

# Concept of Tidy Data, Vectors & Pivoting

CVEN 5837 - Summer 2023

Lars Schöbitz

<https://cven5873-ss23.github.io/website/>

<https://cven5873-ss23.github.io/website/>

# Learning Objectives (for this week)

1. Learners can apply functions from the dplyr R Package to transform their data from a wide to a long format and vice versa.
2. Learners can list the four main atomic vector types in R.
3. Learners can explain the three characteristics of tidy data.

# Part 1: Data types and vectors

# Why care about data types?

<https://cven5873-ss23.github.io/website/>

<https://cven5873-ss23.github.io/website/>

# Example: survey data

via GIPHY

<b>id</b>	<b>job</b>	<b>price_glass</b>
1	Student	0
2	Retired	0
3	Other	0
4	Employed	10
5	Employed	See comment
6	Student	05-Oct
7	Student	0
8	Retired	0
9	Student	10
10	Employed	0
11	Employed	20 (2chf per person with 10 people in the WG)
12	Student	10
13	Student	10

<b>id</b>	<b>job</b>	<b>price_glass</b>
14	Employed	0
15	Student	10
16	Student	0
17	Employed	5 to 10
18	Other	0
19	Student	0
20	Employed	10
21	Employed	0
22	Employed	5

# Oh why won't you work?!

```
1 survey_data_small |>  
2 summarise(mean_price_glass = mean(price_glass))
```

```
# A tibble: 1 × 1  
  mean_price_glass  
    <dbl>  
1 NA
```

# Oh why won't you still work??!!

```
1 survey_data_small |>  
2 summarise(mean_price_glass = mean(price_glass, na.rm = TRUE))  
  
# A tibble: 1 × 1  
  mean_price_glass  
            <dbl>  
1                NA
```

# Take a breath and look at your data

<b>id</b>	<b>job</b>	<b>price_glass</b>
1	Student	0
2	Retired	0
3	Other	0
4	Employed	10
5	Employed	See comment
6	Student	05-Oct
7	Student	0
8	Retired	0
9	Student	10
10	Employed	0
11	Employed	20 (2chf per person with 10 people in the WG)
12	Student	10
13	Student	10

<b>id</b>	<b>job</b>	<b>price_glass</b>
14	Employed	0
15	Student	10
16	Student	0
17	Employed	5 to 10
18	Other	0
19	Student	0
20	Employed	10
21	Employed	0
22	Employed	5

# Very common data tidying step!

```
1 survey_data_small |>  
2   mutate(price_glass_new = case_when(  
3     price_glass == "5 to 10" ~ "7.5",  
4     price_glass == "05-Oct" ~ "7.5",  
5     str_detect(price_glass, pattern = "2chf") == TRUE ~ "20",  
6     str_detect(price_glass, pattern = "See comment") == TRUE ~ NA_character,  
7     TRUE ~ price_glass  
8   ))
```

# Very common data tidying step!

<b>id</b>	<b>job</b>	<b>price_glass_new</b>	<b>price_glass</b>
1	Student	0	0
2	Retired	0	0
3	Other	0	0
4	Employed	10	10
5	Employed	NA	See comment
6	Student	7.5	05-Oct
7	Student	0	0
8	Retired	0	0
9	Student	10	10
10	Employed	0	0
11	Employed	20	20 (2chf per person with 10 people in the WG)
12	Student	10	10
13	Student	10	10

<b>id</b>	<b>job</b>	<b>price_glass_new</b>	<b>price_glass</b>
14	Employed	0	0
15	Student	10	10
16	Student	0	0
17	Employed	7.5	5 to 10
18	Other	0	0
19	Student	0	0
20	Employed	10	10
21	Employed	0	0
22	Employed	5	5

# Sumamrise? Argh!!!!

```
1 survey_data_small |>
2   mutate(price_glass_new = case_when(
3     price_glass == "5 to 10" ~ "7.5",
4     price_glass == "05-Oct" ~ "7.5",
5     str_detect(price_glass, pattern = "20") == TRUE ~ "20",
6     str_detect(price_glass, pattern = "See comment") == TRUE ~ NA_character_,
7     TRUE ~ price_glass
8   )) |>
9   summarise(mean_price_glass = mean(price_glass_new, na.rm = TRUE))
```

```
# A tibble: 1 × 1
  mean_price_glass
  <dbl>
1       NA
```

# Always respect your data types!

! Taking the mean of vector with type “character” is not possible.

```
# A tibble: 22 × 4
  id   job      price_glass price_glass_new
  <int> <chr>    <chr>        <chr>
1 1   Student  0             0
2 2   Retired   0             0
3 3   Other     0             0
4 4   Employed  10            10
5 5   Employed  See comment <NA>
6 6   Student   05-Oct       7.5
7 7   Student   0             0
8 8   Student   0             0
```

# Always respect your data types!

```
1 survey_data_small |>
2   mutate(price_glass_new = case_when(
3     price_glass == "5 to 10" ~ "7.5",
4     price_glass == "05-Oct" ~ "7.5",
5     str_detect(price_glass, pattern = "20") == TRUE ~ "20",
6     str_detect(price_glass, pattern = "See comment") == TRUE ~ NA_character_,
7     TRUE ~ price_glass
8   )) |>
9   mutate(price_glass_new = as.numeric(price_glass_new)) |>
10  summarise(mean_price_glass = mean(price_glass_new, na.rm = TRUE))
```

```
# A tibble: 1 × 1
  mean_price_glass
                <dbl>
1                 4.76
```

# Live Coding Exercise: Vectors

# live-04a-vectors

1. Head over to rstudio.cloud
2. Open the workspace for the course (cven5837-ss22)
3. Open “Projects”
4. Open the “course-materials” project
5. Follow along with me

# Break One

<https://cven5873-ss23.github.io/website/>



<https://cven5873-ss23.github.io/website/>

# Part 2: `tidyverse` - long and wide formats

# ‘TIDY DATA’

is a standard way of mapping the meaning of a dataset to its structure. ’’

