Notes on the data:

This is a modified version [of the computers dataset](https://rdrr.io/cran/Ecdat/man/Computers.html) that I retrieved from the Ecdat package in R.

**Data prep & profiling**

First we’ll use Excel Power Query to profile the data.

* Click in dataset > Data > Get Data > From Table/Range
* Data should be detected as table correctly
* In the Power Query Editor go to View and turn on Column quality, Column distribution, Column profile
* Make sure to turn on Column profiling for ENTIRE dataset!
  + See at bottom of screen

Looks like there are some odd things happening here:

* Check the headers at the top of each column… which ones have missing values?
  + screen and premium
  + We are going to filter these out for now, just in case.
* We can also see the number of unique values for each column. ram is odd, there is 1 unique value let’s click on that column. We will see a larger visualization of this variable’s distribution:

Chart, histogram

Description automatically generated

* + I don’t think any PC has 160 RAM, this is an issue! We’ll filter it too.
  + cd has that issue too! There is an errant “blue” in there, let’s filter it.
* One more thing, we are going to add an index column to this data. You’ll see why soon 😊
  + Add Column > Index Column > From 1

We are done! Click File > Close & Load to > PivotChart.

**Ad-hoc analyzing**

For this part we are going to lean on PivotTables to be able to reshape and visualize our data.

PivotTables are designed to *aggregate* or roll up our data, but with the index column we get a helpful workaround.

I would like to compare the overall distribution of prices for computers with and without a CD-ROM. A histogram is a good choice here.

Excel does *not* have a way to overlay two histograms in the same plot but we will have a way here!

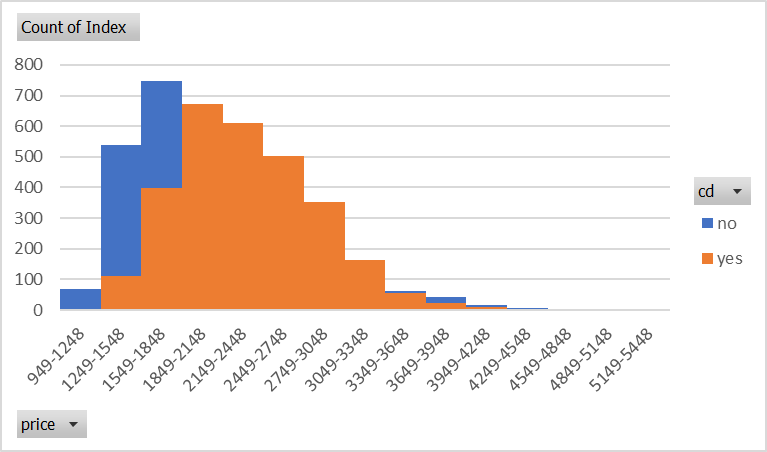
* Drag price to Axis, cd to Legend and Count of Index to Value

Graphical user interface, application, table, Excel

Description automatically generated

* It’s going to look pretty bad right now, let’s fix that:
  + Right-click on the price along the Axis > Group > OK
  + Right-click any of the resulting bars on the PivotChart
    - Set Series Overlap to 100%
    - Set Gap Width to 0%
* Probably still need to adjust the groups in the PivotChart, go ahead and do that.

Your completed analysis should look something like this:



**Pulling data fast**

Power Query and PivotCharts are awesome and work great together, but sometimes speed is the name of the game and for that dynamic arrays are perfect. You can code them directly in Excel and get the results directly in Excel, no drag-and-drop or launching a second editor.

As the name suggests dynamic arrays will return an entire array (like a range) of cells as a result, and they will automatically update with any changes (compare to the old Ctrl + Shift + Enter-type formulas).

We want to analyze the CLEANED copy of the Power Query data, so let’s load that into the workbook:

* Data > Queries & Connections
* Right-click on computers query > Reference
* Close & Load
* Name the table something like computers\_clean in Formulas > Name Manager

OK, let’s try some dynamic arrays now!

UNIQUE() – This should probably really say “distinct,” but it will remove duplicates in the data.

* UNIQUE(computers\_clean) will give the number of unique rows
* Do ROWS(UNIQUE(computers\_clean)) = ROWS(computers\_clean)? Why?

FILTER() – We can filter to get resulting records.

* We only want the premium computers?
  + FILTER(computers\_clean, computers\_clean[premium] = "yes")
* Add the headers to the results: computers\_clean[#Headers]
* AND criteria with \*
  + FILTER(computers\_clean,(computers\_clean[premium]="yes")\*computers\_clean[speed]>30)
* OR criteria with +
  + FILTER(computers\_clean,(computers\_clean[premium]="yes+computers\_clean[speed]>30

SORTBY() – This function lets you sort by multiple columns and you can return whichever arrays you want. Operates a lot like SUMIFS().

* Sort by price, descending: =SORTBY(computers\_clean, computers\_clean[price], -1)
* Sort by speed descending and premium ascending: =SORTBY(computers\_clean, computers\_clean[speed], -1, computers\_clean[premium], 1)
* Sort by price, descending but *only* return the index column: =SORTBY(computers\_clean[Index], computers\_clean[price], -1)

Graphical user interface, application

Description automatically generated