


Follow Up: %>% vs. |>

- Both %>% and |> indicate a "piping" operation for sending the output of one function into the input of another function
- %>% is the older version; |> is the newer version in all recent versions of R
- Older code (that I forgot to change) may still show %>% but you can replace those with |>
- Check Tools > Global Options... > Code > "Use native pipe operator" 
- In RStudio you can use CMD/Ctrl + Shift + M to make it appear

Data Transformation and Summary

Prof. Roger D. Peng

*Department of Statistics and Data Sciences
University of Texas at Austin*

Spring 2024

Tidy Data

- There is one observation per row
- Each column represents a variable or measure or characteristic
- Every row / column combination is a single value
- R object is usually a `data.frame` or `tibble`

Data Transformation Verbs

- **select()** / **slice()**: return a subset of the columns / rows of a data frame
- **filter()**: extract a subset of rows from a data frame based on logical conditions
- **arrange()**: reorder rows of a data frame
- **rename()**: rename variables in a data frame
- **mutate()**: add new variables/columns or transform existing variables
- **group_by()** / **summarize()**: generate summary statistics of different variables in the data frame, possibly within grouped strata

Example: Collaborative Filtering Algorithm

- A system for recommending products / items to a person based on their similarity to other people
- a.k.a. Recommender systems
- a.k.a. "People like you also bought...", "You might also like..."
- Based on purchasing patterns, find a person that is similar to you that bought something that you **haven't yet purchased**

Amazon Reviews Data

- A sample of reviews on many different categories of products sold on Amazon including star ratings and full-text reviews
- Each row is one product review (this is the **unit of observation**)
- Reviews span the time period 1995--2015
- Dataset provided by Amazon
- We will focus on the "Pet Products" category only
- Available on Kaggle (22Gb): <https://www.kaggle.com/datasets/cynthiarempel/amazon-us-customer-reviews-dataset>

Reading in the Data

Don't run this code without the data file!

```
amazon <- read_tsv("amazon_reviews_us_Pet_Products_v1_00.tsv.bz2",  
  col_types = "cccccccdddcccccD")
```

```
> amazon  
# A tibble: 2,632,139 × 15  
  market...1 custo...2 revie...3 produ...4 produ...5 produ...6 produ...7 star_...8 helpf...9 total...x vine  verif...x  
  <chr>      <chr>    <chr>    <chr>    <chr>    <chr>    <chr>      <dbl>    <dbl>    <dbl> <chr> <chr>  
1 US      287948... REAKC2... B00Q0K... 510387... (8-Pac... Pet Pr...      5      0      0 N    Y  
2 US      114889... R3NU70... B00MBW... 912374... Warren... Pet Pr...      2      0      1 N    Y  
3 US      432149... R14QJW... B0084O... 902215... Tyson'... Pet Pr...      5      0      0 N    Y  
4 US      128350... R2HB7A... B001GS... 568880... Soft S... Pet Pr...      5      0      0 N    Y  
5 US      263340... RGKMPD... B004AB... 692846... EliteF... Pet Pr...      5      0      0 N    Y  
6 US      222836... R1DJCV... B00AX0... 590674... Carlso... Pet Pr...      5      0      0 N    Y  
7 US      144698... R3V52E... B00DQF... 688538... Dog Se... Pet Pr...      3      0      0 N    Y  
8 US      508963... R3DK08... B00DIR... 742358... The Bi... Pet Pr...      2      0      0 N    Y  
9 US      184405... R764DB... B00JRC... 869798... Cat Be... Pet Pr...      5      1      1 N    N  
10 US     505023... RW1853... B000L3... 501118... PetSaf... Pet Pr...      5      0      0 N    Y  
# ... with 2,632,129 more rows, 3 more variables: review_headline <chr>, review_body <chr>,  
# review_date <date>, and abbreviated variable names 1marketplace, 2customer_id, 3review_id,  
# 4product_id, 5product_parent, 6product_title, 7product_category, 8star_rating,  
# 9helpful_votes, xtotal_votes, xverified_purchase  
# i Use `print(n = ...)` to see more rows, and `colnames()` to see all variable names
```

