

# Grid 2050 – User Manual

Sarah Troise<sup>a</sup>, M. Granger Morgan<sup>a</sup>

<sup>a</sup>Carnegie Mellon University Department of Engineering and Public Policy

## Abstract

This paper introduces Grid2050, a web-based decision support framework that allows users to build and assess US low-carbon electricity generation portfolios for 2050. Although base assumptions from various sources are integrated in the tool to support a default portfolio mix, the framework encourages users to alter assumptions about the techno-economic performance and socio-political acceptance of different technologies to create their own portfolios. To support that process, the framework includes built-in analytics to estimate deviations from base assumptions, a capability to explore alternative assumptions about technological learning and scale-up, and the premiums that users are willing to incur to avoid or deploy different technologies. This suite of features offers a way to benchmark the consistency and feasibility of users' energy system preferences. Its use should help inform energy system planning studies and regulatory interventions to bend the global warming curve.

## Table of Contents

<i>Overview .....</i>	<i>3</i>
<i>System Requirements and Licenses.....</i>	<i>4</i>
<i>Download and installation.....</i>	<i>5</i>
<i>Adding a new technology.....</i>	<i>6</i>
<i>How to run Grid2050 – web-based.....</i>	<i>7</i>
<i>How to run Grid2050 – customized.....</i>	<i>8</i>

## 1. Overview

## 2. System Requirements and Licenses

Grid2050 can be found for web use at this [link](#). To edit and customize Grid2050 to fit a specific project, the following software tools are required:

1. **R.** The Grid2050 tool is coded in R, so in order to work with the code R is required.

Grid2050 was built using R version 4.3.0, but is programmed in a way that it should be compatible with older and newer versions.

2. **Microsoft Excel.** The main input file for Grid2050 is stored as a Microsoft Excel file (.xlsx).

3. **HTML.** Several input files for the Grid2050 tool are written in HTML.

### **3. Download and installation**

1. Follow this link ()
2. Download the ZIP file by clicking on the green 'Code' button and selecting 'Download ZIP'
3. Unzip and keep all files in the same folder
4. RStudio is recommended to work with the files

## 4. Adding a new technology

1. Add Technology info to Excel worksheet “Tool Data.xlsx”

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
1	Technology	LCOE_2020	LCOE_2050_Mean_User	LCOE_2050_Min_User	LCOE_2050_Max_User	Generation_2020	Generation_2050_Mean_User	Generation_2050_Min_User	Generation_2050_Max_User	Generation_adoption	retire	retire2	learn	maxGrowth	
2	Solar	36.0	36.0	30.0	49.0	145.0	145.0	145.0	145.0	0.00	0.0	0.0	0.0000	32%	
3	OnshoreWind	38.0	38.0	30.0	66.0	435.0	435.0	435.0	435.0	0.00	0.0	0.0	0.0000	33%	
4	Hydropower	64.0	64.0	49.0	83.0	261.0	261.0	261.0	261.0	0.00	0.0	0.0	0.0000	0%	
5	Nuclear	88.0	88.0	83.0	99.0	771.0	771.0	771.0	771.0	0.00	0.0	0.0	0.0000	4%	
6	OffshoreWind	136.0	136.0	110.0	170.0	0.2	0.2	0.2	0.2	0.00	0.0	0.0	0.0000	NA	
7	CCS-NG-Retrofit	115.0	115.0	79.0	153.0	1.1	1.1	1.1	1.1	0.0	0.0	0.0	0.0	NA	
8	CCS-NG-New	115.0	115.0	79.0	153.0	1.1	1.1	1.1	1.1	0.0	0.0	0.0	0.0	NA	
9	CCS-Coal-Retrofit	115.0	115.0	79.0	153.0	1.1	1.1	1.1	1.1	0.0	0.0	0.0	0.0	NA	
10															

2. Update the “numTechnology” value

```
#The below packages are needed to run the app
library(shiny)
library(shinydashboard)
library(readxl)
library(ggplot2)

# Data is imported from an external excel spreadsheet
data <- read_excel("ToolData.xlsx", sheet = "toolData")
data <- as.data.frame(data)
row.names(data) <- data$Technology
numTechnology <- 8
```

