

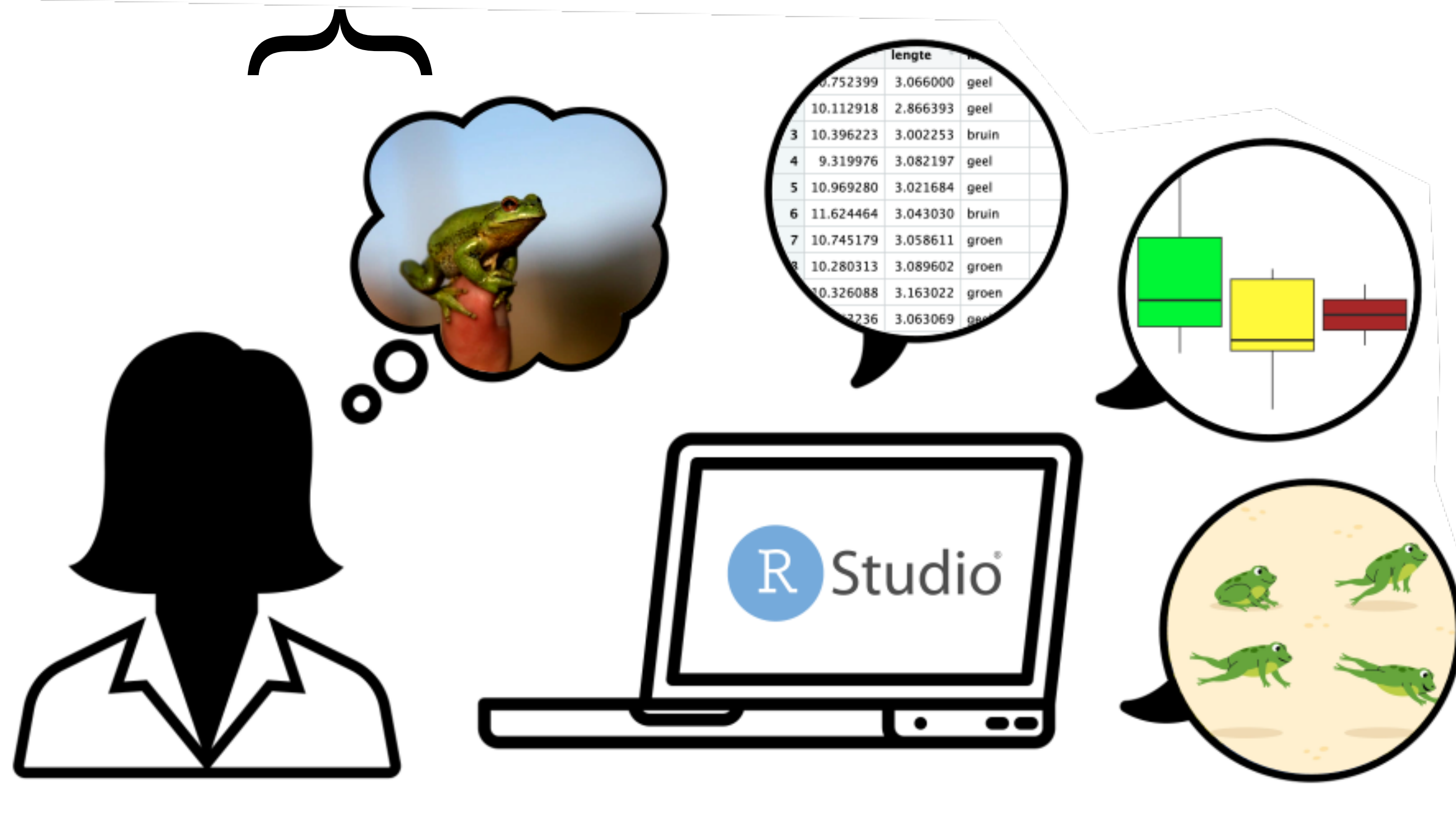
Introductie programmeren met Rstudio

Bram van Dijk (hij/hem)





jij als bioloog
(veldwerk, microscopie, gedragsobservaties...)



jij als bioloog
(veldwerk, microscopie, gedragsobservaties...)

... een boel data.



jij als bioloog
(veldwerk, microscopie, gedragsobservaties...)



... een boel data.

Zo valt het te overzien!

jij als bioloog
(veldwerk, microscopie, gedragsobservaties...)

... een boel data.



Zo valt het te overzien!

Zou ik kunnen voorspellen
wat er met de kikkerpopulatie
gaat gebeuren?

Wat is dat dan, programmeren?

Wat is dat dan, programmeren?

- Programmeren is het **geven van instructies aan een computer** om bepaalde taken uit te voeren, zoals het **analyseren van data**, het **maken van modellen** of het **uitvoeren van simulaties**.

Wat is dat dan, programmeren?

- Programmeren is het **geven van instructies aan een computer** om bepaalde taken uit te voeren, zoals het **analyseren van data**, het **maken van modellen** of het **uitvoeren van simulaties**.
- We moeten **algoritmisch leren denken**: dat betekent dat we problemen stap voor stap oplossen door duidelijke, logische instructies te bedenken die een computer kan volgen.

Wat is dat dan, programmeren?

- Programmeren is het **geven van instructies aan een computer** om bepaalde taken uit te voeren, zoals het **analyseren van data**, het **maken van modellen** of het **uitvoeren van simulaties**.
- We moeten **algoritmisch leren denken**: dat betekent dat we problemen stap voor stap oplossen door duidelijke, logische instructies te bedenken die een computer kan volgen.
- Er bestaan veel programmeertalen (naast **R** ook **python**, C, C++, C#, perl, Java, ❤️ Javascript ❤️, Rust, Fortran, de lijst is eindeloos).

Wat is dat dan, programmeren?

- Programmeren is het **geven van instructies aan een computer** om bepaalde taken uit te voeren, zoals het **analyseren van data**, het **maken van modellen** of het **uitvoeren van simulaties**.
- We moeten **algoritmisch leren denken**: dat betekent dat we problemen stap voor stap oplossen door duidelijke, logische instructies te bedenken die een computer kan volgen.
- Er bestaan veel programmeertalen (naast **R** ook **python**, C, C++, C#, perl, Java, ❤️ Javascript ❤️, Rust, Fortran, de lijst is eindeloos).
- Syntax (hoe je instructies geeft) is per taal net iets anders, maar de achterliggende algoritmen niet! Verlies je niet op de syntax, maar probeer algoritmisch te leren denken!

Wat is dat dan, programmeren?

- Programmeren is het **geven van instructies aan een computer** om bepaalde taken uit te voeren, zoals het **analyseren van data**, het **maken van modellen** of het **uitvoeren van simulaties**.
- We moeten **algorithmisch leren denken**: dat betekent dat we problemen stap voor stap oplossen door duidelijke, logische instructies te bedenken die een computer kan volgen.
- Er bestaan veel programmeertalen (naast **R** ook **python**, C, C++, C#, perl, Java, ❤️ Javascript ❤️, Rust, Fortran, de lijst is eindeloos).
- Syntax (hoe je instructies geeft) is per taal net iets anders, maar de achterliggende algoritmen niet! Verlies je niet op de syntax, maar probeer algorithmisch te leren denken!
- R is makkelijk te installeren, en wordt veel door biologen gebruikt. Hierdoor is er ook veel hulp te vinden.

Wat is dat dan, programmeren?

- Programmeren is het **geven van instructies aan een computer** om bepaalde taken uit te voeren, zoals het **analyseren van data**, het **maken van modellen** of het **uitvoeren van simulaties**.
- We moeten **algorithmisch leren denken**: dat betekent dat we problemen stap voor stap oplossen door duidelijke, logische instructies te bedenken die een computer kan volgen.
- Er bestaan veel programmeertalen (naast **R** ook **python**, C, C++, C#, perl, Java, ❤️ Javascript ❤️, Rust, Fortran, de lijst is eindeloos).
- Syntax (hoe je instructies geeft) is per taal net iets anders, maar de achterliggende algoritmen niet! Verlies je niet op de syntax, maar probeer algorithmisch te leren denken!
- R is makkelijk te installeren, en wordt veel door biologen gebruikt. Hierdoor is er ook veel hulp te vinden.

Wat is dat dan, programmeren?

- Programmeren is het **geven van instructies aan een computer** om bepaalde taken uit te voeren, zoals het **analyseren van data**, het **maken van modellen** of het **uitvoeren van simulaties**.
- We moeten **algoritmisch leren denken**: dat betekent dat we problemen stap voor stap oplossen door duidelijke, logische instructies te bedenken die een computer kan volgen.
- Er bestaan veel programmeertalen (naast **R** ook **python**, C, C++, C#, perl, Java, ❤️ Javascript ❤️, Rust, Fortran, de lijst is eindeloos).
- Syntax (hoe je instructies geeft) is per taal net iets anders, maar de achterliggende algoritmen niet! Verlies je niet op de syntax, maar probeer algoritmisch te leren denken!
- R is makkelijk te installeren, en wordt veel door biologen gebruikt. Hierdoor is er ook veel hulp te vinden.

Wat is dat dan, programmeren?

- Programmeren is het **geven van instructies aan een computer** om bepaalde taken uit te voeren, zoals het **analyseren van data**, het **maken van modellen** of het **uitvoeren van simulaties**.
- We moeten **algorithmisch leren denken**: dat betekent dat we problemen stap voor stap oplossen door duidelijke, logische instructies te bedenken die een computer kan volgen.
- Er bestaan veel programmeertalen (naast **R** ook **python**, C, C++, C#, perl, Java, ❤️ Javascript ❤️, Rust, Fortran, de lijst is eindeloos).
- Syntax (hoe je instructies geeft) is per taal net iets anders, maar de achterliggende algoritmen niet! Verlies je niet op de syntax, maar probeer algorithmisch te leren denken!
- R is makkelijk te installeren, en wordt veel door biologen gebruikt. Hierdoor is er ook veel hulp te vinden.

Wat is dat dan, programmeren?

- Programmeren is het **geven van instructies aan een computer** om bepaalde taken uit te voeren, zoals het **analyseren van data**, het **maken van modellen** of het **uitvoeren van simulaties**.
- We moeten **algorithmisch leren denken**: dat betekent dat we problemen stap voor stap oplossen door duidelijke, logische instructies te bedenken die een computer kan volgen.
- Er bestaan veel programmeertalen (naast **R** ook **python**, C, C++, C#, perl, Java, ❤️ Javascript ❤️, Rust, Fortran, de lijst is eindeloos).
- Syntax (hoe je instructies geeft) is per taal net iets anders, maar de achterliggende algoritmen niet! Verlies je niet op de syntax, maar probeer algorithmisch te leren denken!
- R is makkelijk te installeren, en wordt veel door biologen gebruikt. Hierdoor is er ook veel hulp te vinden.

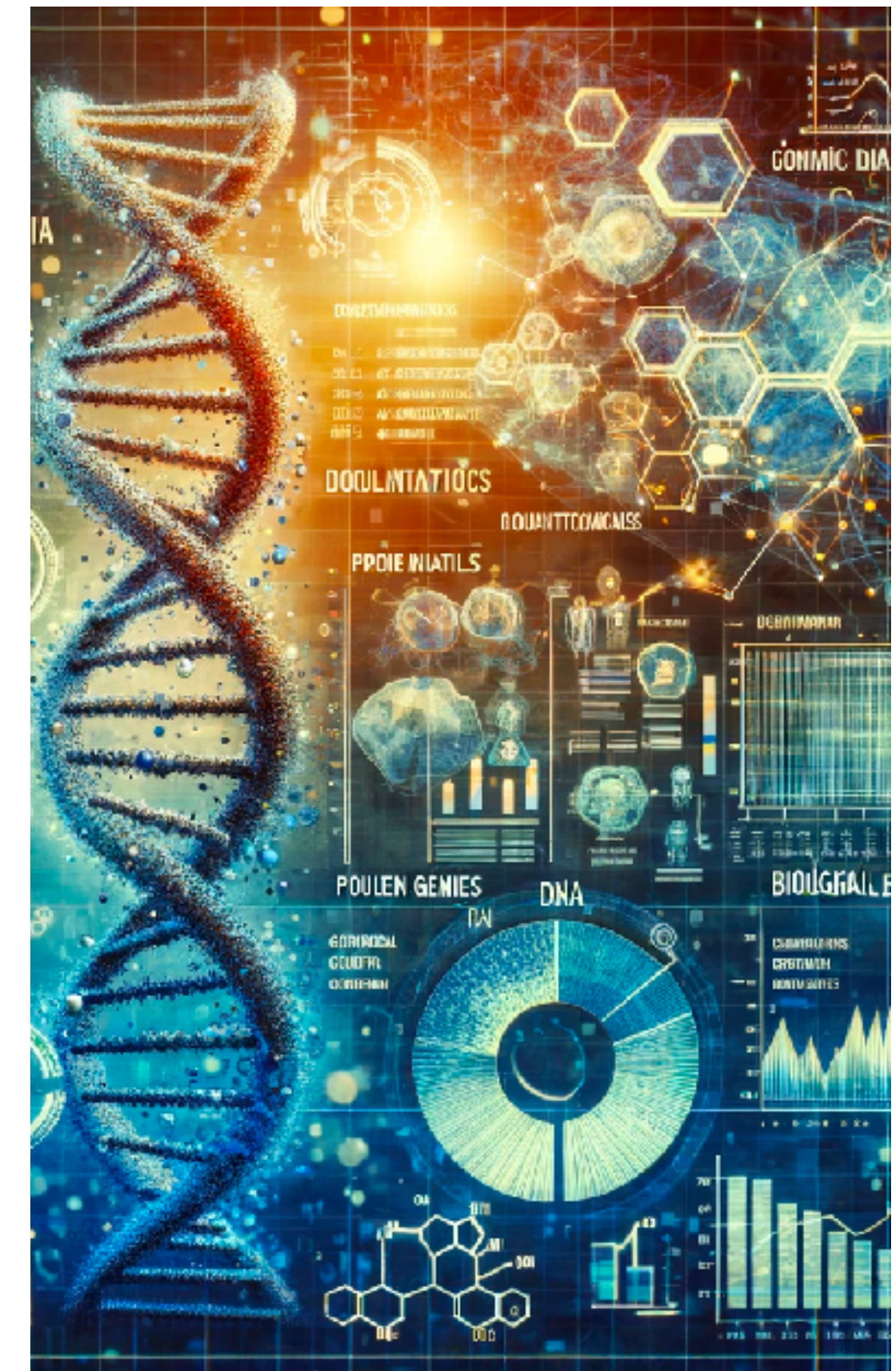
Wat is dat dan, programmeren?

- Programmeren is het **geven van instructies aan een computer** om bepaalde taken uit te voeren, zoals het **analyseren van data**, het **maken van modellen** of het **uitvoeren van simulaties**.
- We moeten **algorithmisch leren denken**: dat betekent dat we problemen stap voor stap oplossen door duidelijke, logische instructies te bedenken die een computer kan volgen.
- Er bestaan veel programmeertalen (naast **R** ook **python**, C, C++, C#, perl, Java, ❤️ Javascript ❤️, Rust, Fortran, de lijst is eindeloos).
- Syntax (hoe je instructies geeft) is per taal net iets anders, maar de achterliggende algoritmen niet! Verlies je niet op de syntax, maar probeer algorithmisch te leren denken!
- R is makkelijk te installeren, en wordt veel door biologen gebruikt. Hierdoor is er ook veel hulp te vinden.

Wat is dat dan, programmeren?

