

```

1  stop
2  *****
3  * Intro to Data Analysis using Stata / Oscar Torres-Reyna / Fall 2025
4  *****
5  * Set working directory. Using menu go to File → Change working directory
6
7  cd "PATH_TO_DATA_HERE"
8
9  *** Create a log file in the logs folder. For more details type: help log
10
11 log using "mylog.log", replace
12
13 *** Import csv file into Stata
14 * Import CSV files. In menu go to File → Import
15 * Source of data:
16 * https://sports-statistics.com/sports-data/nba-basketball-datasets-csv-files/
17
18 import delimited "2019-20_pbp.csv", clear
19
20 * Check the data, notice the different colors
21 * See slide 13 here: https://www.princeton.edu/~otorres/StataTutorial.pdf
22 browse
23
24 *** Save as Stata. Save the file as Stata files
25 * If you get a 'file not found' message, that is fine since it did not exist before
26 save "2019-20_pbp.dta", replace
27
28 * Create a sequence id (indexing)
29 * See slides 41, 42 here: https://www.princeton.edu/~otorres/StataTutorial.pdf
30 * For more details type: help gen
31 gen sequence = _n
32
33 *** Move variable to the front of the data. For more details type: help order
34 * See also: help move
35 order sequence
36
37 * Saving the data
38 save, replace
39
40 *** Import Excel files. Import Excel files. In menu go to File → Import
41 * The data below comes from the same NBA data, the locations variable
42 * is broken down by its different components and re-attached to the
43 * main data.
44
45 import excel "NBALocations.xlsx", sheet("locations") firstrow clear
46 save "NBALocations.dta", replace
47
48 * The data below was created from the following site:
49 * https://www2.census.gov/geo/pdfs/maps-data/maps/reference/us_regdiv.pdf
50
51 import excel "Regions.xlsx", sheet("Regions and Divisions") firstrow clear
52 save "Regions.dta", replace
53
54 *** Merge files
55 * For more details type: help merge
56 * See https://www.princeton.edu/~otorres/Merge101.pdf
57 * Open main file
58 use "2019-20_pbp.dta", clear
59
60 merge m:1 location using "NBALocations.dta", gen(merge1)
61 tab merge1
62
63 merge m:1 state using "Regions.dta", gen(merge2)
64 tab merge2
65 drop if merge2 == 2
66
67 *** Describe the data
68 describe
69
70

```

```

71  *** Generate a date variable
72  * See https://www.princeton.edu/~otorres/TS101.pdf
73
74  gen datevar = date(date, "MDY")
75  format datevar %td
76  order sequence datevar date
77  sort sequence
78
79  *** Count individual games using egen/group. For more details type: help egen
80  * What identifies a single game?
81
82  egen gameid = group(datevar awayteam hometeam)
83  sum gameid
84  order sequence gameid
85
86  *** Frequencies - descriptive statistics for categorical variables
87  * For more details type: help tab
88
89  tab nameregion
90  tab nameregion, missing
91  describe nameregion
92  browse if nameregion == ""
93
94  *** Replace values within variables. For more details type: help replace
95
96  replace nameregion = "Foreign" if nameregion == ""
97  tab nameregion, missing
98  tab venue
99  tab state#
100 tab city
101
102 *** Cross-tabulation
103 * See slides 25-27 here: https://www.princeton.edu/~otorres/StataTutorial.pdf
104 tab shotttype shotoutcome, col row cell chi2
105
106 *** What if we want to know the % of games per location?
107 * Need to identify when a game ends. For more details type: help string functions
108
109 sort sequence
110 browse datevar gameid quarter awayplay homeplay
111
112 gen end_of = strpos(awayplay, "End of")
113 tab end_of
114 tab awayplay if end_of == 1
115 tab location if awayplay == "End of Game"
116
117 browse date location end_of awayplay if awayplay == "End of Game"
118
119 * For more on conditionals:
120 * See slide 12 here: https://www.princeton.edu/~otorres/StataTutorial.pdf
121
122 *** Time string to time variable
123 * See https://www.stata.com/manuals13/ddatetime.pdf#ddatetime
124
125 gen double timevar = clock(time, "hm")
126 format timevar %tcHH:MM
127 tab timevar
128 order sequence gameid datevar date timevar time
129
130 tab timevar if awayplay == "End of Game"
131 tab location if timevar == clock("12:00", "hm")
132 tab location awayteam if timevar == clock("12:00", "hm") & awayplay == "End of Game"
133
134 *** Revert the seconds left variable to seconds play
135
136 gen seconds = 720 - secleft
137 label variable seconds "Play-by-play time per quarter (seconds)"
138 move seconds secleft
139 sort sequence
140 browse datevar gameid seconds secleft awayplay homeplay

