Working with: factors

Data Tidying

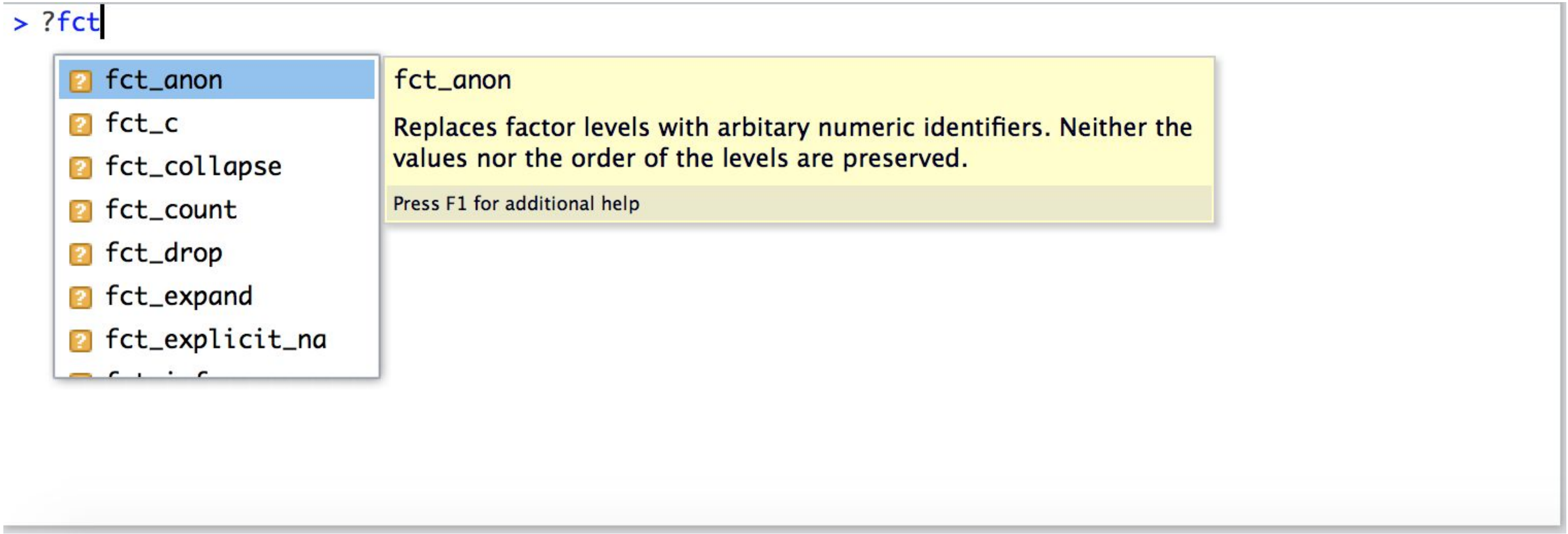
Factors are for **categorical variables**.

Categorical variables: there are a limited number of possible values any data point can take

Example: **months**

- There are 12 possible months in a calendar year
- For a factor variable containing information about month, there are only 12 possible values each data point can have

https://forcats.tidyverse.org/



```
> ?fct
```

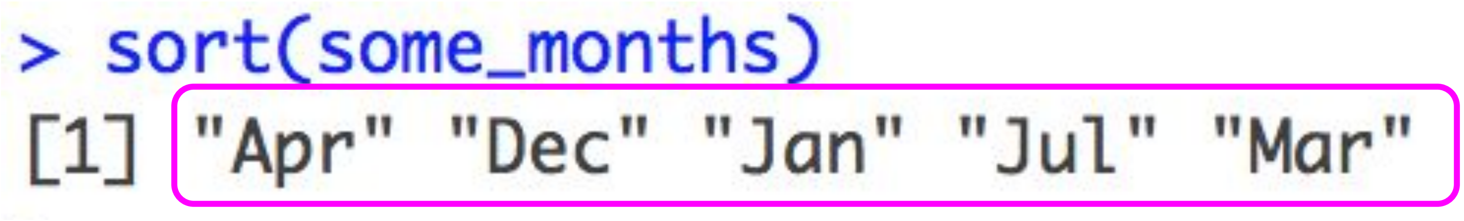
fct_anon

Replaces factor levels with arbitary numeric identifiers. Neither the values nor the order of the levels are preserved.

