DONALD JOHN MACALLISTER 08/10/2020

The tidyverse: Manipulating and visualising large datasets

British Geological Survey

Overview

- What is the tidyverse?

- What is tidy data?

- Using tidyverse to understand performance of rural water supplies in Ethiopia during drought:
  - Data manipulation with dplyr
  - Plotting with ggplot2

WHAT IS THE TIDYVERSE?

“The tidyverse is an opinionated collection of R packages designed for data science. All packages share an underlying design philosophy, grammar, and data structures”

What is the tidyverse?

reading data
ggplot2
dplyr
readr
TIBBLE
data visualisation
modern data frames
data wrangling
tidyverse
data tidying
string manipulation
stringr
forcats
tidyr
purrr
functional programming
dealing with factors

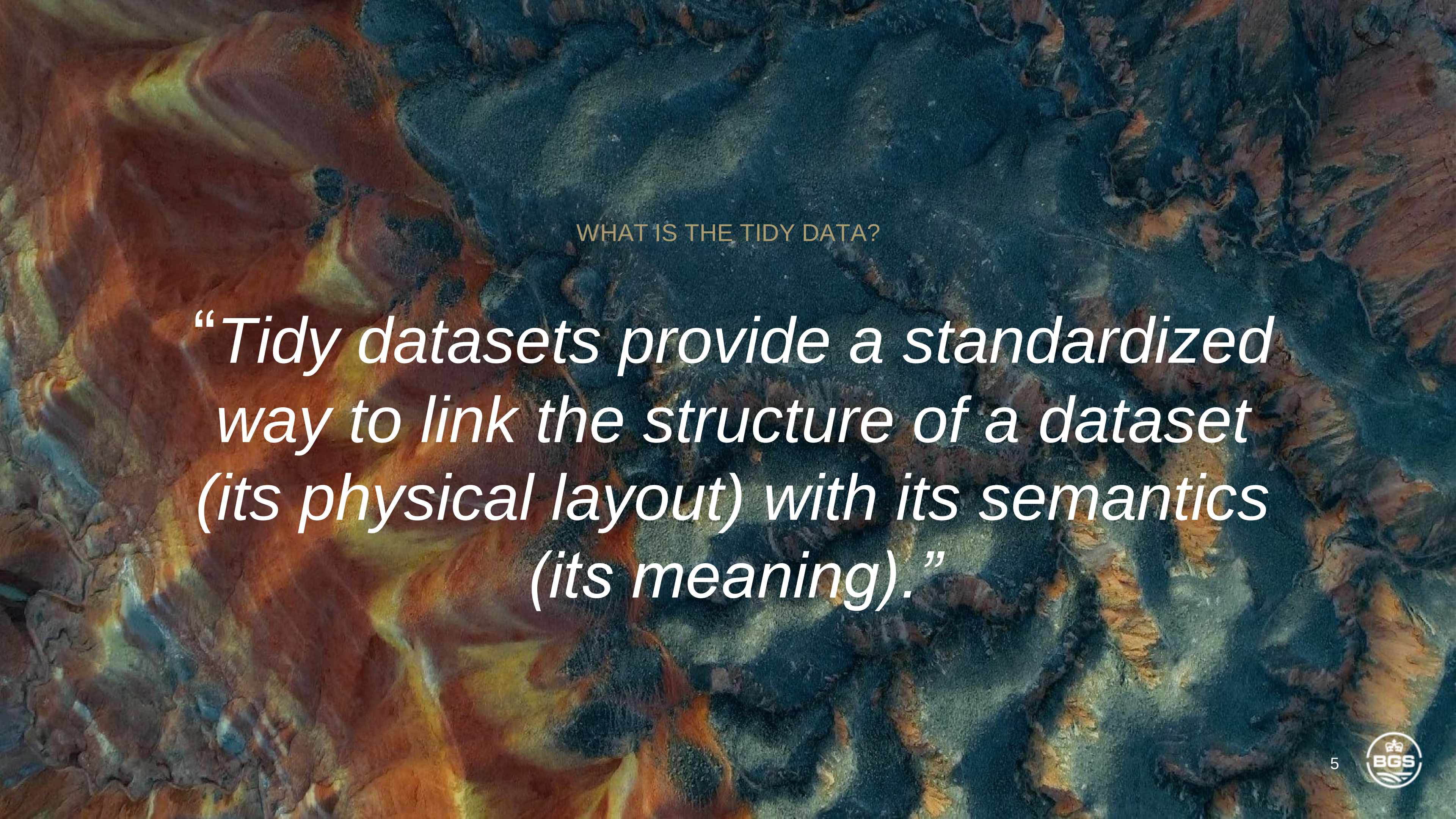
- **ggplot2: data visualisation**
- **dplyr: data wrangling**

- readr: reading data
- stringr: string manipulation
- tidyr: data tidying

Wickham, Hadley, et al. "Welcome to the Tidyverse." Journal of Open Source Software 4.43 (2019): 1686.

Wickham, Hadley. "Tidy data." Journal of Statistical Software 59.10 (2014): 1-23.

https://www.tidyverse.org

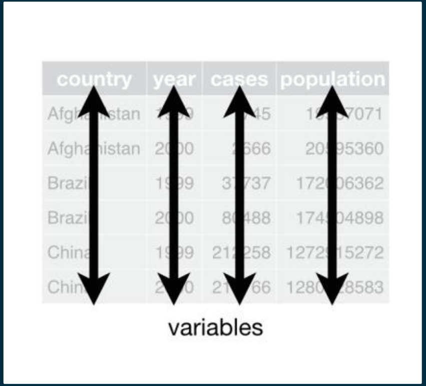


WHAT IS THE TIDY DATA?

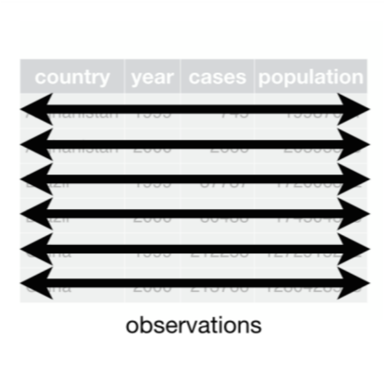
“Tidy datasets provide a standardized way to link the structure of a dataset (its physical layout) with its semantics (its meaning).”

What is tidy data?

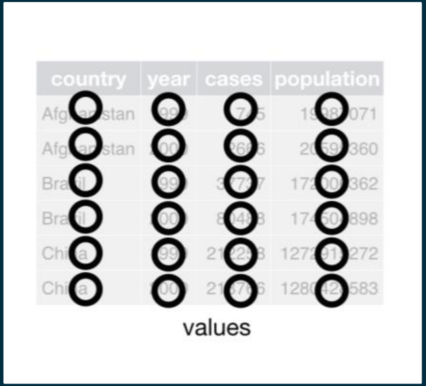
- A dataset is:
  - A collection of values, either numbers (quantitative) or strings (qualitative).
  - Every value belongs to a variable and an observation.
  - A variable contains all values that measure the same underlying attribute.
  - An observation contains all values measured on the same unit.



variables



observations

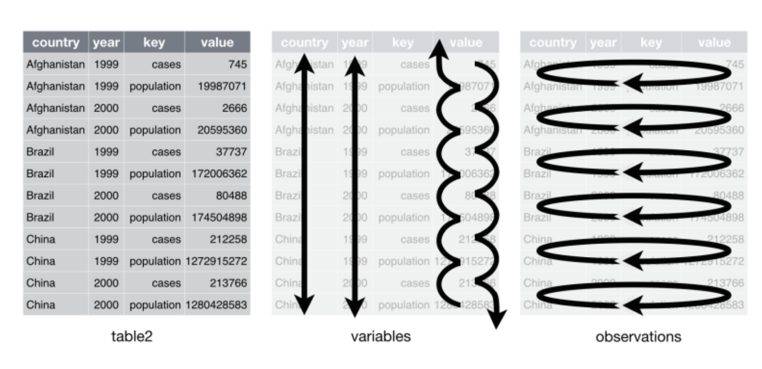


values

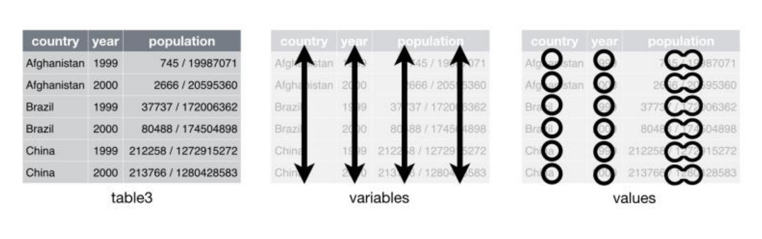
https://garrettgman.github.io/tidying/

What is not tidy data?

