

Data visualization

Quiz

- What do you find interesting in today's VotW?
- What is data dictionary and why do we need it?
- Explain the differences between nominal and ordinal data types with examples.

Tidy Data

What is
Tidy Data?

What is tidy data?

- Each variable forms a column.
- Each observation forms a row.
- Each type of observational unit forms a table.

Variables? Observations?

	treatmenta	treatmentb
John Smith	—	2
Jane Doe	16	11
Mary Johnson	3	1

	John Smith	Jane Doe	Mary Johnson
treatmenta	—	16	3
treatmentb	2	11	1

Variables

Observations

person	treatment	result
John Smith	a	—
Jane Doe	a	16
Mary Johnson	a	3
John Smith	b	2
Jane Doe	b	11
Mary Johnson	b	1

Tidy or not? Why? Can you make it tidy if not?

religion	<\$10k	\$10–20k	\$20–30k	\$30–40k	\$40–50k	\$50–75k
Agnostic	27	34	60	81	76	137
Atheist	12	27	37	52	35	70
Buddhist	27	21	30	34	33	58
Catholic	418	617	732	670	638	1116
Don't know/refused	15	14	15	11	10	35
Evangelical Prot	575	869	1064	982	881	1486
Hindu	1	9	7	9	11	34
Historically Black Prot	228	244	236	238	197	223
Jehovah's Witness	20	27	24	24	21	30
Jewish	19	19	25	25	30	95

religion	income	freq
Agnostic	<\$10k	27
Agnostic	\$10–20k	34
Agnostic	\$20–30k	60
Agnostic	\$30–40k	81
Agnostic	\$40–50k	76
Agnostic	\$50–75k	137
Agnostic	\$75–100k	122
Agnostic	\$100–150k	109
Agnostic	>150k	84
Agnostic	Don't know/refused	96

Tidy or not? Why? Can you make it tidy if not?

year	artist	track	time	date.entered	wk1	wk2	wk3
2000	2 Pac	Baby Don't Cry	4:22	2000-02-26	87	82	72
2000	2Ge+her	The Hardest Part Of ...	3:15	2000-09-02	91	87	92
2000	3 Doors Down	Kryptonite	3:53	2000-04-08	81	70	68
2000	98~0	Give Me Just One Nig...	3:24	2000-08-19	51	39	34
2000	A*Teens	Dancing Queen	3:44	2000-07-08	97	97	96
2000	Aaliyah	I Don't Wanna	4:15	2000-01-29	84	62	51
2000	Aaliyah	Try Again	4:03	2000-03-18	59	53	38
2000	Adams, Yolanda	Open My Heart	5:30	2000-08-26	76	76	74

Table 7: The first eight Billboard top hits for 2000. Other columns not shown are `wk4`, `wk5`, ..., `wk75`.

year	artist	time	track	date	week	rank
2000	2 Pac	4:22	Baby Don't Cry	2000-02-26	1	87
2000	2 Pac	4:22	Baby Don't Cry	2000-03-04	2	82
2000	2 Pac	4:22	Baby Don't Cry	2000-03-11	3	72
2000	2 Pac	4:22	Baby Don't Cry	2000-03-18	4	77
2000	2 Pac	4:22	Baby Don't Cry	2000-03-25	5	87
2000	2 Pac	4:22	Baby Don't Cry	2000-04-01	6	94
2000	2 Pac	4:22	Baby Don't Cry	2000-04-08	7	99
2000	2Ge+her	3:15	The Hardest Part Of ...	2000-09-02	1	91
2000	2Ge+her	3:15	The Hardest Part Of ...	2000-09-09	2	87
2000	2Ge+her	3:15	The Hardest Part Of ...	2000-09-16	3	92
2000	3 Doors Down	3:53	Kryptonite	2000-04-08	1	81
2000	3 Doors Down	3:53	Kryptonite	2000-04-15	2	70
2000	3 Doors Down	3:53	Kryptonite	2000-04-22	3	68
2000	3 Doors Down	3:53	Kryptonite	2000-04-29	4	67
2000	3 Doors Down	3:53	Kryptonite	2000-05-06	5	66

