week 9 more like week i need some wine

Wu Xingyi (Kyra)

2023-10-16

Questions

Code along! lets go

```
library(tidyverse)
```

```
## -- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 --
## v dplyr     1.1.2     v readr     2.1.4
## v forcats   1.0.0     v stringr   1.5.0
## v ggplot2   3.4.3     v tibble    3.2.1
## v lubridate 1.9.2     v tidyr     1.3.0
## v purrr     1.0.2
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
## i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become errors
```

```
tidydata <- tribble(
~country,  ~year,  ~cases, ~population,
"Afghanistan", 1999,    745,   19987071,
"Afghanistan",  2000,   2666,   20595360,
"Brazil", 1999,  37737,  172006362,
"Brazil", 2000,  80488,  174504898,
"China",1999, 212258, 1272915272,
"China", 2000, 213766, 1280428583)

tidydata
```

```
## # A tibble: 6 x 4
##   country      year  cases population
##   <chr>       <dbl>  <dbl>      <dbl>
## 1 Afghanistan  1999    745   19987071
## 2 Afghanistan  2000   2666   20595360
## 3 Brazil       1999  37737  172006362
## 4 Brazil       2000  80488  174504898
## 5 China        1999 212258 1272915272
## 6 China        2000 213766 1280428583
```

next up! non tidy data

```
nontidydata <- tribble(
 ~country,~year,~rate,
 "Afghanistan",  1999, "745/19987071",
"Afghanistan",  2000, "2666/20595360",
"Brazil", 1999,  "37737/172006362",
"Brazil", 2000,  "80488/174504898",
"China",1999, "212258/1272915272",
"China", 2000, "213766/1280428583")

nontidydata
```

```
## # A tibble: 6 x 3
##   country      year rate
##   <chr>       <dbl> <chr>
## 1 Afghanistan  1999 745/19987071
## 2 Afghanistan  2000 2666/20595360
## 3 Brazil       1999 37737/172006362
## 4 Brazil       2000 80488/174504898
## 5 China        1999 212258/1272915272
## 6 China        2000 213766/1280428583
```

next up!   non tidy data

```
nontidydata <- tribble(
 ~country,~year,~rate,
 "Afghanistan",  1999, "745/19987071",
"Afghanistan",  2000, "2666/20595360",
"Brazil", 1999,  "37737/172006362",
"Brazil", 2000,  "80488/174504898",
"China",1999, "212258/1272915272",
"China", 2000, "213766/1280428583")

nontidydata
```

```
## # A tibble: 6 x 3
##   country      year rate
##   <chr>       <dbl> <chr>
## 1 Afghanistan  1999 745/19987071
## 2 Afghanistan  2000 2666/20595360
## 3 Brazil       1999 37737/172006362
## 4 Brazil       2000 80488/174504898
## 5 China        1999 212258/1272915272
## 6 China        2000 213766/1280428583
```

tidying data   non tidy examples

```
tidieddata <- nontidydata %>%
  separate(rate, into = c("cases",
                          "population"),
                            sep = "/")
tidieddata
```

```
## # A tibble: 6 x 4
##   country      year cases  population
##   <chr>       <dbl> <chr>  <chr>
## 1 Afghanistan  1999 745    19987071
## 2 Afghanistan  2000 2666   20595360
## 3 Brazil       1999 37737  172006362
## 4 Brazil       2000 80488  174504898
## 5 China        1999 212258 1272915272
## 6 China        2000 213766 1280428583
```

new tidied data tidying more?

```
newtidieddata <- tidieddata %>%
  pivot_longer(
    cols = cases:population,
    names_to = "measurement",
    values_to = "value"
  )
newtidieddata
```

```
## # A tibble: 12 x 4
##    country      year measurement value
##    <chr>       <dbl> <chr>       <chr>
##  1 Afghanistan  1999 cases       745
##  2 Afghanistan  1999 population  19987071
##  3 Afghanistan  2000 cases       2666
##  4 Afghanistan  2000 population  20595360
##  5 Brazil       1999 cases       37737
##  6 Brazil       1999 population  172006362
##  7 Brazil       2000 cases       80488
##  8 Brazil       2000 population  174504898
##  9 China        1999 cases       212258
## 10 China        1999 population  1272915272
## 11 China        2000 cases       213766
## 12 China        2000 population  1280428583
```

