Week-4: Code-along

Annette

2023-09-04

# II. Code to edit and execute using the Code-along.Rmd file

## A. Data Wrangling

### 1. Loading packages (Slide #16)

library(tidyverse)

## ── Attaching core tidyverse packages ──────────────────────── tidyverse 2.0.0 ──  
## ✔ dplyr 1.1.2 ✔ readr 2.1.4  
## ✔ forcats 1.0.0 ✔ stringr 1.5.0  
## ✔ ggplot2 3.4.3 ✔ tibble 3.2.1  
## ✔ lubridate 1.9.2 ✔ tidyr 1.3.0  
## ✔ purrr 1.0.2   
## ── Conflicts ────────────────────────────────────────── tidyverse\_conflicts() ──  
## ✖ dplyr::filter() masks stats::filter()  
## ✖ dplyr::lag() masks stats::lag()  
## ℹ Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become errors

### 2. Loading data-set (Slide #16)

hotels <- read\_csv("hotels.csv")

## Rows: 119390 Columns: 32  
## ── Column specification ────────────────────────────────────────────────────────  
## Delimiter: ","  
## chr (13): hotel, arrival\_date\_month, meal, country, market\_segment, distrib...  
## dbl (18): is\_canceled, lead\_time, arrival\_date\_year, arrival\_date\_week\_numb...  
## date (1): reservation\_status\_date  
##   
## ℹ Use `spec()` to retrieve the full column specification for this data.  
## ℹ Specify the column types or set `show\_col\_types = FALSE` to quiet this message.

### 3. List names of the variables in the data-set (Slide #19)

names(hotels)

## [1] "hotel" "is\_canceled"   
## [3] "lead\_time" "arrival\_date\_year"   
## [5] "arrival\_date\_month" "arrival\_date\_week\_number"   
## [7] "arrival\_date\_day\_of\_month" "stays\_in\_weekend\_nights"   
## [9] "stays\_in\_week\_nights" "adults"   
## [11] "children" "babies"   
## [13] "meal" "country"   
## [15] "market\_segment" "distribution\_channel"   
## [17] "is\_repeated\_guest" "previous\_cancellations"   
## [19] "previous\_bookings\_not\_canceled" "reserved\_room\_type"   
## [21] "assigned\_room\_type" "booking\_changes"   
## [23] "deposit\_type" "agent"   
## [25] "company" "days\_in\_waiting\_list"   
## [27] "customer\_type" "adr"   
## [29] "required\_car\_parking\_spaces" "total\_of\_special\_requests"   
## [31] "reservation\_status" "reservation\_status\_date"

### 4. Glimpse of contents of the data-set (Slide #20)

glimpse(hotels)

## Rows: 119,390  
## Columns: 32  
## $ hotel <chr> "Resort Hotel", "Resort Hotel", "Resort…  
## $ is\_canceled <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 1, 0, 0, …  
## $ lead\_time <dbl> 342, 737, 7, 13, 14, 14, 0, 9, 85, 75, …  
## $ arrival\_date\_year <dbl> 2015, 2015, 2015, 2015, 2015, 2015, 201…  
## $ arrival\_date\_month <chr> "July", "July", "July", "July", "July",…  
## $ arrival\_date\_week\_number <dbl> 27, 27, 27, 27, 27, 27, 27, 27, 27, 27,…  
## $ arrival\_date\_day\_of\_month <dbl> 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, …  
## $ stays\_in\_weekend\_nights <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, …  
## $ stays\_in\_week\_nights <dbl> 0, 0, 1, 1, 2, 2, 2, 2, 3, 3, 4, 4, 4, …  
## $ adults <dbl> 2, 2, 1, 1, 2, 2, 2, 2, 2, 2, 2, 2, 2, …  
## $ children <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, …  
## $ babies <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, …  
## $ meal <chr> "BB", "BB", "BB", "BB", "BB", "BB", "BB…  
## $ country <chr> "PRT", "PRT", "GBR", "GBR", "GBR", "GBR…  
## $ market\_segment <chr> "Direct", "Direct", "Direct", "Corporat…  
## $ distribution\_channel <chr> "Direct", "Direct", "Direct", "Corporat…  
## $ is\_repeated\_guest <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, …  
## $ previous\_cancellations <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, …  
## $ previous\_bookings\_not\_canceled <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, …  
## $ reserved\_room\_type <chr> "C", "C", "A", "A", "A", "A", "C", "C",…  
## $ assigned\_room\_type <chr> "C", "C", "C", "A", "A", "A", "C", "C",…  
## $ booking\_changes <dbl> 3, 4, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, …  
## $ deposit\_type <chr> "No Deposit", "No Deposit", "No Deposit…  
## $ agent <chr> "NULL", "NULL", "NULL", "304", "240", "…  
## $ company <chr> "NULL", "NULL", "NULL", "NULL", "NULL",…  
## $ days\_in\_waiting\_list <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, …  
## $ customer\_type <chr> "Transient", "Transient", "Transient", …  
## $ adr <dbl> 0.00, 0.00, 75.00, 75.00, 98.00, 98.00,…  
## $ required\_car\_parking\_spaces <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, …  
## $ total\_of\_special\_requests <dbl> 0, 0, 0, 0, 1, 1, 0, 1, 1, 0, 0, 0, 3, …  
## $ reservation\_status <chr> "Check-Out", "Check-Out", "Check-Out", …  
## $ reservation\_status\_date <date> 2015-07-01, 2015-07-01, 2015-07-02, 20…

