Pivoting Table with R

JM Muriuki

2024-06-29

attach libraries

```
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   3.4.2     v tibble    3.2.1
## v lubridate 1.9.2     v tidyr     1.3.0
## v purrr     1.0.1
## -- 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
```

```
library(readxl)
library(here)
```

```
## here() starts at C:/Users/admin/OneDrive/Documents/Data Analyst James/R for Data Science
```

```
library(skimr) # install.packages('skimr')
```

```
## Warning: package 'skimr' was built under R version 4.3.3
```

```
library(kableExtra) # install.packages('kableExtra')
```

```
## Warning: package 'kableExtra' was built under R version 4.3.3
```

```
##
## Attaching package: 'kableExtra'
##
## The following object is masked from 'package:dplyr':
##
##     group_rows
```

getting to know my directory

```
getwd()
```

```
## [1] "C:/Users/admin/OneDrive/Documents/Data Analyst James/R for Data Science"
```

read in data

```
lobsters <- read_excel(here("lobsters.xlsx"),skip=4)
```

pivoting a data

```
#data %>% group_by() %>% summarize()
```

group_by one variable

```
lobsters %>% group_by(year) %>% summarize(count_by_year = n())
```

```
## # A tibble: 5 x 2
##    year count_by_year
##   <dbl>         <int>
## 1  2012           231
## 2  2013           243
## 3  2014           510
## 4  2015          1100
## 5  2016           809
```

when you don't group_by first?

```
lobsters %>%  summarise(count=n())
```

```
## # A tibble: 1 x 1
##   count
##   <int>
## 1  2893
```

what if we only group_by?

```
lobsters %>% group_by(year)
```

```
## # A tibble: 2,893 x 7
## # Groups:   year [5]
##     year month date    site  transect replicate size_mm
##    <dbl> <dbl> <chr>   <chr>    <dbl> <chr>       <dbl>
##  1  2012     8 8/20/12 ivee         3 A              70
##  2  2012     8 8/20/12 ivee         3 B              60
##  3  2012     8 8/20/12 ivee         3 B              65
##  4  2012     8 8/20/12 ivee         3 B              70
##  5  2012     8 8/20/12 ivee         3 B              85
##  6  2012     8 8/20/12 ivee         3 C              60
##  7  2012     8 8/20/12 ivee         3 C              65
##  8  2012     8 8/20/12 ivee         3 C              67
##  9  2012     8 8/20/12 ivee         3 D              70
## 10  2012     8 8/20/12 ivee         4 B              85
## # i 2,883 more rows
```

group_by multiple variables

```
lobsters %>%  group_by(site,year) %>% summarise(count_by_siteyear=n())
```

```
## `summarise()` has grouped output by 'site'. You can override using the
## `.groups` argument.
```

```
## # A tibble: 25 x 3
```

```
## # Groups:   site [5]
##    site   year count_by_siteyear
##    <chr> <dbl>              <int>
##  1 aque   2012                 38
##  2 aque   2013                 32
##  3 aque   2014                100
##  4 aque   2015                 83
##  5 aque   2016                 48
##  6 carp   2012                 78
##  7 carp   2013                 93
##  8 carp   2014                 79
##  9 carp   2015                 90
## 10 carp   2016                231
## # i 15 more rows
```

summarize multiple variables

```
lobsters %>% group_by(site, year) %>% summarize(count_by_siteyear = n() ,mean_size_mm = mean(size_mm))
```

```
## `summarise()` has grouped output by 'site'. You can override using the
## `.groups` argument.
```

```
## # A tibble: 25 x 4
## # Groups:   site [5]
##    site   year count_by_siteyear mean_size_mm
##    <chr> <dbl>              <int>        <dbl>
##  1 aque   2012                 38         71
##  2 aque   2013                 32         72.1
##  3 aque   2014                100         76.9
##  4 aque   2015                 83         68.5
##  5 aque   2016                 48         68.7
##  6 carp   2012                 78         74.4
##  7 carp   2013                 93         76.6
##  8 carp   2014                 79         NA
##  9 carp   2015                 90         70.7
## 10 carp   2016                231         68.9
## # i 15 more rows
```

```
# some of the means are  passed as NA because one or more   values in that year are NA.

lobsters %>% group_by(site, year) %>% summarize(count_by_siteyear =  n(), mean_size_mm = mean(size_mm, r
```

```
## `summarise()` has grouped output by 'site'. You can override using the
## `.groups` argument.
```

```
## # A tibble: 25 x 5
## # Groups:   site [5]
##    site   year count_by_siteyear mean_size_mm sd_size_mm
##    <chr> <dbl>              <int>        <dbl>      <dbl>
##  1 aque   2012                 38         71        10.2
##  2 aque   2013                 32         72.1      12.3
##  3 aque   2014                100         76.9       9.32
##  4 aque   2015                 83         68.5      12.6
##  5 aque   2016                 48         68.7      12.5
##  6 carp   2012                 78         74.4      14.6
##  7 carp   2013                 93         76.6       8.71
```

```
##  8 carp   2014                79         79.1       8.57
##  9 carp   2015                90         70.7      14.6
## 10 carp   2016               231         68.9      12.5
## # i 15 more rows
```

