SBA Statistical Business Analyst using SAS

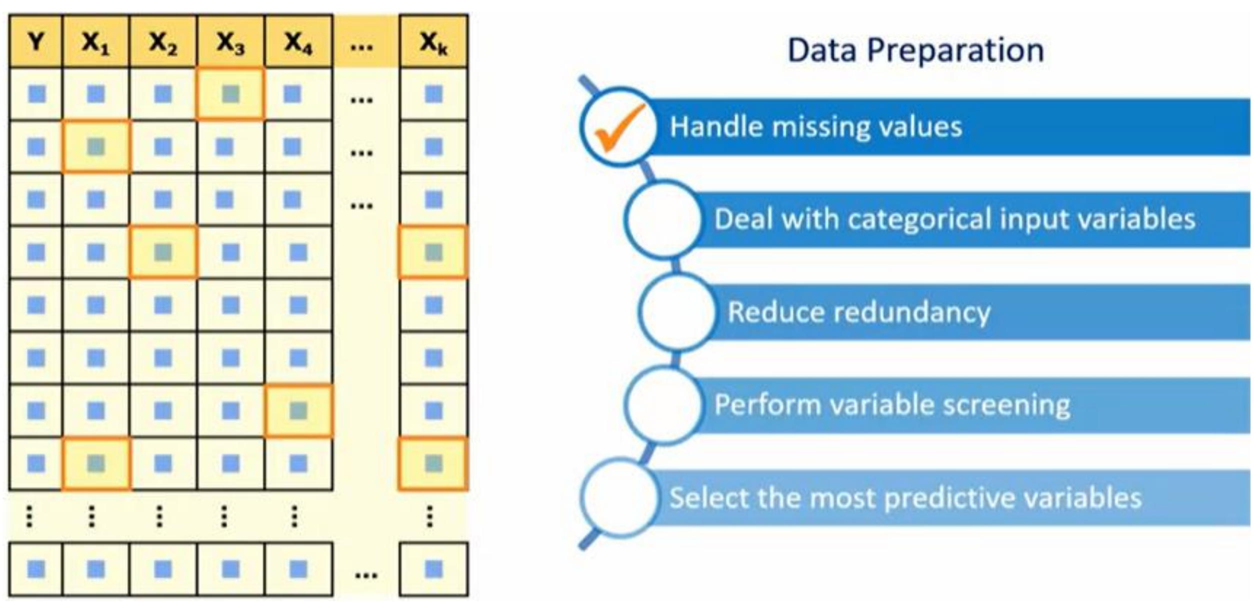
SBA3 Predictive Modeling with Logistic Regression

W3a Handling Missing Values

Data Preparation Overview

Handling Missing Values

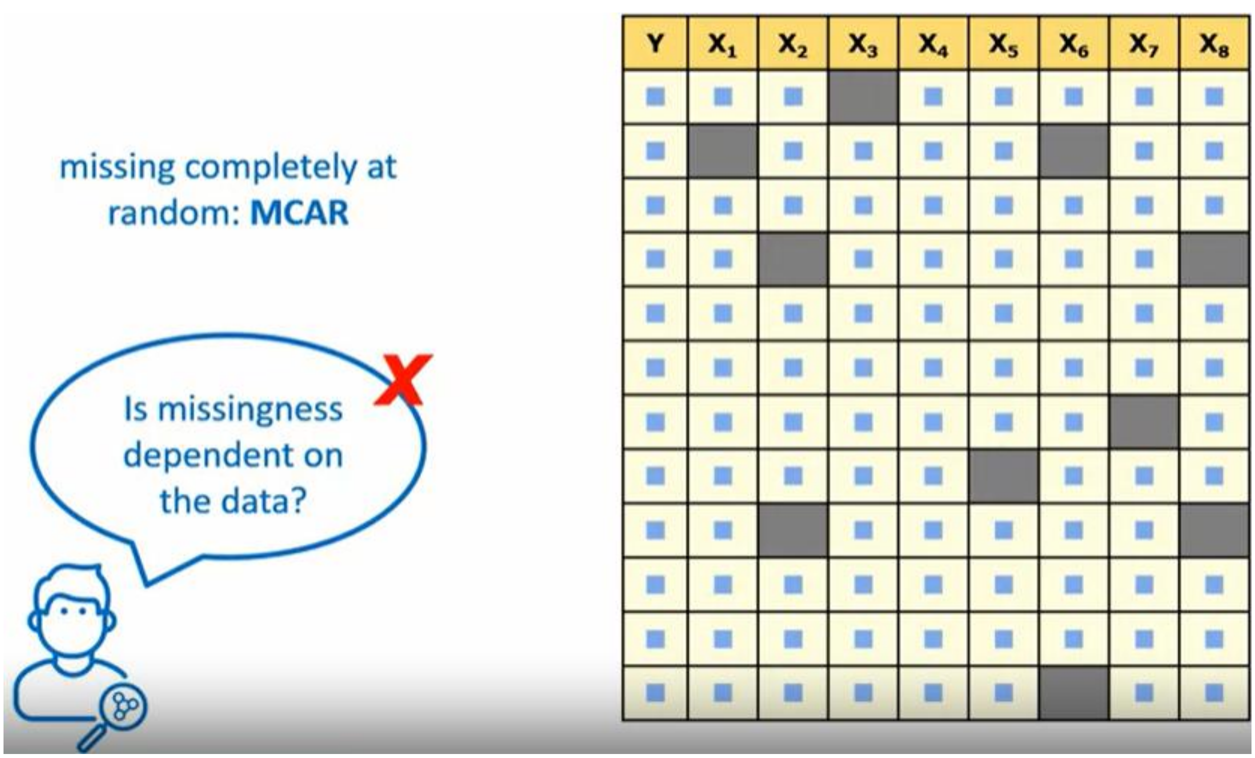
Introduction



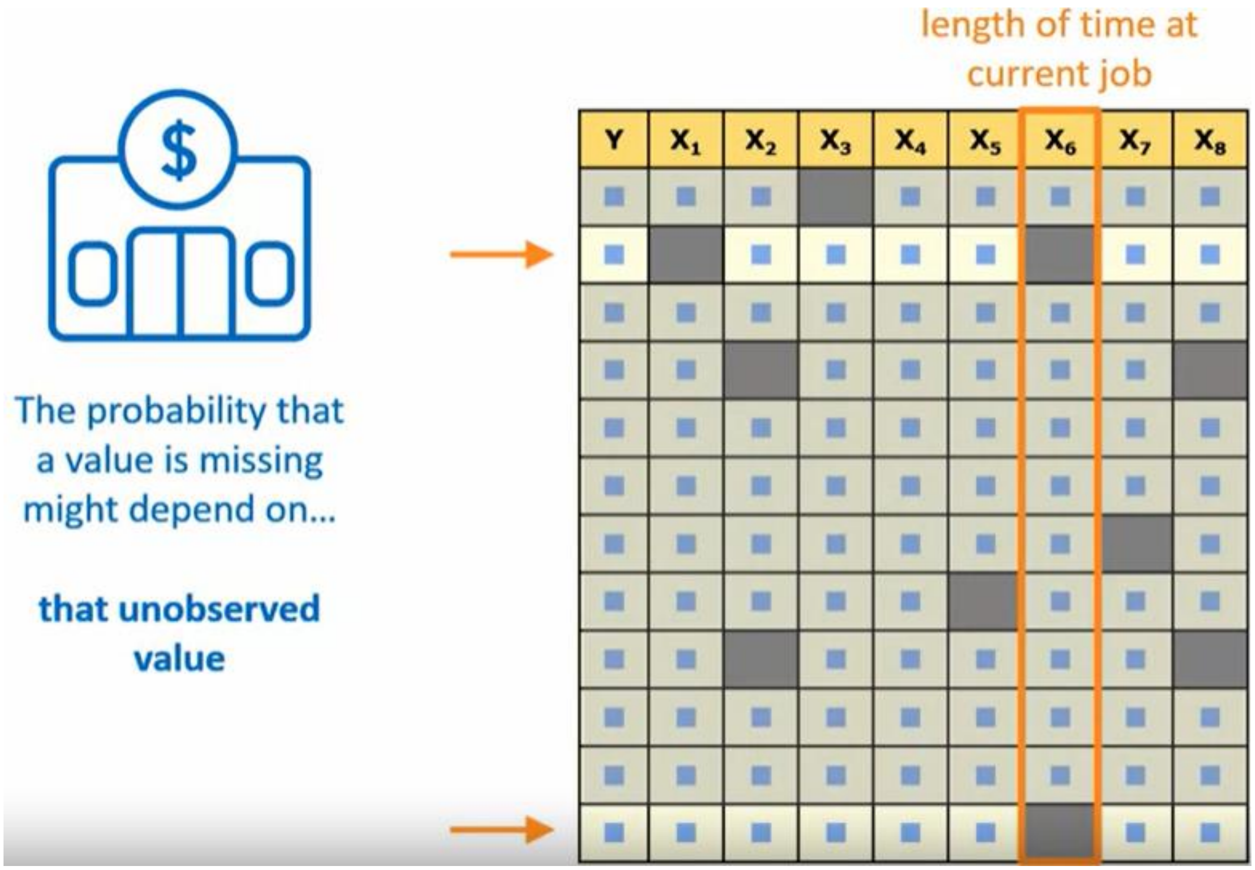
In this topic, you learn to do the following:

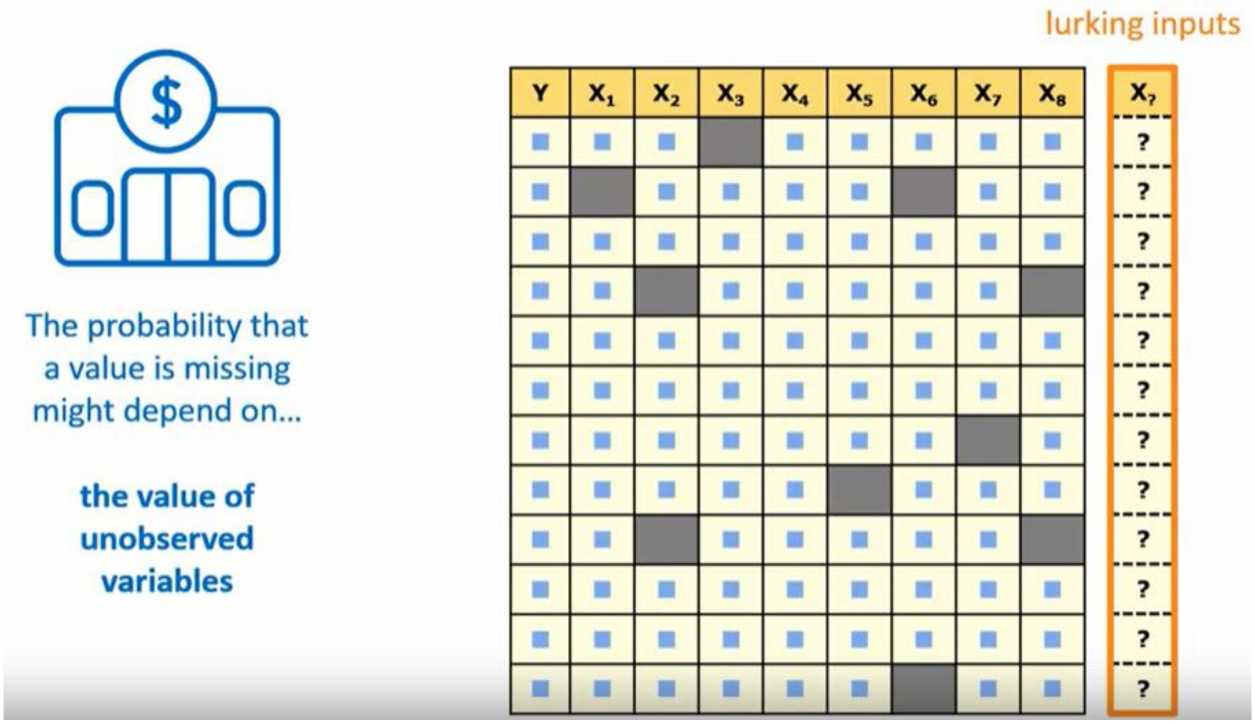
- identify the possible reasons for missing values
- identify the limitations of complete case analysis for predictive modeling
- identify common methods of missing value imputation
- identify the advantages of using missing value indicator variables
- impute missing values using the STDIZE procedure

