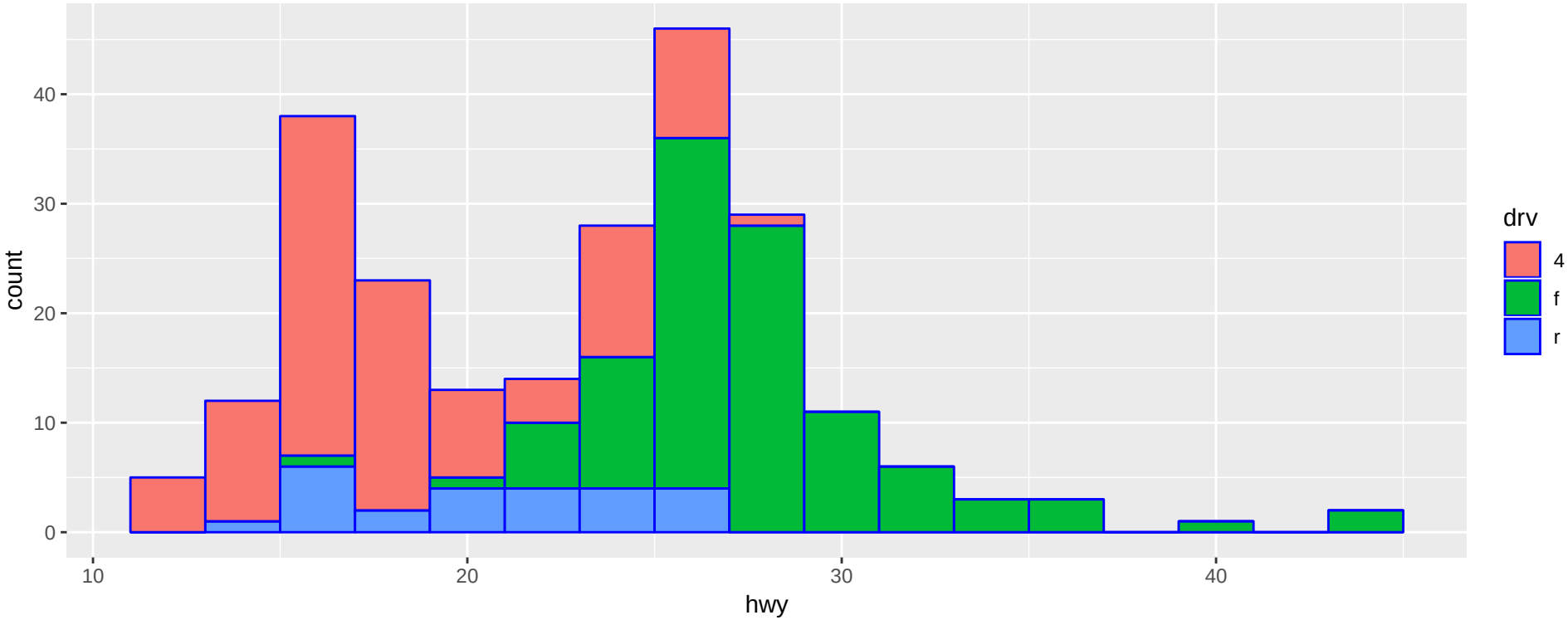
```
ggplot(mpg, aes(hwy))+
    geom_histogram(binwidth = 2, aes(fill=drv))
```



ggplot2

`fill` užpildo, `color` daro linijas (čia color už `aes` ribų, nes veikia ne pavienį subset, o visą geom obejktą)

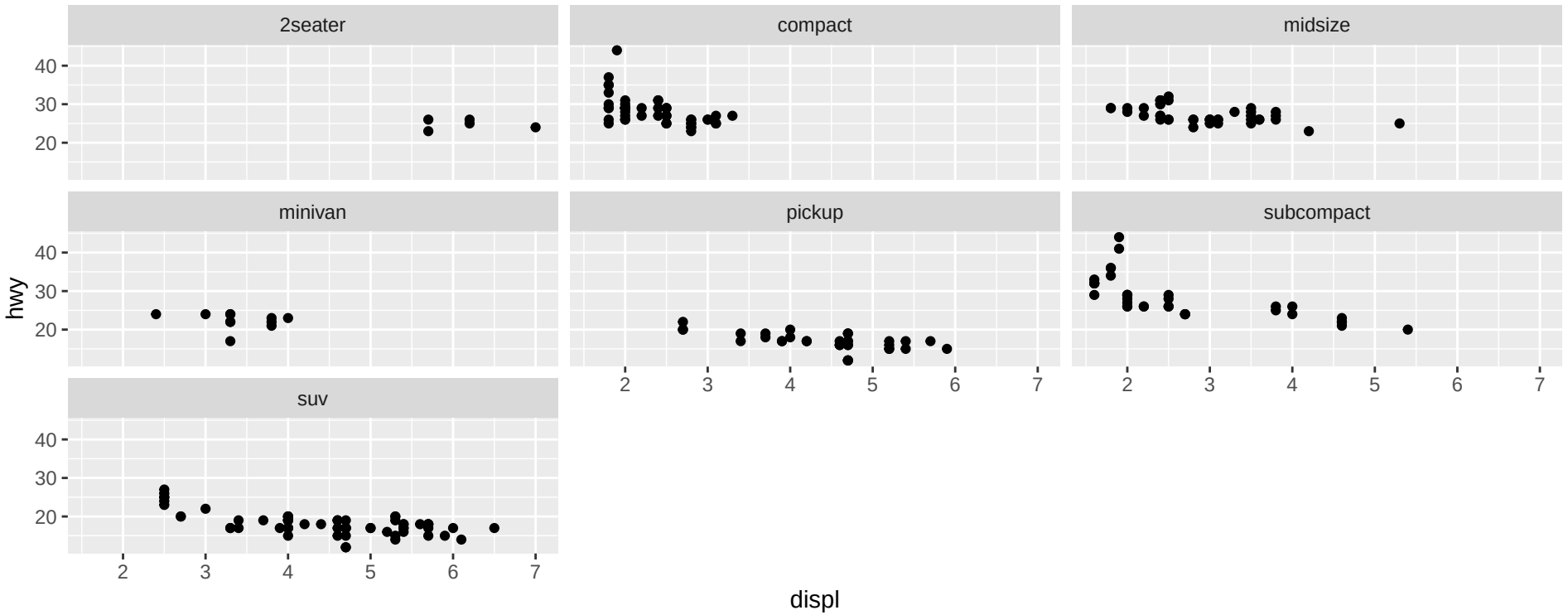
```
ggplot(mpg, aes(hwy))+
    geom_histogram(binwidth = 2, aes(fill=drv), color="blue")
```



ggplot2

- kondicionalūs grafikai: suskaidymas vieno grafiko į kelis skirtingus.