tidied data example 2

```
df <- tribble(
  ~id,  ~bp1, ~bp2,
   "A",  100,  120,
   "B",  140,  115,
   "C",  120,  125
)
df
```

```
## # A tibble: 3 x 3
##   id      bp1   bp2
##   <chr> <dbl> <dbl>
## 1 A       100   120
## 2 B       140   115
## 3 C       120   125
```

```r
df %>%
  pivot_longer(
    cols = bp1:bp2,
    names_to = "measurement",
    values_to = "value"
)
```

```
## # A tibble: 6 x 3
##   id    measurement value
##   <chr> <chr>       <dbl>
## 1 A     bp1           100
## 2 A     bp2           120
## 3 B     bp1           140
## 4 B     bp2           115
## 5 C     bp1           120
## 6 C     bp2           125
```

tidied data   example 3

```r
newtidieddata
```

```
## # A tibble: 12 x 4
##    country      year measurement value
##    <chr>       <dbl> <chr>       <chr>
##  1 Afghanistan  1999 cases       745
##  2 Afghanistan  1999 population  19987071
##  3 Afghanistan  2000 cases       2666
##  4 Afghanistan  2000 population  20595360
##  5 Brazil       1999 cases       37737
##  6 Brazil       1999 population  172006362
##  7 Brazil       2000 cases       80488
##  8 Brazil       2000 population  174504898
##  9 China        1999 cases       212258
## 10 China        1999 population  1272915272
## 11 China        2000 cases       213766
## 12 China        2000 population  1280428583
```

```r
newtidieddata %>%
  pivot_wider(names_from="measurement",
              values_from="value")
```

```
## # A tibble: 6 x 4
##   country      year cases  population
##   <chr>       <dbl> <chr>  <chr>
## 1 Afghanistan  1999 745    19987071
## 2 Afghanistan  2000 2666   20595360
## 3 Brazil       1999 37737  172006362
## 4 Brazil       2000 80488  174504898
## 5 China        1999 212258 1272915272
## 6 China        2000 213766 1280428583
```

tidied data example 4

```
df <- tribble(
  ~id, ~measurement, ~value,
  "A",        "bp1",    100,
  "B",        "bp1",    140,
  "B",        "bp2",    115,
  "A",        "bp2",    120,
  "A",        "bp3",    105
)
df
```

```
## # A tibble: 5 x 3
##   id    measurement value
##   <chr> <chr>       <dbl>
## 1 A     bp1           100
## 2 B     bp1           140
## 3 B     bp2           115
## 4 A     bp2           120
## 5 A     bp3           105
```

```
df %>%
  pivot_wider(
    names_from = measurement,
    values_from = value
  )
```

```
## # A tibble: 2 x 4
##   id      bp1   bp2   bp3
##   <chr> <dbl> <dbl> <dbl>
## 1 A       100   120   105
## 2 B       140   115    NA
```

scraping data from the web trying it out

```
#install.packages("rvest")
library(rvest)
```

```
##
## Attaching package: 'rvest'

## The following object is masked from 'package:readr':
##
##     guess_encoding
```

```
webpage <- read_html("https://books.toscrape.com/")
table <-html_elements(webpage,"body")
```

