

**Pulse oximetry and supplemental oxygen supplement**

Appendix 1:	Extracting and cleaning supplemental oxygen	Page 2
Table 1:	CDW vital signs domain	Page 3
Figure 1:	Consort flow diagram	Page 4
Table 2a:	Different values for VitalResultNumeric and supplemental oxygen	Page 6
Table 2b:	Same values for VitalResultNumeric and supplemental oxygen	Page 6
Appendix 2:	Data extraction and standardization code of pulse oximetry and supplemental oxygen	Page 7






## Appendix 1. Extracting and cleaning supplemental oxygen

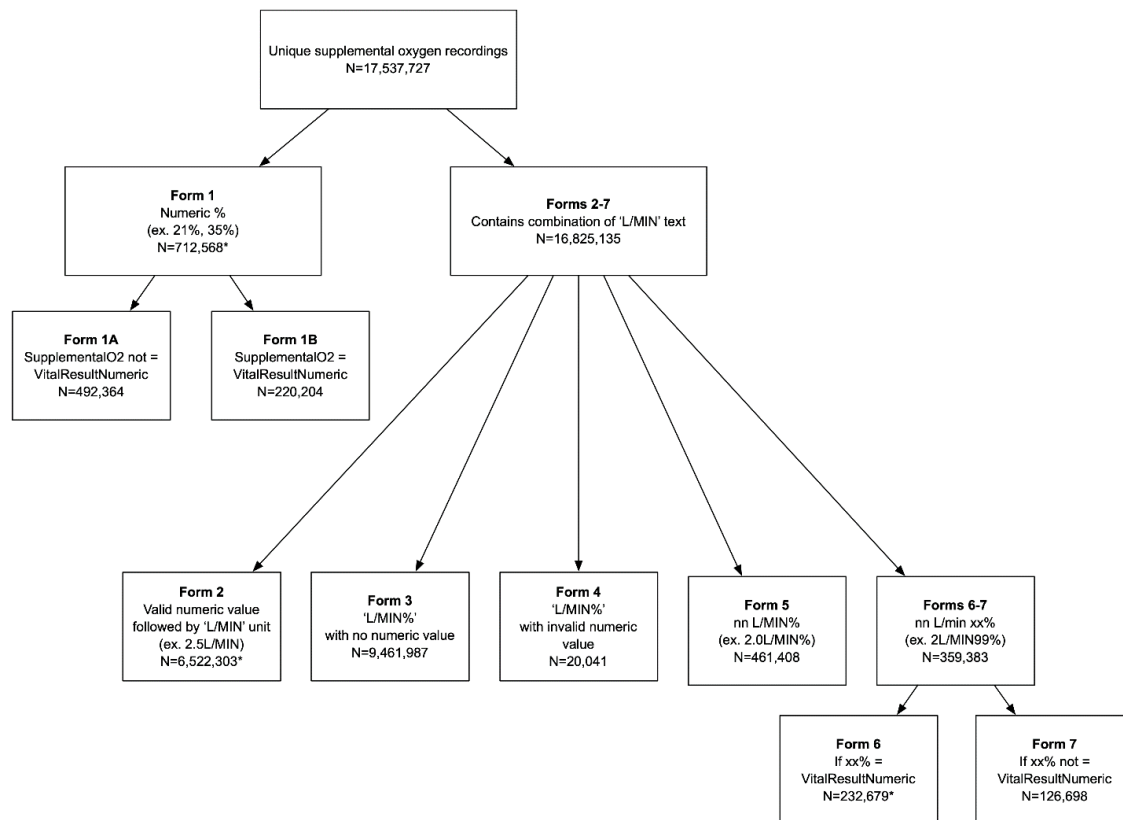
Our goal was to extract and standardize supplemental oxygen data from the VA's Corporate Data Warehouse (CDW). We used data from two fields in the CDW's Vital Signs Domain: (1) the supplemental oxygen text field ('SupplementalO2') and (2) the vital results numeric field ('VitalResultNumeric') (see Table 1).

