Assignment 4: Data Wrangling (Fall 2024)

Cammie Moore

OVERVIEW

This exercise accompanies the lessons in Environmental Data Analytics on Data Wrangling

Directions

1. Rename this file `<FirstLast>_A04_DataWrangling.Rmd` (replacing `<FirstLast>` with your first and last name).
2. Change "Student Name" on line 3 (above) with your name.
3. Work through the steps, **creating code and output** that fulfill each instruction.
4. Be sure to **answer the questions** in this assignment document.
5. When you have completed the assignment, **Knit** the text and code into a single PDF file.
6. Ensure that code in code chunks does not extend off the page in the PDF.

Set up your session

1a. Load the `tidyverse`, `lubridate`, and `here` packages into your session.

1b. Check your working directory.

1c. Read in all four raw data files associated with the EPA Air dataset, being sure to set string columns to be read in a factors. See the README file for the EPA air datasets for more information (especially if you have not worked with air quality data previously).

2. Add the appropriate code to reveal the dimensions of the four datasets.

```
#1a
library(tidyverse)
library(lubridate)
library(here)
```

```
## Warning: package 'here' was built under R version 4.4.2
```

```
#1b
getwd()
```

```
## [1] "C:/Users/cammi/OneDrive/Documents/EDE_Spring2025"
```

```
here()
```

```
## [1] "C:/Users/cammi/OneDrive/Documents/EDE_Spring2025"
```

```
#1c
NC2018 <- read.csv(here('Data/Raw/EPAair_O3_NC2018_raw.csv'),
                   stringsAsFactors = TRUE)
NC2019 <- read.csv(here('Data/Raw/EPAair_O3_NC2019_raw.csv'),
                   stringsAsFactors = TRUE)
pm25_2018 <- read.csv(here('Data/Raw/EPAair_PM25_NC2018_raw.csv'),
                      stringsAsFactors = TRUE)
pm25_2019 <- read.csv(here('Data/Raw/EPAair_PM25_NC2019_raw.csv'),
                      stringsAsFactors = TRUE)

#2
str(NC2018)
```

```
## 'data.frame':    9737 obs. of  20 variables:
##  $ Date                                : Factor w/ 364 levels "01/01/2018","01/02/2018",..: 60 61 62
##  $ Source                              : Factor w/ 1 level "AQS": 1 1 1 1 1 1 1 1 1 1 ...
##  $ Site.ID                             : int  370030005 370030005 370030005 370030005 370030005 3700
##  $ POC                                 : int  1 1 1 1 1 1 1 1 1 1 ...
##  $ Daily.Max.8.hour.Ozone.Concentration: num  0.043 0.046 0.047 0.049 0.047 0.03 0.036 0.044 0.049 0
##  $ UNITS                               : Factor w/ 1 level "ppm": 1 1 1 1 1 1 1 1 1 1 ...
##  $ DAILY_AQI_VALUE                     : int  40 43 44 45 44 28 33 41 45 40 ...
##  $ Site.Name                           : Factor w/ 40 levels "","Beaufort",..: 35 35 35 35 35 35 35 3
##  $ DAILY_OBS_COUNT                     : int  17 17 17 17 17 17 17 17 17 17 ...
##  $ PERCENT_COMPLETE                    : num  100 100 100 100 100 100 100 100 100 100 ...
##  $ AQS_PARAMETER_CODE                  : int  44201 44201 44201 44201 44201 44201 44201 44201 44201 4
##  $ AQS_PARAMETER_DESC                  : Factor w/ 1 level "Ozone": 1 1 1 1 1 1 1 1 1 1 ...
