Challenge-9

Loy Yee Keen

2023-10-18

```
knitr::opts_chunk$set(tidy.opts=list(width.cutoff=80), tidy=TRUE)
```

#Code Along

```
library(tidyverse) #load tidyverse package
```

```
## Warning: package 'tidyverse' was built under R version 4.2.3

## Warning: package 'ggplot2' was built under R version 4.2.3

## Warning: package 'tibble' was built under R version 4.2.3

## Warning: package 'tidyr' was built under R version 4.2.3

## Warning: package 'readr' was built under R version 4.2.3

## Warning: package 'purrr' was built under R version 4.2.3

## Warning: package 'dplyr' was built under R version 4.2.3

## Warning: package 'stringr' was built under R version 4.2.3

## Warning: package 'forcats' was built under R version 4.2.3

## Warning: package 'lubridate' was built under R version 4.2.3

## -- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 --
## v dplyr     1.1.1     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
```

```r
#create a dataframe, tidydata, with values manually keyed into the tribble function
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
```

```r
#create a dataframe, nontidydata, with values manually keyed into the tribble function
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
```

```r
#separate the column rate into 2 columns--cases and population
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
```

```
# Assign the column names of all the columns from cases to population as the values of a
#new column called measurement.
#Assign the values of all the columns from cases to population into a new column called value
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
```

```
# create a new dataframe, df, with the values manually keyed into the function tribble
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
```

```
# Assign the column names of all the columns from bp1 to bp2 as the values of a
#new column called measurement.
# Assign the values of all the columns from bp1 to bp2 into a new column called value
```

```
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
```

```
#Create new columns based on the values of the column "measurement"
#Assign to these columns the values from the column "value"
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
```

```
#Create a dataframe, df, with values manually keyed into the tribble function
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
```

```
#Create new columns based on the values of the column "measurement"
#Assign to these columns the values from the column "value"
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
```

#Challenge In your console, type, billboard A. It will open a dataset where each observation/row is a song B. Columns wk1-wk76 have the rank of the songs in that week

```
library(tidyverse)
billboard
```

```
## # A tibble: 317 x 79
##    artist     track date.entered   wk1   wk2   wk3   wk4   wk5   wk6   wk7   wk8
##    <chr>      <chr> <date>       <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>
##  1 2 Pac      Baby~ 2000-02-26      87    82    72    77    87    94    99    NA
##  2 2Ge+her    The ~ 2000-09-02      91    87    92    NA    NA    NA    NA    NA
##  3 3 Doors D~ Kryp~ 2000-04-08      81    70    68    67    66    57    54    53
##  4 3 Doors D~ Loser 2000-10-21      76    76    72    69    67    65    55    59
##  5 504 Boyz   Wobb~ 2000-04-15      57    34    25    17    17    31    36    49
##  6 98^0       Give~ 2000-08-19      51    39    34    26    26    19     2     2
##  7 A*Teens    Danc~ 2000-07-08      97    97    96    95   100    NA    NA    NA
##  8 Aaliyah    I Do~ 2000-01-29      84    62    51    41    38    35    35    38
##  9 Aaliyah    Try ~ 2000-03-18      59    53    38    28    21    18    16    14
## 10 Adams, Yo~ Open~ 2000-08-26      76    76    74    69    68    67    61    58
## # i 307 more rows
## # i 68 more variables: wk9 <dbl>, wk10 <dbl>, wk11 <dbl>, wk12 <dbl>,
## #   wk13 <dbl>, wk14 <dbl>, wk15 <dbl>, wk16 <dbl>, wk17 <dbl>, wk18 <dbl>,
## #   wk19 <dbl>, wk20 <dbl>, wk21 <dbl>, wk22 <dbl>, wk23 <dbl>, wk24 <dbl>,
## #   wk25 <dbl>, wk26 <dbl>, wk27 <dbl>, wk28 <dbl>, wk29 <dbl>, wk30 <dbl>,
## #   wk31 <dbl>, wk32 <dbl>, wk33 <dbl>, wk34 <dbl>, wk35 <dbl>, wk36 <dbl>,
## #   wk37 <dbl>, wk38 <dbl>, wk39 <dbl>, wk40 <dbl>, wk41 <dbl>, wk42 <dbl>, ...
```

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())

```
billboard %>% pivot_longer(
  cols=starts_with("wk"),
  names_to="week",
  values_to="rank"
  )
```

```
## # A tibble: 24,092 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 2 Pac  Baby Don't Cry (Keep... 2000-02-26   wk8      NA
##  9 2 Pac  Baby Don't Cry (Keep... 2000-02-26   wk9      NA
## 10 2 Pac  Baby Don't Cry (Keep... 2000-02-26   wk10     NA
## # i 24,082 more rows
```

Clean the data by removing observations/rows with entries NA (Use: values_drop_na = TRUE in pivot longer)

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

```
## # 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
```

Ensure that the column, week, has only the number of the week, 1 for wk1, 2 for wk2 and so on. (Use: mutate(week = parse_number(week)))

