Duomenų analizės įvadas

4.1. dalis

Justas Mundeikis

VU EVAF

2019-05-16

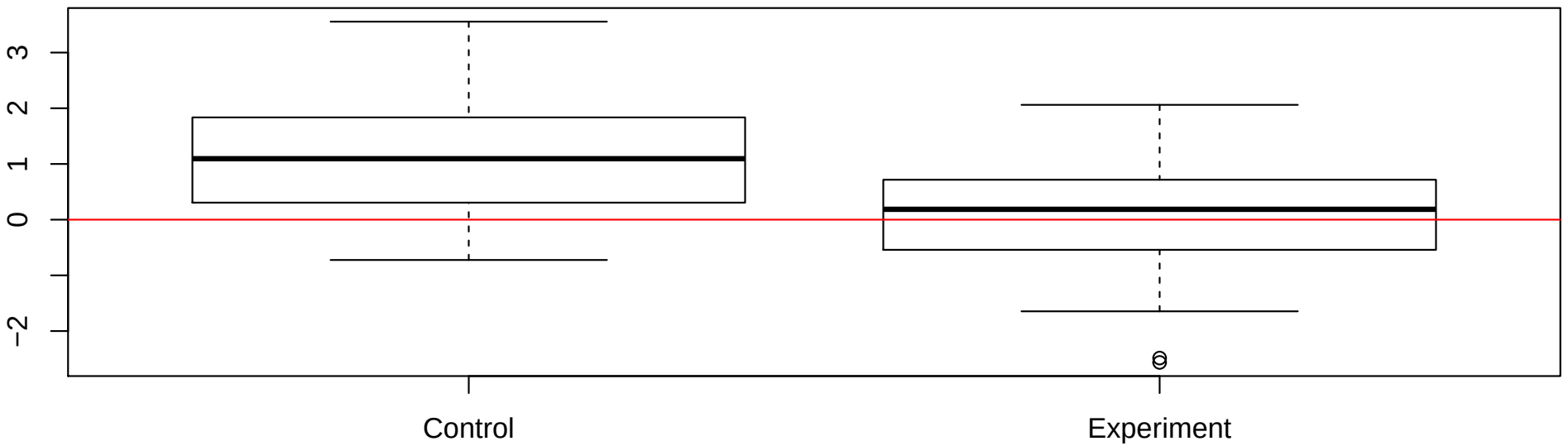
Turinys

Analitinių grafikų principai

Analitinių grafikų principai

1. Parodykite skirtumus
   - Hypotzių įrodymai visada yra relatyvus alternatyviai hipotezei
   - Ar grafikas atsako į klausimą: "Palyginus su kuo?"

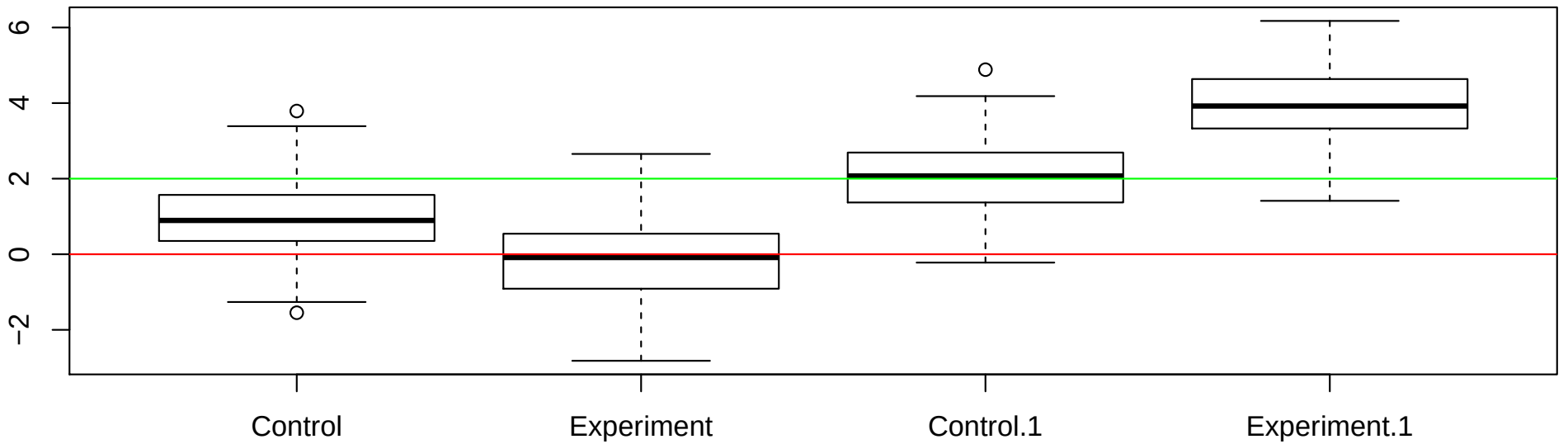
```
df <- data.frame(Control=rnorm(100,1), Experiment =rnorm(100,0))
boxplot(df)
abline(h=0, col="red")
```



Analitinių grafikų principai

2. Parodykite priežastinius-pasekminius ryšius
   - Nebūtinai tikras priežastinis ryšis, bet kaip Jūs / teorija mano

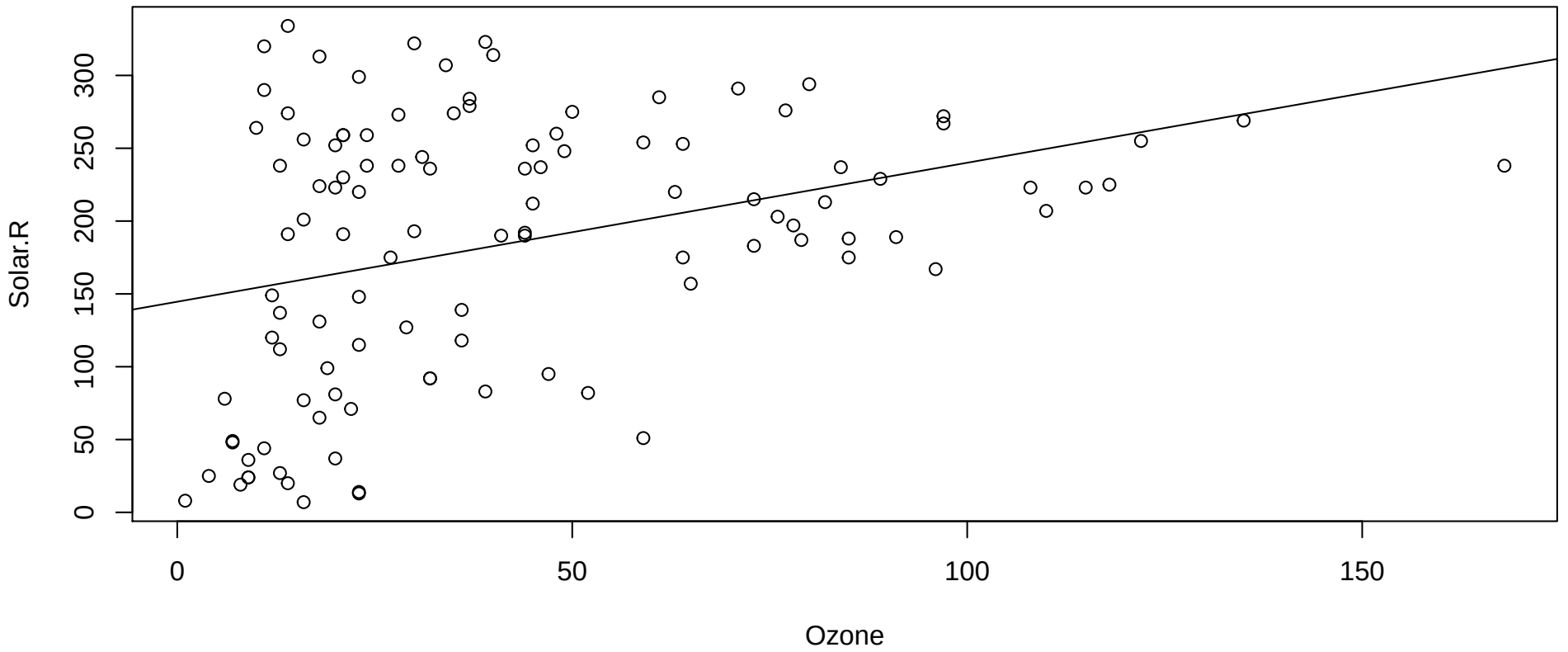
```
df <- data.frame(Control=rnorm(100,1), Experiment =rnorm(100,0), Control=rn
boxplot(df)
abline(h=0, col="red")
abline(h=2, col="green")
```



Analitinių grafikų principai

3. Parodykite *multivariate data*
   - *multivariate* = daugiau nei 2 kintamieji

```
with(airquality, plot(Ozone, Solar.R))
with(airquality, abline(lm(Solar.R~Ozone)))
```



Analitinių grafikų principai

4. Integruokite skirtingus įrodymus
   - dažnai grafikai yra iškalbingesni
   - tačiau kartais lentelės gali būti naudingesnės
   - grafikų, lentelių derinys
5. Tvarkingai aprašykite
   - Pavadinimai, ašys
   - Šaltiniai, geriausia nurodyti lentelės ID (pvz., Eurostat (nama_10_q))
6. *Content is king*
   - Jeigu neturite įdomios "istorijos", joks grafikas Jūsų neišgelbės

Šaltinis Edward Tufte (2006), Beutiful Evidence

Kam naudojami grafikai

- Suprasti duomenų savybes
- Atrasti dėsningumus
- Identifikuoti sąsajas, kurios kurtų prielaidas modeliavimui
- Komunikuoti gautus rezultatus

EDA grafikai

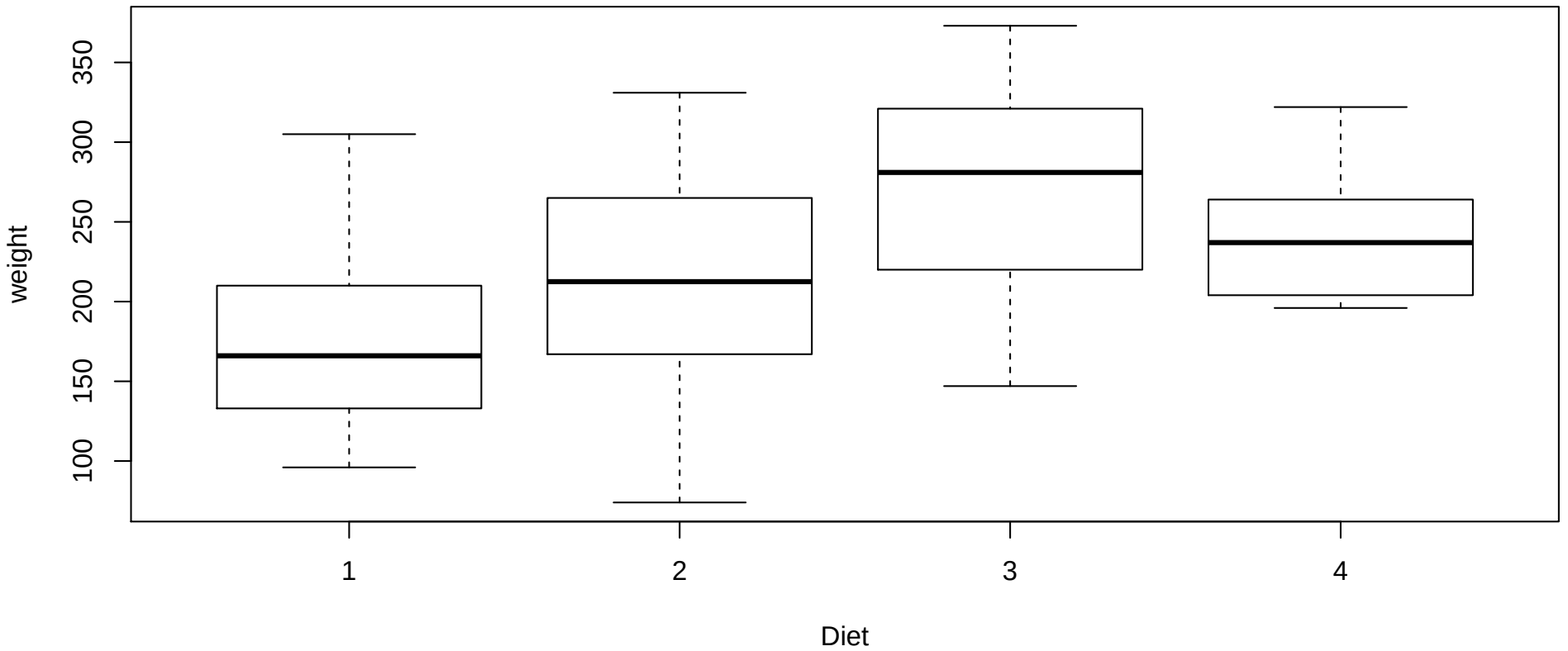
- EDA - *exploratory data analysis*
- greitai ir paprastai sugeneruoti grafikai
- daug grafikų
- padeda analitikui suprasti sąsajas
- grožis kuriamas su `ggplot2` (vėliau)