Press F1 for additional help

```
## all 12 months
all_months <- c("Jan", "Feb", "Mar", "Apr", "May", "Jun", "Jul", "Aug", "Sep",
                "Oct", "Nov", "Dec")

## our data
some_months <- c("Mar", "Dec", "Jan",  "Apr", "Jul")
```

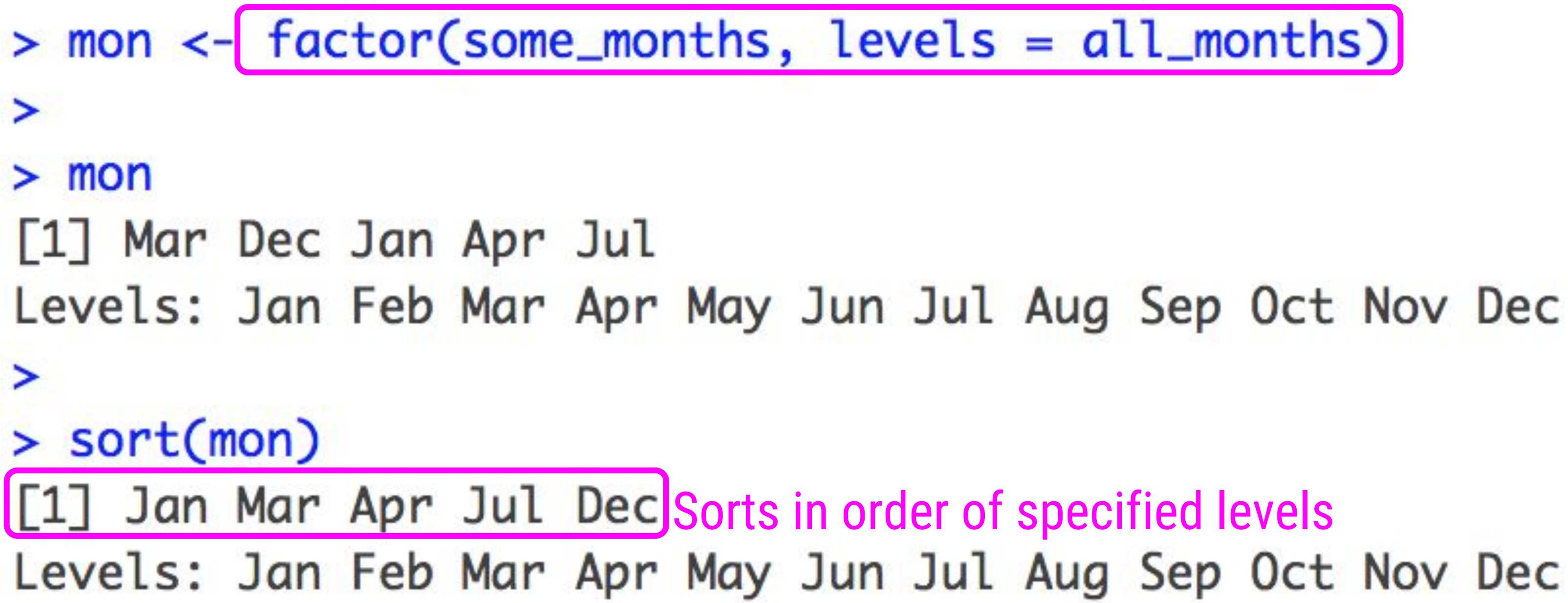


```
> sort(some_months)
[1] "Apr" "Dec" "Jan" "Jul" "Mar"
```

Sorts alphabetically

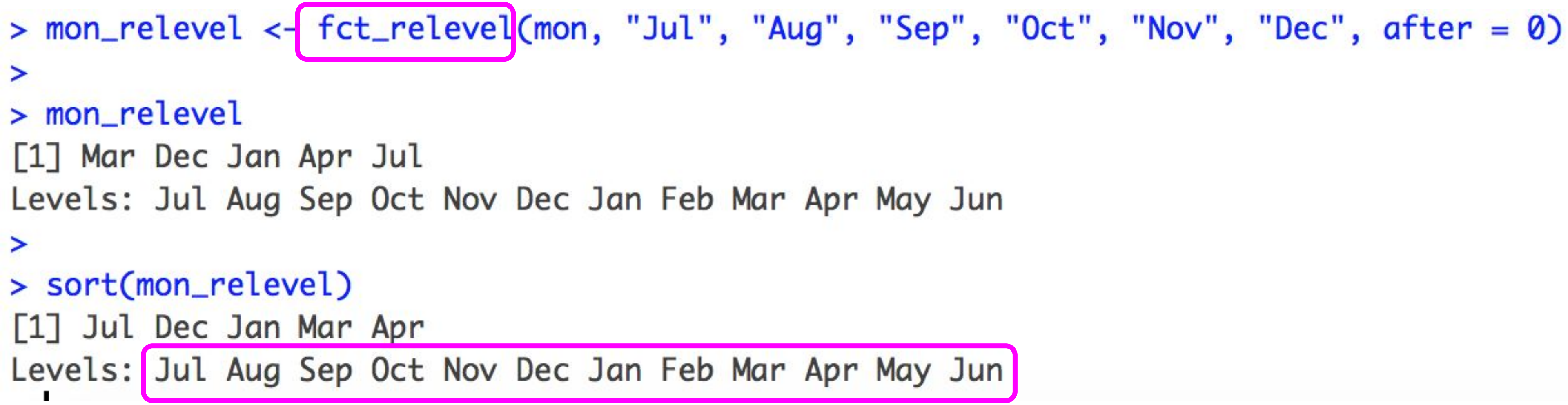
```
## all 12 months
all_months <- c("Jan", "Feb", "Mar", "Apr", "May", "Jun", "Jul", "Aug", "Sep",
                "Oct", "Nov", "Dec")

## our data
some_months <- c("Mar", "Dec", "Jan",  "Apr", "Jul")
```



```
> mon <- factor(some_months, levels = all_months)
>
> mon
[1] Mar Dec Jan Apr Jul
Levels: Jan Feb Mar Apr May Jun Jul Aug Sep Oct Nov Dec
>
> sort(mon)
[1] Jan Mar Apr Jul Dec
Levels: Jan Feb Mar Apr May Jun Jul Aug Sep Oct Nov Dec
```

