

ProginR3

Sherine

2023-09-21

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:

Read the persons.csv, food.csv, drinks.csv, and dinners.csv datasets. Create a new version of the dinners tibble (name it dinners_explicit) that includes exactly the following columns:

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

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

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

Use the dinners_explicit tibble that you just created to now create a summary tibble for the 2 most popular drinks and a summary tibble for the 2 most popular foods. Hints: use the head or tail functions to select the top or bottom rows of a tibble; check out the drop_na function in the dplyr package to remove rows with missing values

```
dinners<-read.csv("C:\\Users\\Sherine\\Downloads\\dinners.csv")
dinners_explicit<-dinners %>%
  left_join( drinks,by=c('drink_id'='item_id')) %>%
  left_join(food,by=c('food_id'))%>%
  left_join(persons, by=c('person_id'='id'))%>%
  select(item_name,price.x ,name,price.y,first_name,last_name,age) %>%
  rename("drink" = "item_name",
         "drink_price" = "price.x",
         "food" = "name",
         "food_price" = "price.y")
dinners_explicit
```

##	drink	drink_price	food	food_price	first_name	last_name	age
## 1	<NA>	<NA>	pasta	\$8.50	Valter	Evangelista	34
## 2	<NA>	<NA>	ice cream	\$4.50	Polly	Verity	61
## 3	water	\$1.00	<NA>	<NA>	<NA>	<NA>	NA
## 4	beer	\$5.00	cake	\$4.50	Aysha	Freitas	55
## 5	<NA>	<NA>	pizza	\$12	Rayno	Van Kann	29
## 6	water	\$1.00	fish	\$15.00	Valter	Evangelista	34
## 7	<NA>	<NA>	pizza	\$12	Rayno	Van Kann	29

## 8	sparkling water	\$2.00	ice cream	\$4.50	Ksenya	Dunai	31
## 9	soda	\$2.50	pop corn	\$1.50	Polly	Verity	61
## 10	water	\$1.00	salad	\$5.00	Aysha	Freitas	55
## 11	wine	\$9.00	<NA>	<NA>	<NA>	<NA>	NA
## 12	soda	\$2.50	salad	\$5.00	<NA>	<NA>	NA
## 13	beer	\$5.00	cake	\$4.50	Aysha	Freitas	55
## 14	<NA>	<NA>	steak	\$12.00	<NA>	<NA>	NA
## 15	wine	\$9.00	<NA>	<NA>	Valter Evangelista		34
## 16	wine	\$9.00	pop corn	\$1.50	<NA>	<NA>	NA
## 17	<NA>	<NA>	fish	\$15.00	<NA>	<NA>	NA
## 18	water	\$1.00	fries	\$3.00	Polly	Verity	61
## 19	sparkling water	\$2.00	burger	\$5.00	Polly	Verity	61
## 20	soda	\$2.50	steak	\$12.00	Polly	Verity	61
## 21	<NA>	<NA>	fries	\$3.00	Valter Evangelista		34
## 22	<NA>	<NA>	<NA>	<NA>	<NA>	<NA>	NA
## 23	<NA>	<NA>	pasta	\$8.50	<NA>	<NA>	NA
## 24	soda	\$2.50	pasta	\$8.50	<NA>	<NA>	NA
## 25	sparkling water	\$2.00	salad	\$5.00	<NA>	<NA>	NA
## 26	sparkling water	\$2.00	pizza	\$12	Rayno	Van Kann	29
## 27	sparkling water	\$2.00	ice cream	\$4.50	<NA>	<NA>	NA
## 28	sparkling water	\$2.00	fish	\$15.00	<NA>	<NA>	NA
## 29	<NA>	<NA>	fries	\$3.00	Polly	Verity	61
## 30	beer	\$5.00	pasta	\$8.50	Valter Evangelista		34
## 31	water	\$1.00	burger	\$5.00	<NA>	<NA>	NA
## 32	<NA>	<NA>	pasta	\$8.50	Rayno	Van Kann	29
## 33	<NA>	<NA>	salad	\$5.00	Aysha	Freitas	55
## 34	<NA>	<NA>	fries	\$3.00	<NA>	<NA>	NA
## 35	soda	\$2.50	cake	\$4.50	<NA>	<NA>	NA
## 36	sparkling water	\$2.00	fish	\$15.00	<NA>	<NA>	NA
## 37	soda	\$2.50	<NA>	<NA>	<NA>	<NA>	NA
## 38	beer	\$5.00	pop corn	\$1.50	<NA>	<NA>	NA
## 39	wine	\$9.00	fish	\$15.00	<NA>	<NA>	NA
## 40	<NA>	<NA>	salad	\$5.00	<NA>	<NA>	NA
## 41	soda	\$2.50	<NA>	<NA>	<NA>	<NA>	NA
## 42	<NA>	<NA>	fries	\$3.00	<NA>	<NA>	NA
## 43	wine	\$9.00	pizza	\$12	Polly	Verity	61
## 44	soda	\$2.50	<NA>	<NA>	Aysha	Freitas	55
## 45	water	\$1.00	pizza	\$12	<NA>	<NA>	NA
## 46	water	\$1.00	fries	\$3.00	Polly	Verity	61
## 47	<NA>	<NA>	pop corn	\$1.50	<NA>	<NA>	NA
## 48	soda	\$2.50	pizza	\$12	Rayno	Van Kann	29
## 49	sparkling water	\$2.00	pizza	\$12	Aysha	Freitas	55
## 50	<NA>	<NA>	steak	\$12.00	Polly	Verity	61

```

top2_drinks<-dinners_explicit %>%
  group_by(drink) %>%
  drop_na(drink)%>%
  summarise(Count = n()) %>%
  arrange(desc(Count))%>%
  rename("n" = "Count")%>%
  head(2)

top2_foods<- dinners_explicit %>%
  group_by(food) %>%

