

Supplement to:

**“I’m not hoarding, I’m just stocking up before the hoarders get here.”
Behavioral causes of phantom ordering in supply chains**

John D. Sterman*
MIT Sloan School of Management, Cambridge MA, USA

Gokhan Dogan
TAV, Istanbul, Turkey

We are grateful to Rachel Croson, Karen Donohue, and Elena Katok for providing the data analyzed here, and for helpful comments. Financial support from the Project on Innovation in Markets and Organizations at the MIT Sloan School of Management.

* Corresponding author: jsterman@mit.edu

Supplement to:

**“I’m not hoarding, I’m just stocking up before the hoarders get here.”
Behavioral causes of phantom ordering in supply chains**

0. Introduction

Here we provide documentation of the methods used to estimate the parameters and confidence intervals for the different models reported in the paper, and a guide to the data, models and scripts needed to replicate the analysis, all of which are included in the electronic appendices to this supplement.

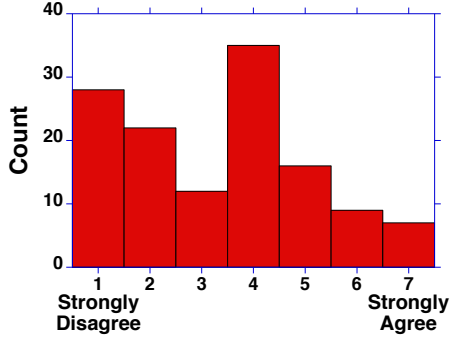
1. Questionnaire and Responses

CDKS (2014) presented each participant a questionnaire after the session ended, shown here:

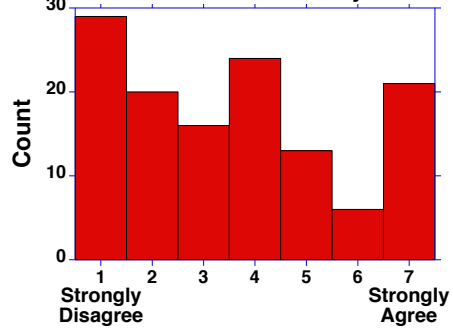
	Strongly Disagree			Neither agree nor disagree			Strongly agree
1	My team probably performed better [had lower costs] than the other teams.						
	☺	☺	☺	☺	☺	☺	☺
2	I probably performed better [had lower costs] than the other members of my team.						
	☺	☺	☺	☺	☺	☺	☺
3	The position I played was harder than the other positions in my team.						
	☺	☺	☺	☺	☺	☺	☺
4	My teammates made sensible decisions.						
	☺	☺	☺	☺	☺	☺	☺
5	It was easy to anticipate the orders I received.						
	☺	☺	☺	☺	☺	☺	☺
6	It was easy to keep my inventory close to the value I desired.						
	☺	☺	☺	☺	☺	☺	☺
7	Some of my teammates made mistakes that raised my costs.						
	☺	☺	☺	☺	☺	☺	☺
8	Some of my teammates made mistakes that raised our team costs						
	☺	☺	☺	☺	☺	☺	☺

Histograms of the responses to these questions are shown below:

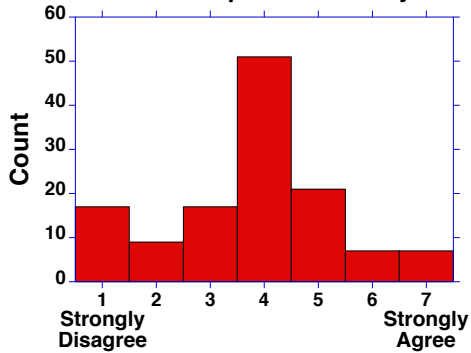
My team probably performed better [had lower costs] than the other teams.



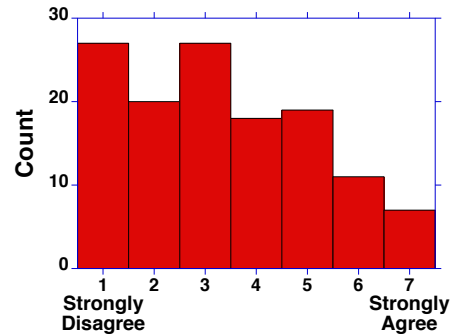
I probably performed better [had lower costs] than the other members of my team.