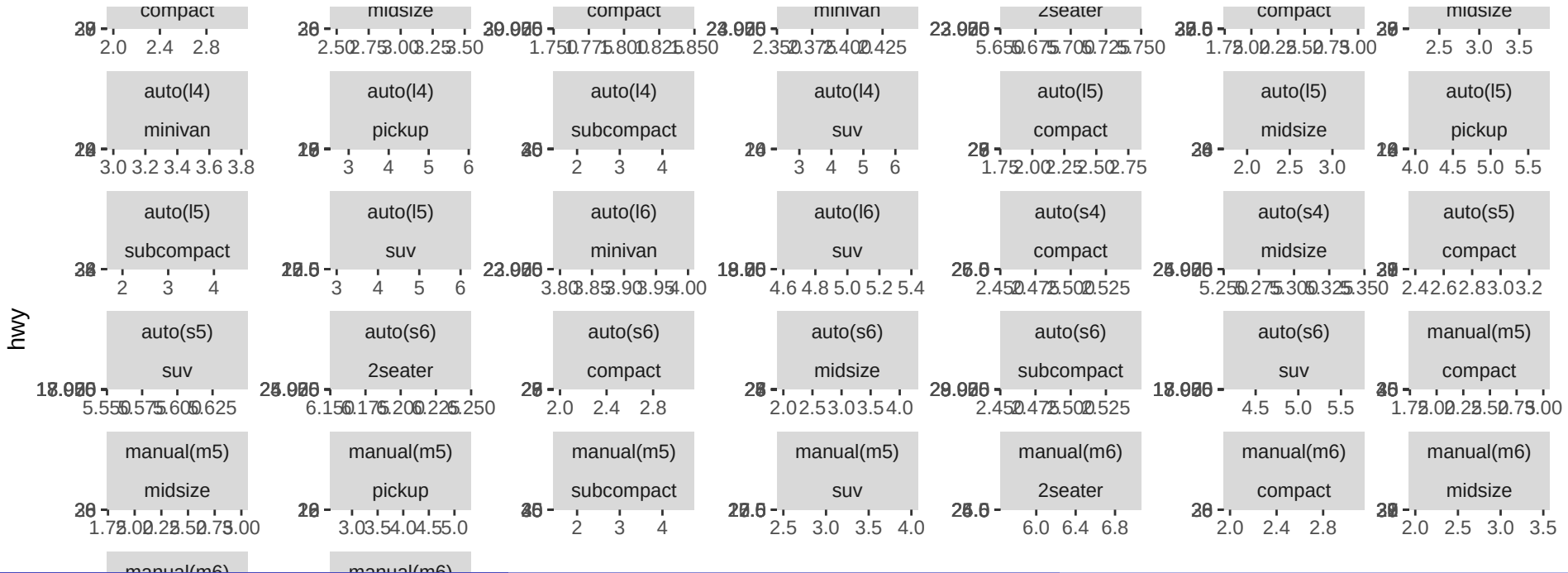
```
ggplot(mpg, aes(displ, hwy))+
    geom_point()+
    facet_wrap(~class)
```



ggplot2

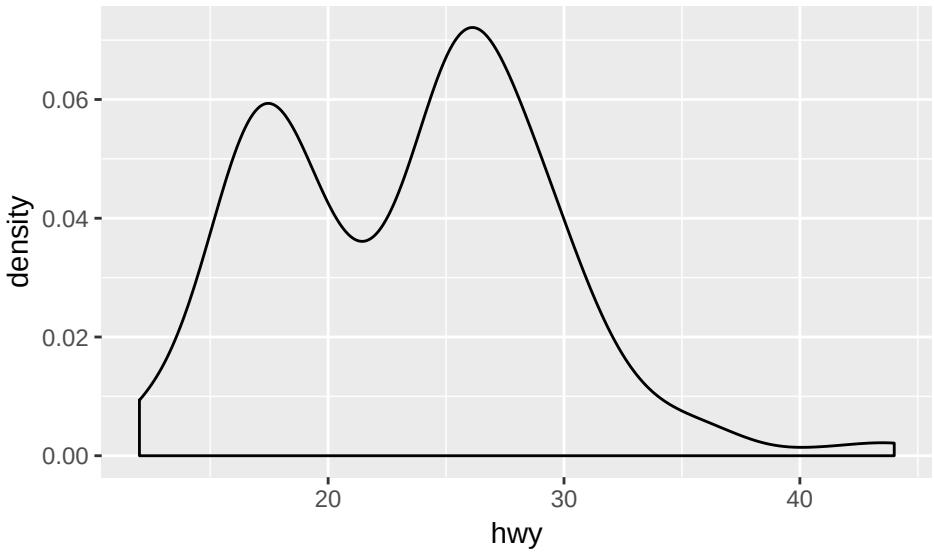
- kondicionalūs grafikai: suskaidymas vieno grafiko į kelis skirtingus
- naudojant labai skirtingus duomenis, `scales="free"` kiekvienam grafikui leidžia individualiai pasirinkti jam tinkamą koordinačių sistema

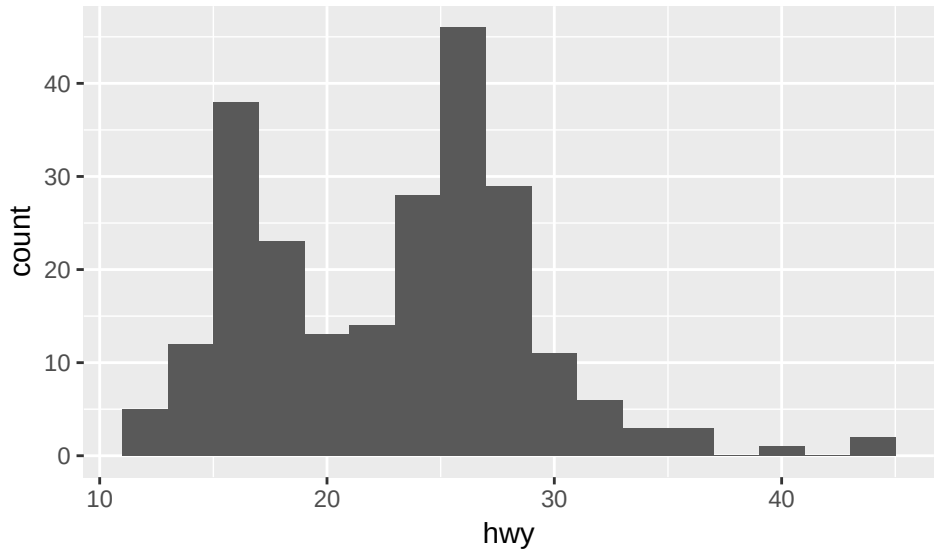
```
ggplot(mpg, aes(displ, hwy))+
    geom_point()+
    facet_wrap(trans~class, scales = "free")
```



