Factors

Getting and Cleaning Data

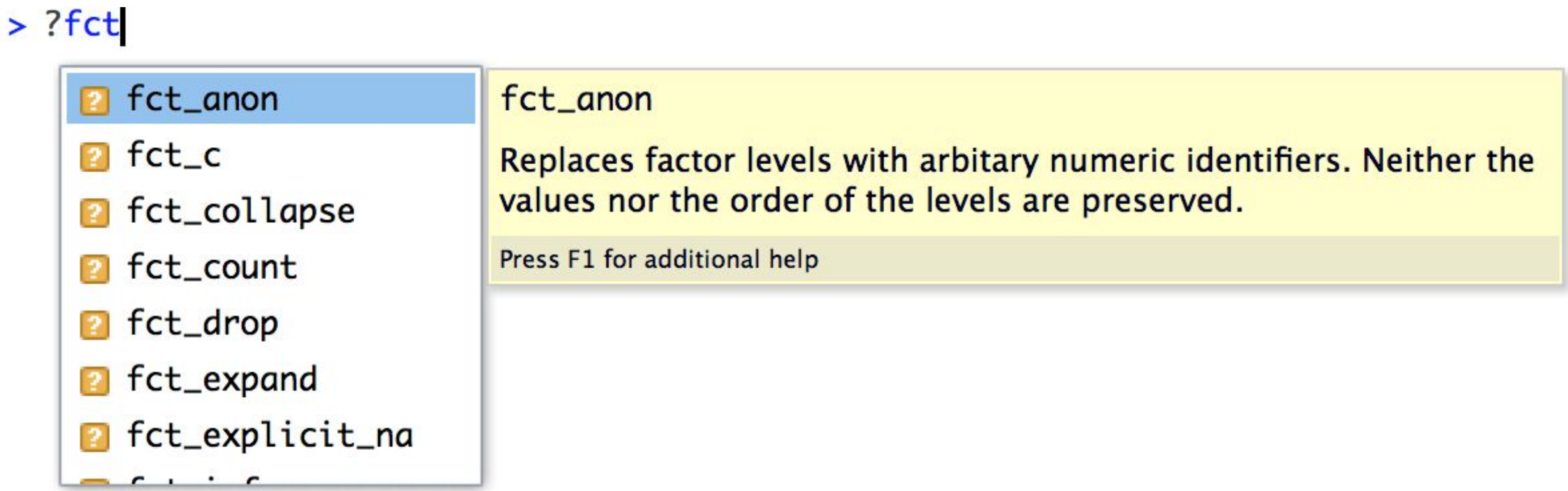
Factors are for **categorical variables**

