Wikipedia

December 2017

Introduction

Wikipedia stores most of its tabular data in HTML tables. The R function `XML::readHTMLTable` scrapes all HTML tables from any website (or HTML file). As an initial foray into downloading data directly from the internet into R, Wikipedia tables provide a nice introduction. In the supplementary R Markdown file associated with the Wikipedia data analysis, we also walk through some of the useful aspects of using `dplyr` to wrangle the data[1].

Data information & loading data

The Wikipedia analysis given in the fully curated files explores an HTML table on sales of music (physical and digital) in 2014, https://en.wikipedia.org/wiki/Music_industry. One variable gives an indication of whether the retail value of the music sales has increased or decreased. Using the country-level music data, we perform a t-test, a Wilcoxon rank sum test, data transformations, and boxplots to investigate music retail sales (analysis given in supplementary materials, not shown here). By grouping the data into two categories we can investigate whether there is any statistical difference between the total average retail sales (in US$) between those countries for whom retail sales increased versus those that decreased. The p-value for the initial t-test is reasonably large, but the boxplot shows that the difference in variability across the two groups is also large with a sample that either has large outliers or a long skewed right tail. Because the technical assumptions do not appear to be met, a log transformation of the data or a non-parametric test might be better assessments of the data (see Figure 2). The analysis leads to conversations about the source of the data and the reasons why p-values are non-significant. The example extends easily to each student choosing their own Wikipedia page and data table, graphical representations, and statistical analyses.

```
#read in our data using a web address, strings as factors
# is set to false, because we are not treating the strings as categories
url = RCurl::getURL("https://en.wikipedia.org/wiki/Music_industry#Total_revenue_by_year")
parsedDoc = readHTMLTable(url, stringsAsFactors=FALSE)

#return value is a list of data frames or matrices
mytable = data.frame(parsedDoc[[5]]) #accessing the 6th HTML table

# check out the other HTML tables:  parsedDoc[[7]]


names(mytable) #what variables are in the table?
```

```
## [1] "Ranking"                    "Market"
## [3] "Retail.value.US...millions." "X...Change"
## [5] "Physical"                   "Digital"
## [7] "Performance.rights"         "Synchronization"
```

```
names(mytable) = c("Rank", "Market", "Retail", "PerChange", "Physical",
                   "Digital", "PerfRights", "Sync")
names(mytable)  #these names of variables will be slightly eaiser to use
```

[1] Original idea for this example provided by Nick Horton, Amherst College.

```
## [1] "Rank"       "Market"     "Retail"     "PerChange"  "Physical"
## [6] "Digital"    "PerfRights" "Sync"
```

`parsedDoc` is a list of all the html tables found on the Wikipedia page. If you look through them, some are just useless things like header data and bibliography, etc. Consult the "How to Use R's XML Package" document to find and view your table.

Notice that the data do not appear exactly as they did on Wikipedia. We will need to fix these problems with wrangling!

Cleaning your Table

In this section we will use the `dplyr` package to remove problematic elements from your table, including but not limited to (there is no limit to the strange things you will have to clean!) ill formatted entries, NaN's (not a number), as well as columns of data you do not need. There are additionally elements you may want to add to the dataset, for example if you find multiple tables corresponding to the same time period, you may want to splice these tables together and compare entries with a t-test or plot the entries on the same graph!

Right now your data are in a Data Frame format, so many of these operations are straight from the data frame specification. As a matter of fact, `dplyr` limits your options when it comes to manipulation: this constraint helps you organize ans systematize your approach to data manipulation. `dplyr` can work with data frames as is, but if you are dealing with large data, it is worthwhile to convert them to a `tbl_df`: this is a wrapper around a data frame that will not accidentally print a lot of data to the screen.

Grabbing a Column using Select

When you read your table in, R should have automatically identified the row and column names, meaning that you should be able to access the ith row using the ith row's variable name. Run `attach(mytable)` for easy access of each column. If you don't want to do this, you can select the columns using `dplyr::select(tableName,columnNumber)`.

What dplyr allows you to do

You don't have to trust me! Check out this awesome tutorial http://cran.rstudio.com/web/packages/dplyr/vignettes/introduction.html.