id	artist	track	time	id	date	rank
1	2 Pac	Baby Don't Cry	4:22	1	2000-02-26	87
2	2Ge+her	The Hardest Part Of ...	3:15	1	2000-03-04	82
3	3 Doors Down	Kryptonite	3:53	1	2000-03-11	72
4	3 Doors Down	Loser	4:24	1	2000-03-18	77
5	504 Boyz	Wobble Wobble	3:35	1	2000-03-25	87
6	98^0	Give Me Just One Nig...	3:24	1	2000-04-01	94
7	A*Teens	Dancing Queen	3:44	1	2000-04-08	99
8	Aaliyah	I Don't Wanna	4:15	2	2000-09-02	91
9	Aaliyah	Try Again	4:03	2	2000-09-09	87
10	Adams, Yolanda	Open My Heart	5:30	2	2000-09-16	92
11	Adkins, Trace	More	3:05	3	2000-04-08	81
12	Aguilera, Christina	Come On Over Baby	3:38	3	2000-04-15	70
13	Aguilera, Christina	I Turn To You	4:00	3	2000-04-22	68
14	Aguilera, Christina	What A Girl Wants	3:18	3	2000-04-29	67
15	Alice DeeJay	Better Off Alone	6:50	3	2000-05-06	66

Why tidy
data?

Why tidy data?

- Many ways to store the same dataset.
- Having a standard, canonical format that is easy to understand means reusability of tools and ease of cleaning process.
- What would be the best format?

Tidy data and data visualization

Simple & consistent data manipulation and analysis built on tidy data



dplyr

Overview

dplyr is a grammar of data manipulation, providing a consistent set of verbs that help you solve the most common data manipulation challenges:

- `mutate()` adds new variables that are functions of existing variables
- `select()` picks variables based on their names.
- `filter()` picks cases based on their values.
- `summarise()` reduces multiple values down to a single summary.
- `arrange()` changes the ordering of the rows.

These all combine naturally with `group_by()` which allows you to perform any operation “by group”. You can learn more about them in [vignette\("dplyr"\)](#). As well as these single-table verbs, dplyr also provides a variety of two-table verbs, which you can learn about in [vignette\("two-table"\)](#).




ggplot2

Overview

ggplot2 is a system for declaratively creating graphics, based on [The Grammar of Graphics](#). You provide the data, tell ggplot2 how to map variables to aesthetics, what graphical primitives to use, and it takes care of the details.

Tidy data is also often referred as “long-form”

seaborn

Installing Gallery **Tutorial** API Releases Citing FAQ

An introduction to seaborn

Overview of seaborn plotting functions

Data structures accepted by seaborn

The seaborn.objects interface

Properties of Mark objects

Visualizing statistical relationships

Visualizing distributions of data

Visualizing categorical data

Statistical estimation and error bars

Estimating regression fits

Building structured multi-plot grids

Controlling figure aesthetics

Choosing color palettes

Long-form vs. wide-form data

Most plotting functions in seaborn are oriented towards *vectors* of data. When plotting `x` against `y`, each variable should be a vector. Seaborn accepts data *sets* that have more than one vector organized in some tabular fashion. There is a fundamental distinction between “long-form” and “wide-form” data tables, and seaborn will treat each differently.

Long-form data

A long-form data table has the following characteristics:

- Each variable is a column
- Each observation is a row

As a simple example, consider the “flights” dataset, which records the number of airline passengers who flew in each month from 1949 to 1960. This dataset has three variables (*year*, *month*, and number of *passengers*):