—HADLEY WICKHAM

## In tidy data:

- each variable forms a column
- each observation forms a row
- each cell is a single measurement

each column a variable

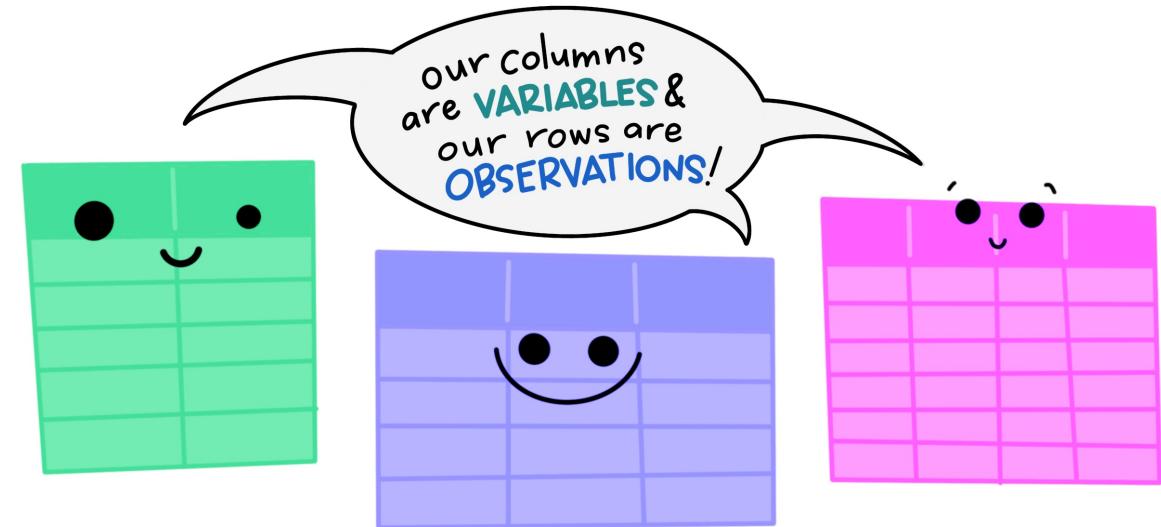
id	name	color
1	floof	gray
2	max	black
3	cat	orange
4	donut	gray
5	merlin	black
6	panda	calico

each row an observation

Wickham, H. (2014). Tidy Data. Journal of Statistical Software 59 (10). DOI: 10.18637/jss.v059.i10

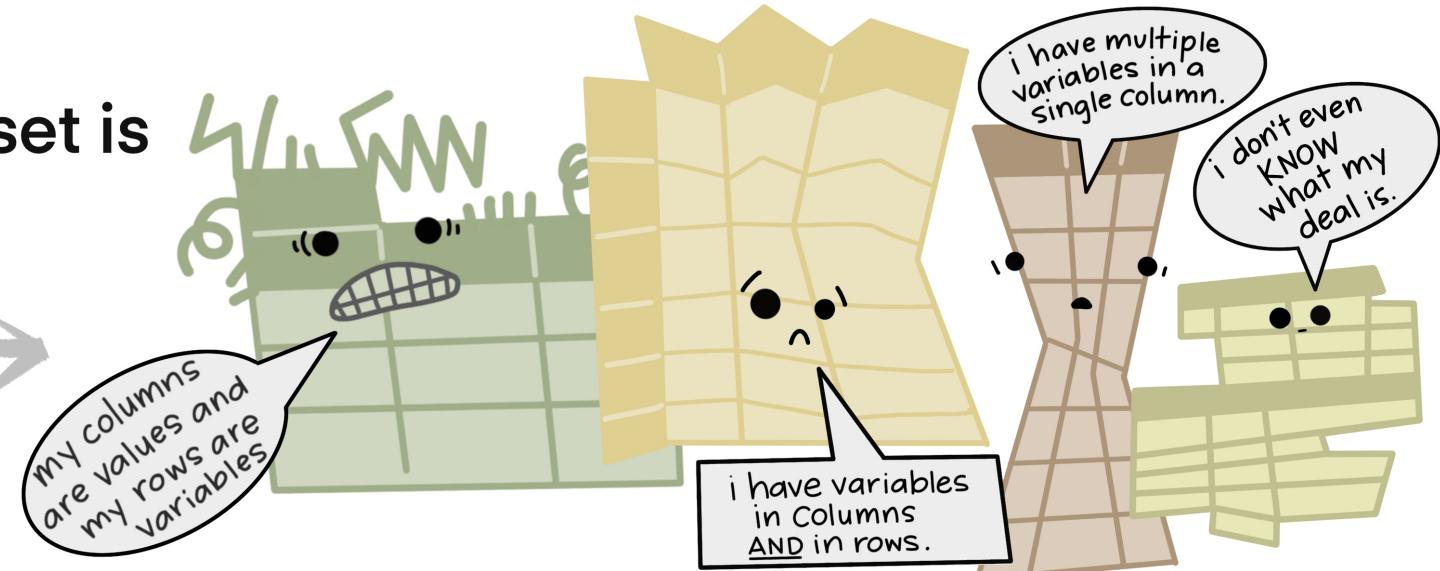
Illustrations from the [Openscapes](#) blog [Tidy Data for reproducibility, efficiency, and collaboration](#) by  
<https://cven5873-ss23.github.io/website/>

The standard structure of  
tidy data means that  
“tidy datasets are all alike...”



“...but every messy dataset is  
messy in its own way.”

