README

1. Overview

The code in this replication package reproduces all the results in the paper and Online Appendix for "Rigging the Scores: Corruption through Scoring Rule Manipulation in Public Procurement Auctions." Replication package compiled October 2024.

The R and Stata codes replicate the empirical results in Sections 4-8 of the paper and Sections A-I of the Online Appendix. The R and Stata codes use the empirical data discussed below.

2. Data Availability

2.1. Statement about Rights

We certify that the manuscript's author has legitimate access to and permission to use the data used in this manuscript.

2.2. Raw Data Sources

We list the raw data sources here. We describe where one can download the data. We use this raw data to extract the data needed for the model simulations. We do not provide raw data here, as we do not have the necessary license to do so. The last character of each data is used to represent the data source for 2.3. Detailed description of how to obtain the raw data and how to process them to get each dataset is included in Section 4.

a. Public procurement data: https://gdgpo.czt.gd.gov.cn/cms-gd/site/guangdong/cggg/index.html **[P]**
b. Firm registration data: https://www.qcc.com/ [People outside China need a VPN to get access to it] **[F]**
c. Bid evaluation with expert randomness: https://www.szggzy.com/jygg/list.html?id=zfcg [Retrieved a subsample on 2023-2-23] **[R]**
d. Corruption investigation data from Wang and Dickson 2020: https://dataverse.harvard.edu/api/access/datafile/3831952 **[I]**
e. Other corruption investigation data: provincial website https://www.gdjct.gd.gov.cn/ with every prefecture website from Commission for Discipline Inspection **[I]**
f. County level panel statistics data: https://cnki.nbsti.net/CSYDMirror/trade/Yearbook/Single/N2022040099?z=Z001 **[C]**

2.3. Details of Data

Description	File Name	Location	Provided
R datasets			
Auction results with auction variables **[P]**	scores.Rdata	data/	Yes
Firm registration data of bidders **[F]**	firms.Rdata	data/	Yes
(call-for-tender) CFP variables and results variables together **[P]**	outcome_announce.Rdata	data/	Yes
Bids and auction char indicators for estimations	df.Rdata	data/	Yes

and tests **[P]**			
Expert audit survey data	evaluation.Rdata	data/	Yes
Price bids of bidders **[P]**	price.Rdata	data/	Yes
Region info of auctions **[P]**	region.Rdata	data/	Yes
Evaluation randomness sample **[R]**	random.Rdata	data/	Yes
Firm local/small indicators for asymmetry **[F]**	covariates.Rdata	data/	Yes
STATA datasets			
CFP variables with corruption investigation for all departments **[P] [I]**	announce_cor_full.dta	data/stata	Yes
CFP variables with corruption investigation for departments only ever under investigation **[P] [I]**	announce_cor.dta	data/stata	Yes
Auction outcomes with corruption investigation for all departments **[P] [I]**	nbidders_cor_full.dta	data/stata	Yes
Auction outcomes with corruption investigation for all departments only ever under investigation **[P] [I]**	nbidders_cor.dta	data/stata	Yes
New supplier for the local indicator with corruption investigation for all departments **[F] [I]**	newfirm.dta	data/stata	Yes
Firm characteristics **[F]**	firms.dta	data/stata	Yes
Firm TFP with characteristics **[F]**	dffirm.dta	data/stata	Yes
Linkage of region and county id **[C]**	countyid.dta	data/stata	Yes
County panel variables **[C]**	county_covs.dta	data/stata	Yes
US data: Kang and Miller (2022)			
Us federal public procurement contracts	final_data.dta	data/us	Yes

3. Descriptions of Code

3.1. Main codes

Main	rcode/Main.R
	statacode/main.do
Appendix	rcode/Appendix.R
	statacode/Appendix.do

3.2. Most computational consuming codes

The setting of the computer: 16 cores, 32G RAM, R version 4.4.1 (2024-06-14 ucrt) with platform x86_64-w64-mingw32

