

Introduktion til ggplot

Statistik E24 (15 ECTS)

ved Mikkeline Munk Nielsen



Hvad er ggplot2?

- **ggplot2** er en datavisualiseringspakke i R
- Den er bygget på *grammar of graphics* og giver fleksibilitet til at skabe avancerede plots
- Det basale i ggplot2: *data*, *aes*, og *geometriske objekter (geoms)*



De tre grundelementer i ggplot2

1. **Data:** Data, du vil plotte
2. **Aesthetic mappings (aes):** Kortlægning af variabler
3. **Geometriske objekter (geom):** plot-type, f.eks. søjlediagram, scatterplot, eller boksplot

```
1 library(ggplot2)
2
3 ggplot(data = <DATA>) +
4   aes(x = <X-variabel>, y = <Y-variabel>) +
5   geom_<GEOM>()
```

Syntaks i ggplot

- ggplot2 er som at bygge med LEGO-klodser: Du tilføjer forskellige lag for at skabe det endelige plot
- Hvert lag bygger på det forrige og tilføjer nye visuelle elementer
- Basen er data, og du bygger videre med lag som akser, farver og geometriske former

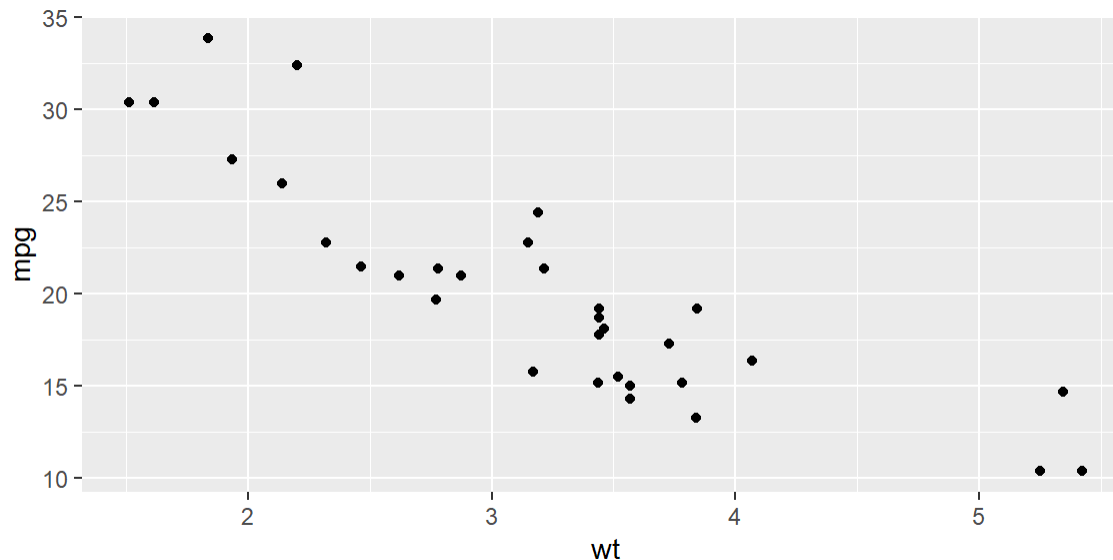
```
1 ggplot(data = <DATA>) +  
2   aes(x = <X-variabel>, y = <Y-variabel>) +  
3   geom_<GEOM>()
```

Eksempel på et simpelt plot

- Data fra **mtcars**-datasættet (built in datasæt på biler)
- Kortlægger variablene **wt** (vægt) til x-aksen og **mpg** (miles per gallon) til y-aksen
- `geom_point()` tilføjer punkterne til plottet