## B. Choosing rows or columns

### 5. Select a single column (Slide #24)

select(hotels,lead\_time)

## # A tibble: 119,390 × 1  
## lead\_time  
## <dbl>  
## 1 342  
## 2 737  
## 3 7  
## 4 13  
## 5 14  
## 6 14  
## 7 0  
## 8 9  
## 9 85  
## 10 75  
## # ℹ 119,380 more rows

### 6. Select multiple columns (Slide #25)

select(hotels,lead\_time, agent, market\_segment)

## # A tibble: 119,390 × 3  
## lead\_time agent market\_segment  
## <dbl> <chr> <chr>   
## 1 342 NULL Direct   
## 2 737 NULL Direct   
## 3 7 NULL Direct   
## 4 13 304 Corporate   
## 5 14 240 Online TA   
## 6 14 240 Online TA   
## 7 0 NULL Direct   
## 8 9 303 Direct   
## 9 85 240 Online TA   
## 10 75 15 Offline TA/TO   
## # ℹ 119,380 more rows

### 7. Arrange entries of a column (Slide #28)

arrange(hotels,lead\_time)

## # A tibble: 119,390 × 32  
## hotel is\_canceled lead\_time arrival\_date\_year arrival\_date\_month  
## <chr> <dbl> <dbl> <dbl> <chr>   
## 1 Resort Hotel 0 0 2015 July   
## 2 Resort Hotel 0 0 2015 July   
## 3 Resort Hotel 0 0 2015 July   
## 4 Resort Hotel 0 0 2015 July   
## 5 Resort Hotel 0 0 2015 July   
## 6 Resort Hotel 0 0 2015 July   
## 7 Resort Hotel 0 0 2015 July   
## 8 Resort Hotel 0 0 2015 July   
## 9 Resort Hotel 0 0 2015 July   
## 10 Resort Hotel 0 0 2015 July   
## # ℹ 119,380 more rows  
## # ℹ 27 more variables: arrival\_date\_week\_number <dbl>,  
## # arrival\_date\_day\_of\_month <dbl>, stays\_in\_weekend\_nights <dbl>,  
## # stays\_in\_week\_nights <dbl>, adults <dbl>, children <dbl>, babies <dbl>,  
## # meal <chr>, country <chr>, market\_segment <chr>,  
## # distribution\_channel <chr>, is\_repeated\_guest <dbl>,  
## # previous\_cancellations <dbl>, previous\_bookings\_not\_canceled <dbl>, …

### 8. Arrange entries of a column in the descending order (Slide #30)

arrange(hotels, desc(lead\_time))

## # A tibble: 119,390 × 32  
## hotel is\_canceled lead\_time arrival\_date\_year arrival\_date\_month  
## <chr> <dbl> <dbl> <dbl> <chr>   
## 1 Resort Hotel 0 737 2015 July   
## 2 Resort Hotel 0 709 2016 February   
## 3 City Hotel 1 629 2017 March   
## 4 City Hotel 1 629 2017 March   
## 5 City Hotel 1 629 2017 March   
## 6 City Hotel 1 629 2017 March   
## 7 City Hotel 1 629 2017 March   
## 8 City Hotel 1 629 2017 March   
## 9 City Hotel 1 629 2017 March   
## 10 City Hotel 1 629 2017 March   
## # ℹ 119,380 more rows  
## # ℹ 27 more variables: arrival\_date\_week\_number <dbl>,  
## # arrival\_date\_day\_of\_month <dbl>, stays\_in\_weekend\_nights <dbl>,  
## # stays\_in\_week\_nights <dbl>, adults <dbl>, children <dbl>, babies <dbl>,  
## # meal <chr>, country <chr>, market\_segment <chr>,  
## # distribution\_channel <chr>, is\_repeated\_guest <dbl>,  
## # previous\_cancellations <dbl>, previous\_bookings\_not\_canceled <dbl>, …

