

Objective: explore how some mammals sleep

Data: msleep

The dataset is part of the package `ggplot2` and comes as a `tibble`

Display `msleep` and its columns

```
```{r}
```

Write your answer here

```
## First countings

#### What are the column types? Summarise by counting how many are they per type

```{block, opts.label = "tip"}
The `dplyr` function `across()` can apply a dedicated function to all columns. Here, it would be
efficient to use it inside a `summarise()`. A subsequent pivoting will help to to count the number of each type.
```{r}
```

Write your answer here

There are 5 columns with the type (categories as character) and the 6 last are double/float

```
#### How many animals are present? Per order, which are the 3 dominant ones?

```{r}
# Write your answer here
#There are 83 animals.
```{r}
```

Write your answer here

Per order, the 3 dominant ones are Rodentia, Primates and Carnivora

```
#### Compute the mean of the body weight and total sleep hours for the 3 main orders

```{block, opts.label = "tip"}
look in the help pages for `msleep` to find out the definition of each column.
`slicing_max()` will you to filter the 3 top rows for the relevant column.
```{r}
```

Write your answer here

```
#### We observe some extreme difference in body weight, plot the distribution of body weights per order, and colored by the `vore` variable. Do herbivores are really the biggest animals?

```{block, opts.label = "tip"}
a basic plot would be dots colored by `vore`, and the order on the x-axis.
But some steps makes a much better graph:

- `order` on the y-axis makes the labels trivial to read
- sorting the body mass by their median, order nicely the order. Use `fct_reorder()` that does the job for you
- we face a huge range of data, using `scale_x_log10()` keep linear scale labels but on a log-scale.
- `annotation_logticks()` for the x-axis offers helpful ticks for the log-scale.
- dots are overlapping, using `geom_jitter()` a little randomness help seeing all dots
- add `geom_violin()` to better see the range of each distribution. `trim = FALSE` might look better.
- being explicit in the `labs()` to describe axis, titles help the reader
```{r}
```

Write your answer here

```
```{r}
# Write your answer here
```

Body / Brain weight

It seems reasonable to expect some correlation between the brain and body weights.

Plot the relationship between the body mass and brain mass

```
```{r}
```

Write your answer here

```
##### We see some extreme values, solve that issue and conclude if the linear trend is true.

```{block, opts.label = "tip"}
You can get rid of the warning concerning the 27 missing values using by filtering upfront `ggplot()`
or use the `na.rm` argument in `geom_point()`

```{r}
```

Write your answer here

```
##### Explore if more variables are linked, better log-scaling them first.

```{block, opts.label = "tip"}
Using the package `GGally` and the function `ggpairs()` helps.
Select only the columns that are numeric doubles.

```{r}
```

Write your answer here

```
##### Do you observe some interesting patterns on top of brain/body weight?

```{r}
# Write your answer here
```

Confirm the linear time constraint of the trade-off between `sleep_total` and `awake`

```
```{r}
```

Write your answer here

```
##### Look at the relationship between REM sleep and sleep total.

```{block, opts.label = "tip"}
plot the REM in function of sleep total.
it is nice to add some animal names. To avoid overlap of text, the package `ggrepel` is doing a fantastic job.
See the little example below. We use the `data =` argument that can take a function to subset data points

```{r} library(ggrepel) mtcars %>% rownames_to_column(var = "name") %>% ggplot(aes(x = wt, y = mpg, colour = factor(am))) + geom_point() + geom_label_repel(data = function(x) filter(x, cyl == 4, hp > 5), # avoid the letter a in the legend aes(label = name), show.legend = FALSE) + labs(title = "How car weight affects gas consumption", x = "weight", y = "mileage per US gallon", colour = "Automatic\nManual", caption = "source: mtcars") + theme(plot.caption = element_text(face = "italic"), plot.caption.position = "plot") + # this will allow the text outside of the plot panel coord_cartesian(clip = "off")

```{r}
# Write your answer here
```

Conservation status

Most of the animals present in this table possess a conservation status. Which means that they are more or less in danger for their own survival because of human behavior and activities.

