Homework 1

Andey Nunes, MS
Jordan Hilton
Mengyu Li
Peter Boss

January 22, 2019

Document Setup

The first step for this week is to set up the R Markdown document options. Be sure that prior to executing code in this document that the following R packages are installed and updated in your R session:

- knitr
- readxl
- tidyverse

Tidyverse is an ecosystem of packages that work nicely together for data science tools. When the tidyverse package is installed, all the packages and their dependencies are automatically loaded into the R session. The packages included in the tidyverse package are listed here.

> broom, cli, crayon, dplyr, dbplyr, forcats, ggplot2, haven, hms, httr, jsonlite, lubridate, magrittr, modelr, purrr, readr, readxl (>=, reprex, rlang, rstudioapi, rvest, stringr, tibble, tidyr, xml2, tidyverse

Next step, load the data sets for the homework. Summaries are included in the appendix.

```
catalog <- read_excel("catalog.xls")
customers <- read_excel("customers.xls")
order_lines <- read_excel("order_lines.xls")#, sheet = "Sheet 1")
orders <- read_excel("orders.xls")
```

At first try, the `order_lines` data table did not load properly. We had to open the file in Excel to find that there are three sheets, two of which are pivot tables of the sheet containing all the data. These pivot tables are ahead of the actual data, so we manually reordered the sheets to put the data as the first sheet (labeled as *Sheet1* in .xls file). While in Excel, we also had to manually fix the column `customer_id` because the file name argument of the VLOOKUP command referenced a file path to the `orders` data file that was not accurate for our project folder. That formula was fixed, and the cell reference for that column updated. Then the file was resaved and used in our analysis.

Custom functions

This section is for building some custom functions that will come in handy later. In this section, we create a custom function called `countNA` to find the total missing values in a vector. We get the range of a numeric vector by taking the difference between the high and low values from the range output, and if the vector is not numeric, then provide NA. Next, we create `make_partBtable` which is a function that generates the generic structure for the tables in Part B. The variable_class use of *map_chr()* will throw an error on the data-time object because that class has multiple assignments. The `value_type` column is temporarily NA, because depending on the field, we will reassign one of: “question”, “answer”, “link”. We also add another important feature `count_unique` which provides information on the variation of entries in any field. The reason this is important is discussed further in section B.

```
countNA <- function(x) {sum(is.na(x)) } # count the number of missing data entries
get_range <- function(x) {ifelse(is.numeric(x), diff(range(x)), NA)}
```

```
make_partBtable <- function(x){
   df <- tibble(variable_name = names(x),
                variable_type = NA,
                variable_class = map_chr(x, class),
                count_missing = map_int(x, countNA),
                count_unique = map_dbl(x, ~length(unique(.x)) ),
                variable_range = map_dbl(x, get_range))
   return(df)

}
```

Homework Questions

Part A: General Questions

1. Key business questions

- What is the company's revenue?
- How many orders are there for each product?
- What is the total revenue for each product?
- Which products are not generating sales?
- How many active customers are there?
- Which market segment (international, domestic, or military) has the most sales growth over time?

2. How does each table relate to answering those questions?

- The catalog table lists each product along with information about that product (such as price, manufacturer, and name).
- The customers table lists each of the company's customers, along with information about that customer (such as location and name).
- The orders table has one record for every order a customer made, with the total cost of that order and information about the number of items in the order and its shipping weight.
- The order_lines table has one record for each different item that was purchased in a single order, along with links to the order.
- The `orders` data has an `order_date` and a `total_amount` for each unique `order_id`, which can be used to join the `order_lines` table to capture the `customer_id`. The `bt_state` field can be reclassified as one of three categories: "domestic" for US states, "international" indicated by the value `INTL`, and "military" indicated by the value "APO". This rebinned field can be used to classify the orders by market segment, using a table made from joining on the `customer_id` field. This final table can be summarized for total order amounts by month or quarter for each market segment then visualized on a timeline to spot trends in sales.

3. How do I have to link the tables in order to be able to answer those questions?

- What is the company's revenue? - simply sum total order amounts from the orders table
- How many orders are there for each product? - sum ordered units from the order_lines table, join with the catalog table for information about the product
- What is the total revenue for each product? - sum ordered dollar amounts from the order_lines table, join with the catalog table for information about the product

- Which products are not generating sales? - join the catalog table with the order_lines table, find records from the catalog table with zero or few ordered units
- How many active customers are there? - join the order table with customers table, find records from the customers table with a minimum threshold of orders
- Which market segment (international, domestic, or military) has the most sales growth over time? - join the order table with customers table, find records from the customers table with a minimum threshold of orders

In any case, the tables will be linked using common fields of unique identifiers such as `*_id` columns or new columns formed from prior aggregations. These common fields are keys and their operators are set functions such as union, intersection, and setdiff (or their `dplyr` equivalent joining functions).

Part B: Specific Questions

For each data set, we include a table that gives the field (variable) names, whether they are a *link*, *answer* or *question* field, the data class, how many missing observations, how many unique entries are in each column, and if the variable is numeric, a range is given.