Summary

```
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

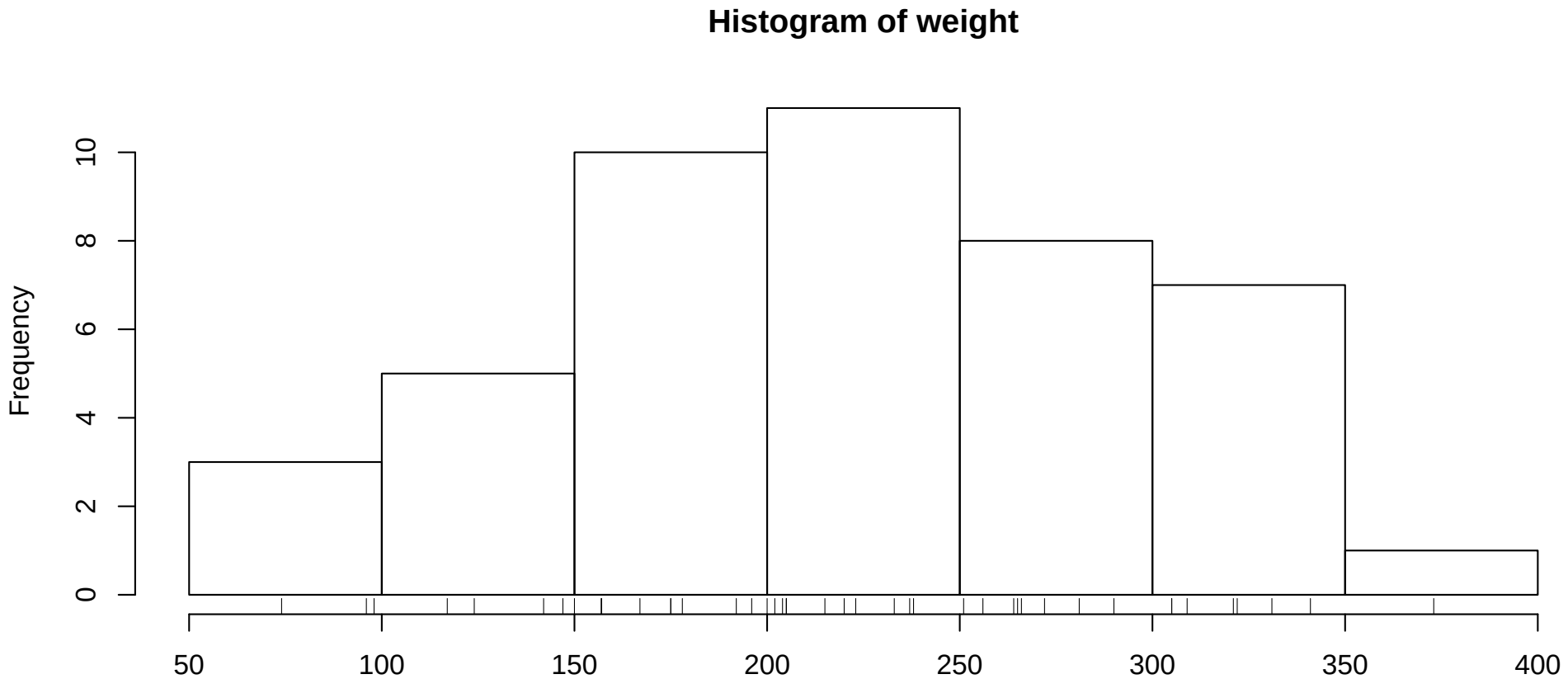
```
with(subset(ChickWeight, Time==21), boxplot(weight~Diet))
```



Histrogram

- `rug` plottina pavienius elementus
- stulpelių skaičius savo nuožiūrą

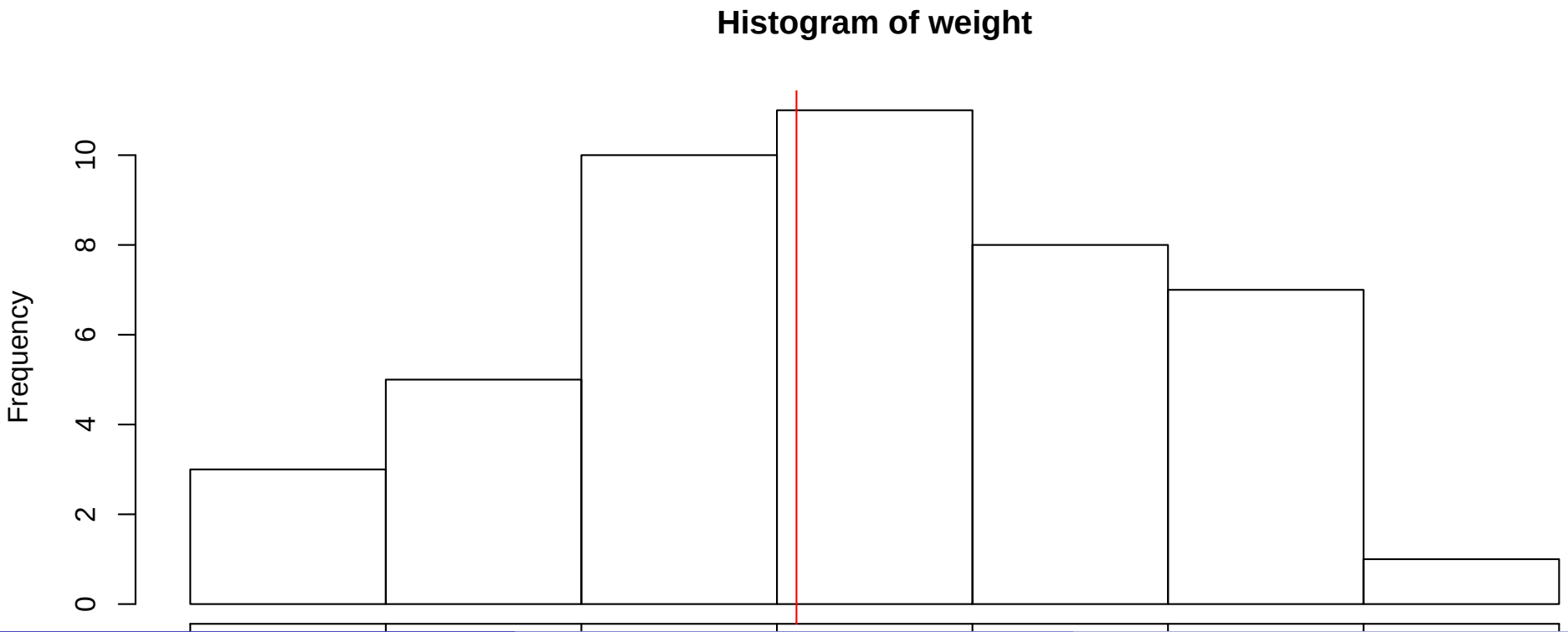
```
with(subset(ChickWeight, Time==21), hist(weight))
with(subset(ChickWeight, Time==21), rug(weight))
```



Histrogram

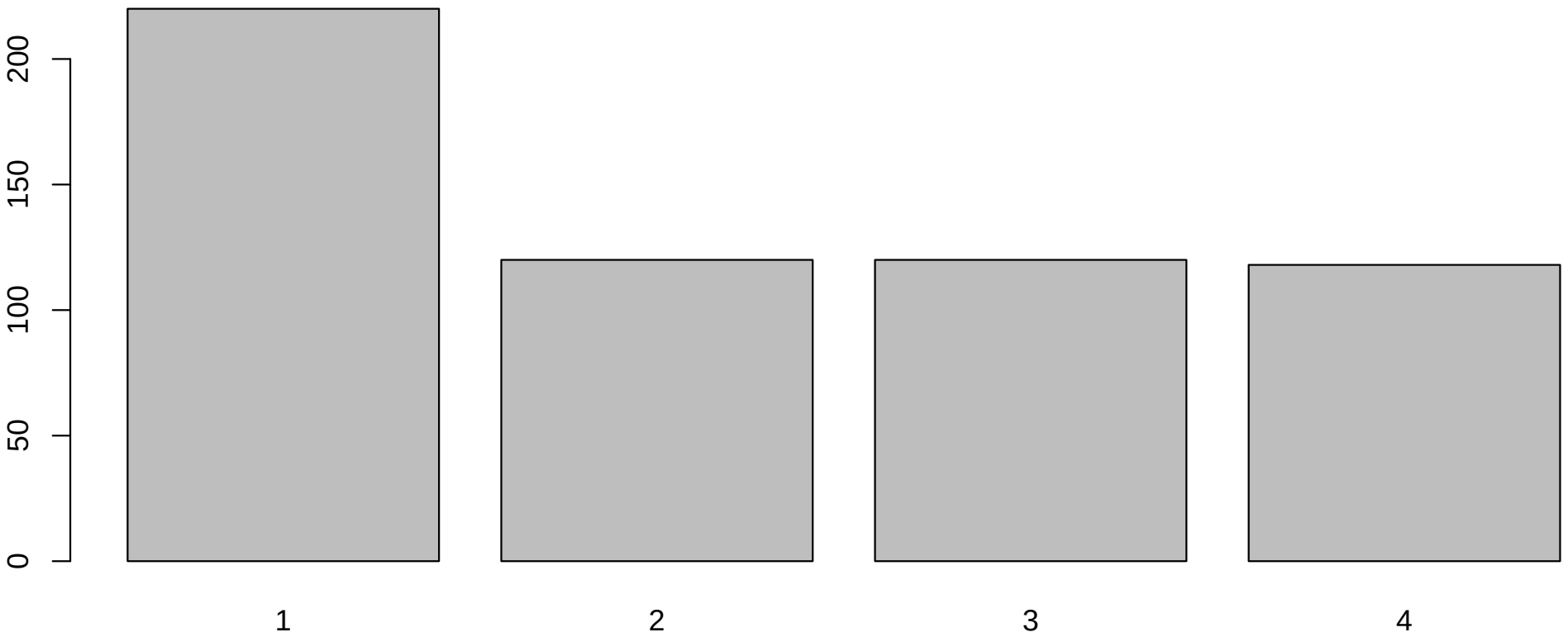
- `abline` brėžia tieses
- `v=..`
- `h=..`

```
with(subset(ChickWeight, Time==21), hist(weight))
abline(v=median(ChickWeight$weight[ChickWeight$Time==21]), col=2)
```



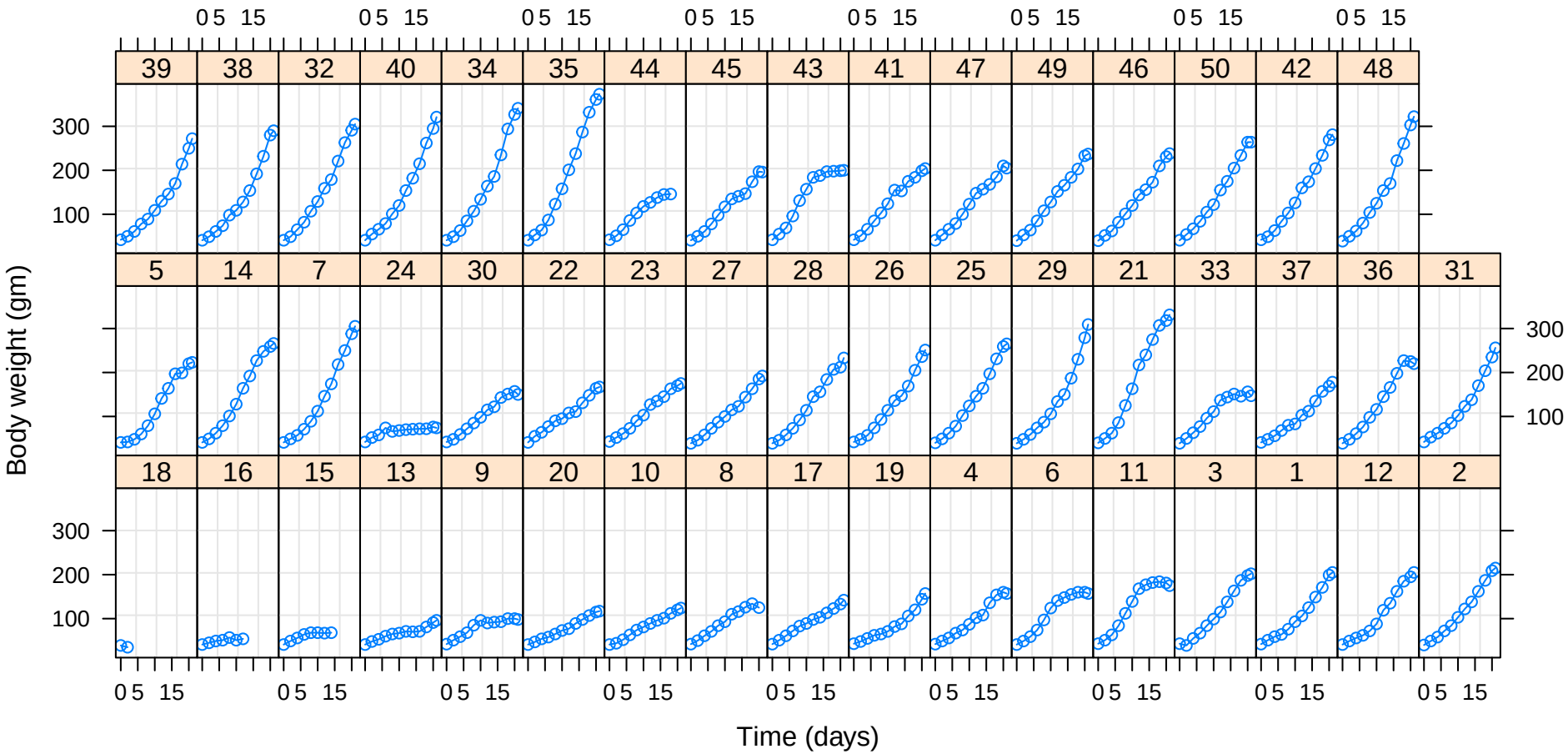
Barplot

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



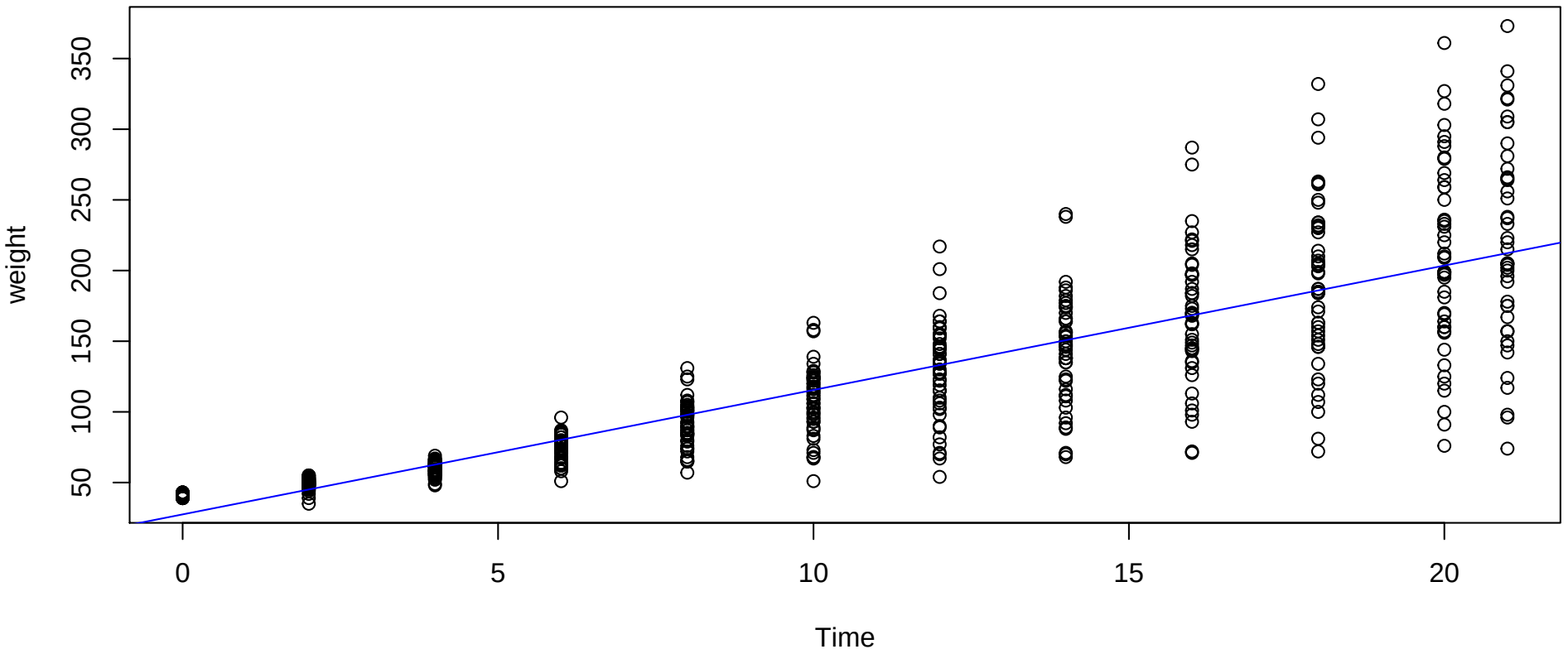
Scatterplot

