Week-4: Code-along

Yang Juan Hun

2023-09-03

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)
```

```
## Warning: package 'tidyverse' was built under R version 4.2.2
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.2 --
## v ggplot2 3.4.0     v purrr   1.0.1
## v tibble  3.1.8     v dplyr   1.1.0
## v tidyr   1.2.1     v stringr 1.5.0
## v readr   2.1.3     v forcats 0.5.2
```

```
## Warning: package 'ggplot2' was built under R version 4.2.2
```

```
## Warning: package 'tibble' was built under R version 4.2.1
```

```
## Warning: package 'tidyr' was built under R version 4.2.1
```

```
## Warning: package 'readr' was built under R version 4.2.2
```

```
## Warning: package 'purrr' was built under R version 4.2.2
```

```
## Warning: package 'dplyr' was built under R version 4.2.2
```

```
## Warning: package 'stringr' was built under R version 4.2.2
```

```
## Warning: package 'forcats' was built under R version 4.2.2
```

```
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

2. Loading data-set (Slide #16)

```r
# 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)

```r
# Enter code here
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)

```r
# 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 x 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
```

6. Select multiple columns (Slide #25)

```
# Enter code here
select(hotels, lead_time, agent, market_segment)
```

```
## # A tibble: 119,390 x 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
```

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

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

```
## # A tibble: 119,390 x 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>, ...
```

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

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

```
## # A tibble: 119,390 x 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>, ...
```

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

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

```
## # A tibble: 119,390 x 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
```

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

```r
# Enter code here
hotels %>%
  select(lead_time) %>%
  arrange(desc(lead_time))
```

```
## # A tibble: 119,390 x 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
```

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

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

```
## # A tibble: 8,590 x 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 8,580 more rows
```

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

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

```
## # A tibble: 5,106 x 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
## # ... with 5,096 more rows
```

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

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

```
## # A tibble: 5 x 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>, ...
```

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

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

```
## # A tibble: 3 x 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>, ...
```

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

```
# Enter code here
hotels %>%
 distinct(hotel)
```

```
## # A tibble: 2 x 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(hotel, little_ones,children,babies)
```

```
## # A tibble: 119,390 x 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
## # ... with 119,380 more rows
```

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

```
# Enter code here
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 x 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
## # ... with 119,380 more rows
```

D. More operations with examples

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

```
# Enter code here
hotels %>%
 count(market_segment)
```

```
## # A tibble: 8 x 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 x 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 x 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 x 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 x 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 x 2
##   hotel        count
##   <chr>        <int>
## 1 City Hotel   79330
## 2 Resort Hotel 40060
```

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

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

```
## # A tibble: 1 x 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)

```
# Enter code here
hotels %>%
 select(hotel, lead_time) %>%
 slice(1:5) %>%
 arrange(lead_time)
```

```
## # A tibble: 5 x 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 x 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(hotel == "City Hotel")
```

```
## # A tibble: 79,330 x 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       6    2015 July         27       1       0       2      1
##  2 City ~       1      88    2015 July         27       1       0       4      2
##  3 City ~       1      65    2015 July         27       1       0       4      1
##  4 City ~       1      92    2015 July         27       1       2       4      2
##  5 City ~       1     100    2015 July         27       2       0       2      2
##  6 City ~       1      79    2015 July         27       2       0       3      2
##  7 City ~       0       3    2015 July         27       2       0       3      1
##  8 City ~       1      63    2015 July         27       2       1       3      1
##  9 City ~       1      62    2015 July         27       2       2       3      2
## 10 City ~       1      62    2015 July         27       2       2       3      2
## # ... with 79,320 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>, ...
```

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 x 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
```

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

```
# Enter code here
hotels %>%
 count(market_segment) %>%
 arrange(desc(n))
```

```
## # A tibble: 8 x 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 = children + babies) %>%
 select(children, babies, little_ones) %>%
 arrange(desc(little_ones))
```

```
## # A tibble: 119,390 x 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
## # ... with 119,380 more rows
```

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

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

```
## # A tibble: 3,929 x 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
## # ... with 3,919 more rows
```