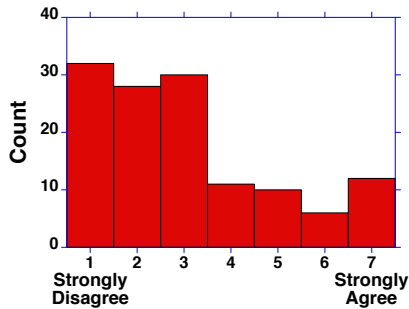
The position I played was harder than the other positions in my team



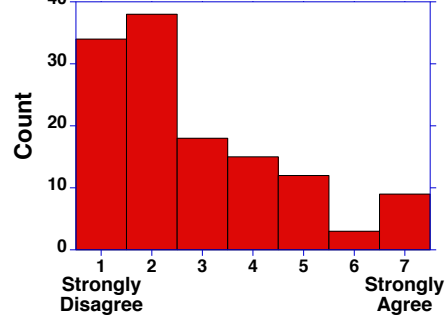
My teammates made sensible decisions.



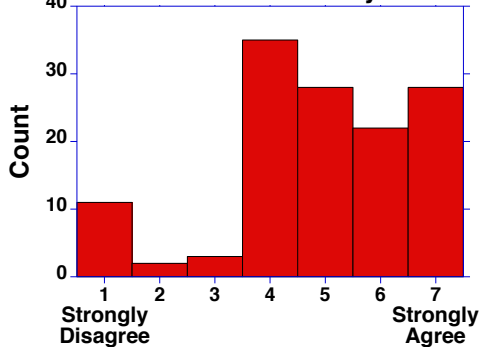
It was easy to anticipate the orders I received



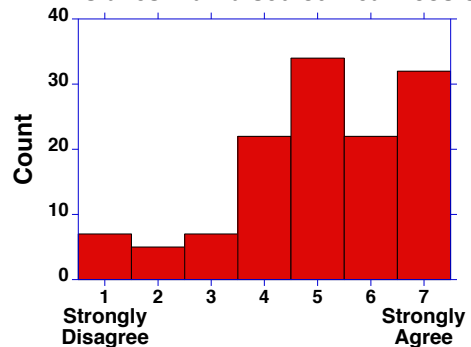
It was easy to keep my inventory close to the the value I desired.



Some of my teammates made mistakes that raised my costs.



Some of my teammates made mistakes that raised our team costs.



Participants were also asked several open-ended questions:

1. Please explain the strategy you used in the game. For example, consider: How did you decide how much to order each week? How much inventory did you want to have? Why? What factors did you consider in placing your orders?
2. What strategy do you think your customer used to place the orders you received? Note: If you were the Factory, your customer is the Distributor; the customer of the Distributor is the Wholesaler, and the customer of the Wholesaler is the Retailer.
3. What caused your inventory/backlog to behave as it did?
4. What information would have helped you do better?

The paper provides a sample of responses to question 1 and 3 to illustrate how some participants set desired on-hand inventory to respond to both the incoming orders and variability in them, and to show how many participants found supplier deliveries to be late and inconsistent. Below we provide additional examples of responses to question 3. Retailers, who always received constant demand, naturally focus on poor supplier performance to a greater degree than upstream individuals, whose explanations for the behavior of their inventory included unexpected variability in incoming orders along with poor supplier delivery performance:

“I did not receive the amount of beer as quickly as I thought I would have.” [Retailer]

“My wholesaler couldn't cover my inventory when I had a low inventory. I'm not sure why they didn't plan to have enough on hand even though they knew that I would need 4 per week....” [Retailer]

“I couldn't predict when I would receive the orders” [Retailer]

“The unpredictableness [sic] of the shipment I would receive was the cause of my backlog or inventory. “ [Retailer]

“Insufficient orders being filled at times.” [Retailer]

“My provider sent me some bad shipments some weeks.” [Wholesaler]