```
plot(ChickWeight)
```



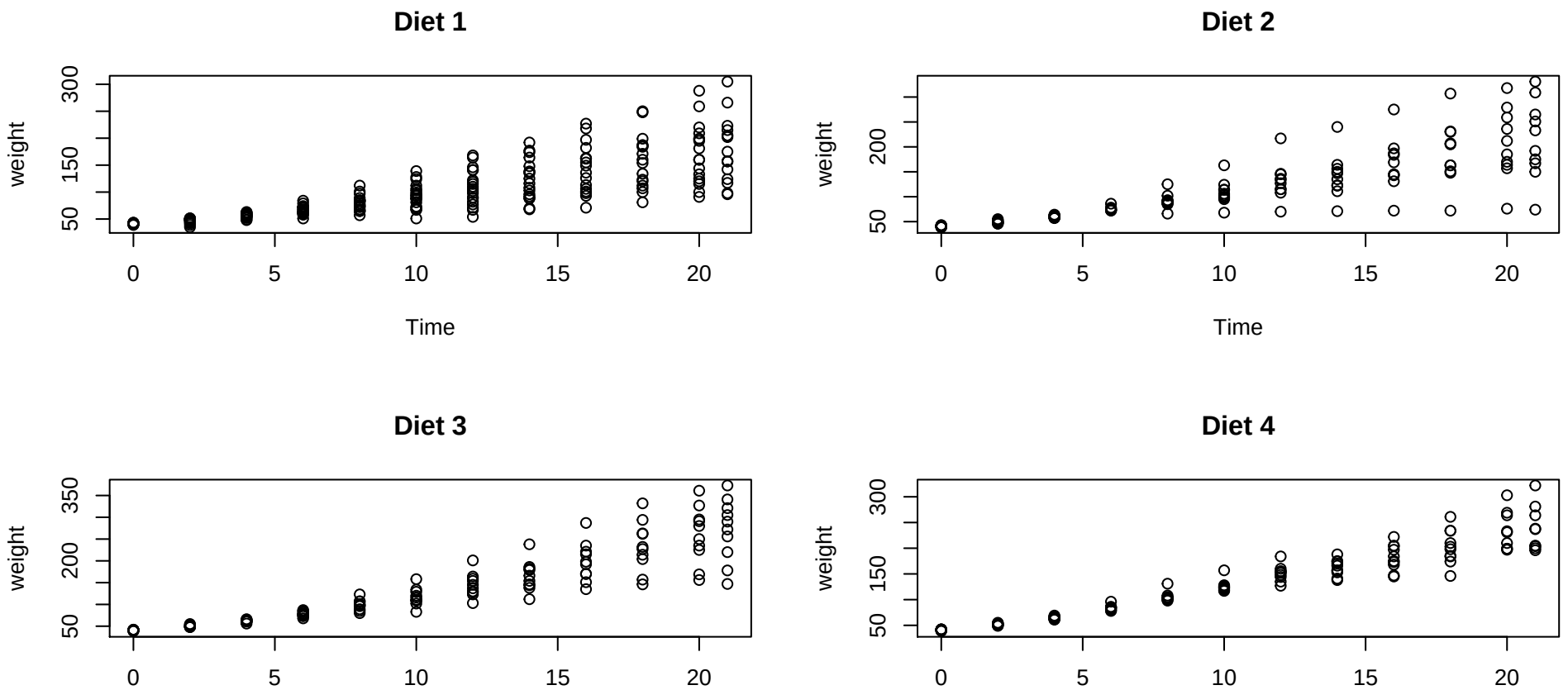
Scatterplot

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



Scatterplot

```
par(mfrow=c(2,2))
with(subset(ChickWeight, Diet==1), plot(Time, weight, main="Diet 1"))
with(subset(ChickWeight, Diet==2), plot(Time, weight, main="Diet 2"))
with(subset(ChickWeight, Diet==3), plot(Time, weight, main="Diet 3"))
with(subset(ChickWeight, Diet==4), plot(Time, weight, main="Diet 4"))
```



Base Graphics parametrai

- pch - the plotting symbol
- lty - the line type
- lwd - thje line width
- col - color
- xlab - charackter string x-axis label
- ylab - charackter string y-axis label
- main - charackter string main label

par

- 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 :

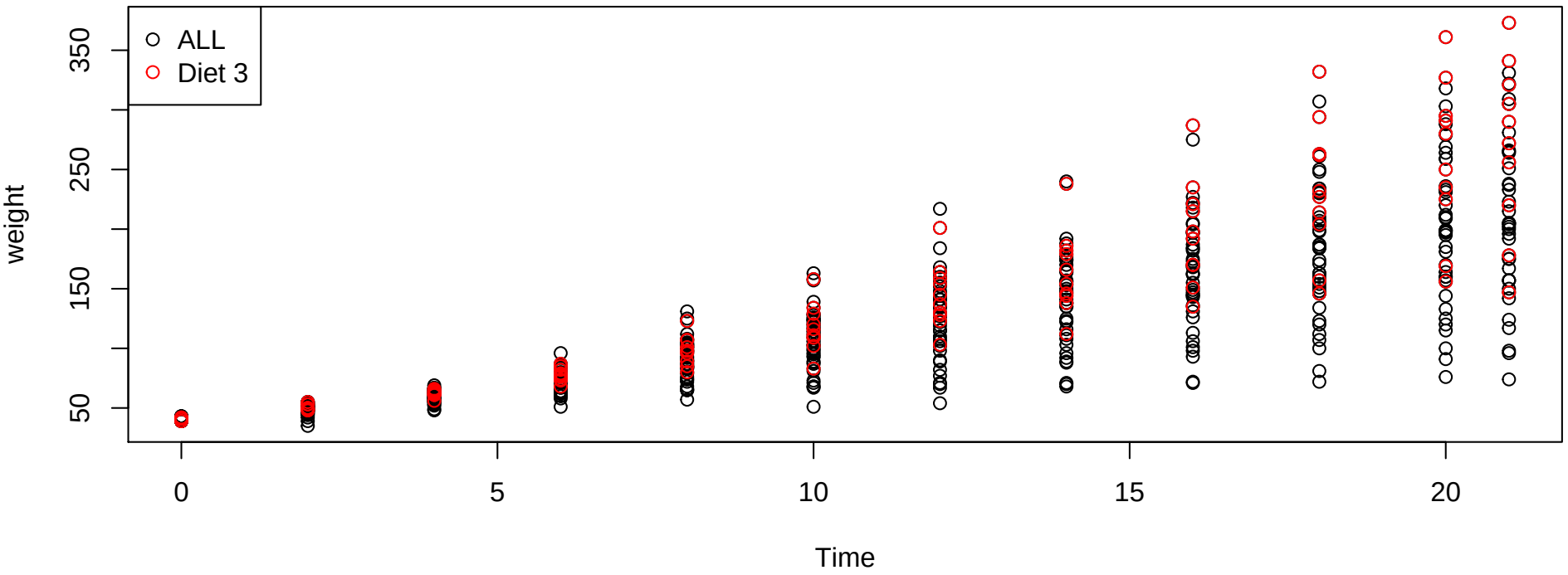
```
par("bg")
## [1] "transparent"
par("mar")
## [1] 5.1 4.1 4.1 2.1
```

Base plotting funkcijos

- plot sukuria pagrinį grafika
- lines - prideda linijas (vektorius)
- points - prideda taškus
- text - prideda tekstą
- title - prideda anotacijas
- axis - prideda ašių *ticks* ir *labels*

Scatterplot

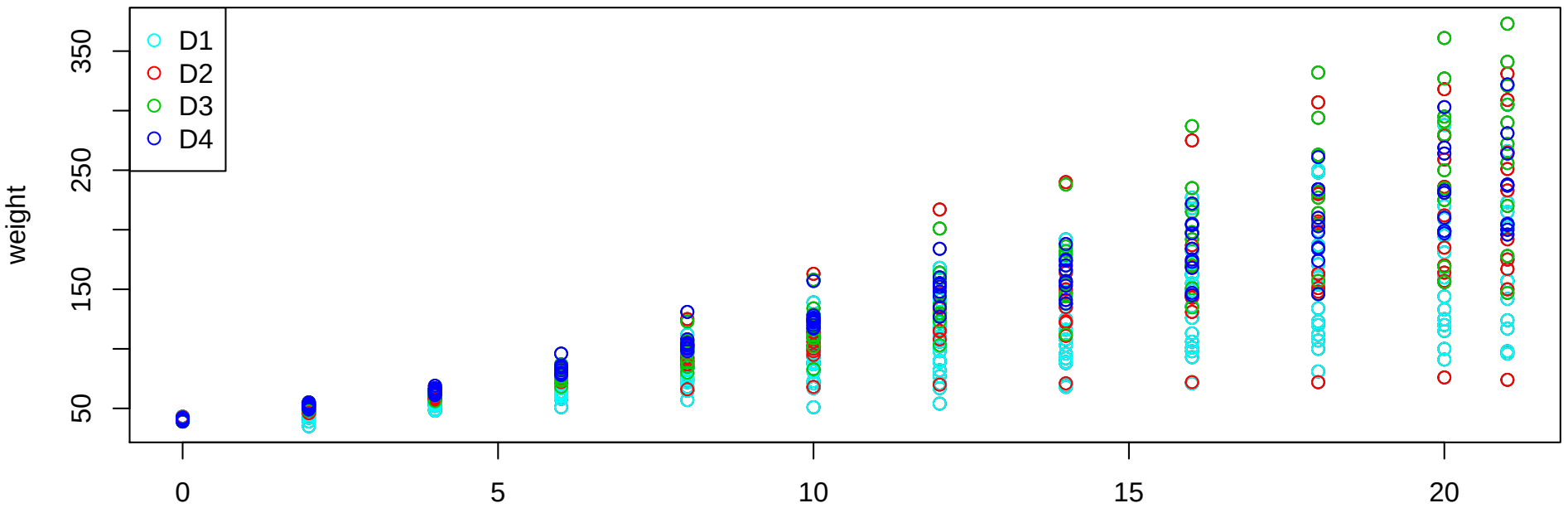
```
with(ChickWeight, plot(Time, weight))
with(subset(ChickWeight, Diet==3), points(Time, weight, col="red"))
legend("topleft", pch=1, col=c("black", "red"), legend=c("ALL", "Diet 3"))
```



Scatterplot

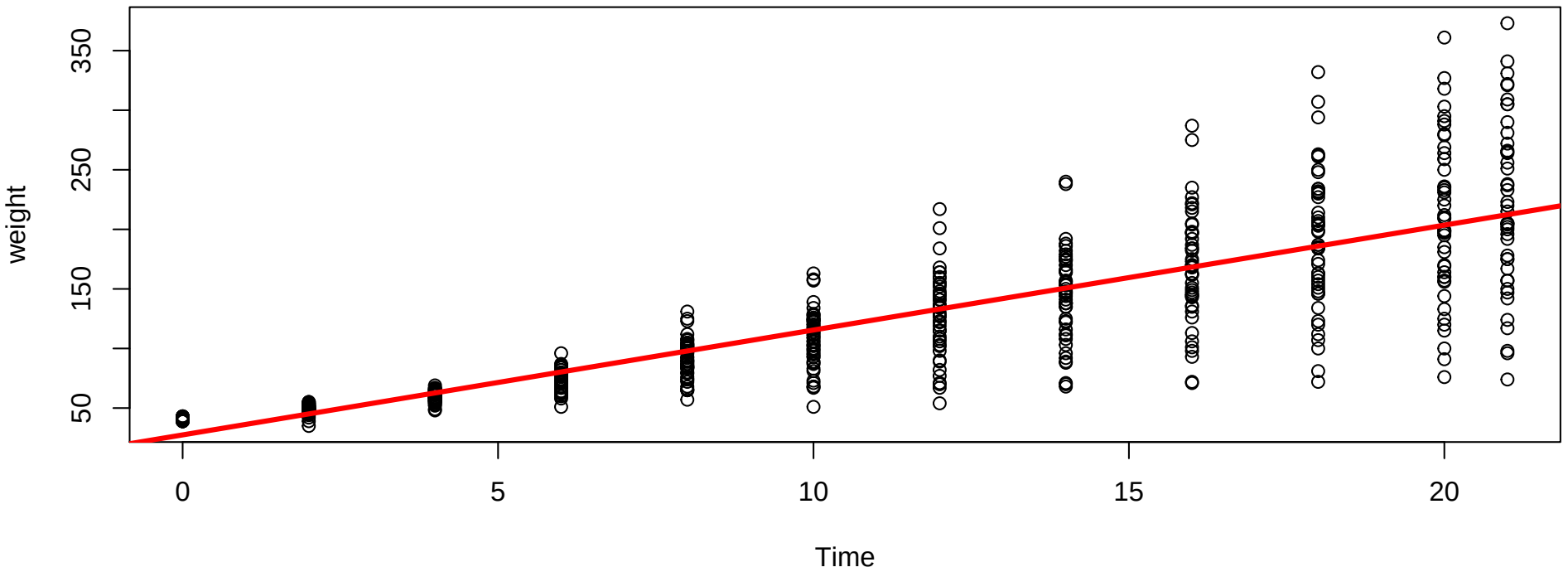
- `type="n"` nepiešia nieko, tik sukuria bazę

