# Exploring the Parameter Space

Pt. 1: Sensitivity Analysis

Nick DeRobertis<sup>1</sup>

<sup>1</sup>University of Florida Department of Finance, Insurance, and Real Estate

November 25, 2020

DeRobertis (UF) Sensitivity Analysis November 25, 2020

Table of Contents

- 1 Introduction to Parameter Exploration
- Sensitivity Analysis Theory
- 3 Sensitivity Analysis in Excel with Data Tables
- 4 Python List Comprehensions, Installing Packages, and More on Dictionaries
- 5 Sensitivity Analysis in Python

DeRobertis (UF) Sensitivity Analysis November 25, 2020 2/40

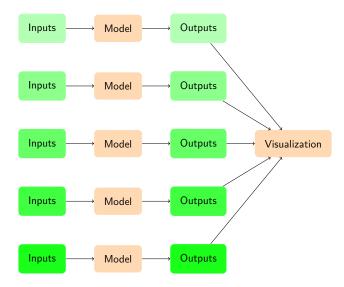
 So far, I have given you some inputs to use and you have been getting one or more outputs from those inputs

 We have not considered how those inputs may change, and how that affects the outputs

 This is where building a model vs. doing a calculation really starts to pay off

DeRobertis (UF) Sensitivity Analysis November 25, 2020 3/40

4 / 40



DeRobertis (UF) Sensitivity Analysis November 25, 2020

SA Python

# Methods of Parameter Exploration

Explore Parameters

- In this lecture, we will be discussing sensitivity analysis as an approach to exploring the parameter space.
- After we cover probabilistic modeling, we will revisit exploring the parameter space with other methods: scenario analysis and Monte Carlo Simulation.
- In sensitivity analysis, a fixed set of values for the parameters are chosen, while in Monte Carlo Simulation, each parameter is assigned a distribution.
- In scenario analysis, several realistic cases of the inputs are chosen which represent possible real-world situations
- All three methods may be used together to fully understand a model.

DeRobertis (UF) Sensitivity Analysis November 25, 2020 5/40

#### Table of Contents

- Sensitivity Analysis Theory

DeRobertis (UF) Sensitivity Analysis November 25, 2020 6 / 40 For the model given by:

Explore Parameters

$$y = f(X) \tag{1}$$

$$X = [x_1, x_2, ..., x_n]$$
 (2)

- y : Model output
- X : Model input matrix
- $x_i$ : Value of ith x variable

Follow the following steps:

- Choose a set of values for each  $x_i$
- 2 Take the cartesian product of these values as  $[X_1, X_2, ..., X_m]$
- **3** For each  $X_i$  calculate  $y_i = f(X_i)$
- Store the values of  $X_i$  mapped to  $y_i$
- $\bullet$  Visualize  $y_i$  versus  $X_i$

# Sensitivity Analysis Example Model

Let's take a simple demand model as an example:

$$D = c - EP \tag{3}$$

$$X = [c, E, P] \tag{4}$$

- D : Quantity demanded
- c : Demand constant
- E: Elasticity of demand
- P : Price
- X : Model input matrix

DeRobertis (UF) Sensitivity Analysis November 25, 2020 8 / 40

9/40

# Sensitivity Analysis Example

$$D = c - EP \tag{5}$$

Follow the following steps:

- ① Choose c = (60000, 100000), E = (200, 500), P = (50, 100)
- Take the cartesian product of these values, yielding  $[X_1, X_2, ..., X_m]$ :

С	Ε	Ρ
60000	200	50
60000	200	100
60000	500	50
60000	500	100
100000	200	50
100000	200	100
100000	500	50
100000	500	100

DeRobertis (UF) Sensitivity Analysis November 25, 2020

10 / 40

$$D = c - EP \tag{6}$$

Continue following the steps:

Explore Parameters

- **3** For each  $X_i$  calculate  $y_i = f(X_i)$
- Store the values of  $X_i$  mapped to  $y_i$

С	Ε	Ρ	D
60000	200	50	50000
60000	200	100	40000
60000	500	50	35000
60000	500	100	10000
100000	200	50	90000
100000	200	100	80000
100000	500	50	75000
100000	500	100	50000

DeRobertis (UF) Sensitivity Analysis November 25, 2020

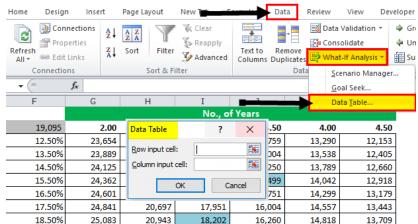
#### Table of Contents

- 3 Sensitivity Analysis in Excel with Data Tables

DeRobertis (UF) Sensitivity Analysis November 25, 2020 11 / 40

# Sensitivity Analysis in Excel

### Two-Variable Data Table in Excel



DeRobertis (UF) Sensitivity Analysis November 25, 2020 12 / 40

SA Python

# Visualizing Sensitivity Analysis in Excel

• There are two main ways to visualize sensitivity analysis results in Excel: graphing and conditional formatting.

