Working with Data Frames - Notes

Key

2025-12-26

! How to use the Notes files

This notes file follows along with the lesson materials. Your job is to fill out the missing code, often denoted with ________, and to complete the "you try it" type problems presented in the notes.
Render this document regularly, ensuring that it is complete before you submit. Before you submit, remove the `error: true` line in the YAML header then render.

Setup

```
library(ggplot2)                # replace the blank line with `ggplot2`
library(gtsummary)
ncbirths <- openintro::ncbirths
```

Frequency Tables

You can create a basic frequency table by using the `table()` function.

```
table(ncbirths$lowbirthweight)
```

```

    low not low 
    111     889 
```

Relative frequencies (proportions or percentages) are calculated by putting the results of the `table` function inside the `prop_table` function.

Characteristic	**N = 1,000**[1]
visits	12 (10, 15)
Unknown	9
lowbirthweight	
low	111 (11%)
not low	889 (89%)

[1] Median (Q1, Q3); n (%)

```r
prop.table(
  table(ncbirths$lowbirthweight)
)
```

```

    low not low 
  0.111   0.889 
```

Summary Statistics

The function `summary()` prints out the five number summary, and includes the mean. This function also displays the number of missing values for that variable.

```r
summary(ncbirths$visits)
```

```
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.    NA's 
    0.0    10.0    12.0    12.1    15.0    30.0       9 
```

Fancy summary tables

The `gtsummary` package provides a single function `tbl_summary` to create a really nicely formatted summary table for both quantitative and categorical data types.

```r
tbl_summary(ncbirths,
            include = c(visits, lowbirthweight)
            )
```

The `statistic` argument can be used to change what values are displayed.

Characteristic	N = 1,000[1]
visits	12 (4)
Unknown	9
lowbirthweight	
low	111 / 1,000 (11%)
not low	889 / 1,000 (89%)

[1] Mean (SD); n / N (%)

```
tbl_summary(ncbirths,
            include = c(visits, lowbirthweight),
            statistic = list(
                all_continuous() ~ "{mean} ({sd})",
                all_categorical() ~ "{n} / {N} ({p}%)"
              )
            )
```

Missing Data

We can see 4 out of the first 6 values for the variable `fage` (fathers age) in the `ncbirths` data set are missing.

```
head(ncbirths$fage)
```

```
[1] NA NA 19 21 NA NA
```

Problem 1: `R` can't do arithmetic on missing data.

```
mean(ncbirths$fage)
```

```
[1] NA
```

> Fix this error
>
> Add the argument `na.rm=TRUE` to the `mean()` function inside the right hand parenthesis to calculate the mean after excluding missing values.

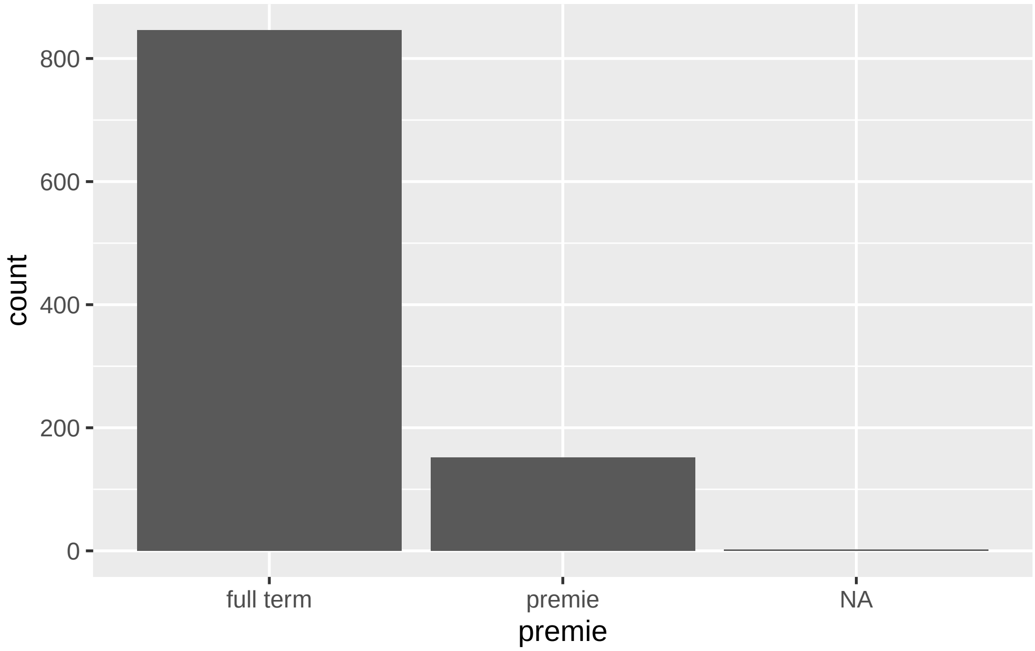
```
mean(ncbirths$fage, na.rm = TRUE)
```

```
[1] 30.25573
```

