transform your data with tidyverse

Seun Odeyemi

October 13, 2018

Contents

```r
knitr::opts_knit$set(root.dir = "~/tidyverse")
```

```r
devtools::session_info()
```

```
## Session info -------------------------------------------------------------
```

```
##  setting  value
##  version  R version 3.5.1 (2018-07-02)
##  system   x86_64, mingw32
##  ui       RTerm
##  language (EN)
##  collate  English_United States.1252
##  tz       America/Chicago
##  date     2018-10-27
```

```
## Packages -----------------------------------------------------------------
```

```
##  package    * version date       source
##  backports    1.1.2   2017-12-13 CRAN (R 3.5.0)
##  base       * 3.5.1   2018-07-02 local
##  compiler     3.5.1   2018-07-02 local
##  crayon       1.3.4   2017-09-16 CRAN (R 3.5.1)
##  datasets   * 3.5.1   2018-07-02 local
##  devtools     1.13.6  2018-06-27 CRAN (R 3.5.1)
##  digest       0.6.17  2018-09-12 CRAN (R 3.5.1)
##  evaluate     0.11    2018-07-17 CRAN (R 3.5.1)
##  graphics   * 3.5.1   2018-07-02 local
##  grDevices  * 3.5.1   2018-07-02 local
##  htmltools    0.3.6   2017-04-28 CRAN (R 3.5.1)
##  knitr        1.20    2018-02-20 CRAN (R 3.5.1)
##  magrittr     1.5     2014-11-22 CRAN (R 3.5.1)
##  memoise      1.1.0   2017-04-21 CRAN (R 3.5.1)
##  methods    * 3.5.1   2018-07-02 local
##  pillar       1.3.0   2018-07-14 CRAN (R 3.5.1)
##  Rcpp         0.12.19 2018-10-01 CRAN (R 3.5.1)
##  rlang        0.2.2   2018-08-16 CRAN (R 3.5.1)
##  rmarkdown    1.10    2018-06-11 CRAN (R 3.5.1)
##  rprojroot    1.3-2   2018-01-03 CRAN (R 3.5.1)
##  rstudioapi   0.8     2018-10-02 CRAN (R 3.5.1)
##  stats      * 3.5.1   2018-07-02 local
##  stringi      1.2.4   2018-07-20 CRAN (R 3.5.1)
##  stringr      1.3.1   2018-05-10 CRAN (R 3.5.1)
##  tibble       1.4.2   2018-01-22 CRAN (R 3.5.1)
##  tools        3.5.1   2018-07-02 local
##  utils      * 3.5.1   2018-07-02 local
##  withr        2.1.2   2018-03-15 CRAN (R 3.5.1)
##  yaml         2.2.0   2018-07-25 CRAN (R 3.5.1)
```

