Day 4: R Markdown and Data Viz

Data Carpentries for Social Sciences

Leiden University, TU Delft, Erasmus University Rotterdam, Vrije Universiteit

2022-10-28 *(Updated: 2022-10-28)*

R Markdown

Why R Markdown?

R Markdown allows you to **seamlessly combine executable R code, its output, and text** in a single document.

These documents **can be converted to multiple static and dynamic output formats**, including PDF (.pdf), Word (.docx), and HTML (.html).

The benefit of a well-prepared R Markdown document is **full reproducibility**.

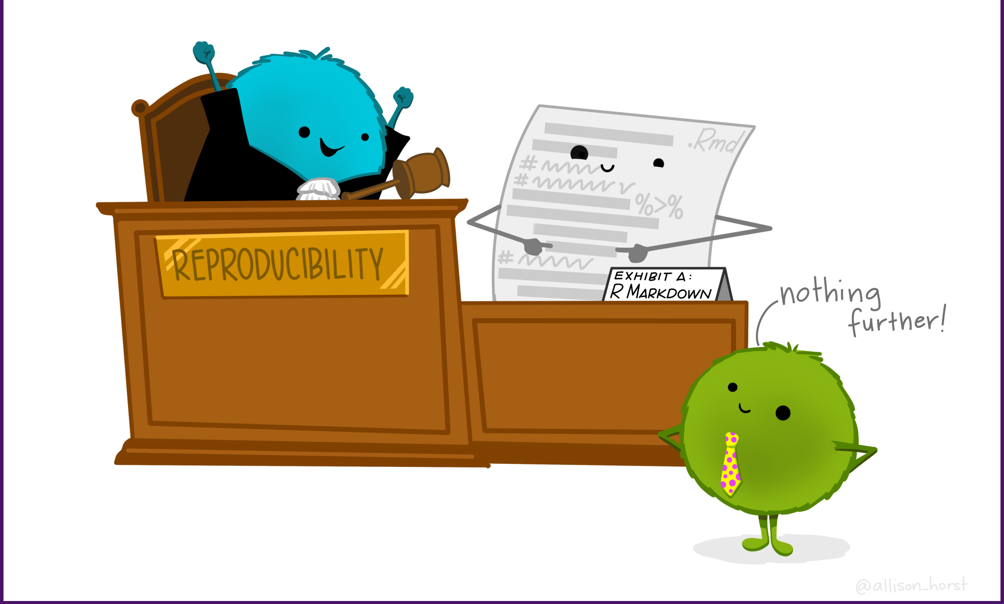


Image: Allison Horst

Why R Markdown?

Full reproducibility also means that, if you
are able to add more data to your analysis, you will be able to recompile the report
without making any changes in the actual document.

And if you make a mistake…

GIPHY

R Markdown Exercises

Exercise 1

⌚ 4 mins

Play around with the different options in the chunk with the code for the table, and re-Knit to see what each option does to the output.

What happens if you use `eval=FALSE` and `echo=FALSE`?

What is the difference between this and `include=FALSE`?

04:00

Exercise 1: Solution

Create a chunk with `eval=FALSE, echo=FALSE`

then create another chunk with `include=FALSE` to compare.

`eval=FALSE, echo=FALSE` will neither run the code in the chunk, nor show the code in the knitted document. The code chunk essentially doesn't exist in the knitted document…

…whereas `include=FALSE` will run the code and store the output for later use.

See a comprehensive list of chunk options here

Data Visualisation

Why ggplot2?

