

# **Data Wrangling and Data Analysis**

## **Exploratory Data Analysis**

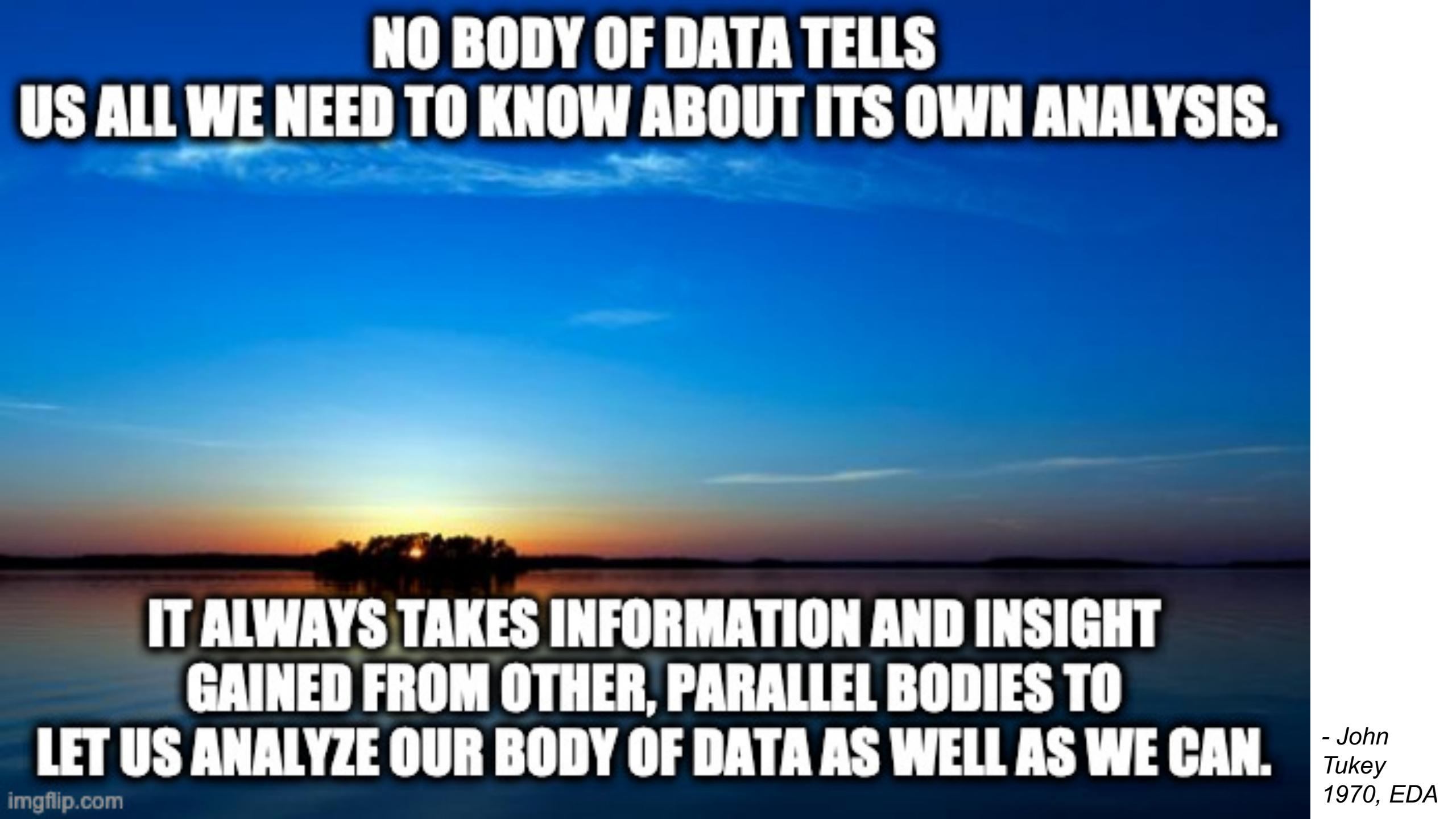
**Daniel L. Oberski & Erik-Jan van Kesteren**

Department of Methodology & Statistics

Utrecht University



Utrecht University

The background of the image is a photograph of a sunset or sunrise over a body of water. The sky is a gradient from deep blue at the top to warm orange and yellow near the horizon. Silhouetted trees are visible along the shore in the foreground.

**NO BODY OF DATA TELLS  
US ALL WE NEED TO KNOW ABOUT ITS OWN ANALYSIS.**

**IT ALWAYS TAKES INFORMATION AND INSIGHT  
GAINED FROM OTHER, PARALLEL BODIES TO  
LET US ANALYZE OUR BODY OF DATA AS WELL AS WE CAN.**

*- John  
Tukey  
1970, EDA*

# This week

- Data visualization principles
- The grammar of graphics
- **Exploratory data analysis (EDA)**
  
- Goal of the week: know how to create and improve data visualizations, and know how to use them for exploration

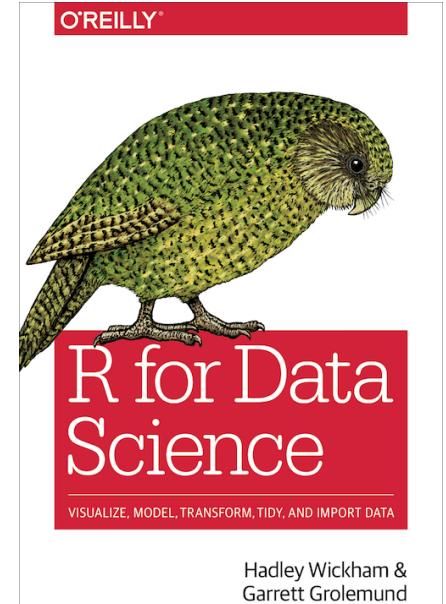


# Reading materials for this week

- **R for Data Science (R4DS)**  
*(Wickham & Grolemund 2017(ish) )*
- <https://r4ds.had.co.nz>
- Chapters 3,5, 7

*Optional:*

- **Exploratory Data Analysis with R** (*Peng 2020*)
- <https://bookdown.org/rdpeng/exdata/>



Exploratory Data  
Analysis with R



Roger D. Peng



# Assignments this week

- Monday: Tidy data visualization with dplyr and ggplot
- Tuesday: Exploratory data analysis
- Thursday: either (a) resit for the test, or (b) assignment on advanced data visualization



# Exploratory vs. confirmatory



# Exploratory vs confirmatory

- Exploratory analysis
  - Generate new insights
  - Create new hypotheses
  - Analysis may depend on data
- Confirmatory analysis
  - Test theory
  - A priori hypothesis
  - Analysis predefined
  - Example: registered report



# Exploratory vs confirmatory

*You work at Google and your colleague has implemented a new search algorithm for German searches. Your task is to check whether this algorithm performs better than the existing one.*

- **Confirmatory.**
- Theory testing, hypothesis: new works better than old
- Analysis can be defined in advance: which outcome variables, how to sample from the population, which method?
- Full analysis script could be written before the data even exists



# Exploratory vs confirmatory

*You have obtained access to your company's customer relations database. Your task is to find data-driven ways in which your company can improve customer retention.*

- **Exploratory.**
- Generate new insights - at which touchpoint do customers drop out?
- Create new hypotheses - there may be two types of customers who drop out
- Analysis cannot be defined in advance, task is to explore associations between features



# Exploratory vs confirmatory

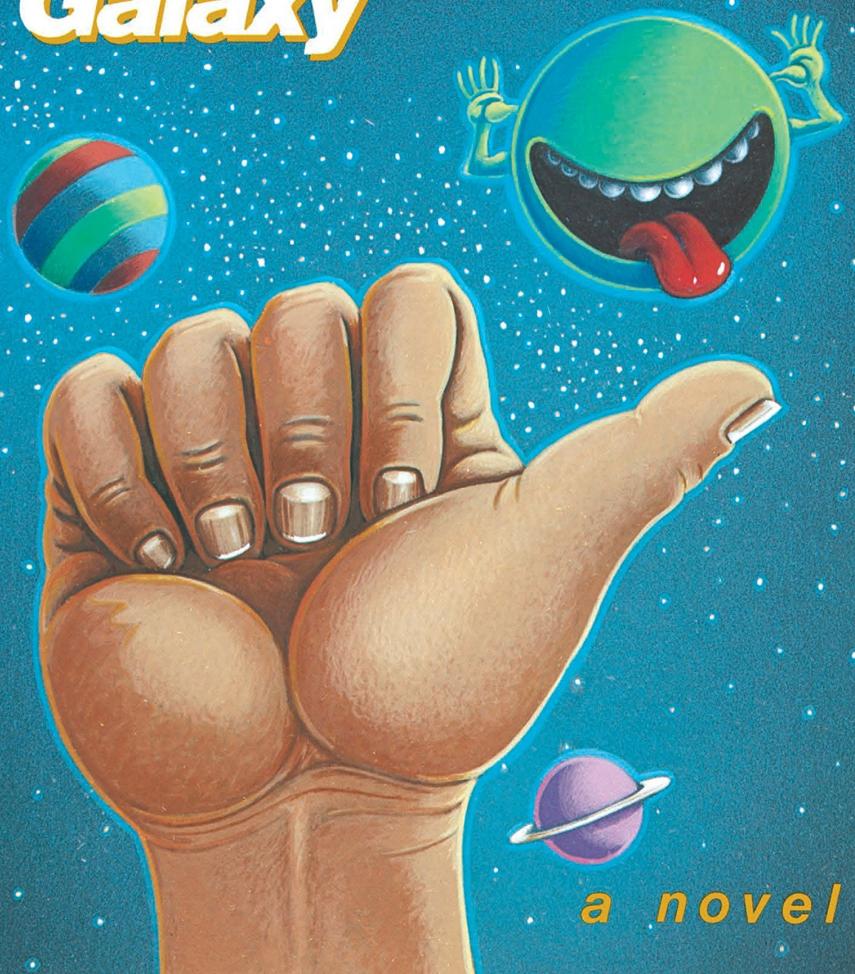
- Both are necessary and complementary
- Typical scientific questions: exploratory -> confirmatory
- Then build on the tested theories to generate new insights:  
confirmatory -> exploratory?
- This lecture: **exploratory analysis**





DOUGLAS ADAMS

*The Hitchhiker's  
Guide to the  
Galaxy*



# Some good advice on data exploration



# Peng's EDA checklist

1. Formulate your question
2. Read in your data
3. Check the packaging, (run `str()`)
4. Look at the top and the bottom of your data
5. Check your “n”s
6. Validate with at least one external data source
7. Try the easy solution first
8. Challenge your solution
9. Follow up

Exploratory Data  
Analysis with R



Roger D. Peng



# Tukey's approach to EDA

- Look at center and spread
- Find comparisons
- “Straightening and flattening”:
- Use logarithms and other transforms
- Use models and residuals



hinges.

If we have 9 values in all, the 5th from either end will be the median, since  $\frac{1}{2}(1 + 9) = 5$ . Since  $\frac{1}{2}(1 + 5) = 3$ , the third from either end will be a hinge. If we have 13 values, the 7th will be the median--and the 4th from each end a hinge. In folded form, a particular set of 13 values appears as follows:

|      |     |     |     |     |
|------|-----|-----|-----|-----|
| -3.2 |     | 1.5 |     | 9.8 |
| -1.7 |     | 1.2 | 1.8 | 6.4 |
| -0.4 | 0.3 |     | 2.4 | 4.3 |
|      | 0.1 |     | 3.0 |     |

The five summary numbers are, in order, -3.2, 0.1, 1.5, 3.0, and 9.8, one at each folding point.