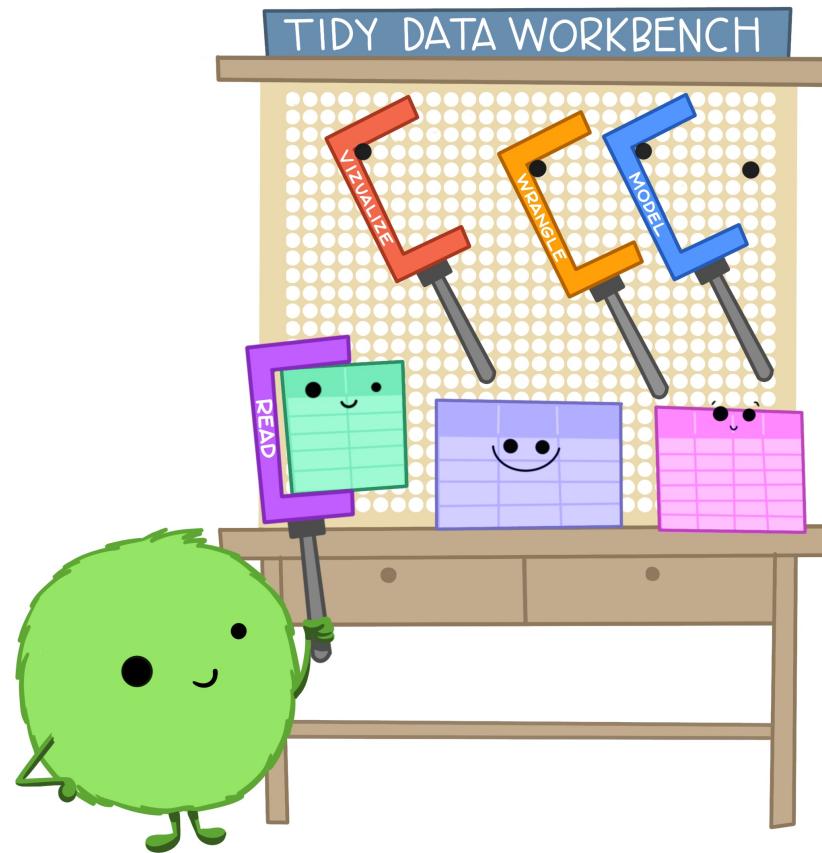
-HADLEY WICKHAM



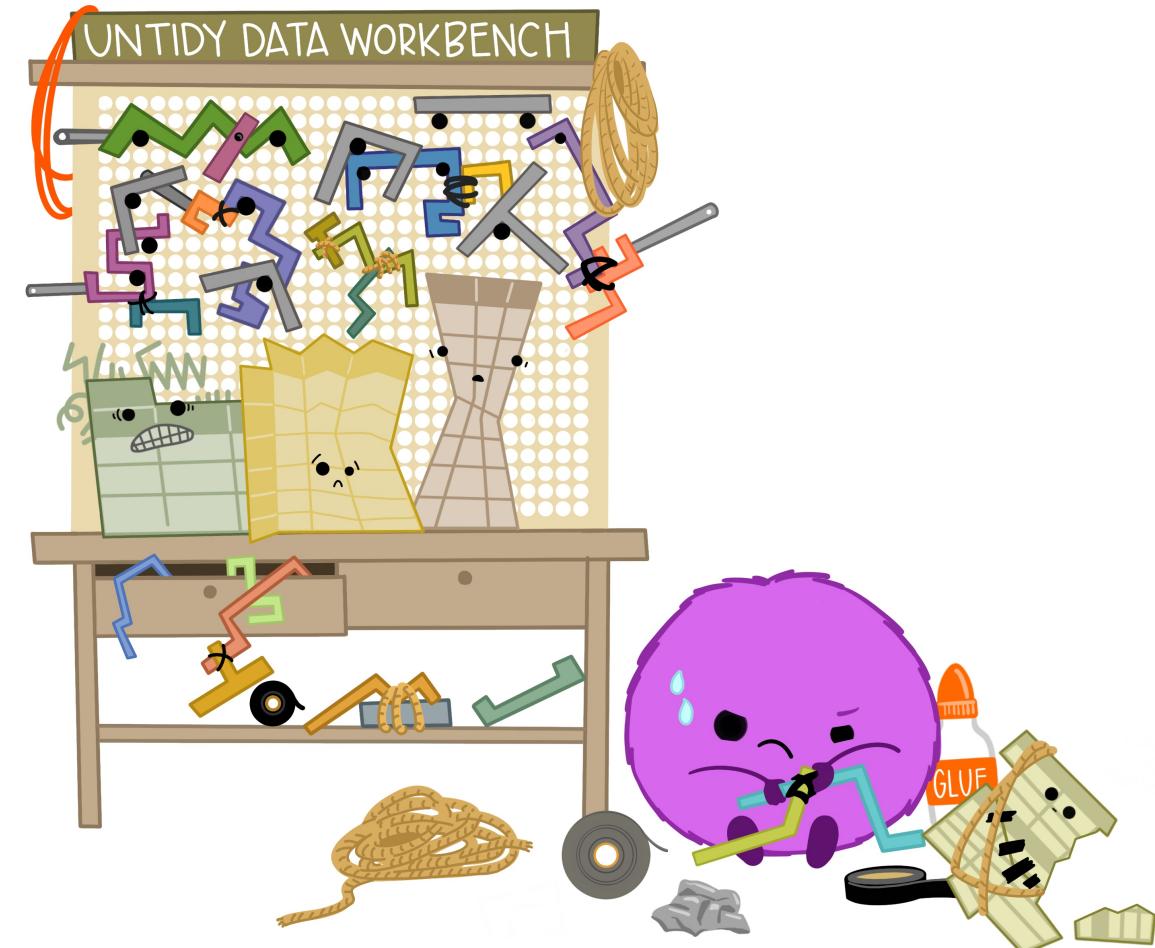
Illustrations from the [Openscapes](#) blog [Tidy Data for reproducibility, efficiency, and collaboration](#) by  
<https://cven5873-ss23.github.io/website/>

- 

When working with tidy data,  
we can use the **same tools** in  
**similar ways** for different datasets...