...because these are 'base' plots

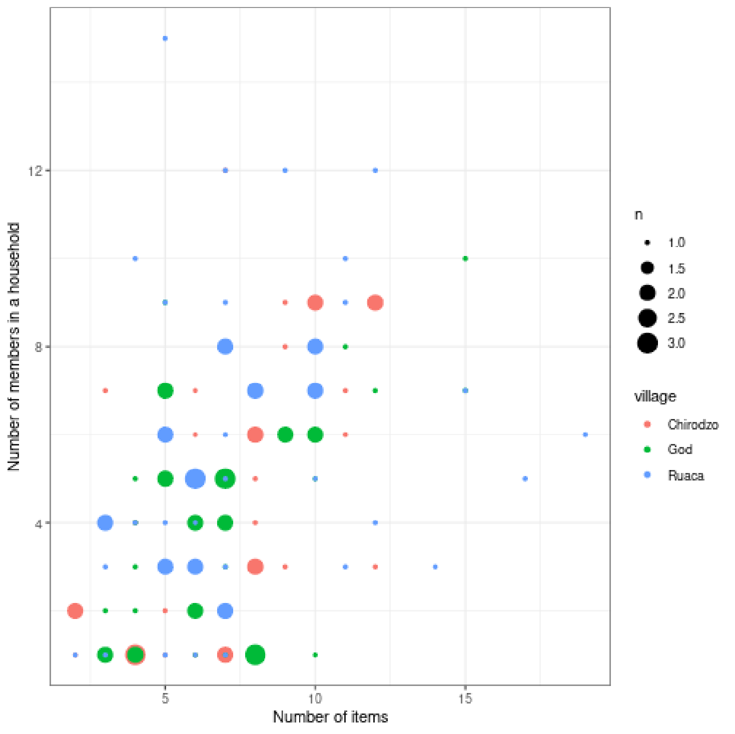
```r
plot(number_items ~ no_membrs,
     interviews_plotting,
     col = "blue")
```

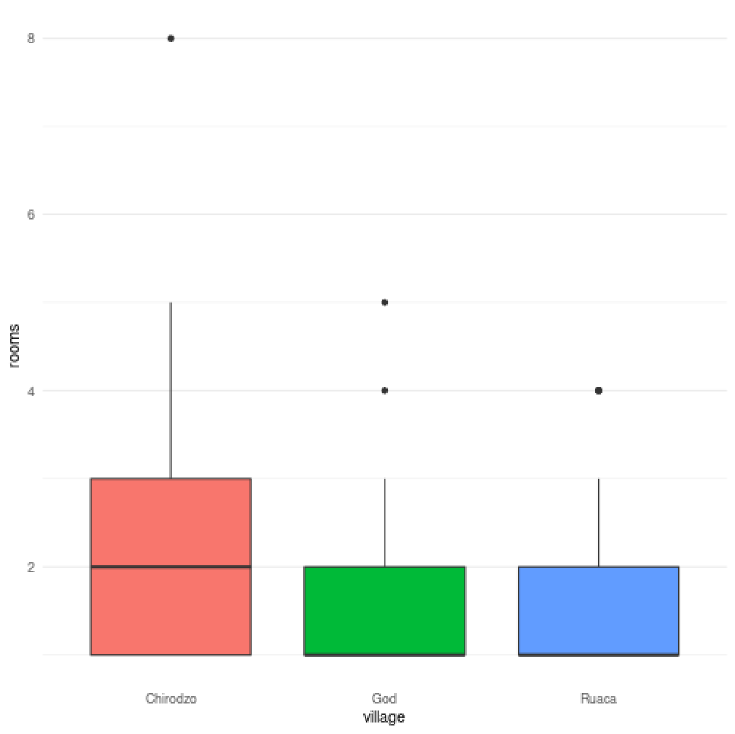
```r
boxplot(rooms ~ village,
        interviews_plotting,
        col = c("blue", "green",
```