We usually symbolize the 5 numbers (extremes, hinges, median) that make up a

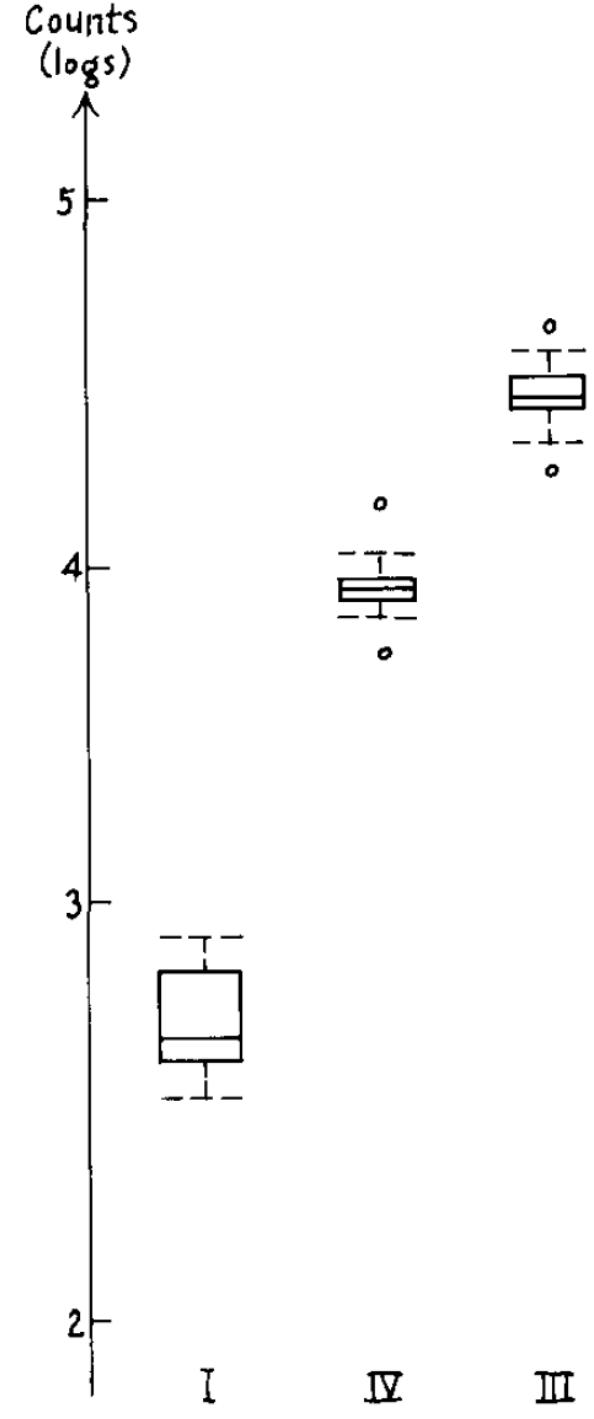
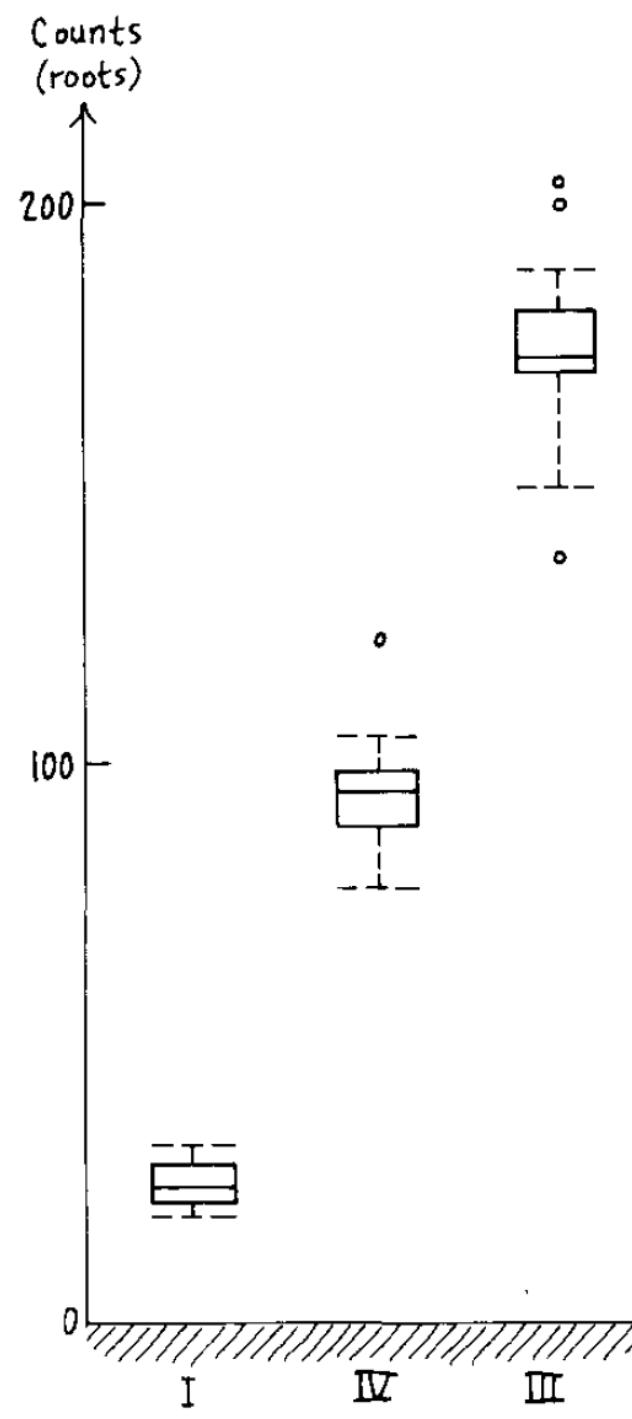
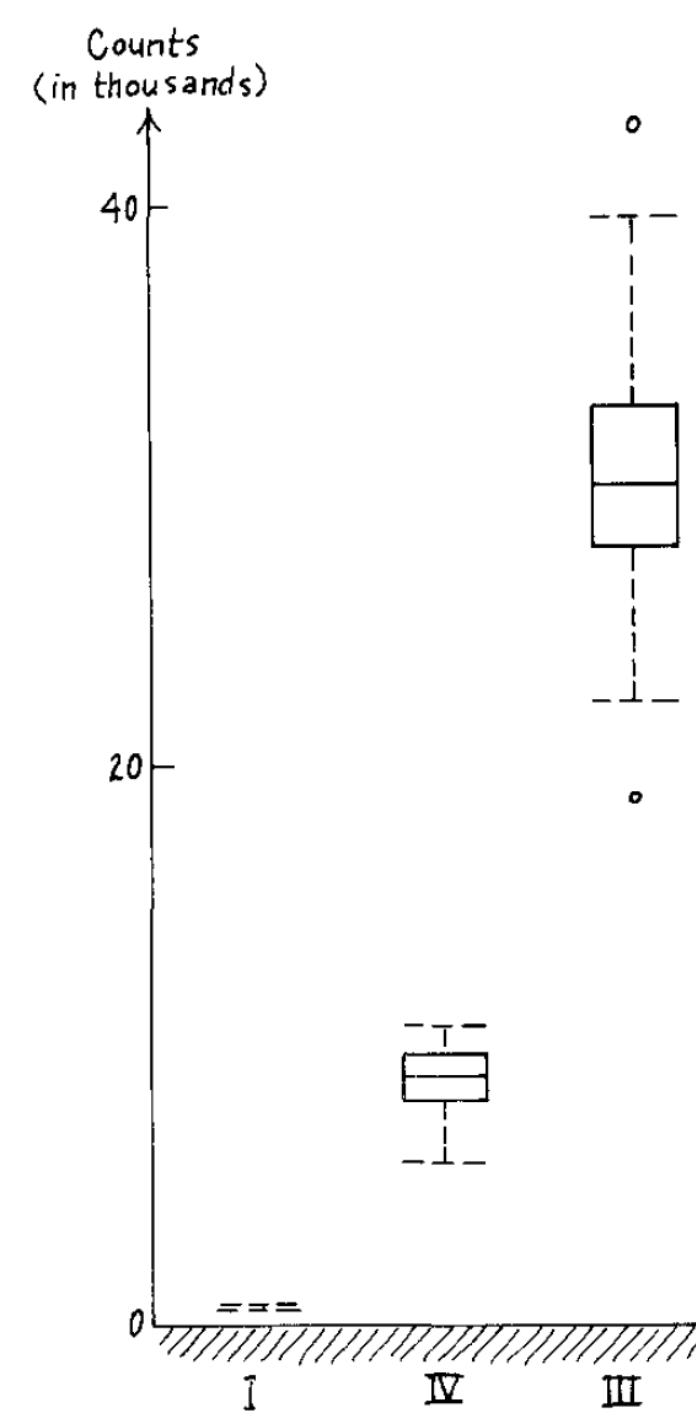
### 5-number summary

by a simple summary scheme like this:

|     |
|-----|
| #13 |
| M7  |
| H4  |
| 1   |

|      |     |
|------|-----|
|      | 1.5 |
| 0.1  | 3.0 |
| -3.2 | 9.8 |

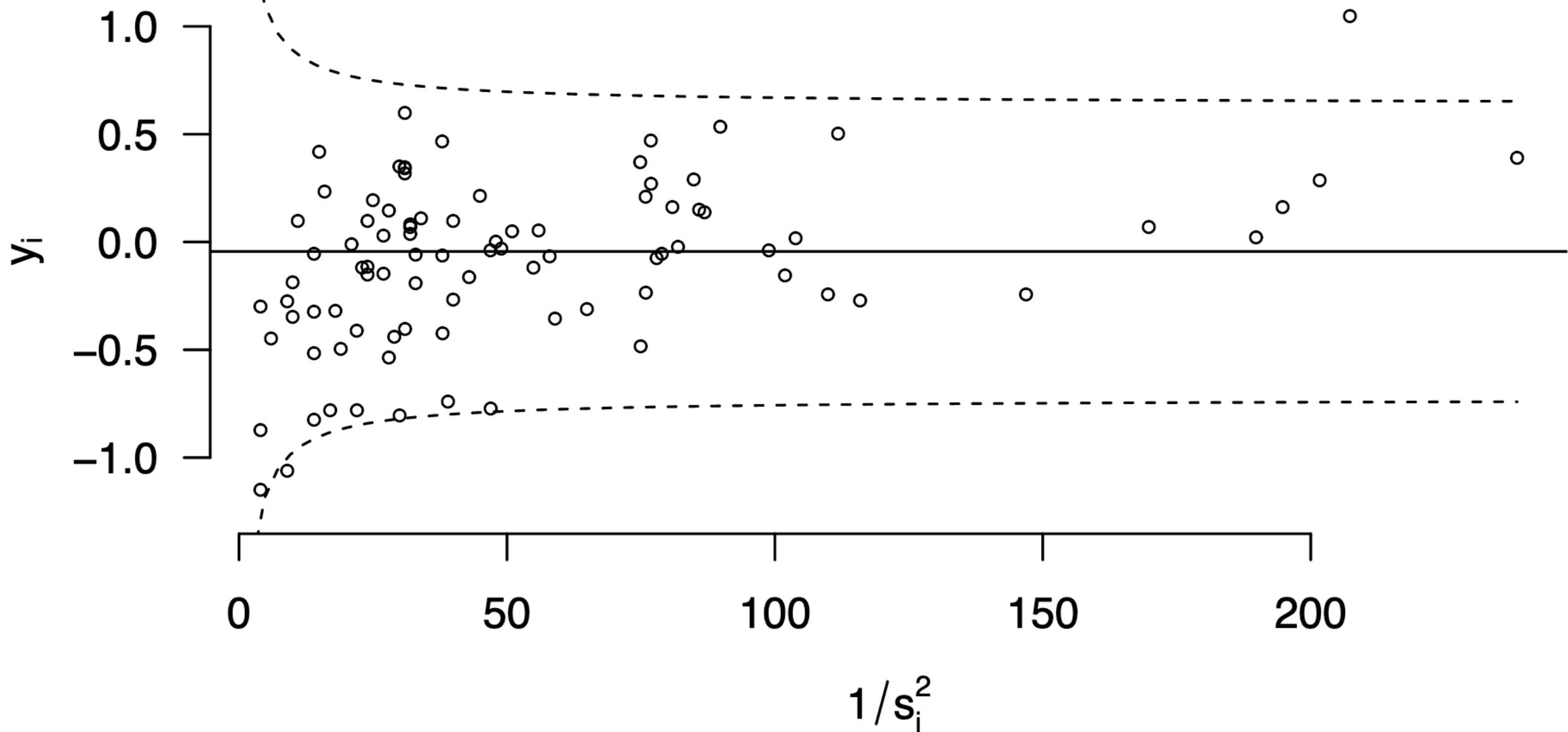
(Tukey 1970)



(Tukey 1970)

# Adverse events in Dutch hospitals

(Lenz & Oberski)



# Example exploration of current corona cases in the Netherlands





## Dataset: COVID-19 case counts in The Netherlands

---

CoronaWatchNL collects numbers on COVID-19 disease count cases in **The Netherlands**. The numbers are collected from various sources on a daily base, like [RIVM \(National Institute for Public Health and the Environment\)](#), [LCPS \(Landelijk Coördinatiecentrum Patiënten Spreiding\)](#), [NICE \(Nationale Intensive Care Evaluatie\)](#), and the [National Corona Dashboard](#). This project standardizes, and publishes data and makes it **Findable, Accessible, Interoperable, and Reusable (FAIR)**. We aim to collect a complete time series and prepare a dataset for reproducible analysis and academic use.

# Read data, check packaging

```
url_icu <-  
"https://raw.githubusercontent.com/J535D165/CoronaWatchNL/master/data-  
ic/data-nice/NICE_IC_long_latest.csv"  
  
icu <- read_csv(url_icu)  
  
> dim(icu)  
[1] 1935     3
```



# Look at top and bottom

```
> icu %>% tail  
# A tibble: 6 x 3  
  Datum      Type          Aantal  
  <date>    <chr>        <dbl>  
1 2020-09-28 Toename ontslag (ziekenhuis)     0  
2 2020-09-28 Totaal opnamen (IC)            144  
3 2020-09-28 Toename opnamen (IC)            7  
4 2020-09-28 Totaal ontslag (IC)             56  
5 2020-09-28 Cumulatief opnamen (IC)       3261  
6 2020-09-28 Toename ontslag (overleden)     0
```



# Look at top and bottom

```
> icu %>% head  
# A tibble: 6 x 3  
  Datum      Type          Aantal  
  <date>    <chr>        <dbl>  
1 2020-02-27 Totaal ingezette IC's     5  
2 2020-02-27 Cumulatief opnamen (IC)  7  
3 2020-02-27 Totaal ontslag (IC)       0  
4 2020-02-27 Cumulatief ontslag (ziekenhuis) 0  
5 2020-02-27 Cumulatief ontslag (overleden) 0  
6 2020-02-27 Toename ontslag (overleden)  0
```



# Validate with external source



## CoronaTracker

[Home](#)[Travel Alert](#)[What is COVID-19](#)

### Netherlands Overview

**114,540**

Confirmed

+2,914 new cases

**3**

Recovered

Share: [f](#) [t](#)