2. Models and Estimation

Models: The paper reports estimation results for three models for each of the participants in the sample. Model 0 (see eq. 8 in the paper) assumes, as in prior studies, that the desired on-hand and on-order stocks, S^* and SL^* , are constant. However, defining effective total desired on-hand and on-order stock $S' = S^* + bSL^*$, and assuming an additive error term, the system to be estimated becomes:

$$\begin{aligned}
\widehat{O}_t &= \max\left(0, D_t^e + \alpha_s(S' - S_t - \beta SL_t) + \varepsilon_t\right) \\
D_t^e &= \psi D^c + (1 - \psi) D_t^p \\
D_t^p &= \theta D_{t-1} + (1 - \theta) D_{t-1}^p \\
&\text{subject to } 0 \leq \psi, \theta, \alpha_s, \beta \leq 1; 0 \leq S'
\end{aligned} \tag{Model 0}$$

Models 1 and 2 explicitly separate the desired on-hand and on-order stocks:

$$\widehat{O}_t = \max\left(0, D_t^e + \alpha_s\left((S_t^* - S_t) + \beta(SL_t^* - SL_t)\right) + \varepsilon_t\right) \tag{Models 1-2}$$

with D^e as in Model 0, and the following formulations for S^* and SL^* . In Models 1 and 2, desired on-hand stock is an estimated target coverage goal of expected orders, $S_t^* = \gamma D_t^e$ (eq. 10).

Desired on-order stock, SL^* is modeled as

$$SL_t^* = \lambda_t^e R_t^* . \tag{11}$$

where λ^e is the expected lag in receiving orders and R^* is the desired acquisition rate—the rate at which the participant seeks to receive goods from the supplier.

We formulate the expected delivery delay as a linear function of the current delivery delay, denoted the perceived delay, λ^p , but constrained to be less than some maximum, λ^M :

$$\lambda_t^e = \min\left(\lambda^M, \kappa + \omega \lambda_t^p\right) \tag{12}$$

where κ , ω and λ^M are parameters to be estimated. As discussed in the paper, the steady-state delivery delay is, by Little's Law, the ratio of the supply line to deliveries received, SL/R .

However, the delivery delay can never be less than the normal value λ^* , even if the supply line falls to zero. Similarly, participants may receive nothing from their supplier, but we assume that their estimate of the delivery delay remains finite by constraining expected deliveries to be the maximum of actual deliveries, R_t , and 1 case/week. Hence,

$$\lambda_t^p = \max\left(\lambda^*, SL_t / \max(R_t, 1)\right). \tag{13}$$

Two alternatives are considered for R^* , the desired rate at which participants seek to receive goods from their supplier: (i) participants seek to receive goods at the rate they expect their customers will place orders:

$$R_t^* = D_t^e \tag{14a}$$

or (ii) participants desire to receive goods at the rate needed to fill expected customer orders *and*

eliminate any inventory gap they have:

$$R_t^* = \max\left(0, D_t^e + \alpha_s (S^* - S)\right). \quad (14b)$$

where the max function ensures that R^* remains nonnegative even if there is surplus inventory..

The table below summarizes the differences among the models:

	Constant desired supply line	Desired Supply Line rises with expected supplier lead time	
	$SL_t^* = \text{constant}$	$SL_t^* = \lambda_t^e D_t^e$	$SL_t^* = \lambda_t^e * \left(D_t^e + \alpha_s (S_t^* - S_t)\right)$
$S_t^* = \text{constant}$	Model 0		
$S_t^* = \gamma D_t^e$		Model 1	Model 2

Parameter Estimation: All models were estimated by nonlinear least squares using the Vensim simulation software, VensimDSSDP (Vensim Decision Support System, double precision), Version 6.2 (<http://vensim.com>). Vensim uses a modified Powell hill-climbing algorithm for optimization and parameter estimation (the Vensim documentation provides details). To limit the possibility of finding only a local optimum in parameter space, we ran the optimizer multiple times for each model/participant pair, using random starting points in parameter space. The maximum number of starting points was set to 1000. Each model typically executed between 100,000 and 1,000,000 simulations before converging to the best-fit parameters. Section 3 reports the computational burden for the analysis. The Vensim Optimization Control files (appended to this supplement) contain the parameters for the Vensim Powell algorithm and the list of parameters to be estimated, with their allowable ranges. The Vensim Payoff Definition file <payoff.vpd> specifies the criterion to be minimized, specifically the sum of the squared error between simulated and actual orders for each participant over the 48 weeks of each trial.