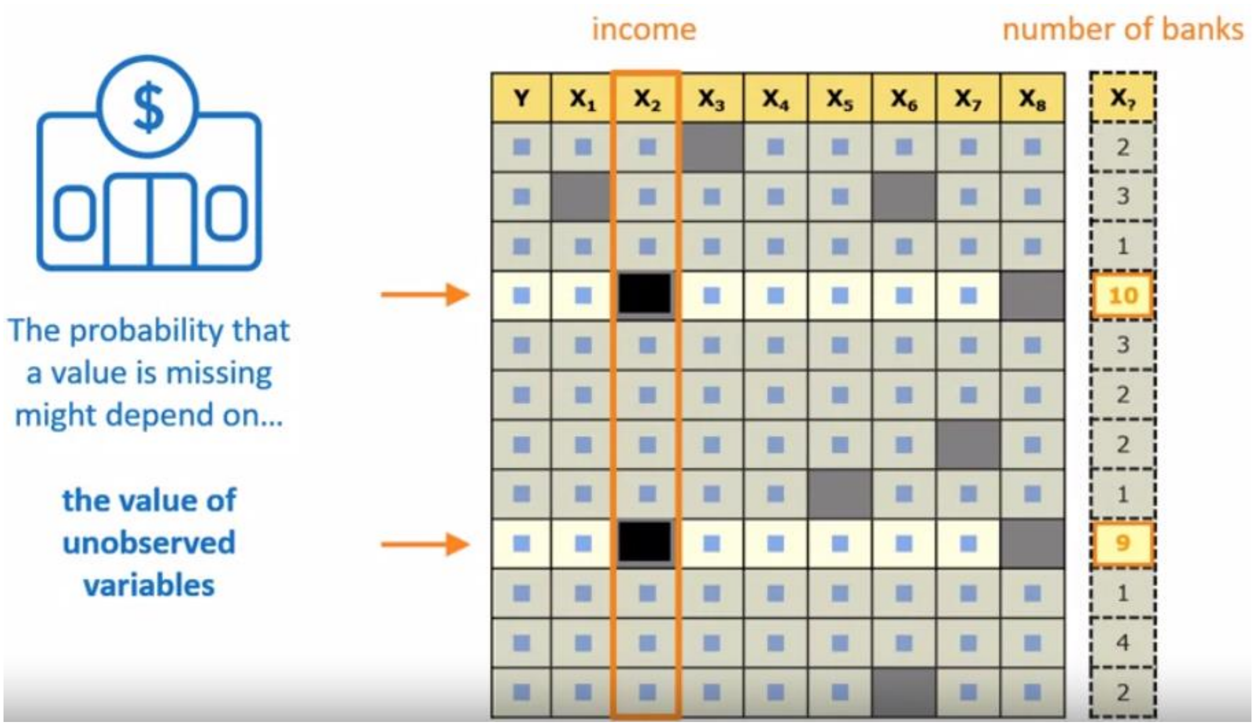
Reasons for Missing Data

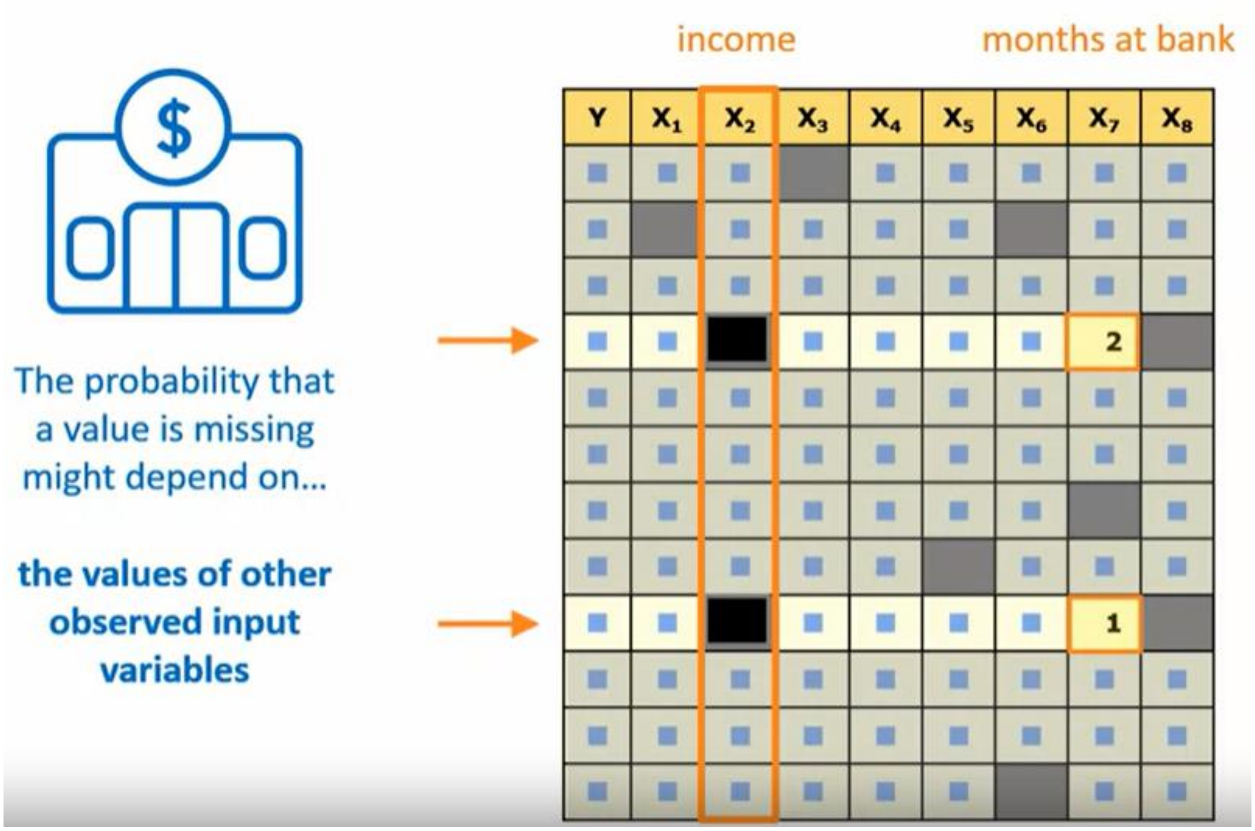


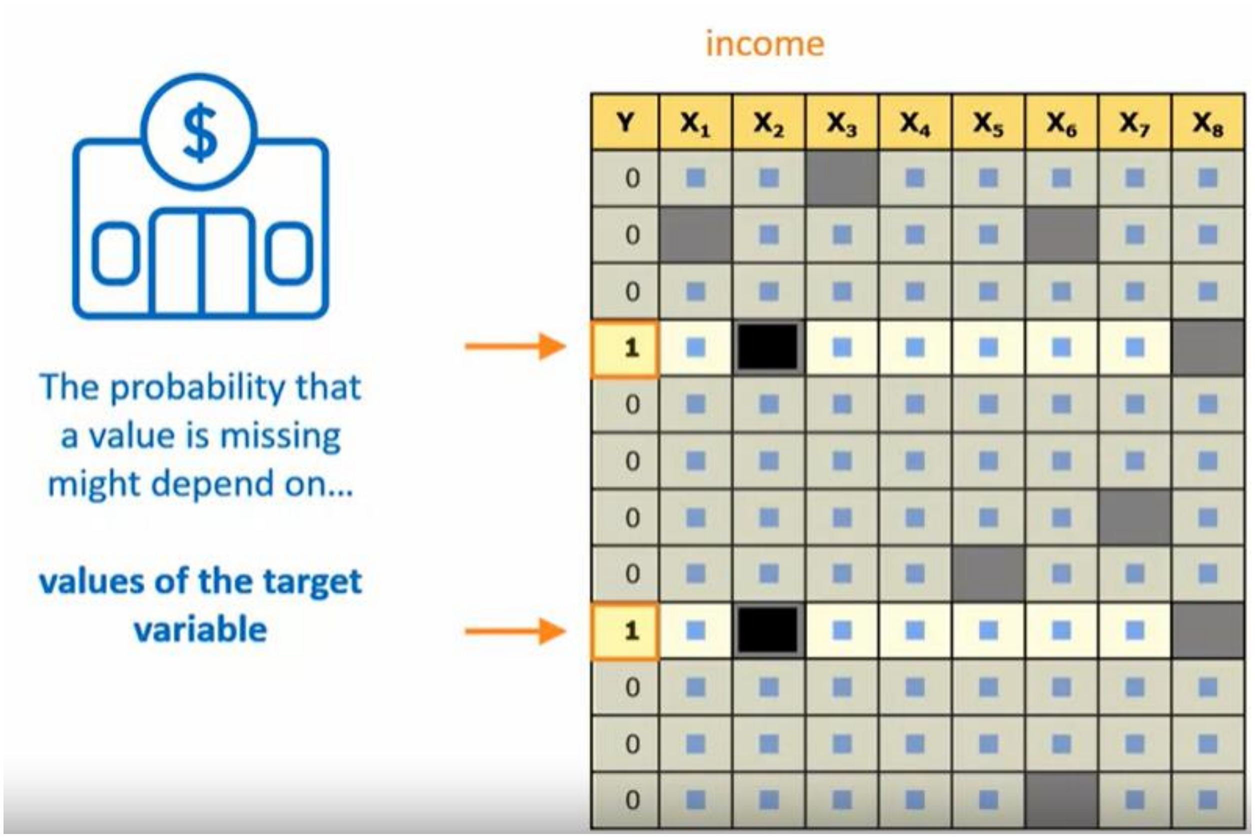

Randomized
Experimental Designs
missing completely at
random: **MCAR**


length of time at
current job
Y X_1 X_2 X_3 X_4 X_5 X_6 X_7 X_8
$
The probability that
a value is missing
might depend on...
**that unobserved
value**

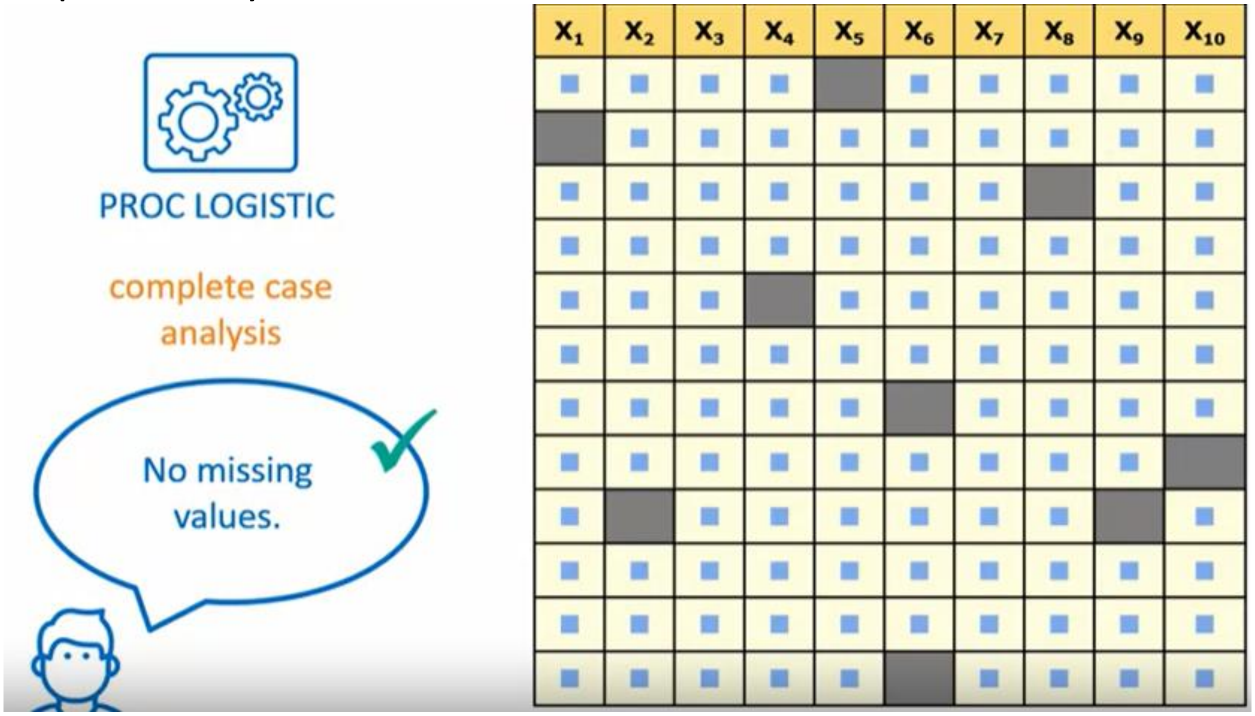

lurking inputs
The probability that a value is missing might depend on...
the value of unobserved variables
Y
X_1
X_2
X_3
X_4
X_5
X_6
X_7
X_8
$X_?$
?
?
?
?
?
?
?
?
?
?
?
?

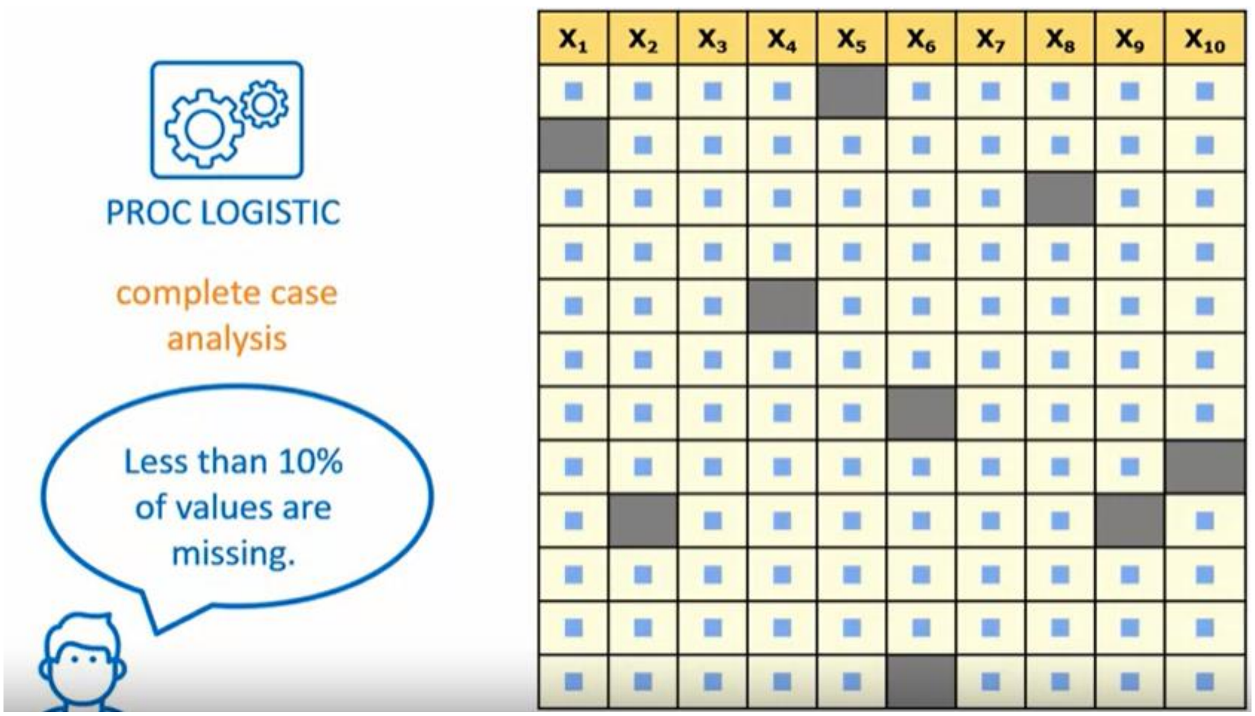

income
number of banks
The probability that a value is missing might depend on...
the value of unobserved variables
Y
X_1
X_2
X_3
X_4
X_5
X_6
X_7
X_8
$X_?$
2
3
1
10
3
2
2
1
9
1
4
2

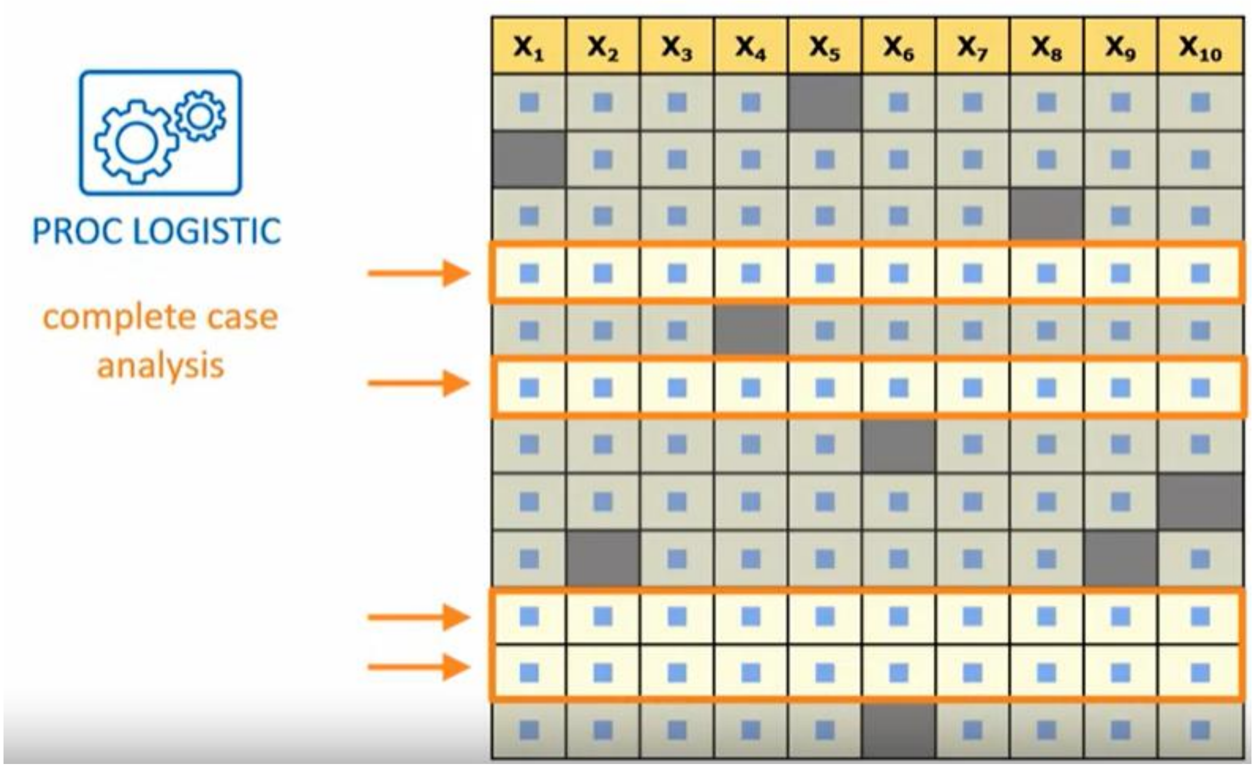

income
months at bank
Y
X_1
X_2
X_3
X_4
X_5
X_6
X_7
X_8
2
1
The probability that a value is missing might depend on...
the values of other observed input variables