The reason to include a column for unique entries is to identify two types of columns: unique identifiers, and fields that contain only one kind of entry. If the number of unique entries in a field is equal to the number of observations in the data table, then that variable can be considered a unique identifier and should not be considered to be a number for calculations nor a factor for grouping, rather it is a way to link unique rows between two separate data frames. A perfect exampe of this is the customer id field or the order id. Occasionally, date-time columns will yield this, but its also a good check for duplicate values in those types of columns. When a field contains only one unique entry (NA values are considered a type of entry) then it indicates a value that is descriptive of the entire table and is meaningless in differentiaing observations. It may not be a useless variable, because it could be indicative that our table is a filtered subset of a much larger table where that field had other values, but we would not know unless we knew how the table we are looking at was constructed. The large numbers of unique values also gives us a sense of the size of state space for that field and will indicate where descritizing actions may need to be focused.

Catalog

This data set has 761 observations on 7 variables with details as follows:

```
catalog_table <- make_partBtable(catalog)
catalog_table$variable_type <- c("link", "link", "answer", "question",
                                 "question", "question", "answer")
kable(catalog_table, caption = "Catalog Data Table Details")
```

Table 1: Catalog Data Table Details

variable_name	variable_type	variable_class	count_missing	count_unique	variable_range
id	link	numeric	0	761	818
product_code	link	character	1	761	NA
catalog_price	answer	numeric	0	134	654
category1	question	character	645	10	NA
manufact_id	question	numeric	0	5	8
vendor_id	question	numeric	0	5	8
name	answer	character	1	756	NA

Customers

Many of these fields are character string fields or identification fields. While the range values are given, they are not applicable to this data table.

This data set has 22070 observations on 10 variables with details as follows:

```
customers_table <- make_partBtable(customers)

customers_table$variable_type <- c("link", "link", rep("question", 6), "question or answer", "link")
# id variables and customer code are "links"
# names and bt_* are questions of who and where

kable(customers_table, caption = "Customers Data Table Details")
```

Table 2: Customers Data Table Details

variable_name	variable_type	variable_class	count_missing	count_unique	variable_range
cust_id	link	numeric	0	22070	22482
merchant_id	link	numeric	0	2	1
firstName	question	character	12070	502	NA
lastName	question	character	12070	1001	NA
bt_city	question	character	1	9032	NA
bt_state	question	character	137	67	NA
bt_country	question	character	0	79	NA
bt_zip	question	character	0	12434	NA
cc_type	question or answer	character	0	4	NA
custcode	link	character	0	22069	NA

Order_lines

This data set has 31233 observations on 21 variables with details as follows:

```
order_lines_table <- tibble(
  variable_name = names(order_lines),
  variable_type = c("link", "question", # which line in the order?
                    "link", "question", # what line status
                    "question & answer", # time intervals, when
                    rep("answer", 2), # how many
                    "question & answer", # time intervals, when
                    rep("unused", 3),# empty columns
                    "link", "question", # what is the list price
                    rep("unused", 2),# empty columns
                    "link", "questions", # which products
                    rep("question",2),
                    "link", "unused"), # last column is empty
                    # assign one of: "question", "answer", "link"
  variable_class = c(rep("numeric", 3), "character", "date-time",
                     "numeric", "numeric", "date-time",
                     rep("logical", 3), "numeric", "numeric",
                     "logical", "logical", "numeric", "character",
                     rep("numeric", 3), "logical"),
  count_missing = map_int(order_lines, countNA),
  count_unique = map_dbl(order_lines, ~length(unique(.x))),
```

```
  variable_range = map_dbl(order_lines, get_range))


kable(order_lines_table, caption = "Order_lines Data Table Details")
```

Table 3: Order_lines Data Table Details

variable_name	variable_type	variable_class	count_missing	count_unique	variable_range
order_id	link	numeric	2	23266	NA
order_line	question	numeric	2	22	NA
customer_id	link	numeric	14	22035	NA
line_status	question	character	2	5	NA
line_status_date	question & answer	date-time	2	1843	NA
order_qty	answer	numeric	1	43	NA
shipped_qty	answer	numeric	1	35	NA
bo_exp_date	question & answer	date-time	7055	186	NA
internal_note	unused	logical	31233	1	NA
spec_proc_note	unused	logical	31233	1	NA
spec_proc_id	unused	logical	31233	1	NA
order_line_id	link	numeric	2	31232	NA
list_price	question	numeric	2	272	NA
gift_note	unused	logical	31233	1	NA
distrib_id	unused	logical	31233	1	NA
product_id	link	numeric	2	678	NA
Product name	questions	character	1159	651	NA
Shipped Total	question	numeric	2	757	NA
Ordered Total	question	numeric	2	912	NA
format_id	link	numeric	2	7	NA
options	unused	logical	31233	1	NA

