Warning Messages

In Python, we sometimes deal with Warning or Error messages when working with data.

As you know, the latter prevent the code from running, while the former – do not. In other words, warning messages are there to make sure you know what you are doing might be different from what you intend, while error messages prevent your code from compiling.

For instance, warning messages might want to make you aware that a method you are using is being updated in a newer version of the library, or that you are trying to pass values between two objects with different sizes.

In the first case, it gives you a heads up that your code might suddenly stop working (or not work as intended) once the new version of the package (library) is implemented. Of course, you might want to adjust this if you intend to re-use your file in the future. In academia this is important when publishing because others will probably try and replicate your results using your code.

However, the warning message we have is from a different kind.

```
C:\Users\365_Careers_Designer\Anaconda3\lib\site-packages\ipykernel_launcher.py:1: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: http://pandas.pydata.org/pandas-docs/stable/indexing.html#indexing-view-versus-copy
  """Entry point for launching an IPython kernel.
```

Ignoring the Messages

This message indicates that the way we are creating our new column is problematic and we should use the “loc” method to be on the safe.

However, doing so will only complicate our code, since our current syntax does that precise job just as well, so we can stick with it.

If these error messages worry you, we can write some code to get rid of them. Simply import the library “warnings” and ignore all warnings by setting the filter value to “ignore”.

```
import warnings
warnings.filterwarnings("ignore")
```

Once you run this cell, any future warning messages will be ignored and won’t appear on screen. This will be more prominent when we add more columns to the data frame as the course progresses.

In most cases that would be when dealing with returns, integrated values or other types of data, where we have fewer elements than the original prices we are dealing with. When these happen, simply add the two lines of code in the “Pre-processing” section of your code and don’t worry about them too much.

What if I Want to See the Warnings?

Now, if you don't wish to ignore all the warning messages, you can set the argument of the method to something different form "ignore". Here is a list of alternatives you can use when using this method based on your preferences, extracted from the Python website. https://docs.python.org/3/library/warnings.html

- *action* is one of the following strings:

Value	Disposition
`"default"`	print the first occurrence of matching warnings for each location (module + line number) where the warning is issued
`"error"`	turn matching warnings into exceptions
`"ignore"`	never print matching warnings
`"always"`	always print matching warnings
`"module"`	print the first occurrence of matching warnings for each module where the warning is issued (regardless of line number)
`"once"`	print only the first occurrence of matching warnings, regardless of location

If you want to learn more about the syntax of the method, just follow the abovementioned link.