##  $ CBSA_CODE                           : int  25860 25860 25860 25860 25860 25860 25860 25860 25860 2
##  $ CBSA_NAME                           : Factor w/ 17 levels "","Asheville, NC",..: 9 9 9 9 9 9 9 9 9
##  $ STATE_CODE                          : int  37 37 37 37 37 37 37 37 37 37 ...
##  $ STATE                               : Factor w/ 1 level "North Carolina": 1 1 1 1 1 1 1 1 1 1 ...
##  $ COUNTY_CODE                         : int  3 3 3 3 3 3 3 3 3 3 ...
##  $ COUNTY                              : Factor w/ 32 levels "Alexander","Avery",..: 1 1 1 1 1 1 1 1
##  $ SITE_LATITUDE                       : num  35.9 35.9 35.9 35.9 35.9 ...
##  $ SITE_LONGITUDE                      : num  -81.2 -81.2 -81.2 -81.2 -81.2 ...
```

```
str(NC2019)
```

```
## 'data.frame':    10592 obs. of  20 variables:
##  $ Date                                : Factor w/ 365 levels "01/01/2019","01/02/2019",..: 1 2 3 4 5
##  $ Source                              : Factor w/ 2 levels "AirNow","AQS": 1 1 1 1 1 1 1 1 1 1 ...
##  $ Site.ID                             : int  370030005 370030005 370030005 370030005 370030005 3700
##  $ POC                                 : int  1 1 1 1 1 1 1 1 1 1 ...
##  $ Daily.Max.8.hour.Ozone.Concentration: num  0.029 0.018 0.016 0.022 0.037 0.037 0.029 0.038 0.038 0
##  $ UNITS                               : Factor w/ 1 level "ppm": 1 1 1 1 1 1 1 1 1 1 ...
##  $ DAILY_AQI_VALUE                     : int  27 17 15 20 34 34 27 35 35 28 ...
##  $ Site.Name                           : Factor w/ 38 levels "","Beaufort",..: 33 33 33 33 33 33 33 3
##  $ DAILY_OBS_COUNT                     : int  24 24 24 24 24 24 24 24 24 24 ...
##  $ PERCENT_COMPLETE                    : num  100 100 100 100 100 100 100 100 100 100 ...
##  $ AQS_PARAMETER_CODE                  : int  44201 44201 44201 44201 44201 44201 44201 44201 44201 4
##  $ AQS_PARAMETER_DESC                  : Factor w/ 1 level "Ozone": 1 1 1 1 1 1 1 1 1 1 ...
##  $ CBSA_CODE                           : int  25860 25860 25860 25860 25860 25860 25860 25860 25860 2
##  $ CBSA_NAME                           : Factor w/ 15 levels "","Asheville, NC",..: 8 8 8 8 8 8 8 8 8
##  $ STATE_CODE                          : int  37 37 37 37 37 37 37 37 37 37 ...
```

```
##  $ STATE                             : Factor w/ 1 level "North Carolina": 1 1 1 1 1 1 1 1 1 1 ...
##  $ COUNTY_CODE                       : int  3 3 3 3 3 3 3 3 3 3 ...
##  $ COUNTY                            : Factor w/ 30 levels "Alexander","Avery",..: 1 1 1 1 1 1 1 1
##  $ SITE_LATITUDE                     : num  35.9 35.9 35.9 35.9 35.9 ...
##  $ SITE_LONGITUDE                    : num  -81.2 -81.2 -81.2 -81.2 -81.2 ...
```

```
str(pm25_2018)
```

```
## 'data.frame':    8983 obs. of  20 variables:
##  $ Date                          : Factor w/ 365 levels "01/01/2018","01/02/2018",..: 2 5 8 11 14 17
##  $ Source                        : Factor w/ 1 level "AQS": 1 1 1 1 1 1 1 1 1 1 ...
##  $ Site.ID                       : int  370110002 370110002 370110002 370110002 370110002 370110002 3
##  $ POC                           : int  1 1 1 1 1 1 1 1 1 1 ...
##  $ Daily.Mean.PM2.5.Concentration: num  2.9 3.7 5.3 0.8 2.5 4.5 1.8 2.5 4.2 1.7 ...