Orders

This data set has 23256 observations on 18 variables with details as follows:

```
orders_table <- tibble(
  variable_name = names(orders),
  variable_type = c(rep("link",2), "question", #when
                    rep("link",2),
                    rep("question", 2),# which
                    rep("answer", 7),# how much /total
                    rep("question",4)), # when
  # assign one of: "question", "answer", "link"
  variable_class = c("numeric", "numeric","date-time", "character",
                     "numeric", "character", "character",
                     "numeric", "character",rep("numeric", 5), "date-time",
                     "numeric", "logical", "logical"),
  count_missing = map_int(orders, countNA),
  count_unique = map_dbl(orders, ~length(unique(.x))),
  variable_range = map_dbl(orders, get_range))

kable(orders_table, caption = "Orders Data Table Details")
```

Table 4: Orders Data Table Details

variable_name	variable_type	variable_class	count_missing	count_unique	variable_range
order_id	link	numeric	0	23256	23575
merchant_id	link	numeric	0	2	1
order_date	question	date-time	0	2641	NA
po_number	link	character	22742	442	NA
cust_id	link	numeric	0	22034	32482
order_status	question	character	0	4	NA
ship_method	question	character	186	16	NA
items_amount	answer	numeric	0	2105	9590
amt_bracket	answer	character	0	4	NA
total_weight	answer	numeric	0	444	483
total_ship	answer	numeric	0	2298	631
total_hand	answer	numeric	0	1	0
total_tax	answer	numeric	0	1	0
total_amount	answer	numeric	0	10444	9584
order_status_date	question	date-time	0	1801	NA
send_inv_to_bill	question	numeric	0	2	1
coupon_code	question	logical	23256	1	NA
spec_instr	question	logical	23256	1	NA

Part C. Filter/Select Operations

For all these answers indicate clearly what fields you used, and why you chose those particular fields. If there were other fields you could have considered, indicate why you did not choose those.

4. Top 10 states for orders by dollar volume

We need the "state" field from the customers table, along with summed order totals from the order table, so we'll need to join those two tables and group by state.

```
# join customers and orders using cust_id link
# filter out the two non-state labels from bt_state
# pull out the two fields of interest and group the data by state
# summarize the observations to get a total by state and arrange in
# descending order, then rename the state column and keep only rows 1:10
orders_top_states <- customers %>%
  inner_join(orders, by="cust_id") %>%
  filter(bt_state != "APO",
         bt_state != "INTL") %>%
  select(bt_state, total_amount) %>%
  group_by(bt_state) %>%
  summarize(order_volume = sum(total_amount)) %>%
  arrange(desc(order_volume)) %>%
  rename(state = bt_state) %>%
  slice(1:10)

kable(orders_top_states, caption = "Top 10 states for orders by dollar volume")
```

Table 5: Top 10 states for orders by dollar volume

state	order_volume
CA	174920
TX	128754
FL	89137
NY	84202
VA	72133
NC	56886
WA	56838
OR	55147
IL	54843
PA	50150

5. Top 10 countries for orders by dollar volume

```
#head(orders)
#head(customers)

top10_Order_Dollar_byCountry <- inner_join(orders, customers, by = c('cust_id')) %>%
  group_by(bt_country) %>%
  summarise(totDollarVol =  sum(total_amount)) %>%
  arrange(desc(totDollarVol)) %>%
  top_n(10)
```

```
## Selecting by totDollarVol
```

```
kable(top10_Order_Dollar_byCountry, caption="Top 10 Countries by Dollar Volume")
```

Table 6: Top 10 Countries by Dollar Volume

bt_country	totDollarVol
United States	1695959
Canada	75096
Singapore	22706
Australia	17456
United Kingdom	14136
France	11361
Qatar	9081
Germany	9063
Spain	7924
Denmark	6439

6. Top 10 selling products by units; then by dollar volume

```
#head(order_lines)
Top10_SellProduct_ByUnit <- inner_join(order_lines, catalog, by = c('product_id'='id')) %>%
  group_by(name) %>%
  summarise(totUnit =  sum(shipped_qty)) %>%
  arrange(desc(totUnit)) %>%
  top_n(10)
```

```
## Selecting by totUnit
```

```
kable(Top10_SellProduct_ByUnit, caption="Top 10 Products by Unit Volume")
```

Table 7: Top 10 Products by Unit Volume

name	totUnit
Sheath: Large - Black	682
Infinity Ultra Task Lightâ„¢ - White L.E.D. - Black	475
MP800/Diesel & Gator Replacement Sheath	391
Multi-PlierÂ® 400 - Compact Sport- Needlenose	374
Ultralight L.S.T.Â® - Fine Edge	373
Guardian Back-UpÂ® - Double Fine Edge	357
L.S.T.Â® - Fine Edge	349
Sheath: Medium - Black	339
Multi-PlierÂ® 800 - Legend	322
Tool Kit for MP400, MP600, MP800	313

```
Top10_SellProduct_ByDollar <- inner_join(order_lines, catalog, by = c('product_id'='id')) %>%
  group_by(name) %>%
  summarise(totDollar =  sum(`Shipped Total`)) %>%
  arrange(desc(totDollar)) %>%
  top_n(10)
```

```
## Selecting by totDollar
```

```
kable(Top10_SellProduct_ByDollar, caption="Top 10 Products by Dollar Volume")
```

