DATA 607: Week 7

Cindy Lin

2025-03-11

INTRODUCTION

Week 7 goal is to work with different data format: HTML, JSON, XML, and parquet. The first step is to convert the received data to each format and then perform analysis.

Loading library

```
options(repos = c(CRAN = "https://cloud.r-project.org/"))
install.packages("arrow")
```

```
##
## The downloaded binary packages are in
##  /var/folders/y0/6tdnwf3d5yz6sbyqmnrdzvj40000gn/T//RtmpGiyxjN/downloaded_packages
```

```
library (tidyverse)
```

```
## -- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 --
## v dplyr     1.1.4     v readr     2.1.5
## v forcats   1.0.0     v stringr   1.5.1
## v ggplot2   3.5.1     v tibble    3.2.1
## v lubridate 1.9.3     v tidyr     1.3.1
## v purrr     1.0.2
## -- Conflicts ---------------------------------------- tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
## i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become errors
```

```
library(knitr) #HTML
library(jsonlite) #JSON
```

```
##
## Attaching package: 'jsonlite'
##
## The following object is masked from 'package:purrr':
##
##     flatten
```

```
library(xml2) #XML
library(arrow) #Parquet
```

```
##
## Attaching package: 'arrow'
##
## The following object is masked from 'package:lubridate':
##
##     duration
##
## The following object is masked from 'package:utils':
##
##     timestamp
```

```
library(rvest) #HTML
```

```
##
## Attaching package: 'rvest'
##
## The following object is masked from 'package:readr':
##
##     guess_encoding
```

I am loading the tidyverse library because there are functions that can help tidy the loaded data.

Loading the data

```
get_data <- read.csv("CUNYMart.csv", header = TRUE, sep = ",")
```