calling APIs

```r
#install.packages(c("httr","jsonlite"))
library(jsonlite)
```

```
##
## Attaching package: 'jsonlite'

## The following object is masked from 'package:purrr':
##
##     flatten
```

```r
library(httr)

# current data
current_county_data_url <- "https://api.covidactnow.org/v2/counties.csv?apiKey=33382de96fd8441fb6c"
raw_data <- GET(current_county_data_url)
raw_data$status
```

```
## [1] 403
```

```r
head(raw_data$content)
```

```
## [1] 7b 22 65 72 72 6f
```

```r
#install.packages(c("httr","jsonlite"))
library(jsonlite)
library(httr)

# historic data
historic_county_data_url <-
"https://api.covidactnow.org/v2/counties.timeseries.csv?apiKey=33382de96fd8441fb6c1eca82b3bd4ec"
raw_data <- GET(historic_county_data_url)
raw_data$status
```

```
## [1] 200
```

```r
head(raw_data$content)
```

```
## [1] 64 61 74 65 2c 63
```

```r
#install.packages(c("httr","jsonlite"))
library(jsonlite)
library(httr)

# individual location data
individual_loc_data_url <-
"https://api.covidactnow.org/v2/county/{49}.csv?apiKey=33382de96fd8441fb6c1eca82b3bd4ec"
raw_data <- GET(individual_loc_data_url)
raw_data$status
```

```
## [1] 403
```

```r
raw_data$content
```

```
##   [1] 3c 3f 78 6d 6c 20 76 65 72 73 69 6f 6e 3d 22 31 2e 30 22 20 65 6e 63 6f 64
##  [26] 69 6e 67 3d 22 55 54 46 2d 38 22 3f 3e 0a 3c 45 72 72 6f 72 3e 3c 43 6f 64
##  [51] 65 3e 41 63 63 65 73 73 44 65 6e 69 65 64 3c 2f 43 6f 64 65 3e 3c 4d 65 73
##  [76] 73 61 67 65 3e 41 63 63 65 73 73 20 44 65 6e 69 65 64 3c 2f 4d 65 73 73 61
## [101] 67 65 3e 3c 52 65 71 75 65 73 74 49 64 3e 45 43 44 31 4b 42 43 34 4d 53 4b
## [126] 42 54 45 32 50 3c 2f 52 65 71 75 65 73 74 49 64 3e 3c 48 6f 73 74 49 64 3e
## [151] 72 4e 36 47 43 4e 38 36 46 45 34 36 39 56 4f 34 46 71 5a 35 33 64 47 4f 74
## [176] 5a 72 68 36 4f 2b 32 53 45 51 31 4b 38 48 62 52 59 66 6c 50 48 4d 79 42 33
## [201] 42 49 61 33 43 66 6a 61 4a 55 69 6e 51 73 58 52 6d 6d 41 61 41 65 42 73 6f
## [226] 3d 3c 2f 48 6f 73 74 49 64 3e 3c 2f 45 72 72 6f 72 3e
```

```r
head(raw_data$content)
```

```
## [1] 3c 3f 78 6d 6c 20
```

now for the challenge

loading the packages

```r
library(tidyverse)
```

Pivot longer to arrange the names of the columns, wk1 to wk76 under a new variable/column week (Hint use: cols = starts_with("wk") as the argument to pivot_longer())

```r
billboard_long <- billboard %>%
  pivot_longer(cols = starts_with("wk"), names_to = "week", values_to = "rank", values_drop_na = TRUE)

billboard_long
```

```
## # A tibble: 5,307 x 5
##    artist  track                   date.entered week   rank
##    <chr>   <chr>                   <date>       <chr> <dbl>
##  1 2 Pac   Baby Don't Cry (Keep... 2000-02-26   wk1      87
##  2 2 Pac   Baby Don't Cry (Keep... 2000-02-26   wk2      82
##  3 2 Pac   Baby Don't Cry (Keep... 2000-02-26   wk3      72
##  4 2 Pac   Baby Don't Cry (Keep... 2000-02-26   wk4      77
##  5 2 Pac   Baby Don't Cry (Keep... 2000-02-26   wk5      87
##  6 2 Pac   Baby Don't Cry (Keep... 2000-02-26   wk6      94
##  7 2 Pac   Baby Don't Cry (Keep... 2000-02-26   wk7      99
##  8 2Ge+her The Hardest Part Of ... 2000-09-02   wk1      91
##  9 2Ge+her The Hardest Part Of ... 2000-09-02   wk2      87
## 10 2Ge+her The Hardest Part Of ... 2000-09-02   wk3      92
## # i 5,297 more rows
```

Clean the "week" column to have only the week numbers (1 for wk1, 2 for wk2, etc.)