Table 8: Top 10 Products by Dollar Volume

name	totDollar
Multi-PlierÂ® 800 - Legend	27507
LMFâ„¢ II Infantry - Black	21592
Multi-PlierÂ® 600 Series - D.E.T.	20356
Guardian Back-UpÂ® - Double Fine Edge	19116
Applegate-Fairbairnâ„¢ Covert - Double Bevel - Black Oxide	18437
06 Automaticâ„¢ - Serrated Edge - Drop Point	17879
Multi-PlierÂ® 600 Series - Maintenance Kit	17739
06 Automaticâ„¢ - Serrated Edge - Tanto	17674
Hinderer Rescueâ„¢ - Serrated Edge	16511
Applegate-Fairbairnâ„¢ Combat Folder - Double Bevel - Sheath	16347

7. For each of the top two US states and each of the top two countries (excluding the US) in questions 1 and 2, what are the 5 top selling products by units? By dollar volume? (5%)

Our top two states are CA and TX. The top 5 products in CA by units are:

```
Top5CAbyUnits <- inner_join(order_lines, catalog, by = c('product_id'='id')) %>%
  inner_join(customers, by=c("customer_id"="cust_id")) %>%
  filter(bt_state == "CA") %>%
  group_by(name) %>%
  summarise(totUnit =  sum(shipped_qty)) %>%
  arrange(desc(totUnit)) %>%
  top_n(5)
```

```
## Selecting by totUnit
```

```
kable(Top5CAbyUnits, caption="Top 5 Products by Unit in CA")
```

Table 9: Top 5 Products by Unit in CA

name	totUnit
EZ Outâ„¢ Skeleton - Serrated Edge	70
Crucialâ„¢ - Black	66
Sheath: Large - Black	56
EZ Outâ„¢ Jr. - Serrated Edge	43
Guardian Back-UpÂ® - Double Fine Edge	35

And by dollar volume:

```
Top5CAbyDollar <- inner_join(order_lines, catalog, by = c('product_id'='id')) %>%
  inner_join(customers, by=c("customer_id"="cust_id")) %>%
  filter(bt_state == "CA") %>%
  group_by(name) %>%
  summarise(totDollar =  sum(`Shipped Total`)) %>%
  arrange(desc(totDollar)) %>%
  top_n(5)
```

```
## Selecting by totDollar
```

```
kable(Top5CAbyDollar, caption="Top 5 Products by Dollar in CA")
```

Table 10: Top 5 Products by Dollar in CA

name	totDollar
LMFâ„¢ II Infantry - Black	2909
06 Automaticâ„¢ - Serrated Edge - Tanto	2659
Multi-PlierÂ® 800 - Legend	2576
Crucialâ„¢ - Black	2373
EZ Outâ„¢ Skeleton - Serrated Edge	2344

The same two queries for Texas:

```
Top5TXbyUnits <- inner_join(order_lines, catalog, by = c('product_id'='id')) %>%
  inner_join(customers, by=c("customer_id"="cust_id")) %>%
  filter(bt_state == "TX") %>%
  group_by(name) %>%
  summarise(totUnit =  sum(shipped_qty)) %>%
  arrange(desc(totUnit)) %>%
  top_n(5)
```

```
## Selecting by totUnit
```

```
kable(Top5TXbyUnits, caption="Top 5 Products by Unit in TX")
```

Table 11: Top 5 Products by Unit in TX

name	totUnit
EZ Outâ„¢ Skeleton - Fine Edge	49
Microlight LST - Black	47
Sheath: Large - Black	43
Multi-PlierÂ® 800 - Legend	36
L.S.T.Â® - Fine Edge	32

```
Top5TXbyDollar <- inner_join(order_lines, catalog, by = c('product_id'='id')) %>%
  inner_join(customers, by=c("customer_id"="cust_id")) %>%
  filter(bt_state == "CA") %>%
  group_by(name) %>%
  summarise(totDollar =  sum(`Shipped Total`)) %>%
  arrange(desc(totDollar)) %>%
  top_n(5)
```

```
## Selecting by totDollar
```

```
kable(Top5TXbyDollar, caption="Top 5 Products by Dollar in TX")
```

Table 12: Top 5 Products by Dollar in TX

name	totDollar
LMFâ„¢ II Infantry - Black	2909
06 Automaticâ„¢ - Serrated Edge - Tanto	2659
Multi-PlierÂ® 800 - Legend	2576
Crucialâ„¢ - Black	2373
EZ Outâ„¢ Skeleton - Serrated Edge	2344

Now the same thing for Canada and Singapore:

```
Top5CADbyUnits <- inner_join(order_lines, catalog, by = c('product_id'='id')) %>%
  inner_join(customers, by=c("customer_id"="cust_id")) %>%
  filter(bt_country == "Canada") %>%
  group_by(name) %>%
  summarise(totUnit =  sum(shipped_qty)) %>%
  arrange(desc(totUnit)) %>%
  top_n(5)
```

```
## Selecting by totUnit
```

```
kable(Top5CADbyUnits, caption="Top 5 Products by Unit in Canada")
```