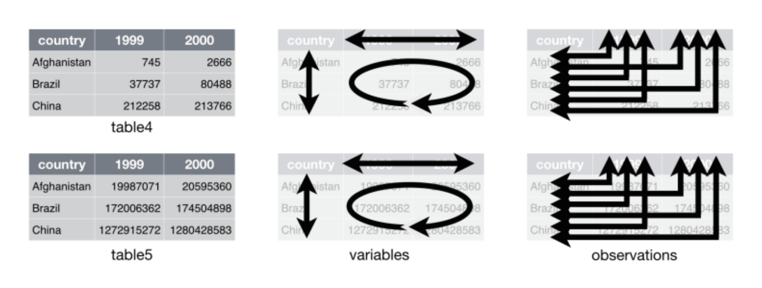
- Messy datasets, might include (but are not limited to) the following problems :
  - Column headers are values, not variable names.
  - Multiple variables are stored in one column.
  - Variables are stored in both rows and columns.
  - Mixing values of different types (i.e. text and numbers)



country	year	key	value
Afghanistan	1999	cases	745
Afghanistan	1999	population	19987071
Afghanistan	2000	cases	2666
Afghanistan	2000	population	20595360
Brazil	1999	cases	37737
Brazil	1999	population	172006362
Brazil	2000	cases	80488
Brazil	2000	population	174504898
China	1999	cases	212258
China	1999	population	1272915272
China	2000	cases	213766
China	2000	population	1280428583



country	year	population
Afghanistan	1999	745 / 19987071
Afghanistan	2000	2666 / 20595360
Brazil	1999	37737 / 172006362
Brazil	2000	80488 / 174504898
China	1999	212258 / 1272915272
China	2000	213766 / 1280428583



country	1999	2000
Afghanistan	745	2666
Brazil	37737	80488
China	212258	213766

country	1999	2000
Afghanistan	19987071	20595360
Brazil	172006362	174504898
China	1272915272	1280428583

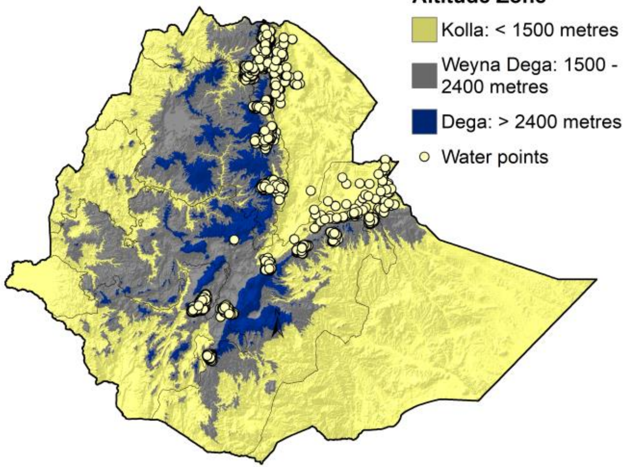
https://garrettgman.github.io/tidying/

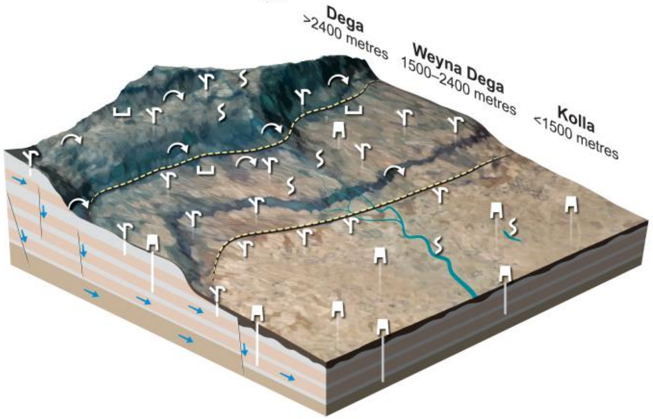
USING THE TIDYVERSE IN A HYDROGEOLOGY:
AN EXAMPLE FROM ETHIOPIA

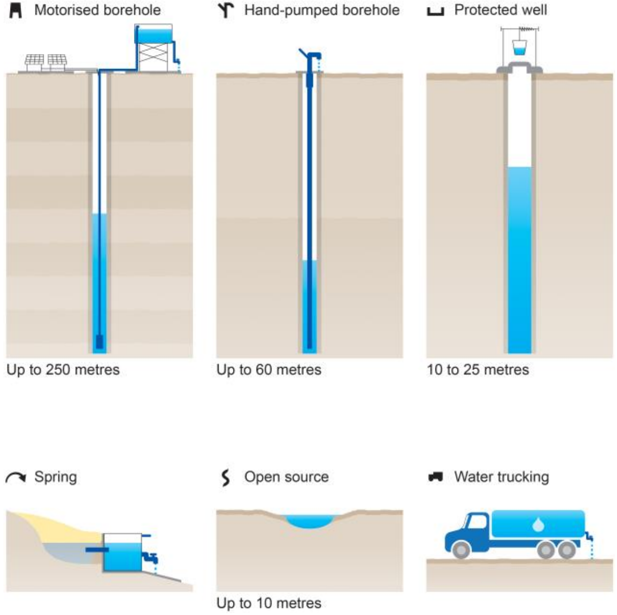
“Comparative performance of rural water supplies during drought”

Using the tidyverse: An example from Ethiopia

- El Nino water supply drought monitoring
- 5196 individual water points
- 12 weeks monitoring
- > 28,000 site visits (observations)
- Data on functionality, user numbers, water quantity, etc. (variables)







MacAllister, D. J., et al. "Comparative performance of rural water supplies during drought." Nature communications 11.1 (2020): 1-13.

Using the tidyverse: An example from Ethiopia

This is our dataset, lets call it: ETH

WP_ID	DATE	WEEK	AREA	TYPE	FCAT	USERS	TRAVEL_TIME	WATER_QUANT
105s-nq41-sbqa	15/02/2016	3	Weyna Dega	Hand-pumped borehole	Functional	600	Between 30 minutes & 1 hour	Yes
105s-nq41-sbqa	04/03/2016	5	Weyna Dega	Hand-pumped borehole	Partial functionality	600	Between 30 Minutes & 1 Hour	No
108x-qwpm-qhuj	28/01/2016	0	Dega	Motorised borehole	Non-functional	NA	NA	NA
109m-2g9q-ga68	12/02/2016	2	Kolla	Open source	Functional	1000	More than 1 Hour	No
109m-2g9q-ga68	20/02/2016	3	Kolla	Open source	Functional	1000	More than 1 Hour	No
109m-2g9q-ga68	05/03/2016	5	Kolla	Open source	Functional	1200	Between 30 Minutes & 1 Hour	No
10e2-guw4-j39y	14/03/2016	7	Open source	Kolla	Functional	400	More than 1 Hour	No
10ea-sncf-2w16	08/02/2016	2	Spring	Dega	Functional	NA	15	Yes
10ea-sncf-2w16	13/02/2016	2	Spring	Dega	Functional	150	20	Yes
10ea-sncf-2w16	19/02/2016	3	Spring	Dega	Functional	200	35	Yes

Example of messy data

- Lets look at our travel time variable in more details

```
unique(ETH$TRAVEL_TIME)
```

```
"Between 30 Minutes & 1 Hour" "Between 30 minutes & 1 hour" "NaN“ "More than 1 Hour"
"Less than 30 Minutes"      "Less than 30 minutes"      "0"                  "35"
NA                      "30"                    "15"                    "9"
"1"                     "120"                   "45"                    "20"
"10"                    "2"                     "50"                    "60"
"2400"                  "80"                    "90"                    "40"
"16"                    "32"                    "25"                    "150"
"180"                   "4"                     "5"                     "12"
"140"                   "75"                    "8"                     "240"
"68"                    "100"                   "70"                    "31"
```

