Lab Seven

Noah Gallego

2024-09-29

Advanced Dplyr Assignment with Multiple Excel Files

Problem 1: Data Manipulation (PeopleData)

```
##
## Attaching package: 'dplyr'

## The following objects are masked from 'package:stats':
##
##     filter, lag

## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union

## # A tibble: 2 × 3
##   sex    avg_age total_count
##   <chr>    <dbl>       <int>
## 1 Female    68.1          13
## 2 Male      68.1          11
```

Problem 2: TransactionsData and EmployeeData Analysis

```
head(transaction_data)

## # A tibble: 6 × 4
##   customer_id transaction_date    amount category
##         <dbl> <dttm>               <dbl> <chr>
## 1           8 2022-03-30 00:00:00  459.  Electronics
## 2          11 2022-10-19 00:00:00  116.  Food
## 3          21 2022-09-20 00:00:00  279.  Electronics
## 4           7 2022-03-09 00:00:00  363.  Clothes
## 5          23 2022-04-30 00:00:00  205.  Clothes
## 6           8 2022-08-20 00:00:00   56.1 Electronics

# Group Transaction Data By Category & Sum
transaction_data %>%
  group_by(category) %>%
  summarize(
    total_amount = sum(amount, na.rm = TRUE),
    average_amount = mean(amount, na.rm = TRUE)
  )
```

```
## # A tibble: 4 × 3
##   category    total_amount average_amount
##   <chr>              <dbl>          <dbl>
## 1 Clothes           12085.           247.
## 2 Electronics       11166.           228.
## 3 Food              13018.           241.
## 4 Travel            11954.           249.
```

```
# Join the Transaction Data w/ Employee Data
#head(employee_data)
#merged_data = transaction_data %>%
# left_join(employee_data, by = "employee_id")

# There is no common column??
```

Problem 3: Advanced Grouping and Ranking

```
people_data$age_group = df1$age_group

# Group By Age and Race
grouped_data = people_data %>%
  group_by(age_group, race) %>%
  summarise(
    avg_age = mean(age, na.rm = TRUE),
    total_people = n()
  )
```

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

```
# Rank Individuals by avg_drinks_perday within each age_group
ranked_data = people_data %>%
  group_by(age_group) %>%
  mutate(rank = dense_rank(desc(avg_drinks_perday))) %>%
  filter(rank <= 3)

ranked_data = ranked_data[order(ranked_data$rank), ]

# Display the top three individuals in each group
ranked_data
```

```
## # A tibble: 10 × 7
## # Groups:   age_group [3]
##      age sex    race     housing  avg_drinks_perday age_group  rank
##    <dbl> <chr>  <chr>    <chr>                <dbl> <fct>     <int>
##  1    67 Male   Black    Homeless               4.9 Senior        1
##  2    26 Female White    Rent                   4.8 Young         1
##  3    41 Female Black    Homeless               5   Adult         1
##  4    56 Female Hispanic Homeless               4.9 Senior        1
##  5    30 Male   Black    Rent                   4.7 Young         2
```

```
##  6    31 Male   White    Rent                  4.9 Adult         2
##  7    54 Female Black    Rent                  4.8 Senior        2
##  8    31 Male   Asian    Own                   4.7 Adult         3
##  9    55 Male   Black    Rent                  4.6 Senior        3
## 10    26 Male   Hispanic Own                   4.5 Young         3
```

Problem 4: Joining Data & Complex Summaries

```
## Perform an inner-join between PeopleData and TransactionsData using a
common column
#joined_data = people_data %>%
# inner_join(transaction_data, by = "customer_id")
# No common column???? If the data was correct, it would look like:

## Calculate total transaction amount for Heavy Drinkers
# total_transaction_amount = joined_data %>%
#  summarise(
#    total_amount = sum(case_when(
#      avg_drinks_perday > 3 ~ amount,
#      TRUE ~ 0
#    ), na.rm = TRUE)
#  )


# avg_transaction_by_race = joined_data %>%
#  filter(avg_drinks_perday > 3) %>%
#  group_by(race) %>%
#  summarise(avg_amount = mean(amount, na.rm = TRUE))
```

Problem 5: Pollution Characteristics

```
pollution_df = read.table("../Data/pollution1-1.txt", header = TRUE)

str(pollution_df)
```

```
## 'data.frame':    6940 obs. of  8 variables:
##  $ city      : chr  "chic" "chic" "chic" "chic" ...
##  $ tmpd      : num  31.5 33 33 29 32 40 34.5 29 26.5 32.5 ...
##  $ dptp      : num  31.5 29.9 27.4 28.6 28.9 ...
##  $ date      : chr  "1987-01-01" "1987-01-02" "1987-01-03" "1987-01-04"
...
##  $ pm25tmean2: num  NA NA NA NA NA NA NA NA NA NA ...
##  $ pm10tmean2: num  34 NA 34.2 47 NA ...
##  $ o3tmean2  : num  4.25 3.3 3.33 4.38 4.75 ...
##  $ no2tmean2 : num  20 23.2 23.8 30.4 30.3 ...
```

Problem 6: Pollution Data Selection & Cleaning

```
# Clean DataFrame
pollution_df$date = as.Date(pollution_df$date, format = "%Y-%m-%d")
pollution_df = pollution_df %>%
  mutate(across(where(is.numeric), ~ ifelse(is.na(.), mean(., na.rm = TRUE),
.)))
```

```r
# Use select() to keep every variable that ends with a two
pollution_df_four = pollution_df %>%
  select(ends_with("2"))

# Display the first few rows
head(pollution_df_four)
```

```
##   pm25tmean2 pm10tmean2 o3tmean2 no2tmean2
## 1   16.23096   34.00000 4.250000  19.98810
## 2   16.23096   33.89521 3.304348  23.19099
## 3   16.23096   34.16667 3.333333  23.81548
## 4   16.23096   47.00000 4.375000  30.43452
## 5   16.23096   33.89521 4.750000  30.33333
## 6   16.23096   48.00000 5.833333  25.77233
```