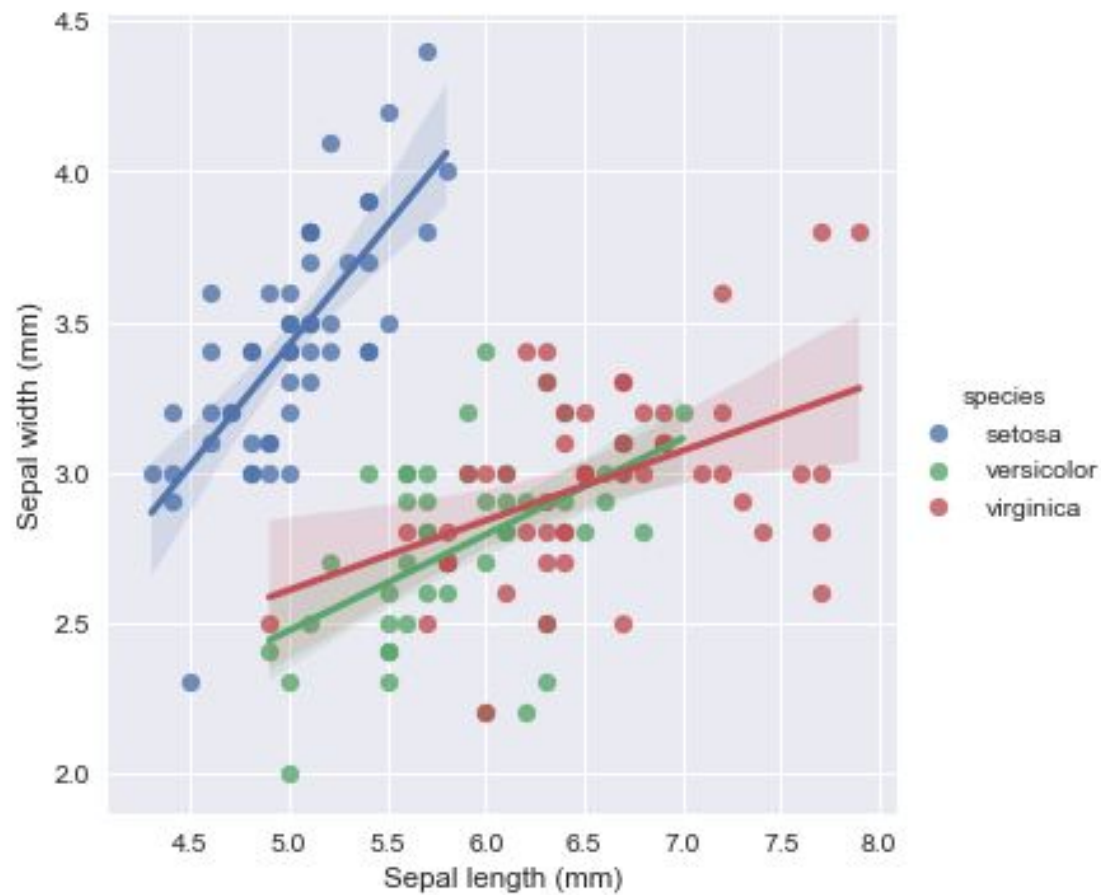
```
flights = sns.load_dataset("flights")
flights.head()
```

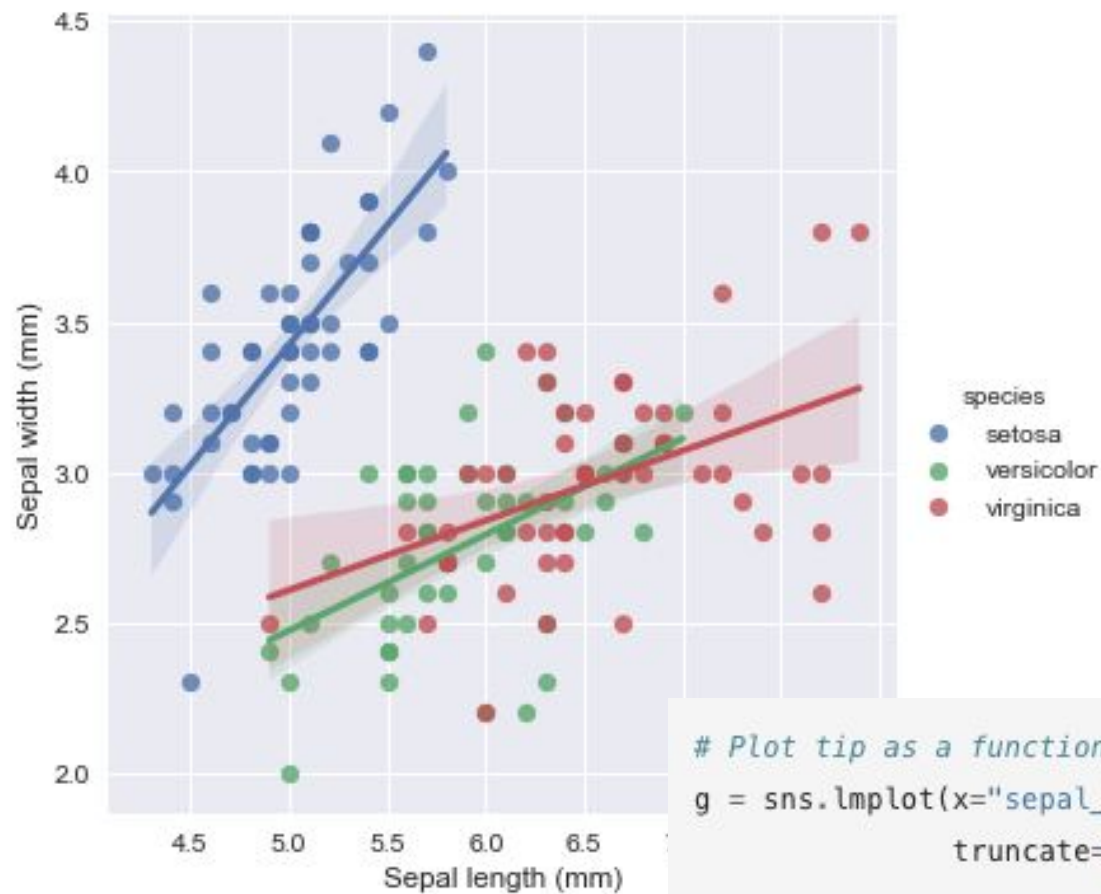
	year	month	passengers
0	1949	Jan	112
1	1949	Feb	118
2	1949	Mar	132
3	1949	Apr	129
4	1949	May	121