Reading in the Data

```
> glimpse(amazon)
Rows: 2,632,139
Columns: 15
$ marketplace      <chr> "US", "US", "US", "US", "US", "US", "US", "US", "US", "US", "US", "US..."
$ customer_id      <chr> "28794885", "11488901", "43214993", "12835065", "26334022", "22283621..."
$ review_id        <chr> "REAKC26P07MDN", "R3NU7OMZ4HQIEG", "R14QJW3XF8Q01P", "R2HB7AX0394ZGY"..."
$ product_id       <chr> "B00Q0K9604", "B00MBW509W", "B0084OHUIO", "B001GS71K2", "B004ABH1LG",..."
$ product_parent   <chr> "510387886", "912374672", "902215727", "568880110", "692846826", "590..."
$ product_title    <chr> "(8-Pack) EZwhelp Belly Band/Wrap", "Warren Eckstein's Hugs & Kisses ..."
$ product_category <chr> "Pet Products", "Pet Products", "Pet Products", "Pet Products", "Pet ..."
$ star_rating      <dbl> 5, 2, 5, 5, 5, 5, 3, 2, 5, 5, 2, 1, 5, 1, 5, 5, 2, 5, 5, 5, 5, 5, 5, ...
$ helpful_votes    <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 2, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, ...
$ total_votes      <dbl> 0, 1, 0, 0, 0, 0, 0, 0, 1, 0, 0, 2, 0, 0, 3, 0, 0, 1, 0, 0, 0, 0, 0, ...
$ vine            <chr> "N", "N", "N", "N", "N", "N", "N", "N", "N", "N", "N", "N", "N", "N", "N",..."
$ verified_purchase <chr> "Y", "Y", "Y", "Y", "Y", "Y", "Y", "Y", "Y", "N", "Y", "Y", "Y", "Y", "Y",..."
$ review_headline  <chr> "A great purchase for \"dribbly\" dogs", "My dogs love Hugs and Kisse..."
$ review_body      <chr> "Best belly bands on the market! These are a great deal for an 8 pac..."
$ review_date      <date> 2015-08-31, 2015-08-31, 2015-08-31, 2015-08-31, 2015-08-31, 2015-08-...
>
```


Subset the Dataset

Turn big data into small data as quickly as possible (Robert Gentleman)

```
> glimpse(amazon)
Rows: 2,632,139
Columns: 15
$ marketplace <chr> "US", "US", "US", "US", "US", "US", "US", "US", "US", "US", "US", "US"...
$ customer_id <chr> "28794885", "11488901", "43214993", "12835065", "26334022", "22283621"...
$ review_id <chr> "REAKC26P07MDN", "R3NU7OMZ4HQIEG", "R14QJW3XF8Q01P", "R2HB7AX0394ZGY"...
$ product_id <chr> "B00Q0K9604", "B00MBW509W", "B0084OHUIO", "B001GS71K2", "B004ABH1LG",...
$ product_parent <chr> "510387886", "912374672", "902215727", "568880110", "692846826", "590...
$ product_title <chr> "(8-Pack) EZwhelp Belly Band/Wrap", "Warren Eckstein's Hugs & Kisses ...
$ product_category <chr> "Pet Products", "Pet Products", "Pet Products", "Pet Products", "Pet ...
$ star_rating <dbl> 5, 2, 5, 5, 5, 5, 3, 2, 5, 5, 2, 1, 5, 1, 5, 5, 2, 5, 5, 5, 5, 5, 5, ...
$ helpful_votes <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 2, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, ...
$ total_votes <dbl> 0, 1, 0, 0, 0, 0, 0, 0, 1, 0, 0, 2, 0, 0, 3, 0, 0, 1, 0, 0, 0, 0, 0, ...
$ vine <chr> "N", "N", "N", "N", "N", "N", "N", "N", "N", "N", "N", "N", "N", "N", "N",...
$ verified_purchase <chr> "Y", "Y", "Y", "Y", "Y", "Y", "Y", "Y", "Y", "N", "Y", "Y", "Y", "Y", "Y",...
$ review_headline <chr> "A great purchase for \"dribbly\" dogs", "My dogs love Hugs and Kisse...
$ review_body <chr> "Best belly bands on the market! These are a great deal for an 8 pac...
$ review_date <date> 2015-08-31, 2015-08-31, 2015-08-31, 2015-08-31, 2015-08-31, 2015-08-...
>
```