The final data table we'll use also has a variable called PerChange which is based on the column in the Wikipedia table "% Change". The percent change is presumably over the last year, but a glimpse at the source data does not provide any additional information, http://www.riaj.or.jp/e/issue/pdf/RIAJ2015E.pdf. A great conversation with your students would be centered around how the observations are collected (sampled), how the variables are defined, and whether the data appear to be of high quality.

```
music = mytable

dashfunc <- function(x){sub("(-|-|-)","-",x)}  # need to substitute out the hyphen for a minus sign
music = data.frame(apply(music, 2, dashfunc))
music[,-c(1:2)] = apply(music[,-c(1:2)], 2, readr::parse_number)
music = music %>% mutate(change = ifelse(PerChange < 0, "decrease", "increase"))

glimpse(music)
```

```
## Observations: 20
## Variables: 9
## $ Rank       <fctr> 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15,...
## $ Market     <fctr> United States, Japan, Germany, United Kingdom, Fra...
```

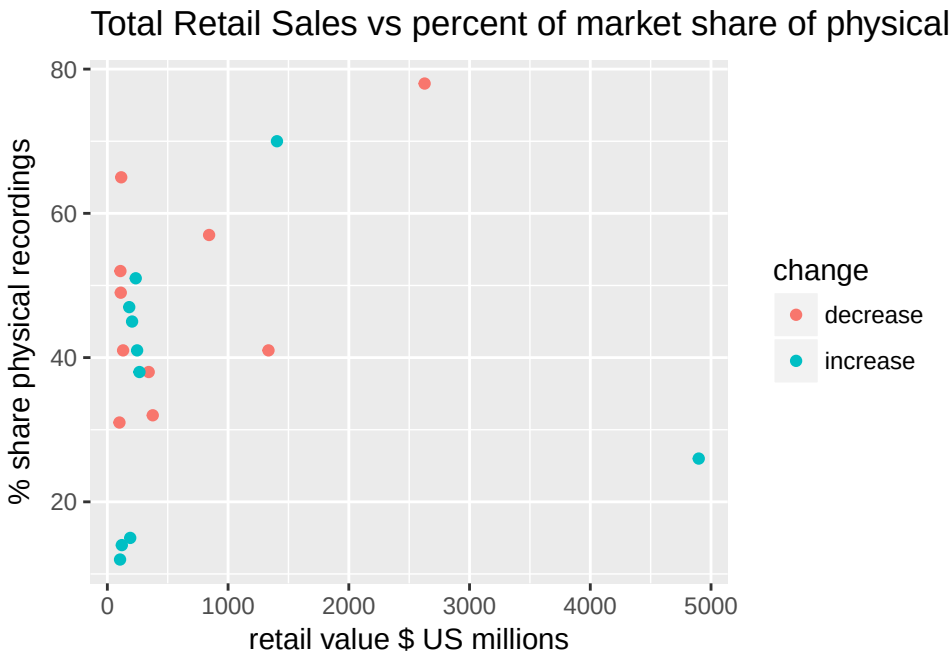

Figure 1: The y-axis gives the percentage of the market share of total music industry revenue which is due to physical recordings. The x-axis gives the total retail sales in US dollars. The coloring is broken down over whether there has been a decrease or increase in retail sales (presumably over the previous year).

```
## $ Retail     <dbl> 4898.3, 2627.9, 1404.8, 1334.6, 842.8, 376.1, 342.5...
## $ PerChange  <dbl> 2.1, -5.5, 1.9, -2.8, -3.4, -6.8, -11.3, 19.2, 2.0,...
## $ Physical   <dbl> 26, 78, 70, 41, 57, 32, 38, 38, 41, 51, 45, 15, 47,...
## $ Digital    <dbl> 71, 17, 22, 45, 27, 56, 53, 58, 37, 33, 38, 73, 35,...
## $ PerfRights <dbl> 0, 3, 7, 12, 13, 9, 6, 3, 21, 13, 16, 10, 17, 4, 12...
## $ Sync       <dbl> 4, 1, 1, 2, 3, 2, 2, 1, 1, 3, 1, 2, 1, 2, 2, 1, 0, ...
## $ change     <chr> "increase", "decrease", "increase", "decrease", "de...
```

Using dynamic data within a typical classroom

The data in the table include 4 columns representing the information on how retail sales are broken down: physical (e.g., CDs), digital (e.g., iTunes), performance rights (e.g., another band playing the song), and synchronization (e.g., songs played on TV or video games).