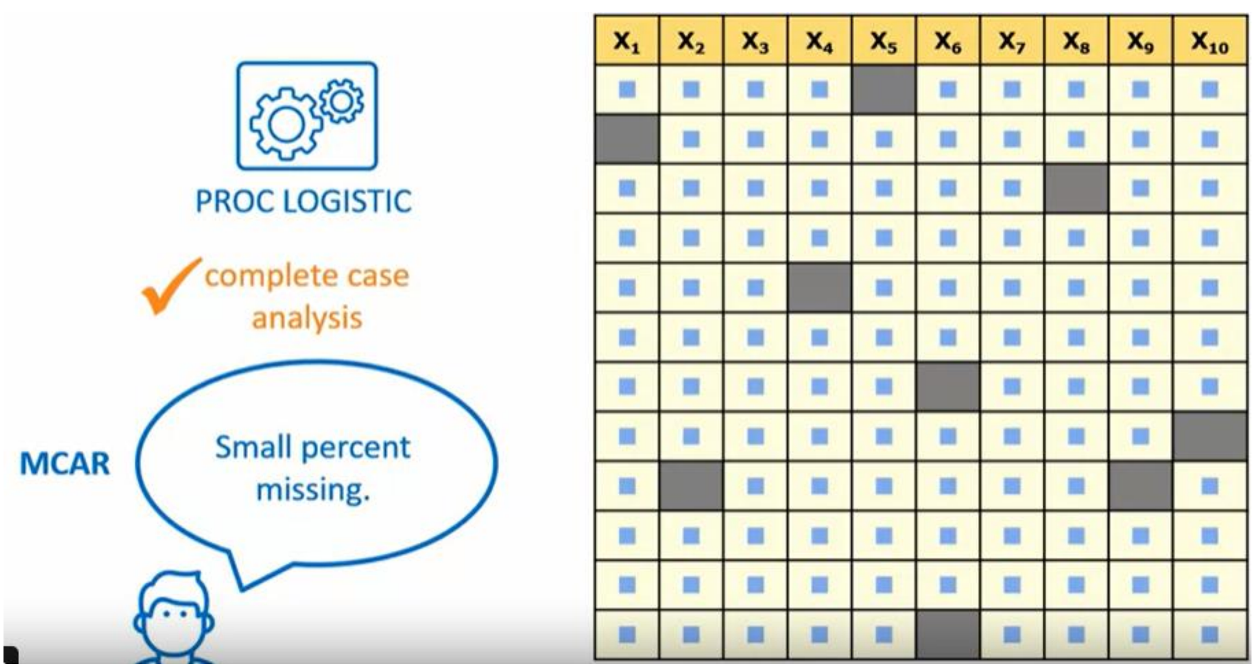

income
Y
X_1
X_2
X_3
X_4
X_5
X_6
X_7
X_8
0
0
0
1
0
0
0
0
1
0
0
0
The probability that a value is missing might depend on...
values of the target variable

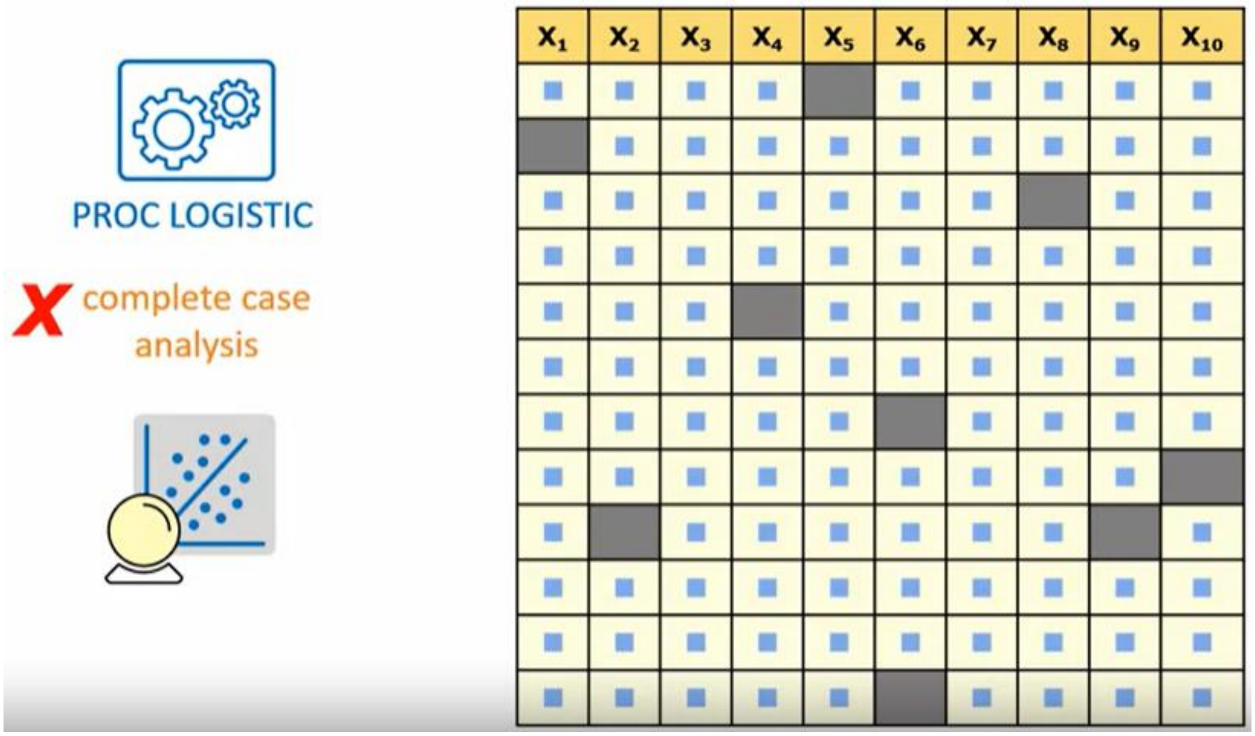
Complete Case Analysis



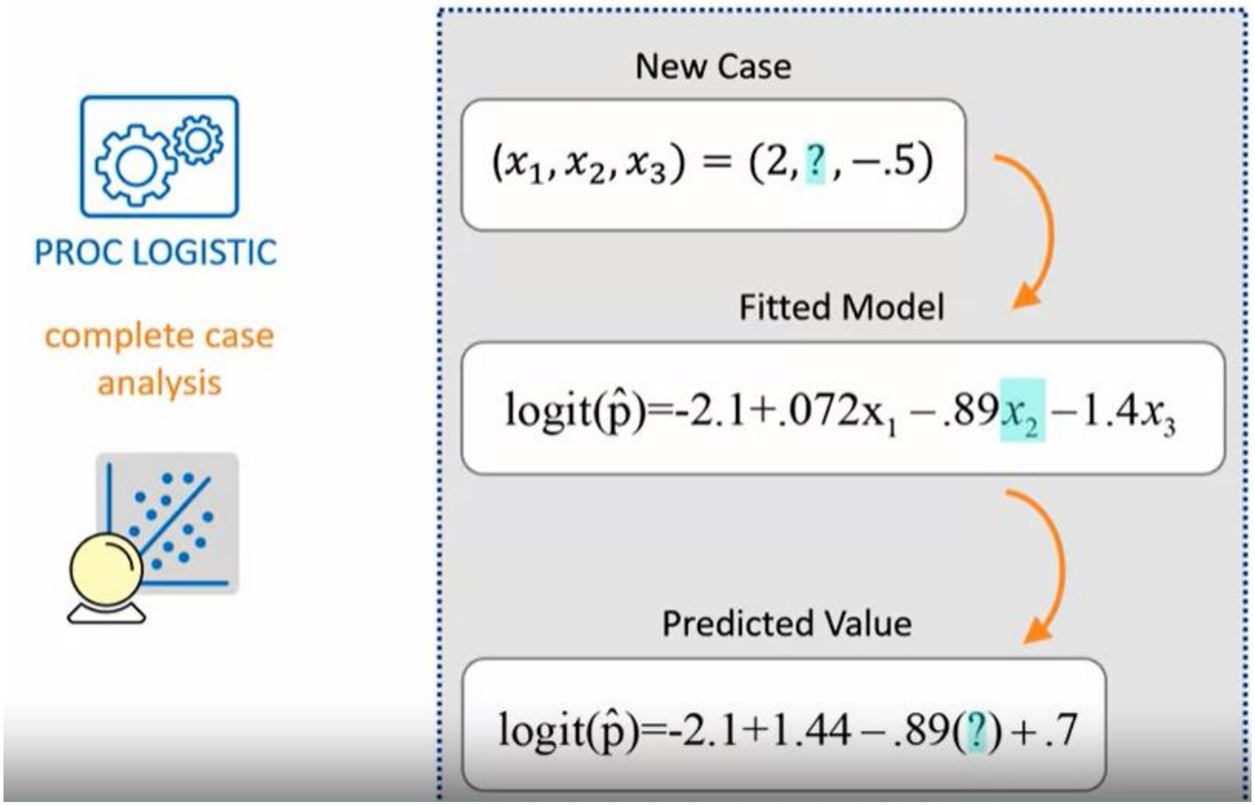



PROC LOGISTIC
complete case analysis
X_1 X_2 X_3 X_4 X_5 X_6 X_7 X_8 X_9 X_{10}


PROC LOGISTIC
complete case analysis
MCAR
Small percent missing.
X_1 X_2 X_3 X_4 X_5 X_6 X_7 X_8 X_9 X_{10}


PROC LOGISTIC
complete case analysis
X_1 X_2 X_3 X_4 X_5 X_6 X_7 X_8 X_9 X_{10}


PROC LOGISTIC
complete case analysis
score
new cases
No missing values!



Methods for Imputing Missing Values

X_1	X_2	X_3	X_4	X_5	X_6	X_7	X_8
6	03	2.6	0	8.3	42	66	C03
12	04	1.8	0	0.5	86	65	C14
?	01	?	?	4.8	37	?	C00
8	01	2.1	1	4.8	37	64	C08
6	01	2.8	1	9.6	22	66	?
3	?	2.7	0	1.1	28	64	C00
2	02	2.1	1	5.9	21	63	C03
10	03	2.0	0	?	?	63	?
7	01	2.5	0	5.5	62	67	C12
?	01	2.4	0	0.9	29	?	C05

Common Imputation Methods

- x_1: median
- x_2: mode
- x_3: mean
- x_4: mean

X_1	X_2	X_3	X_4	X_5	X_6	X_7	X_8
6	03	2.6	0	8.3	42	66	C03
12	04	1.8	0	0.5	86	65	C14
6.5	01	**2.3**	**.33**	4.8	37	**66**	C00
8	01	2.1	1	4.8	37	64	C08
6	01	2.8	1	9.6	22	66	**C99**
3	**01**	2.7	0	1.1	28	64	C00
2	02	2.1	1	5.9	21	63	C03
10	03	2.0	0	**0.8**	**0**	63	**C99**
7	01	2.5	0	5.5	62	67	C12
6.5	01	2.4	0	0.9	29	**63**	C05

Median is better to be used for imputation method if the values are only 0 and 1 for X4.

Common Imputation Methods

- x_1: median
- x_2: mode
- x_3: mean
- x_4: mean
- x_5: regression
- x_6: subject-matter knowledge

X_1	X_2	X_3	X_4	X_5	X_6	X_7	X_8
6	03	2.6	0	8.3	42	66	C03
12	04	1.8	0	0.5	86	65	C14
6.5	01	**2.3**	**.33**	4.8	37	**66**	C00
8	01	2.1	1	4.8	37	64	C08
6	01	2.8	1	9.6	22	66	**C99**
3	**01**	2.7	0	1.1	28	64	C00
2	02	2.1	1	5.9	21	63	C03
10	03	2.0	0	**0.8**	**0**	63	**C99**
7	01	2.5	0	5.5	62	67	C12
6.5	01	2.4	0	0.9	29	**63**	C05

Num_Items_Purchased

Common Imputation Methods

- x_1: median
- x_2: mode
- x_3: mean
- x_4: mean
- x_5: regression
- x_6: subject-matter knowledge
- x_7: hot-deck imputation
- x_8: new category

X_1	X_2	X_3	X_4	X_5	X_6	X_7	X_8
6	03	2.6	0	8.3	42	66	C03
12	04	1.8	0	0.5	86	65	C14
6.5	01	**2.3**	**.33**	4.8	37	**66**	C00
8	01	2.1	1	4.8	37	64	C08
6	01	2.8	1	9.6	22	66	**C99**
3	**01**	2.7	0	1.1	28	64	C00
2	02	2.1	1	5.9	21	63	C03
10	03	2.0	0	**0.8**	**0**	63	**C99**
7	01	2.5	0	5.5	62	67	C12
6.5	01	2.4	0	0.9	29	**63**	C05

