

RHEA Analysis and Visualization

This document contains examples on analysis and visualisations on output from the RHEA agent-based-model for the “Digging into Data” project. Each example is presented with accompanying R code as well as further documentation on the functions used to create the visualization or analysis. The examples require data provided in the accompanying `csv/` directory. Further details on the data sets and the data items/variables are provided at the end of the document.

Description of the RHEA Agent-Based Model

RHEA is the acronym for **R**isks and **H**edonics in an **E**mpirical **A**gent-Based Land-Market. The model simulates a bilateral, heterogeneous housing market in an empirical landscape. Buyers search for properties in a seaside town where both coastal amenities and environmental risks of flooding are spatially correlated. A realtor agent serves as a mediating agent that learns about the current market trend and willingness to pay for various spatial attributes of properties (Filatova, 2014). At each time step, sellers of the properties advertise their ask prices while buyers select and offer bids on properties. Then, the two groups engage in dyadic (pair-wise) negotiation which may or may not result in a successful transaction (or trade).

The ask prices are primarily market-driven: sellers (1) consult with a realtor on the appropriate price given the current trends as estimated by a hedonic analysis of the recent simulated transactions and (2) adjust the ask price if a property remains too long on a market. Buyers bid on the single property that brings them maximum utility and is affordable for their budgets. A buyer’s budget constraints include their income, preference for amenities, a necessity to pay insurance premiums for flood-prone properties, and the possibility to activate an annual income growth. Spatial patterns of price dynamics and intensity of trade are the emergent outputs of the model. When studying them under various parameter settings, we noticed that relative market power of buyers and sellers plays a role. The explanation is twofold. Firstly, a parcel that is very attractive will most likely receive several bid offers, out of which its seller chooses the highest, driving prices for most desirable properties up. Secondly, sellers of the properties that are less desirable may receive only one or even no bid offers, which can result in their accepting a bid that is below their ask prices or reducing their ask prices after a number of unsuccessful trades. Thus, excess demand drives prices up while excess supply pushed them down.

A Note About the Data Sets

The RHEA output data sets employed in the following examples are encoded as flat `data.frames` in R. The columns need to include (in no particular order) the run seed (or incrementing id number is sufficient), time step or `tick`, experimental variables/conditions, and outcome variables. For example, the following table shows the first three and several other rows of the data employed in the first example (i.e., the data frame `x.bs.200`):

Ins	RH	s	t	avPriceFP0	avPriceFP100	avPriceFP500	avPriceCoastFront	...
0	0	0	0	239169.97	266217.58	193210.70	461214.67	...
0	0	0	1	239578.40	266698.20	193525.24	462776.09	...
0	0	0	2	240034.15	267186.23	193877.64	463939.95	...
...
1	0	2	199	525784.50	516614.00	470629.55	905632.60	...
1	0	2	200	528258.68	518983.84	472518.26	910007.72	...
1	0	3	0	239169.97	266217.58	193210.70	461214.67	...
1	0	3	1	239559.08	266593.64	193442.31	462448.64	...
...

In this data, the 1st two columns contain the experimental values (in **RH** and **Ins**) and the next two columns contain the random number seed and the time step (in **s** and **t** respectively). The subsequent columns contain the model outcomes.

User-defined data sets to be used with the visualization and analysis routines need to follow this tabular format.

Details of all of the data sets and data items/variables are provided at the end of the document.

A Note About the Functions

All of the functions in the examples require prior loading of the primary script file (**ut_did.R**) in the R session using the following command:

```
source("ut_did.R")
```

One may consider **ut_did.R** to be a library of functions which contain the plotting functions of the examples below as well as auxiliary functions.

Also, all functions have an **outfile** parameter. If **outfile** is set, then the plotting function will write its output to a PDF file specified by **outfile**. Users may also set **width** and **height** (in inches), the defaults of which are both 7 if not already specified by the function, to adjust the *x*- and *y*- dimensions of the PDF plot as they see fit. It is advisable to include the **.pdf** file extension in the **outfile** filename, e.g., **outfile="figure_1.pdf"**.

The function producing regression models will write its output in text (not PDF). It is advisable to include the **.txt** file extension in the **outfile** filename for regression model outputs, e.g., **outfile="table_1.txt"**.

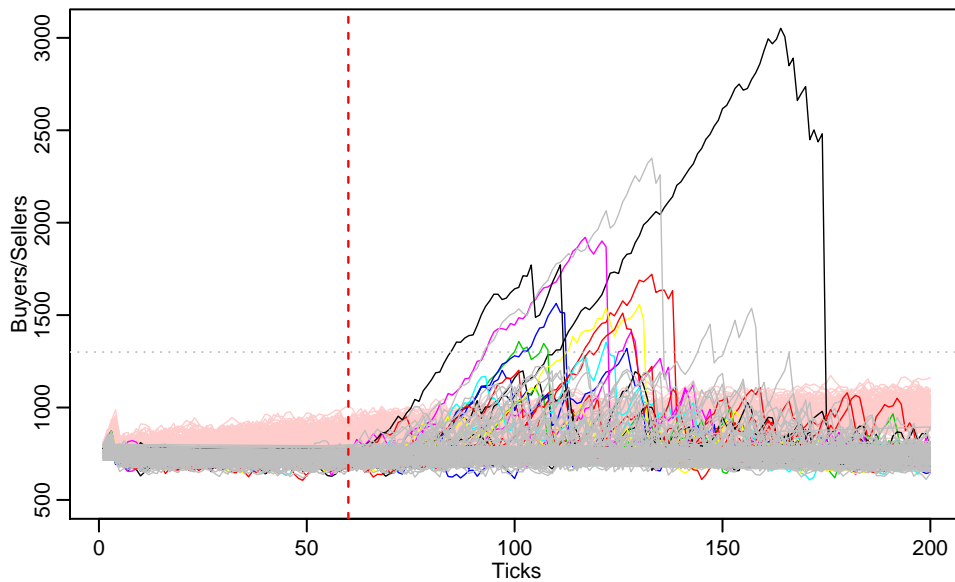
Example 1 - Buyers and Sellers - Saturation and Catastrophe

Example 1a - Buyer Spike Across Seeds

In the RHEA model, we can track the number of buyers and sellers at each time period (or tick) to reveal those seeds/runs that deviate from the nominal behavior. The goal of this diagnostic visual analysis is to ascertain the characteristics of those simulation instances in which the population counts of either agent type pool deviates from a nominal distribution.

One needs first load in the dataset employed and run the plotting function for charting two variables across 25 seeds/runs over time horizon of 200 ticks, highlighting those trajectories that surpass a defined threshold. The following lines should be run in the R session to produce the figure below:

```
if (!exists("x.bs.200")) x.bs.200 = read.csv("csv/x_bs_200.csv")
fig.nB.did(data=x.bs.200)
```



The light pink trails indicate the seller count per tick, which appears to increase at a slight rate. The gray trails indicate most of the buyer counts, which initially decrease and reach a steady state. The multi-colored trails that climb indicate distinct runs/seeds which incur buyer counts exceeding the nominal threshold of 1300. The graph reveals that for some seeds, the buyer counts climb high after which it quickly collapses to a nominal level. This observation tells us that the curvilinear mean Buyer/Seller difference (for $\text{coef} = 0.70$) suffers from these “mountain” outlier influences. The seller counts on the otherhand appear wholly stable and monotonically increasing and does not contribute to the curvilinear mean. Furthermore, we observe a small spike into both buyer and seller counts near the start of the simulation.