```

```
drop_na(food) %>%
summarise(Count = n()) %>%
arrange(desc(Count))%>%
rename("n" = "Count")%>%
head(2)
```

```
top2_foods
```

```
## # A tibble: 2 x 2
##   food      n
##   <chr> <int>
## 1 pizza      7
## 2 fries      6
```

```
top2_drinks
```

```
## # A tibble: 2 x 2
##   drink      n
##   <chr>    <int>
## 1 soda      9
## 2 sparkling water  8
```

Use the summary tibbles for the top 2 drinks and top 2 foods together with the `dinners_explicit` tibble to create a tibble that corresponds to all the rows in the `dinners_explicit` tibble where the food is in the top 2 foods and the drink is not in the top 2 drinks. Only keep rows without missing values in your final table.

```
final<-dinners_explicit%>%
  filter(
    food %in% top2_foods$food,
    !(drink %in% top2_drinks$drink),
    !is.na(food) & !is.na(drink)
  )%>%
  drop_na()
final
```

```
##   drink drink_price food food_price first_name last_name age
## 1 water      $1.00 fries      $3.00      Polly    Verity  61
## 2 wine       $9.00 pizza      $12      Polly    Verity  61
## 3 water      $1.00 fries      $3.00      Polly    Verity  61
```

Read the `singers.csv` dataset 1 – Change the tibble so that all first and last names are lower cased. – Add a column named `full_name` obtained by concatenating the first and last name with a single space between them. – Use the `day`, `month`, `year` columns to create a column named `date_of_birth` that with the date of birth of each singer. The type of this column should be `Date` (i.e., the tibble should indicate the type of the column as when you print out the tibble). – Calculate their current age in years (if all of them were still alive today). Note that this doesn't have to be an integer value. – Filter the tibble so that it only contains the subset of singers whose last name begins with 2 consecutive consonants.

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

singers <- singers %>%
```

```

mutate(first_name = tolower(first_name),
       last_name = tolower(last_name))%>%
mutate(full_name = paste(first_name, last_name, sep = " ")) %>%
mutate(date_of_birth = make_date(year, month, day))%>%
mutate(age_today = (today() - date_of_birth) / dyears(1))%>%
filter(str_detect(last_name, "^[^aeiou]{2}"))

```

singers

```

##   first_name last_name day month year   full_name date_of_birth age_today
## 1      rod   stewart  10     1 1945   rod stewart   1945-01-10  78.69405
## 2      mark  knopfler  12     8 1949 mark knopfler   1949-08-12  74.10815

```