Problem 2: Some plots will show `NA` as it's own category

Sometimes this is fine, other times this is undesirable. We'll see later how we can adjust this plot to remove that column of NA.

```
ggplot(ncbirths, aes(premie)) + geom_bar()
```



Identifying missing values

To find out how many values in a particular variable are missing we can use several different approaches.

Look at the raw data

```
head(ncbirths)
```

```
# A tibble: 6 x 13
   fage  mage mature    weeks premie visits marital gained weight lowbirthweight
  <int> <int> <fct>     <int> <fct>   <int> <fct>    <int>  <dbl> <fct>
1    NA    13 younger ~    39 full ~     10 not ma~     38   7.63 not low
2    NA    14 younger ~    42 full ~     15 not ma~     20   7.88 not low
3    19    15 younger ~    37 full ~     11 not ma~     38   6.63 not low
4    21    15 younger ~    41 full ~      6 not ma~     34   8    not low
5    NA    15 younger ~    39 full ~      9 not ma~     27   6.38 not low
6    NA    15 younger ~    38 full ~     19 not ma~     22   5.38 low
# i 3 more variables: gender <fct>, habit <fct>, whitemom <fct>
```

Look at data summaries

Functions such as `table()` have a `useNA="always"` option to show how many records have missing values, and `summary()` will always show a column for NA.

```
table(ncbirths$habit, useNA="always")
```

```

nonsmoker    smoker      <NA>
      873       126         1
```

```
summary(ncbirths$fage)
```

```
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.    NA's
  14.00   25.00   30.00   30.26   35.00   55.00     171
```

Use a logical statement

The function `is.na()` returns TRUE or FALSE for each element in the provided vector for whether or not that element is missing.

```
x <- c("green", NA, 3)
is.na(x)
```

```
[1] FALSE  TRUE FALSE
```

```
sum(is.na(ncbirths$fage))
```

```
[1] 171
```

There are 171 records in this data set where the age for the father is not present.

ℹ Negating `is.na()` to find the non-missing values

Sometimes you want to operate only only the non-missing values. Recall from **?@sec-logical** we can use the `!` to negate a boolean argument.

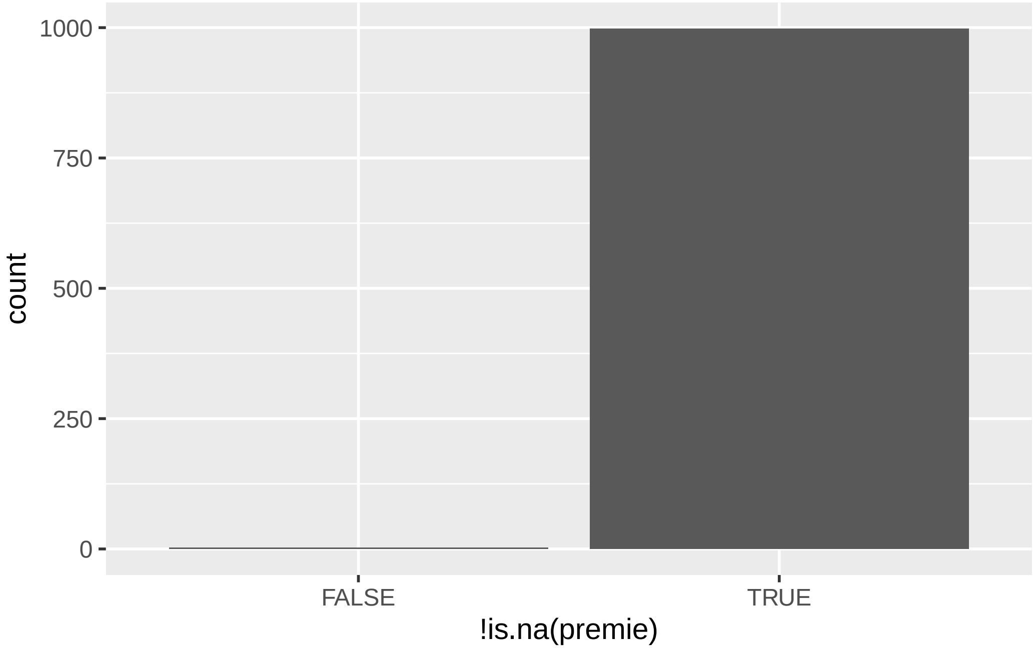
```r
!is.na(x)
```

```
[1]  TRUE FALSE  TRUE
```