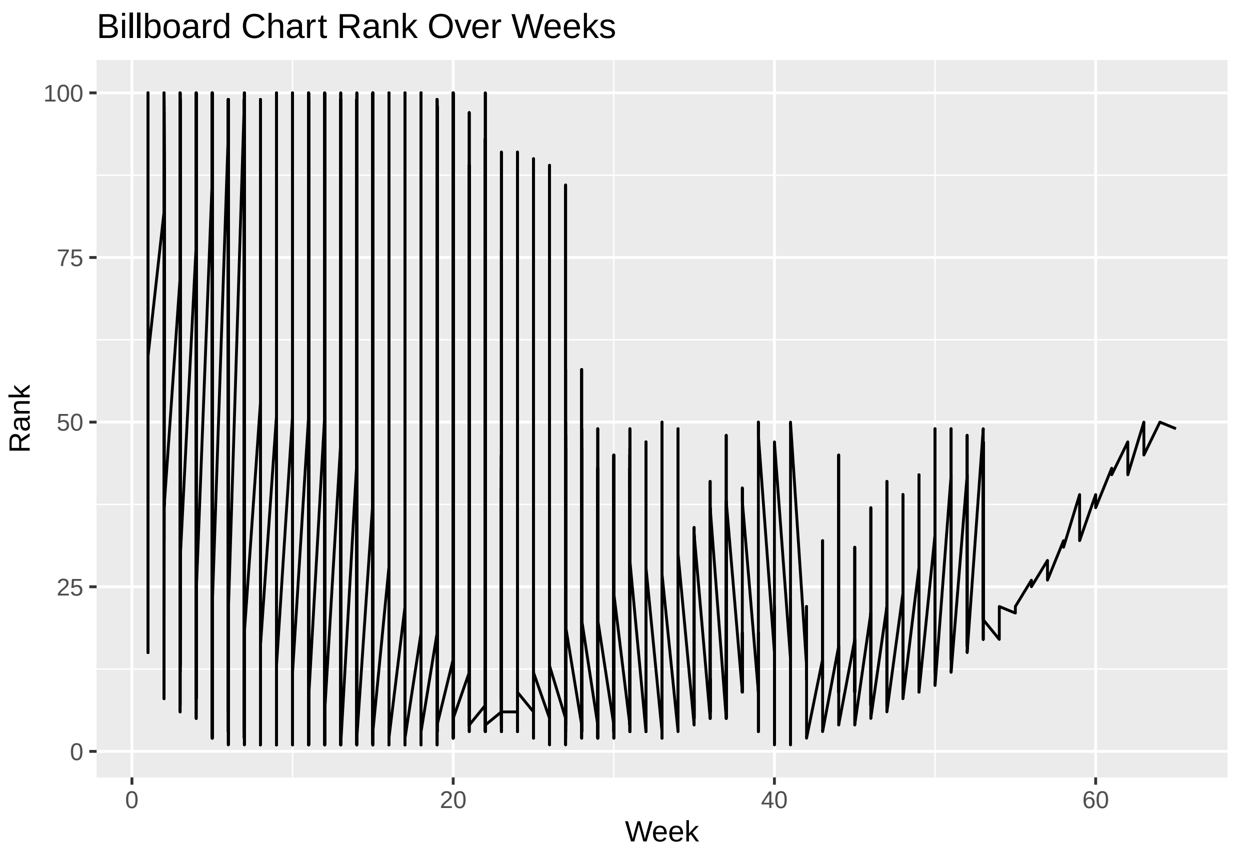
```
billboard_long <- billboard_long %>%
  mutate(week = parse_number(week))

billboard_long
```

```
## # A tibble: 5,307 x 5
##    artist  track                   date.entered  week  rank
##    <chr>   <chr>                   <date>       <dbl> <dbl>
##  1 2 Pac   Baby Don't Cry (Keep... 2000-02-26       1    87
##  2 2 Pac   Baby Don't Cry (Keep... 2000-02-26       2    82
##  3 2 Pac   Baby Don't Cry (Keep... 2000-02-26       3    72
##  4 2 Pac   Baby Don't Cry (Keep... 2000-02-26       4    77
##  5 2 Pac   Baby Don't Cry (Keep... 2000-02-26       5    87
##  6 2 Pac   Baby Don't Cry (Keep... 2000-02-26       6    94
##  7 2 Pac   Baby Don't Cry (Keep... 2000-02-26       7    99
##  8 2Ge+her The Hardest Part Of ... 2000-09-02       1    91
##  9 2Ge+her The Hardest Part Of ... 2000-09-02       2    87
## 10 2Ge+her The Hardest Part Of ... 2000-09-02       3    92
## # i 5,297 more rows
```

Plot the rank along the y-axis and week along the x-axis, joining the data points with 'geom_line()'

```
ggplot(billboard_long, aes(x = week, y = rank)) +
  geom_line() +
  labs(title = "Billboard Chart Rank Over Weeks", x = "Week", y = "Rank")
```