Missing Value Imputation with Missing Value Indicator Variables

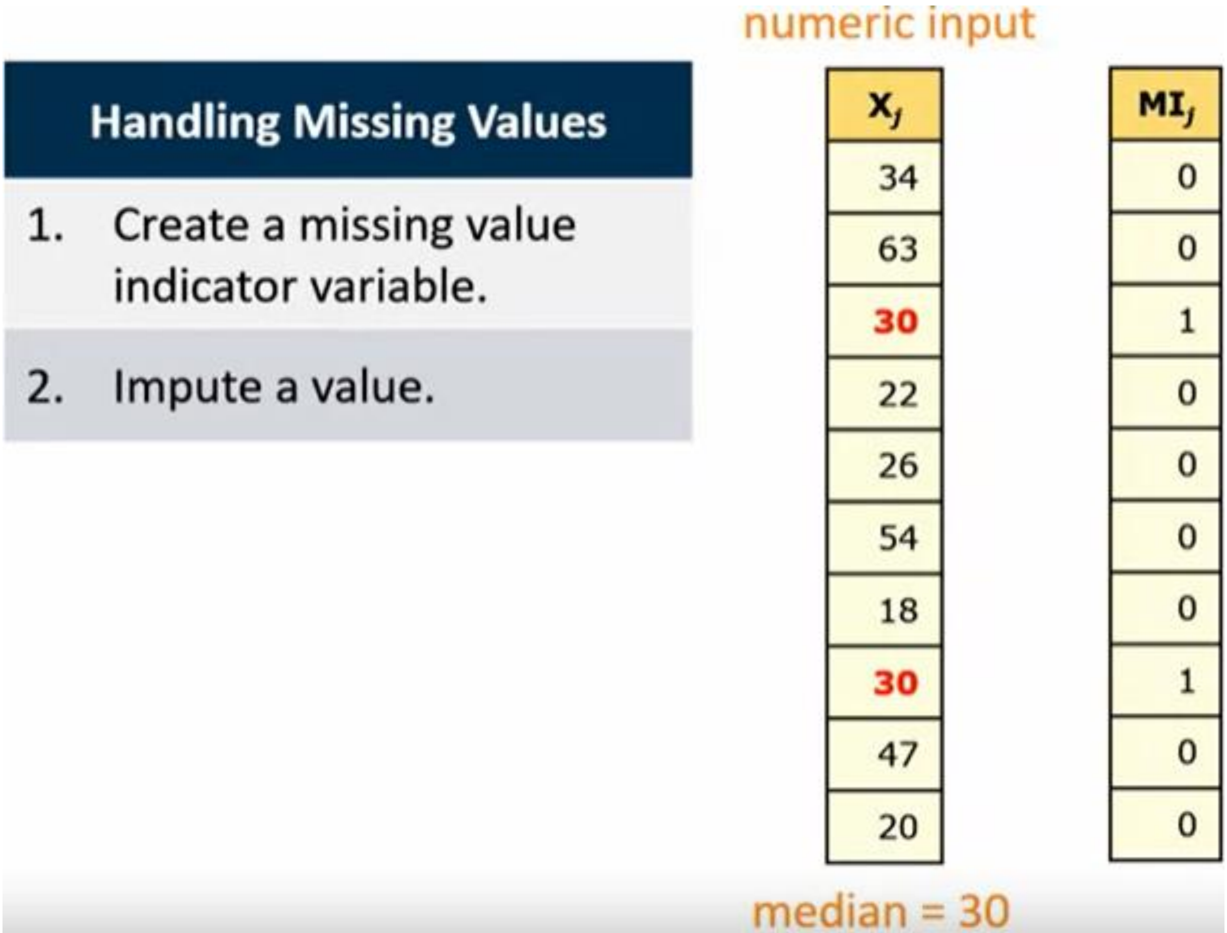
Handling Missing Values

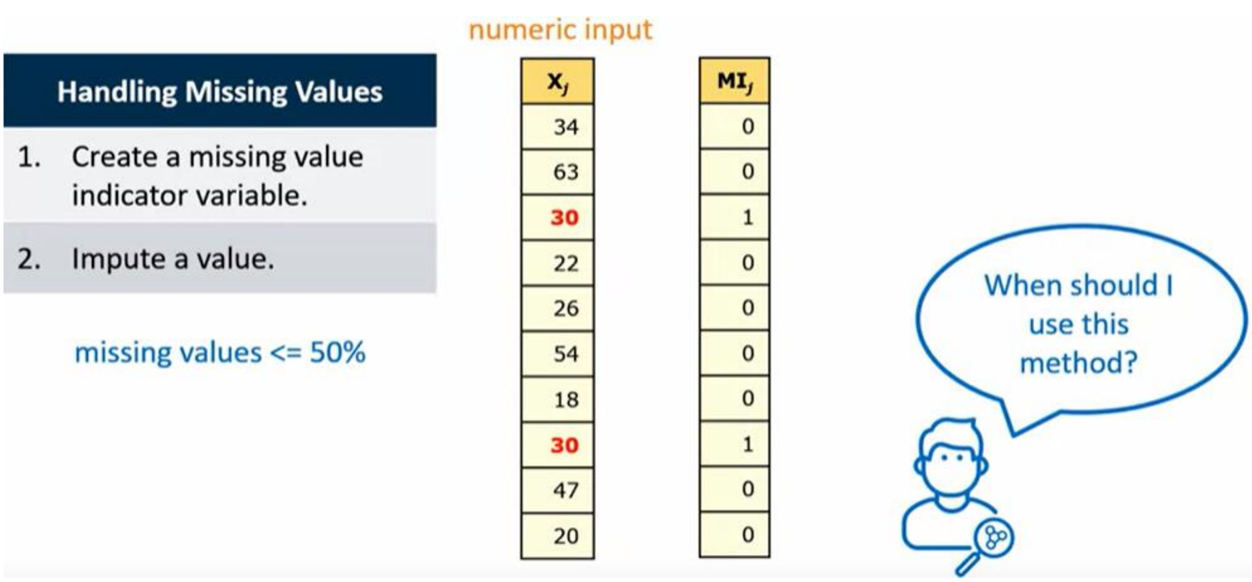
1. Create a missing value indicator variable.
2. Impute a value.

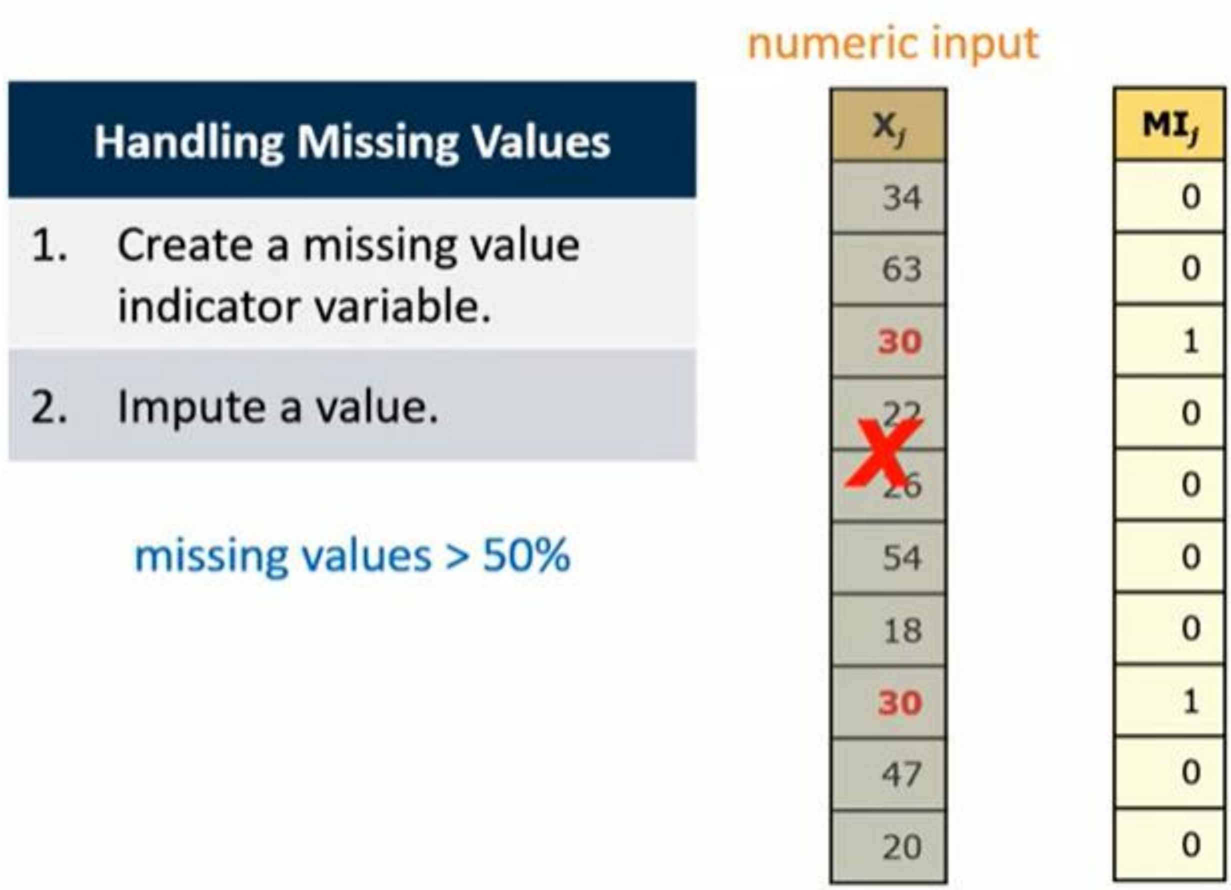
numeric input

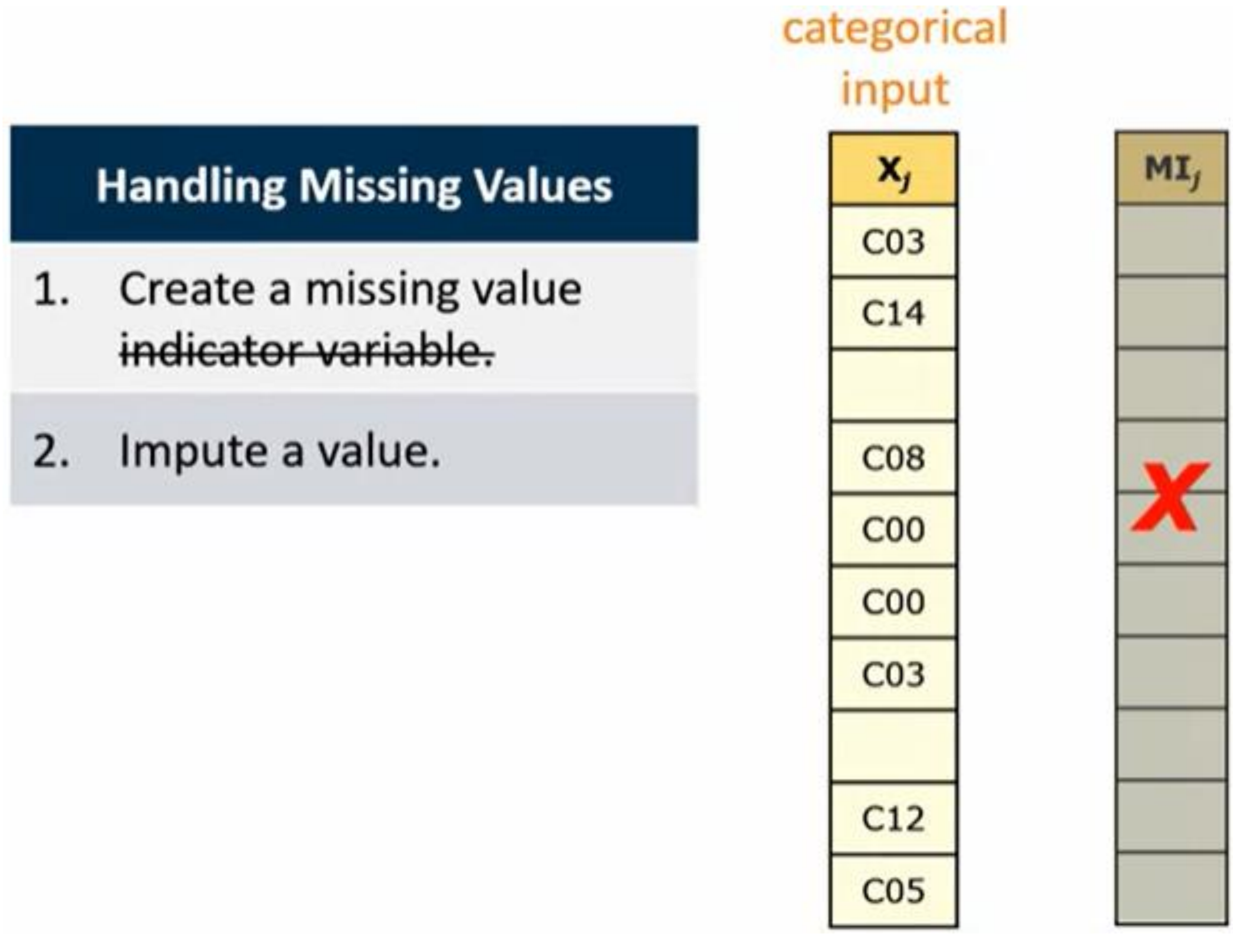
X_j	MI_j
34	0
63	0
.	1
22	0
26	0
54	0
18	0
.	1
47	0
20	0