##  $ UNITS                         : Factor w/ 1 level "ug/m3 LC": 1 1 1 1 1 1 1 1 1 1 ...
##  $ DAILY_AQI_VALUE               : int  12 15 22 3 10 19 8 10 18 7 ...
##  $ Site.Name                     : Factor w/ 25 levels "","Blackstone",..: 15 15 15 15 15 15 15 15 15
##  $ DAILY_OBS_COUNT               : int  1 1 1 1 1 1 1 1 1 1 ...
##  $ PERCENT_COMPLETE              : num  100 100 100 100 100 100 100 100 100 100 ...
##  $ AQS_PARAMETER_CODE            : int  88502 88502 88502 88502 88502 88502 88502 88502 88502 88502
##  $ AQS_PARAMETER_DESC            : Factor w/ 2 levels "Acceptable PM2.5 AQI & Speciation Mass",..: 1
##  $ CBSA_CODE                     : int  NA NA NA NA NA NA NA NA NA NA ...
##  $ CBSA_NAME                     : Factor w/ 14 levels "","Asheville, NC",..: 1 1 1 1 1 1 1 1 1 1 ..
##  $ STATE_CODE                    : int  37 37 37 37 37 37 37 37 37 37 ...
##  $ STATE                         : Factor w/ 1 level "North Carolina": 1 1 1 1 1 1 1 1 1 1 ...
##  $ COUNTY_CODE                   : int  11 11 11 11 11 11 11 11 11 11 ...
##  $ COUNTY                        : Factor w/ 21 levels "Avery","Buncombe",..: 1 1 1 1 1 1 1 1 1 1 ..
##  $ SITE_LATITUDE                 : num  36 36 36 36 36 ...
##  $ SITE_LONGITUDE                : num  -81.9 -81.9 -81.9 -81.9 -81.9 ...
```

```
str(pm25_2019)
```

```
## 'data.frame':    8581 obs. of  20 variables:
##  $ Date                          : Factor w/ 365 levels "01/01/2019","01/02/2019",..: 3 6 9 12 15 18
##  $ Source                        : Factor w/ 2 levels "AirNow","AQS": 2 2 2 2 2 2 2 2 2 2 ...
##  $ Site.ID                       : int  370110002 370110002 370110002 370110002 370110002 370110002 3
##  $ POC                           : int  1 1 1 1 1 1 1 1 1 1 ...
##  $ Daily.Mean.PM2.5.Concentration: num  1.6 1 1.3 6.3 2.6 1.2 1.5 1.5 3.7 1.6 ...
##  $ UNITS                         : Factor w/ 1 level "ug/m3 LC": 1 1 1 1 1 1 1 1 1 1 ...
##  $ DAILY_AQI_VALUE               : int  7 4 5 26 11 5 6 6 15 7 ...
##  $ Site.Name                     : Factor w/ 25 levels "","Board Of Ed. Bldg.",..: 14 14 14 14 14 14
##  $ DAILY_OBS_COUNT               : int  1 1 1 1 1 1 1 1 1 1 ...
##  $ PERCENT_COMPLETE              : num  100 100 100 100 100 100 100 100 100 100 ...
##  $ AQS_PARAMETER_CODE            : int  88502 88502 88502 88502 88502 88502 88502 88502 88502 88502
##  $ AQS_PARAMETER_DESC            : Factor w/ 2 levels "Acceptable PM2.5 AQI & Speciation Mass",..: 1
##  $ CBSA_CODE                     : int  NA NA NA NA NA NA NA NA NA NA ...
##  $ CBSA_NAME                     : Factor w/ 14 levels "","Asheville, NC",..: 1 1 1 1 1 1 1 1 1 1 ..
##  $ STATE_CODE                    : int  37 37 37 37 37 37 37 37 37 37 ...
##  $ STATE                         : Factor w/ 1 level "North Carolina": 1 1 1 1 1 1 1 1 1 1 ...
##  $ COUNTY_CODE                   : int  11 11 11 11 11 11 11 11 11 11 ...
##  $ COUNTY                        : Factor w/ 21 levels "Avery","Buncombe",..: 1 1 1 1 1 1 1 1 1 1 ..
##  $ SITE_LATITUDE                 : num  36 36 36 36 36 ...
##  $ SITE_LONGITUDE                : num  -81.9 -81.9 -81.9 -81.9 -81.9 ...
```

