Exercise of Data Transform

Jamia Begum

2022-10-13

Exercises (data transform) : 1. Using flights dataset, find all flights that a) Had an arrival delay of two or more hours b)Flew to Houston (IAH or HOU) c) Arrived more than two hours late, but didn't leave late d) Were delayed by at least an hour, but made up over 30 minutes in flight

```
library(tidyverse)
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.2 --
## v ggplot2 3.3.6      v purrr   0.3.5
## v tibble  3.1.8      v dplyr   1.0.10
## v tidyr   1.2.1      v stringr 1.4.1
## v readr   2.1.3      v forcats 0.5.2
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```
library("nycflights13")
library("dplyr")
flights
```

```
## # A tibble: 336,776 x 19
##     year month   day dep_time sched_de~1 dep_d~2 arr_t~3 sched~4 arr_d~5 carrier
##    <int> <int> <int>    <int>      <int>   <dbl>   <int>   <int>   <dbl> <chr>
##  1  2013     1     1      517        515       2     830     819      11 UA
##  2  2013     1     1      533        529       4     850     830      20 UA
##  3  2013     1     1      542        540       2     923     850      33 AA
##  4  2013     1     1      544        545      -1    1004    1022     -18 B6
##  5  2013     1     1      554        600      -6     812     837     -25 DL
##  6  2013     1     1      554        558      -4     740     728      12 UA
##  7  2013     1     1      555        600      -5     913     854      19 B6
##  8  2013     1     1      557        600      -3     709     723     -14 EV
##  9  2013     1     1      557        600      -3     838     846      -8 B6
## 10  2013     1     1      558        600      -2     753     745       8 AA
## # ... with 336,766 more rows, 9 more variables: flight <int>, tailnum <chr>,
## #   origin <chr>, dest <chr>, air_time <dbl>, distance <dbl>, hour <dbl>,
## #   minute <dbl>, time_hour <dttm>, and abbreviated variable names
## #   1: sched_dep_time, 2: dep_delay, 3: arr_time, 4: sched_arr_time,
## #   5: arr_delay
```

```
filter(flights, arr_delay >= 120)
```

```
## # A tibble: 10,200 x 19
##     year month   day dep_time sched_de~1 dep_d~2 arr_t~3 sched~4 arr_d~5 carrier
##    <int> <int> <int>    <int>      <int>   <dbl>   <int>   <int>   <dbl> <chr>
##  1  2013     1     1      811        630     101    1047     830     137 MQ
##  2  2013     1     1      848       1835     853    1001    1950     851 MQ
##  3  2013     1     1      957        733     144    1056     853     123 UA
##  4  2013     1     1     1114        900     134    1447    1222     145 UA
##  5  2013     1     1     1505       1310     115    1638    1431     127 EV
##  6  2013     1     1     1525       1340     105    1831    1626     125 B6
##  7  2013     1     1     1549       1445      64    1912    1656     136 EV
##  8  2013     1     1     1558       1359     119    1718    1515     123 EV
##  9  2013     1     1     1732       1630      62    2028    1825     123 EV
## 10  2013     1     1     1803       1620     103    2008    1750     138 MQ
## # ... with 10,190 more rows, 9 more variables: flight <int>, tailnum <chr>,
## #   origin <chr>, dest <chr>, air_time <dbl>, distance <dbl>, hour <dbl>,
## #   minute <dbl>, time_hour <dttm>, and abbreviated variable names
## #   1: sched_dep_time, 2: dep_delay, 3: arr_time, 4: sched_arr_time,
## #   5: arr_delay
```

```
select(flights,dest)
```

```
## # A tibble: 336,776 x 1
##    dest
##    <chr>
##  1 IAH
##  2 IAH
##  3 MIA
##  4 BQN
##  5 ATL
##  6 ORD
##  7 FLL
##  8 IAD
##  9 MCO
## 10 ORD
## # ... with 336,766 more rows
```

```
filter(flights, dest == "IAH" | dest == "HOU")
```

```