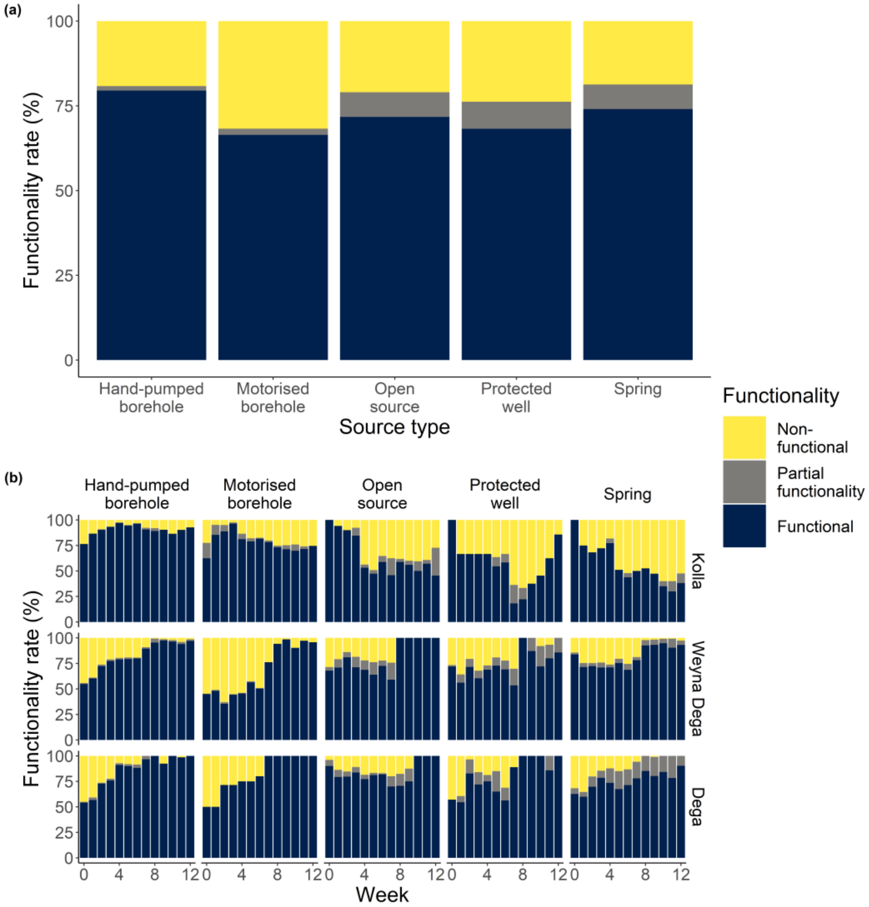
Cleaning messy data

- dplyr provides a consistent set of verbs (it is a grammar) for data manipulation

```
ETH <- ETH %>%
          mutate(TRAVEL_TIME = case_when(
                    is.na(TRAVEL_TIME) == TRUE ~ TRAVEL_TIME,
                    TRAVEL_TIME < 30 ~ "<30 mins",
                    TRAVEL_TIME >= 30 & TRAVEL_TIME < 60 ~ "30-60 mins",
                    TRAVEL_TIME > 60 ~ ">60 mins"))


ETH <- ETH %>%
          mutate(TRAVEL_TIME = case_when(
                    TRAVEL_TIME == "Between 30 Minutes & 1 Hour" ~ "30-60 mins",
                    TRAVEL_TIME == "Less than 30 Minutes" ~ "<30 mins",
                    TRAVEL_TIME == "More than 1 Hour" ~ ">60 mins",
                    TRAVEL_TIME == "NaN" ~ "NA))
```

Using the tidyverse: focusing on water point functionality



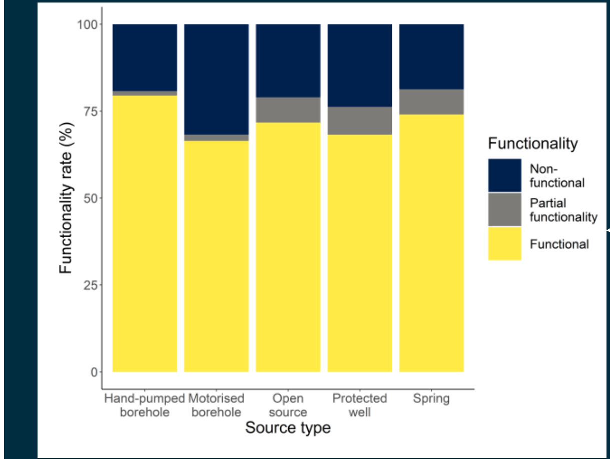
Manipulating tidy data:

grouping, summarising, counting and joining datasets

- Group by variables, count observations and peform calculations in a few lines of code:

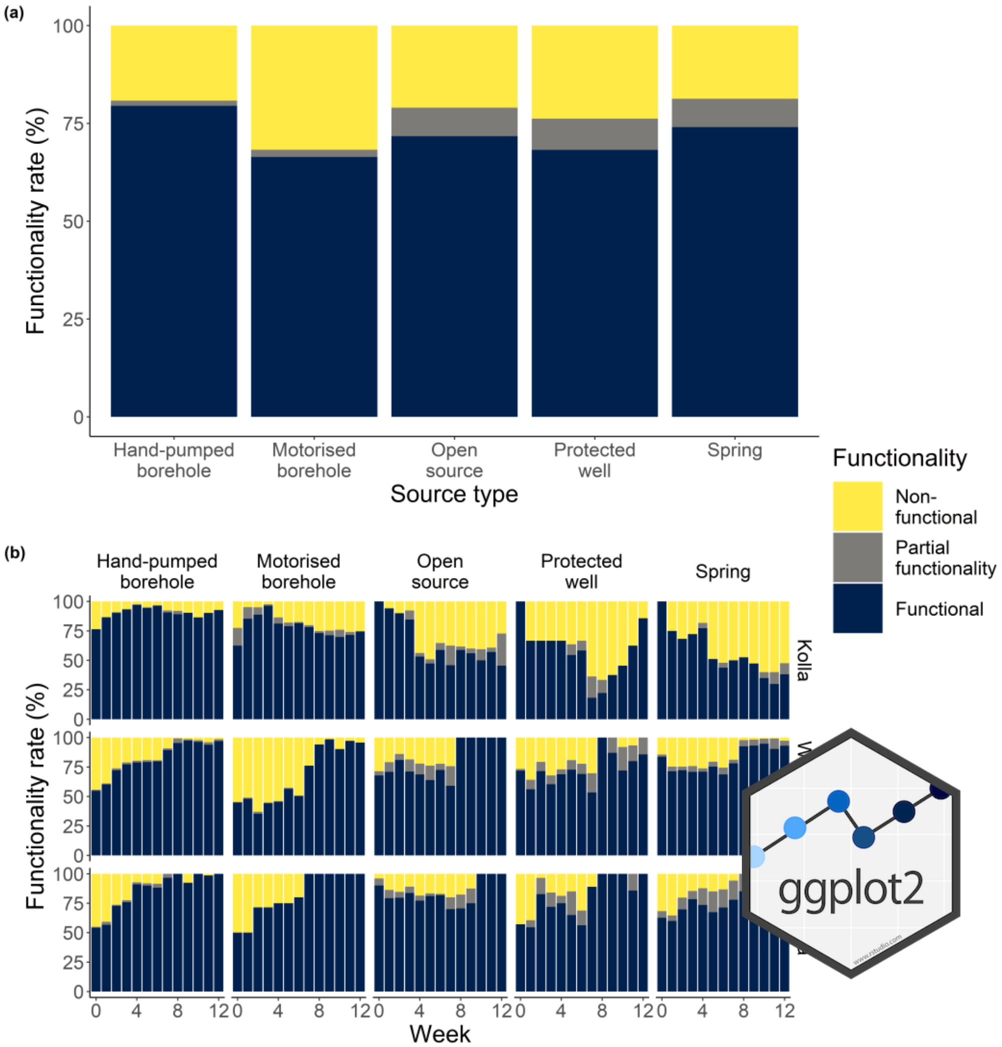
```
FUNC <- ETH %>%
  group_by(TYPE, FCAT) %>%
  tally(name = FCAT_TOTAL) %>%
  mutate(TYPE_TOTAL = sum(TYPE)) %>%
  na.omit() %>%
  mutate(FUNC_RATE = (FCAT_TOTAL /
TYPE_TOTAL )*100)
```