The graph reveals that for some seeds, the buyer counts climb high after which it quickly collapses to a nominal level. This observation tells us that the curvilinear mean Buyer/Seller difference (for $\text{coef} = 0.70$) suffers from these “mountain” outlier influences. The seller counts on the otherhand appear wholly stable and monotonically increasing and does not contribute to the curvilinear mean. Furthermore, we observe a small spike into both buyer and seller counts near the start of the simulation.

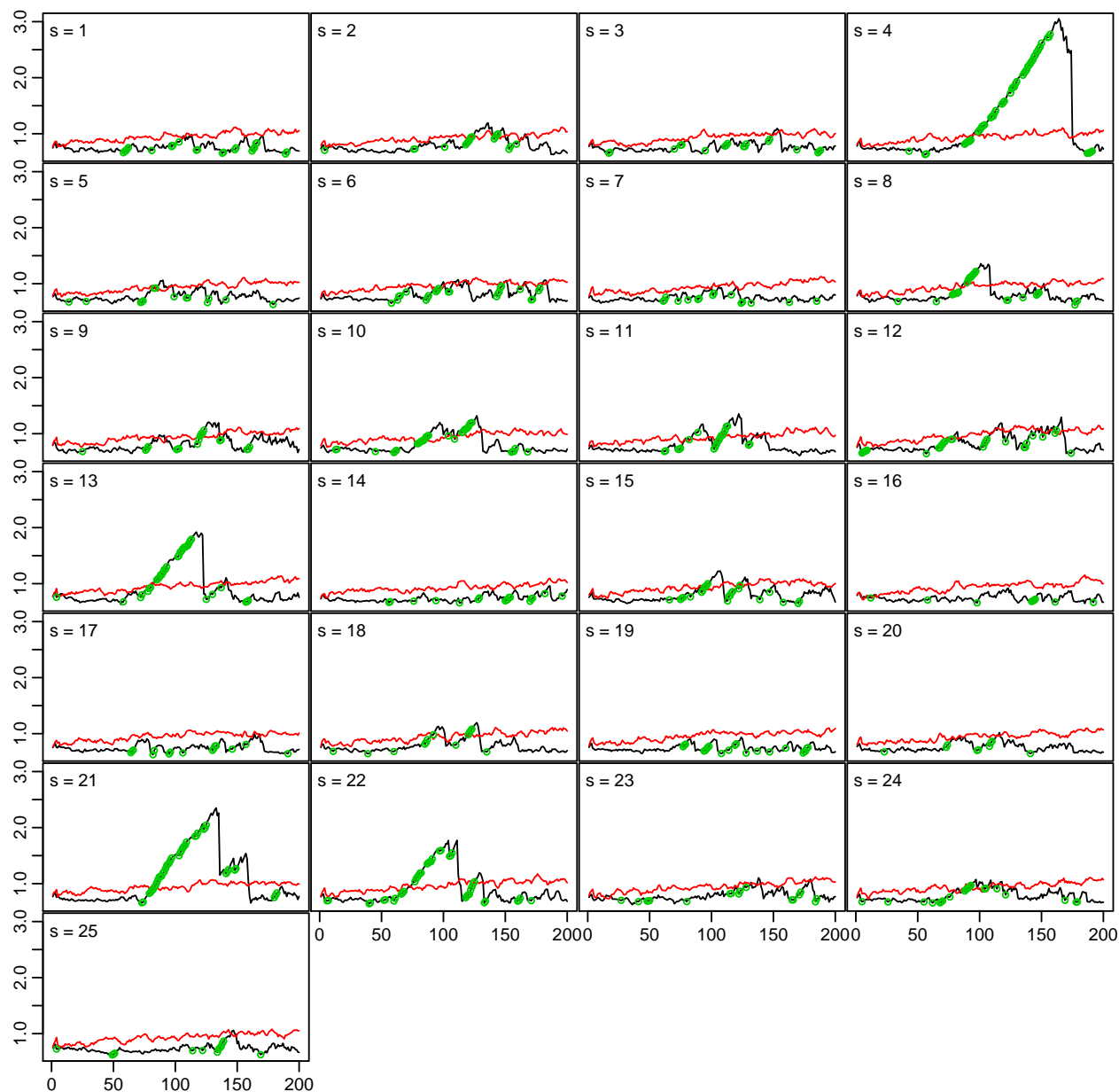
Description of `fig.nB.did` The `fig.nB.did` plotting function has the following arguments which may be employed to generalize its functionalities. As a general function, it can plot trajectories of two outcomes for individual runs and also graphically highlight those values which exceed a threshold.

Argument	Default	Description
<code>data</code>	<code>x.bs.200</code>	Dataset containing variables in <code>yvar</code>
<code>yvar</code>	<code>c("nB","nS")</code>	Two outcome variables of data to plot
<code>S</code>	<code>0:24</code>	Values/range of the seeds to consider
<code>svar</code>	<code>"s"</code>	Variable with the seeds
<code>tvar</code>	<code>"t"</code>	Variable with time step values
<code>yM</code>	<code>30</code>	Vertical dashed line denoting mortgage years (assumes each tick is semi-annual)
<code>crit</code>	<code>1300</code>	Critical value of <code>yvar[1]</code> , above which trajectories are highlighted by different colors
<code>sd.mult</code>	<code>0</code>	If <code>sd.mult = 0</code> , then use <code>crit</code> directly. If <code>sd.mult > 0</code> , then <code>crit = $\mu + \sigma \cdot \text{sd.mult}$</code> (of <code>yvar[1]</code>). If <code>sd.mult = -1</code> , then <code>crit = max(yvar[2])</code>
<code>xlab</code>	<code>"Ticks"</code>	Label of x -axis
<code>ylab</code>	<code>"Buyers/Sellers"</code>	Label of y -axis
<code>y0</code>	<code>500</code>	Minimum of y -axis values displayed