## # A tibble: 9,313 x 19
##     year month   day dep_time sched_de~1 dep_d~2 arr_t~3 sched~4 arr_d~5 carrier
##    <int> <int> <int>    <int>      <int>   <dbl>   <int>   <int>   <dbl> <chr>
##  1  2013     1     1      517        515       2     830     819      11 UA
##  2  2013     1     1      533        529       4     850     830      20 UA
##  3  2013     1     1      623        627      -4     933     932       1 UA
##  4  2013     1     1      728        732      -4    1041    1038       3 UA
##  5  2013     1     1      739        739       0    1104    1038      26 UA
##  6  2013     1     1      908        908       0    1228    1219       9 UA
##  7  2013     1     1     1028       1026       2    1350    1339      11 UA
##  8  2013     1     1     1044       1045      -1    1352    1351       1 UA
##  9  2013     1     1     1114        900     134    1447    1222     145 UA
## 10  2013     1     1     1205       1200       5    1503    1505      -2 UA
## # ... with 9,303 more rows, 9 more variables: flight <int>, tailnum <chr>,
## #   origin <chr>, dest <chr>, air_time <dbl>, distance <dbl>, hour <dbl>,
```

```
## #   minute <dbl>, time_hour <dttm>, and abbreviated variable names
## #   1: sched_dep_time, 2: dep_delay, 3: arr_time, 4: sched_arr_time,
## #   5: arr_delay
```

```
filter(flights, arr_delay > 120, dep_delay <= 0)
```

```
## # A tibble: 29 x 19
##     year month   day dep_time sched_de~1 dep_d~2 arr_t~3 sched~4 arr_d~5 carrier
##    <int> <int> <int>    <int>      <int>   <dbl>   <int>   <int>   <dbl> <chr>
##  1  2013     1    27     1419       1420      -1    1754    1550     124 MQ
##  2  2013    10     7     1350       1350       0    1736    1526     130 EV
##  3  2013    10     7     1357       1359      -2    1858    1654     124 AA
##  4  2013    10    16      657        700      -3    1258    1056     122 B6
##  5  2013    11     1      658        700      -2    1329    1015     194 VX
##  6  2013     3    18     1844       1847      -3      39    2219     140 UA
##  7  2013     4    17     1635       1640      -5    2049    1845     124 MQ
##  8  2013     4    18      558        600      -2    1149     850     179 AA
##  9  2013     4    18      655        700      -5    1213     950     143 AA
## 10  2013     5    22     1827       1830      -3    2217    2010     127 MQ
## # ... with 19 more rows, 9 more variables: flight <int>, tailnum <chr>,
## #   origin <chr>, dest <chr>, air_time <dbl>, distance <dbl>, hour <dbl>,
## #   minute <dbl>, time_hour <dttm>, and abbreviated variable names
## #   1: sched_dep_time, 2: dep_delay, 3: arr_time, 4: sched_arr_time,
## #   5: arr_delay
```

```
filter(flights, dep_delay >= 60, dep_delay - arr_delay > 30)
```

```
## # A tibble: 1,844 x 19
##     year month   day dep_time sched_de~1 dep_d~2 arr_t~3 sched~4 arr_d~5 carrier
##    <int> <int> <int>    <int>      <int>   <dbl>   <int>   <int>   <dbl> <chr>
##  1  2013     1     1     2205       1720     285      46    2040     246 AA
##  2  2013     1     1     2326       2130     116     131      18      73 B6
##  3  2013     1     3     1503       1221     162    1803    1555     128 UA
##  4  2013     1     3     1839       1700      99    2056    1950      66 AA
##  5  2013     1     3     1850       1745      65    2148    2120      28 AA
##  6  2013     1     3     1941       1759     102    2246    2139      67 UA
##  7  2013     1     3     1950       1845      65    2228    2227       1 B6
##  8  2013     1     3     2015       1915      60    2135    2111      24 9E
##  9  2013     1     3     2257       2000     177      45    2224     141 9E
## 10  2013     1     4     1917       1700     137    2135    1950     105 AA
## # ... with 1,834 more rows, 9 more variables: flight <int>, tailnum <chr>,
## #   origin <chr>, dest <chr>, air_time <dbl>, distance <dbl>, hour <dbl>,
## #   minute <dbl>, time_hour <dttm>, and abbreviated variable names
## #   1: sched_dep_time, 2: dep_delay, 3: arr_time, 4: sched_arr_time,
## #   5: arr_delay
```

