README for Aryal, Ciliberto, and Leyden (2021): Coordinated Capacity Reductions and Public Communication in the Airline Industry

Overview

The code provided here replicates the analysis in Aryal, Ciliberto, and Leyden (2021), using a combination of Stata and Python. The full set of figures and tables can be recreated using GNU Make, as described below.

Data Summary

All of the cleaned data used in the analysis have been provided with this replication package. The raw data sources are listed below. Note that some of the raw data, namely the OAG Market Intelligence-Schedules dataset is not publicly available. That data is available via OAG.

Raw Data Sources

U.S. Bureau of Transportation Statistics's (BTS) T-100 Domestic Segment for U.S. Carriers dataset: Data on U.S. carriers' non-stop segment flights.

- This data is publicly available at BTS Transtats.

BTS's On-Time Performance dataset: Data on U.S. carriers' flight times.

- This data is publicly available via BTS Transtats.

OAG Market Inteligence: Data identifying the operating and ticketing carrier for each segment at the quarterly level

- This data is available for purchase from at https://www.oag.com

Provided data files

All of the data used in the analysis is provided in `./data/`. The primary data file used in the analysis is `manual_capacity_disipline_airport.dta`.

- `20170928_manual_talk_indicators.csv`: A table indicating whether capacity discipline was discussed for each available transcript.
- `auto_capacity_discipline_airport.dta`: Data on flights and communication decisions for airport-pair markets, using only the automated communication classification process, as discussed in Appendix D.
- `manual_capacity_discipline_airport.dta`: Data on flights and communication decisions for airport-pair markets, using the automated + manual communication classification process described in Section 2.2.
- `manual_capacity_discipline_airport_cong.dta`: Market level data on flight departure crowding and communication decisions for airport-pair markets, using the automated + manual communication classification. See Section 3.3.1.
- `manual_capacity_discipline_airport_control-function.dta`: Data on flights and communication decisions for airport-pair markets, using the automated + manual communication classification process described in Section 2.2, including the distance measurements used in Appendix A.
- `manual_capacity_discipline_airport_prices-markets.dta`: Quarterly data on flight prices and communication decisions for airport-pair markets, using the automated + manual communication classification process described in Section 2.2. See Section 3.3.2.
- `manual_capacity_discipline_airport_realQuarter.dta`: Data on flights and communication decisions for airport-pair markets, using the automated + manual communication classification process described in Section 2.2, where the communication and flight data has been aligned according to the alternative process described in Appendix E.

- manual_capacity_discipline_city.dta: Data on flights and communication decisions for city-pair markets, using the automated + manual communication classification process described in Section 2.2. See Appendix C.
- ra_capacity_discipline_airport.dta: Data on flights and communication decisions for airport-pair markets, using communication classification conducted independently by a research assistant, as described in Appendix D.
- transcript_status.csv: A table indicating when a carrier had a transcript available, and the reason for any missing transcripts.

Computational requirements

Software Requirements

- Stata (code was last run with Stata MP-4 version 16)
  - estout (installed 2019-12-27)
  - tabout (installed 2019-12-27)
  - vallist (installed 2019-12-30)
  - To install all Stata dependencies, run ./stata_setup.do.
- Python 3.9.4
  - matplotlib 3.3.3
  - numpy 1.17.2
  - palettable 3.3.0
  - pandas 1.1.5
  - seaborn 0.11.2
  - To install all Python dependencies, please run "pip install -r requirements.txt" as the first step. See https://pip.readthedocs.io/en/1.1/requirements.html for further instructions on using the "requirements.txt" file. It is recommended that you set up a virtual environment before installing these dependencies. See Creation of virtual environments.
- GNU Make
  - The master file for running all of the analysis in this paper is ./Makefile, which can be run using GNU Make. GNU Make allows the replicator to easily replicate the entire paper, and is available on Linux and MacOS. See below for more information on using GNU Make to replicate this paper's analysis.
  - Note that all of the analysis in this paper can be replicated without the use of GNU Make, by running the Python and Stata files directly.

Computational Details

The published analysis was run on a 2017 Retina 5K iMac, with a 4.2 GHz Quad-Core Intel Core i7 processor and 64 GD 2400 MHZ DDR4 RAM, running MacOS Version 11.5.2.

Notes and Instructions for Replicators

Notes on Directory Structure

- All files for constructing the figures in the paper are available in figures/.
- All files for estimating the regressions that involve airport-pair markets, other than the conditional exogeneity regressions (Section 4.1) and the control function regressions (Appendix A) are available in regressions/code/airport/.
- All files for estimating the conditional exogeneity regressions are available in regressions/code/cond_exog/.
- All files for estimating the control function regression are available in regressions/code/control_function/.
- All files for constructing tables in the paper are available in tables/code/. Note that to construct tables with regression results, the replicator should first estimate the relevant regressions in regressions/code.
- The files available in misc_descriptive/ can be used to construct a few ad hoc statistics cited in the paper.