Required packages in R:

```
install.package(“pacman”)
pacman::p_load(
  BiocManager, pacman, pkgbuild, remotes, installr, ggplot2, ggpattern, dplyr, data.table,
  haven, np, iterators, itertools, foreach, parallel, doParallel, stats,
  nleqslv, MASS, tictoc, kableExtra, purrr, desc, RStata, msm, mixtools,
  factoextra, REBayes, qvalue, Rmosek, bde, doRNG, RhpcBLASctl, optimx
)
```

Especially, an optimization package “Rmosek” is needed to install manually following the instruction here:

https://docs.mosek.com/latest/rmosek/install-interface.html

Codes	Description & Time
rcode/Section6_2.R	Estimate the Gshat and gshat; Each iteration with 12 cores parallel runs for 20 mins
rcode/Section6_3.R	Auction level mean test; Each iteration with 12 cores parallel runs for 6 hrs
rcode/Section6_4.R	Auction level rank test; Each iteration with 12 cores parallel runs for 6 hrs
rcode/Section6_5.R	Pooled test runs for 15 mins
rcode/Section8_Counter1.R	Counterfactual 1 runs for 15 mins
rcode/Section8_Counter2.R	Counterfactual 2 runs for 15 mins
rcode/AppendixF2.R	Auction level mean test, only one iteration with 12 cores parallel runs for 8 hrs

rcode/AppendixF3.R	Auction level rank test, only one iteration with 12 cores parallel runs for 8 hrs
rcode/AppendixG2.R	Auction level mean test, only one iteration with 12 cores parallel runs for 8 hrs
rcode/AppendixG3.R	Auction level rank test, only one iteration with 12 cores parallel runs for 8 hrs

3.3. Summary of Files that Generate Each Figure or Table

Main Paper

Figure/Table	Codes	Output File
Figure 3	rcode/Figure3.do	graph/Figure3.png
Figure 4(a)	rcode/Figure4.do	graph/Figure4_a.png
Figure 4(b)	rcode/Figure4.do	graph/Figure4_b.png
Table 2	rcode/Table2.R	table/table2.tex
Figure 5 (a)	rcode/Figure5.R	graph/Figure5_1.png
Figure 5 (b)	rcode/Figure5.R	graph/Figure5_2.png
Figure 6 (a)	rcode/Figure6.R	graph/Figure6_1.png
Figure 6 (b)	rcode/Figure6.R	graph/Figure6_2.png
Table 3	rcode/Table3.R	table/table3.tex
Table 4	rcode/Table4.R	table/table4.tex
Figure 8 (a)	statacode/openauction.do	graph/Figure8_a.png
Figure 8 (b)	statacode/nbidders.do	graph/Figure8_b.png
Figure 8 (c)	statacode/corruption.do	graph/Figure8_c.png
Figure 8 (d)	statacode/newfirm.do	graph/Figure8_d.png
Figure 9 (a)	statacode/nbidders.do	graph/Figure9_a.png
Figure 9 (b)	statacode/nbidders.do	graph/Figure9_b.png
Figure 9 (c)	statacode/corruption.do	graph/Figure9_c.png
Figure 9 (d)	statacode/corruption.do	graph/Figure9_d.png
Table 5	rcode/Section8_2.R	table/table5.tex
Figure 10	rcode/Section8_2.R	graph/Figure10.png