ggplot2 density plot

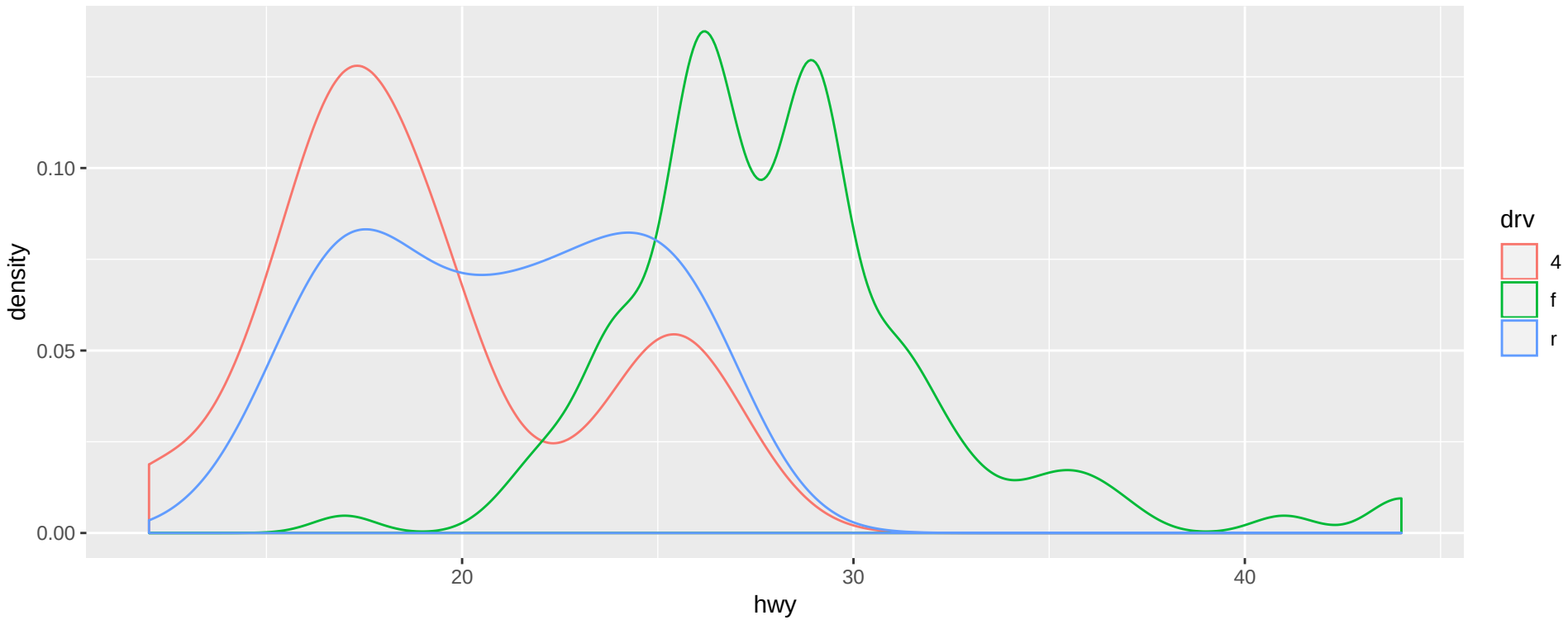
```
library(gridExtra) #padeda sudėti kelis į vieną
plot1 <- ggplot(mpg, aes(hwy))+geom_density()
plot2 <- ggplot(mpg, aes(hwy))+geom_histogram(binwidth = 2)
grid.arrange(plot1, plot2, ncol=2)
```