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

Example: **<u>months</u>**

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

https://forcats.tidyverse.org/

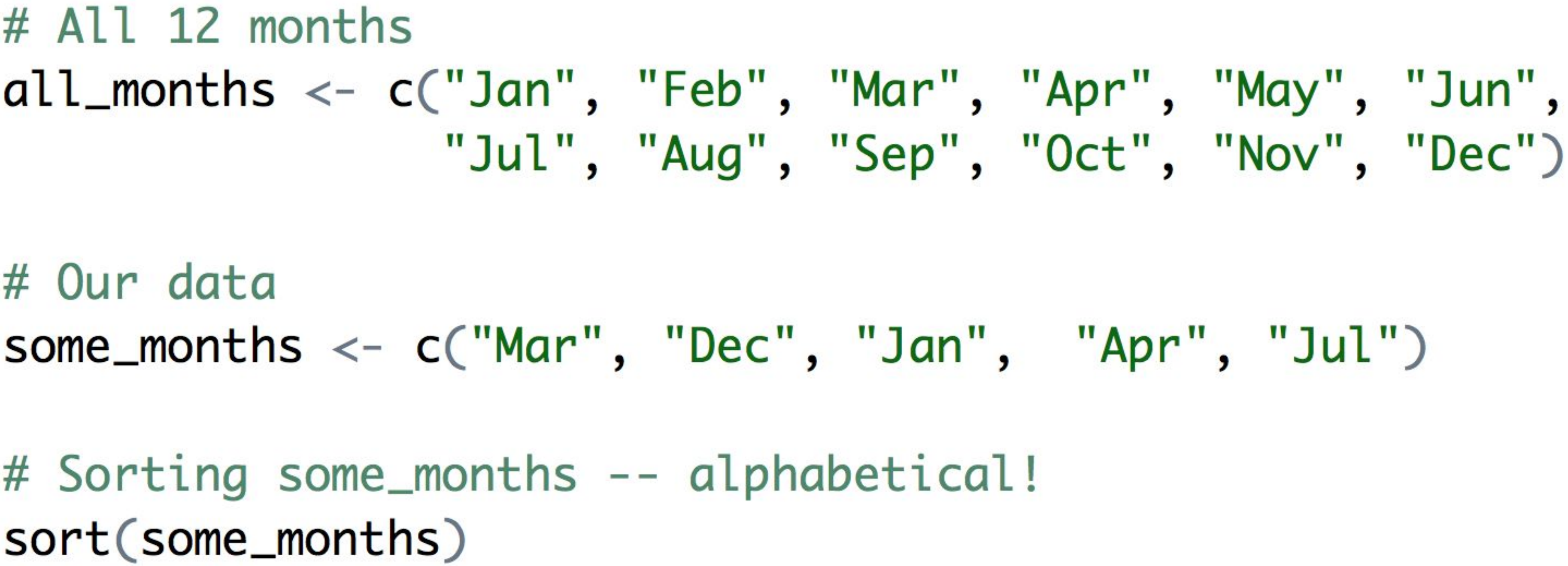


```
> ?fct
```

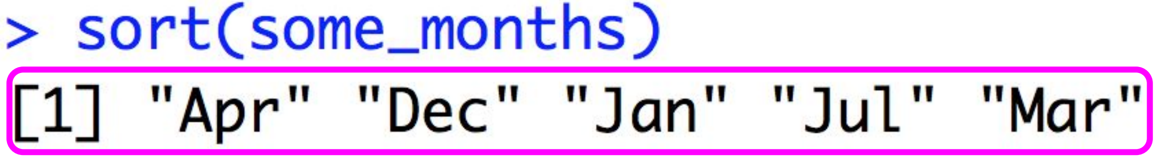
fct_anon

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

Press F1 for additional help

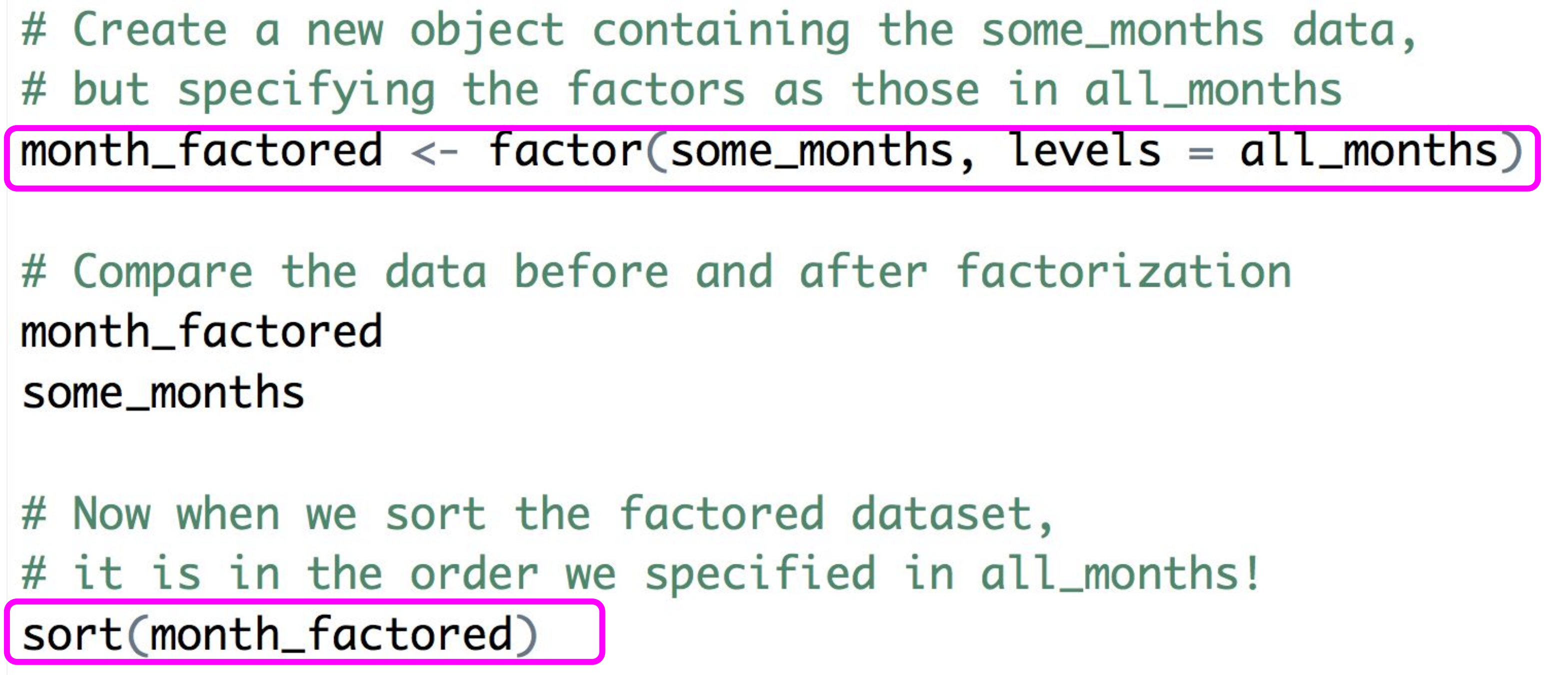


```r
# All 12 months
all_months <- c("Jan", "Feb", "Mar", "Apr", "May", "Jun",
                "Jul", "Aug", "Sep", "Oct", "Nov", "Dec")

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

# Sorting some_months -- alphabetical!
sort(some_months)
```