Example 1b - Buyer Spikes for Individual Seeds

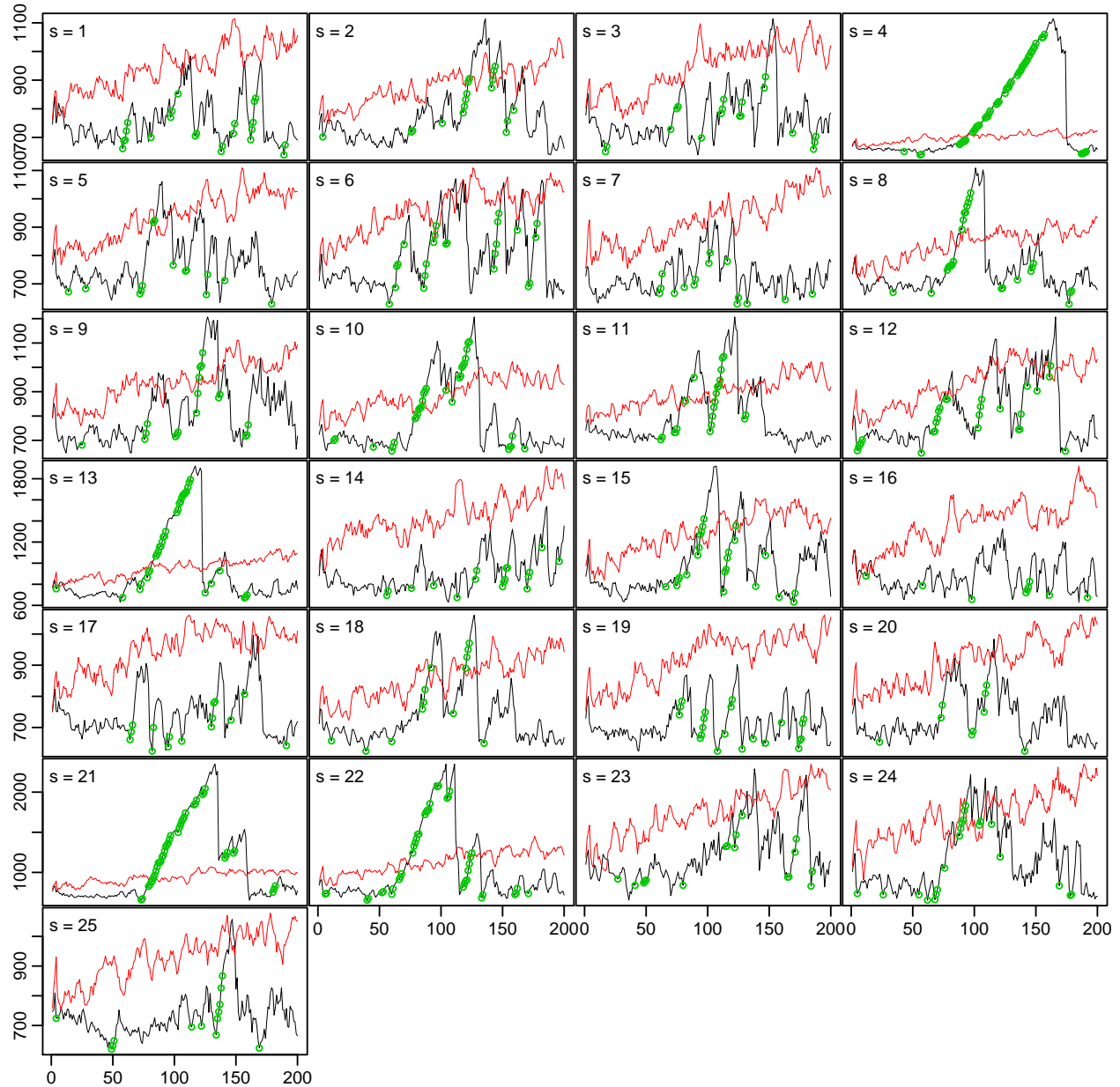
In order to understand these “mountains”, we display the Buyer (black) and Seller (red) counts for each run/seed in separate plots. These graphs have a common y -axis range to allow for comparison. The green dots highlight sequences in which the buyer counts increase for at least 5 consecutive ticks. The following lines should be run in the R session to produce the figure below:

```
if (!exists("x.bs.200")) x.bs.200 = read.csv("csv/x_bs_200.csv")
fig.nBS.did(cex=.75,abs=T)
```



We observe that the larger spikes consist of multiple rising (green) sequences. The next figure displays the same data but using the y -axis range of each seed's data. Note that the y -axis values which appear on the leftmost graphs pertain only to their immediate graphs. The following lines should be run in the R session to produce the figure below:

```
if (!exists("x.bs.200")) x.bs.200 = read.csv("csv/x_bs_200.csv")
fig.nBS.did(cex=.75,abs=F,lwd=.5)
```



We observe that the “mountains” entail multiple sequences of increasing buyer counts suggesting that such sequences are required to sustain the growth of these counts. We also observe less concentrated rising sequences and almost all runs contain these.

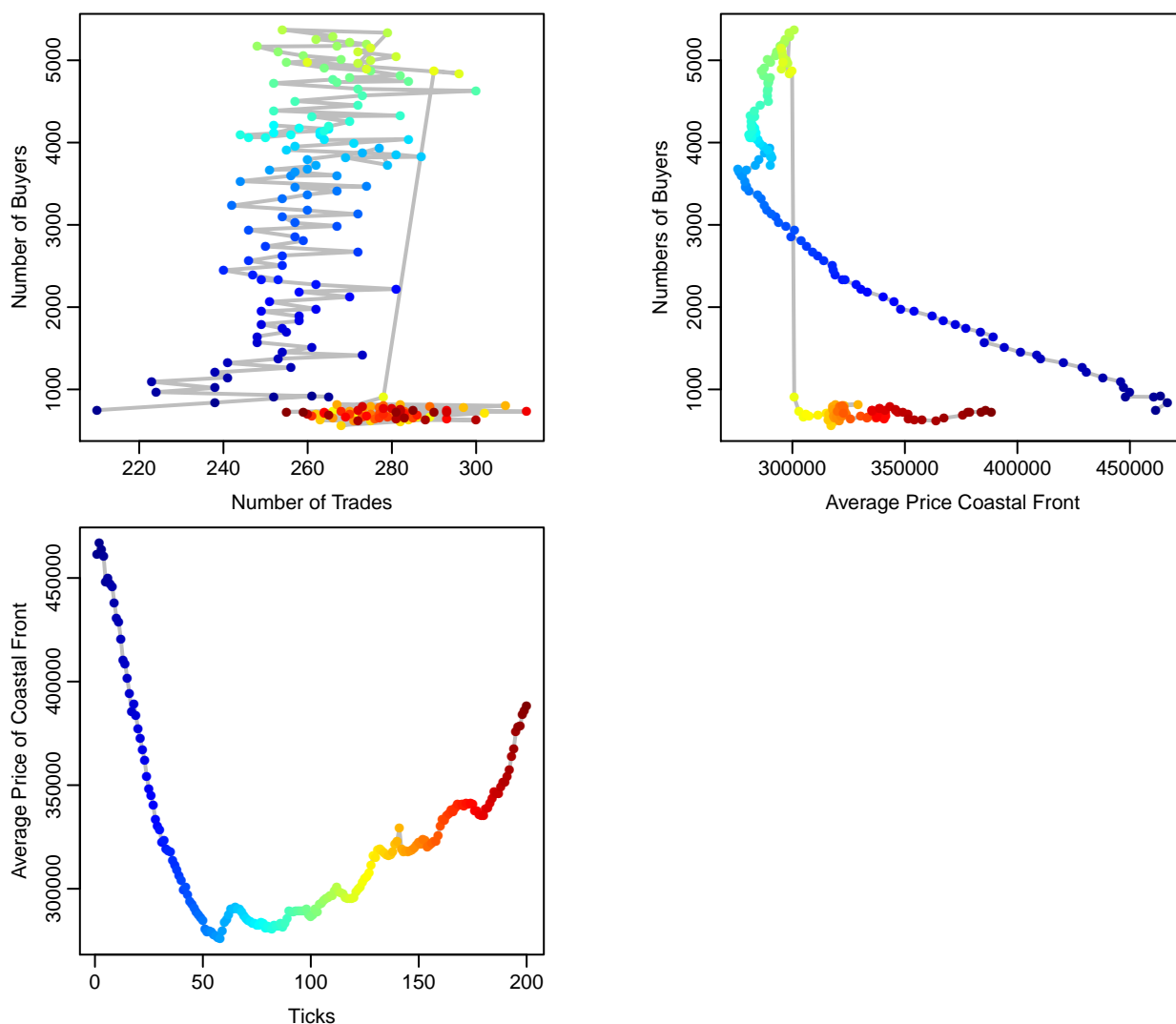
Description of `fig.nBS.did` The `fig.nBS.did` plotting function has the following arguments which may be employed to generalize its functionalities. The function produces a set of plots containing two variables over time with each plot representing results from a distinct seed.