2. Another useful dplyr filtering helper is between(). What does it do? Can you use it to simplify the code needed to answer the previous challenges?

between(x, left, right) gives values in a numeric vector bounded in a specified range (left, right)

Using it in the previous questions does not simplify the codes here.

```
filter(flights, between(arr_delay, 120, Inf) )
```

```
## # A tibble: 10,200 x 19
##     year month   day dep_time sched_de~1 dep_d~2 arr_t~3 sched~4 arr_d~5 carrier
##    <int> <int> <int>    <int>      <int>   <dbl>   <int>   <int>   <dbl> <chr>
##  1  2013     1     1      811        630     101    1047     830     137 MQ
##  2  2013     1     1      848       1835     853    1001    1950     851 MQ
##  3  2013     1     1      957        733     144    1056     853     123 UA
##  4  2013     1     1     1114        900     134    1447    1222     145 UA
##  5  2013     1     1     1505       1310     115    1638    1431     127 EV
##  6  2013     1     1     1525       1340     105    1831    1626     125 B6
##  7  2013     1     1     1549       1445      64    1912    1656     136 EV
##  8  2013     1     1     1558       1359     119    1718    1515     123 EV
##  9  2013     1     1     1732       1630      62    2028    1825     123 EV
## 10  2013     1     1     1803       1620     103    2008    1750     138 MQ
## # ... with 10,190 more rows, 9 more variables: flight <int>, tailnum <chr>,
## #   origin <chr>, dest <chr>, air_time <dbl>, distance <dbl>, hour <dbl>,
## #   minute <dbl>, time_hour <dttm>, and abbreviated variable names
## #   1: sched_dep_time, 2: dep_delay, 3: arr_time, 4: sched_arr_time,
## #   5: arr_delay
```

```
filter(flights, between(arr_delay, 120, Inf)& between(dep_delay,-Inf,0) )
```

```
## # A tibble: 29 x 19
##     year month   day dep_time sched_de~1 dep_d~2 arr_t~3 sched~4 arr_d~5 carrier
##    <int> <int> <int>    <int>      <int>   <dbl>   <int>   <int>   <dbl> <chr>
##  1  2013     1    27     1419       1420      -1    1754    1550     124 MQ
##  2  2013    10     7     1350       1350       0    1736    1526     130 EV
##  3  2013    10     7     1357       1359      -2    1858    1654     124 AA
##  4  2013    10    16      657        700      -3    1258    1056     122 B6
##  5  2013    11     1      658        700      -2    1329    1015     194 VX
##  6  2013     3    18     1844       1847      -3      39    2219     140 UA
##  7  2013     4    17     1635       1640      -5    2049    1845     124 MQ
##  8  2013     4    18      558        600      -2    1149     850     179 AA
##  9  2013     4    18      655        700      -5    1213     950     143 AA
## 10  2013     5    22     1827       1830      -3    2217    2010     127 MQ
## # ... with 19 more rows, 9 more variables: flight <int>, tailnum <chr>,
## #   origin <chr>, dest <chr>, air_time <dbl>, distance <dbl>, hour <dbl>,
## #   minute <dbl>, time_hour <dttm>, and abbreviated variable names
## #   1: sched_dep_time, 2: dep_delay, 3: arr_time, 4: sched_arr_time,
## #   5: arr_delay
```

```
filter(flights, between(dep_delay, 60, Inf)& between(dep_delay- arr_delay,31,Inf) )
```

```
## # A tibble: 1,844 x 19
##     year month   day dep_time sched_de~1 dep_d~2 arr_t~3 sched~4 arr_d~5 carrier
##    <int> <int> <int>    <int>      <int>   <dbl>   <int>   <int>   <dbl> <chr>
##  1  2013     1     1     2205       1720     285      46    2040     246 AA
##  2  2013     1     1     2326       2130     116     131      18      73 B6
##  3  2013     1     3     1503       1221     162    1803    1555     128 UA
##  4  2013     1     3     1839       1700      99    2056    1950      66 AA
```

```
##  5  2013     1     3     1850       1745      65    2148    2120      28 AA
##  6  2013     1     3     1941       1759     102    2246    2139      67 UA
##  7  2013     1     3     1950       1845      65    2228    2227       1 B6
##  8  2013     1     3     2015       1915      60    2135    2111      24 9E
##  9  2013     1     3     2257       2000     177      45    2224     141 9E
## 10  2013     1     4     1917       1700     137    2135    1950     105 AA
## # ... with 1,834 more rows, 9 more variables: flight <int>, tailnum <chr>,
## #   origin <chr>, dest <chr>, air_time <dbl>, distance <dbl>, hour <dbl>,
## #   minute <dbl>, time_hour <dttm>, and abbreviated variable names
## #   1: sched_dep_time, 2: dep_delay, 3: arr_time, 4: sched_arr_time,
## #   5: arr_delay
```