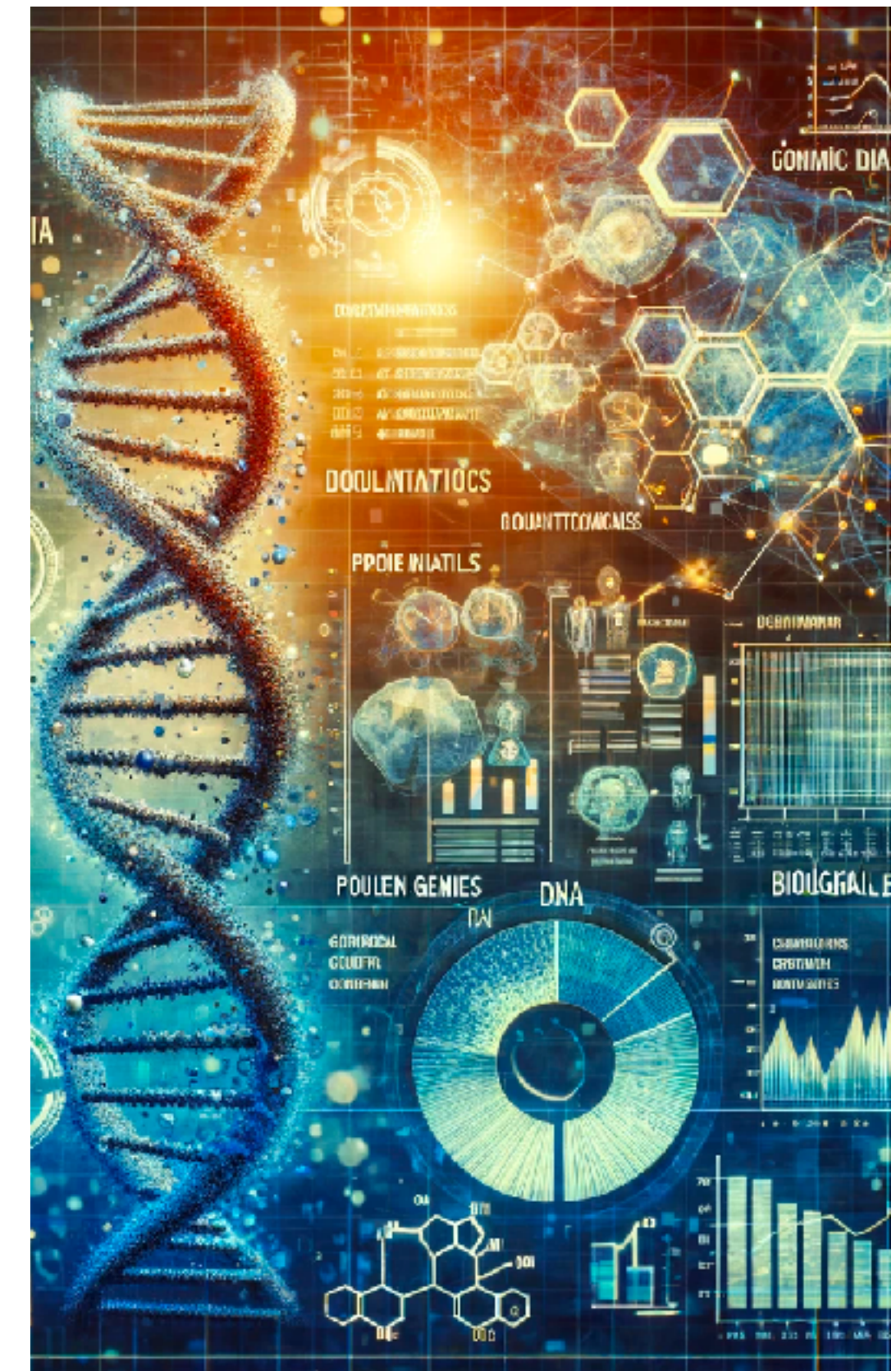
- Programmeren is het **geven van instructies aan een computer** om bepaalde taken uit te voeren, zoals het **analyseren van data**, het **maken van modellen** of het **uitvoeren van simulaties**.
- We moeten **algorithmisch leren denken**: dat betekent dat we problemen stap voor stap oplossen door duidelijke, logische instructies te bedenken die een computer kan volgen.
- Er bestaan veel programmeertalen (naast **R** ook **python**, C, C++, C#, perl, Java, ❤️ Javascript ❤️, Rust, Fortran, de lijst is eindeloos).
- Syntax (hoe je instructies geeft) is per taal net iets anders, maar de achterliggende algoritmen niet! Verlies je niet op de syntax, maar probeer algorithmisch te leren denken!
- R is makkelijk te installeren, en wordt veel door biologen gebruikt. Hierdoor is er ook veel hulp te vinden.

Biologie van de 21e eeuw...



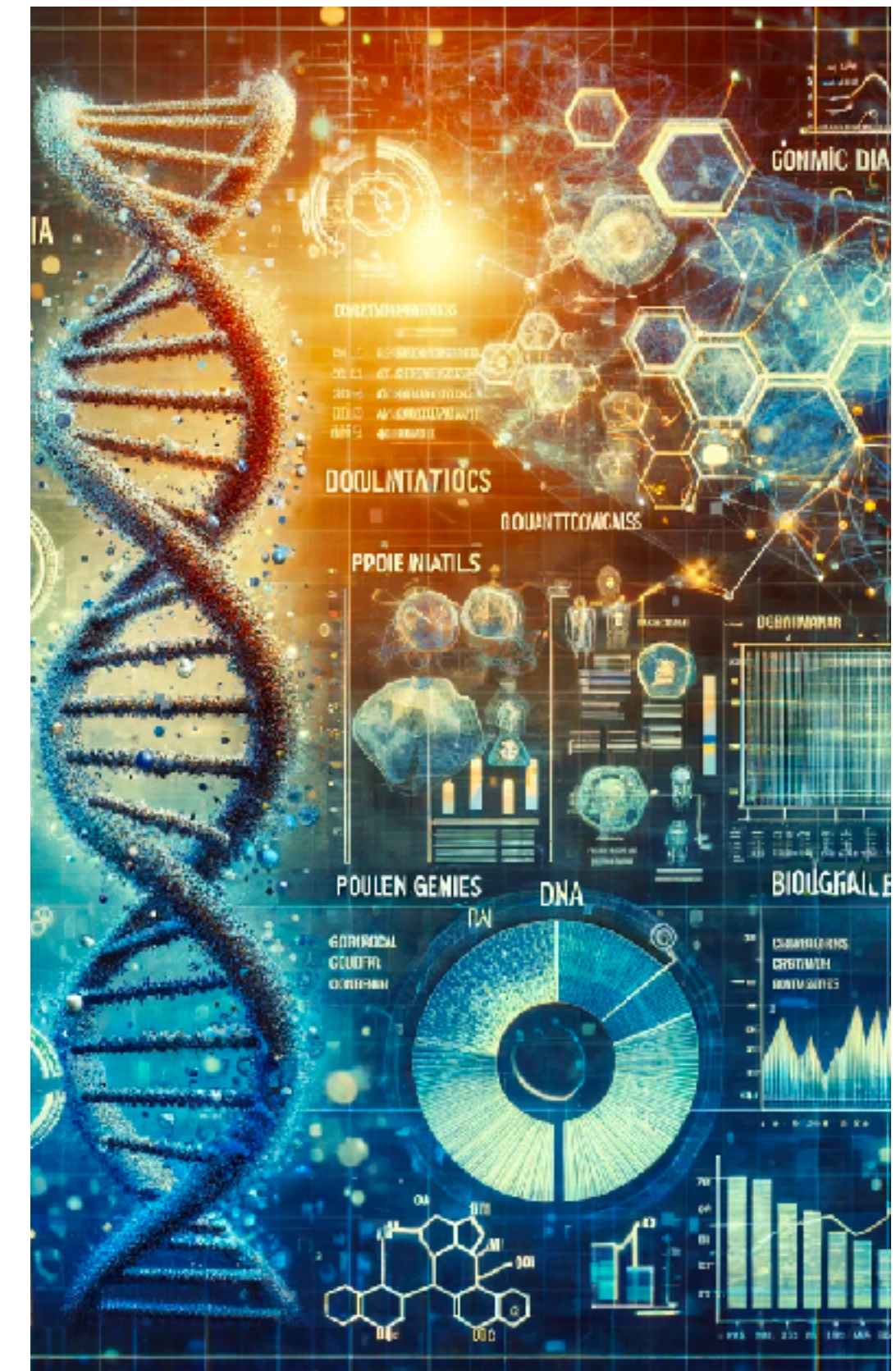
Biologie van de 21e eeuw...

- “Big data” laat zien dat biologie op alle domeinen uit erg veel interacties bestaat:
 - ➔ Het immuunsysteem
 - ➔ Complexe genregulatiernetwerken
 - ➔ Voedselwebben met 100-en soorten



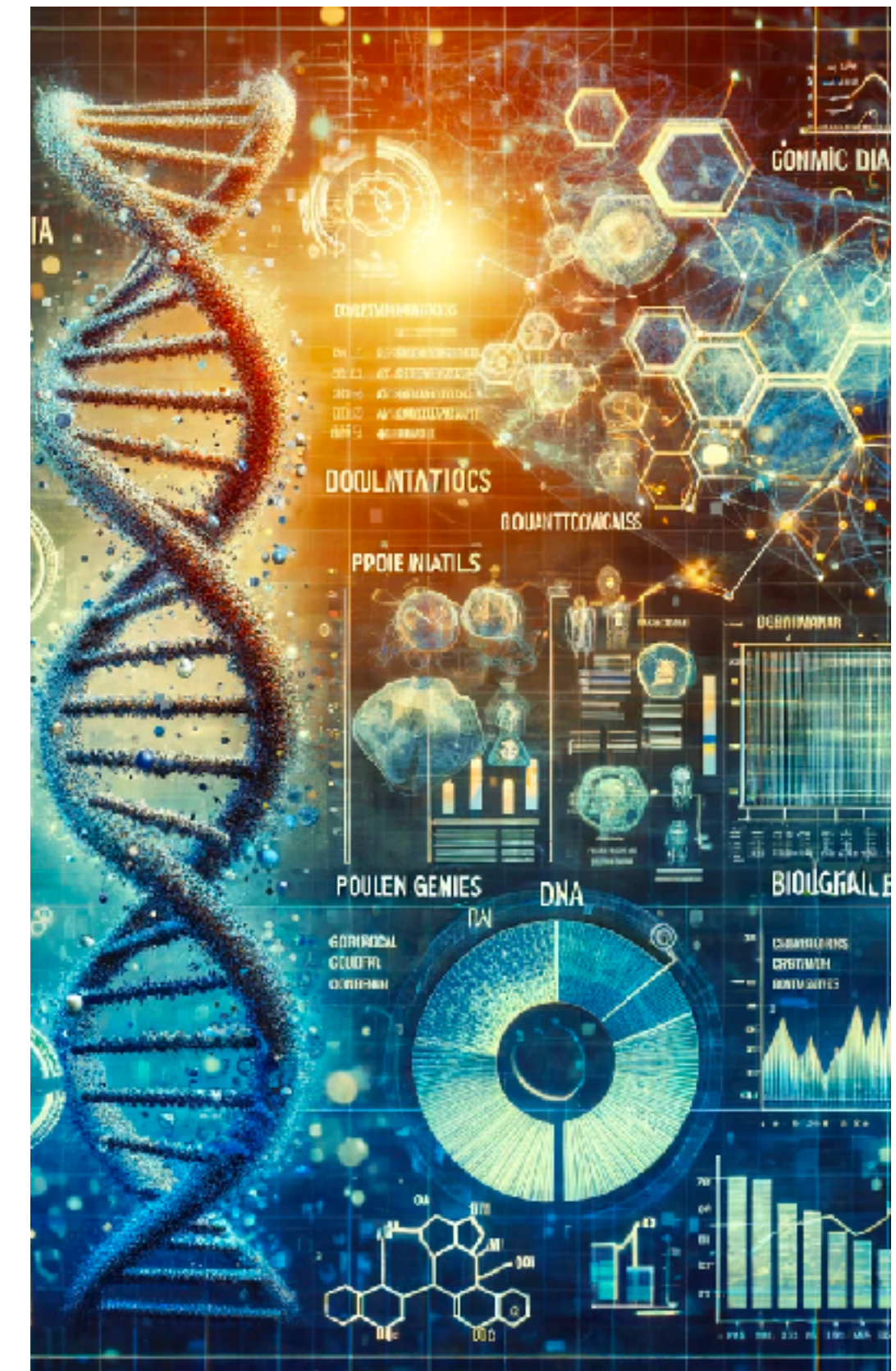
Biologie van de 21e eeuw...

- “Big data” laat zien dat biologie op alle domeinen uit erg veel interacties bestaat:
 - ➔ Het immuunsysteem
 - ➔ Complexe genregulatiernetwerken
 - ➔ Voedselwebben met 100-en soorten
- Veel data (zelfs van simpele interacties) geeft enorm veel mogelijkheden



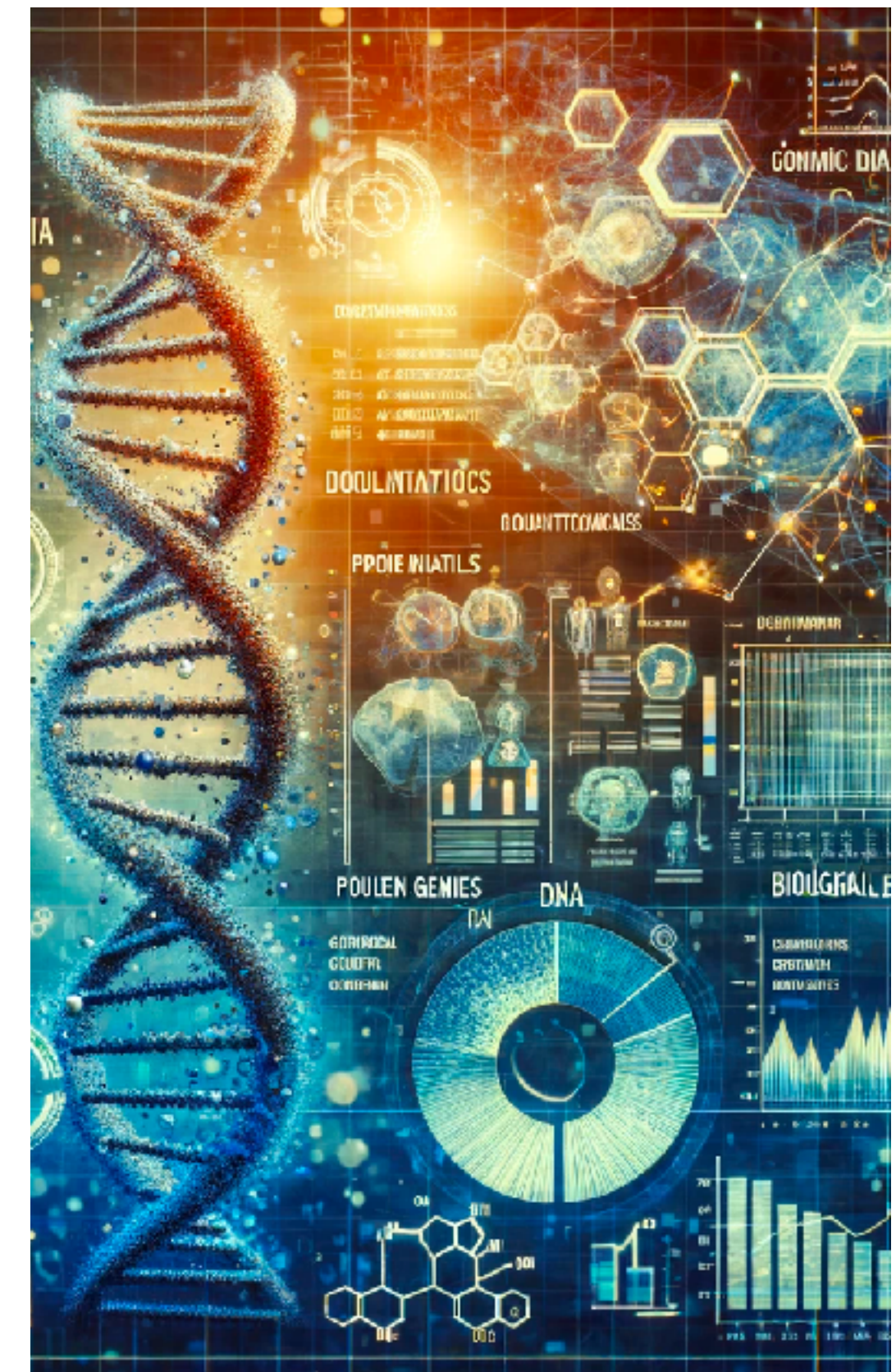
Biologie van de 21e eeuw...