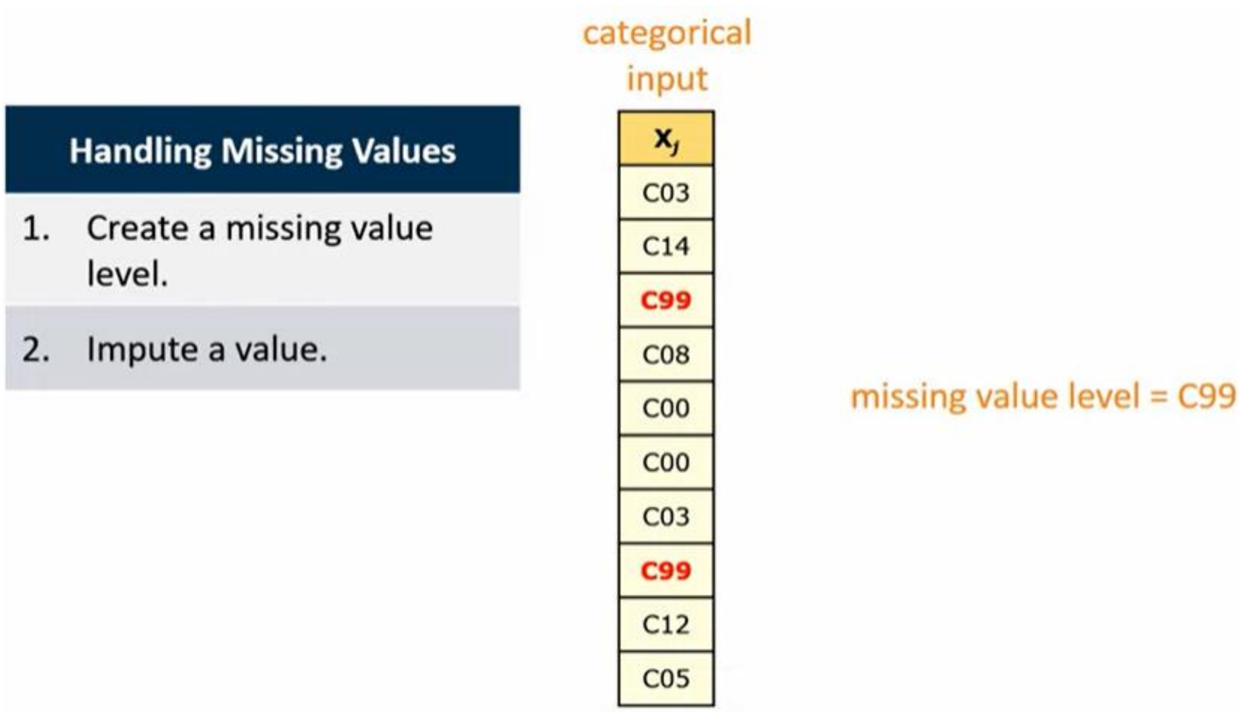
missingness ⟷ target variable

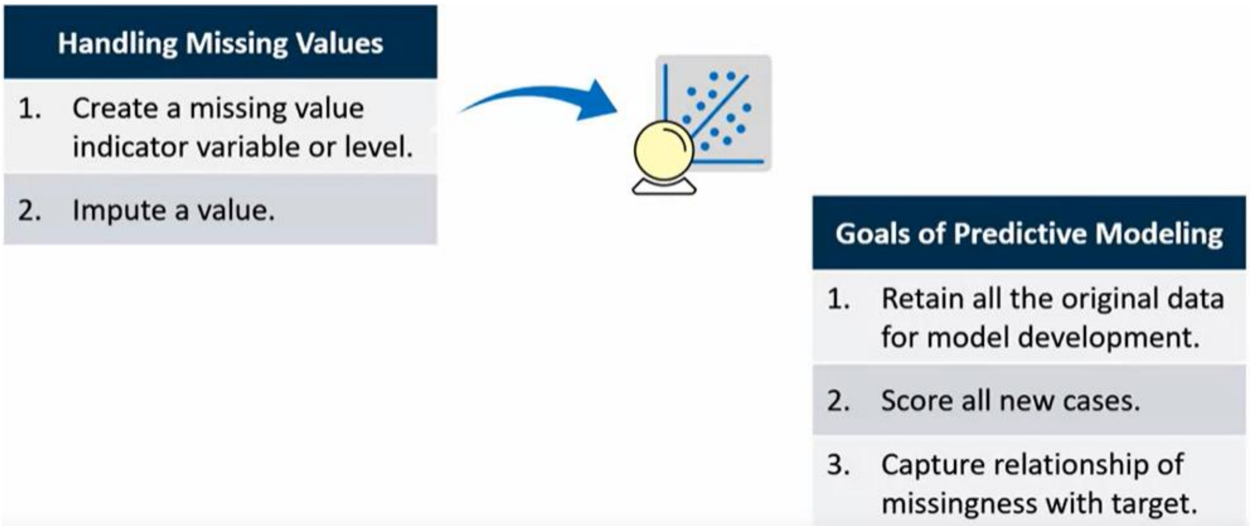

numeric input
Handling Missing Values
1. Create a missing value indicator variable.
2. Impute a value.
X_j
34
63
30
22
26
54
18
30
47
20
MI_j
0
0
1
0
0
0
0
1
0
0
median = 30

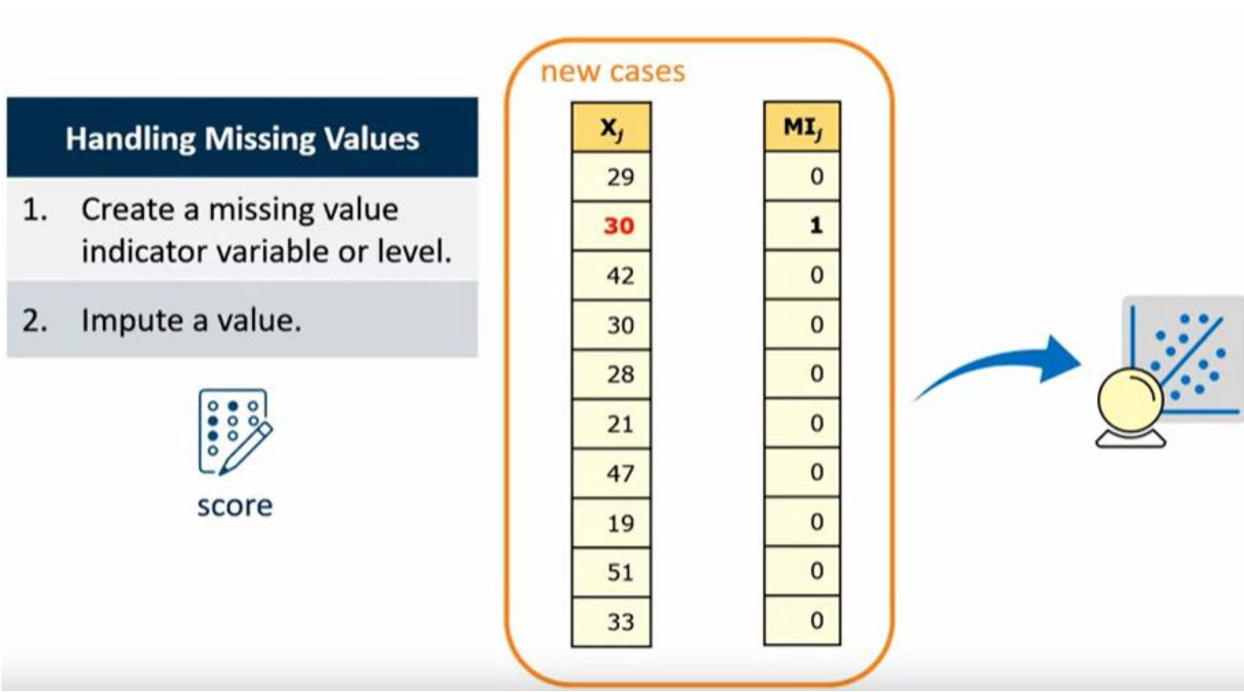

numeric input
Handling Missing Values
1. Create a missing value indicator variable.
2. Impute a value.
missing values <= 50%
X_j
34
63
30
22
26
54
18
30
47
20
MI_j
0
0
1
0
0
0
0
1
0
0
When should I use this method?


numeric input
Handling Missing Values
1. Create a missing value indicator variable.
2. Impute a value.
missing values > 50%
X_j
34
63
30
22
26
54
18
30
47
20
MI_j
0
0
1
0
0
0
0
1
0
0


categorical
input
Handling Missing Values
1. Create a missing value ~~indicator variable.~~
2. Impute a value.
x_j
C03
C14
C08
C00
C00
C03
C12
C05
MI_j


categorical
input
Handling Missing Values
1. Create a missing value level.
2. Impute a value.
x_j
C03
C14
C99
C08
C00
C00
C03
C99
C12
C05
missing value level = C99





Question 3.01

Which of the following statements is true regarding missing values in predictive modeling applications?

Missing value indicator variables can be used to capture the relationship between the target variable and missing inputs.

Demo Imputing Missing Values

The SAS System

The MEANS Procedure

Variable	Label	N	N Miss	Mean	Minimum	Maximum
AcctAge	Age of Oldest Account	30194	2070	5.9086772	0.3000000	61.5000000
DDA	Checking Account	32264	0	0.8156459	0	1.0000000
DDABal	Checking Balance	32264	0	2170.02	-774.8300000	278093.83
Dep	Checking Deposits	32264	0	2.1346082	0	28.0000000
DepAmt	Amount Deposited	32264	0	2232.76	0	484893.67
CashBk	Number Cash Back	32264	0	0.0159621	0	4.0000000
Checks	Number of Checks	32264	0	4.2599182	0	49.0000000
DirDep	Direct Deposit	32264	0	0.2955616	0	1.0000000
NSF	Number Insufficient Fund	32264	0	0.0870630	0	1.0000000
NSFAmt	Amount NSF	32264	0	2.2905464	0	666.8500000
Phone	Number Telephone Banking	28131	4133	0.4056024	0	30.0000000
Teller	Teller Visits	32264	0	1.3652678	0	27.0000000
Sav	Saving Account	32264	0	0.4668981	0	1.0000000
SavBal	Saving Balance	32264	0	3170.60	0	700026.94
ATM	ATM	32264	0	0.6099368	0	1.0000000
ATMAmt	ATM Withdrawal Amount	32264	0	1235.41	0	427731.26
POS	Number Point of Sale	28131	4133	1.0756816	0	54.0000000
POSAmt	Amount Point of Sale	28131	4133	48.9261782	0	3293.49

pmlr03d01.sas

```
title1 "Variables with Missing Values";
proc print data=work.train(obs=15);
   var ccbal ccpurc income hmown;
run;
title1 ;
```

Variables with Missing Values

Obs	CCBal	CCPurc	Income	HMOwn
1	0.00	1	4	1
2	65.76	0	125	1
3	85202.99	0	55	1
4	.	.	20	0
5	0.00	0	25	1
6	0.00	0	8	1
7	0.00	0	100	1
8	323.13	0	13	1
9	32366.86	0	.	1
10	0.00	0	9	0
11	1378.46	1	60	1
12	.	.	25	0

pmlr03d01.sas

```
/* Create missing indicators */
data work.train_mi(drop=i);
   set work.train;
   /* name the missing indicator variables */
   array mi{*} MIAcctAg MIPhone MIPOS MIPOSAmt
               MIInv MIInvBal MICC MICCBal
               MICCPurc MIIncome MIHMOwn MILORes
               MIHMVal MIAge MICRScor;
```

```
/* select variables with missing values */
array x{*} acctage phone pos posamt
           inv invbal cc ccbal
           ccpurc income hmown lores
           hmval age crscore;
```

```
   do i=1 to dim(mi);
      mi{i}=(x{i}=.);
      nummiss+mi{i};
   end;
run;
```

```
/* Impute missing values with the median */
proc stdize data=work.train_mi reponly method=median out=work.train_impute
   var &inputs;
run;
```

```
title1 "Imputed Values with Missing Indicators";
proc print data=work.train_imputed(obs=12);
   var ccbal miccbal ccpurc miccpurc income miincome hmown mihmown nummis
run;
title1 ;
```

Imputed Values with Missing Indicators

Obs	CCBal	MICCBal	CCPurc	MICCPurc	Income	MIIncome	HMOwn	MIHMOwn	nummiss
1	0.00	0	1	0	4	0	1	0	0
2	65.76	0	0	0	125	0	1	0	0
3	85202.99	0	0	0	55	0	1	0	0
4	0.00	1	0	1	20	0	0	0	8
5	0.00	0	0	0	25	0	1	0	8
6	0.00	0	0	0	8	0	1	0	9
7	0.00	0	0	0	100	0	1	0	9
8	323.13	0	0	0	13	0	1	0	9
9	32366.86	0	0	0	35	1	1	0	13
10	0.00	0	0	0	9	0	0	0	13
11	1378.46	0	1	0	60	0	1	0	13
12	0.00	1	0	1	25	0	0	0	21

```
/* Run this code before demo l3d1 */

/* ===================================================== */
/* Lesson 1, Section 1: l1d1.sas
  Demonstration: Examining the Code for Generating
  Descriptive Statistics and Frequency Tables         */
/* ===================================================== */

data work.develop;
  set pmlr.develop;
run;

%global inputs;
%let inputs=ACCTAGE DDA DDABAL DEP DEPAMT CASHBK
        CHECKS DIRDEP NSF NSFAMT PHONE TELLER
        SAV SAVBAL ATM ATMAMT POS POSAMT CD
        CDBAL IRA IRABAL LOC LOCBAL INV
        INVBAL ILS ILSBAL MM MMBAL MMCRED MTG
        MTGBAL CC CCBAL CCPURC SDB INCOME
        HMOWN LORES HMVAL AGE CRSCORE MOVED
        INAREA;

proc means data=work.develop n nmiss mean min max;
  var &inputs;
run;

proc freq data=work.develop;
  tables ins branch res;
run;
```

```
/* ===================================================== */
/* Lesson 1, Section 2: l1d2.sas
   Demonstration: Splitting the Data   */
/* ===================================================== */


/* Sort the data by the target in preparation for stratified sampling. */

proc sort data=work.develop out=work.develop_sort;
   by ins;
run;

/* The SURVEYSELECT procedure will perform stratified sampling
   on any variable in the STRATA statement. The OUTALL option
   specifies that you want a flag appended to the file to
   indicate selected records, not simply a file comprised
   of the selected records. */

proc surveyselect noprint data=work.develop_sort
              samprate=.6667 stratumseed=restore
              out=work.develop_sample
              seed=44444 outall;
   strata ins;
run;

/* Verify stratification. */

proc freq data=work.develop_sample;
```

```
  tables ins*selected;
run;

/* Create training and validation data sets. */

data work.train(drop=selected SelectionProb SamplingWeight)
    work.valid(drop=selected SelectionProb SamplingWeight);
  set work.develop_sample;
  if selected then output work.train;
  else output work.valid;
run;



/* ===================================================== */
/* Lesson 2, Section 1: l2d1.sas
  Demonstration: Fitting a Basic Logistic
  Regression Model, Parts 1 and 2                    */
/* ===================================================== */

title1 "Logistic Regression Model for the Variable Annuity Data Set";
proc logistic data=work.train
          plots(only maxpoints=none)=(effect(clband x=(ddabal depamt checks res))
          oddsratio (type=horizontalstat));
  class res (param=ref ref='S') dda (param=ref ref='0');
  model ins(event='1')=dda ddabal dep depamt
           cashbk checks res / stb clodds=pl;
  units ddabal=1000 depamt=1000 / default=1;
  oddsratio 'Comparisons of Residential Classification' res / diff=all cl=pl;
```

```
  effectplot slicefit(sliceby=dda x=ddabal) / noobs;
  effectplot slicefit(sliceby=dda x=depamt) / noobs;
run;
title1;


/* ===================================================== */
/* Lesson 2, Section 1: l2d2.sas
   Demonstration: Scoring New Cases                    */
/* ===================================================== */

/* Score a new data set with one run of the LOGISTIC procedure with the
   SCORE statement.  */

proc logistic data=work.train noprint;
  class res (param=ref ref='S');
  model ins(event='1')= res dda ddabal dep depamt cashbk checks;
  score data = pmlr.new out=work.scored1;
run;

title1 "Predicted Probabilities from Scored Data Set";
proc print data=work.scored1(obs=10);
  var p_1 dda ddabal dep depamt cashbk checks res;
run;

title1 "Mean of Predicted Probabilities from Scored Data Set";
proc means data=work.scored1 mean nolabels;
  var p_1;
run;
```

```
/* Score a new data set with the OUTMODEL= amd INMODEL= options */

proc logistic data=work.train outmodel=work.scoredata noprint;
  class res (param=ref ref='S');
  model ins(event='1')= res dda ddabal dep depamt cashbk checks;
run;

proc logistic inmodel=work.scoredata noprint;
  score data = pmlr.new out=work.scored2;
run;

title1 "Predicted Probabilities from Scored Data Set";
proc print data=work.scored2(obs=10);
  var p_1 dda ddabal dep depamt cashbk checks res;
run;

/* Score a new data set with the CODE Statement  */

proc logistic data=work.train noprint;
  class res (param=ref ref='S');
  model ins(event='1')= res dda ddabal dep depamt cashbk checks;
  code file="&PMLRfolder/pmlr_score.txt";
run;

data work.scored3;
  set pmlr.new;
  %include "&PMLRfolder/pmlr_score.txt";
run;
```

```
title1 "Predicted Probabilities from Scored Data Set";
proc print data=work.scored3(obs=10);
  var p_ins1 dda ddabal dep depamt cashbk checks res;
run;
title1 ;



/* ===================================================== */
/* Lesson 2, Section 2: l2d3.sas
  Demonstration: Correcting for Oversampling          */
/* ===================================================== */

/* Specify the prior probability to correct for oversampling.  */
%global pi1;
%let pi1=.02;

/* Correct predicted probabilities */

proc logistic data=work.train noprint;
  class res (param=ref ref='S');
  model ins(event='1')=dda ddabal dep depamt cashbk checks res;
  score data=pmlr.new out=work.scored4 priorevent=&pi1;
run;

title1 "Adjusted Predicted Probabilities from Scored Data Set";
proc print data=work.scored4(obs=10);
  var p_1 dda ddabal dep depamt cashbk checks res;
run;
```

```
title1 "Mean of Adjusted Predicted Probabilities from Scored Data Set";
proc means data=work.scored4 mean nolabels;
  var p_1;
run;
title1 ;

/* Correct probabilities in the Score Code */

proc logistic data=work.train noprint;
  class res (param=ref ref='S');
  model ins(event='1')=dda ddabal dep depamt cashbk checks res;
  /* File suffix "txt" is used so you can view the file */
  /* with a native text editor. SAS prefers "sas", but  */
  /* when specified as a filename, SAS does not care.   */
  code file="&PMLRfolder/pmlr_score_adj.txt";
run;

%global rho1;
proc SQL noprint;
  select mean(INS) into :rho1
  from work.train;
quit;

data new;
  set pmlr.new;
  off=log(((1-&pi1)*&rho1)/(&pi1*(1-&rho1)));
run;
```

```
data work.scored5;
  set work.new;
  %include "&PMLRfolder/pmlr_score_adj.txt";
  eta=log(p_ins1/p_ins0) - off;
  prob=1/(1+exp(-eta));
run;

title1 "Adjusted Predicted Probabilities from Scored Data Set";
proc print data=scored5(obs=10);
  var prob dda ddabal dep depamt cashbk checks res;
run;
title1 ;



/* ===================================================== */
/* Lesson 3, Section 1: l3d1.sas
  Demonstration: Imputing Missing Values
  [m643_1_h; derived from pmlr03d01.sas]             */
/* ===================================================== */

title1 "Variables with Missing Values";
proc print data=work.train(obs=15);
  var ccbal ccpurc income hmown;
run;
title1 ;

/* Create missing indicators */
data work.train_mi(drop=i);
  set work.train;
```

```
  /* name the missing indicator variables */
  array mi{*} MIAcctAg MIPhone MIPOS MIPOSAmt
           MIInv MIInvBal MICC MICCBal
           MICCPurc MIIncome MIHMOwn MILORes
           MIHMVal MIAge MICRScor;
  /* select variables with missing values */
  array x{*} acctage phone pos posamt
          inv invbal cc ccbal
          ccpurc income hmown lores
          hmval age crscore;
  do i=1 to dim(mi);
    mi{i}=(x{i}=.);
    nummiss+mi{i};
  end;
run;

/* Impute missing values with the median */
proc stdize data=work.train_mi reponly method=median out=work.train_imputed;
  var &inputs;
run;

title1 "Imputed Values with Missing Indicators";
proc print data=work.train_imputed(obs=12);
  var ccbal miccbal ccpurc miccpurc income miincome hmown mihmown nummiss;
run;
title1 ;
```