Graphing is usually appropriate for one-way data tables

Conditional formatting is usually appropriate for two-way data tables

Sensitivity Analysis November 25, 2020 13 / 40

Explore Parameters

14 / 40

### Conditional Formatting in Excel

	А	ВС		D	E	F	G	
1	City 🚚	Jan 🔻	Feb 🔻	Mar 🔻	Apr ▼	May 🔻	Jun 🔻	
2	Barstow	80	84	84	97	95	98	
3	California City	78	86	84	96	98	102	
4	Cinco	83	86	86	97	95	103	
5	Hesperia	78	85	87	98	97	102	
6	Lancaster	78	85	86	99	95	101	
7	Mojave	82	85	86	98	96	99	
8	Palmdale	81	84	85	97	95	101	
9	Ridgecrest	81	87	87	97	96	98	
10	Rosamond	82	86	88	99	97	101	
11	Santa Clarita	79	85	87	95	96	103	

DeRobertis (UF) Sensitivity Analysis November 25, 2020

# Sensitivity Analysis in Excel

Explore Parameters

#### Adding Sensitivity Analysis to the Dynamic Retirement Excel Model

- I will now go through adding sensitivity analysis to the Dynamic Salary Retirement Model in Excel
- The completed exercise on the course site, "Dynamic Salary Retirement Model Sensitivity.xlsx"

DeRobertis (UF) Sensitivity Analysis November 25, 2020 15 / 40

#### Adding Sensitivity Analysis to Project 1 - Excel

- Add sensitivity analysis to your Excel model from Project 1
- See how the NPV changes when the number of machines and initial demand change
- Oo a one-way Data Table with a graph for each of the two inputs, then a two-way data table with conditional formatting

Resources: Slide 42

DeRobertis (UF) Sensitivity Analysis November 25, 2020 16 / 40

#### Table of Contents

- Python List Comprehensions, Installing Packages, and More on **Dictionaries**

DeRobertis (UF) Sensitivity Analysis November 25, 2020

We'll cover a couple more Python patterns and a new data type before jumping into sensitivity analysis

Dictionaries

Explore Parameters

- List comprehensions
- Python import system and custom code

DeRobertis (UF) Sensitivity Analysis November 25, 2020 18 / 40

### What is a Dictionary?

- A dictionary, or dict for short, is another basic Python data type like lists, numbers, and strings.
- Like a list, it is a collection: it holds other objects.

 Unlike a list, a dict is composed of key-value pairs. It holds relationships between objects.



#### Basic Dictionary Example

Explore Parameters

```
>>> coffee_levels_emotions = {
>>>
        'high': 'happy',
>>>
        'pretty high': 'happy',
       'medium': 'neutral',
>>>
       'low': 'sad'.
>>>
>>>
        'empty': 'desparate'
>>> }
>>> coffee_levels_emotions['pretty high']
'happy'
>>> for coffee_level, emotion in coffee_levels_emotions.items():
>>>
       print(f"I'm {emotion} when my coffee is {coffee_level}")
I'm happy when my coffee is high
I'm happy when my coffee is pretty high
I'm neutral when my coffee is medium
I'm sad when my coffee is low
I'm desparate when my coffee is empty
```

DeRobertis (UF)

21 / 40

Explore Parameters

#### Add and Delete Items from Dictionaries

```
>>> coffee_levels_emotions.update({'overflowing': 'burned'})
>>> coffee_levels_emotions['negative'] = 'confused'
>>> high_value = coffee_levels_emotions.pop('high')
>>> coffee_levels_emotions
{'pretty high': 'happy',
 'medium': 'neutral'.
 'low': 'sad'.
 'empty': 'desparate',
 'overflowing': 'burned',
 'negative': 'confused'}
>>> high_value
'happy'
```

DeRobertis (UF) Sensitivity Analysis November 25, 2020

### More About Dictionaries in Python

#### Using Dictionaries

- I will now start going through the example notebook called "Python Dicts, List comprehensions, and Imports.ipynb"
- I will go through the Dictionaries section for now