Argument	Default	Description
<code>data</code>	<code>x.bs.200</code>	Dataset containing variables in <code>yvar</code>
<code>S</code>	<code>0:24</code>	Value(s)/range of seeds to display
<code>yvar</code>	<code>c("nB", "nS")</code>	Two variables in <code>data</code> to plot
<code>svar</code>	<code>"s"</code>	Variable with the seeds
<code>tvar</code>	<code>"t"</code>	Variable with time step values
<code>scale</code>	<code>1000</code>	Dividing factor for <code>yvar</code> (for purposes of displaying shorter numerical labels)
<code>abs</code>	<code>TRUE</code>	Is the <i>y</i> -axis scale fixed and absolute across all plots?

Example 1c - Trajectory Plots

For one of the seeds that produce a saturation/crash pattern, we plot the trajectory of several outcomes across a 2D space over a time horizon of 200 ticks. The following lines should be run in the R session to produce the figures below:

```
if (!exists("x.yM.10.200")) x.yM.10.200 = read.csv("csv/x_yM_10_200.csv")
fig.yM.traj.did(data = x.yM.10.200, x = "nT", y = "nB", s = 1, fx = I,
  fy = I, xlab = "Number of Trades", ylab = "Number of Buyers")
fig.yM.traj.did(data = x.yM.10.200, x = "avPriceCoastalFront", y = "nB",
  s = 1, fx = I, fy = I, xlab = "Average Price Coastal Front", ylab = "Numbers of Buyers")
fig.yM.traj.did(data = x.yM.10.200, x = "t", y = "avPriceCoastalFront", s = 1,
  fx = I, fy = I, xlab = "Ticks", ylab = "Average Price of Coastal Front")
```



The first one charts the number of buyers and number of trades; the second one, the number buyers against the average coastal property price; and the third, the average coastal property price against time (ticks). In each plot, the colors denote time: dark blue is the beginning of the simulation while dark red represents the end of the simulation.

The crash on the market occurs because many buyers leave without finding an affordable house. Over time,

prices, which sellers gradually adjust if they are not able to sell, drop to the level when buyers may at least attempt to purchase a house by bidding on it. Yet, at this moment the market is already so saturated with buyers, that this excess demand creates a tremendous competition among them. As a result, many buyers leave since they face several unsuccessful trade attempts. At the same time, prices start to grow since sellers select the highest bid out of many bids that are now being submitted.

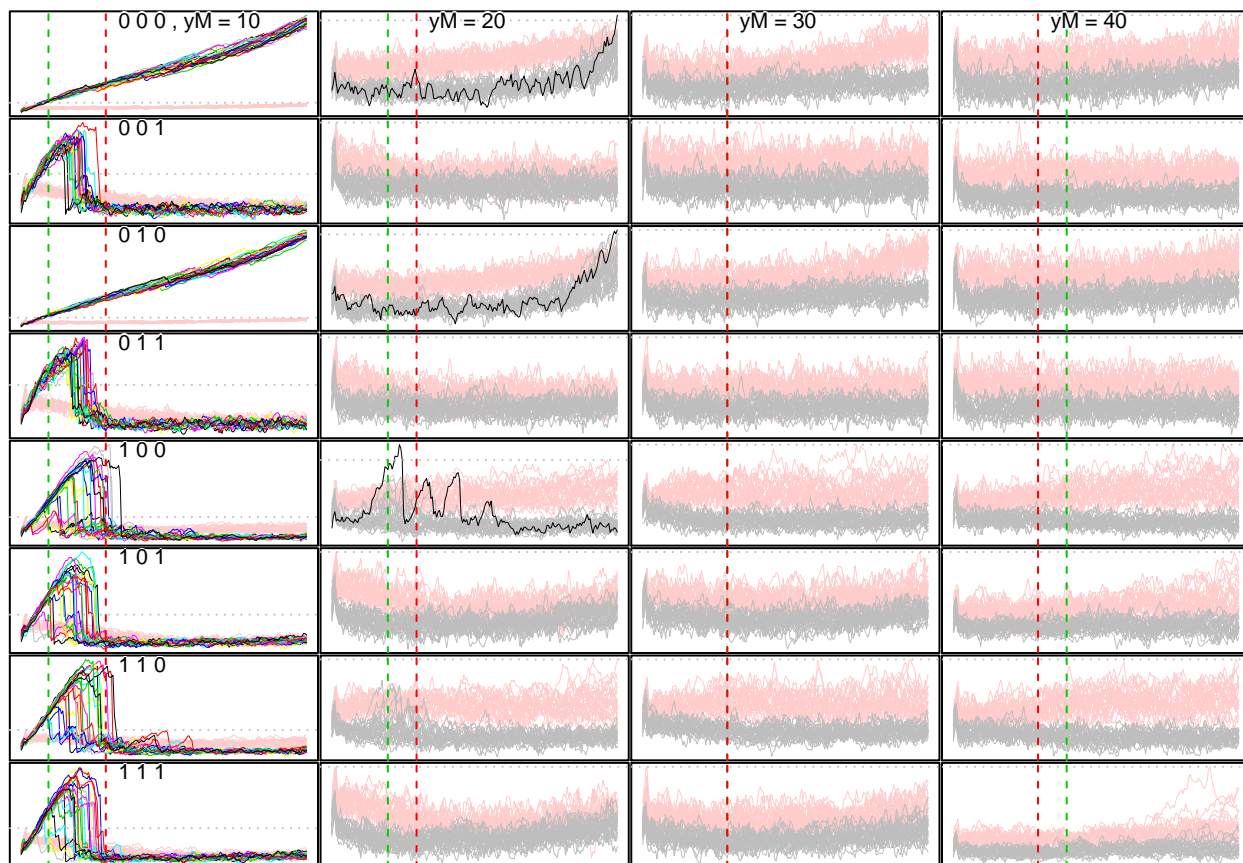
Description of `fig.yM.traj.did` The `fig.yM.traj.did` plotting function has the following arguments which may be employed to generalize its functionalities. As a generic function, it plots the trajectory of model behavior in a 2D space defined two of the outcomes.

Argument	Default	Description
<code>data</code>	<code>x.yM.10.200</code>	Dataset containing variables <code>x</code> and <code>y</code>
<code>x</code>	<code>"nT"</code>	Variable for x -axis (default = number of trades)
<code>y</code>	<code>"nB"</code>	Variable for y -axis (default = number of buyers)
<code>s</code>	<code>1</code>	Single seed value from <code>data</code> to plot (assuming <code>data</code> contains outcomes from multiple seeds)
<code>fx</code>	<code>I</code>	Transformation function for <code>x</code> variable (default = indicator function)
<code>fy</code>	<code>I</code>	Transformation function for <code>y</code> variable (default = indicator function)
<code>xlab</code>	<code>"Number of Trades"</code>	Label for x -axis
<code>ylab</code>	<code>"Number of Buyers"</code>	Label for y -axis

Example 1d - Effects of Mortgage on Buyer Spikes

In next figure, we display the buyer counts under a new depreciation mechanism in which is applied cumulatively and an increase in the proportion of properties on sale (fraction on sale or `fos= 0.20`). The following lines should be run in the R session to produce the figure below:

```
if (!exists("x.yM2.200")) x.yM2.200 = read.csv("csv/x_yM2_200.csv")
fig.yM2.did(data=x.yM2.200,lwd=.25)
```



The columns in the figure indicate each of the mortgage years, $yM \in \{10, 20, 30, 40\}$ and the rows indicate each combination of the three conditions: realtor hedonic (0=static), insurance, and income growth. For example, the first row is (0,0,0) (static, no insurance, and no income growth). The second row is (0,0,1) (same but with income growth). The final row is (1,1,1) which equates to engagement of all three adaptive, insurance, and income growth. For presentation purposes, the axes labels have been removed.