Subset the Dataset

```
dat <- amazon |>
  select(customer_id,
         product_id:product_title,
         star_rating,
         starts_with("review")) |>
  rename(rating = star_rating)
```

New name

Old name

all columns between (and including)
'product_id' and 'product_title'

all columns that start with the word
"review"

```
> dat
# A tibble: 2,632,139 × 9
  customer_id product_id product_parent product_title rating review...1 review...2 review...3 review_d...4
  <chr>        <chr>        <chr>        <chr>        <dbl> <chr>      <chr>      <chr>      <date>
1 28794885    B00Q0K9604 510387886    (8-Pack) EZwh... 5 REAKC2... "A gre... Best b... 2015-08-31
2 11488901    B00MBW509W 912374672    Warren Eckste... 2 R3NU70... "My do... My dog... 2015-08-31
3 43214993    B00840HUIO 902215727    Tyson's True ... 5 R14QJW... "I hav... I have... 2015-08-31
4 12835065    B001GS71K2 568880110    Soft Side Pet... 5 R2HB7A... "it is... It is ... 2015-08-31
5 26334022    B004ABH1LG 692846826    EliteField 3-... 5 RGKMPD... "Dog c... Worked... 2015-08-31
6 22283621    B00AX0LFM4 590674141    Carlson 68-In... 5 R1DJCV... "Five ... I love... 2015-08-31
7 14469895    B00DQFZGZ0 688538603    Dog Seat Cove... 3 R3V52E... "Seat ... Didn't... 2015-08-31
8 50896354    B00DIRF9US 742358789    The Bird Catc... 2 R3DK08... "Great... I had ... 2015-08-31
9 18440567    B00JRCBFUG 869798483    Cat Bed - Pur... 5 R764DB... "My ca... The pa... 2015-08-31
10 50502362    B000L3XYZ4 501118658    PetSafe Drink... 5 RW1853... "Five ... My cat... 2015-08-31
# ... with 2,632,129 more rows, and abbreviated variable names 1review_id, 2review_headline,
# 3review_body, 4review_date
# i Use `print(n = ...)` to see more rows
>
```

Subset the Dataset

```
dat <- amazon |>
  select(customer_id,
         product_id:product_title,
         star_rating,
         starts_with("review")) |>
  rename(rating = star_rating)
```

New name

Old name

all columns between (and including)
'product_id' and 'product_title'

all columns that start with the word
"review"

```
> dat
# A tibble: 2,632,139 × 9
  customer_id product_id product_parent product_title rating review_id1 review_headline2 review_body3 review_date4
  <chr>        <chr>        <chr>        <chr>        <dbl> <chr>        <chr>        <chr>        <date>
1 28794885    B00Q0K9604 510387886    (8-Pack) EZwh... 5 REAKC2... "A gre... Best b... 2015-08-31
2 11488901    B00MBW509W 912374672    Warren Eckste... 2 R3NU70... "My do... My dog... 2015-08-31
3 43214993    B00840HUIO 902215727    Tyson's True ... 5 R14QJW... "I hav... I have... 2015-08-31
4 12835065    B001GS71K2 568880110    Soft Side Pet... 5 R2HB7A... "it is... It is ... 2015-08-31
5 26334022    B004ABH1LG 692846826    EliteField 3-... 5 RGKMPD... "Dog c... Worked... 2015-08-31
6 22283621    B00AX0LFM4 590674141    Carlson 68-In... 5 R1DJCV... "Five ... I love... 2015-08-31
7 14469895    B00DQFZGZ0 688538603    Dog Seat Cove... 3 R3V52E... "Seat ... Didn't... 2015-08-31
8 50896354    B00DIRF9US 742358789    The Bird Catc... 2 R3DK08... "Great... I had ... 2015-08-31
9 18440567    B00JRCBFUG 869798483    Cat Bed - Pur... 5 R764DB... "My ca... The pa... 2015-08-31
10 50502362    B000L3XYZ4 501118658    PetSafe Drink... 5 RW1853... "Five ... My cat... 2015-08-31
# ... with 2,632,129 more rows, and abbreviated variable names 1review_id, 2review_headline,
# 3review_body, 4review_date
# i Use `print(n = ...)` to see more rows
>
```

