An anonymised five-year learner-level dataset for 2017 to 2021

15 December 2022[1]

Contents

1 Introduction

The dataset described in the current document consists of the following three tables, all Stata version 16 files, with the first one containing labelling:

Table 1: Details on the two tables

File name	Size in KB	Variables	Observations
learner-2017-2021-v1	2,247,764	12	65,762,826
school-2017-2021-v1	497	11	26,353

The file names indicate what each observation represents, and the version. The file sizes are those seen after the anonymisation process. The two tables are linked by the variable *emiscode*, which is unique in the data file *school-2017-2021-v1*.

The purpose of the dataset is largely to facilitate analysis of learner movements through the system over the five-year period 2017 to 2021, a period which saw major disruptions due to the pandemic. However, the dataset would also be of use in examining the evolution of class size, a matter which has become increasingly important in the context of serious budget constraints and an increase in the population of school-going age.

2 The original source data

The raw data for the years 2018 to 2021 were compiled by the DBE's EMIS directorate in mid-2022. The raw data for 2017 were compiled in 2018 in response to a need for an earlier analysis. Moreover, EC data from a separate channel were used for 2020, as the national 2020 raw data received did not include this province. The source for the 2020 EC data is described in Department of Basic Education (2021). WC 2018 data were also obtained separately, in late 2022, because this was missing in the original 'national' dataset.

For the data file *school*-2017-2021-v1, various national master lists available on the DBE website were used. The details are as follows:

[1] Produced by Martin Gustafsson (mgustafsson@sun.ac.za) for the Department of Basic Education.

2017	[National database *Operational_ Ordinary School.xlsx* describing 2017 schools and downloaded in 2018. File no longer available on the DBE website, can be obtained on request from the author of the current report.]
2018	Title on DBE website: 'Quarter 3 of 2018: December 2018'
2019	Title on DBE website: 'Quarter 3 of 2019: December 2019'
2020	Title on DBE website: 'Quarter 1 of 2020: June 2020'
2021	Title on DBE website: 'Quarter 1 of 2021: June 2021'

Note that the only details extracted from the schools master lists were whether a school was public or independent, and only details for ordinary schools were extracted. There are many further school-level details in the master lists. However, these further details have not been pulled into the dataset described here, in part because that would involve decisions around what is needed, and how to standardise values. Users of the dataset wanting further details on schools, such as the school quintile, should obtain that from the original DBE schools master lists, using the school identifier *emiscode* for linking to the learner-level dataset.

3 Merging and normalisation occurring before anonymisation

Table 2 below compares learners in the raw data to officially reported learners. The raw data totals are what is left after a tiny number of learners, one to five per year, was dropped as the learners had no *emiscode* value. 'Other' in the table refers to learners who are either not in grades R to 12, or who are not in an ordinary school. Much of 'Other' would be accounted for by special school learners, who came officially to 119,403 in the last published *Education statistics in South Africa*, of 2016. The province-specific 'Official' totals in the table, which cover only ordinary schools (public and independent) and grades R to 12, are from the abridged *School realities* series.

Across all years, the gap between official totals and the data, either with respect to grades R to 12 in ordinary ('Total'), or with respect to the grand total ('Final'), is fairly close to a zero percentage[2]. In interpreting the table, what should be taken into account is that some learners from the data would be incorrectly entered under 'Other' because their grade is not specified. And the latter could occur because values appear in the wrong column, due to problems with column separators in the original text files. There was no attempt to fix this problem, in part because it is limited to relatively few observations, and in part because there appears to be no easy solution. A key problem appears to be the presence of the hashtag character "#" within the variable *accessionno*, given that this character is also the column separator in the raw data.

The matter of values appearing in the wrong column is discussed further below in relation to missing gender values.

[2] The 2019 'Total' and 'Official' value deviates from the published total for grades 1 to 12 plus the published total for Grade R, though the two should agree exactly. The publication under-states enrolment by around 20,000. This is due to incorrect aggregation for certain provinces and grades in the 2019 publication. For 'Total' and 'Official' in Table 2 below, the underlying grade-specific values in the publication were aggregated. This problems appears to be unique to the 2019 publication.

Table 2: Official totals versus totals in the data

	2017			2018			2019			2020			2021		
	Official	Data	%	Official	Data	%	Official	Data	%	Official	Data	%	Official	Data	%
EC	1,792,264	1,792,196	0.00	1,840,780	1,827,777	-0.71	1,852,225	1,841,410	-0.58	1,840,999	1,840,178	-0.04	1,846,066	1,843,899	-0.12
FS	699,742	697,940	-0.26	705,471	668,457	-5.25	707,891	714,420	0.92	718,209	715,808	-0.33	724,986	724,919	-0.01
GP	2,402,299	2,340,388	-2.58	2,396,305	2,383,755	-0.52	2,457,915	2,439,911	-0.73	2,499,805	2,491,159	-0.35	2,557,582	2,548,147	-0.37
KN	2,858,712	2,824,696	-1.19	2,818,142	2,772,252	-1.63	2,839,121	2,841,079	0.07	2,864,128	2,857,177	-0.24	2,890,827	2,883,355	-0.26
LP	1,775,252	1,759,670	-0.88	1,723,639	1,723,022	-0.04	1,752,839	1,752,839	0.00	1,758,169	1,751,337	-0.39	1,798,038	1,796,310	-0.10
MP	1,095,951	1,074,925	-1.92	1,045,667	1,041,621	-0.39	1,094,607	1,094,607	0.00	1,107,135	1,106,111	-0.09	1,134,299	1,126,906	-0.65
NC	291,979	290,236	-0.60	294,880	294,250	-0.21	298,093	298,093	0.00	303,277	302,852	-0.14	303,777	302,823	-0.31
NW	825,426	811,047	-1.74	840,402	838,427	-0.24	852,375	852,375	0.00	862,613	858,717	-0.45	872,189	868,726	-0.40
WC	1,125,277	1,125,277	0.00	1,141,057	1,141,057	0.00	1,185,542	1,185,623	0.01	1,239,438	1,235,851	-0.29	1,261,066	1,261,064	0.00
Total	12,866,902	12,716,375	-1.17	12,806,343	12,690,618	-0.90	13,040,608	13,020,357	-0.16	13,193,773	13,159,190	-0.26	13,388,830	13,356,149	-0.24
Other		175,898			128,963			156,463			162,297			196,516	
Final	12,892,273	12,892,273	0.00	12,819,542	12,819,581	0.00	13,041,198	13,176,820	1.04	13,216,472	13,321,487	0.79	13,409,249	13,552,665	1.07

The data file *school-2017-2021-v1* contains only the *emiscode* values of ordinary schools, not for instance special schools. This means that there are *emiscode* values in the table *learner* which would not be found in the table *school.* However, all *emiscode* values in *school* are found somewhere in *learner*, by design. Any *emiscode* value seen in the downloaded schools master lists which did not appear in the learner-level data resulted in the school record being dropped.

The pre-anonymisation process of normalisation followed a few principles, keeping in mind that an aim was to enhance the linking of the same learners across years. First, all lower case alpha characters became upper case, so “Thabo” became “THABO”. Standardisation within and across years with respect to descriptors was pursued. For instance, differences such as having “1” and “2” for the two genders and “Male” and “Female” were dealt with. Normalisation also involved looking out for strange patterns. This led to the discovery that in *race* the values for Indian and coloured learners had clearly been switched in the case of WC in 2020 and 2021. This was fixed. It is not impossible that other such strange patterns were missed, and the analyst should be mindful of this.

Table 3 below refers to the 11 variables of the *pre*-anonymised data, and where variables are missing. The 2017 and 2019 data have neither *class* nor *race*, as these were missing in the source data. In the case of 2020, *accessionno* and *class* are missing from the EC data, which has a different source. The same can be said of *class* and *race* in 2018 with respect to WC.

***Table 3: Variables present per year in table* learner**

	2017	2018	2019	2020	2021
year	•	•	•	•	•
emiscode	•	•	•	•	•
idno	•	•	•	•	•
accessionno	•	•	•	(•)	•
birthdate	•	•	•	•	•
fname	•	•	•	•	•
sname	•	•	•	•	•
grade	•	•	•	•	•
class		(•)		(•)	•
gender	•	•	•	•	•
race		(•)		•	•

The variables *emiscode* and *year* can never have missing values, but all other variables can. Table 4 and Table 5 display the extent of this. Missing *idno* values are clearly a problem, as the national ID number is perhaps the most reliable unique identifier of learners. Table 5 indicates that this problem is concentrated in GP. Missing *birthdate* values in 2017 are a problem, for instance in terms of determining the ages of learners (but also because this variable can assist in the linking process). Missing *birthdate* largely overlaps with missing *idno* in 2017, leaving little space for imputing the date using the national ID number. Many values appearing in the two tables below as zero are not truly zero. For instance, in Table 4 the gender value for 2021 is 0.0007%, implying around 90 learners with gender missing. It appears virtually all of this is due to the problem referred to previously of values in the incorrect column. Missing *grade* is not necessarily problematic as this could point to a learner being in a grade other than grades R to 12, for instance pre-Grade R. Any grade not in the range of 0 (Grade 12) to 12 was coded as 99 in the normalisation process, so missing grade in the following two tables means 99.