Just having a table of data doesn't do much for us. Let's use our statistical background and intuition about what interesting stories might be told about our data. The percent change variable speaks to the direction of the change in retail sales in a given country. Indeed, it is worth considering variables that might speak to the changing world of music sales.

```
ggplot(music, aes(x=Retail, y=Physical, colour=change)) +
  geom_point() +
  xlab("retail value $ US millions") +
  ylab("% share physical recordings") +
  ggtitle("Total Retail Sales vs percent of market share of physical recordings")
```

By grouping the data into two categories we can investigate whether there is any statistical difference between the total average retail sales (in US$) between those countries for whom retail sales increased versus those

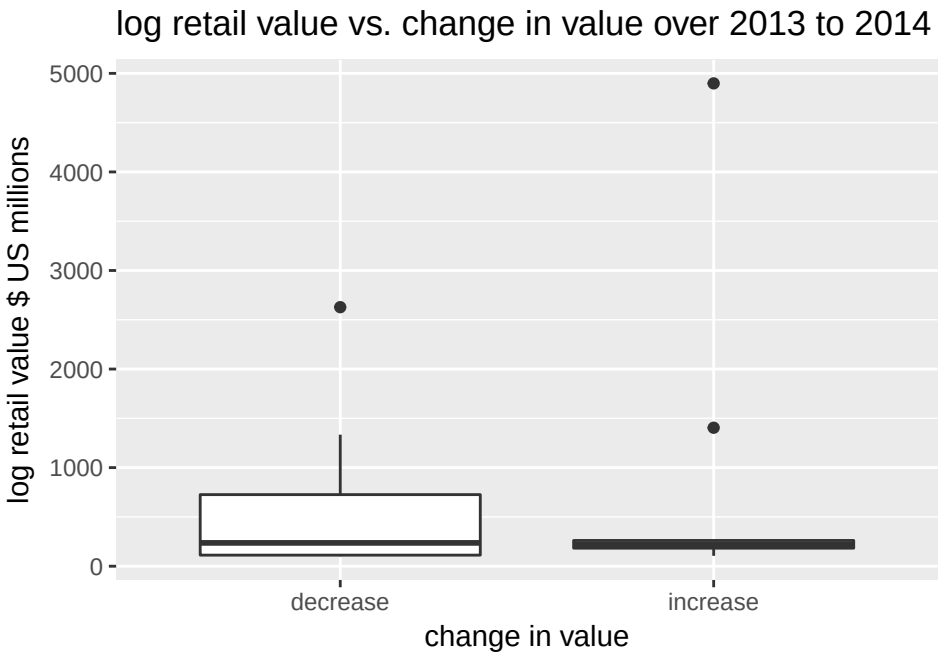

Figure 2: The retail value of music sales broken down by whether or not the sales have increased or decreased (presumably over the previous year, although the Wikipedia documentation does not specify the time period over which the change is measured).

that decreased.

```
t.test(Retail ~ change, data=music)
```

```
##
##  Welch Two Sample t-test
##
## data:  Retail by change
## t = -0.32701, df = 13.937, p-value = 0.7485
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -1332.5821   980.1221
## sample estimates:
## mean in group decrease mean in group increase
##                 608.87                 785.10
```

```
ggplot(music, aes(x=change, y=Retail)) +
  geom_boxplot() +
  ylab("log retail value $ US millions") +
  xlab("change in value") +
  ggtitle("log retail value vs. change in value over 2013 to 2014")
```

The p-value is reasonably large, but the boxplot shows that the difference in variability across the two groups is also large with a sample that either has large outliers or a long skewed right tail. Because the technical assumptions do not appear to be met, a log transformation of the data or a non-parametric test might be better assessments of the data.

```
ggplot(music, aes(x=change, y=log(Retail))) +
  geom_boxplot() +
  ylab("log retail value $ US millions") +
```