- “Big data” laat zien dat biologie op alle domeinen uit erg veel interacties bestaat:
 - ➔ Het immuunsysteem
 - ➔ Complexe genregulatiernetwerken
 - ➔ Voedselwebben met 100-en soorten
- Veel data (zelfs van simpele interacties) geeft enorm veel mogelijkheden
- Meer data $\neq$ meer kennis!



Biologie van de 21e eeuw...

- “Big data” laat zien dat biologie op alle domeinen uit erg veel interacties bestaat:
 - ➔ Het immuunsysteem
 - ➔ Complexe genregulatiernetwerken
 - ➔ Voedselwebben met 100-en soorten
- Veel data (zelfs van simpele interacties) geeft enorm veel mogelijkheden
- Meer data $\neq$ meer kennis!
- Kennis = **begrip van hoe dingen werken**

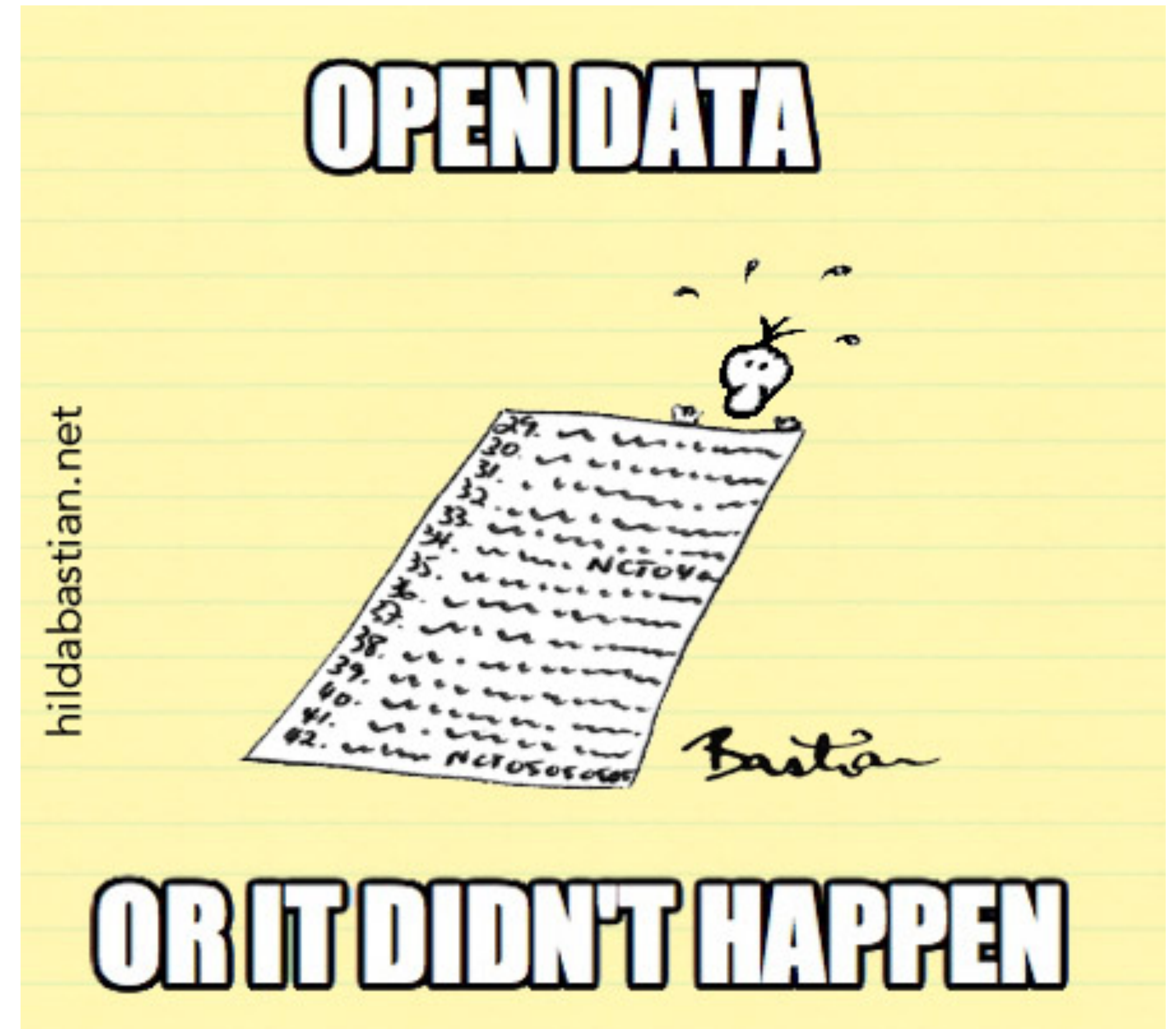


Biologie van de 21e eeuw...

- “Big data” laat zien dat biologie op alle domeinen uit erg veel interacties bestaat:
 - ➔ Het immuunsysteem
 - ➔ Complexe genregulatiernetwerken
 - ➔ Voedselwebben met 100-en soorten
- Veel data (zelfs van simpele interacties) geeft enorm veel mogelijkheden
- Meer data $\neq$ meer kennis!
- Kennis = **begrip van hoe dingen werken**
- Dus: hoe te voorkomen we dat we verdrinken in deze zee van data?

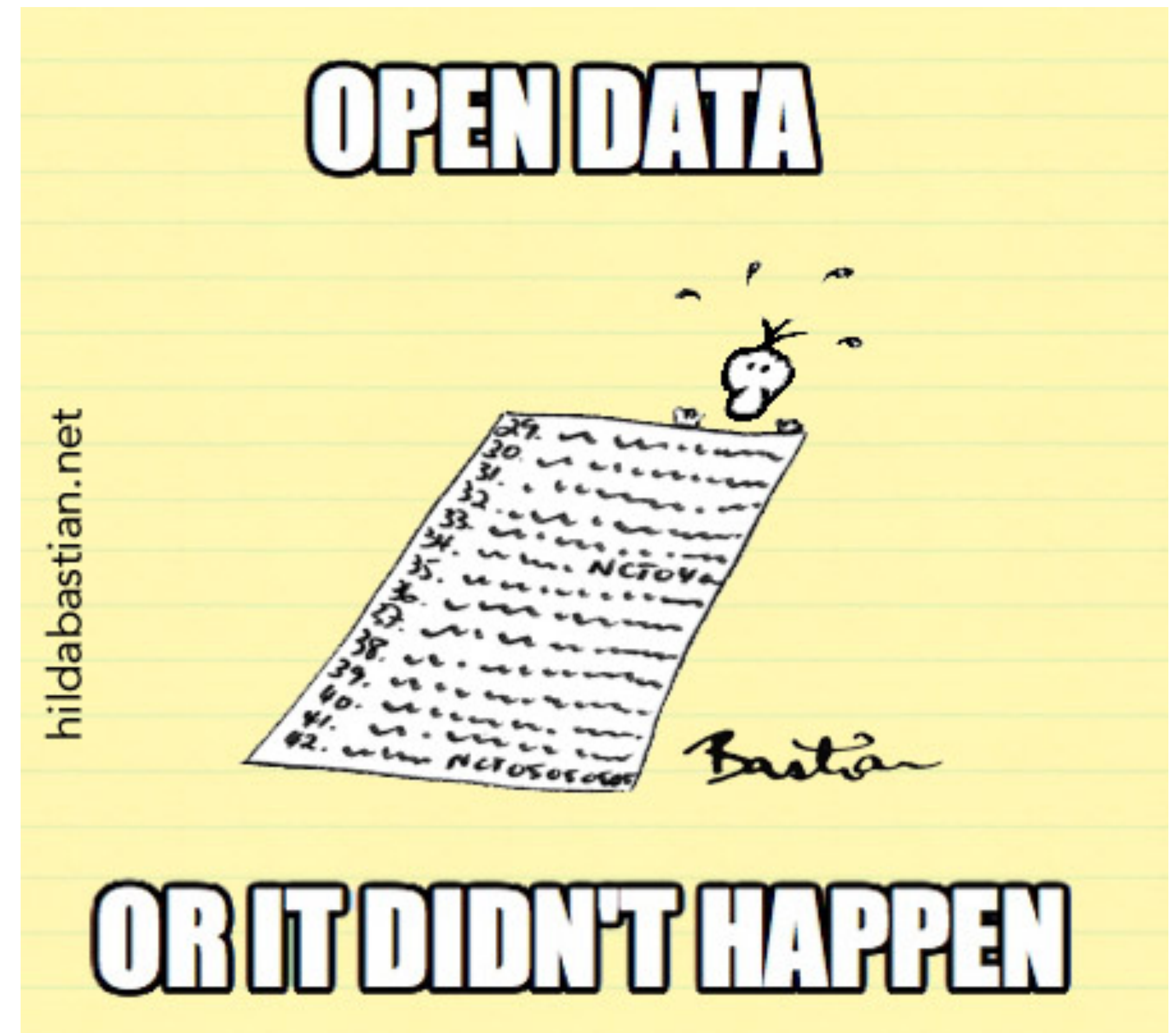


Data en reproduceerbaarheid



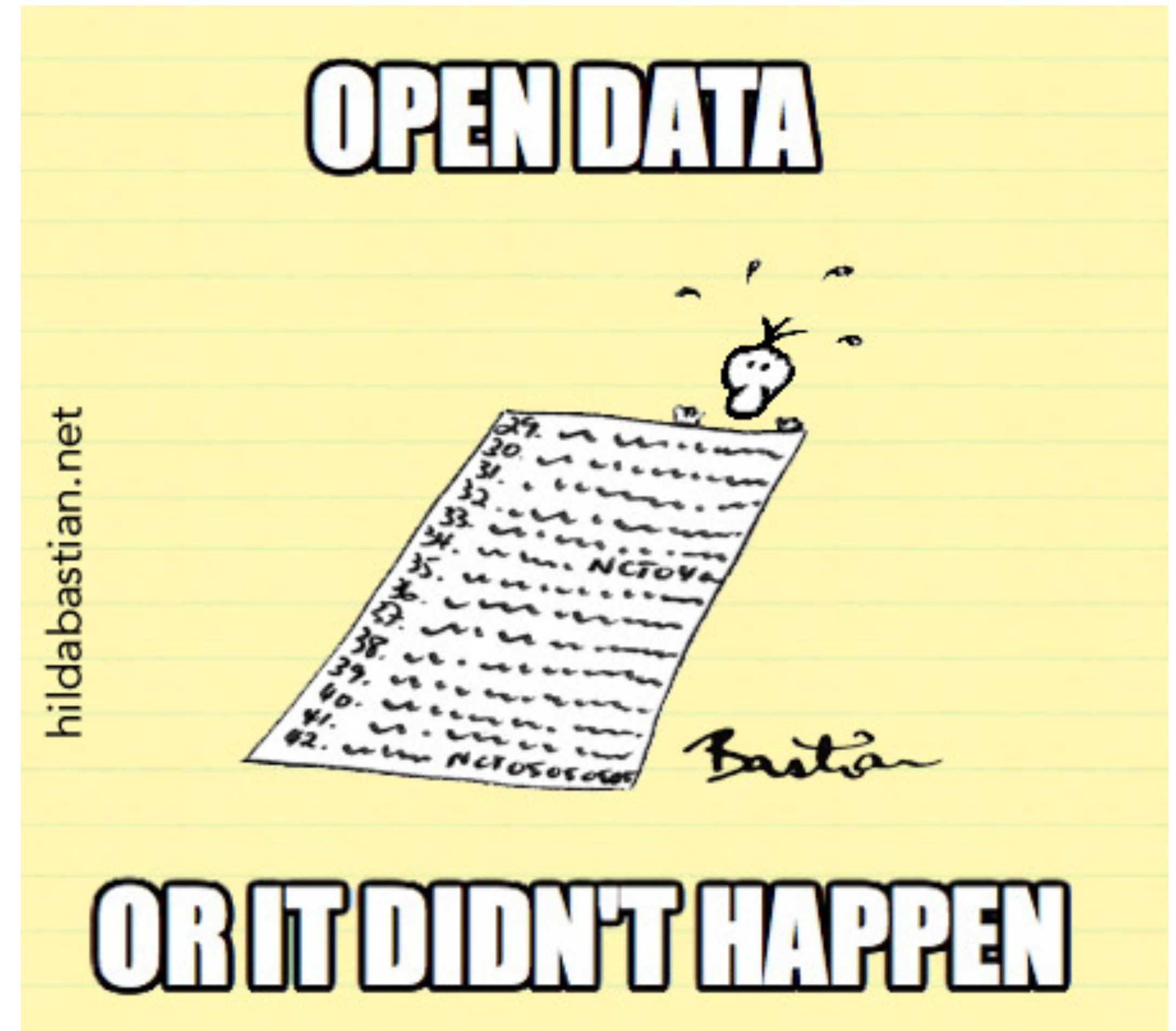
Data en reproduceerbaarheid

- **Open Science:** andere wetenschappers moeten niet alleen je experiment, maar ook je analyse kunnen reproduceren



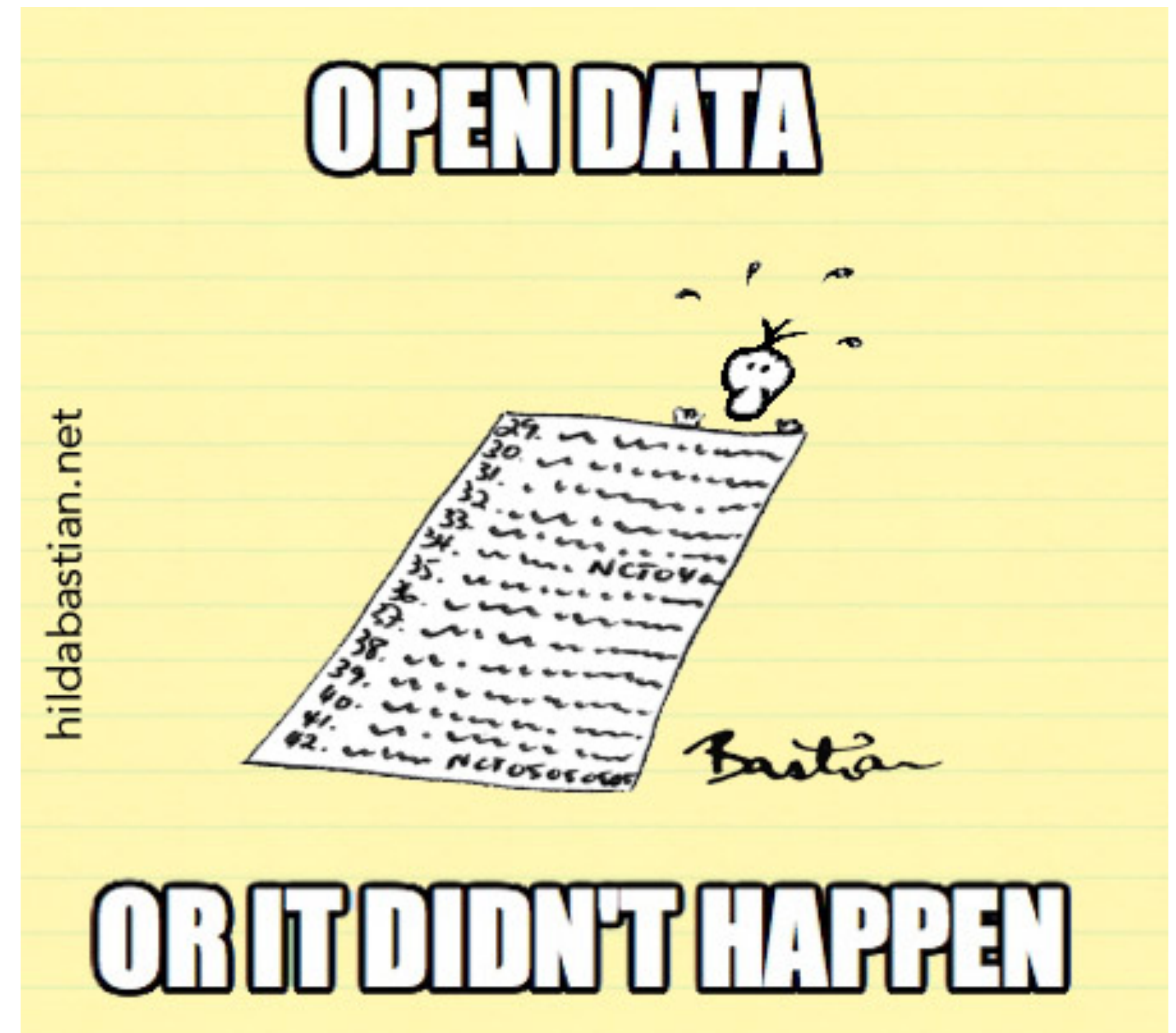
Data en reproduceerbaarheid

- **Open Science:** andere wetenschappers moeten niet alleen je experiment, maar ook je analyse kunnen reproduceren
- **Open Data:** om deze analyse te kunnen reproduceren, hebben ze ook de **data+code** nodig.