The parameters to be estimated, with their allowable ranges for the search algorithm, are:

$$\text{All Models: } 0 \leq \psi, \theta, \alpha_s, \beta \leq 1$$

$$\text{Model 0: } 0 \leq S' \leq 5000 \text{ cases}$$

$$\text{Models 1, 2: } 0 \leq \gamma \leq 100 \text{ weeks; } 0 \leq \kappa \leq 50; 0 \leq \omega \leq 10; 4 \leq \lambda^M \leq 20000.$$

To replicate the parameter estimation for each model and participant:

1. Make sure the appropriate version of Vensim is installed (VensimDSSDP).
2. Open the folder Parameter Estimation. Open Model0JS.mdl in Vensim. Save.
3. From the File menu, choose Edit File. Select the file Model0JS estimation.cmd.
4. From the File menu, choose Run Commands. The command script (.cmd file) will execute, estimating the parameters for each subject in turn.
5. Repeat steps 2-5 for the other models and command scripts (e.g., Model3JS and Model3JS estimation.cmd) to generate the parameter estimates for the other models.

Bootstrapping: We used bootstrapping to estimate the confidence intervals around the best-fit parameters for each model/participant combination (Dogan 2006 and Struben, Keith, Sterman (forthcoming) provide examples and compare bootstrapping to other methods e.g. the Wald test). For each of the models estimated for each participant, we generated a sample of 500 bootstrap simulations. The bootstrap procedure is:

1. For each model, m , and participant, i , generate a set of bootstrap samples, indexed by $j \in \{1, 500\}$. The ordering decision rule with the best-fit parameters for each model/participant combination is used to generate a new sample of “actual” orders according to:

$$\text{Bootstrap Orders}_{mijt} = \text{MAX}(0, \text{Estimated Orders}_{mit} + \text{Bootstrap Error}_{mijt}) \quad (\text{S1})$$

where Estimated Orders_{mit} is the set of orders, for $t = 1$ to 48 weeks, produced by the best-fit model for each model/participant combination. The MAX function ensures that the orders remain nonnegative. The Bootstrap Errors for each sample j and period t are distributed as $N(0, \sigma_{mi}^2)$, where σ_{mi} is the standard deviation of the residuals of the best-fit estimates for each model/participant combination. A different random number generator seed is used for each bootstrap j . The vensim model <order_generator.mdl> and associated command scripts needed to generate the bootstrap samples for each model/participant combination are appended to this supplement (see below for step-by-step instructions to replicate the bootstrap analysis).

2. For each of the 500 bootstrap orders for each model and participant, estimate the parameters of the relevant model using the same procedure described under Estimation, above. The Vensim command scripts for the estimation for each model/participant combination are appended to this supplement.

3. Use the distribution of estimated parameters to estimate the confidence intervals for each parameter or test hypotheses. We used the 2.5th and 97.5th percentiles of the resulting distribution as our estimators of the 95% confidence interval for each parameter.

To replicate the bootstrap sample generation and parameter estimation for each model and participant:

1. Make sure the appropriate version of Vensim is installed (VensimDSSDP).
2. Open the folder Bootstrapping. Bootstrap scripts are set up for each model reported in the paper (Model0, Model3, Model4). To keep the command scripts and their execution time at reasonable length, we created subfolders for each model, and within these, for the first 100 bootstrap samples and the 101st through 500th samples for different groups of subjects. The instructions here apply to the folder <model 0 part 1 first 100>, which generates the first 100 bootstrap samples using Model 0 for the first five participants. The procedure is the same for the other subfolders.
3. Open order_generator.mdl in Vensim. Save.
4. From the File menu, choose Edit File. Select the file <order generator script model0 part1.cmd>.
5. From the File menu, choose Run Commands. The command script (.cmd file) will execute, generating 100 bootstrap samples for each of the first five participants and storing the results in .vdf files.
6. After the command script finishes, close the .cmd file and order_generator model. Open <Model0JS.mdl>. Save.
7. From the File menu, choose Edit File. Select the file <bootstrap script model0 part1.cmd>.
8. From the File menu, choose Run Commands. The command script (.cmd file) will execute, estimating the parameters for each of the first 100 bootstrap samples for each participant in turn.
9. Repeat steps 3-8 for the other models and command scripts (e.g., model 0 part 2 first 100) to generate the parameter estimates for the other participants, models, and bootstrap samples.