Subset the Dataset

Keep everything BUT these variables

```
dat <- dat |>
  select(-review_body, -review_id)
```

```
> dat |>
+   glimpse()
Rows: 2,632,139
Columns: 7
$ customer_id      <chr> "28794885", "11488901", "43214993", "12835065", "26334022", "22283621",...
$ product_id       <chr> "B00Q0K9604", "B00MBW509W", "B00840HUI0", "B001GS71K2", "B004ABH1LG", "...
$ product_parent   <chr> "510387886", "912374672", "902215727", "568880110", "692846826", "59067...
$ product_title    <chr> "(8-Pack) EZwhelp Belly Band/Wrap", "Warren Eckstein's Hugs & Kisses Vi...
$ rating           <dbl> 5, 2, 5, 5, 5, 5, 3, 2, 5, 5, 2, 1, 5, 1, 5, 5, 2, 5, 5, 5, 5, 5, 5, 5,...
$ review_headline  <chr> "A great purchase for \"dribbly\" dogs", "My dogs love Hugs and Kisses"...
$ review_date      <date> 2015-08-31, 2015-08-31, 2015-08-31, 2015-08-31, 2015-08-31, 2015-08-31...
>
```

Distribution of Ratings

```
dat |>  
  group_by(rating) |>  
  summarize(n = n())
```

Create a new (summarized) variable
named 'n' that represents the number of
rows in each group computed by the
function 'n()'

```
# A tibble: 6 × 2
```

	rating	n
	<dbl>	<int>
1	1	247833
2	2	150599
3	3	215685
4	4	379691
5	5	1638308
6	NA	23

Number of 1-star ratings in the dataset

group_by() / summarize

```
dat |>  
  group_by(rating) |>  
  summarize(n = n())
```

rating	product_id	customer_id
4	B00Q0K9604	28794885
4	B00Q0K9604	11488901
2	B00Q0K9604	43214993
2	B00Q0K9604	12835065
1	B00Q0K9604	26334022

group_by() / summarize

```
dat |>  
  group_by(rating) |>  
  summarize(n = n())
```

group_by()

rating	product_id	customer_id
4	B00Q0K9604	28794885
4	B00Q0K9604	11488901

rating	product_id	customer_id
2	B00Q0K9604	43214993
2	B00Q0K9604	12835065

rating	product_id	customer_id
1	B00Q0K9604	26334022

rating	product_id	customer_id
4	B00Q0K9604	28794885
4	B00Q0K9604	11488901
2	B00Q0K9604	43214993
2	B00Q0K9604	12835065
1	B00Q0K9604	26334022

group_by() / summarize

```
dat |>  
  group_by(rating) |>  
  summarize(n = n())
```

rating	product_id	customer_id
4	B00Q0K9604	28794885
4	B00Q0K9604	11488901
2	B00Q0K9604	43214993
2	B00Q0K9604	12835065
1	B00Q0K9604	26334022

rating	product_id	customer_id	n()
4	B00Q0K9604	28794885	2
4	B00Q0K9604	11488901	

rating	product_id	customer_id	n()
2	B00Q0K9604	43214993	2
2	B00Q0K9604	12835065	

rating	product_id	customer_id	n()
1	B00Q0K9604	26334022	1

group_by() / summarize

rating	product_id	customer_id
4	B00Q0K9604	28794885
4	B00Q0K9604	11488901



2

summarize()

rating	product_id	customer_id
2	B00Q0K9604	43214993
2	B00Q0K9604	12835065



2

rating	product_id	customer_id
1	B00Q0K9604	26334022



1

rating	n
4	2
2	2
1	1

Distribution of Ratings

```
dat |>  
  group_by(rating) |>  
  summarize(n = n())
```

Create a new (summarized) variable
named 'n' that represents the number of
rows in each group computed by the
function 'n()'

```
# A tibble: 6 × 2
```

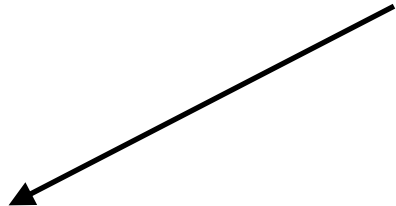
	rating	n
	<dbl>	<int>
1	1	247833
2	2	150599
3	3	215685
4	4	379691
5	5	1638308
6	NA	23