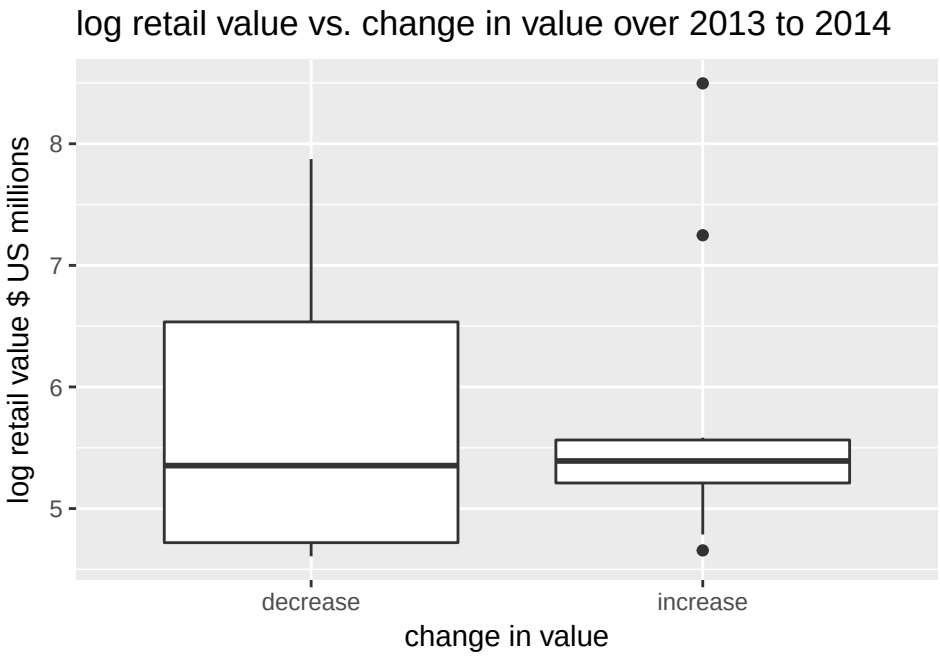

Figure 3: The log of the retail value of music sales broken down by whether or not the sales have increased or decreased (presumably over the previous year, although the Wikipedia documentation does not specify the time period over which the change is measured).

```
  xlab("change in value") +
  ggtitle("log retail value vs. change in value over 2013 to 2014")
```

```
t.test(log(Retail) ~ change, data=music)
```

```
##
##  Welch Two Sample t-test
##
## data:  log(Retail) by change
## t = -0.058478, df = 18, p-value = 0.954
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -1.154839  1.092291
## sample estimates:
## mean in group decrease mean in group increase
##               5.718957               5.750230
```

```
wilcox.test(Retail ~ change, data=music)
```

```
##
##  Wilcoxon rank sum test
##
## data:  Retail by change
## W = 46, p-value = 0.7959
## alternative hypothesis: true location shift is not equal to 0
```

The technical assumptions for the t-test seem better now (though still not great), but neither the updated p-value nor the p-value on the Wilcoxon Rank Sum test are significant. The data for the top 20 countries (in terms of market share of retail music sales) are allocated into the "increase" and "decrease" groups at no

different an organization than random chance (with respect to average retail sales).

The classroom discussion can open up to interpretation of p-values. We might have expected those countries with increased retail sales to have a higher average retail sale! So why didn't they? Possibly the test is not powerful enough - sample size is not high with these data. Possibly there is no effect. Possibly there was a sampling bias - maybe the top 20 countries are systematically different from other countries? What other suggestions can your students come up with to consider the data?

Thinking outside the box

Among the variables in the Wikipedia music dataset are the breakdown (percentages) of how the retail sales are distributed across physical, digital, performance rights, and synchronization. We might want to see whether there is a dependency of total retail sales on the breakdown of types of products. The problem is not well suited to introductory statistics as there is not an obvious statistic we can use within a sampling distribution (to create a p-value, etc.). Because there does not seem to be an obvious mechanism for evaluating the breakdown of products (and how "different" they are), we consider an ad-hoc measure and perform a permutation test to assess significance.

One way to measure a discrepancy between the retail sales and the consistency of product breakdown is to correlate the retail sales with the sum of squared distances from the average breakdown of product types. We see that the metric we created to find a relationship between retail sales and breakdown of types of product does not show significance.