As illustrated in Table 1, the raw data for supplemental oxygen were recorded using a variety of notations (e.g. 3.0 l/min %, l/min %, 2 l/min, 35%). We binned these notations into 7 separate forms following a standardization process shown in Figure 1. In total, we extracted 17,537,727 unique supplemental oxygen values from the VA's CDW for the 2013-2017 Veteran's Affairs Patient Database (VAPD) and placed each value into one of 7 smaller datasets according to the value of the supplemental oxygen recording and the vital results numeric field.

In the remaining sections of the appendix, we provide our standardization strategy for each of these 7 forms. Standardization of the supplemental oxygen text field resulted in a usable variable for clinical research: O2\_LPM (supplemental oxygen measured in liters per minute).

**Table 1.** CDW vital signs domain

 VitalResult	 VitalResultNumeric	 Systolic	 Diastolic	 Supplemental IO2
95	95.0000	.	.	3.0 l/min %
94	94.0000	.	.	l/min %
94	94.0000	.	.	l/min %
93	93.0000	.	.	l/min %
98	98.0000	.	.	l/min %
93	93.0000	.	.	l/min %
92	92.0000	.	.	l/min %
94	94.0000	.	.	l/min %
100	100.0000	.	.	l/min %
93	93.0000	.	.	3.0 l/min
92	92.0000	.	.	2 l/min
100	100.0000	.	.	5 l/min
100	100.0000	.	.	l/min %
95	95.0000	.	.	l/min %
95	95.0000	.	.	2.0 l/min
97	97.0000	.	.	3 l/min
91	91.0000	.	.	3 l/min
91	91.0000	.	.	l/min %
100	100.0000	.	.	35%
94	94.0000	.	.	3 l/min
99	99.0000	.	.	l/min %
98	98.0000	.	.	l/min %
96	96.0000	.	.	l/min %
96	96.0000	.	.	l/min %
94	94.0000	.	.	l/min %
65	65.0000	.	.	l/min %
97	97.0000	.	.	2 l/min

**Figure 1.** Consort flow diagram of supplemental oxygen text fields, 2013-2017

**Figure 1 Legend**

Form 1 consisted of 712,568 supplemental oxygen recordings that contained a numeric value followed by the '%' character (e.g., 21%, 35%, 55%, 95%). For these observations, we used the formula  $LPM = (FiO_2 - 0.21) / 0.03$  to obtain liters per minute of oxygen (O2\_LPM). Because Form 1 contained values in the supplemental oxygen field that were not always consistent with values in the VitalResultNumeric field (Tables 2a and 2b), we used two approaches for standardization. When supplemental oxygen values differed from VitalResultNumeric values (N=492,364), we considered the supplemental oxygen field to be the valid measure of FiO<sub>2</sub>, (Table 2a). However, when supplemental oxygen values were equal to the VitalResultNumeric values (N=220,204), we considered these to be SpO<sub>2</sub> values that were miscoded in the SupplementalO<sub>2</sub> field and excluded them (Table 2b).

Form 2 consisted of 6,522,303 supplemental oxygen recordings with a numeric value followed by the 'L/MIN' unit (e.g. 2.5L/MIN). These recordings reflected the number of liters per minute of oxygen that patients were receiving at the time of the recording and were considered valid numeric supplemental oxygen values. Values in Form 2 required the numeric digits to be extracted from the string of text characters (e.g. separating '2.5' from 'L/MIN').

Form 3 consisted of 9,461,987 supplemental oxygen recordings that contained the text characters 'L/MIN%' with no numeric value. This form was coded as '0' to indicate that patients were not receiving supplemental oxygen.

Form 4 contained 20,041 observations with the text character 'L/MIN%' and a numeric value. However, the numeric value for this form represented an invalid supplemental oxygen recording (e.g. L/MIN95%) and was dropped.

Form 5 included 461,408 valid numeric supplemental oxygen recordings followed by 'L/MIN%' (e.g. 2.5L/MIN%). For these observations, we extracted the numeric values from the string characters (e.g. separating '2.5' from 'L/MIN%').