```
with(ChickWeight, plot(Time, weight), type="n")
with(subset(ChickWeight, Diet==1), points(Time, weight, main="Diet 1", col=
with(subset(ChickWeight, Diet==2), points(Time, weight, main="Diet 2", col=
with(subset(ChickWeight, Diet==3), points(Time, weight, main="Diet 3", col=
with(subset(ChickWeight, Diet==4), points(Time, weight, main="Diet 4", col=
legend("topleft", pch=1, col=c(5,2,3,4), legend=c("D1", "D2", "D3", "D4"))
```



Tiesinė regresija

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

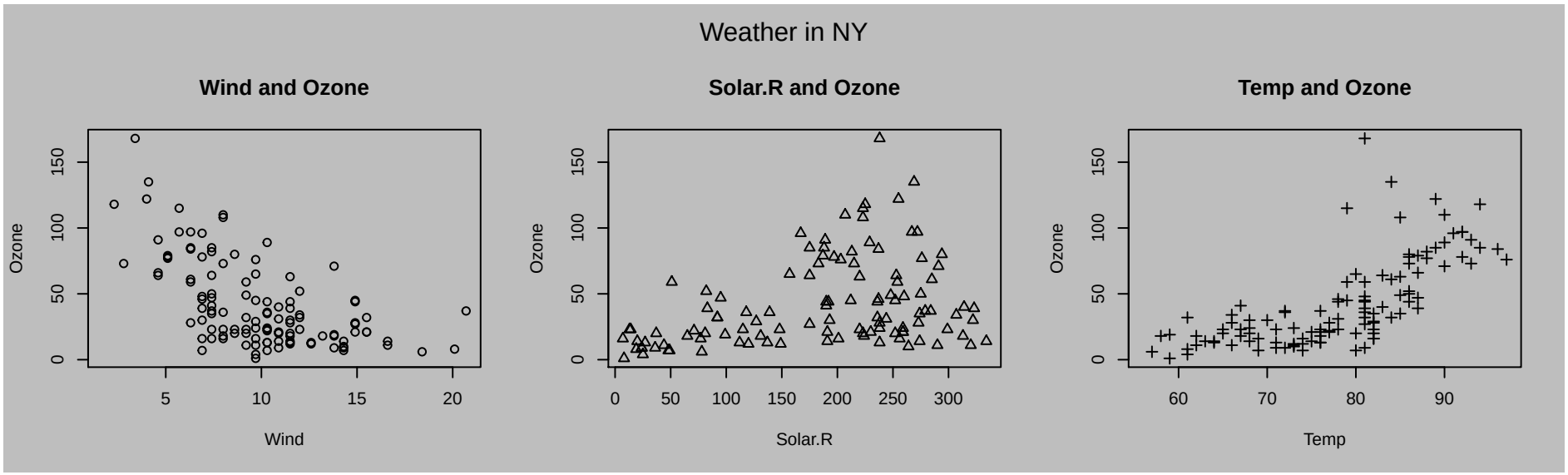


mar ir oma

```
par("mar")
## [1] 5.1 4.1 4.1 2.1
par("oma")
## [1] 0 0 0 0
```

outer ir mtext

```
par(mfrow=c(1,3),par=c(1,1,1,1),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)
})
```



example(points)

- išbandykite: example(points)

Graphics devices

- ? Devices
- Ekrane (windows(). quartz(), x11())
- Vektoriniai formatai
  - pdf
  - svg
  - ...
- Bitmap formatai
  - png
  - jpeg
  - tiff
  - bmp
- dev.copy()

Graphics devices

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

Graphics devices

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

ggplot2

ggplot2

- gg - Grammer of Graphics (Leland Wilkinson)
- parašyta Hadley Wickham (taip kur ir dplyr...)
- `install.packages(ggplot2)`
- cheatsheet ggplot2

ggplot2

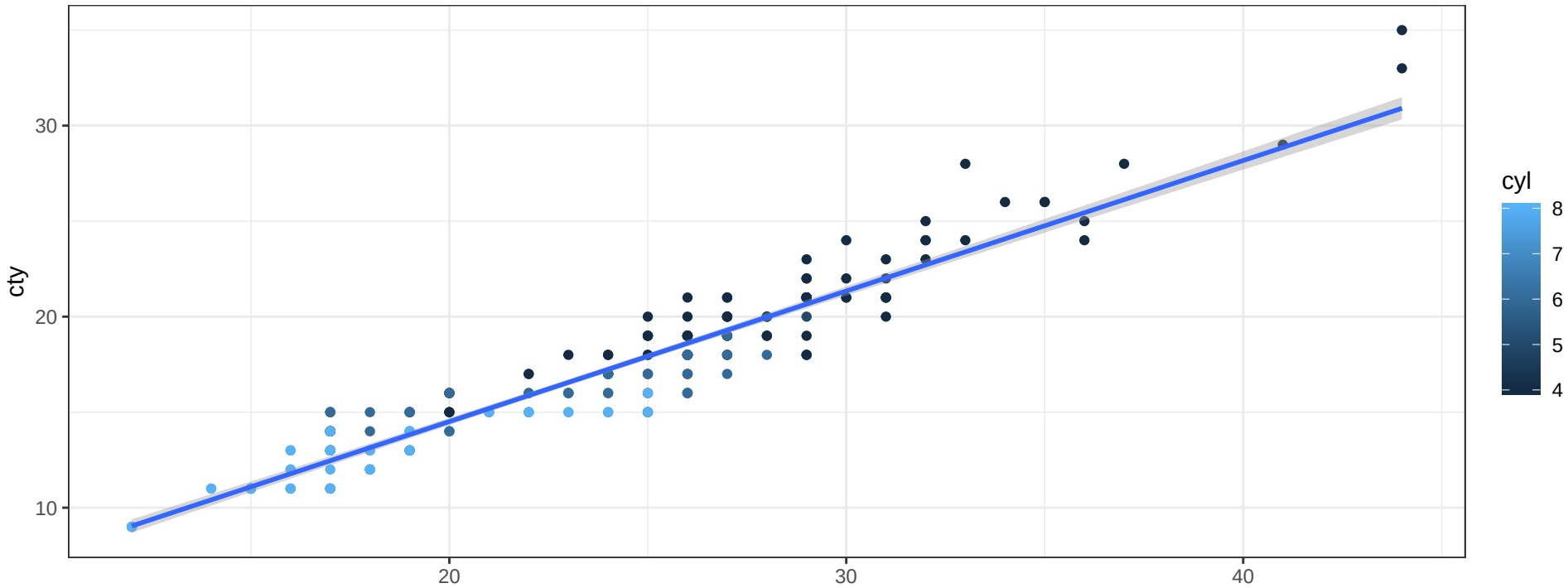
- gg - Grammer of Graphics (Leland Wilkinson)
- parašyta Hadley Wickham (taip kur ir dplyr...)
- duomenys turi būti `dataframe` objekte, geriausia `long` formatu
- `install.packages(ggplot2)`
- cheatsheet ggplot2

ggplot2

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

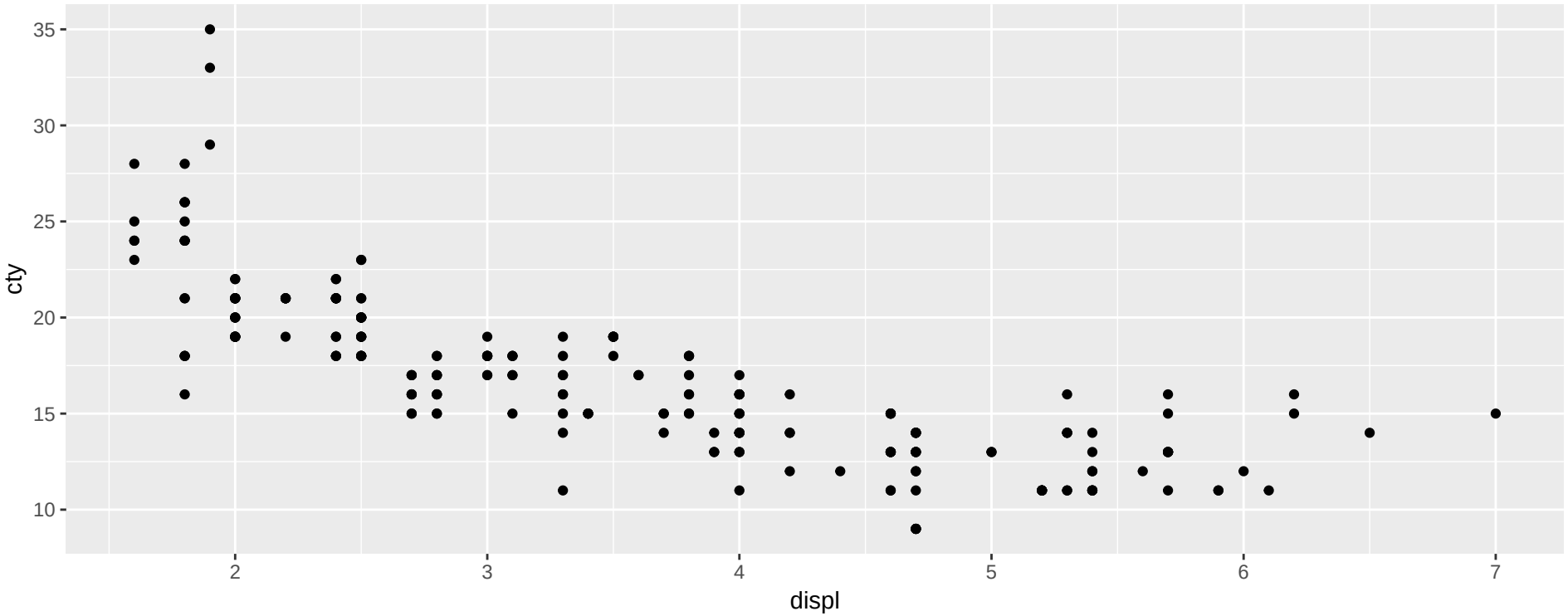
ggplot2

```
# library(ggplot2)
ggplot(mpg, aes(hwy, cty)) +
    geom_point(aes(color = cyl)) +
    geom_smooth(method ="lm") +
    coord_cartesian() +
    scale_color_gradient() +
    theme_bw()
```



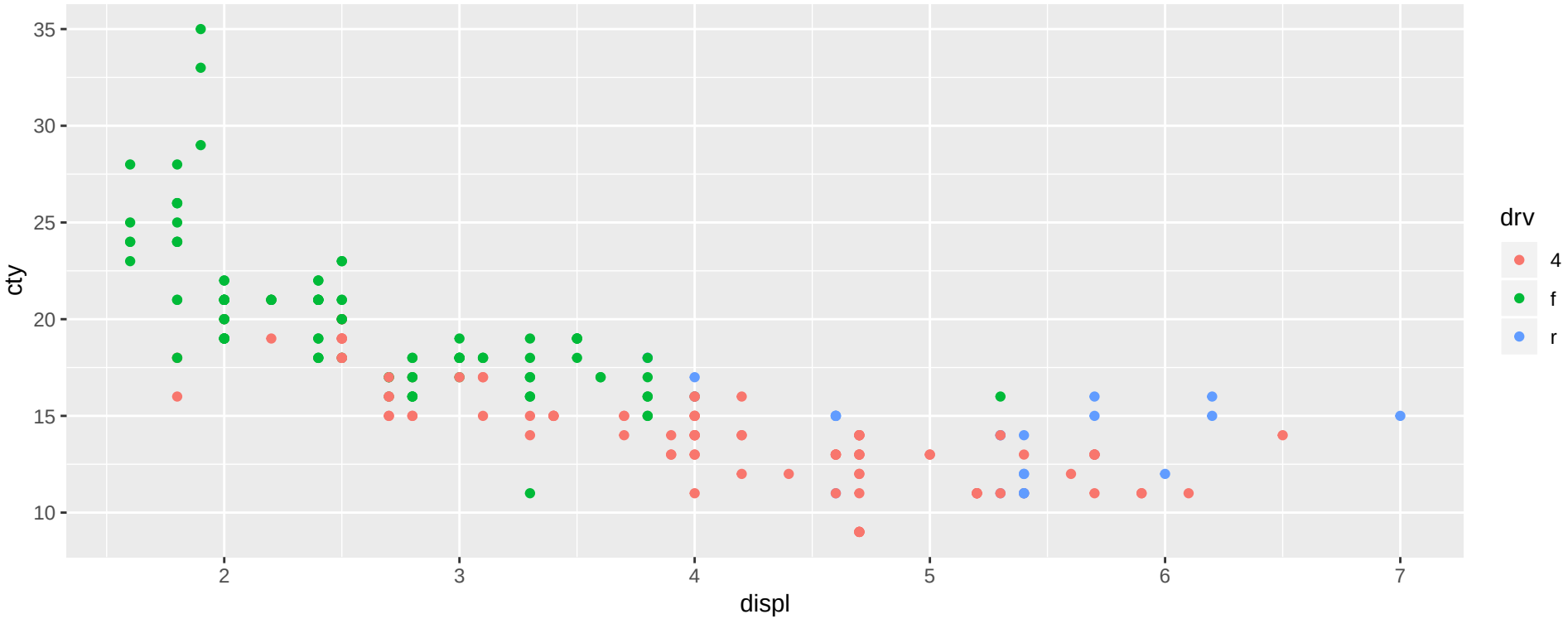
ggplot2