Table 4: Percentage missing by year

	2017	2018	2019	2020	2021
idno	13.3	8.5	7.5	6.6	6.8
accessionno	0.2	0.0	0.0	13.9	0.0
birthdate	13.4	0.0	0.0	0.0	0.0
fname	0.0	0.0	0.0	0.5	0.0
sname	0.0	0.0	0.0	0.0	0.0
grade	0.2	0.2	0.3	0.3	0.3
class	100.0	8.9	100.0	13.9	0.0
gender	0.0	0.0	0.0	0.0	0.0
race	100.0	9.0	100.0	0.0	0.0

Table 5: Percentage missing by province

	EC	FS	GP	KN	LP	MP	NC	NW	WC
idno	3.6	5.8	18.9	8.0	3.7	9.4	1.9	6.7	6.4
accessionno	20.1	0.0	0.0	0.2	0.0	0.0	0.0	0.0	0.0
birthdate	0.8	1.2	5.1	3.2	1.1	3.3	0.4	2.6	2.1
fname	0.7	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
sname	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
grade	0.2	0.3	0.5	0.2	0.1	0.1	0.2	0.3	0.3
class	59.8	40.0	39.2	40.0	40.0	40.0	39.5	39.4	57.7
gender	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
race	39.7	40.0	39.3	40.0	40.0	40.0	39.5	39.4	57.7

4 How the anonymisation occurred

Five variables with personal information had their values anonymised: *idno, accessionno, birthdate, fname* and *sname.* In each case, the anonymised variable takes on the suffix *_anon*, creating for instance *fname_anon.*

Anonymisation involved distributing, on a random basis, sequential integer numbers starting with 1 to each personal value[3]. This occurred separately for each of the five anonymised variables. The maximum replacement value for each anonymised variable would equal the number of non-missing unique values in the original variable.

In the case of *idno*, missing leading zeros were dealt with, before the anonymisation, as follows. Any *idno* value with a length of 10 characters had "000" added to the beginning. Any *idno* with a length of 11 characters acquired "00" at the beginning and a length of 12 resulted in the addition of "0" at the beginning.

For the first name and surname variables, *fname* and *sname*, irregularities within the values were removed before anonymisation. All characters were made upper case, any characters not in the range "A" to "Z" were removed, meaning blanks were removed, and the maximum length was made to be 15 characters. After these adjustments, anonymisation occurred.

Before birthdate was removed, a new variable *birthyear* was constructed. To illustrate, the *birthyear* value would be 2011.0 for someone born in the first half of 2011, and 2011.5 for someone born in 2011, but from 1 July onwards. This specification of half years is important given that the age of admission policies use half years.

[3] One discovery regarding Stata made during this process is that in coding such as 'gen newid = _n' Stata does not automatically generate long when float is not sufficient for all values (though Stata does automatically generate float when int is not sufficient!). The number of unique learners across all years is a little over 17 million, which exceeds Stata's float limit of 16,777,216. A command such as 'gen long newid = _n' thus had to be used.

Table 6 indicates what percentage of learners would be uniquely identified. Note that in this process only identifiers which are not missing were used to identify learners. Identification is clearly worse for 2017 than for the subsequent years. The national ID number identifies just over 90% of learners in 2018 to 2021 uniquely. The variable *accessionno* is clearly not a very effective unique identifier. The combination of identifiers in column D covers around 100% of learners in 2018 to 2021. In the case of 2017, unique identification rises to 90% if three different identification approaches are employed: one using *idno*, another *accessionno*, and a third the combination of four variables appearing in column D. This can be seen in the last column of the table Note this is not the same as identifying learners using the six variables simultaneously. To illustrate, if a learner has a unique ID number, but has first name missing, then the learner would not be considered to have a unique identification if all six variables are combined, given the approach of discarding learners with missing identifier values.

Table 6: Percentage of learners with unique identification per year

	A	B	C	D	E	A or D	A, B or D
idno	●						
accessionno		●					
birthdate			●	●	●		
fname			●	●			
sname			●	●	●		
gender			●	●	●		
race			●				
2017	84	35		85	62	85	90
2018	91	29	90	99	70	100	100
2019	92	37		99	70	100	100
2020	93	33	99	99	70	100	100
2021	92	36	99	99	69	99	100

It is clear from Table 6 that first name, *fname*, is important. Without it, unique identification declines considerably – compare columns D and E.

Table 7 indicates the extent of across-year linking that is possible. For instance, if in 2018 only learners with a unique non-missing *idno* value are considered, and if the same is done for 2019, then 74% of 2018 learners can be found in the next year's data using a common *idno* value. The value would never be 100% because learners leave school, especially at the secondary level and Grade 12. However, if learners in grades 1 to 6 in 2018 are considered, the degree of linkage to 2019 can be expected to be higher: in fact it is 80% using *idno*, according to Table 7. For grades 1 to 6 learners, and in the case of 2018 to 2020, around 97% of learners from the one year can be found in the next year, if a multi-pronged approach is used of A, then D, then B. This means first linkages using *idno* are explored, and linked learners are removed. The remaining learners are then linked using the four variables behind D: *birthdate, fname, sname* and *gender.* After this approach has been exhausted, finally linking of unlinked learners using *accessionno* is attempted.

Importantly, virtually all pre-anonymised *idno* values in WC for 2018 have an alphabetic letter in position 7 and 8 of the 13-digit identifier. Why this is the case is not known, and this pattern is not seen in other years. This means linking via this variable for WC between 2017 and 2018 and between 2018 and 2019 is not possible.

Table 7: Linking to next year

	A	B	D	A then D	A then D then B
All					
2017	67	14	61	75	76
2018	74	15	84	89	89
2019	84	26	85	90	90
2020	85	26	85	90	90
Only grades 1 to 6					
2017	73	13	67	82	83
2018	80	15	92	96	96
2019	90	27	93	96	97
2020	90	27	93	96	97

Degrees of linking are slightly lower than what is shown in Table 7 if links displaying inconsistent grades are excluded. A learner must stay in the same grade or move to the next grade. But such exclusions make a small difference. For instance, in the final column of Table 7, under 'All', the values would decline by around one percentage point if grade consistency were taken into account.

The following four graphs provide a picture of the degree of linking by year, province and grade if the 'A then D then B' approach is employed. Here only linking to a valid grade in the next year – the same grade or one grade up – is considered a successful link. GP consistently displays the worst linking in all four graphs. The dip in Grade 7 across all graphs is largely the result of inconsistent data management practices across separate primary and secondary schools. The far worse degree of linking seen for 2017 to 2018 is visible in Figure 1. For the other three years, patterns are fairly consistent.

Importantly, the meaning of the grade-specific values seen in the graph can easily be misinterpreted. To illustrate, what Figure 2 does *not* suggest is that around 90% of learners reach Grade 12. This 90% suggests instead that 90% of Grade 11 learners move to Grade 12 in the next year, but those Grade 11 learners were already a reduced group of learners as around 10% of Grade 10 learners did not make it to Grade 11, and so on. The aim of the graphs presented here is to explain the extent of linking across years, not to illustrate levels of participation.