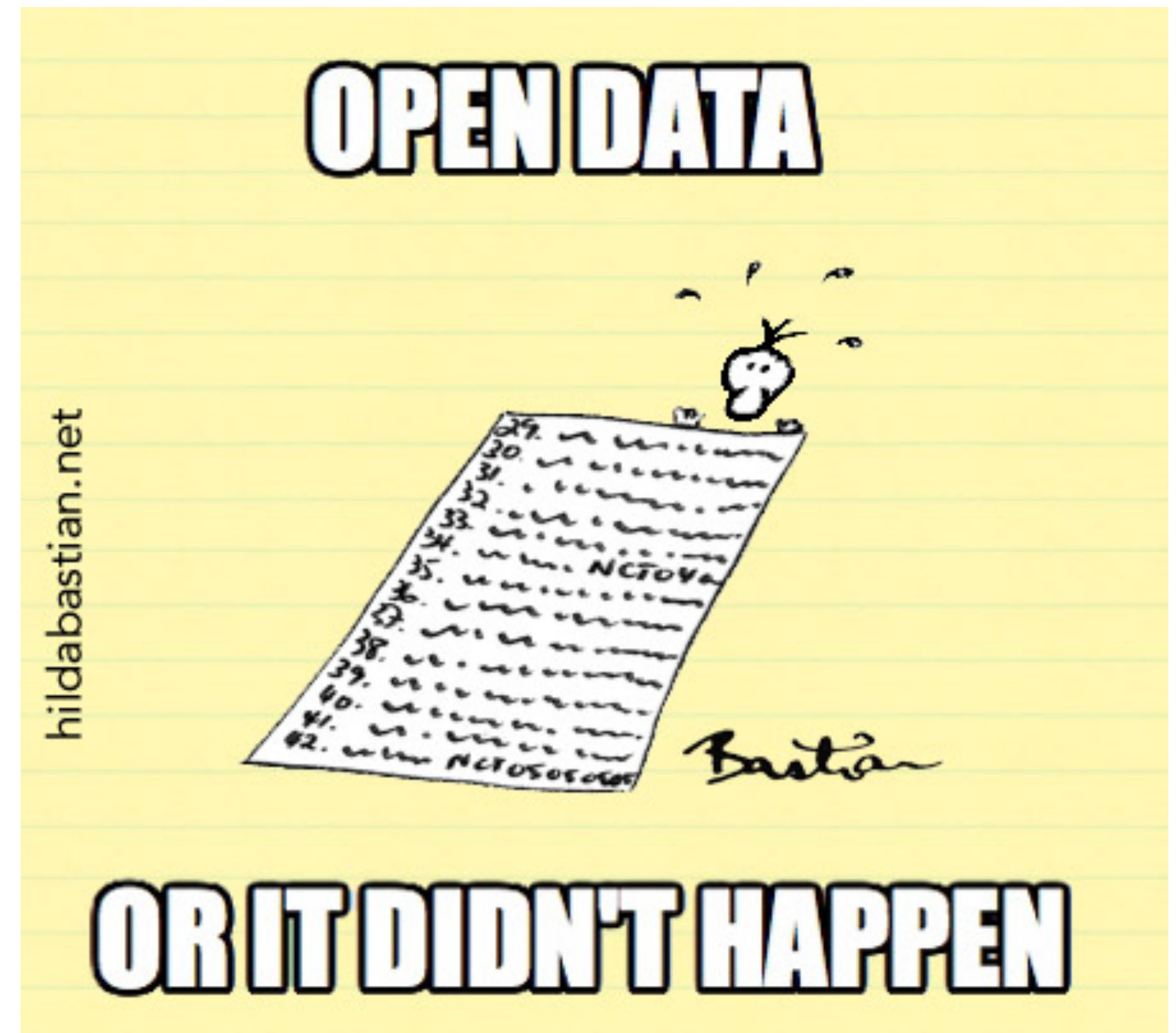
Data en reproduceerbaarheid

- **Open Science:** andere wetenschappers moeten niet alleen je experiment, maar ook je analyse kunnen reproduceren
- **Open Data:** om deze analyse te kunnen reproduceren, hebben ze ook de **data+code** nodig.
- Spreadsheets (e.g. Excel) zijn niet geschikt voor datasets van deze grootteorde



Data en reproduceerbaarheid

- **Open Science:** andere wetenschappers moeten niet alleen je experiment, maar ook je analyse kunnen reproduceren
- **Open Data:** om deze analyse te kunnen reproduceren, hebben ze ook de **data+code** nodig.
- Spreadsheets (e.g. Excel) zijn niet geschikt voor datasets van deze grootteorde
- Data wordt daarom eigenlijk altijd in *plain text* opgeslagen (bijvoorbeeld in 'csv' formaat)



Data en reproduceerbaarheid

- **Open Science:** andere wetenschappers moeten niet alleen je experiment, maar ook je analyse kunnen reproduceren
- **Open Data:** om deze analyse te kunnen reproduceren, hebben ze ook de **data+code** nodig.
- Spreadsheets (e.g. Excel) zijn niet geschikt voor datasets van deze grootteorde
- Data wordt daarom eigenlijk altijd in *plain text* opgeslagen (bijvoorbeeld in 'csv' formaat)

```
Version;Date_of_report;Date_of_statistics;Perc_ARI_symptoms;MA_perc_ARI
1;2025-01-29 10:00:00;2020-11-01;8,50;NA
1;2025-01-29 10:00:00;2020-11-02;6,76;NA
1;2025-01-29 10:00:00;2020-11-03;7,23;NA
1;2025-01-29 10:00:00;2020-11-04;7,83;7,64
1;2025-01-29 10:00:00;2020-11-05;8,14;7,61
1;2025-01-29 10:00:00;2020-11-06;7,71;7,86
1;2025-01-29 10:00:00;2020-11-07;7,29;8,00
1;2025-01-29 10:00:00;2020-11-08;8,28;7,79
1;2025-01-29 10:00:00;2020-11-09;8,55;7,62
1;2025-01-29 10:00:00;2020-11-10;8,17;7,50
1;2025-01-29 10:00:00;2020-11-11;6,40;7,52
1;2025-01-29 10:00:00;2020-11-12;6,93;7,54
1;2025-01-29 10:00:00;2020-11-13;6,86;7,27
1;2025-01-29 10:00:00;2020-11-14;7,48;7,08
1;2025-01-29 10:00:00;2020-11-15;8,39;7,16
1;2025-01-29 10:00:00;2020-11-16;6,63;7,10
1;2025-01-29 10:00:00;2020-11-17;6,84;7,22
1;2025-01-29 10:00:00;2020-11-18;6,97;7,22
1;2025-01-29 10:00:00;2020-11-19;6,51;7,20
1;2025-01-29 10:00:00;2020-11-20;7,74;7,26
1;2025-01-29 10:00:00;2020-11-21;7,43;7,59
1;2025-01-29 10:00:00;2020-11-22;8,31;7,66
1;2025-01-29 10:00:00;2020-11-23;7,05;7,74
1;2025-01-29 10:00:00;2020-11-24;9,13;7,81
1;2025-01-29 10:00:00;2020-11-25;7,45;7,80
1;2025-01-29 10:00:00;2020-11-26;7,10;7,94
1;2025-01-29 10:00:00;2020-11-27;8,18;8,05
1;2025-01-29 10:00:00;2020-11-28;7,39;7,88
1;2025-01-29 10:00:00;2020-11-29;9,27;7,73
1;2025-01-29 10:00:00;2020-11-30;7,85;7,90
1;2025-01-29 10:00:00;2020-12-01;7,92;7,87
1;2025-01-29 10:00:00;2020-12-02;6,38;8,09
1;2025-01-29 10:00:00;2020-12-03;8,34;8,00
```

Data en reproduceerbaarheid

- **Open Science:** andere wetenschappers moeten niet alleen je experiment, maar ook je analyse kunnen reproduceren
- **Open Data:** om deze analyse te kunnen reproduceren, hebben ze ook de **data+code** nodig.
- Spreadsheets (e.g. Excel) zijn niet geschikt voor datasets van deze grootteorde
- Data wordt daarom eigenlijk altijd in *plain text* opgeslagen (bijvoorbeeld in ‘csv’ formaat)

Version	Date_of_report	Date_of_statistics	Perc_ARI_symptoms	MA_perc_ARI_symptoms
1	2025-01-29 10:00:00	2020-11-01	8,50	NA
1	2025-01-29 10:00:00	2020-11-02	6,76	NA
1	2025-01-29 10:00:00	2020-11-03	7,23	NA
1	2025-01-29 10:00:00	2020-11-04	7,83	7,64
1	2025-01-29 10:00:00	2020-11-05	8,14	7,61
1	2025-01-29 10:00:00	2020-11-06	7,71	7,86
1	2025-01-29 10:00:00	2020-11-07	7,29	8,00
1	2025-01-29 10:00:00	2020-11-08	8,28	7,79
1	2025-01-29 10:00:00	2020-11-09	8,55	7,62
1	2025-01-29 10:00:00	2020-11-10	8,17	7,50
1	2025-01-29 10:00:00	2020-11-11	6,40	7,52
1	2025-01-29 10:00:00	2020-11-12	6,93	7,54
1	2025-01-29 10:00:00	2020-11-13	6,86	7,27
1	2025-01-29 10:00:00	2020-11-14	7,48	7,08
1	2025-01-29 10:00:00	2020-11-15	8,39	7,16
1	2025-01-29 10:00:00	2020-11-16	6,63	7,10
1	2025-01-29 10:00:00	2020-11-17	6,84	7,22
1	2025-01-29 10:00:00	2020-11-18	6,97	7,22
1	2025-01-29 10:00:00	2020-11-19	6,51	7,20
1	2025-01-29 10:00:00	2020-11-20	7,74	7,26

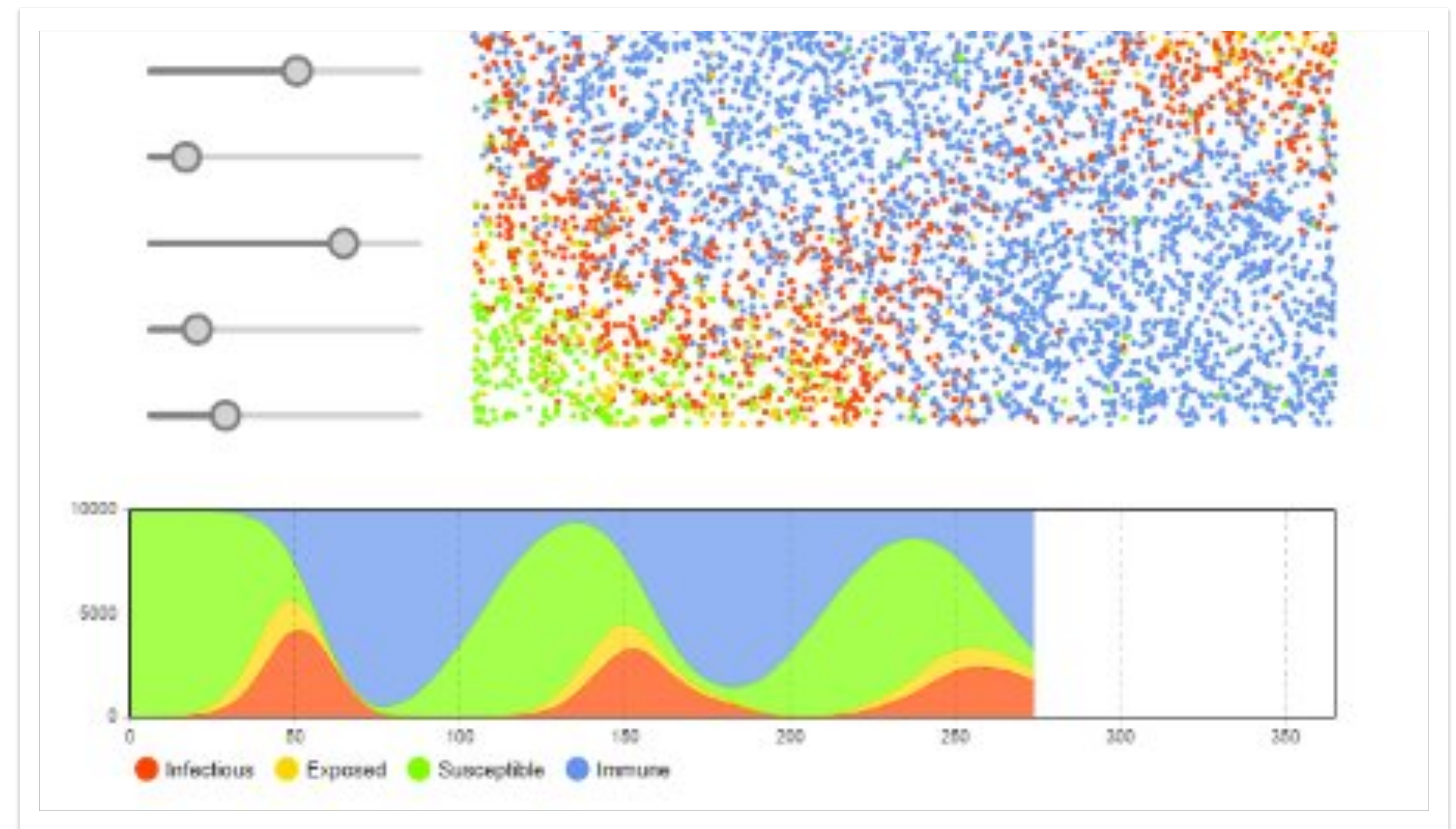
Modellen doorrekenen

Modellen doorrekenen

- Naast **data** speelt de computer ook een belangrijke rol in het doorrekenen van (voorspellende) modellen

