Procedures for Cleaning, Coding and Formatting Health Professional Licensure Survey Data

Bowen Center for Health Workforce Research and Policy

Version Date: July 18, 2018

Table of Contents

I. Storage of licensure and survey data files from IPLA

a. License and survey data files are periodically transferred from IPLA to the Bowen Center using WINSCP

   i. An email from the IPLA will notify The Bowen Center when the most recent license and survey data files are ready for secure transfer.

   ii. IPLA will provide information on the licensed professionals included in the data files and instructions for accessing the data files on WINSCP

b. Original data files will be stored in the HWS Data folder under PLA Source Data, inside a new folder with the title "PLA Data Extract [Date of Data Extract]".

   i. Only designated Bowen Center team members will have access to the HWS Data fold and be able to transfer the licensure and survey data files.

The following procedures outline the formatting, cleaning and coding of licensure survey data using SAS

II. Formatting Survey Data

a. The raw survey data file contains survey data for all health professionals who renewed their license online. Each health professional will have one observation per question asked in the online survey.

b. Import data into SAS and create a raw survey data set for a specific health professional by restricting by the license prefix of the specific health professional. See Appendix A for list of license prefixes for licensed Indiana health professionals.

c. Transpose the survey question answers to create a data set in which each health professional had one observation. The transposed data set will contain the following variables:

   i. Unique license number
   ii. Profession ID
   iii. Profession Name
   iv. License Type ID
   v. License Type Name
   vi. Health Professional's Full Name
   vii. Health Professional's Sort Name (Last Name, First Name)
   viii. Create Date
   ix. Update Date
   x. Survey Answers (number of questions vary by health profession)

   ** The survey data file contains two license status variables (License Status ID and License Status Name) which should not be used. These variables reflect the license status at the time the survey was created, not the status of the license after it has been renewed. Only the license status variable in the license file (sec_lic_status) should be used for the final licensure survey data. **

III. Coding Survey Data

a. All fields in the original survey data file are formatted as text fields and will need to be recoded and formatted. Each survey questions will be coded according to the coding definitions outlined in the health professional data dictionary. Coding schemes of variables commonly used in licensure surveys are available in Appendix B.

b. Reported practice hours will be used to determine practice FTE for all practice locations. Below is a table which outlines this conversion:

Hours per Week in Patient Care	**FTE Conversion**
0 hours per week	0
1 – 4 hours per week	0.1
5 – 8 hours per week	0.2
9 – 12 hours per week	0.3
13 – 16 hours per week	0.4
17 – 20 hours per week	0.5
21 – 24 hours per week	0.6
25 – 28 hours per week	0.7
29 – 32 hours per week	0.8
33 – 36 hours per week	0.9
37 – 40 hours per week	1
41 or more hours per week	1

c. Create the variable Respondent to indicate whether the health professional responded to the online survey.

i. Health professional answered at least one survey question: Respondent=1

ii. Health professional did not answer any survey question: Respondent=0

IV. Formatting License Data

a. The license data file contains the unique license number, mailing address, birth date, license status and issues and expiration dates for all Indiana health professionals. All fields in the original license data file are formatted as text fields and some will need to be reformatted in order to be used in the licensure survey data set.

b. Import the license data into SAS and create a raw license data set for a specific health professional by restricting by the license prefix of the specific health professional. See Appendix A for list of license prefixes for licensed Indiana health professionals.

i. Data will also be restricted by the license suffix. Only licenses ending with "A" will be included in the license data set.

c. Formating date_of_birth

i. The variable date_of_birth will need to be converted from a datetime variable to a date variable. This can be done using the datepart(date_of_birth) function.

V. Merging and Formatting License and Survey Data Sets

a. Merge the license and survey data set by unique license number

b. If Respondent status is blank for a health professional *and* their license status is not active (sec_lic_status in 1, 15, 50) then remove the health professional from the new data set. Survey respondents with inactive licenses will remain in the data set. They will be removed when the inclusion and exclusion criteria is applied to create a survey sample for the data report. Appendix C outlines the inclusion and exclusion criteria for creating the survey sample.

c. Formating survey completion date
   1. Update_date will be used as the survey date
   2. If update_date is blank for a respondent, then use create_date as the survey date

d. Create a new variable called SurveyStatus to indicate the survey status of health professionals who renewed their license online and offline.
   i. If Respondent is equal to 1, then SurveyStatus equals "Respondent".
   ii. If Respondent is equal to 0, then SurveyStatus equals "Non-Respondent".
   iii. If Respondent is blank then, SurveyStatus equals "Offline".
   iv. Remove the variable Respondent once this variable has been created.