3. Sort flights to find the fastest flights.

fastest flights mean having highest average speed calculated by distance/air time

```
arrange(flights, desc(distance / air_time))
```

```
## # A tibble: 336,776 x 19
##     year month   day dep_time sched_de~1 dep_d~2 arr_t~3 sched~4 arr_d~5 carrier
##    <int> <int> <int>    <int>      <int>   <dbl>   <int>   <int>   <dbl> <chr>
##  1  2013     5    25     1709       1700       9    1923    1937     -14 DL
##  2  2013     7     2     1558       1513      45    1745    1719      26 EV
##  3  2013     5    13     2040       2025      15    2225    2226      -1 EV
##  4  2013     3    23     1914       1910       4    2045    2043       2 EV
##  5  2013     1    12     1559       1600      -1    1849    1917     -28 DL
##  6  2013    11    17      650        655      -5    1059    1150     -51 DL
##  7  2013     2    21     2355       2358      -3     412     438     -26 B6
##  8  2013    11    17      759        800      -1    1212    1255     -43 AA
##  9  2013    11    16     2003       1925      38      17      36     -19 DL
## 10  2013    11    16     2349       2359     -10     402     440     -38 B6
## # ... with 336,766 more rows, 9 more variables: flight <int>, tailnum <chr>,
## #   origin <chr>, dest <chr>, air_time <dbl>, distance <dbl>, hour <dbl>,
## #   minute <dbl>, time_hour <dttm>, and abbreviated variable names
## #   1: sched_dep_time, 2: dep_delay, 3: arr_time, 4: sched_arr_time,
## #   5: arr_delay
```

4. Create a new data frame having variables with the dep string.

```
dep_str=select(flights, contains("dep"))
dep_str
```

```
## # A tibble: 336,776 x 3
##    dep_time sched_dep_time dep_delay
##       <int>          <int>     <dbl>
##  1      517            515         2
##  2      533            529         4
##  3      542            540         2
##  4      544            545        -1
##  5      554            600        -6
##  6      554            558        -4
##  7      555            600        -5
```

```
##  8      557            600        -3
##  9      557            600        -3
## 10      558            600        -2
## # ... with 336,766 more rows
```

5. Create a new data frame having the hour and minute of depature (Hint: information is in the variable dep_time with format HHMM or HMM. Use %/% or %% when appropriate)

we need to split hour digits and minute digits from dep_time. ## %/% gives quotient so gives the hour digits here ## %% gives remainder so gives the minute digits here

```
dp_flights<-mutate(flights,dep_hour=dep_time %/% 100, dep_min = dep_time %% 100)
select(dp_flights,dep_time,dep_hour,dep_min)
```

```
## # A tibble: 336,776 x 3
##    dep_time dep_hour dep_min
##       <int>    <dbl>   <dbl>
##  1      517        5      17
##  2      533        5      33
##  3      542        5      42
##  4      544        5      44
##  5      554        5      54
##  6      554        5      54
##  7      555        5      55
##  8      557        5      57
##  9      557        5      57
## 10      558        5      58
## # ... with 336,766 more rows
```

