Universiteit Utrecht

Leveraging register data to estimate causal effects of policy interventions

Workshop ODISSEI

Oisín Ryan & Erik-Jan van Kesteren

About us

Erik-Jan van Kesteren

- Background in statistics / social science
- Assistant professor @ methodology & statistics UU
- Social Data Science team lead @ ODISSEI (consortium of universities)

SoDA

Some stuff I work on:

Latent variables, high-dimensional data, optimization, regularization, visualisation, Bayesian statistics, multilevel models, spatial data, generalized linear models, privacy, synthetic data, high-performance computing, software development, open science & reproducibility

About us

Erik-Jan van Kesteren

- Background in statistics / social science
- Assistant professor @ methodology & statistics UU
- Social Data Science team lead @ ODISSEI (consortium of universities)

SoDA

Some stuff I work on:

Latent variables, high-dimensional data, optimization, regularization, visualisation, Bayesian statistics, multilevel models, spatial data, generalized linear models, privacy, synthetic data, high-performance computing, software development, open science & reproducibility

github.com/sodascience/workshop_causal_impact_assessment

Today's plan: morning

- Introduction (60 minutes)
  - Policy Interventions and Causal Inference
  - Pre-Post Analyses and Difference-in-Difference
- Practical (30 minutes)
- Break (15 minutes)

- Interrupted Time Series (45 minutes)
- Practical (30 minutes)

- Lunch around 12:00 ; re-start at 13:00

Today's plan: afternoon

- Synthetic Control Methods (45 minutes)
- Practical (45 minutes)
- Break (15 minutes)

- CausalImpact (45 minutes)
- Practical (45 minutes)
- Break (15 minutes)

- Discussion session (30 minutes)

- Finish around 17:00

Context: “Policy Evaluations”

Many social science **research questions** concern evaluating what **the effect** of implementing a particular **policy** or **intervention** was on some outcome of interest

Examples:

- What was the effect of raising the maximum speed limit on road deaths?
- What effect did introducing students loans have on post-graduation debt levels?
- Did introducing an after-school programme in disadvantaged neighbourhoods lead to improved educational outcomes in children from that neighbourhood?

Context: “Policy Evaluations”

Sometimes referred to as “policy evaluation” research or “comparative case studies”

Basic Structure:

- We have some **unit** which we observe **before** and **after** some intervention or action
- Did the intervention produces changes in the outcome for that unit?

Methods for Policy Evaluation

Many different methods have been developed to answer these types of research questions

These methods differ in terms of:

- The **amount** and **type** of information they use
  - Amount of time-points and amount of potential “control” units
- The specific **statistical approach** they take
- The types of **assumptions** they make

Today we aim to give you a brief introduction to many of these different methods!

Big Table of Methods

Causal Inference: A primer

Potential Outcomes

Causal Inference is (broadly) concerned with using data to estimate what the effect is of **intervening** on a particular variable.

Using the **potential outcomes** framework, we can define causal inference as a missing data problem

Potential Outcomes

Let Y_i represent your headache level, and let A_i be whether you take aspirin or not (A =1 you take it, A = 0 you don't)

You only want to take an aspirin if your headache levels **after taking aspirin** would be lower than your headache level **without taking aspirin**

There are **two possible versions** of the outcome variable

- Y_i^1 your headache level **if you would take aspirin**
- Y_i^0 your headache level **if you would take aspirin**

Causal Effects

We can define the **causal effect** of taking aspirin on your headache levels as the difference in potential outcomes

$$CE_i = Y_i^1 - Y_i^0$$

The **fundamental problem of causal inference:** You only ever observe one of the potential outcomes!