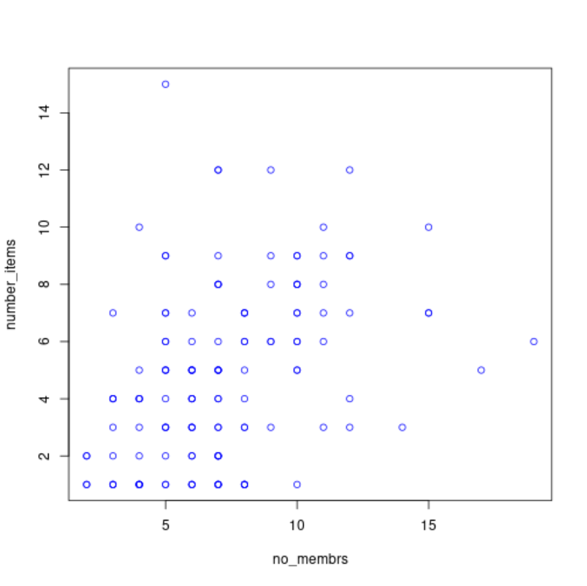
...and these are ggplots 😎

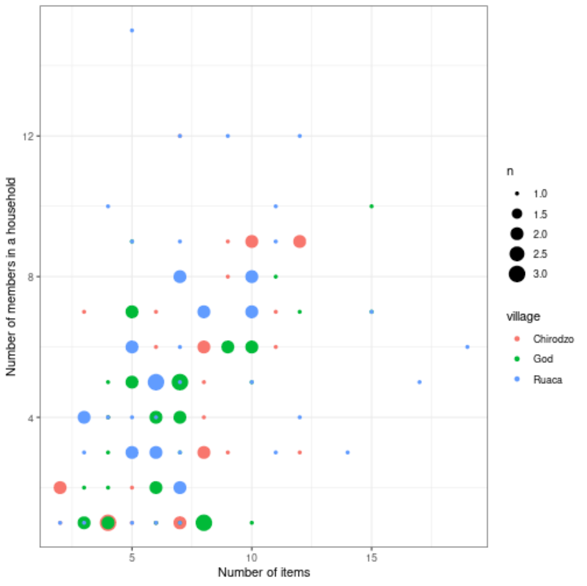
```
interviews_plotting %>%
   ggplot(aes(x = no_membrs, y =
      geom_count() +
      theme_bw() +
      labs(x = "Number of items",
           y = "Number of members
```

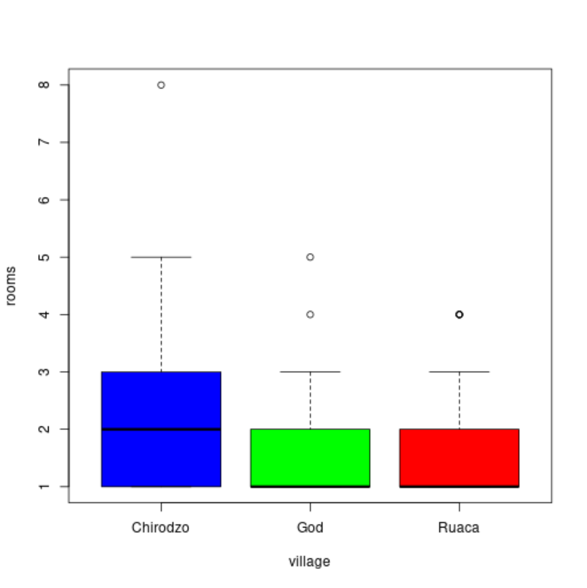
```
interviews_plotting %>%
  ggplot(aes(x = village, y = roo
    geom_boxplot() +
    theme_minimal() +
    theme(legend.position = "none
        panel.grid.major.x = elem
```

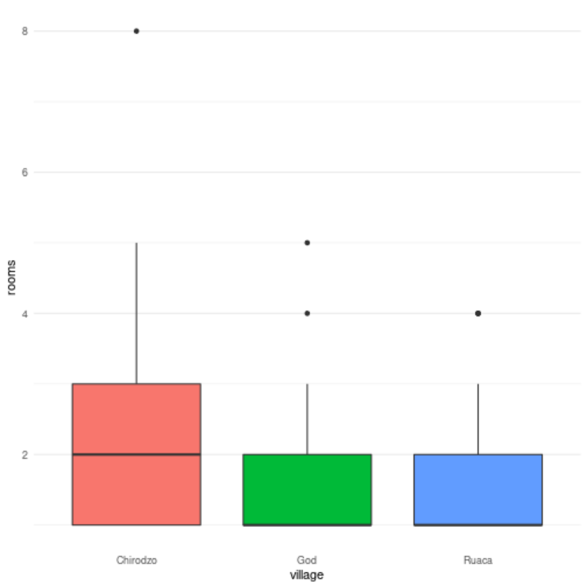





number_items
2
4
6
8
10
12
14
5
10
15
no_membrs


Number of members in a household
4
8
12
5
10
15
Number of items
n
1.0
1.5
2.0
2.5
3.0
village
Chirodzo
God
Ruaca


rooms
1
2
3
4
5
6
7
8
Chirodzo
God
Ruaca
village


rooms
2
4
6
8
Chirodzo
God
Ruaca
village

ggplot2

ggplot2 is a package (included in **tidyverse**) for creating highly customisable plots that are built step-by-step by adding layers.

The separation of a plot into layers allows a high degree of flexibility with minimal effort.

```
<DATA> %>%
    ggplot(aes(<MAPPINGS>)) +
    <GEOM_FUNCTION>() +
    <CUSTOMISATION>
```

More artwork by Allison Horst

Data Visualisation Exercises

Exercise 2

🕥 **6 mins**

Create a new code chunk with the label `rooms-village-scatter`.