### 9. Select columns and arrange the entries of a column (Slide #31)

arrange( select(hotels, lead\_time), desc(lead\_time))

## # A tibble: 119,390 × 1  
## lead\_time  
## <dbl>  
## 1 737  
## 2 709  
## 3 629  
## 4 629  
## 5 629  
## 6 629  
## 7 629  
## 8 629  
## 9 629  
## 10 629  
## # ℹ 119,380 more rows

### 10. Select columns and arrange the entries of a column using the pipe operator (Slide #37)

hotels %>%  
 select(lead\_time) %>%  
 arrange(desc(lead\_time))

## # A tibble: 119,390 × 1  
## lead\_time  
## <dbl>  
## 1 737  
## 2 709  
## 3 629  
## 4 629  
## 5 629  
## 6 629  
## 7 629  
## 8 629  
## 9 629  
## 10 629  
## # ℹ 119,380 more rows

### 11. Pick rows matching a condition (Slide #44)

hotels %>%  
 filter(children >=1) %>%  
 select(hotel,children)

## # A tibble: 8,590 × 2  
## hotel children  
## <chr> <dbl>  
## 1 Resort Hotel 1  
## 2 Resort Hotel 2  
## 3 Resort Hotel 2  
## 4 Resort Hotel 2  
## 5 Resort Hotel 1  
## 6 Resort Hotel 1  
## 7 Resort Hotel 2  
## 8 Resort Hotel 2  
## 9 Resort Hotel 1  
## 10 Resort Hotel 2  
## # ℹ 8,580 more rows

### 12. Pick rows matching multiple conditions (Slide #46)

hotels %>%  
 filter(children >=1, hotel == "City Hotel") %>%  
 select(hotel,children)

## # A tibble: 5,106 × 2  
## hotel children  
## <chr> <dbl>  
## 1 City Hotel 1  
## 2 City Hotel 2  
## 3 City Hotel 1  
## 4 City Hotel 1  
## 5 City Hotel 1  
## 6 City Hotel 1  
## 7 City Hotel 1  
## 8 City Hotel 1  
## 9 City Hotel 1  
## 10 City Hotel 1  
## # ℹ 5,096 more rows

### 13. Non-conditional selection of rows: sequence of indices (Slide #49)

hotels %>% slice(1:5)

## # A tibble: 5 × 32  
## hotel is\_canceled lead\_time arrival\_date\_year arrival\_date\_month  
## <chr> <dbl> <dbl> <dbl> <chr>   
## 1 Resort Hotel 0 342 2015 July   
## 2 Resort Hotel 0 737 2015 July   
## 3 Resort Hotel 0 7 2015 July   
## 4 Resort Hotel 0 13 2015 July   
## 5 Resort Hotel 0 14 2015 July   
## # ℹ 27 more variables: arrival\_date\_week\_number <dbl>,  
## # arrival\_date\_day\_of\_month <dbl>, stays\_in\_weekend\_nights <dbl>,  
## # stays\_in\_week\_nights <dbl>, adults <dbl>, children <dbl>, babies <dbl>,  
## # meal <chr>, country <chr>, market\_segment <chr>,  
## # distribution\_channel <chr>, is\_repeated\_guest <dbl>,  
## # previous\_cancellations <dbl>, previous\_bookings\_not\_canceled <dbl>,  
## # reserved\_room\_type <chr>, assigned\_room\_type <chr>, …

### 14. Non-conditional selection of rows: non-consecutive/specific indices (Slide #50)

hotels %>% slice(1,3,5)

## # A tibble: 3 × 32  
## hotel is\_canceled lead\_time arrival\_date\_year arrival\_date\_month  
## <chr> <dbl> <dbl> <dbl> <chr>   
## 1 Resort Hotel 0 342 2015 July   
## 2 Resort Hotel 0 7 2015 July   
## 3 Resort Hotel 0 14 2015 July   
## # ℹ 27 more variables: arrival\_date\_week\_number <dbl>,  
## # arrival\_date\_day\_of\_month <dbl>, stays\_in\_weekend\_nights <dbl>,  
## # stays\_in\_week\_nights <dbl>, adults <dbl>, children <dbl>, babies <dbl>,  
## # meal <chr>, country <chr>, market\_segment <chr>,  
## # distribution\_channel <chr>, is\_repeated\_guest <dbl>,  
## # previous\_cancellations <dbl>, previous\_bookings\_not\_canceled <dbl>,  
## # reserved\_room\_type <chr>, assigned\_room\_type <chr>, …

