

Week-4: Code-along

Tan Ee Xuan

2023-09-06

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

A. Data Wrangling

1. Loading packages (Slide #16)

```
# Load package tidyverse
library("tidyverse")
```

```
## — Attaching packages — tidyverse 1.3.2 —
## ✓ ggplot2 3.3.6      ✓ purrr 0.3.4
## ✓ tibble 3.1.8      ✓ dplyr 1.0.9
## ✓ tidyr 1.2.0       ✓ stringr 1.4.0
## ✓ readr 2.1.2       ✓ forcats 0.5.1
## — Conflicts — tidyverse_conflicts() —
## ✖ dplyr::filter() masks stats::filter()
## ✖ dplyr::lag() masks stats::lag()
```

2. Loading data-set (Slide #16)

```
# Read data from the hotels.csv file and assign it to a variable named, "hotels"
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)

```
# Enter code here
names(hotels)
```

```
## [1] "hotel"
## [3] "lead_time"
## [5] "arrival_date_month"
## [7] "arrival_date_day_of_month"
## [9] "stays_in_week_nights"
## [11] "children"
## [13] "meal"
## [15] "market_segment"
## [17] "is_repeated_guest"
## [19] "previous_bookings_not_canceled"
## [21] "assigned_room_type"
## [23] "deposit_type"
## [25] "company"
## [27] "customer_type"
## [29] "required_car_parking_spaces"
## [31] "reservation_status"

"is_canceled"
"arrival_date_year"
"arrival_date_week_number"
"stays_in_weekend_nights"
"adults"
"babies"
"country"
"distribution_channel"
"previous_cancellations"
"reserved_room_type"
"booking_changes"
"agent"
"days_in_waiting_list"
"adr"
"total_of_special_requests"
"reservation_status_date"
```

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

```
# Enter code here
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)

```
# Enter code here
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
## # ... with 119,380 more rows
## # i Use `print(n = ...)` to see more rows
```

6. Select multiple columns (Slide #25)

```
# Enter code here
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
## # ... with 119,380 more rows
## # i Use `print(n = ...)` to see more rows
```

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

```
# Enter code here
arrange(hotels, lead_time)
```

```
## # A tibble: 119,390 × 32
##   hotel is_ca...1 lead_...2 arriv...3 arriv...4 arriv...5 arriv...6 stays...7 stays...8 adults
##   <chr>   <dbl>   <dbl>   <dbl> <chr>       <dbl>   <dbl>   <dbl>   <dbl>   <dbl>
## 1 Resor...     0     0   2015 July       27     1     0     2     2
## 2 Resor...     0     0   2015 July       27     1     0     1     2
## 3 Resor...     0     0   2015 July       27     2     0     1     2
## 4 Resor...     0     0   2015 July       27     2     0     1     2
## 5 Resor...     0     0   2015 July       27     2     0     1     2
## 6 Resor...     0     0   2015 July       28     5     1     0     2
## 7 Resor...     0     0   2015 July       28     6     0     0     1
## 8 Resor...     0     0   2015 July       28     7     0     1     1
## 9 Resor...     0     0   2015 July       28     7     0     1     3
## 10 Resor...    0     0   2015 July       28     7     0     1     1
## # ... with 119,380 more rows, 22 more variables: 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>, booking_changes <dbl>,
## #   deposit_type <chr>, agent <chr>, company <chr>, days_in_waiting_list <dbl>,
## #   customer_type <chr>, adr <dbl>, required_car_parking_spaces <dbl>, ...
## # i Use `print(n = ...)` to see more rows, and `colnames()` to see all variable names
```

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

```
# Enter code here
arrange(hotels, desc(lead_time))
```

```
## # A tibble: 119,390 × 32
##   hotel is_ca...1 lead_...2 arriv...3 arriv...4 arriv...5 arriv...6 stays...7 stays...8 adults
##   <chr>   <dbl>   <dbl>   <dbl> <chr>       <dbl>   <dbl>   <dbl>   <dbl>   <dbl>
## 1 Resor...     0   737   2015 July       27     1     0     0     2
## 2 Resor...     0   709   2016 Februa...     9   25     8    20     2
## 3 City ...     1   629   2017 March       13   30     0     1     1
## 4 City ...     1   629   2017 March       13   30     0     1     1
## 5 City ...     1   629   2017 March       13   30     0     2     2
## 6 City ...     1   629   2017 March       13   30     0     2     2
## 7 City ...     1   629   2017 March       13   30     0     2     2
## 8 City ...     1   629   2017 March       13   30     0     2     2
## 9 City ...     1   629   2017 March       13   30     0     2     2
## 10 City ...    1   629   2017 March       13   30     0     2     2
## # ... with 119,380 more rows, 22 more variables: 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>, booking_changes <dbl>,
## #   deposit_type <chr>, agent <chr>, company <chr>, days_in_waiting_list <dbl>,
## #   customer_type <chr>, adr <dbl>, required_car_parking_spaces <dbl>, ...
## # i Use `print(n = ...)` to see more rows, and `colnames()` to see all variable names
```