The alternative depreciation mechanism appears to mitigate the buyer spikes especially for the longer mortgage periods. However, we see that the spikes are even more prevalent when the mortgage period is curtailed ($yM = 10$), which reduces the purchasing power of the buyers. In the absence of adaptive price assessment (static hedonic) and income growth (rows 000 and 010), the proportionally fewer and fewer buyers are able to afford properties so the influx of new buyers creates a monotonically increasing trend in the buyer counts.

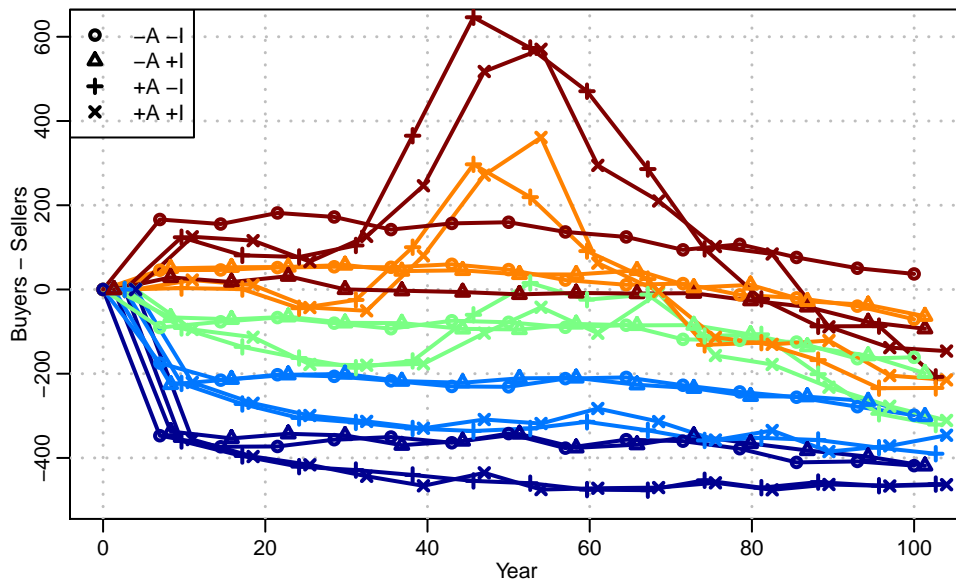
Description of `fig.yM2.did` The `fig.yM2.did` plotting function has the following arguments which may be employed to generalize its functionalities. The function implicitly seeks all combinations of the 1st three variables (enumerated in the plotted rows) against a fourth variables (enumerated in the plotted columns). These four variables are specified in `zvar`. In each plot, the outcomes of `yvar` are shown for all seeds in `S`.

Argument	Default	Description
<code>data</code>	<code>x.yM2.200</code>	Dataset containing variables of <code>yvar</code> and <code>zvar</code>
<code>S</code>	<code>0:24</code>	Range or vector of seeds to plot
<code>mfrow</code>	<code>c(8,4)</code>	Dimensional arrangement of subplots
<code>yvar</code>	<code>c("nB","nS")</code>	Two variables of <code>data</code> to plot
<code>svar</code>	<code>"s"</code>	Variable with seeds
<code>zvar</code>	<code>c("RH","Ins","G","yM")</code>	Vector specifying the experimental conditions/variables
<code>RH</code>	<code>0:1</code>	1st condition values/range (default: realtor hedonics)
<code>Ins</code>	<code>0:1</code>	2nd condition values/range (default: insurance)
<code>G</code>	<code>0:1</code>	3rd condition values/range (default: income growth)
<code>YM</code>	<code>c(10,20,30,40)</code>	4th condition values/range (default: mortgage years)

Example 2 - New Buyer Coefficient

In the next figure, we examine the effects of adjusting the `new buyer coefficient` which guides the rate at which new buyers enter the market. The new buyer coefficient (default of 0.70) to have a range of values $\in \{0.60, 0.65, 0.70, 0.75, 0.80\}$. The colors follow a spectrum from dark blue for the lowest values to dark red for the highest. Each coefficient is applied to each of the four conditions arising from the realtor hedonic and insurance binary conditions. The order of display for overlapping points have been randomized so as to allow better viewing of these points and accompanying lines. The following lines should be run in the `R` session to produce the figure below:

```
if (!exists("x.bs.coef.200")) x.bs.coef.200 = read.csv("csv/x_bs_coef_200.csv")
fig.coef.did(data = x.bs.coef.200, Y = "dBS", off.div = 0.75, ylab = "Buyers - Sellers",
  rel = F, ylim = c(-500, 620))
```



The main observed effect is the new buyers coefficient having a positive impact on the difference between buyers and sellers. This is clearly expected as the coefficient directly determines the influx of new buyers at each time step. However, we also observe a curvilinear trend in those conditions in which the realtor hedonics are Adaptive (or +A). These non-monotonic trends are invariably linked to the buyer count spikes and crashes observed in the previous examples. Moreover, in general, the adaptive hedonics condition appears to consistently result in lower buyer/seller difference (within each buyer coefficient setting), the spikes notwithstanding. The inference from this observation is that the adaptive condition eventually results in fewer buyers remaining in the market.

Description of `fig.coef.did` The `fig.coef.did` plotting function has the following arguments which may be employed to generalize its functionalities. As a general function, it shows separate lines for three conditions as defined by `yvar`. Note that the x -axis values are divided by 2 to reflect the semi-annual frequency of each tick.

Argument	Default	Description
<code>data</code>	<code>x.bs.coef.200</code>	Dataset containing variables of <code>yvar</code>
<code>Y</code>	<code>"avPriceCoastFront"</code>	Variable for y -axis
<code>yvar</code>	<code>c("RH","Ins","nBC")</code>	Variables of the experimental conditions
<code>RH</code>	<code>0:1</code>	1st variables' values/range (default for realtor hedonics)
<code>Ins</code>	<code>0:1</code>	2nd variables' values/range (default for insurance)
<code>nbc</code>	<code>c(0.60,0.65,0.70,0.75,0.80)</code>	3rd variables' values/range (default for new buyer coefficient)
<code>len</code>	<code>15</code>	Number of points to plot on top of lines
<code>ylab</code>	<code>"% @ Time 1"</code>	Label for y -axis
<code>xlab</code>	<code>"Year"</code>	Label for x -axis
<code>ylim</code>	<code>c(100,300)</code>	Plotting limits for the y -axis
<code>legend</code>	<code>c("-A -I","-A +I","+A -I","+A +I")</code>	Legend labels for all combinations of first two conditions
<code>off.div</code>	<code>0.75</code>	Offset divisor for x -axis position to prevent overlapping points
<code>rel</code>	<code>TRUE</code>	Are outcomes are relative to first value of <code>Y</code> (as a percentage)?

Example 3 - Examining Depreciation Effects on the Stability of the Hedonic Model Using Multidimensional Scaling

Depreciation on a property's price occurs when the seller cannot sell for a given duration. We examine the effects of three methods of operationalizing depreciation: 1) depreciation is performed on the current property price ($dm = 0$), 2) the depreciation cumulatively appreciates ($dm = 1$) for multiple sale failures and is applied to the market-based asking price, 3) depreciation is performed on the market-based asking price and resets in the next transaction, if any, and is non-cumulative ($dm = 2$). We regard the cumulative depreciation ($dm = 1$) to be the more correct application of depreciation.