Figure 1: Degrees of linking by single grade and province for 2017

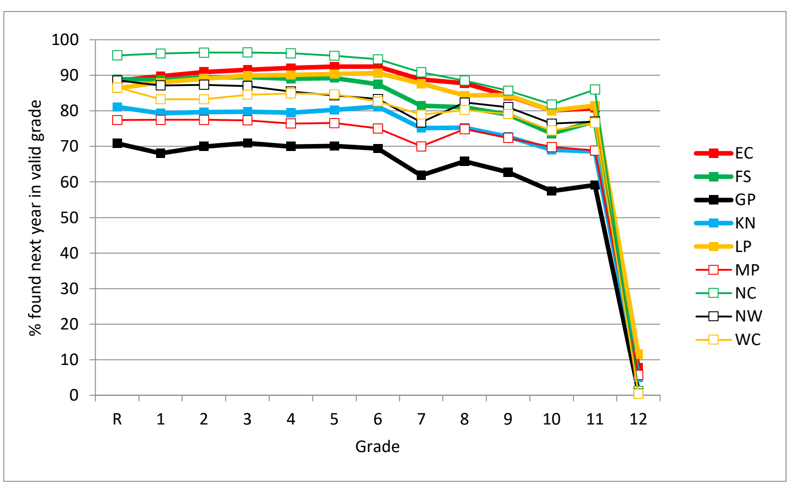

Figure 2: Degrees of linking by single grade and province for 2018

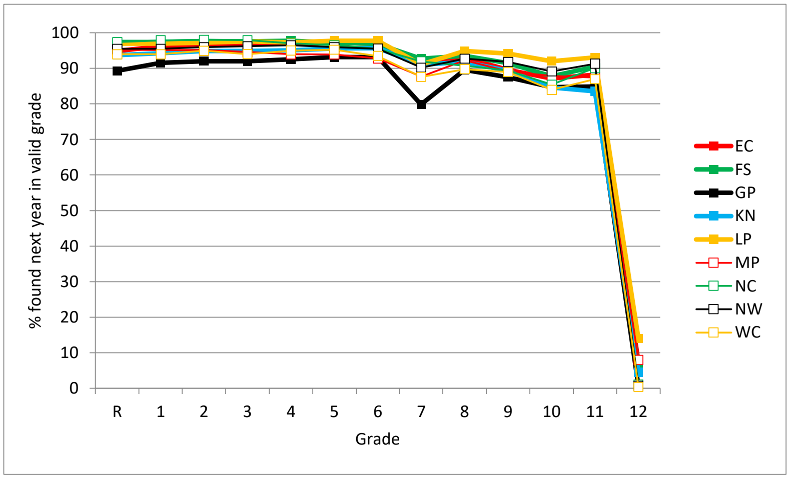

Figure 3: Degrees of linking by single grade and province for 2019

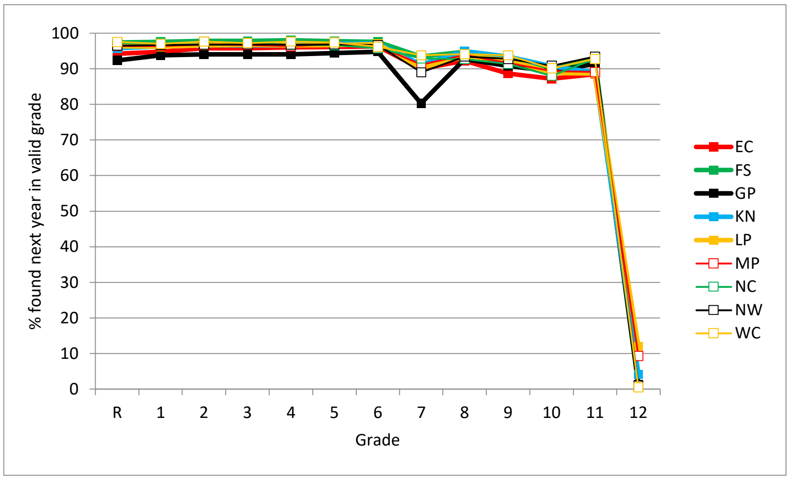

Figure 4: Degrees of linking by single grade and province for 2020

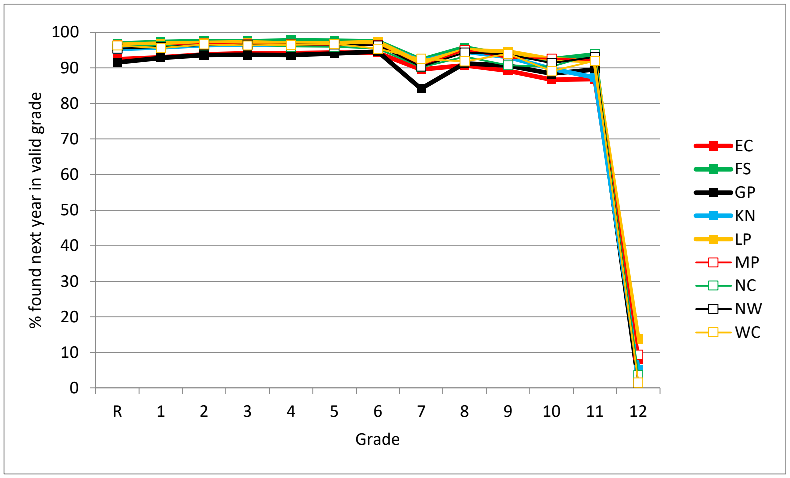

Table 8 below indicates what percentage of learners in one year could be linked to a record in the next year, using 'A then D then B', though the grade was *not* right. The 0.6% of learners for 2018 represents 92,675 learners. Of these, 72% were linked using the ID number (option A), 14% were linked using the combination of birth date, both names and gender (D), and 14% were linked using the accession number (B). Of the 66,989 linked using the ID number, virtually all were in one of three somewhat plausible categories: 18% had a missing or irregular grade in one year, 55% had apparently moved up by two grades, and 26% had moved back by one grade. There were virtually no across-grade movements exceeding two. The fact that these linked learners displayed relatively consistent grade patterns strengthens the the evidence considerably that the learners in question are *not* accidentally linked. It is possible that some schools did, for instance, promote learners by two grades on academic grounds, even though policy discourages this. Of the 13,138 learners linked using the combination of four variables, 17% displayed missing or irregular grades, 50% displayed a move up two grades, and 13% displayed a move down one grade. Here 20% of linked learners displayed other, and more erratic, grade movements, suggesting that some links could be accidental, in other words links across two individuals who are not the same. The overall finding is encouraging, however. But 20% of 13,138 represents just 0.02% of all 2018 learners, so this is a tiny problem that would not affect analysis in any substantive way. If linked learners with non-standard grades are found to be mostly legitimate, this also strengthens the evidence that the entire linking process is valid.

Table 8: Percentage of learners linked but to strange grade

	2017	2018	2019	2020
EC	0.8	0.4	0.4	1.0
FS	0.4	0.3	0.3	0.4
GP	1.2	1.0	0.6	1.0
KN	1.3	1.1	0.7	0.8
LP	0.8	0.3	0.6	0.8
MP	2.1	0.4	0.5	1.0
NC	0.3	0.3	0.5	0.5
NW	0.3	0.4	0.3	0.6
WC	0.1	0.4	0.6	0.8
SA	1.0	0.6	0.5	0.8

References

Department of Basic Education (2021). *Impacts of the COVID-19 pandemic on school enrolments*. Pretoria.