```
ggplot(mpg)+
    geom_point(aes(displ, cty))
```



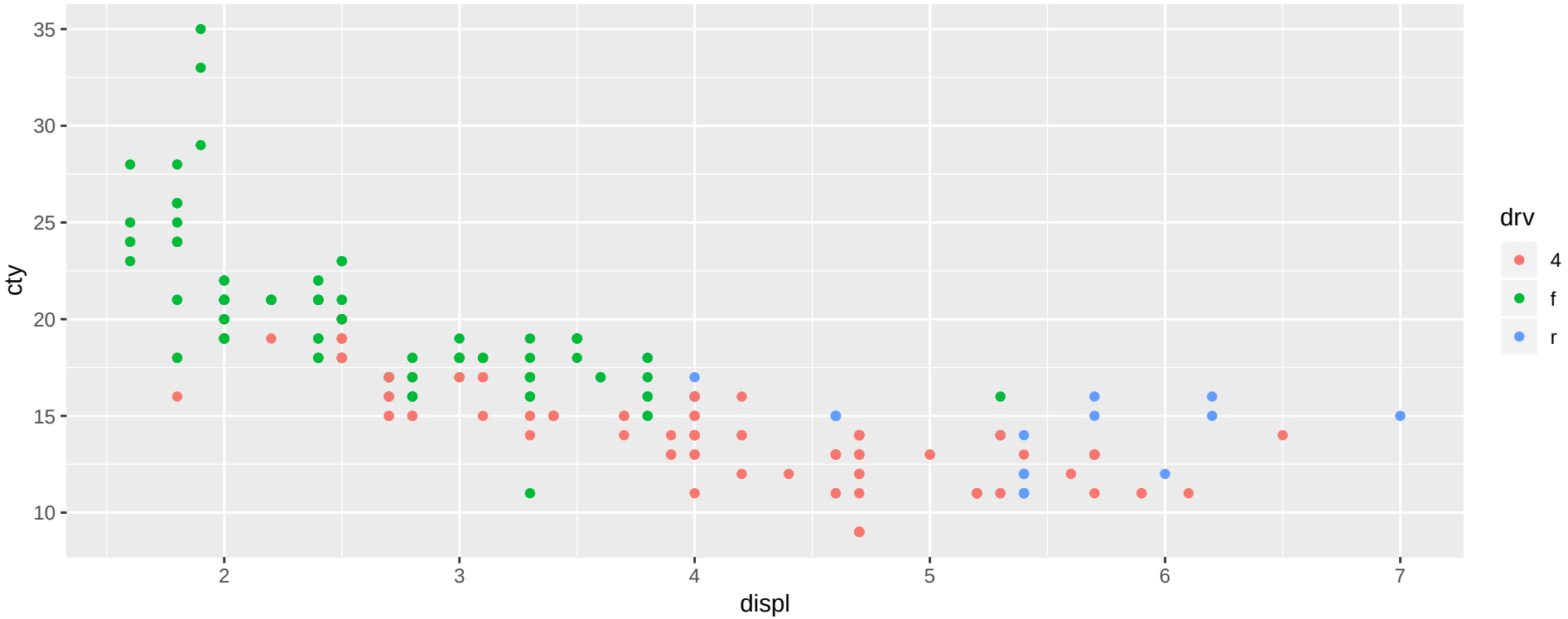
ggplot2

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



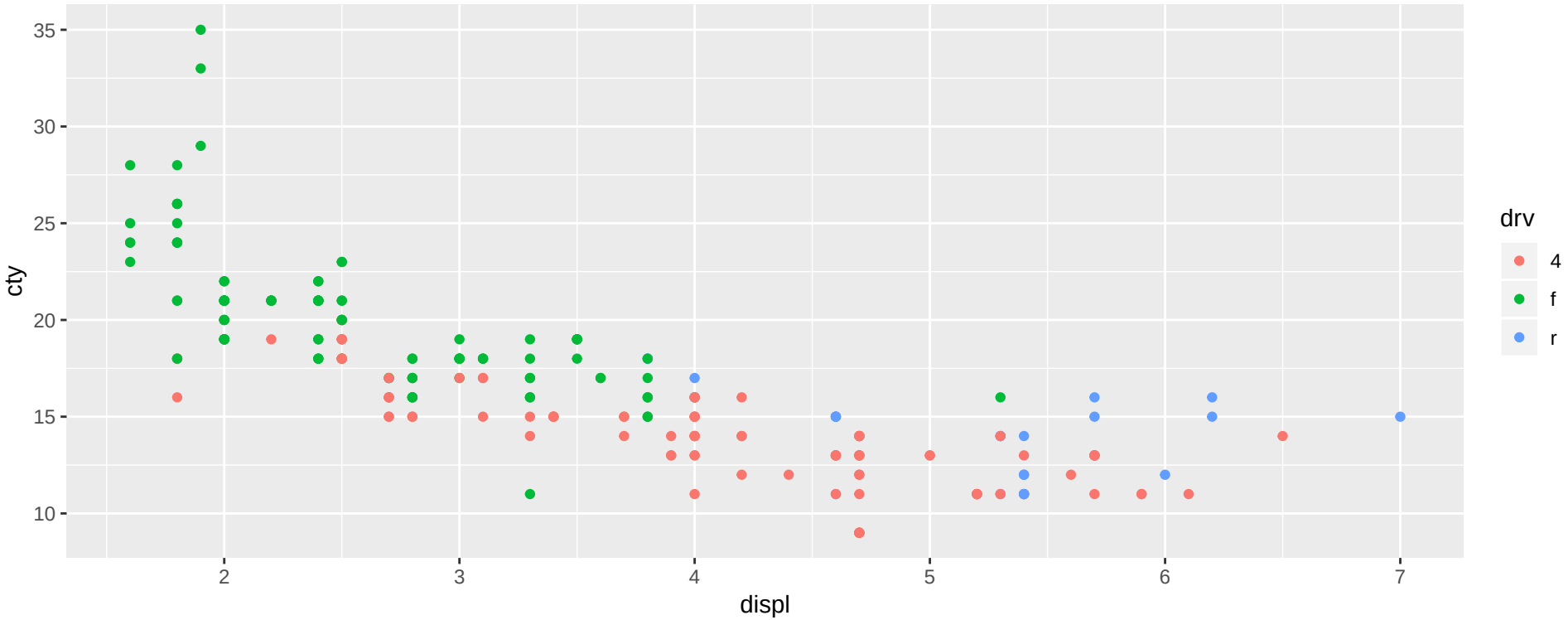
ggplot2

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



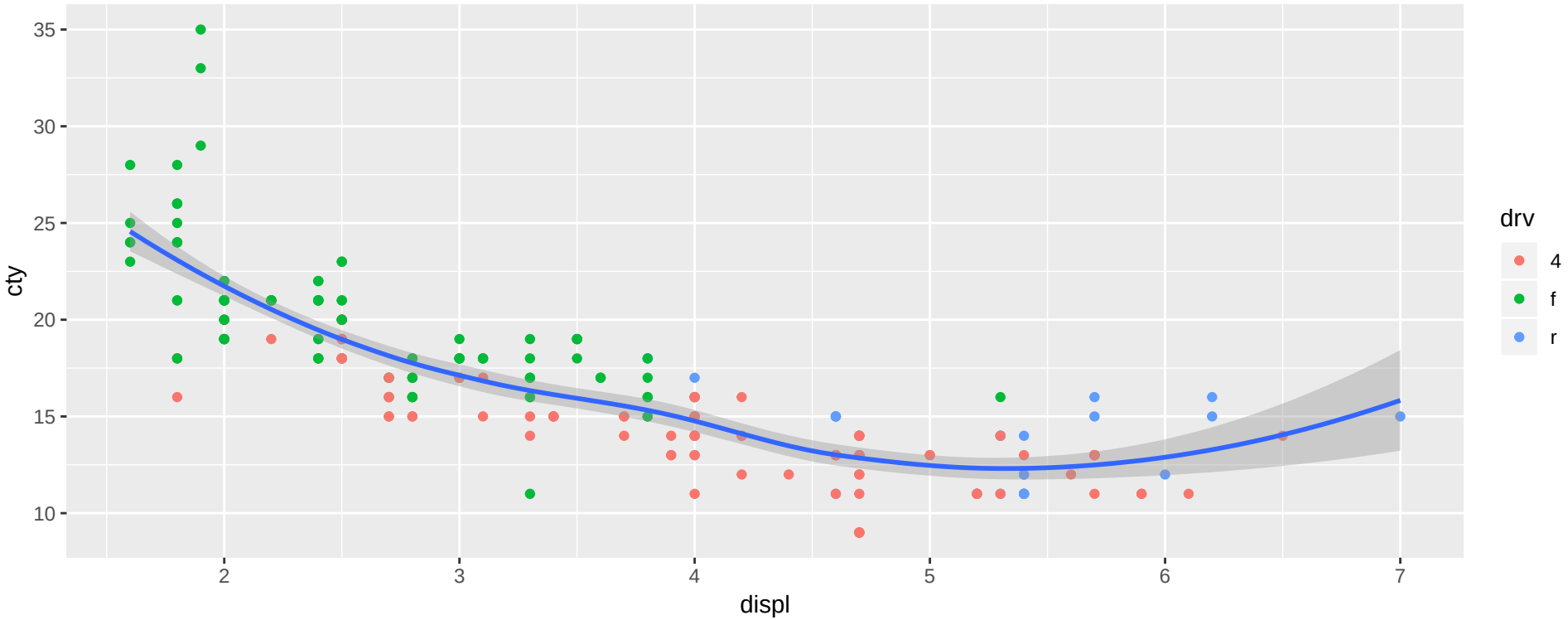
ggplot2

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



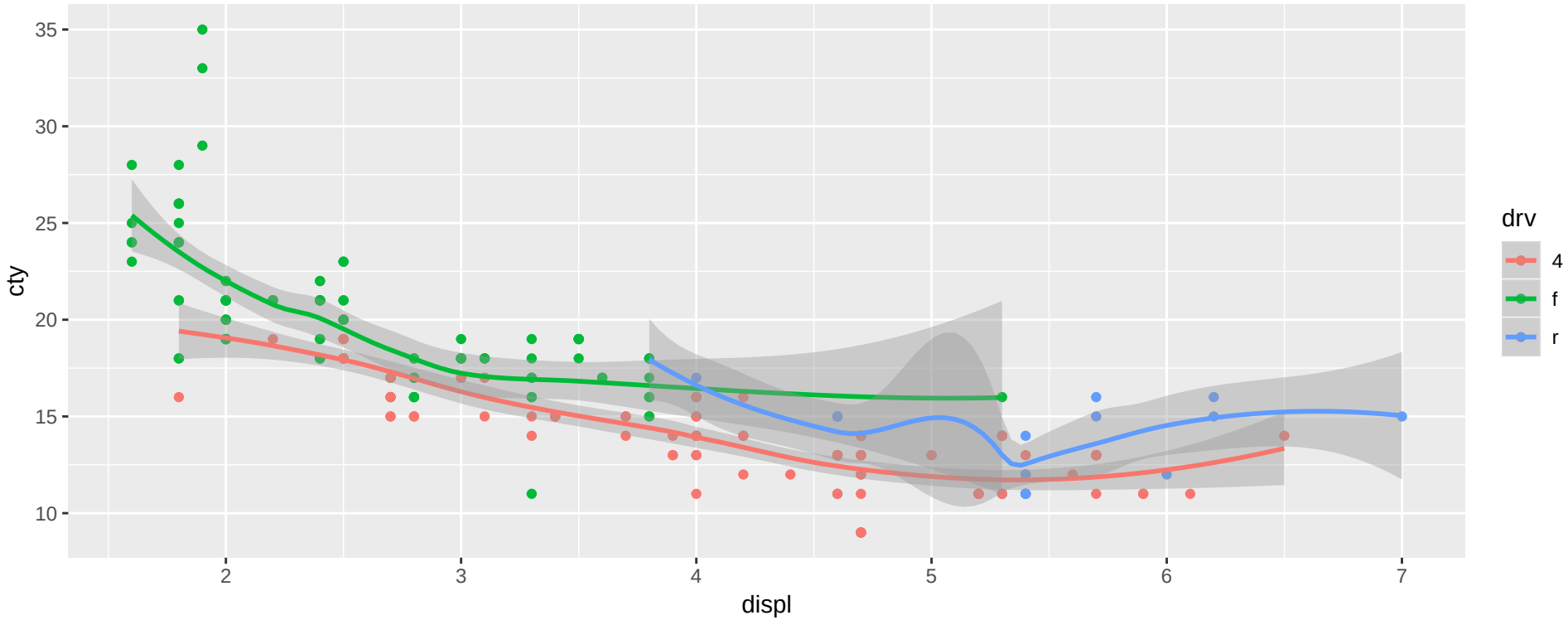
ggplot2

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



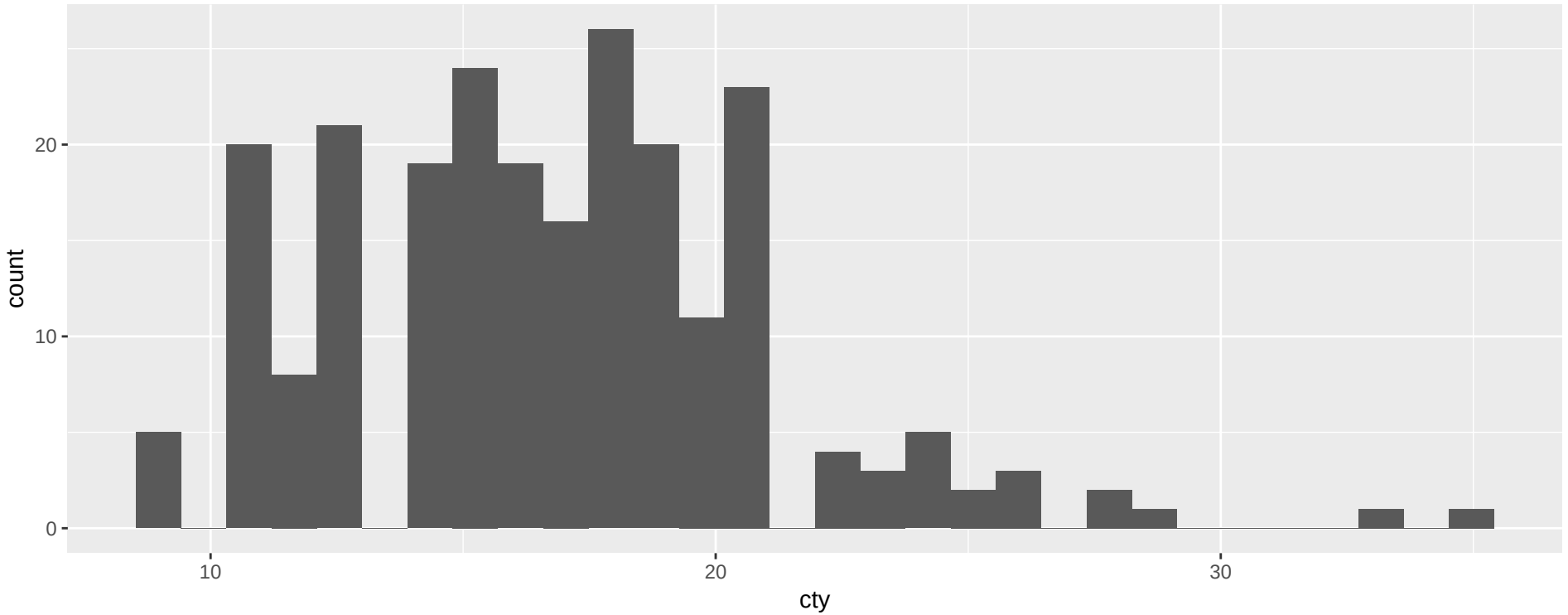
ggplot2

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



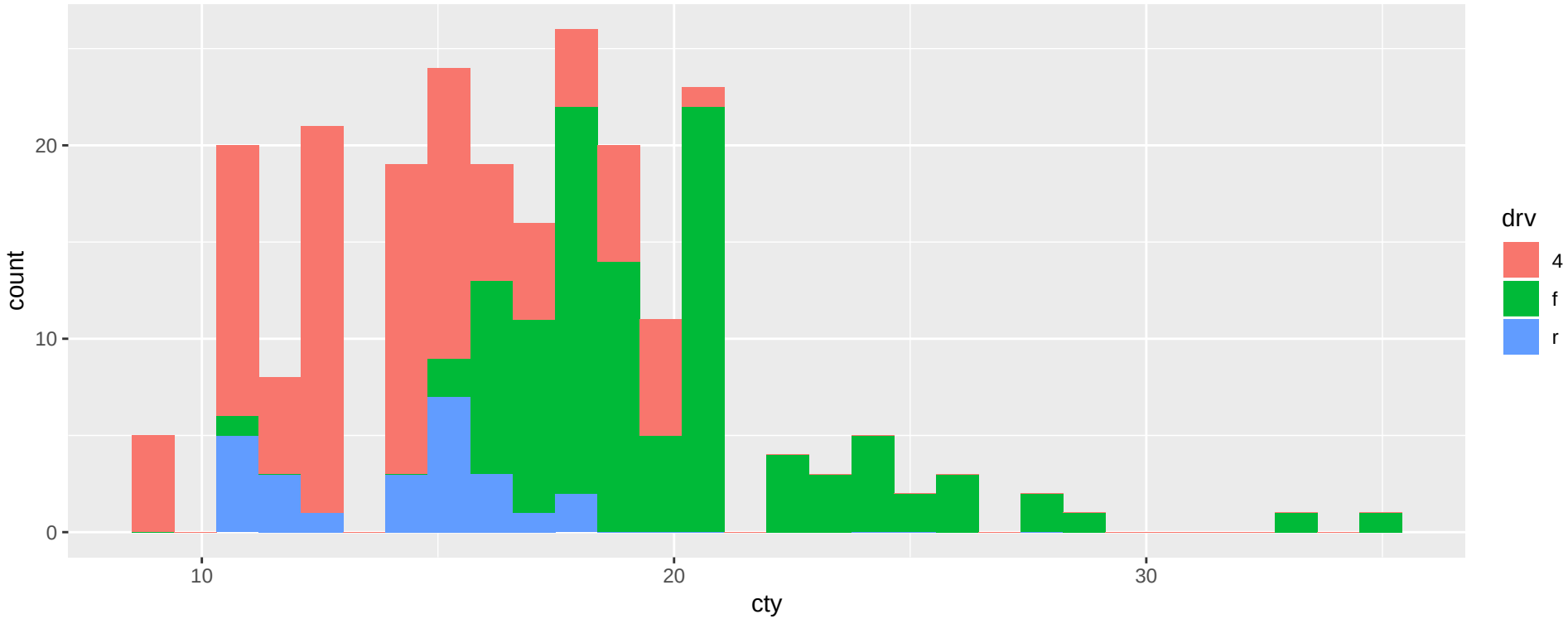
ggplot2