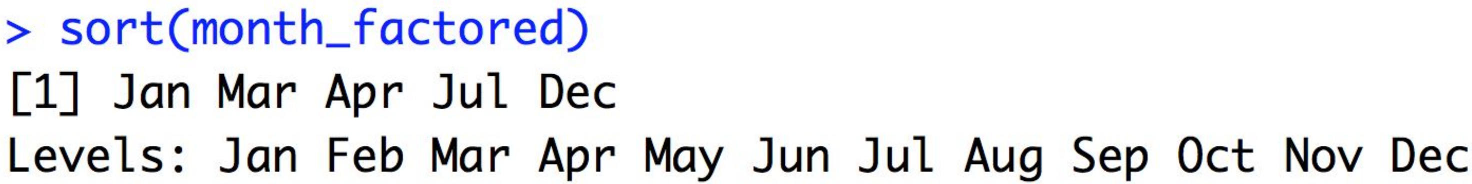


```
> sort(some_months)
[1] "Apr" "Dec" "Jan" "Jul" "Mar"
```

Sorts alphabetically

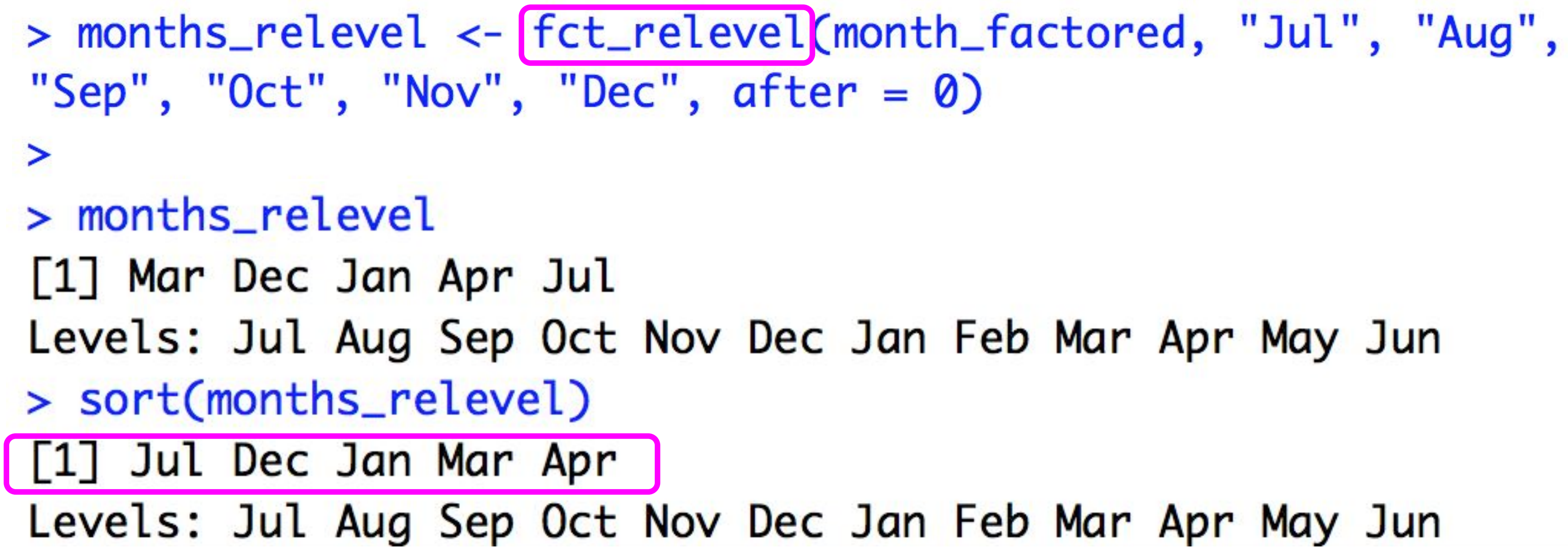


```
# Create a new object containing the some_months data,
# but specifying the factors as those in all_months
month_factored <- factor(some_months, levels = all_months)

# Compare the data before and after factorization
month_factored
some_months

# Now when we sort the factored dataset,
# it is in the order we specified in all_months!
sort(month_factored)
```