Table 13: Top 5 Products by Unit in Canada

name	totUnit
Remixâ„¢ - Serrated Edge	30
Hinderer Rescueâ„¢ - Serrated Edge	29
LMFâ„¢ II Infantry - Black	22
Pocket Sharpener	20
Multi-PlierÂ® 800 - Legend	19

```
Top5CADbyDollar <- inner_join(order_lines, catalog, by = c('product_id'='id')) %>%
  inner_join(customers, by=c("customer_id"="cust_id")) %>%
  filter(bt_country == "Canada") %>%
  group_by(name) %>%
  summarise(totDollar =  sum(`Shipped Total`)) %>%
  arrange(desc(totDollar)) %>%
  top_n(5)
```

```
## Selecting by totDollar
```

```
kable(Top5CADbyDollar, caption="Top 5 Products by Dollar in Canada")
```

Table 14: Top 5 Products by Dollar in Canada

name	totDollar
Hinderer Rescueâ„¢ - Serrated Edge	2519
LMFâ„¢ II Infantry - Black	1999
Multi-PlierÂ® 800 - Legend	1597
Multi-PlierÂ® 600 - Black - Bluntnose with Tungsten Carbide Inserts - Nylon Sheath	962
Dieselâ„¢ - Stainless Steel - Black	959

```
Top5SingaporebyUnits <- inner_join(order_lines, catalog, by = c('product_id'='id')) %>%
  inner_join(customers, by=c("customer_id"="cust_id")) %>%
  filter(bt_country == "Singapore") %>%
  group_by(name) %>%
  summarise(totUnit =  sum(shipped_qty)) %>%
  arrange(desc(totUnit)) %>%
  top_n(5)
```

```
## Selecting by totUnit
```

```
kable(Top5SingaporebyUnits, caption="Top 5 Products by Unit in Singapore")
```

Table 15: Top 5 Products by Unit in Singapore

name	totUnit
LMFâ„¢ II Survival - Coyote Brown	1
MP400 Series - Compact Sport w/ Corkscrew	1
MP400 Series - Fisherman	1
MP600 Series - Fisherman	1
Multi-PlierÂ® 600 Pro Scout - Needlenose	1
Multi-PlierÂ® 600 Series - D.E.T.	1
Tool Kit for MP400, MP600, MP800	1

```
Top5SingaporebyDollar <- inner_join(order_lines, catalog, by = c('product_id'='id')) %>%
  inner_join(customers, by=c("customer_id"="cust_id")) %>%
  filter(bt_country == "Singapore") %>%
  group_by(name) %>%
  summarise(totDollar =  sum(`Shipped Total`)) %>%
  arrange(desc(totDollar)) %>%
  top_n(5)
```

```
## Selecting by totDollar
```

```
kable(Top5SingaporebyDollar, caption="Top 5 Products by Dollar in Singapore")
```

Table 16: Top 5 Products by Dollar in Singapore

name	totDollar
Multi-PlierÂ® 600 Series - D.E.T.	118
LMFâ„¢ II Survival - Coyote Brown	98
MP600 Series - Fisherman	70
Multi-PlierÂ® 600 Pro Scout - Needlenose	65
MP400 Series - Fisherman	43

8. Provide the customer ID's, order dates, and order amounts for all customers who have ordered more than once. (5%)

```
# make a copy of orders that uses table() function to get counts of the customer IDs
rc <- as.data.frame(table(orders$cust_id))
# filter for the ones we want
rc <- rc[rc$"Freq" > 1,]
# copy it in a format that will behave well with the subset() in the next line
rc2 <- as.character(rc$Var1)
# pull a subset in our list
repeat_customers <- subset(orders, orders$cust_id %in% rc2)
kable(repeat_customers[c(1:10, 2210:2219),c(5,3,14)],
      caption = "Repeat customers, with order dates and dollar totals (first 10 lines and last 10 lines)
```

Table 17: Repeat customers, with order dates and dollar totals (first 10 lines and last 10 lines)

cust_id	order_date	total_amount
10034	2003-10-17	64
10002	2003-10-12	103
10002	2003-10-13	35
10004	2003-10-12	23
10049	2003-10-20	19
10034	2003-10-22	60
10066	2003-10-23	14
10078	2003-10-26	56
10088	2003-10-27	120
10109	2003-11-03	85
24282	2010-12-23	59
31868	2010-12-25	52
32326	2010-12-28	36
10649	2010-12-28	30
17898	2010-12-29	49
32007	2010-12-29	87
32326	2010-12-30	89
32007	2010-12-31	56
32446	2011-01-15	115
32446	2011-01-20	88

```
# not in the code: the number of lines in the repeated customers table matches the total of rc$Freq (22
```

Part D. Sales increasing strategies

A quick list of sales increasing strategies include;

- We know we have one time and repeat customers, but perhaps are there any other ways to segment customers and offer special promotions to see which customer segments respond to particular sales promotions.
- It appears that many of the top selling products are accessories instead of knives themselves; it might be good to expand the line of sheaths and multitools.
- Sales are heavily concentrated in the US but there's a long tail of international buyers; there may be a growth opportunity in marketing in Europe and SE Asia.