```
ggplot(mpg, aes(cty))+
    geom_histogram()
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



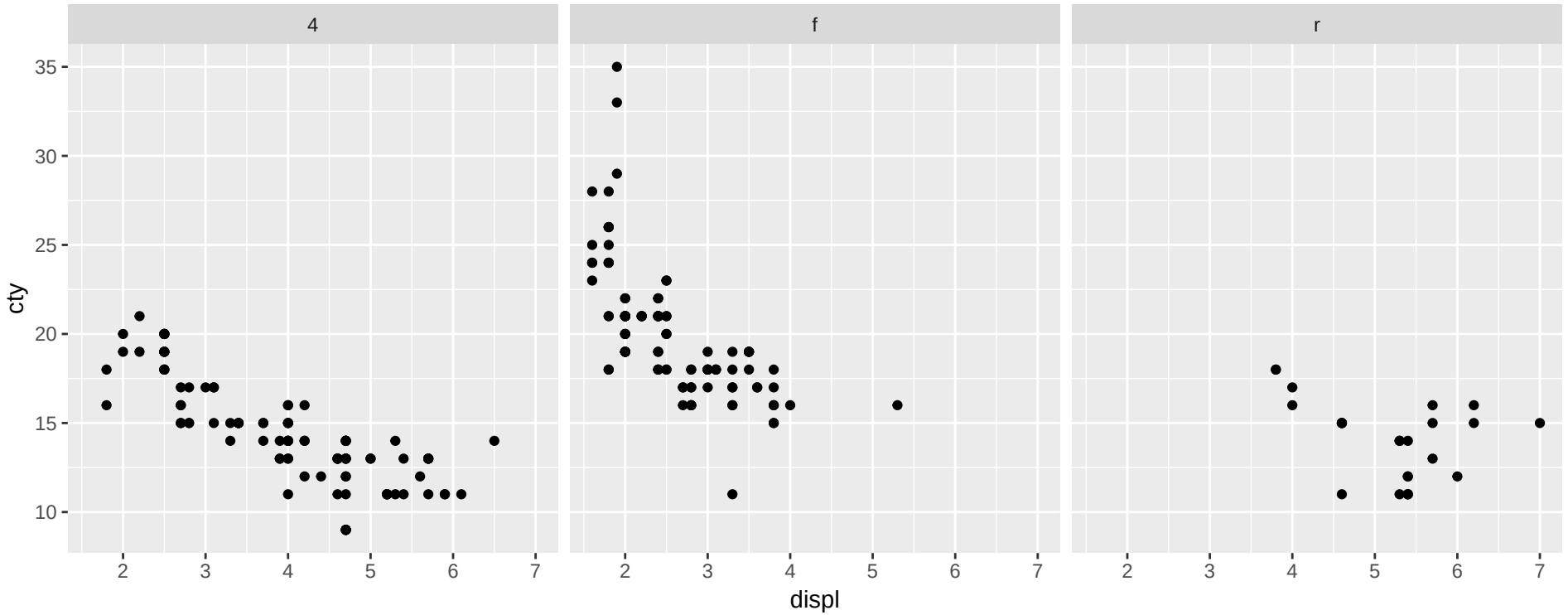
ggplot2

```
ggplot(mpg, aes(cty))+
    geom_histogram(aes(fill=drv))
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



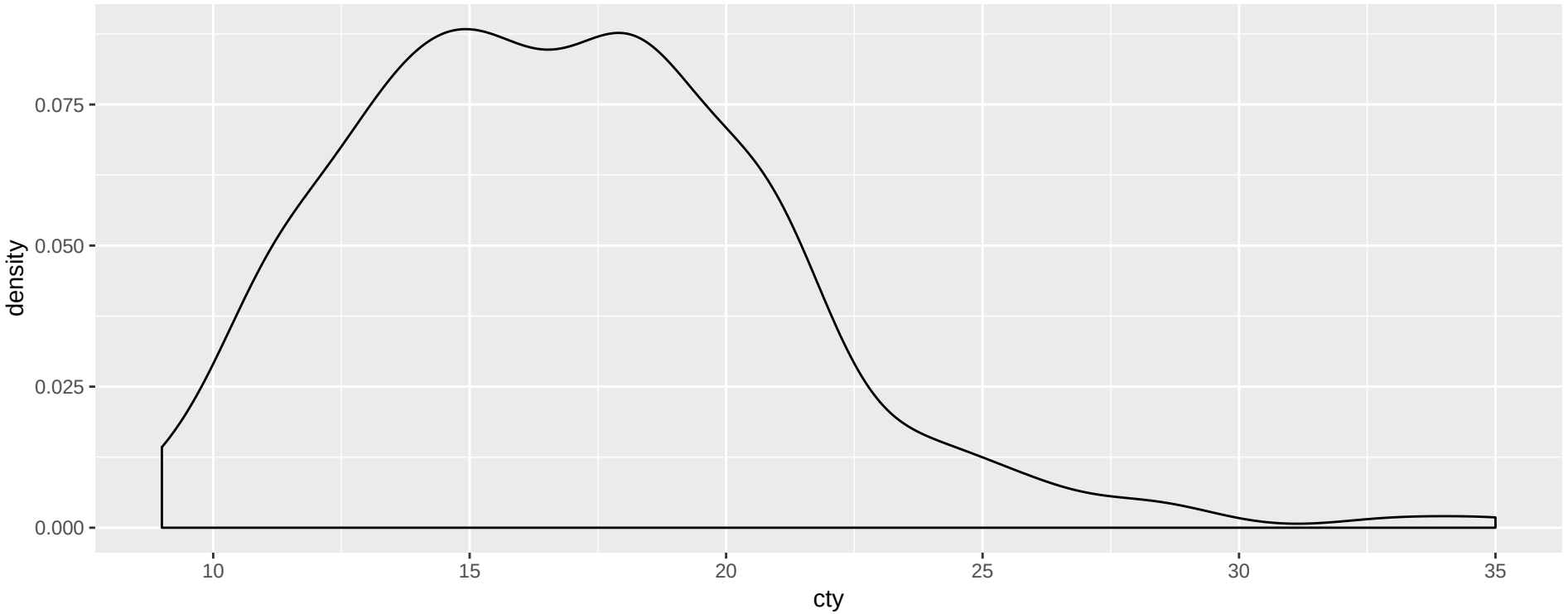
ggplot2

```
ggplot(mpg, aes(displ, cty))+
    geom_point()+
    facet_grid(~ drv)
```



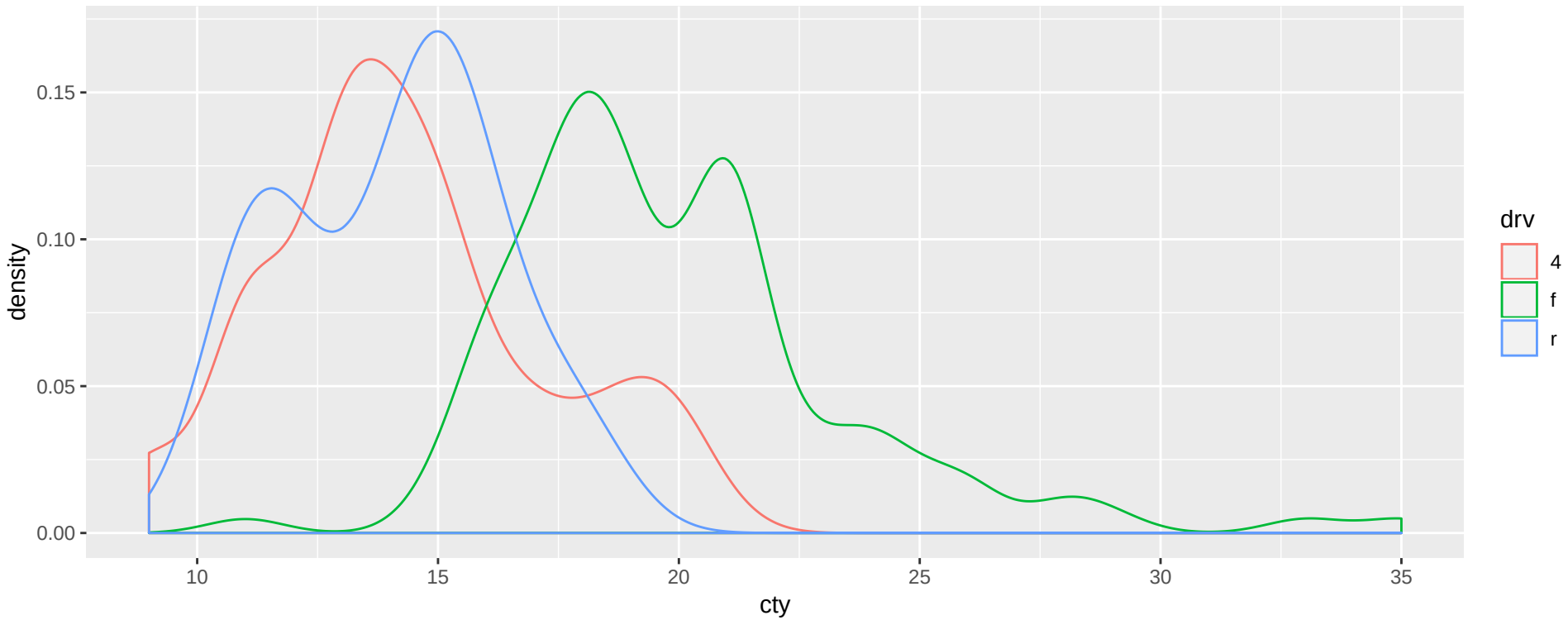
ggplot2