Loading the data

HTML - PRO and CON

```
html <-kable(get_data, format = "html")

writeLines(html, "get_data.html")
#save html file to directory

print(html)
```

```
## <table>
##  <thead>
##   <tr>
##    <th style="text-align:left;"> Category </th>
##    <th style="text-align:left;"> Item.Name </th>
##    <th style="text-align:right;"> Item.ID </th>
##    <th style="text-align:left;"> Brand </th>
```

```
##    <th style="text-align:right;"> Price </th>
##    <th style="text-align:left;"> Variation.ID </th>
##    <th style="text-align:left;"> Variation.Details </th>
##   </tr>
##  </thead>
## <tbody>
##   <tr>
##    <td style="text-align:left;"> Electronics </td>
##    <td style="text-align:left;"> Smartphone </td>
##    <td style="text-align:right;"> 101 </td>
##    <td style="text-align:left;"> TechBrand </td>
##    <td style="text-align:right;"> 699.99 </td>
##    <td style="text-align:left;"> 101-A </td>
##    <td style="text-align:left;"> Color: Black, Storage: 64GB </td>
##   </tr>
##   <tr>
##    <td style="text-align:left;"> Electronics </td>
##    <td style="text-align:left;"> Smartphone </td>
##    <td style="text-align:right;"> 101 </td>
##    <td style="text-align:left;"> TechBrand </td>
##    <td style="text-align:right;"> 699.99 </td>
##    <td style="text-align:left;"> 101-B </td>
##    <td style="text-align:left;"> Color: White, Storage: 128GB </td>
##   </tr>
##   <tr>
##    <td style="text-align:left;"> Electronics </td>
##    <td style="text-align:left;"> Laptop </td>
##    <td style="text-align:right;"> 102 </td>
##    <td style="text-align:left;"> CompuBrand </td>
##    <td style="text-align:right;"> 1099.99 </td>
##    <td style="text-align:left;"> 102-A </td>
##    <td style="text-align:left;"> Color: Silver, Storage: 256GB </td>
##   </tr>
##   <tr>
##    <td style="text-align:left;"> Electronics </td>
##    <td style="text-align:left;"> Laptop </td>
##    <td style="text-align:right;"> 102 </td>
##    <td style="text-align:left;"> CompuBrand </td>
##    <td style="text-align:right;"> 1099.99 </td>
##    <td style="text-align:left;"> 102-B </td>
##    <td style="text-align:left;"> Color: Space Gray, Storage: 512GB </td>
##   </tr>
##   <tr>
##    <td style="text-align:left;"> Home Appliances </td>
##    <td style="text-align:left;"> Refrigerator </td>
##    <td style="text-align:right;"> 201 </td>
##    <td style="text-align:left;"> HomeCool </td>
##    <td style="text-align:right;"> 899.99 </td>
##    <td style="text-align:left;"> 201-A </td>
##    <td style="text-align:left;"> Color: Stainless Steel, Capacity: 20 cu ft </td>
##   </tr>
##   <tr>
##    <td style="text-align:left;"> Home Appliances </td>
##    <td style="text-align:left;"> Refrigerator </td>
```

```
##    <td style="text-align:right;"> 201 </td>
##    <td style="text-align:left;"> HomeCool </td>
##    <td style="text-align:right;"> 899.99 </td>
##    <td style="text-align:left;"> 201-B </td>
##    <td style="text-align:left;"> Color: White, Capacity: 18 cu ft </td>
##   </tr>
##   <tr>
##    <td style="text-align:left;"> Home Appliances </td>
##    <td style="text-align:left;"> Washing Machine </td>
##    <td style="text-align:right;"> 202 </td>
##    <td style="text-align:left;"> CleanTech </td>
##    <td style="text-align:right;"> 499.99 </td>
##    <td style="text-align:left;"> 202-A </td>
##    <td style="text-align:left;"> Type: Front Load, Capacity: 4.5 cu ft </td>
##   </tr>
##   <tr>
##    <td style="text-align:left;"> Home Appliances </td>
##    <td style="text-align:left;"> Washing Machine </td>
##    <td style="text-align:right;"> 202 </td>
##    <td style="text-align:left;"> CleanTech </td>
##    <td style="text-align:right;"> 499.99 </td>
##    <td style="text-align:left;"> 202-B </td>
##    <td style="text-align:left;"> Type: Top Load, Capacity: 5.0 cu ft </td>
##   </tr>
##   <tr>
##    <td style="text-align:left;"> Clothing </td>
##    <td style="text-align:left;"> T-Shirt </td>
##    <td style="text-align:right;"> 301 </td>
##    <td style="text-align:left;"> FashionCo </td>
##    <td style="text-align:right;"> 19.99 </td>
##    <td style="text-align:left;"> 301-A </td>
##    <td style="text-align:left;"> Color: Blue, Size: S </td>
##   </tr>
##   <tr>
##    <td style="text-align:left;"> Clothing </td>
##    <td style="text-align:left;"> T-Shirt </td>
##    <td style="text-align:right;"> 301 </td>
##    <td style="text-align:left;"> FashionCo </td>
##    <td style="text-align:right;"> 19.99 </td>
##    <td style="text-align:left;"> 301-B </td>
##    <td style="text-align:left;"> Color: Red, Size: M </td>
##   </tr>
##   <tr>
##    <td style="text-align:left;"> Clothing </td>
##    <td style="text-align:left;"> T-Shirt </td>
##    <td style="text-align:right;"> 301 </td>
##    <td style="text-align:left;"> FashionCo </td>
##    <td style="text-align:right;"> 19.99 </td>
##    <td style="text-align:left;"> 301-C </td>
##    <td style="text-align:left;"> Color: Green, Size: L </td>
##   </tr>
##   <tr>
##    <td style="text-align:left;"> Clothing </td>
##    <td style="text-align:left;"> Jeans </td>
```

```
##    <td style="text-align:right;"> 302 </td>
##    <td style="text-align:left;"> DenimWorks </td>
##    <td style="text-align:right;"> 49.99 </td>
##    <td style="text-align:left;"> 302-A </td>
##    <td style="text-align:left;"> Color: Dark Blue, Size: 32 </td>
##   </tr>
##   <tr>
##    <td style="text-align:left;"> Clothing </td>
##    <td style="text-align:left;"> Jeans </td>
##    <td style="text-align:right;"> 302 </td>
##    <td style="text-align:left;"> DenimWorks </td>
##    <td style="text-align:right;"> 49.99 </td>
##    <td style="text-align:left;"> 302-B </td>
##    <td style="text-align:left;"> Color: Light Blue, Size: 34 </td>
##   </tr>
##   <tr>
##    <td style="text-align:left;"> Books </td>
##    <td style="text-align:left;"> Fiction Novel </td>
##    <td style="text-align:right;"> 401 </td>
##    <td style="text-align:left;"> - </td>
##    <td style="text-align:right;"> 14.99 </td>
##    <td style="text-align:left;"> 401-A </td>
##    <td style="text-align:left;"> Format: Hardcover, Language: English </td>
##   </tr>
##   <tr>
##    <td style="text-align:left;"> Books </td>
##    <td style="text-align:left;"> Fiction Novel </td>
##    <td style="text-align:right;"> 401 </td>
##    <td style="text-align:left;"> - </td>
##    <td style="text-align:right;"> 14.99 </td>
##    <td style="text-align:left;"> 401-B </td>
##    <td style="text-align:left;"> Format: Paperback, Language: Spanish </td>
##   </tr>
##   <tr>
##    <td style="text-align:left;"> Books </td>
##    <td style="text-align:left;"> Non-Fiction Guide </td>
##    <td style="text-align:right;"> 402 </td>
##    <td style="text-align:left;"> - </td>
##    <td style="text-align:right;"> 24.99 </td>
##    <td style="text-align:left;"> 402-A </td>
##    <td style="text-align:left;"> Format: eBook, Language: English </td>
##   </tr>
##   <tr>
##    <td style="text-align:left;"> Books </td>
##    <td style="text-align:left;"> Non-Fiction Guide </td>
##    <td style="text-align:right;"> 402 </td>
##    <td style="text-align:left;"> - </td>
##    <td style="text-align:right;"> 24.99 </td>
##    <td style="text-align:left;"> 402-B </td>
##    <td style="text-align:left;"> Format: Paperback, Language: French </td>
##   </tr>
##   <tr>
##    <td style="text-align:left;"> Sports Equipment </td>
##    <td style="text-align:left;"> Basketball </td>
```

```
##    <td style="text-align:right;"> 501 </td>
##    <td style="text-align:left;"> SportsGear </td>
##    <td style="text-align:right;"> 29.99 </td>
##    <td style="text-align:left;"> 501-A </td>
##    <td style="text-align:left;"> Size: Size 7, Color: Orange </td>
##   </tr>
##   <tr>
##    <td style="text-align:left;"> Sports Equipment </td>
##    <td style="text-align:left;"> Tennis Racket </td>
##    <td style="text-align:right;"> 502 </td>
##    <td style="text-align:left;"> RacketPro </td>
##    <td style="text-align:right;"> 89.99 </td>
##    <td style="text-align:left;"> 502-A </td>
##    <td style="text-align:left;"> Material: Graphite, Color: Black </td>
##   </tr>
##   <tr>
##    <td style="text-align:left;"> Sports Equipment </td>
##    <td style="text-align:left;"> Tennis Racket </td>
##    <td style="text-align:right;"> 502 </td>
##    <td style="text-align:left;"> RacketPro </td>
##    <td style="text-align:right;"> 89.99 </td>
##    <td style="text-align:left;"> 502-B </td>
##    <td style="text-align:left;"> Material: Aluminum, Color: Silver </td>
##   </tr>
## </tbody>
## </table>
```