Appendix

Summary tables

```
# this whole code chunk can be updated to be "include = FALSE"
# the use of head() is redundant since glimpse() shows more of the same information
# but also tells you how many observations are in the data set
# and doesn't truncate the list of variables

#kable(summary(catalog), caption = "catalog summary table")
#head(catalog)
glimpse(catalog)
```

```
## Observations: 761
## Variables: 7
## $ id            <dbl> 446, 455, 445, 444, 443, 442, 438, 439, 440, 441...
## $ product_code  <chr> "G79761", "plastic", "G75329", "G75328", "G75231...
## $ catalog_price <dbl> 9.9, 0.0, 11.9, 10.9, 12.9, 11.9, 9.5, 6.0, 6.0,...
## $ category1     <chr> "accessories", NA, "fishing", "fillet", "fillet"...
## $ manufact_id   <dbl> 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, ...
## $ vendor_id     <dbl> 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, ...
## $ name          <chr> "Exchange-A-Blade Sheath for 7 inch saw", "Plast...
```

```
summary(catalog)
```

```
##        id       product_code       catalog_price  category1
##  Min.   : 307   Length:761         Min.   :  0   Length:761
##  1st Qu.: 525   Class :character   1st Qu.: 18   Class :character
##  Median : 728   Mode  :character   Median : 34   Mode  :character
##  Mean   : 725                      Mean   : 49
##  3rd Qu.: 930                      3rd Qu.: 57
##  Max.   :1125                      Max.   :654
##   manufact_id    vendor_id       name
##  Min.   :0.0   Min.   :0.0   Length:761
##  1st Qu.:1.0   1st Qu.:1.0   Class :character
##  Median :1.0   Median :1.0   Mode  :character
##  Mean   :1.2   Mean   :1.2
##  3rd Qu.:1.0   3rd Qu.:1.0
```

```
##  Max.   :8.0   Max.   :8.0
```

```
#kable(summary(customers), caption = "customers summary table")
#head(customers)
glimpse(customers)
```

```
## Observations: 22,070
## Variables: 10
## $ cust_id     <dbl> 20696, 15465, 19830, 25532, 16044, 32394, 29572, 3...
## $ merchant_id <dbl> 2, 1, 2, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,...
## $ firstName   <chr> "Kristina", "Paige", "Sherri", "Gretchen", "Karen"...
## $ lastName    <chr> "Chung", "Chen", "Melton", "Hill", "Puckett", "Son...
## $ bt_city     <chr> "Piedmont", "Cincinnati", "Shelbyville", "North ri...
## $ bt_state    <chr> "OK", "OH", "TN", "AZ", "ON", "OR", "GA", "VA", "K...
## $ bt_country  <chr> "United States", "United States", "United States",...
## $ bt_zip      <chr> "73078", "45227", "37160", "86052", "K8H 2X3", "97...
## $ cc_type     <chr> "Visa", "Visa", "Mastercard", "Visa", "Visa", "Mas...
## $ custcode    <chr> "P20696", "G15465", "P19830", "G25532", "G16044", ...
```

```
summary(customers)
```

```
##     cust_id       merchant_id    firstName           lastName
##  Min.   :10000   Min.   :1.00   Length:22070       Length:22070
##  1st Qu.:15930   1st Qu.:1.00   Class :character   Class :character
##  Median :21448   Median :1.00   Mode  :character   Mode  :character
##  Mean   :21408   Mean   :1.05
##  3rd Qu.:26965   3rd Qu.:1.00
##  Max.   :32482   Max.   :2.00
##    bt_city            bt_state          bt_country
##  Length:22070       Length:22070       Length:22070
##  Class :character   Class :character   Class :character
##  Mode  :character   Mode  :character   Mode  :character
##
##
##
##     bt_zip            cc_type            custcode
##  Length:22070       Length:22070       Length:22070
##  Class :character   Class :character   Class :character
##  Mode  :character   Mode  :character   Mode  :character
##
##
##
```

```
#kable(summary(order_lines), caption = "order_lines summary table")
#head(order_lines)
glimpse(order_lines)
```

```
## Observations: 31,233
## Variables: 21
## $ order_id         <dbl> 34462, 26061, 35964, 35217, 14053, 15586, 167...
## $ order_line       <dbl> 1, 1, 1, 1, 1, 1, 1, 1, 1, 2, 1, 1, 1, 1, 1, ...
## $ customer_id      <dbl> 29522, 21537, 30924, 30246, 10052, 11518, 127...
## $ line_status      <chr> "S", "S", "S", "S", "S", "S", "S", "S", "S", ...
## $ line_status_date <dttm> 2009-12-18, 2007-07-31, 2010-08-31, 2010-04-...
## $ order_qty        <dbl> 1, 6, 5, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, ...
## $ shipped_qty      <dbl> 11, 6, 5, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2,...
```

```
## $ bo_exp_date      <dttm> 1899-12-31, 1899-12-31, 1899-12-31, 1899-12-...
## $ internal_note    <lgl> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, N...
## $ spec_proc_note   <lgl> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, N...
## $ spec_proc_id     <lgl> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, N...
## $ order_line_id    <dbl> 27539, 16544, 29509, 28514, 163, 2217, 3732, ...
## $ list_price       <dbl> 18, 18, 16, 19, 18, 18, 18, 18, 18, 18, 18, 1...
## $ gift_note        <lgl> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, N...
## $ distrib_id       <lgl> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, N...
## $ product_id       <dbl> 307, 307, 307, 307, 307, 307, 307, 307, 307, ...
## $ `Product name`   <chr> "Carbide Cutter Insert Replacements", "Carbid...
## $ `Shipped Total`  <dbl> 197, 108, 80, 38, 36, 36, 36, 36, 36, 36, 36,...
## $ `Ordered Total`  <dbl> 18, 108, 80, 38, 36, 36, 36, 36, 36, 36, 36, ...
## $ format_id        <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, ...
## $ options          <lgl> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, N...
```

```
summary(order_lines)
```

```