Appendix 1: Stata code used for the normalisation and anonymisation

```
* PUTTING TOGETHER A SINGLE NORMALISED FILE, WITH EC 2020 PATCH AND WC 2018
PATCH
* The aim here is to produce a standardised NON-anonymised dataset. At this point, ID numbers are on
the whole left left-truncated. This is fixed in the anonymisation process.

* 2018
import excel "C:\My Documents\Resources (Data)\Department of Education\Databases downloaded off
DoE website\Schools list for 2018 09 downloaded 2022 09\2018 Masterlist Ordinary schools National
Masterlist.xlsx", sheet("National") firstrow clear
replace Sector = upper(Sector)
replace Sector = substr(Sector, 1, 1)
rename Type_DoE Type
replace Type = substr(Type, 1, 1)
rename NatEmis emiscode
keep emiscode Sector Type
rename Sector Sector18
rename Type Type18
save "C:\My Documents\Numbercrunching\LURITS\temp0.dta", replace
import delimited "D:\Lurits\2018\Learners.txt", delimiter("#") bindquote(nobind) clear
drop if _n==_N | _n==_N - 1 // 2 - end-of-file markers
rename importyear year
rename ïemiscode emiscode
drop if emiscode<"000000000" | emiscode>"999999999" // 1
replace idno = subinstr(idno, " " , "", .)
replace idno = upper(idno)
replace idno = "" if idno=="NULL"
replace idno = substr(idno, 1, 13)
replace accessionno = subinstr(accessionno, " ", "", .)
replace accessionno = upper(accessionno)
foreach n of varlist fname sname {
  replace `n' = subinstr(`n', " ", "", .)
  replace `n' = substr(`n', 1, 20)
  replace `n' = upper(`n')
  rename `n' temp
  gen `n' = ""
  gen templength = length(temp)
  quietly summ templength
  quietly forvalues j = 1 / `r(max)' {
    replace `n' = `n' + substr(temp, `j', 1) if inrange(substr(temp, `j', 1), "A", "Z")
            noisily display `j'
  }
  replace `n' = substr(`n', 1, 15)
  drop temp*
}
codebook grade, tab(50)
destring grade, force replace
replace grade = 99 if grade<0 | grade>12
replace gender = upper(gender)
codebook gender, tab(50)
replace gender = cond(gender=="FEMALE", "F", cond(gender=="MALE", "M", ""))
replace race = upper(race)
replace race = subinstr(race, " ", "", .)
codebook race, tab(500)
rename race temp
gen race = "A" if substr(temp, 1, 7)=="AFRICAN"
replace race = "C" if substr(temp, 1, 8)=="COLOURED"
replace race = "I" if substr(temp, 1, 6)=="INDIAN" | substr(temp, 1, 5)=="ASIAN"
```

```
replace race = "W" if substr(temp, 1, 5)=="WHITE"
replace race = "O" if substr(temp, 1, 5)=="OTHER"
drop temp
codebook race, tab(50)
keep year emiscode idno accessionno birthdate fname sname grade class gender race
order year emiscode idno accessionno birthdate fname sname grade class gender race
compress
save "C:\My Documents\Numbercrunching\LURITS\temp3.dta", replace
import delimited "D:\Lurits\2018\2018_WC.txt", delimiter(",") clear
rename v2 year
rename v1 emiscode
tostring emiscode, replace
drop if emiscode<"100000000" | emiscode>"199999999" // 0
rename v7 idno
replace idno = subinstr(idno, " " , "", .)
replace idno = upper(idno)
replace idno = "" if idno=="NULL"
replace idno = substr(idno, 1, 13)
rename v10 accessionno
format accessionno %13.0f
tostring accessionno, replace force format(%13.0f)
replace accessionno = subinstr(accessionno, " ", "", .)
replace accessionno = upper(accessionno)
rename v8 fname
rename v9 sname
foreach n of varlist fname sname {
  replace `n' = subinstr(`n', " ", "", .)
  replace `n' = substr(`n', 1, 20)
  replace `n' = upper(`n')
  rename `n' temp
  gen `n' = ""
  gen templength = length(temp)
  quietly summ templength
  quietly forvalues j = 1 / `r(max)' {
    replace `n' = `n' + substr(temp, `j', 1) if inrange(substr(temp, `j', 1), "A", "Z")
          noisily display `j'
  }
  replace `n' = substr(`n', 1, 15)
  drop temp*
}
rename v11 grade
replace grade = subinstr(grade, "Grade ", "", .)
replace grade = subinstr(grade, "R", "0", .)
destring grade, force replace
replace grade = 99 if grade<0 | grade>12
rename v12 gender
replace gender = upper(gender)
codebook gender, tab(50)
replace gender = cond(gender=="FEMALE", "F", cond(gender=="MALE", "M", ""))
gen temp = length(idno)
codebook temp, tab(50) // All 13
gen tempy = substr(idno, 1, 2)
destring tempy, force gen(tempynum)
gen tempyyyy = cond(tempynum<=18, "20", "19") + tempy
gen tempbirth = tempyyyy + substr(idno, 3, 4)
destring tempbirth, gen(birthdate) force
drop temp*
keep year emiscode idno accessionno birthdate fname sname grade gender
order year emiscode idno accessionno birthdate fname sname grade gender
compress
append using "C:\My Documents\Numbercrunching\LURITS\temp3.dta"
save "C:\My Documents\Numbercrunching\LURITS\temp3.dta", replace
use "C:\My Documents\Numbercrunching\LURITS\temp3.dta", clear
gen ones = 1
collapse (sum) ones, by(emiscode grade)
merge m:1 emiscode using "C:\My Documents\Numbercrunching\LURITS\temp0.dta"
drop if _merge==2
```

```
drop _merge
gen statssaprov = substr(emiscode, 1, 1)
destring statssaprov, replace
merge m:1 statssaprov using "C:\My Documents\Numbercrunching\Tools\provinces.dta"
gen core = ones if grade!=99 & Type=="O"
tabstat core, stat(sum) by(myprov)
tabstat ones if grade==99 | Type!="O", stat(sum)
* 2017
use "C:\My Documents\Resources (Data)\Department of Education\Databases downloaded off DoE website\Schools list for 2017 downloaded 2018 03\Produced by MG\Operational_ Ordinary School.dta", clear
rename sector Sector17
replace Sector17 = upper(Sector)
replace Sector17= substr(Sector, 1, 1)
rename type_doe Type17
replace Type17 = substr(Type, 1, 1)
keep if (Sector17=="P" | Sector17=="I") & Type17=="O" // 11
rename natemis emiscode
keep emiscode Sector17 Type17
merge 1:1 emiscode using "C:\My Documents\Numbercrunching\LURITS\temp0.dta"
drop _merge
save "C:\My Documents\Numbercrunching\LURITS\temp0.dta", replace
import delimited "C:\My Documents\Resources (Data)\Department of Education\LURITS from JM 2018 06\individual learner per school in 2017.txt", delimiter("|") varnames(1) clear
drop if province=="" // 2
drop datayear
gen year = 2017
rename natemis emiscode
drop if emiscode<"000000000" | emiscode>"999999999" // 2
replace idno = subinstr(idno, " " , "", .)
replace idno = upper(idno)
replace idno = "" if idno=="NULL"
replace idno = substr(idno, 1, 13)
replace accessionno = subinstr(accessionno, " ", "", .)
replace accessionno = upper(accessionno)
gen temp = length(idno)
codebook temp, tab(50)
replace idno = "0" + idno if temp==12
replace idno = "00" + idno if temp==11
replace idno = "000" + idno if temp==10
drop temp
gen temp = length(idno)
gen tempy = substr(idno, 1, 2) if temp==13
destring tempy, force gen(tempynum)
gen tempyyyy = cond(tempynum<=17, "20", "19") + tempy if temp==13
gen tempbirth = tempyyyy + substr(idno, 3, 4) if temp==13
destring tempbirth, gen(birthdate) force
drop temp*
rename learner_first_names fname
rename learner_surname sname
foreach n of varlist fname sname {
  replace `n' = subinstr(`n', " ", "", .)
  replace `n' = substr(`n', 1, 20)
  replace `n' = upper(`n')
  rename `n' temp
  gen `n' = ""
  gen templength = length(temp)
  quietly summ templength
  quietly forvalues j = 1 / `r(max)' {
    replace `n' = `n' + substr(temp, `j', 1) if inrange(substr(temp, `j', 1), "A", "Z")
            noisily display `j'
  }
  replace `n' = substr(`n', 1, 15)
  drop temp*
}
replace grade = upper(grade)
codebook grade, tab(500)
```

```
rename grade temp
gen grade = 0 if temp=="GRADE R"
forvalues g = 1 / 12 {
  replace grade = `g' if temp=="GRADE `g'"
}
codebook grade, tab(100)
drop temp
replace grade = 99 if grade==.
replace gender = upper(gender)
codebook gender, tab(50)
replace gender = cond(gender=="FEMALE", "F", cond(gender=="MALE", "M", ""))