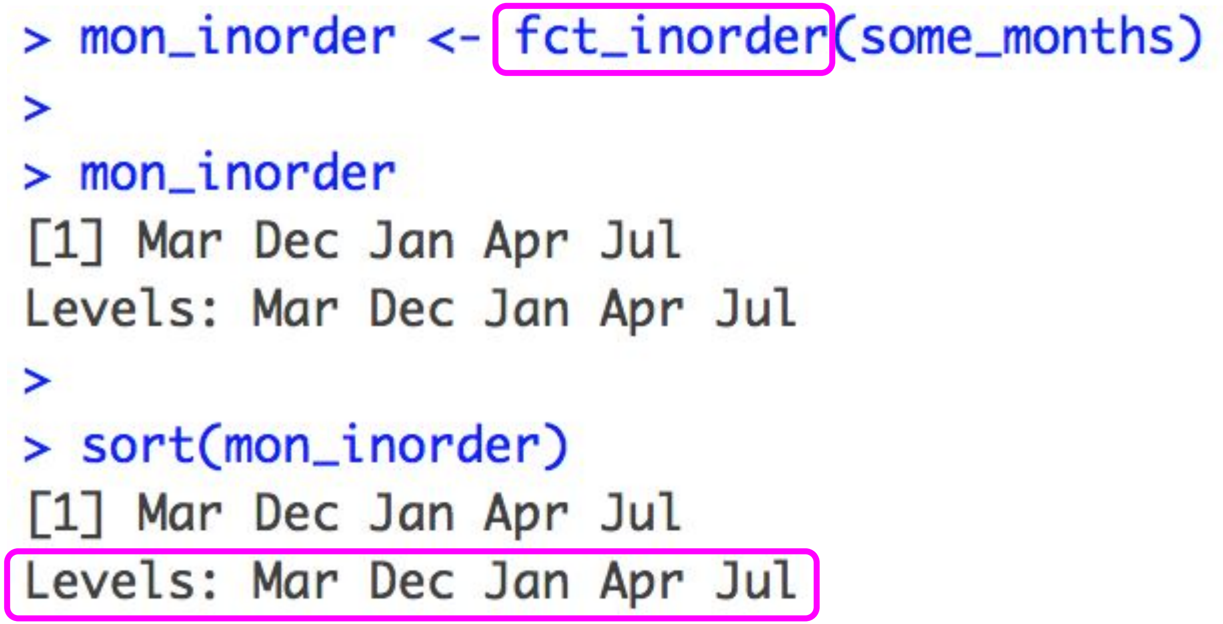
Sorts in order of specified levels



```
> mon_relevel <- fct_relevel(mon, "Jul", "Aug", "Sep", "Oct", "Nov", "Dec", after = 0)
>
> mon_relevel
[1] Mar Dec Jan Apr Jul
Levels: Jul Aug Sep Oct Nov Dec Jan Feb Mar Apr May Jun
>
> sort(mon_relevel)
[1] Jul Dec Jan Mar Apr
Levels: Jul Aug Sep Oct Nov Dec Jan Feb Mar Apr May Jun
```