**6,380**

Deaths

+6 new deaths

**5.6%**

OF TOTAL CASES

Fatality

### Critical Cases treated in ICU

**144**

0.1% of total cases

### Daily Cases Receiving Treatment

**0**

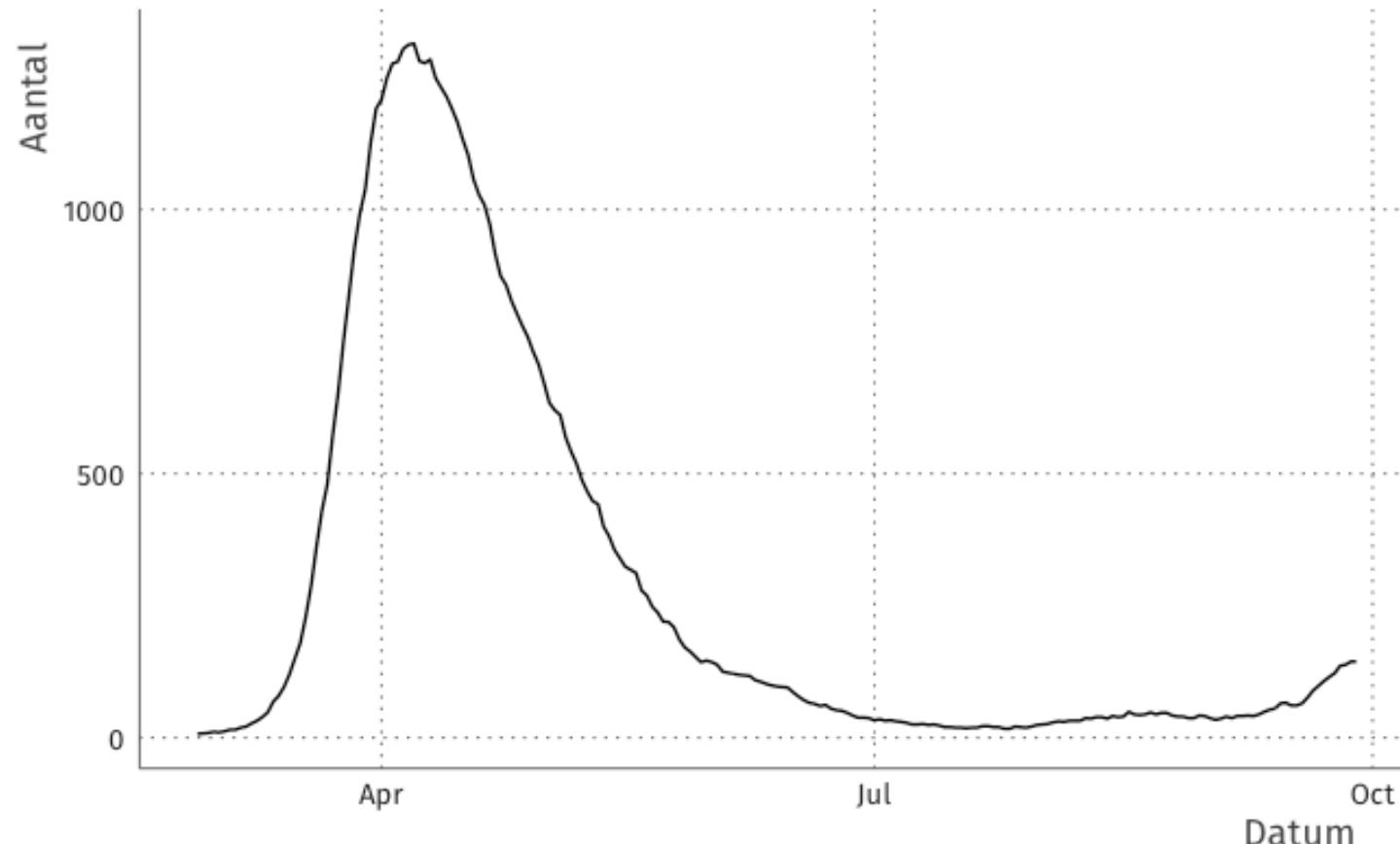
0.0% of total cases

# Check n's

```
> icu %>% group_by(Type) %>% summarize(n())  
  
# A tibble: 9 x 2  
  Type                `n()`  
  <chr>               <int>  
1 Cumulatief ontslag (overleden)    215  
2 Cumulatief ontslag (ziekenhuis)   215  
3 Cumulatief opnamen (IC)          215  
4 Toename ontslag (overleden)       215  
5 Toename ontslag (ziekenhuis)      215  
6 Toename opnamen (IC)             215  
7 Totaal ingezette IC's            215  
8 Totaal ontslag (IC)              215  
9 Totaal opnamen (IC)              215
```

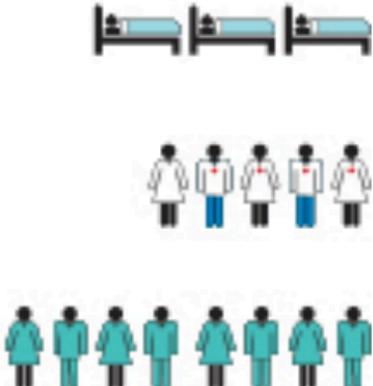


```
icu %>%
  filter(Type == "Totaal opnamen (IC)") %>%
  ggplot(aes(Datum, Aantal)) + geom_line() +
  theme_fira()
```



# Understand what you're looking at

Bezetting IC units



|  |                           | IC-units in Nederland |      |      |
|--|---------------------------|-----------------------|------|------|
|  |                           | gem.                  | min. | max. |
|  | bedden                    | 14                    | 1    | 35   |
|  | intensivisten (in fte)    | 7                     | 1    | 18   |
|  | verpleegkundigen (in fte) | 50                    | 10   | 152  |

(84 ICUs)

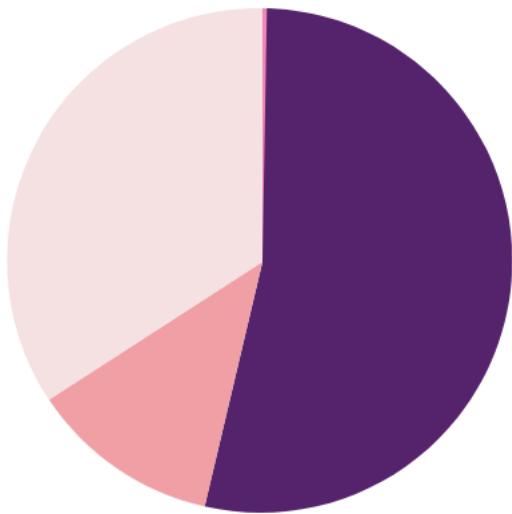
**Back-of-the-envelope:**

$$14 \times 84 \approx 1200 \text{ beds}$$

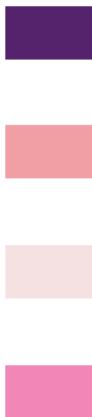
**Geometric average:**  
 $\sqrt{1 \times 35} \approx 6 \text{ beds}$

# Understand what you're looking at

## Aantal opnamen per opnametype

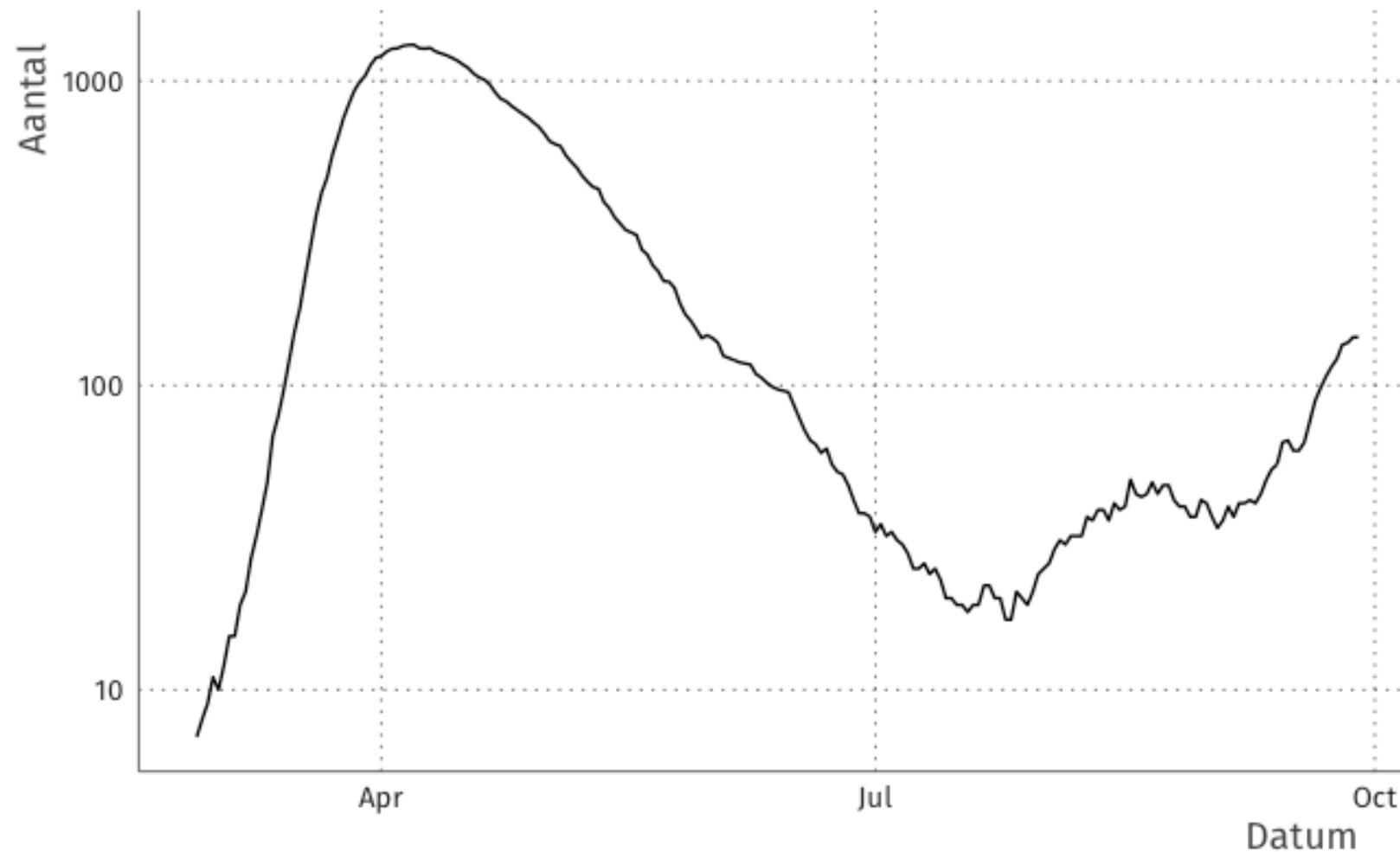


IC-units in Nederland



| IC-units in Nederland |        |      |
|-----------------------|--------|------|
| Type                  | aantal | %    |
| Alle opnamen          | 73.979 | 100  |
| Medisch               | 39.654 | 53,6 |
| Spoed chirurgie       | 8.632  | 11,7 |
| Geplande chirurgie    | 25.499 | 34,5 |
| Overige               | 194    | 0,3  |

```
icu %>%
  filter(Type == "Totaal opnamen (IC)") %>%
  ggplot(aes(Datum, Aantal)) + geom_line() +
  theme_fira() + scale_y_log10()
```



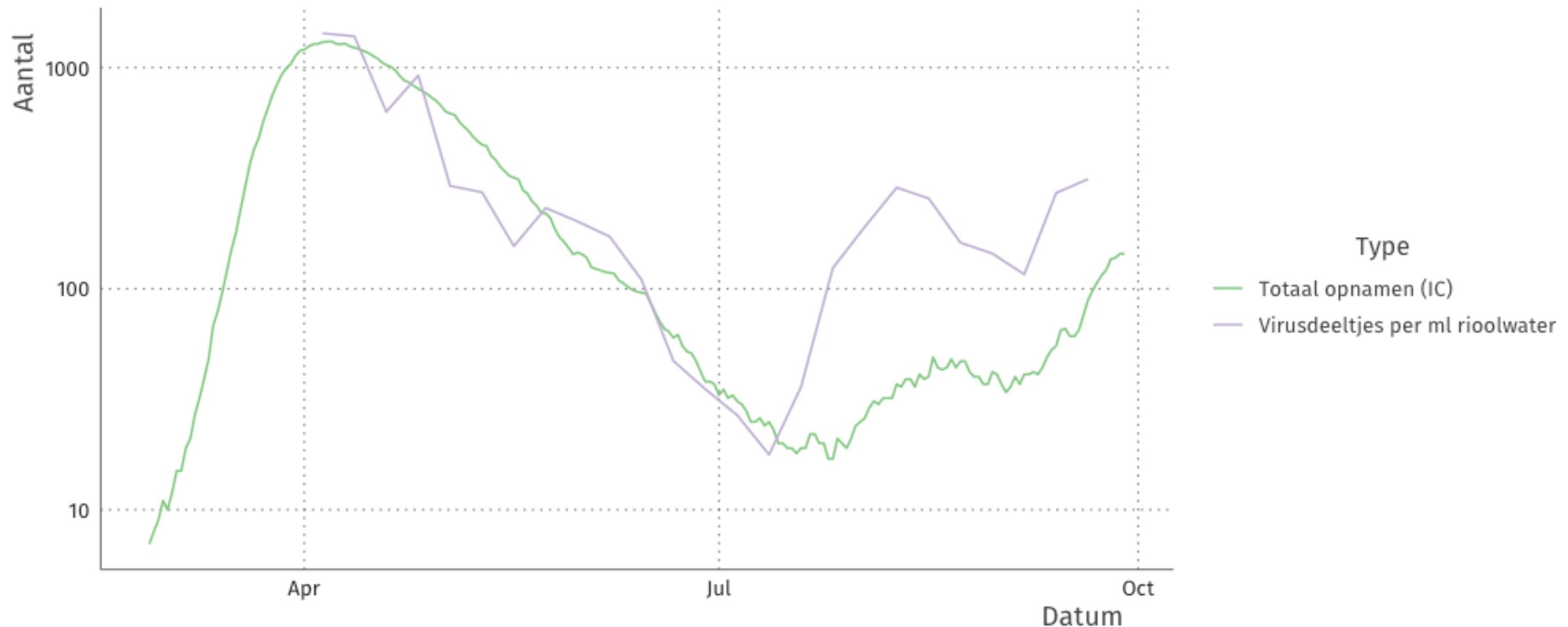
```
url_sewage <-
"https://raw.githubusercontent.com/J535D165/CoronaWatchNL/master/data-
dashboard/data-sewage/RIVM_NL_sewage_counts.csv"

sewage <- read_csv(url_sewage)

joined <- bind_rows(sewage, icu)
```