6. Create a summary of each airline (variable carrier) describing the total number of flights, the average, median, IQR of arr_delay

```
flights %>%
  group_by(carrier) %>% summarise( count = n(), mean_delay = mean(arr_delay, na.rm = TRUE)
  ,median_delay = median(arr_delay, na.rm = TRUE),
  IQR_delay = IQR(arr_delay, na.rm = TRUE))
```

```
## # A tibble: 16 x 5
##    carrier count mean_delay median_delay IQR_delay
##    <chr>   <int>      <dbl>        <dbl>     <dbl>
##  1 9E      18460      7.38            -7      36
##  2 AA      32729      0.364           -9      29
##  3 AS        714     -9.93           -17      34
##  4 B6      54635      9.46            -3      31
##  5 DL      48110      1.64            -8      28
##  6 EV      54173     15.8             -1      40
##  7 F9        685     21.9              6      40
##  8 FL       3260     20.1              5      31
##  9 HA        342     -6.92           -13      30.5
## 10 MQ      26397     10.8             -1      31
## 11 OO         32     11.9             -7      22
## 12 UA      58665      3.56            -6      30
```

```
## 13 US      20536      2.13           -6      23
## 14 VX       5162      1.76           -9      31
## 15 WN      12275      9.65           -3      30
## 16 YV        601     15.6            -2      40.2
```

Exercise for Tibble

Jamia Begum

2022-10-12

Exercise:1 How can you know whether an object is a tibble? (Hint: try printing mtcars, which is a regular data frame).

```
library(tidyverse)
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.2 --
## v ggplot2 3.3.6      v purrr   0.3.5
## v tibble  3.1.8      v dplyr   1.0.10
## v tidyr   1.2.1      v stringr 1.4.1
## v readr   2.1.3      v forcats 0.5.2
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```
is_tibble(mtcars)
```

```
## [1] FALSE
```

```
mtcars
```

```
##                     mpg cyl  disp  hp drat    wt  qsec vs am gear carb
## Mazda RX4           21.0   6 160.0 110 3.90 2.620 16.46  0  1    4    4
## Mazda RX4 Wag       21.0   6 160.0 110 3.90 2.875 17.02  0  1    4    4
## Datsun 710          22.8   4 108.0  93 3.85 2.320 18.61  1  1    4    1
## Hornet 4 Drive      21.4   6 258.0 110 3.08 3.215 19.44  1  0    3    1
## Hornet Sportabout   18.7   8 360.0 175 3.15 3.440 17.02  0  0    3    2
## Valiant             18.1   6 225.0 105 2.76 3.460 20.22  1  0    3    1
## Duster 360          14.3   8 360.0 245 3.21 3.570 15.84  0  0    3    4
## Merc 240D           24.4   4 146.7  62 3.69 3.190 20.00  1  0    4    2
## Merc 230            22.8   4 140.8  95 3.92 3.150 22.90  1  0    4    2
## Merc 280            19.2   6 167.6 123 3.92 3.440 18.30  1  0    4    4