With long-form data, columns in the table are given roles in the plot by explicitly assigning them to one of the variables. For example, making a monthly plot of the number of passengers per year looks like this:

```
sns.relplot(data=flights, x="year", y="passengers", hue="month", kind="line")
```

Tidy data and data visualization





```
# Plot tip as a function of total bill across days  
g = sns.lmplot(x="sepal_length", y="sepal_width", hue="species",  
               truncate=True, size=5, data=iris)
```

- Should we ALWAYS keep our dataset TIDY?
- Are tidy datasets most efficient?
- Should we try to use tidy tools and tidy data as much as possible?

Not necessarily

- Should we ALWAYS keep our dataset TIDY?
- Are tidy datasets most efficient? Not necessarily
- Should we try to use tidy tools and tidy data as much as possible?

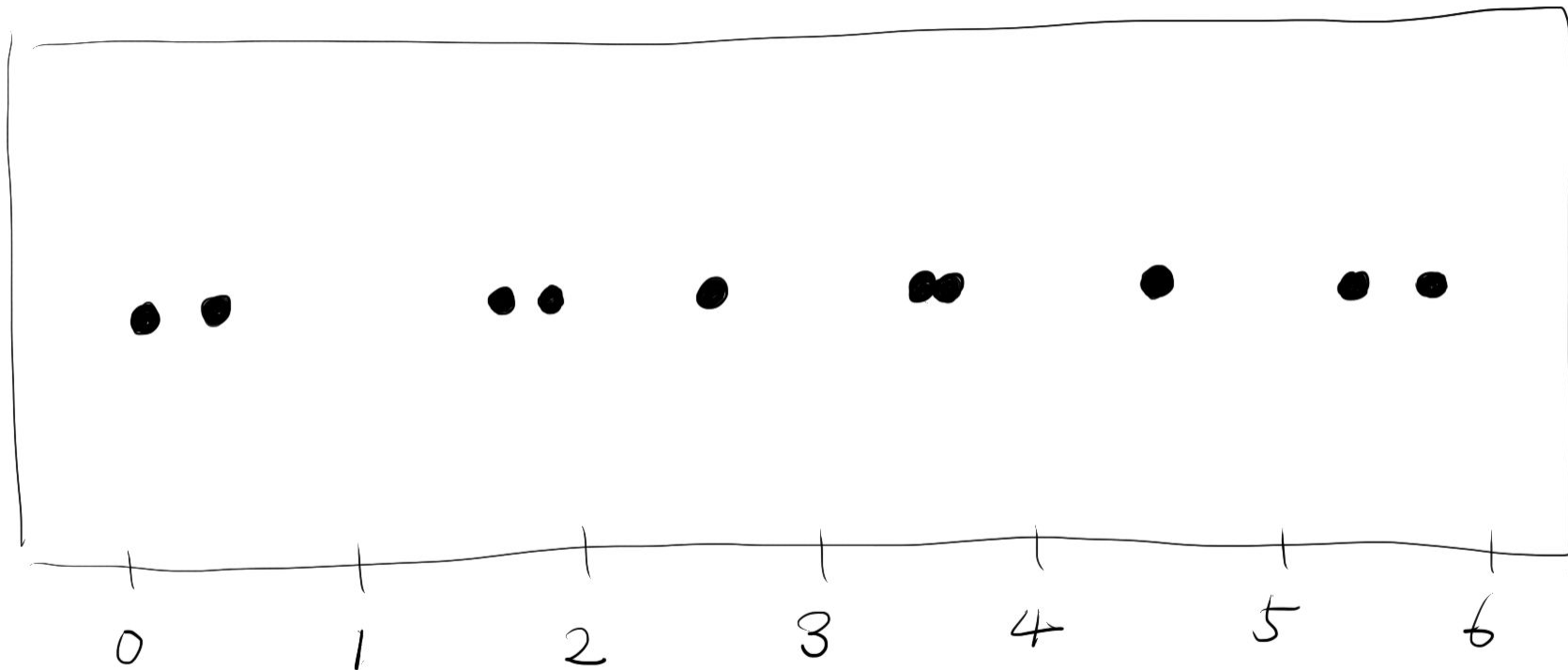
Probably yes

1-D data

Person	Income
1	\$20,000
2	\$150,000
3	\$40,000
4	\$55,000
...	...