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

```
# Enter code here
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
## # ... with 119,380 more rows
## # i Use `print(n = ...)` to see more rows
```

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

```
# Enter code here

hotels %>%
  select(lead_time, adr, agent) %>%
  arrange(lead_time)
```

```
## # A tibble: 119,390 × 3
##   lead_time   adr agent
##   <dbl> <dbl> <chr>
## 1         0  107  NULL
## 2         0  107.  NULL
## 3         0  147  NULL
## 4         0  118.  240
## 5         0  123  NULL
## 6         0  85.1  306
## 7         0    0  250
## 8         0  110.  240
## 9         0  195  NULL
## 10        0  110.  240
## # ... with 119,380 more rows
## # i Use `print(n = ...)` to see more rows
```

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

```
# Enter code here
hotels %>%
  filter(children >= 1) %>%
  select(children, hotel)
```

```
## # A tibble: 8,590 × 2
##   children hotel
##   <dbl> <chr>
## 1         1 Resort Hotel
## 2         2 Resort Hotel
## 3         2 Resort Hotel
## 4         2 Resort Hotel
## 5         1 Resort Hotel
## 6         1 Resort Hotel
## 7         2 Resort Hotel
## 8         2 Resort Hotel
## 9         1 Resort Hotel
## 10        2 Resort Hotel
## # ... with 8,580 more rows
## # i Use `print(n = ...)` to see more rows
```

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

```
# Enter code here
hotels %>%
  filter(hotel == "Resort Hotel", children >= 1) %>%
  select(hotel, children)
```

```
## # A tibble: 3,484 × 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
## # ... with 3,474 more rows
## # i Use `print(n = ...)` to see more rows
```

13. Non-conditional selection of rows: sequence of indices

(Slide #49)

```
# Enter code here
hotels %>%
  slice(1:5)
```

```
## # A tibble: 5 × 32
##   hotel    is_ca...1 lead_...2 arriv...3 arriv...4 arriv...5 arriv...6 stays...7 stays...8 adults
##   <chr>      <dbl>    <dbl>    <dbl> <chr>      <dbl>    <dbl>    <dbl>    <dbl>    <dbl>
## 1 Resort...      0      342    2015 July      27      1      0      0      2
## 2 Resort...      0      737    2015 July      27      1      0      0      2
## 3 Resort...      0       7    2015 July      27      1      0      1      1
## 4 Resort...      0      13    2015 July      27      1      0      1      1
## 5 Resort...      0      14    2015 July      27      1      0      2      2
## # ... with 22 more variables: 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>, booking_changes <dbl>, deposit_type <chr>,
## #   agent <chr>, company <chr>, days_in_waiting_list <dbl>,
## #   customer_type <chr>, adr <dbl>, required_car_parking_spaces <dbl>, ...
## # i Use `colnames()` to see all variable names
```

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

```
# Enter code here
hotels %>%
  slice(1,3,5)
```

```
## # A tibble: 3 × 32
##   hotel    is_ca...1 lead_...2 arriv...3 arriv...4 arriv...5 arriv...6 stays...7 stays...8 adults
##   <chr>      <dbl>    <dbl>    <dbl> <chr>      <dbl>    <dbl>    <dbl>    <dbl>    <dbl>
## 1 Resort...      0      342    2015 July      27      1      0      0      2
## 2 Resort...      0       7    2015 July      27      1      0      1      1
## 3 Resort...      0      14    2015 July      27      1      0      2      2
## # ... with 22 more variables: 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>, booking_changes <dbl>, deposit_type <chr>,
## #   agent <chr>, company <chr>, days_in_waiting_list <dbl>,
## #   customer_type <chr>, adr <dbl>, required_car_parking_spaces <dbl>, ...
## # i Use `colnames()` to see all variable names
```

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

```
# Enter code here
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)

```
# Enter code here
hotels %>%
  mutate(little_ones = children + babies) %>%
  select(little_ones, children, babies, hotel)
```

```
## # A tibble: 119,390 × 4
##   little_ones children babies hotel
##   <dbl>      <dbl> <dbl> <chr>
## 1         0         0     0 Resort Hotel
## 2         0         0     0 Resort Hotel
## 3         0         0     0 Resort Hotel
## 4         0         0     0 Resort Hotel
## 5         0         0     0 Resort Hotel
## 6         0         0     0 Resort Hotel
## 7         0         0     0 Resort Hotel
## 8         0         0     0 Resort Hotel
## 9         0         0     0 Resort Hotel
## 10        0         0     0 Resort Hotel
## # ... with 119,380 more rows
## # i Use `print(n = ...)` to see more rows
```

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