Scatterplot

Kode

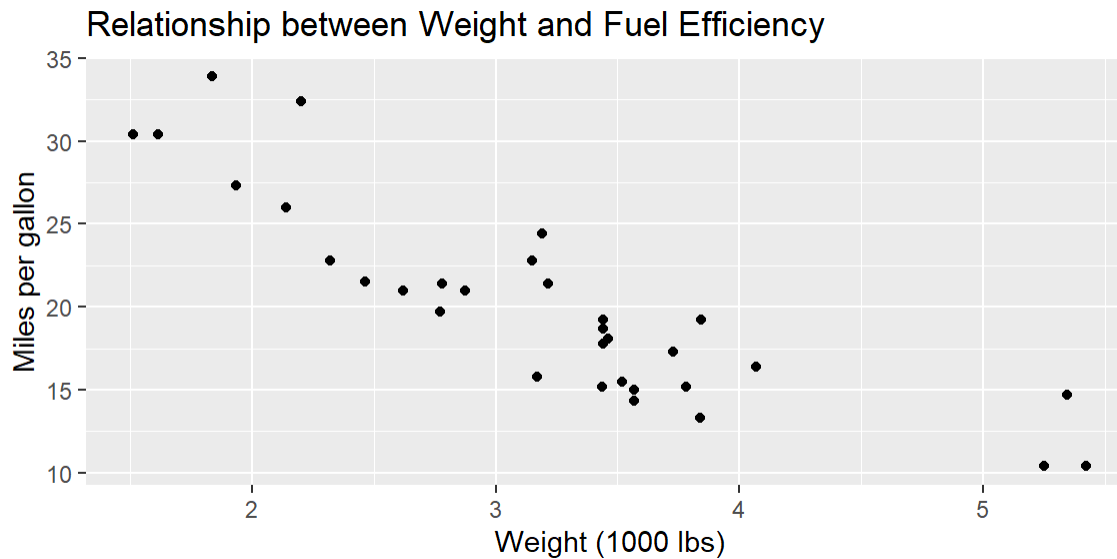


Ændring af plot-udseende

- Man kan bruge funktionen `labs()` til at tilføje titel og asketitler

Scatterplot

Kode

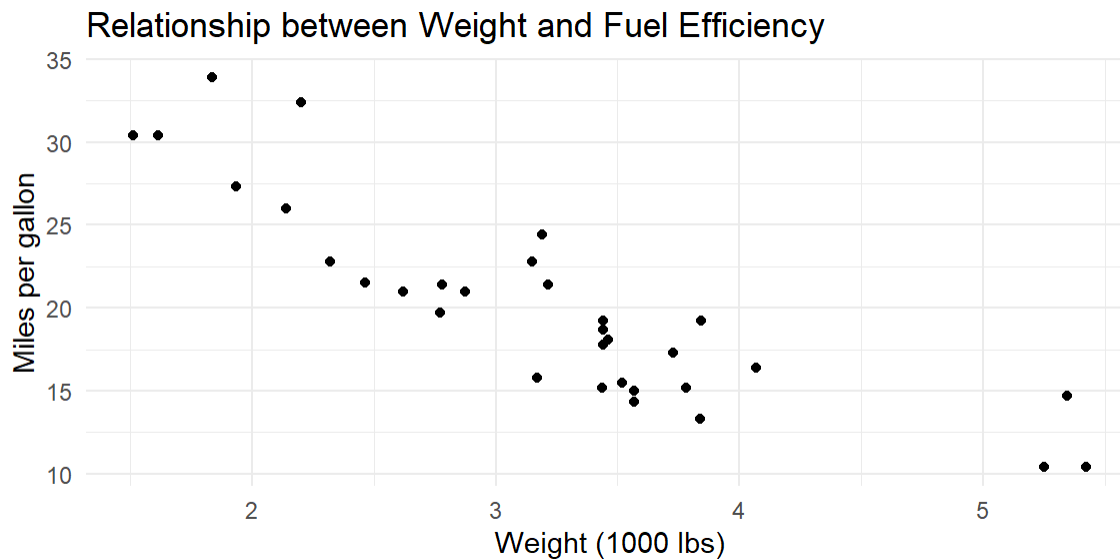


Ændring af plot-udseende

- Man kan ændre dine plots visuelle udseende med `themes()`
- Man kan f.eks. ændre baggrunde, akser, tekst, grids, osv., ved at bruge forskellige temaer
- ggplot2 kommer med flere indbyggede temaer, f.eks. `theme_minimal()`