Load Packages

```r
library(tidyverse)
library(skimr)
library(here)
```

Import data

```
baker_result <- read_csv("datasets/baker_results.csv")
```

```
## Parsed with column specification:
## cols(
##   .default = col_integer(),
##   baker_full = col_character(),
##   baker = col_character(),
##   occupation = col_character(),
##   hometown = col_character(),
##   baker_last = col_character(),
##   baker_first = col_character(),
##   technical_median = col_double(),
##   first_date_appeared = col_date(format = ""),
##   last_date_appeared = col_date(format = ""),
##   first_date_us = col_date(format = ""),
##   last_date_us = col_date(format = ""),
##   percent_episodes_appeared = col_double(),
##   percent_technical_top3 = col_double()
## )
```

```
## See spec(...) for full column specifications.
```

```
dplyr::slice(baker_result, 1:6)
```

```
## # A tibble: 6 x 24
##   series baker_full baker   age occupation hometown baker_last baker_first
##    <int> <chr>      <chr> <int> <chr>      <chr>    <chr>      <chr>
## 1      1 Annetha M~ Anne~    30 Single mo~ Essex    Mills      Annetha
## 2      1 David Cha~ David    31 Entrepren~ Milton ~ Chambers   David
## 3      1 "Edward \~ Edd      24 Debt coll~ Bradford Kimber     Edward
## 4      1 Jasminder~ Jasm~    45 Assistant~ Birming~ Randhawa   Jasminder
## 5      1 Jonathan ~ Jona~    25 Research ~ St Alba~ Shepherd   Jonathan
## 6      1 Lea Harris Lea      51 Retired    Midloth~ Harris     Lea
## # ... with 16 more variables: star_baker <int>, technical_winner <int>,
## #   technical_top3 <int>, technical_bottom <int>, technical_highest <int>,
## #   technical_lowest <int>, technical_median <dbl>, series_winner <int>,
## #   series_runner_up <int>, total_episodes_appeared <int>,
## #   first_date_appeared <date>, last_date_appeared <date>,
## #   first_date_us <date>, last_date_us <date>,
## #   percent_episodes_appeared <dbl>, percent_technical_top3 <dbl>
```

```
# Create skill variable with 3 levels

bakers_skill <- baker_result %>%
  mutate(skill = case_when(star_baker > technical_winner ~ "super_star",
                           star_baker < technical_winner ~ "high_tech",
                           TRUE ~ "well_rounded"))
```

```
# Filter zeroes to examine skill variable

bakers_skill %>%
  filter(star_baker == 0 & technical_winner == 0) %>%
  count(skill)
```

```
## # A tibble: 1 x 2
##   skill            n
##   <chr>        <int>
## 1 well_rounded    41
```

```r
# Edit skill variable to have 4 levels

bakers_skill <- baker_result %>%
  mutate(skill = case_when(
    star_baker > technical_winner ~ "super_star",
    star_baker < technical_winner ~ "high_tech",
    star_baker == 0 & technical_winner == 0 ~ NA_character_,
    star_baker == technical_winner  ~ "well_rounded"
  ))
```

```r
# Add pipe to drop skill = NA

bakers_skill <- bakers_skill %>% drop_na(skill)
```

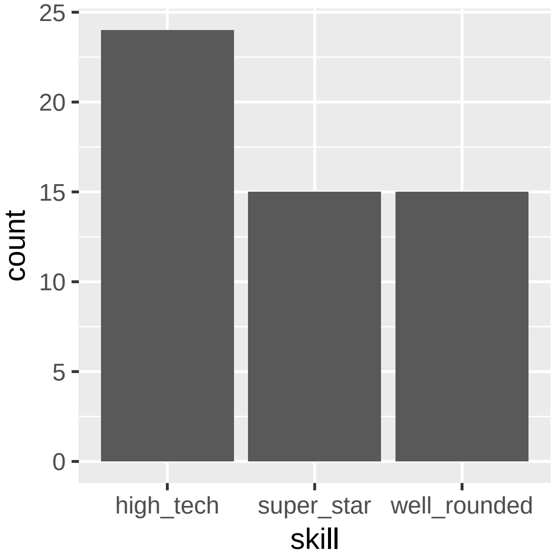
```r
# Count bakers by skill
bakers_skill %>%  count(skill) #count(baker, skill) %>%
```

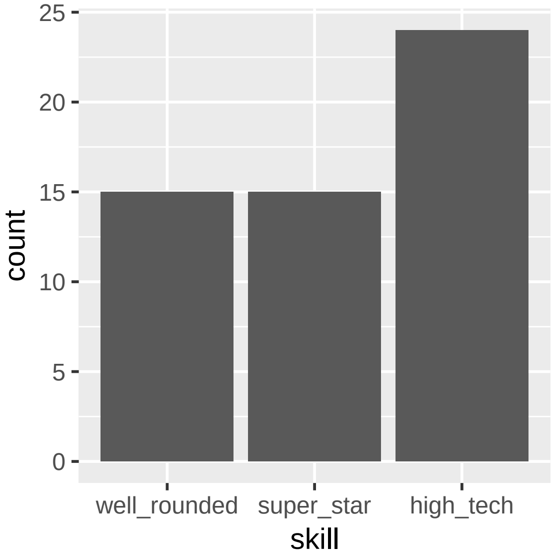
```
## # A tibble: 3 x 2
##   skill            n
##   <chr>        <int>
## 1 high_tech       24
## 2 super_star      15
## 3 well_rounded    15
```

```r
bakers_skill %>%
  count (skill, sort = TRUE) %>%
  mutate(prop = n/sum(n))
```

```
## # A tibble: 3 x 3
##   skill            n  prop
##   <chr>        <int> <dbl>
## 1 high_tech       24 0.444
## 2 super_star      15 0.278
## 3 well_rounded    15 0.278
```

```r
ggplot(bakers_skill, aes(skill)) + geom_bar()

ggplot(bakers_skill, aes(fct_rev(fct_infreq(skill))))  + geom_bar() + xlab("skill")
```