Form 6 consisted of 232,679 observations in which the supplemental oxygen value was recorded in the form 'xx%' and was consistent with the numeric value recorded in the 'VitalResultNumeric' field. Supplemental oxygen values in this form appear to contain values for both pulse oximetry and Supplemental oxygen in the text string. When the number before a '%' unit was equal to the value in the VitalResultNumeric field, then we extracted the numeric digits before the '%' unit and coded them as Supplemental oxygen values.

Form 7 consisted of 126,698 observations in which 'xx%' was linked to a numeric value that was not consistent with the numeric value in the 'VitalResultNumeric' field. Supplemental oxygen recordings in this form appear to contain values for both pulse oximetry and supplemental oxygen in the text string. Unlike form 5, when the number before a '%' unit was NOT equal to the VitalResultNumeric value, these were considered incoherent data and were dropped.

\* A total of 24 observations were dropped from Form 1 due to invalid liters per minute of oxygen of less than 0 (O2\_LPM<0). 13 observations were dropped from Form 2 due to missing values for liters per minute of oxygen. 6 observations were dropped from Form 6 due to data entry errors in the text character field for SupplementalO<sub>2</sub>. Exclusions are noted in the data extraction and standardization code of Appendix 2.

**Table 2a.** Different values for VitalResultNumeric and Supplemental oxygen

vital_date	year	VitalResultNumeric	Supplemental IO2	SpO2	O2_LPM	incoherent
06/10/2016	2016	88.0000	50%	88	9.6666666667	0
12/20/2014	2014	97.0000	23%	97	0.6666666667	0
01/24/2015	2015	97.0000	35%	97	4.6666666667	0
02/14/2015	2015	93.0000	50%	93	9.6666666667	0
06/06/2013	2013	94.0000	50%	94	9.6666666667	0
09/19/2015	2015	94.0000	35%	94	4.6666666667	0
05/30/2014	2014	95.0000	50%	95	9.6666666667	0
02/28/2017	2017	97.0000	21%	97	0	0

**Table 2b.** Same values for VitalResultNumeric and Supplemental oxygen

vital_date	year	VitalResultNumeric	Supplemental IO2	SpO2	O2_LPM	incoherent
10/20/2015	2015	97.0000	97%	97	25.3333333333	1
02/21/2013	2013	96.0000	96%	96	25	1
10/01/2013	2013	97.0000	97%	97	25.3333333333	1
09/02/2014	2014	96.0000	96%	96	25	1
01/14/2016	2016	95.0000	95%	95	24.6666666667	1
02/25/2016	2016	92.0000	92%	92	23.6666666667	1
03/16/2016	2016	97.0000	97%	97	25.3333333333	1
06/01/2015	2015	97.0000	97%	97	25.3333333333	1
03/13/2013	2013	98.0000	98%	98	25.6666666667	1
06/03/2013	2013	97.0000	97%	97	25.3333333333	1
06/18/2013	2013	96.0000	96%	96	25	1
05/12/2014	2014	97.0000	97%	97	25.3333333333	1
08/22/2013	2013	99.0000	99%	99	26	1
05/07/2014	2014	100.0000	100%	100	26.3333333333	1
-----	----	-----	---	--	-----	.