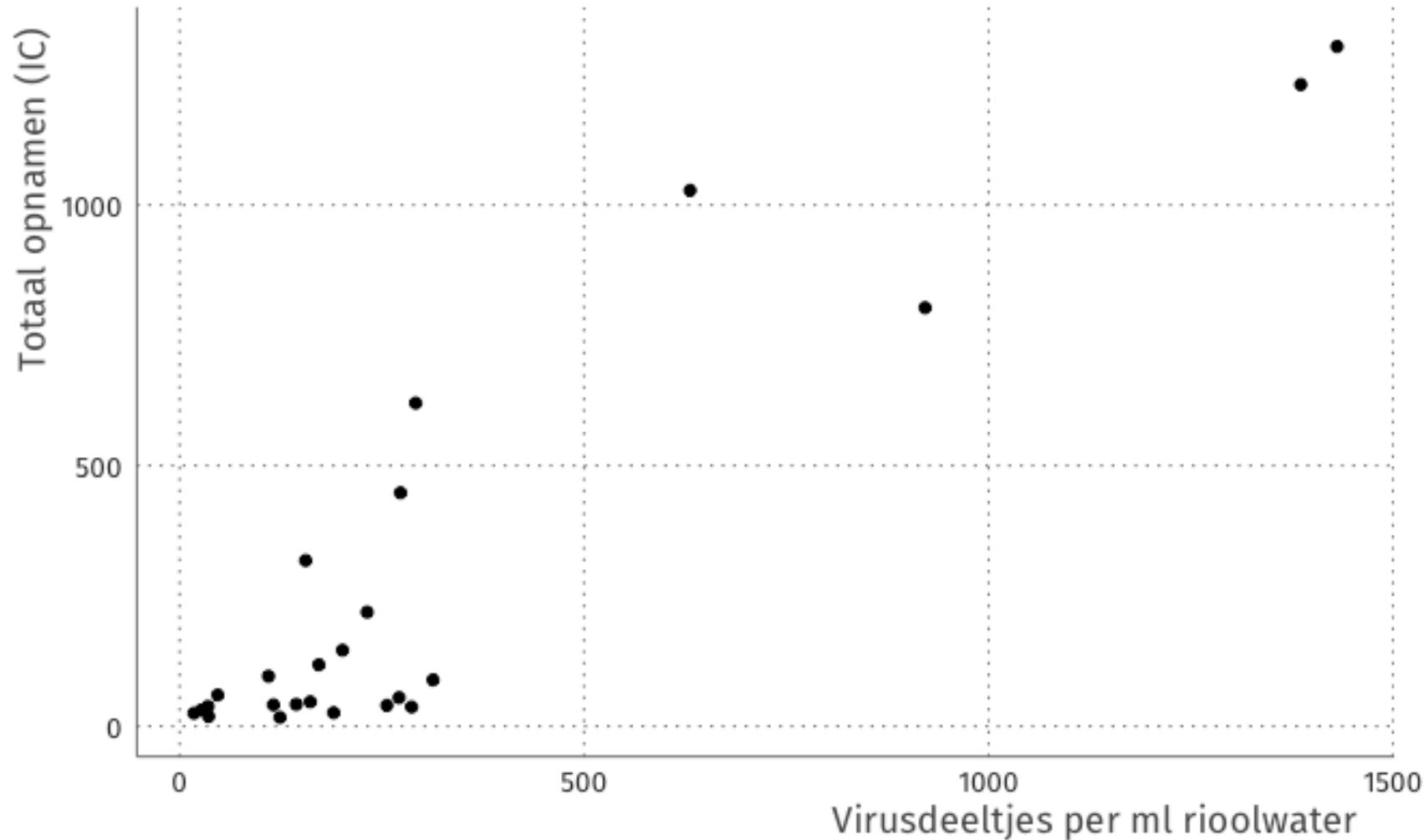
```
joined %>%
  filter(Type == "Totaal opnamen (IC)" |
         Type == "Virusdeeltjes per ml rioolwater") %>%
ggplot(aes(Datum, Aantal, color = Type)) + geom_line() +
scale_y_log10() +
theme_fira() + scale_color_brewer(type = "qual")
```



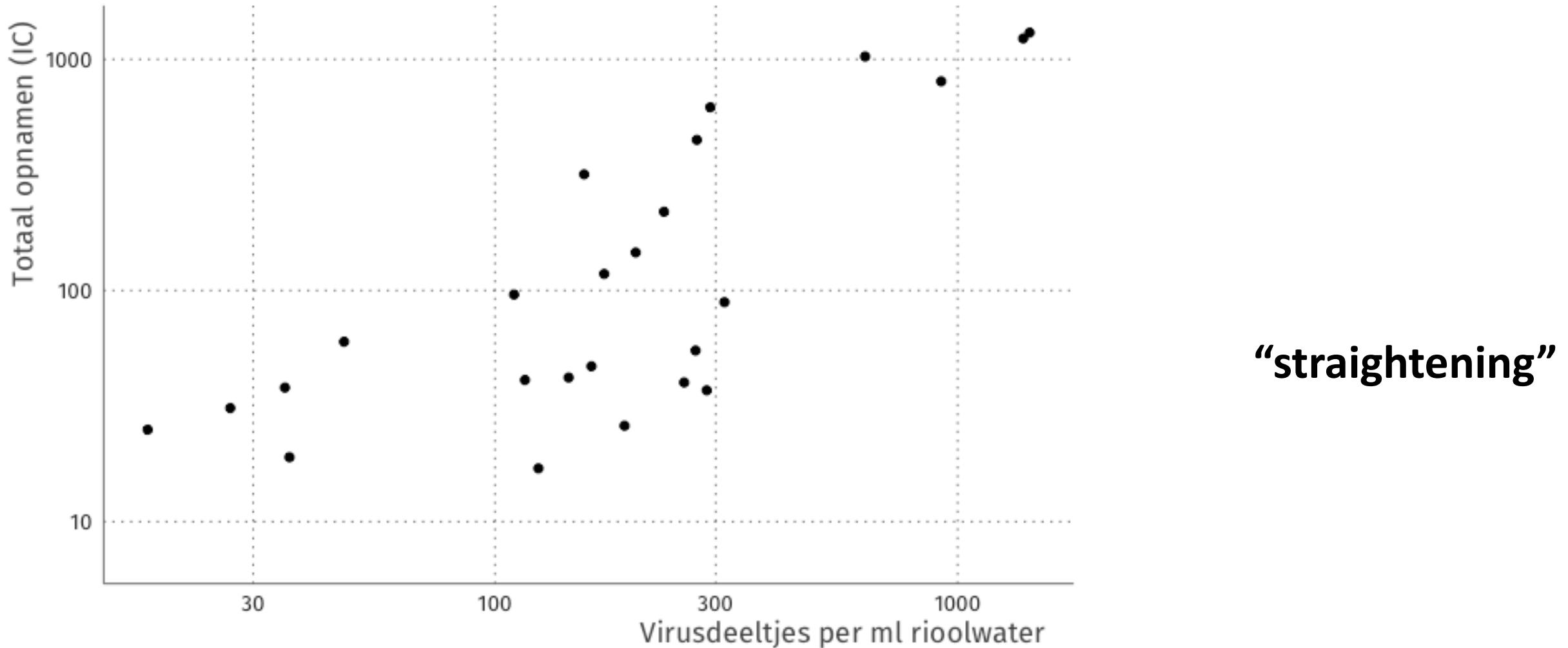
```
joined_wide <- joined %>%
  filter(Type == "Totaal opnamen (IC)" |
         Type == "Virusdeeltjes per ml rioolwater") %>%
pivot_wider(names_from = Type, values_from = Aantal)
```



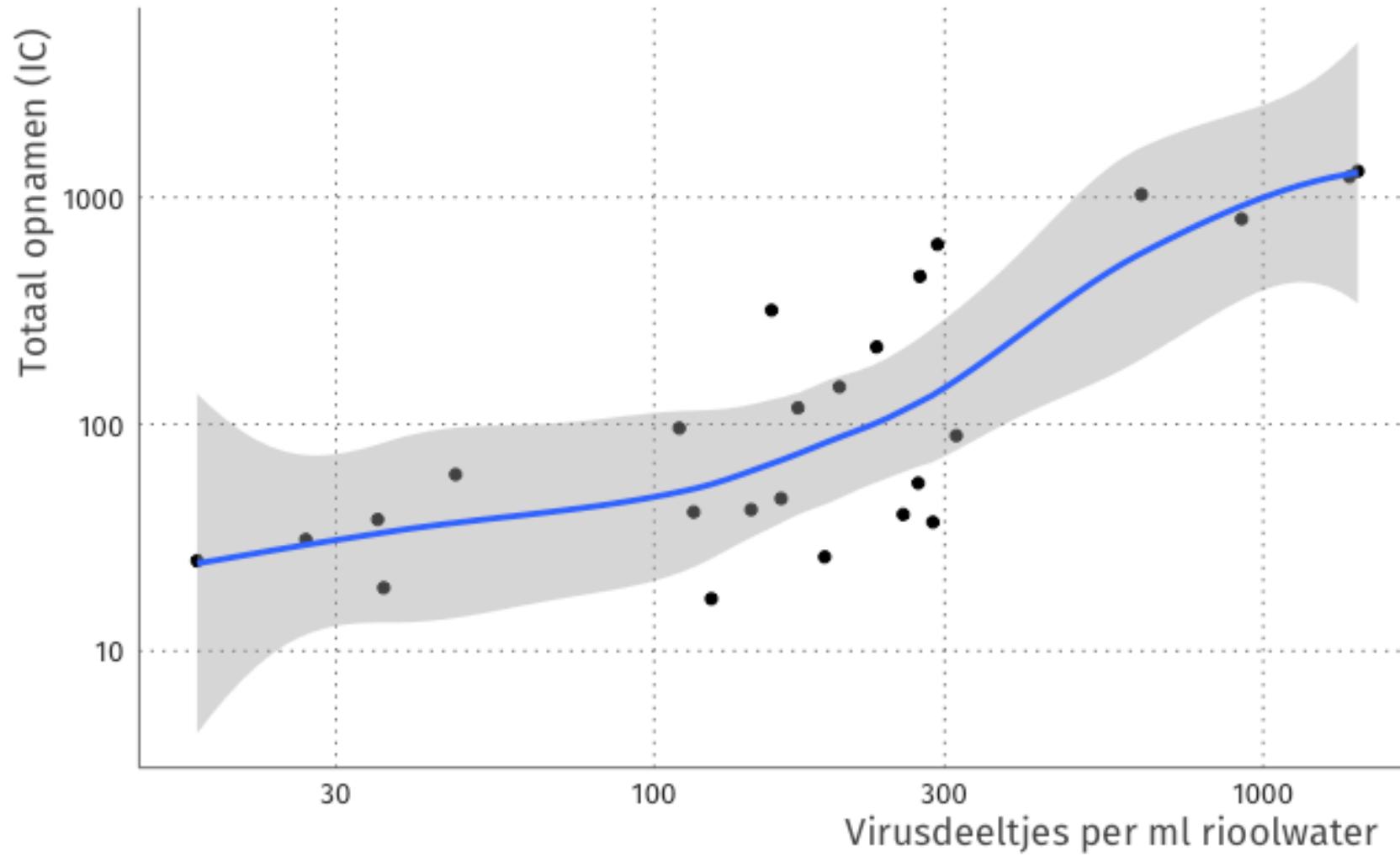
```
joined_wide %>%
ggplot(aes(`Virusdeeltjes per ml rioolwater`, `Totaal opnamen (IC)`)) +
  geom_point() + theme_fira()
```



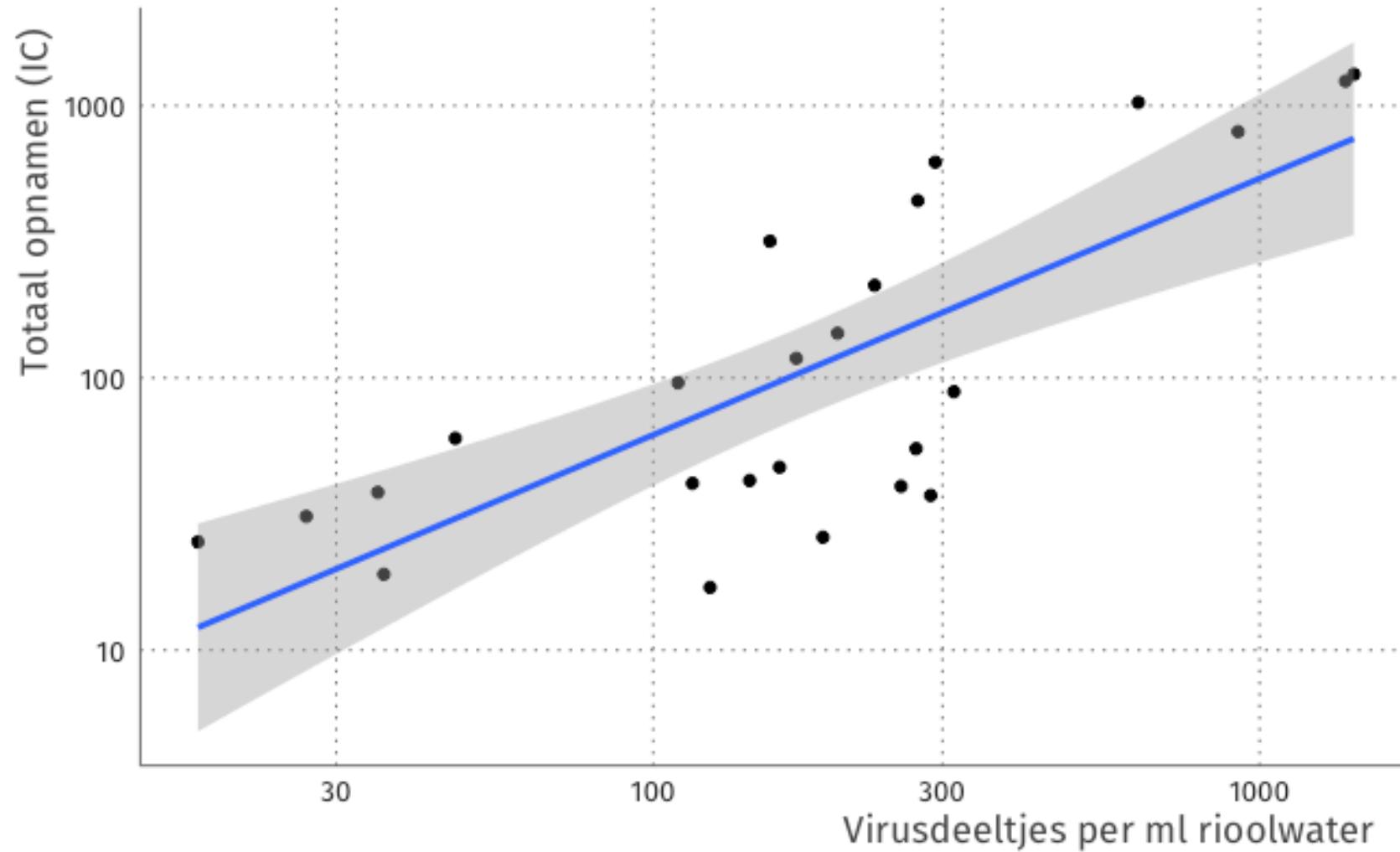
```
joined_wide %>%
ggplot(aes(`Virusdeeltjes per ml rioolwater`, `Totaal opnamen (IC)`)) +
  geom_point() + scale_y_log10() + scale_x_log10() + theme_fira()
```



```
joined_wide %>%
ggplot(aes(`Virusdeeltjes per ml rioolwater`, `Totaal opnamen (IC)`)) +
  geom_point() + geom_smooth() +
  scale y log10() + scale x log10() + theme fira()
```



```
joined_wide %>%
ggplot(aes(`Virusdeeltjes per ml rioolwater`, `Totaal opnamen (IC)`)) +
  geom_point() + geom_smooth(method = "lm") +
  scale y log10() + scale x log10() + theme fira()
```