3. Sensitivity Analysis: Alternate models

In addition to Models 1 and 2, we tested the sensitivity of the results to an alternative formulation for hoarding in which S^* is assumed to be constant. The formulations for SL^* remained as in Models 1 and 2. Constant S^* allows us to estimate the extent of hoarding ($S^* \gg 0$, the optimal level) independently of the participants' estimate of expected demand. The alternate models with constant S^* improved on the RMSE of Models 1 and 2 for a few participants, and the estimated parameters, including those related to phantom ordering, were consistent with the models above. For clarity, therefore, we do not report the results in the paper. However, for

completeness, the files needed appended to this supplement include the models and scripts needed for all four alternative models. Note that in the files supplied here, the models denoted Models 1 and 2 in the paper are denoted Models 3 and 4, while the alternative models with constant S^* are denoted models 1 and 2. The table below shows the differences among these models:

	Constant desired supply line	Desired Supply Line rises with expected supplier lead time	
	$SL_t^* = \text{constant}$	$SL_t^* = \lambda_t^e D_t^e$	$SL_t^* = \lambda_t^e * (D_t^e + \alpha_S (S_t^* - S_t))$
$S_t^* = \text{constant}$	Model 0	Model 1 (not reported in paper but contained in Vensim scripts)	Model 2 (not reported in paper but contained in Vensim scripts)
$S_t^* = \gamma D_t^e$	NA	Model 3 (denoted Model 1 in paper)	Model 4 (denoted Model 2 in paper)

4. Computational Burden

Estimating the parameters for each model/participant combination required $5 * 171 = 855$ estimations. Estimating the parameters for each of the 500 bootstrap samples, for Models 0, 3, and 4, required $500 * 3 * 171 = 256,500$ estimations. On average, estimation of any given model/participant combination required approximately 200,000 simulations of the model to ensure that the estimates converged to the global minimum sum of squared errors. Estimating the five models for each participant, and 500 bootstrap samples for Models 0, 3, and 4, for each of the 171 participants, required a total of approximately 25.8 billion simulations. The estimation and bootstrap analysis were carried out over a period of about two months (calendar time, not execution time) on different computers, including four Macbook Pro machines and a virtual PC with 8 cores running on the university network.

5. Vensim optimization control files

The files listed below document the settings for the Powell optimizer used by Vensim to estimate the parameters, along with the parameters and their allowable ranges.

Model 0 (Model0.voc):

```
:OPTIMIZER=Powell
:SENSITIVITY=Off
:MULTIPLE_START=Random
:RANDOM_NUMER=Linear
:SEED=3
:OUTPUT_LEVEL=Off
:TRACE=Off
:MAX_ITERATIONS=1000
:RESTART_MAX=1000
:PASS_LIMIT=2
:FRACTIONAL_TOLERANCE=0.0003
:TOLERANCE_MULTIPLIER=21
:ABSOLUTE_TOLERANCE=1
:SCALE_ABSOLUTE=1
:VECTOR_POINTS=25
0<=Theta =0.5<=1
0<=Fraction of Discrepancy Ordered Each Period =0.5<=1
0<=Fraction of Supply Line Taken into Account =0.5<=1
0<=Constant Desired Inventory=4<=5000
0<=Weight on Customer Orders=.5<=1
```

Models 1-2 (Model12.voc)

```
:OPTIMIZER=Powell
:SENSITIVITY=Off
:MULTIPLE_START=Random
:RANDOM_NUMER=Linear
:SEED=3
:OUTPUT_LEVEL=Off
:TRACE=Off
:MAX_ITERATIONS=1000
:RESTART_MAX=1000
:PASS_LIMIT=2
:FRACTIONAL_TOLERANCE=0.0003
:TOLERANCE_MULTIPLIER=21
:ABSOLUTE_TOLERANCE=1
:SCALE_ABSOLUTE=1
:VECTOR_POINTS=25
0<=Weight on Customer Orders=.5<=1
0<=Theta =0.5<=1
0<=Fraction of Discrepancy Ordered Each Period =0.5<=1
0<=Fraction of Supply Line Taken into Account =0.5<=1
0<=Constant Desired Inventory=4<=500
0<=Constant Desired Supply Line=16<=50
0<=Sens of Des Supply Line to Exp Delivery Delay=1<=10
4<=Maximum Expected Delivery Delay=500<=20000
```