## Appendix 2. Data extraction and standardization code of pulse oximetry and supplemental oxygen

### SQL code for data extraction from CDW

```

SELECT a.Sta3n, a.PatientSID, a.vitalSignTakenDateTime, a.VitalResultNumeric, a.SupplementalO2, a.VitalTypeSID,
      B.VitalType, c.patienticn
into #NewPulseOx20132014_SW20200609
FROM [Src].[Vital_VitalSign] as A
left JOIN [CDWWORK].[Dim].[VitalType] as B ON A.VitalTypeSID =B.VitalTypeSID
left join Src.SPatient_SPatient c on a.patientsid=c.patientsid
WHERE (a.VitalSignTakenDateTime >= '20170101' and a.VitalSignTakenDateTime < '20180101') /*times pulled*/
      AND (a.CohortName='Cohort20200427') /*new cohort updated*/
AND (
/*1) pull all where SupplementalO2 not null, type=PulseOX and VitalResultNumeric > 0 and VitalResultNumeric <= 100 */
(a.SupplementalO2 <> 'NULL' and b.VitalType='PULSE OXIMETRY' and a.VitalResultNumeric > 0 and a.VitalResultNumeric <= 100)

/*OR 2) also pull pulseox, and SupplementalO2 is NULL and VitalResultNumeric > 0 and VitalResultNumeric <= 100*/
OR (a.SupplementalO2 is NULL and b.VitalType='PULSE OXIMETRY' and a.VitalResultNumeric > 0 and a.VitalResultNumeric <= 100)
)

```

### SAS code for data cleaning and standardization

#### Step 1. Before data cleaning, standardize the supplemental oxygen field by changing all its values into capital letters and remove the extra spaces.

```

/*clean SupplementalO2 to take out extra space and turn everything into CAPS*/
DATA PulseOx20132017_v2 (compress=yes rename=SupplementalO2_v3=SupplementalO2); /*n=17469574*/
SET PulseOx20132017_v1;
SupplementalO2_v2=upcase(SupplementalO2); /*turn all units into uppercase*/
SupplementalO2_v3=compress(SupplementalO2_v2); /*removes all blanks*/
drop SupplementalO2_v2 SupplementalO2;
/*drop the original SupplementalO2 field and rename SupplementalO2_v3 as SupplementalO2*/
RUN;

```

#### Step 2. Clean Supplemental oxygen field by separating them out into different forms.

##### Form 1 Code

```

PROC SQL;
CREATE TABLE pulse_noLmin (compress=yes) AS
SELECT *
FROM PulseOx20132017_v2
WHERE SupplementalO2 not like '%L/MIN%';
QUIT;

PROC MEANS DATA=pulse_noLmin MIN MAX MEAN MEDIAN Q1 Q3;
VAR VitalResultNumeric;
RUN;

PROC FREQ DATA=pulse_noLmin order=freq;
TABLE SupplementalO2;
RUN;

/*remove the % sign and change to numeric value and look at descriptive*/
DATA pulse_noLmin2 (compress=yes);
SET pulse_noLmin;
supple_char=compress(SupplementalO2,'%');
supple_num=input(supple_char,3.);
RUN;

```

```

/*check if suppl_num NE (not equal) VitalResultNumeric*/
data checking;
set pulse_noLmin2;
if suppl_num NE VitalResultNumeric;
run;

PROC FREQ DATA=pulse_noLmin2;
TABLE suppl_num;
RUN;

PROC MEANS DATA=pulse_noLmin2 MIN /*0*/ MAX/*100*/ MEAN /*58.5*/ MEDIAN /*50*/ Q1 /*35*/ Q3 /*95*/;
VAR suppl_num;
RUN;

/*use Jack's conversions to get SpO2 and O2 (L/MIN), call this form1_20132017.
create new fields:SpO2=VitalResultNumeric, O2_LPM, and incoherent*/
DATA form1_20132017_test (compress=yes);
SET pulse_noLmin2;
SpO2=VitalResultNumeric;
O2_LPM=((suppl_num/100)-0.21)/0.03; /*turn into % first*/
if O2_LPM < 0 then delete; /*on 3/2/20 Jack said it's okay to delete*/
RUN;

PROC MEANS DATA=form1_20132017_test MIN MAX MEAN MEDIAN Q1 Q3;
VAR VitalResultNumeric SpO2 O2_LPM;
RUN;

PROC FREQ DATA=form1_20132017_test order=freq;
TABLE O2_LPM;
RUN;

/*there's form 1A and 1B: in general, SupplementalO2 should not equal VitalResultNumeric, Jack said this is
incoherent on 6/17/20*/
DATA form1A_20132017 (compress=yes)
form1B_20132017 (compress=yes);
SET form1_20132017_test;
if suppl_num=VitalResultNumeric then incoherent=1;
else incoherent=0;
if incoherent=0 then output form1A_20132017;
if incoherent=1 then output form1B_20132017;
keep patientcn PatientSID Sta3n VitalTypeSID VitalType obs vitalSignTakenDateTime vital_date year
VitalResultNumeric SupplementalO2 SpO2 O2_LPM incoherent;
RUN;

DATA form1A_20132017 (compress=yes);
retain patientcn PatientSID Sta3n VitalTypeSID VitalType obs vitalSignTakenDateTime vital_date year
VitalResultNumeric SupplementalO2 SpO2 O2_LPM incoherent;
set form1A_20132017;
RUN;

DATA form1B_20132017 (compress=yes);
retain patientcn PatientSID Sta3n VitalTypeSID VitalType obs vitalSignTakenDateTime vital_date year
VitalResultNumeric SupplementalO2 SpO2 O2_LPM incoherent;
set form1B_20132017;
RUN;
/***** END OF FORM 1 CODES *****/