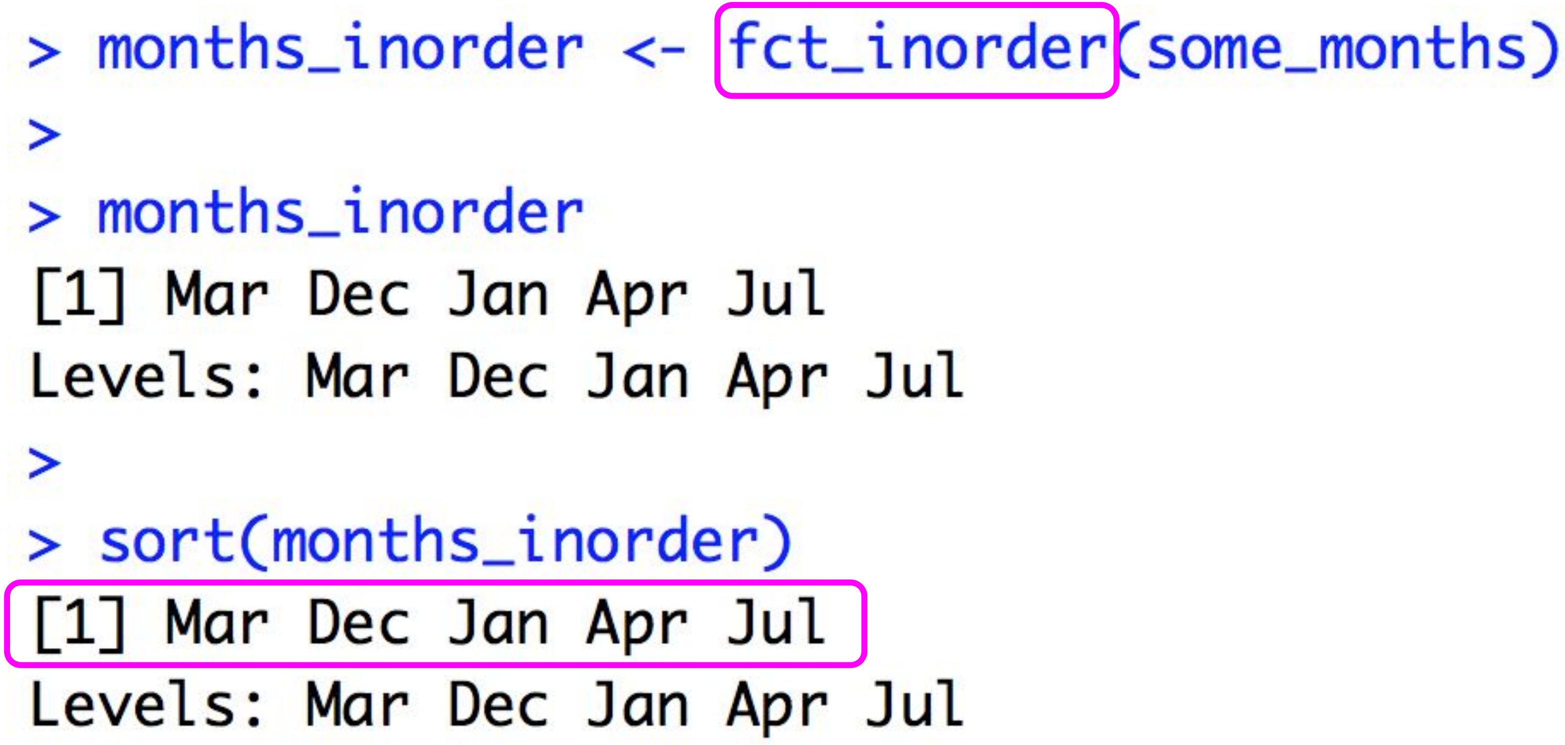
```
> sort(month_factored)
[1] Jan Mar Apr Jul Dec
Levels: Jan Feb Mar Apr May Jun Jul Aug Sep Oct Nov Dec
```

Sorts in order of specified levels!



