

R-programmering VT224

Föreläsning 7

Johan Alenlöv

Linköpings Universitet

Föreläsning 7

- Grafik med ggplot2
- Grundläggande statistik
- Linjär regression

ggplot2

- Skapat av Hadley Wickham för över 10 år sedan
- Baseras på “Grammar of Graphics” av Leland Wilkinson
- Alternativ till basgrafiken
- Grunden är alltid en `data.frame`

- Abstraktion av grafiska idéer
 - Tänk språk med ordklasser/satsdeelar
- Ger ett teoretiskt ramverk för att bygga grafik.
- Bygga upp grafik lager för lager

- Bygger upp en graf av flera delar:
 - data: en `data.frame` med **all** data
 - aes: aesthetic mappings
 - geom: geometriska objekt
 - facets: subplottar
 - scales: skalar
 - coordinate system: koordinatsystem

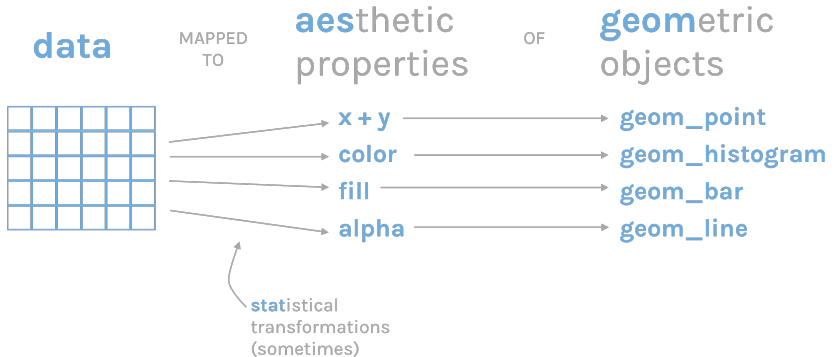


Bild från “R for the rest of us”

- `ggplot2` bygger upp en plot med olika lager
 - När plotten är klar så visas den
 - Kan också visa med `print()`
- Utgår från `ggplot()`
 - Returnerar ett objekt
- Adderar lager med `+`
 - t.ex. `+ geom_point()`
- Speciella klasser för `ggplot2`

"In brief, the grammar tells us that a statistical graphic is a mapping from data to aesthetic attributes (colour, shape, size) of geometric objects (points, lines, bars). The plot may also contain statistical transformations of the data and is drawn on a specific coordinate system."

Från "ggplot2 book" av Hadley Wickham

Kopplar ihop färg, form och utseende till data

aes	Beskrivning
x	x-axel
y	y-axel
size	storlek
color	färg
shape	form

Vilken geometrisk representation ska användas

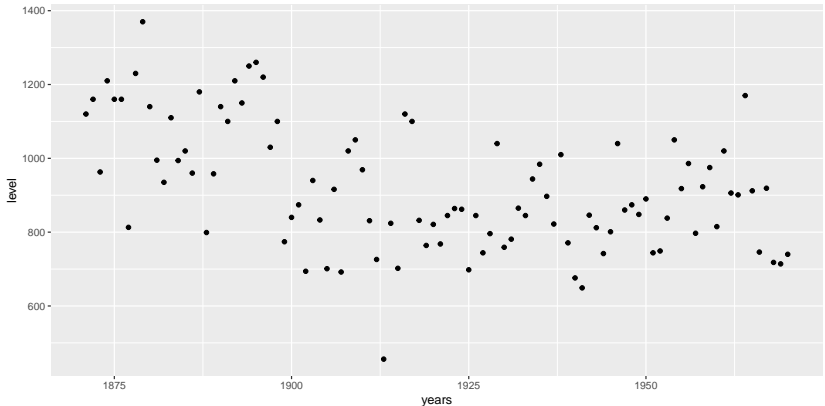
geom	Beskrivning
geom_point	Scatterplot
geom_line	Line graph
geom_bar	Barplot
geom_boxplot	Boxplot
geom_histogram	Histogram

Finns även speciella aesthetics för vissa geoms

geom	aes
geom_points	point shape, point size
geom_line	line type, line size
geom_bar	y min, y max, fill color, outline color