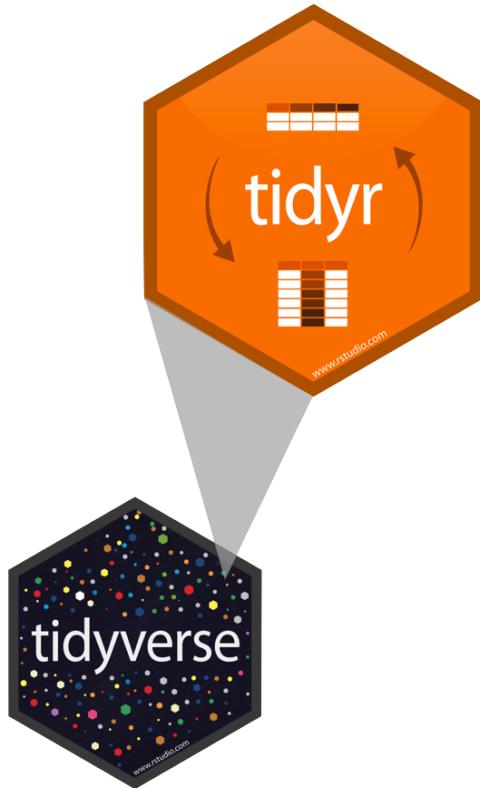


...but working with untidy data often means  
reinventing the wheel with **one-time**  
**approaches** that are **hard to iterate or reuse**.



Illustrations from the [Openscapes](#) blog [Tidy Data for reproducibility, efficiency, and collaboration](#) by  
<https://cven5873-ss23.github.io/website/>

# A grammar of data tidying



The goal of `tidyr` is to help you tidy your data via

- pivoting for going between wide and long data
- splitting and combining character columns
- nesting and unnesting columns
- clarifying how `NA`s should be treated

# Pivoting data

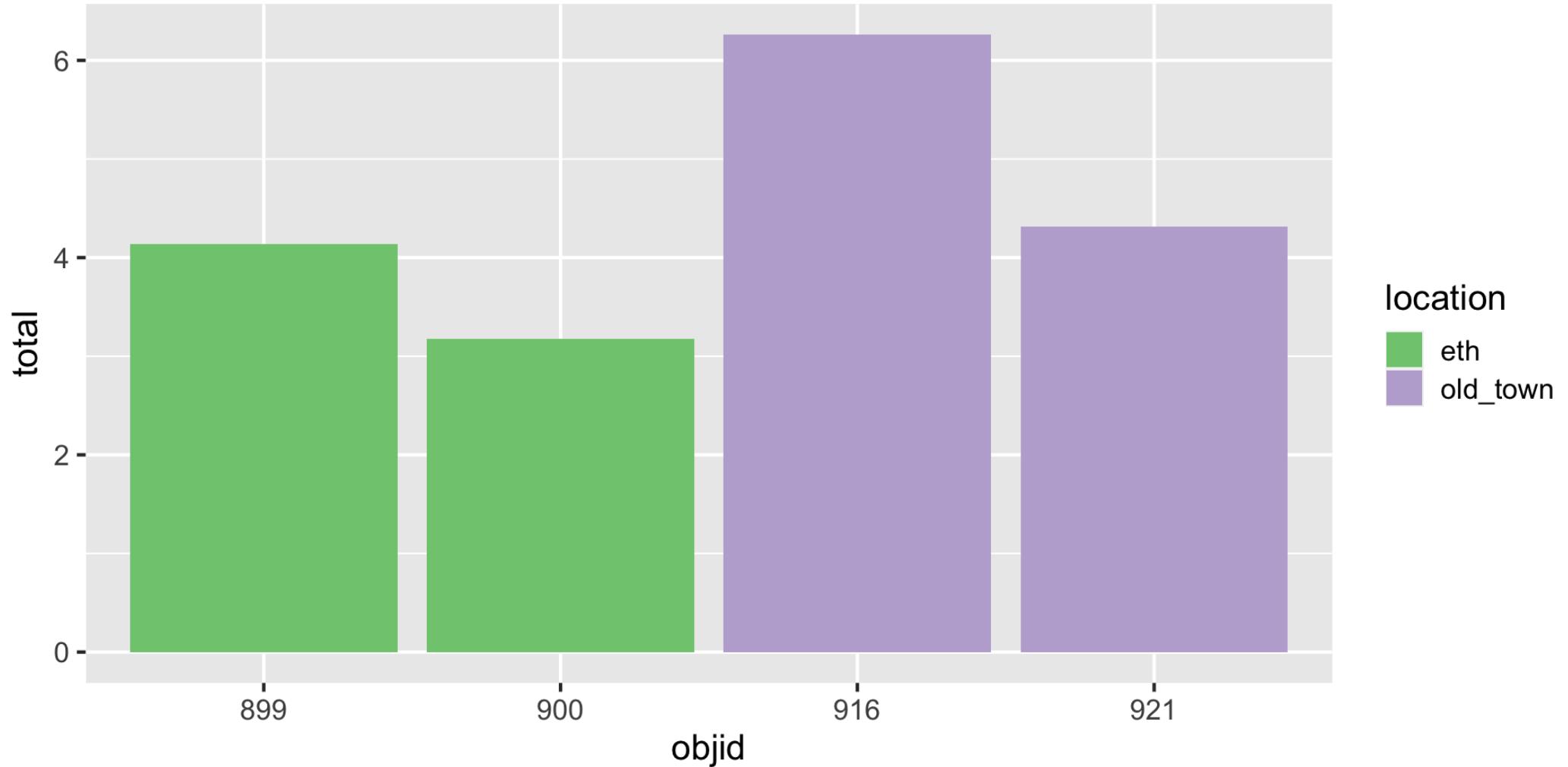
wide

id	x	y	z
1	a	c	e
2	b	d	f

# Waste characterisation data

objid	location	pet	metal_alu	glass	paper	other	total
900	eth	0.06	0.06	0.58	0.21	1.14	2.05
899	eth	0.14	0.01	0.18	0.28	3.04	3.64
921	old_town	0.00	0.00	0.00	0.41	1.57	1.99
916	old_town	0.17	0.04	0.80	0.55	0.62	2.19
900	eth	0.10	0.04	0.00	0.40	0.58	1.12
899	eth	0.08	0.03	0.00	0.05	0.34	0.50
921	old_town	0.08	0.03	0.30	0.40	1.52	2.33
916	old_town	0.11	0.04	0.92	1.01	1.99	4.07