Scatterplot

Kode



Ændring af plot-udseende

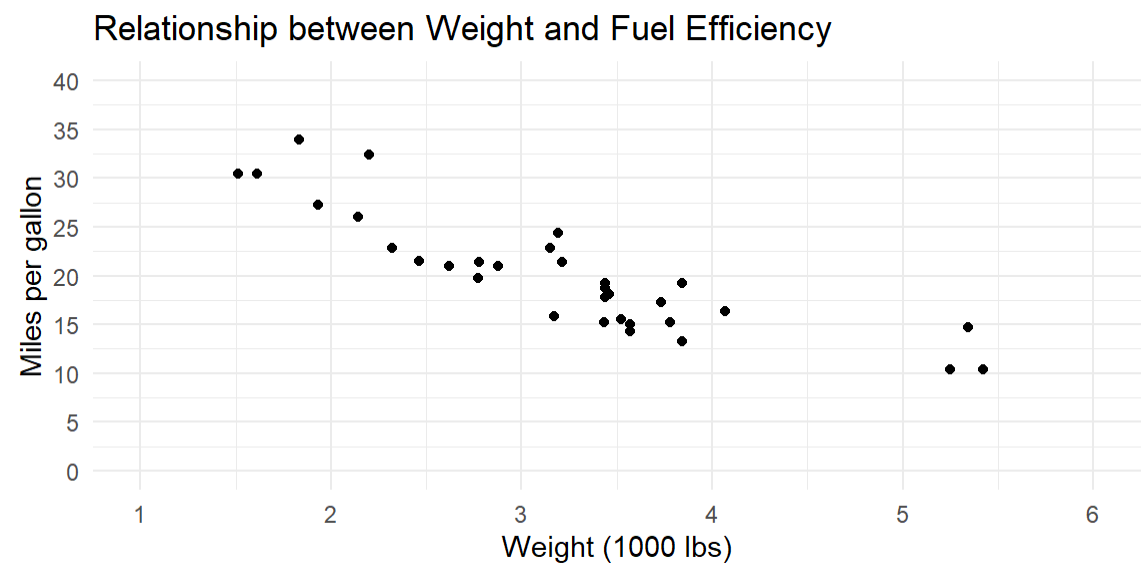
Det er ofte nødvendigt at skrue på værdierne på akserne. Afhængigt af hvilken akse man vil justere bruger man:

- `scale_x_discrete()`: Bruges til diskrete (kategoriske) værdier på x-aksen
- `scale_x_continuous()`: Bruges til kontinuære (numeriske) værdier på x-aksen
- `scale_y_discrete()`: Bruges til diskrete værdier på y-aksen
- `scale_y_continuous()`: Bruges til kontinuære værdier på y-aksen.

Ændring af plot-udseende

Scatterplot

Kode

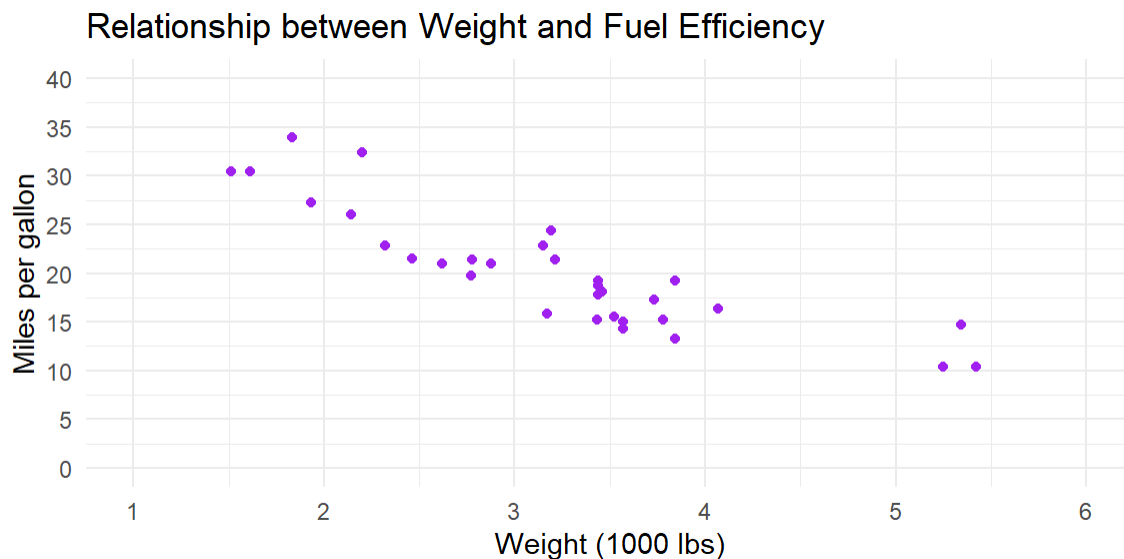


Ændring af plot-udseende

Man kan også ændre farver på sine ggplots. Afhængigt af hvilken type geometrisk objekt du arbejder med, bruger man `fill()` eller `color()`. `color()` ændrer kantfarver og punktfarver, `fill()` ændrer fyldfarver på objekter, der har en indre fyldning. I et scatterplot med punkter bruges `color()`:

Scatterplot

Kode



Ændring af plot-udseende

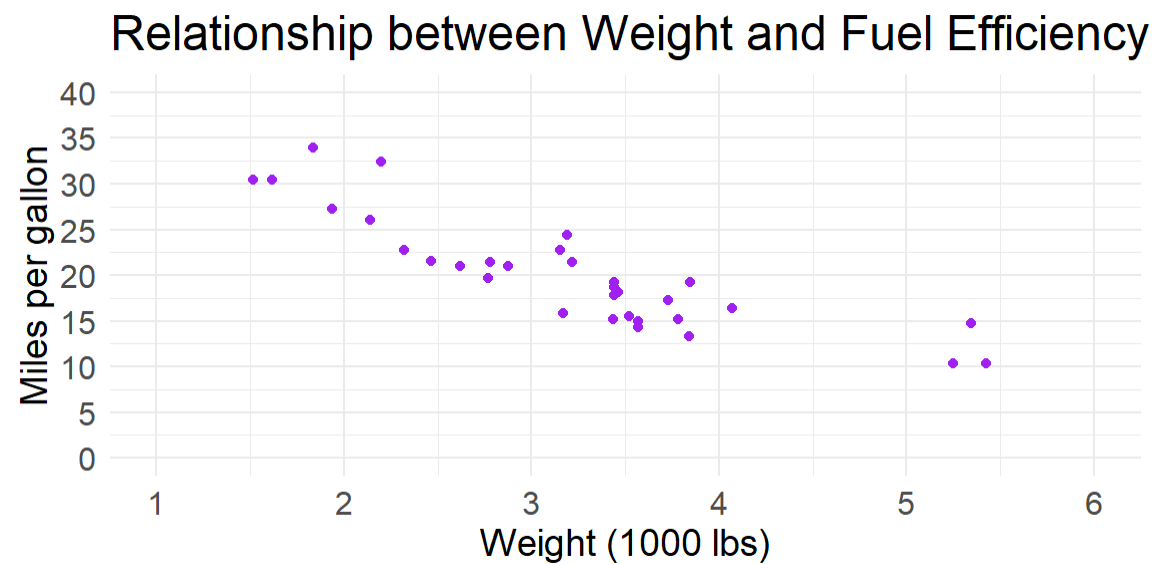
Man kan ændre tekst-størrelsen på plottets elementer med `theme()`:

```
1  theme(  
2    plot.title = element_text(size = 16),          # Ændrer titelstørrelsen  
3    axis.title.x = element_text(size = 12),        # Ændrer størrelsen på x-aksetitlen  
4    axis.title.y = element_text(size = 12),        # Ændrer størrelsen på y-aksetitlen  
5    axis.text.x = element_text(size = 10),         # Ændrer størrelsen på x-akse tekst  
    (ticks)  
6    axis.text.y = element_text(size = 10)          # Ændrer størrelsen på y-akse tekst  
    (ticks)  
7  )
```

Ændring af plot-udseende

Scatterplot

Kode

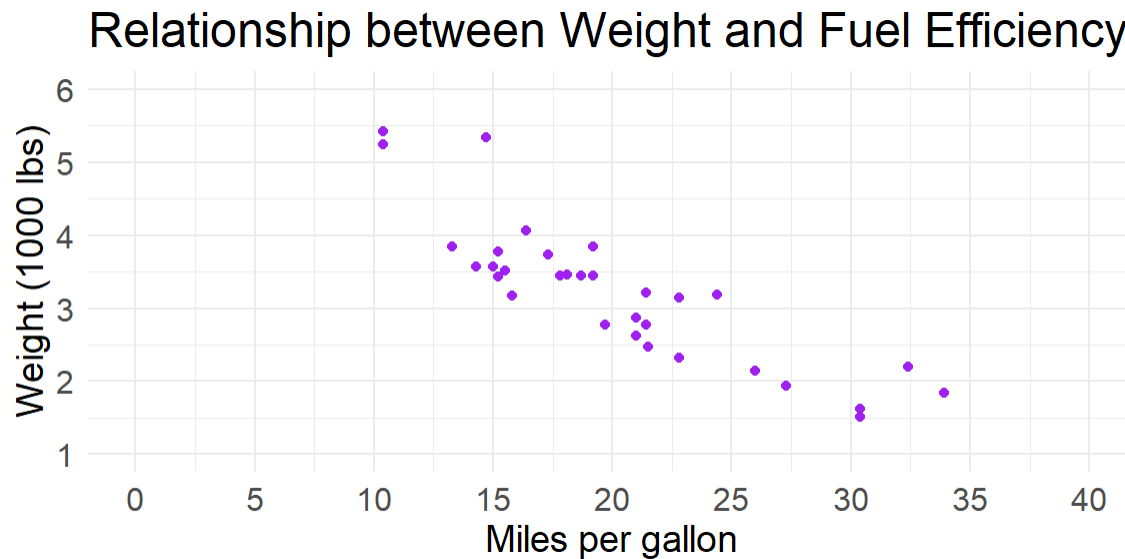


Ændring af plot-udseende

Nogle gange kan det også være nyttigt at rotere sit plot med `coord_flip()`:

Scatterplot

Kode



Typer af plots

Der findes mange typer af plots eller **geoms**, som er gode til forskellige ting. Her er et par grundlæggende:

Visualisering af én variabel

- Brug `geom_bar()` til kategoriske variable
- Brug `geom_histogram()` eller `geom_boxplot` til numeriske variable

Visualisering af flere variable

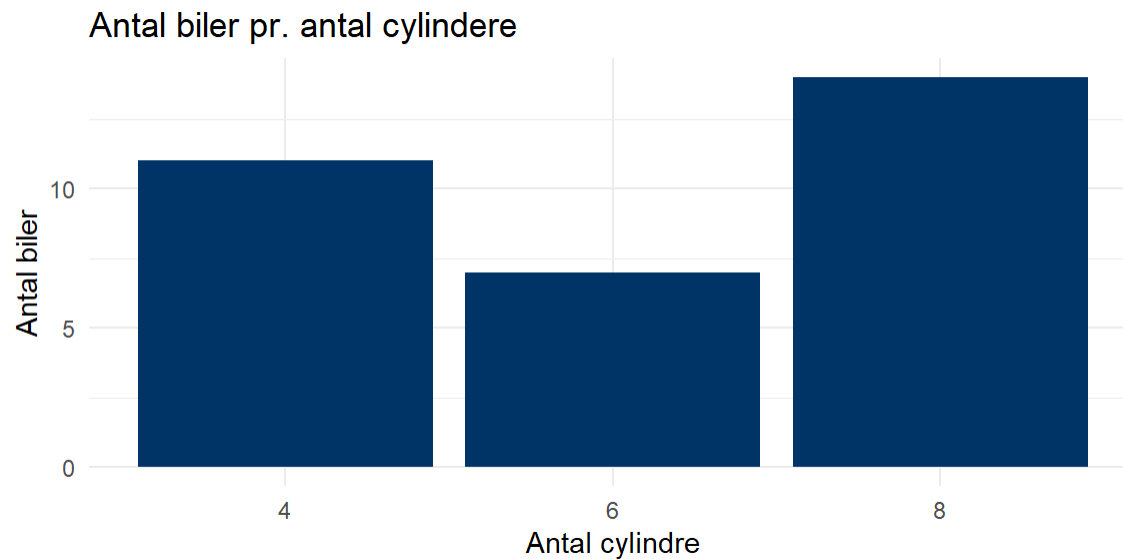
- Brug `geom_bar()` eller `geom_jitter()` til to kategoriske variable
- Brug `geom_boxplot` til en numerisk variabel og en kategoriske variabel.
- Brug `geom_point()` til to numeriske variable.

Barplot (univariat)

Variablen **cyl** (antal cylindre), som er en kategorisk variabel, kan visualiseres via. et barplot:

Plot

Kode

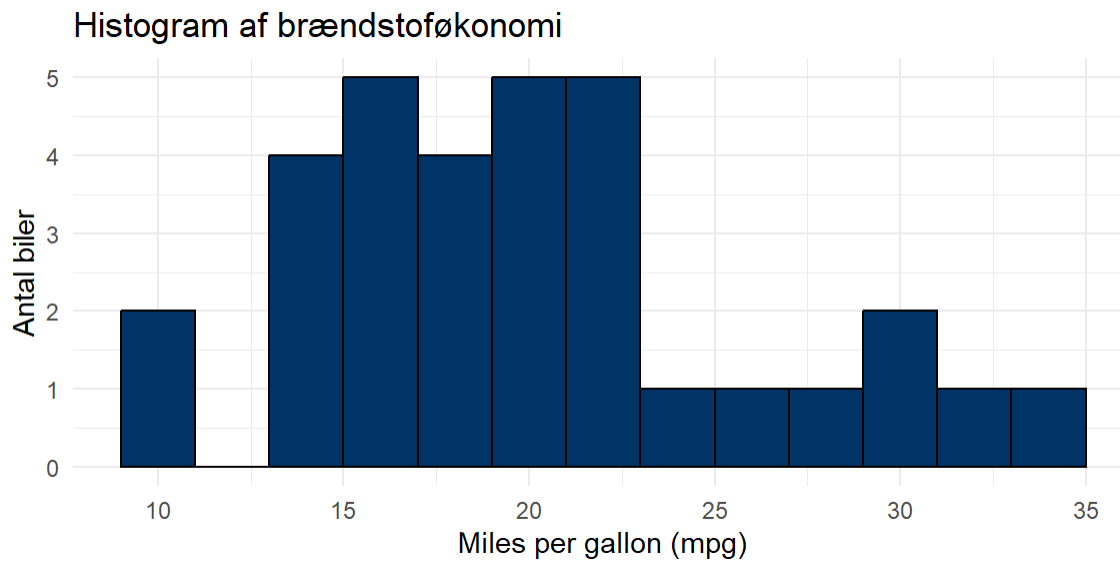


Histogram (univariat)