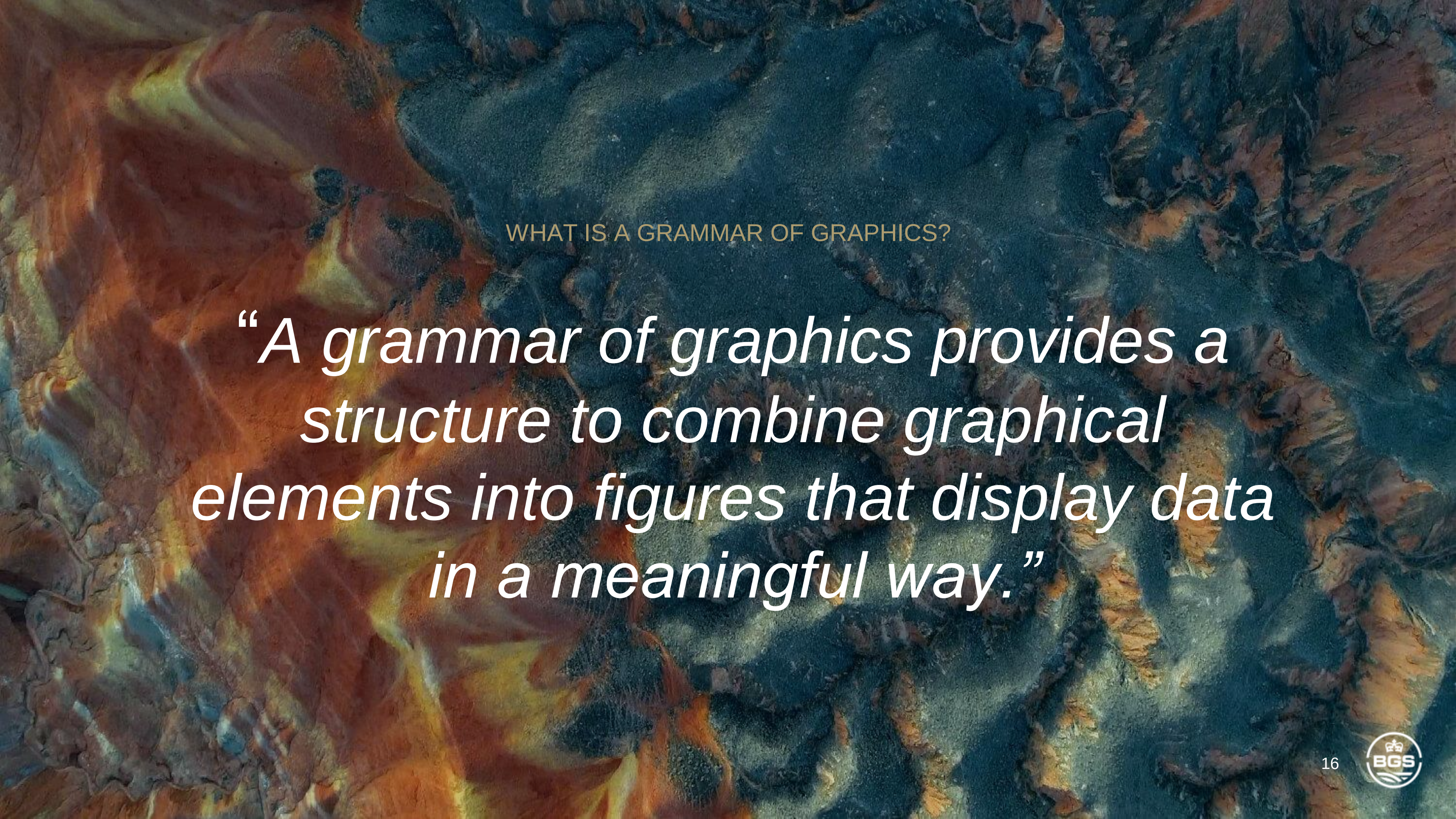
TYPE	FCAT	FCAT_ TOTAL	TYPE_ TOTAL	FUNC_ RATE
Hand-pumped borehole	Non- functional	2314	12053	19.20
Hand-pumped borehole	Partial functionality	157	12053	1.30
Hand-pumped borehole	Functional	9582	12053	79.50
Motorised borehole	Non- functional	978	3081	31.74
Motorised borehole	Partial functionality	56	3081	1.82
Motorised borehole	Functional	2047	3081	66.44



ggplot2

- Designed for data visualisation
- Breaks up graphs into semantic components
- Based on The Grammar of Graphics
- ggplot2 works in four basic steps:
  - assign data
  - map variables to aesthetics (colours, shapes, size, etc)
  - assign graphical elements (lines, points, etc)
  - ggplot2 does the rest





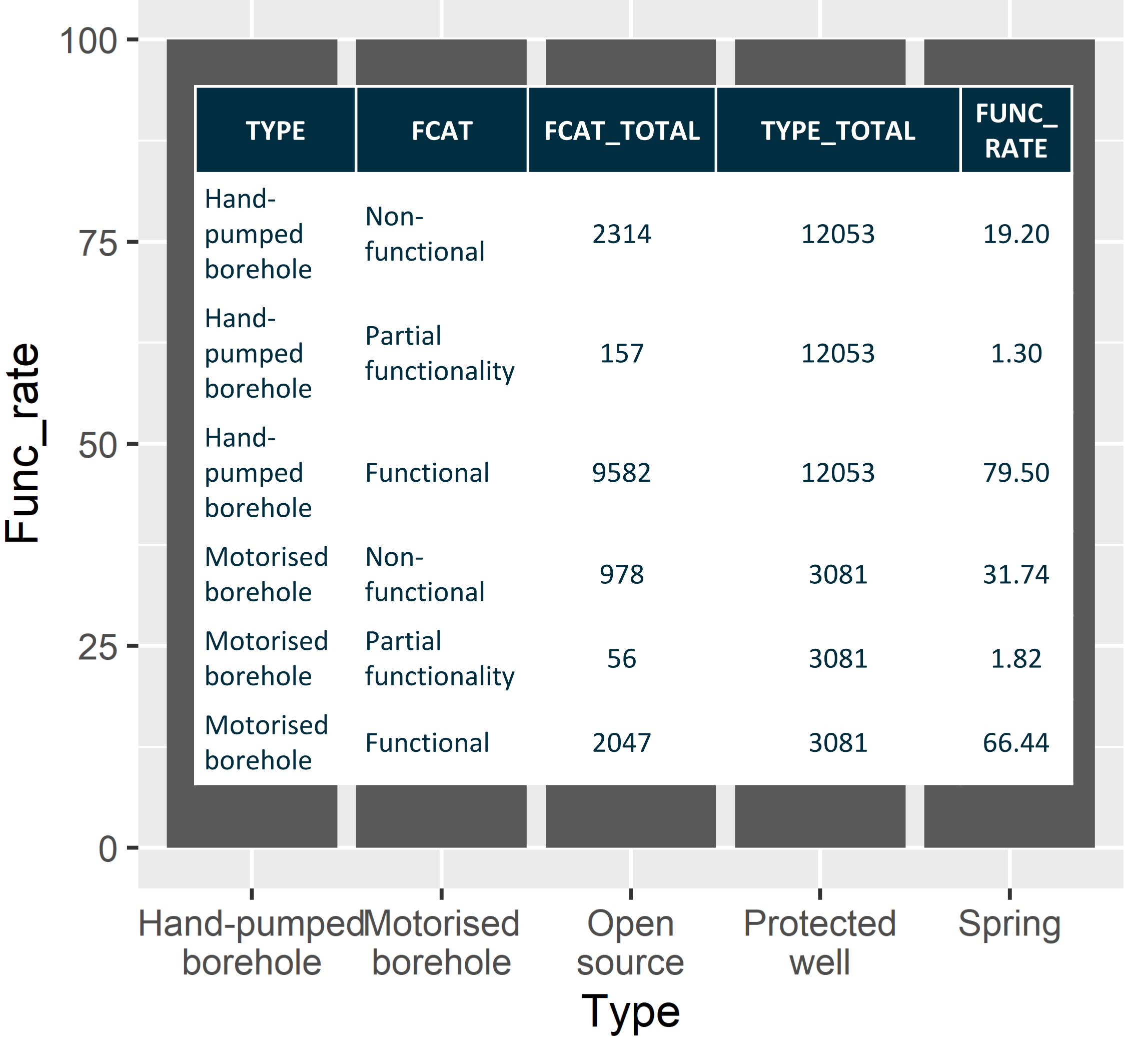
WHAT IS A GRAMMAR OF GRAPHICS?