keep year emiscode idno accessionno birthdate fname sname grade gender
order year emiscode idno accessionno birthdate fname sname grade gender
compress
save "C:\My Documents\Numbercrunching\LURITS\temp2.dta", replace
use "C:\My Documents\Numbercrunching\LURITS\temp2.dta", clear
gen ones = 1
collapse (sum) ones, by(emiscode grade)
merge m:1 emiscode using "C:\My Documents\Numbercrunching\LURITS\temp0.dta"
drop if _merge==2
drop _merge
gen statssaprov = substr(emiscode, 1, 1)
destring statssaprov, replace
merge m:1 statssaprov using "C:\My Documents\Numbercrunching\Tools\provinces.dta"
gen core = ones if grade!=99 & Type17=="O"
tabstat core, stat(sum) by(myprov)
tabstat ones if grade==99 | Type17!="O", stat(sum)
*2019
import excel "C:\My Documents\Resources (Data)\Department of Education\Databases downloaded off
DoE website\Schools list for 2019 09 downloaded 2022 09\National.xlsx", sheet("Ordinary schools")
firstrow clear
replace Sector = upper(Sector)
replace Sector = substr(Sector, 1, 1)
rename Sector Sector19
rename Type_DoE Type19
replace Type19 = substr(Type, 1, 1)
rename NatEMIS emiscode
keep emiscode Sector19 Type19
merge 1:1 emiscode using "C:\My Documents\Numbercrunching\LURITS\temp0.dta"
drop _merge
order Sector19 Type19, after(Type18)
save "C:\My Documents\Numbercrunching\LURITS\temp0.dta", replace
import delimited "D:\Lurits\2019\Learners_2019.txt", delimiter("#") bindquote(nobind) varnames(1) clear
drop if _n==_N | _n==_N - 1 // 2
gen year = 2019
rename ïemiscode emiscode
drop if emiscode<"000000000" | emiscode>"999999999" // 4
replace idno = subinstr(idno, " " , "", .)
replace idno = upper(idno)
replace idno = "" if idno=="NULL"
replace idno = substr(idno, 1, 13)
replace accessionno = subinstr(accessionno, " ", "", .)
replace accessionno = upper(accessionno)
destring birthdate, replace force
foreach n of varlist fname sname {
  replace `n' = subinstr(`n', " ", "", .)
  replace `n' = substr(`n', 1, 20)
  replace `n' = upper(`n')
  rename `n' temp
  gen `n' = ""
  gen templength = length(temp)
  quietly summ templength
  quietly forvalues j = 1 / `r(max)' {
    replace `n' = `n' + substr(temp, `j', 1) if inrange(substr(temp, `j', 1), "A", "Z")
          noisily display `j'
  }
  replace `n' = substr(`n', 1, 15)
```

```
  drop temp*
}
codebook grade, tab(50)
destring grade, force replace
replace grade = 99 if grade<0 | grade>12
replace gender = upper(gender)
codebook gender, tab(500)
replace gender = cond(gender=="FEMALE" | gender=="2", "F", cond(gender=="MALE" | gender=="1",
"M", ""))
keep year emiscode idno accessionno birthdate fname sname grade gender
order year emiscode idno accessionno birthdate fname sname grade gender
compress
save "C:\My Documents\Numbercrunching\LURITS\temp4.dta", replace
use "C:\My Documents\Numbercrunching\LURITS\temp4.dta", clear
gen ones = 1
collapse (sum) ones, by(emiscode grade)
merge m:1 emiscode using "C:\My Documents\Numbercrunching\LURITS\temp0.dta"
drop if _merge==2
drop _merge
gen statssaprov = substr(emiscode, 1, 1)
destring statssaprov, replace
merge m:1 statssaprov using "C:\My Documents\Numbercrunching\Tools\provinces.dta"
gen core = ones if grade!=99 & Type19=="O"
tabstat core, stat(sum) by(myprov)
tabstat ones if grade==99 | Type19!="O", stat(sum)
* 2020 - except EC
import excel "C:\My Documents\Resources (Data)\Department of Education\Databases downloaded off
DoE website\Schools list for 2020 03 downloaded 2022 09\National- Ordinary school.xlsx",
sheet("Sheet1") firstrow clear
replace Sector = upper(Sector)
replace Sector = substr(Sector, 1, 1)
rename Sector Sector20
rename Type_PED Type20
replace Type20 = substr(Type20, 1, 1)
rename NatEmis emiscode
keep emiscode Sector20 Type20
merge 1:1 emiscode using "C:\My Documents\Numbercrunching\LURITS\temp0.dta"
drop _merge
order Sector20 Type20, after(Type19)
save "C:\My Documents\Numbercrunching\LURITS\temp0.dta", replace
import delimited "D:\Lurits\2020\Learners.txt", delimiter("#") bindquote(nobind) varnames(1) clear
drop if _n==_N | _n==_N - 1 // 2
drop importyear
gen year = 2020
rename ïemiscode emiscode
drop if emiscode<"000000000" | emiscode>"999999999" // 5
replace idno = subinstr(idno, " " , "", .)
replace idno = upper(idno)
replace idno = "" if idno=="NULL"
replace idno = substr(idno, 1, 13)
replace accessionno = subinstr(accessionno, " ", "", .)
replace accessionno = upper(accessionno)
destring birthdate, replace force
foreach n of varlist fname sname {
  replace `n' = subinstr(`n', " ", "", .)
  replace `n' = substr(`n', 1, 20)
  replace `n' = upper(`n')
  rename `n' temp
  gen `n' = ""
  gen templength = length(temp)
  quietly summ templength
  quietly forvalues j = 1 / `r(max)' {
    replace `n' = `n' + substr(temp, `j', 1) if inrange(substr(temp, `j', 1), "A", "Z")
            noisily display `j'
  }
  replace `n' = substr(`n', 1, 15)
  drop temp*
```

```
}
codebook grade, tab(50)
destring grade, force replace
replace grade = 99 if grade<0 | grade>12
replace gender = upper(gender)
codebook gender, tab(500)
replace gender = cond(gender=="FEMALE" | gender=="2", "F", cond(gender=="MALE" | gender=="1",
"M", ""))
replace race = upper(race)
replace race = subinstr(race, " ", "", .)
codebook race, tab(500)  // Seems INDIAN and COLOURED have been switched!
gen statssaprov = substr(emiscode, 1, 1)
destring statssaprov, replace
merge m:1 statssaprov using "C:\My Documents\Numbercrunching\Tools\provinces.dta"
drop if _merge==2
gen tempuse = 1 if substr(race, 1, 6)=="INDIAN" | substr(race, 1, 5)=="ASIAN" | substr(race, 1,
8)=="COLOURED"
gen ones = 1
table myprov race if tempuse==1, content(sum ones) // Seems the switching occurred only in WC
rename race temp
gen race = "A" if substr(temp, 1, 7)=="AFRICAN" | temp=="BLACKAFRICAN"
replace race = "C" if substr(temp, 1, 8)=="COLOURED" & myprov!="WC"
replace race = "I" if substr(temp, 1, 6)=="INDIAN" | substr(temp, 1, 5)=="ASIAN" & myprov!="WC"
replace race = "W" if substr(temp, 1, 5)=="WHITE"
replace race = "O" if substr(temp, 1, 5)=="OTHER"
replace race = "I" if substr(temp, 1, 8)=="COLOURED" & myprov=="WC"
replace race = "C" if substr(temp, 1, 5)=="ASIAN" & myprov=="WC"
drop temp
table myprov race, content(sum ones)
keep year emiscode idno accessionno birthdate fname sname grade class gender race
order year emiscode idno accessionno birthdate fname sname grade class gender race
compress
save "C:\My Documents\Numbercrunching\LURITS\temp5.dta", replace
* 2020 - EC
use "C:\My Documents\Numbercrunching\LURITS\temp10.dta", clear // This is before the duplicate
removal process referred to in the 2021 enrolments report published by the DBE. The code to produce
this is in '2021 enrolment trends - new 9 of 9.do'.
count // 1,853,270 - agrees with Table 1 in the 2021 enrolments report. The duplicates removal process
referred to there reduce the EC enrolments by only 2K.
gen year = 2020
rename EmisCode emiscode
tostring emiscode, replace
drop if emiscode<"000000000" | emiscode>"999999999" // 0
rename IDNo idno
replace idno = subinstr(idno, " " , "", .)
replace idno = upper(idno)
replace idno = "" if idno=="NULL"
replace idno = substr(idno, 1, 13)
rename BirthDate birthdate
rename FName fname
rename SName sname
foreach n of varlist fname sname {
  replace `n' = subinstr(`n', " ", "", .)
  replace `n' = substr(`n', 1, 20)
  replace `n' = upper(`n')
  rename `n' temp
  gen `n' = ""
  gen templength = length(temp)
  quietly summ templength
  quietly forvalues j = 1 / `r(max)' {
    replace `n' = `n' + substr(temp, `j', 1) if inrange(substr(temp, `j', 1), "A", "Z")
            noisily display `j'
  }
  replace `n' = substr(`n', 1, 15)