Find out how many of each category of conservation status are present

```
```{r}
```

Write your answer here

```
##### Using the unique values of `conservation`, find out their meaning and create a `tibble` of correspondences between abbreviations and full names. Assign to the name `cons_status`

Right now only `domesticated` is clearly stated.

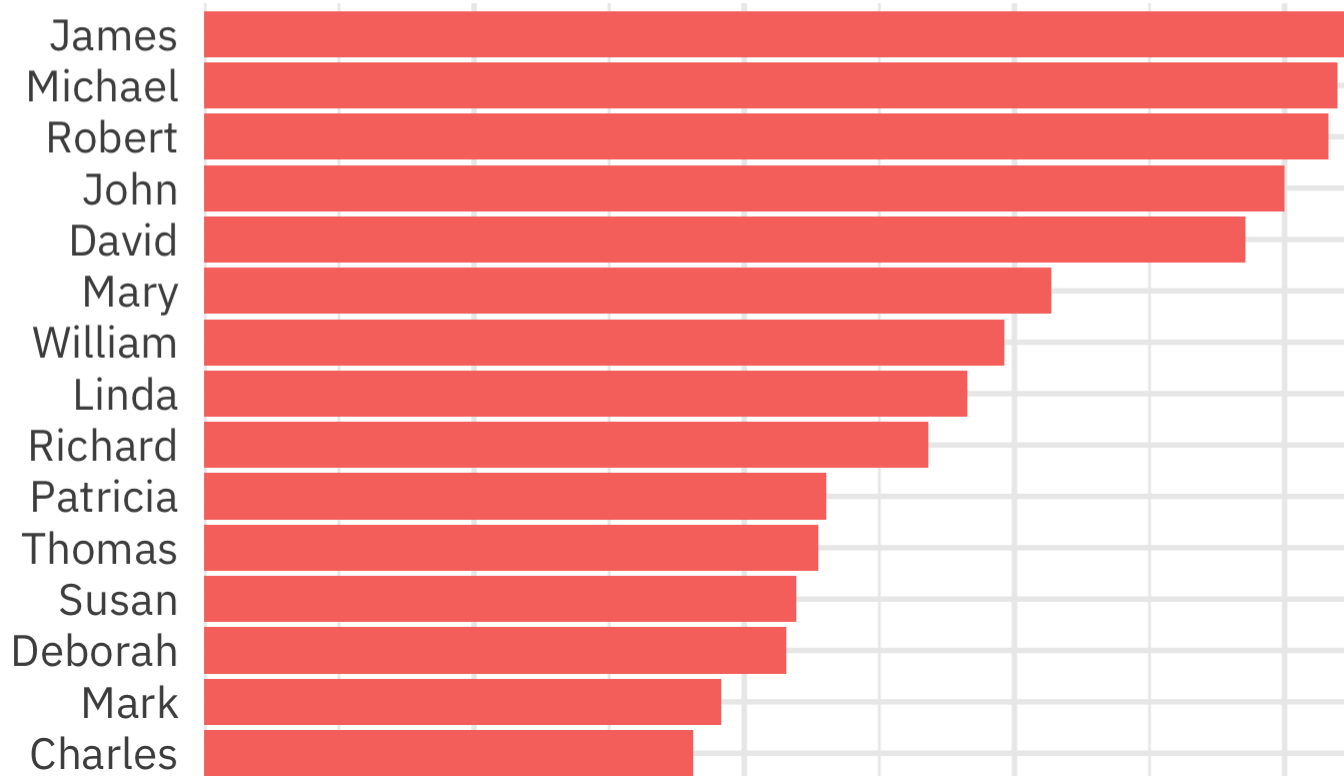
```{r}
# Write your answer here
```

What endangered animals are eating?