If you have like 10 data points,
what would be the most **direct way** to visualize this?

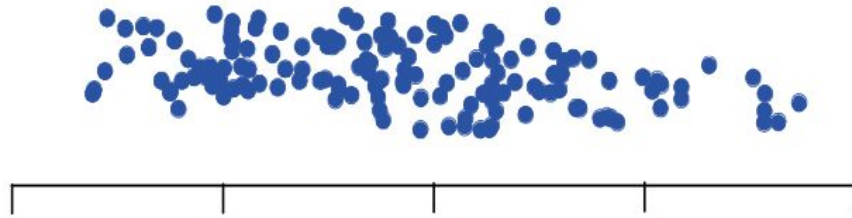
1-d Scatterplot or “strip chart”





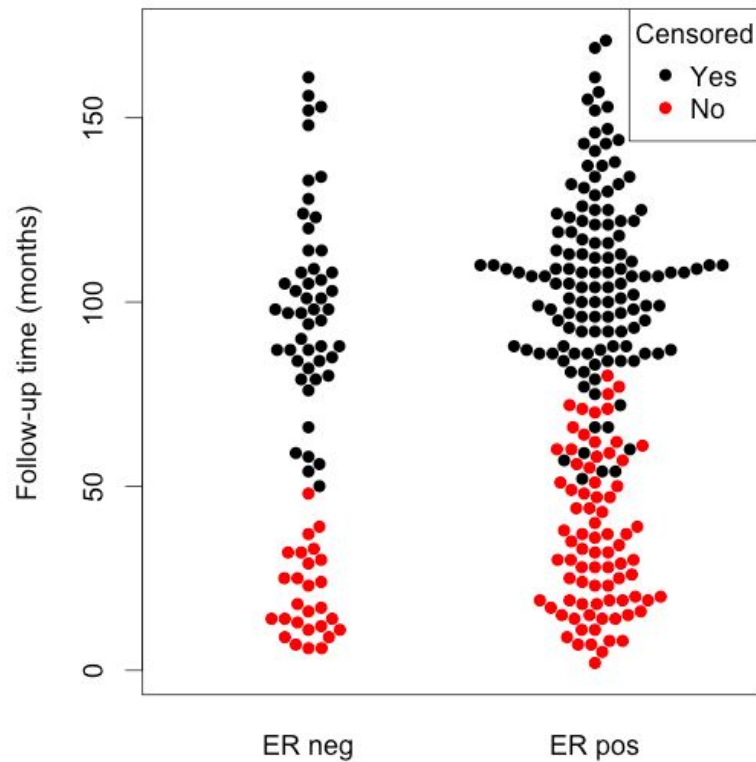
Problems?

1-d “Jittered” Scatterplot

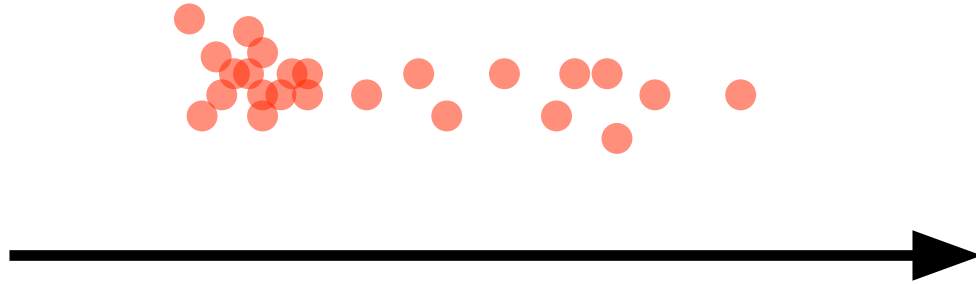


Problem?