```

```

141 *** Generate a dummy variable to identify games that were over time
142
143 tab quarter if awayplay == "End of Game"
144 gen overtime1 = (quarter > 4) if !missing(quarter)
145 tab quarter overtime1
146
147 * Adds a 1 to all plays of games that went overtime
148 egen overtime2 = max(overtime1), by(gameid)
149 browse datevar gameid quarter awayplay quarter overtime1 overtime2
150
151 * FYI: Same as above but it re-sorts the datafile
152 bysort gameid: egen overtime3 = max(overtime1)
153
154 * Keep only 4 quarter games
155 drop if overtime2 == 1
156
157 *** Show average away score per quarter
158 * Using sum. For more details type: help sum
159
160 bysort quarter: sum awayscore
161
162 *** Using tabstat. For more details type: help tabstat
163
164 tabstat awayscore, s(mean median sd min max range cv) by(quarter)
165
166 *** Using table statistics in rows, quarters in columns
167 * For more details type: help table. See also: help table##stat
168 * See slide 19-27 here: https://www.princeton.edu/~otorres/Outreg2.pdf
169
170 table () (quarter), statistic(mean awayscore) ///
171         statistic(sd awayscore) ///
172         statistic(cv awayscore) ///
173         nformat(%8.2f mean sd median min max range cv) ///
174         name(table1) replace
175
176 collect set table1
177 collect style cell result[sd min range], shading(background(lightblue))
178 collect title "Table XX. Away score. Descriptive statistics by quarter"
179 collect export "table1.docx", replace
180
181 * Using table statistics in columns, quarters in rows
182 table (quarter) (), statistic(mean awayscore) ///
183         statistic(sd awayscore) ///
184         statistic(cv awayscore) ///
185         nformat(%8.2f mean sd median min max range cv) ///
186         name(table1a) replace
187
188 collect set table1a
189 collect style cell quarter[2 4], shading(background(lightblue))
190 collect title "Table XX. Away score. Descriptive statistics by quarter"
191 collect export "table1a.docx", replace
192
193 *** Generate a variable with the average away score by time, seconds, and quarter
194 * For more details type: help egen
195
196 egen awayscore1 = mean(awayscore), by(timevar quarter seconds)
197
198 *** Scatterplots with line fit
199
200 twoway scatter awayscore1 seconds if quarter == 1
201
202 twoway scatter awayscore1 seconds if quarter == 1 || ///
203         lfit awayscore1 seconds if quarter == 1 || ///
204         scatter awayscore1 seconds if quarter == 2 || ///
205         lfit awayscore1 seconds if quarter == 2, ///
206         title("Awaycore by time play per quarter") ///
207         ytitle("Awaycore") xtitle("Seconds") ///
208         legend(label(1 "Q1") label(2 "LfitQ1") label(3 "Q2") label(4 "LfitQ2") pos(6) rows(1))
209
210 graph export "figure1.png", replace