Models 3-4 (Model34.voc)

```
:OPTIMIZER=Powell
:SENSITIVITY=Off
:MULTIPLE_START=Random
:RANDOM_NUMER=Linear
:SEED=3
:OUTPUT_LEVEL=Off
:TRACE=Off
:MAX_ITERATIONS=1000
:RESTART_MAX=1000
:PASS_LIMIT=2
:FRACTIONAL_TOLERANCE=0.0003
:TOLERANCE_MULTIPLIER=21
:ABSOLUTE_TOLERANCE=1
:SCALE_ABSOLUTE=1
:VECTOR_POINTS=25
0<=Weight on Customer Orders=.5<=1
0<=Theta =0.5<=1
0<=Fraction of Discrepancy Ordered Each Period =0.5<=1
0<=Fraction of Supply Line Taken into Account =0.5<=1
0<=Constant Desired Supply Line=16<=50
0<=Sens of Des Supply Line to Exp Delivery Delay=1<=10
4<=Maximum Expected Delivery Delay=500<=20000
0<=Desired Inv Coverage=1<=100
```

6. Appended files to replicate analysis

The following files and folders are appended in electronic form as a .zip archive, which contains:

Folder	File Name	Type	Explanation
Data and Results	Estimation Results.xlsx	Excel	Results of estimation and bootstrap analysis for all models and participants
	All Human Data.xls	Excel	Raw data for all participants
Parameter Estimation	Model0JS.mdl	Vensim model	Model 0
	Model1JS.mdl		Model 1 (not reported in paper)
	Model2JS.mdl		Model 2 (not reported in paper)
	Model3JS.mdl		Model 3 (denoted in paper as Model 1)
	Model4JS.mdl		Model 4 (denoted in paper as Model 2)
	Model0.voc	Vensim Optimization Control file	Optimization specifications and parameters for Model 0
	Model12.voc		Optimization specifications and parameters for Models 1 and 2
	Model34.voc		Optimization specifications and parameters for Models 3 and 4
	Model0JS estimation.cmd	Vensim command script	Script to estimate parameters for each subject, using the specified model
	Model1JS estimation.cmd		
	Model2JS estimation.cmd		
	Model3JS estimation.cmd		
	Model4JS estimation.cmd		
	Payoff.vpd	Vensim payoff definition	The payoff function for the estimation (minimize sum of squared errors between actual and estimated orders)
Savelist.lst	Vensim variable list file	The list of the variables to be saved for each estimation	
subjectX.vdf	Vensim data file	Data for each subject X	
Bootstrapping			Contains subfolders with models and files needed to generate bootstrap samples for each model and participant, e.g. Model 0 Bootstraps 0-100, Model, Model 0 Bootstraps 101-500, etc.
Example: Model 0 Bootstraps 0-100>model 0 part 1 first 100			Folder to run the first 100 bootstrap samples for the first five participants
	Order_generator.mdl	Vensim model	Model to generate the bootstrap samples using the estimated decision rule and standard deviation of residuals for each participant
	order generator script model0 part1.cmd	Vensim command script	Script to generate the bootstrap samples for model 0 part 1 (first five participants)
	Model0JS.mdl	Vensim model	Model 0.
	Model0.voc	Vensim Optimization Control file	Optimization specifications and parameters for Model 0
	bootstrap script model0 part1.cmd	Vensim command script	Script to estimate the best-fit parameters for each of the bootstrap samples for model0 (first five participants)
	Payoff.vpd	Vensim payoff definition	The payoff function for the estimation (minimize sum of squared errors between actual and estimated orders)
	btstrp_savelist	Vensim variable list file	The list of variables to be saved for each bootstrap estimation
	sX-model0js_stdev_1.vdf	Vensim data file	Data file for participant X containing data specifying the standard deviation of the residuals for model 0, participant X; used to generate the bootstrap samples for that participant

7. References

Dogan, G. 2006. Bootstrapping for confidence interval estimation and hypothesis testing for parameters of system dynamics models, *System Dynamics Review* 23(4): 415-436.

Struben, J., Sterman, J., Keith, D. (forthcoming). Parameter and Confidence Interval Estimation in Dynamic Models: Maximum Likelihood and Bootstrapping Methods. *Analytical Handbook for Dynamic Modelers*. R. Oliva, H. Rahmandad and N. Osgood (eds). Cambridge, MA, MIT Press.