Coding Assignment 7

Noah Foilb

10/13/2020

|  |
| --- |
| ## MAKE SURE YOUR CODE IS PROPERLY COMMENTED THROUGOUT! |

## Reminders

### Pivot\_longer()

* pivot\_longer() “lengthens” data, increasing the number of rows and decreasing the number of columns.
* format pivot\_longer(name of data frame, cols=name(s) of columns to pivot names\_to=name of new column created from the original column names values\_to=name of new column created from the values in the column(s) being pivoted values\_drop\_na=FALSE)

### Pivot\_wider()

* pivot\_wider() “widens” data, increasing the number of columns and decreasing the number of rows.
* format pivot\_wider(name of data frame, names\_from = column to use to get new column headers values\_from = column to use to fill in values for newly created columns)

#### 1. Load the tidyr and dplyr libraries

# Load Librarys  
library(dplyr)

##   
## Attaching package: 'dplyr'

## The following objects are masked from 'package:stats':  
##   
## filter, lag

## The following objects are masked from 'package:base':  
##   
## intersect, setdiff, setequal, union

library(tidyr)

#### 2. We will be using the data set called USArrests.

* Ask for help on this data set so that you can read about it,
* get the dimensions of the data set (rows and columns), and
* print out the first 6 rows.

# Help for USArrest  
help("USArrests")

## starting httpd help server ... done

# Structure which gives dimensions  
str(USArrests)

## 'data.frame': 50 obs. of 4 variables:  
## $ Murder : num 13.2 10 8.1 8.8 9 7.9 3.3 5.9 15.4 17.4 ...  
## $ Assault : int 236 263 294 190 276 204 110 238 335 211 ...  
## $ UrbanPop: int 58 48 80 50 91 78 77 72 80 60 ...  
## $ Rape : num 21.2 44.5 31 19.5 40.6 38.7 11.1 15.8 31.9 25.8 ...

# Display first six values  
head(USArrests,6)

## Murder Assault UrbanPop Rape  
## Alabama 13.2 236 58 21.2  
## Alaska 10.0 263 48 44.5  
## Arizona 8.1 294 80 31.0  
## Arkansas 8.8 190 50 19.5  
## California 9.0 276 91 40.6  
## Colorado 7.9 204 78 38.7

#### 3. The column UrbanPop displays the percentage of the population in each state that live in an urban area. The other three columns present number of crimes in each category per 100,000 people. Create a new data frame called USCrime that omits the UrbanPop column and keeps the three crime rate columns. Then display the dimension of this new data frame (it should have 50 rows and 3 columns) and diplay first 6 rows of this new data frame.

# Assign variable  
USCrime <- USArrests %>%  
   
 # Select the columns we want  
 select(Murder,Assault,Rape)  
  
# Find Dimensions  
dim(USCrime)

## [1] 50 3

# Display first six values  
head(USCrime)

## Murder Assault Rape  
## Alabama 13.2 236 21.2  
## Alaska 10.0 263 44.5  
## Arizona 8.1 294 31.0  
## Arkansas 8.8 190 19.5  
## California 9.0 276 40.6  
## Colorado 7.9 204 38.7

#### 4. Put USCrime into long format. Name the new data set USCrime\_long. Display the first 10 rows of USCrime\_long and display the total number of rows using the nrow() command.

* The three crimes (murder, assault, and rape) should end up in one column. Name this column “crime”.
* The arrest rates (the numerical values) should end up in a second column. Name this column “number\_per\_100000”.
* If done correctly, Murder will be in the top row with a rate of 13.2 and there will be 150 rows.

# Use Pivot Longer to get new data  
USCrime\_long <- pivot\_longer(USCrime,cols = Murder:Rape, names\_to = "crime", values\_to = "number\_per\_100000")  
  
# Display data  
head(USCrime\_long,10)

## # A tibble: 10 x 2  
## crime number\_per\_100000  
## <chr> <dbl>  
## 1 Murder 13.2  
## 2 Assault 236   
## 3 Rape 21.2  
## 4 Murder 10   
## 5 Assault 263   
## 6 Rape 44.5  
## 7 Murder 8.1  
## 8 Assault 294   
## 9 Rape 31   
## 10 Murder 8.8

#### 5. Create a summary table showing the median state-wide crime rate per 100,000 for each of the three crime categories. Arrange the data from highest to lowest.