DeRobertis (UF) Sensitivity Analysis November 25, 2020 22 / 40

#### Learning How to Use Dictionaries

- For this Python section, lab exercises are in the Jupyter notebook Dicts and List Comprehensions Lab.ipynb
- Complete the exercises in the dictionaries section for now

Resources: Slide 43

DeRobertis (UF) Sensitivity Analysis November 25, 2020 23 / 40

# An Easier way to Build Simple Lists from Loops

#### The Original Way

```
>>> out values = []
>>> for i in range(5):
        out values.append(i + 10)
>>> out values
[10, 11, 12, 13, 14]
```

#### With List Comprehension

```
>>> out_values = [i + 10 for i in range(5)]
>>> out values
[10, 11, 12, 13, 14]
```

#### Notice

You never need to use list comprehension, it is just for convenience. The original for loop syntax will always work fine.

DeRobertis (UF) Sensitivity Analysis November 25, 2020 24/40

### Easier Loops in Python

#### Using List Comprehensions

- I will continue going through the example notebook "Python Dicts, List comprehensions, and Imports.ipynb"
- I will go through the List Comprehensions section for now

Sensitivity Analysis November 25, 2020 25/40

### List Comprehensions Lab

Explore Parameters

#### Learning How to Use List Comprehensions

- Continue working on the same Jupyter notebook from the previous lab exercise
- 2 Complete the exercises in the List Comprehensions section for now

Resources: Slide 44

DeRobertis (UF) Sensitivity Analysis November 25, 2020 26 / 40

### Understanding Python imports

 In the past we have used import to load packages such as numpy and pandas

Extra Python Basics

000000000000000

- These packages are just Python files. We can also write our own Python files and import them the same way
- When you import something, Python first searches the current directory for a file something.py and if it doesn't find it, it searches your installed packages
- In fact if you added a numpy.py in the current directory and tried to import numpy it would import the contents of that file rather than the numpy package.

DeRobertis (UF) Sensitivity Analysis November 25, 2020 27 / 40

# Importing Custom Code

- You can write your own functions and classes, then put them in a Python file and import them into your notebook.
- When you import a file, it executes the contents of that file. So you
  generally want just function and class definitions, and not really
  anything outside of def or class statements.
- Using Python files is a more maintainable structure for building complex models and apps versus Jupyter notebooks only.

DeRobertis (UF) Sensitivity Analysis November 25, 2020 28 / 40

# Installing Packages

Explore Parameters

- Sometimes you will need a package which does not already come installed with Anaconda
- The general way to do this is with pip install mypackage replacing mypackage with the package you want to install
- You would run this in Anaconda Prompt, or in Jupyter you can run it but you need to put an exclaimation mark before it to say you want to run it in a terminal. So in Jupyter it would be !pip install mypackage

DeRobertis (UF) Sensitivity Analysis November 25, 2020 29 / 40

# Installing Packages in Python

#### How to Install Packages

Explore Parameters

- I will continue going through the example notebook "Python Dicts, List comprehensions, and Imports.ipynb"
- I will go through the Imports and Installing Packages section for now

DeRobertis (UF) Sensitivity Analysis November 25, 2020 30 / 40

SA Python

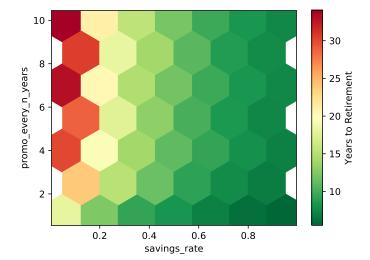
31 / 40

#### Table of Contents

- Sensitivity Analysis in Python

DeRobertis (UF) Sensitivity Analysis November 25, 2020

# Sensitivity Analysis in Python - Hex-Bin



DeRobertis (UF) Sensitivity Analysis November 25, 2020 32 / 40

SA Python

### Sensitivity Analysis in Python - Styled DataFrame

Result - value1 vs. value2

10 11 12 13 14 15 16 17 18 19

#### value1

values										
0	15	16	17	18	19	20	21	22	23	24
1	16	17	18	19	20	21	22	23	24	25
2	17	18	19	20	21	22	23	24	25	26
3	18	19	20	21	22	23	24	25	26	27
4	19	20	21	22	23	24	25	26	27	28
5	20	21	22	23	24	25	26	27	28	29
6	21	22	23	24	25	26	27	28	29	30
7	22	23	24	25	26	27	28	29	30	31
8	23	24	25	26	27	28	29	30	31	32
9	24	25	26	27	28	29	30	31	32	33