## Merc 280C           17.8   6 167.6 123 3.92 3.440 18.90  1  0    4    4
## Merc 450SE          16.4   8 275.8 180 3.07 4.070 17.40  0  0    3    3
## Merc 450SL          17.3   8 275.8 180 3.07 3.730 17.60  0  0    3    3
## Merc 450SLC         15.2   8 275.8 180 3.07 3.780 18.00  0  0    3    3
## Cadillac Fleetwood  10.4   8 472.0 205 2.93 5.250 17.98  0  0    3    4
## Lincoln Continental 10.4   8 460.0 215 3.00 5.424 17.82  0  0    3    4
## Chrysler Imperial   14.7   8 440.0 230 3.23 5.345 17.42  0  0    3    4
## Fiat 128            32.4   4  78.7  66 4.08 2.200 19.47  1  1    4    1
## Honda Civic         30.4   4  75.7  52 4.93 1.615 18.52  1  1    4    2
```

```
## Toyota Corolla      33.9   4  71.1  65 4.22 1.835 19.90  1  1    4    1
## Toyota Corona       21.5   4 120.1  97 3.70 2.465 20.01  1  0    3    1
## Dodge Challenger    15.5   8 318.0 150 2.76 3.520 16.87  0  0    3    2
## AMC Javelin         15.2   8 304.0 150 3.15 3.435 17.30  0  0    3    2
## Camaro Z28          13.3   8 350.0 245 3.73 3.840 15.41  0  0    3    4
## Pontiac Firebird    19.2   8 400.0 175 3.08 3.845 17.05  0  0    3    2
## Fiat X1-9           27.3   4  79.0  66 4.08 1.935 18.90  1  1    4    1
## Porsche 914-2       26.0   4 120.3  91 4.43 2.140 16.70  0  1    5    2
## Lotus Europa        30.4   4  95.1 113 3.77 1.513 16.90  1  1    5    2
## Ford Pantera L      15.8   8 351.0 264 4.22 3.170 14.50  0  1    5    4
## Ferrari Dino        19.7   6 145.0 175 3.62 2.770 15.50  0  1    5    6
## Maserati Bora       15.0   8 301.0 335 3.54 3.570 14.60  0  1    5    8
## Volvo 142E          21.4   4 121.0 109 4.11 2.780 18.60  1  1    4    2
```

```
mtcars.tib<-as_tibble(mtcars)
mtcars.tib
```

```
## # A tibble: 32 x 11
##      mpg   cyl  disp    hp  drat    wt  qsec    vs    am  gear  carb
##    <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>
##  1  21       6  160    110  3.9   2.62  16.5     0     1     4     4
##  2  21       6  160    110  3.9   2.88  17.0     0     1     4     4
##  3  22.8     4  108     93  3.85  2.32  18.6     1     1     4     1
##  4  21.4     6  258    110  3.08  3.22  19.4     1     0     3     1
##  5  18.7     8  360    175  3.15  3.44  17.0     0     0     3     2
##  6  18.1     6  225    105  2.76  3.46  20.2     1     0     3     1
##  7  14.3     8  360    245  3.21  3.57  15.8     0     0     3     4
##  8  24.4     4  147.    62  3.69  3.19  20       1     0     4     2
##  9  22.8     4  141.    95  3.92  3.15  22.9     1     0     4     2
## 10  19.2     6  168.   123  3.92  3.44  18.3     1     0     4     4
## # ... with 22 more rows
```

```
is_tibble(mtcars.tib)
```

```
## [1] TRUE
```