Variablen **mpg** (miles per gallon), som er en numerisk, kan visualiseres via. et histogram:

Plot

Kode

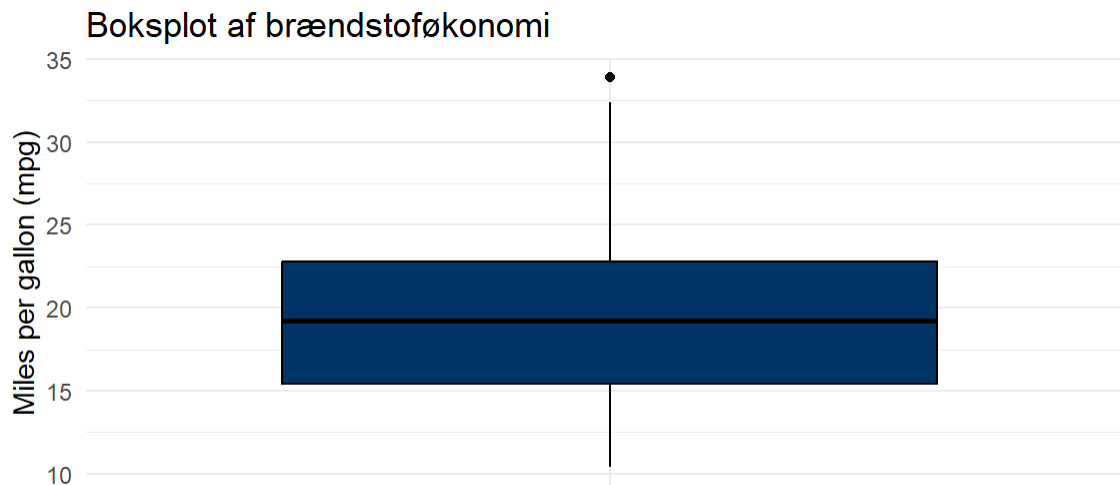


Boksplot (univariat)

Variablen **mpg** (miles per gallon), som er en numerisk, kan også visualiseres via. et boksplot:

Plot

Kode

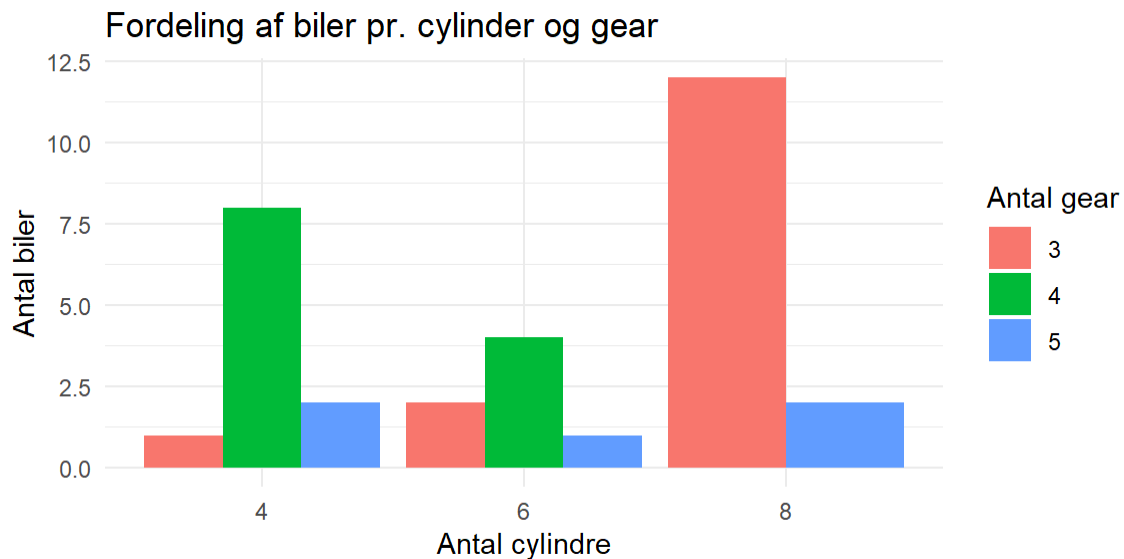


Barplot (bivariat)