  drop temp*
}
rename Grade grade
```

```
codebook grade, tab(50)
replace grade = 99 if grade<0 | grade>12
rename gender temp
gen gender = "M" if temp==1
replace gender = "F" if temp==2
drop temp
codebook race, tab(50)  // "O" is also there
keep year emiscode idno birthdate fname sname grade gender race
order year emiscode idno birthdate fname sname grade gender race
compress
append using "C:\My Documents\Numbercrunching\LURITS\temp5.dta"
order year emiscode idno accessionno birthdate fname sname grade class gender race
save "C:\My Documents\Numbercrunching\LURITS\temp5.dta", replace
use "C:\My Documents\Numbercrunching\LURITS\temp5.dta", clear
gen ones = 1
collapse (sum) ones, by(emiscode grade)
merge m:1 emiscode using "C:\My Documents\Numbercrunching\LURITS\temp0.dta"
drop if _merge==2
drop _merge
gen statssaprov = substr(emiscode, 1, 1)
destring statssaprov, replace
merge m:1 statssaprov using "C:\My Documents\Numbercrunching\Tools\provinces.dta"
gen core = ones if grade!=99 & Type20=="O"
tabstat core, stat(sum) by(myprov)
tabstat ones if grade==99 | Type20!="O", stat(sum)
*2021
use "C:\My Documents\Resources (Data)\Department of Education\Databases downloaded off DoE
website\Schools list for 2021 03 downloaded 2021 10\Ordinary Schools.dta", clear
rename sector Sector21
replace Sector21 = upper(Sector)
replace Sector21= substr(Sector, 1, 1)
rename type_ped Type21
replace Type21 = substr(Type21, 1, 1)
keep if (Sector21=="P" | Sector21=="I") & Type21=="O" // 36 - largely pre-primary
rename natemis emiscode
tostring emiscode, replace
keep emiscode Sector21 Type21
merge 1:1 emiscode using "C:\My Documents\Numbercrunching\LURITS\temp0.dta"
drop _merge
order Sector21 Type21, after(Type20)
save "C:\My Documents\Numbercrunching\LURITS\temp0.dta", replace
import delimited "D:\Lurits\2021\Learner2021.txt", delimiter("#") bindquote(nobind) varnames(1) clear
drop if _n==_N | _n==_N - 1 // 2
drop importyear
gen year = 2021
rename ïemiscode emiscode
drop if emiscode<"000000000" | emiscode>"999999999" // 5
replace idno = subinstr(idno, " " , "", .)
replace idno = upper(idno)
replace idno = "" if idno=="NULL"
replace idno = substr(idno, 1, 13)
replace accessionno = subinstr(accessionno, " ", "", .)
replace accessionno = upper(accessionno)
destring birthdate, replace force
foreach n of varlist fname sname {
  replace `n' = subinstr(`n', " ", "", .)
  replace `n' = substr(`n', 1, 20)
  replace `n' = upper(`n')
  rename `n' temp
  gen `n' = ""
  gen templength = length(temp)
  quietly summ templength
  quietly forvalues j = 1 / `r(max)' {
    replace `n' = `n' + substr(temp, `j', 1) if inrange(substr(temp, `j', 1), "A", "Z")
          noisily display `j'
  }
  replace `n' = substr(`n', 1, 15)
```

```
  drop temp*
}
codebook grade, tab(50)
destring grade, force replace
replace grade = 99 if grade<0 | grade>12
replace gender = upper(gender)
codebook gender, tab(500)
replace gender = cond(gender=="FEMALE" | gender=="2", "F", cond(gender=="MALE" | gender=="1",
"M", ""))
replace race = upper(race)
replace race = subinstr(race, " ", "", .)
codebook race, tab(500) // Seems same problem as in 2020
gen statssaprov = substr(emiscode, 1, 1)
destring statssaprov, replace
merge m:1 statssaprov using "C:\My Documents\Numbercrunching\Tools\provinces.dta"
drop if _merge==2
gen tempuse = 1 if substr(race, 1, 6)=="INDIAN" | substr(race, 1, 5)=="ASIAN" | substr(race, 1,
8)=="COLOURED"
gen ones = 1
table myprov race if tempuse==1, content(sum ones) // Same WC problem
rename race temp
gen race = "A" if substr(temp, 1, 7)=="AFRICAN" | temp=="BLACKAFRICAN"
replace race = "C" if substr(temp, 1, 8)=="COLOURED" & myprov!="WC"
replace race = "I" if substr(temp, 1, 6)=="INDIAN" | substr(temp, 1, 5)=="ASIAN" & myprov!="WC"
replace race = "W" if substr(temp, 1, 5)=="WHITE"
replace race = "O" if substr(temp, 1, 5)=="OTHER"
replace race = "I" if substr(temp, 1, 8)=="COLOURED" & myprov=="WC"
replace race = "C" if substr(temp, 1, 5)=="ASIAN" & myprov=="WC"
drop temp
codebook race, tab(50)
keep year emiscode idno accessionno birthdate fname sname grade class gender race
order year emiscode idno accessionno birthdate fname sname grade class gender race
compress
save "C:\My Documents\Numbercrunching\LURITS\temp6.dta", replace
use "C:\My Documents\Numbercrunching\LURITS\temp6.dta", clear
gen ones = 1
collapse (sum) ones, by(emiscode grade)
merge m:1 emiscode using "C:\My Documents\Numbercrunching\LURITS\temp0.dta"
drop if _merge==2
drop _merge
gen statssaprov = substr(emiscode, 1, 1)
destring statssaprov, replace
merge m:1 statssaprov using "C:\My Documents\Numbercrunching\Tools\provinces.dta"
gen core = ones if grade!=99 & Type21=="O"
tabstat core, stat(sum) by(myprov)
tabstat ones if grade==99 | Type21!="O", stat(sum)

* MISSING DATA

* By year...
forvalues f = 2 / 6 {
  use "C:\My Documents\Numbercrunching\LURITS\temp`f'.dta", clear
  display "file `f'"
  quietly foreach v in "idno" "accessionno" "birthdate" "fname" "sname" "grade" "class" "gender" "race" {
    capture confirm variable `v'
    if _rc==0 {
            if "`v'"=="birthdate" | "`v'"=="grade" {
              if "`v'"=="birthdate" {
                    gen temp = cond(`v'==., 1, 0)
                  }
                  else {
                    gen temp = cond(`v'==99, 1, 0)
                  }
            }
            else {
              gen temp = cond(`v'=="", 1, 0)
            }
```

```
              mean temp
              local perc = _b[temp] * 100
              drop temp
              noisily display "`v' " _column(15) `perc'
            }
  }
}
* By province
quietly foreach v in "idno" "accessionno" "birthdate" "fname" "sname" "grade" "class" "gender" "race" {
  clear
  set obs 0
  gen emiscode = ""
  gen missing = .
  save "C:\My Documents\Numbercrunching\LURITS\temp11.dta", replace
  forvalues f = 2 / 6 {
    use "C:\My Documents\Numbercrunching\LURITS\temp`f'.dta", clear
            capture confirm variable `v'
    if _rc==0 {
              if "`v'"=="birthdate" | "`v'"=="grade" {
                if "`v'"=="birthdate" {
                        gen missing = cond(`v'==., 1, 0)
                      }
                      else {
                        gen missing = cond(`v'==99, 1, 0)
                      }
              }
              else {
                gen missing = cond(`v'=="", 1, 0)
              }
              keep emiscode missing
              append using "C:\My Documents\Numbercrunching\LURITS\temp11.dta"
              save "C:\My Documents\Numbercrunching\LURITS\temp11.dta", replace
            }
            else {
              gen missing = 1
              keep emiscode missing
              append using "C:\My Documents\Numbercrunching\LURITS\temp11.dta"
              save "C:\My Documents\Numbercrunching\LURITS\temp11.dta", replace
            }
  }
  noisily display "`v'"
  gen statssaprov = substr(emiscode, 1, 1)
  destring statssaprov, replace
  merge m:1 statssaprov using "C:\My Documents\Numbercrunching\Tools\provinces.dta"
  noisily tabstat missing, by(myprov)
}

* ANONYMISATION

* idno
forvalues f = 2 / 6 {
  use "C:\My Documents\Numbercrunching\LURITS\temp`f'.dta", clear
  keep idno
  drop if idno==""
  gen temp = length(idno)
  replace idno = "0" + idno if temp==12
  replace idno = "00" + idno if temp==11
  replace idno = "000" + idno if temp==10
  contract idno
  drop _freq
  if `f'!=2 {
    append using "C:\My Documents\Numbercrunching\LURITS\temp11.dta"
  }
  contract idno
  drop _freq
  save "C:\My Documents\Numbercrunching\LURITS\temp11.dta", replace
}
```

```
use "C:\My Documents\Numbercrunching\LURITS\temp11.dta", clear