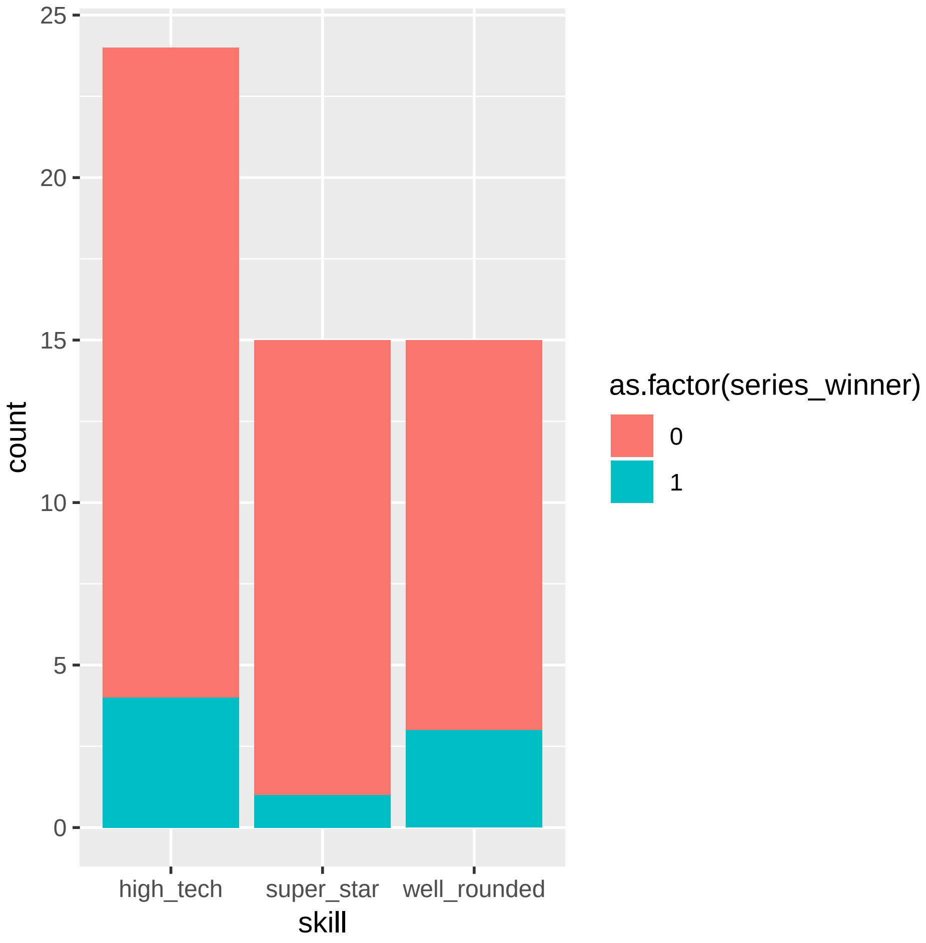


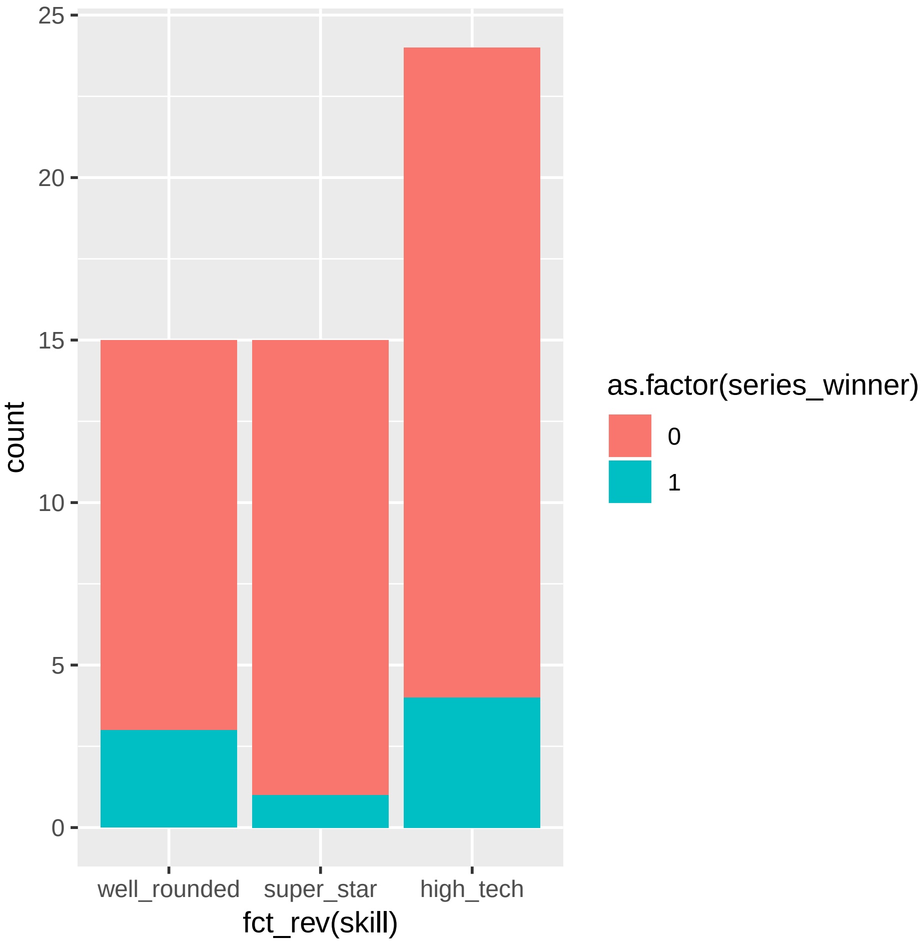


```
# Cast skill as a factor
bakers_skill <- bakers_skill %>%
  mutate(skill = as.factor(skill))

bakers_skill %>% dplyr::pull(skill) %>% levels()
```

```
## [1] "high_tech"    "super_star"   "well_rounded"
```

```
# Plot counts of bakers by skill, fill by winner
ggplot(bakers_skill,  aes(x = skill, fill = as.factor(series_winner)))  + geom_bar()

# Edit to reverse x-axis order
ggplot(bakers_skill, aes(x = fct_rev (skill), fill = as.factor(series_winner))) +
  geom_bar()
```


25
20
15
10
5
0
count
high_tech
super_star
well_rounded
skill
as.factor(series_winner)
0
1



Working with Dates using the Lubridate Package

```
hosts <- tibble::tribble(
  ~host, ~bday, ~premiere,
  "Mary", "24 March 1935", "August 17th, 2010",
  "Paul", "1 March 1966", "August 17th, 2010")
```

```
library(lubridate)
```

```
##
## Attaching package: 'lubridate'

## The following object is masked from 'package:here':
##
##     here

## The following object is masked from 'package:base':
##
##     date
```

```
hosts <- hosts %>%
  mutate(bday = dmy(bday),
         premiere = mdy(premiere))
glimpse(hosts)
```

```
## Observations: 2
## Variables: 3
## $ host     <chr> "Mary", "Paul"
## $ bday     <date> 1935-03-24, 1966-03-01
## $ premiere <date> 2010-08-17, 2010-08-17
```

```
(hosts <- hosts %>%
  mutate(age_int = interval(bday, premiere)))
```

```
## # A tibble: 2 x 4
##   host  bday       premiere   age_int
##   <chr> <date>     <date>     <S4: Interval>
## 1 Mary  1935-03-24 2010-08-17 1935-03-24 UTC--2010-08-17 UTC
## 2 Paul  1966-03-01 2010-08-17 1966-03-01 UTC--2010-08-17 UTC
```