```
ggplot(mpg, aes(cty))+
    geom_density()
```



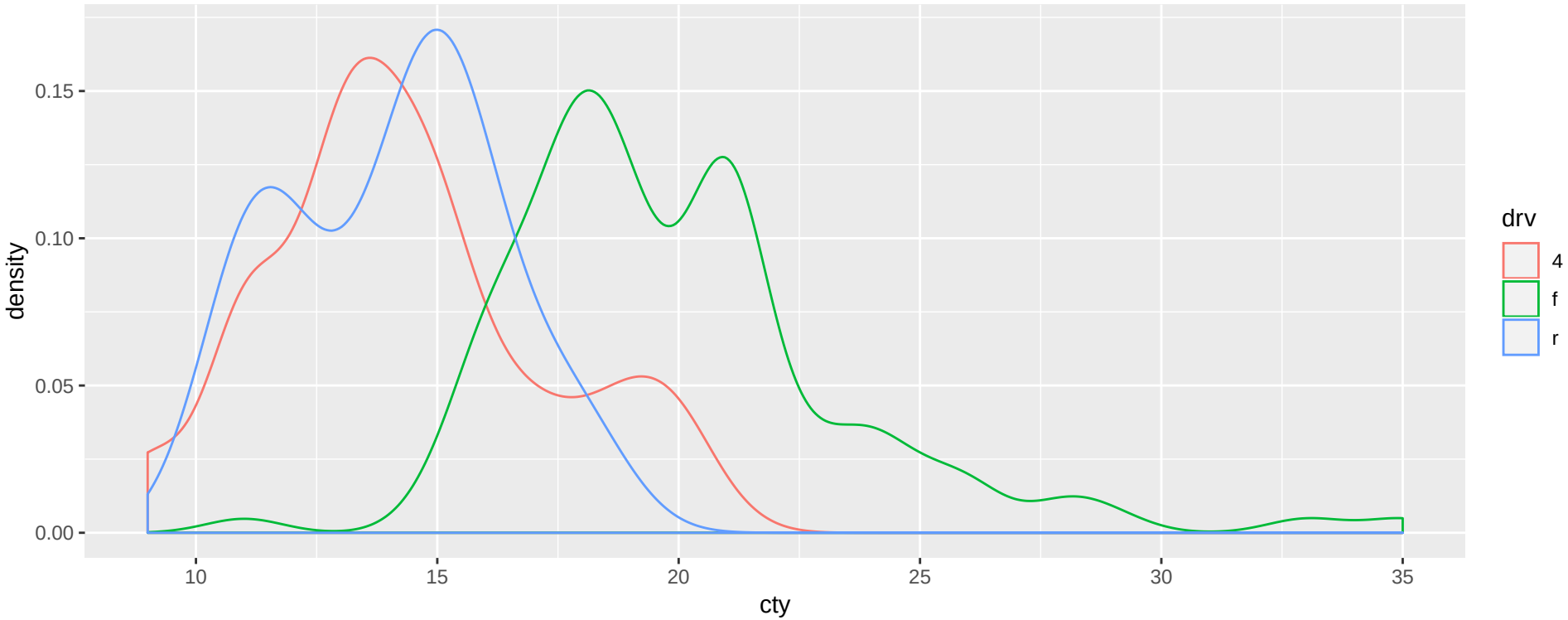
ggplot2

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



ggplot2

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

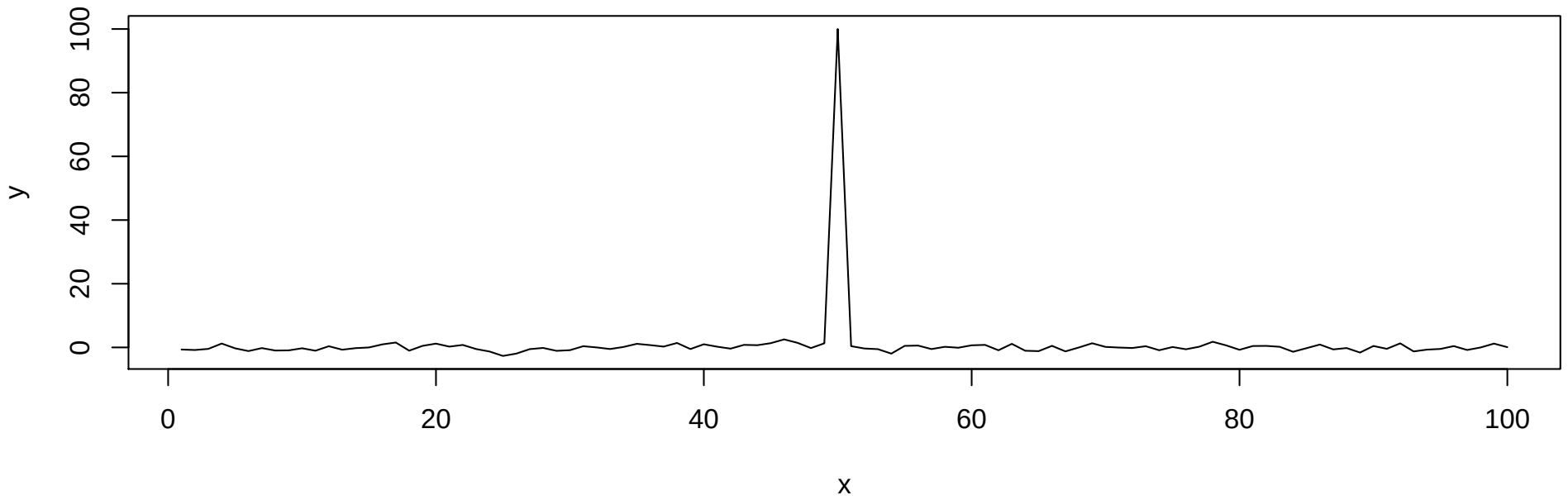


Outlayeriai

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

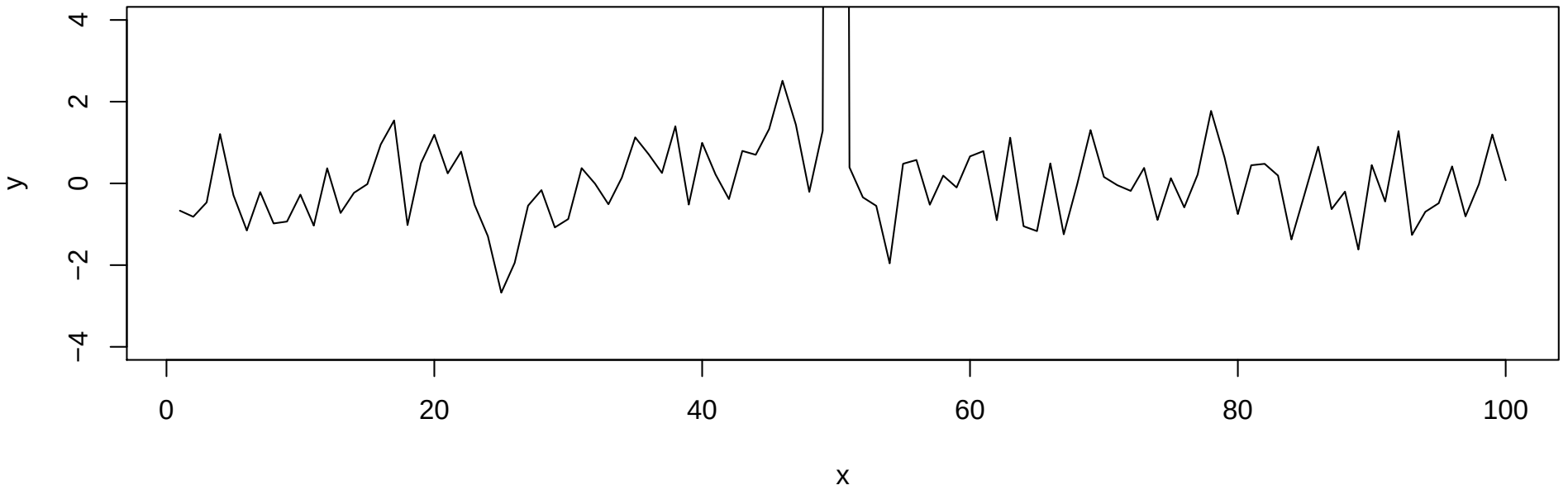
Outlayeriai

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



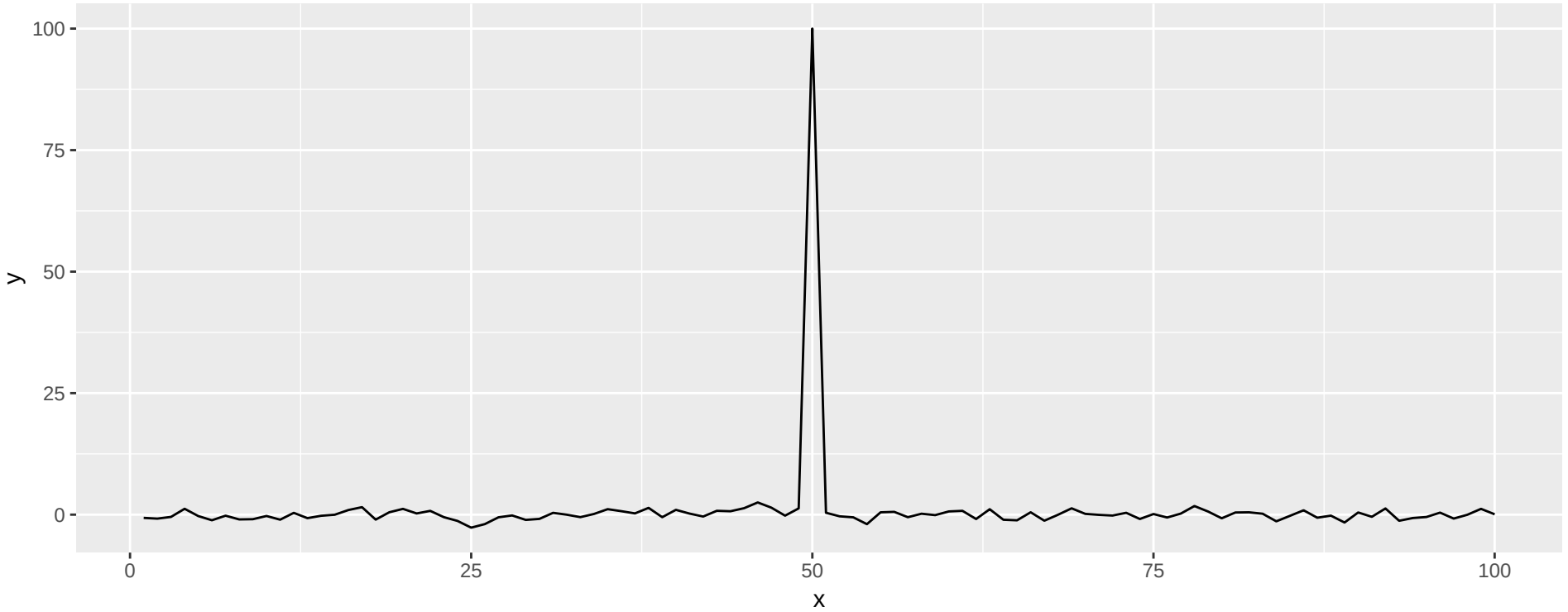
Outlayeriai

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



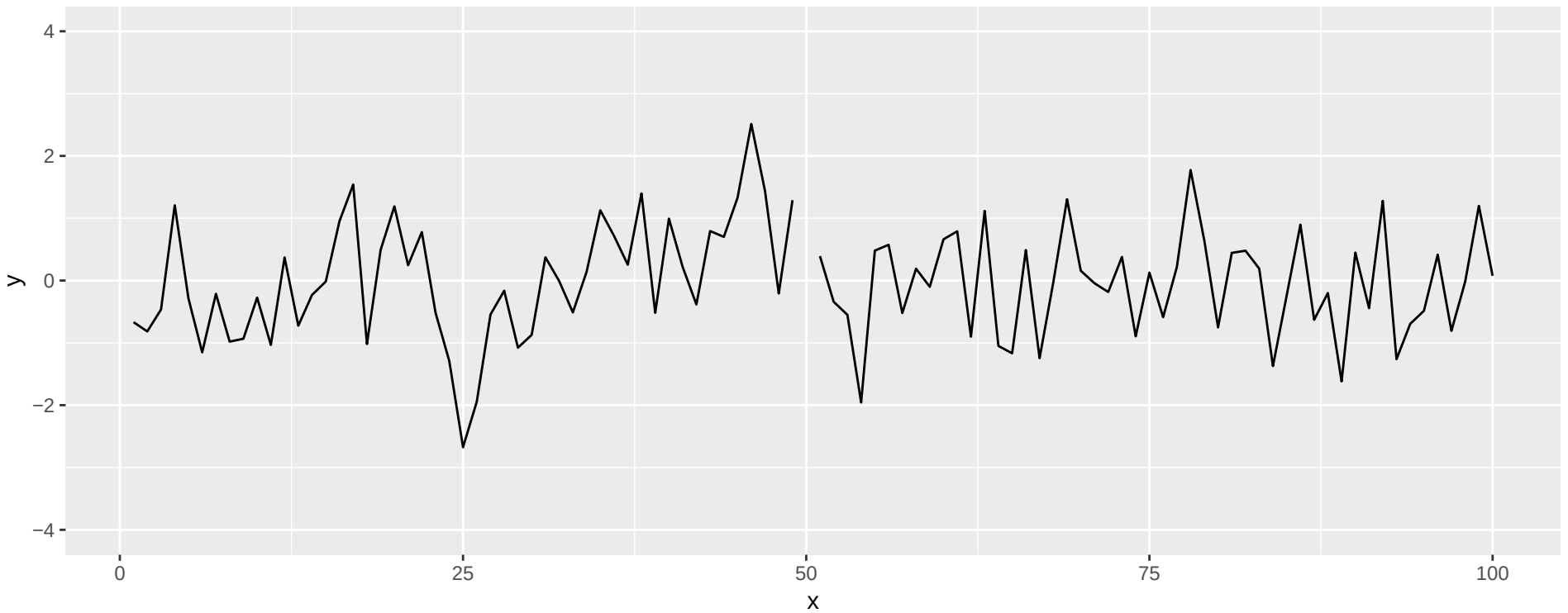
Outlayeriai

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



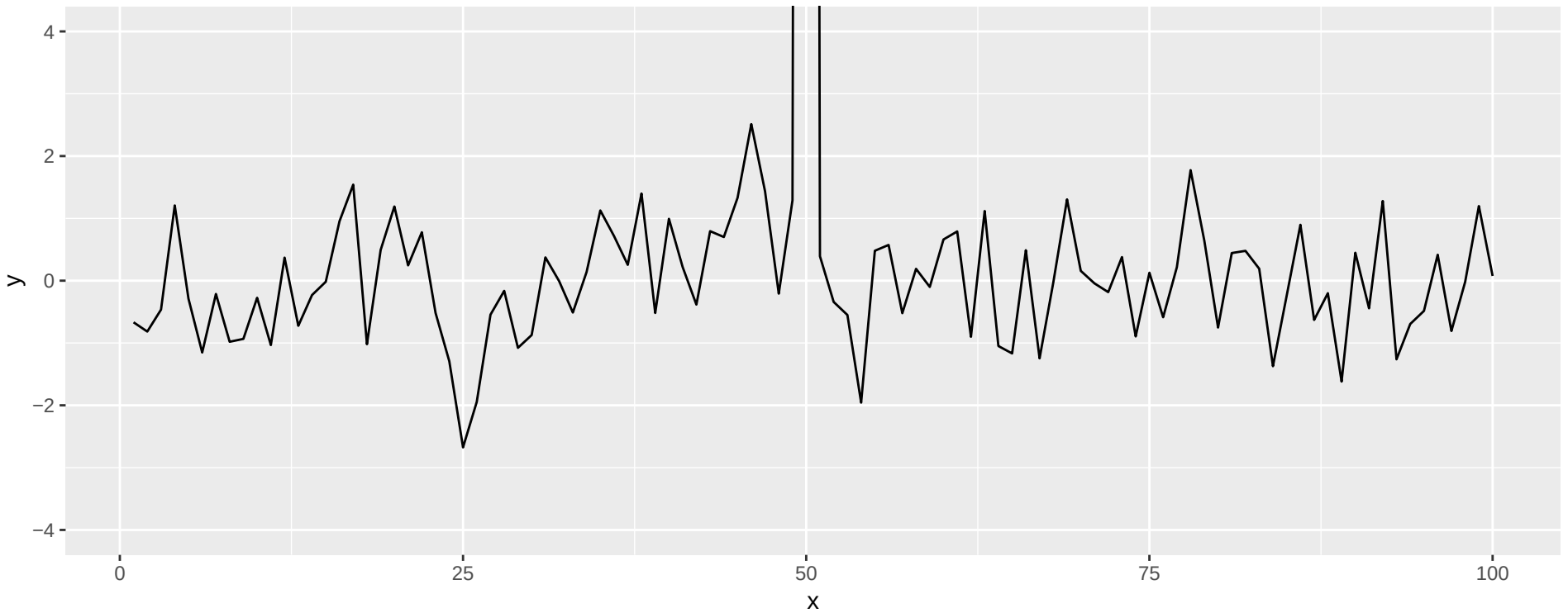
Outlayeriai

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



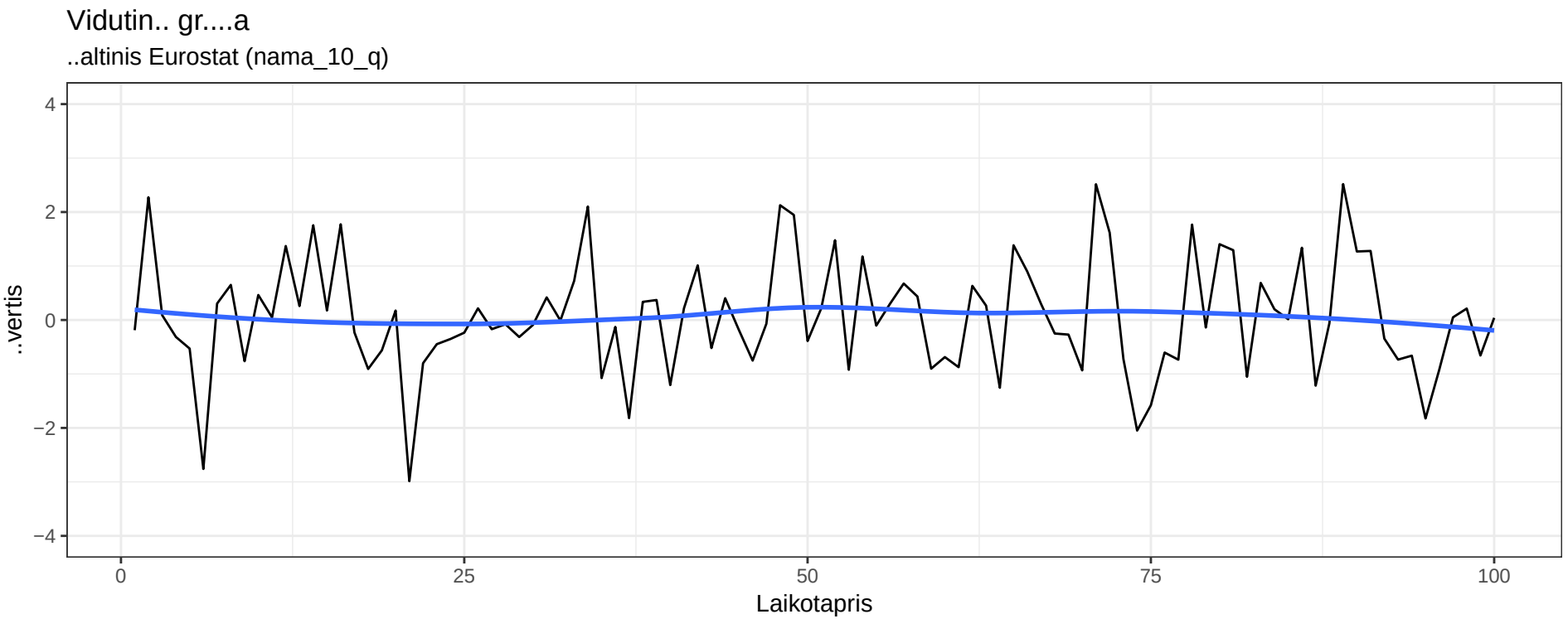
Outlayeriai

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



labs()

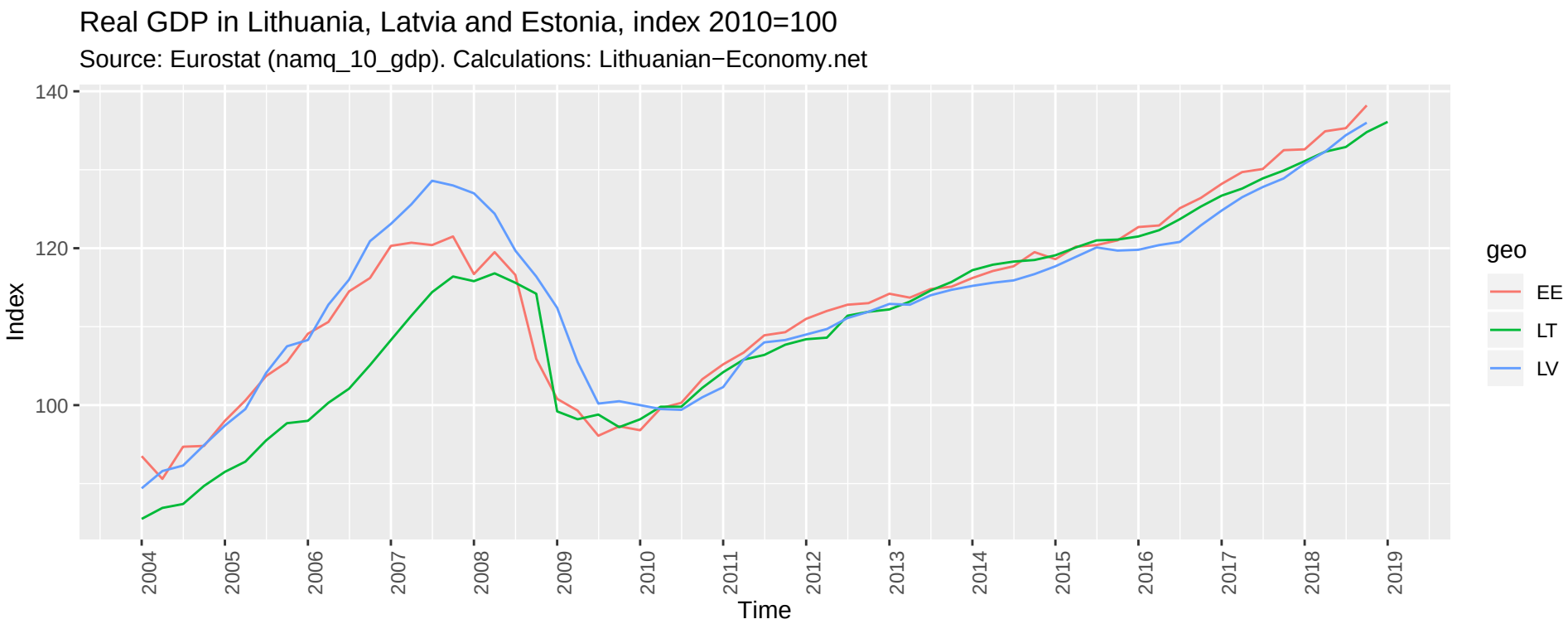
```
df<- data.frame(x=1:100, y=rnorm(100))
ggplot(df, aes(x=x,y=y))+theme_bw()+
    geom_line()+ geom_smooth(se=FALSE, method = "loess")+
    coord_cartesian(ylim=c(-4,4))+
    labs(x="Laikotapris", y="Įvertis", title= "Vidutinė grąža",
         subtitle = "Šaltinis Eurostat (nama_10_q)")
```



1 Hands on...

- parašykite skirptą, kuris, importuoja duomenis iš Eurostat
- apdoroja duomenis su dplyr
- nubraižo grafiką geom_line()
- Duomenys:
  - 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

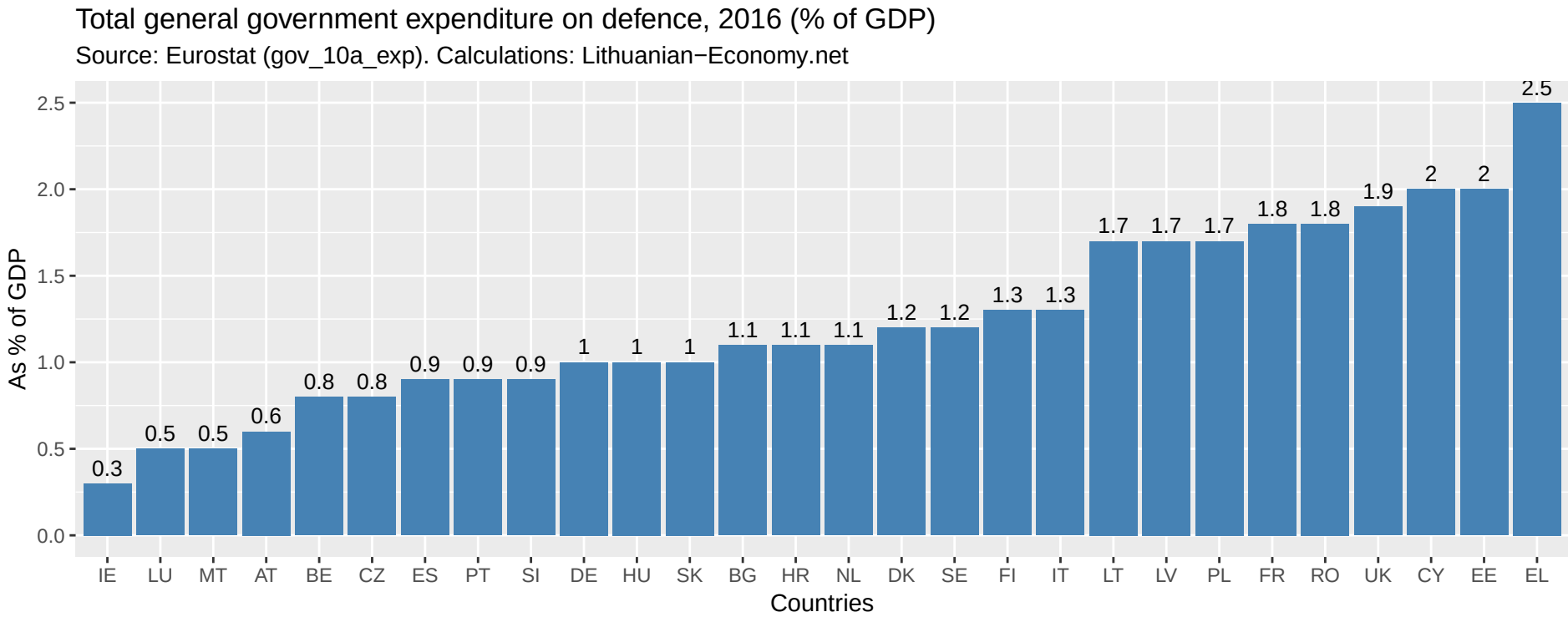
1 Hands on...


Real GDP in Lithuania, Latvia and Estonia, index 2010=100