How to Replicate Analysis

To replicate the results of the paper on Linux or MacOS:

- Edit line 2 in `Makefile` to provide your machine's CLI command for running Stata.
- The Makefile can be used to produce any of the final exhibits in the paper, by navigating to the root directory of this package in your shell client, and then running `make <file to be recreated`.
  - E.g., `make tables/output/tab_airport_primary.tex` will reproduce Table 4, and will automatically run the relevant regressions (if they have not already been estimated).
  - The "Aggregate Commands" section of `Makefile`, beginning on line 612, provide a few commands for running specific portions of the paper. E.g., `make run_all` will run everything, and `make run-main-regressions` will run only the regressions in the main paper (i.e., not in the appendix).

Note that all Python and Stata files can be run separately, but that care should be taken to provide the required arguments to those files.

- Many regressions require that a `fe_name` argument be specified to run the regression with carrier-market fixed effects (`cm`) or carrier-market-structure fixed effects (`cms`).
- Some regressions require that the argument `realQuarter` be specified, indicating whether the analysis should be run with the default alignment of flights and communication (`0`) or with the alternative alignment considered in Appendix E (`1`).
- If the replicator chooses to not use GNU Make to replicate the paper, they are still encouraged to review the commands in `Makefile` if there are any questions as to what arguments should be provided for recreating a specific regression.

List of tables and programs

The code to create tables is located in `tables/code/`, and the output is saved in `tables/output/`. The code to create all figures is located in `figures/code/` and the output is saved to `figures/output/`.

Figure/Table #	Program	Command Line Arguments	Output file
Table 1	create_tab_talk_stats.do		talk-stats.tex
Table 2	create_tab_sum-stats.do		sum-stats_airport.tex
Table 3	n.a. (no data)		
Table 4	create_tab_primary-results.do		tab_airport_primary.tex
Table 5	create_tab_cong-price-results.do	cm	tab_airport_cond-exog_fe-cm.tex
Table 6	create_tab_trans-results.do	cm	tab_airport_cong-price_fe-cm.tex
Table 7	create_tab_cond-exog.do	cm slow weakness domestically internationally stable pace	tab_airport_cond-exog_fe-cm.tex
Figure 1	create_transcript-boxes.py		transcript_status.pdf
Figure 2	create_transcript-boxes.py		talk_manual_capacity_discipline_status.pdf
Figure 3	n.a. (no data)		

Figure/Table #	Program	Command Line Arguments	Output file
Table A.1	create_tab_distance_stats.do		sum-stats_distance.tex
Table A.2	create_tab_cf-first-stage.do	cm	tab_airport_cf-first_fe-cm.tex

Figure/Table #	Program	Command Line Arguments	Output file
Table A.3	create__tab__cf.do	1000 cm	tab__airport__cf__fe-cm.tex
Table B.1	create__tab__sum-stats.do		sum-stats__market-sizes__airport.tex
Table B.2	create__tab__pop-biz-results.do	cm	tab__airport__pop-biz__fe-cm.tex
Table C.3	create__tab__sum-stats.do		sum-stats__city.tex
Table C.4	n.a. (no data)		
Table C.5	create__tab__city-results.do	cm	tab__city__primary__fe-cm.tex
Table D.1	create__tab__ra-results.do	cm	tab__airport__ra__fe-cm.tex
Table E.2	create__tab__realQuarter-results.do	cm	tab__airport__realQuarter__fe-cm.tex
Table F.1	create__tab__20102015-results.do	cm	tab__airport__20102015__fe-cm.tex
Table H.1	create__tab__trans-results.do	cms	tab__airport__trans__fe-cms.tex
Table H.2	create__tab__cond-exog.do	cms	tab__airport__cond-exog__fe-cms.tex
Table H.3	create__tab__pop-biz-results.do	cms	tab__airport__pop-biz__fe-cms.tex
Table H.4	create__tab__city-results.do	cms	tab__city__primary__fe-cms.tex
Table H.5	create__tab__ra-results.do	cms	tab__airport__ra__fe-cms.tex
Table H.6	create__tab__realQuarter-results.do	cms	tab__airport__realQuarter__fe-cms.tex
Table H.7	create__tab__20102015-results.do	cms	tab__airport__20102015__fe-cms.tex
Figure A.1	create__hist__distance.do		hist__distance.pdf, hist__distance__nonzero.pdf
Figure A.2	n.a. (no data)		