VI. Address Cleaning

Beginning January 2018, all address cleaning will be performed by the Polis Center using 360Science. This software compares input addresses to recently updated USPS address data and outputs a corrected address with standardized street spelling and ZIP codes. The small number of addresses that cannot be standardized using 360Science will be cleaned using a custom cleaning step developed by Polis.

The steps below provide the process for standardizing license and practice addresses in SAS.

a. Cleaning License Address
   i. License address cleaning will only occur for addresses that are located in Indiana.

   ii. The license file contains the license address for each licensed health professional. The physical address comes in two lines or variables (addr_line_1 and addr_line_2).

   iii. Both address variables will be cleaned and standardized using the code found in Appendix C. The variable which contains the physical address will be copied to a new license address variable (new_license_address).

      1. Remove suite, room, apartment and building numbers from the practice address. This can be done using the index function (See Appendix C for more details).
      2. The table below outlines the parts of street names to convert.

Word	Conversion
Ordinals	
First	1st
Second	2nd
Third	3rd
Fourth	4th
Fifth	5th
Sixth	6th
Seventh	7th
Ninth	9th
Tenth	10th
Cardinal Directions	
South	S
North	N
East	E
West	W

Word	Conversion
Street Title	
Road	Rd
Street	St
Lane	Ln
Avenue	Ave
Drive	Dr
Boulevard	Blvd
Court	Ct
Parkway	Pkwy
Square	Sq
Terrace	Ter
Circle	Cir
Other Street Names	
SR/St. Rd.	State Road
CR/Co. Rd.	County Road

3. In order to have a uniform address variable, a third address variable will be created using pattern matching (newLicenseAddress). The practice address variable must only include the street number, street name street title.

iv. License address city will be formatted as proper case

v. License address ZIP code will be formatted as a 5-digit number.

i. License Address City

1. Convert practice city to proper case and ensure correct spelling.
2. If the practice city contains a state name, the state must be removed from the field. This can be done using the *index* function.

ii. Practice ZIP Code

3. Format the practice ZIP code variable as a five digit number.

b. Cleaning Practice Address

i. Practice Address Variable

1. Convert words in the practice address to proper case.
2. Suite, room, apartment and building numbers will be removed from the practice address. This can be done using the index command. (See Appendix C for SAS code used to clean addresses)
3. The table below outlines the parts of street names to convert.

Word	Conversion
Ordinals	
First	1st
Second	2nd
Third	3rd
Fourth	4th
Fifth	5th
Sixth	6th
Seventh	7th
Ninth	9th
Tenth	10th

Word	Conversion
Street Title	
Road	Rd
Street	St
Lane	Ln
Avenue	Ave
Drive	Dr
Boulevard	Blvd
Court	Ct
Parkway	Pkwy
Square	Sq

Word	Conversion
Ordinals	
Cardinal Directions	
South	S
North	N
East	E
West	W

Word	Conversion
Street Title	
Terrace	Ter
Circle	Cir
Other Street Names	
SR/St. Rd.	State Road
CR/Co. Rd.	County Road

ii. In order to have a uniform address variable, a second address variable will be created using pattern matching (newPrimaryAddress). The practice address variable must only include the street number, street name street title.

iii. If practice address variable contains an incomplete address or extraneous content (e.g. “N/A”, “not currently working”, “retired”gb , etc.), convert the field to missing.

c. Practice City

i. Convert practice city to proper case and ensure correct spelling

ii. If the practice city contains a state name, the state must be removed from the field. This can be done using the *index* function.

d. Practice ZIP Code

i. Format the practice ZIP code variable as a five digit number.

VII. Finalizing the Health Professional Data Set

a. Reorder the variables in the data set to the order outlined in the data dictionary.

b. Format the coded variables using the format values outlined in the data dictionary.

c. Create a permanent SAS data set in the current year of the health professional survey folder under ‘SAS’.

d. The final merged license survey data set will be sent to Biostatistics (transferred to the J drive), where it will be stored in the longitudinal relational database (Indiana Health Professions Database).