Source: Eurostat (namq_10_gdp). Calculations: Lithuanian−Economy.net
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:
  - gov_10a_exp
  - visos ES šalys! (28)
  - Total expenditure
  - General government
  - 2017m
  - procentais nuo BVP

2 Hands on...


Total general government expenditure on defence, 2016 (% of GDP)
Source: Eurostat (gov_10a_exp). Calculations: Lithuanian−Economy.net
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

KNITR

3 Hands on

- sukurkite 2 funkcijas bruto_neto
- 2019 ir 2020 metais
- apskaičiuokite jose ITR (visi mokesčiai / darbo vietos kaina)
- nudownloadinkite Sodros draudžiamų pajamų duomenis
- nubraižykite ITR_2019 ir ITR_2020

3 Hands on

Pagalba: * `min()` ir `max()` nepriima vektorių, juos reiktų pakeisti, žr ?min * funkcijos pabaigoje sukurkite list objektą, kuriam priskirkite norimus rodiklius * GPM įstyatymas * https://e-seimas.lrs.lt/portal/legalActEditions/lt/TAD/TAIS.171369 * 2019 prog VDU * http://finmin.lrv.lt/lt/aktualus-valstybes-finansu-duomenys/ekonomines-raidos-scenariju

3 Hands on

```
bruto_neto <- function(x) {
        GPM_1 <- 0.20
        GPM_2 <- 0.27
        PSD <- 0.0698
        SODRA <- 0.1252
        MMA <- 555
        VDU <- 1283.2
        lubos <- 10*VDU
        NPD <- 300
        NPD_coef <- 0.15
        bruto <- x
        npd <- max(NPD - NPD_coef* max(0,(bruto - MMA)),0)
        mok_baz <- max(0,(bruto-npd))
        gpm <- ifelse(bruto<=lubos, mok_baz*GPM_1, lubos *GPM_1+(bruto-lubo
        sodra <- min(bruto*SODRA, lubos * SODRA)
        psd <- bruto*PSD
        neto <- bruto - gpm - sodra - psd
        neto
}
```