### 15. Pick unique rows using distinct() (Slide #52)

hotels %>% distinct(hotel)

## # A tibble: 2 × 1  
## hotel   
## <chr>   
## 1 Resort Hotel  
## 2 City Hotel

## C. Creating new columns

### 16. Creating a single column with mutate() (Slide #56)

hotels %>%  
 mutate(little\_ones = children + babies) %>%  
 select(hotel, little\_ones, children, babies)

## # A tibble: 119,390 × 4  
## hotel little\_ones children babies  
## <chr> <dbl> <dbl> <dbl>  
## 1 Resort Hotel 0 0 0  
## 2 Resort Hotel 0 0 0  
## 3 Resort Hotel 0 0 0  
## 4 Resort Hotel 0 0 0  
## 5 Resort Hotel 0 0 0  
## 6 Resort Hotel 0 0 0  
## 7 Resort Hotel 0 0 0  
## 8 Resort Hotel 0 0 0  
## 9 Resort Hotel 0 0 0  
## 10 Resort Hotel 0 0 0  
## # ℹ 119,380 more rows

### 17. Creating multiple columns with mutate() (Slide #58)

hotels %>%  
 mutate(little\_ones = children + babies, average\_little\_ones = mean(little\_ones)) %>%  
 select(hotel, little\_ones, children, babies, average\_little\_ones)

## # A tibble: 119,390 × 5  
## hotel little\_ones children babies average\_little\_ones  
## <chr> <dbl> <dbl> <dbl> <dbl>  
## 1 Resort Hotel 0 0 0 NA  
## 2 Resort Hotel 0 0 0 NA  
## 3 Resort Hotel 0 0 0 NA  
## 4 Resort Hotel 0 0 0 NA  
## 5 Resort Hotel 0 0 0 NA  
## 6 Resort Hotel 0 0 0 NA  
## 7 Resort Hotel 0 0 0 NA  
## 8 Resort Hotel 0 0 0 NA  
## 9 Resort Hotel 0 0 0 NA  
## 10 Resort Hotel 0 0 0 NA  
## # ℹ 119,380 more rows

## D. More operations with examples

### 18. count() to get frequencies (Slide #60)

hotels %>% count(market\_segment)

## # A tibble: 8 × 2  
## market\_segment n  
## <chr> <int>  
## 1 Aviation 237  
## 2 Complementary 743  
## 3 Corporate 5295  
## 4 Direct 12606  
## 5 Groups 19811  
## 6 Offline TA/TO 24219  
## 7 Online TA 56477  
## 8 Undefined 2

### 19. count() to get frequencies with sorting of count (Slide #61)

hotels %>% count(market\_segment, sort = TRUE)

## # A tibble: 8 × 2  
## market\_segment n  
## <chr> <int>  
## 1 Online TA 56477  
## 2 Offline TA/TO 24219  
## 3 Groups 19811  
## 4 Direct 12606  
## 5 Corporate 5295  
## 6 Complementary 743  
## 7 Aviation 237  
## 8 Undefined 2

### 20. count() multiple variables (Slide #62)

hotels %>% count(hotel, market\_segment)

## # A tibble: 14 × 3  
## hotel market\_segment n  
## <chr> <chr> <int>  
## 1 City Hotel Aviation 237  
## 2 City Hotel Complementary 542  
## 3 City Hotel Corporate 2986  
## 4 City Hotel Direct 6093  
## 5 City Hotel Groups 13975  
## 6 City Hotel Offline TA/TO 16747  
## 7 City Hotel Online TA 38748  
## 8 City Hotel Undefined 2  
## 9 Resort Hotel Complementary 201  
## 10 Resort Hotel Corporate 2309  
## 11 Resort Hotel Direct 6513  
## 12 Resort Hotel Groups 5836  
## 13 Resort Hotel Offline TA/TO 7472  
## 14 Resort Hotel Online TA 17729

### 21. summarise() for summary statistics (Slide #63)

hotels %>% summarise (mean\_adr = mean(adr))

## # A tibble: 1 × 1  
## mean\_adr  
## <dbl>  
## 1 102.

### 22. summarise() by using group\_by to find mean (Slide #64)

hotels %>%   
 group\_by(hotel) %>%  
 summarise(mean\_adr= mean(adr))

## # A tibble: 2 × 2  
## hotel mean\_adr  
## <chr> <dbl>  
## 1 City Hotel 105.   
## 2 Resort Hotel 95.0

### 23. summarise() by using group\_by to get count (Slide #65)

hotels %>%   
 group\_by(hotel) %>%  
 summarise(count = n())

