Lab 3: Wrangling sales data

Minsu Kang and Minsung Kim

2022-06-30

Exercise 1

```
select(coffeeshop, !cogs)
```

```
## # A tibble: 1,844 x 8
##    date       market  product_line product      sales state total_expenses type
##    <date>     <chr>   <chr>        <chr>        <dbl> <chr>          <dbl> <chr>
##  1 2012-01-01 Central Beans        Decaf Irish~   234 Colo~             38 Decaf
##  2 2012-01-01 Central Beans        Decaf Irish~   234 Illi~             52 Decaf
##  3 2012-01-01 Central Beans        Decaf Espre~   180 Colo~             55 Decaf
##  4 2012-01-01 Central Beans        Decaf Espre~   456 Illi~             88 Decaf
##  5 2012-01-01 Central Beans        Decaf Espre~   130 Ohio              56 Decaf
##  6 2012-01-01 East    Beans        Decaf Irish~   200 Flor~             49 Decaf
##  7 2012-01-01 East    Beans        Decaf Espre~   180 Flor~             53 Decaf
##  8 2012-01-01 South   Beans        Decaf Irish~   190 Texas             39 Decaf
##  9 2012-01-01 South   Beans        Decaf Espre~   134 Texas             26 Decaf
## 10 2012-01-01 West    Beans        Decaf Espre~   546 Cali~            109 Decaf
## # ... with 1,834 more rows
```

It extract column without 'cogs' variable.

```
select(coffeeshop, starts_with('prod'))
```

```
## # A tibble: 1,844 x 2
##    product_line product
##    <chr>        <chr>
##  1 Beans        Decaf Irish Cream
##  2 Beans        Decaf Irish Cream
##  3 Beans        Decaf Espresso
##  4 Beans        Decaf Espresso
##  5 Beans        Decaf Espresso
##  6 Beans        Decaf Irish Cream
##  7 Beans        Decaf Espresso
##  8 Beans        Decaf Irish Cream
##  9 Beans        Decaf Espresso
## 10 Beans        Decaf Espresso
## # ... with 1,834 more rows
```

It makes the chart starts from ’product’ variable.

```
select(coffeeshop, contains("pe"))
```

```
## # A tibble: 1,844 x 2
##    total_expenses type
##             <dbl> <chr>
##  1             38 Decaf
##  2             52 Decaf
##  3             55 Decaf
##  4             88 Decaf
##  5             56 Decaf
##  6             49 Decaf
##  7             53 Decaf
##  8             39 Decaf
##  9             26 Decaf
## 10            109 Decaf
## # ... with 1,834 more rows
```

It sort out the variable that contains letter "pe". The function extract the colomns, which contains "pe".

```
select(coffeeshop, caffeination = type)
```

```
## # A tibble: 1,844 x 1
##    caffeination
##    <chr>
##  1 Decaf
##  2 Decaf
##  3 Decaf
##  4 Decaf
##  5 Decaf
##  6 Decaf
##  7 Decaf
##  8 Decaf
##  9 Decaf
## 10 Decaf
## # ... with 1,834 more rows
```

The caffeination = type part, rename the name of ’type’ column to ’caffeination’.

Exercise 2

```
coffeeshop %>% select(caffeination = type)
```

```
## # A tibble: 1,844 x 1
##    caffeination
##    <chr>       
##  1 Decaf       
##  2 Decaf       
##  3 Decaf       
##  4 Decaf       
##  5 Decaf       
##  6 Decaf       
##  7 Decaf       
##  8 Decaf       
##  9 Decaf       
## 10 Decaf       
## # ... with 1,834 more rows
```

```
coffeeshop %>% select(starts_with('prod')) %>%
            filter(product == "Darjeelisng")
```

```
## # A tibble: 0 x 2
## # ... with 2 variables: product_line <chr>, product <chr>
```

Exercise 3

```
coffeeshop %>%
  mutate(product_profit = sales - (cogs + total_expenses))
```

```
## # A tibble: 1,844 x 10
##     cogs date       market product_line product sales state total_expenses type 
##    <dbl> <date>     <chr>  <chr>        <chr>   <dbl> <chr>          <dbl> <chr>
##  1    95 2012-01-01 Centr~ Beans        Decaf ~   234 Colo~             38 Decaf
##  2    95 2012-01-01 Centr~ Beans        Decaf ~   234 Illi~             52 Decaf
##  3    72 2012-01-01 Centr~ Beans        Decaf ~   180 Colo~             55 Decaf
##  4   228 2012-01-01 Centr~ Beans        Decaf ~   456 Illi~             88 Decaf
##  5    58 2012-01-01 Centr~ Beans        Decaf ~   130 Ohio              56 Decaf
##  6    84 2012-01-01 East   Beans        Decaf ~   200 Flor~             49 Decaf
##  7    77 2012-01-01 East   Beans        Decaf ~   180 Flor~             53 Decaf
##  8    83 2012-01-01 South  Beans        Decaf ~   190 Texas             39 Decaf
##  9    54 2012-01-01 South  Beans        Decaf ~   134 Texas             26 Decaf
## 10   234 2012-01-01 West   Beans        Decaf ~   546 Cali~            109 Decaf
## # ... with 1,834 more rows, and 1 more variable: product_profit <dbl>
```

```r
coffeeshop %>%
  mutate(sales - (cogs + total_expenses))
```

```
## # A tibble: 1,844 x 10
##     cogs date       market product_line product sales state total_expenses type
##    <dbl> <date>     <chr>  <chr>        <chr>   <dbl> <chr>          <dbl> <chr>
##  1    95 2012-01-01 Centr~ Beans        Decaf ~   234 Colo~             38 Decaf
##  2    95 2012-01-01 Centr~ Beans        Decaf ~   234 Illi~             52 Decaf
##  3    72 2012-01-01 Centr~ Beans        Decaf ~   180 Colo~             55 Decaf
##  4   228 2012-01-01 Centr~ Beans        Decaf ~   456 Illi~             88 Decaf
##  5    58 2012-01-01 Centr~ Beans        Decaf ~   130 Ohio              56 Decaf
##  6    84 2012-01-01 East   Beans        Decaf ~   200 Flor~             49 Decaf
##  7    77 2012-01-01 East   Beans        Decaf ~   180 Flor~             53 Decaf
##  8    83 2012-01-01 South  Beans        Decaf ~   190 Texas             39 Decaf
##  9    54 2012-01-01 South  Beans        Decaf ~   134 Texas             26 Decaf
## 10   234 2012-01-01 West   Beans        Decaf ~   546 Cali~            109 Decaf
## # ... with 1,834 more rows, and 1 more variable:
## #   ‘sales - (cogs + total_expenses)‘ <dbl>
```