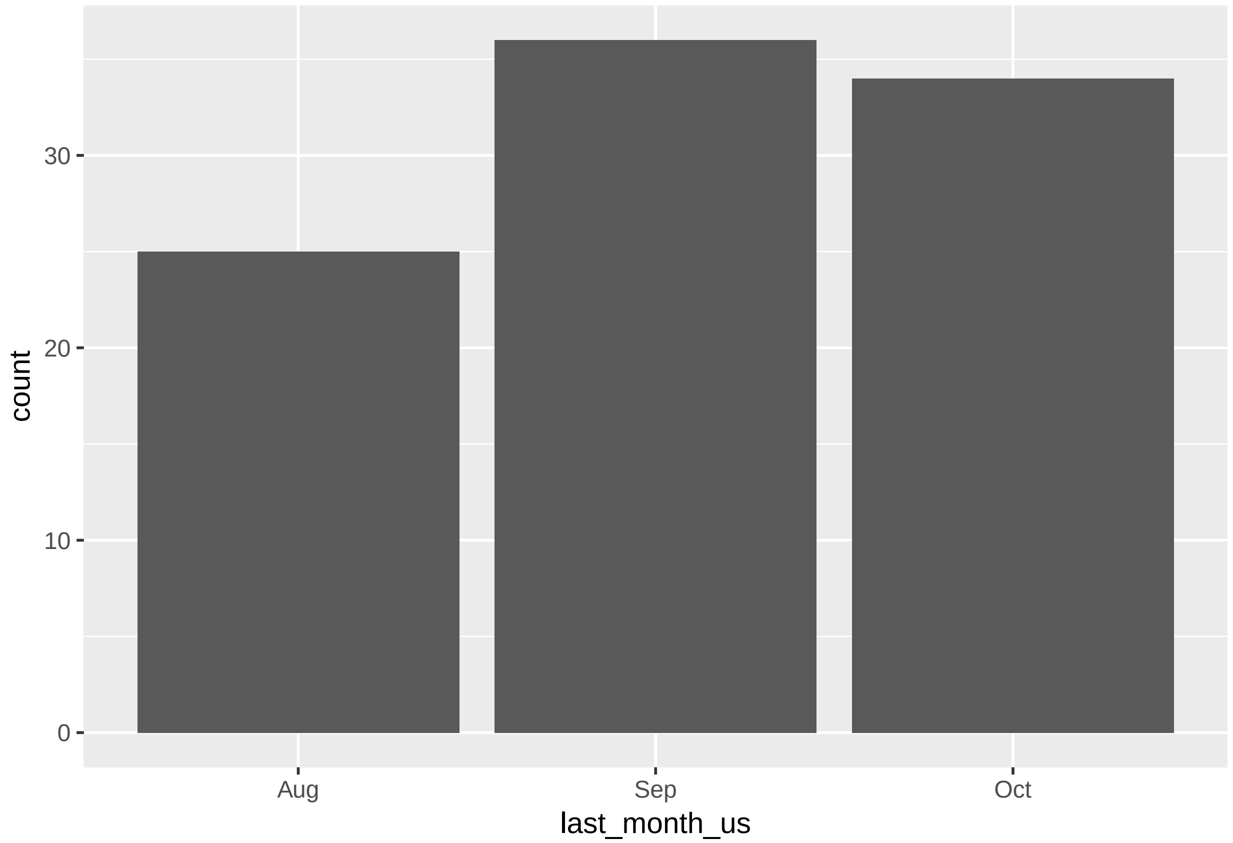
```
(hosts %>%
  mutate(years_decimal = age_int / years(1),
         years_whole = age_int %/% years(1)) )
```

```
## Note: method with signature 'Timespan#Timespan' chosen for function '%/%',
##  target signature 'Interval#Period'.
##  "Interval#ANY", "ANY#Period" would also be valid

## # A tibble: 2 x 6
##   host  bday       premiere   age_int                        years_decimal
##   <chr> <date>     <date>     <S4: Interval>                         <dbl>
## 1 Mary  1935-03-24 2010-08-17 1935-03-24 UTC--2010-08-17 UTC          75.4
## 2 Paul  1966-03-01 2010-08-17 1966-03-01 UTC--2010-08-17 UTC          44.5
## # ... with 1 more variable: years_whole <dbl>
```

```
# Add a line to extract labeled month
baker_dates_cast <- baker_result %>% select(series, baker, contains("date"))

(baker_dates_cast <-  baker_dates_cast %>%
  mutate(last_date_appeared_us = ymd(last_date_appeared)) %>%
  mutate(last_month_us = month(last_date_appeared, label = TRUE)))
```

```
## # A tibble: 95 x 8
##    series baker first_date_appe~ last_date_appea~ first_date_us
##     <int> <chr> <date>           <date>           <date>
##  1      1 Anne~ 2010-08-17       2010-08-24       NA
##  2      1 David 2010-08-17       2010-09-07       NA
##  3      1 Edd   2010-08-17       2010-09-21       NA
##  4      1 Jasm~ 2010-08-17       2010-09-14       NA
##  5      1 Jona~ 2010-08-17       2010-08-31       NA
##  6      1 Lea   2010-08-17       2010-08-17       NA
##  7      1 Loui~ 2010-08-17       2010-08-24       NA
##  8      1 Mark  2010-08-17       2010-08-17       NA
##  9      1 Mira~ 2010-08-17       2010-09-21       NA
## 10      1 Ruth  2010-08-17       2010-09-21       NA
## # ... with 85 more rows, and 3 more variables: last_date_us <date>,
## #   last_date_appeared_us <date>, last_month_us <ord>
```

```