Number of 1-star ratings in the dataset

Distribution of Ratings

```
dat |>
  group_by(rating) |>
  summarize(n = n()) |>
  mutate(pct = 100 * n / sum(n))
```



Create a new variable in the summarized data frame that is the percentage of all ratings with that number of stars



```
# A tibble: 6 × 3
```

	rating <dbl>	n <int>	pct <dbl>
1	1	247833	9.42
2	2	150599	5.72
3	3	215685	8.19
4	4	379691	14.4
5	5	1638308	62.2
6	NA	23	0.000874

Distribution of Ratings



```
dat |>
  filter(!is.na(rating)) |>  Remove ratings that are NA
  group_by(rating) |>
  summarize(n = n()) |>
  mutate(pct = 100 * n / sum(n)) |>
  arrange(pct) 
```

Sort rows in ascending
order as determined by the
'pct' column

```
# A tibble: 5 × 3
```

	rating	n	pct
	<dbl>	<int>	<dbl>
1	2	150599	5.72
2	3	215685	8.19
3	1	247833	9.42
4	4	379691	14.4
5	5	1638308	62.2

Distribution of Ratings

```
dat |>
  filter(!is.na(rating)) |>  Remove ratings that are NA
  group_by(rating) |>
  summarize(n = n()) |>
  mutate(pct = 100 * n / sum(n)) |>
  arrange(desc(pct)) 
```

Sort rows in descending
order as determined by the
'pct' column

```
# A tibble: 5 × 3
```

	rating	n	pct
	<dbl>	<int>	<dbl>
1	5	1638308	62.2
2	4	379691	14.4
3	1	247833	9.42
4	3	215685	8.19
5	2	150599	5.72

Sample a Review

```
dat |>  
  sample_n(1) ← Sample one row at random
```

```
# A tibble: 1 × 7  
  customer_id product_id product_parent product_title rating review...1 review_d...2  
  <chr>      <chr>      <chr>      <chr>      <dbl> <chr>      <date>  
1 14431019    0763004227 46792424    Golden Retriever Metal Calenda... 2 Poor q... 2014-12-18  
# ... with abbreviated variable names 1review_headline, 2review_date
```

Collaborative Filtering

- Suggest products from "people like you"
- Search the entire dataset for other people who have reviewed that same product
- Subset to people who have given a similar (or the same) rating
- Find what other products those (other) people have reviewed

Sample a Review

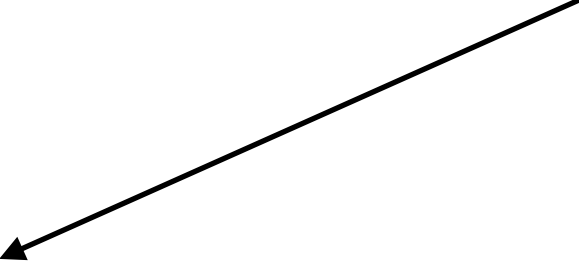
```
dat |>  
  sample_n(1)
```

```
# A tibble: 1 × 7  
  customer_id product_id product_parent product_title rating review_headline1 review_date2  
  <chr>      <chr>      <chr>      <chr>      <dbl> <chr>      <date>  
1 14431019 0763004227 46792424 Golden Retriever Metal Calenda... 2 Poor q... 2014-12-18  
# ... with abbreviated variable names 1review_headline, 2review_date
```


Filtering

**Filter to rows that just
review this product**

```
dat |>  
  filter(product_id == "0763004227")
```



```
# A tibble: 1 × 7  
  customer_id product_id product_parent product_title rating review...1 review_d...2  
  <chr>      <chr>      <chr>      <chr>      <dbl> <chr>      <date>  
1 14431019    0763004227 46792424    Golden Retriever Metal Calenda...      2 Poor q... 2014-12-18  
# ... with abbreviated variable names 1review_headline, 2review_date
```

No one else has purchased this product!