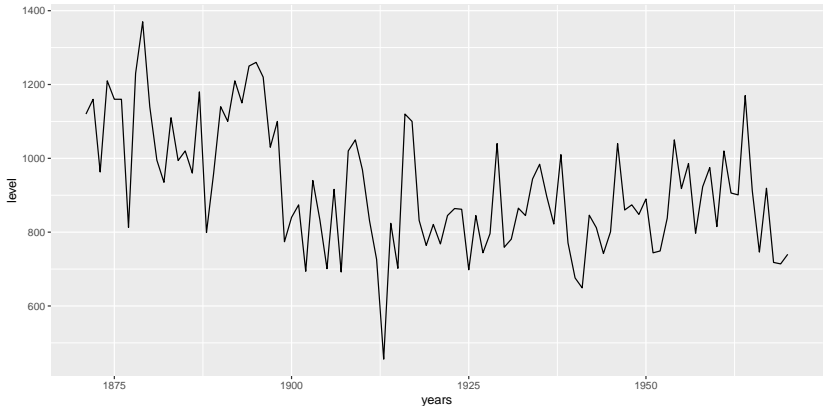
Exempel - I

```
ggplot(data = Nile) +  
  aes(x = years, y = level) +  
  geom_point()
```



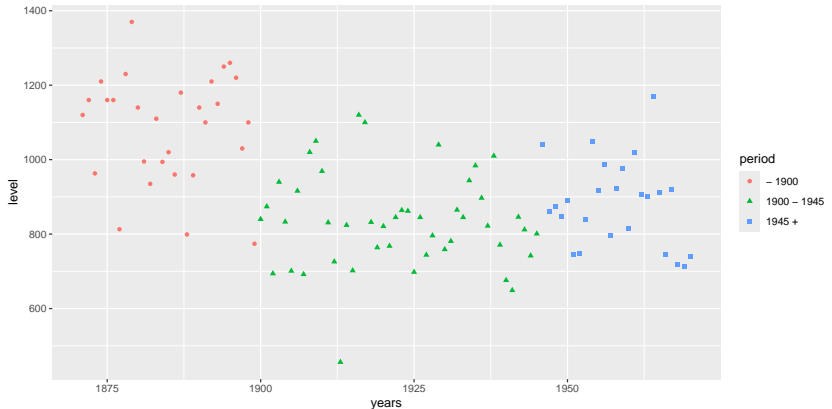
Exempel - II

```
ggplot(data = Nile) +  
  aes(x = years, y = level) +  
  geom_line()
```



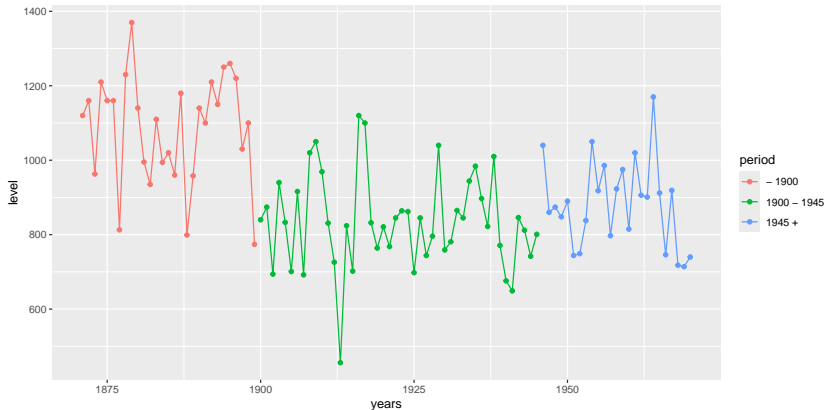
Exempel - III

```
ggplot(data = Nile) +  
  aes(x = years, y = level, color = period) +  
  geom_point(aes(shape = period))
```



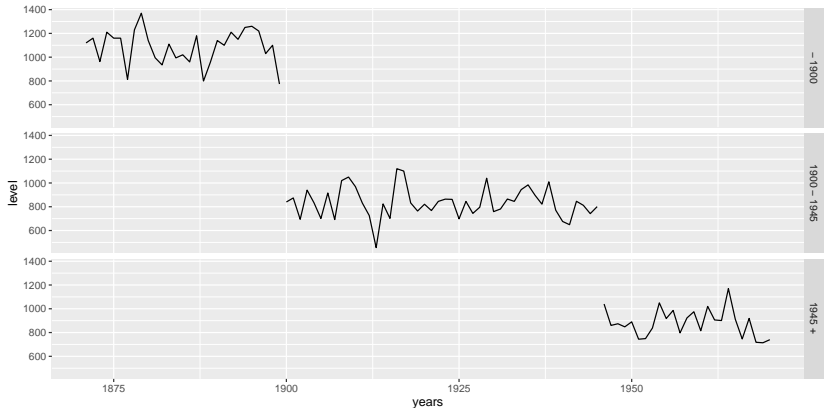
Exempel - IV

```
ggplot(data = Nile) +  
  aes(x = years, y = level, color = period) +  
  geom_line() +  
  geom_point()
```