ggplot2

```
ggplot(mpg, aes(hwy))+
    geom_density(aes(col=drv))
```

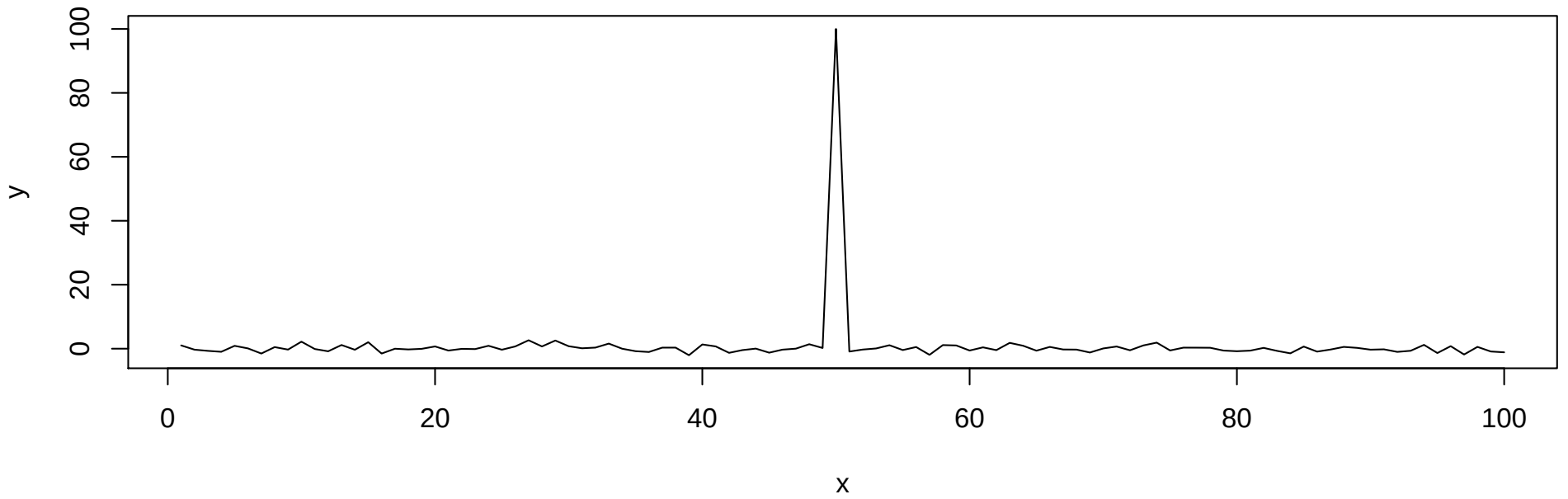


Outlayeriai

```
df<- data.frame(x=1:100, y=rnorm(100))
df[50,2] <-100
```

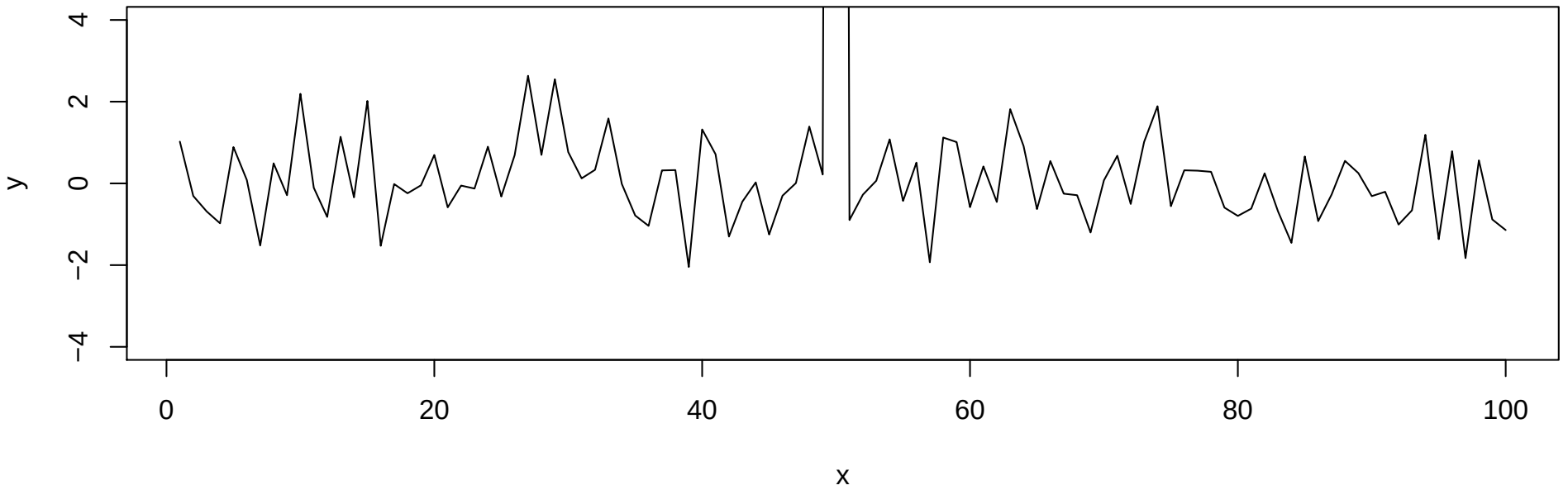
Outlayeriai

```
with(df, plot(x,y, type="l"))
```



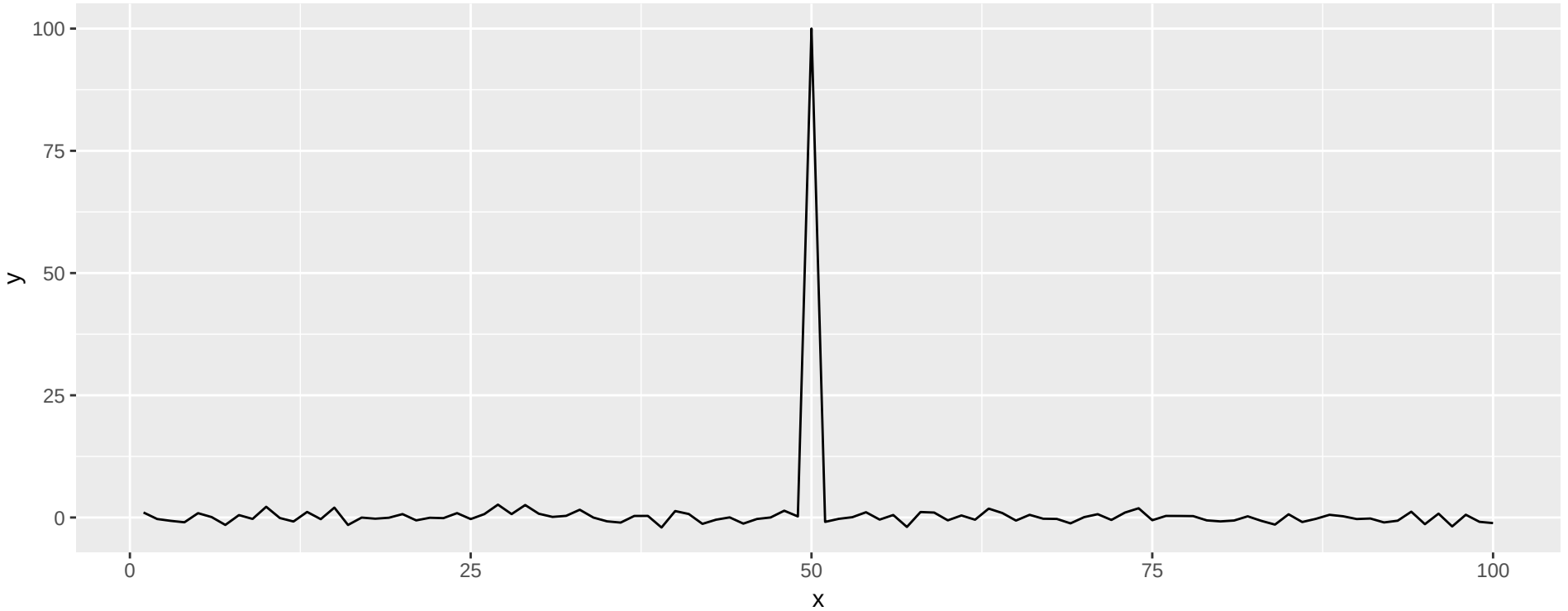
Outlayeriai

```
with(df, plot(x,y, type="l", ylim=c(-4,4)))
```