##     order_id       order_line    customer_id    line_status
##  Min.   :    0   Min.   : 1.0   Min.   :    0   Length:31233
##  1st Qu.:19842   1st Qu.: 1.0   1st Qu.:15484   Class :character
##  Median :25622   Median : 1.0   Median :20974   Mode  :character
##  Mean   :25707   Mean   : 1.4   Mean   :21083
##  3rd Qu.:31514   3rd Qu.: 2.0   3rd Qu.:26584
##  Max.   :37575   Max.   :21.0   Max.   :32482
##  NA's   :2       NA's   :2      NA's   :14
##  line_status_date                order_qty      shipped_qty
##  Min.   :2003-10-10 00:00:00   Min.   :    0   Min.   :    0
##  1st Qu.:2006-05-01 00:00:00   1st Qu.:    1   1st Qu.:    0
##  Median :2007-06-05 00:00:00   Median :    1   Median :    1
##  Mean   :2007-08-02 15:07:40   Mean   :    3   Mean   :    2
##  3rd Qu.:2008-12-15 00:00:00   3rd Qu.:    1   3rd Qu.:    1
##  Max.   :2011-01-21 00:00:00   Max.   :41409   Max.   :28257
##  NA's   :2                     NA's   :1       NA's   :1
##   bo_exp_date                  internal_note  spec_proc_note
##  Min.   :1899-12-31 00:00:00   Mode:logical   Mode:logical
##  1st Qu.:1899-12-31 00:00:00   NA's:31233     NA's:31233
##  Median :1899-12-31 00:00:00
##  Mean   :1903-04-11 12:05:08
##  3rd Qu.:1899-12-31 00:00:00
##  Max.   :2008-02-15 00:00:00
##  NA's   :7055
##  spec_proc_id   order_line_id     list_price  gift_note
##  Mode:logical   Min.   :   90   Min.   :  0   Mode:logical
##  NA's:31233     1st Qu.: 8174   1st Qu.: 18   NA's:31233
##                 Median :15982   Median : 35
##                 Mean   :15956   Mean   : 43
##                 3rd Qu.:23790   3rd Qu.: 55
##                 Max.   :31597   Max.   :361
##                 NA's   :2       NA's   :2
##  distrib_id       product_id   Product name       Shipped Total
##  Mode:logical   Min.   : 307   Length:31233       Min.   :   0
##  NA's:31233     1st Qu.: 408   Class :character   1st Qu.:   0
##                 Median : 560   Mode  :character   Median :  22
##                 Mean   : 586                      Mean   :  36
##                 3rd Qu.: 744                      3rd Qu.:  46
```

```
##                 Max.   :1101                     Max.   :6982
##                 NA's   :2                        NA's   :2
##  Ordered Total    format_id  options
##  Min.   :   0   Min.   : 0   Mode:logical
##  1st Qu.:  20   1st Qu.: 0   NA's:31233
##  Median :  37   Median : 0
##  Mean   :  54   Mean   : 0
##  3rd Qu.:  59   3rd Qu.: 0
##  Max.   :9590   Max.   :11
##  NA's   :2      NA's   :2
```

```
#kable(summary(orders), caption = "orders summary table")
#head(orders)
glimpse(orders)
```

```
## Observations: 23,256
## Variables: 18
## $ order_id          <dbl> 14035, 14034, 14033, 14032, 14031, 14030, 14...
## $ merchant_id       <dbl> 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,...
## $ order_date        <dttm> 2003-10-17, 2003-10-16, 2003-10-16, 2003-10...
## $ po_number         <chr> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, ...
## $ cust_id           <dbl> 10034, 10033, 10032, 10031, 10030, 10029, 10...
## $ order_status      <chr> "S", "S", "S", "S", "S", "S", "S", "S", "S",...
## $ ship_method       <chr> "GND", "3DS", "GND", "GND", "3DS", "1DA", "G...
## $ items_amount      <dbl> 58.9, 8.9, 50.0, 11.9, 9.9, 109.9, 23.9, 40....
## $ amt_bracket       <chr> "C", "A", "B", "B", "A", "D", "B", "B", "A",...
## $ total_weight      <dbl> 2.3, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.2, 1.0,...
## $ total_ship        <dbl> 5.5, 9.0, 5.2, 5.4, 9.0, 27.3, 5.3, 6.1, 5.4...
## $ total_hand        <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,...
## $ total_tax         <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,...
## $ total_amount      <dbl> 64, 18, 55, 17, 19, 137, 29, 46, 15, 23, 29,...
## $ order_status_date <dttm> 2003-10-17, 2003-10-17, 2003-10-17, 2003-10...
## $ send_inv_to_bill  <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,...
## $ coupon_code       <lgl> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, ...
## $ spec_instr        <lgl> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, ...
```

```
summary(orders)
```

```
##     order_id      merchant_id     order_date
##  Min.   :14000   Min.   :1.00   Min.   :2003-10-10 00:00:00
##  1st Qu.:20134   1st Qu.:1.00   1st Qu.:2006-04-28 00:00:00
##  Median :25948   Median :1.00   Median :2007-07-02 00:00:00
##  Mean   :25918   Mean   :1.05   Mean   :2007-08-11 16:51:42
##  3rd Qu.:31761   3rd Qu.:1.00   3rd Qu.:2008-12-19 00:00:00
##  Max.   :37575   Max.   :2.00   Max.   :2011-01-21 00:00:00
##   po_number           cust_id      order_status       ship_method
##  Length:23256       Min.   :    0   Length:23256       Length:23256
##  Class :character   1st Qu.:15778   Class :character   Class :character
##  Mode  :character   Median :21302   Mode  :character   Mode  :character
##                     Mean   :21295
##                     3rd Qu.:26849
##                     Max.   :32482
##   items_amount  amt_bracket         total_weight   total_ship
##  Min.   :   0   Length:23256       Min.   :  0   Min.   :  0
##  1st Qu.:  28   Class :character   1st Qu.:  1   1st Qu.:  7
```

```
##  Median :  48   Mode  :character   Median :  2   Median :  8
##  Mean   :  73                      Mean   :  3   Mean   : 11
##  3rd Qu.:  80                      3rd Qu.:  3   3rd Qu.: 10
##  Max.   :9590                      Max.   :483   Max.   :631
##    total_hand   total_tax  total_amount  order_status_date
##  Min.   :0    Min.   :0   Min.   :   6   Min.   :2003-10-10 00:00:00
##  1st Qu.:0    1st Qu.:0   1st Qu.:  36   1st Qu.:2006-05-30 18:00:00
##  Median :0    Median :0   Median :  57   Median :2007-07-12 00:00:00
##  Mean   :0    Mean   :0   Mean   :  84   Mean   :2007-08-21 21:51:27
##  3rd Qu.:0    3rd Qu.:0   3rd Qu.:  94   3rd Qu.:2008-12-26 00:00:00
##  Max.   :0    Max.   :0   Max.   :9590   Max.   :2011-01-21 00:00:00
##  send_inv_to_bill coupon_code    spec_instr
##  Min.   :0.00     Mode:logical   Mode:logical
##  1st Qu.:0.00     NA's:23256     NA's:23256
##  Median :0.00
##  Mean   :0.05
##  3rd Qu.:0.00
##  Max.   :1.00
```

```
unique_cat <- map_dbl(catalog, ~length(unique(.x)))
kable(unique_cat, caption = "Catalog Data: unique entry counts by data field")
```