Vi kan visualisere sammenhængen mellem variablene **cyl** (antal cylindre) og **gear** (antal gear), som begge er kategoriske variabler via `barplot()`:

Plot

Kode



Vi kan også visualisere sammenhængen mellem **cyl** (antal cylindre) og **gear** (antal gear) via `geom_jitter()`. Et jitter-plot er en variant af et scatterplot, hvor punkterne bliver “spredt” (jittered) lidt tilfældigt, når de overlapper hinanden:

Kode

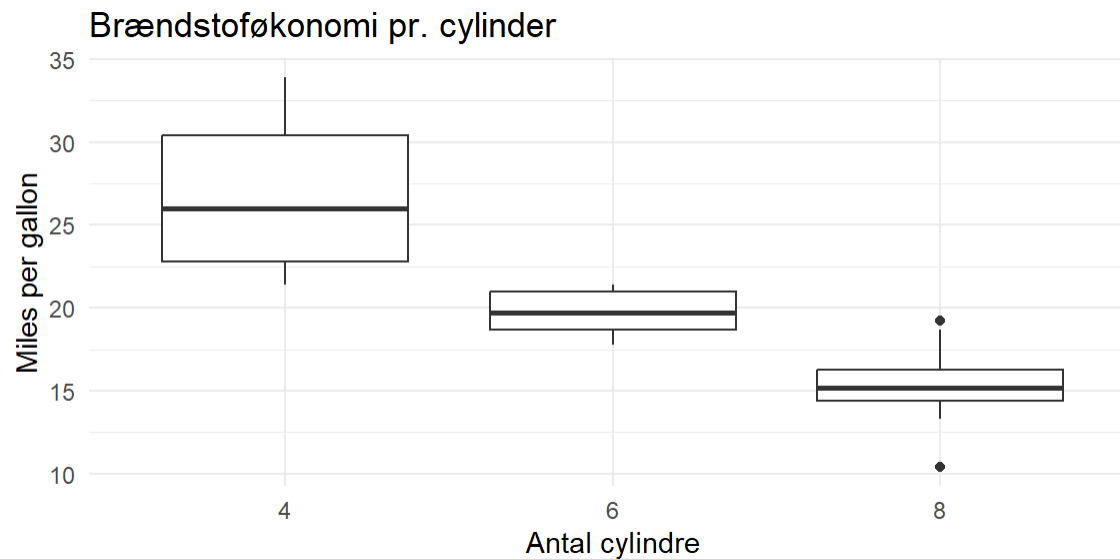


Boksplot (bivariat)

Vi kan bruge et boksplot til at se fordelingen af variablene **mpg** (nummerisk) over **cyl** (kategorisk):

Plot

Kode

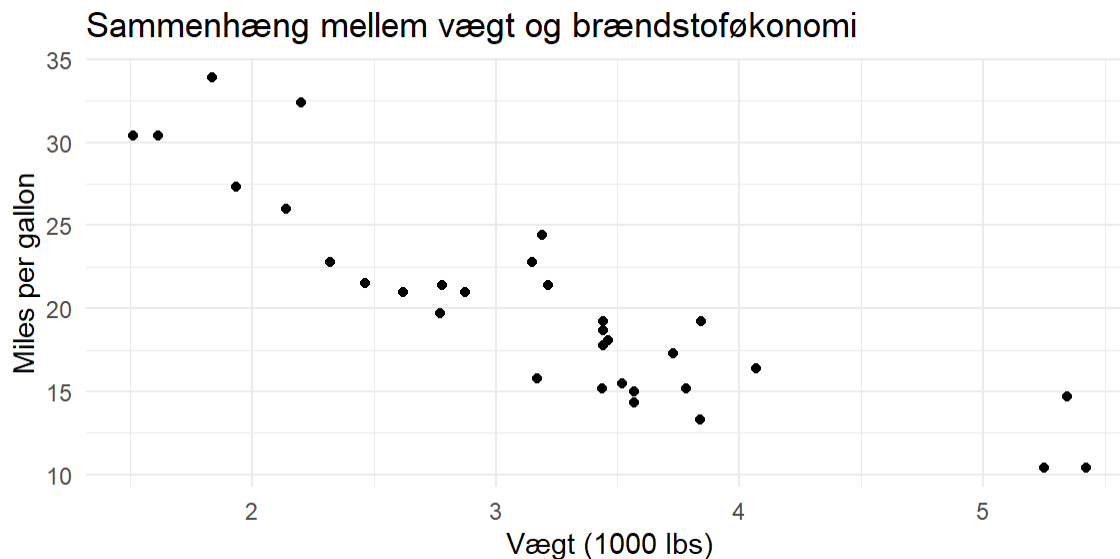


Scatterplot (bivariat)