All four datasets should have the same number of columns but unique record counts (rows). Do your datasets follow this pattern? Yes.

Wrangle individual datasets to create processed files.

3. Change the Date columns to be date objects.
4. Select the following columns: Date, DAILY_AQI_VALUE, Site.Name, AQS_PARAMETER_DESC, COUNTY, SITE_LATITUDE, SITE_LONGITUDE
5. For the PM2.5 datasets, fill all cells in AQS_PARAMETER_DESC with "PM2.5" (all cells in this column should be identical).
6. Save all four processed datasets in the Processed folder. Use the same file names as the raw files but replace "raw" with "processed".

```
#3
NC2018 <- NC2018 |>
  mutate(Date = mdy(Date))
NC2019 <- NC2019 |>
  mutate(Date = mdy(Date))
pm25_2018 <- pm25_2018 |>
  mutate(Date = mdy(Date))
pm25_2019 <- pm25_2019 |>
  mutate(Date = mdy(Date))

#4
NC2018 <- NC2018 |>
  select(Date, DAILY_AQI_VALUE, Site.Name, AQS_PARAMETER_DESC, COUNTY, SITE_LATITUDE, SITE_LONGITUDE)
NC2019 <- NC2019 |>
  select(Date, DAILY_AQI_VALUE, Site.Name, AQS_PARAMETER_DESC, COUNTY, SITE_LATITUDE, SITE_LONGITUDE)
pm25_2018 <- pm25_2018 |>
  select(Date, DAILY_AQI_VALUE, Site.Name, AQS_PARAMETER_DESC, COUNTY, SITE_LATITUDE, SITE_LONGITUDE)
pm25_2019 <- pm25_2019 |>
  select(Date, DAILY_AQI_VALUE, Site.Name, AQS_PARAMETER_DESC, COUNTY, SITE_LATITUDE, SITE_LONGITUDE)


#5
pm25_2018 <- pm25_2018 |>
  mutate(AQS_PARAMETER_DESC = "PM2.5")
pm25_2019 <- pm25_2019 |>
  mutate(AQS_PARAMETER_DESC = "PM2.5")

#6
write.csv(NC2018, file = "./Data/Processed/EPAair_O3_NC2018_processed")
write.csv(NC2019, file = "./Data/Processed/EPAair_O3_NC2019_processed")
write.csv(pm25_2018, file = "./Data/Processed/EPAair_PM25_NC2018_processed")
write.csv(pm25_2019, file = "./Data/Processed/EPAair_PM25_NC2019_processed")
```

Combine datasets

7. Combine the four datasets with `rbind`. Make sure your column names are identical prior to running this code.

8. Wrangle your new dataset with a pipe function (%>%) so that it fills the following conditions:

- Include only sites that the four data frames have in common:

"Linville Falls", "Durham Armory", "Leggett", "Hattie Avenue",
"Clemmons Middle", "Mendenhall School", "Frying Pan Mountain", "West Johnston Co.", "Garinger High School", "Castle Hayne", "Pitt Agri. Center", "Bryson City", "Millbrook School"

(the function `intersect` can figure out common factor levels - but it will include sites with missing site information, which you don't want...)

- Some sites have multiple measurements per day. Use the split-apply-combine strategy to generate daily means: group by date, site name, AQS parameter, and county. Take the mean of the AQI value, latitude, and longitude.
- Add columns for "Month" and "Year" by parsing your "Date" column (hint: `lubridate` package)
- Hint: the dimensions of this dataset should be 14,752 x 9.