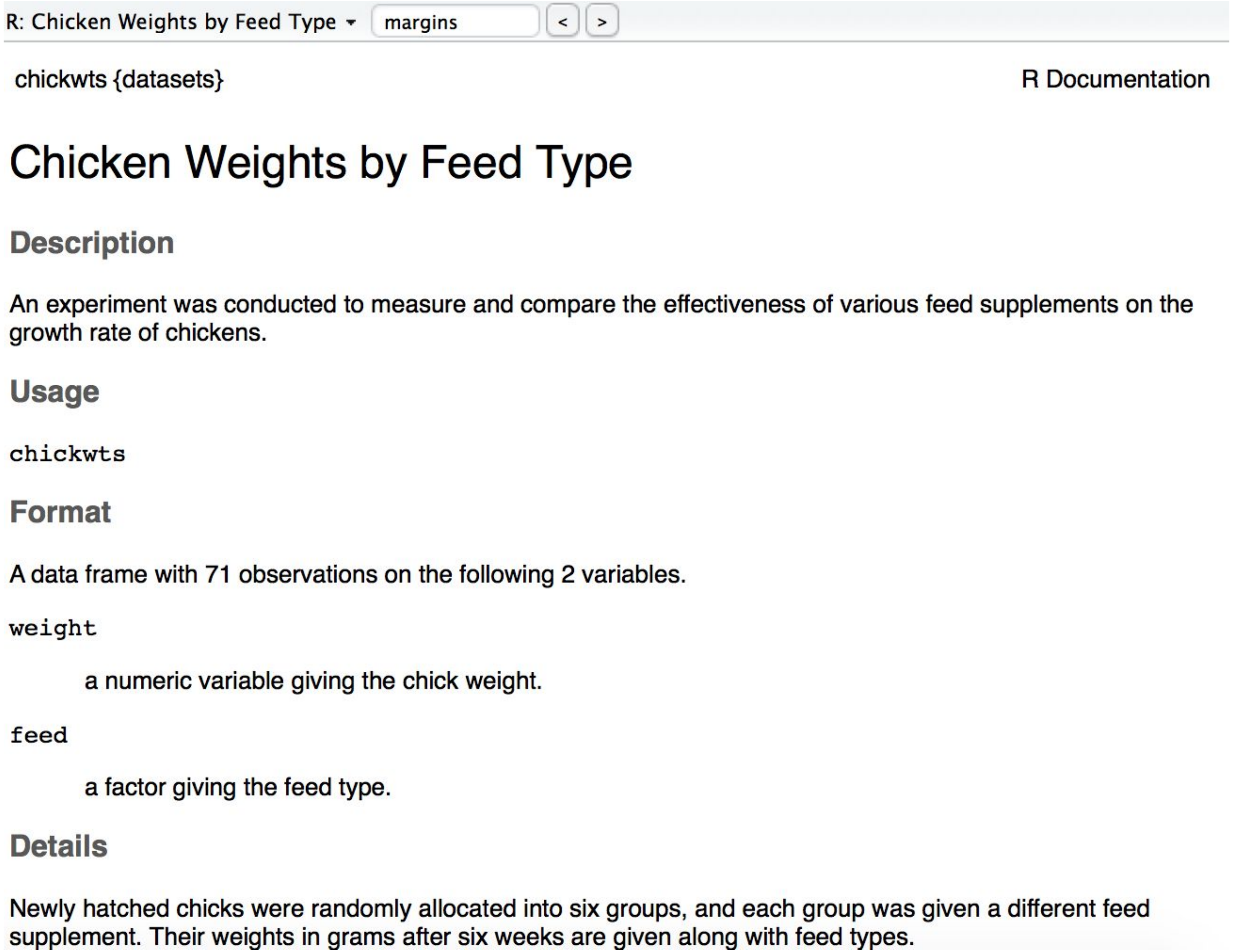
Sorts in order of re-ordered levels

```
some_months <- c("Mar", "Dec", "Jan",  "Apr", "Jul")
```



```
> mon_inorder <- fct_inorder(some_months)
>
> mon_inorder
[1] Mar Dec Jan Apr Jul
Levels: Mar Dec Jan Apr Jul
>
> sort(mon_inorder)
[1] Mar Dec Jan Apr Jul
Levels: Mar Dec Jan Apr Jul
```

Levels match order of appearance in data



chickwts {datasets}	R Documentation

Chicken Weights by Feed Type

Description

An experiment was conducted to measure and compare the effectiveness of various feed supplements on the growth rate of chickens.

Usage

```
chickwts
```

Format

A data frame with 71 observations on the following 2 variables.