## # A tibble: 2 × 2  
## hotel count  
## <chr> <int>  
## 1 City Hotel 79330  
## 2 Resort Hotel 40060

### 24. summarise() for multiple summary statistics (Slide #67)

hotels %>%  
 summarise(  
 min\_adr = min(adr),  
 mean\_adr = mean(adr),  
 median\_adr = median(adr),  
 max\_adr = max(adr)  
 )

## # A tibble: 1 × 4  
## min\_adr mean\_adr median\_adr max\_adr  
## <dbl> <dbl> <dbl> <dbl>  
## 1 -6.38 102. 94.6 5400

### 25. select(), slice() and arrange() (Slide #68)

hotels %>%  
 select(hotel, lead\_time) %>%  
 slice(1:5) %>%  
 arrange(lead\_time)

## # A tibble: 5 × 2  
## hotel lead\_time  
## <chr> <dbl>  
## 1 Resort Hotel 7  
## 2 Resort Hotel 13  
## 3 Resort Hotel 14  
## 4 Resort Hotel 342  
## 5 Resort Hotel 737

### 26. select(), arrange() and slice() (Slide #69)

hotels %>%  
 select(hotel, lead\_time) %>%  
 arrange(lead\_time) %>%  
 slice(1:5)

## # A tibble: 5 × 2  
## hotel lead\_time  
## <chr> <dbl>  
## 1 Resort Hotel 0  
## 2 Resort Hotel 0  
## 3 Resort Hotel 0  
## 4 Resort Hotel 0  
## 5 Resort Hotel 0

### 27. filter() to select rows based on conditions (Slide #73)

hotels %>%  
 filter(  
 adults == 0,  
 children >= 1  
 ) %>%  
 select(adults, babies, children)

## # A tibble: 223 × 3  
## adults babies children  
## <dbl> <dbl> <dbl>  
## 1 0 0 3  
## 2 0 0 2  
## 3 0 0 2  
## 4 0 0 2  
## 5 0 0 2  
## 6 0 0 3  
## 7 0 1 2  
## 8 0 0 2  
## 9 0 0 2  
## 10 0 0 2  
## # ℹ 213 more rows

### 28. filter() to select rows based on complicated conditions (Slide #74)

hotels %>%  
 filter(adults == 1,  
 children >= 1 | babies >= 1) %>%  
 select(adults, babies, children)

## # A tibble: 450 × 3  
## adults babies children  
## <dbl> <dbl> <dbl>  
## 1 1 0 2  
## 2 1 0 2  
## 3 1 0 1  
## 4 1 1 0  
## 5 1 0 1  
## 6 1 0 1  
## 7 1 0 2  
## 8 1 0 2  
## 9 1 0 1  
## 10 1 0 1  
## # ℹ 440 more rows

### 29. count() and arrange() (Slide #76)

hotels %>%  
 count(market\_segment) %>%  
 arrange(desc(n))

## # A tibble: 8 × 2  
## market\_segment n  
## <chr> <int>  
## 1 Online TA 56477  
## 2 Offline TA/TO 24219  
## 3 Groups 19811  
## 4 Direct 12606  
## 5 Corporate 5295  
## 6 Complementary 743  
## 7 Aviation 237  
## 8 Undefined 2

### 30. mutate(), select() and arrange() (Slide #77)

hotels %>%  
 mutate(little\_ones = children + babies) %>%  
 select(children, babies, little\_ones) %>%  
 arrange(desc(little\_ones))

## # A tibble: 119,390 × 3  
## children babies little\_ones  
## <dbl> <dbl> <dbl>  
## 1 10 0 10  
## 2 0 10 10  
## 3 0 9 9  
## 4 2 1 3  
## 5 2 1 3  
## 6 2 1 3  
## 7 3 0 3  
## 8 2 1 3  
## 9 2 1 3  
## 10 3 0 3  
## # ℹ 119,380 more rows

### 31. mutate(), filter() and select() (Slide #78)

hotels %>%  
 mutate(little\_ones = children + babies) %>%  
 filter(  
 little\_ones >= 1,  
 hotel == "Resort Hotel"  
 ) %>%  
 select(hotel, little\_ones)

## # A tibble: 3,929 × 2  
## hotel little\_ones  
## <chr> <dbl>  
## 1 Resort Hotel 1  
## 2 Resort Hotel 2  
## 3 Resort Hotel 2  
## 4 Resort Hotel 2  
## 5 Resort Hotel 1  
## 6 Resort Hotel 1  
## 7 Resort Hotel 2  
## 8 Resort Hotel 2  
## 9 Resort Hotel 1  
## 10 Resort Hotel 1  
## # ℹ 3,919 more rows