"Beeswarm"

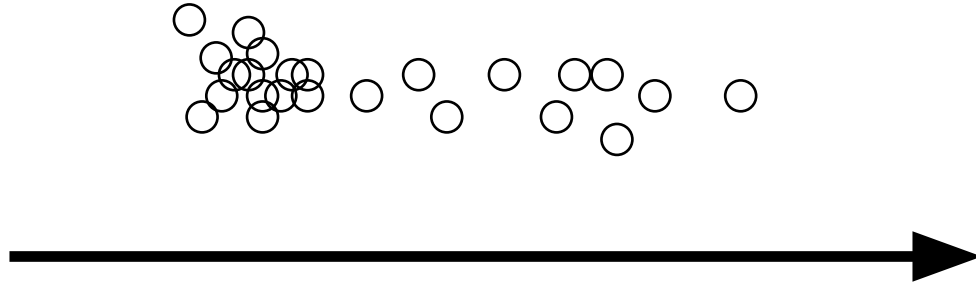


using “alpha”

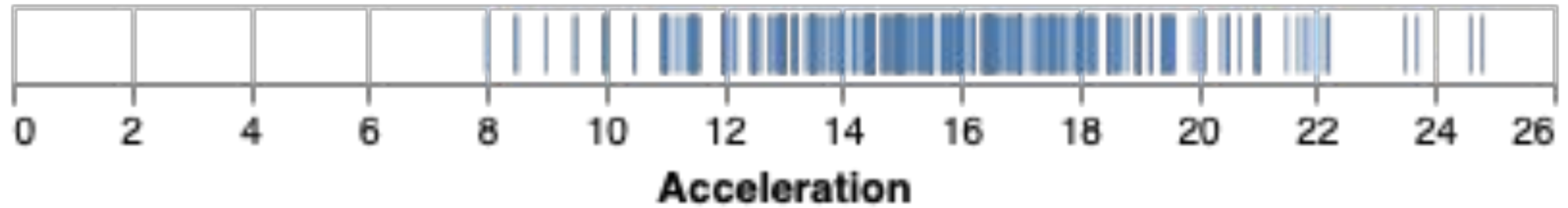


Problem?

Using empty symbols



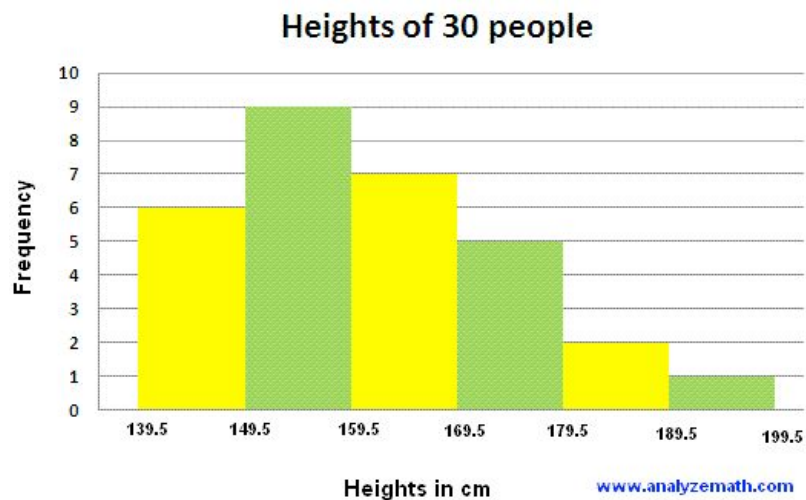
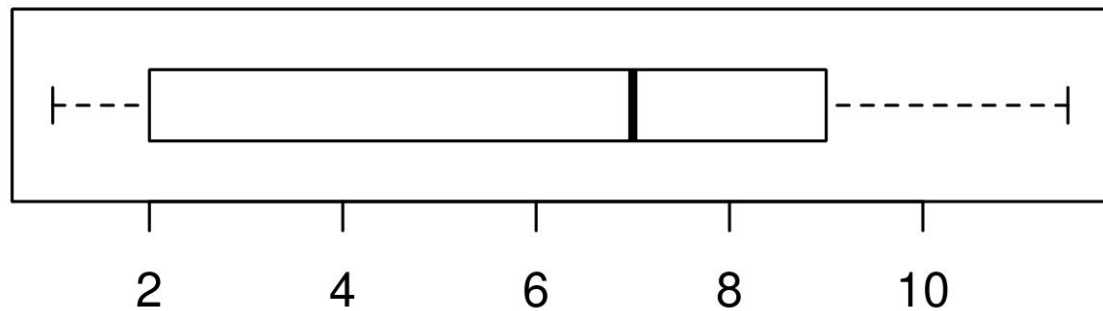
Use lines



What if there are millions
of data points?