# Looking at residuals

```
joined_wide <- joined_wide %>%
  mutate(
    log_icu = log(`Totaal opnamen (IC)` + 1),
    log_sewage = log(`Virusdeeltjes per ml rioolwater` + 1))

fit_lm <- lm(log_icu ~ log_sewage, data = joined_wide)

joined_wide <- joined_wide %>%
  mutate(log_icu_predicted = predict(fit_lm, newdata = .),
        log_icu_residual = log_icu - log_icu_predicted)
```

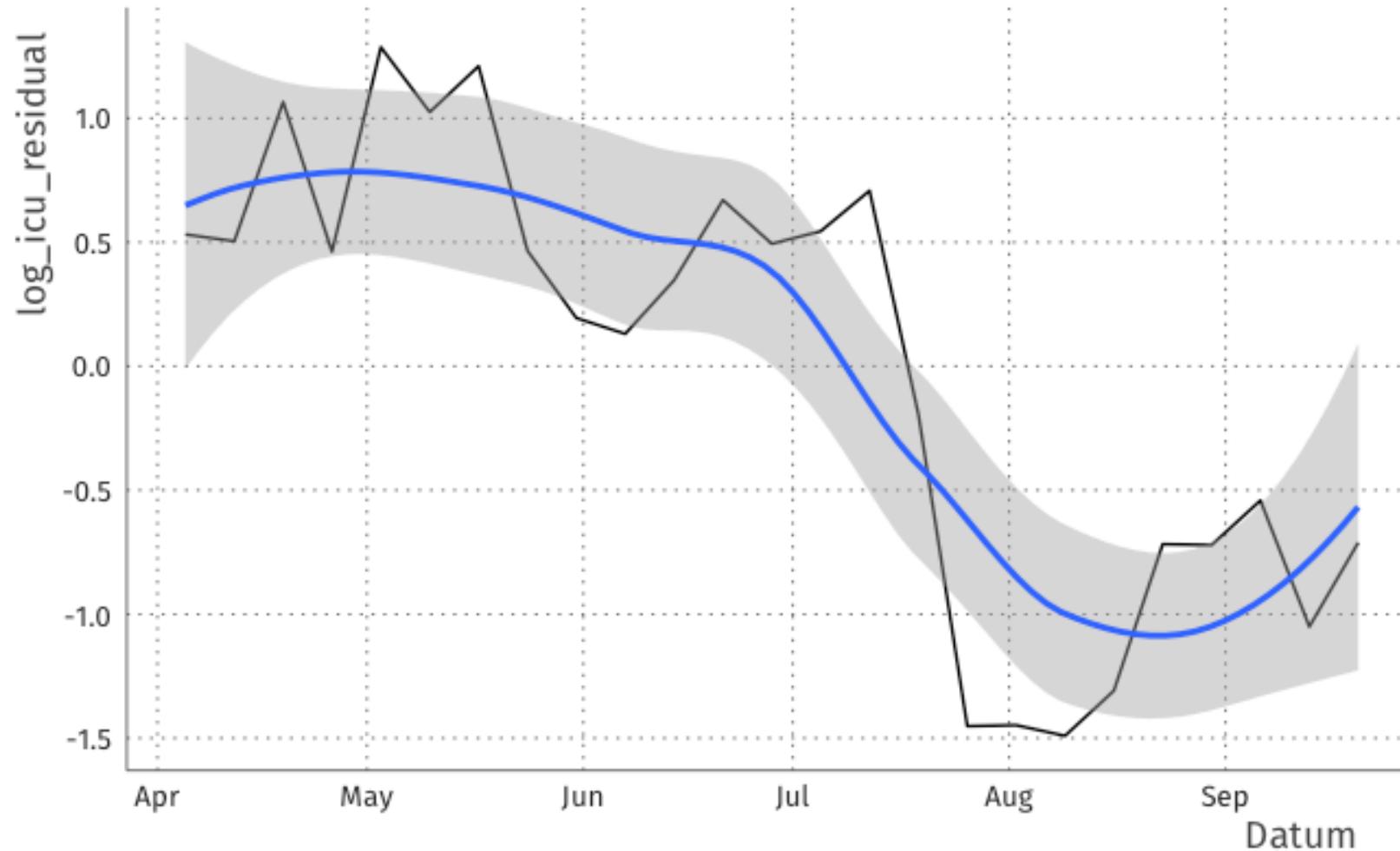


```
joined_wide %>%  
  filter(!is.na(log_icu_residual)) %>%  
  ggplot(aes(Datum, log_icu_residual)) +  
  geom_line() + theme_fira()
```



**“flattening”**

```
joined_wide %>%  
  filter(!is.na(log_icu_residual)) %>%  
  ggplot(aes(Datum, log_icu_residual)) +  
  geom_line() + geom_smooth() + theme_fira()
```



**"flattening"**

# Using some standard plots for EDA



# Visual EDA: two questions

- Variation: How are the features distributed?
  - Univariate
  - Specific: among people, among time, among flights
  - General: among the unit of measurement, among examples
  - In tabular data, generally one example/unit per row
- Association: What type of covariation occurs in the data?
  - Which features covary? When A is high, B is high?
  - Multidimensional, multivariate.



# Visual EDA: two questions

- **Variation: How are the features distributed?**
  - Univariate
  - Specific: among people, among time, among flights
  - General: among the unit of measurement, among examples
  - In tabular data, generally one example/unit per row
- Association: What type of covariation occurs in the data?
  - Which features covary? When A is high, B is high?
  - Multidimensional, multivariate.



```
library(nycflights13)
flights
#> # A tibble: 336,776 x 19
#>   year month   day dep_time sched_dep_time dep_delay arr_time sched_arr_time
#>   <int> <int> <int>     <int>           <int>     <dbl>     <int>           <int>
#> 1 2013     1     1      517             515        2       830          819
#> 2 2013     1     1      533             529        4       850          830
#> 3 2013     1     1      542             540        2       923          850
#> 4 2013     1     1      544             545       -1      1004         1022
#> 5 2013     1     1      554             600       -6       812          837
#> 6 2013     1     1      554             558       -4       740          728
#> # ... with 3.368e+05 more rows, and 11 more variables: arr_delay <dbl>,
#> #   carrier <chr>, flight <int>, tailnum <chr>, origin <chr>, dest <chr>,
#> #   air_time <dbl>, distance <dbl>, hour <dbl>, minute <dbl>, time_hour <dttm>
```



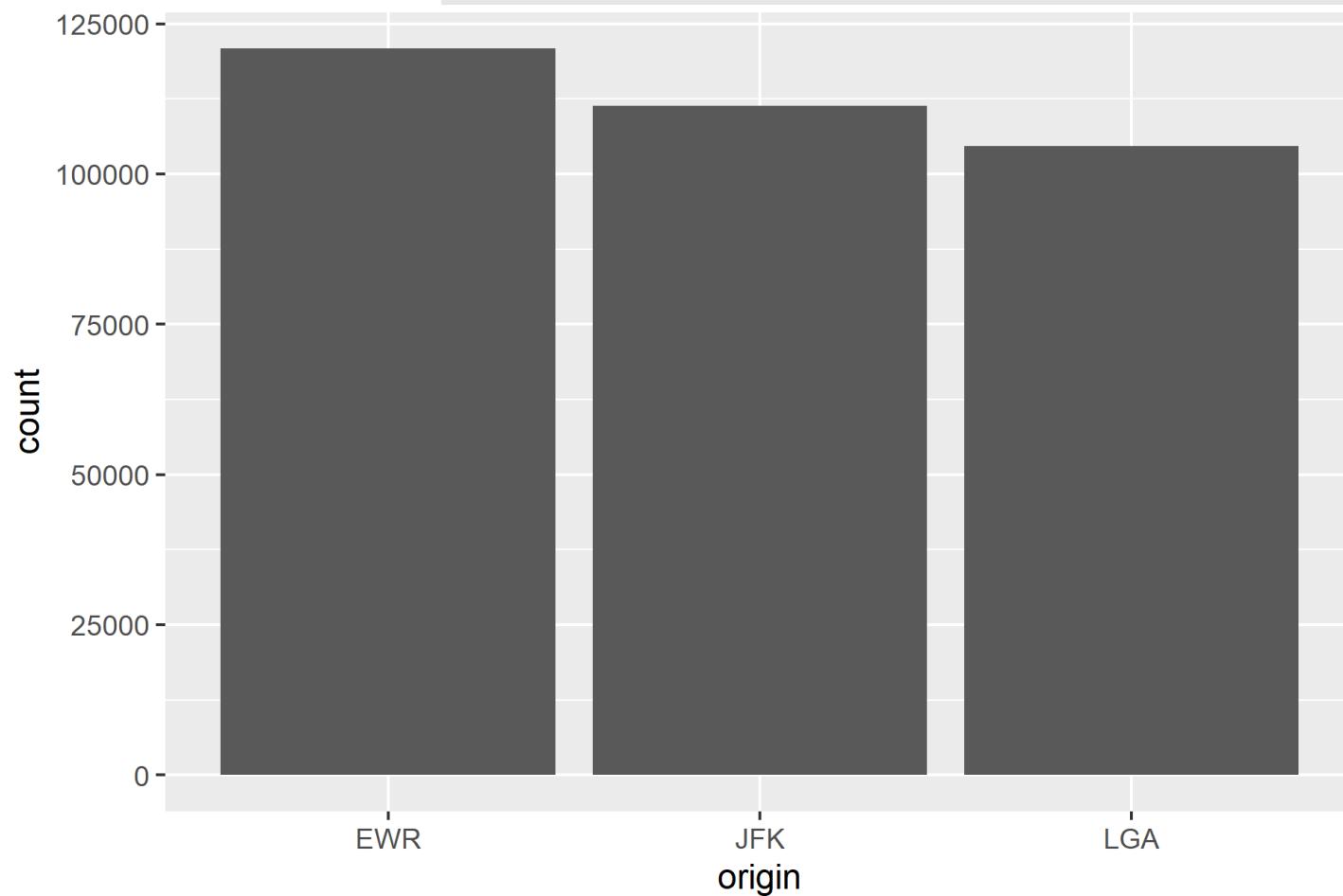
# Bar chart

- X position: feature of interest
  - Y position: count of occurrences (statistical transformation)
  - Geom: bars/rectangle
- 
- For categorical features e.g., flight origin