Adding a variable assignment to that first line:

```
siteyear_summary <- lobsters %>% group_by(site, year) %>% summarize(count_by_siteyear =  n(), mean_size_
```

```
## `summarise()` has grouped output by 'site'. You can override using the
## `.groups` argument.
```

inspect our new variable

```
siteyear_summary
```

```
## # A tibble: 25 x 5
## # Groups:   site [5]
##    site   year count_by_siteyear mean_size_mm sd_size_mm
##    <chr> <dbl>             <int>        <dbl>      <dbl>
##  1 aque   2012                38         71        10.2
##  2 aque   2013                32         72.1      12.3
##  3 aque   2014               100         76.9       9.32
##  4 aque   2015                83         68.5      12.6
##  5 aque   2016                48         68.7      12.5
##  6 carp   2012                78         74.4      14.6
##  7 carp   2013                93         76.6       8.71
##  8 carp   2014                79         79.1       8.57
##  9 carp   2015                90         70.7      14.6
## 10 carp   2016               231         68.9      12.5
## # i 15 more rows
```

Table formatting with kable()

kable() offers a nice presentation of our table

```
## make a table with our new variable
siteyear_summary %>% kable()
```

site	year	count_by_siteyear	mean_size_mm	sd_size_mm
aque	2012	38	71.00000	10.150223
aque	2013	32	72.12500	12.262584
aque	2014	100	76.92000	9.321074
aque	2015	83	68.45783	12.555536
aque	2016	48	68.68750	12.510687
carp	2012	78	74.35897	14.616282
carp	2013	93	76.56989	8.709562
carp	2014	79	79.08974	8.569329
carp	2015	90	70.65556	14.646517
carp	2016	231	68.90476	12.470122
ivee	2012	26	66.07692	12.092719
ivee	2013	40	73.77500	7.640941
ivee	2014	132	76.02273	17.860984

site	year	count_by_siteyear	mean_size_mm	sd_size_mm
ivee	2015	361	69.80332	17.470534
ivee	2016	193	71.61658	13.450454
mohk	2012	83	77.25301	10.587433
mohk	2013	15	71.86667	10.190098
mohk	2014	36	75.75000	10.038142
mohk	2015	296	59.19932	16.770357
mohk	2016	210	63.01286	11.875763
napl	2012	6	73.00000	11.747340
napl	2013	63	75.31746	12.989854
napl	2014	163	79.51572	9.556531
napl	2015	270	78.24074	12.438899
napl	2016	127	74.39370	10.732060