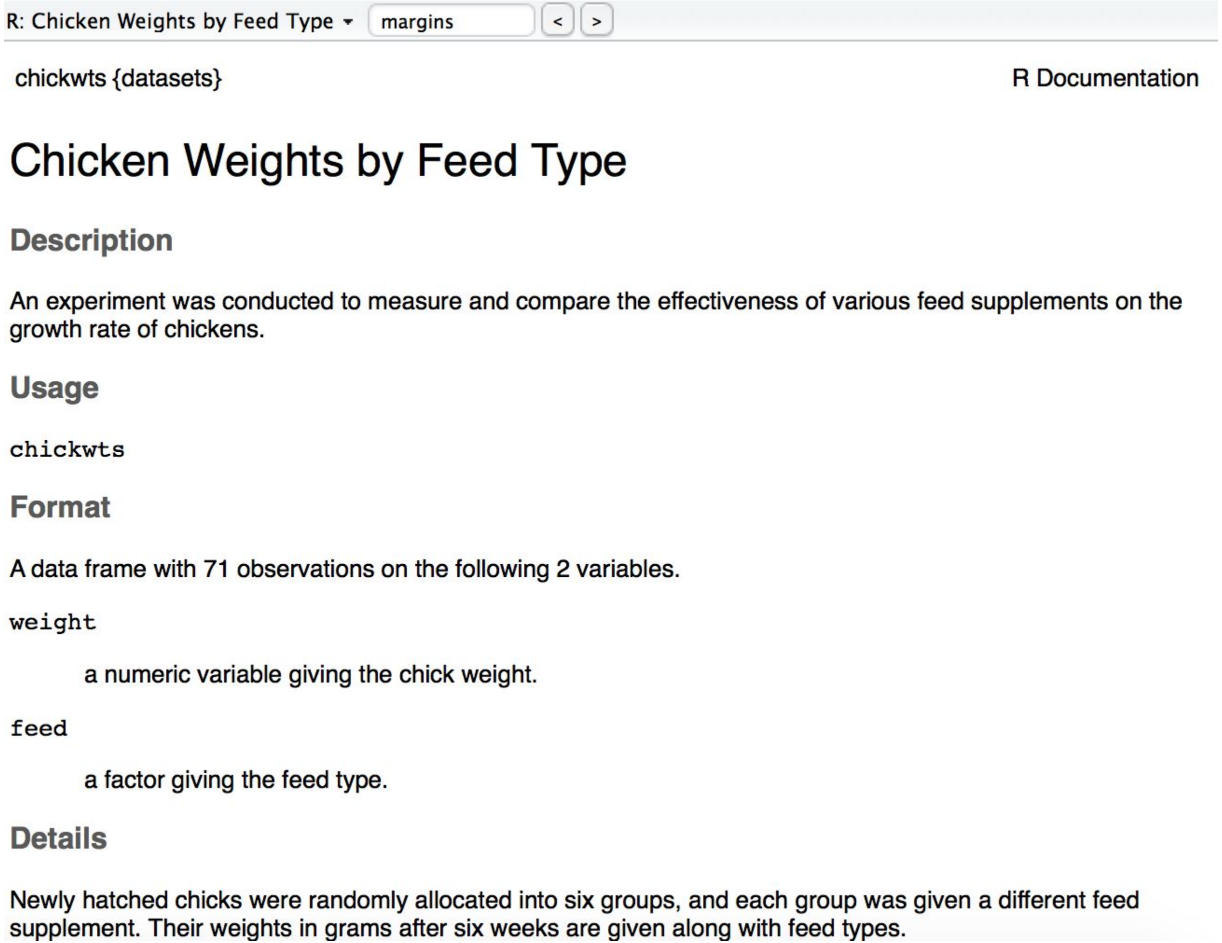
```
> months_relevel <- fct_relevel(month_factored, "Jul", "Aug",
"Sep", "Oct", "Nov", "Dec", after = 0)
>
> months_relevel
[1] Mar Dec Jan Apr Jul
Levels: Jul Aug Sep Oct Nov Dec Jan Feb Mar Apr May Jun
> sort(months_relevel)
[1] Jul Dec Jan Mar Apr
Levels: Jul Aug Sep Oct Nov Dec Jan Feb Mar Apr May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

Sorts in order of re-ordered levels



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

Levels match order of appearance in the some_months dataset



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`

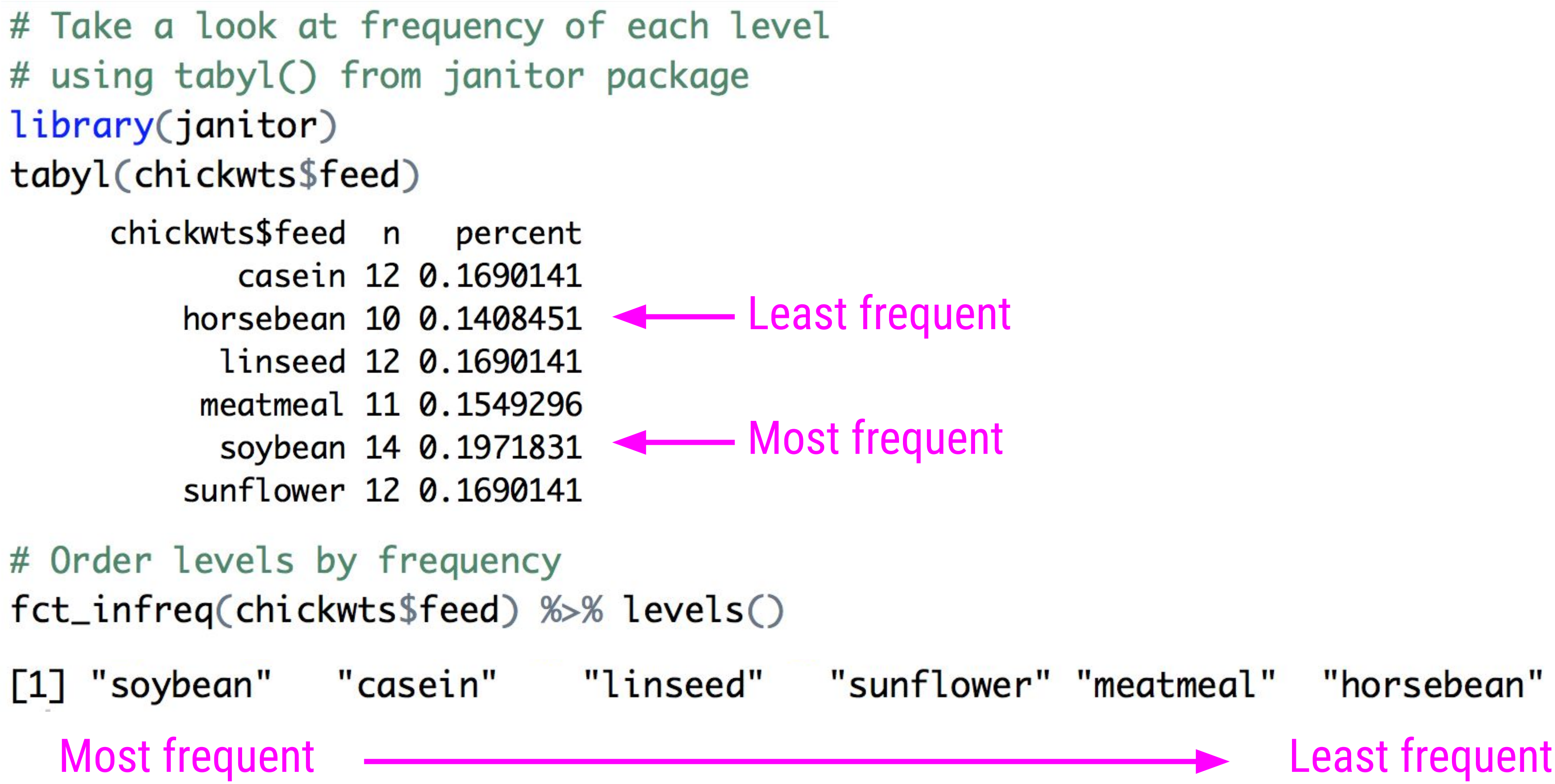
a numeric variable giving the chick weight.

`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.