> *"A grammar of graphics provides a structure to combine graphical elements into figures that display data in a meaningful way."*

Constructing a plot

Plotting functionality of water points

```
ggplot(FUNC,
aes(x=TYPE, y=FUNC_RATE)) +
geom_col()
```

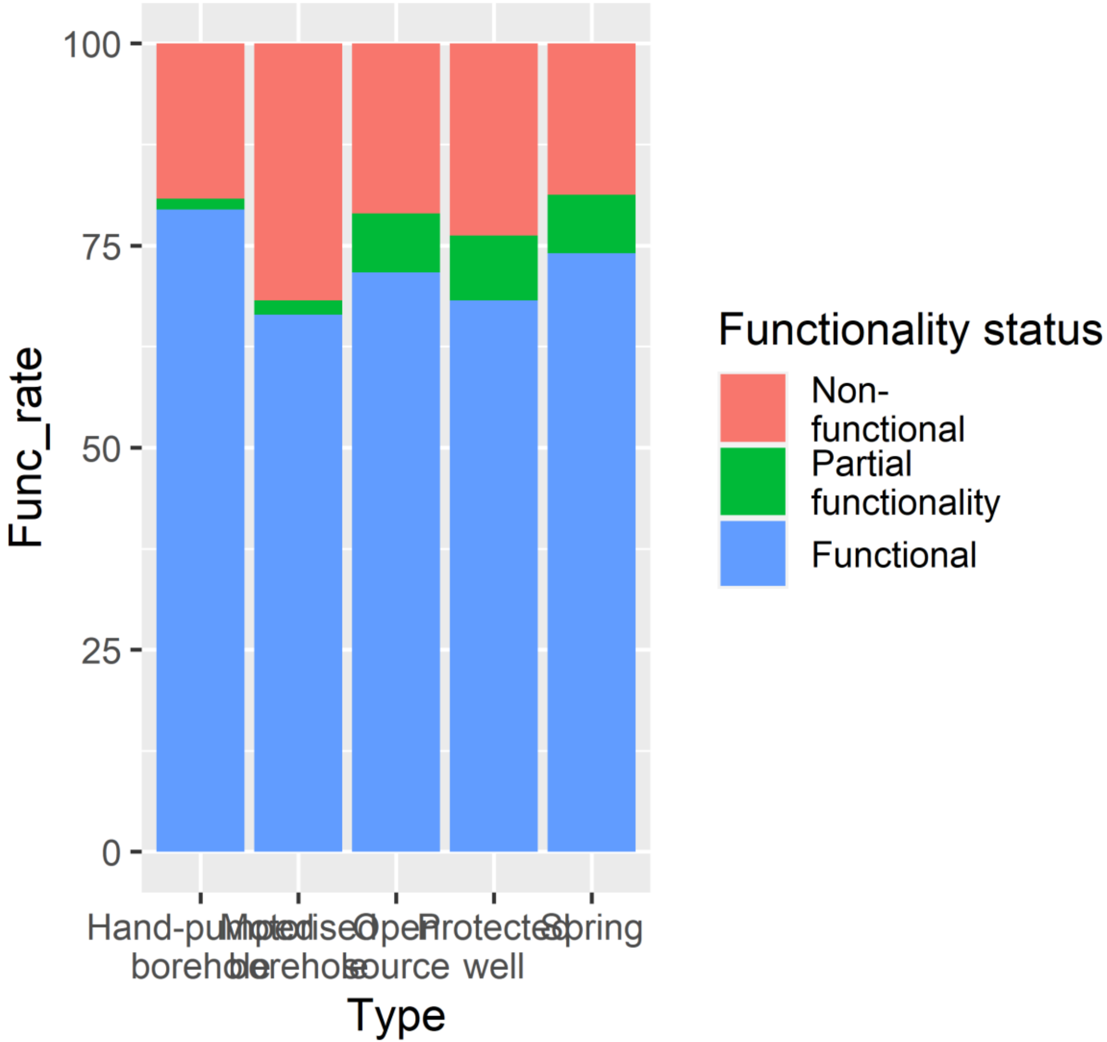


TYPE	FCAT	FCAT_TOTAL	TYPE_TOTAL	FUNC_RATE
Hand-pumped borehole	Non-functional	2314	12053	19.20
Hand-pumped borehole	Partial functionality	157	12053	1.30
Hand-pumped borehole	Functional	9582	12053	79.50
Motorised borehole	Non-functional	978	3081	31.74
Motorised borehole	Partial functionality	56	3081	1.82
Motorised borehole	Functional	2047	3081	66.44

Constructing a plot:

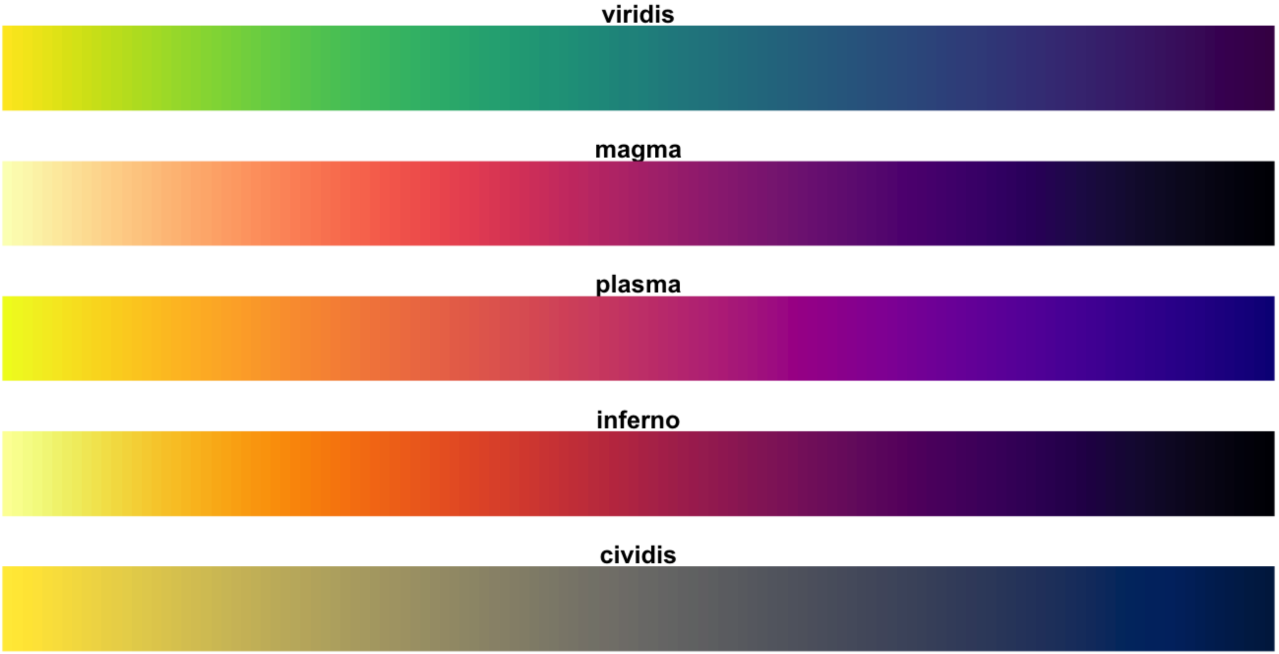
Plotting functionality of water points

```
ggplot(FUNC,
aes(x=TYPE, y=FUNC_RATE,
fill=FCAT)) +
geom_col()
```