# Bar chart

```
ggplot(data = flights) +  
  geom_bar(mapping = aes(x = origin))
```



# Bar chart

- Fewest flights depart from LGA
- Almost 125000 flights from EWR



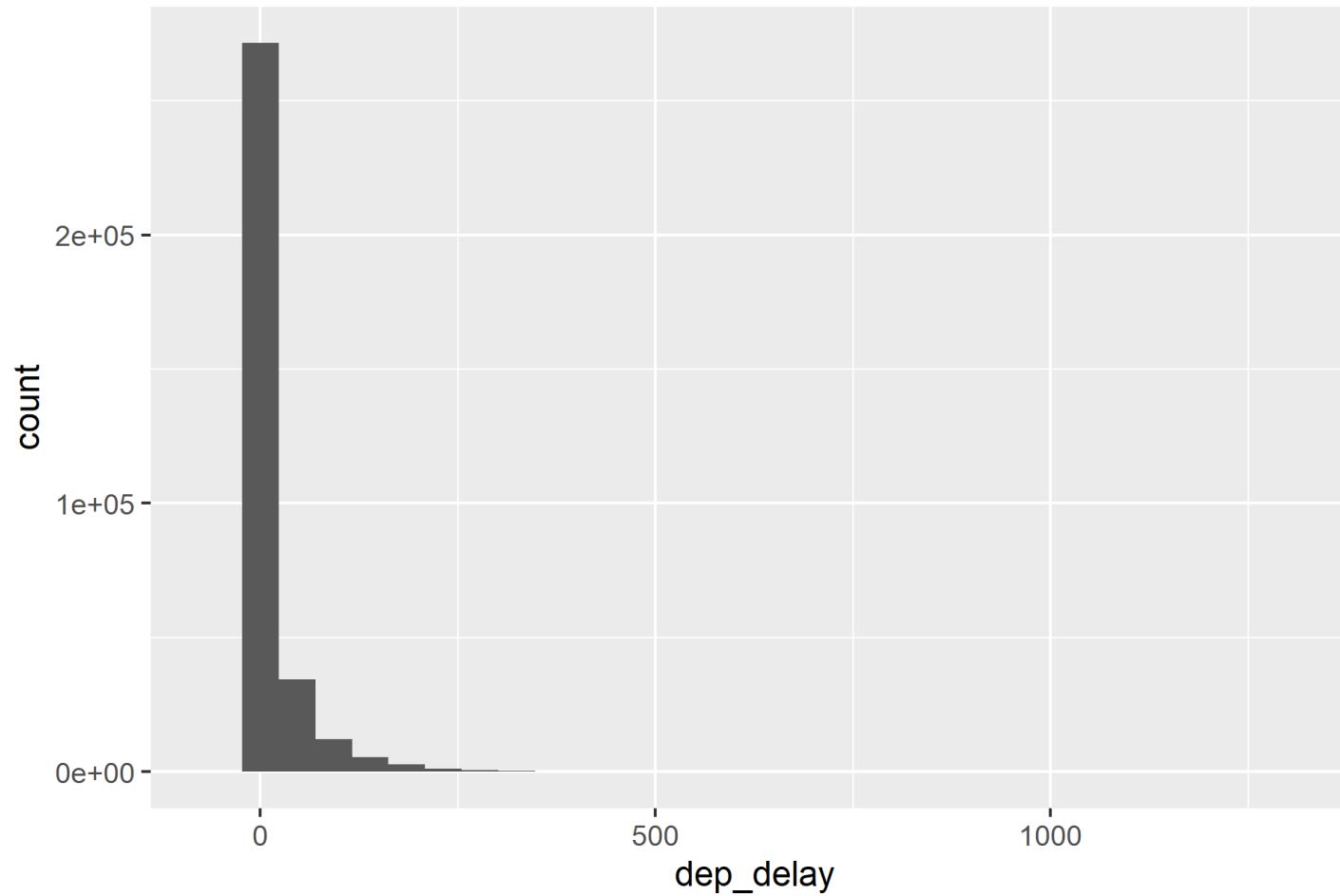
# Histogram

- X position: feature of interest
  - Y position: count of occurrences **in bins** (statistical transformation)
  - Geom: bars/rectangle
- 
- For continuous features, e.g., departure delay in minutes



# Histogram

```
ggplot(data = flights) +  
  geom_histogram(mapping = aes(x = dep_delay))
```



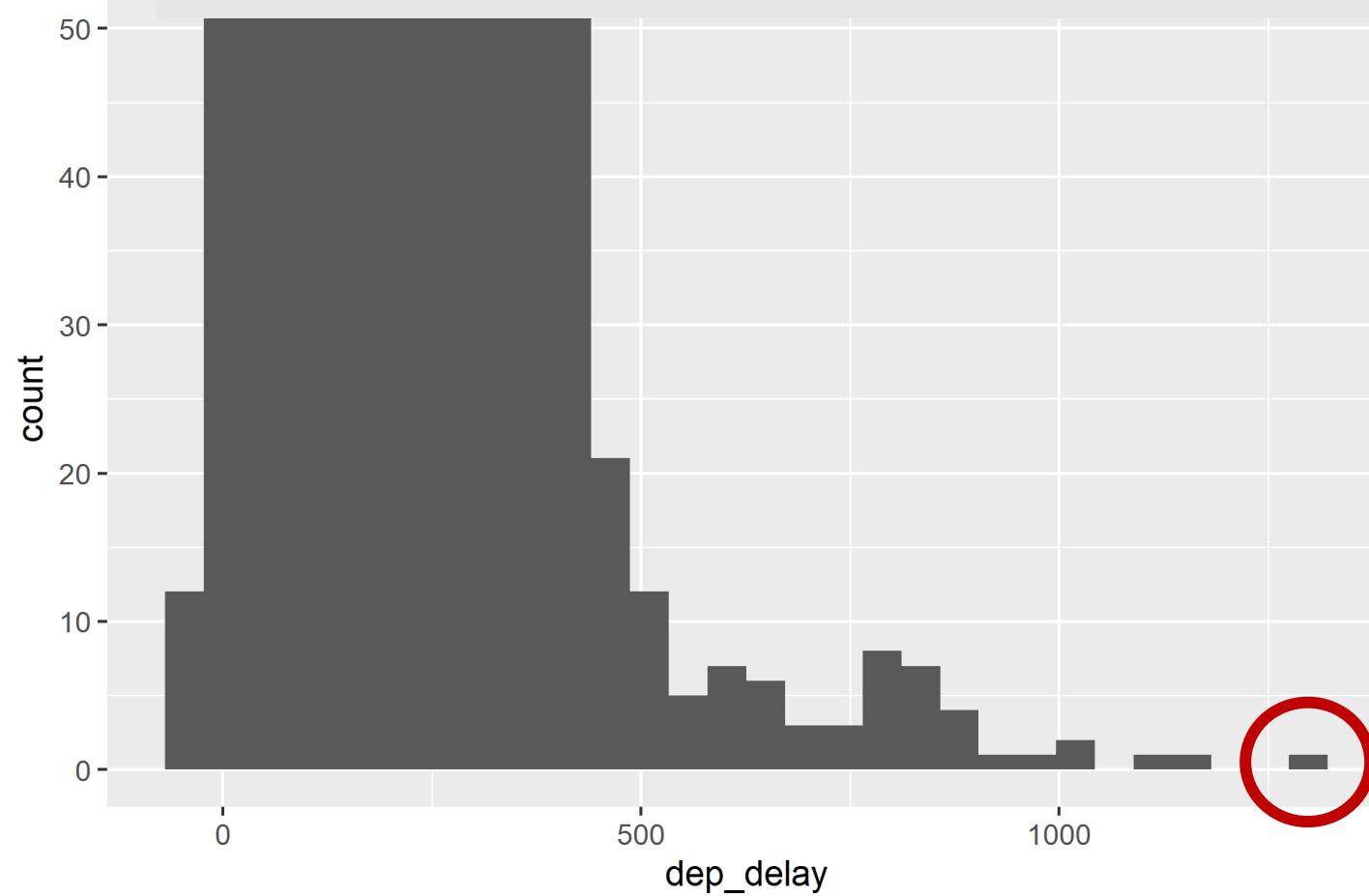
# Histogram

- New question: why the strange distribution? What happens at the tail end?



# Histogram (zoomed)

```
ggplot(data = flights) +  
  geom_histogram(mapping = aes(x = dep_delay)) +  
  coord_cartesian(ylim = c(0, 50))
```



That's a long delay!



# Histogram

- Most flights are approximately on time
- There are even some flights that depart early
- There are a few flights with more than 1000 minutes delay



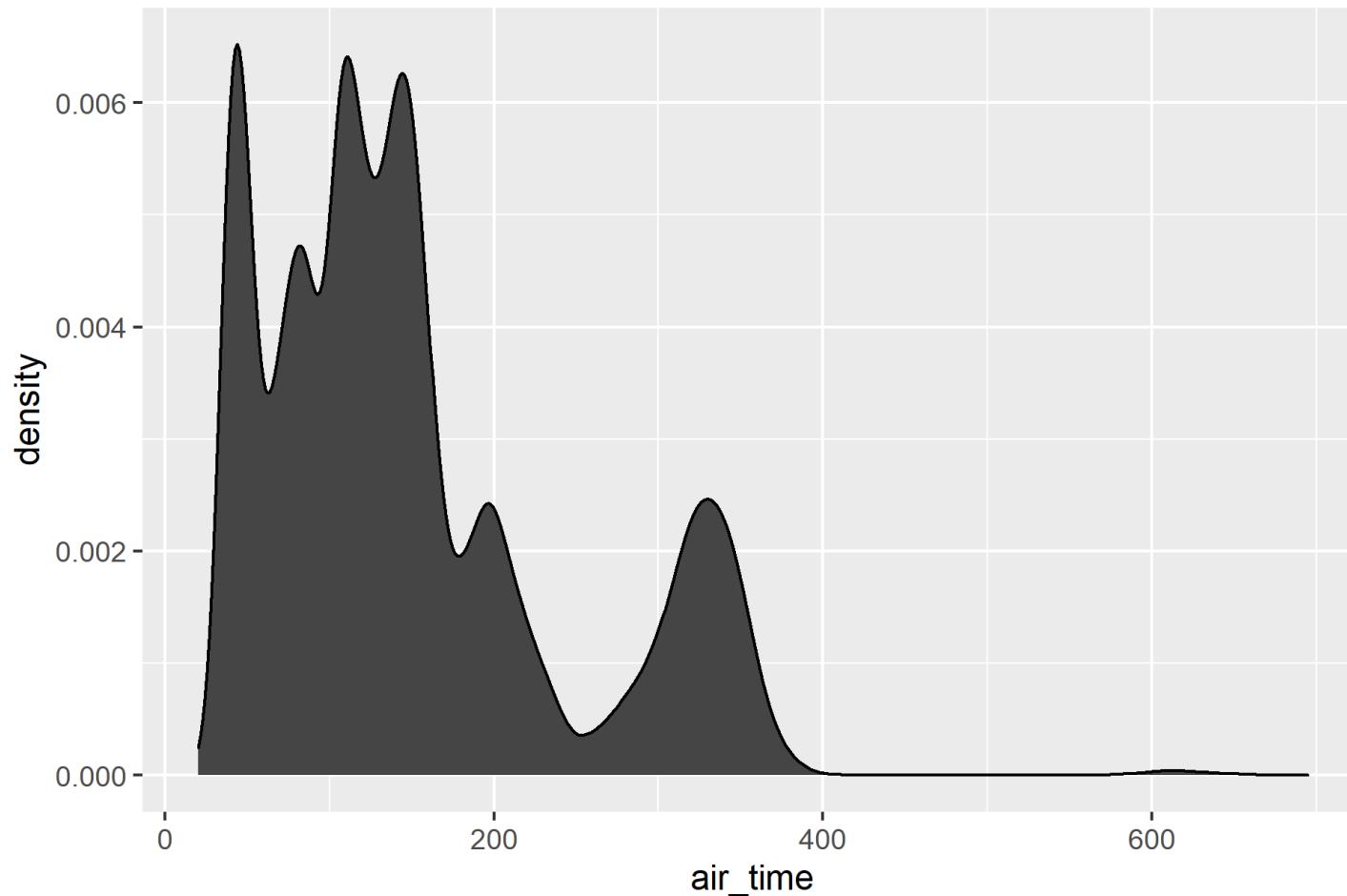
# Density

- X position: feature of interest
  - Y position: density (statistical transformation, smoothed histogram)
  - Geom: polygon / line
- 
- For continuous features, e.g., airtime in minutes



# Density

```
ggplot(data = flights) +  
  geom_density(mapping = aes(x = air_time))
```



# Density

- Most flights from NY are under 200 minutes (3.3 hours)
- Few flights are between 200 and 250 minutes
- Quite some flights are between 250 and 400 minutes
- Some flights are over 600 minutes (weird bump? Remember!)
- Air times are not normally distributed!



# Visual EDA: two questions

- Variation: How are the features distributed?
  - Univariate
  - Specific: among people, among time, among flights
  - General: among the unit of measurement, among examples
  - In tabular data, generally one example/unit per row
- **Association: What type of covariation occurs in the data?**
  - Which features covary? When A is high, B is high?
  - Multidimensional, multivariate.



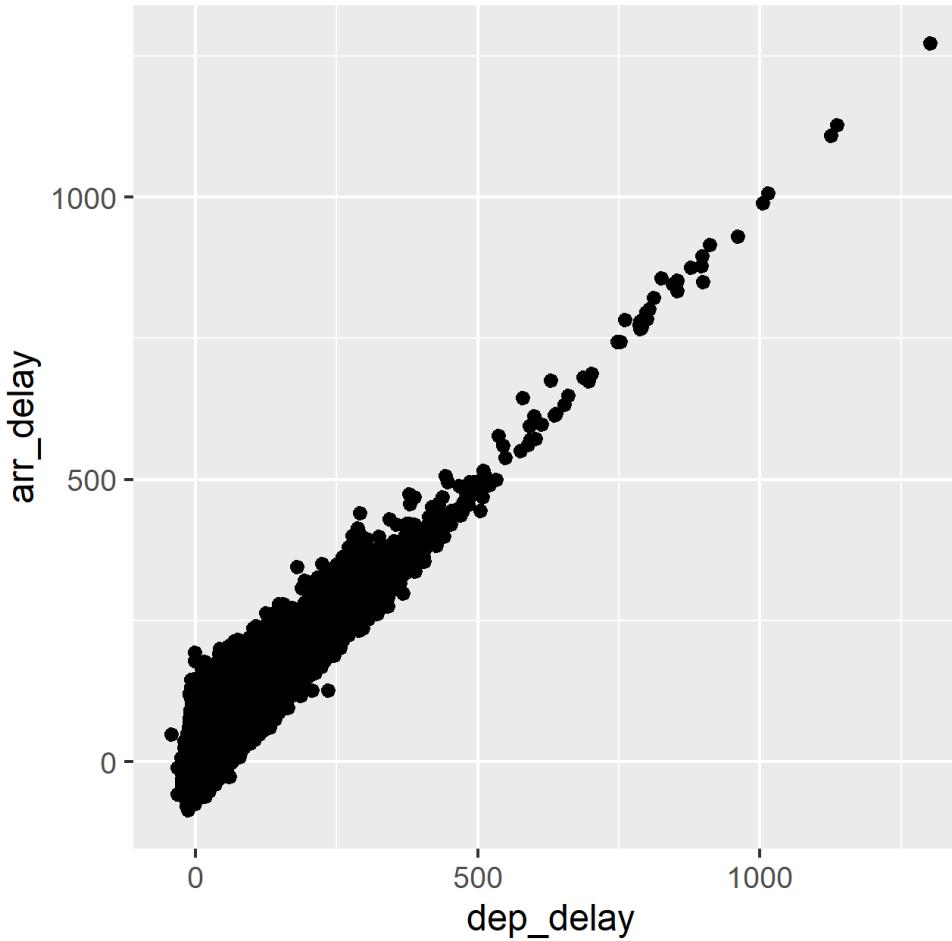
# Scatter plot

- X position: feature A of interest (in regression, generally predictor)
  - Y position: feature B of interest (in regression, outcome)
  - Geom: points/dots
- 
- For continuous features, e.g., departure delay in minutes and arrival delay in minutes