`weight`

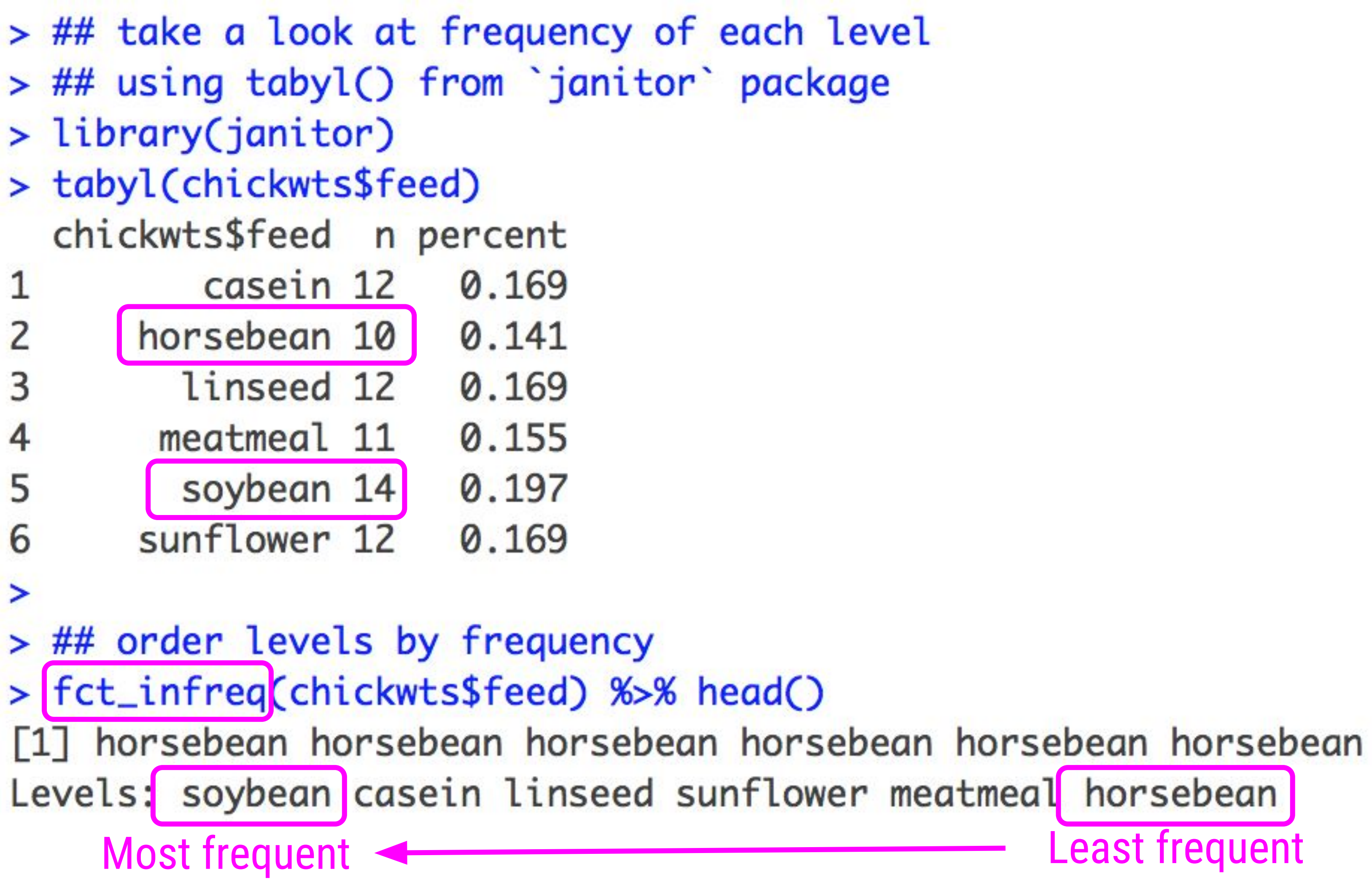
a numeric variable giving the chick weight.

`feed`

a factor giving the feed type.