Problem

- Some products only have one review
- Some customers have only reviewed one product

Collaborative Filtering

- Search the entire dataset for other people who have reviewed that same product
 - **Requires products to have more than one review**
- Subset to people who have given a similar (or the same) rating
- Find what other products those (other) people have reviewed
 - **Requires customers to have reviewed more than one product**

Number of Reviews per Product

```
dat |>
  group_by(product_id) |>
  summarize(n = n())
```

← Count the number of reviews for each product

```
# A tibble: 238,936 × 2
```

	product_id	n
	<chr>	<int>
1	0310824230	2
2	039480001X	299
3	0615553605	7
4	0684836483	7
5	0761129804	2
6	0763004227	1
7	0764102885	16
8	0764118269	2
9	0764122940	4
10	0764132660	1

← One row for each unique product ID

How many products have more than 2 reviews?


```
# ... with 238,926 more rows
```

```
# i Use `print(n = ...)` to see more rows
```

Distribution of Number of Reviews per Product

```
dat |>  
  group_by(product_id) |>  
  summarize(n = n()) |>  
  select(n) |>  
  summary()
```

**Summarize the
distribution of the number
of reviews per product**



	n
Min.	: 1.00
1st Qu.:	1.00
Median :	2.00
Mean :	11.02
3rd Qu.:	5.00
Max.	:5252.00

**Half of all products have 2
or fewer reviews**



Identify Products That Have More Than 2 Reviews

Create a vector storing the product IDs with more than 2 reviews ('prodlist' is NOT a data frame)


```
prodlist <- dat |>
  group_by(product_id) |>
  summarize(n = n()) |>
  filter(n > 2) |>
  pull(product_id)
```

Extract just the 'product_id' variable from the data frame

```
> glimpse(prodlist)
chr [1:96458] "039480001X" "0615553605" "0684836483" "0764102885" "0764122940" "0793837871" ...
```

Number of unique product IDs that have more than 2 reviews

Collaborative Filtering

- Search the entire dataset for other people who have reviewed that same product
 - **Requires products to have more than one review** 
- Subset to people who have given a similar (or the same) rating
- Find what other products those (other) people have reviewed
 - **Requires customers to have reviewed more than one product**

Identify Users Who Have Reviewed More Than 1 Product

```
dat |>
  group_by(customer_id) |>
  summarize(n = n())
```

Count the number of
reviews that each
customer has done

```
# A tibble: 1,411,182 × 2
```

	customer_id	n
	<chr>	<int>
1	10000016	1
2	10000020	1
3	10000069	1
4	10000110	3
5	10000116	1
6	10000191	1
7	10000192	1
8	10000271	2
9	10000288	1
10	10000294	3

```
# ... with 1,411,172 more rows
```

```
# i Use `print(n = ...)` to see more rows
```


Identify Users Who Have Reviewed More Than 1 Product

```
dat |>  
  group_by(customer_id) |>  
  summarize(n = n()) |>  
  select(n) |>  
  summary()
```

Summarize the distribution of the
number of reviews per customer



	n
Min.	: 1.000
1st Qu.:	1.000
Median :	1.000
Mean :	1.865
3rd Qu.:	2.000
Max.	: 257.000

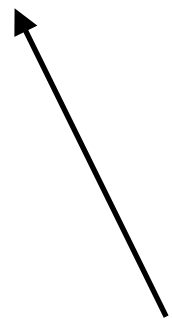
Half of all customers have
only reviewed 1 product



Identify Users Who Have Reviewed More Than 1 Product

```
userlist <- dat |>  
  group_by(customer_id) |>  
  summarize(n = n()) |>  
  filter(n > 1) |>  
  pull(customer_id)
```

```
> glimpse(userlist)  
chr [1:444301] "10000110" "10000271" "10000294" "10000392" "10000697" "10000810" "10000818" ...
```



Number of unique customer IDs that have reviewed more than 1 products

Subset the Data!

```
ratings <- dat |>
  filter(customer_id %in% userlist
         & product_id %in% prodlist)
```