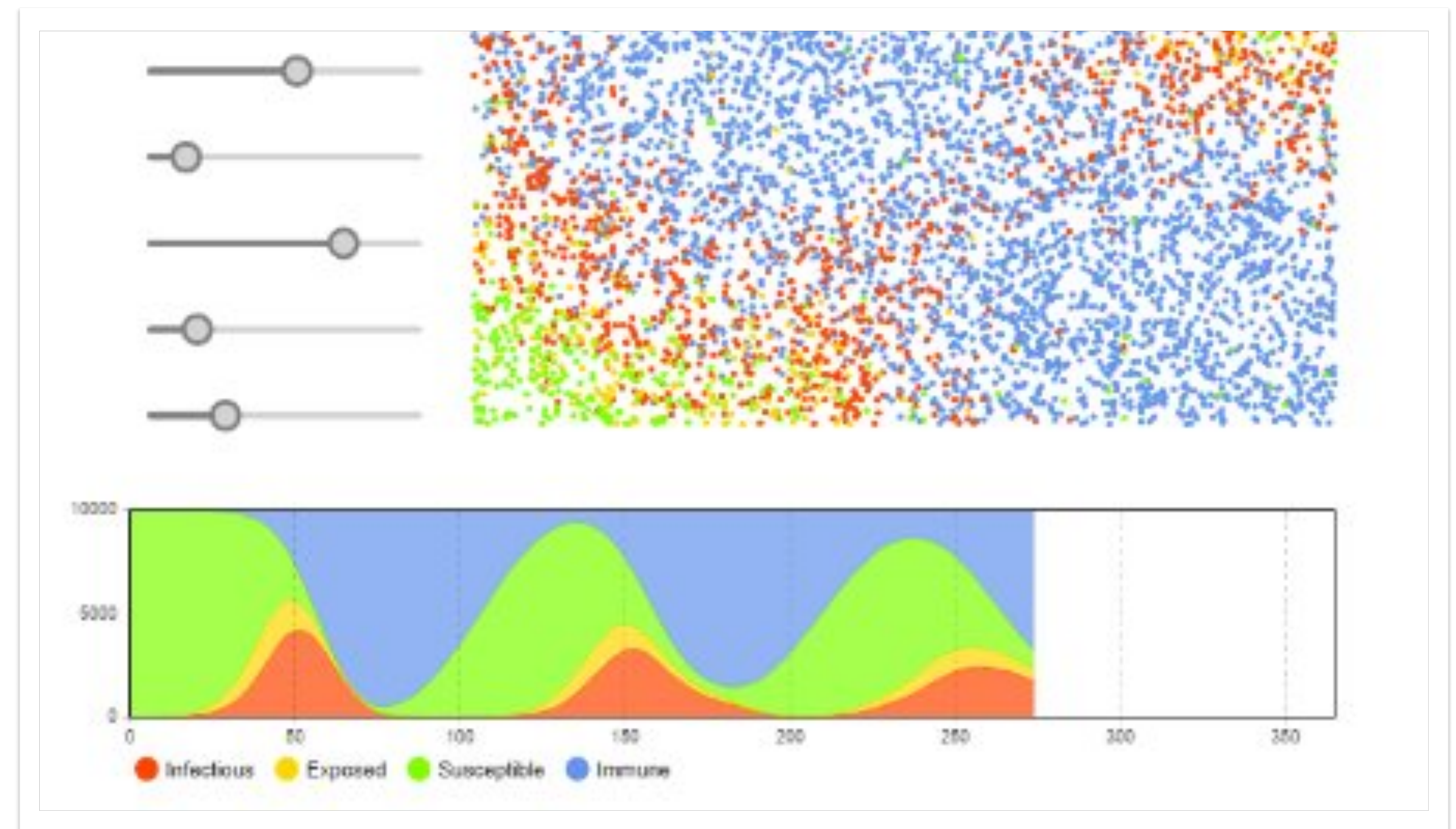
Modellen doorrekenen

- Naast **data** speelt de computer ook een belangrijke rol in het doorrekenen van (voorspellende) modellen



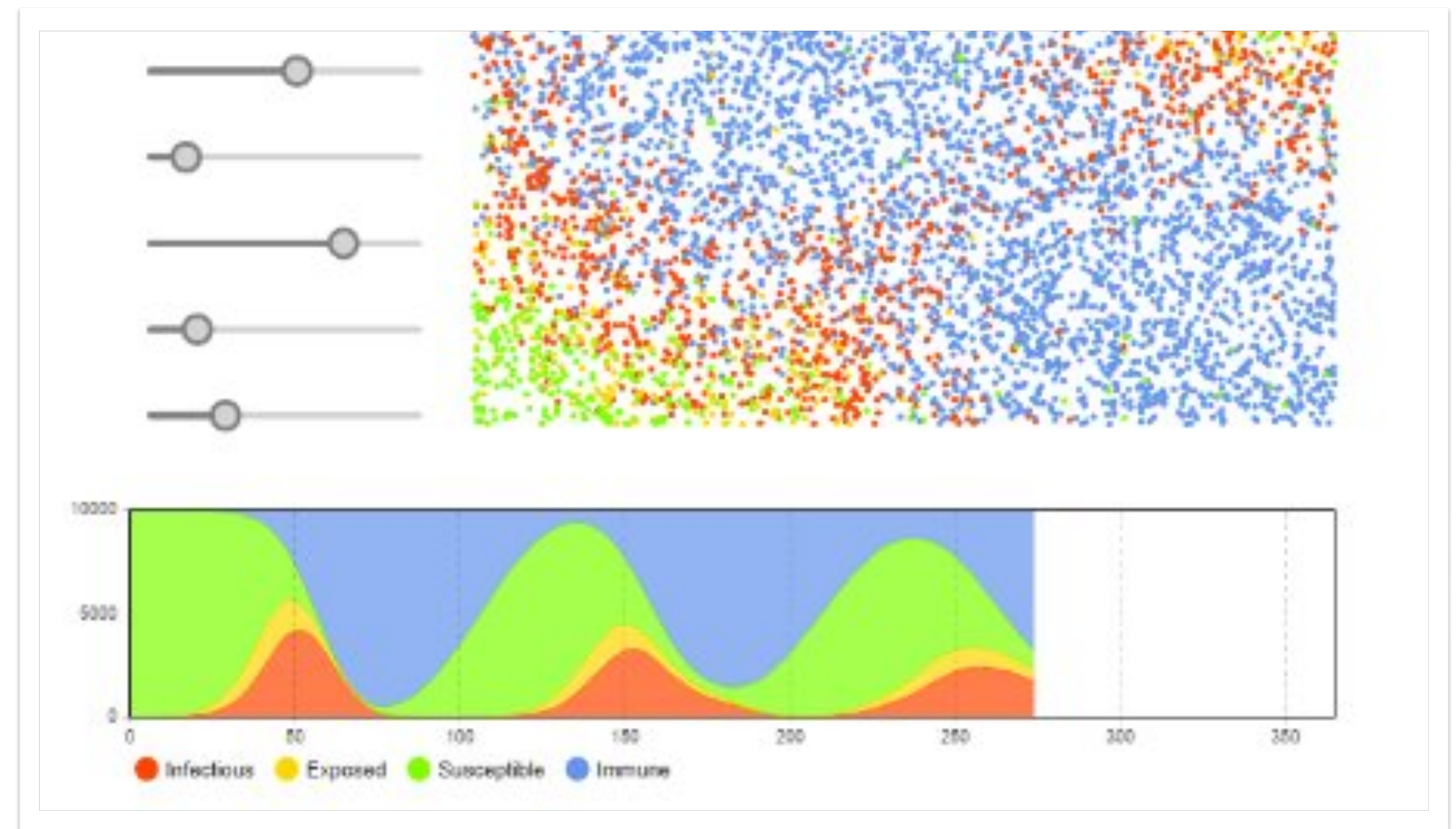
Modellen doorrekenen

- Naast **data** speelt de computer ook een belangrijke rol in het doorrekenen van (voorspellende) modellen
- Wat is het huidige risico van een ziekteuitbraak?



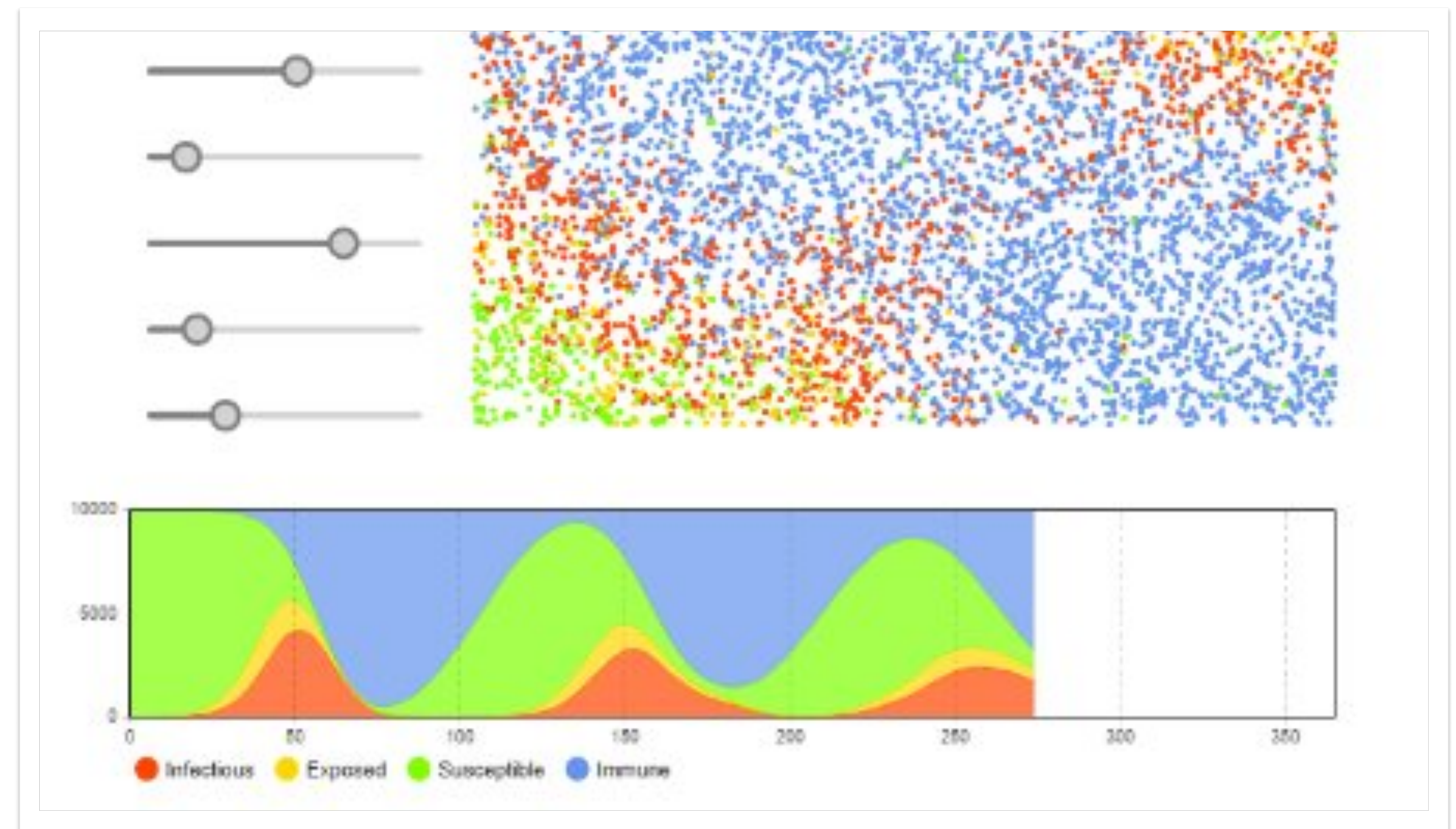
Modellen doorrekenen

- Naast **data** speelt de computer ook een belangrijke rol in het doorrekenen van (voorspellende) modellen
- Wat is het huidige risico van een ziekteuitbraak?
- Hoe snel verloopt de ziekteuitbraak?



Modellen doorrekenen

- Naast **data** speelt de computer ook een belangrijke rol in het doorrekenen van (voorspellende) modellen
- Wat is het huidige risico van een ziekteuitbraak?
- Hoe snel verloopt de ziekteuitbraak?
- Welke methoden werken het beste om een ziekteuitbraak te voorkomen



Rstudio



Rstudio

- Rstudio is een zogeheten IDE (*integrated development environment*), een programma waarin je snel met R aan de slag kan.



Installatie van R en Rstudio

1.2 R zelf installeren

R kun je eenvoudig installeren. Het is te vinden op de [CRAN website](#), en de instructies voor Mac en Windows gebruikers zijn daar ook te vinden:

- Windows-gebruikers: ga naar "Download R for Windows", klik op de link "base" en vervolgens op de link "Download R x.x.x for Windows" om het installatiebestand te downloaden. Open het gedownloadde `.exe` bestand en volg de instructies om R te installeren.
- Mac-gebruikers: "Download R for (Mac) OS X". Download het juiste installatiebestand voor jouw macOS-versie (let op: als je een oudere Mac gebruikt moet je de "Intel Mac" versie downloaden). Open het `.pkg` bestand en volg de instructies.

1.3 R-studio installeren

Nadat je R geïnstalleerd hebt, kun je Rstudio installeren. Deze downloaden we vanaf de [website van Rstudio](#). Deze website legt ook uit dat je wel eerst R moet installeren, maar dat hebben we al gedaan. We kunnen dus meteen naar stap 2: Install Rstudio. De website herkent vanzelf welk besturingssysteem je gebruikt, dus hoeft je het alleen nog maar te installeren:

- Windows-gebruikers: volg de instructies van de installatie-wizard
- Mac-gebruiker: open de `.dmg` en sleep "Rstudio" naar je Applications folder

Nu kun je Rstudio opstarten, en zou alles moeten werken¹

SCRIPT

```

1 library(nycflights13) ## package containing flights dataset
2 library(lubridate)
3 library(dplyr)
4 library(ggplot2)
5
6 head(flights, n = 3)
7 daily <- flights %>%
8   mutate(date = make_date(year, month, day)) %>%
9   count(date) %>%
10  mutate(wday = wday(date, label = TRUE))
11 head(daily, n = 3)
12 ggplot(daily, aes(wday, n)) +
13   geom_boxplot(outlier.colour = "hotpink") +
14   labs(x = "Weekday", y = "Flights",
15        subtitle = "Number of 2013 New York Flights Each Weekday")
16
1:1 (Top Level)

```

ENVIRONMENT

Global Environment

Data

daily 365 obs. of 3 variables

- \$ date: Date[1:365], format: "2013-01-01" "2013-01-02" ..
- \$ n : int [1:365] 842 943 914 915 720 832 933 899 902..
- \$ wday: Ord.factor w/ 7 levels "Sun"<"Mon"<"Tue"<...: 3 ..