The upper code chunk assign sales - (cogs + total_expenses)) to product_profit.
So mutate function create new column that name is "product_profit".
However, second doesn’t designate the specific column name.
Therefore, sales - (cogs + total_expenses)) itself became a name of the column.

Exercise 4

```r
coffeeshop %>%
  mutate(product_profit = sales - (cogs + total_expenses),
         expiration_date = date + 2)
```

```
## # A tibble: 1,844 x 11
##     cogs date       market product_line product sales state total_expenses type
##    <dbl> <date>     <chr>  <chr>        <chr>   <dbl> <chr>          <dbl> <chr>
##  1    95 2012-01-01 Centr~ Beans        Decaf ~   234 Colo~             38 Decaf
##  2    95 2012-01-01 Centr~ Beans        Decaf ~   234 Illi~             52 Decaf
##  3    72 2012-01-01 Centr~ Beans        Decaf ~   180 Colo~             55 Decaf
##  4   228 2012-01-01 Centr~ Beans        Decaf ~   456 Illi~             88 Decaf
##  5    58 2012-01-01 Centr~ Beans        Decaf ~   130 Ohio              56 Decaf
##  6    84 2012-01-01 East   Beans        Decaf ~   200 Flor~             49 Decaf
##  7    77 2012-01-01 East   Beans        Decaf ~   180 Flor~             53 Decaf
##  8    83 2012-01-01 South  Beans        Decaf ~   190 Texas             39 Decaf
##  9    54 2012-01-01 South  Beans        Decaf ~   134 Texas             26 Decaf
## 10   234 2012-01-01 West   Beans        Decaf ~   546 Cali~            109 Decaf
## # ... with 1,834 more rows, and 2 more variables: product_profit <dbl>,
## #   expiration_date <date>
```

The expiration_date column has been added. Added number 2, two days added to the 'date' variables.

Exercise 5

```
coffeeshop_updated <- mutate(
  coffeeshop,
  product_profit = sales - (cogs + total_expenses)
 )
```

It makes permanent change in the environment window, you can see coffeeshop_updated data is uploaded with now variable, product_profit. This is because we assign to the new, updated date.

Exercise 6

```
coffeeshop_updated %>%
  group_by(product) %>%
  summarize(avg_profit = mean(product_profit))
```

```
## # A tibble: 13 x 2
##    product            avg_profit
##    <chr>                   <dbl>
##  1 Amaretto                 40.5
##  2 Caffe Latte              51.0
##  3 Caffe Mocha              53.0
##  4 Chamomile                61.7
##  5 Colombian                99.1
##  6 Darjeeling               65.8
##  7 Decaf Espresso           58.9
##  8 Decaf Irish Cream        49.7
##  9 Earl Grey                77.8
## 10 Green Tea                43.0
## 11 Lemon                    60.4
## 12 Mint                     65.2
## 13 Regular Espresso        174.
```

Amaretto has the highest average profit.

```
coffeeshop %>%
  group_by(market) %>%
  summarize(
    total_profit = sum(sales - (cogs + total_expenses)),
    total_sales = sum(sales),
    profit_margin = total_profit / total_sales
  ) %>%
  arrange(desc(profit_margin))
```

```
## # A tibble: 4 x 4
##   market  total_profit total_sales profit_margin
##   <chr>          <dbl>       <dbl>         <dbl>
## 1 East           29031       79894         0.363
## 2 Central        38873      122112         0.318
## 3 West           37681      123466         0.305
## 4 South          13703       47058         0.291
```

East market has the greatest profit margin.

Exercise 7

```
coffeeshop %>%
  group_by(product_line, type) %>%
  summarize(
    total_profit = sum(sales - (cogs + total_expenses)),
    total_sales = sum(sales),
    profit_margin = total_profit / total_sales
    ) %>%
    arrange(desc(total_profit))
```

```
## `summarise()` has grouped output by 'product_line'. You can override using the
## `.groups` argument.
```

```
## # A tibble: 4 x 5
##   product_line type    total_profit total_sales profit_margin
##   <chr>        <chr>          <dbl>       <dbl>         <dbl>
## 1 Beans        Regular        44748      135922         0.329
## 2 Leaves       Decaf          28491       93280         0.305
## 3 Leaves       Regular        26374       79325         0.332
## 4 Beans        Decaf          19675       64003         0.307
```

The leaves has the highest profit_margin.
The beans has the greatest total_profit.
The Regular has higher profit_margin than Decaf.