```r
# Take a look at frequency of each level
# using tabyl() from janitor package
library(janitor)
tabyl(chickwts$feed)
```

```
 chickwts$feed  n   percent
        casein 12 0.1690141
     horsebean 10 0.1408451
       linseed 12 0.1690141
      meatmeal 11 0.1549296
       soybean 14 0.1971831
     sunflower 12 0.1690141
```

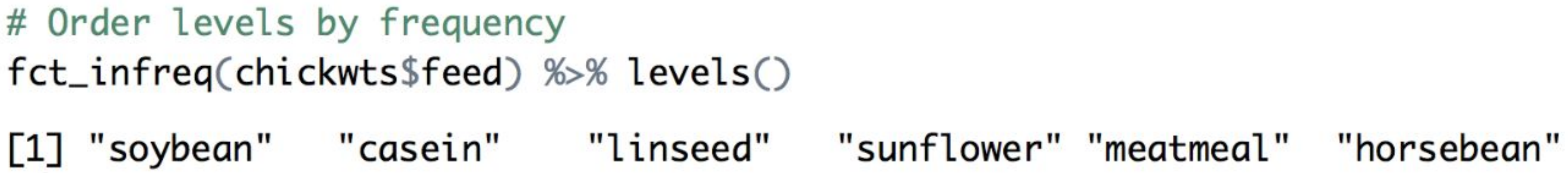
Least frequent

Most frequent

```r
# Order levels by frequency
fct_infreq(chickwts$feed) %>% levels()
```

```
[1] "soybean"   "casein"    "linseed"   "sunflower" "meatmeal"  "horsebean"
```

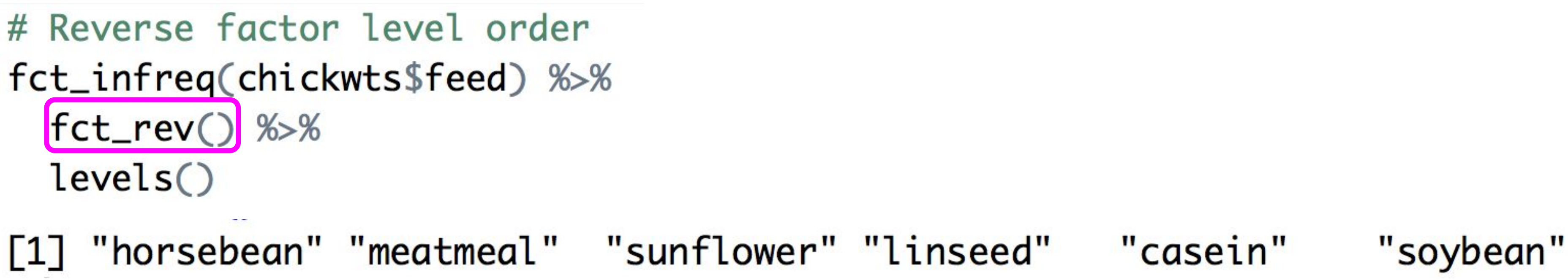
Most frequent → Least frequent