As seen on the above examples,the printing pattern is different for tibble than dataframe. Tibble shows only the first 10 rows, and all the columns that fit on screen by default. Also,it gives number of rows and columns and prints data type for each column.Another way to check whether the object is a tibble or not is to use is_tibble command which returns TRUE if it's a tibble.

Exercise:2 If you have the name of a variable stored in an object, e.g. var <- "mpg", how can you extract the reference variable from a tibble?

```
var<-"mpg"

mtcars.tib$var
```

```
## Warning: Unknown or uninitialised column: `var`.
```

```
## NULL
```

```
mtcars.tib[[var]]
```

```
##  [1] 21.0 21.0 22.8 21.4 18.7 18.1 14.3 24.4 22.8 19.2 17.8 16.4 17.3 15.2 10.4
## [16] 10.4 14.7 32.4 30.4 33.9 21.5 15.5 15.2 13.3 19.2 27.3 26.0 30.4 15.8 19.7
## [31] 15.0 21.4
```

Here, $ searchs for the column name var so gives unknown result.

Exercise:3 What option controls how many additional column names are printed at the footer of a tibble?

```
print(nycflights13::flights)
```

```
## # A tibble: 336,776 x 19
##     year month   day dep_time sched_de~1 dep_d~2 arr_t~3 sched~4 arr_d~5 carrier
##    <int> <int> <int>    <int>      <int>   <dbl>   <int>   <int>   <dbl> <chr>
##  1  2013     1     1      517        515       2     830     819      11 UA
##  2  2013     1     1      533        529       4     850     830      20 UA
##  3  2013     1     1      542        540       2     923     850      33 AA
##  4  2013     1     1      544        545      -1    1004    1022     -18 B6
##  5  2013     1     1      554        600      -6     812     837     -25 DL
##  6  2013     1     1      554        558      -4     740     728      12 UA
##  7  2013     1     1      555        600      -5     913     854      19 B6
##  8  2013     1     1      557        600      -3     709     723     -14 EV
##  9  2013     1     1      557        600      -3     838     846      -8 B6
## 10  2013     1     1      558        600      -2     753     745       8 AA
## # ... with 336,766 more rows, 9 more variables: flight <int>, tailnum <chr>,
## #   origin <chr>, dest <chr>, air_time <dbl>, distance <dbl>, hour <dbl>,
## #   minute <dbl>, time_hour <dttm>, and abbreviated variable names
## #   1: sched_dep_time, 2: dep_delay, 3: arr_time, 4: sched_arr_time,
## #   5: arr_delay
```

```
print(nycflights13::flights,n_extra=3)
```

```
## Warning: The ‘n_extra‘ argument of ‘print()‘ is deprecated as of pillar 1.6.2.
## i Please use the ‘max_extra_cols‘ argument instead.
```

```
## # A tibble: 336,776 x 19
##     year month   day dep_time sched_de~1 dep_d~2 arr_t~3 sched~4 arr_d~5 carrier
##    <int> <int> <int>    <int>      <int>   <dbl>   <int>   <int>   <dbl> <chr>
##  1  2013     1     1      517        515       2     830     819      11 UA
##  2  2013     1     1      533        529       4     850     830      20 UA
##  3  2013     1     1      542        540       2     923     850      33 AA
##  4  2013     1     1      544        545      -1    1004    1022     -18 B6
##  5  2013     1     1      554        600      -6     812     837     -25 DL
##  6  2013     1     1      554        558      -4     740     728      12 UA
##  7  2013     1     1      555        600      -5     913     854      19 B6
##  8  2013     1     1      557        600      -3     709     723     -14 EV
##  9  2013     1     1      557        600      -3     838     846      -8 B6
## 10  2013     1     1      558        600      -2     753     745       8 AA
## # ... with 336,766 more rows, 9 more variables: flight <int>, tailnum <chr>,
## #   origin <chr>, ..., and abbreviated variable names 1: sched_dep_time,
## #   2: dep_delay, 3: arr_time, 4: sched_arr_time, 5: arr_delay
```

By default all the column information are printed at the footer.To customize the number of column names printed in the footer we can use n_extra.

Exercise:4 Practice creating new variables in the following data frame

tbl <- tibble(age = c(14, 18, 22, 12, 16, 19, 21, 24), chol = c(172, 180, 185, 170, 175, 188, 190, 192), sex = c("male", "male", "female", "female", "female", "male", "male", "male")) by: + Extracting the variable called sex. + Plotting a scatterplot of age vs chol. + Creating a new column called chol2 which is chol to the power of 2. + Rename the columns to one, two and three.