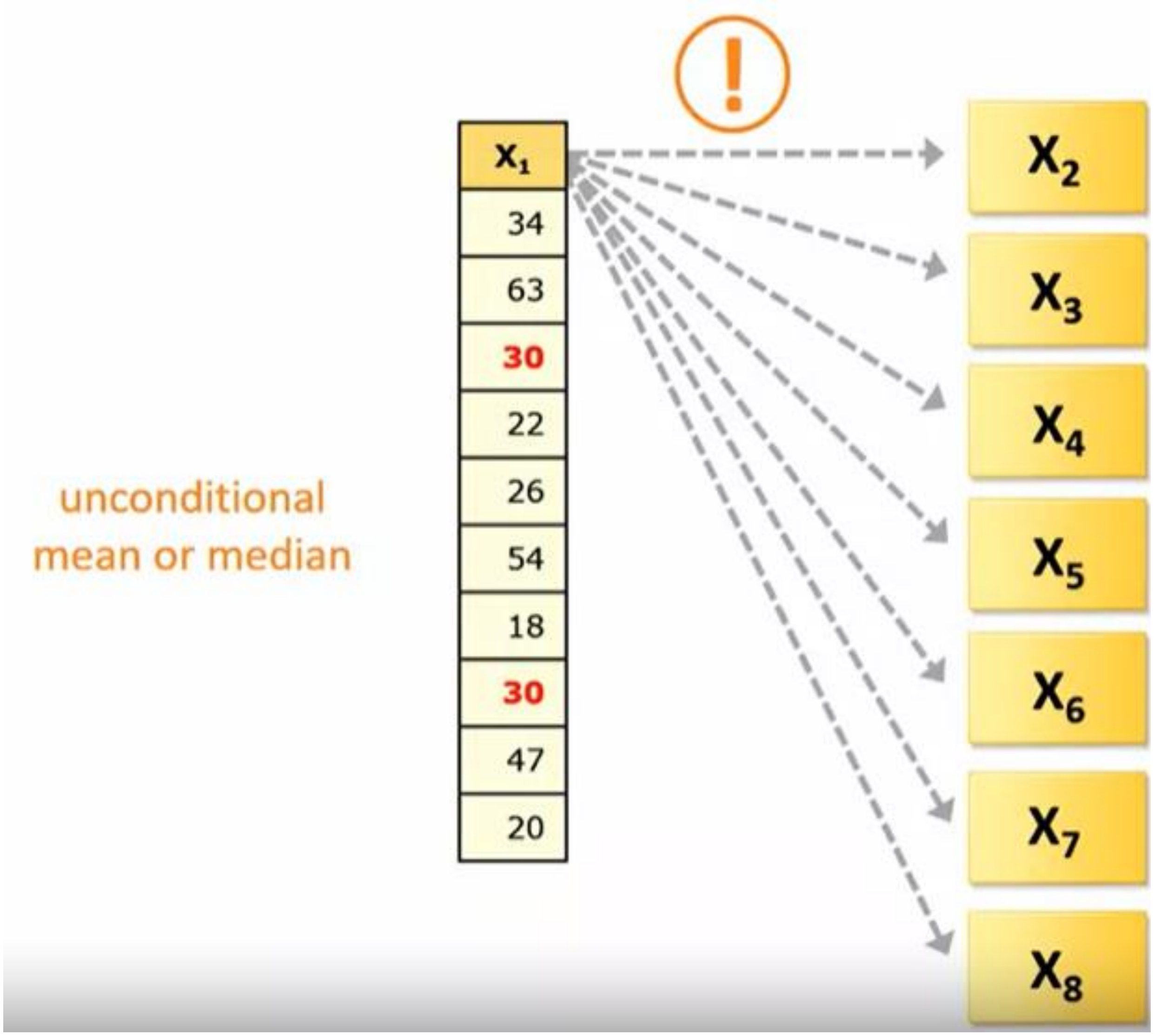
Variables with Missing Values

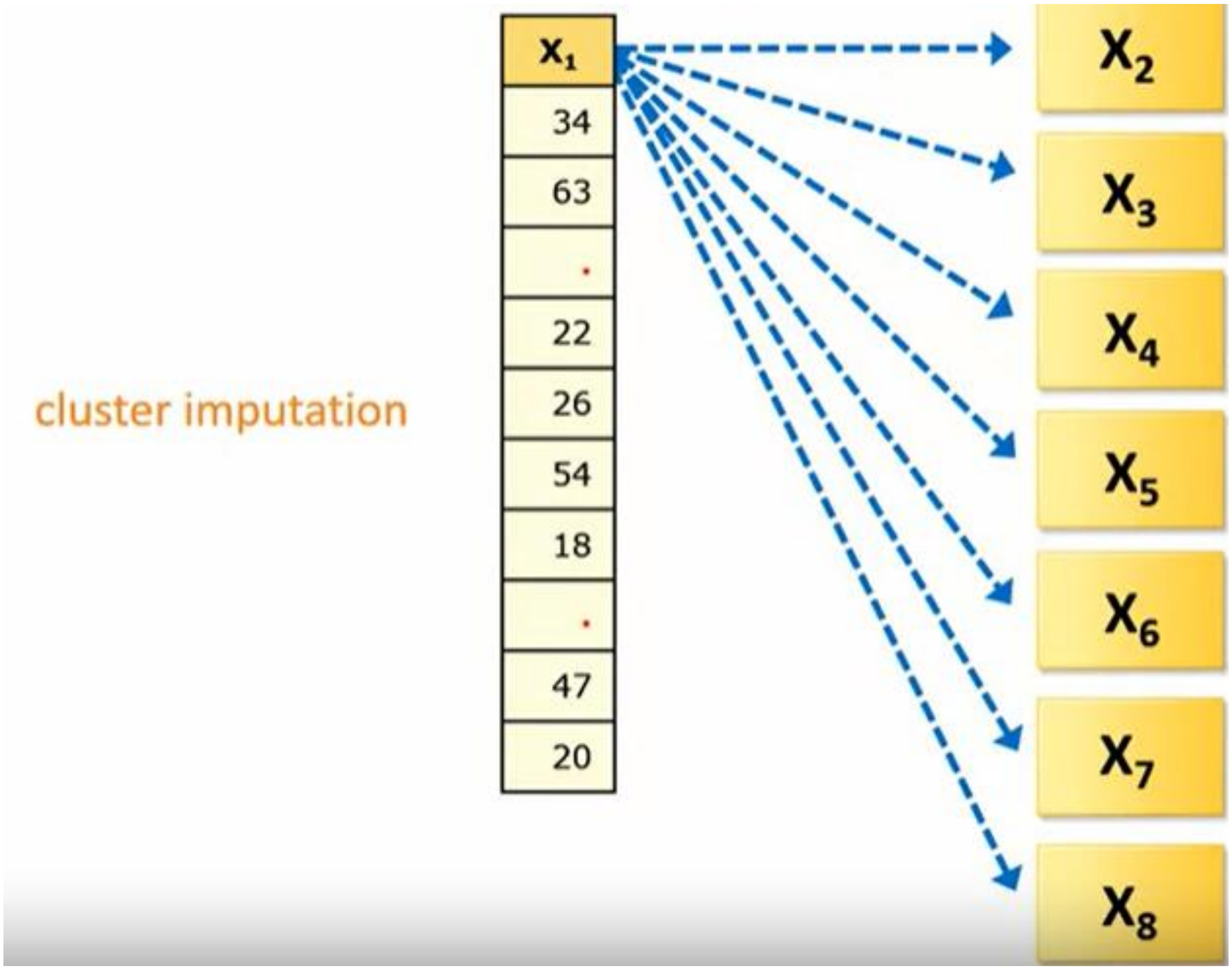
Obs	CCBal	CCPurc	Income	HMOwn
1	0.00	1	4	1
2	65.76	0	125	1
3	85202.99	0	55	1
4	.	.	20	0
5	0.00	0	25	1
6	0.00	0	8	1
7	0.00	0	100	1
8	323.13	0	13	1
9	32366.86	0	.	1
10	0.00	0	9	0
11	1378.46	1	60	1
12	.	.	25	0
13	0.00	0	54	0
14	1466.87	0	45	0
15	.	.	31	0