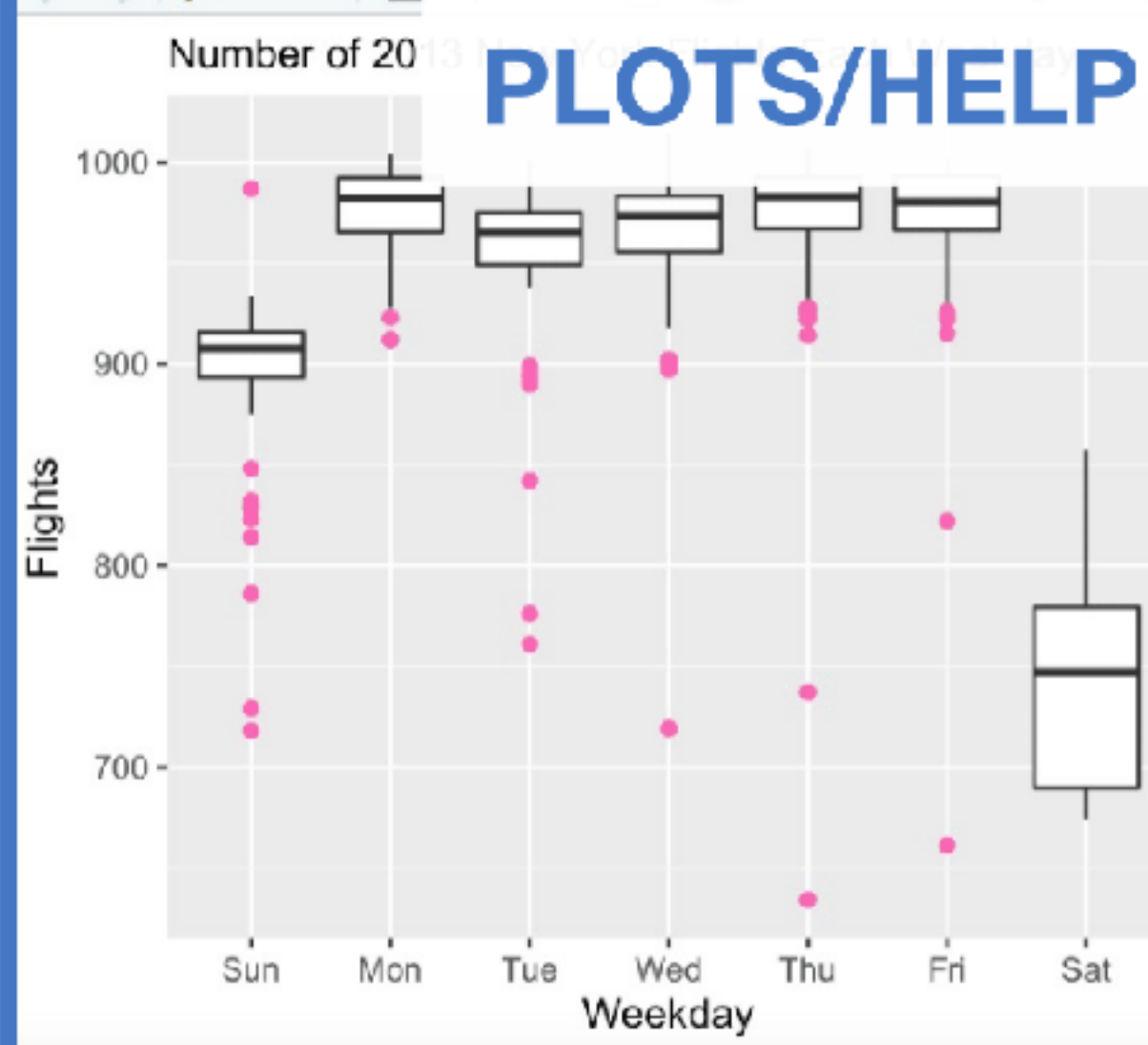
CONSOLE

```

# A tibble: 3 x 19
  year month   day dep_time sched_dep_time dep_delay arr_time sched_a
<int> <int> <int>   <int>         <int>         <dbl>   <int>
1  2013     1     1     517             515           2     830
2  2013     1     1     533             529           4     850
3  2013     1     1     542             540           2     923
# ... with 9 more variables: flight <int>, tailnum <chr>, origin <chr>, dest <chr>, air_time <dbl>,
# distance <dbl>, hour <dbl>, minute <dbl>, time_hour <dtm>
> daily <- flights %>%
+   mutate(date = make_date(year, month, day)) %>%
+   count(date) %>%
+   mutate(wday = wday(date, label = TRUE))
> head(daily, n = 3)
# A tibble: 3 x 3
  date           n wday
<date>   <int> <ord>
1 2013-01-01   842 Tue
2 2013-01-02   943 Wed
3 2013-01-03   914 Thu
> ggplot(daily, aes(wday, n)) +
+   geom_boxplot(outlier.colour = "hotpink") +
+   labs(x = "Weekday", y = "Flights",
+        subtitle = "Number of 2013 New York Flights Each Weekday")
>

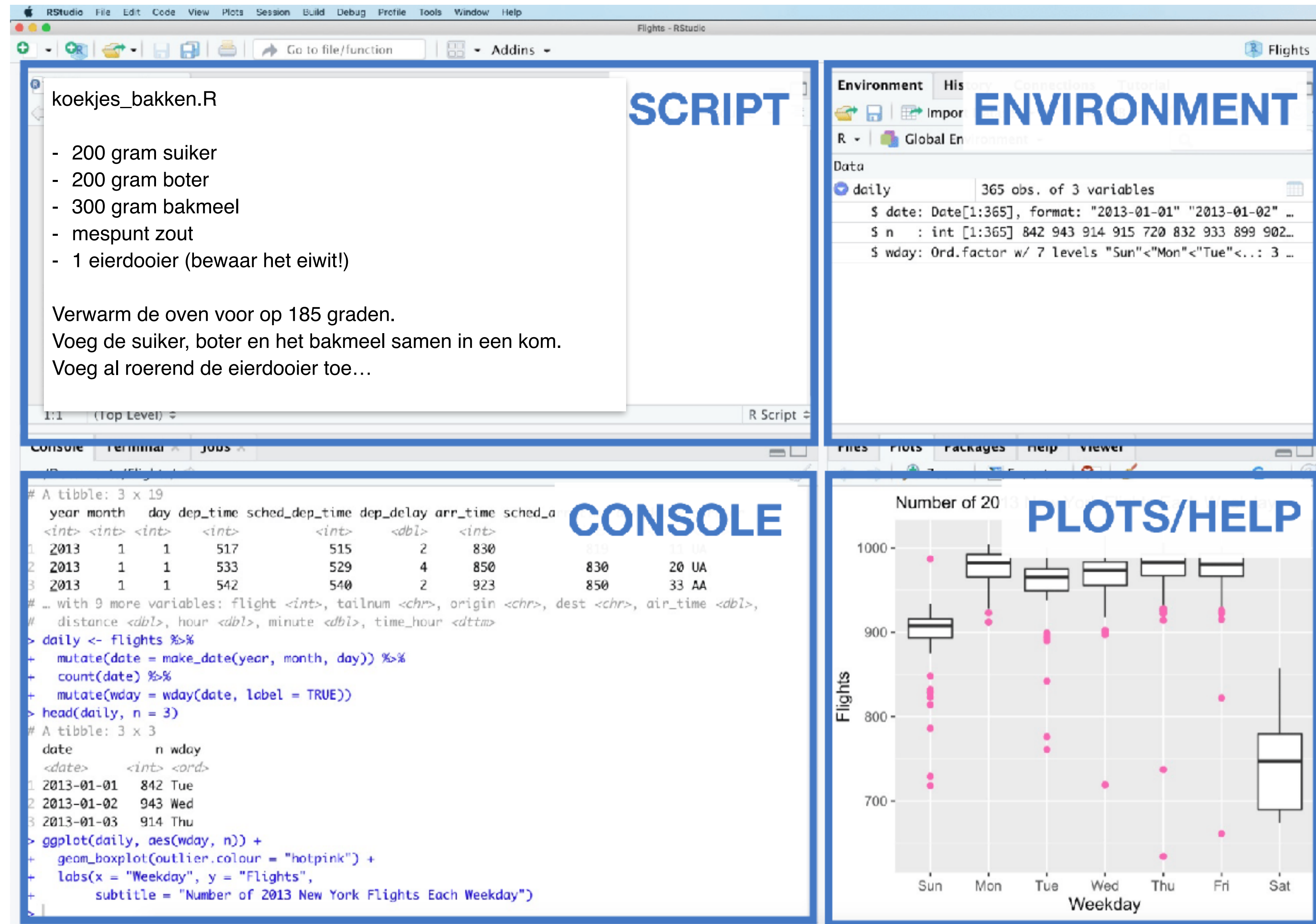
```

PLOTS/HELP



Number of 2013 New York Flights Each Weekday

Weekday	Median	Q1	Q3	Min	Max
Sun	~900	~890	~910	~880	~920
Mon	~950	~940	~960	~930	~970
Tue	~940	~930	~950	~920	~960
Wed	~940	~930	~950	~920	~960
Thu	~940	~930	~950	~920	~960
Fri	~940	~930	~950	~920	~960
Sat	~750	~690	~780	~680	~850



SCRIPT

```
koekjes_bakken.R
```

- 200 gram suiker
- 200 gram boter
- 300 gram bakmeel
- mespunt zout
- 1 eierdooier (bewaar het eiwit!)

Verwarm de oven voor op 185 graden.
Voeg de suiker, boter en het bakmeel samen in een kom.
Voeg al roerend de eierdooier toe...

ENVIRONMENT

Global Environment

Data

daily 365 obs. of 3 variables

\$ date: Date[1:365], format: "2013-01-01" "2013-01-02" ...

\$ n : int [1:365] 842 943 914 915 720 832 933 899 902...

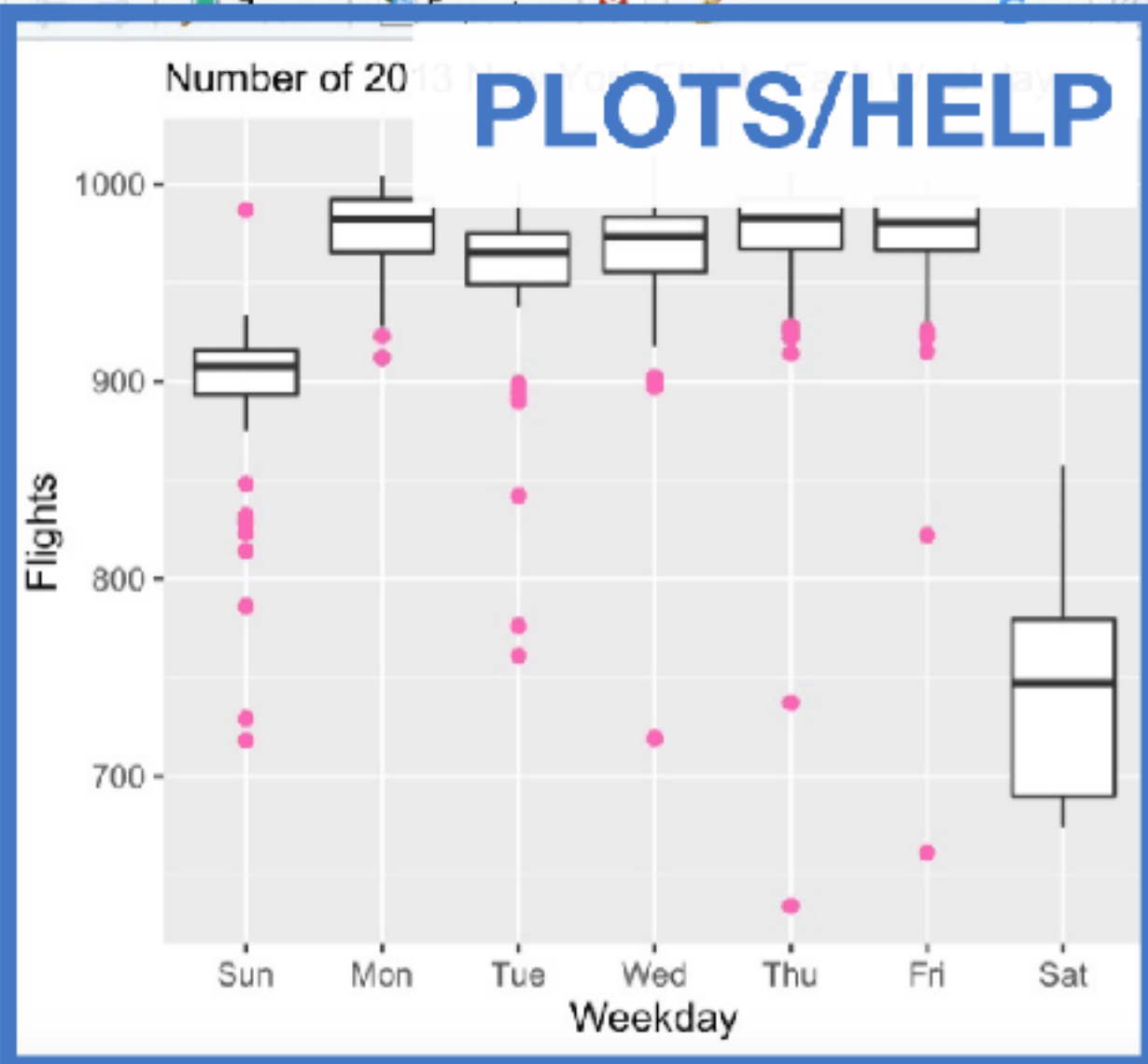
\$ wday: Ord.factor w/ 7 levels "Sun"<"Mon"<"Tue"<...: 3 ...

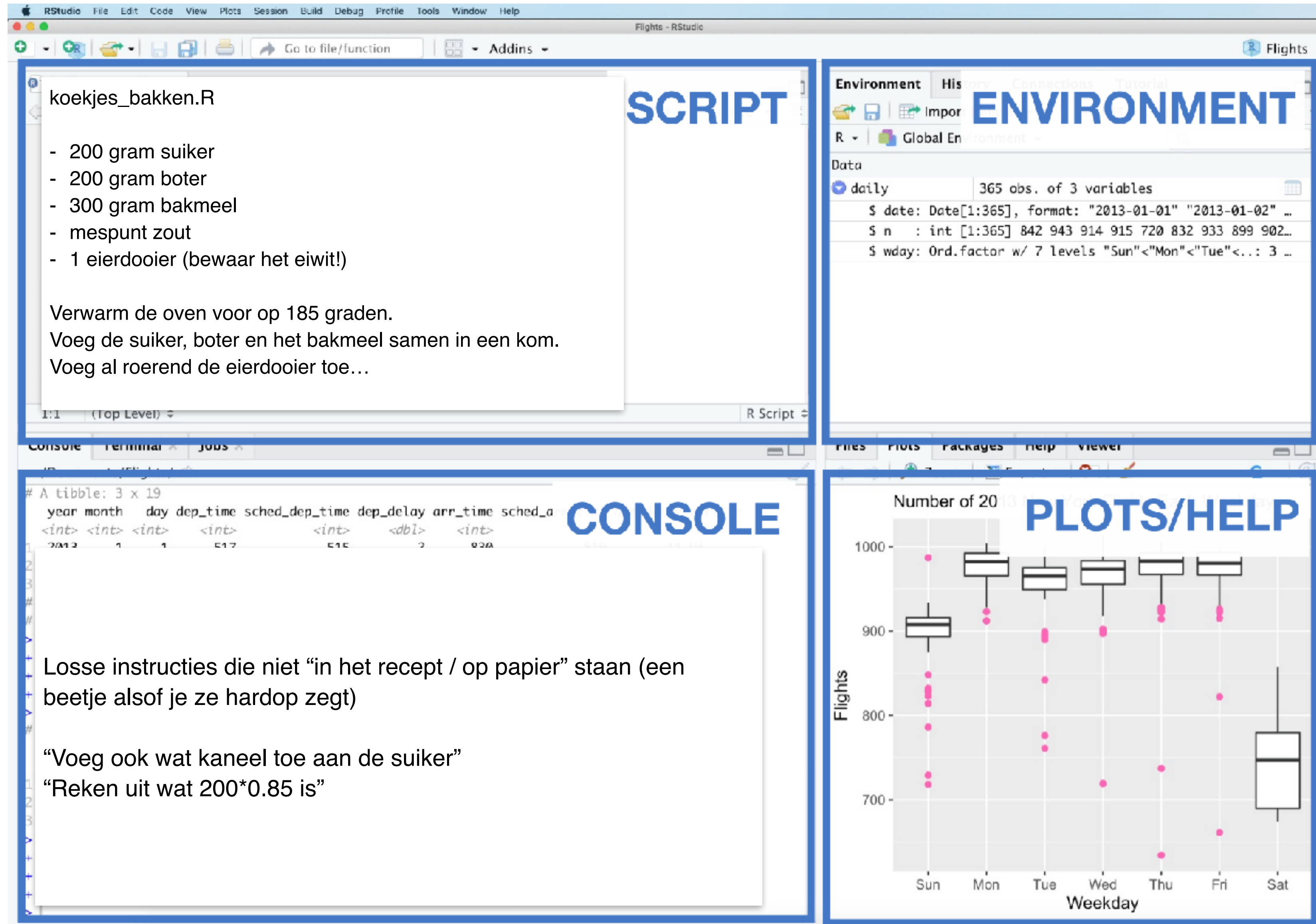
CONSOLE

```
# A tibble: 3 x 19
  year month   day dep_time sched_dep_time dep_delay arr_time sched_a
  <int> <int> <int>   <int>         <int>         <dbl>   <int>
1  2013     1     1     517             515           2     830
2  2013     1     1     533             529           4     850
3  2013     1     1     542             540           2     923
# ... with 9 more variables: flight <int>, tailnum <chr>, origin <chr>, dest <chr>, air_time <dbl>,
# distance <dbl>, hour <dbl>, minute <dbl>, time_hour <dtm>
> daily <- flights %>%
+   mutate(date = make_date(year, month, day)) %>%
+   count(date) %>%
+   mutate(wday = wday(date, label = TRUE))
> head(daily, n = 3)
# A tibble: 3 x 3
  date           n wday
  <date>     <int> <ord>
1 2013-01-01   842 Tue
2 2013-01-02   943 Wed
3 2013-01-03   914 Thu
> ggplot(daily, aes(wday, n)) +
+   geom_boxplot(outlier.colour = "hotpink") +
+   labs(x = "Weekday", y = "Flights",
+        subtitle = "Number of 2013 New York Flights Each Weekday")
>
```