# Make bar chart by last month
ggplot(baker_dates_cast, aes(last_month_us)) + geom_bar()
```



```
# # Create interval between first and last UK dates
# (baker_dates_cast <- baker_dates_cast %>%
#   mutate(time_on_air = interval(first_date_appeared, last_date_appeared )
#
# baker_dates_cast <- baker_dates_cast %>%
#   select(-c(last_month_us, time_on_air))
#
# glimpse(baker_dates_cast)
# #
# # baker_dates_cast <- baker_dates_cast %>%
# #   rename( first_date_appeared_us = first_date_us,
# #          last_date_appeared_us = last_date_us)
```

```
# Create interval between first and last UK dates
# (baker_dates_cast <- baker_dates_cast %>%
#   mutate(time_on_air = lubridate::interval(baker_dates_cast$first_date_appeared_uk, baker_dates_cast$
#          weeks_on_air = time_on_air / weeks(1), # Add a line to create weeks on air variable
#          months_on_air = time_on_air %/% months(1))) # Add a line to create whole months on air varia

# head(baker_dates_cast)
```

Working with strings in tidyverse

```
library(stringr)

baker_result %>%
  mutate(baker_full = str_to_upper(baker_full),
         occupation = str_to_upper(occupation),
         student = str_detect(occupation, "STUDENT")) %>%
  filter(student == TRUE) %>%
  select(baker, occupation, student)
```

```
## # A tibble: 8 x 3
##   baker   occupation                            student
##   <chr>   <chr>                                 <lgl>
## 1 Jason   CIVIL ENGINEERING STUDENT             TRUE
## 2 James   MEDICAL STUDENT                       TRUE
## 3 John    LAW STUDENT                           TRUE
## 4 Ruby    HISTORY OF ART AND PHILOSOPHY STUDENT TRUE
## 5 Martha  STUDENT                               TRUE
## 6 Michael STUDENT                               TRUE
## 7 Rav     STUDENT SUPPORT                       TRUE
## 8 Liam    STUDENT                               TRUE
```