Recommender

Importing libraries

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.0 --
```

```
## v ggplot2 3.3.2     v purrr   0.3.4
## v tibble  3.0.4     v dplyr   1.0.2
## v tidyr   1.1.2     v stringr 1.4.0
## v readr   1.4.0     v forcats 0.5.0
```

```
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::between()   masks data.table::between()
## x dplyr::filter()    masks stats::filter()
## x dplyr::first()     masks data.table::first()
## x dplyr::lag()       masks stats::lag()
## x dplyr::last()      masks data.table::last()
## x purrr::transpose() masks data.table::transpose()
```

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

```
## The following objects are masked from 'package:data.table':
##
##     hour, isoweek, mday, minute, month, quarter, second, wday, week,
##     yday, year
```

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

Importing raw data file

use read excel function and remove white space

Inspect the data using skim() function

Data summary

Name	Piped data
Number of rows	541909
Number of columns	8
_____	
Column type frequency:	
character	4
numeric	3
POSIXct	1
_____	

Group variables	None

Variable type: character

skim_variable	n_missing	complete_rate	min	max	empty	n_unique	whitespace
InvoiceNo	0	1	6	7	0	25900	0
StockCode	0	1	1	12	0	4070	0
Description	1454	1	1	35	0	4211	0
Country	0	1	3	20	0	38	0

Variable type: numeric

skim_variable	n_missing	complete_rate	mean	sd	p0	p25	p50	p75	p100	hist
Quantity	0	1.00	9.55	218.08	-80995.00	1.00	3.00	10.00	80995	▁▇▁▁
UnitPrice	0	1.00	4.61	96.76	-11062.06	1.25	2.08	4.13	38970	▁▇▁▁
CustomerID	135080	0.75	15287.69	1713.60	12346.00	13953.00	15152.00	16791.00	18287	▇▇▇▇▇

Variable type: POSIXct

skim_variable	n_missing	complete_rate	min	max	median	n_unique
InvoiceDate	0	1	2010-12-01 08:26:00	2011-12-09 12:50:00	2011-07-19 17:17:00	23260

Cleaning the data ^.^

Using the filter function

```
##   .
## 1 1
## 2 2
## 3 4
```

Using grepl which generates TRUE/FALSE vector for each row by matching with a string, like Italy below

```
## .
##  FALSE   TRUE
## 541106    803
```

Using filter an grepl to remove rows that have a cancellation

Check how many rows have a cancellation using summarise() function

```
## # A tibble: 1 x 1
##   Total
##   <int>
## 1  9288
```

Now remove the rows that have a cancellation

Making a table and using group_by function

```
## `summarise()` ungrouping output (override with `.groups` argument)
```

```
## # A tibble: 2 x 2
##   price count
##   <chr> <int>
## 1 1         6
## 2 2         1
```

```
## `summarise()` ungrouping output (override with `.groups` argument)
```

```
## # A tibble: 5 x 2
##   animal count
##   <chr>  <int>
## 1 Ant        1
## 2 Bat        1
## 3 Cow        1
## 4 Genda      2
## 5 Pig        2
```

```
## `summarise()` regrouping output by 'animal' (override with `.groups` argument)
```

```
## # A tibble: 6 x 3
## # Groups:   animal [5]
##   animal price count
##   <chr>  <chr> <int>
## 1 Ant    1         1
## 2 Bat    1         1
## 3 Cow    1         1
## 4 Genda  1         2
## 5 Pig    1         1
## 6 Pig    2         1
```

```
## `summarise()` regrouping output by 'animal' (override with `.groups` argument)
```

```
## # A tibble: 6 x 3
## # Groups:   animal [5]
##   animal price count
##   <chr>  <chr> <int>
## 1 Genda  1         2
## 2 Ant    1         1
## 3 Bat    1         1
## 4 Cow    1         1
## 5 Pig    1         1
## 6 Pig    2         1
```

```
## `summarise()` regrouping output by 'animal' (override with `.groups` argument)
```

```
## # A tibble: 6 x 3
##   animal price count
##   <chr>  <chr> <int>
## 1 Genda  1         2
## 2 Ant    1         1
## 3 Bat    1         1
## 4 Cow    1         1
## 5 Pig    1         1
## 6 Pig    2         1
```