# Scatter plot

```
ggplot(data = flights) +  
  geom_point(aes(x = dep_delay, y = arr_delay))
```



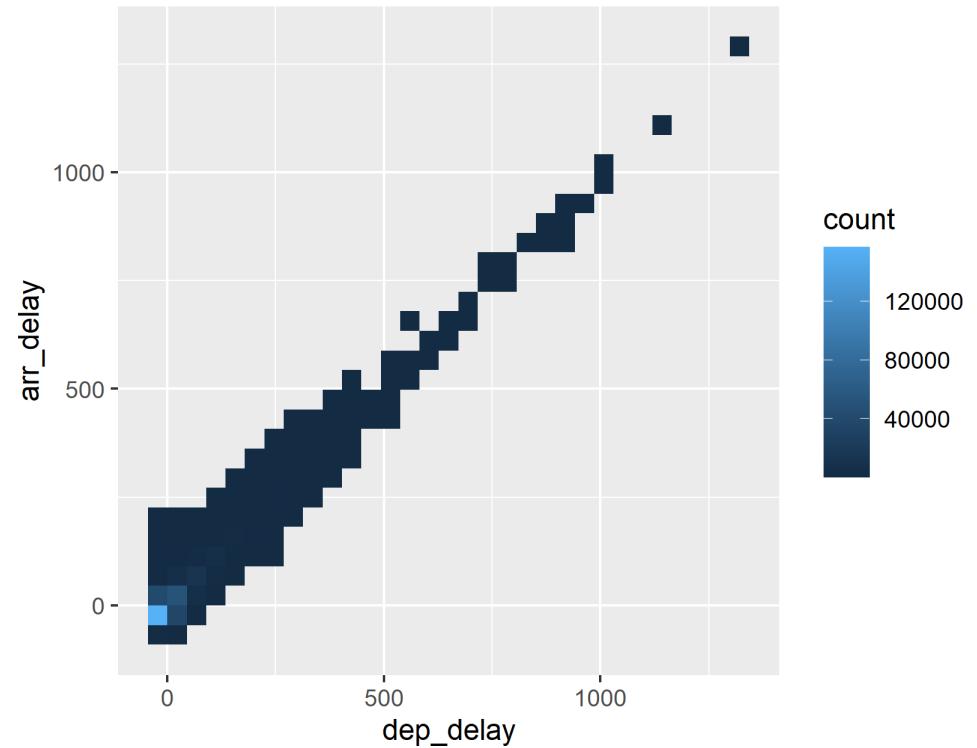
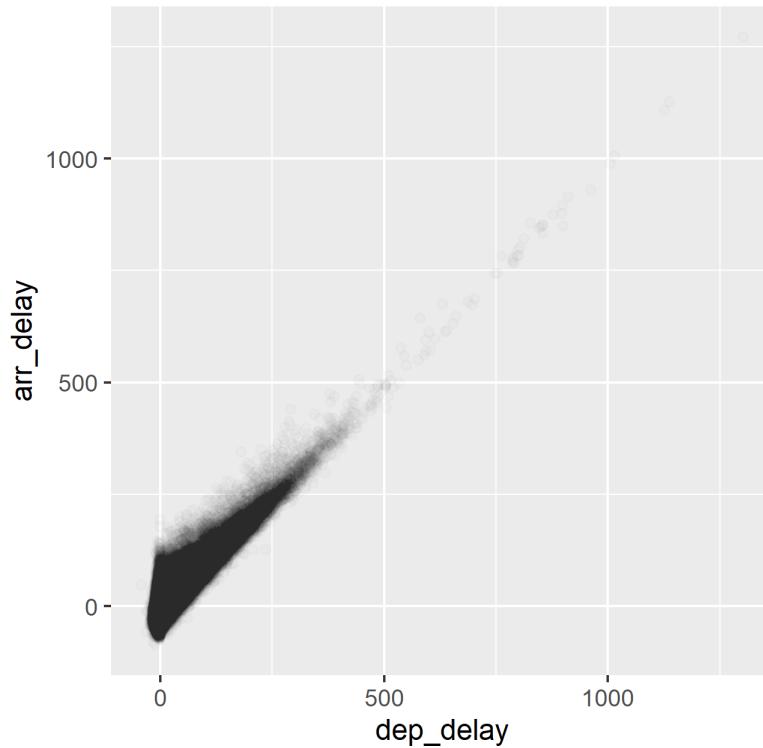
# Scatter plot

- Departure and arrival delay correlate strongly
- Arrival delay is generally higher than departure delay
- Only a few delays are above 500 minutes



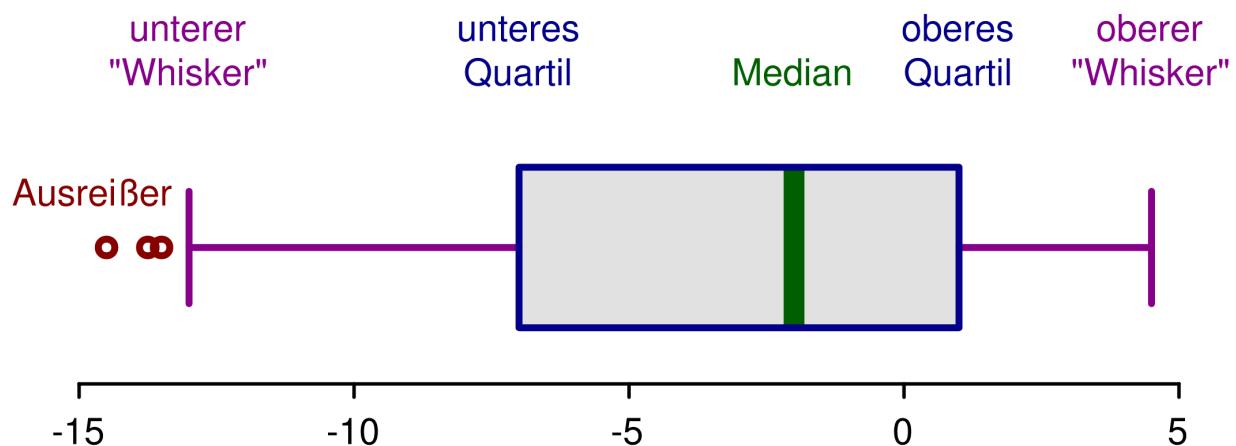
# Solutions to overplotting

- Transparency (alpha) or binning (geom\_bin2d)



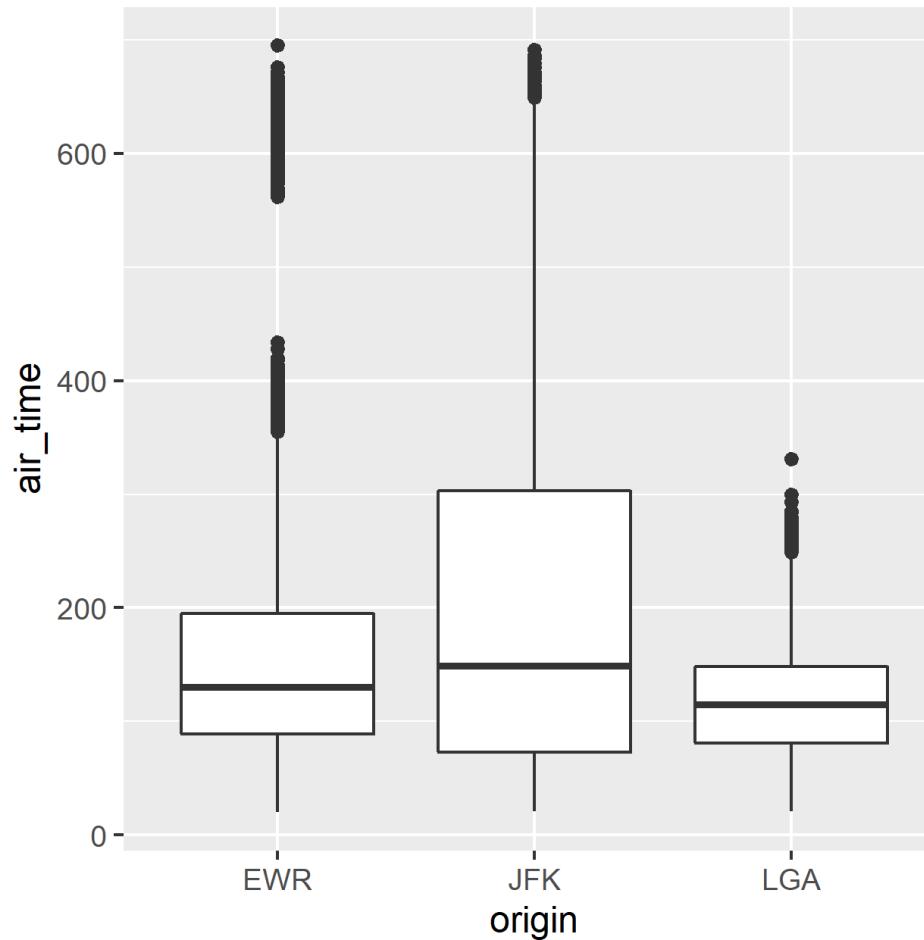
# Box plot (Tukey)

- X position: feature A of interest
- Y position: feature B of interest
- Geom: rectangles for box, lines (whiskers), points for outliers
- Statistical transformations:
  - median, 25% and 75% percentile  
**(inter quartile range, IQR),**  
1.5×IQR for “whiskers”
- Continuous vs. categorical, e.g., origin and airtime



# Box plot

```
ggplot(data = flights) +  
  geom_boxplot(aes(x = origin, y = air_time))
```

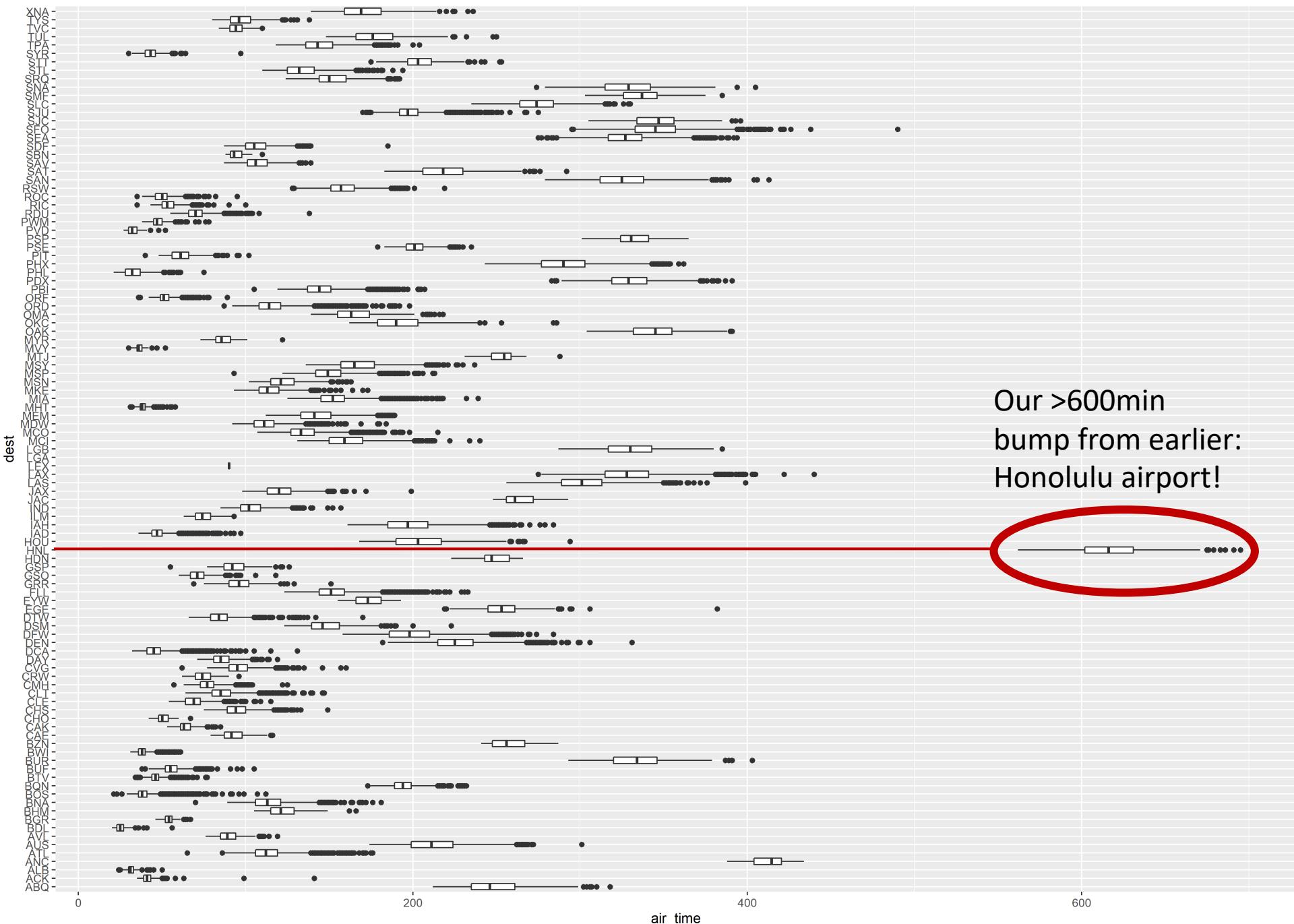


# Box plot

- Flights from LGA do not have the >600 minute bump
  - Flights from JFK take longest, on average (median)
  - Flight times from JFK have largest IQR
- 
- Let's look at destination rather than origin?