Table 6	rcode/Section8_2.R	table/table6.tex

Appendix

Figure/Table	Codes	Output File
Appendix A		
Table A2	rcode/TableA2.R	table/tableA2.tex
Table A4	rcode/TableA4.R	table/tableA4.tex
Figure A2(a)	rcode/AppendixA.R	graph/FigureA2_a.png
Figure A2(b)	rcode/AppendixA.R	graph/FigureA2_b.png
Figure A3(a)	rcode/AppendixA.R	graph/FigureA3_a.png
Figure A3(b)	rcode/AppendixA.R	graph/FigureA3_b.png
Figure A5(a)	statacode/Figure4.do	graph/FigureA5_a.png
Figure A5(b)	statacode/Figure4.do	graph/FigureA5_b.png
Figure A5(c)	statacode/Figure4.do	graph/FigureA5_c.png
Figure A5(d)	statacode/Figure4.do	graph/FigureA5_d.png
Figure A6	statacode/FigureA6.do	graph/FigureA6.png
Figure A7 (a)	statacode/openauction.do	graph/FigureA7_a.png
Figure A7 (b)	statacode/nbidders.do	graph/FigureA7_b.png
Figure A7 (c)	statacode/corruption.do	graph/FigureA7_c.png
Figure A7 (d)	statacode/newfirm.do	graph/FigureA7_d.png
Figure A7 (e)	statacode/FigureA7.do	graph/FigureA7_e.png
Figure A7 (f)	statacode/FigureA7.do	graph/FigureA7_f.png
Figure A8 (a)	statacode/openauction.do	graph/FigureA8_a.png
Figure A8 (b)	statacode/openauction.do	graph/FigureA8_b.png
Figure A9 (a)	rcode/Section8_1.R	graph/FigureA9_a.png
Figure A9 (b)	rcode/Section8_1.R	graph/FigureA9_b.png
Table A5	statacode/Section8.do	table/tableA5.tex
Table A6	statacode/Section8.do	table/tableA6.tex
Figure A10 (a)	statacode/Section8.do	graph/FigureA10_a.png
Figure A10 (b)	statacode/Section8.do	graph/FigureA10_b.png
Figure A11 (a)	statacode/Section8.do	graph/FigureA11_a.png
Figure A11 (b)	statacode/Section8.do	graph/FigureA11_b.png

Figure A12 (a)	statacode/Section8.do	graph/FigureA12_a.png
Figure A12 (b)	statacode/Section8.do	graph/FigureA12_b.png
Figure A13 (a)	statacode/Section8.do	graph/FigureA13_a.png
Figure A13 (b)	statacode/Section8.do	graph/FigureA13_b.png
Table A7	rcode/Section8_2.R	table/tableA7.tex
Table A8	rcode/Section8_2.R	table/tableA8.tex
Appendix B		
Figure B2 (a)	rcode/AppendixB.R	graph/FigureB2_a.png
Figure B2 (b)	rcode/AppendixB.R	graph/FigureB2_b.png
Appendix D		
Figure D1 (a)	rcode/AppendixD1.R	graph/FigureD1_a.png
Figure D1 (b)	rcode/AppendixD1.R	graph/FigureD1_b.png
Figure D1 (c)	rcode/AppendixD1.R	graph/FigureD1_c.png
Figure D1 (d)	rcode/AppendixD1.R	graph/FigureD1_d.png
Figure D2	rcode/AppendixD4.R	graph/FigureD2.png
Figure D3 (a)	rcode/AppendixD4.R	graph/FigureD3_a.png
Figure D3 (b)	rcode/AppendixD5.R	graph/FigureD3_b.png
Figure D4 (a)	rcode/AppendixD1.R	graph/FigureD4_a.png
Figure D4 (b)	rcode/AppendixD1.R	graph/FigureD4_b.png
Figure D5 (a)	rcode/AppendixD1.R	graph/FigureD5_a.png
Figure D5 (b)	rcode/AppendixD1.R	graph/FigureD5_b.png
Figure D6 (a)	rcode/AppendixD2.R	graph/FigureD6_a.png
Figure D6 (b)	rcode/AppendixD2.R	graph/FigureD6_b.png
Figure D6 (c)	rcode/AppendixD3.R	graph/FigureD6_c.png
Figure D6 (d)	rcode/AppendixD3.R	graph/FigureD6_d.png
Figure D7 (a)	rcode/AppendixD6.R	graph/FigureD7_a.png
Figure D7 (b)	rcode/AppendixD6.R	graph/FigureD7_b.png
Figure D8 (a)	rcode/AppendixD6.R	graph/FigureD8_a.png
Figure D8 (b)	rcode/AppendixD6.R	graph/FigureD8_b.png
Figure D8 (c)	rcode/AppendixD6.R	graph/FigureD8_c.png
Figure D8 (d)	rcode/AppendixD6.R	graph/FigureD8_d.png
Figure D9 (a)	rcode/AppendixD7.R	graph/FigureD9_a.png