Create a scatter plot of `rooms` by `village` with the `respondant_wall_type` showing in different colours.

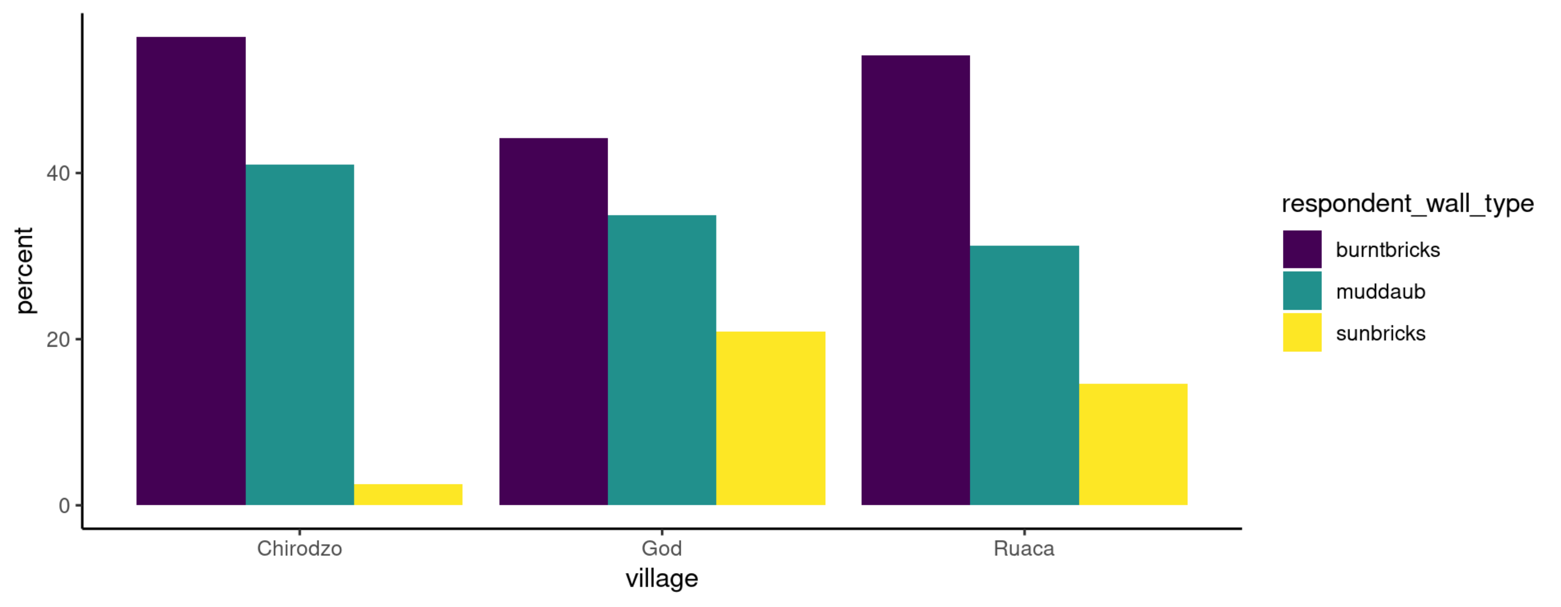
Does this seem like a good way to display the relationship between these variables?

What other kinds of plots might you use to show this type of data?

06:00

Exercise 2: Solution

~~~
```{r rooms-village-scatter}
percent_wall_type %>%
    ggplot(aes(x = village, y = percent, fill = respondent_wall_type))
    geom_bar(stat = "identity", position = "dodge") +
    theme_classic() +
    scale_fill_viridis_d() # add colourblind-friendly palette
```
~~~



Captioning

Now that we have created the plot, we can also create a caption.

e.g.

~~~
```{r rooms-village-scatter, fig.cap="This plot shows the relationship
but doesn't do a very good job at it."}
percent_wall_type %>%
    ggplot(aes(x = village, y = percent, fill = respondent_wall_type))
    geom_bar(stat = "identity", position = "dodge") +
    theme_classic() +
    scale_fill_viridis_d() # add colourblind-friendly palette
```
~~~