The importance of colour

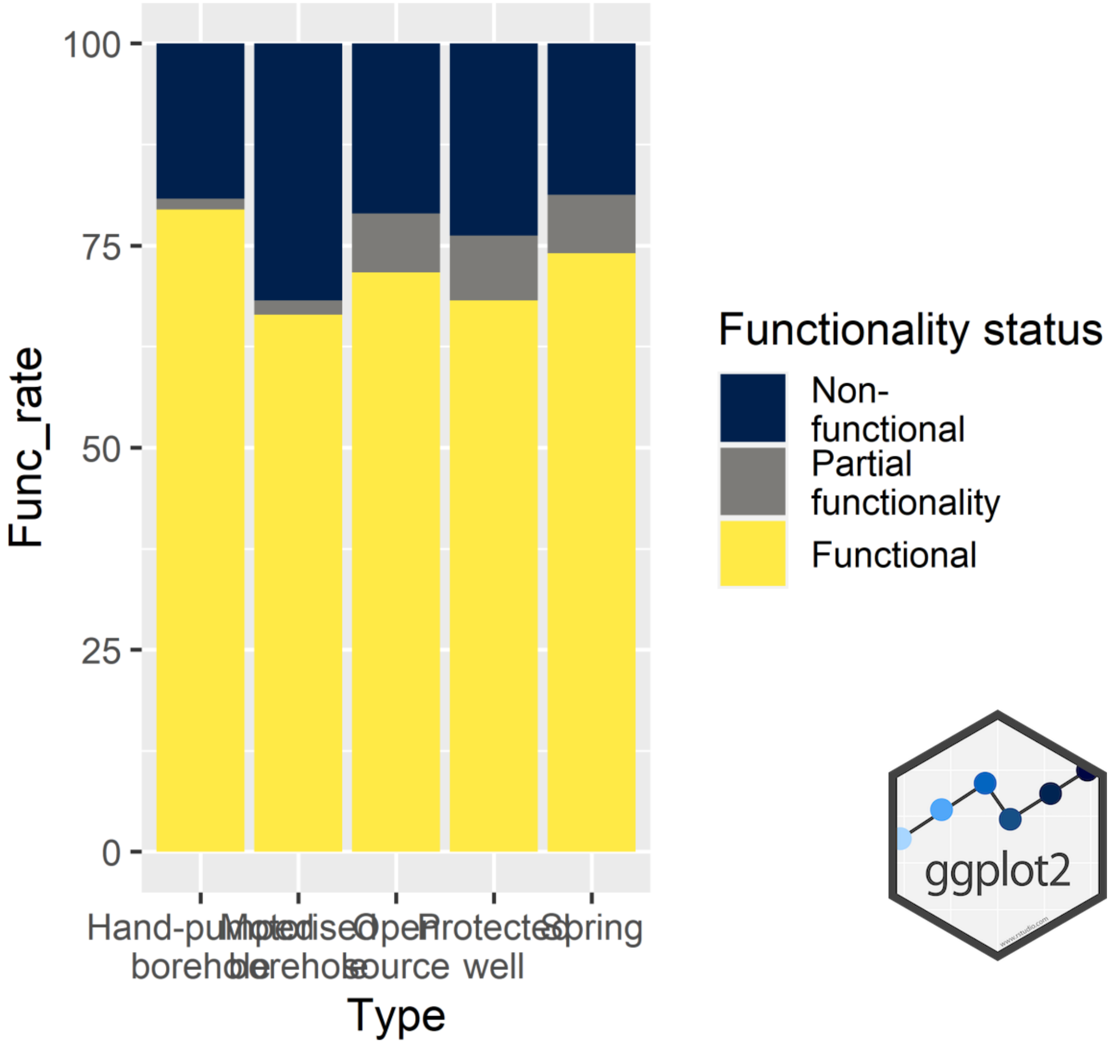
- Colour blind friendly colour scales are available in the R package “viridis”.
- Viridis scales are perceptually uniform in both colour and black-and-white.



Constructing a plot:

Plotting functionality of water points

```
Plot <- ggplot(FUNC,
aes(x=TYPE,
y=FUNC_RATE, fill=FUNC)) +
  geom_col()+
scale_fill_viridis(discrete=
TRUE, option = "E")
```

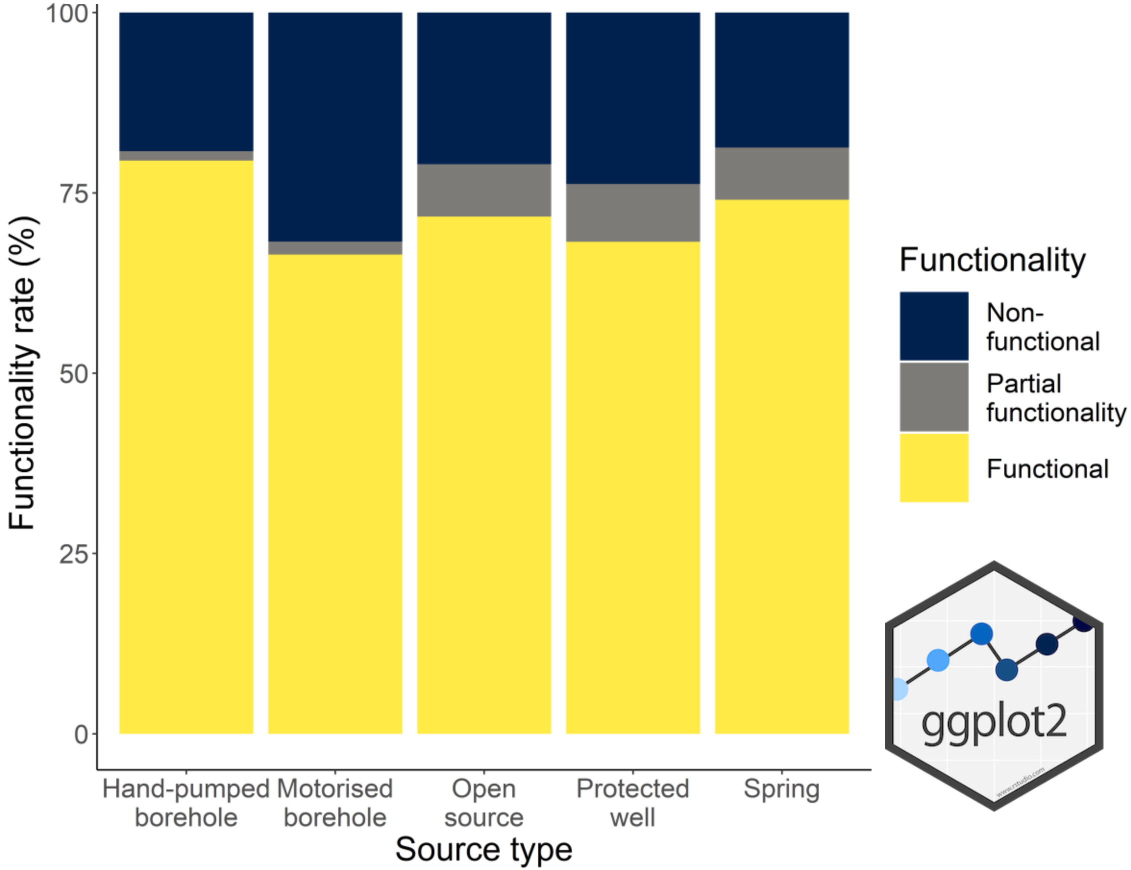


Constructing a plot:

Plotting functionality of water points

- Add axis titles and legend titles, change themes:

```
Plot +
  ylab("Functionality rate (%)") +
  xlab("Source type") +
  theme_classic() +
theme(element_text(size=20), +
  guides(fill = guide_legend(title
= "Functionality"))
```

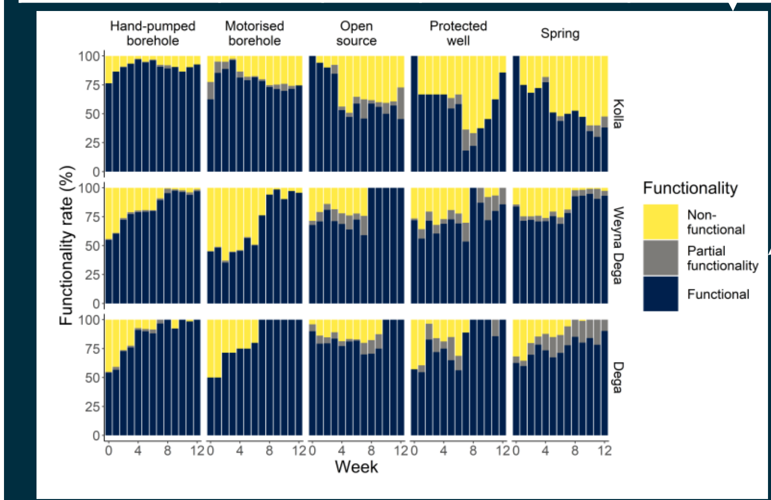


Further data manipulation based on variables

- Group by MORE variables, count observations and perform calculations in JUST as few lines of code as before:

```
FUNC <- ETH %>%
    group_by(TYPE, WEEK, AREA, FCAT) %>%
    tally(name = FCAT_TOTAL) %>%
    mutate(TYPE_TOTAL = sum(TYPE)) %>%
    na.omit() %>%
    mutate(FUNC_RATE= (FCAT_TOTAL/ TYPE_TOTAL )*100)
```