In the following figures, we specifically examine the effects of the depreciation methods on the hedonic regression coefficients. Naturally, we employ the adaptive realtor hedonic condition for these results. Each of the coefficients across time steps have been scaled by the initial value and so they represent relative changes. Each (relative) coefficient set (vector) at each time period has a Euclidean distance to each of the vectors from the other time periods. These distances appear in a $m \times m$ distance matrix (where $m = \text{time steps} = 60$); in this example, we consider 60 time steps (or 30 years reflecting the default mortgage period). Each cell (i, j) in this matrix refers to the Euclidean distance between the relative coefficients of tick i and those of j .

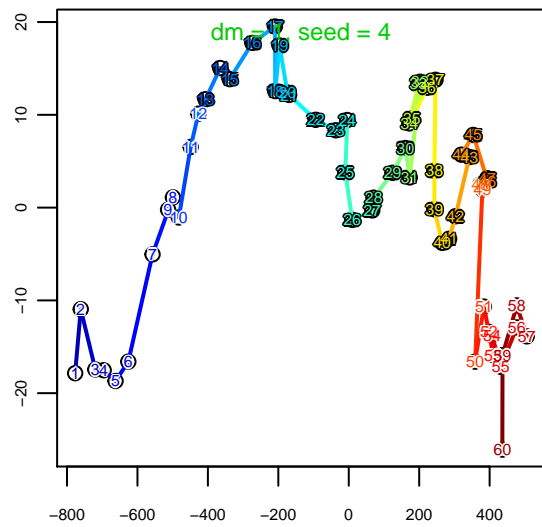
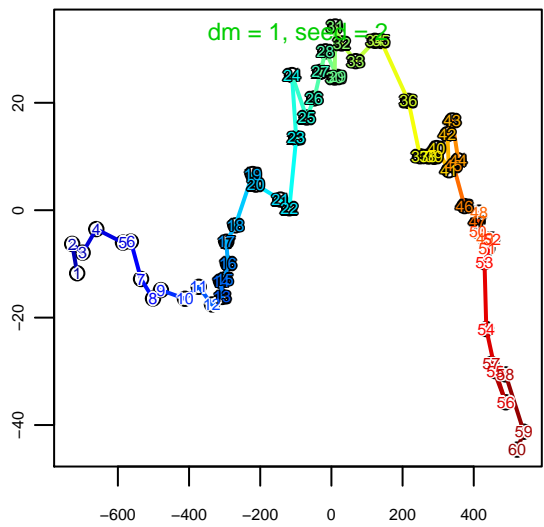
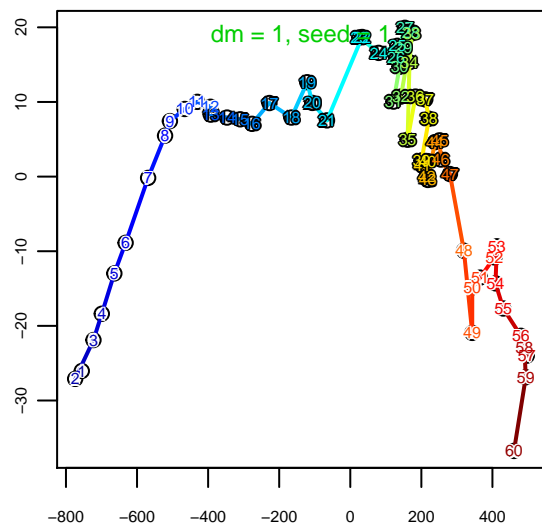
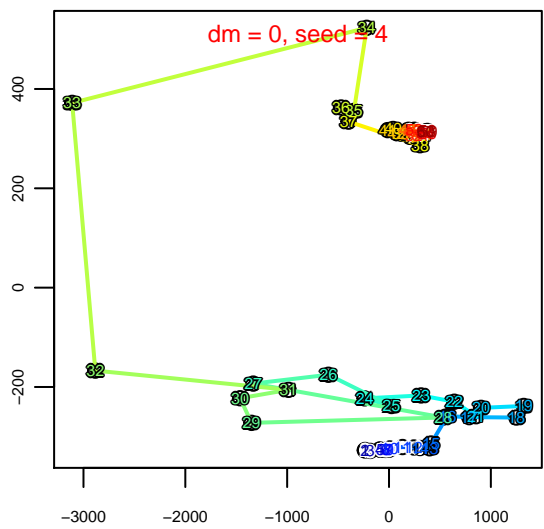
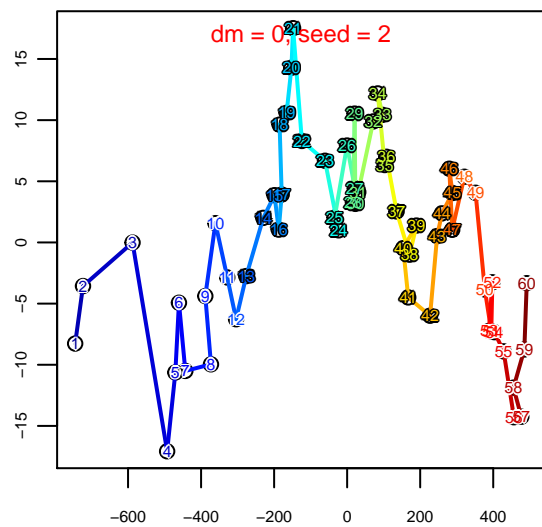
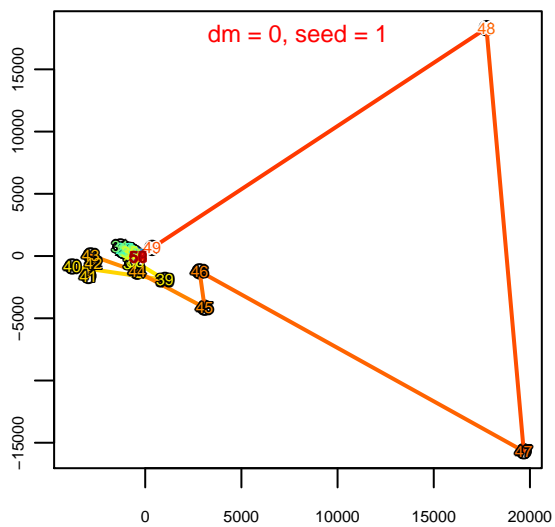
The distance matrix occupies an m -dimensional space which we dimensionally-reduce using dimension reduction methods. Here, we employ classical multidimensional scaling (MDS aka principal coordinate analysis) which employs an eigen-decomposition of the distance matrix. An MDS' result is similar to PCA (principal component analysis); in fact, the unscaled PCA coordinates are identical to MDS. Here, we reduce the m -dimensional set of coefficient vectors to 2-dimensions for visualization purposes.