PLOTS/HELP

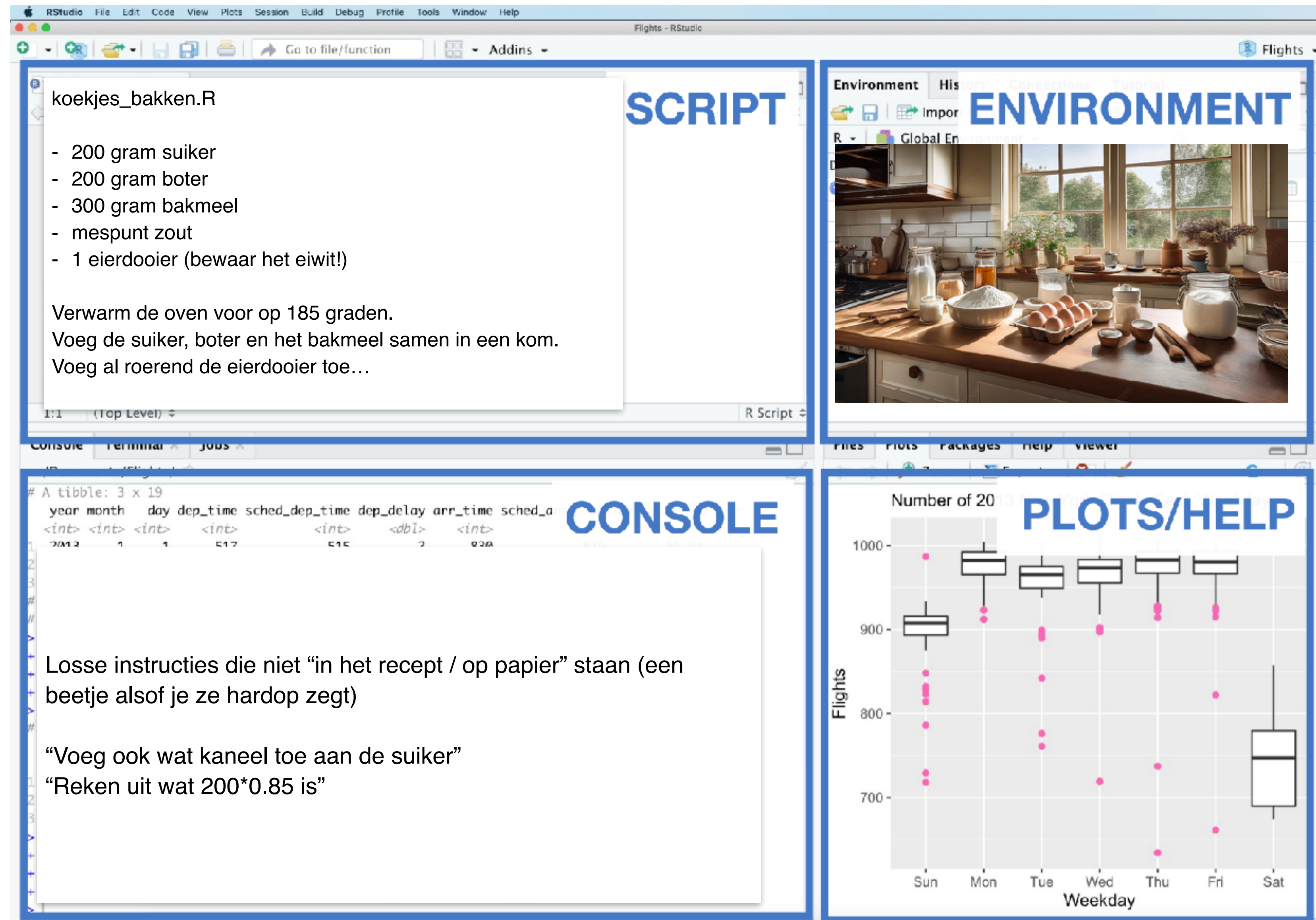
Number of 2013 New York Flights Each Weekday





The screenshot shows the RStudio interface with four panels highlighted by blue borders:

- SCRIPT:** Contains a file named `koekjes_bakken.R` with a list of ingredients and instructions.
 - Ingredients:
 - 200 gram suiker
 - 200 gram boter
 - 300 gram bakmeel
 - mespunt zout
 - 1 eierdooier (bewaar het eiwit!)
 - Instructions:
 - Verwarm de oven voor op 185 graden.
 - Voeg de suiker, boter en het bakmeel samen in een kom.
 - Voeg al roerend de eierdooier toe...
- ENVIRONMENT:** Shows the `daily` data frame with 365 observations and 3 variables: `date`, `n`, and `wday`.
- CONSOLE:** Displays the output of a command, showing a tibble with 3 rows and 19 columns. Below the output, there are instructions:
 - Losse instructies die niet “in het recept / op papier” staan (een beetje alsof je ze hardop zegt)
 - “Voeg ook wat kaneel toe aan de suiker”
 - “Reken uit wat 200×0.85 is”
- PLOTS/HELP:** Displays a box plot titled "Number of 2013 Flights" showing the distribution of flights by weekday. The y-axis is labeled "Flights" and ranges from 700 to 1000. The x-axis is labeled "Weekday" and includes Sun, Mon, Tue, Wed, Thu, Fri, and Sat. The plot shows that Saturday has the lowest number of flights, while the other days have higher medians and more variability.



SCRIPT

koekjes_bakken.R

- 200 gram suiker
- 200 gram boter
- 300 gram bakmeel
- mespunt zout
- 1 eierdooier (bewaars het eiwit!)

Verwarm de oven voor op 185 graden.
Voeg de suiker, boter en het bakmeel samen in een kom.
Voeg al roerend de eierdooier toe...

ENVIRONMENT

Global Environment



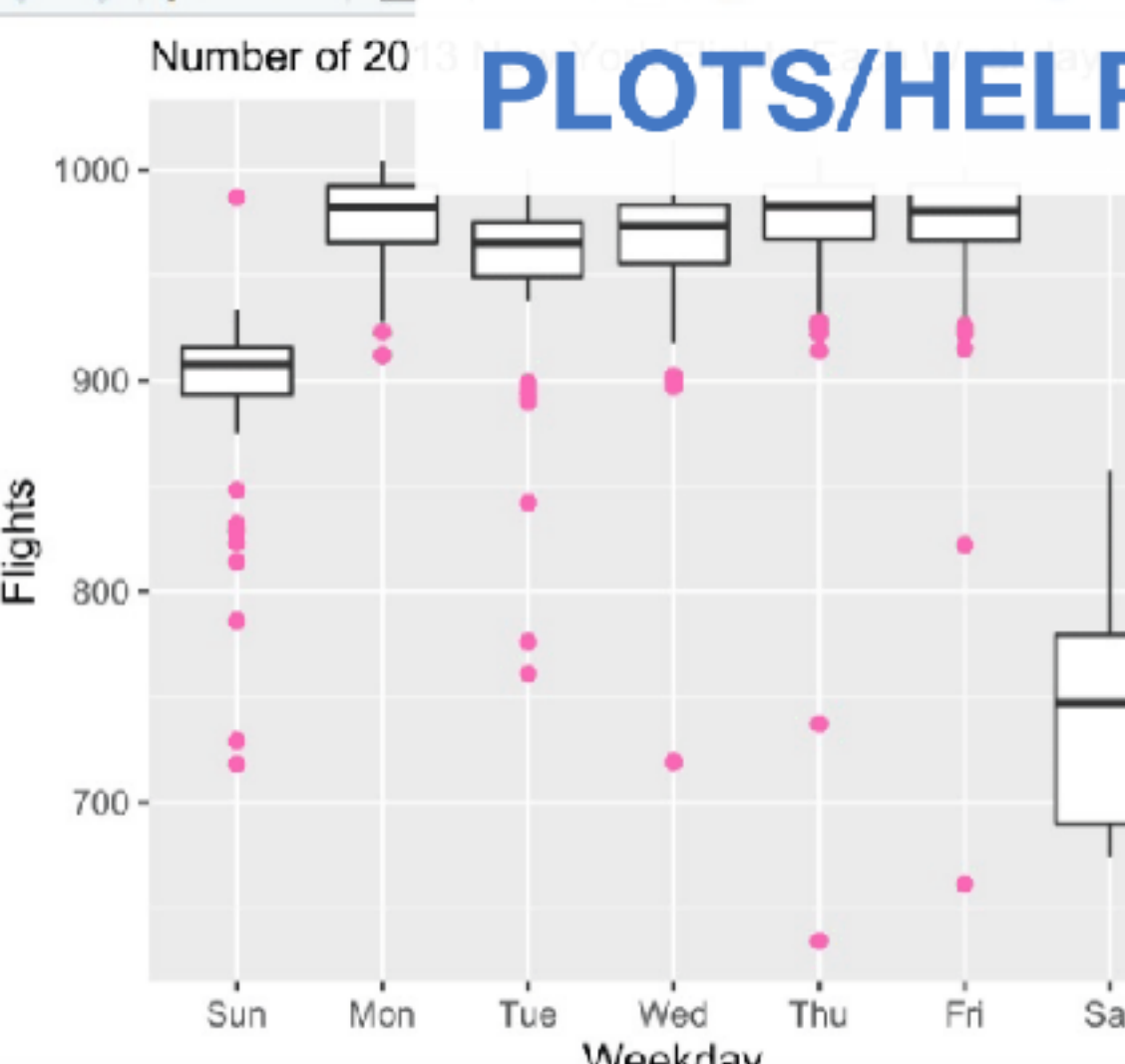
CONSOLE

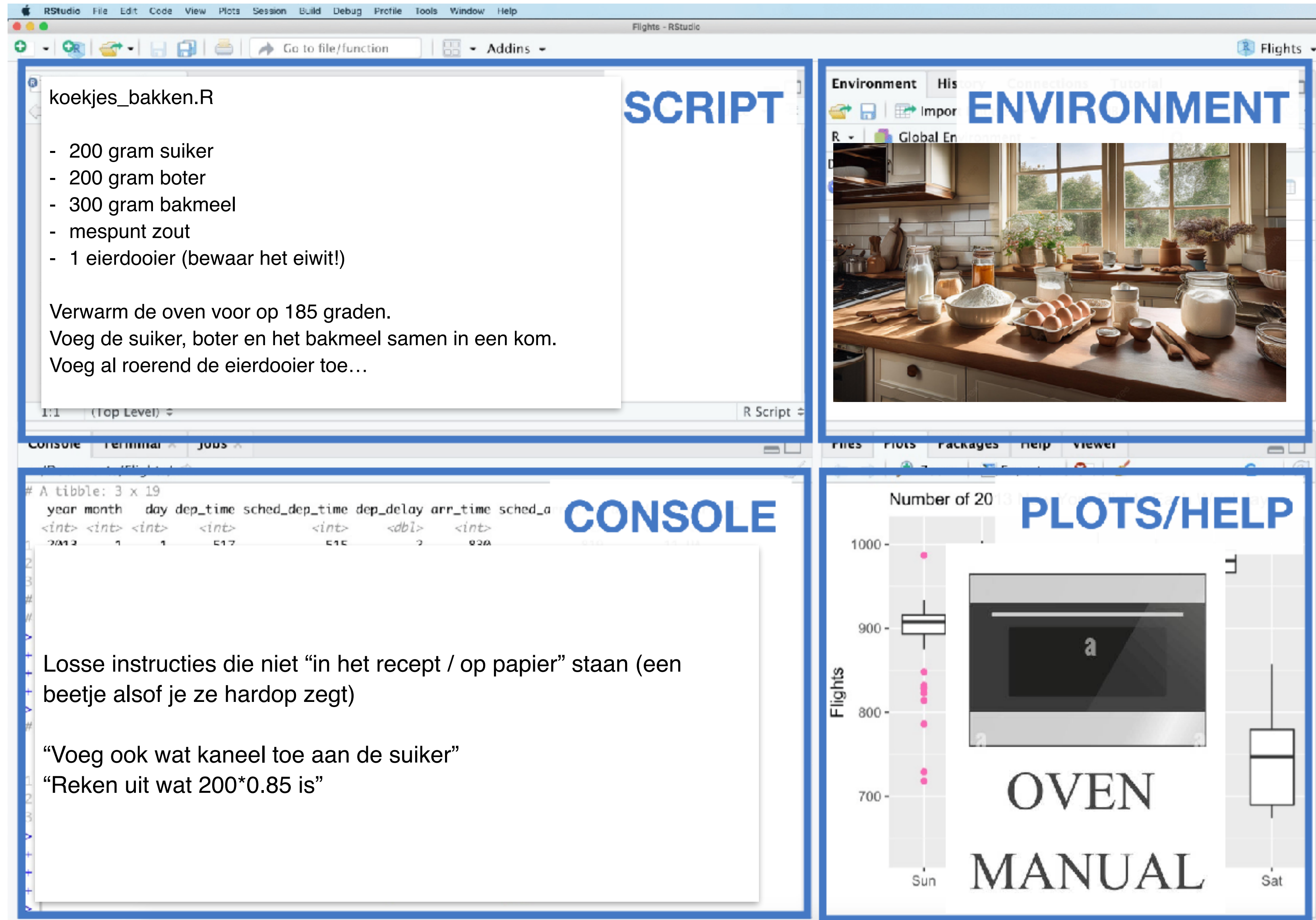
```
# A tibble: 3 x 19
  year month   day dep_time sched_dep_time dep_delay arr_time sched_a
  <int> <int> <int>   <int>         <int>         <dbl>   <int>   <int>
1 2013     1     1     517             515         2      820
2
3
#
#
```

Losse instructies die niet “in het recept / op papier” staan (een beetje alsof je ze hardop zegt)

“Voeg ook wat kaneel toe aan de suiker”
“Reken uit wat 200×0.85 is”