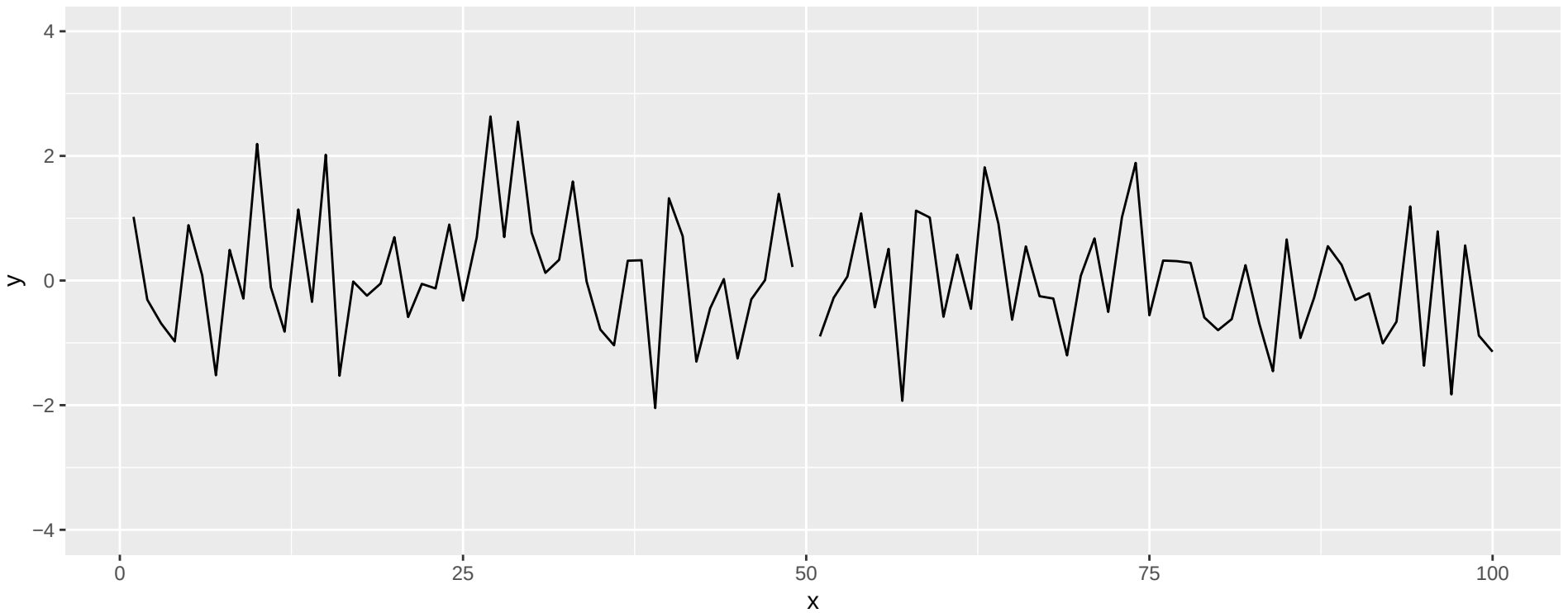
Outlayeriai

```
ggplot(df, aes(x,y))+
    geom_line()
```



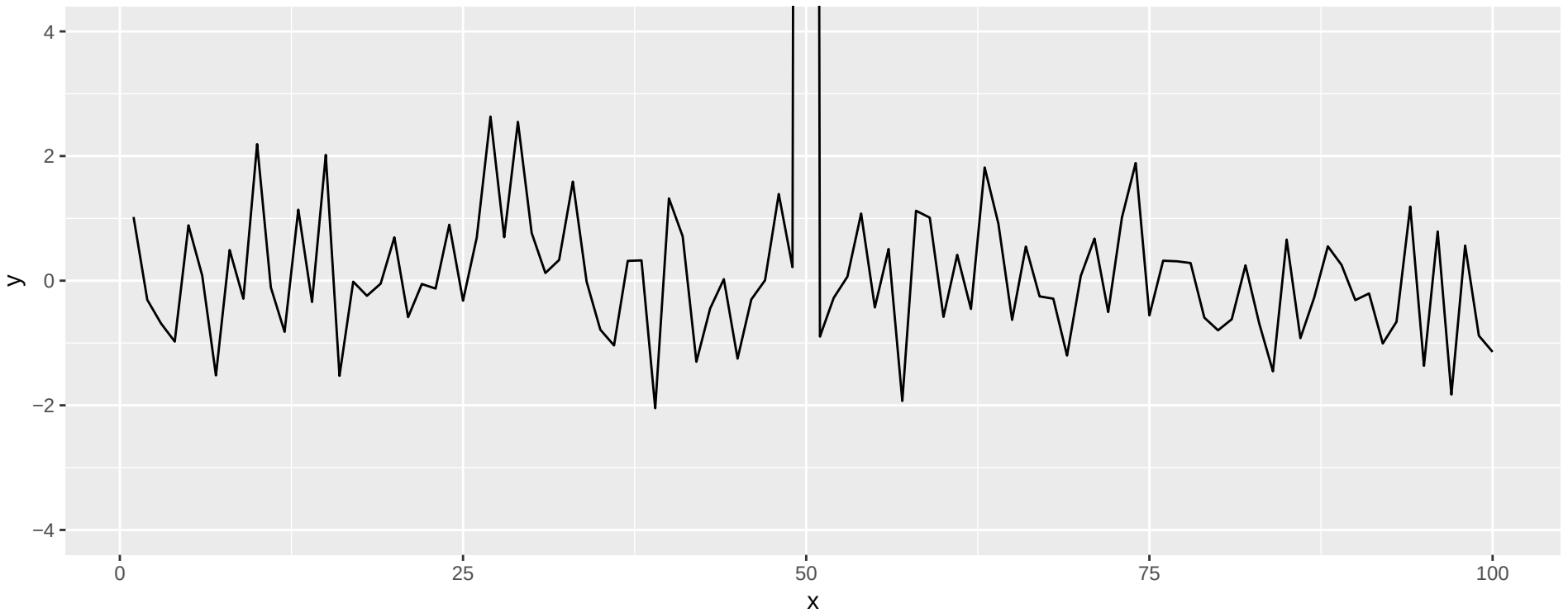
Outlayeriai

```
ggplot(df, aes(x=x,y=y))+
    geom_line()+
    scale_y_continuous(limits=c(-4,4))
```