next question: loading the packages

```r
library(tidyverse)
```

Create as many columns as the distinct entries of the variable, measure_cd

```r
pivot_wider(data = cms_patient_experience,
            names_from = measure_cd,
            values_from = prf_rate)
```

```
## # A tibble: 500 x 9
##    org_pac_id org_nm           measure_title CAHPS_GRP_1 CAHPS_GRP_2 CAHPS_GRP_3
##    <chr>      <chr>            <chr>               <dbl>       <dbl>       <dbl>
##  1 0446157747 USC CARE MEDICA~ CAHPS for MI~          63          NA          NA
##  2 0446157747 USC CARE MEDICA~ CAHPS for MI~          NA          87          NA
##  3 0446157747 USC CARE MEDICA~ CAHPS for MI~          NA          NA          86
##  4 0446157747 USC CARE MEDICA~ CAHPS for MI~          NA          NA          NA
##  5 0446157747 USC CARE MEDICA~ CAHPS for MI~          NA          NA          NA
##  6 0446157747 USC CARE MEDICA~ CAHPS for MI~          NA          NA          NA
##  7 0446162697 ASSOCIATION OF ~ CAHPS for MI~          59          NA          NA
##  8 0446162697 ASSOCIATION OF ~ CAHPS for MI~          NA          85          NA
##  9 0446162697 ASSOCIATION OF ~ CAHPS for MI~          NA          NA          83
## 10 0446162697 ASSOCIATION OF ~ CAHPS for MI~          NA          NA          NA
## # i 490 more rows
## # i 3 more variables: CAHPS_GRP_5 <dbl>, CAHPS_GRP_8 <dbl>, CAHPS_GRP_12 <dbl>
```

Create as many columns as the distinct entries of the variable, measure_cd, the values in the columns should correspond to the ones listed in the column, prf_rate

```
pivot_wider(data = cms_patient_experience,
            names_from = measure_cd,
            values_from = prf_rate)
```

```
## # A tibble: 500 x 9
##    org_pac_id org_nm           measure_title CAHPS_GRP_1 CAHPS_GRP_2 CAHPS_GRP_3
##    <chr>      <chr>            <chr>               <dbl>       <dbl>       <dbl>
##  1 0446157747 USC CARE MEDICA~ CAHPS for MI~          63          NA          NA
##  2 0446157747 USC CARE MEDICA~ CAHPS for MI~          NA          87          NA
##  3 0446157747 USC CARE MEDICA~ CAHPS for MI~          NA          NA          86
##  4 0446157747 USC CARE MEDICA~ CAHPS for MI~          NA          NA          NA
##  5 0446157747 USC CARE MEDICA~ CAHPS for MI~          NA          NA          NA
##  6 0446157747 USC CARE MEDICA~ CAHPS for MI~          NA          NA          NA
##  7 0446162697 ASSOCIATION OF ~ CAHPS for MI~          59          NA          NA
##  8 0446162697 ASSOCIATION OF ~ CAHPS for MI~          NA          85          NA
##  9 0446162697 ASSOCIATION OF ~ CAHPS for MI~          NA          NA          83
## 10 0446162697 ASSOCIATION OF ~ CAHPS for MI~          NA          NA          NA
## # i 490 more rows
## # i 3 more variables: CAHPS_GRP_5 <dbl>, CAHPS_GRP_8 <dbl>, CAHPS_GRP_12 <dbl>
```

The output doesn't look quite right; we still seem to have multiple rows for each organization. That's because, we also need to tell pivot_wider() which column or columns have values that uniquely identify each row; in this case those are the variables starting with "org"

```
pivot_wider(data = cms_patient_experience,
            names_from = measure_cd,
            values_from = prf_rate,
            id_cols = starts_with("org"))
```

```
## # A tibble: 95 x 8
##    org_pac_id org_nm CAHPS_GRP_1 CAHPS_GRP_2 CAHPS_GRP_3 CAHPS_GRP_5 CAHPS_GRP_8
##    <chr>      <chr>        <dbl>       <dbl>       <dbl>       <dbl>       <dbl>
##  1 0446157747 USC C~          63          87          86          57          85
##  2 0446162697 ASSOC~          59          85          83          63          88
##  3 0547164295 BEAVE~          49          NA          75          44          73
##  4 0749333730 CAPE ~          67          84          85          65          82
##  5 0840104360 ALLIA~          66          87          87          64          87
##  6 0840109864 REX H~          73          87          84          67          91
##  7 0840513552 SCL H~          58          83          76          58          78
##  8 0941545784 GRITM~          46          86          81          54          NA
##  9 1052612785 COMMU~          65          84          80          58          87
## 10 1254237779 OUR L~          61          NA          NA          65          NA
## # i 85 more rows
## # i 1 more variable: CAHPS_GRP_12 <dbl>
```