

# Executive Summary

---

The real estate analyzer is a simple program that will help a real estate investor determine the worth and feasibility of possible investments. This is a spreadsheet that will have to be done in steps, however, due to scope limitations.

This first phase of the project helps an investor create pro forma operating statements and to check up-to-date market data for current interest rates. This is the last step an investor will need to do an evaluation, to see if after valuation and financing is lined up, if the investment will actually have positive cash flows in the first few years. Once initially calculated, an investor can check the sensitive of their results by creating a new spreadsheet with slightly different inputs.

In the future, the program will also help an investor through valuation, financing and tax decisions, simple risk analysis, and the rent vs buy option.

# Write-up

---

## Purpose

To determine if a project will provide positive cash flows by creating pro forma operating statements for the first 1-10 years of the life of the investment. These pro formas will help an investor see where a project has 'speed bumps' over the life of the project as well as to help determine some variables that the investor may be able to change with some ease.

## Current Functionality

As the spreadsheet opens, the investor will have a few options to choose from. Most options are still being worked out, but currently they can choose to produce operating statements or to get current market data.

**Operating Statements:** If they select produce operating statement, they will be given an input form. This form asks for 25 different variables, 4 of which are optional. The variables cover the project valuation, equity investments, mortgage information, forecast operating and sales data, and tax information (most of this data will be easily available to an investor).

From this information, the macro will take over and create a new pro forma that is formatted. The advantage to creating this macro over asking the user to place the inputs in a spreadsheet is control. In this way, the user is unable to change any significant equations and invalidate the spreadsheet. It simply ensures the spreadsheet stays relevant.

**Market Data:** This macro will automatically go out to the internet and get current market data for interest rates. It will return 7 different rates that the investor may use to evaluate how close their rates are to the market.

The user may choose to export the data to any cells they would like to use the data inside the spreadsheets the program has created.

## Use

The program is created to be extremely user friendly and to minimize user interaction, and thus limit the user's ability to change the 'guts' of the program. The program is saved as template to force the user to create a new file each time the program is run new, this is simply to allow the original file to stay clean and uncorrupted.

The biggest worry about the program is if the user inputs items that the program can't process, such as characters or numbers that are out of scope. The program has a number of fail safes however to prevent wrong numbers from being input and from the scope of numbers getting too high to calculate properly.

For example, any entry that asks for a percentage can't be greater than 1. Some entries, such as mortgage years and forecast years are prevented from going any higher than is

common and necessary (40 years and 10 years respectively). Some values are redundant, such as future sales price and going out cap rate; in this case the program will ask you to choose which value you prefer for the analysis. It also checks to insure numbers are sound, for instance, you can't invest more in a property than it is worth, so it will not let the program complete if you entered into too many 0's for your investment.

The program also formats the entries while it is still in the form so the user can check that everything is correct (decimals in the right places and the right number of zeros, etc.). This happens whenever the mouse is moved.

Once the data is entered and process, the user has a few options to choose from. At this point they may call the main menu to get to other options in the spreadsheet. They may save the page, where they are asked for a file name and the file is saved in the same location as the original. They may test the sensitivity of the file by recalling the form and changing any values that they feel will adjust the results, a new sheet will then be created so they can compare results. (The form will not be filled with the last entries however, but from the entries of the spreadsheet it was called from.) They may delete the sheet. They may call the market data form. And last they can close the entire spreadsheet.

The market data option is very straight forward. It will only require user interaction if they choose to export the data. In that case, the user can select any cell or set of cells, and it will paste the data into those cells in a format where the user can still manipulate the data.

## Future Functionality

**Rent vs Own:** This will be a simple tool that will produce some pro forma statements to see if it is more advantageous to lease or to own a property.

**Valuation:** This will require a few inputs to help the investor determine how much a property is worth. It will allow the investor to choose from a few different valuation methods.

**Multitenant Valuation:** This will be much like valuation screen, but with a few differences to account for some of the unique elements of multitenant investments.

**Financing:** This will help an investor determine the ideal amount to invest given the size of the investment and current rates.

**Risk Analysis:** This will be a simple risk analysis tool to help an investor see where are over/underleveraging a transaction, or where some tax challenges are, etc. It is simple a quick summary of line items for the investor to consider to improve the appeal of a particular investment.

## Learning

- Initial Checks to Maintain Data Integrity—it was a challenge to get these to work correctly and to have enough to ensure the program always works correctly.

- With Statements—though I only figured this out near the end, so much of the code is still sloppy and redundant.
- Using dynamic arrays—I didn't get a lot of chance to use these much in the semester, but had to use these all over in this assignment.
- Using loops to do mass math—again, it was good practice because of the amount of these I had to use and intertwine.
- Putting data from macro variables into a spreadsheet—while these are simple, it became a challenge when doing it across 200-600 cells.
- Formatting an entire spreadsheet from a macro—this was probably the hardest part of the program because of the breadth of entries I had to format.
- Adding macro buttons to a spreadsheet—this was simple to do, but it took some time to figure out how to do it.
- Getting the right data to update to the a recalled form—Again, simple to do, a challenge to figure out.
- Preventing the user from saving over the original—I went through a bunch of options to get this to work, none of them worked well. I finally landed on 'save as' macro and saved the file as a template.
- Getting data from the internet—we had enough practice in class projects that this was pretty straight forward
- Getting data from the internet onto a spreadsheet in an editable form—The data I pulled from online was strings which included the '%' symbol. It was simple put the data onto the spreadsheet, difficult to figure out how to change the data from a string to a number that looked good right, but was easy to manipulate as a number.