# How would you plot this?

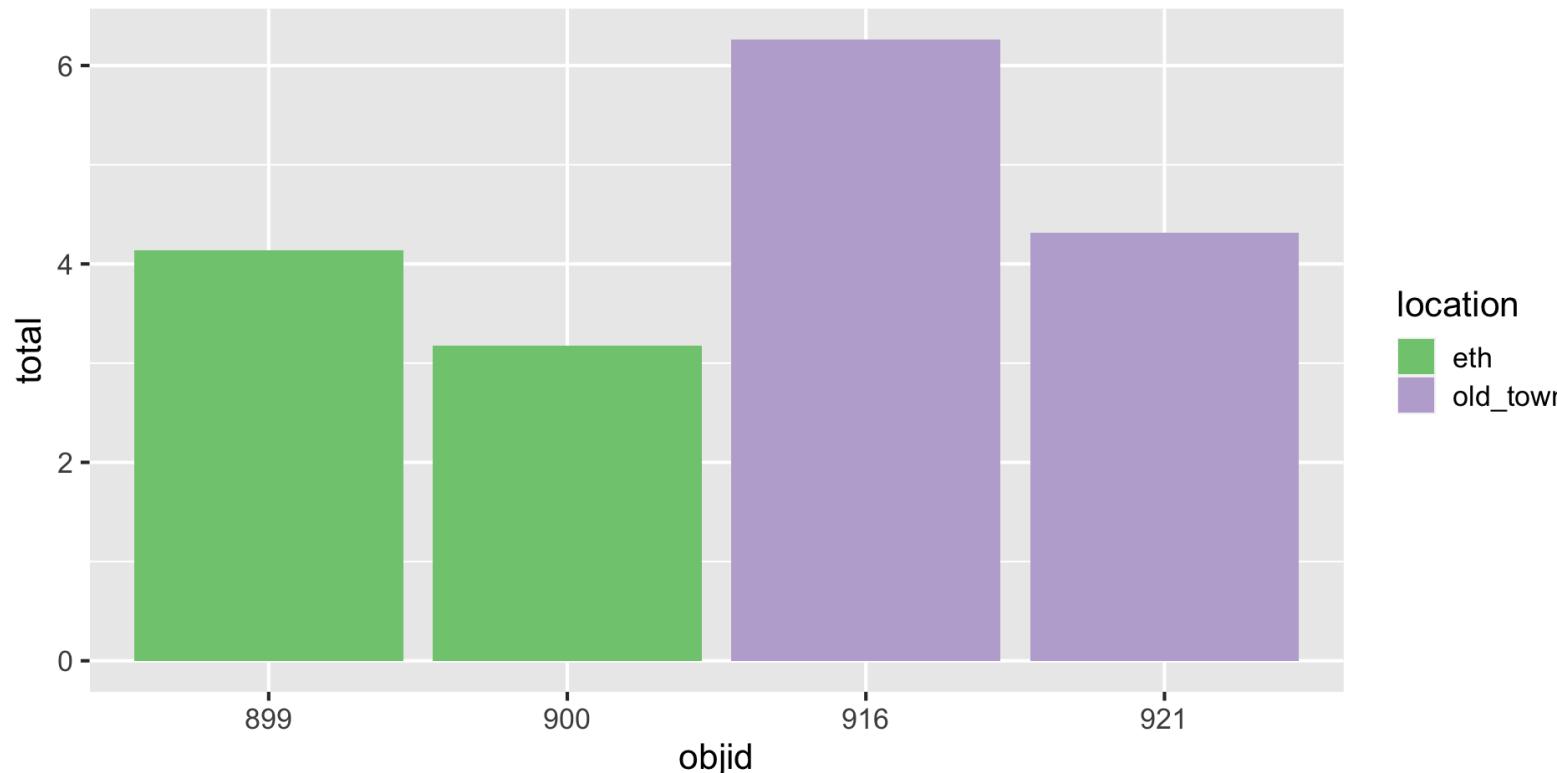


# Three variables

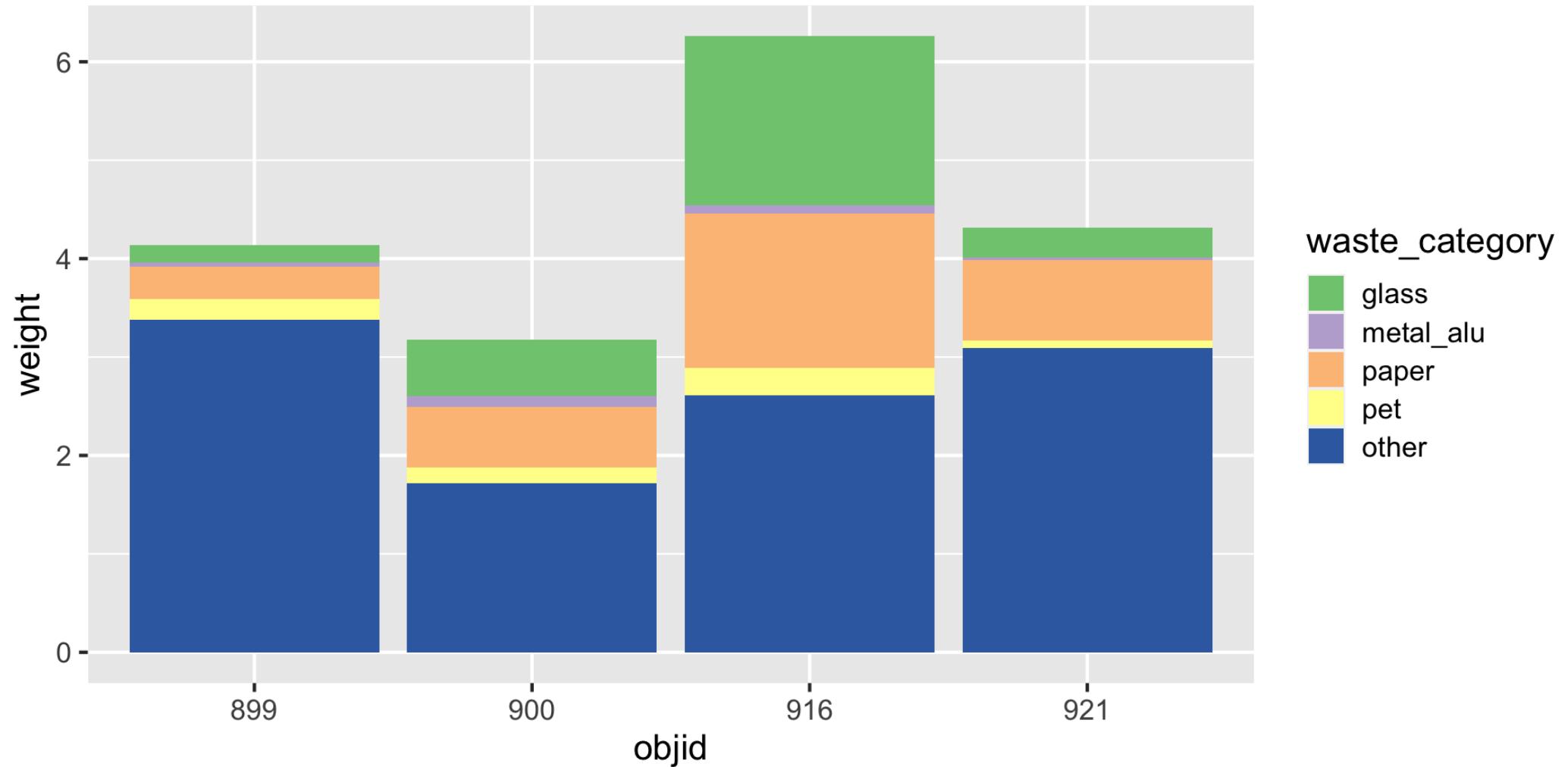
objid	location	total
900	eth	2.05
899	eth	3.64
921	old_town	1.99
916	old_town	2.19
900	eth	1.12
899	eth	0.50
921	old_town	2.33
916	old_town	4.07

# Three variables -> three aesthetics

```
1 ggplot(data = waste_data_untidy,  
2         mapping = aes(x = objid,  
3                           y = total,  
4                           fill = location)) +  
5   geom_col() +  
6   scale_fill_brewer(type = "qual")
```



# And how to plot this?



# Reminder: Data (in wide format)

objid	location	pet	metal_alu	glass	paper	other
900	eth	0.06	0.06	0.58	0.21	1.14
899	eth	0.14	0.01	0.18	0.28	3.04
921	old_town	0.00	0.00	0.00	0.41	1.57
916	old_town	0.17	0.04	0.80	0.55	0.62
900	eth	0.10	0.04	0.00	0.40	0.58
899	eth	0.08	0.03	0.00	0.05	0.34
921	old_town	0.08	0.03	0.30	0.40	1.52
916	old_town	0.11	0.04	0.92	1.01	1.99

# You need: A long format

objid	location	waste_category	weight
900	eth	pet	0.06
900	eth	metal_alu	0.06
900	eth	glass	0.58
900	eth	paper	0.21
900	eth	other	1.14
899	eth	pet	0.14
899	eth	metal_alu	0.01
899	eth	glass	0.18
899	eth	paper	0.28
899	eth	other	3.04
921	old_town	pet	0.00
921	old_town	metal_alu	0.00
921	old_town	glass	0.00

<https://cven5873-ss23.github.io/website/>

objid	location	waste_category	weight
921	old_town	paper	0.41
921	old_town	other	1.57
916	old_town	pet	0.17
916	old_town	metal_alu	0.04
916	old_town	glass	0.80
916	old_town	paper	0.55
916	old_town	other	0.62
900	eth	pet	0.10
900	eth	metal_alu	0.04
900	eth	glass	0.00
900	eth	paper	0.40
900	eth	other	0.58
899	eth	pet	0.08
899	eth	metal_alu	0.03
899	eth	glass	0.00
899	eth	paper	0.05