Exercise 3

🕡 4 mins

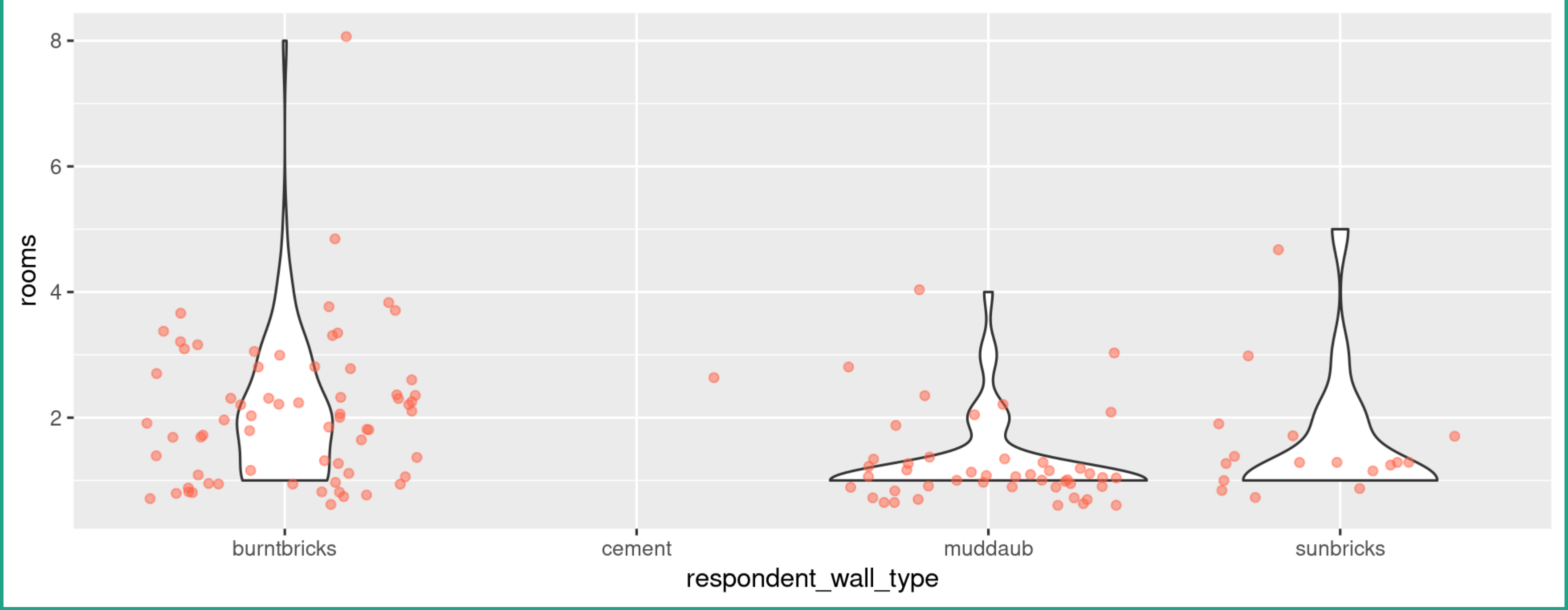
Boxplots are useful summaries, but hide the shape of the distribution. For example, if the distribution is bimodal, we would not see it in a boxplot.

Replace the box plot with a violin plot
see `geom_violin()`

05:00

Exercise 3: Solution

```
interviews_plotting %>%
  ggplot(aes(x = respondent_wall_type, y = rooms)) +
  geom_violin() +
  geom_jitter(alpha = 0.5, color = "tomato")
```



Exercise 4

🕓 **8 mins**

Create a bar plot showing the proportion of respondents in each village who are or are not part of an irrigation association (`memb_assoc`).

Include only respondents who answered that question in the calculations and plot.

Which village had the lowest proportion of respondents in an irrigation association?

Hint

```r
percent_memb_assoc <- interviews_plotting %>%
  filter(!is.na(memb_assoc)) %>%
  count(village, memb_assoc) %>%
  group_by(village) %>%
  mutate(percent = (n / sum(n)) * 100) %>%
  ungroup()
```

10:00

Exercise 4: Solution

```
percent_memb_assoc <- interviews_plotting %>%
  filter(!is.na(memb_assoc)) %>%
  count(village, memb_assoc) %>%
  group_by(village) %>%
  mutate(percent = (n / sum(n)) * 100) %>%
  ungroup()

percent_memb_assoc %>%
   ggplot(aes(x = village, y = percent, fill = memb_assoc)) +
    geom_bar(stat = "identity", position = "dodge")
```

Exercise 5

🕘 **4 mins**

Experiment with at least two different themes. Build the previous plot using each of those themes.