9. Spread your datasets such that AQI values for ozone and PM2.5 are in separate columns. Each location on a specific date should now occupy only one row.
10. Call up the dimensions of your new tidy dataset.
11. Save your processed dataset with the following file name: "EPAair_O3_PM25_NC1819_Processed.csv"

```
#7
epa_air <- rbind(NC2018, NC2019, pm25_2018, pm25_2019)

#8
epa_air <- epa_air |>
  filter(Site.Name %in% c("Linville Falls", "Durham Armory", "Leggett",
                          "Hattie Avenue", "Clemmons Middle", "Mendenhall School",
                          "Frying Pan Mountain", "West Johnston Co.", "Garinger High School",
                          "Castle Hayne", "Pitt Agri. Center", "Bryson City", "Millbrook School")) |>
  group_by(Date, Site.Name, AQS_PARAMETER_DESC, COUNTY) |> #ask a question about this because i am conf
  summarise(
    mean_AQI = mean(DAILY_AQI_VALUE, na.rm = TRUE),
    mean_latitude = mean(SITE_LATITUDE, na.rm = TRUE),
    mean_longitude = mean(SITE_LONGITUDE, na.rm = TRUE)
  )|>
  mutate(month = month(Date),
         year = year(Date))
```

```
## `summarise()` has grouped output by 'Date', 'Site.Name', 'AQS_PARAMETER_DESC'.
## You can override using the `.groups` argument.
```

```
#9
epa_air <- epa_air |>
  pivot_wider(names_from = AQS_PARAMETER_DESC , values_from = mean_AQI)

#10
str(epa_air)
```

```
## gropd_df [8,976 x 9] (S3: grouped_df/tbl_df/tbl/data.frame)
##  $ Date          : Date[1:8976], format: "2018-01-01" "2018-01-01" ...
##  $ Site.Name     : Factor w/ 51 levels "","Beaufort",..: 6 10 12 16 18 19 23 28 32 40 ...
##  $ COUNTY        : Factor w/ 37 levels "Alexander","Avery",..: 29 24 10 8 22 10 9 31 26 16 ...
##  $ mean_latitude : num [1:8976] 35.4 34.4 36 36 35.2 ...
##  $ mean_longitude: num [1:8976] -83.4 -77.8 -80.3 -78.9 -80.8 ...
##  $ month         : num [1:8976] 1 1 1 1 1 1 1 1 1 ...
##  $ year          : num [1:8976] 2018 2018 2018 2018 2018 ...
##  $ PM2.5         : num [1:8976] 35 13 24 31 20 22 14 28 15 24 ...
##  $ Ozone         : num [1:8976] NA NA NA NA 32 NA NA 34 NA NA ...
##  - attr(*, "groups")= tibble [8,976 x 3] (S3: tbl_df/tbl/data.frame)
##   ..$ Date     : Date[1:8976], format: "2018-01-01" "2018-01-01" ...
##   ..$ Site.Name: Factor w/ 51 levels "","Beaufort",..: 6 10 12 16 18 19 23 28 32 40 ...
##   ..