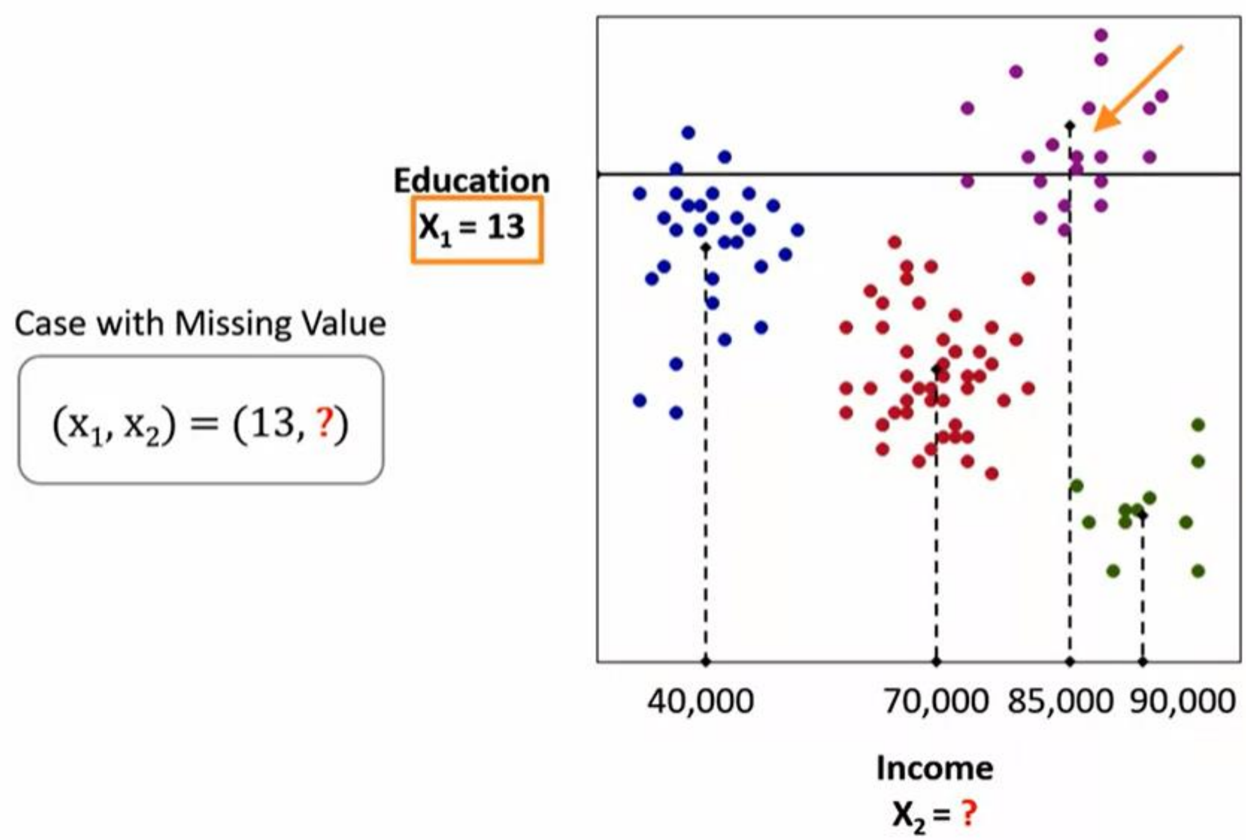
Imputed Values with Missing Indicators

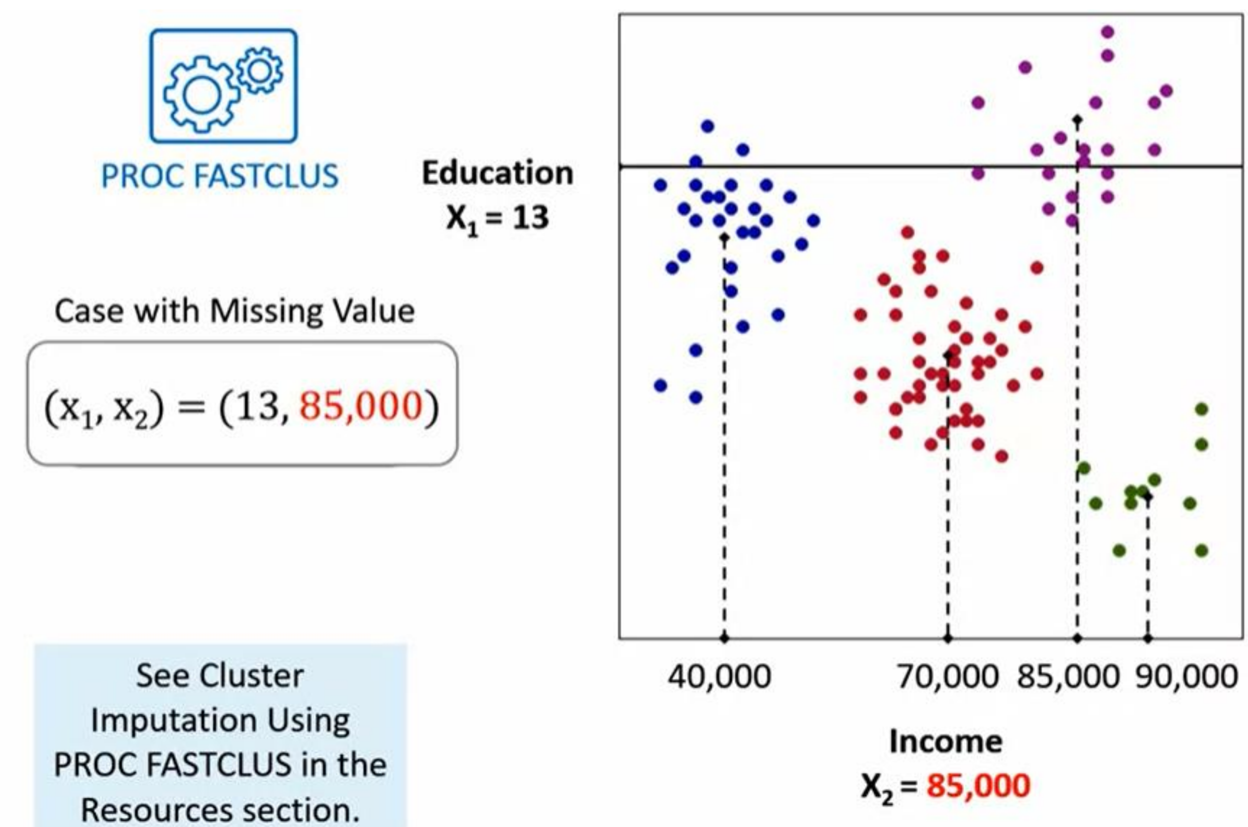
Obs	CCBal	MICCBal	CCPurc	MICCPurc	Income	MIIncome	HMOwn	MIHMOwn	nummiss
1	0.00	0	1	0	4	0	1	0	0
2	65.76	0	0	0	125	0	1	0	0
3	85202.99	0	0	0	55	0	1	0	0
4	0.00	1	0	1	20	0	0	0	8
5	0.00	0	0	0	25	0	1	0	8
6	0.00	0	0	0	8	0	1	0	9
7	0.00	0	0	0	100	0	1	0	9
8	323.13	0	0	0	13	0	1	0	9
9	32366.86	0	0	0	35	1	1	0	13
10	0.00	0	0	0	9	0	0	0	13
11	1378.46	0	1	0	60	0	1	0	13
12	0.00	1	0	1	25	0	0	0	21

Cluster Imputation




cluster imputation
x_1
34
63
.
22
26
54
18
.
47
20
X_2
X_3
X_4
X_5
X_6
X_7
X_8


Education
$X_1 = 13$
Case with Missing Value
$(x_1, x_2) = (13, ?)$
40,000
70,000
85,000
90,000
Income
$X_2 = ?$


PROC FASTCLUS
Education
$X_1 = 13$
Case with Missing Value
$(x_1, x_2) = (13, 85{,}000)$
See Cluster Imputation Using PROC FASTCLUS in the Resources section.
40,000
70,000
85,000
90,000
Income
$X_2 = 85{,}000$

```
/* Run this code before doing practice l3p1 */


/* ===================================================== */
/* Lesson 1, Practice 1
  Practice: Exploring the Veterans' Organization Data
  Used in the Practices                           */
/* ===================================================== */


data pmlr.pva(drop=control_number
              MONTHS_SINCE_LAST_PROM_RESP
              FILE_AVG_GIFT
              FILE_CARD_GIFT);
  set pmlr.pva_raw_data;
  STATUS_FL=RECENCY_STATUS_96NK in("F","L");
  STATUS_ES=RECENCY_STATUS_96NK in("E","S");
  home01=(HOME_OWNER="H");
  nses1=(SES="1");
  nses3=(SES="3");
  nses4=(SES="4");
  nses_=(SES="?");
  nurbr=(URBANICITY="R");
  nurbu=(URBANICITY="U");
  nurbs=(URBANICITY="S");
  nurbt=(URBANICITY="T");
  nurb_=(URBANICITY="?");
run;


proc contents data=pmlr.pva;
run;
```

```
proc means data=pmlr.pva mean nmiss max min;
  var _numeric_;
run;

proc freq data=pmlr.pva nlevels;
  tables _character_;
run;


/* ===================================================== */
/* Lesson 1, Practice 2
  Practice: Splitting the Data                    */
/* ===================================================== */

proc sort data=pmlr.pva out=work.pva_sort;
  by target_b;
run;

proc surveyselect noprint data=work.pva_sort
            samprate=0.5 out=pva_sample seed=27513
            outall stratumseed=restore;
  strata target_b;
run;

data pmlr.pva_train(drop=selected SelectionProb SamplingWeight)
    pmlr.pva_valid(drop=selected SelectionProb SamplingWeight);
  set work.pva_sample;
  if selected then output pmlr.pva_train;
```

```
  else output pmlr.pva_valid;
run;



/* ===================================================== */
/* Lesson 2, Practice 1
  Practice: Fitting a Logistic Regression Model        */
/* ===================================================== */

/* Modifications for your SAS software:
  ----------------------------------------------------
  (Optional) To avoid a warning in the log about the
  suppression of plots that have more than 5000
  observations, you can add the MAXPOINTS= option
  to the PROC LOGISTIC statement like this:
  plots(maxpoints=none only). Omitting the
  MAXPOINTS= option does not affect the results
  of the practices in this course.
*/

%global ex_pi1;
%let ex_pi1=0.05;

title1 "Logistic Regression Model of the Veterans' Organization Data";
proc logistic data=pmlr.pva_train plots(only)=
        (effect(clband x=(pep_star recent_avg_gift_amt
        frequency_status_97nk)) oddsratio (type=horizontalstat));
  class pep_star (param=ref ref='0');
  model target_b(event='1')=pep_star recent_avg_gift_amt
```

```
            frequency_status_97nk / clodds=pl;
   effectplot slicefit(sliceby=pep_star x=recent_avg_gift_amt) / noobs;
   effectplot slicefit(sliceby=pep_star x=frequency_status_97nk) / noobs;
   score data=pmlr.pva_train out=work.scopva_train priorevent=&ex_pi1;
run;

title1 "Adjusted Predicted Probabilities of the Veteran's Organization Data";
proc print data=work.scopva_train(obs=10);
   var p_1 pep_star recent_avg_gift_amt frequency_status_97nk;

run;
title;
```

The CONTENTS Procedure

Data Set Name	PMLR.PVA	Observations	19372
Member Type	DATA	Variables	58
Engine	V9	Indexes	0
Created	09/18/2021 20:51:40	Observation Length	432
Last Modified	09/18/2021 20:51:40	Deleted Observations	0
Protection		Compressed	NO
Data Set Type		Sorted	NO
Label			
Data Representation	SOLARIS_X86_64, LINUX_X86_64, ALPHA_TRU64, LINUX_IA64		
Encoding	utf-8 Unicode (UTF-8)		

Engine/Host Dependent Information	
Data Set Page Size	131072
Number of Data Set Pages	65
First Data Page	1
Max Obs per Page	303
Obs in First Data Page	281
Number of Data Set Repairs	0
Filename	/home/u58304328/EPMLR51/data/pva.sas7bdat
Release Created	9.0401M6
Host Created	Linux
Inode Number	10881694450
Access Permission	rw-r--r--
Owner Name	u58304328
File Size	8MB
File Size (bytes)	8650752

The MEANS Procedure

Variable	Mean	N Miss	Maximum	Minimum
TARGET_B	0.2500000	0	1.0000000	0
TARGET_D	15.6243444	14529	200.0000000	1.0000000
MONTHS_SINCE_ORIGIN	73.4099732	0	137.0000000	5.0000000
DONOR_AGE	58.9190506	4795	87.0000000	0
IN_HOUSE	0.0731984	0	1.0000000	0
INCOME_GROUP	3.9075434	4392	7.0000000	1.0000000
PUBLISHED_PHONE	0.4977287	0	1.0000000	0
MOR_HIT_RATE	3.3616560	0	241.0000000	0
WEALTH_RATING	5.0053967	8810	9.0000000	0

The FREQ Procedure

Number of Variable Levels	
Variable	Levels
URBANICITY	6
SES	5
CLUSTER_CODE	54
HOME_OWNER	2
DONOR_GENDER	4
OVERLAY_SOURCE	4
RECENCY_STATUS_96NK	6

URBANICITY	Frequency	Percent	Cumulative Frequency	Cumulative Percent
?	454	2.34	454	2.34
C	4022	20.76	4476	23.11
R	4005	20.67	8481	43.78
S	4491	23.18	12972	66.96
T	3944	20.36	16916	87.32
U	2456	12.68	19372	100.00

SES	Frequency	Percent	Cumulative Frequency	Cumulative Percent
1	5924	30.58	5924	30.58
2	9284	47.92	15208	78.51
3	3323	17.15	18531	95.66
4	387	2.00	18918	97.66
?	454	2.34	19372	100.00

CLUSTER_CODE	Frequency	Percent	Cumulative Frequency	Cumulative Percent
.	454	2.34	454	2.34
01	239	1.23	693	3.58
02	380	1.96	1073	5.54
03	300	1.55	1373	7.09
04	113	0.58	1486	7.67
05	199	1.03	1685	8.70
06	123	0.63	1808	9.33
07	184	0.95	1992	10.28
08	378	1.95	2370	12.23
09	153	0.79	2523	13.02
10	387	2.00	2910	15.02
11	484	2.50	3394	17.52
12	631	3.26	4025	20.78
13	579	2.99	4604	23.77
14	454	2.34	5058	26.11
15	223	1.15	5281	27.26
16	384	1.98	5665	29.24
17	349	1.80	6014	31.04
18	619	3.20	6633	34.24
19	98	0.51	6731	34.75
20	317	1.64	7048	36.38
21	353	1.82	7401	38.20
22	251	1.30	7652	39.50
23	293	1.51	7945	41.01
24	795	4.10	8740	45.12
25	273	1.41	9013	46.53

26	202	1.04	9215	47.57
27	666	3.44	9881	51.01
28	343	1.77	10224	52.78
29	170	0.88	10394	53.65
30	519	2.68	10913	56.33
31	249	1.29	11162	57.62
32	152	0.78	11314	58.40
33	109	0.56	11423	58.97
34	284	1.47	11707	60.43
35	727	3.75	12434	64.19
36	716	3.70	13150	67.88
37	204	1.05	13354	68.93
38	240	1.24	13594	70.17
39	512	2.64	14106	72.82
40	830	4.28	14936	77.10
41	431	2.22	15367	79.33
42	284	1.47	15651	80.79
43	468	2.42	16119	83.21
44	383	1.98	16502	85.18
45	482	2.49	16984	87.67
46	369	1.90	17353	89.58
47	185	0.95	17538	90.53
48	180	0.93	17718	91.46
49	675	3.48	18393	94.95
50	156	0.81	18549	95.75
51	460	2.37	19009	98.13
52	60	0.31	19069	98.44
53	303	1.56	19372	100.00

HOME_OWNER	Frequency	Percent	Cumulative Frequency	Cumulative Percent
H	10606	54.75	10606	54.75
U	8766	45.25	19372	100.00

DONOR_GENDER	Frequency	Percent	Cumulative Frequency	Cumulative Percent
A	1	0.01	1	0.01
F	10401	53.69	10402	53.70
M	7953	41.05	18355	94.75
U	1017	5.25	19372	100.00

OVERLAY_SOURCE	Frequency	Percent	Cumulative Frequency	Cumulative Percent
B	8732	45.08	8732	45.08
M	1480	7.64	10212	52.72
N	4392	22.67	14604	75.39
P	4768	24.61	19372	100.00

RECENCY_STATUS_96NK	Frequency	Percent	Cumulative Frequency	Cumulative Percent
A	11918	61.52	11918	61.52
E	427	2.20	12345	63.73
F	1521	7.85	13866	71.58
L	93	0.48	13959	72.06
N	1192	6.15	15151	78.21
S	4221	21.79	19372	100.00

Logistic Regression Model of the Veterans' Organization Data

The LOGISTIC Procedure

Model Information	
Data Set	PMLR.PVA_TRAIN
Response Variable	TARGET_B
Number of Response Levels	2
Model	binary logit
Optimization Technique	Fisher's scoring

Number of Observations Read	9687
Number of Observations Used	9687

Response Profile		
Ordered Value	TARGET_B	Total Frequency
1	0	7265
2	1	2422

Probability modeled is TARGET_B=1.

Class Level Information		
Class	Value	Design Variables
PEP_STAR	0	0
	1	1

Model Convergence Status
Convergence criterion (GCONV=1E-8) satisfied.