Data and Potential Outcomes

ID	Y	A
1	7	0
2	9	0
3	6	0
4	5	0
5	6	0
6	2	1
7	3	1
8	1	1
…	…	…
I	2	1

Data and Potential Outcomes

ID	Y	A
1	7	0
2	9	0
3	6	0
4	5	0
5	6	0
6	2	1
7	3	1
8	1	1
...	...	...
I	2	1

Data and Potential Outcomes

ID	Y	A	Y^0	Y^1
1	7	0	7	NA
2	9	0	9	NA
3	6	0	6	NA
4	5	0	5	NA
5	6	0	6	NA
6	2	1	NA	2
7	3	1	NA	3
8	1	1	NA	1
…	…	…	…	…
I	2	1	NA	2

Data and Potential Outcomes

ID	Y	A	Y^0	Y^1
1	7	0	7	NA
2	9	0	9	NA
3	6	0	6	NA
4	5	0	5	NA
5	6	0	6	NA
6	2	1	NA	2
7	3	1	NA	3
8	1	1	NA	1
…	…	…	…	…
I	2	1	NA	2

Causal Inference

In cross-sectional settings, we typically aim to make inferences about the **average causal effect.** This is known as a **causal estimand:**

$$ACE = E[Y^1] - E[Y^0]$$

In a **Randomized Controlled Trial,** we often use the (sample) difference in treated and untreated groups as an **estimator** of this causal effect:

$$\widehat{ACE} = E[Y|\, A = 1] - E[Y\,|A = 0]$$

Causal Inference

In cross-sectional settings, we typically aim to make inferences about the **average causal effect.** This is known as a **causal estimand:**

$$ACE = E[Y^1] - E[Y^0]$$

In a **Randomized Control Trial,** we often use the (sample) difference in treated and untreated groups as an **estimator** of this causal effect:

$$\widehat{ACE} = E[Y|\, A = 1] - E[Y\,|A = 0]$$

Causal Inference

ID	Y	A	Y^0	Y^1
1	7	0	7	NA
2	9	0	9	NA
3	6	0	6	NA
4	5	0	5	NA
5	6	0	6	NA
6	2	1	NA	2
7	3	1	NA	3
8	1	1	NA	1
…	…	…	…	…
I	2	1	NA	2

Causal Inference

In cross-sectional settings, we typically aim to make inferences about the **average causal effect.** This is known as a **causal estimand:**

$$ACE = E[Y^1] - E[Y^0]$$

In a **Randomized Control Trial,** we often use the (sample) difference in treated and untreated groups as an **estimator** of this causal effect:

$$\widehat{ACE} = E[Y|\, A = 1] - E[Y\,|A = 0]$$

Causal Inference

ID	Y	A	Y^0	Y^1
1	7	0	7	NA
2	9	0	9	NA
3	6	0	6	NA
4	5	0	5	NA
5	6	0	6	NA
6	2	1	NA	2
7	3	1	NA	3
8	1	1	NA	1
…	…	…	…	…
I	2	1	NA	2

Causal Inference Assumptions

This type of **inference** about causal effects from **observed data** is only possible under certain **conditions** or **assumptions**

Exchangeability

- Essentially relates to the absence of (unaccounted for) **confounder variables**
- If we were to reverse treatment assignment we would observe the same group differences. Information is exchangeable between groups
- **RCTs** are powerful because **randomization** ensures exchangeability
- In practice we need **conditional exchangeability**; to control for **confounders!**

Causal Inference Assumptions

This type of **inference** about causal effects from **observed data** is only possible under certain **conditions** or **assumptions**

Stable Unit Treatment Value (also known as SUTVA)

- **No Interference:** The potential outcomes of one unit does not depend on the treatment assigned to another unit. E.g.: My taking an aspirin does not influence your headache levels if you do or do not take one
- **Consistency:** Only one version of treatment, treatment is unambiguously defined.
- I can directly observe **one of the potential outcomes**.
- If person I takes aspirin, then $Y_i = Y_i^1$

Causal Inference Assumptions

These two generic assumptions essentially always appear in causal inference problems, and as we will see, we will have to deal with concerns around **confounders** and **no interference** repeatedly today

Other assumptions or conditions may also be needed depending on the specific **design** and **analytic approach you take**

Causal Inference and Policy Evaluations

Todays Topic

Policy evaluation is a special case of causal inference

We have **one unit** observed repeatedly over time

At some point in time an **intervention** takes place

Data and Potential Outcomes

$Time$	Y_t	A_t
1	7	0
2	9	0
3	6	0
4	5	0
5	6	0
6	2	1
7	3	1
8	1	1
…	…	…
I	2	1

Todays Topic

Policy evaluation is a special case of causal inference

We have **one unit** observed repeatedly over time

At some point in time an **intervention** takes place

Pre-intervention we observe Y_t^0 and **post-intervention** Y_t^1

Data and Potential Outcomes

$Time$	Y_t	A_t	Y_t^0	Y_t^1
1	7	0	7	NA
2	9	0	9	NA
3	6	0	6	NA
4	5	0	5	NA
5	6	0	6	NA
6	2	1	NA	2
7	3	1	NA	3
8	1	1	NA	1
…	…	…	…	…
I	2	1	NA	2

