Data Tidying

Jessica Ernakovich

1/16/2019

Loading packages

We need to call these packages when they are needed for a program, even though it's installed
It's good practice to include all your library calls in one chunk

```
library(dplyr)
library(tidyr)
library(plotrix) #julia told me about this and I didn't use it, but it can give you std err
```

this will give a warning message about certain packages because their names are redundant to this package and others. By default, RStudio will assume you want to call the function from the most recently loaded library.

If necessary, you can call from the other package using the package_name::function_name(...)

Data Cleaning

Read in the datafile

```
catch_df <- read.csv(url("https://knb.ecoinformatics.org/knb/d1/mn/v2/object/df35b.302.1", method = "li
                   stringsAsFactors = FALSE)
head(catch_df)
```

```
##   Region Year Chinook Sockeye Coho Pink Chum All notesRegCode
## 1    SSE 1886       0       5    0    0    0   5
## 2    SSE 1887       0     155    0    0    0 155
## 3    SSE 1888       0     224   16    0    0 240
## 4    SSE 1889       0     182   11   92    0 285
## 5    SSE 1890       0     251   42    0    0 292
## 6    SSE 1891       0     274   24    0    0 298
```

cmd + shift + M will result in %>%, the pipe operator (which allows operations to be linked)

- remove marginal sum and notes column
- move from wide to long format

```
catch_long <- catch_df %>%
  select(Region, Year, Chinook, Sockeye, Coho, Pink, Chum) %>%
  gather(key = "species", value = "catch", -Year, -Region)

#including function calls on new lines makes this cleaner, but is unnecessary
#could also have used "-" to just drop the columns we wanted to get rid of.

head(catch_long)
```

```
##   Region Year species catch
## 1    SSE 1886 Chinook     0
## 2    SSE 1887 Chinook     0
## 3    SSE 1888 Chinook     0
## 4    SSE 1889 Chinook     0
## 5    SSE 1890 Chinook     0
## 6    SSE 1891 Chinook     0
```

If <- reads as "gets" and %>% reads as "then", then the first two lines above read: catch_cleaned gets the catch_df then a select of the catch_df.

- erroneious value due to OCR issue - change "I" to one
- create "catch" column that multiplies by 1000 to get true numbers

```
catch_cleaned <- catch_long %>%
  rename(catch_thousands = catch) %>%
mutate(catch_thousands = ifelse(catch_thousands == "I", 1, catch_thousands)) %>%
mutate(catch_thousands = as.integer(catch_thousands))  %>%
mutate(catch = catch_thousands *1000)

tail (catch_cleaned) #tail is more meaningful than head in this case, because the head is all zero's
```

```
##      Region Year species catch_thousands  catch
## 8535    NOP 1992    Chum             342 342000
## 8536    NOP 1993    Chum             135 135000
## 8537    NOP 1994    Chum              84  84000
## 8538    NOP 1995    Chum              99  99000
## 8539    NOP 1996    Chum              68  68000
## 8540    NOP 1997    Chum              97  97000
```

There are some values in the catch_thousands column that are not integers, but rather text. So, we are trying to find and force to be integers. But, it didn't work. So then we looked for it with the "which" command. And then we visualized it. Next, we will change this to a number.

Split-Apply-Combine

Calculate total catch by region

```
catch_total <- catch_cleaned %>%
  group_by(species, Year) %>%
  summarize(catch_region = sum(catch),
            average = mean(catch),
            stderr = std.error(catch), #from plotrix package
            n_obs = n())

catch_total
```

```
## # A tibble: 600 x 6
## # Groups:   species [?]
##    species  Year catch_region average stderr n_obs
```

```
##    <chr>   <int>        <dbl>   <dbl>  <dbl> <int>
##  1 Chinook  1878            0       0     NA     1
##  2 Chinook  1879            0       0     NA     1
##  3 Chinook  1880            0       0     NA     1
##  4 Chinook  1881            0       0     NA     1
##  5 Chinook  1882            0       0      0     2
##  6 Chinook  1883            0       0      0     3
##  7 Chinook  1884            0       0      0     4
##  8 Chinook  1885            0       0      0     4
##  9 Chinook  1886            0       0      0     5
## 10 Chinook  1887            0       0      0     5
## # ... with 590 more rows
```

Filter for Chinook Salmon

```
#names(catch_cleaned)
catch_chinook <- catch_cleaned %>%
  filter(species == "Chinook" & Region == "SSE" & Year > 1990) %>%
# "|" is called a logical "or"
  arrange(-Year)

head(catch_chinook)
```

```
##   Region Year species catch_thousands catch
## 1    SSE 1997 Chinook              38 38000
## 2    SSE 1996 Chinook              24 24000
## 3    SSE 1995 Chinook              32 32000
## 4    SSE 1994 Chinook              56 56000
## 5    SSE 1993 Chinook              98 98000
## 6    SSE 1992 Chinook              88 88000
```