PRO - HyperText Markup Language is useful for webpages and styling the content and formatting for display so from the visualization aspect of analysis, it is useful as a presentation tool. Since every browser supports it, it is "inexpensive" in the regards that you do not need extra software to support it.

CON - It is static and the data in the html format is unstructured so it is not ideal of analysis. I would not select this to store inventory.

JSON - PRO and CON

```r
json_data <- toJSON(get_data, pretty = TRUE)
write_json(get_data, "get_data.json")
#save json to directory

print(json_data)
```

```
## [
##   {
##     "Category": "Electronics",
##     "Item.Name": "Smartphone",
##     "Item.ID": 101,
##     "Brand": "TechBrand",
##     "Price": 699.99,
##     "Variation.ID": "101-A",
##     "Variation.Details": "Color: Black, Storage: 64GB "
##   },
```

```
##   {
##     "Category": "Electronics",
##     "Item.Name": "Smartphone",
##     "Item.ID": 101,
##     "Brand": "TechBrand",
##     "Price": 699.99,
##     "Variation.ID": "101-B",
##     "Variation.Details": "Color: White, Storage: 128GB "
##   },
##   {
##     "Category": "Electronics",
##     "Item.Name": "Laptop",
##     "Item.ID": 102,
##     "Brand": "CompuBrand",
##     "Price": 1099.99,
##     "Variation.ID": "102-A",
##     "Variation.Details": "Color: Silver, Storage: 256GB "
##   },
##   {
##     "Category": "Electronics",
##     "Item.Name": "Laptop",
##     "Item.ID": 102,
##     "Brand": "CompuBrand",
##     "Price": 1099.99,
##     "Variation.ID": "102-B",
##     "Variation.Details": "Color: Space Gray, Storage: 512GB "
##   },
##   {
##     "Category": "Home Appliances",
##     "Item.Name": "Refrigerator",
##     "Item.ID": 201,
##     "Brand": "HomeCool",
##     "Price": 899.99,
##     "Variation.ID": "201-A",
##     "Variation.Details": "Color: Stainless Steel, Capacity: 20 cu ft "
##   },
##   {
##     "Category": "Home Appliances",
##     "Item.Name": "Refrigerator",
##     "Item.ID": 201,
##     "Brand": "HomeCool",
##     "Price": 899.99,
##     "Variation.ID": "201-B",
##     "Variation.Details": "Color: White, Capacity: 18 cu ft "
##   },
##   {
##     "Category": "Home Appliances",
##     "Item.Name": "Washing Machine",
##     "Item.ID": 202,
##     "Brand": "CleanTech",
##     "Price": 499.99,
##     "Variation.ID": "202-A",
##     "Variation.Details": "Type: Front Load, Capacity: 4.5 cu ft "
##   },
```

```
##   {
##     "Category": "Home Appliances",
##     "Item.Name": "Washing Machine",
##     "Item.ID": 202,
##     "Brand": "CleanTech",
##     "Price": 499.99,
##     "Variation.ID": "202-B",
##     "Variation.Details": "Type: Top Load, Capacity: 5.0 cu ft "
##   },
##   {
##     "Category": "Clothing",
##     "Item.Name": "T-Shirt",
##     "Item.ID": 301,
##     "Brand": "FashionCo",
##     "Price": 19.99,
##     "Variation.ID": "301-A",
##     "Variation.Details": "Color: Blue, Size: S "
##   },
##   {
##     "Category": "Clothing",
##     "Item.Name": "T-Shirt",
##     "Item.ID": 301,
##     "Brand": "FashionCo",
##     "Price": 19.99,
##     "Variation.ID": "301-B",
##     "Variation.Details": "Color: Red, Size: M "
##   },
##   {
##     "Category": "Clothing",
##     "Item.Name": "T-Shirt",
##     "Item.ID": 301,
##     "Brand": "FashionCo",
##     "Price": 19.99,
##     "Variation.ID": "301-C",
##     "Variation.Details": "Color: Green, Size: L "
##   },
##   {
##     "Category": "Clothing",
##     "Item.Name": "Jeans",
##     "Item.ID": 302,
##     "Brand": "DenimWorks",
##     "Price": 49.99,
##     "Variation.ID": "302-A",
##     "Variation.Details": "Color: Dark Blue, Size: 32 "
##   },
##   {
##     "Category": "Clothing",
##     "Item.Name": "Jeans",
##     "Item.ID": 302,
##     "Brand": "DenimWorks",
##     "Price": 49.99,
##     "Variation.ID": "302-B",
##     "Variation.Details": "Color: Light Blue, Size: 34 "
##   },
```

```
##   {
##     "Category": "Books",
##     "Item.Name": "Fiction Novel",
##     "Item.ID": 401,
##     "Brand": "-",
##     "Price": 14.99,
##     "Variation.ID": "401-A",
##     "Variation.Details": "Format: Hardcover, Language: English "
##   },
##   {
##     "Category": "Books",
##     "Item.Name": "Fiction Novel",
##     "Item.ID": 401,
##     "Brand": "-",
##     "Price": 14.99,
##     "Variation.ID": "401-B",
##     "Variation.Details": "Format: Paperback, Language: Spanish "
##   },
##   {
##     "Category": "Books",
##     "Item.Name": "Non-Fiction Guide",
##     "Item.ID": 402,
##     "Brand": "-",
##     "Price": 24.99,
##     "Variation.ID": "402-A",
##     "Variation.Details": "Format: eBook, Language: English "
##   },
##   {
##     "Category": "Books",
##     "Item.Name": "Non-Fiction Guide",
##     "Item.ID": 402,
##     "Brand": "-",
##     "Price": 24.99,
##     "Variation.ID": "402-B",
##     "Variation.Details": "Format: Paperback, Language: French "
##   },
##   {
##     "Category": "Sports Equipment",
##     "Item.Name": "Basketball",
##     "Item.ID": 501,
##     "Brand": "SportsGear",
##     "Price": 29.99,
##     "Variation.ID": "501-A",
##     "Variation.Details": "Size: Size 7, Color: Orange "
##   },
##   {
##     "Category": "Sports Equipment",
##     "Item.Name": "Tennis Racket",
##     "Item.ID": 502,
##     "Brand": "RacketPro",
##     "Price": 89.99,
##     "Variation.ID": "502-A",
##     "Variation.Details": "Material: Graphite, Color: Black "
##   },
```

```
##   {
##     "Category": "Sports Equipment",
##     "Item.Name": "Tennis Racket",
##     "Item.ID": 502,
##     "Brand": "RacketPro",
##     "Price": 89.99,
##     "Variation.ID": "502-B",
##     "Variation.Details": "Material: Aluminum, Color: Silver "
##   }
## ]
```

PRO - JavaScript Object Notation (JSON) is easy to read and simple text format. It uses key-value pairs which makes the structure easy to understand so in the case of the CUNYMart data, I can see each item as it's own part.