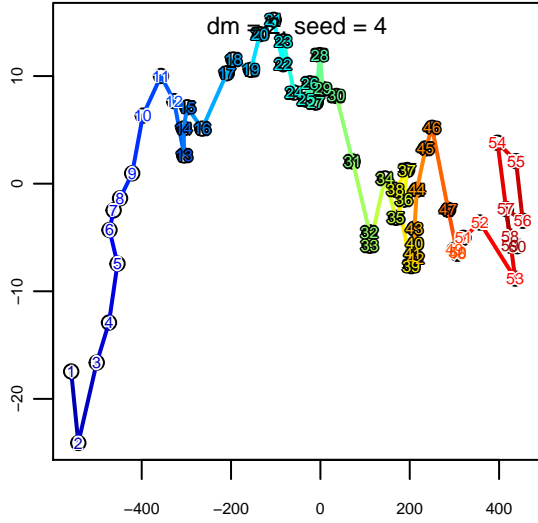
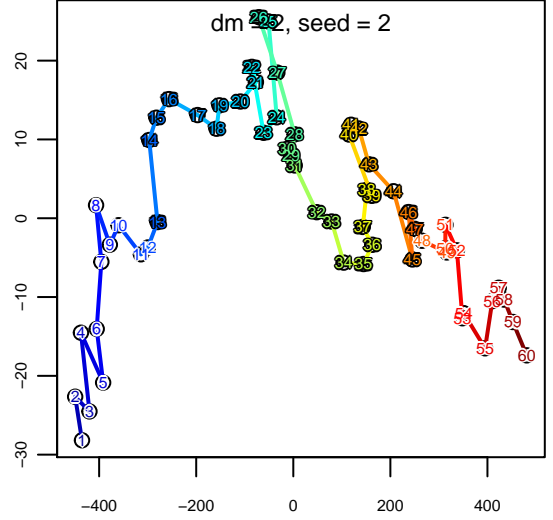
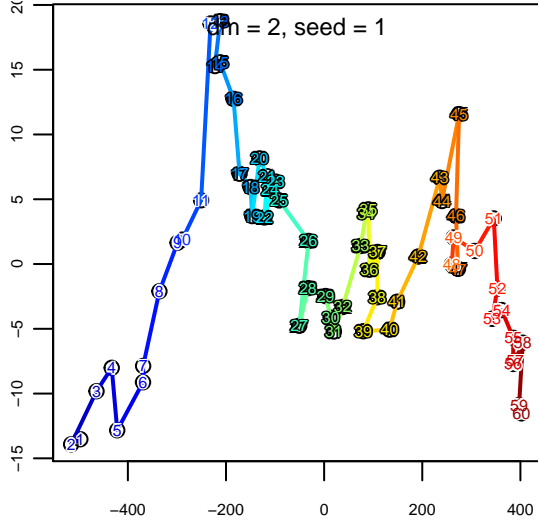
The following figures contain MDS's resulting from the evolution of the hedonic regression from each of the depreciation methods (for three seeds) in order to evaluate both the model's behavior in the context of the hedonic price mechanism. The following lines should be run in the R session to produce the figures below:

```
if (!exists("x.deprec.60")) x.deprec.60 = read.csv("csv/x_deprec_60.csv")
fig.deprec.rdist.did(data=x.deprec.60,mode="rmd")
```

or

```
if (!exists("x.deprec.60")) x.deprec.60 = read.csv("csv/x_deprec_60.csv")
for (dm in 0:2) {
  for (seed in c(1,2,4)) {
    plot.deprec.rdist(data=x.deprec.60,dm=dm,s=seed,cex=.5,cex.axis=.5,lab.line=-1)
  }
}
```





The number within the plot as well as the coloring denote the time steps (dark blue to dark red). The x and y axes denote coordinates in the reduced space and are unitless, hence, no axes labels are shown. The interpretation of the space delineated by the axes is that pairs of points (i.e., vectors of relative coefficients) that far apart in the plot are also far apart according to their Euclidean distances.

The size of the space of each plot can be measured by the range of the axes. In the case of the $\mathbf{dm} = 0$, the trajectories of the coefficient vectors (for seeds 1 and 4) span a much larger space indicating wider variation in their relative values and the instability of the hedonic pricing mechanism under this depreciation mode. The remaining depreciation modes occupy similarly sized spaces, but $\mathbf{dm} = 1$ (the “correct” mode) appears to incur a slightly larger space. Furthermore, the trajectories of $\mathbf{dm} = 1$ and 2 are also roughly unidirectional along one of the dimensions while the two salient trajectories of $\mathbf{dm} = 1$ incur switchbacks (i.e., bi-directionality along with dimensions).

Description of `fig.deprec.rdist.did` The `fig.deprec.rdist.did` plotting function has the following arguments which may be employed to generalize its functionalities. As a general function, it can display an MDS of distances between outcome set (over time) from values of a single experimental variable over selected seeds.

Argument	Default	Description
<code>data</code>	<code>x.deprec.60</code>	Data containing outcomes
<code>D</code>	<code>0:2</code>	Experimental conditions (default: depreciation modes)
<code>S</code>	<code>c(1,2,4)</code>	Selected seeds to visualize
<code>ignore</code>	<code>c(1,2,3,26-0:4)</code>	Column ids of variables in the data to ignore in distance matrix calculation
<code>mode</code>	<code>"rel"</code>	Transformation of the outcomes used in the MDS (default: relative to values at time = 0; alternative: <code>"scale"</code> for standardization)
<code>scalingfn</code>	<code>"cmdscale"</code>	Name of MDS scaling function (default: classical MDS, alternatives are <code>"isoMDS"</code> and <code>"monoMDS"</code> which require the <code>MASS</code> and <code>vegan</code> libraries respectively)

Example 4 - Linear Model

We examine the extent of linear predictability of a key outcome, the average price of coastal front property `avPriceCoastFront`, through three binary, parametric conditions: adaptive/static realtor hedonic, `A`; presence of insurance in the utility calculation, `I`; and engagement of buyer's income growth, `G`.

We report separate models for three mortgage years parameters ($yM \in \{10, 20, 30\}$) and two settings for the parameter that sets the initial fraction of properties on sale ($fos \in \{0.14, 0.20\}$ where the former has been the default for earlier RHEA runs and analysis). We output the table of regression models as plain text as well as laid out as a HTML/PDF table in R markdown (using the `kable()` function). The following lines should be run in the R session to produce the regression results table:

```
if (!exists("x.fos.dm1.200")) x.fos.dm1.200 = read.csv("csv/x_fos_dm1_200.csv")
z = 1.lm.yM2.fos.did(data=x.fos.dm1.200, do.res=F, lr=3, r=0, mode=0, yM=c(10, 20, 30), ret=T, print=T)
```

Predictor	yM=10		yM=20		yM=30	
	fos=0.14	fos=0.20	fos=0.14	fos=0.20	fos=0.14	fos=0.20
Intercept	420*** (4)	420*** (4)	450*** (9)	460*** (6)	403*** (16)	395*** (16)
I(nsurance)	-103*** (4)	-105*** (4)	165*** (10)	146*** (7)	436*** (18)	481*** (19)
A(daptive)	17** (6)	21*** (6)	13 (13)	6 (9)	70** (24)	81** (25)
G(rowth)	47*** (4)	55*** (4)	36*** (10)	22*** (7)	147*** (18)	166*** (19)
IxA	-37*** (7)	-39*** (8)	-61*** (17)	-30** (11)	-153*** (31)	-181*** (32)
IxG	-35*** (7)	-41*** (8)	-26 (17)	-11 (11)	-142*** (31)	-161*** (32)
IxAxG	73*** (8)	63*** (9)	47* (20)	42** (13)	275*** (36)	289*** (37)
Adj-R ²	0.883	0.887	0.707	0.833	0.870	0.878
n	200	200	200	200	200	200

```
kable(z, align=c('l', rep(c('r', 'l'), 6)))
```