VIII. Geocoding

a. Geocoding of license addresses will be conducted by the Polis Center.

b. Designated members of the Polis Center team will be given access to the longitudinal data base in order to extract license address data for cleaning and geocoding. This process will take approximately 3 weeks.

c. After geocoding is completed, the new geographical information (match level, census tract and county assignment) data will be imported into the longitudinal database.

IX. Reporting Sample

a. The reporting sample will be selected based on criteria outlined below. This sample will be used for data reports and data visualization on the Bowen Data Portal.

Include	Exclude
Total license renewals - All health professionals who renewed their license online, and health professional who renewed their license offline and their license status equals 1, 15 or 50	
	Non-Active Licenses - All health professionals whose license status (sec_lic_status) not in (1, 15, 50)
Total Active Licenses - All health professionals who license status (sec_lic_status) in 1, 15, 50	
	Total non-respondents - Health professionals who did not complete the survey
Total Respondents - Health professionals who responded to at least one survey question	
	Not actively practicing - Health professionals who reported not actively practicing in a field that requires a license
Actively practicing - Health professionals reported actively practicing in a field that requires a license	
	No license address in Indiana - Health professionals who have a license address state of Indiana
License address in Indiana - Health professionals who have a license address state of Indiana	
	Unable to confirm Indiana practice address - Indiana practice address could not be geocoded
Had a confirmed Indiana practice address: SURVEY SAMPLE - Indiana practice address could be geocoded and an Indiana county was identified for the practice address	

Appendix A: Health Professional License Prefix

Health Profession	License Prefix
Physician: Allopathic	01
Physician: Osteopathic	02
Physical Therapist (PT)	05
Podiatrist	07
Chiropractor	08
Physician Assistant (PA)	10
Dentist	12
Dental Hygienist	13
Hearing Aid Dealer	17
Optometrist	18
Psychologist	20
Speech Pathologist	22
Audiologist	23
Pharmacist	26
Licensed Practical Nurse (LPN)	27
Registered Nurse (RN)	28
APN Prescriptive Authority	71
Occupational Therapist	31
Social Worker	33
Clinical Social Worker	34
Marriage & Family Therapist	35
Mental Health Counselor	39
Marriage & Family Associate	85
Addiction Counselor Associate	869
Addiction Counselor	860
Clinical Addiction Counselor Associate	879
Clinical Addiction Counselor	870
Mental Health Associate	88

Appendix B: Coding Scheme for Commonly Used variables in Health Professional Licensure Surveys

Variable	Category	Code
Sex	Male	1
	Female	2
Hispanic_Latino	Yes	1
	No	2
Race	White	1
	American Indian or Alaska Native	2
	Native Hawaiian or Other Pacific Islander	3
	Black or African American	4
	Asian	5
	Multiracial (more than one race selected)	6
QualifyingDegreeState, MedicalDegreeState, ResidencyState	Indiana	1
	Michigan	2
	Illinois	3
	Kentucky	4
	Ohio	5
	Another State (not listed)	6
	Another Country (not U.S.)	7
PrimaryPracticeHours, SecondPracticeHours, ThirdPracticeHours, PrimaryHoursWorked, SecondHoursWorked, PrimaryHoursTotal, SecondHoursTotal, PrimaryHoursPtCare SecondHoursPtCare	0 hours per week	1
	1 – 4 hours per week	2
	5 – 8 hours per week	3
	9 – 12 hours per week	4
	13 – 16 hours per week	5
	17 – 20 hours per week	6
	21 – 24 hours per week	7
	25 – 28 hours per week	8
	29 – 32 hours per week	9
	33 – 36 hours per week	10
	37 – 40 hours per week	11
	41 or more hours we week	12

Appendix C: Address Cleaning SAS Code

Practice Address Cleaning Code:

Practice Address Variable

1. Formatting Primary Practice Address as proper case

```
newPrimaryAddress=Propcase(PrimaryPracticeAddress);
```

2. Removing unit number preceded by #

```
find2=index(newPrimaryAddress,"#");
if find2 gt 1 then newPrimaryAddress=substr(newPrimaryAddress,1,find2-1);
```