* If done correctly, Assault will be in the top row with a median arrest rate of 159 per 100,000

# Call the Variable  
USCrime\_long %>%  
   
 # Group by crime  
 group\_by(crime)%>%  
   
 # Use summarise to creaste a summary table and use median on number per 100000 to find the medians  
 summarise(median\_arrest = median(number\_per\_100000),.groups = 'drop') %>%  
   
 # Arrange in desc order  
 arrange(desc(median\_arrest))

## # A tibble: 3 x 2  
## crime median\_arrest  
## <chr> <dbl>  
## 1 Assault 159   
## 2 Rape 20.1   
## 3 Murder 7.25

#### 6. We will now be using the data set called fish\_encounters.

* Ask for help on this data set so that you can read about it,
* get the dimensions of the data set (rows and columns), and
* print out the first 50 rows.
* Note that the seen column has the same value of 1 for all rows. This means that the fish was seen.

# Use Help to find info about variable  
help("fish\_encounters")  
  
# FInd dimensions  
dim(fish\_encounters)

## [1] 114 3

# Display first 50 values  
head(fish\_encounters,50)

## # A tibble: 50 x 3  
## fish station seen  
## <fct> <fct> <int>  
## 1 4842 Release 1  
## 2 4842 I80\_1 1  
## 3 4842 Lisbon 1  
## 4 4842 Rstr 1  
## 5 4842 Base\_TD 1  
## 6 4842 BCE 1  
## 7 4842 BCW 1  
## 8 4842 BCE2 1  
## 9 4842 BCW2 1  
## 10 4842 MAE 1  
## # ... with 40 more rows

#### 7. Put fish\_encounters into wide format. Call the new data set fish\_encounters\_wide. Display the entire data set.

* The new column headers will come from the column called station
* The values will come from the column called seen
* You should end up with many cells listed as NA. This means the fish wasn’t seen.

# Use pivot longer to make a more detailed data set from fish encounters  
fish\_encounters\_wide <- pivot\_wider(fish\_encounters,names\_from = "station", values\_from = "seen")  
  
# Display data set  
fish\_encounters\_wide

## # A tibble: 19 x 12  
## fish Release I80\_1 Lisbon Rstr Base\_TD BCE BCW BCE2 BCW2 MAE MAW  
## <fct> <int> <int> <int> <int> <int> <int> <int> <int> <int> <int> <int>  
## 1 4842 1 1 1 1 1 1 1 1 1 1 1  
## 2 4843 1 1 1 1 1 1 1 1 1 1 1  
## 3 4844 1 1 1 1 1 1 1 1 1 1 1  
## 4 4845 1 1 1 1 1 NA NA NA NA NA NA  
## 5 4847 1 1 1 NA NA NA NA NA NA NA NA  
## 6 4848 1 1 1 1 NA NA NA NA NA NA NA  
## 7 4849 1 1 NA NA NA NA NA NA NA NA NA  
## 8 4850 1 1 NA 1 1 1 1 NA NA NA NA  
## 9 4851 1 1 NA NA NA NA NA NA NA NA NA  
## 10 4854 1 1 NA NA NA NA NA NA NA NA NA  
## 11 4855 1 1 1 1 1 NA NA NA NA NA NA  
## 12 4857 1 1 1 1 1 1 1 1 1 NA NA  
## 13 4858 1 1 1 1 1 1 1 1 1 1 1  
## 14 4859 1 1 1 1 1 NA NA NA NA NA NA  
## 15 4861 1 1 1 1 1 1 1 1 1 1 1  
## 16 4862 1 1 1 1 1 1 1 1 1 NA NA  
## 17 4863 1 1 NA NA NA NA NA NA NA NA NA  
## 18 4864 1 1 NA NA NA NA NA NA NA NA NA  
## 19 4865 1 1 1 NA NA NA NA NA NA NA NA

#### 8. Put fish\_encounters into wide format once again, but this time use the values\_fill= specifier to put in zeros in all spots that had NAs in the table above. Call the new data set fish\_encounters\_wide2. Display the entire data set.

# Use pivot longer to make a more detailed data set from fish encounters  
fish\_encounters\_wide2 <- pivot\_wider(fish\_encounters,names\_from = "station", values\_from = "seen",values\_fill = 0)  
  