Which do you like best?

Hint

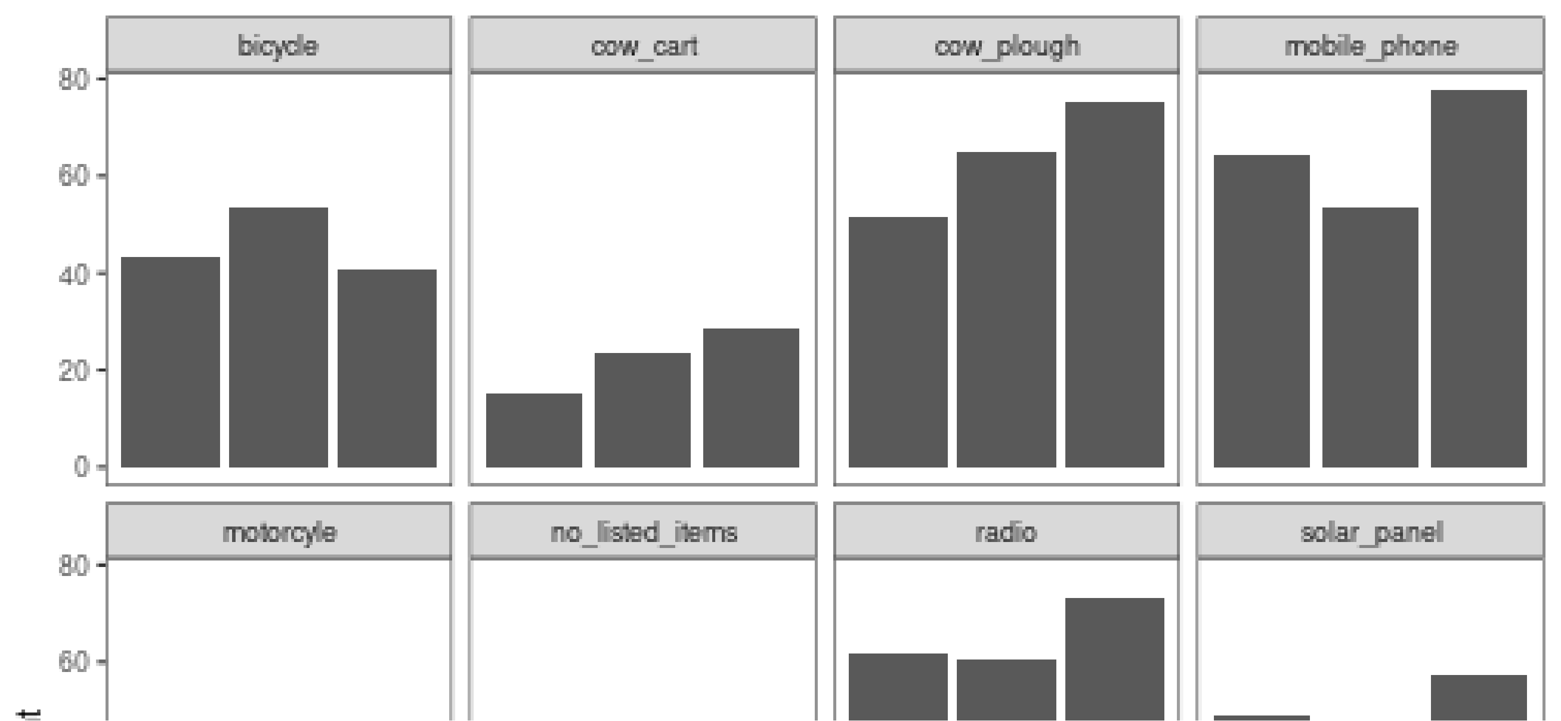
theme_minimal
theme_void
theme_classic

theme_dark
theme_grey
theme_light

05:00

Exercise 5: Solution

```
percent_items %>%
    ggplot(aes(x = village, y = percent)) +
    geom_bar(stat = "identity", position = "dodge") +
    facet_wrap(~ items) +
    theme_bw() +
    theme(panel.grid = element_blank())
```



Exercise 5: Solution

```
percent_items %>%
    ggplot(aes(x = village, y = percent)) +
    geom_bar(stat = "identity", position = "dodge") +
    facet_wrap(~ items) +
    theme_bw() +
    theme(panel.grid = element_blank())
```