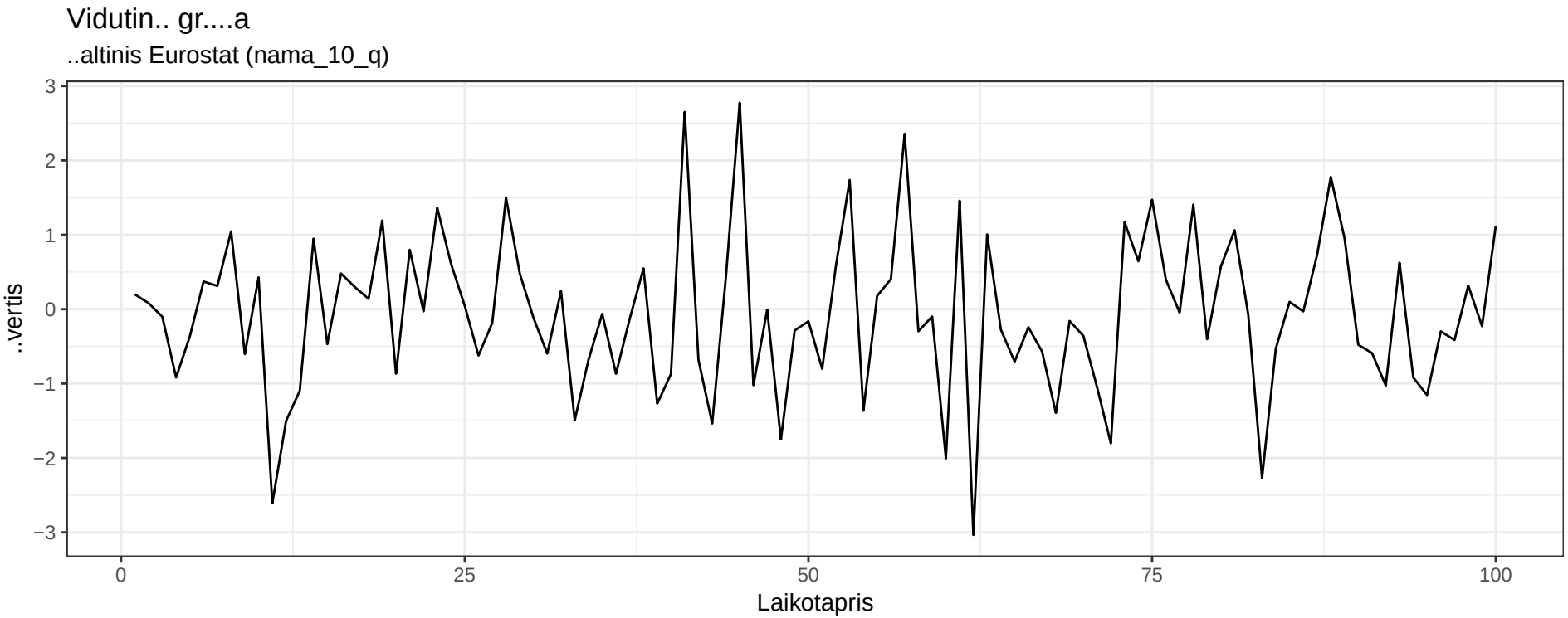
Outlayeriai

```
ggplot(df, aes(x=x,y=y))+
    geom_line()+
    coord_cartesian(ylim=c(-4,4))
```



labs()

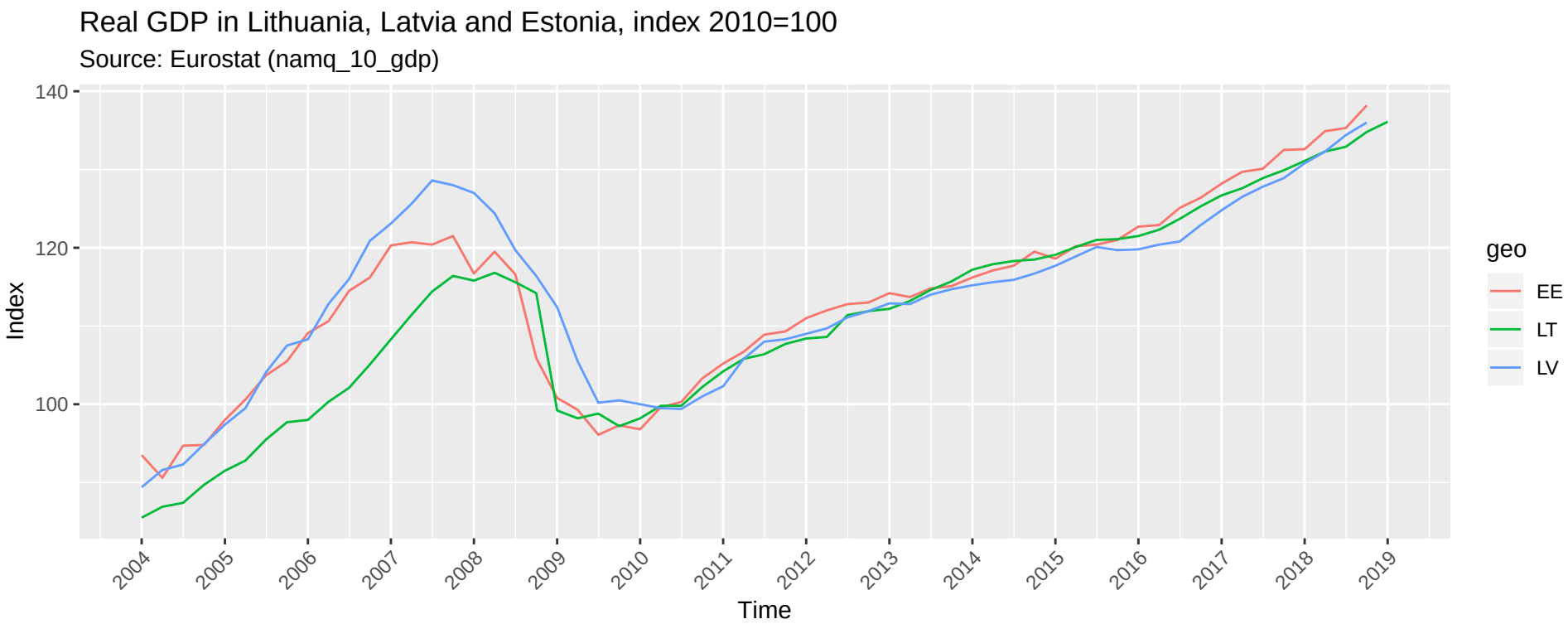
```
df<- data.frame(x=1:100, y=rnorm(100))
ggplot(df, aes(x,y))+theme_bw()+
    geom_line()+
    labs(x="Laikotapris", y="Įvertis", title= "Vidutinė grąža",
         subtitle = "Šaltinis Eurostat (nama_10_q)")
```



1 Hands on...

- parašykite skriptą, kuris, importuoja duomenis iš Eurostat
- apdoroja duomenis su dplyr
- nubraižo grafiką su `geom_line()`
- Duomenys:
  - ketvirtiniai BVP duomenys iš `namq_10_gdp`
  - Lietuvos, Latvijos ir Estijos duomenys
  - Gross domestic product at market prices
  - Seasonally and calendar adjusted data
  - nuo 2004 m.
  - Chain linked volumes, index 2010=100