CON - It is not ideal for large data set. Since each item is its own part, I would imagine it is error prone and can get overwhemling with a large dataset. For example, if I have the same category, item name and brand but price differently, each price point would be it's own part.

XML - PRO and CON

```
xml_doc <- xml_new_root("inventory")

for (i in 1:nrow(get_data)) {
  item <- xml_add_child(xml_doc, "item")
  xml_add_child(item, "Category", get_data$Category[i])
  xml_add_child(item, "Item.Name", get_data$Item.Name[i])
  xml_add_child(item, "Item.ID", get_data$Item.ID[i])
  xml_add_child(item, "Brand", get_data$Brand[i])
  xml_add_child(item, "Price", get_data$Price[i])
  xml_add_child(item, "Variation.ID", get_data$Variation.ID[i])
  xml_add_child(item, "Variation.Details", get_data$Variation.Details[i])
}

write_xml(xml_doc, "get_data.xml")
#save xml to directory

print(xml_doc)
```

```
## {xml_document}
## <inventory>
##  [1] <item>\n  <Category>Electronics</Category>\n  <Item.Name>Smartphone</Ite ...
##  [2] <item>\n  <Category>Electronics</Category>\n  <Item.Name>Smartphone</Ite ...
##  [3] <item>\n  <Category>Electronics</Category>\n  <Item.Name>Laptop</Item.Na ...
##  [4] <item>\n  <Category>Electronics</Category>\n  <Item.Name>Laptop</Item.Na ...
##  [5] <item>\n  <Category>Home Appliances</Category>\n  <Item.Name>Refrigerato ...
##  [6] <item>\n  <Category>Home Appliances</Category>\n  <Item.Name>Refrigerato ...
##  [7] <item>\n  <Category>Home Appliances</Category>\n  <Item.Name>Washing Mac ...
##  [8] <item>\n  <Category>Home Appliances</Category>\n  <Item.Name>Washing Mac ...
##  [9] <item>\n  <Category>Clothing</Category>\n  <Item.Name>T-Shirt</Item.Name ...
## [10] <item>\n  <Category>Clothing</Category>\n  <Item.Name>T-Shirt</Item.Name ...
## [11] <item>\n  <Category>Clothing</Category>\n  <Item.Name>T-Shirt</Item.Name ...
```

```
## [12] <item>\n  <Category>Clothing</Category>\n  <Item.Name>Jeans</Item.Name>\ ...
## [13] <item>\n  <Category>Clothing</Category>\n  <Item.Name>Jeans</Item.Name>\ ...
## [14] <item>\n  <Category>Books</Category>\n  <Item.Name>Fiction Novel</Item.N ...
## [15] <item>\n  <Category>Books</Category>\n  <Item.Name>Fiction Novel</Item.N ...
## [16] <item>\n  <Category>Books</Category>\n  <Item.Name>Non-Fiction Guide</It ...
## [17] <item>\n  <Category>Books</Category>\n  <Item.Name>Non-Fiction Guide</It ...
## [18] <item>\n  <Category>Sports Equipment</Category>\n  <Item.Name>Basketball ...
## [19] <item>\n  <Category>Sports Equipment</Category>\n  <Item.Name>Tennis Rac ...
## [20] <item>\n  <Category>Sports Equipment</Category>\n  <Item.Name>Tennis Rac ...
```

XML - extensible markup language similar to HTML but unlike HTML, it focuses on carrying data - not just how the data looks.

PRO - like the other format, it is easily readable by a human. It also can nest elements which I can see as helpful for complex relationships.So if I have multiple categories for an item, this would be helpful in storing.

CON - the file size can be inefficient which leads to slower processing. For large inventory dataset, this might not be ideal.