```
# Order levels by frequency
fct_infreq(chickwts$feed) %>% levels()

[1] "soybean"   "casein"    "linseed"   "sunflower" "meatmeal"  "horsebean"
```

Most frequent → Least frequent

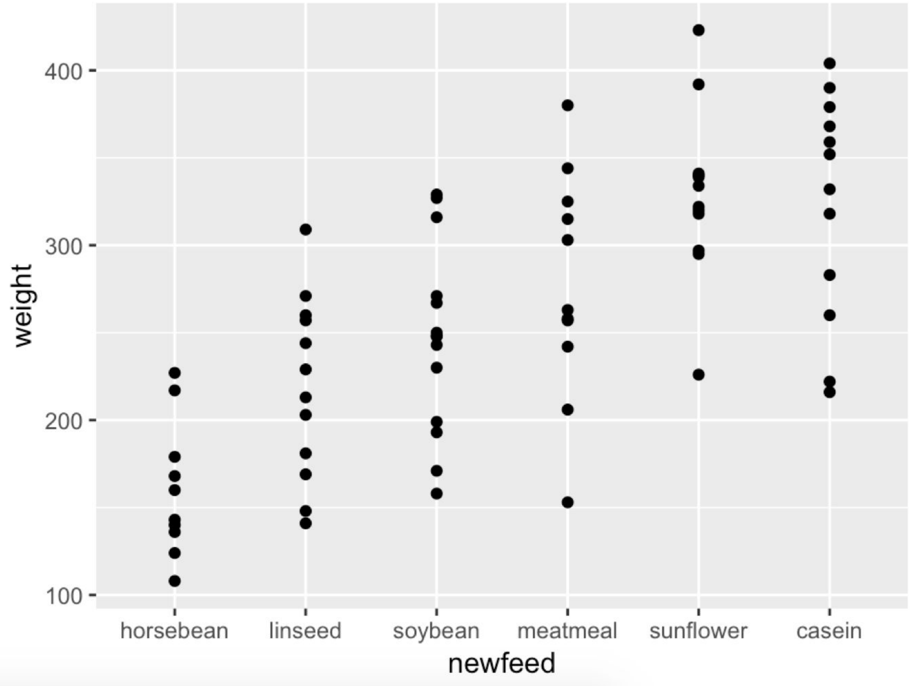
fct_rev()



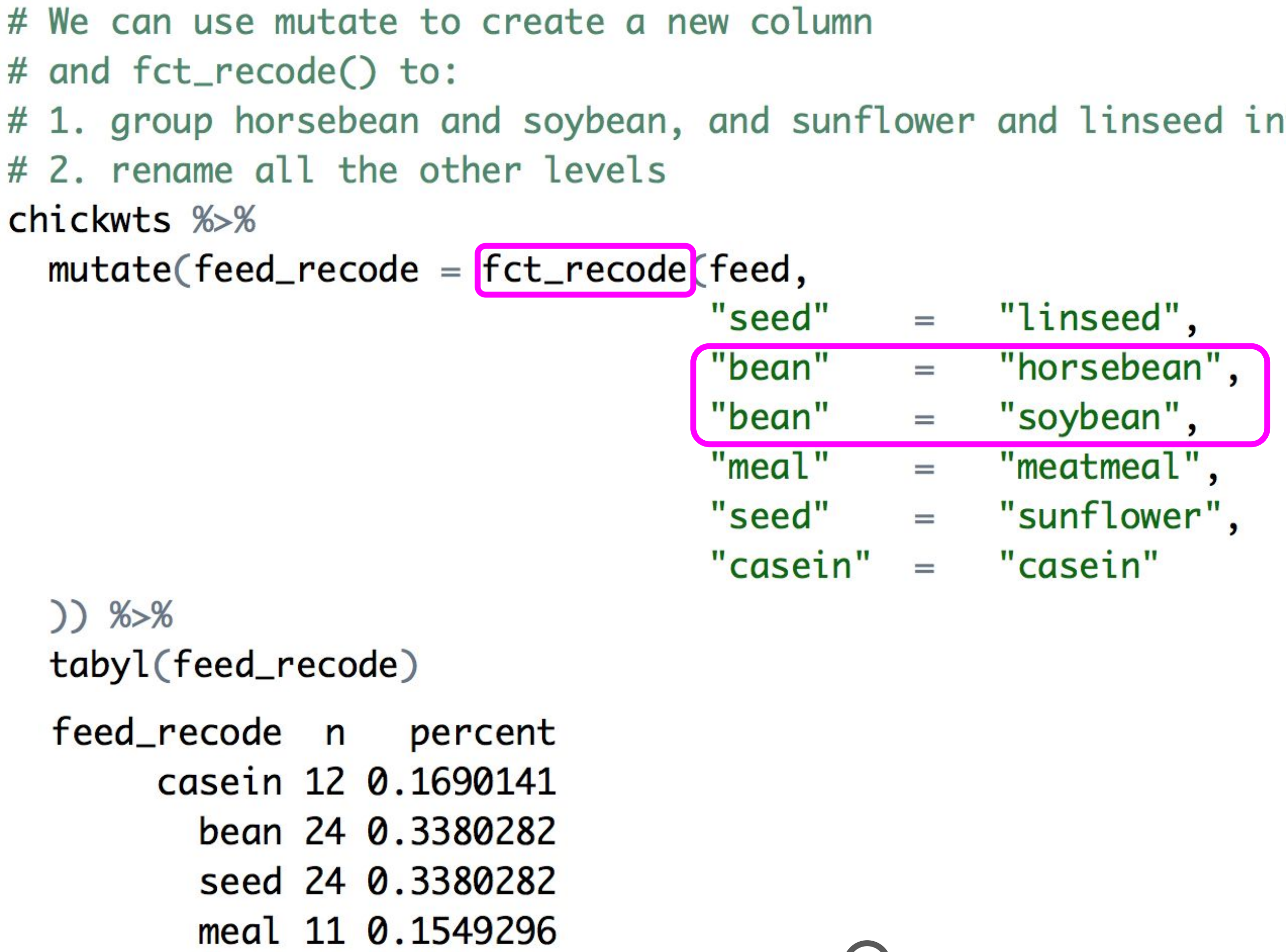
```
# Reverse factor level order
fct_infreq(chickwts$feed) %>%
  fct_rev() %>%
  levels()