Figure D9 (b)	rcode/AppendixD7.R	graph/FigureD9_b.png
Figure D10 (a)	rcode/AppendixD7.R	graph/FigureD10_a.png
Figure D10 (b)	rcode/AppendixD8.R	graph/FigureD10_b.png
Figure D11 (a)	rcode/AppendixD7.R	graph/FigureD11_a.png
Figure D11 (b)	rcode/AppendixD8.R	graph/FigureD11_b.png
Table D1	rcode/AppendixD9.R	table/tableD1.tex
Appendix F		
Table F1	Rcode/AppendixF4.R	Table/tableF1.tex
Table F2	Rcode/AppendixF4.R	Table/tableF2.tex
Appendix G		
Table G1	Rcode/AppendixG4.R	Table/tableG1.tex
Table G2	Rcode/AppendixG4.R	Table/tableG2.tex
Figure G2 (a)	rcode/AppendixG1.R	graph/FigureG2_a.png
Figure G2 (b)	rcode/AppendixG1.R	graph/FigureG2_b.png
Figure G2 (c)	rcode/AppendixG1.R	graph/FigureG2_c.png
Appendix H		
Table H1	statacode/FigureH1.do	table/tableH1.tex
Appendix I		
Table I1	statacode/AppendixI.do	table/tableI1.tex
Figure I1 (a)	rcode/AppendixI.R	graph/FigureI1_a.png
Figure I1 (b)	rcode/AppendixI.R	graph/FigureI1_b.png
Figure I2 (a)	rcode/AppendixI.R	graph/FigureI2_a.png
Figure I2 (b)	rcode/AppendixI.R	graph/FigureI2_b.png
Figure I3 (a)	rcode/AppendixI.R	graph/FigureI3_a.png
Figure I3 (b)	rcode/AppendixI.R	graph/FigureI3_b.png

4. Description of Data Scraping and Processing

4.1. Public procurement data [P]

https://gdgpo.czt.gd.gov.cn/cms-gd/site/guangdong/cggg/index.html

Data Retrieval and Source

- The dataset was retrieved from the old version of a government procurement website in **2022**.

- In mid-2022, all data were migrated to a new website. However, procurement data from before 2022 can still be accessed through the **"Old Website Announcement" tab** on the new website, though this requires an **internal login**.

Data Collection and Processing

To obtain and process the data, we scraped two distinct sets of information:

1. **Call-for-Tender Notices**
2. **Contract Award Notices**

Both sets of notices are unstructured and do not come in a standardized dataframe format. All relevant information was extracted through **text analysis techniques**, including the use of **regular expressions**, to parse and retrieve key details from the raw text.

Details of the Datasets

1. **Call-for-Tender Notices**
    - Contains information such as:
        - Buyer's name
        - Budget
        - Publication date
        - Procurement code
        - URL
        - Category
        - Procurement method
        - Individuals responsible for the procurement process
    - These details were systematically extracted from the unstructured text of each notice.
2. **Contract Award Notices**
    - Provides information about the successful bidder for each contract, including:
        - Winner's name
        - Winning price
        - Date of the award
    - Additionally, these notices publish the **scoring and auction results** for each contract, offering insights into the evaluation process.
    - However, the tables containing these results are highly inconsistent, with **thousands of variations in format**.