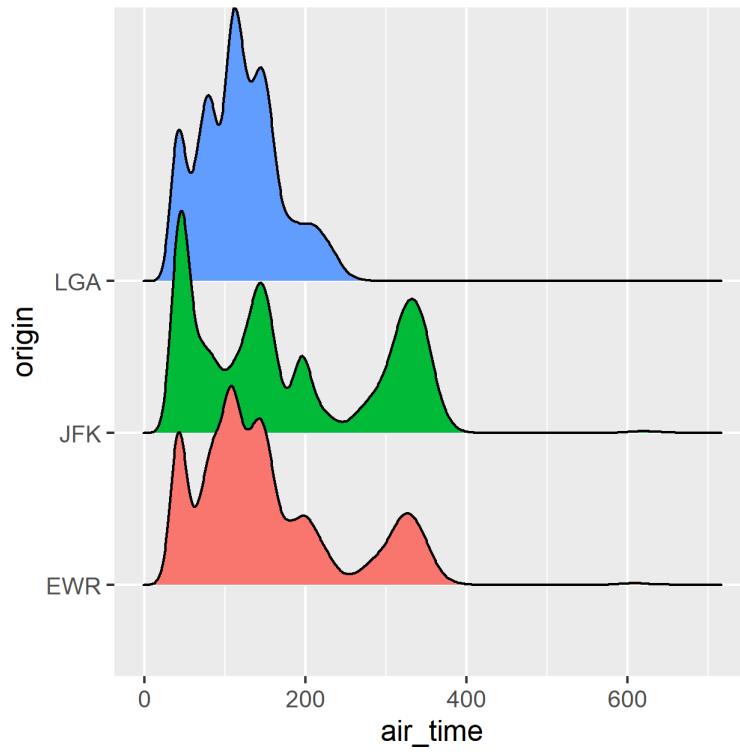


# Box plot by destination

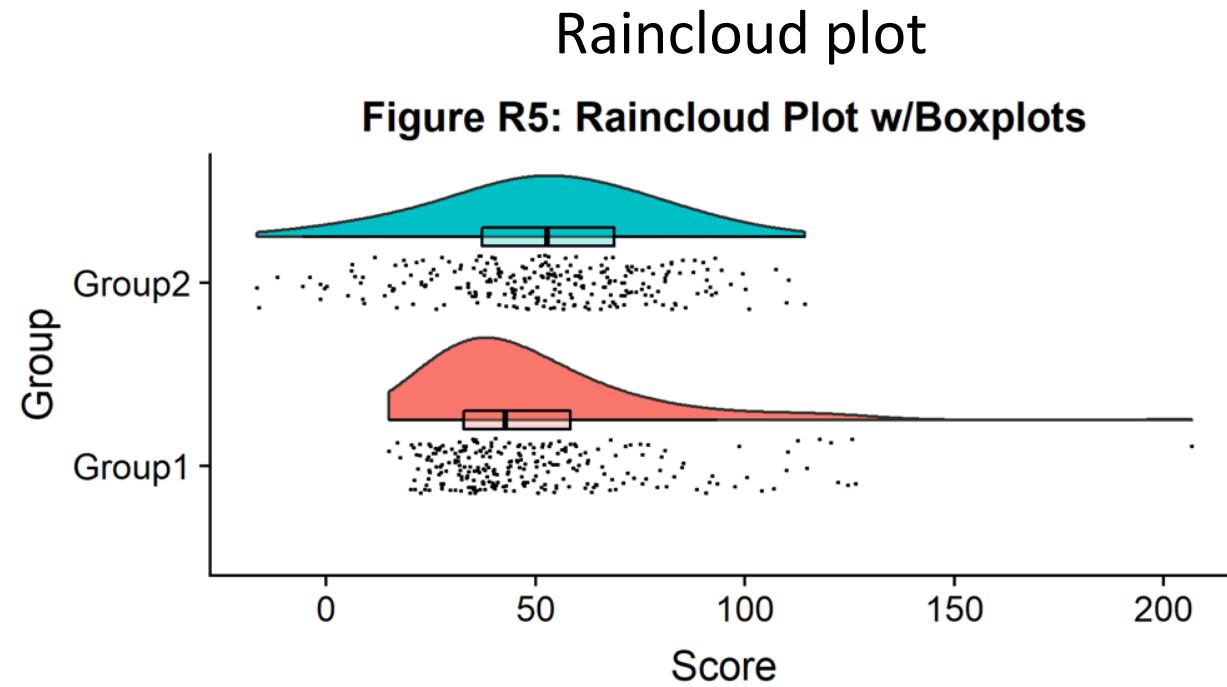


# Many options available for categorical vs continuous

ggridges (ridgeline plot)



Raincloud plot

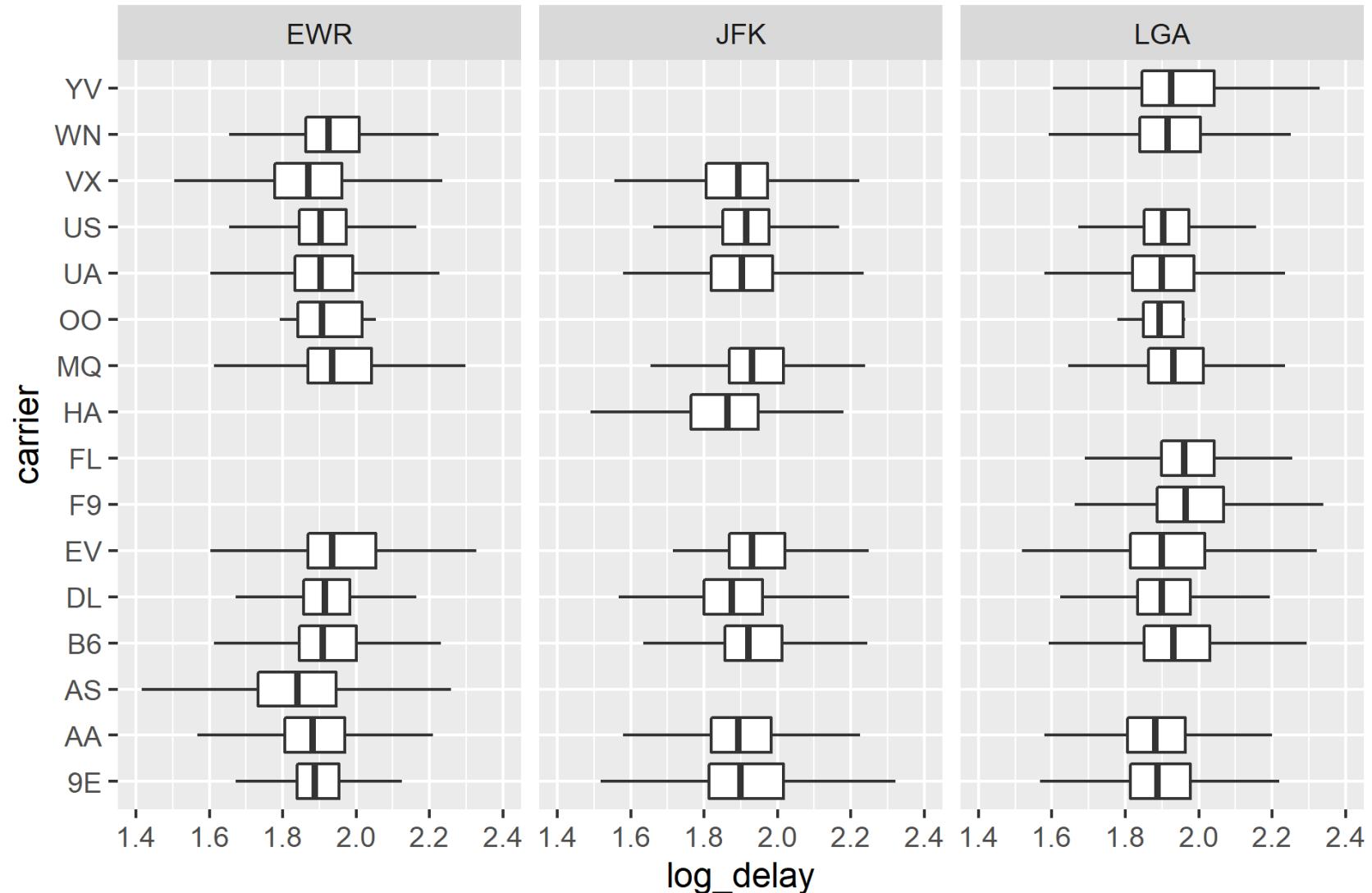


# Facets

- Faceting a plot means creating a subplot per category
  - Also called “small multiples” (tufte)
  - Allows for another categorical variable in the visualisation
  - Danger: clutter
- + `facet_wrap(~feature)`



# Boxplot of (log) delay by carrier, faceted by origin



# Facets

- Some carriers do not fly from some airports (missing data)
- Delays are quite similar across origin airports and carriers

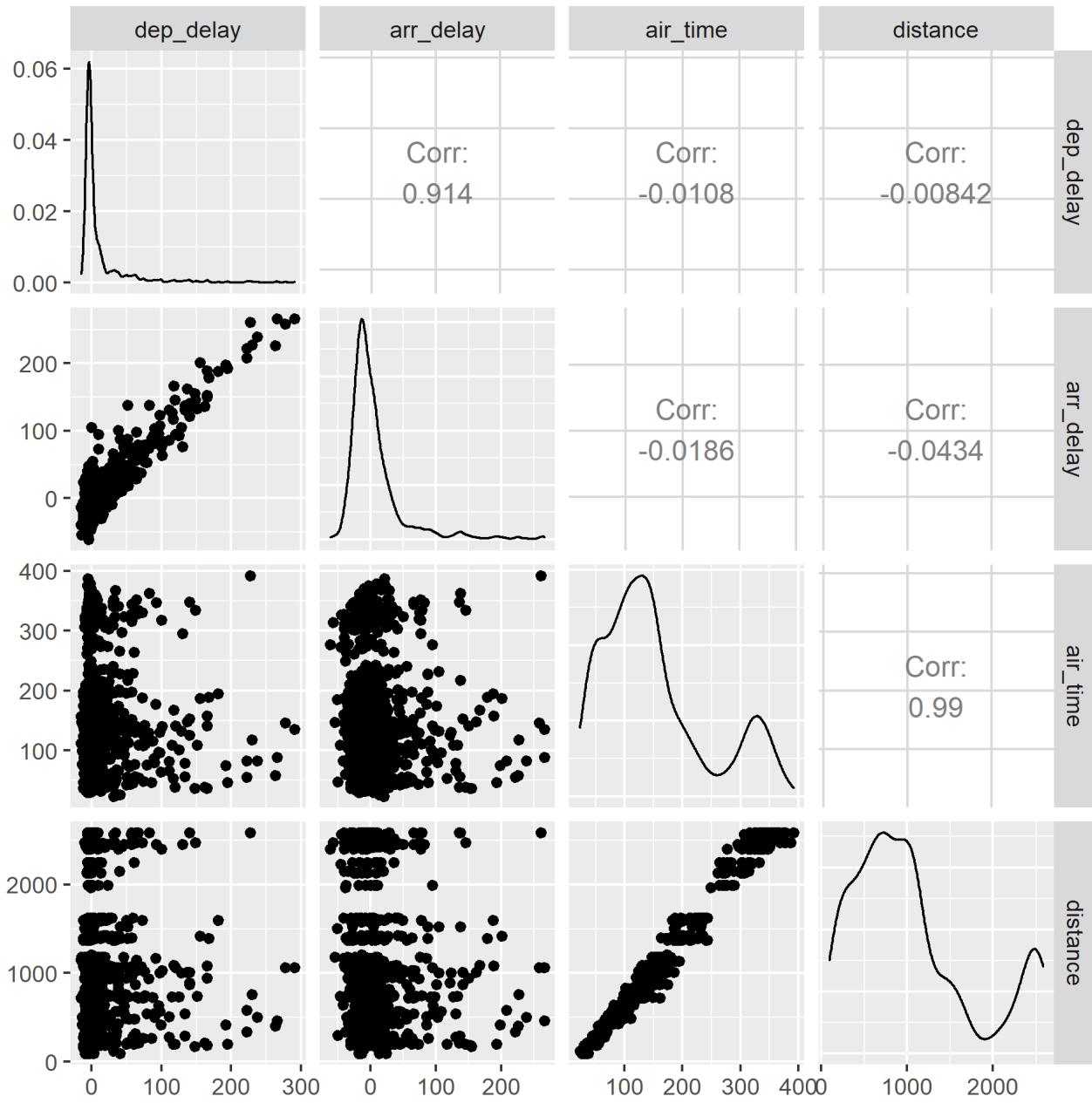


# Pairs plot

- Pairs plot: a scatter plot for each variable
- Multivariate
- Difficult in ggplot: facet by variable (not a feature in the original dataset)
- Data frame needs to be
- Package available: GGally (`ggpairs`)



# Pairs plot



# Pairs plot

- Air time and distance correlate strongly
- Departure and arrival delay correlate strongly
- Air time / distance do not correlate with delay
- In other words: distance cannot predict delays well



# Conclusion

- Exploration is key to understanding things you did not know already
- EDA walks a fine line between seeing useful and less useful things (overfitting, if you like)
- Some useful principles are:
  - Peng's checklist
  - Understanding what you're seeing
  - Finding interesting comparisons
  - “Straightening and flattening” (using models and residuals)
  - Standard graphs to look at variation and association