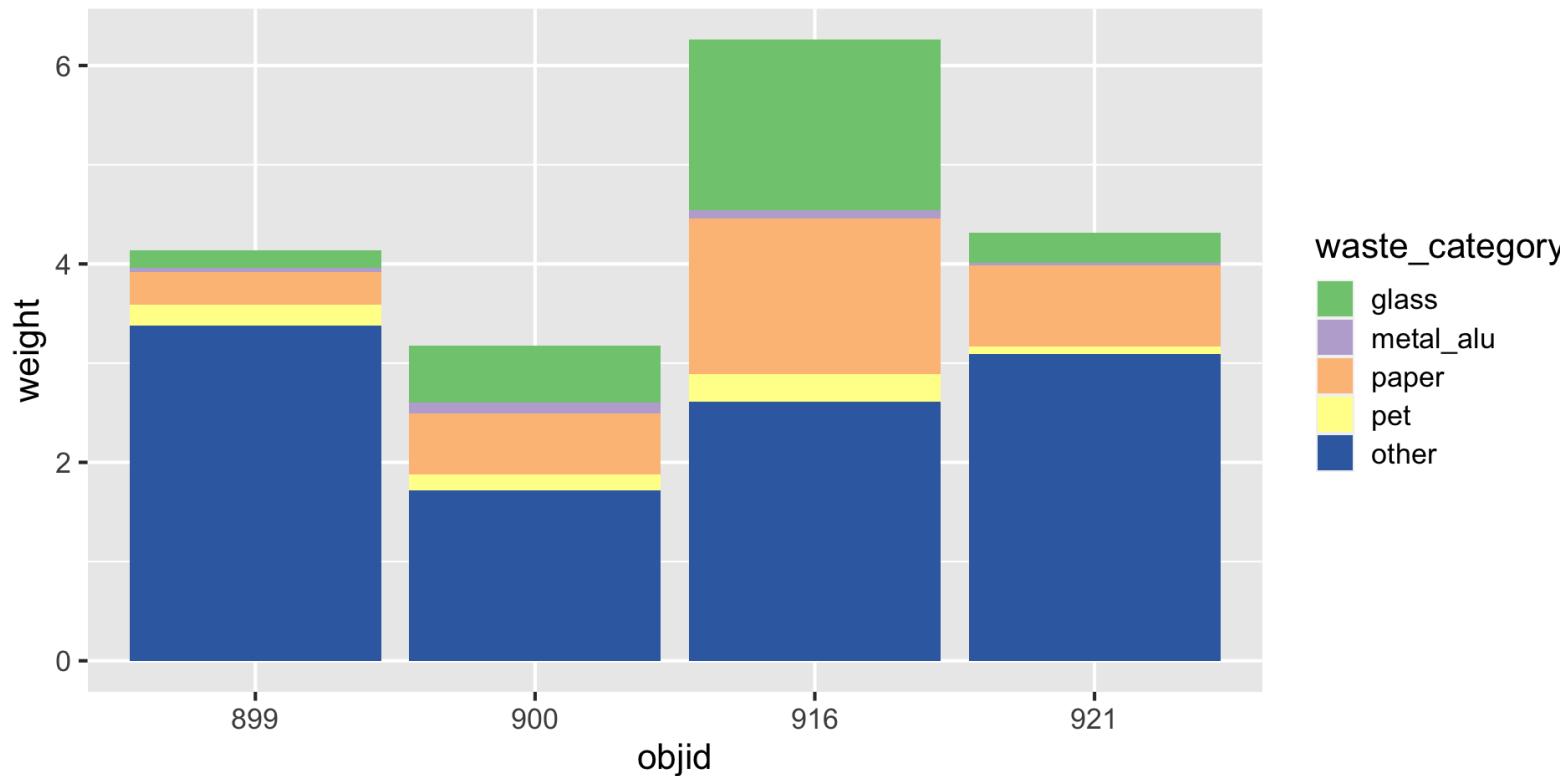
objid	location	waste_category	weight
899	eth	other	0.34
921	old_town	pet	0.08
921	old_town	metal_alu	0.03
921	old_town	glass	0.30
921	old_town	paper	0.40
921	old_town	other	1.52
916	old_town	pet	0.11
916	old_town	metal_alu	0.04
916	old_town	glass	0.92
916	old_town	paper	1.01
916	old_town	other	1.99

# Three variables -> three aesthetics

```

1 ggplot(data = waste_data_tidy,
2         mapping = aes(x = objid,
3                         y = weight,
4                         fill = waste_category)) +
5   geom_col() +
6   scale_fill_brewer(type = "qual")

```



# How to

## 1 waste\_data\_untidy

objid	location	pet	metal_alu	glass	paper	other
900	eth	0.06	0.06	0.58	0.21	1.14
899	eth	0.14	0.01	0.18	0.28	3.04
921	old_town	0.00	0.00	0.00	0.41	1.57
916	old_town	0.17	0.04	0.80	0.55	0.62
900	eth	0.10	0.04	0.00	0.40	0.58
899	eth	0.08	0.03	0.00	0.05	0.34
921	old_town	0.08	0.03	0.30	0.40	1.52
916	old_town	0.11	0.04	0.92	1.01	1.99

# How to

```

1 waste_data_untidy |>
2   pivot_longer(cols = pet:other,
3                 names_to = "waste_category",
4                 values_to = "weight")

```

objid	location	waste_category	weight
900	eth	pet	0.06
900	eth	metal_alu	0.06
900	eth	glass	0.58
900	eth	paper	0.21
900	eth	other	1.14
899	eth	pet	0.14
899	eth	metal_alu	0.01
899	eth	glass	0.18
899	eth	paper	0.28
899	eth	other	3.04
921	old_town	pet	0.00

objid	location	waste_category	weight
921	old_town	metal_alu	0.00
921	old_town	glass	0.00
921	old_town	paper	0.41
921	old_town	other	1.57
916	old_town	pet	0.17
916	old_town	metal_alu	0.04
916	old_town	glass	0.80
916	old_town	paper	0.55
916	old_town	other	0.62
900	eth	pet	0.10
900	eth	metal_alu	0.04
900	eth	glass	0.00
900	eth	paper	0.40
900	eth	other	0.58
899	eth	pet	0.08
899	eth	metal_alu	0.03

<https://cven5873-ss23.github.io/website/>

objid	location	waste_category	weight
899	eth	glass	0.00
899	eth	paper	0.05
899	eth	other	0.34
921	old_town	pet	0.08
921	old_town	metal_alu	0.03
921	old_town	glass	0.30
921	old_town	paper	0.40
921	old_town	other	1.52
916	old_town	pet	0.11
916	old_town	metal_alu	0.04
916	old_town	glass	0.92
916	old_town	paper	1.01
916	old_town	other	1.99

# How to

```

1 waste_category_levels <- c("glass", "metal_alu", "paper", "pet", "other")
2
3 waste_data_untidy |>
4   pivot_longer(cols = pet:other,
5                 names_to = "waste_category",
6                 values_to = "weight") |>
7   mutate(waste_category = factor(waste_category, levels = waste_category_

```

objid	location	waste_category	weight
900	eth	pet	0.06
900	eth	metal_alu	0.06
900	eth	glass	0.58
900	eth	paper	0.21
900	eth	other	1.14
899	eth	pet	0.14
899	eth	metal_alu	0.01
899	eth	glass	0.18
899	eth	paper	0.28

objid	location	waste_category	weight
899	eth	other	3.04
921	old_town	pet	0.00
921	old_town	metal_alu	0.00
921	old_town	glass	0.00
921	old_town	paper	0.41
921	old_town	other	1.57
916	old_town	pet	0.17
916	old_town	metal_alu	0.04
916	old_town	glass	0.80
916	old_town	paper	0.55
916	old_town	other	0.62
900	eth	pet	0.10
900	eth	metal_alu	0.04
900	eth	glass	0.00
900	eth	paper	0.40
900	eth	other	0.58