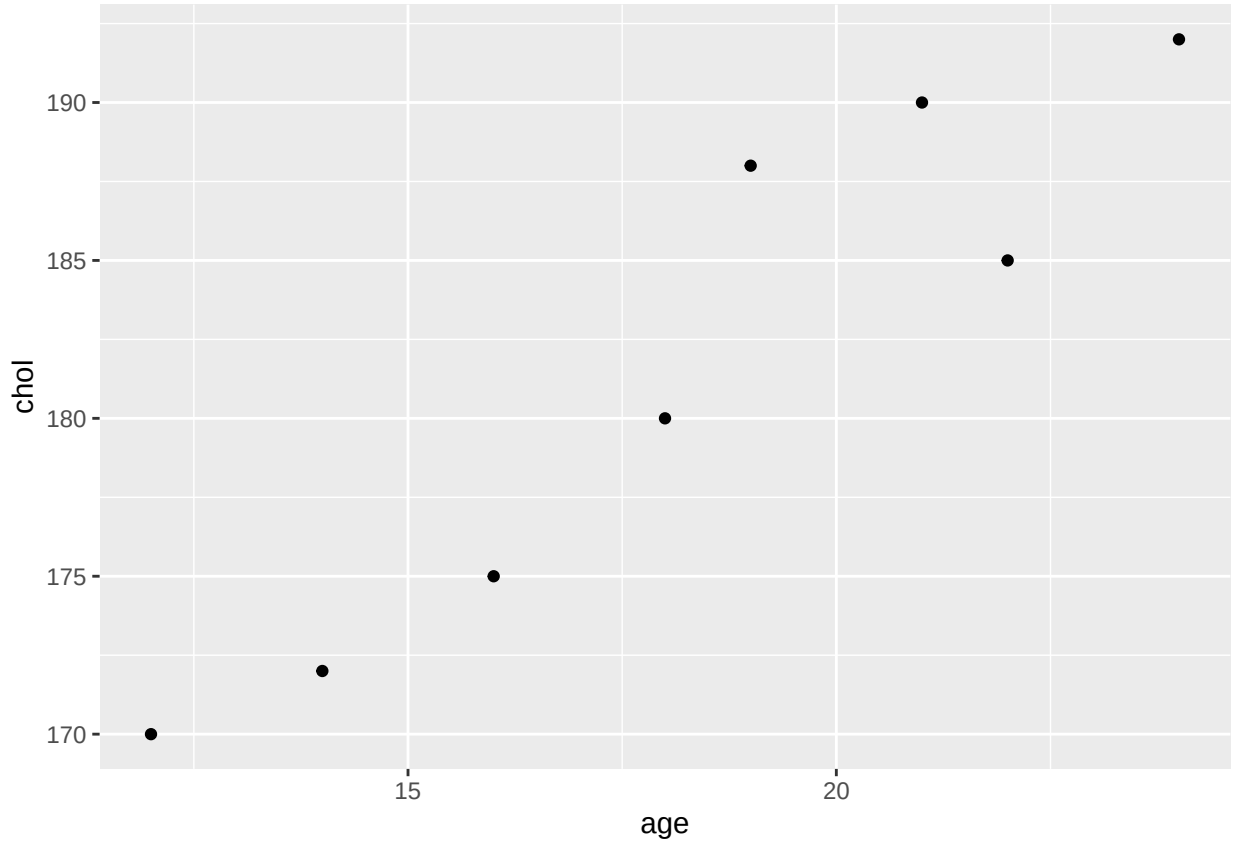
```
library(tibble)
tbl <- tibble( age = c(14, 18, 22, 12, 16, 19, 21, 24),
               chol = c(172, 180, 185, 170, 175, 188,190, 192),
      sex = c("male", "male","female",
              "female", "female", "male", "male", "male" ))
tbl
```

```
## # A tibble: 8 x 3
##     age  chol sex
##   <dbl> <dbl> <chr>
## 1    14   172 male
## 2    18   180 male
## 3    22   185 female
## 4    12   170 female
## 5    16   175 female
## 6    19   188 male
## 7    21   190 male
## 8    24   192 male
```

```
tbl$sex
```

```
## [1] "male"   "male"   "female" "female" "female" "male"   "male"   "male"
```

```
library(tidyverse)
ggplot(tbl, aes(age, chol)) + geom_point()
```




```
mutate(tbl,chol2= chol^2)
```

```
## # A tibble: 8 x 4
##     age  chol sex    chol2
##   <dbl> <dbl> <chr>  <dbl>
## 1    14   172 male   29584
## 2    18   180 male   32400
## 3    22   185 female 34225
## 4    12   170 female 28900
## 5    16   175 female 30625
## 6    19   188 male   35344
## 7    21   190 male   36100
## 8    24   192 male   36864
```

```
transmute(tbl,one="age",two="chol",three="sex")
```

```
## # A tibble: 8 x 3
##   one   two   three
##   <chr> <chr> <chr>
## 1 age   chol  sex
## 2 age   chol  sex
## 3 age   chol  sex
## 4 age   chol  sex
## 5 age   chol  sex
## 6 age   chol  sex
```

```
## 7 age   chol  sex
## 8 age   chol  sex
```