3. Compressing practice address variable to remove extra spaces, dashes, commas and periods

```
newPrimaryAddress=compress(newPrimaryAddress,,'adks');
```

4. Removing suite, room and building numbers from address string

```
find2=index(newPrimaryAddress,"Suite");
if find2 gt 1 then newPrimaryAddress=substr(newPrimaryAddress,1,find2-1);

find2=index(newPrimaryAddress,"Ste");
if find2 gt 1 then newPrimaryAddress=substr(newPrimaryAddress,1,find2-1);

find2=index(newPrimaryAddress," Room");
if find2 gt 1 then newPrimaryAddress=substr(newPrimaryAddress,1,find2-1);

find2=index(newPrimaryAddress," Bldg ");
if find2 gt 1 then newPrimaryAddress=substr(newPrimaryAddress,1,find2-1);
```

5. Converting ordinals, cardinal directions and street titles

```
newPrimaryAddress=tranwrd(newPrimaryAddress, "First", "1st");
newPrimaryAddress=tranwrd(newPrimaryAddress, "Second", "2nd");
newPrimaryAddress=tranwrd(newPrimaryAddress, "Third", "3rd");
newPrimaryAddress=tranwrd(newPrimaryAddress, "Fourth", "4th");
newPrimaryAddress=tranwrd(newPrimaryAddress, "Fifth", "5th");
newPrimaryAddress=tranwrd(newPrimaryAddress, "Sixth", "6th");
newPrimaryAddress=tranwrd(newPrimaryAddress, "Seventh", "7th");
newPrimaryAddress=tranwrd(newPrimaryAddress, "Eighth", "8th");
newPrimaryAddress=tranwrd(newPrimaryAddress, "Ninth", "9th");
newPrimaryAddress=tranwrd(newPrimaryAddress, "Tenth", "10th");

newPrimaryAddress=tranwrd(newPrimaryAddress, " South ", " S ");
newPrimaryAddress=tranwrd(newPrimaryAddress, " North ", " N ");
newPrimaryAddress=tranwrd(newPrimaryAddress, " East ", " E ");
newPrimaryAddress=tranwrd(newPrimaryAddress, " West ", " W ");

newPrimaryAddress=tranwrd(newPrimaryAddress, " Road", " Rd");
newPrimaryAddress=tranwrd(newPrimaryAddress, "Street", "St");
newPrimaryAddress=tranwrd(newPrimaryAddress, "Highway", "Hwy ");
newPrimaryAddress=tranwrd(newPrimaryAddress, "Lane", "Ln");
newPrimaryAddress=tranwrd(newPrimaryAddress, "Avenue", "Ave");
newPrimaryAddress=tranwrd(newPrimaryAddress, "Drive", "Dr");
newPrimaryAddress=tranwrd(newPrimaryAddress, "Boulevard", "Blvd");
newPrimaryAddress=tranwrd(newPrimaryAddress, " Court ", " Ct ");
newPrimaryAddress=tranwrd(newPrimaryAddress, " Parkway ", " Pkwy ");
newPrimaryAddress=tranwrd(newPrimaryAddress, " Place ", " Pl ");
newPrimaryAddress=tranwrd(newPrimaryAddress, " Square ", " Sq ");
```

```
newPrimaryAddress=tranwrd(newPrimaryAddress, " Circle ", " Cir ");

newPrimaryAddress=tranwrd(newPrimaryAddress, " St Rd ", " State Road ");
newPrimaryAddress=tranwrd(newPrimaryAddress, " Sr ", " State Road ");
```

6. Pattern matching
   a. Conventional street address pattern

```
pattstreetaddr="/(\d+)(\s+\w+)+ (St|Ave|Dr|Rd|Pkwy|Blvd|Ct|Ln|Way|Pl|Cir) /i";
pattstreetid=prxparse(pattstreetaddr);

retain pattstreetid;

call prxsubstr(pattstreetid, newPrimaryAddress, position, length);

if position ne 0 then do;
pPrimaryAddress=substr(newPrimaryAddress, position, length);
end;
```

   b. Unconventional street address pattern (example: 1001 State Road 35)

```
pattstreetaddr2="/(\d+)(\s+\w+)+/i";
pattstreetid2=prxparse(pattstreetaddr2);

retain pattstreetid2;

call prxsubstr(pattstreetid2, newPrimaryAddress, position2, length2);

if pPrimaryAddress='' and position2 ne 0 then do;
pPrimaryAddress=substr(newPrimaryAddress, position2, length2);
end;
```