# Display data set  
fish\_encounters\_wide2

## # A tibble: 19 x 12  
## fish Release I80\_1 Lisbon Rstr Base\_TD BCE BCW BCE2 BCW2 MAE MAW  
## <fct> <int> <int> <int> <int> <int> <int> <int> <int> <int> <int> <int>  
## 1 4842 1 1 1 1 1 1 1 1 1 1 1  
## 2 4843 1 1 1 1 1 1 1 1 1 1 1  
## 3 4844 1 1 1 1 1 1 1 1 1 1 1  
## 4 4845 1 1 1 1 1 0 0 0 0 0 0  
## 5 4847 1 1 1 0 0 0 0 0 0 0 0  
## 6 4848 1 1 1 1 0 0 0 0 0 0 0  
## 7 4849 1 1 0 0 0 0 0 0 0 0 0  
## 8 4850 1 1 0 1 1 1 1 0 0 0 0  
## 9 4851 1 1 0 0 0 0 0 0 0 0 0  
## 10 4854 1 1 0 0 0 0 0 0 0 0 0  
## 11 4855 1 1 1 1 1 0 0 0 0 0 0  
## 12 4857 1 1 1 1 1 1 1 1 1 0 0  
## 13 4858 1 1 1 1 1 1 1 1 1 1 1  
## 14 4859 1 1 1 1 1 0 0 0 0 0 0  
## 15 4861 1 1 1 1 1 1 1 1 1 1 1  
## 16 4862 1 1 1 1 1 1 1 1 1 0 0  
## 17 4863 1 1 0 0 0 0 0 0 0 0 0  
## 18 4864 1 1 0 0 0 0 0 0 0 0 0  
## 19 4865 1 1 1 0 0 0 0 0 0 0 0

#### 9. Put fish\_encounters\_wide2 back into long format. Put the stations back into a column called “station”, and the values into a column called “fish\_observed”. Call the new data set fish\_encouters\_long2. Display the entire data set.

* Now, there will be both 1s indicating that the fish was seen and 0s indicating that it was not.
* The first station should be Release
* There should be 209 rows. Note that the original fish\_encounters table had 114 rows because fish that weren’t seen were not recorded.

# Assign Variable  
fish\_encounters\_long2 <- pivot\_longer(fish\_encounters\_wide2,cols = "Release":"MAW", names\_to = "station", values\_to = "fish\_observed")  
  
# Display dataset  
fish\_encounters\_long2

## # A tibble: 209 x 3  
## fish station fish\_observed  
## <fct> <chr> <int>  
## 1 4842 Release 1  
## 2 4842 I80\_1 1  
## 3 4842 Lisbon 1  
## 4 4842 Rstr 1  
## 5 4842 Base\_TD 1  
## 6 4842 BCE 1  
## 7 4842 BCW 1  
## 8 4842 BCE2 1  
## 9 4842 BCW2 1  
## 10 4842 MAE 1  
## # ... with 199 more rows

#### 10. Use fish\_encouters\_long2 to create a summary table that groups the data by fish and displays the total number of stations that observed that fish. Arrange the fish from most to least frequently observed.

* If done correctly, fish 4842 will be on top with 11 stations observing the fish.

# Call variable  
fish\_encounters\_long2 %>%  
   
 # Group by fish  
 group\_by(fish) %>%  
   
 # Summarize and sum all fish observed  
 summarise(number\_of\_stations = sum(fish\_observed),.groups = 'drop') %>%  
   
 # Rearrange data  
 arrange(desc(number\_of\_stations))

## # A tibble: 19 x 2  
## fish number\_of\_stations  
## <fct> <int>  
## 1 4842 11  
## 2 4843 11  
## 3 4844 11  
## 4 4858 11  
## 5 4861 11  
## 6 4857 9  
## 7 4862 9  
## 8 4850 6  
## 9 4845 5  
## 10 4855 5  
## 11 4859 5  
## 12 4848 4  
## 13 4847 3  
## 14 4865 3  
## 15 4849 2  
## 16 4851 2  
## 17 4854 2  
## 18 4863 2  
## 19 4864 2

#### 11. Use fish\_encouters\_long2 to create a summary table that groups the data by station and displays the number of unique fish species that were observed at each station. Arrange the stations from stations that saw the most to the least number of different species.

* If done correctly, station 180\_1 will be on top with 19 different species observed.

# Call variable  
fish\_encounters\_long2 %>%  
   
 # group by station  
 group\_by(station) %>%  
   
 # Summarize and sum all fish observed  
 summarise(number\_of\_stations = sum(fish\_observed),.groups = 'drop') %>%  
   
 # Rearrange data  
 arrange(desc(number\_of\_stations))

## # A tibble: 11 x 2  
## station number\_of\_stations  
## <chr> <int>  
## 1 I80\_1 19  
## 2 Release 19  
## 3 Lisbon 13  
## 4 Rstr 12  
## 5 Base\_TD 11  
## 6 BCE 8  
## 7 BCW 8  
## 8 BCE2 7  
## 9 BCW2 7  
## 10 MAE 5  
## 11 MAW 5