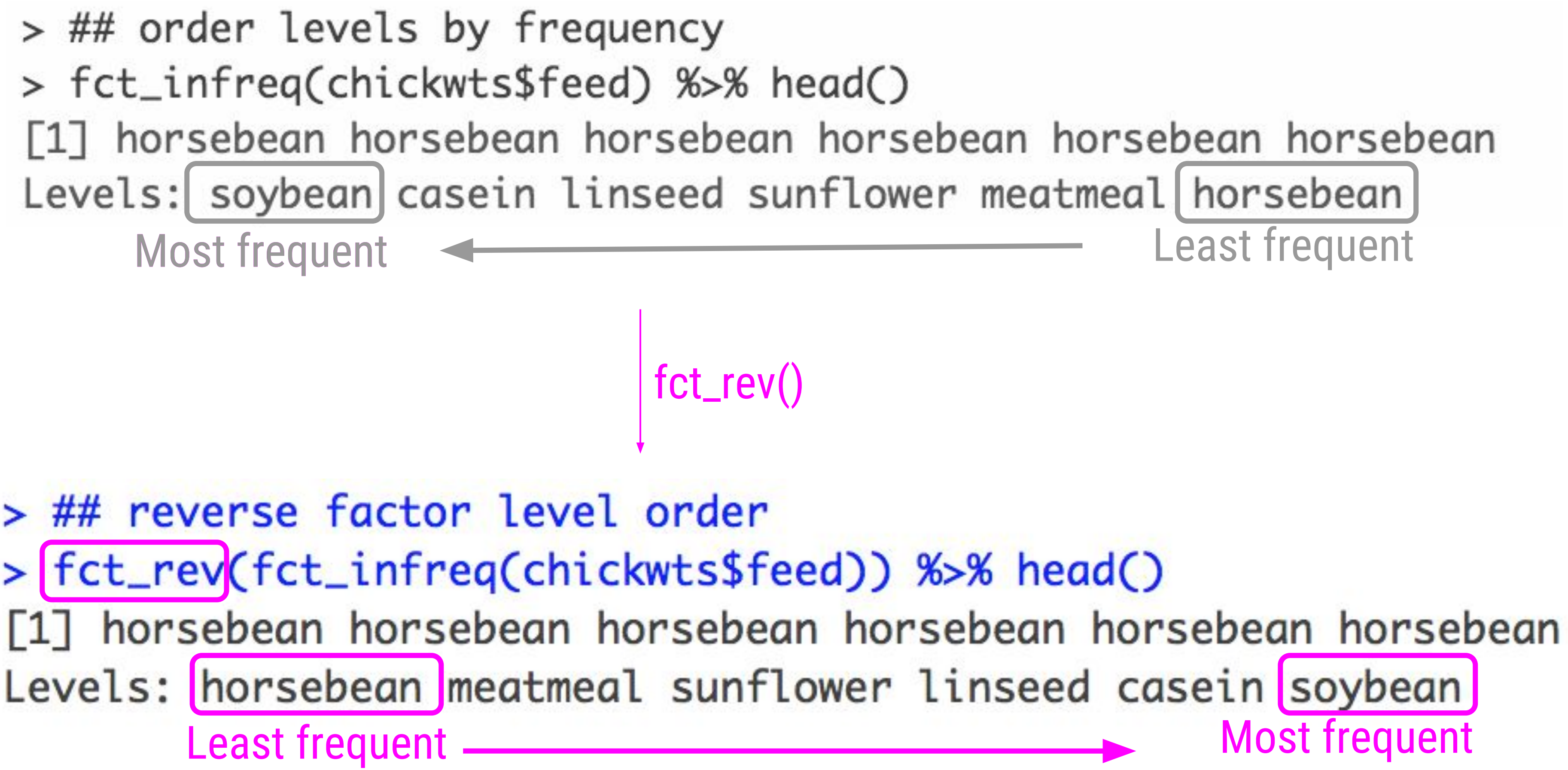
Details

Newly hatched chicks were randomly allocated into six groups, and each group was given a different feed supplement. Their weights in grams after six weeks are given along with feed types.



```
> ## take a look at frequency of each level
> ## using tabyl() from `janitor` package
> library(janitor)
> tabyl(chickwts$feed)
  chickwts$feed  n percent
1        casein 12   0.169
2     horsebean 10   0.141
3       linseed 12   0.169
4      meatmeal 11   0.155
5       soybean 14   0.197
6     sunflower 12   0.169
>
> ## order levels by frequency
> fct_infreq(chickwts$feed) %>% head()
[1] horsebean horsebean horsebean horsebean horsebean horsebean
Levels: soybean casein linseed sunflower meatmeal horsebean
```

Most frequent ← Least frequent



```
> ## order levels by frequency
> fct_infreq(chickwts$feed) %>% head()
[1] horsebean horsebean horsebean horsebean horsebean horsebean
Levels: soybean casein linseed sunflower meatmeal horsebean
```