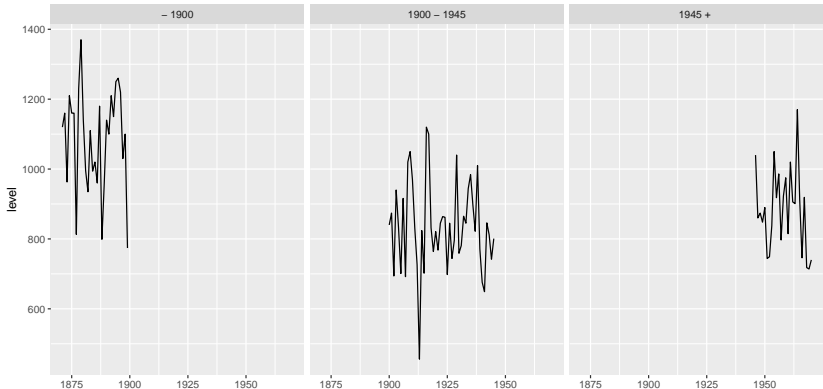
Exempel - V

```
ggplot(data = Nile) +  
  aes(x = years, y = level) +  
  facet_grid(period ~ .) +  
  geom_line()
```



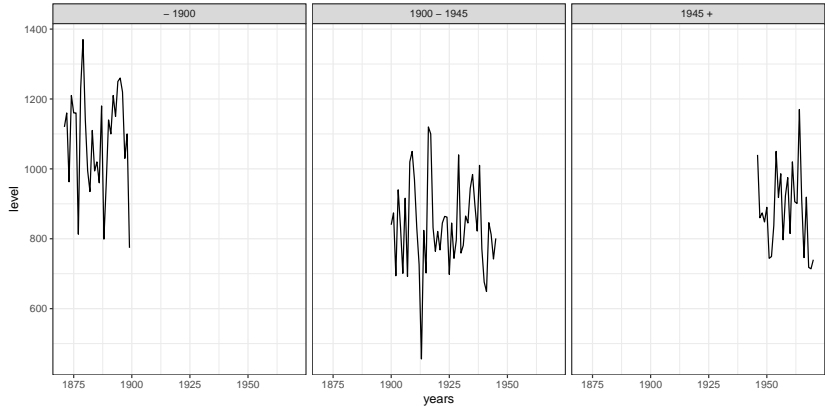
Exempel - VI

```
p <- ggplot(data = Nile) +  
  aes(x = years, y = level) +  
  facet_grid(~ period) +  
  geom_line()  
print(p)
```



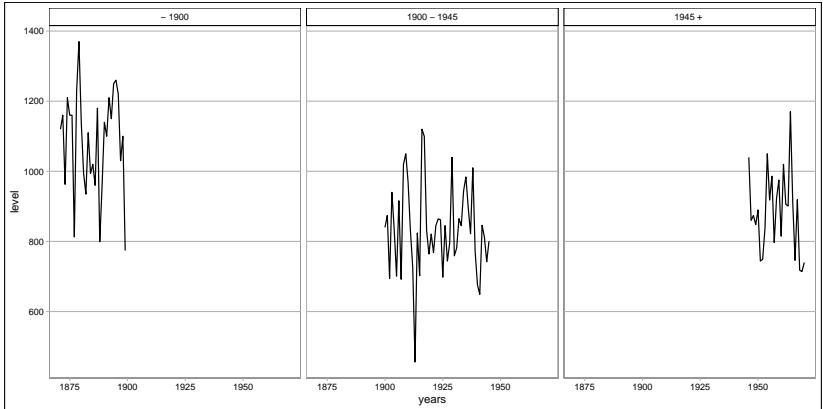
Exempel - VII : Teman

```
p + theme_bw()
```



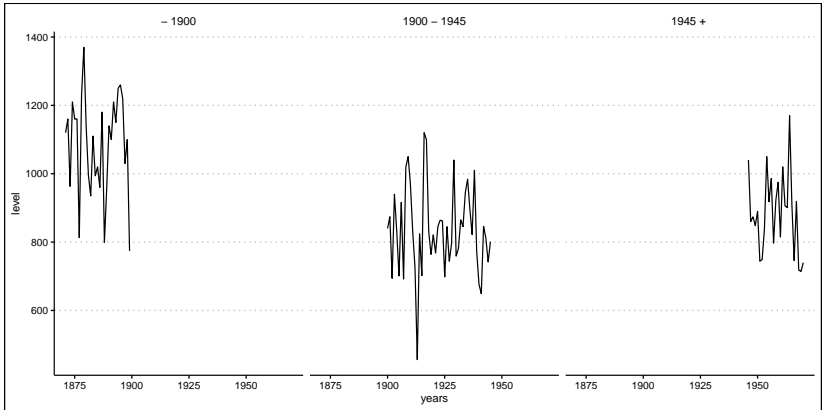
Exempel - VIII : Teman

```
p + theme_calc()
```