Causal Inference

In cross-sectional settings, we typically aim to make inferences about the **average causal effect.** This is known as a **causal estimand:**

$$ACE = E[Y^1] - E[Y^0]$$

In a **Randomized Control Trial,** we often use the (sample) difference in treated and untreated groups as an **estimator**

$$\widehat{ACE} = E[Y|\ A = 1] - E[Y\ |A = 0]$$

Causal Inference

We can define the **causal effect** of taking aspirin on your headache levels as the difference in potential outcomes

$$CE_i = Y_i^1 - Y_i^0$$

The **fundamental problem of causal inference:** You only ever observe one of the potential outcomes!

Data visualisation with ggplot2

https://r4ds.had.co.nz/data-visualisation.html

Raw data maps to:

- Aesthetics: data-bound properties of the picture (position, shape, colour, ...)
- Geometric objects, or geom: visual objects on the plot (points, lines, bars, polygons ...)
- Scales: how data values map to aesthetic values (continuous or discrete)
- Facets: subplots / small multiples

Additionally, can apply:

- Statistical transformations: transform data before mapping
- Alternative coordinate system (cartesian, polar, ...)

Example dataset: cars

```
> mpg
# A tibble: 234 × 11
   manufacturer model      displ  year   cyl trans      drv     cty   hwy fl    class
   <chr>        <chr>      <dbl> <int> <int> <chr>      <chr> <int> <int> <chr> <chr>
 1 audi         a4           1.8  1999     4 auto(l5)   f        18    29 p     compact
 2 audi         a4           1.8  1999     4 manual(m5) f        21    29 p     compact
 3 audi         a4           2    2008     4 manual(m6) f        20    31 p     compact
 4 audi         a4           2    2008     4 auto(av)   f        21    30 p     compact
 5 audi         a4           2.8  1999     6 auto(l5)   f        16    26 p     compact
 6 audi         a4           2.8  1999     6 manual(m5) f        18    26 p     compact
 7 audi         a4           3.1  2008     6 auto(av)   f        18    27 p     compact
 8 audi         a4 quattro   1.8  1999     4 manual(m5) 4        18    26 p     compact
 9 audi         a4 quattro   1.8  1999     4 auto(l5)   4        16    25 p     compact
10 audi         a4 quattro   2    2008     4 manual(m6) 4        20    28 p     compact
# … with 224 more rows
# ℹ Use `print(n = ...)` to see more rows
```

Data wrangling with dplyr

Which are the most efficient (combined city-highway) cars after 2000 in terms of litres / 100km?

```
1  library(tidyverse)
2
3  l100k ←
4    mpg ▷
5    filter(year > 2000) ▷
6    select(manufacturer, model, year, displ, cty, hwy, class) ▷
7    mutate(
8      mpg_combined = .55*cty + .45*hwy,
9      l_per_100km = 378.5411784 / (1.609344 * mpg_combined)
10   ) ▷
11   select(-cty, -hwy, -mpg_combined) ▷
12   arrange(l_per_100km)
13
```

Data wrangling with dplyr

```
# A tibble: 117 × 6
   manufacturer model          year displ class      l_per_100km
   <chr>        <chr>         <int> <dbl> <chr>            <dbl>
 1 toyota       corolla        2008   1.8 compact           7.34
 2 toyota       corolla        2008   1.8 compact           7.83
 3 honda        civic          2008   1.8 subcompact        7.85
 4 honda        civic          2008   1.8 subcompact        7.95
 5 honda        civic          2008   1.8 subcompact        8.00
 6 nissan       altima         2008   2.5 midsize           8.70
 7 nissan       altima         2008   2.5 midsize           8.84
 8 toyota       camry solara   2008   2.4 compact           9.03
 9 chevrolet    malibu         2008   2.4 midsize           9.19
10 hyundai      sonata         2008   2.4 midsize           9.22
# … with 107 more rows
# ℹ Use `print(n = ...)` to see more rows
```

Data visualisation with ggplot2

https://r4ds.had.co.nz/data-visualisation.html

Raw data maps to:

- Aesthetics: data-bound properties of the picture (position, shape, colour, ...)
- Geometric objects, or geom: visual objects on the plot (points, lines, bars, polygons ...)
- Scales: how data values map to aesthetic values (continuous or discrete)
- Facets: subplots / small multiples

Additionally, can apply:

- Statistical transformations: transform data before mapping
- Alternative coordinate system (cartesian, polar, ...)

