This document describes the data files and program files used for "Bundle-size Pricing as an Approximation to Mixed Bundling." There are three groups of files:

1. The MATLAB programs used to calculate optimal prices and profits for various combinations of taste distributions and marginal cost structures, as described in section 2.2 of the paper. These are in the directory called `pricing_solutions`.
2. A Stata dataset containing the relevant output from these MATLAB programs, and some Stata do-files used to generate tables and figures from these data. In the directory called `numerical_analyses`.
3. The data and MATLAB programs used in the empirical application to theater ticket pricing, as described in section 3 of the paper. In the directory called `theater_application`.

The included files are listed on the following pages.

In `pricing_solutions`:

`sim(#).m`	Calculates optimal prices and profits for a firm selling `(#)` products. The ranges of parameters for the taste distribution and marginal cost structure are passed as arguments. (See the comments in the files for further explanation.)
`sim(#)cap.m`	Same as the `sim(#).m` files, but modified to handle capacity constraints.
`DrawValuations.m`	Subroutine for drawing a large set of simulated consumers from the specified distribution of tastes. See comments in file for more explanation.
`Constraints.m`	Determines constraints on prices depending on the pricing scheme. See comments in file for more explanation.
`MBConstraints.m`	Constraints on prices under mixed bundling. See comments in file for more explanation.
`SmoothProfit_(#).m`	Subroutine that calculates the profit function for the `(#)`-good case. See comments in file for more explanation.
`SmoothQuants_(#).m`	Subroutine that calculates purchased quantities for the `(#)`-good case. See comments in file for more explanation.
`ConsumerSurplus_(#).m`	Subroutine for calculating consumer surplus in the `(#)`-good case. See comments in file for more explanation.
`CapProfit_(#).m`	Modification of `SmoothProfit_(#).m` that incorporates capacity constraints.
`CapSurplus_(#).m`	Modification of `ConsumerSurplus_(#).m` that incorporates capacity constraints.
`UPQuants_(#).m`	Subroutine that calculates quantities under UP in the scenarios with capacity constraints. See comments in file for more explanation.

In `numerical_analyses`:

`all-sims.dta`	Stata dataset containing the output from the MATLAB programs listed above. Tables and figures in Section 2.2 of the paper are all based on this dataset. The variable names are self-explanatory.
`pdiff_table`	Subdirectory containing files for creating Table 3 in the paper. (The `notes.txt` file explains the files.)
`regression.do`	Generates the numbers in Table 4.
`welfare_table.do`	Generates most of the numbers in Table 5.
`welfare_table_pb.do`	Generates the PB column of Table 5.
`figure2_k.do`	Makes Figure 2 in the paper.
`figure2_correlation.do`	Makes Figure 3.
`figure2_mc.do`	Makes Figure 5.
`figure2_family.do`	Makes Figure 6.
`figure4`	Subdirectory containing files for generating Figure 4. See `notes.txt` for explanation.
`appendix-tables.do`	Stata do-file that generates the tables in the online appendix.

In `theater_application`:

`gmm_algorithm.m`	The main estimation program. Calls all other program components.
`datafile.m`	Loads the relevant data for estimating the structural model. Called by `gmm_algorithm.m`. *(Note: please contact the authors to obtain this file. It can be made available, but since it contains proprietary information of TheatreWorks, we will want to coordinate with them to make sure any necessary confidentiality agreements are made.)*
`gmmobj_fxn.m`	Returns the GMM objective function. Called by `gmm_algorithm.m`.
`gmmmoments_fxn.m`	Returns the moments of the GMM function, used to compute the standard errors. Called by `gmm_algorithm.m`.
`negprofit_fxn.m`	Returns -1 times profits as a function of prices, used to compute the pricing optimality condition. Called by `gmmobj_fxn.m` and `gmmmoments_fxn.m`.
`priceelast.m:`	Computes price elasticities. Called by `gmm_algorithm.m`.
`fminsearch10.m`	A slight modification of Matlab's standard `fminsearch` Nelder-Mead simplex search program that initializes with a larger initial simplex. Called by `gmm_algorithm.m` for search over parameters; called by `gmmobj_fxn.m` and `gmmmoments_fxn.m` for search over prices.