A cursory plot shows us that there is likely a relationship (slope) between the percent of retail which is physical and the total retail sales. The size of the plots speaks to the digital percent which is quite high for some low total retail sales but also quite high for the largest total retail sales. The color says that the largest performance rights countries have middle range retail sales. And we clearly understand that the four percentage variables are constrained to sum to one.

```
ggplot(music, aes(y=log(Retail), x=Physical, size=Digital, color=PerfRights) ) +
  geom_point()
```

Because there does not seem to be an obvious mechanism for evaluating the breakdown of products (and how "different" they are), we will consider an ad-hoc measure and then perform a permutation test to assess significance. The average breakdown of retail sales is given by the following values. One way to measure a discrepancy between the retail sales and the consistency of product breakdown is to correlate the retail sales with the sum of squared distances from the average breakdown of product types.

```
apply(music[,3:8],2,mean)
```

```
##     Retail  PerChange   Physical    Digital PerfRights       Sync
##    696.985     -0.215     42.150     46.150      9.900      1.650
```

```
music = music %>% mutate(PerCh2 = (PerChange-mean(PerChange)^2),
                         Phy2 = (Physical - mean(Physical))^2,
                         Dig2 = (Digital - mean(Digital))^2,
                         PerRt2 = (PerfRights - mean(PerfRights))^2,
                         Sync2 = (Sync - mean(Sync))^2,
                         break.SSE = PerCh2 + Phy2 + Dig2 + PerRt2 + Sync2)

ggplot(music, aes(x=break.SSE, y=log(Retail))) + geom_point()
```

```
mosaic::cor(log(Retail) ~ break.SSE, data=music)
```

```
## [1] 0.153715
```

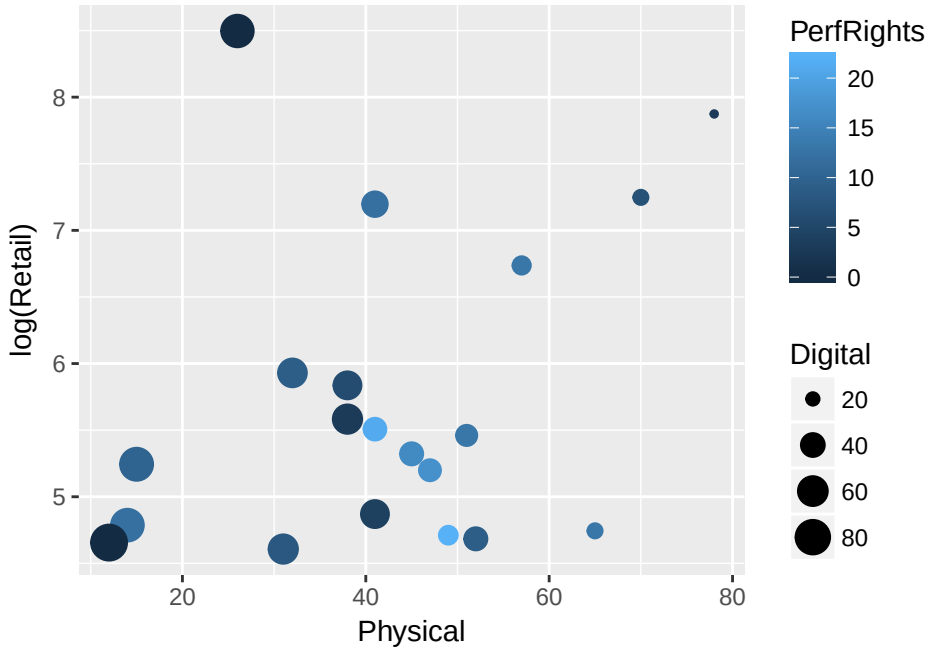

Figure 4: Natural log of retail sales by amoung of physical sales. The size of the point is given by the amount of digit sales. The color is given by the amount of revenue due to performance rights.