gen myrand = uniform()
sort myrand
drop myrand
gen long idno_anon = _n
format idno_anon %8.0f
compress
save "C:\My Documents\Numbercrunching\LURITS\idno_key.dta", replace
forvalues f = 2 / 6 {
  use "C:\My Documents\Numbercrunching\LURITS\temp`f'.dta", clear
  merge m:1 idno using "C:\My Documents\Numbercrunching\LURITS\idno_key.dta"
  drop if _merge==2
  drop _merge
  order idno_anon, after(idno)
  drop idno
  compress
  save "C:\My Documents\Numbercrunching\LURITS\temp`f'.dta", replace
}
* accessionno
forvalues f = 2 / 6 {
  use "C:\My Documents\Numbercrunching\LURITS\temp`f'.dta", clear
  keep accessionno
  drop if accessionno==""
  contract accessionno
  drop _freq
  if `f'!=2 {
    append using "C:\My Documents\Numbercrunching\LURITS\temp11.dta"
  }
  contract accessionno
  drop _freq
  save "C:\My Documents\Numbercrunching\LURITS\temp11.dta", replace
}
use "C:\My Documents\Numbercrunching\LURITS\temp11.dta", clear
gen myrand = uniform()
sort myrand
drop myrand
gen long accessionno_anon = _n
format accessionno_anon %8.0f
compress
save "C:\My Documents\Numbercrunching\LURITS\accessionno_key.dta", replace
forvalues f = 2 / 6 {
  use "C:\My Documents\Numbercrunching\LURITS\temp`f'.dta", clear
  merge m:1 accessionno using "C:\My Documents\Numbercrunching\LURITS\accessionno_key.dta"
  drop if _merge==2
  drop _merge
  order accessionno_anon, after(accessionno)
  drop accessionno
  compress
  save "C:\My Documents\Numbercrunching\LURITS\temp`f'.dta", replace
}
* birthdate (also insertion of birthyear)
forvalues f = 2 / 6 {
  use "C:\My Documents\Numbercrunching\LURITS\temp`f'.dta", clear
  keep birthdate
  drop if birthdate==.
  contract birthdate
  drop _freq
  if `f'!=2 {
    append using "C:\My Documents\Numbercrunching\LURITS\temp11.dta"
  }
  contract birthdate
  drop _freq
  save "C:\My Documents\Numbercrunching\LURITS\temp11.dta", replace
}
use "C:\My Documents\Numbercrunching\LURITS\temp11.dta", clear
gen myrand = uniform()
sort myrand
```

```
drop myrand
gen long birthdate_anon = _n
format birthdate_anon %8.0f
compress
save "C:\My Documents\Numbercrunching\LURITS\birthdate_key.dta", replace
forvalues f = 2 / 6 {
  use "C:\My Documents\Numbercrunching\LURITS\temp`f'.dta", clear
  tostring birthdate, gen(temp1)
  gen tempyear = substr(temp1, 1, 4)
  destring tempyear, replace force
  gen tempmonth = substr(temp1, 5 , 2)
  destring tempmonth, replace force
  gen birthyear = tempyear + cond(tempmonth>6, .5, 0)
  drop temp*
  format birthyear %4.1f
  merge m:1 birthdate using "C:\My Documents\Numbercrunching\LURITS\birthdate_key.dta"
  drop if _merge==2
  drop _merge
  order birthdate_anon, after(birthdate)
  drop birthdate
  compress
  save "C:\My Documents\Numbercrunching\LURITS\temp`f'.dta", replace
}
* fname and sname
foreach n in "fname" "sname" {
  forvalues f = 2 / 6 {
    use "C:\My Documents\Numbercrunching\LURITS\temp`f'.dta", clear
    keep `n'
    drop if `n'==""
    contract `n'
    drop _freq
    if `f'!=2 {
      append using "C:\My Documents\Numbercrunching\LURITS\temp11.dta"
    }
    contract `n'
    drop _freq
    save "C:\My Documents\Numbercrunching\LURITS\temp11.dta", replace
  }
  use "C:\My Documents\Numbercrunching\LURITS\temp11.dta", clear
  gen myrand = uniform()
  sort myrand
  drop myrand
  gen long `n'_anon = _n
  format `n'_anon %8.0f
  compress
  save "C:\My Documents\Numbercrunching\LURITS\\`n'_key.dta", replace
  forvalues f = 2 / 6 {
    use "C:\My Documents\Numbercrunching\LURITS\temp`f'.dta", clear
    merge m:1 `n' using "C:\My Documents\Numbercrunching\LURITS\\`n'_key.dta"
    drop if _merge==2
    drop _merge
    order `n'_anon, after(`n')
    drop `n'
    compress
    save "C:\My Documents\Numbercrunching\LURITS\temp`f'.dta", replace
  }
}

* CREATION OF ONE LEARNER FILE

forvalues f = 2 / 6 {
  use "C:\My Documents\Numbercrunching\LURITS\temp`f'.dta", clear
  if `f'!=2 {
    append using "C:\My Documents\Numbercrunching\LURITS\learner2022_12.dta"
  }
  compress
  save "C:\My Documents\Numbercrunching\LURITS\learner2022_12.dta", replace
```

```
}
use "C:\My Documents\Numbercrunching\LURITS\learner2022_12.dta", clear
codebook gender
rename gender temp
encode temp, gen(gender)
drop temp
codebook race
rename race temp
encode temp, gen(race)
drop temp
compress
order year emiscode idno_anon accessionno_anon birthdate_anon fname_anon sname_anon grade
class gender race birthyear
save "C:\My Documents\Numbercrunching\LURITS\learner2022_12.dta", replace
use "C:\My Documents\Numbercrunching\LURITS\learner2022_12.dta", clear
drop if class==""
contract class
drop _freq
gen myrand = uniform()
sort myrand
drop myrand
gen class_code = _n
save "C:\My Documents\Numbercrunching\LURITS\class_key.dta", replace
use "C:\My Documents\Numbercrunching\LURITS\learner2022_12.dta", clear
merge m:1 class using "C:\My Documents\Numbercrunching\LURITS\class_key.dta"
drop _merge
order class_code, after(class)
drop class
compress
save "C:\My Documents\Numbercrunching\LURITS\learner2022_12.dta", replace
use "C:\My Documents\Numbercrunching\LURITS\learner2022_12.dta", clear
destring emiscode, replace
format emiscode %9.0f
compress
sort year emiscode grade
save "C:\My Documents\Numbercrunching\LURITS\learner2022_12.dta", replace

* LEARNERS IDENTIFIED UNIQUELY WITHIN EACH YEAR

use "C:\My Documents\Numbercrunching\LURITS\learner2022_12.dta", clear
by year idno_anon, sort: egen temp = count(_n)
gen idno_id = cond(temp==1 & idno_anon!=., 1, 0)
drop temp
tabstat idno_id, by(year)
by year accessionno_anon, sort: egen temp = count(_n)
gen accessionno_id = cond(temp==1 & accessionno_anon!=., 1, 0)
drop temp
tabstat accessionno_id, by(year)
by year birthdate_anon fname_anon sname_anon gender race, sort: egen temp = count(_n)
gen combo1_id = cond(temp==1 & birthdate_anon!=. & fname_anon!=. & sname_anon!=. & gender!=. &
race!=., 1, 0)
drop temp
tabstat combo1_id, by(year)
by year birthdate_anon fname_anon sname_anon gender, sort: egen temp = count(_n)
gen combo2_id = cond(temp==1 & birthdate_anon!=. & fname_anon!=. & sname_anon!=. & gender!=.,
1, 0)
drop temp
tabstat combo2_id, by(year)
by year birthdate_anon sname_anon gender, sort: egen temp = count(_n)
gen combo3_id = cond(temp==1 & birthdate_anon!=. & sname_anon!=. & gender!=., 1, 0)
drop temp
tabstat combo3_id, by(year)
egen composite1 = rowmax(idno_id combo2_id)
tabstat composite1, by(year)
egen composite2 = rowmax(idno_id accessionno_id combo2_id)
tabstat composite2, by(year)
```

```
* SCHOOL FILE

use "C:\My Documents\Numbercrunching\LURITS\temp0.dta", clear
compress
save "C:\My Documents\Numbercrunching\LURITS\school2022_12.dta", replace

* LINKING TO NEXT YEAR

use "C:\My Documents\Numbercrunching\LURITS\learner2022_12.dta", clear
gen L = 1
tabstat L if grade>=1 & grade<=6, stat(sum) by(year)
* A: idno
use "C:\My Documents\Numbercrunching\LURITS\learner2022_12.dta", clear
gen L = 1
drop if idno_anon==.
by year idno_anon, sort: egen temp = count(_n)
keep if temp==1
drop temp
forvalues y = 2017 / 2020 {
  gen temp = 1 if year==`y' & grade>=1 & grade<=6
  by idno_anon, sort: egen prim`y' = max(temp)
  drop temp
}
keep idno_anon year prim* L
reshape wide L, i(idno_anon prim*) j(year)
quietly forvalues y = 2017 / 2020 {
  local yplus = `y' + 1
  gen templinked = 1 if L`y'==1 & L`yplus'==1
  summ templinked
  local all = r(sum)
  summ templinked if prim`y'==1
  local prim = r(sum)
  noisily display `y' _column(8) `all' _column(18) `prim'
  drop temp*
}
* B: accessionno
use "C:\My Documents\Numbercrunching\LURITS\learner2022_12.dta", clear
gen L = 1
drop if accessionno_anon==.