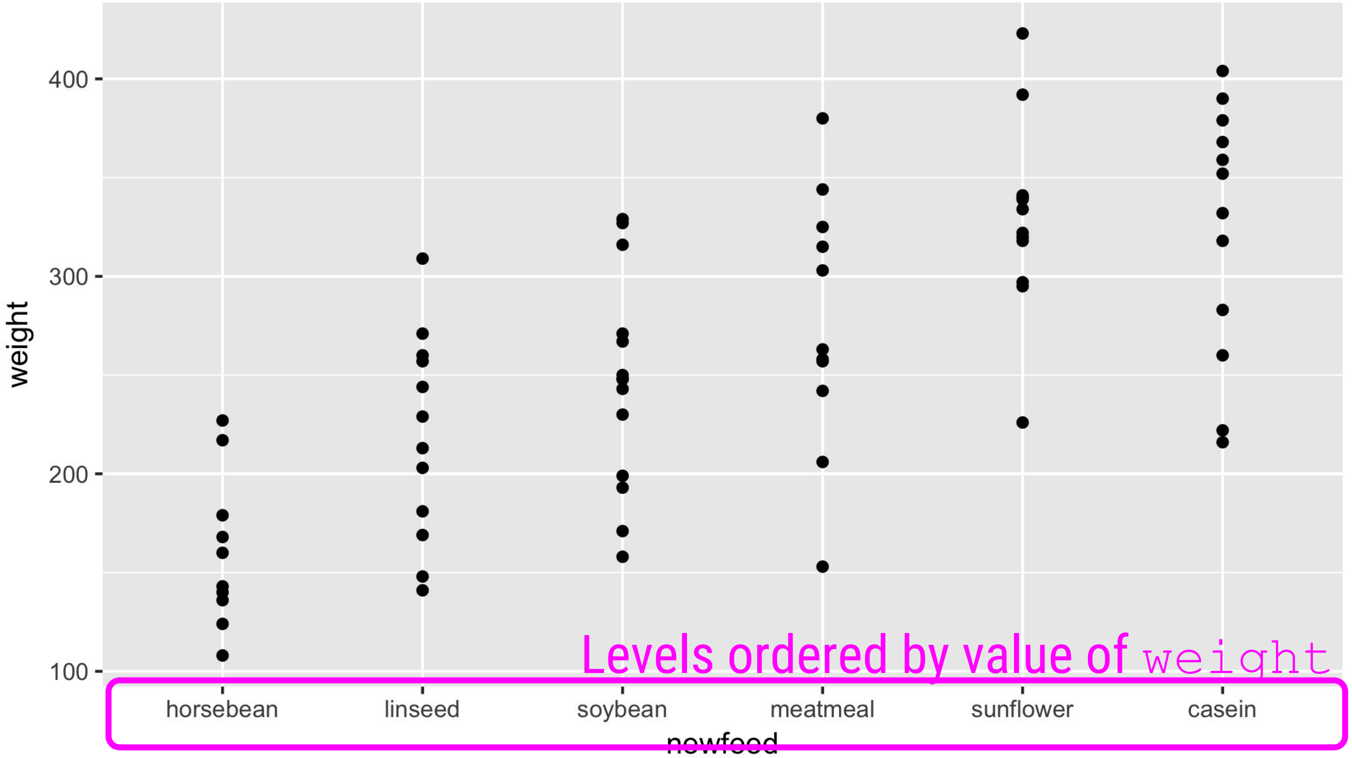
Most frequent ← Least frequent

fct_rev()

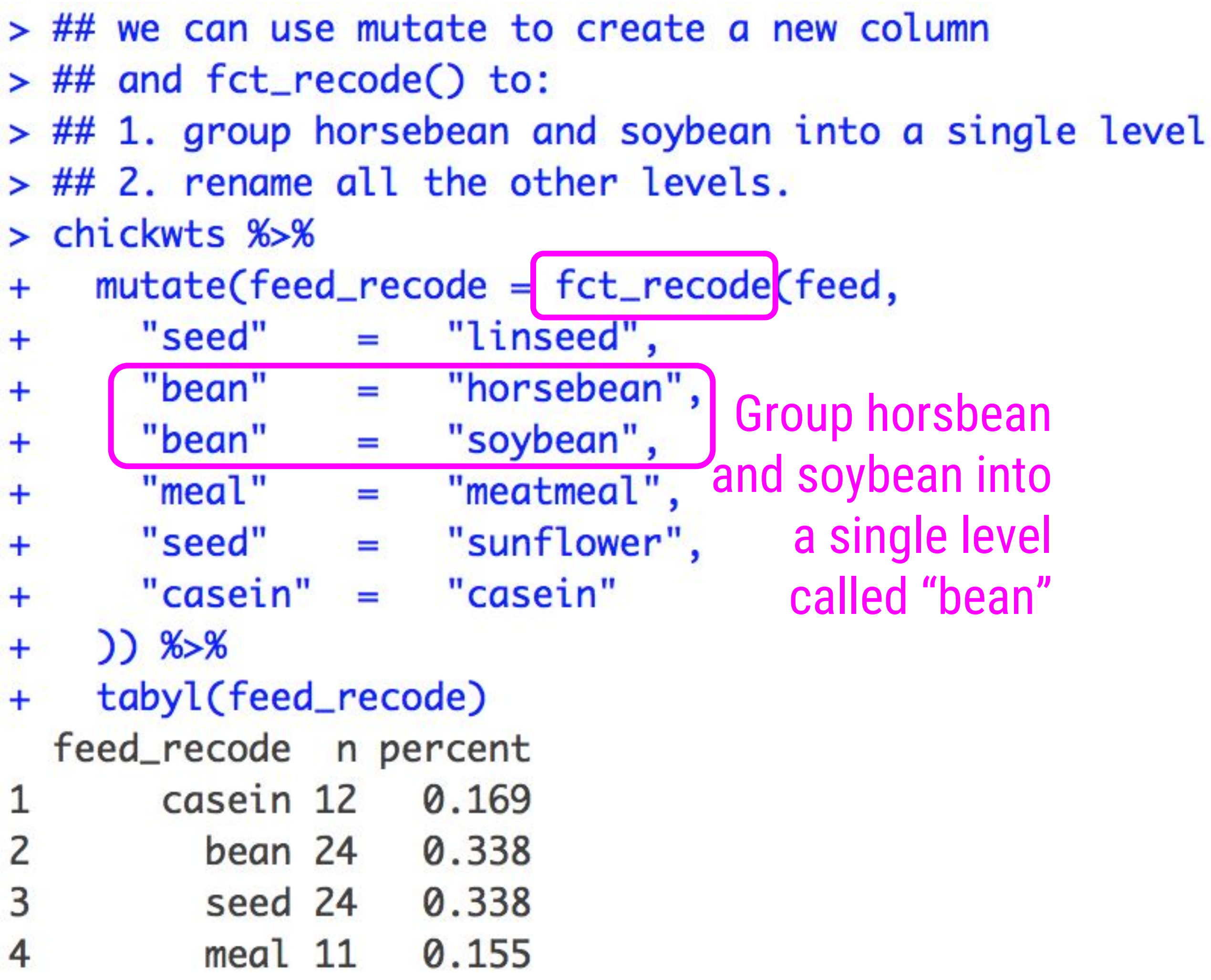
```
> ## reverse factor level order
> fct_rev(fct_infreq(chickwts$feed)) %>% head()
[1] horsebean horsebean horsebean horsebean horsebean horsebean
Levels: horsebean meatmeal sunflower linseed casein soybean
```