Activity

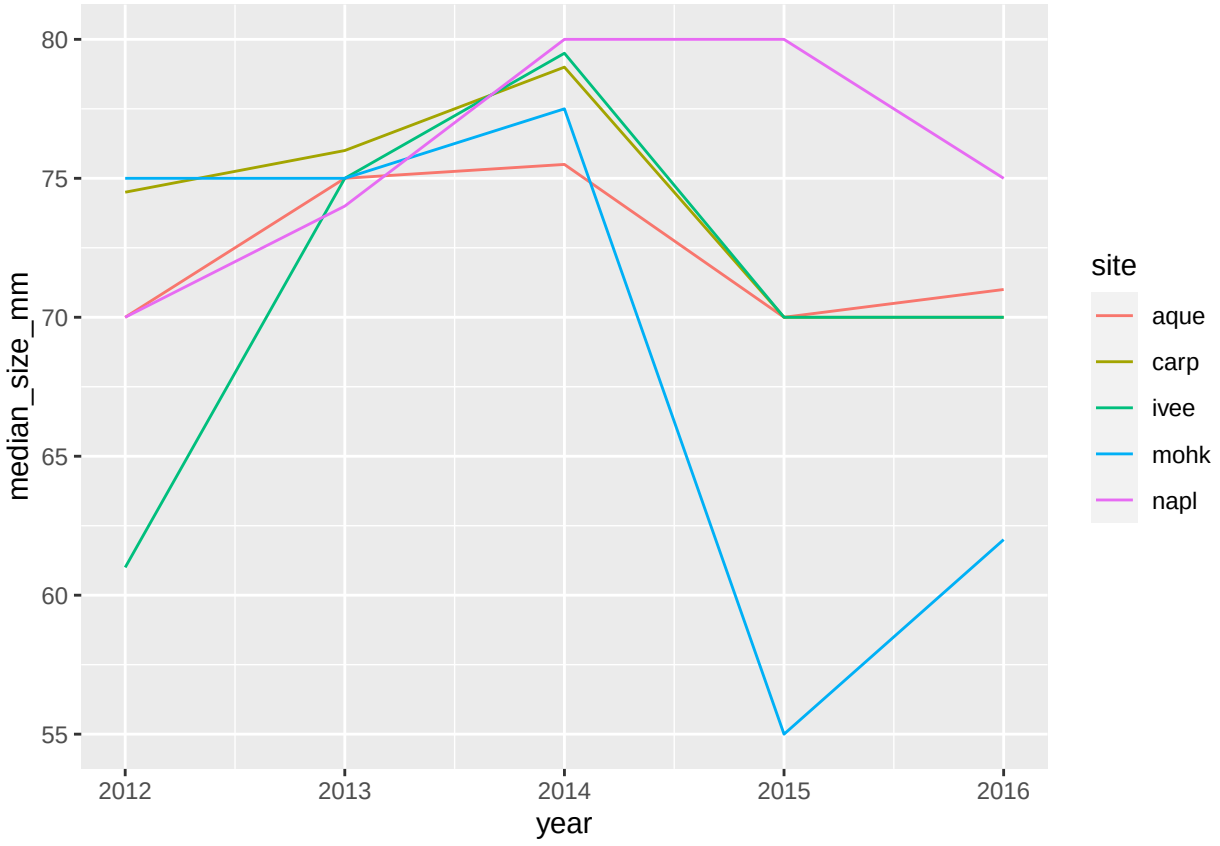
Building from our analysis

```
siteyear_summary <- lobsters %>% group_by(site,year) %>% summarise(count_by_siteyear= n(),mean_size_mm=m
```

```
## `summarise()` has grouped output by 'site'. You can override using the
## `.groups` argument.
```

a ggplot option:

```
ggplot(data = siteyear_summary,aes(x=year,y=median_size_mm,color=site))+geom_line()
```