```



## Form 2 Code

```

/*select everything else (non % only) not in pulse_noLmin dataset, and do a frequency to see what units it entail*/
PROC SQL;
CREATE TABLE pulse_lmin (COMPRESS=YES) AS
SELECT A.* FROM PulseOx20132017_v2 AS A
WHERE A.obs NOT IN (SELECT obs FROM pulse_noLmin);
QUIT;

PROC FREQ DATA=pulse_lmin order=freq;
TABLE SupplementalO2;
RUN;

/*look at those only with L/MIN units*/
PROC SQL;
CREATE TABLE pulse_lmin_only (compress=yes) AS
SELECT *
FROM pulse_lmin
WHERE SupplementalO2 like '%L/MIN' or SupplementalO2 like 'L/MIN';
QUIT;

/*check*/
PROC FREQ DATA=pulse_lmin_only order=freq;
TABLE SupplementalO2;
RUN;

/*there were some types for LPM, remove 'L/MIN', turn into numeric to look at descriptive*/
DATA pulse_lmin_only2 (compress=yes);
SET pulse_lmin_only;
if SupplementalO2='3EL/MIN' then SupplementalO2='3L/MIN';
if SupplementalO2='5LTL/MIN' then SupplementalO2='5L/MIN';
if SupplementalO2='4LITERSL/MIN' then SupplementalO2='4L/MIN';
if SupplementalO2='3LPML/MIN' then SupplementalO2='3L/MIN';
if SupplementalO2='4LPML/MIN' then SupplementalO2='4L/MIN';
if SupplementalO2='2L.NCL/MIN' then SupplementalO2='2L/MIN';
if SupplementalO2='2LPML/MIN' then SupplementalO2='2L/MIN';
if SupplementalO2='3LTL/MIN' then SupplementalO2='3L/MIN';
if SupplementalO2='RAL/MIN' then SupplementalO2='0L/MIN';
SupplementalO2_char=compress(SupplementalO2,'L/MIN'); /*first drop the characters, removes 'L/MIN' in units*/
SupplementalO2_num=input(SupplementalO2_char,3.); /*turn the numeric digits into numbers*/
RUN;

PROC FREQ DATA=pulse_lmin_only2 order=freq;
TABLE SupplementalO2;
RUN;

PROC MEANS DATA=pulse_lmin_only2 MIN MAX MEAN MEDIAN Q1 Q3;
VAR SupplementalO2_num VitalResultNumeric;
RUN;

/*call this form2*/
DATA form2_20132017_test (compress=yes);
SET pulse_lmin_only2;
SpO2=VitalResultNumeric;
incoherent=0;
O2_LPM=SupplementalO2_num;
if SupplementalO2='RAL/MIN' then O2_LPM=0; /*recode RAL/MIN =0, don't exclude*/
if O2_LPM =. then delete; /*3/2/20: Jack said it's ok to delete*/
RUN;