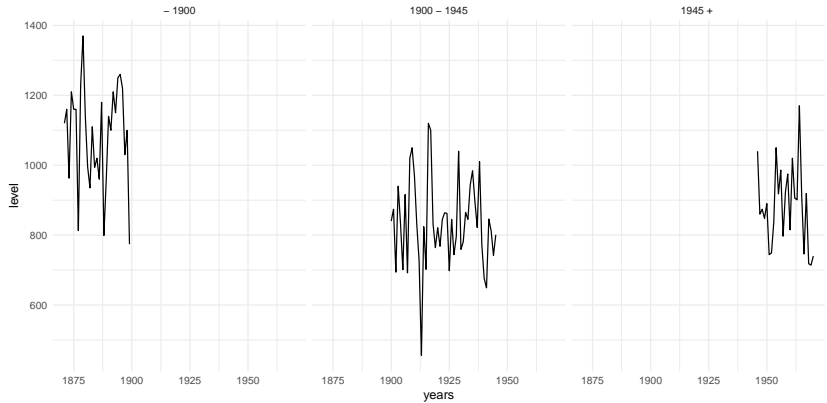
Exempel - IX : Teman

```
p + theme_clean()
```



Exempel - X : Teman

```
p + theme_minimal()
```



- `qplot()` liknar `plot()`
- Bra för snabba grafer
- För mer kontroll använd `ggplot()`

Demo: ggplot

Statistik

- Finns massor av olika statistiska tester
 - Väldigt många finns i R också
- För t-tester används `t.test()`
- För χ^2 -tester används
 - `chisq.test()`, `fisher.test()`
- Korrelation och kovarians kan beräknas och testas
 - `cor()` och `cov()`
 - `cor.test()`

Exempel: t.test() - I

```
data("chickwts")  
horsebean <- chickwts$weight[chickwts$feed == "horsebean"]  
sunflower <- chickwts$weight[chickwts$feed == "sunflower"]  
  
mean(horsebean)
```

```
## [1] 160.2
```

```
mean(sunflower)
```

```
## [1] 328.9167
```

Exempel: t.test() - II

```
t.test(horsebean, alternative = "two.sided",  
       mu = 150, conf.level = 0.95)
```

```
##  
## One Sample t-test  
##  
## data: horsebean  
## t = 0.83507, df = 9, p-value = 0.4253  
## alternative hypothesis: true mean is not equal to 150  
## 95 percent confidence interval:  
## 132.5687 187.8313  
## sample estimates:  
## mean of x  
## 160.2
```

Exempel: t.test() - III

```
t.test(horsebean, sunflower,  
       alternative = "two.sided",  
       mu = 0, conf.level = 0.95)
```

```
##  
## Welch Two Sample t-test  
##  
## data: horsebean and sunflower  
## t = -9.0449, df = 19.964, p-value = 1.69e-08  
## alternative hypothesis: true difference in means is not  
## 95 percent confidence interval:  
## -207.6313 -129.8021  
## sample estimates:  
## mean of x mean of y  
## 160.2000 328.9167
```

Demo: Statistik

Linjär regression

- I R finns formelobjektet som beskriver relationer mellan variabler
 - Formel skapas med `~`
 - Exempel: `y ~ x1 + x2`
- Att arbeta med modeller i R kan delas in i fyra steg:
 1. Anpassa (träna) en modell
 2. Analysera/studera resultatet
 3. Diagnostisera
 4. Använda modellen och resultaten
- Linjär regression handlar om att hitta en linjär modell

Linjär regression - Anpassa en modell

- Behöver en formel och data
- Data behöver samma variabler som formeln

```
library(MASS)
library(car)
data(Prestige)
```

```
mod1 <- lm(prestige ~ income + women + education, data=Pres
```

```
mod2 <- lm(prestige ~ income + women + education - 1, data=
```

```
mod3 <- lm(prestige ~ income:women + education, data=Pres
```

- Använd följande funktioner för att studera resultatet
 - `summary()`
 - `anova()`

Exempel:

```
summary(mod1)
anova(mod1)
anova(mod1, mod2, test = "Chisq")
```

- Finns ett antal olika metoder, ex:

```
plot(mod1)
durbinWatsonTest(mod1)
qqplot(mod1)
```

- När vi har en modell kan vi göra olika saker:
 - Publicera modellen
 - Studera residualer
 - Prediktion
- Vi kan spara vår modell och använda
 - `resid()`
 - `predict()`

Demo: Linjär regression