Least frequent → Most frequent

```
## order levels by a second numeric variable
chickwts %>%
  mutate(newfeed = fct_reorder(feed, weight)) %>%
  ggplot(., aes(newfeed,weight)) +
  geom_point()
```

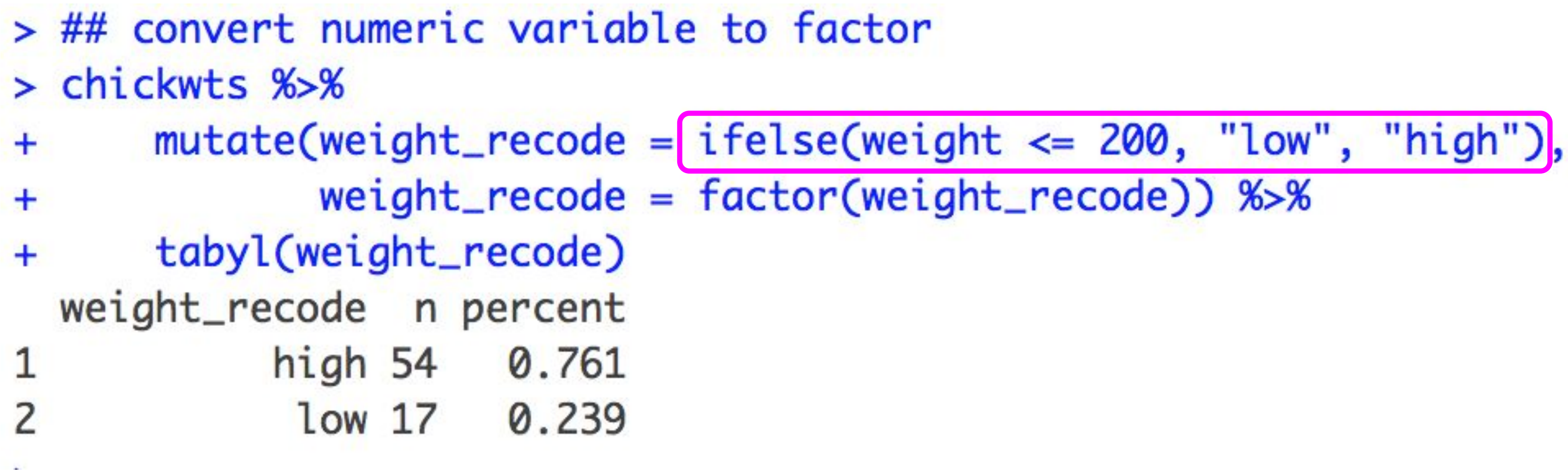


Levels ordered by value of `weight`



```
> ## we can use mutate to create a new column
> ## and fct_recode() to:
> ## 1. group horsebean and soybean into a single level
> ## 2. rename all the other levels.
> chickwts %>%
+   mutate(feed_recode = fct_recode(feed,
+     "seed"      =    "linseed",
+     "bean"      =    "horsebean",
+     "bean"      =    "soybean",
+     "meal"      =    "meatmeal",
+     "seed"      =    "sunflower",
+     "casein"    =    "casein"
+   )) %>%
+   tabyl(feed_recode)
  feed_recode  n percent
1      casein 12   0.169
2        bean 24   0.338
3        seed 24   0.338
4        meal 11   0.155
```

Group horsbean and soybean into a single level called “bean”



```
> ## convert numeric variable to factor
> chickwts %>%
+     mutate(weight_recode = ifelse(weight <= 200, "low", "high"),
+            weight_recode = factor(weight_recode)) %>%
+     tabyl(weight_recode)
  weight_recode  n percent
1          high 54   0.761
2           low 17   0.239
```

https://forcats.tidyverse.org/