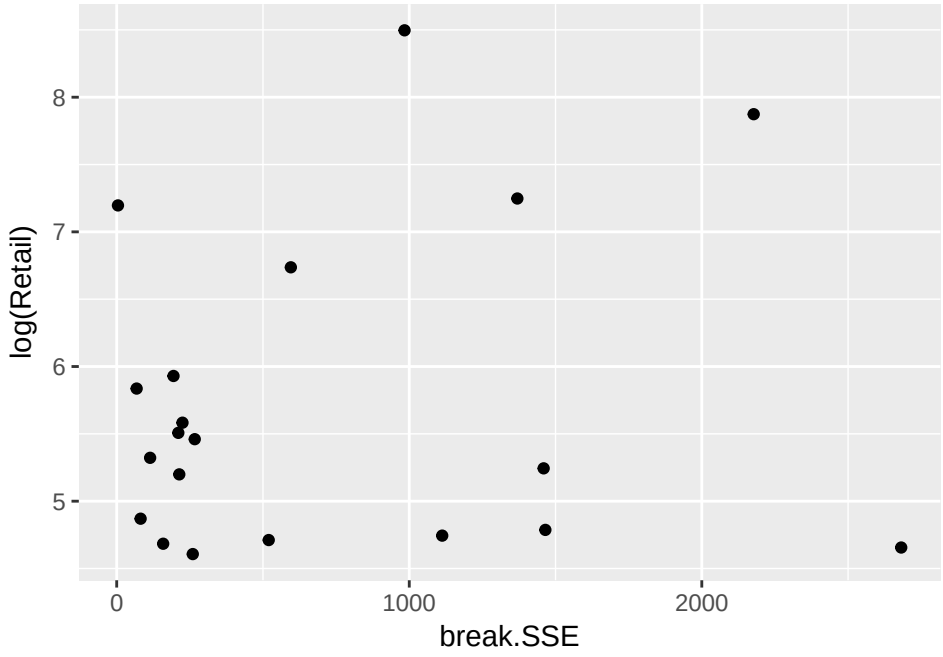

Figure 5: Natural log of the retail versus the breakdown SSE (the sum of squared errors from average for each revenue measure).

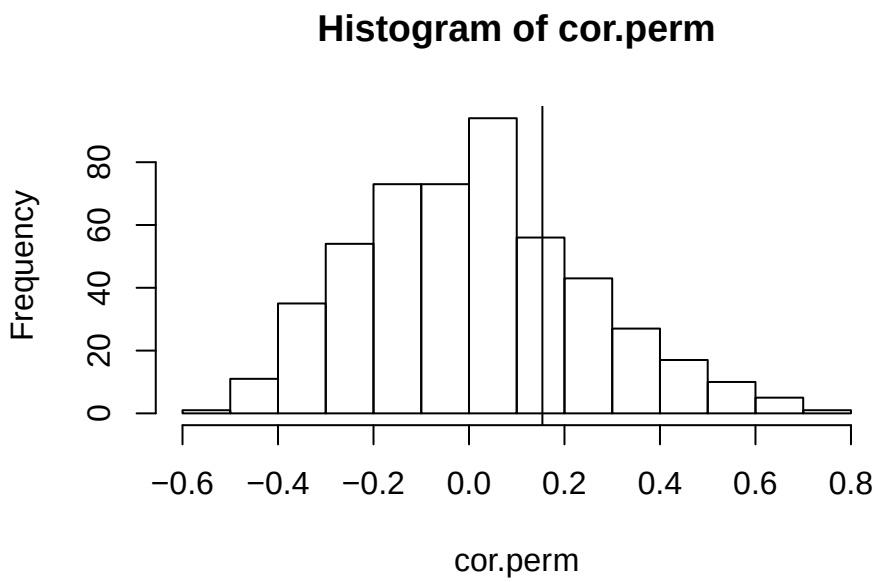

Figure 6: Histogram of the correlation between the permuted retail sales and the measure of breakdown SSE. The horizontal line gives the observed correlation value, indicating no deviation from the null hypothesis.

The correlation between our created measure (breakdown SSE) and the log of the retail sales is low, but we don't have a sense for how low. We can permute the data to see what the relationship between the log retails sales and breakdown SSE would be just by chance.

```
cor.perm = c()
reps = 500
for(j in 1:reps){
  perm = sample(1:20, replace=FALSE)
  cor.perm = c(cor.perm, mosaic::cor(log(Retail)[perm] ~ break.SSE, data=music))}

hist(cor.perm)
abline(v=mosaic::cor(log(Retail) ~ break.SSE, data=music)  )
```

```
sum(cor.perm > mosaic::cor(log(Retail) ~ break.SSE, data=music)) / reps
```

```
## [1] 0.254
```

From the histogram and empirical p-value, we see that the metric we created to find a relationship between retail sales and breakdown of types of product does not show significance.

Additional ideas for analysis:

A student project could be to think about different ways to measure how the breakdowns can be considered to be different.