Parquet - PRO and CON

```r
par <- write_parquet(get_data, "get_data.parquet")
# save parquet to directory

print(par)
```

```
##            Category         Item.Name Item.ID      Brand   Price Variation.ID
## 1       Electronics        Smartphone     101  TechBrand  699.99        101-A
## 2       Electronics        Smartphone     101  TechBrand  699.99        101-B
## 3       Electronics            Laptop     102 CompuBrand 1099.99        102-A
## 4       Electronics            Laptop     102 CompuBrand 1099.99        102-B
## 5   Home Appliances      Refrigerator     201   HomeCool  899.99        201-A
## 6   Home Appliances      Refrigerator     201   HomeCool  899.99        201-B
## 7   Home Appliances   Washing Machine     202  CleanTech  499.99        202-A
## 8   Home Appliances   Washing Machine     202  CleanTech  499.99        202-B
## 9          Clothing           T-Shirt     301  FashionCo   19.99        301-A
## 10         Clothing           T-Shirt     301  FashionCo   19.99        301-B
## 11         Clothing           T-Shirt     301  FashionCo   19.99        301-C
## 12         Clothing             Jeans     302 DenimWorks   49.99        302-A
## 13         Clothing             Jeans     302 DenimWorks   49.99        302-B
## 14            Books     Fiction Novel     401          -   14.99        401-A
## 15            Books     Fiction Novel     401          -   14.99        401-B
## 16            Books Non-Fiction Guide     402          -   24.99        402-A
## 17            Books Non-Fiction Guide     402          -   24.99        402-B
## 18 Sports Equipment        Basketball     501 SportsGear   29.99        501-A
## 19 Sports Equipment     Tennis Racket     502  RacketPro   89.99        502-A
## 20 Sports Equipment     Tennis Racket     502  RacketPro   89.99        502-B
##                            Variation.Details
## 1                Color: Black, Storage: 64GB
## 2               Color: White, Storage: 128GB
## 3              Color: Silver, Storage: 256GB
## 4           Color: Space Gray, Storage: 512GB
```

```
## 5  Color: Stainless Steel, Capacity: 20 cu ft
## 6            Color: White, Capacity: 18 cu ft
## 7       Type: Front Load, Capacity: 4.5 cu ft
## 8         Type: Top Load, Capacity: 5.0 cu ft
## 9                        Color: Blue, Size: S
## 10                        Color: Red, Size: M
## 11                      Color: Green, Size: L
## 12                Color: Dark Blue, Size: 32
## 13               Color: Light Blue, Size: 34
## 14       Format: Hardcover, Language: English
## 15       Format: Paperback, Language: Spanish
## 16           Format: eBook, Language: English
## 17        Format: Paperback, Language: French
## 18                Size: Size 7, Color: Orange
## 19           Material: Graphite, Color: Black
## 20          Material: Aluminum, Color: Silver
```

Parquet - a storage file format that stores data by columns and allow for fast processing.

PRO - it is structured data unlike JSON and it can read only the necessary information unlike some format where it reads all the data.

CON - it might not be as widely supported as the other format. For example, with HTML - all browser supports it so regardless of the software installed, it can be opened.

Analysis JSON

```
json_data <- fromJSON("get_data.json")

top_ten_expensive_json <- json_data %>%
  arrange(desc(Price)) %>%
  select(Item.Name, Price, Category, Variation.Details) %>%
  head(10)

print(top_ten_expensive_json)
```

```
##          Item.Name   Price         Category
## 1           Laptop 1099.99      Electronics
## 2           Laptop 1099.99      Electronics
## 3     Refrigerator  899.99  Home Appliances
## 4     Refrigerator  899.99  Home Appliances
## 5       Smartphone  699.99      Electronics
## 6       Smartphone  699.99      Electronics
## 7  Washing Machine  499.99  Home Appliances
## 8  Washing Machine  499.99  Home Appliances
## 9    Tennis Racket   89.99 Sports Equipment
## 10   Tennis Racket   89.99 Sports Equipment
##                            Variation.Details
## 1               Color: Silver, Storage: 256GB
## 2           Color: Space Gray, Storage: 512GB
## 3  Color: Stainless Steel, Capacity: 20 cu ft
## 4            Color: White, Capacity: 18 cu ft
```

```
## 5                 Color: Black, Storage: 64GB
## 6                Color: White, Storage: 128GB
## 7       Type: Front Load, Capacity: 4.5 cu ft
## 8         Type: Top Load, Capacity: 5.0 cu ft
## 9            Material: Graphite, Color: Black
## 10          Material: Aluminum, Color: Silver
```

Grab the JSON file from directory and used that for analysis to see the top ten most expensive items. This was easily done (being able to just grab it from the file and work on it).