```

```

211
212 *** Install binscatter
213 * For more details: help binscatter
214 ssc install binscatter
215
216 binscatter awayscore seconds, by(quarter) ///
217         legend(label(1 "Q1") label(2 "Q2") label(3 "Q3") label(4 "Q4") pos(6) rows(1))
218
219 graph export "output\figure2.png", replace
220
221 *** Is the average score between home and away teams the same?
222
223 * ttest is used to compare two means. For more details type: help ttest
224 * For a nice table see slides 29-34 here: https://www.princeton.edu/~otorres/Outreg2.pdf
225
226 ttest awayscore = homescore
227 ttest awayscore = homescore if quarter == 1 & awayplay == "End of 1st quarter"
228 ttest awayscore = homescore if quarter == 2 & awayplay == "End of 2nd quarter"
229 ttest awayscore = homescore if quarter == 3 & awayplay == "End of 3rd quarter"
230 ttest awayscore = homescore if quarter == 4 & awayplay == "End of 4th quarter"
231 ttest awayscore = homescore if awayplay == "End of Game"
232
233 *** Creating a dummy variable to identify games after 8 pm
234
235 gen evening = (timevar >= clock("20:00", "hm")) if !missing(timevar)
236 tab timevar evening
237 ttest awayscore if awayplay == "End of Game", by(evening)
238 ttest homescore if awayplay == "End of Game", by(evening)
239
240 *** ANOVA is used to compare more than two means. For more details type: tab oneway
241 * For more details type: tab encode
242
243 oneway awayscore nameregion if awayplay == "End of Game", bonferroni
244
245 *** Regression
246
247 * Do committing more fouls help higher score? Create a dummy to identify any type of foul
248 gen foul = (foultype != "")
249
250 * Cumulative fouls per game/quarter
251
252 bysort gameid quarter (sequence): gen foulnumqtr = sum(foul)
253 label variable foulnumqtr "Cumulative fouls per quarter"
254
255 reg awayscore foulnumqtr, robust
256 reg awayscore foulnumqtr seconds, robust
257
258 * Adding quarter fixed effects
259
260 reg awayscore foulnumqtr seconds i.quarter, robust
261
262
263 * Adding region fixed effects
264
265 encode nameregion, gen(nameregion1)
266 label variable nameregion1 "Census regions"
267
268 reg awayscore foulnumqtr seconds i.quarter i.nameregion1, robust
269
270 *** Nice regression tables using outreg2
271 * After install it, type: help outreg2
272 * See https://www.princeton.edu/~otorres/Outreg2.pdf
273 ssc install outreg2
274
275 reg awayscore foulnumqtr seconds i.quarter i.nameregion1, robust
276 outreg2 using "myregs.doc", replace label ctitle(Awayscore)
277
278 reg homescore foulnumqtr seconds i.quarter i.nameregion1, robust
279 outreg2 using "myregs.doc", append label ctitle(Homescore)
280

```

```

281  *** USING PRESERVE / RESTORE
282
283  *** Estimate the number of games per month and compare the awayscore vs the homescore
284  *** Step 1: create monthly variable
285
286  gen monthly = mofd(datevar)
287  format monthly %tm
288
289  *** Step 2: keep only the end of each game per month
290  * NOTE: when working with string variables, spelling matter. Same with variable names.
291
292  tab awayplay if end_of == 1
293
294  *** NOTE: MUST SELECT FROM PRESERVE TO RESTORE AND RUN THE CODE ALL AT ONCE.
295
296  preserve
297
298  keep if awayplay == "End of Game"
299
300  * Collapse the data by monthly
301  * For more details type: help collapse
302
303  collapse (mean) awayscore ///
304           (mean) homescore ///
305           (count) games = gameid, by(monthly)
306
307  twoway bar games monthly, ytitle("Num. of games") ///
308         color(green%33) || ///
309         scatter games monthly, mlabel(games) msymbol(none) mlabpos(12) || ///
310         line awayscore homescore monthly, yaxis(2) ///
311         lcolor(black red) lwidth(thick thick) lpattern(solid dash) ///
312         title("Away and home scores 2019-2020") ///
313         ytitle("Average score", axis(2)) xtitle(Month) ///
314         tlabel(2019m10(1)2020m10, format(%tmM,_CY) angle(45) labsize(2.5)) ///
315         legend(order(1 "Num. of games" 3 "Awayscore" 4 "Homescore") pos(6) rows(1))
316
317  graph export "figure3.png", replace
318
319  restore
320

```