<https://cven5873-ss23.github.io/website/>

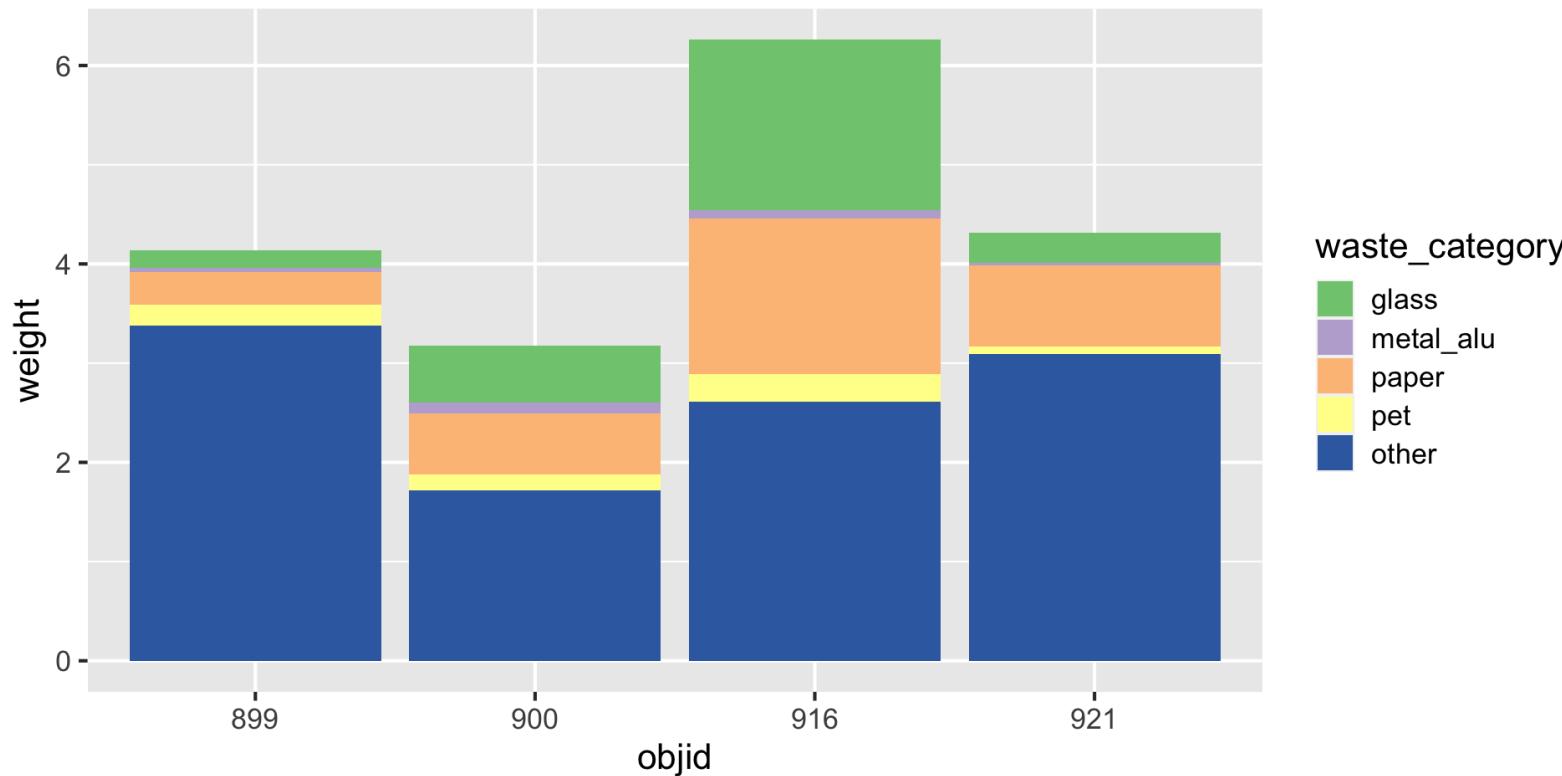
objid	location	waste_category	weight
899	eth	pet	0.08
899	eth	metal_alu	0.03
899	eth	glass	0.00
899	eth	paper	0.05
899	eth	other	0.34
921	old_town	pet	0.08
921	old_town	metal_alu	0.03
921	old_town	glass	0.30
921	old_town	paper	0.40
921	old_town	other	1.52
916	old_town	pet	0.11
916	old_town	metal_alu	0.04
916	old_town	glass	0.92
916	old_town	paper	1.01
916	old_town	other	1.99

# Three variables -> three aesthetics

```

1 ggplot(data = waste_data_tidy,
2         mapping = aes(x = objid,
3                         y = weight,
4                         fill = waste_category)) +
5   geom_col() +
6   scale_fill_brewer(type = "qual")

```



# Live Coding Exercise: Pivoting

<https://cven5873-ss23.github.io/website/>

# live-05a-tidyr-pivoting

1. Head over to posit.cloud
2. Open the workspace for the course (cven5837-ss23)
3. Open “Projects”
4. Open the “course-materials” project
5. Follow along with me

# Homework week 5

<https://cven5873-ss23.github.io/website/>

# Homework due dates

- All material on [course website](#)
- Homework assignment & learning reflection due: **Friday, 7th July**

Thanks! 🌻

Slides created via revealjs and Quarto:

<https://quarto.org/docs/presentations/revealjs/> Access slides as PDF on GitHub

All material is licensed under [Creative Commons Attribution Share Alike 4.0 International.](#)

# References

Ben Aleya, Ali, Daniel Biek, Lin Boynton, Julia Jaeggi, Sebastian Camilo Loos, Chiara Meyer-Piening, Jonathan Olal Ogwang, et al. 2022. "Research Beyond the Lab, Spring Term 2022, Global Health Engineering, ETH Zurich. Raw Data and Analysis-Ready Derived Data on Waste Management in Public Spaces in Zurich, Switzerland." Zenodo. <https://doi.org/10.5281/ZENODO.7331119>.

<https://cven5873-ss23.github.io/website/>