by year accessionno_anon, sort: egen temp = count(_n)
keep if temp==1
drop temp
forvalues y = 2017 / 2020 {
  gen temp = 1 if year==`y' & grade>=1 & grade<=6
  by accessionno_anon, sort: egen prim`y' = max(temp)
  drop temp
}
keep accessionno_anon year prim* L
reshape wide L, i(accessionno_anon prim*) j(year)
quietly forvalues y = 2017 / 2020 {
  local yplus = `y' + 1
  gen templinked = 1 if L`y'==1 & L`yplus'==1
  summ templinked
  local all = r(sum)
  summ templinked if prim`y'==1
  local prim = r(sum)
  noisily display `y' _column(8) `all' _column(18) `prim'
  drop temp*
}
* D: birthdate fname sname gender
use "C:\My Documents\Numbercrunching\LURITS\learner2022_12.dta", clear
gen L = 1
drop if birthdate_anon==. | fname_anon==. | sname_anon==. | gender==.
by year birthdate_anon fname_anon sname_anon gender, sort: egen temp = count(_n)
keep if temp==1
drop temp
forvalues y = 2017 / 2020 {
```

```
  gen temp = 1 if year==`y' & grade>=1 & grade<=6
  by birthdate_anon fname_anon sname_anon gender, sort: egen prim`y' = max(temp)
  drop temp
}
keep birthdate_anon fname_anon sname_anon gender year prim* L
reshape wide L, i(birthdate_anon fname_anon sname_anon gender prim*) j(year)
quietly forvalues y = 2017 / 2020 {
  local yplus = `y' + 1
  gen templinked = 1 if L`y'==1 & L`yplus'==1
  summ templinked
  local all = r(sum)
  summ templinked if prim`y'==1
  local prim = r(sum)
  noisily display `y' _column(8) `all' _column(18) `prim'
  drop temp*
}
* A then D
quietly forvalues y = 2017 / 2020 {
  local yplus = `y' + 1
  use "C:\My Documents\Numbercrunching\LURITS\learner2022_12.dta", clear
  keep if year==`y' | year==`yplus'
  keep year grade idno_anon birthdate_anon fname_anon sname_anon gender
  gen long newid = _n
  save "C:\My Documents\Numbercrunching\LURITS\temp11.dta", replace
  keep year grade newid idno_anon
  drop if idno==.
  by year idno, sort: egen temp = count(_n)
  keep if temp==1
  drop temp
  by idno, sort: egen temp = count(_n)
  keep if temp==2
  drop temp
  save "C:\My Documents\Numbercrunching\LURITS\temp12.dta", replace
  use "C:\My Documents\Numbercrunching\LURITS\temp11.dta", clear
  merge 1:1 newid using "C:\My Documents\Numbercrunching\LURITS\temp12.dta", keepusing(newid)
  drop if _merge==3
  drop _merge
  keep year grade newid birthdate_anon fname_anon sname_anon gender
  drop if birthdate_anon==. | fname_anon==. | sname_anon==. | gender==.
  by year birthdate_anon fname_anon sname_anon gender, sort: egen temp = count(_n)
  keep if temp==1
  drop temp
  by birthdate_anon fname_anon sname_anon gender, sort: egen temp = count(_n)
  keep if temp==2
  drop temp
  count
  append using "C:\My Documents\Numbercrunching\LURITS\temp12.dta"
  count if year==`y'
  local all = r(N)
  count if year==`y' & grade>=1 & grade<=6
  local prim = r(N)
  noisily display `y' _column(8) `all' _column(18) `prim'
}
* A then D then B
quietly forvalues y = 2017 / 2020 { // Normally 2017 / 2020. But make this range cover just one year in
order to extract temp13.dta specific to that year.
  noisily display "Year is `y'
*****************************************************************************************************"
  local yplus = `y' + 1
  use "C:\My Documents\Numbercrunching\LURITS\learner2022_12.dta", clear
  keep if year==`y' | year==`yplus'
  tostring emiscode, replace force
  replace emiscode = substr(emiscode, 1, 1)
  destring emiscode, replace
  rename emiscode statssaprov
  merge m:1 statssaprov using "C:\My Documents\Numbercrunching\Tools\provinces.dta"
  gen L = 1
```

```
  noisily table grade myprov if grade!=99 & year==`y', content(sum L)
  keep year myprov grade idno_anon accessionno_anon birthdate_anon fname_anon sname_anon
gender
  gen long newid = _n
  save "C:\My Documents\Numbercrunching\LURITS\temp11.dta", replace
  use "C:\My Documents\Numbercrunching\LURITS\temp11.dta", clear
  keep year myprov grade newid idno_anon
  drop if idno==.
  by year idno, sort: egen temp = count(_n)
  keep if temp==1
  drop temp
  by idno, sort: egen temp = count(_n)
  keep if temp==2
  drop temp
  gen group = "A"
  sort idno year
  save "C:\My Documents\Numbercrunching\LURITS\temp12.dta", replace
  use "C:\My Documents\Numbercrunching\LURITS\temp11.dta", clear
  merge 1:1 newid using "C:\My Documents\Numbercrunching\LURITS\temp12.dta", keepusing(newid)
  drop if _merge==3
  drop _merge
  keep year myprov grade newid birthdate_anon fname_anon sname_anon gender
  drop if birthdate_anon==. | fname_anon==. | sname_anon==. | gender==.
  by year birthdate_anon fname_anon sname_anon gender, sort: egen temp = count(_n)
  keep if temp==1
  drop temp
  by birthdate_anon fname_anon sname_anon gender, sort: egen temp = count(_n)
  keep if temp==2
  drop temp
  gen group = "D"
  sort birthdate_anon fname_anon sname_anon gender year
  append using "C:\My Documents\Numbercrunching\LURITS\temp12.dta"
  save "C:\My Documents\Numbercrunching\LURITS\temp12.dta", replace
  use "C:\My Documents\Numbercrunching\LURITS\temp11.dta", clear
  merge 1:1 newid using "C:\My Documents\Numbercrunching\LURITS\temp12.dta", keepusing(newid)
  drop if _merge==3
  drop _merge
  keep year myprov grade newid accessionno_anon
  drop if accessionno_anon==.
  by year accessionno_anon, sort: egen temp = count(_n)
  keep if temp==1
  drop temp
  by accessionno_anon, sort: egen temp = count(_n)
  keep if temp==2
  drop temp
  gen group = "B"
  sort accessionno_anon year
  append using "C:\My Documents\Numbercrunching\LURITS\temp12.dta"
  gen gradeokay = 1 if year==`y' & (grade[_n + 1] - grade==1 | grade[_n + 1] - grade==0)
  count if year==`y'
  local all = r(N)
  count if year==`y' & grade>=1 & grade<=6
  local prim = r(N)
  count if year==`y' & gradeokay==1
  local gradechecked = r(N)
  noisily display `y' _column(8) `all' _column(18) `prim' _column(28) `gradechecked'
  save "C:\My Documents\Numbercrunching\LURITS\temp13.dta", replace
  keep if year==`y'
  drop if grade==99
  gen L = 1
  noisily table grade myprov, content(sum L)
  noisily table grade myprov if gradeokay==1, content(sum L)
}

* DIGGING DEEPER INTO THE LINKED WITH THE WRONG GRADE USING 2018-2019

use "C:\My Documents\Numbercrunching\LURITS\temp13.dta", clear
```

```
tabstat year if year==2018 & gradeokay!=1, by(group) stat(count) // 92,675
keep if group=="A"
sort idno_anon year
by idno_anon, sort: egen temp = max(gradeokay)
keep if temp==.
drop temp
gen temp1 = grade if year==2018
by idno_anon, sort: egen tempstart = mean(temp1)
gen temp2 = grade if year==2019
by idno_anon, sort: egen tempend = mean(temp2)
gen gradediff = tempend - tempstart
drop temp*
replace gradediff = 85 if gradediff>85
replace gradediff = -85 if gradediff<-85
gen outofrange = cond(gradediff==85 | gradediff==-85, 1, 0)
tabstat outofrange if year==2018 // .1831495
codebook gradediff if year==2018 & outofrange!=1, tab(500)
use "C:\My Documents\Numbercrunching\LURITS\temp13.dta", clear
keep if group=="D"
sort birthdate_anon fname_anon sname_anon gender year
by birthdate_anon fname_anon sname_anon gender, sort: egen temp = max(gradeokay)
keep if temp==.
drop temp
gen temp1 = grade if year==2018
by birthdate_anon fname_anon sname_anon gender, sort: egen tempstart = mean(temp1)
gen temp2 = grade if year==2019
by birthdate_anon fname_anon sname_anon gender, sort: egen tempend = mean(temp2)
gen gradediff = tempend - tempstart
drop temp*
replace gradediff = 85 if gradediff>85
replace gradediff = -85 if gradediff<-85
gen outofrange = cond(gradediff==85 | gradediff==-85, 1, 0)
tabstat outofrange if year==2018 // .1743704
codebook gradediff if year==2018 & outofrange!=1, tab(500)
```