```
# Enter code here
which(is.na(hotels$children))
```

```
## [1] 40601 40668 40680 41161
```

```
which(is.na(hotels$babies))
```

```
## integer(0)
```

```
hotels %>%
  slice(-c(40601, 40668, 40680, 41161)) %>%
  mutate(little_ones = children + babies,
         average_little_ones = mean(little_ones)) %>%
  select(little_ones, average_little_ones, children, babies)
```

```
## # A tibble: 119,386 × 4
##   little_ones average_little_ones children babies
##   <dbl>         <dbl>         <dbl>   <dbl>
## 1         0         0.112         0       0
## 2         0         0.112         0       0
## 3         0         0.112         0       0
## 4         0         0.112         0       0
## 5         0         0.112         0       0
## 6         0         0.112         0       0
## 7         0         0.112         0       0
## 8         0         0.112         0       0
## 9         0         0.112         0       0
## 10        0         0.112         0       0
## # ... with 119,376 more rows
## # i Use `print(n = ...)` to see more rows
```

D. More operations with examples

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

```
# Enter code here
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)

```
# Enter code here
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)

```
# Enter code here
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)

```
# Enter code here
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)

```
# Enter code here
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)

```
# Enter code here
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)

```
# Enter code here
summarise(hotels,
  mean_adr = mean(adr),
  min_adr = min(adr),
  max_adr = max(adr),
  median_adr = median(adr))
```

```
## # A tibble: 1 × 4
##   mean_adr min_adr max_adr median_adr
##   <dbl>   <dbl>   <dbl>      <dbl>
## 1    102.    -6.38    5400      94.6
```

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

```
# Enter code here
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)

```
# Enter code here
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)

```
# Enter code here
hotels %>%
  filter(adults == 0, children >= 1)
```

```
## # A tibble: 223 × 32
##   hotel is_ca...1 lead_...2 arriv...3 arriv...4 arriv...5 arriv...6 stays...7 stays...8 adults
##   <chr>    <dbl>    <dbl>    <dbl> <chr>    <dbl>    <dbl>    <dbl>    <dbl>    <dbl>
## 1 City ...      0      1    2015 August      33     10      1      1      0
## 2 City ...      0    104    2015 August      33     11      0      3      0
## 3 City ...      0      3    2015 August      34     16      2      0      0
## 4 City ...      0     15    2015 August      35     28      0      1      0
## 5 City ...      1     48    2015 October     43     19      1      3      0
## 6 City ...      1      6    2015 Decemb...   51     13      1      0      0
## 7 City ...      0      6    2015 Decemb...   51     18      0      1      0
## 8 City ...      0      1    2015 Decemb...   52     23      0      3      0
## 9 City ...      1     12    2015 Decemb...   52     24      0      2      0
## 10 City ...     0      7    2015 Decemb...   52     26      2      1      0
## # ... with 213 more rows, 22 more variables: 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>, booking_changes <dbl>,
## #   deposit_type <chr>, agent <chr>, company <chr>, days_in_waiting_list <dbl>,
## #   customer_type <chr>, adr <dbl>, required_car_parking_spaces <dbl>, ...
## # i Use `print(n = ...)` to see more rows, and `colnames()` to see all variable names
```

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

```
# Enter code here
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
## # ... with 440 more rows
## # i Use `print(n = ...)` to see more rows
```

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

```
# Enter code here
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)

```
# Enter code here
hotels %>%
  mutate(little_ones = babies + children) %>%
  select(little_ones, babies, children) %>%
  arrange(desc(little_ones))
```

```
## # A tibble: 119,390 × 3
##   little_ones babies children
##   <dbl> <dbl> <dbl>
## 1      10      0      10
## 2      10     10       0
## 3       9      9       0
## 4       3      1       2
## 5       3      1       2
## 6       3      1       2
## 7       3      0       3
## 8       3      1       2
## 9       3      1       2
## 10      3      0       3
## # ... with 119,380 more rows
## # i Use `print(n = ...)` to see more rows
```

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

```
# Enter code here
hotels %>%
  mutate(little_ones = babies + children) %>%
  filter(little_ones >= 1, hotel == "Resort Hotel") %>%
  select(little_ones, hotel)
```

```
## # A tibble: 3,929 × 2
##   little_ones hotel
##   <dbl> <chr>
## 1         1 Resort Hotel
## 2         2 Resort Hotel
## 3         2 Resort Hotel
## 4         2 Resort Hotel
## 5         1 Resort Hotel
## 6         1 Resort Hotel
## 7         2 Resort Hotel
## 8         2 Resort Hotel
## 9         1 Resort Hotel
## 10        1 Resort Hotel
## # ... with 3,919 more rows
## # i Use `print(n = ...)` to see more rows
```