PLOTS/HELP

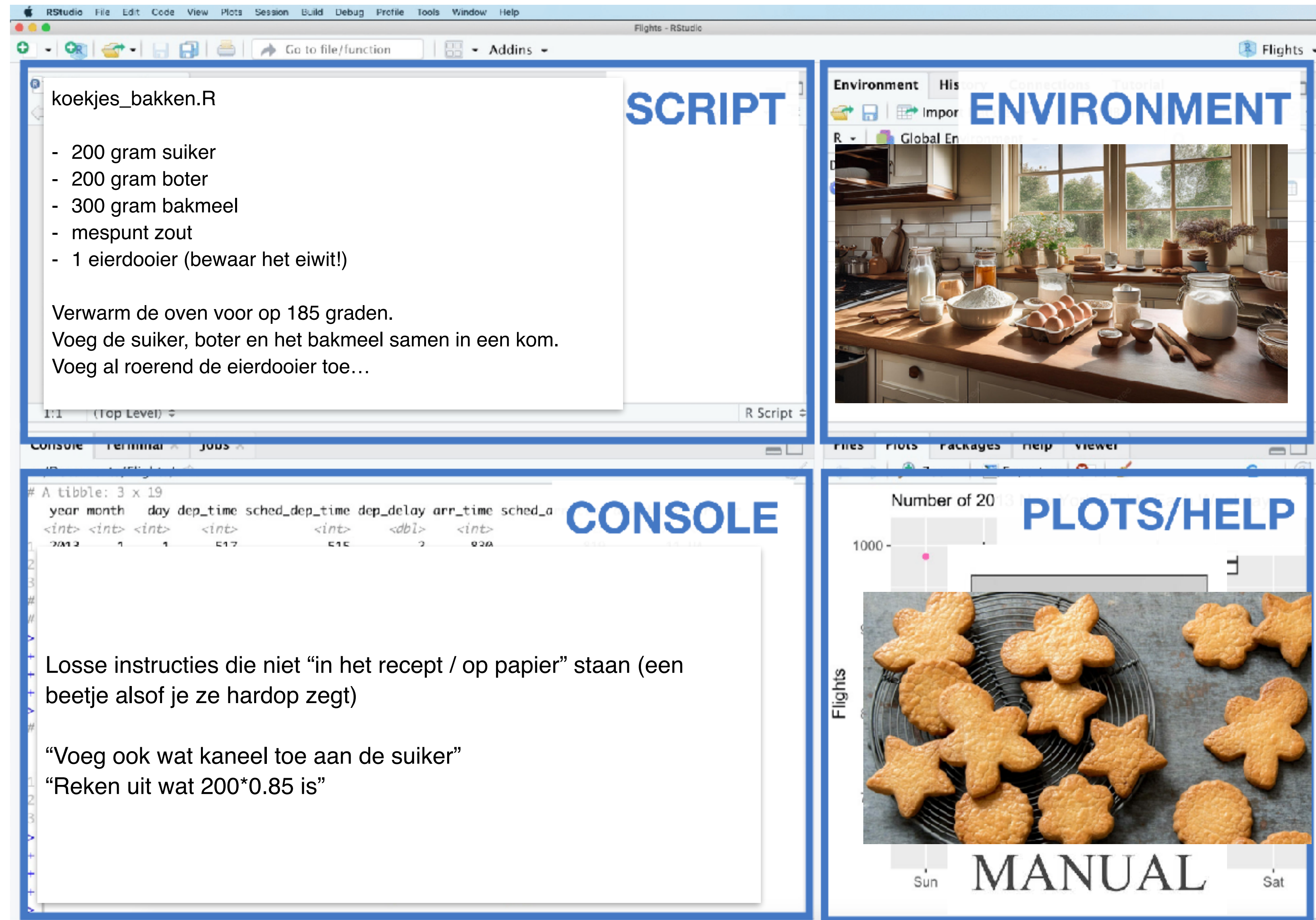




The screenshot shows the RStudio interface with four panels highlighted by blue borders:

- SCRIPT:** Contains a file named `koekjes_bakken.R` with a list of ingredients and instructions for baking cookies.
 - 200 gram suiker
 - 200 gram boter
 - 300 gram bakmeel
 - mespunt zout
 - 1 eierdooier (bewaar het eiwit!)

Verwarm de oven voor op 185 graden.
Voeg de suiker, boter en het bakmeel samen in een kom.
Voeg al roerend de eierdooier toe...
- ENVIRONMENT:** Displays a photograph of a kitchen counter with various baking ingredients like flour, eggs, and butter.
- CONSOLE:** Shows the output of an R command, displaying a tibble with 3 rows and 19 columns. Below the tibble, there are instructions:
 - Losse instructies die niet “in het recept / op papier” staan (een beetje alsof je ze hardop zegt)
 - “Voeg ook wat kaneel toe aan de suiker”
 - “Reken uit wat 200×0.85 is”
- PLOTS/HELP:** Displays a box plot titled "Number of 20" showing the distribution of flights for Sunday (Sun) and Saturday (Sat). The y-axis is labeled "Flights" and ranges from 700 to 1000. The plot shows that Sunday has a higher median number of flights than Saturday. Below the plot, the text "OVEN MANUAL" is displayed.



SCRIPT

koekjes_bakken.R

- 200 gram suiker
- 200 gram boter
- 300 gram bakmeel
- mespunt zout
- 1 eierdooier (bewaar het eiwit!)

Verwarm de oven voor op 185 graden.
Voeg de suiker, boter en het bakmeel samen in een kom.
Voeg al roerend de eierdooier toe...

ENVIRONMENT

Environment History Connections Tutorial
R Global Environment

CONSOLE

```
# A tibble: 3 x 19
  year month   day dep_time sched_dep_time dep_delay arr_time sched_a
  <int> <int> <int>   <int>         <int>      <dbl>   <int>
1 2013     1     1      517             515        2      520
```

Losse instructies die niet “in het recept / op papier” staan (een beetje alsof je ze hardop zegt)

“Voeg ook wat kaneel toe aan de suiker”
“Reken uit wat 200×0.85 is”

PLOTS/HELP

Number of 20
1000

Flights

Sun **MANUAL** Sat

Demonstratie van een superhandige rekenmachine



Comments in code

```
# a, b, en c waarden toekennen (assigning variables)
```

```
a <- 1
```

```
b <- 3
```

```
c <- 2.5
```

```
# rekenen met a, b, en c
```

```
a+b-c # a plus b min c
```

```
a*b # a keer b
```

```
a/b # a gedeeld door b
```

```
b*b # b keer b
```

```
b^3 # b tot de macht 3
```

Cursusboek: code vs. resultaten

```
# definieer een waarde voor x en y
x <- 10
y <- 9
# Met 'cat' kunnen we tekst en berekeningen aan elkaar plakken
cat("De som van x en y is:", x+y, "en x min y is:", x-y, "\n... einde berekening!")
```

De resultaten van code worden zo weergegeven:

```
De som van x en y is: 19 en x min y is: 1
... einde berekening!
```

Soms staat er ook een stukje code midden in een regel. Dat ziet er zo uit.

Werkcollege (groepen)

Groep	Studentassistent
1	Mees
2	Marit
3	Noémi
4	Nikita
5	Wouter
6	Maarten S.
7	Amber
8	Vince (5wk) / Maarten W. (5 wk)
9	Luuk
10	Madelon
11	Floor
12	Stef
13	Oscar
14	Herm
15	Sofie

Werkcollege (groepen)

Groep	Studentassistent
1	Mees
2	Marit
3	Noémi
4	Nikita
5	Wouter
6	Maarten S.
7	Amber
8	Vince (5wk) / Maarten W. (5 wk)
9	Luuk
10	Madelon
11	Floor
12	Stef
13	Oscar
14	Herm
15	Sofie

- **Inzet tijdens werkcolleges doet je cijfer goed!**

Werkcollege (groepen)

Groep	Studentassistent
1	Mees
2	Marit
3	Noémi
4	Nikita
5	Wouter
6	Maarten S.
7	Amber
8	Vince (5wk) / Maarten W. (5 wk)
9	Luuk
10	Madelon
11	Floor
12	Stef
13	Oscar
14	Herm
15	Sofie

- **Inzet tijdens werkcolleges doet je cijfer goed!**
- Zorg dat je wordt afgetekend!

Werkcollege (groepen)

Groep	Studentassistent
1	Mees
2	Marit
3	Noémi
4	Nikita
5	Wouter
6	Maarten S.
7	Amber
8	Vince (5wk) / Maarten W. (5 wk)
9	Luuk
10	Madelon
11	Floor
12	Stef
13	Oscar
14	Herm
15	Sofie

- **Inzet tijdens werkcolleges doet je cijfer goed!**
- Zorg dat je wordt afgetekend!
- Ikzelf probeer een beetje rond te vliegen over de Uithof, maar ik kan niet alle zalen aantikken (Dalton en Bol liggen ver van elkaar af...)

Werkcollege: Rstudio installeren en een beetje rekenen

Werkcollege: Rstudio installeren en een beetje rekenen

Oefening 1.1 (Rekenen met R)



Maak een nieuw (leeg) Rscript in Rstudio, en voer de volgende berekeningen uit:

- Definieer $a=2$, en reken uit wat $a \cdot a \cdot a = a^3$ is.
- Definieer nu $b=9.5$ en reken uit wat b^{19} is (let op: je hoeft niet 19 keer b te typen)
- Stel je zet eerst 3 stappen naar links (laten we deze afstand a noemen), dan 4 stappen naar voren (laten we deze b noemen), en loopt vervolgens in een rechte lijn terug naar je beginpositie. Schrijf een script met R die uitrekent hoe ver je in totaal hebt gelopen. Zorg dat de uitkomst van je berekening **niet** in de environment staat, maar slechts in de console wordt weergegeven.
- Bereken nu uit hoever je gelopen zou hebben als je in plaats hiervan 5 stappen naar links en 6 stappen naar boven had gezet. Zorg dit keer dat de uitkomst **wel** in de environment staat onder de naam "afstand". Voeg de onderstaande regel code toe aan het eind van je script (hint: er is een knop om de code te kopiëren!). Voer je script uit. Hoeveel stappen heb je gelopen? Sla je script op onder de naam **Pythagoras.R**.

```
print(paste('Je hebt', afstand, 'stappen gelopen'))
```

Werkcollege: Rstudio installeren en een beetje rekenen

Oefening 1.1 (Rekenen met R)



Maak een nieuw (leeg) Rscript in Rstudio, en voer de volgende berekeningen uit:

- Definieer $a=2$, en reken uit wat $a \cdot a \cdot a = a^3$ is.
- Definieer nu $b=9.5$ en reken uit wat b^{19} is (let op: je hoeft niet 19 keer b te typen)
- Stel je zet eerst 3 stappen naar links (laten we deze afstand a noemen), dan 4 stappen naar voren (laten we deze b noemen), en loopt vervolgens in een rechte lijn terug naar je beginpositie. Schrijf een script met R die uitrekent hoe ver je in totaal hebt gelopen. Zorg dat de uitkomst van je berekening **niet** in de environment staat, maar slechts in de console wordt weergegeven.
- Bereken nu uit hoever je gelopen zou hebben als je in plaats hiervan 5 stappen naar links en 6 stappen naar boven had gezet. Zorg dit keer dat de uitkomst **wel** in de environment staat onder de naam "afstand". Voeg de onderstaande regel code toe aan het eind van je script (hint: er is een knop om de code te kopiëren!). Voer je script uit. Hoeveel stappen heb je gelopen? Sla je script op onder de naam **Pythagoras.R**.

```
print(paste('Je hebt', afstand, 'stappen gelopen'))
```

Werkcollege: Rstudio installeren en een beetje rekenen

Oefening 1.1 (Rekenen met R)

Maak een nieuw (leeg) Rscript in Rstudio, en voer de volgende berekeningen uit:

- a. Definieer $a=2$, en reken
- b. Definieer nu $b=9.5$ en
- c. Stel je zet eerst 3 stappen voren (laten we deze i Schrijf een script met je berekening **niet** in c
- d. Bereken nu uit hoever stappen naar boven h de naam "afstand". Vo een knop om de code script op onder de na

```
print(paste('Je hebt'
```



Oefening 4.1

Los de volgende vergelijkingen op:

a.

$$3x + 5 = 11$$

b.

$$x^2 + 2x - 8 = 0$$

c.

$$x^2 - 9 = 0$$



Werkcollege: Rstudio installeren en een beetje rekenen

Oefening 1.1 (Rekenen met R)

Maak een nieuw (leeg) Rscript in Rstudio, en voer de volgende berekeningen uit:

- Definieer $a=2$, en reken
- Definieer nu $b=9.5$ en
- Stel je zet eerst 3 stappen voren (laten we deze i Schrijf een script met je berekening **niet** in c
- Bereken nu uit hoever stappen naar boven h de naam "afstand". Vo een knop om de code script op onder de na

```
print(paste('Je hebt'
```

Oefening 4.1

Los de volgende vergelijkingen op:

a.

$$3x + 5 = 11$$

b.

$$x^2 + 2x - 8 =$$

c.

$$x^2 - 9 = 0$$

Oefening 15.4 (Cheaters en cooperators)

In hoofdstuk [Hoofdstuk 13](#) heb je kennis gemaakt met *individual based modelling*, en heb je gezien hoe cooperators en cheaters kunnen samenleven. Echter zagen we ook dat als het systeem te sterk gemengd werd, dat minder goed ging. Sterker nog, onder goed gemengde condities ging het eigenlijk helemaal mis, omdat de cheaters teveel toenamen en daarna het hele systeem uitstierf. Laten we kijken hoe dit uitpakt in een ODE die vergelijkbare aannames doet. Bekijk het volgende ODE model:

$$\begin{aligned}\frac{dC_1}{dt} &= b_1 C_1^2 \left(1 - \frac{C_1 + C_2}{K}\right) - dC_1 \\ \frac{dC_2}{dt} &= b_2 C_1 C_2 \left(1 - \frac{C_1 + C_2}{K}\right) - dC_2\end{aligned}$$

Anders dan in het individual-based model, waar we twee cooperators en één cheater hadden,

Werkcollege: Rstudio installeren en een beetje rekenen

Oefening 1.1 (Rekenen met R)

Maak een nieuw (leeg) Rscript in Rstudio, en voer de volgende berekeningen uit:

- a. Definieer $a=2$, en reken
- b. Definieer nu $b=9.5$ en
- c. Stel je zet eerst 3 stappen voor (laten we deze 3 stappen voorstellen). Schrijf een script met je berekening **niet** in code, maar in een commentaar. Bereken nu uit hoeveel stappen naar boven h de naam "afstand". Voer een knop om de code te runnen op onder de naam
- d. Bereken nu uit hoeveel stappen naar boven h de naam "afstand". Voer een knop om de code te runnen op onder de naam

```
print(paste('Je hebt'
```

Oefening 4.1

Los de volgende vergelijkingen op:

a.

$$3x + 5 = 11$$

b.

$$x^2 + 2x - 8 = 0$$

c.

$$x^2 - 9 = 0$$

Oefening 15.4 (Cheaters en cooperators)

In hoofdstuk [Hoofdstuk 13](#) heb je kennis gemaakt met *individual based modelling*, en heb je gezien hoe cooperators en defectors gemengd werken. In dit hoofdstuk gaan we kijken hoe het eigenlijk helerom werkt. Laten we kijken naar het model:

Anders dan in

Oefening 10.1 (Een toename van slechts 5%)

We plotten het resultaat van de loop hierboven op een log-lineaire schaal (zie [Hoofdstuk 6](#)), zodat we goed kunnen zien of de groei exponentiële is. Laten we nu aannemen dat de groei van het aantal rijstkorrels niet verdubbelt, maar met 5 procent toeneemt per hokje op het schaakbord:

$$R(t+1) = 1.05 \cdot R(t)$$

Pas de R-code hierboven aan om deze nieuwe differentievergelijking door te rekenen.

a. Is er sprake van exponentiele groei?

b. Hoeveel rijstkorrels heb je nu in totaal? (hint: je kan de `sum` functie gebruiken om de som van een vector in één keer uit te rekenen!)

Morgen: écht programmeren met R