**Keep rows where the value of
'customer_id' is contained in
the vector 'userlist'**

AND

**Keep rows where the value of
'product_id' is contained in
the vector 'prodlist'**

```
> ratings
# A tibble: 1,553,185 × 7
  customer_id product_id product_parent product_title rating review...1 review_d...2
  <chr>        <chr>        <chr>        <chr>        <dbl> <chr>      <date>
1 43214993    B00840HUIO 902215727    Tyson's True Chews Premium Je...     5 I have... 2015-08-31
2 50896354    B00DIRF9US 742358789    The Bird Catcher Pro Pets Can...     2 Great ... 2015-08-31
3 18440567    B00JRCBFUG 869798483    Cat Bed - Purrfect Thermal Ca...     5 My cat... 2015-08-31
4 50502362    B000L3XYZ4 501118658    PetSafe Drinkwell Platinum Pe...     5 Five S... 2015-08-31
5 33930128    B00BOEXWFG 454737777    Contech ZenDog Calming Compre...     2 Also h... 2015-08-31
6 43534290    B001HBBQKY 420905252    Wellness Crunchy Puppy Bites ...     1 DO NOT... 2015-08-31
7 45555864    B00701FHB0 302588963    Rx Vitamins Essentials 1 Piec...     5 Five S... 2015-08-31
8 11147406    B001P3NU30 525778264    Virbac C.E.T. Enzymatic Oral ...     1 Receiv... 2015-08-31
9 6495678     B00ZP6HS6S 414117299    Kitty Shack - 2 in 1 Tube Cat...     5 Five S... 2015-08-31
10 2019416     B00IP05CUA 833937853    Wellness Kittles Crunchy Natu...     5 kitty ... 2015-08-31
# ... with 1,553,175 more rows, and abbreviated variable names 1review_headline, 2review_date
# i Use `print(n = ...)` to see more rows
```

Find a Customer/Product

```
set.seed(2023)
ratings |>
  sample_n(1) |>
  select(customer_id, product_id,
         rating, product_title)
```

**Set random number generator
seed with an integer to make
sampling reproducible**



```
# A tibble: 1 × 4
  customer_id product_id rating product_title
  <chr>      <chr>    <dbl> <chr>
1 45464833 B001U4M9I6 2 Pet Parade's Dog Tweeter Training Aid, all breed, all size dogs
```

Find other people who have reviewed this product

Find Other Customers

```
ratings |>
  filter(product_id == "B001U4M9I6") |>
  select(customer_id, rating,
         review_headline, review_date)
```



```
# A tibble: 7 × 4
  customer_id rating review_headline review_date
  <chr>      <dbl> <chr>
1 41861197    1 This did not work at all for me & my ... 2014-07-17
2 12485248    1 Dog hated it 2014-02-20
3 45464833    2 Does not effect my dog one bit 2012-12-30
4 10508186    4 helpful to re-enforce training 2012-02-10
5 34029469    3 Good training aid 2011-02-06
6 11213388    3 Good 2011-01-28
7 50994974    1 Fell apart immediately 2010-06-20
```


Original reviewer

Which of the other people are most similar?

Characterize Other Customers

```
ratings |>
  filter(customer_id %in% c("45464833", "50994974",
                           "41861197", "12485248")) |>
  group_by(customer_id) |>
  summarize(mean_rating = mean(rating),
            num_rating = n())
```

**Compute the mean rating
and the number of ratings
by customer ID**



```
# A tibble: 4 × 3
```

	customer_id	mean_rating	num_rating
	<chr>	<dbl>	<int>
1	12485248	4.13	15
2	41861197	4	4
3	45464833	2.5	2
4	50994974	3.75	4

Original reviewer



group_by() / summarize

```
ratings |>
  filter(customer_id %in% c("45464833", "50994974",
                           "41861197", "12485248")) |>
  group_by(customer_id) |>
  summarize(mean_rating = mean(rating),
            num_rating = n())
```

rating	customer_id
4	28794885
3	28794885
1	43214993
5	43214993
1	26334022

group_by() / summarize

`group_by(customer_id)`

rating	customer_id
4	28794885
3	28794885

rating	customer_id
1	43214993
5	43214993

rating	customer_id
1	26334022

rating	customer_id
4	28794885
3	28794885
1	43214993
5	43214993
1	26334022

group_by() / summarize

rating	customer_id
4	28794885
3	28794885

mean()

→ 3.5

rating	customer_id
4	28794885
3	28794885
1	43214993
5	43214993
1	26334022

rating	customer_id
1	43214993
5	43214993

→ 3

rating	customer_id
1	26334022

→ 1

group_by() / summarize

rating	customer_id
4	28794885
3	28794885

`summarize(mean_rating = mean(rating))`

→ 3.5

rating	customer_id
1	43214993
5	43214993

→ 3

rating	customer_id
1	26334022


→ 1

mean_rating	customer_id
3.5	28794885
3	43214993
1	26334022

Characterize Other Customers

```
ratings |>
  filter(customer_id %in% c("45464833", "50994974",
                           "41861197", "12485248")) |>
  group_by(customer_id) |>
  summarize(mean_rating = mean(rating),
            num_rating = n())
```