What would be good
ways to approach this?

Summarization vs.
aggregation



Box plot

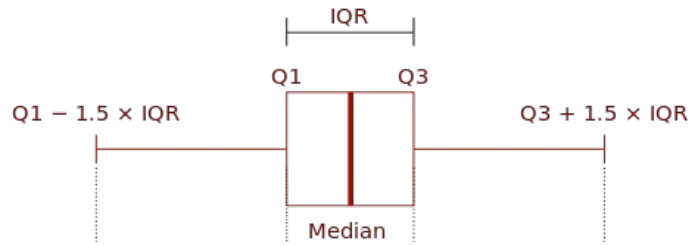
14
14
15
16
16
18
18
19
19
21
22
25
25
29
30
30



Draw two box plots



-1
3
3
4
15
16
16
17
23
24
24
25
35
36
37
46



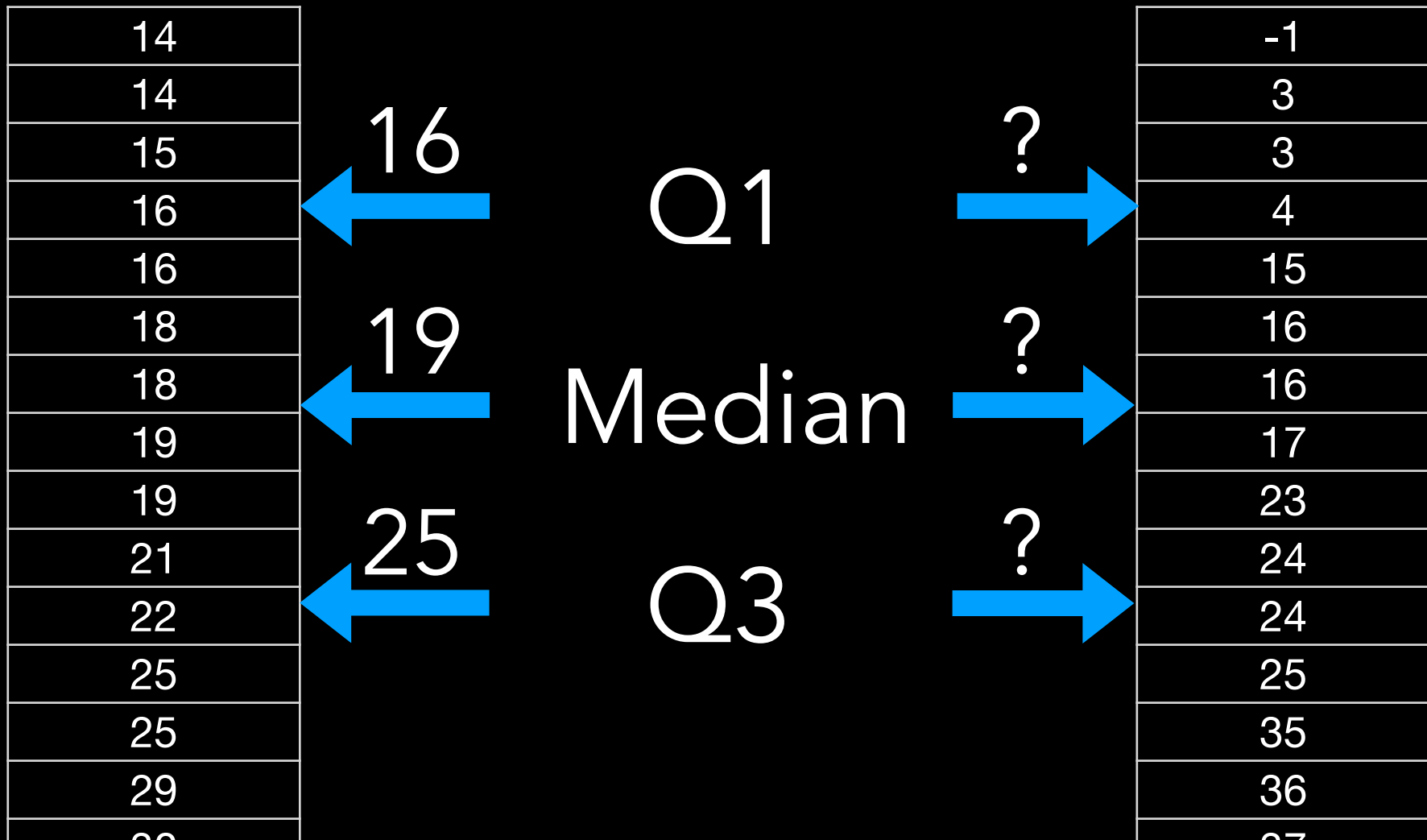