Example dataset: l100k

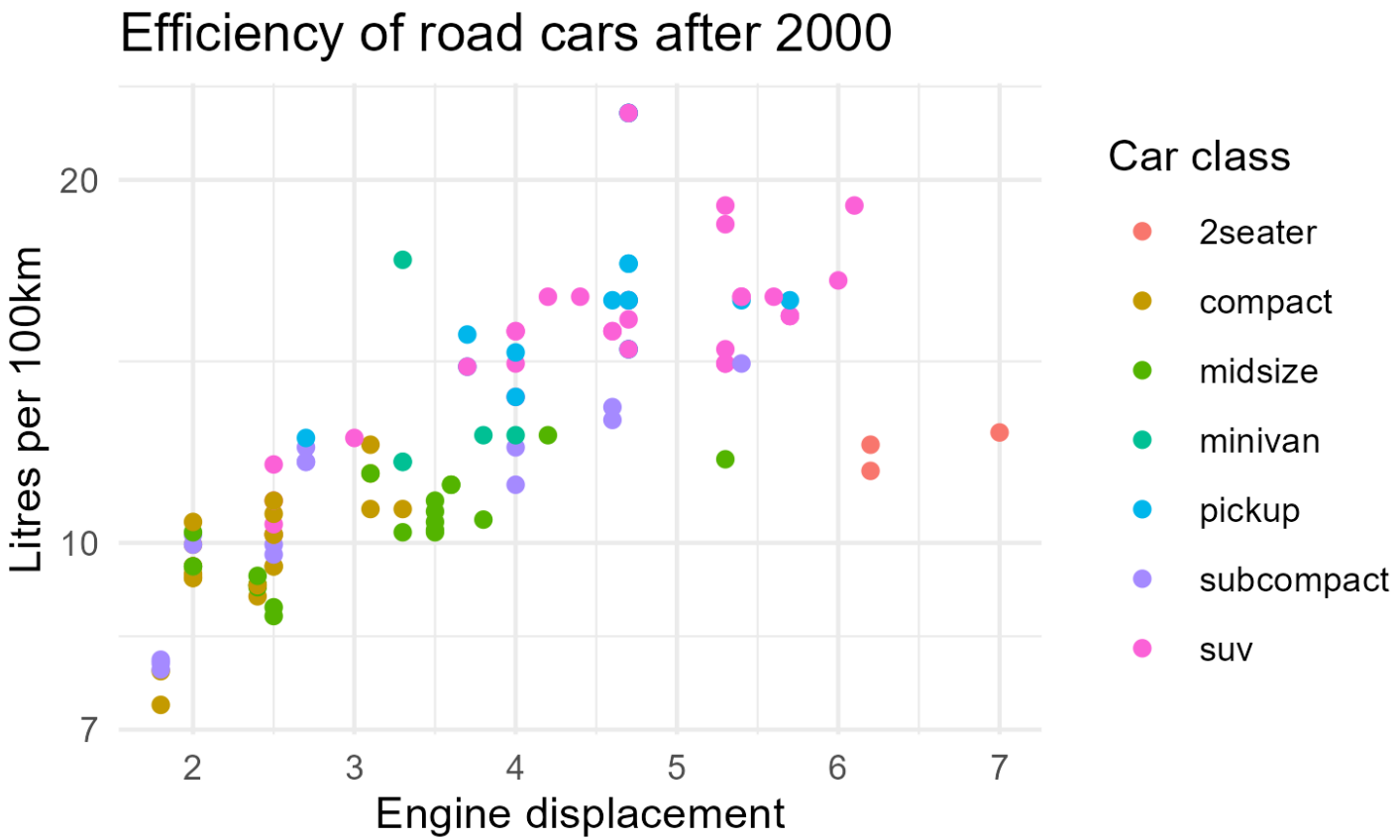
```
# A tibble: 117 × 6
   manufacturer model          year displ class      l_per_100km
   <chr>        <chr>         <int> <dbl> <chr>            <dbl>
 1 toyota       corolla        2008   1.8 compact           7.34
 2 toyota       corolla        2008   1.8 compact           7.83
 3 honda        civic          2008   1.8 subcompact        7.85
 4 honda        civic          2008   1.8 subcompact        7.95
 5 honda        civic          2008   1.8 subcompact        8.00
 6 nissan       altima         2008   2.5 midsize           8.70
 7 nissan       altima         2008   2.5 midsize           8.84
 8 toyota       camry solara   2008   2.4 compact           9.03
 9 chevrolet    malibu         2008   2.4 midsize           9.19
10 hyundai      sonata         2008   2.4 midsize           9.22
# … with 107 more rows
# ℹ Use `print(n = ...)` to see more rows
```