TYPE	WEEK	AREA	FCAT	FCAT_TOTAL	TYPE_TOTAL	FUNC_RATE
Hand-pumped borehole	0	Kolla	Non-functional	34	144	23.61
Hand-pumped borehole	0	Kolla	Functional	110	144	76.39
Hand-pumped borehole	0	Weyna Dega	Non-functional	313	708	44.21
Hand-pumped borehole	0	Weyna Dega	Partial functionality	7	708	0.99
Hand-pumped borehole	0	Weyna Dega	Functional	388	708	54.80
Hand-pumped borehole	0	Dega	Non-functional	114	253	45.06

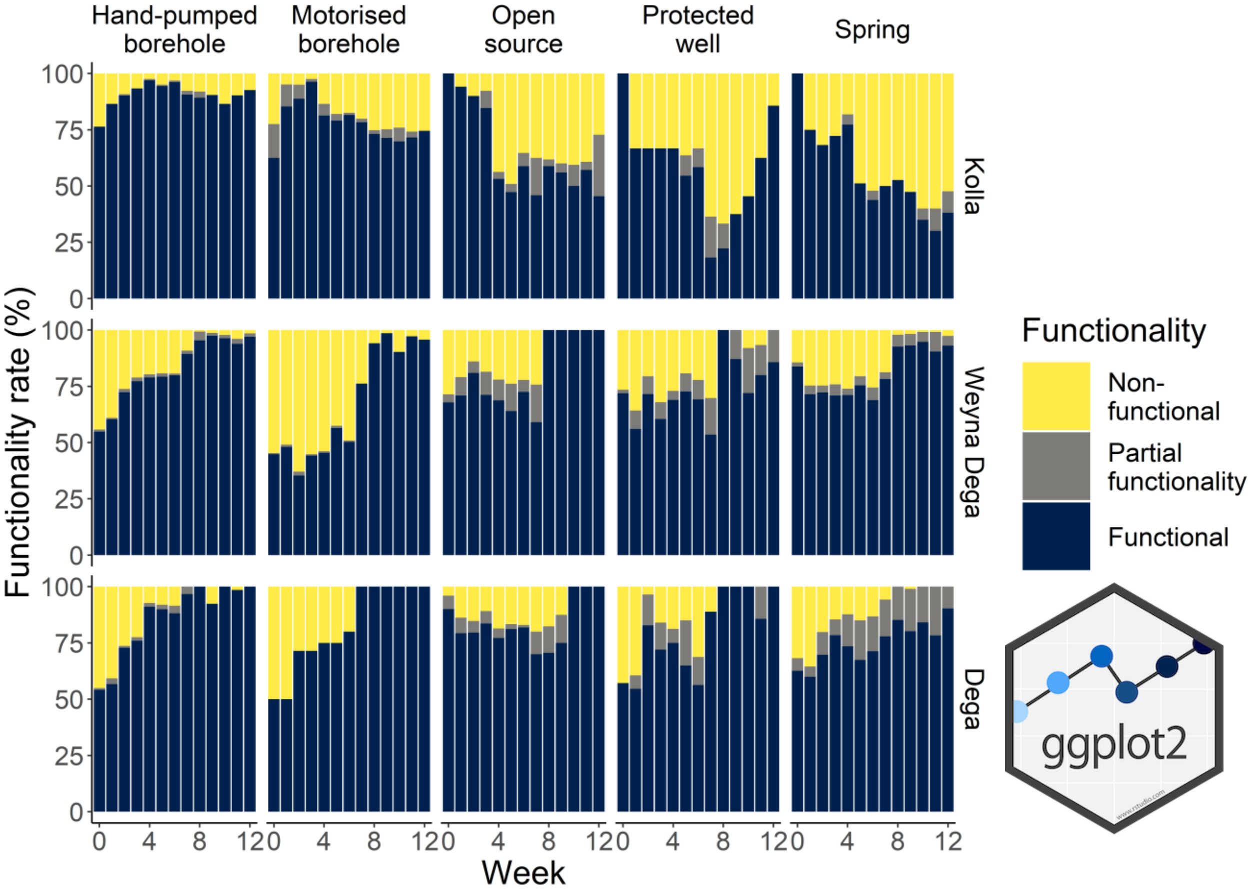


Plotting more variables using facets

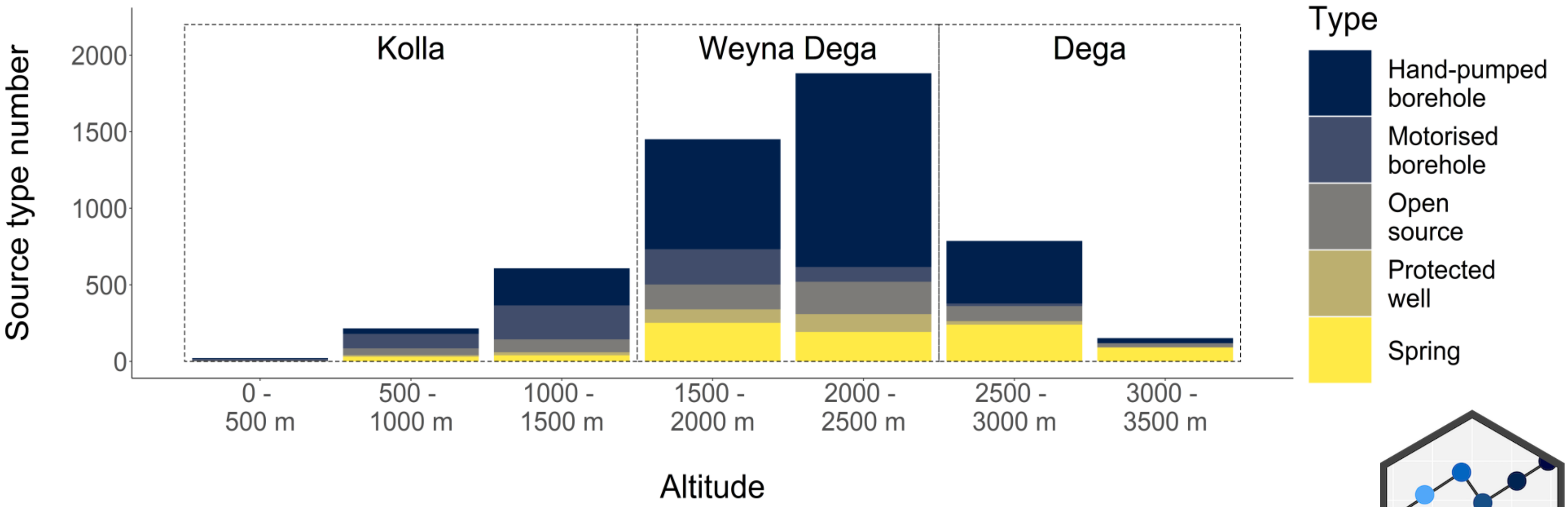
- To plot facets we use:
  - the previous code,
  - Change the x variable in aes()
  - add one additional line

ggplot(FUNC, aes(**x=WEEK**, y=FCAT, fill = FUNC_RATE) +

facet_grid(AREA ~ TYPE)