Data Cleaning and Organization

To address the challenge of inconsistent formats in the auction results:

1. **Saving Result Tables**
    - Used the **unique URL code** associated with each winner award notice to save the result table as a separate CSV file.
    - Each CSV file typically includes:
        - List of bidders
        - Quality scores

        - Price scores
        - Submitted prices
        - Final rankings
2. **Categorizing and Processing Tables**
    - Categorized the result tables into different groups based on their format.
    - For tables with **consistent formats**:
        - Used **Python** to process them systematically.
        - Appended the unique URL code and combined them into a unified dataframe.
    - For tables with **inconsistent formats**:
        - Manually reviewed and corrected the data to ensure accuracy and consistency.

Merging the Two Datasets

The two datasets—call-for-tender notices and contract award notices—were merged using a **multi-step approach**:

1. **Primary Merge Key**: Procurement code (unique identifier for each tender).
2. **Fallback Merge Keys**:
    - If the procurement code was missing, datasets were matched using the **title of the notice** and the **publication time**.
3. **Standardizing Buyer Names**:
    - Many departments underwent name changes or reorganizations over the years.
    - Used **fuzzy matching** to clean and standardize buyer names.
    - Created a unique identifier, **departID2**, to serve as a consistent reference for each buyer across the dataset.

Handling Missing Categories

For the **category field**, where some entries were missing:

1. Identified observations with **non-missing categories**.
2. Applied **fuzzy matching** to the title and project description of notices with missing categories.
3. Assigned the closest matching category from the non-missing data, ensuring consistency and completeness.

Processing Auction Results

To derive meaningful insights from the auction results:

1. **Price Weights**:
    - Determined by the **highest price score** achieved by any bidder in each auction.
2. **Quality Weights**:
    - Calculated as the complement to the price weights:

Quality Weight=100−Price WeightQuality Weight=100−Price Weight

3. **Reconstructing Original Prices**:
    - For cases where original prices were not listed, we used a **two-step calculation**:
        - Step 1: Calculate the **lowest price**:

Lowest Price=Winner Price×Winner Price ScorePrice WeightLowest Price=Price WeightWinner Price ×Winner Price Score

        - Step 2: Calculate the **original prices** for all bidders:

Original Price=Lowest PricePrice Score/Price WeightOriginal Price=Price Score/Price WeightLowest Price

    - This method ensured a complete and consistent representation of bidding dynamics for each auction.

Cleaned datasets:

1. **outcome_announce.Rdata**: Merges the call-for-tender and contract award datasets.
2. **scores.Rdata**: Merges the outcome_announce data with the scoring auction results dataset.
3. **df.Rdata**: Contains the scoring auction results dataset used for estimation.
4. **price.Rdata**: Includes the original price of each bid associated with the scoring auction results.
5. **region.Rdata**: A subset of the outcome_announce dataset focusing on the region of procurement.
6. **random.Rdata [R]**: the small data were manually collected from https://www.szggzy.com/jygg/list.html?id=zfcg [Retrieved a subsample on 2023-2-23] to show cases of the variation in bid evaluation. The dataset is only used in Appendix G.

4.2. Firm Registration Data [F]

1. **Obtaining the Firm List**
    - The firm list was extracted from the **scores.Rdata** dataset, which includes all firm names that participated in public procurement competitions.
    - **Challenges Identified**:
        - There was **no unique firm identifier** associated with the firm names.
        - Firms did not always use the **exact same name** across different records.
        - Firm name changes over time were **not uncommon**, further complicating the matching process.
2. **Linking Firms to the Qichacha Platform**
    - To address these challenges, we linked the firm list to **Qichacha** (https://www.qcc.com/), a comprehensive firm registration record platform in China.
    - Qichacha provides a **batch matching function**, which automatically compares the firm names in our list against its database to identify potential matches.
    - This function helped to:
        - Group together records that likely refer to the same firm, despite variations in naming.
        - Identify firms that had undergone name changes over time.