DeRobertis (UF) Sensitivity Analysis November 25, 2020 33 / 40

# How to Do Sensitivity Analysis in Python (The Hard Way)

- Generally, to do sensitivity analysis in Python without any special tools, you would just create one nested for loop for each input, and finally within all the loops, run your model with the inputs from the loops
- This will work fine, but you will have many nested loops which can become hard to read. Also it is a fair bit of setup involved.
- You can avoid the nested loops with itertools.product but then this becomes more difficult to use and read

DeRobertis (UF) Sensitivity Analysis November 25, 2020 34 / 40

35 / 40

# Sensitivity Analysis Example (Hard Way)

 Say you have a function which runs your model, called model, which takes inputs of inp1 and inp2

### Sensitivity Analysis in Python with No Libraries

```
inp1_values = [1, 2]
inp2\_values = [4, 5]
results = []
for inp1 in inp1_values:
    for inp2 in inp2_values:
        result = model(inp1, inp2,)
        results.append(
            (inp1, inp2, result)
        )
pd.DataFrame(results, columns=['inp1', 'inp2', 'Result'])
   inp1
       inp2 Result
         5
```

DeRobertis (UF) Sensitivity Analysis November 25, 2020

# How to Do Sensitivity Analysis in Python (The Easy Way)

Extra Python Basics

 When I first created this course, I thought there should be a good sensitivity analysis tool in Python and I couldn't find it

- The beauty of Python is if you want a tool that doesn't exist, you can create it, and share it with others so that nobody else has to deal with the problem.
- So I created sensitivity a package for sensitivity analysis in Python, which makes it very easy

DeRobertis (UF) Sensitivity Analysis November 25, 2020 36 / 40

# Sensitivity Analysis Example (Easy Way)

 Say you have a function which runs your model, called model, which takes inputs of inp1 and inp2

### Sensitivity Analysis in Python with sensitivity

```
from sensitivity import SensitivityAnalyzer
sensitivity values = {
    'inp1': [1, 2],
    'inp2': [4, 5],
}
     SensitivityAnalyzer(sensitivity_values, model)
sa.df
   inp1
       inp2 Result
               6
```

DeRobertis (UF)

6

7

4

5

### Intro to Sensitivity Analysis in Python

#### An Overview of the Manual and Automated Approaches

- I will now go through Sensitivity Analysis example Jupyter notebook
- This notebook shows both the standard approach and using the sensitivity package

DeRobertis (UF) Sensitivity Analysis November 25, 2020 38 / 40

SA Python

# Applying Sensitivity Analysis in Python

Explore Parameters

#### Adding Sensitivity Analysis to the Dynamic Retirement Python Model

- I will now go through adding sensitivity analysis to the Dynamic Salary Retirement Model in Python
- The completed exercise available on the course site is called "Dynamic Salary Retirement Model Sensitivity.ipynb"

DeRobertis (UF) Sensitivity Analysis November 25, 2020 39 / 40

#### Adding Sensitivity Analysis to Project 1 - Python

- Add sensitivity analysis to your Python model from Project 1
- See how the NPV changes when the number of machines and initial demand change
- Output both a hex-bin plot and a styled DataFrame

Resources: Slide 45

DeRobertis (UF) Sensitivity Analysis November 25, 2020 40 / 40

41 / 40

#### Lecture Resources

#### Lecture Resources

- Slides Exploring the Parameter Space
- 2 Lecture Notes Exploring the Parameter Space
- Opnio Salary Retirement Model Excel
- Python Dicts, List Comprehensions, and Imports
- Dictionaries, List Comprehensions, and Imports Labs
- Guide to Python Imports
- Sensitivity Analysis
- Oynamic Salary Retirement Model Python

42 / 40

### Sensitivity Analysis in Excel Lab Resources

#### Adding Sensitivity Analysis to Project 1 - Excel Resources

Slides - Exploring the Parameter Space

Exercise: Slide 16

DeRobertis (UF) Sensitivity Analysis November 25, 2020

#### Dictionaries Lab Resources

#### Learning How to Use Dictionaries Resources

- Slides Exploring the Parameter Space
- Python Dicts, List Comprehensions, and Imports
- Oictionaries, List Comprehensions, and Imports Labs

Exercise: Slide 23

### List Comprehensions Lab Resources

#### Learning How to Use List Comprehensions Resources

- Slides Exploring the Parameter Space
- Python Dicts, List Comprehensions, and Imports
- Oictionaries, List Comprehensions, and Imports Labs

Exercise: Slide 26

### Sensitivity Analysis in Python Lab Resources

#### Adding Sensitivity Analysis to Project 1 - Python Resources

Slides - Exploring the Parameter Space

Exercise: Slide 40