```

```

/*check*/
PROC MEANS DATA=form2_20132017_test MIN MAX MEAN MEDIAN Q1 Q3;
VAR VitalResultNumeric O2_LPM SpO2;
RUN;
PROC FREQ DATA=form2_20132017_test order=freq;
TABLE incoherentO2_LPM;
RUN;

DATA form2_20132017 (compress=yes);
retain patientcn PatientSID Sta3n VitalTypeSID VitalType obs vitalSignTakenDateTime vital_date year
VitalResultNumeric SupplementalO2 SpO2 O2_LPM incoherent;
SET form2_20132017_test;
drop SupplementalO2_char SupplementalO2_num;
RUN;

/***** END OF FORM 2 CODES *****/

```

### Form 3 & 4 code

```

/*look at those only with 'L/MIN%' unit*/
PROC SQL;
CREATE TABLE pulse_lminpercent_only (compress=yes) AS
SELECT *
FROM pulse_lmin
WHERE SupplementalO2 like 'L/MIN%';
QUIT;

PROC FREQ DATA= pulse_lminpercent_only order=freq;
TABLE SupplementalO2;
RUN;

/*2/3/20: get the descriptive on X%*/
/*first, get the last 4 digits*/
DATA last_4char_pulse_lminpercent (compress=yes);
SET pulse_lminpercent_only;
last_4=substr(SupplementalO2,length(SupplementalO2)-3,4);
RUN;

PROC FREQ DATA=last_4char_pulse_lminpercent order=freq;
TABLE last_4;
RUN;

/*from last character values, compress "MIN%" characters*/
DATA last_4char_pulse_lminpercent2 (compress=yes);
SET last_4char_pulse_lminpercent;
last_4char_v2=compress(last_4,'M');
last_4char_v3=compress(last_4char_v2,'I');
last_4char_v4=compress(last_4char_v3,'N');
last_4char_v5=compress(last_4char_v4,'%');
last_4num=input(last_4char_v5,3.);
RUN;

PROC FREQ DATA= last_4char_pulse_lminpercent2 order=freq;
TABLE last_4num last_4char_v5;
RUN;

DATA form3232 (compress=yes);
SET last_4char_pulse_lminpercent2;
if last_4num NE .;
if last_4num = VitalResultNumeric then equal=1; else equal=0;
RUN;

```

```

PROC FREQ DATA=form3232 order=freq;
TABLE equal;
RUN;

PROC MEANS DATA=form3232 MIN MAX MEAN MEDIAN Q1 Q3;
VAR last_4num VitalResultNumeric;
RUN;

/*call this form3*/
DATA form3_20132017_test (compress=yes);
SET last_4char_pulse_lminpercent2;
if SupplementalO2='L/MIN%';
SpO2=VitalResultNumeric;
O2_LPM=0;
incoherent=0;
RUN;

/*check Saturation_LPM forms*/
PROC MEANS DATA=form3_20132017_test MIN MAX MEAN MEDIAN Q1 Q3;
VAR SpO2 O2_LPM;
RUN;
PROC FREQ DATA=form3_20132017_test order=freq;
TABLE O2_LPM SupplementalO2;
RUN;

DATA form3_20132017 (compress=yes);
retain patientcn PatientSID Sta3n VitalTypeSID VitalType obs vitalSignTakenDateTime vital_date year
VitalResultNumeric SupplementalO2 SpO2 O2_LPM incoherent;
SET form3_20132017_test;
drop last_4last_4char_v2-last_4char_v5last_4num;
RUN;

/*2/24/20: Jack decided to label these as incoherent*/
DATA form4_20132017_test (compress=yes);
SET last_4char_pulse_lminpercent2;
if SupplementalO2 NE 'L/MIN%';
SpO2=VitalResultNumeric;
O2_LPM=.;
incoherent=1;
run;

/*Jack said he wants to look at distribution of the nn%*/
DATA form4_20132017 (compress=yes);
retain patientcn PatientSID Sta3n VitalTypeSID VitalType obs vitalSignTakenDateTime vital_date year
VitalResultNumeric SupplementalO2 SpO2 O2_LPM incoherent;
SET form4_20132017_test;
RUN;

PROC MEANS DATA=form4_20132017 MIN MAX MEAN MEDIAN Q1 Q3;
VAR last_4num;
RUN;

proc sgplot data=form4_20132017 noautolegend;
histogram last_4num;
density last_4num;
run;
/***** END OF FORM 3 & 4 CODES *****/