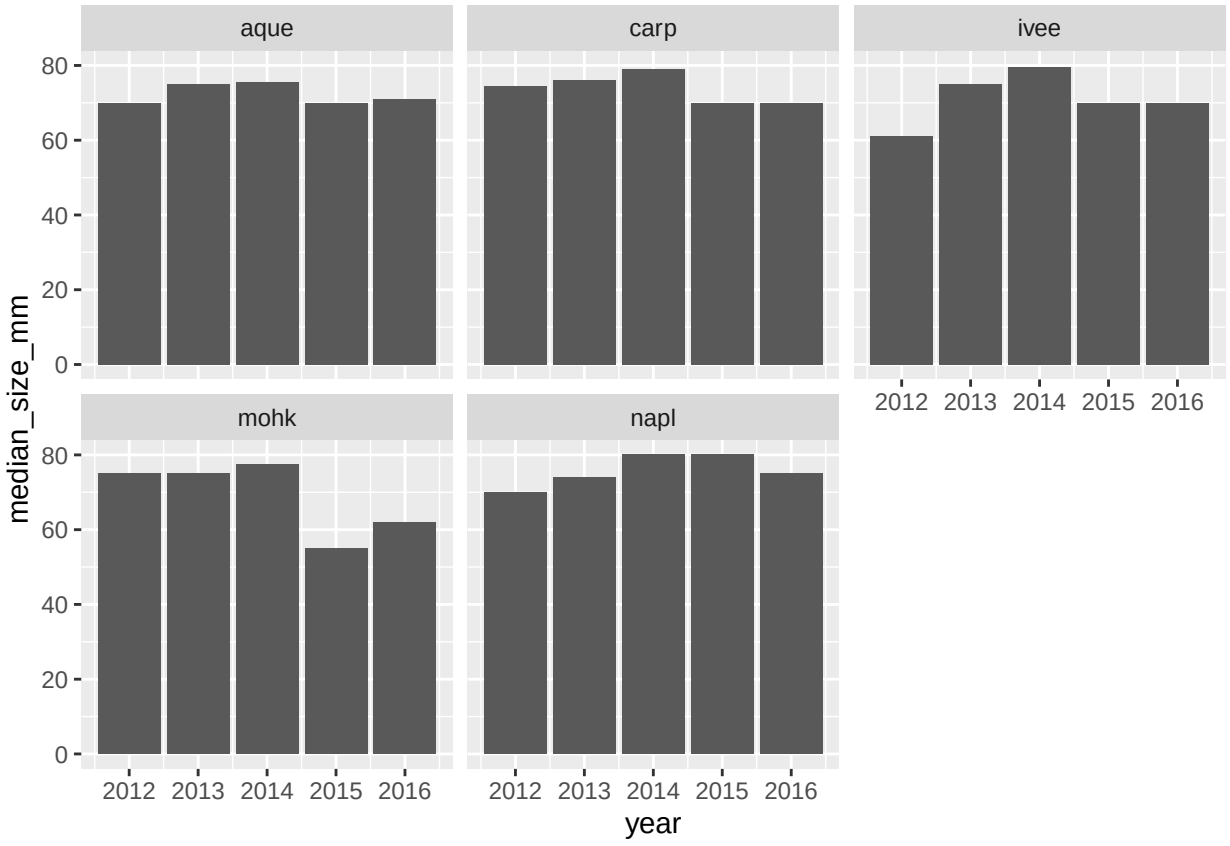
save the image

```
ggsave(here("figures","lobsters-line.png"))
```

```
## Saving 6.5 x 4.5 in image
```

another option

```
ggplot(siteyear_summary,aes(x=year,y=median_size_mm))+geom_col()+facet_wrap(~site)
```



save here

```
ggsave(here("figures","lobsters-col.png"))
```

```
## Saving 6.5 x 4.5 in image
```

Oh no! They sent me the wrong data

```
# now the data becomes
lobsters <- read_xlsx(here("lobsters2.xlsx"),skip=4)
# Then the same procedure follows
```

mutate()

###There are a lot of times where you don't want to summarize your data, but you do want to operate beyond the original daa.This is often done b y adding a column.

```
lobsters %>% mutate(size_m=size_mm/1000)
```

```
## # A tibble: 6,366 x 8
##     year month date    site  transect replicate size_mm size_m
##    <dbl> <dbl> <chr>   <chr>    <dbl> <chr>       <dbl>  <dbl>
##  1  2012     8 8/20/12 ivee         3 A              70  0.07
##  2  2012     8 8/20/12 ivee         3 B              60  0.06
##  3  2012     8 8/20/12 ivee         3 B              65  0.065
##  4  2012     8 8/20/12 ivee         3 B              70  0.07
##  5  2012     8 8/20/12 ivee         3 B              85  0.085
##  6  2012     8 8/20/12 ivee         3 C              60  0.06
```

```
##  7  2012     8 8/20/12 ivee         3 C              65  0.065
##  8  2012     8 8/20/12 ivee         3 C              67  0.067
##  9  2012     8 8/20/12 ivee         3 D              70  0.07
## 10  2012     8 8/20/12 ivee         4 B              85  0.085
## # i 6,356 more rows
```

Adding a column that has the same value repeated

```
lobsters_detailed <- lobsters %>% mutate(size_m=size_mm/1000,millenia=2000,observer="Allison Horst")
```

select()

```
lobsters_detailed %>% select(date,site,size_m)
```

```
## # A tibble: 6,366 x 3
##    date    site  size_m
##    <chr>   <chr>  <dbl>
##  1 8/20/12 ivee   0.07
##  2 8/20/12 ivee   0.06
##  3 8/20/12 ivee   0.065
##  4 8/20/12 ivee   0.07
##  5 8/20/12 ivee   0.085
##  6 8/20/12 ivee   0.06
##  7 8/20/12 ivee   0.065
##  8 8/20/12 ivee   0.067
##  9 8/20/12 ivee   0.07
## 10 8/20/12 ivee   0.085
## # i 6,356 more rows
```