Data visualisation with ggplot2

Display a scatter plot with on the x axis the engine displacement and on the y-axis the efficiency. Colour the points by the car class

```
14  l100k ▷
15    ggplot(aes(x = displ, y = l_per_100km, colour = class)) +
16    geom_point() +
17    scale_y_log10() +
18    labs(
19      x       = "Engine displacement",
20      y       = "Litres per 100km",
21      colour  = "Car class",
22      title   = "Efficiency of road cars after 2000"
23    ) +
24    theme_minimal()
```

Data visualisation with ggplot2



More data wrangling with dplyr

group by

Registers groups by which to perform further operations (usually mutate, summarise)

summarise

Create summaries based on a function applied to each group

Example dataset: l100k

```
# A tibble: 117 × 6
   manufacturer model          year displ class      l_per_100km
   <chr>        <chr>         <int> <dbl> <chr>            <dbl>
 1 toyota       corolla        2008   1.8 compact           7.34
 2 toyota       corolla        2008   1.8 compact           7.83
 3 honda        civic          2008   1.8 subcompact        7.85
 4 honda        civic          2008   1.8 subcompact        7.95
 5 honda        civic          2008   1.8 subcompact        8.00
 6 nissan       altima         2008   2.5 midsize           8.70
 7 nissan       altima         2008   2.5 midsize           8.84
 8 toyota       camry solara   2008   2.4 compact           9.03
 9 chevrolet    malibu         2008   2.4 midsize           9.19
10 hyundai      sonata         2008   2.4 midsize           9.22
# … with 107 more rows
# ℹ Use `print(n = ...)` to see more rows
```

More data wrangling with dplyr

Which type of car should I buy to be the least efficient?

```
28  l100k ▷
29    group_by(class) ▷
30    summarise(avg = mean(l_per_100km)) ▷
31    arrange(desc(avg))
```

```
# A tibble: 7 × 2
  class          avg
  <chr>        <dbl>
1 pickup       16.4
2 suv          15.3
3 minivan      13.0
4 2seater      12.0
5 subcompact   10.7
6 midsize      10.2
7 compact       9.82
```

More data wrangling with dplyr

What is the most efficient car within each class?

```
34  l100k ▷
35    group_by(class) ▷
36    arrange(l_per_100km) ▷
37    summarise(
38      manufacturer = first(manufacturer),
39      model = first(model)
40    )
```

```
# A tibble: 7 × 3
  class      manufacturer model
  <chr>      <chr>        <chr>
1 2seater    chevrolet    corvette
2 compact    toyota       corolla
3 midsize    nissan       altima
4 minivan    dodge        caravan 2wd
5 pickup     toyota       toyota tacoma 4wd
6 subcompact honda        civic
7 suv        subaru       forester awd
```