Practice City Variable

1. Formatting practice city variables as proper case

```
newPrimaryCity=propcase(PrimaryPracticeCity);
```

2. Compressing new practice city variable to remove extra spaces, dashes, commas and periods

```
newPrimaryCity=compress(newPrimaryCity,,'adks');
```

3. Remove commonly found state names from variable

```
find2=index(newPrimaryCity," Indiana ");
if find2 gt 1 then newPrimaryCity=substr(newPrimaryCity,1,find2-1);

find2=index(newPrimaryCity," In ");
if find2 gt 1 then newPrimaryCity=substr(newPrimaryCity,1,find2-1);

find2=index(newPrimaryCity," Michigan ");
if find2 gt 1 then newPrimaryCity=substr(newPrimaryCity,1,find2-1);
```

Practice ZIP code variable

1. Converting ZIP code to a 5-digit number

```
newPrimaryZip=input(PrimaryPracticeZip, 5.);
```

License Address Cleaning Code:

License Address

1. Cleaning addr_line_1 variable

```
new_addr_1=propcase(addr_line_1);

new_addr_1=tranwrd(new_addr_1, "Drive", "Dr.");
new_addr_1=tranwrd(new_addr_1, "Street", "St.");
new_addr_1=tranwrd(new_addr_1, "Road", "Rd.");
new_addr_1=tranwrd(new_addr_1, "Court", "Ct.");
new_addr_1=tranwrd(new_addr_1, "Place", "Pl.");
new_addr_1=tranwrd(new_addr_1, "Lane ", "Ln.");
new_addr_1=tranwrd(new_addr_1, "Drive", "Dr.");
new_addr_1=tranwrd(new_addr_1, " Cir ", "Circle");
new_addr_1=tranwrd(new_addr_1, "Avenue", "Ave.");

new_addr_1=tranwrd(new_addr_1, "Po ", "P.O. ");
new_addr_1=tranwrd(new_addr_1, "P O ", "P.O. ");

addr_county=tranwrd(addr_county, "Saint", "St.");
addr_county=tranwrd(addr_county, "St ", "St. ");

new_addr_1=tranwrd(new_addr_1, "State Rd.", "State Road");
new_addr_1=tranwrd(new_addr_1, " Sr ", " State Road ");
new_addr_1=tranwrd(new_addr_1, "County Rd.", "County Road");
new_addr_1=tranwrd(new_addr_1, " Cr ", " County Road ");
new_addr_1=tranwrd(new_addr_1, " Xing", " Crossing");

if index(new_addr_1, "#") gt 0 then new_addr_1=substr(new_addr_1, 1,
index(new_addr_1, "#")-1);
if index(new_addr_1, "Apt ") gt 0 then new_addr_1=substr(new_addr_1, 1,
index(new_addr_1, "Apt ")-1);
if index(new_addr_1, "Apt.") gt 0 then new_addr_1=substr(new_addr_1, 1,
index(new_addr_1, "Apt.")-1);
if index(new_addr_1, "Apartment") gt 0 then new_addr_1=substr(new_addr_1,
1, index(new_addr_1, "Apartment")-1);
if index(new_addr_1, "Suite") gt 0 then new_addr_1=substr(new_addr_1, 1,
index(new_addr_1, "Suite")-1);
if index(new_addr_1, "Ste ") gt 0 then new_addr_1=substr(new_addr_1, 1,
index(new_addr_1, "Ste ")-1);
if index(new_addr_1, "Ste. ") gt 0 then new_addr_1=substr(new_addr_1, 1,
index(new_addr_1, "Ste. ")-1);
if index(new_addr_1, "Unit ") gt 0 then new_addr_1=substr(new_addr_1, 1,
index(new_addr_1, "Unit ")-1);
```