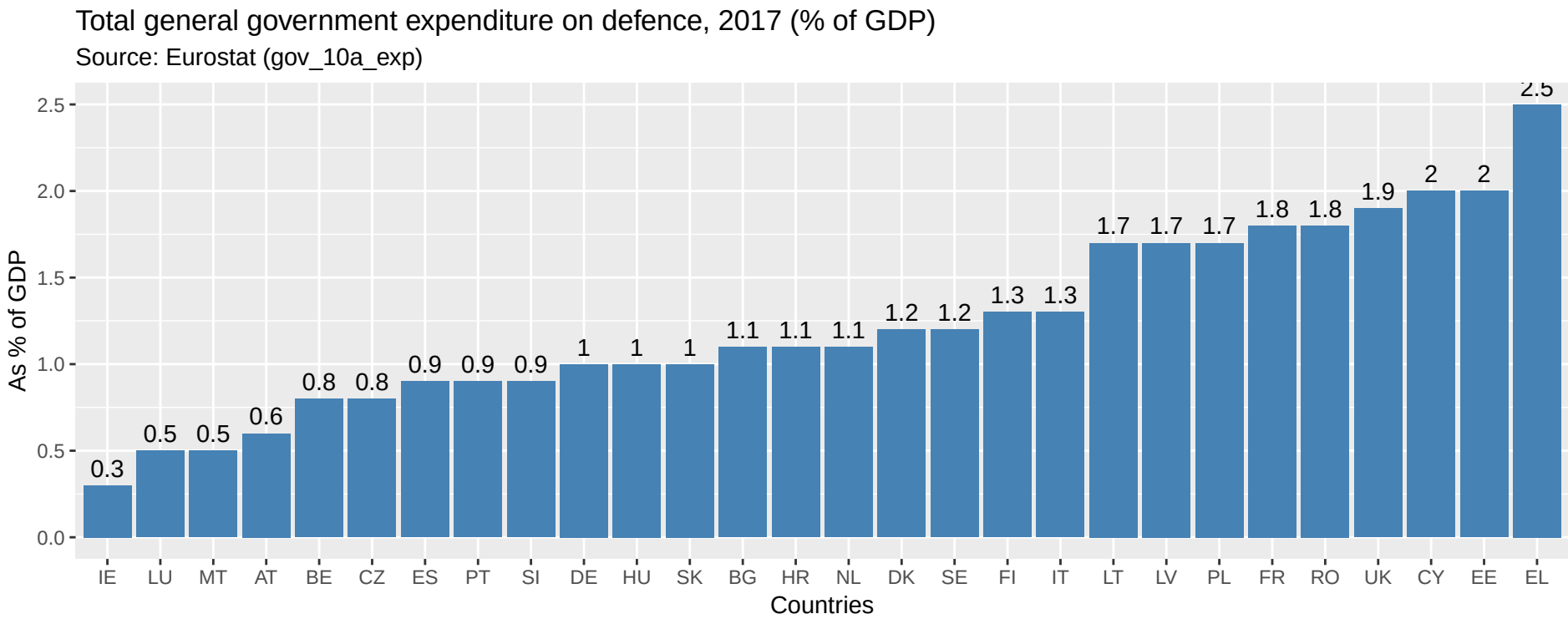
1 Hands on...


Real GDP in Lithuania, Latvia and Estonia, index 2010=100
Source: Eurostat (namq_10_gdp)
Index
140
120
100
2004
2005
2006
2007
2008
2009
2010
2011
2012
2013
2014
2015
2016
2017
2018
2019
Time
geo
EE
LT
LV

2 Hands on...

- parašykite skirptą, kuris, importuoja duomenis iš Eurostat
- apdoroja duomenis su dplyr
- nubraižo grafiką `geom_bar()`
- Duomenys:
  - Metiniai valstybės išlaidos duoenys pagal išlaidų funkcijas: `gov_10a_exp`
  - visos ES šalys! (28) (patarimas, su `Sublime` susitvarkyti 28 šalis)
  - Total expenditure
  - General government
  - 2017m
  - procentais nuo BVP

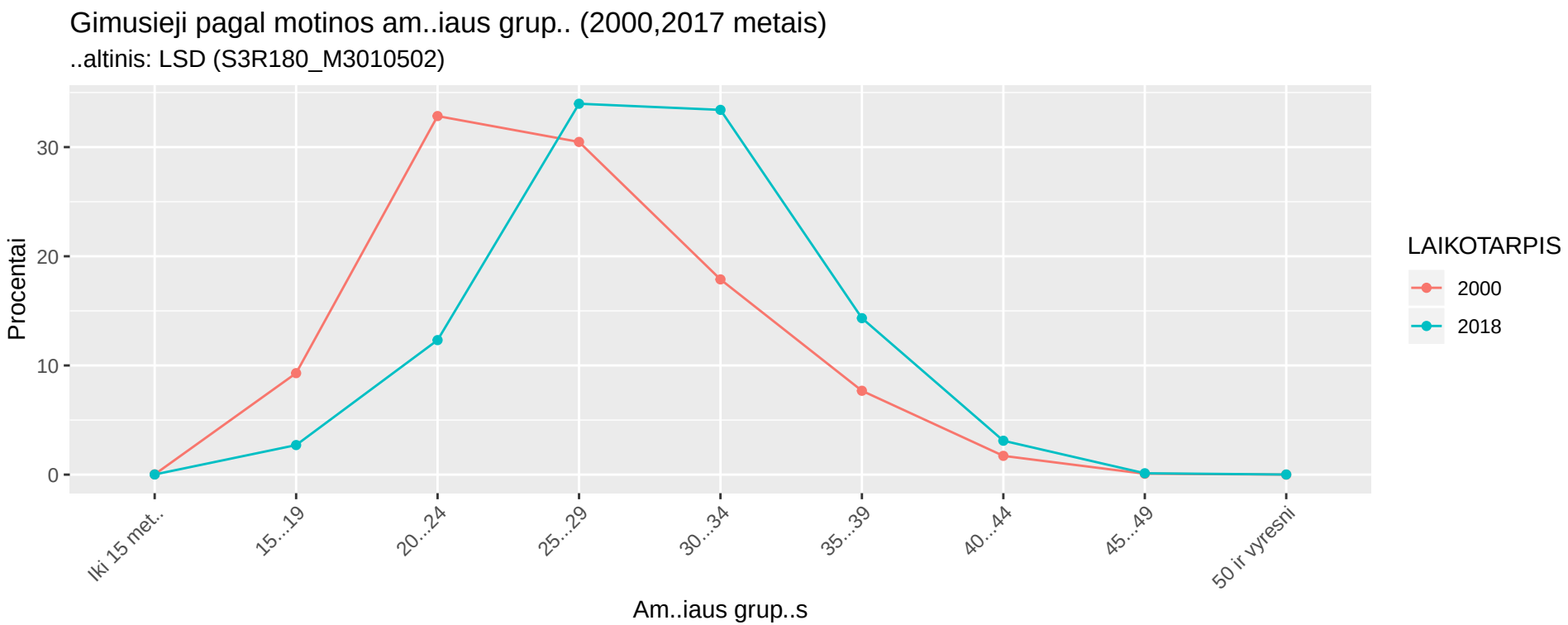
2 Hands on...