Predictor	yM=10		yM=20		yM=30	
	fos=0.14	fos=0.20	fos=0.14	fos=0.20	fos=0.14	fos=0.20
Intercept	420 *** (4)	420 *** (4)	450 *** (9)	460 *** (6)	403 *** (16)	395 *** (16)
I(nsurance)	-103 *** (4)	-105 *** (4)	165 *** (10)	146 *** (7)	436 *** (18)	481 *** (19)
A(daptive)	17 ** (6)	21 *** (6)	13 (13)	6 (9)	70 ** (24)	81 ** (25)
G(rowth)	47 *** (4)	55 *** (4)	36 *** (10)	22 *** (7)	147 *** (18)	166 *** (19)
IxA	-37 *** (7)	-39 *** (8)	-61 *** (17)	-30 ** (11)	-153 *** (31)	-181 *** (32)
IxG	-35 *** (7)	-41 *** (8)	-26 (17)	-11 (11)	-142 *** (31)	-161 *** (32)
IxAxG	73 *** (8)	63 *** (9)	47 * (20)	42 ** (13)	275 *** (36)	289 *** (37)
Adj-R ²	0.883	0.887	0.707	0.833	0.870	0.878
n	200	200	200	200	200	200

The above regression model predicts the dependent variable at time step = 60. The overall impact of yM is lower absolute coefficients for the lower yM (i.e., for both positive and negative coefficient values) despite comparable base prices (i.e., Intercept). However, the trend is not monotonic: some coefficients for $yM = 20$ can reside in between those of $yM = 10$ and 30. And, we also see a lowering of the main effect for insurance even negative for $yM = 10$. Also, fos heightens the impact of all the conditions (hedonics, insurance, and income growth) for $yM \in \{10, 30\}$ but not for 20.

Description of `l.lm.yM2.fos.did` The `l.lm.yM2.fos.did` reporting function has the following arguments which may be employed to generalize its functionalities. As a general function, it reports separate linear regression models for a single dependent variable obtained across combinations of two conditions (here, `yM` and `fos`) predicted by user-defined modelive model of three parametric predictors (here labeled, `A`, `I`, and `G`). The parametric predictors reside in variables named `RH`, `Ins`, and `G` and do not necessarily have to be binary.

Argument	Default	Description
<code>data</code>	<code>x.fos.dm1.200</code>	Data containing predictors and dependent variable
<code>y</code>	<code>"avPriceCoastFront"</code>	Dependent variable for regression models
<code>T</code>	<code>60</code>	Time value(s)/range for the regression data
<code>scale</code>	<code>1000</code>	Scaling value for dependent variable (for presentation purposes)
<code>fn</code>	<code>I</code>	Transformation function on scaled dependent variable (default: indicator function)
<code>yvar</code>	<code>c("yM", "fos")</code>	Variables names for the first two primary conditions
<code>YM</code>	<code>c(10,20,30)</code>	1st condition values/range
<code>FOS</code>	<code>c(0.14,0.20)</code>	2nd condition values/range
<code>form</code>	<code>"RH+Ins+G+Ins:RH+Ins:G+Ins:RH:G"</code>	Predictive formula for regression model
<code>xlab</code>	<code>c("Intercept", "I(nsurance)", "A(daptive)", "G(rowth)", "IxA", "IxG", "IxAxG")</code>	Labels for predictors in the output table
<code>yvar.fmt</code>	<code>c("%d", "%.2f")</code>	<code>sprintf()</code> format strings for <code>yvar</code> values' appearance in the output table
<code>ret</code>	<code>FALSE</code>	Return the final table (w/o aligned cells)
<code>print</code>	<code>TRUE</code>	Print the plain text of the table (w/alignment spacing)

Description of Data Sets

The following table provides basic summaries of each of the data sets (in order of appearance).

Name	Description
x_bs_200.csv	Experiments over insurance and hedonic parameters; contains average prices for property types, buyer/seller stats; 25 seeds per condition; 200 ticks (100 years)
x_yM_10_200.csv	Data borne from insurance disengaged but adaptive realtor hedonics engaged; 200 ticks
x_yM2_200.csv	Experiments over insurance, hedonics, income growth, and mortgage years; 25 seeds per condition; 200 ticks (100 years)
x_bs_coef_200.csv	Similar to x_bs_200.csv but also varying new buyer coefficient from 0.6 to 0.8 in steps of 0.05; 25 seeds per condition; 200 ticks (100 years)
x_deprec_60.csv	Regression coefficients with adaptive realtor hedonics engaged for four seeds with varying depreciation mode (dm); 60 ticks
x_fos_dm1_200.csv	Examines outcomes' dependencies to the fraction_on_sale parameter of 0.14 and 0.20 over mortgage years (yM) of 10, 20, and 30 and depreciation mode (dm) of 1; 200 ticks

Description of the Data Variables/Columns

This table contains descriptions of all of the variables in the data sets employed in this vignette, with the exception of the data set for Example 3 (i.e., `x.deprec.60`); the description of the variables in that data set appears afterwards. A single table is presented for many of the variables span multiple data sets.

Variable	Data Type	Description
Parameter, Seed, and Time Variables:		
Ins	integer (binary)	Insurance
RH	integer (binary)	Adaptive hedonics
G	integer (binary)	Income growth
yM	integer	Mortgage years
nBC	real	New buyers coefficient
s	integer	Random number generator seed
t	integer	Time step (aka tick)
Outcome Variables:		
avPriceFP0	real	Avg price of non-flood risk properties
avPriceFP100	real	Avg price of properties with 1/100 risk
avPriceFP500	real	Avg price of properties with 1/500 risk
avPriceCoastFront	real	Avg price of coastal properties
avPriceNowFP0	real	Avg price of non-flood risk properties (just traded)
avPriceNowFP100	real	Avg price of properties with 1/100 risk (just traded)
avPriceNowFP500	real	Avg price of properties with 1/500 risk (just traded)
avPriceNowCF	real	Avg price of coastal properties (just traded)
nTradedNowFP0	integer	# of traded properties that are non-flood risk
nTradedNowFP100	integer	# of just traded properties that are 1/100 risk
nTradedNowFP500	integer	# of just traded properties that are 1/500 risk
nTradedNowCF	integer	# of just traded properties that are coastal
nT	integer	# of parcels just traded
nB	integer	Current # of buyers
nS	integer	Current # of sellers
nNC	integer	# of newcomers
nOS	integer	# of old sellers
nOB	integer	# of old buyers
nBL	integer	# of buyers who left
dBS	integer	Difference between # of buyers and sellers
nD	integer	Difference between # of buyers and sellers

The following table provides details about the data set employed in Example 3 (`x.deprec.60`). These data are primarily regression coefficients inferred in the hedonic model. These coefficients will evolve only for the ‘adaptive’ realtor hedonic condition.

Variable	Data Type	Description
Parameter, Seed, and Time Variables:		
d	integer	Depreciation Mode
s	integer	Random number generator seed
time	integer	Time step or tick
Regression Coefficients:		
intercept	real	Intercept
bathrooms	real	Number of bathrooms
bathrooms.2	real	(Number of bathrooms) ²
age	real	Age of the house on the property
age.2	real	Age ²
SQFT	real	Square footage of the house
SQFT.2	real	SQFT ²
lotsize	real	Size of the property
lotsize.2	real	lotsize ²
newhome	real	Is the house new?
postFirm	real	Is the property post-FIRM (Flood Insurance Rate Map)
FP100	real	Is the parcel at 1/100 flood risk?
FP500	real	Is the property at 1/500 flood risk?
coastalFront	real	Is the property coastal front?
log.distAmen.	real	log(Distance to beach)
log.distCBD.	real	log(Distance to center of business)
log.distHwy.	real	log(Distance to nearest highway)
log.distPark.	real	log(Distance to nearest park)