Median

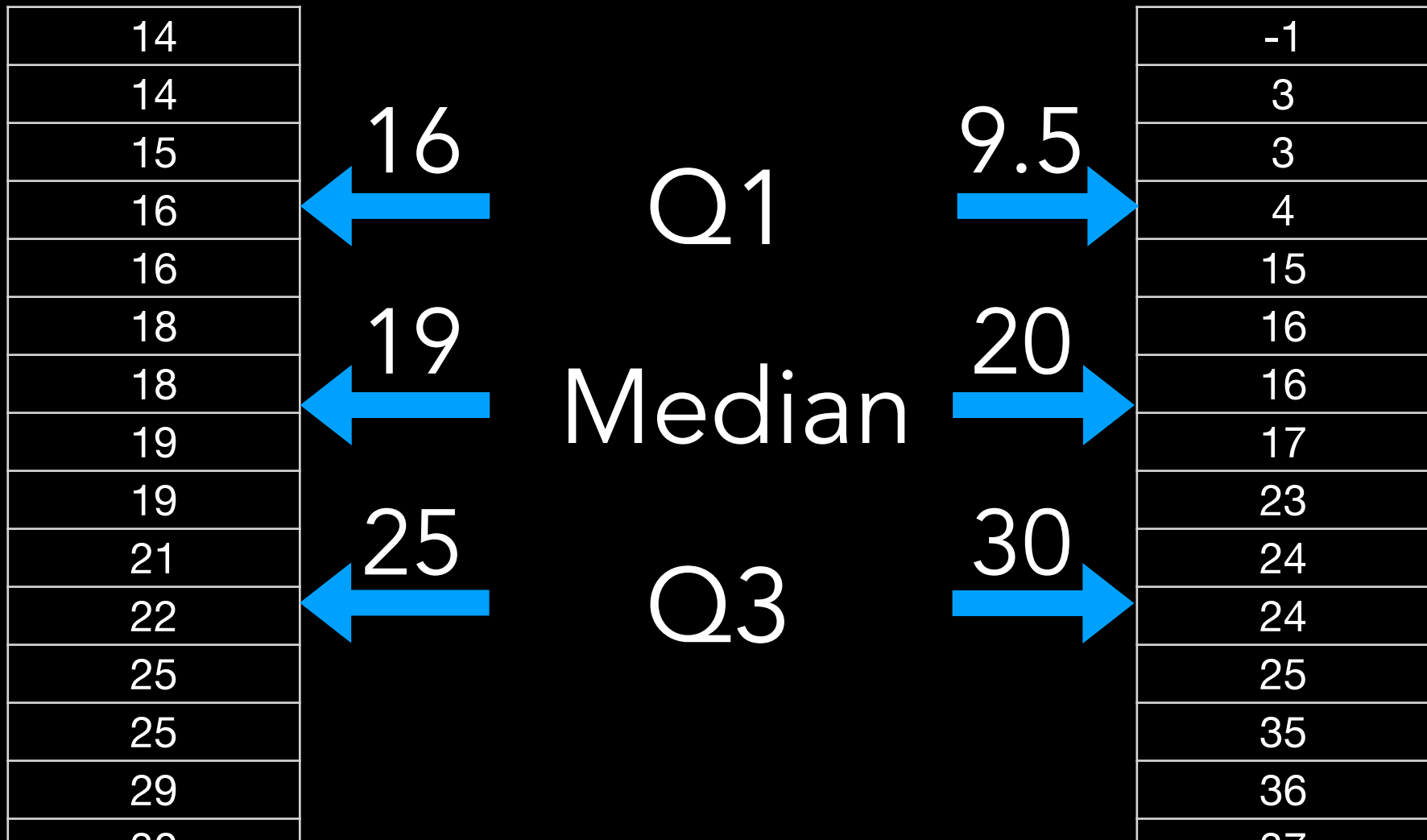
Q1, Q3: the first and the third quartiles

IQR: inter-quartile range
(Q3 - Q1)

14
14
15
16
16
18
18
19
19
21
22
25
25
29
30
30

-1
3
3
4
15
16
16
17
23
24
24
25
35
36
37
46





What are the key ideas in
boxplot?

Let's construct a box plot