Total general government expenditure on defence, 2017 (% of GDP)
Source: Eurostat (gov_10a_exp)
As % of GDP
2.5
2.0
1.5
1.0
0.5
0.0
0.3
0.5
0.5
0.6
0.8
0.8
0.9
0.9
0.9
1
1
1
1.1
1.1
1.1
1.2
1.2
1.3
1.3
1.7
1.7
1.7
1.8
1.8
1.9
2
2
2.5
IE
LU
MT
AT
BE
CZ
ES
PT
SI
DE
HU
SK
BG
HR
NL
DK
SE
FI
IT
LT
LV
PL
FR
RO
UK
CY
EE
EL
Countries

3 Hands on

- parašykite skirptą, kuris, importuoja duomenis iš LSD
- apdoroja duomenis su dplyr
- nubraižo grafiką
- Duomenys:
  - Gimusieji
  - Gyvenamoji vietovė | Motinos amžius (2000 - 2017)
  - Kodą susirasti iš LSD meta failo (patarimas parsisiųsti meta csv failą ...)
  - Nenurodytą amžių išmesti
  - Jeigu reikia, `character` pasiverskite `factor` ir sutvarkykite `levels`.

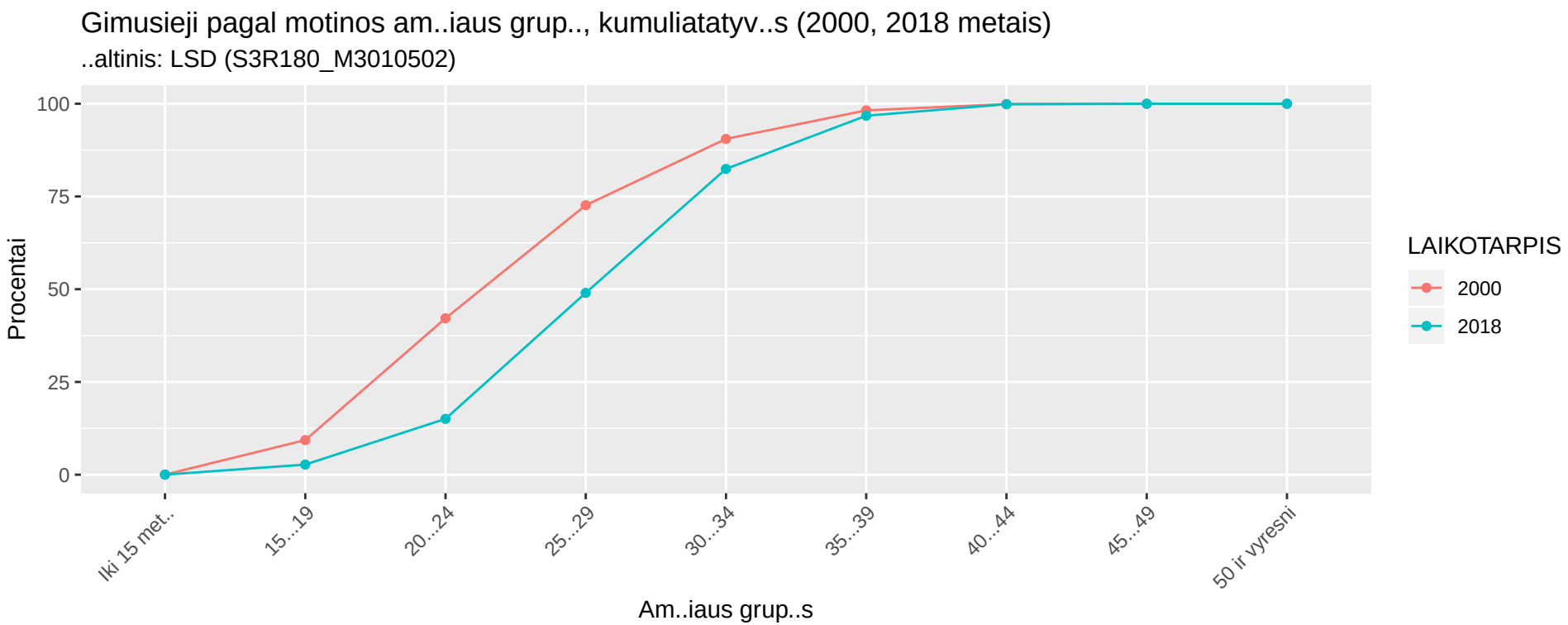
3 Hands on


Gimusieji pagal motinos am..iaus grup.. (2000,2017 metais)
..altinis: LSD (S3R180_M3010502)
Procentai
30
20
10
0
Iki 15 met..
15...19
20...24
25...29
30...34
35...39
40...44
45...49
50 ir vyresni
Am..iaus grup..s
LAIKOTARPIS
2000
2018

4 Hands on

- parašykite skirptą, kuris, importuoja duomenis iš LSD
- apdoroja duomenis su dplyr
- nubraižo grafiką
- Duomenys:
  - Gimusieji
  - Gyvenamoji vietovė | Motinos amžius (2000 - 2017)
  - Kodą susirasti iš LSD meta failo (patarimas parsisiųsti meta csv failą ...)
  - Nenurodytą amžių išmesti
  - `cumsum()` pateikia kumuliatyvią sumą
  - grafikas pateikia kumuliuotas gimdymo tikimybes dviejų metų (2000,2017) cohortoms atskirai

4 Hands on


Gimusieji pagal motinos am..iaus grup.., kumuliatatyv..s (2000, 2018 metais)
..altinis: LSD (S3R180_M3010502)
Procentai
100
75
50
25
0
Iki 15 met..
15...19
20...24
25...29
30...34
35...39
40...44
45...49
50 ir vyresni
Am..iaus grup..s
LAIKOTARPIS
2000
2018