$ .rows    : list<int> [1:8976]
##   .. ..$ : int 1
##   .. ..$ : int 2
##   .. ..$ : int 3
##   .. ..$ : int 4
##   .. ..$ : int 5
##   .. ..$ : int 6
##   .. ..$ : int 7
##   .. ..$ : int 8
##   .. ..$ : int 9
##   .. ..$ : int 10
##   .. ..$ : int 11
##   .. ..$ : int 12
##   .. ..$ : int 13
##   .. ..$ : int 14
##   .. ..$ : int 15
##   .. ..$ : int 16
##   .. ..$ : int 17
##   .. ..$ : int 18
##   .. ..$ : int 19
##   .. ..$ : int 20
##   .. ..$ : int 21
##   .. ..$ : int 22
##   .. ..$ : int 23
##   .. ..$ : int 24
##   .. ..$ : int 25
##   .. ..$ : int 26
##   .. ..$ : int 27
##   .. ..$ : int 28
##   .. ..$ : int 29
##   .. ..$ : int 30
##   .. ..$ : int 31
##   .. ..$ : int 32
##   .. ..$ : int 33
##   .. ..$ : int 34
##   .. ..$ : int 35
##   .. ..$ : int 36
##   .. ..$ : int 37
##   .. ..$ : int 38
##   .. ..$ : int 39
##   .. ..$ : int 40
```

```
##   .. ..$ : int 41
##   .. ..$ : int 42
##   .. ..$ : int 43
##   .. ..$ : int 44
##   .. ..$ : int 45
##   .. ..$ : int 46
##   .. ..$ : int 47
##   .. ..$ : int 48
##   .. ..$ : int 49
##   .. ..$ : int 50
##   .. ..$ : int 51
##   .. ..$ : int 52
##   .. ..$ : int 53
##   .. ..$ : int 54
##   .. ..$ : int 55
##   .. ..$ : int 56
##   .. ..$ : int 57
##   .. ..$ : int 58
##   .. ..$ : int 59
##   .. ..$ : int 60
##   .. ..$ : int 61
##   .. ..$ : int 62
##   .. ..$ : int 63
##   .. ..$ : int 64
##   .. ..$ : int 65
##   .. ..$ : int 66
##   .. ..$ : int 67
##   .. ..$ : int 68
##   .. ..$ : int 69
##   .. ..$ : int 70
##   .. ..$ : int 71
##   .. ..$ : int 72
##   .. ..$ : int 73
##   .. ..$ : int 74
##   .. ..$ : int 75
##   .. ..$ : int 76
##   .. ..$ : int 77
##   .. ..$ : int 78
##   .. ..$ : int 79
##   .. ..$ : int 80
##   .. ..$ : int 81
##   .. ..$ : int 82
##   .. ..$ : int 83
##   .. ..$ : int 84
##   .. ..$ : int 85
##   .. ..$ : int 86
##   .. ..$ : int 87
##   .. ..$ : int 88
##   .. ..$ : int 89
##   .. ..$ : int 90
##   .. ..$ : int 91
##   .. ..$ : int 92
##   .. ..$ : int 93
##   .. ..$ : int 94
```

```
##   .. ..$ : int 95
##   .. ..$ : int 96
##   .. ..$ : int 97
##   .. ..$ : int 98
##   .. ..$ : int 99
##   .. .. [list output truncated]
##   .. ..@ ptype: int(0)
##   ..- attr(*, ".drop")= logi TRUE
```

```
#11
write.csv(epa_air, file = "./Data/Processed/EPAair_O3_PM25_NC1819_Processed.csv")
```