Hide the NA bar

We can use this tactic to fix that barchart from above. Wrap `!is.na()` around the `premie` variable in the `ggplot` code to create a barchart that does not have a bar for missing values.

```r
ggplot(ncbirths, aes(!is.na(premie))) + geom_bar()
```



Data management

Overwrite existing values

Example 1: Too low birthweight

Let's look at the numerical distribution of birthweight (in pounds) of the baby.

```
summary(ncbirths$weight)
```

```
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
  1.000   6.380   7.310   7.101   8.060  11.750 
```

The value of 1 lb seems very low. The researchers you are working with decide that is a mistake and should be excluded from the data. We would then set all records where `weight=1` to missing.

```
ncbirths$weight[ncbirths$weight==1] <- NA
```

```
min(ncbirths$weight, na.rm=TRUE)
```

```
[1] 1.19
```

Your Turn

But what about other weights that aren't quite as low as 1, but still unusually low?

- Write the code to set all birth weights less than 4 lbs (`<4`) to missing (NA).
- Then recalculate the mean to confirm your recode worked.

```
ncbirths$weight[ncbirths$weight < 4] <- NA
min(ncbirths$weight, na.rm=TRUE)
```

```
[1] 4
```

Creating new variables

The following code creates a new variable `wtgain_mom` the weight gained by the mother, that is not due to the baby by subtracting `weight` from `gained`.

```
ncbirths$wtgain_mom <- ncbirths$gained - ncbirths$weight
```

To confirm this variable was created correctly, we look at the data contained in three variables in question.

```
head(ncbirths[,c('gained', 'weight', 'wtgain_mom')])
```

```
# A tibble: 6 x 3
  gained weight wtgain_mom
   <int>  <dbl>      <dbl>
1     38   7.63       30.4
2     20   7.88       12.1
3     38   6.63       31.4
4     34   8          26
5     27   6.38       20.6
6     22   5.38       16.6
```

Dichtomizing data

Let's add a variable to identify if a mother in the North Carolina births data set was underage at the time of birth. Specifically Make a new variable `underage` on the `ncbirths` data set. If `mage` is under 18, then the value of this new variable is `underage`, else it is labeled as `adult`.

```
ncbirths$underage <- ifelse(ncbirths$mage < 18, "underage", "adult")
```

Trust but Verify

```
table(ncbirths$underage, useNA="always")
```

```

   adult underage     <NA>
     963       37        0
```

Next let's check it against the value of `mage` itself. Let's look at all rows where mothers age is either 17 or 18 `mage %in% c(17,18)`, and only the columns of interest.

```
ncbirths[ncbirths$mage %in% c(17,18),c('mage', 'underage')]
```

```
# A tibble: 57 x 2
    mage underage
   <int> <chr>
 1    17 underage
 2    17 underage
 3    17 underage
 4    17 underage
 5    17 underage
 6    17 underage
 7    17 underage
 8    17 underage
 9    17 underage
10    17 underage
# i 47 more rows
```

Chaining commands

Example: Frequency tables & summary statistics using the pipe

First stating the variable, then pipe in the summary function.

```
ncbirths$mature |> table() # instead of table(ncbirths$mature)
```

```

 mature mom younger mom
        133         867
```

```
ncbirths$mage |> mean() #instead of mean(ncbirths$mage)
```

```
[1] 27
```