[1] "horsebean" "meatmeal"  "sunflower" "linseed"   "casein"    "soybean"
```

Least frequent ← Most frequent

```r
# Order levels by a second numeric variable
chickwts %>%
  mutate(newfeed = fct_reorder(feed, weight)) %>% # Reorder feed types by chicken weight
  ggplot(aes(x = newfeed, y = weight)) + # Plot the feed type on the X and chicken weights on the Y axes
  geom_point() # Plot this data as points
```



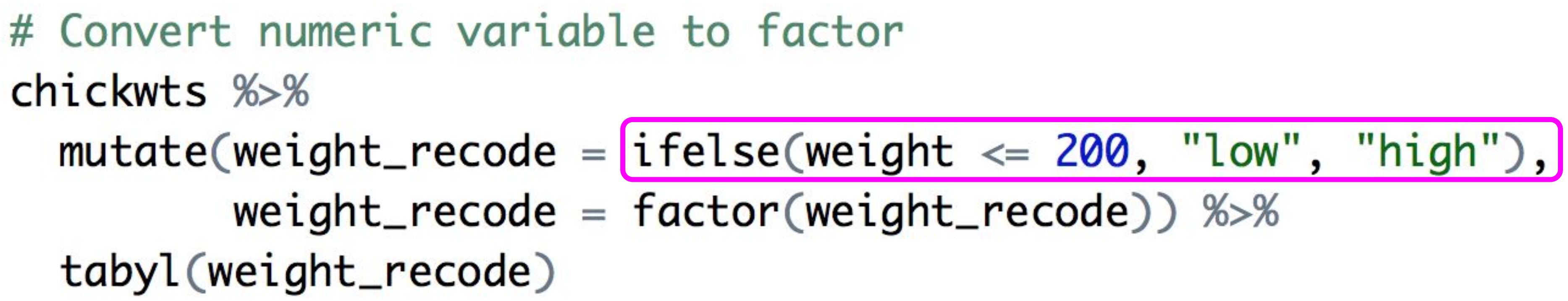
Feed levels ordered by value of `weight`



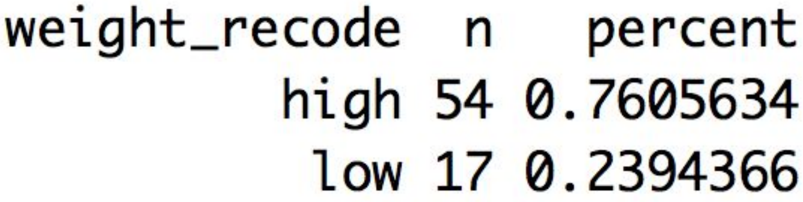
```
# We can use mutate to create a new column
# and fct_recode() to:
# 1. group horsebean and soybean, and sunflower and linseed into single levels
# 2. rename all the other levels
chickwts %>%
  mutate(feed_recode = fct_recode(feed,
                                   "seed"    =    "linseed",
                                   "bean"    =    "horsebean",
                                   "bean"    =    "soybean",
                                   "meal"    =    "meatmeal",
                                   "seed"    =    "sunflower",
                                   "casein"  =    "casein"
  )) %>%
  tabyl(feed_recode)
```

```
  feed_recode  n   percent
       casein 12 0.1690141
         bean 24 0.3380282
         seed 24 0.3380282
         meal 11 0.1549296
```

Group horsebean 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
         high 54 0.7605634
          low 17 0.2394366
```

Summarizing: Factors

Getting and Cleaning Data