```

## Form 5 Code

Select what's left:

```
/*combine pulse_lminpercent_only+pulse_lmin_only+pulse_noLmin2 and see what is not in those datasets, do frequency check of those units*/
```

```
DATA all (compress=yes);
SET pulse_lminpercent_only pulse_lmin_only pulse_noLmin2;
RUN;
```

```
PROC SQL;
CREATE TABLE whatisleft (COMPRESS=YES) AS
SELECT A.* FROM PulseOx20132017_v2 AS A
WHERE A.obs not IN (SELECT obs FROM work.all);
QUIT;
```

```
PROC FREQ DATA=whatisleft order=freq;
TABLE SupplementalO2;
RUN;
```

```
/*1) separate out the ###*/ /*first, get the last 4 digits*/
DATA last_4charV1;
SET whatisleft;
last_4=substr(SupplementalO2,length(SupplementalO2)-3,4);
RUN;
```

```
/*look at list*/
PROC FREQ DATA=last_4charV1 order=freq;
TABLE last_4;
RUN;
```

```
/*2/3/20: look at descriptive for this 126,091 form with only nnL/MIN%*/
DATA LMINpercentonly_v1 (compress=yes);
SET last_4charV1;
if last_4 = 'MIN%';
RUN;
```

```
PROC MEANS DATA=LMINpercentonly_v1 MIN MAX MEAN MEDIAN Q1 Q3;
VAR VitalResultNumeric;
RUN;
```

```
/*get the first 4 characters*/
DATA LMINpercentonly_v2 (compress=yes);
SET LMINpercentonly_v1;
first_4=substr(SupplementalO2,1,4);
RUN;
```

```
PROC FREQ DATA=LMINpercentonly_v2 order=freq;
TABLE first_4;
RUN;
```

```
/*compress L,/,M*/
DATA LMINpercentonly_v3;
SET LMINpercentonly_v2;
first_4char_v2=compress(first_4,'M');
first_4char_v3=compress(first_4char_v2,'/');
first_4char_v4=compress(first_4char_v3,'L');
first_4num=input(first_4char_v4,3.);
RUN;
```

```
PROC MEANS DATA=LMINpercentonly_v3 MIN MAX MEAN MEDIAN Q1 Q3;
VAR first_4num VitalResultNumeric;
RUN;
```

```

/*call this form5*/
DATA form5_20132017_test (compress=yes);
SET LMINpercentonly_v3;
SpO2=VitalResultNumeric;
incoherent=0;
O2_LPM=first_4num;
RUN;

/*check*/
PROC MEANS DATA=form4_20132017_test MIN MAX MEAN MEDIAN Q1 Q3;
VAR SpO2 O2_LPM;
RUN;

PROC FREQ DATA=form5_20132017_test order=freq;
TABLE incoherent SpO2;
RUN;

DATA form5_20132017 (compress=yes);
retain patientcn PatientSID Sta3n VitalTypeSID VitalType obs vitalSignTakenDateTime vital_date year
VitalResultNumeric SupplementalO2 SpO2 O2_LPM incoherent;
SET form4_20132017_test;
drop last_4 first_4 first_4num first_4char_v2-first_4char_v4;
RUN;
/***** END OF FORM 5 CODES *****/