Problem 7: The Filter Function

```r
# Filter by a day, i.e the world championship day
champion_df = pollution_df %>%
  filter(date == "2005-10-23")

# Select Temp & Dew Point from date
temp_info = champion_df %>%
  select(tmpd, dptp)
temp_info
```

```
##   tmpd dptp
## 1   40 36.7
```

```r
# Filter PM2.5 levels greater than 30 and temperature that is greater than 80
filtered_pm_temp = pollution_df %>%
  filter(pm25tmean2 > 30, tmpd > 80)
filtered_pm_temp
```

```
##    city tmpd dptp       date pm25tmean2 pm10tmean2 o3tmean2 no2tmean2
## 1  chic   81 71.2 1998-08-23   39.60000       59.0 45.86364  14.32639
## 2  chic   81 70.4 1998-09-06   31.50000       50.5 50.66250  20.31250
## 3  chic   82 72.2 2001-07-20   32.30000       58.5 33.00380  33.67500
## 4  chic   84 72.9 2001-08-01   43.70000       81.5 45.17736  27.44239
## 5  chic   85 72.6 2001-08-08   38.83750       70.0 37.98047  27.62743
## 6  chic   84 72.6 2001-08-09   38.20000       66.0 36.73245  26.46742
## 7  chic   82 67.4 2002-06-20   33.00000       80.5 47.42673  30.76703
## 8  chic   82 63.5 2002-06-23   42.50000       65.0 54.88043  30.03913
## 9  chic   81 70.4 2002-07-08   33.10000       64.0 45.34969  27.67857
## 10 chic   82 66.2 2002-07-18   38.85000       72.5 44.98045  26.06905
## 11 chic   82 65.1 2003-06-25   33.90000       66.0 56.13666  22.94934
## 12 chic   84 68.4 2003-07-04   32.90000       47.5 45.66146  21.34375
## 13 chic   86 63.4 2005-06-24   31.85714       74.0 50.96649  23.75000
## 14 chic   82 64.6 2005-06-27   51.53750       79.0 55.23586  28.54937
## 15 chic   85 64.1 2005-06-28   31.20000       57.5 50.29144  26.55398
```

```
## 16 chic   84 67.0 2005-07-17   32.70000       42.5 44.64323  16.27083
## 17 chic   84 69.0 2005-08-03   37.90000       64.0 39.32111  23.61932
```

```
# Count the number of days that satisfy the condition
num_days = nrow(filtered_pm_temp)
num_days
```

```
## [1] 17
```

```
# See what months are in the filtered data
library(lubridate)
```

```
##
## Attaching package: 'lubridate'
```

```
## The following objects are masked from 'package:base':
##
##     date, intersect, setdiff, union
```

```
filtered_pm_temp %>%
  mutate(month = month(date)) %>%
  distinct(month)
```

```
##   month
## 1     8
## 2     9
## 3     7
## 4     6
```

Problem 8: Arrange Dates in Descending Order

```
# Arrange Dates in Descending Order
df_arranged = pollution_df %>%
  arrange(desc(date))

head(df_arranged)
```

```
##   city tmpd dptp       date pm25tmean2 pm10tmean2  o3tmean2 no2tmean2
## 1 chic   35 30.1 2005-12-31   15.00000       23.5  2.531250  13.25000
## 2 chic   36 31.0 2005-12-30   15.05714       19.2  3.034420  22.80556
## 3 chic   35 29.4 2005-12-29    7.45000       23.5  6.794837  19.97222
## 4 chic   37 34.5 2005-12-28   17.75000       27.5  3.260417  19.28563
## 5 chic   40 33.6 2005-12-27   23.56000       27.0  4.468750  23.50000
## 6 chic   35 29.6 2005-12-26    8.40000        8.5 14.041667  16.81944
```

Problem 9: Create a Year Variable and Display It

```
# Create year Variable
df_with_year = pollution_df %>%
  mutate(year = year(date))

head(df_with_year)
```

```
##   city tmpd   dptp       date pm25tmean2 pm10tmean2 o3tmean2 no2tmean2
year
## 1 chic 31.5 31.500 1987-01-01   16.23096   34.00000 4.250000  19.98810
1987
## 2 chic 33.0 29.875 1987-01-02   16.23096   33.89521 3.304348  23.19099
1987
## 3 chic 33.0 27.375 1987-01-03   16.23096   34.16667 3.333333  23.81548
1987
## 4 chic 29.0 28.625 1987-01-04   16.23096   47.00000 4.375000  30.43452
1987
## 5 chic 32.0 28.875 1987-01-05   16.23096   33.89521 4.750000  30.33333
1987
## 6 chic 40.0 35.125 1987-01-06   16.23096   48.00000 5.833333  25.77233
1987
```

Problem 10: Group Data By Year & Compute Median of O3 Levels

```
df_median_o3 = df_with_year %>%
  group_by(year) %>%
  summarise(median_o3 = median(o3tmean2, na.rm = TRUE))

head(df_median_o3)
```

```
## # A tibble: 6 × 2
##    year median_o3
##   <dbl>     <dbl>
## 1  1987      18.8
## 2  1988      20.4
## 3  1989      19.3
## 4  1990      19.0
## 5  1991      18.4
## 6  1992      15.2
```