3. **Retrieving Firm Registration Data**
   - After the batch matching process, we retrieved the **firm registration data** and associated firm information from Qichacha.
   - This data included:
     - Unique firm identifiers.
     - Official registered names.
     - Additional firm details such as registration numbers, addresses, capital size, labor size, firm category, and legal representatives.
4. **Cleaned datasets:**
   1) **firms.Rdata**: Contains firm registration data obtained from Qichacha, including details such as firm names, registration numbers, addresses, and legal representatives.
   2) **covariates.Rdata**: Includes covariates used for Appendix F of the analysis. This dataset incorporates: Firm size classification (small vs. large firms) and geographic classification (local vs. non-local firms).
   3) **firms.dta**: Firm registration data stored in STATA format, making it compatible with STATA-based analysis workflows.
   4) **dffirm.dta**: A merged dataset combining firm registration data from Qichacha with firm Total Factor Productivity (TFP) data obtained from Chen et al.(2021)

4.3. Corruption Investigation Data [I]

1. Data Sources

The corruption investigation data was collected from two primary sources:

1. **Wang and Dickson (2020) Dataset**: 2012-2016
   - Obtained from Harvard Dataverse: Link to Dataset.
   - This dataset provides a comprehensive overview of corruption investigations, including details on investigated officials, their positions, and the outcomes of the investigations.
2. **Provincial and Prefecture Websites**: 2016-2022
   - Data was also collected from official websites of provincial and prefecture-level **Commissions for Discipline Inspection (CDI)**.
   - Example: Guangdong Provincial CDI website: https://www.gdjct.gd.gov.cn/ and each prefectural CDI website linked to the provincial website for all cities
   - These websites publish detailed records of corruption investigations, including the names of investigated officials, their positions, and the reasons for investigation.

2. Data Collection Process

The corruption investigation data was collected through a two-step process:

Step 1: Collecting Investigation Details

- **Level of Investigation**: Recorded the administrative level of the investigated officials (e.g., provincial, municipal, county).

- **Reason for Investigation**: Documented the specific reasons or allegations leading to the investigation (e.g., bribery, embezzlement, abuse of power).
- **Position of Officials**: Captured the official positions held by the individuals under investigation (e.g., mayor, department head, bureau director).
- **Department Affiliation**: Linked the investigated officials to their respective departments using the unique identifier **departID2**, which standardizes department names across datasets.
- **Date**: year and month the investigation happened

Step 2: Data Integration and Standardization

- Data from both sources (Wang and Dickson 2020 and CDI websites) was combined into a unified dataset.
- Inconsistent naming conventions and department affiliations were resolved using **departID2** to ensure consistency across records.
- Duplicate entries and incomplete records were manually reviewed and corrected to ensure data accuracy.
- Reformat the investigation dataset to departID2 level investigation case with year-month

3. Cleaned datasets:

1) **announce_cor_full.dta**:
    - Merges the **call-for-tender notice data** with the **investigation data** using **departID2**.
    - Adds the relative months to the investigation year-month.
    - Includes all **departID2** entries, regardless of whether there was ever an investigation.
2) **announce_cor.dta**:
    - Derived from **announce_cor_full.dta**. Retains only the **departID2** entries that have **at least one corruption investigation case**.
3) **nbidders_cor_full.dta**:
    - Merges the **contract award notice data** with the **investigation data** using **departID2**.
    - Adds the relative months to the investigation year-month.
    - Includes all **departID2** entries, regardless of whether there was ever an investigation.
4) **nbidders_cor.dta**:
    - Derived from **nbidders_cor_full.dta**.
    - Retains only the **departID2** entries that have **at least one corruption investigation case**.
5) **newfirm.dta**:
    - Merges **score.Rdata** and **firm.Rdata** to identify **new participant firms**.
    - Further merges this combined data with the **investigation data** using **departID2**.
    - Adds the relative months to the investigation year-month.

4.4 County level panel statistics data [C]

https://cnki.nbsti.net/CSYDMirror/trade/Yearbook/Single/N2022040099?z=Z001

County-level panel statistics data were directly downloaded from the provided links for each variable, including: GDP, GDP per capita, Fiscal expenditure, Revenue. These variables were then combined into a single dataset for analysis.