```

### Form 6 Code

```

/*Question: some last 4 characters have PRN%, NLT%, TO2%, exclude them?*/
PROC FREQ DATA=last_4charV1 order=freq;
TABLE last_4;
RUN;

/*compress 1 step at a time*/
DATA last_4charV2; /*n=6 were deleted*/
SET last_4charV1;
/*Jack ok with deleting these on 3/2/20*/
if last_4 in ('NNC%', 'INO%', 'NLT%', 'PRN%', 'TO2%') then delete;
last_4char_v2=compress(last_4, 'M');
last_4char_v3=compress(last_4char_v2, 'I');
last_4char_v4=compress(last_4char_v3, 'N');
last_4char_v5=compress(last_4char_v4, '%');
last_4num=input(last_4char_v5, 3.);
RUN;

PROC FREQ DATA=last_4charV2 order=freq;
TABLE last_4char_v5;
RUN;

PROC MEANS DATA=last_4charV2 MIN MAX MEAN MEDIAN Q1 Q3;
VAR last_4num;
RUN;

PROC FREQ DATA=last_4charV2;
TABLE last_4num;
RUN;

/*to see if there are any last_4num = VitalResultNumeric*/
DATA form_20132017_test;
SET last_4charV2;

```

```

if (last_4num NE .) and (last_4num = VitalResultNumeric);
RUN;

/*2/3/20: look at nn L/Min descriptive*/
DATA form_20132017_test2 (compress=yes);
SET form_20132017_test;
first_4=substr(SupplementalO2,1,4);
RUN;

PROC FREQ DATA=form_20132017_test2 order=freq;
TABLE first_4;
RUN;

DATA form_20132017_test3 (compress=yes);
SET form_20132017_test2;
first_4char_v2=compress(first_4,'M');
first_4char_v3=compress(first_4char_v2,'/');
first_4char_v4=compress(first_4char_v3,'L');
first_4num=input(first_4char_v4,3.);
RUN;

PROC MEANS DATA=form_20132017_test3 MIN MAX MEAN MEDIAN Q1 Q3;
VAR first_4num VitalResultNumeric;
RUN;

/*call this form6*/
DATA form6_20132017_test (compress=yes);
SET form_20132017_test3;
SpO2=VitalResultNumeric;
incoherent=0;
O2_LPM=first_4num;
RUN;

/*check*/
PROC MEANS DATA=form6_20132017_test MIN MAX MEAN MEDIAN Q1 Q3;
VAR O2_LPM SpO2;
RUN;

DATA form6_20132017 (compress=yes);
retain patientcn PatientSID Sta3n VitalTypeSID VitalType obs vitalSignTakenDateTIme vital_date year
VitalResultNumeric SupplementalO2 SpO2 O2_LPM incoherent;
SET form5_20132017_test;
drop last_4 last_4char_v2-last_4char_v5 last_4num first_4 first_4char_v2-first_4char_v4 first_4num;
RUN;

/***** END OF FORM 6 CODES *****/

```

### Form 7 Code

```

/*look at those that last_4num not equal to vitalresultnumeric*/
DATA val_form;
SET last_4charV2;
if (last_4num NE .) and (last_4num NE VitalResultNumeric);
RUN;

/*call this form6*/
DATA form7_20132017_test (compress=yes);
SET val_form;
SpO2=VitalResultNumeric;
incoherent=1;
O2_LPM=.;

```

```
RUN;

/*check*/
PROC MEANS DATA=form7_20132017_test MIN MAX MEAN MEDIAN Q1 Q3;
VAR O2_LPM SpO2;
RUN;
PROC FREQ DATA=form7_20132017_test order=freq;
TABLE incoherent;
RUN;

DATA form7_20132017 (compress=yes);
retain patientcn PatientSID Sta3n VitalTypeSID VitalType obs vitalSignTakenDateTme vital_date year
VitalResultNumeric SupplementalO2 SpO2 O2_LPM incoherent;
SET form7_20132017_test
drop last_4last_4char_v2-last_4char_v5last_4num;
RUN;

/***** END OF FORM 7 CODES *****/
```

### **Step 3. Combine all of the valid forms to make up a complete dataset**

```
/*combine all forms (cleaned and don't clean) and make sure the totals add up, dropped the incoherent forms*/
DATA PulseOx_20132017_07022020 (compress=yes);
SET DONT_CLEAN_FORM FORM1A_20132017 FORM2_20132017 FORM3_20132017 FORM5_20132017
FORM6_20132017;
RUN;
```