Table 18: Catalog Data: unique entry counts by data field

	x
id	761
product_code	761
catalog_price	134
category1	10
manufact_id	5
vendor_id	5
name	756

```
unique_cust <- map_dbl(customers, ~length(unique(.x)))
kable(unique_cust, caption = "Customers Data: unique entry counts by data field")
```

Table 19: Customers Data: unique entry counts by data field

	x
cust_id	22070
merchant_id	2
firstName	502
lastName	1001
bt_city	9032
bt_state	67
bt_country	79
bt_zip	12434
cc_type	4
custcode	22069

```
unique_OL <- map_dbl(order_lines, ~length(unique(.x)))
kable(unique_OL, caption = "Order Lines Data: unique entry counts by data field")
```

Table 20: Order Lines Data: unique entry counts by data field

	x
order_id	23266
order_line	22
customer_id	22035
line_status	5
line_status_date	1843
order_qty	43
shipped_qty	35
bo_exp_date	186
internal_note	1
spec_proc_note	1
spec_proc_id	1
order_line_id	31232
list_price	272
gift_note	1
distrib_id	1
product_id	678
Product name	651
Shipped Total	757
Ordered Total	912
format_id	7
options	1

```
unique_orders <- map_dbl(orders, ~length(unique(.x)))
kable(unique_orders, caption = "Orders Data Table: unique entry counts by data field")
```

Table 21: Orders Data Table: unique entry counts by data field

	x
order_id	23256
merchant_id	2
order_date	2641
po_number	442
cust_id	22034
order_status	4
ship_method	16
items_amount	2105
amt_bracket	4
total_weight	444
total_ship	2298
total_hand	1
total_tax	1
total_amount	10444
order_status_date	1801
send_inv_to_bill	2
coupon_code	1

	x
spec_instr	1