2. Cleaning addr_line_2 variable

```
new_addr_2=propcase(addr_line_2);

new_addr_2=tranwrd(new_addr_2, "Drive", "Dr.");
new_addr_2=tranwrd(new_addr_2, "Street", "St.");
new_addr_2=tranwrd(new_addr_2, "Road", "Rd.");
new_addr_2=tranwrd(new_addr_2, "Court", "Ct.");
new_addr_2=tranwrd(new_addr_2, "Place", "Pl.");
new_addr_2=tranwrd(new_addr_2, "Lane ", "Ln.");
```

```
new_addr_2=tranwrd(new_addr_2, "Drive", "Dr.");
new_addr_2=tranwrd(new_addr_2, " Circle ", "Cir.");
new_addr_2=tranwrd(new_addr_2, "Avenue", "Ave.");
new_addr_2=tranwrd(new_addr_2, "Terrace", "Ter.");
new_addr_2=tranwrd(new_addr_2, "Boulevard", "Blvd.");

new_addr_2=tranwrd(new_addr_2, " Po ", " P.O. ");
new_addr_2=tranwrd(new_addr_2, "P O", "P.O.");
new_addr_2=tranwrd(new_addr_2, "Pobox", "P.O. Box ");

new_addr_2=tranwrd(new_addr_2, "State Rd.", "State Road");
new_addr_2=tranwrd(new_addr_2, " Sr ", " State Road ");
new_addr_2=tranwrd(new_addr_2, "County Rd.", "County Road");
new_addr_2=tranwrd(new_addr_2, "County Rd ", "County Road ");
new_addr_2=tranwrd(new_addr_2, " Cr ", " County Road ");

if index(new_addr_2, "#") gt 0 then new_addr_2=substr(new_addr_2, 1,
index(new_addr_2, "#")-1);
if index(new_addr_2, "Apt ") gt 0 then new_addr_2=substr(new_addr_2, 1,
index(new_addr_2, "Apt ")-1);
if index(new_addr_2, "Apt.") gt 0 then new_addr_2=substr(new_addr_2, 1,
index(new_addr_2, "Apt.")-1);
if index(new_addr_2, "Apt-") gt 0 then new_addr_2=substr(new_addr_2, 1,
index(new_addr_2, "Apt-")-1);
if index(new_addr_2, "Apartment") gt 0 then new_addr_2=substr(new_addr_2,
1, index(new_addr_2, "Apartment")-1);
if index(new_addr_2, "Suite") gt 0 then new_addr_2=substr(new_addr_2, 1,
index(new_addr_2, "Suite")-1);
if index(new_addr_2, "Ste ") gt 0 then new_addr_2=substr(new_addr_2, 1,
index(new_addr_2, "Ste ")-1);
if index(new_addr_2, "Ste. ") gt 0 then new_addr_2=substr(new_addr_2, 1,
index(new_addr_2, "Ste. ")-1);
if index(new_addr_2, "Unit ") gt 0 then new_addr_2=substr(new_addr_2, 1,
index(new_addr_2, "Unit ")-1);
```

3. Creating new license address variable

```
street_pat1=prxparse("/(\d+)(\s+\w+)+
(St|Ave|Dr|Rd|Pkwy|Blvd|Ct|Ln|Way|Pl|Circle) /i");
street_pat2=prxparse("/(\d+)(\s+\w+)+/i");

call prxsubstr(street_pat1, new_addr_1, position1, length1);
if position1 gt 0 then new_license_address=new_addr_1;

call prxsubstr(street_pat2, new_addr_1, position2, length2);
if position2 gt 0 then new_license_address=new_addr_1;

call prxsubstr(street_pat1, new_addr_2, position3, length3);
if position3 gt 0 then new_license_address=new_addr_2;

call prxsubstr(prxparse(street_pat2 , new_addr_2, position4, length4);
if position4 gt 0 then new_license_address=new_addr_2;

if new_license_address='' and index(new_addr_1, "Box") gt 0 then
new_license_address=new_addr_1;
```

```
   if new_license_address='' and index(new_addr_2, "Box") gt 0 then
   new_license_address=new_addr_2;
```

Formatting License Address City

```
   new_addr_city=propcase(addr_city);
```

Formatting ZIP code

```
   new_addr_zip=input(addr_zipcode, 9.);
```