Og vi kan bruge et scatterplot til at se sammenhængen mellem variablene **wt** (vægt) og **mpg** (miles per gallon), som begge er numeriske variabler med `geom_point()`:

Plot

Kode

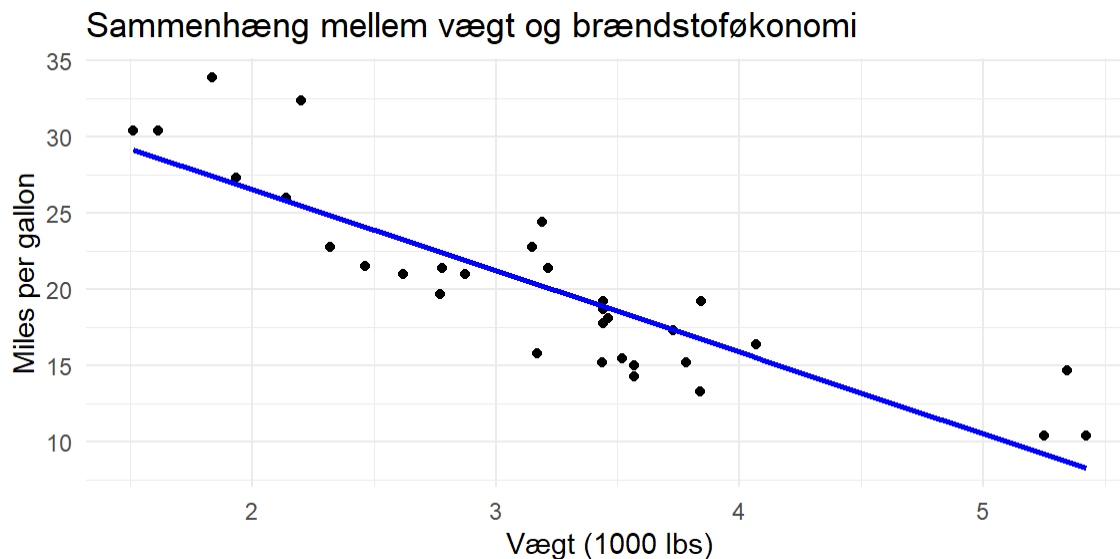


Scatterplot (bivariat) med regressionslinje

Det er også muligt at lægge flere plots ovenpå hinanden med ggplot. F.eks. kan I (senere hen) lægge en regressionslinje indover jeres scatterplot med `geom_smooth()` :

Plot

Kode



Gem dit ggplot

Standardmetoden til at gemme ggplot-grafer er ved hjælp af `ggsave()` funktionen.

- Først gemmes dit plot som et objekt
- Dernæst bruges `ggsave()` med følgende syntaks:

```
1 ggsave("mit_plot.jpeg", plot = mit_ggplot_objekt, width = 10, height = 8)
```

- `mit_ggplot_objekt` er navnet på det objekt, du har gemt dit plot i
- `ggsave()` kan gemme i forskellige filformater: JPEG, PNG, PDF, TIFF osv.
- Gem som udgangspunkt i JPEG
- Juster størrelse med `width` og `height` (i tommer).

Gem dit ggplot

```
1 mit_ggplot_objekt <- ggplot(mtcars) +
2   aes(x = wt, y = mpg) +
3   geom_point() +
4   geom_smooth(method = "lm", color = "blue", se = FALSE) + # Tilføjer en lineær
  regressionslinje
5   labs(
6     title = "Sammenhæng mellem vægt og brændstoføkonomi",
7     x = "Vægt (1000 lbs)",
8     y = "Miles per gallon"
9   ) +
10  theme_minimal()
11
12 # Gem mit plot i R-project mappen på computeren
13 ggsave("mit_plot.jpeg", plot = mit_ggplot_objekt, width = 10, height = 8)
```


Mere om ggplot

Eksemplerne her giver jer nogle skabeloner til at arbejde med en række grundlæggende plots, men mulighederne for at justere på dem og gøre dem flotte er uendelige!

- Brug [ggplot-bogen](#)
- Kig på [ggplot2 cheatsheet](#)
- Søg hjælp hos ChatGPT, der kan fejlsøge i din kode