

hw2

Sherine

2023-09-14

R Markdown

This is an R Markdown document. Markdown is a simple formatting syntax for authoring HTML, PDF, and MS Word documents. For more details on using R Markdown see <http://rmarkdown.rstudio.com>.

When you click the **Knit** button a document will be generated that includes both content as well as the output of any embedded R code chunks within the document. You can embed an R code chunk like this:

1. Using the flights dataset and functions from the dplyr library, create a tibble that summarizes the average gain for the flights of each airline that occurred between July and November (included). The tibble should have a summary column named average_gain and another column with the name of the airline. The rows of the tibble should be ordered by average_gain in decreasing order. Ignore any NAs in the computation of average_gain. Recall that “gain” is defined as arr_delay - dep_delay.

```
new1<-flights %>% filter(month >= 7 & month <= 11) %>%  
  mutate( gain = arr_delay - dep_delay) %>%  
  inner_join( airlines,by=c('carrier')) %>%  
  group_by(name) %>%  
  summarize(average_gain = mean(gain, na.rm = TRUE)) %>%  
  arrange(desc(average_gain))  
new1
```

```
## # A tibble: 16 x 2  
##   name                average_gain  
##   <chr>                <dbl>  
## 1 AirTran Airways Corporation    1.29  
## 2 Frontier Airlines Inc.        0.545  
## 3 Envoy Air                     0.0636  
## 4 Hawaiian Airlines Inc.       -0.910  
## 5 US Airways Inc.              -2.84  
## 6 SkyWest Airlines Inc.        -2.85  
## 7 Mesa Airlines Inc.           -3.02  
## 8 JetBlue Airways              -4.91  
## 9 ExpressJet Airlines Inc.     -5.79  
## 10 Delta Air Lines Inc.        -7.96  
## 11 Southwest Airlines Co.      -8.57  
## 12 American Airlines Inc.     -8.67  
## 13 United Air Lines Inc.       -9.62  
## 14 Virgin America              -9.88  
## 15 Endeavor Air Inc.          -10.3  
## 16 Alaska Airlines Inc.       -22.2
```

2. Using the diamonds dataset and functions from the dplyr library, find the minimum depth percentage and the maximum weight (carat) of diamonds that are priced between \$1,000 and \$2,500 (included), and that have cut quality that is good, very good, or premium.

```
diamonds1<-filter(diamonds,price>=1000 & price<=2500,cut %in% c("Good", "Very Good", "Premium"))

fin<-diamonds1%>%
  group_by(cut)%>%
  summarize( min_depth_percentage = min(depth), max_weight_carat = max(carat))
fin
```

```
## # A tibble: 3 x 3
##   cut      min_depth_percentage max_weight_carat
##   <ord>          <dbl>          <dbl>
## 1 Good              54.3              1.14
## 2 Very Good         56.9              1.06
## 3 Premium           58                1.2
```

3. Read the exams.csv dataset, then: – make it into a tidy tibble – use functions from the dplyr package on your tidy tibble to create a new tibble with columns student and pass where student is the student name and pass is “yes” if the average grade of the student is at least 60 and “no” otherwise – write this tibble to disk in CSV format. Hint: ? ifelse

```
exams<-read.csv("C:\\Users\\Sherine\\Downloads\\exams.csv")

filter_exams <- mutate(
  exams,
  grade = (exam1+exam2+exam3)/3
)

new_exam<-select(filter_exams,-exam1,-exam2,-exam3)
filter_exam <- mutate(
  new_exam,
  pass = ifelse(grade >= 60, "yes", "no")
)

filter_exam1<-select(filter_exam,-grade)
filter_exam1
```

```
##   student pass
## 1   jenna  yes
## 2    bob   no
## 3 robert  yes
```

```
file_path <- "C:\\Users\\Sherine\\Downloads\\sample_data.csv"

# Write the tibble to a CSV file
write_csv(filter_exam1, file_path)
```

4. Read the transactions.csv dataset from the lab of class 4 and make it tidy using the code from the lab. Using functions from the dplyr library, create a tibble with a single column containing the unique months in which either Jenna or John had a transaction (either sale or purchase). Hint: ?is.na and ? distinct

```

# Read the full dataset with appropriate types.
transactions <- read_csv(
  "C:\\Users\\Sherine\\Downloads\\transactions.csv",
  col_types = cols(
    person = col_character(),
    month = col_integer(),
    purchase = col_integer(),
    sale = col_integer()
  )
)
transactions_tidy <-
  transactions %>% gather(purchase,
                        sale,
                        key = transaction_type,
                        value = amount)
transactions_tidy

```

```

## # A tibble: 22 x 4
##   person month transaction_type amount
##   <chr>   <int> <chr>             <int>
## 1 john     1 purchase           1000
## 2 jenna     1 purchase              NA
## 3 frank     8 purchase             400
## 4 bill     4 purchase             150
## 5 laurel    6 purchase              NA
## 6 mary     6 purchase              NA
## 7 joe      8 purchase             200
## 8 john     3 purchase              NA
## 9 frank    12 purchase             100
## 10 mary    10 purchase              75
## # i 12 more rows

```

```

tidy_transactions<-transactions_tidy%>%
  select(month,person,amount)%>%
  filter(!is.na(person)) %>%
  mutate(person=tolower(person))%>%
  distinct(month,person,.keep_all=TRUE)

unique_months<-tidy_transactions %>%
  filter(person %in% c("jenna","john")) %>%
  distinct(month)

unique_months

```

```

## # A tibble: 2 x 1
##   month
##   <int>
## 1     1
## 2     3

```