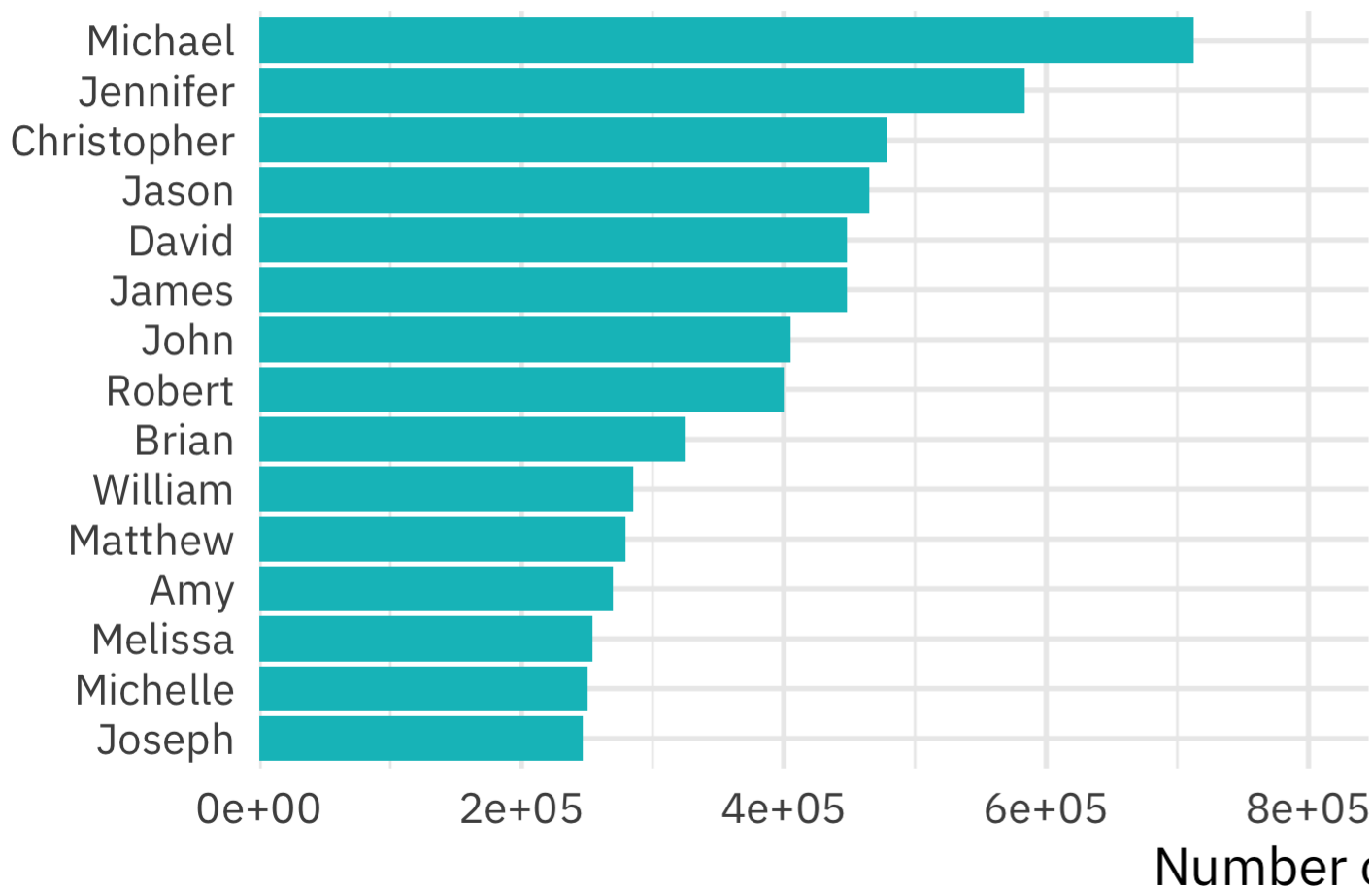
We want to get a horizontal barplots of conservation status, faceted by `vore`. However, of course, the conservation status won't be the same across alimentary regime. Read the [blog post](#) by Julia Silge.

That's a similar picture we would like to obtain in the end:

What were the most common k  
Via US Social Security Administration  
1950



1970



First, count the number of each conservation status per `vore` and join the table `cons_status` to obtain meaningful names. Assign to the name `ms_cons`.

```{block, opts.label = "warning"} Mind if you need `ungroup()`

```
```{r}
# Write your answer here
```

Install the `tidytext` package and follow Julia [instructions](#) to show the count of conservations status faceted by `vore`

```{r}

Write your answer here

```
#### What can you conclude?

```{r}
# Write your answer here
```