Analysis XML

```
xml_data <- read_xml("get_data.xml")

print(xml_data)
```

```
## {xml_document}
## <inventory>
##  [1] <item>\n  <Category>Electronics</Category>\n  <Item.Name>Smartphone</Ite ...
##  [2] <item>\n  <Category>Electronics</Category>\n  <Item.Name>Smartphone</Ite ...
##  [3] <item>\n  <Category>Electronics</Category>\n  <Item.Name>Laptop</Item.Na ...
##  [4] <item>\n  <Category>Electronics</Category>\n  <Item.Name>Laptop</Item.Na ...
##  [5] <item>\n  <Category>Home Appliances</Category>\n  <Item.Name>Refrigerato ...
##  [6] <item>\n  <Category>Home Appliances</Category>\n  <Item.Name>Refrigerato ...
##  [7] <item>\n  <Category>Home Appliances</Category>\n  <Item.Name>Washing Mac ...
##  [8] <item>\n  <Category>Home Appliances</Category>\n  <Item.Name>Washing Mac ...
##  [9] <item>\n  <Category>Clothing</Category>\n  <Item.Name>T-Shirt</Item.Name ...
## [10] <item>\n  <Category>Clothing</Category>\n  <Item.Name>T-Shirt</Item.Name ...
## [11] <item>\n  <Category>Clothing</Category>\n  <Item.Name>T-Shirt</Item.Name ...
## [12] <item>\n  <Category>Clothing</Category>\n  <Item.Name>Jeans</Item.Name>\ ...
## [13] <item>\n  <Category>Clothing</Category>\n  <Item.Name>Jeans</Item.Name>\ ...
## [14] <item>\n  <Category>Books</Category>\n  <Item.Name>Fiction Novel</Item.N ...
## [15] <item>\n  <Category>Books</Category>\n  <Item.Name>Fiction Novel</Item.N ...
## [16] <item>\n  <Category>Books</Category>\n  <Item.Name>Non-Fiction Guide</It ...
## [17] <item>\n  <Category>Books</Category>\n  <Item.Name>Non-Fiction Guide</It ...
## [18] <item>\n  <Category>Sports Equipment</Category>\n  <Item.Name>Basketball ...
## [19] <item>\n  <Category>Sports Equipment</Category>\n  <Item.Name>Tennis Rac ...
## [20] <item>\n  <Category>Sports Equipment</Category>\n  <Item.Name>Tennis Rac ...
```

```
# Extract all <item> elements
xml_items <- xml_data %>% xml_find_all("//item")


xml_df <- tibble(
  Category = xml_items %>% xml_find_first("Category") %>% xml_text(),
  Item_Name = xml_items %>% xml_find_first("Item.Name") %>% xml_text(),
  Item_ID = xml_items %>% xml_find_first("Item.ID") %>% xml_text() %>% as.integer(),
  Brand = xml_items %>% xml_find_first("Brand") %>% xml_text(),
  Price = xml_items %>% xml_find_first("Price") %>% xml_text() %>% as.numeric(),
  Variation_ID = xml_items %>% xml_find_first("Variation.ID") %>% xml_text(),
  Variation_Details = xml_items %>% xml_find_first("Variation.Details") %>% xml_text()
```

```
)

top_ten_expensive_xml <- xml_df %>%
  arrange(desc(Price)) %>%
  select(Item_Name, Price, Category) %>%
  head(10)

print(top_ten_expensive_xml)
```

```
## # A tibble: 10 x 3
##    Item_Name        Price Category
##    <chr>            <dbl> <chr>
##  1 Laptop          1100.  Electronics
##  2 Laptop          1100.  Electronics
##  3 Refrigerator     900.  Home Appliances
##  4 Refrigerator     900.  Home Appliances
##  5 Smartphone       700.  Electronics
##  6 Smartphone       700.  Electronics
##  7 Washing Machine  500.  Home Appliances
##  8 Washing Machine  500.  Home Appliances
##  9 Tennis Racket     90.0 Sports Equipment
## 10 Tennis Racket     90.0 Sports Equipment
```

Grab the XML file from directory and used that for analysis to see the top ten most expensive items. This was more work than JSON as I had to convert the file and the column names.