Generate summary tables

12. Use the split-apply-combine strategy to generate a summary data frame. Data should be grouped by site, month, and year. Generate the mean AQI values for ozone and PM2.5 for each group. Then, add a pipe to remove instances where mean **ozone** values are not available (use the function `drop_na` in your pipe). It's ok to have missing mean PM2.5 values in this result.

13. Call up the dimensions of the summary dataset.

```
#12
summary_epa <- epa_air |>
  group_by(Site.Name, month, year) |>
  summarise(
    mean_ozone = mean(Ozone),
    mean_PM25 = mean(PM2.5))|>
  drop_na(mean_ozone)|>
  drop_na(mean_PM25)
```

```
## `summarise()` has grouped output by 'Site.Name', 'month'. You can override
## using the `.groups` argument.
```

```
#13
str(summary_epa)
```

```
## gropd_df [101 x 5] (S3: grouped_df/tbl_df/tbl/data.frame)
##  $ Site.Name : Factor w/ 51 levels "","Beaufort",..: 6 6 6 6 6 6 10 10 10 10 ...
##  $ month     : num [1:101] 3 4 4 7 9 10 4 4 5 7 ...
##  $ year      : num [1:101] 2018 2018 2019 2019 2018 ...
##  $ mean_ozone: num [1:101] 41.6 44.5 45.4 30.4 25.4 ...
##  $ mean_PM25 : num [1:101] 34.7 28.2 26.7 33.6 25.1 ...
##  - attr(*, "groups")= tibble [74 x 3] (S3: tbl_df/tbl/data.frame)
##   ..$ Site.Name: Factor w/ 51 levels "","Beaufort",..: 6 6 6 6 6 10 10 10 10 10 ...
##   ..$ month    : num [1:74] 3 4 7 9 10 4 5 7 8 10 ...
##   ..$ .rows    : list<int> [1:74]
##   .. ..$ : int 1
##   .. ..$ : int [1:2] 2 3
##   .. ..$ : int 4
##   .. ..$ : int 5
##   .. ..$ : int 6
##   .. ..$ : int [1:2] 7 8
```

```
##   .. ..$ : int 9
##   .. ..$ : int 10
##   .. ..$ : int 11
##   .. ..$ : int 12
##   .. ..$ : int 13
##   .. ..$ : int [1:2] 14 15
##   .. ..$ : int 16
##   .. ..$ : int [1:2] 17 18
##   .. ..$ : int 19
##   .. ..$ : int [1:2] 20 21
##   .. ..$ : int [1:2] 22 23
##   .. ..$ : int 24
##   .. ..$ : int 25
##   .. ..$ : int 26
##   .. ..$ : int 27
##   .. ..$ : int 28
##   .. ..$ : int 29
##   .. ..$ : int 30
##   .. ..$ : int [1:2] 31 32
##   .. ..$ : int 33
##   .. ..$ : int 34
##   .. ..$ : int [1:2] 35 36
##   .. ..$ : int [1:2] 37 38
##   .. ..$ : int 39
##   .. ..$ : int [1:2] 40 41
##   .. ..$ : int [1:2] 42 43
##   .. ..$ : int [1:2] 44 45
##   .. ..$ : int 46
##   .. ..$ : int 47
##   .. ..$ : int 48
##   .. ..$ : int [1:2] 49 50
##   .. ..$ : int [1:2] 51 52
##   .. ..$ : int 53
##   .. ..$ : int 54
##   .. ..$ : int 55
##   .. ..$ : int 56
##   .. ..$ : int 57
##   .. ..$ : int 58
##   .. ..$ : int 59
##   .. ..$ : int 60
##   .. ..$ : int 61
##   .. ..$ : int [1:2] 62 63
##   .. ..$ : int 64
##   .. ..$ : int 65
##   .. ..$ : int 66
##   .. ..$ : int 67
##   .. ..$ : int [1:2] 68 69
##   .. ..$ : int 70
##   .. ..$ : int [1:2] 71 72
##   .. ..$ : int [1:2] 73 74
##   .. ..$ : int [1:2] 75 76
##   .. ..$ : int [1:2] 77 78
##   .. ..$ : int [1:2] 79 80
##   .. ..$ : int 81
```

```
##   .. ..$ : int [1:2] 82 83
##   .. ..$ : int 84
##   .. ..$ : int 85
##   .. ..$ : int 86
##   .. ..$ : int [1:2] 87 88
##   .. ..$ : int [1:2] 89 90
##   .. ..$ : int 91
##   .. ..$ : int 92
##   .. ..$ : int [1:2] 93 94
##   .. ..$ : int 95
##   .. ..$ : int [1:2] 96 97
##   .. ..$ : int [1:2] 98 99
##   .. ..$ : int 100
##   .. ..$ : int 101
##   .. ..@ ptype: int(0)
##   ..- attr(*, ".drop")= logi TRUE
```

14. Why did we use the function `drop_na` rather than `na.omit`? Hint: replace `drop_na` with `na.omit` in part 12 and observe what happens with the dimensions of the summary date frame.

    Answer: The dimensions with na.omit is three times the size of drop_na. I am not exactly sure of the difference.