3. Add Assumptions for new technology

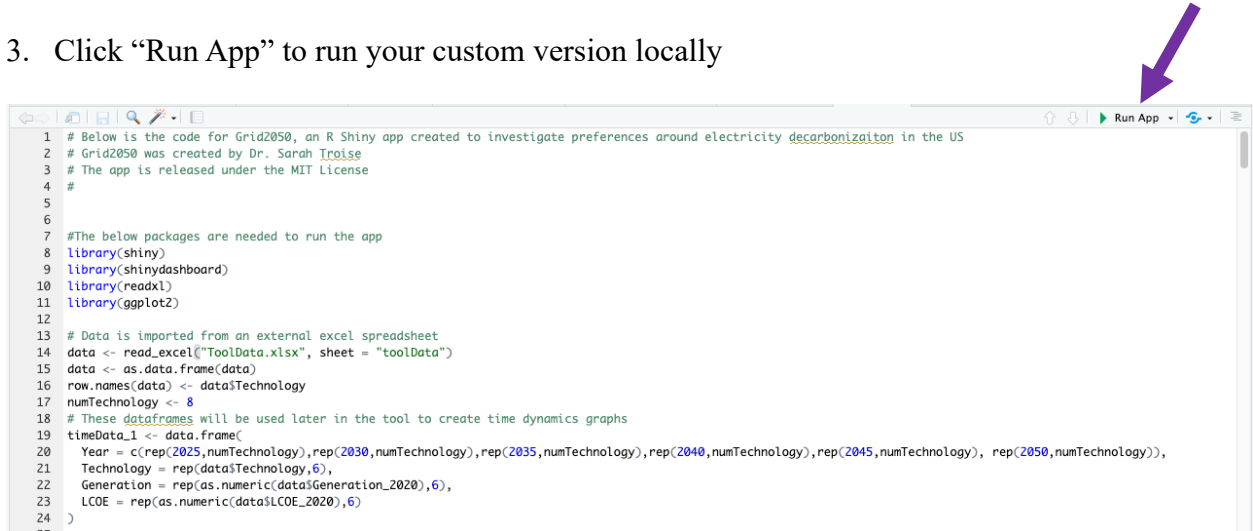
## 5. How to run Grid2050 – web-based

1. Follow the link (<https://troises19.shinyapps.io/Grid2050/>) and use Grid2050 as a web-based tool

## 6. How to run Grid2050 – customized

This provides a step by step guide on to run Grid2050 with custom features.

1. Download and unzip the Grid2050 code as described in Section 3.
2. Open the “app.R” file in R Studio to customize the tool
  - a. See Section 4 on how to add new technologies to the tool
  - b. For surveys, consider adding a demographics page
  - c. Consider changing values in “ToolData.xlsx” to values from other countries for non-US-based markets
  - d. Consider changing values in “ToolData.xlsx” to state or region-specific values for a regional tool
3. Click “Run App” to run your custom version locally



```
1 # Below is the code for Grid2050, an R Shiny app created to investigate preferences around electricity decarbonization in the US
2 # Grid2050 was created by Dr. Sarah Troise
3 # The app is released under the MIT License
4 #
5
6
7 #The below packages are needed to run the app
8 library(shiny)
9 library(shinydashboard)
10 library(readxl)
11 library(ggplot2)
12
13 # Data is imported from an external excel spreadsheet
14 data <- read_excel("ToolData.xlsx", sheet = "toolData")
15 data <- as.data.frame(data)
16 row.names(data) <- data$Technology
17 numTechnology <- 8
18 # These dataframes will be used later in the tool to create time dynamics graphs
19 timeData_1 <- data.frame(
20   Year = c(rep(2025, numTechnology), rep(2030, numTechnology), rep(2035, numTechnology), rep(2040, numTechnology), rep(2045, numTechnology), rep(2050, numTechnology)),
21   Technology = rep(data$Technology, 6),
22   Generation = rep(as.numeric(data$Generation_2020), 6),
23   LCOE = rep(as.numeric(data$LCOE_2020), 6)
24 )
25 }
```



4. If you want to publish your version on the web, click the blue icon in the top right corner.



```
1 # Below is the code for Grid2050, an R Shiny app created to investigate preferences around electricity decarbonization in the US
2 # Grid2050 was created by Dr. Sarah Troise
3 # The app is released under the MIT License
4 #
5
6
7 #The below packages are needed to run the app
8 library(shiny)
9 library(shinydashboard)
10 library(readxl)
11 library(ggplot2)
12
13 # Data is imported from an external excel spreadsheet
14 data <- read_excel("ToolData.xlsx", sheet = "toolData")
15 data <- as.data.frame(data)
16 row.names(data) <- data$Technology
17 numTechnology <- 8
18 # These dataframes will be used later in the tool to create time dynamics graphs
19 timeData_1 <- data.frame(
20   Year = c(rep(2025,numTechnology),rep(2030,numTechnology),rep(2035,numTechnology),rep(2040,numTechnology),rep(2045,numTechnology), rep(2050,numTechnology)),
21   Technology = rep(data$Technology,6),
22   Generation = rep(as.numeric(data$Generation_2020),6),
23   LCOE = rep(as.numeric(data$LCOE_2020),6)
24 )
25
```