Pivoting with tidyr

pivot_longer

Combines various columns into a single “value” column with an additional “name” column to indicate where the value came from

pivot_wider

The opposite: puts rows of different categories in separate columns

Example dataset: l100k

```
# A tibble: 117 × 6
   manufacturer model          year displ class      l_per_100km
   <chr>        <chr>         <int> <dbl> <chr>            <dbl>
 1 toyota       corolla        2008   1.8 compact           7.34
 2 toyota       corolla        2008   1.8 compact           7.83
 3 honda        civic          2008   1.8 subcompact        7.85
 4 honda        civic          2008   1.8 subcompact        7.95
 5 honda        civic          2008   1.8 subcompact        8.00
 6 nissan       altima         2008   2.5 midsize           8.70
 7 nissan       altima         2008   2.5 midsize           8.84
 8 toyota       camry solara   2008   2.4 compact           9.03
 9 chevrolet    malibu         2008   2.4 midsize           9.19
10 hyundai      sonata         2008   2.4 midsize           9.22
# … with 107 more rows
# ℹ Use `print(n = ...)` to see more rows
```

Pivoting with tidyr

Let's generate predictions for efficiency using two models: a linear regression and a regression tree

```
42  # fit
43  fit_linear ← lm(l_per_100km ~ displ + class, l100k)
44  fit_tree   ← rpart(l_per_100km ~ displ + class, l100k)
45
46  # predict & add to data
47  l100k$pred_linear ← predict(fit_linear)
48  l100k$pred_tree ← predict(fit_tree)
```

Pivoting with tidyr

```
# A tibble: 117 × 8
   manufacturer model          year displ class      l_per_100km pred_linear pred_…¹
   <chr>        <chr>         <int> <dbl> <chr>            <dbl>       <dbl>   <dbl>
 1 toyota       corolla        2008   1.8 compact           7.34        8.98    9.37
 2 toyota       corolla        2008   1.8 compact           7.83        8.98    9.37
 3 honda        civic          2008   1.8 subcompact        7.85        9.00    9.37
 4 honda        civic          2008   1.8 subcompact        7.95        9.00    9.37
 5 honda        civic          2008   1.8 subcompact        8.00        9.00    9.37
 6 nissan       altima         2008   2.5 midsize           8.70        9.28    9.37
 7 nissan       altima         2008   2.5 midsize           8.84        9.28    9.37
 8 toyota       camry solara   2008   2.4 compact           9.03        9.88    9.37
 9 chevrolet    malibu         2008   2.4 midsize           9.19        9.13    9.37
10 hyundai      sonata         2008   2.4 midsize           9.22        9.13    9.37
# … with 107 more rows, and abbreviated variable name ¹pred_tree
# ℹ Use `print(n = ...)` to see more rows
```

Pivoting with tidyr

Let's plot these predictions with the following mapped aesthetics:

x: engine size

y: predicted efficiency

colour: model type

???

Pivoting with tidyr

ggplot wants tidy data. Let's pivot our data so that model type becomes a column.

```
50  l100k_long ←
51    l100k ▷
52    pivot_longer(
53      cols = starts_with("pred"),
54      names_to = "model_type",
55      names_prefix = "pred_",
56      values_to = "prediction"
57    )
```