**Compute the mean rating
and the number of ratings
by customer ID**



```
# A tibble: 4 × 3
```

	customer_id	mean_rating	num_rating
	<chr>	<dbl>	<int>
1	12485248	4.13	15
2	41861197	4	4
3	45464833	2.5	2
4	50994974	3.75	4

Original reviewer



Find Other Products

```
ratings |>
  filter(customer_id %in% c("50994974", "41861197", "12485248")) |>
  filter(rating >= 4) |>
  arrange(customer_id) |>
  select(customer_id, product_id,
         rating, product_title)
```

← Sort by customer_id in ascending order

A tibble: 17 × 4

Which product should be suggested?

	customer_id <chr>	product_id <chr>	rating <dbl>	product_title <chr>
1	12485248	B00IWLWT30	5	Nylabone Nutri Dent Complete Dog Treat Bones for Large Dogs up ...
2	12485248	B002QWLC98	5	GREENIES Weight Management Dental Dog Treats
3	12485248	B0035FZN50	5	Premier Busy Buddy Linkables Puzzle Dog Toys
4	12485248	B006WW85H0	5	Rachael Ray Nutrish Just 6 Dog Treats, 8-Ounce Pouch (Pack of ...
5	12485248	B00GSEVVA0	5	HappyDogz Pet Grooming Shedding Brush for Dog & Cat Hair, the D...
6	12485248	B006WW85HU	5	Rachael Ray Nutrish Healthy Weight Treats, Turkey & Rice Crunch...
7	12485248	B00D5AOC20	5	PetArmorPro Advanced Large (45-88.9 lbs)
8	12485248	B006WW84YY	5	Rachael Ray Nutrish Just 6 Dog Treats, 10-Ounce Pouch (Pack of ...
9	12485248	B0002AQQWY	5	Lambert Kay Linatone Shed Relief Skin/Coat Liquid Supplement fo...
10	12485248	B0002AQR8W	5	Lambert Kay Linatone Shed Relief Skin/Coat Liquid Supplement fo...
11	12485248	B000E804UA	4	Pet Parade Dog Repeller and Training Aid
12	41861197	B0030YIAW4	5	Walky Dog Plus Hands Free Dog Bicycle Exerciser Leash 2017 Newe...
13	41861197	B0002AS1B8	5	Bergan Wall Mounted Dispenser
14	41861197	B005ZKVIYC	5	FURminator FurVac Vacuum Accessory
15	50994974	B000084E6V	5	Nylabone Dental Dinosaur Chew
16	50994974	B00063MQLW	5	MidWest Life Stages Heavy-Duty Folding Metal Dog Crates; Single...
17	50994974	B001EQ4XIO	4	Nutri-Vet Hip & Joint Peanut Butter Wafer with Glucosamine for ...

Find Other Products

```
ratings |>
  filter(customer_id %in% c("50994974", "41861197", "12485248")) |>
  filter(rating >= 4) |>
  arrange(customer_id) |>
  select(customer_id, product_id,
         rating, product_title) |>
  slice(15:17)
```

**Just return rows 15--17
from the previous table**

```
# A tibble: 3 × 4
  customer_id product_id rating product_title
  <chr>      <chr>      <dbl> <chr>
1 50994974    B000084E6V      5 Nylabone Dental Dinosaur Chew
2 50994974    B00063MQLW      5 MidWest Life Stages Heavy-Duty Folding Metal Dog Crates; Single ...
3 50994974    B001EQ4XIO      4 Nutri-Vet Hip & Joint Peanut Butter Wafer with Glucosamine for L...
```

Collaborative Filtering Summary

- Subset the data to include
 - Products with multiple reviews
 - Customers with multiple reviews
- Identify a product that a customer has reviewed
- Identify other customers that reviewed that same product
- Find other products that those customers have reviewed well

Collaborative Filtering Summary

- The approach shown here was "model free"
- The data were repeatedly filtered / subsetted
- Only matched customers based on their ratings of the same product (two variables: 'product_id' and 'rating')
- In more complex situations it is easy to run out of data
- More sophisticated statistical models are needed then