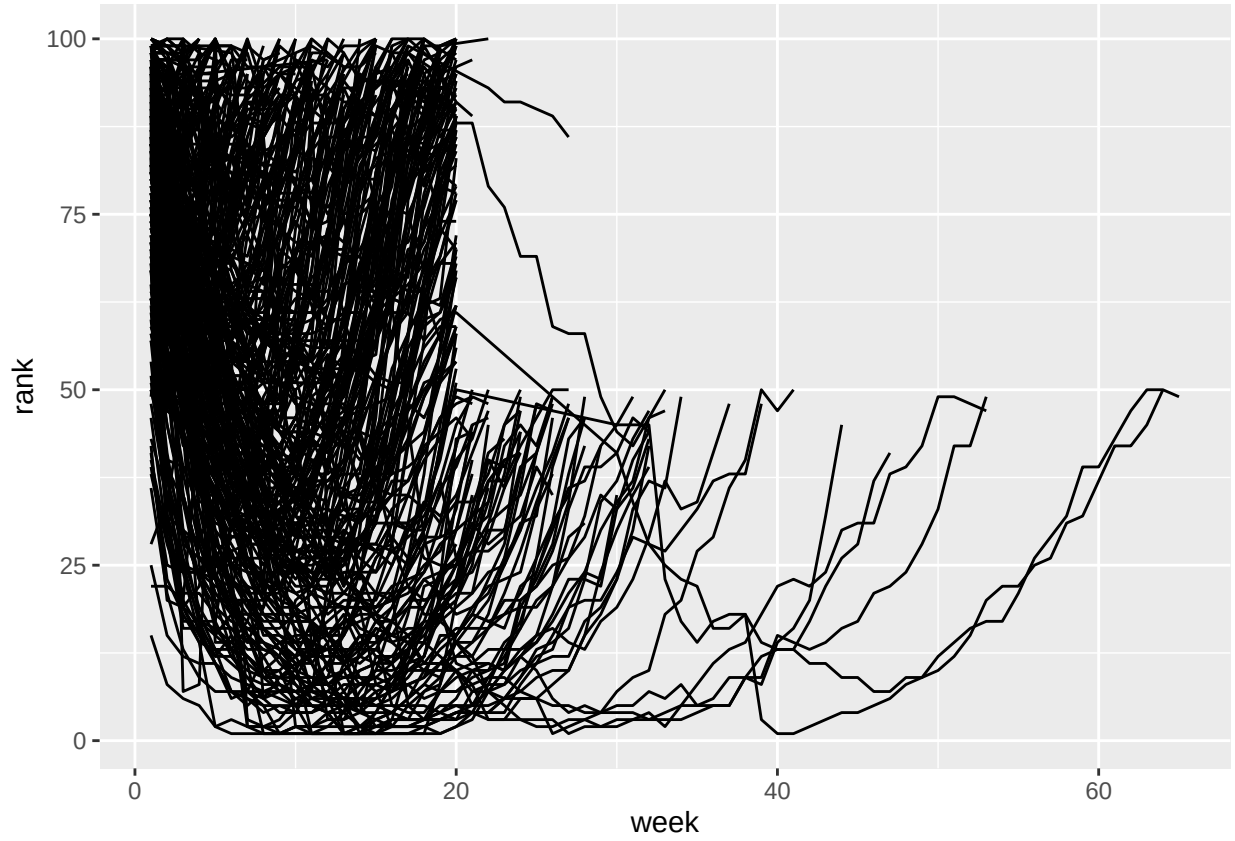
```
billboard %>% pivot_longer(
  cols=starts_with("wk"),
  names_to="week",
  values_to="rank",
  values_drop_na = TRUE) %>%
  mutate(week = parse_number(week)
         )
```

```
## # 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 y axis and week along x axis and join the data points using geom_line()

```
library(ggplot2)
billboard<-billboard %>% pivot_longer(cols=starts_with("wk"),
names_to="week",values_to="rank",
values_drop_na = TRUE) %>%
  mutate(week = parse_number(week)
         )
ggplot(data=billboard)+
  aes(x=week,y=rank,group=track)+
  geom_line()
```



In your console, type cms_patient_experience A. A dataset from the Centers of Medicare and Medicaid services that collects data about patient experiences B. The core unit being studied is an organization, but each organization is spread across six rows, with one row for each measurement taken in the survey organization

```
cms_patient_experience
```

```
## # A tibble: 500 x 5
##    org_pac_id org_nm                              measure_cd measure_title prf_rate
##    <chr>      <chr>                               <chr>      <chr>            <dbl>
##  1 0446157747 USC CARE MEDICAL GROUP INC          CAHPS_GRP~ CAHPS for MI~       63
##  2 0446157747 USC CARE MEDICAL GROUP INC          CAHPS_GRP~ CAHPS for MI~       87
##  3 0446157747 USC CARE MEDICAL GROUP INC          CAHPS_GRP~ CAHPS for MI~       86
##  4 0446157747 USC CARE MEDICAL GROUP INC          CAHPS_GRP~ CAHPS for MI~       57
##  5 0446157747 USC CARE MEDICAL GROUP INC          CAHPS_GRP~ CAHPS for MI~       85
##  6 0446157747 USC CARE MEDICAL GROUP INC          CAHPS_GRP~ CAHPS for MI~       24
##  7 0446162697 ASSOCIATION OF UNIVERSITY PHYSI~    CAHPS_GRP~ CAHPS for MI~       59
##  8 0446162697 ASSOCIATION OF UNIVERSITY PHYSI~    CAHPS_GRP~ CAHPS for MI~       85
##  9 0446162697 ASSOCIATION OF UNIVERSITY PHYSI~    CAHPS_GRP~ CAHPS for MI~       83
## 10 0446162697 ASSOCIATION OF UNIVERSITY PHYSI~    CAHPS_GRP~ CAHPS for MI~       63
## # i 490 more rows
```

Using pivot_wider(), 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

```
cms_patient_experience %>% pivot_wider(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" To your answer to the previous step, include, id_cols = starts_with("org"), as an argument to the function, pivot_wider. Now you will be able to see the id of each organisation, the corresponding name and the metrics

```
cms_patient_experience %>% pivot_wider(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>
```