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 with Power Query

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.

# Visualizing results with PivotTables & Charts

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!

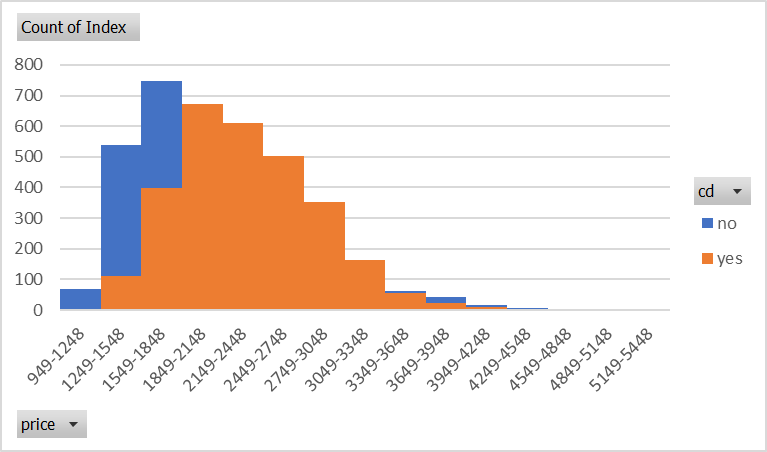
* Drag price to Axis, cd to Legend and Count of Index to Value

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:



A couple more touches:

* Remove the field labels and add a proper title to the PivotChart
* Adjust the fill transparency of each group so help see the overlapping areas

# Pulling data fast with dynamic arrays

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)

# Foundations of exploratory data analysis

**Frequencies**

Create a PivotTable from the source data.

1. Make a frequency table by selecting categories of interest in the Rows/Columns field, then place a Count of the ID field in the Values section.
   1. To convert a field from a Sum to a Count, double-click on that variable header, and select Count in the “Summarize value field by” menu.

Graphical user interface, application, table, Excel

Description automatically generated

**Downloading the Analysis ToolPak**

[See instructions from Microsoft here](https://support.office.com/en-us/article/Load-the-Analysis-ToolPak-in-Excel-6a63e598-cd6d-42e3-9317-6b40ba1a66b4). Note the process is different for Windows and Mac.

**Descriptive Statistics**

1. Go to the Data tab on the home ribbon.
2. Select Data Analysis from the Analyze group (far right of the menu).
3. Select Descriptive Statistics from the menu.

Graphical user interface, application, table, Excel

Description automatically generated

1. Select your Input Range. This will be Column J, year. If your selection includes a header row, make sure to check on the “Labels in First Row” option.
2. By default, the output will be placed in a new worksheet. If you want it elsewhere, click inside “Output Range.” Make sure to double-click inside the dialog box before selecting a new range, otherwise the input range will be re-written.
3. Check on “Summary Statistics.”

Graphical user interface, application, table, Excel

Description automatically generated

**Histograms**

1. Select your input range and go to Insert -> Charts. Histogram should be your third option. Select that. You can cut and paste the resulting histogram elsewhere in the workbook.
2. To change the number of bins in the histogram, right-click on the X-axis and select Format Axis. You can then customize the X-axis on the side menu. *Note: these features are not available on Excel for Mac.*

Chart

Description automatically generated

# Automating Excel analysis & reporting with Python

1. xlwings comes pre-installed with Anaconda, but you do need to install an Excel add-in to work with UDFs:
   1. Open Anaconda Navigator
   2. Execute xlwings addin install
   3. You should see (and follow!) a message like this:  
      Text

      Description automatically generated
2. In Excel, go to File > Options > Trust Center > Macro Settings.
3. Click on “Trust access to the VBA project object model”  
   Graphical user interface, text, application, email

   Description automatically generated
4. Open Anaconda Navigator
   1. Optional: use the cd command to set the directory where you want
5. Enter xlwings quickstart first\_udf in the cursor  
   Text

   Description automatically generated
6. This will open a first\_udf subfolder in the selected directory.   
   Table

   Description automatically generated with low confidence
7. Try moving your UDF file to somewhere else on your computer like C:\py\_udfs\my\_py\_udfs.py.
   1. Set your PYTHONPATH to C:\py\_udfs and UDF Modules to py\_udfs. Do *not* include the *.py* extension for the UDF Modules name.   
      Graphical user interface, application

      Description automatically generated
8. Go ahead and click Import Functions. If you’re not currently in a *.xlsm* workbook, you will need to create one.
9. Open up your py\_udfs.py file. We will take a look at the hello() function. The idea of a function is to take some input, then do something to it behind the scenes and return something to the user.
   1. You can run the *.py* script in VS Code by right-clicking your selection and running. You may need to download an extension for this.