Joins

we will be using a left join to join the region definition to the catch data

using a left join will mean that the number of rows are defined by the left dataframe. The column numbers will be the columns from left + the column from right - the number of key columns.

```
region_defs <- read.csv(url("https://knb.ecoinformatics.org/knb/d1/mn/v2/object/df35b.303.1", method =
                        stringsAsFactors = FALSE)

head(region_defs)
```

```
##      code                                    mgmtArea areaClass regionCode
## 1     GSE              Unallocated Southeast Alaska  mgmtArea          1
## 2     NSE                 Northern Southeast Alaska  mgmtArea          1
## 3     SSE                 Southern Southeast Alaska  mgmtArea          1
## 4     YAK                                   Yakutat  mgmtArea          1
## 5 PWSmgmt      Prince William Sound Management Area  mgmtArea          2
## 6     BER Bering River Subarea Copper River Subarea   subarea          2
##
## 1                                                          Included are Southeastern Alaska catches
```

```
## 2 Northern Southern Alaska includes Districts 9 through 16 from summer straight northwest to and incl
## 3
## 4
## 5
## 6
```

Cleaning up a bit by keeping only the columns we want.

```
region_clean <- region_defs %>%
  select(code, mgmtArea)

head(region_clean)
```

```
##      code                                    mgmtArea
## 1     GSE              Unallocated Southeast Alaska
## 2     NSE                 Northern Southeast Alaska
## 3     SSE                 Southern Southeast Alaska
## 4     YAK                                  Yakutat
## 5 PWSmgmt      Prince William Sound Management Area
## 6     BER Bering River Subarea Copper River Subarea
```

Now it's time for the join.

```
catch_joined <- left_join(catch_cleaned, region_clean, #join syntax changes a bit
                          by = c("Region" = "code"))

head(catch_joined)
```

```
##   Region Year species catch_thousands catch                  mgmtArea
## 1    SSE 1886 Chinook               0     0 Southern Southeast Alaska
## 2    SSE 1887 Chinook               0     0 Southern Southeast Alaska
## 3    SSE 1888 Chinook               0     0 Southern Southeast Alaska
## 4    SSE 1889 Chinook               0     0 Southern Southeast Alaska
## 5    SSE 1890 Chinook               0     0 Southern Southeast Alaska
## 6    SSE 1891 Chinook               0     0 Southern Southeast Alaska
```

Spread

Long format to wide format for data display

```
catch_wide <- catch_cleaned %>%
  filter(Year>1990) %>%
    select(-catch_thousands) %>%
  spread(key = Year, value = catch)

head(catch_wide)
```

```
##   Region species 1991   1992 1993   1994 1995 1996 1997
## 1    ALU Chinook    0      0    0      0    0    0    0
## 2    ALU    Chum    0   2000 1000   1000    0    0    0
```

```
## 3    ALU    Coho    0      0    0      0    0    0    0
## 4    ALU    Pink    0 320000    0 860000    0    0    0
## 5    ALU Sockeye 1000   3000    0      0    0    0    0
## 6    BER Chinook    0      0    0      0    0    0    0
```

Seperate and unite

some fake data ISO date formate is: YYYY-MM-DD

```
dates_df <- data.frame(date = c("5/24/1930",
                                "5/25/1930",
                                "5/26/1930",
                                "5/27/1930",
                                "5/28/1930"),
                       stringsAsFactors = FALSE)

dates_df
```

```
##        date
## 1 5/24/1930
## 2 5/25/1930
## 3 5/26/1930
## 4 5/27/1930
## 5 5/28/1930
```

```
dates_sep <- dates_df %>%
  separate(col = date, into = c("month", "day", "year"), by = "/", remove = F)

head(dates_sep)
```

```
##        date month day year
## 1 5/24/1930     5  24 1930
## 2 5/25/1930     5  25 1930
## 3 5/26/1930     5  26 1930
## 4 5/27/1930     5  27 1930
## 5 5/28/1930     5  28 1930
```

```
dates_unite <- dates_sep %>%
  unite(date_iso, year, month, day, sep = "-")

head(dates_unite)
```

```
##        date  date_iso
## 1 5/24/1930 1930-5-24
## 2 5/25/1930 1930-5-25
## 3 5/26/1930 1930-5-26
## 4 5/27/1930 1930-5-27
## 5 5/28/1930 1930-5-28
```

```
#these don't look amazing because the dates and months are one rather than 2 digits. use "stringer" pac
```