Pivoting with tidyr

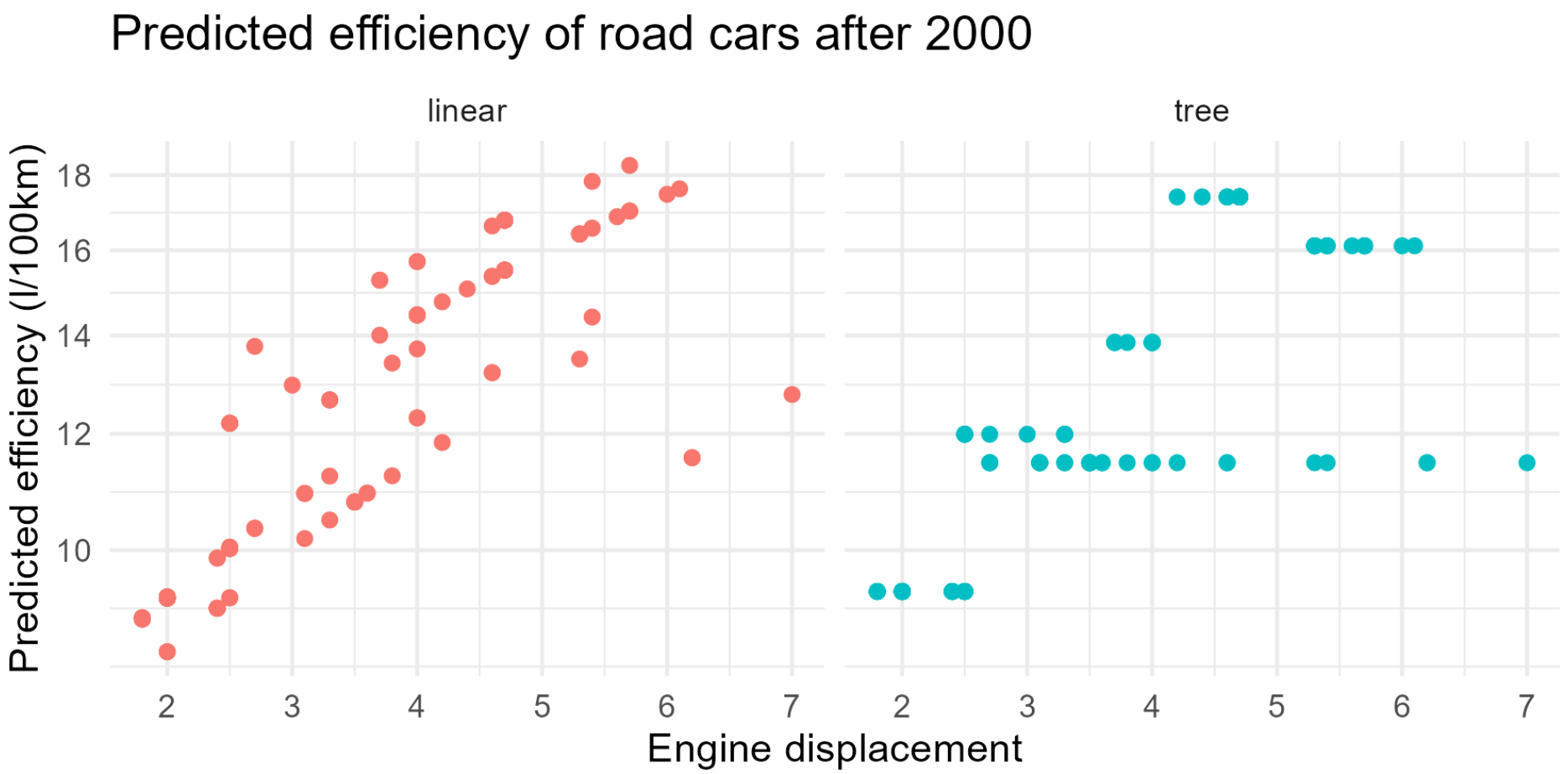
```
# A tibble: 234 × 8
   manufacturer model    year displ class      l_per_100km model_type prediction
   <chr>        <chr>   <int> <dbl> <chr>            <dbl> <chr>            <dbl>
 1 toyota       corolla  2008   1.8 compact           7.34 linear            8.98
 2 toyota       corolla  2008   1.8 compact           7.34 tree              9.37
 3 toyota       corolla  2008   1.8 compact           7.83 linear            8.98
 4 toyota       corolla  2008   1.8 compact           7.83 tree              9.37
 5 honda        civic    2008   1.8 subcompact        7.85 linear            9.00
 6 honda        civic    2008   1.8 subcompact        7.85 tree              9.37
 7 honda        civic    2008   1.8 subcompact        7.95 linear            9.00
 8 honda        civic    2008   1.8 subcompact        7.95 tree              9.37
 9 honda        civic    2008   1.8 subcompact        8.00 linear            9.00
10 honda        civic    2008   1.8 subcompact        8.00 tree              9.37
# … with 224 more rows
# ℹ Use `print(n = ...)` to see more rows
```

Pivoting with tidyr

Let's plot these predictions!

```
59  l100k_long ▷
60    ggplot(aes(x = displ, y = prediction, colour = model_type)) +
61    geom_point() +
62    scale_y_log10() +
63    labs(
64      x       = "Engine displacement",
65      y       = "Predicted efficiency (l/100km)",
66      colour = "Model type",
67      title   = "Predicted efficiency of road cars after 2000"
68    ) +
69    facet_wrap(vars(model_type)) +
70    theme_minimal() +
71    theme(legend.position = "none")
```

Pivoting with tidyr



Practical: dplyr, ggplot, tidyr

Work in your groups!
Take a break from 10:30 to 10:40

Break

Break