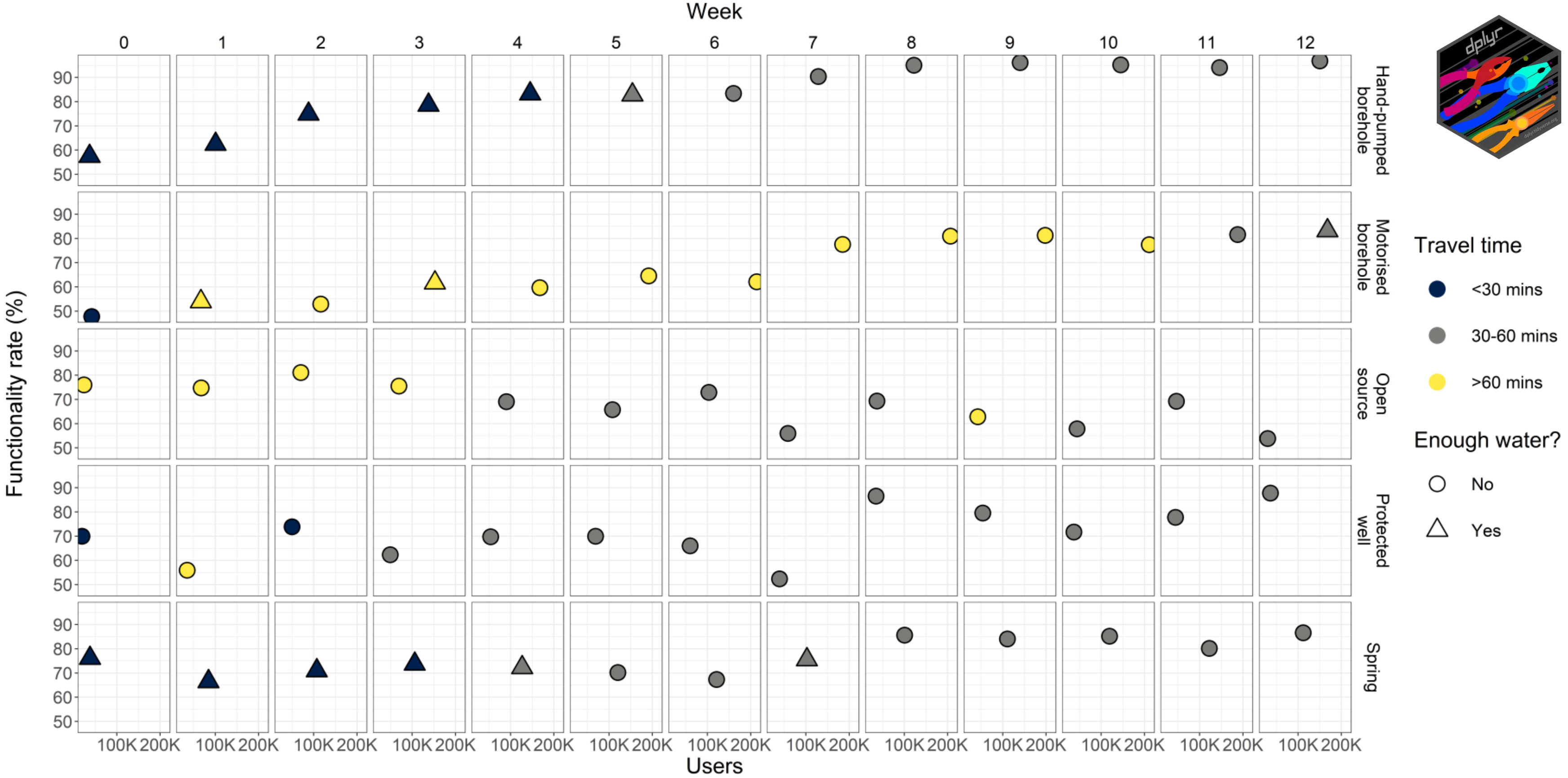
Including other features on plots



geom_rect(aes(linetype = "Kolla", xmin=0.5, xmax=3.5, ymin=0, ymax=2200), fill=NA, col="black") +

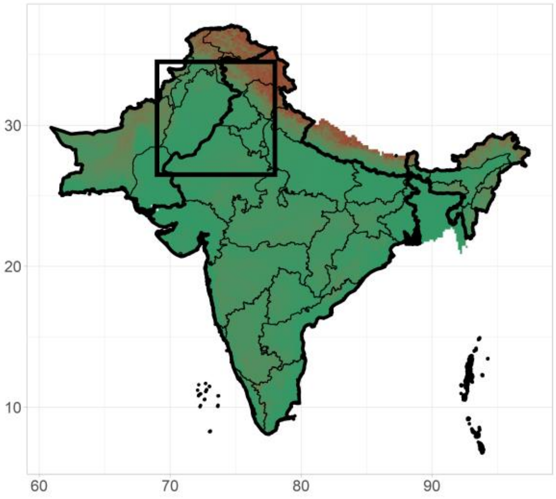
geom_text(aes(x=2,y=2050,label="Kolla"), size = 12)

Bringing it all together: summarising performance of rural water supplies in one plot



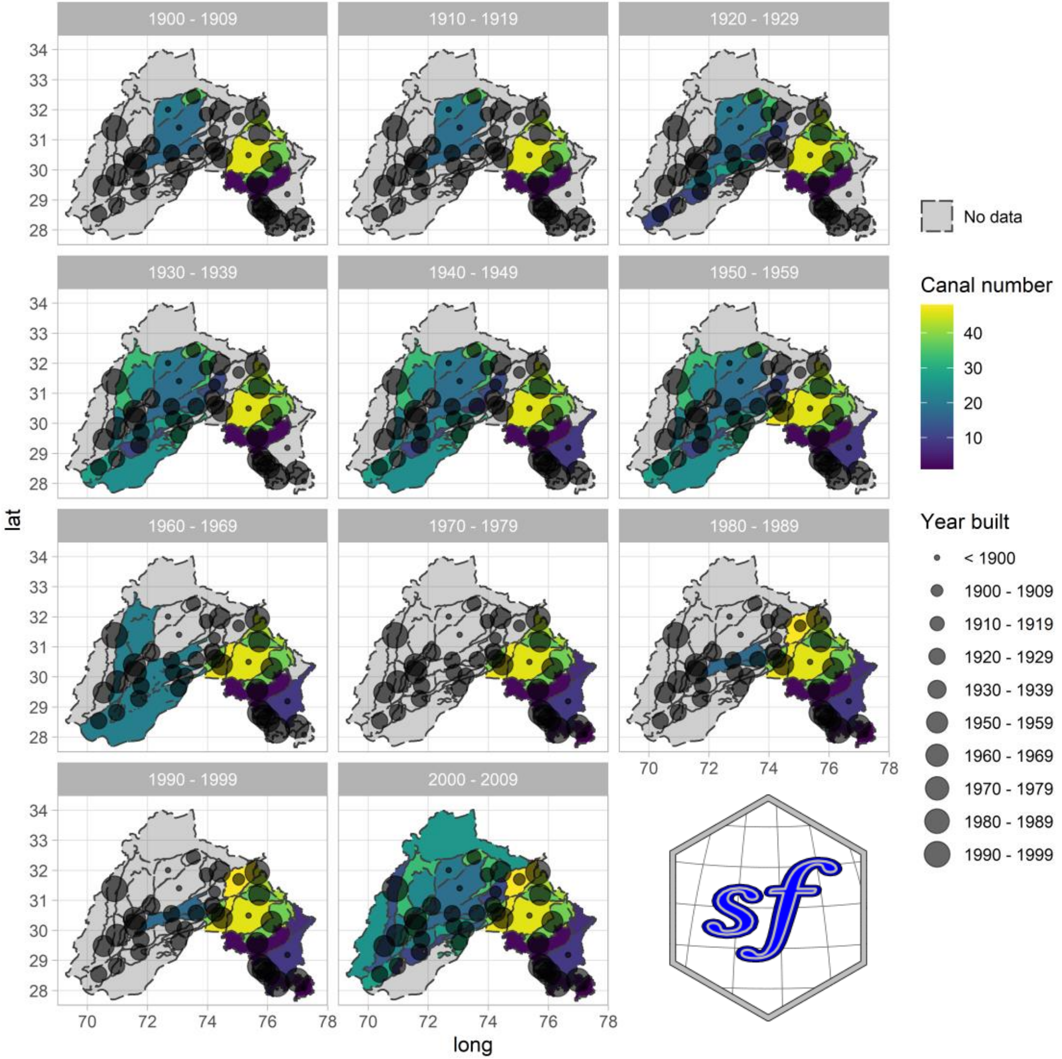
Spatial data

- Same principles apply



dplyr

ggplot2



sf

Summary and conclusions

- The tidyverse uses an underlying data structure and grammar
- These principles make it easy to work with and visualise (large) datasets
- dplyr provides the tools for manipulating (large) datasets
- ggplot offers the user huge control over how that data is visualised
- Both tools share an easy to understand syntax, making them easy to learn and use
- In combination they offer a powerful way to manipulate and visualise (pretty much any) dataset

BGS British Geological Survey

THANK YOU

Any questions?