Analysis HTML

```
html_data <- read_html("get_data.html")

html_table <- html_data %>%
  html_element("table") %>%
  # Get the first table

  html_table(fill = TRUE)
# Convert it to a data frame

# View the extracted table
print(html_table)
```

```
## # A tibble: 20 x 7
##    Category        Item.Name Item.ID Brand  Price Variation.ID Variation.Details
##    <chr>           <chr>       <int> <chr>  <dbl> <chr>        <chr>
##  1 Electronics     Smartpho~     101 Tech~  700.  101-A        Color: Black, St~
##  2 Electronics     Smartpho~     101 Tech~  700.  101-B        Color: White, St~
##  3 Electronics     Laptop        102 Comp~ 1100.  102-A        Color: Silver, S~
##  4 Electronics     Laptop        102 Comp~ 1100.  102-B        Color: Space Gra~
##  5 Home Appliances Refriger~     201 Home~  900.  201-A        Color: Stainless~
##  6 Home Appliances Refriger~     201 Home~  900.  201-B        Color: White, Ca~
##  7 Home Appliances Washing ~     202 Clea~  500.  202-A        Type: Front Load~
```

```
##  8 Home Appliances Washing ~     202 Clea~  500.  202-B        Type: Top Load, ~
##  9 Clothing        T-Shirt       301 Fash~   20.0 301-A        Color: Blue, Siz~
## 10 Clothing        T-Shirt       301 Fash~   20.0 301-B        Color: Red, Size~
## 11 Clothing        T-Shirt       301 Fash~   20.0 301-C        Color: Green, Si~
## 12 Clothing        Jeans         302 Deni~   50.0 302-A        Color: Dark Blue~
## 13 Clothing        Jeans         302 Deni~   50.0 302-B        Color: Light Blu~
## 14 Books           Fiction ~     401 -       15.0 401-A        Format: Hardcove~
## 15 Books           Fiction ~     401 -       15.0 401-B        Format: Paperbac~
## 16 Books           Non-Fict~     402 -       25.0 402-A        Format: eBook, L~
## 17 Books           Non-Fict~     402 -       25.0 402-B        Format: Paperbac~
## 18 Sports Equipme~ Basketba~     501 Spor~   30.0 501-A        Size: Size 7, Co~
## 19 Sports Equipme~ Tennis R~     502 Rack~   90.0 502-A        Material: Graphi~
## 20 Sports Equipme~ Tennis R~     502 Rack~   90.0 502-B        Material: Alumin~
```

```r
top_ten_expensive_html <- html_table %>%
  arrange(desc(Price)) %>%
  select(Item.Name, Price, Category) %>%
  head(10)
print(top_ten_expensive_html)
```

```
## # A tibble: 10 x 3
##    Item.Name        Price Category
##    <chr>            <dbl> <chr>
##  1 Laptop          1100.  Electronics
##  2 Laptop          1100.  Electronics
##  3 Refrigerator     900.  Home Appliances
##  4 Refrigerator     900.  Home Appliances
##  5 Smartphone       700.  Electronics
##  6 Smartphone       700.  Electronics
##  7 Washing Machine  500.  Home Appliances
##  8 Washing Machine  500.  Home Appliances
##  9 Tennis Racket     90.0 Sports Equipment
## 10 Tennis Racket     90.0 Sports Equipment
```

Grab the HTML file from directory and used that for analysis to see the top ten most expensive items. This was similar to XML as I had to convert and go through extra steps. Up to this point, working with JSON file is much easier than XML and HTML

Analysis parquet

```r
parquet_data <- read_parquet("get_data.parquet")

top_ten_expensive_parquet <- parquet_data %>%
  arrange(desc(Price)) %>%
  select(Item.Name, Price, Category) %>%
  head(10)
print(top_ten_expensive_parquet)
```

```
## # A tibble: 10 x 3
##    Item.Name        Price Category
##    <chr>            <dbl> <chr>
```

```
##  1 Laptop          1100.  Electronics
##  2 Laptop          1100.  Electronics
##  3 Refrigerator     900.  Home Appliances
##  4 Refrigerator     900.  Home Appliances
##  5 Smartphone       700.  Electronics
##  6 Smartphone       700.  Electronics
##  7 Washing Machine  500.  Home Appliances
##  8 Washing Machine  500.  Home Appliances
##  9 Tennis Racket     90.0 Sports Equipment
## 10 Tennis Racket     90.0 Sports Equipment
```

Grab the parquet file from directory and used that for analysis to see the top ten most expensive items. This was similar to JSON as I did not have to format like I had to do for XML and HTML.

CONCLUSION

Depending on the goal, each format has it's own pros and cons. Since this data set was small, I think it worked well with all four formats. I think the impact of each pros and cons is minimal in that aspect, but if we are looking at a larger data set, I would prefer parquet.