Graphical user interface, text, application, email

Description automatically generated

1. Let’s create a UDF that can look up the timezone of a phone number. The phone number needs to be listed with the country code, such as +1-440-879-8399 (dashes are not necessary)

import phonenumbers

from phonenumbers import timezone

@xw.func

def phone\_timezone(phone\_number, country\_code):

    number\_parse = phonenumbers.parse(phone\_number, country\_code)

    return timezone.time\_zones\_for\_number(number\_parse)

1. For another example, we can create a UDF that detects whether an email address is in a valid format. This will return a True or False to Excel:

import re

valid\_email\_re = r'\b[A-Za-z0-9.\_%+-]+@[A-Za-z0-9.-]+\.[A-Z|a-z]{2,}\b'

@xw.func

def valid\_email(email):

    if(re.fullmatch(valid\_email\_re, email)):

        return True

    else:

        return False

1. Finally, we can define a function that will take a range and return descriptive statistics of it using pandas. Because we are using an outside data type in this case (the DataFrame) we need an extra argument:

import pandas as pd

@xw.func

@xw.arg("df", pd.DataFrame, index=False, header=True)

def describe(df):

    return df.describe()

# Foundations of data science

**Central limit theorem**

1. Simulate 500 rounds of a roulette spin using RANDBETWEEN(0,36)
2. Plot the resulting frequency distribution as a histogram.
3. Use F9 while in your workbook to refresh it. A screenshot of a cell phone

   Description automatically generated
4. This is a *uniform* distribution.
5. Now simulate a roulette spin 100 times and take the average spin. Do this 500 times and plot the resulting distribution of *sample* means.

A screenshot of a cell phone

Description automatically generated

1. This time we get a normal distribution, due to the central limit theorem.

**Law of large numbers: large-numbers.xlsx**

1. Simulate a roulette toss 500 times in Column B: RANDBETWEEN(0,36)
2. Take a running total in Column C: SUM($B$2:B2)
3. Take a running total in Column D: C2/A2
4. Plot Column D as a line chart. Press F9 to recalculate.
   1. The line converges to the expected mean due to the law of large numbers.
5. Create a new worksheet, including the PivotTables with the records of the two categories you want to compare.

Graphical user interface, table, Excel

Description automatically generated

1. On the ribbon, go to Data -> Data Analysis -> t-Test: Two-Sample Assuming Unequal Variances

Graphical user interface, application

Description automatically generated

1. Select your variable ranges, and set the output range to somewhere on the same worksheet.
2. This gives you the p-value. Return to slides for explanation of confidence interval.
3. To calculate the confidence interval, follow with the formulas used below.

Graphical user interface, application, table

Description automatically generated

Return to slides for explanation of visualizing t-test results

# Predictive analytics & the future of Excel

File: sentiment.xlsx

1. Go to Insert > Get Add-ins
   1. Search for Azure Machine Learning
      1. Click “Add”  
         Graphical user interface, text, application

         Description automatically generated
2. Select “Text Sentiment Analysis” (Excel Add-in Sample) from the menu to the right  
   Graphical user interface, text, application, email

   Description automatically generated
3. Due to the setup of “View Schema,” the input column must be labeled tweet\_text. Change cell C1 to match that.
4. Select the Input and Output range  
   Graphical user interface, text, application

   Description automatically generated
5. Select Predict.
6. You will receive a Sentiment and a Score column. Use the Score column to derive your own sentiment categories. If the Sentiment is greater than .5, it’s positive; otherwise, it’s negative: =IF(E2 > 0.5, "Positive", "Negative")
7. Is there a relationship between the classified sentiment and the stars column? If a review was given one or two stars, it’s classified a 1 here; otherwise, 2.
   1. First thing to do is create a PivotTable counting up the predictions vs the actual polarities.
   2. Next, copy and paste those values into cells directly underneath.
   3. We are going to mathematically determine what these values would be just based on a straight proportion of each group, then use the chi-square test to compare these actual and expected values for any difference.   
      Graphical user interface, application, table

      Description automatically generated