Model Fit Statistics		
Criterion	Intercept Only	Intercept and Covariates
AIC	10897.230	10663.061
SC	10904.409	10691.776
-2 Log L	10895.230	10655.061

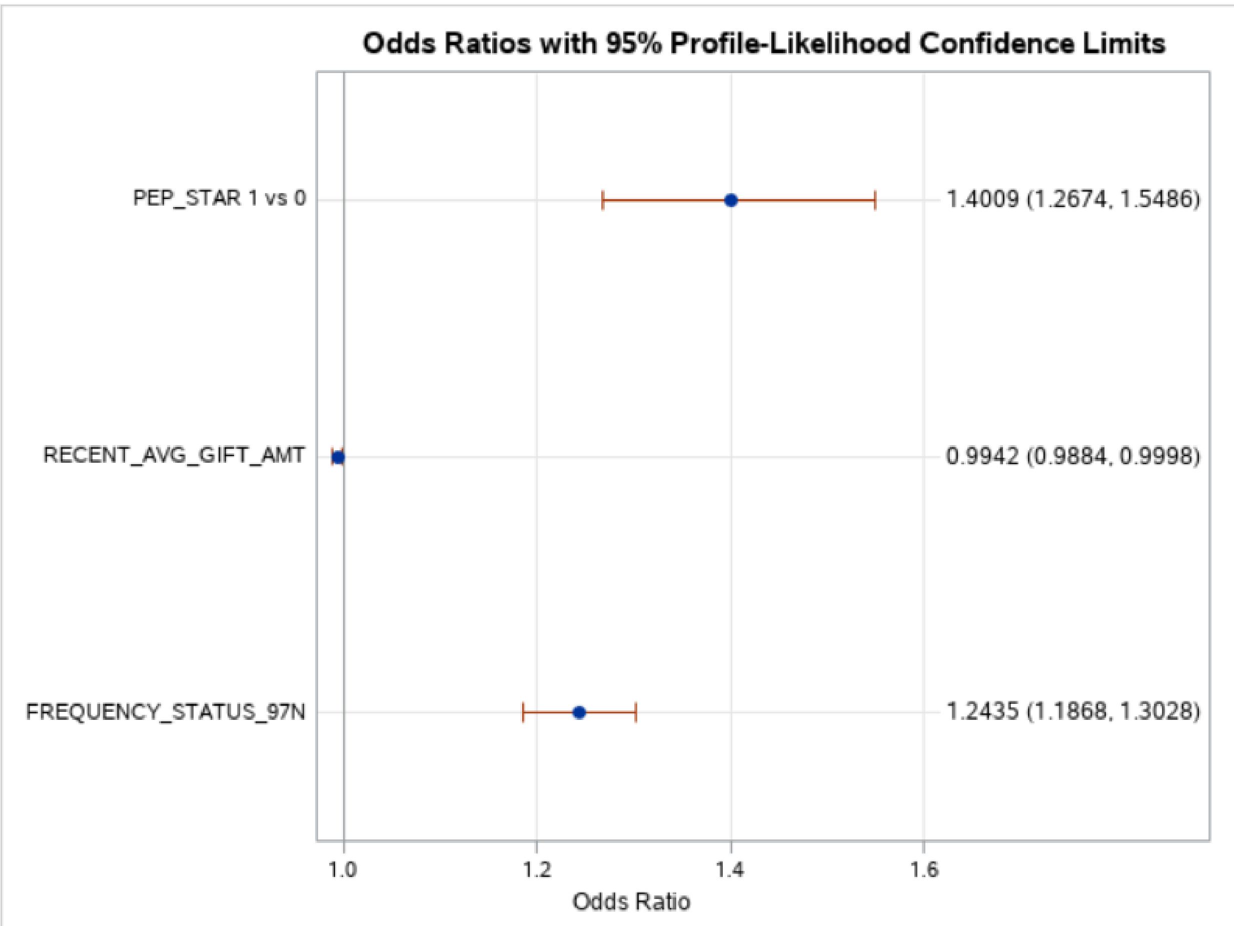
Testing Global Null Hypothesis: BETA=0			
Test	Chi-Square	DF	Pr > ChiSq
Likelihood Ratio	240.1690	3	<.0001
Score	242.9486	3	<.0001
Wald	237.2875	3	<.0001

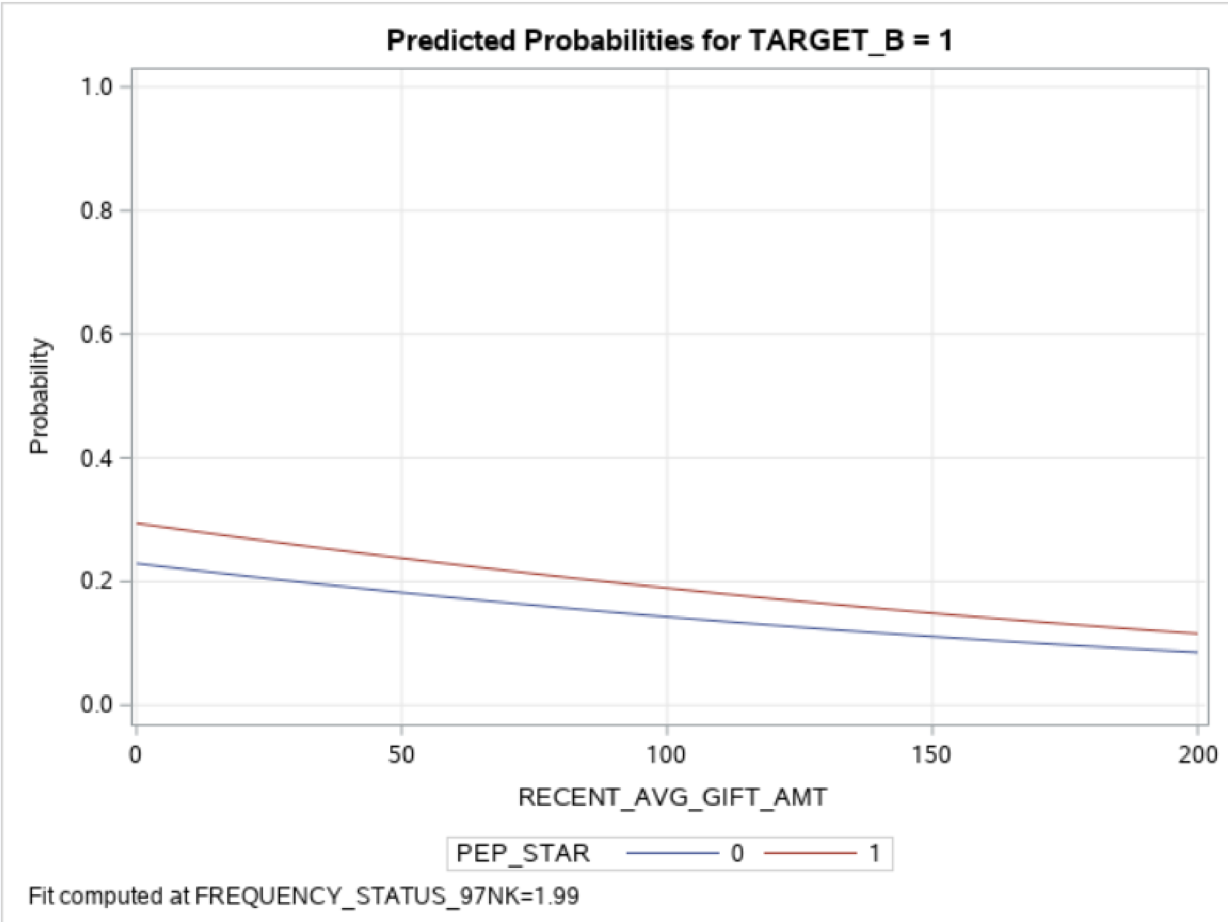
Type 3 Analysis of Effects			
Effect	DF	Wald Chi-Square	Pr > ChiSq
PEP_STAR	1	43.4902	<.0001
RECENT_AVG_GIFT_AMT	1	3.9559	0.0467
FREQUENCY_STATUS_97N	1	83.8209	<.0001

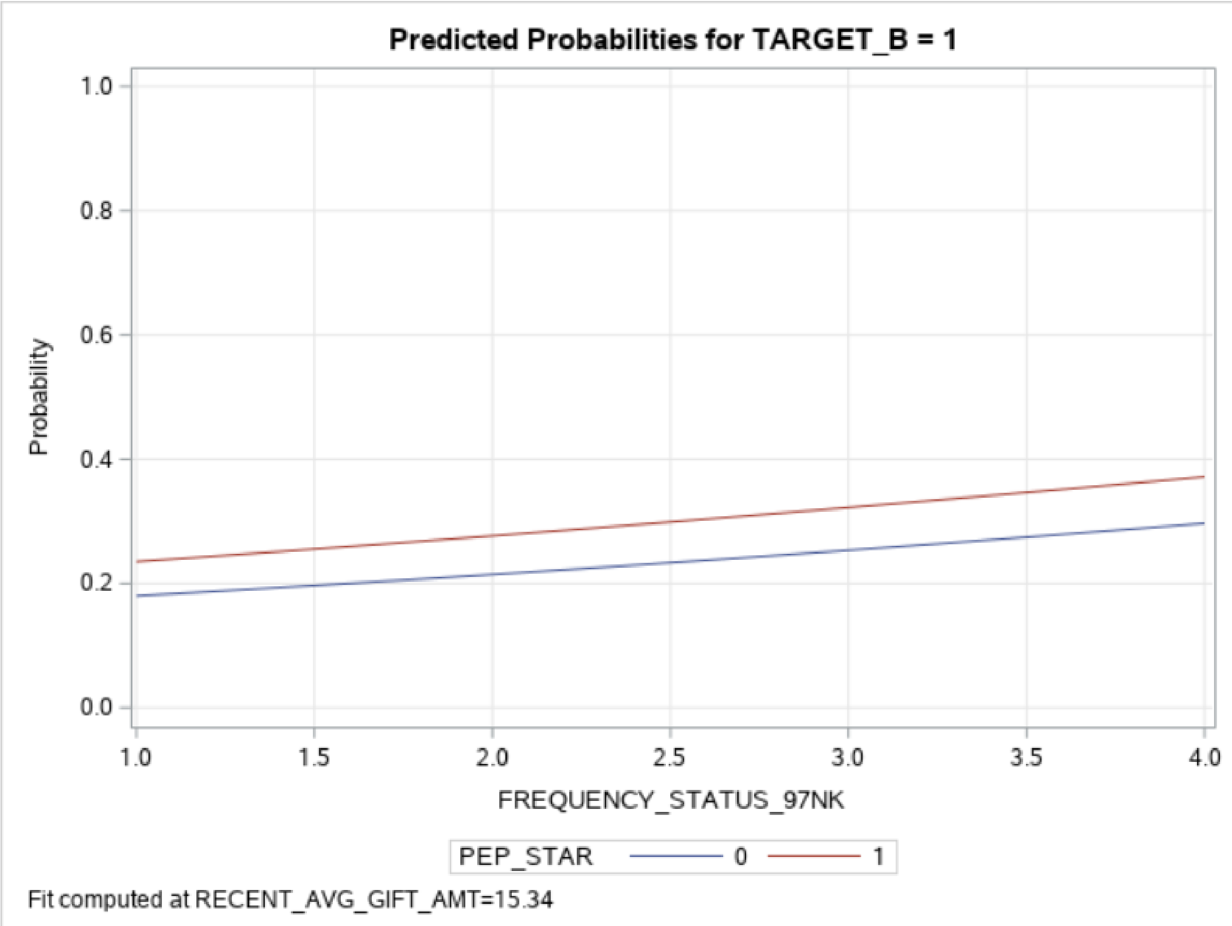
Analysis of Maximum Likelihood Estimates						
Parameter		DF	Estimate	Standard Error	Wald Chi-Square	Pr > ChiSq
Intercept		1	-1.6454	0.0831	392.4480	<.0001
PEP_STAR	1	1	0.3371	0.0511	43.4902	<.0001
RECENT_AVG_GIFT_AMT		1	-0.00579	0.00291	3.9559	0.0467
FREQUENCY_STATUS_97N		1	0.2179	0.0238	83.8209	<.0001

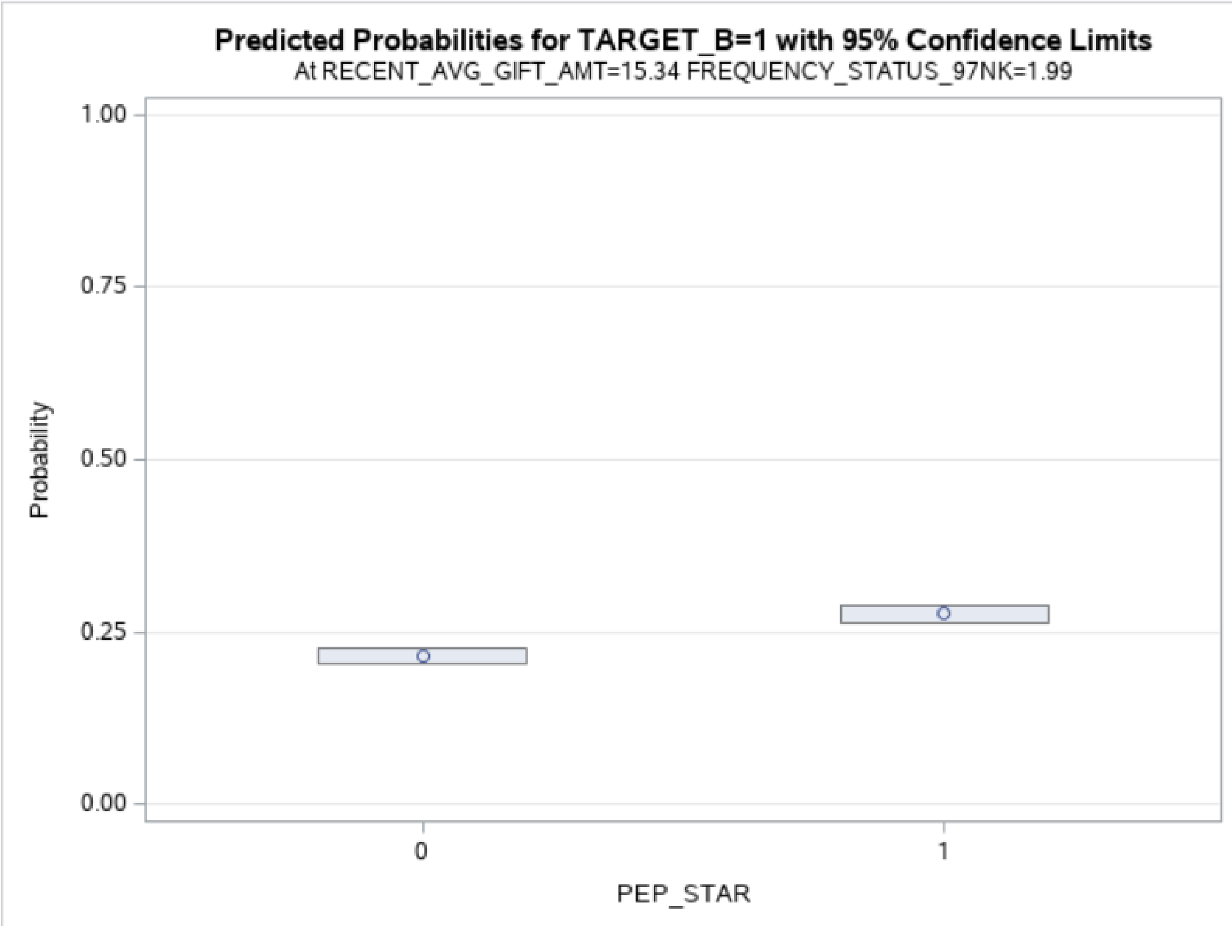
Association of Predicted Probabilities and Observed Responses			
Percent Concordant	59.9	Somers' D	0.208
Percent Discordant	39.0	Gamma	0.211
Percent Tied	1.1	Tau-a	0.078
Pairs	17595830	c	0.604

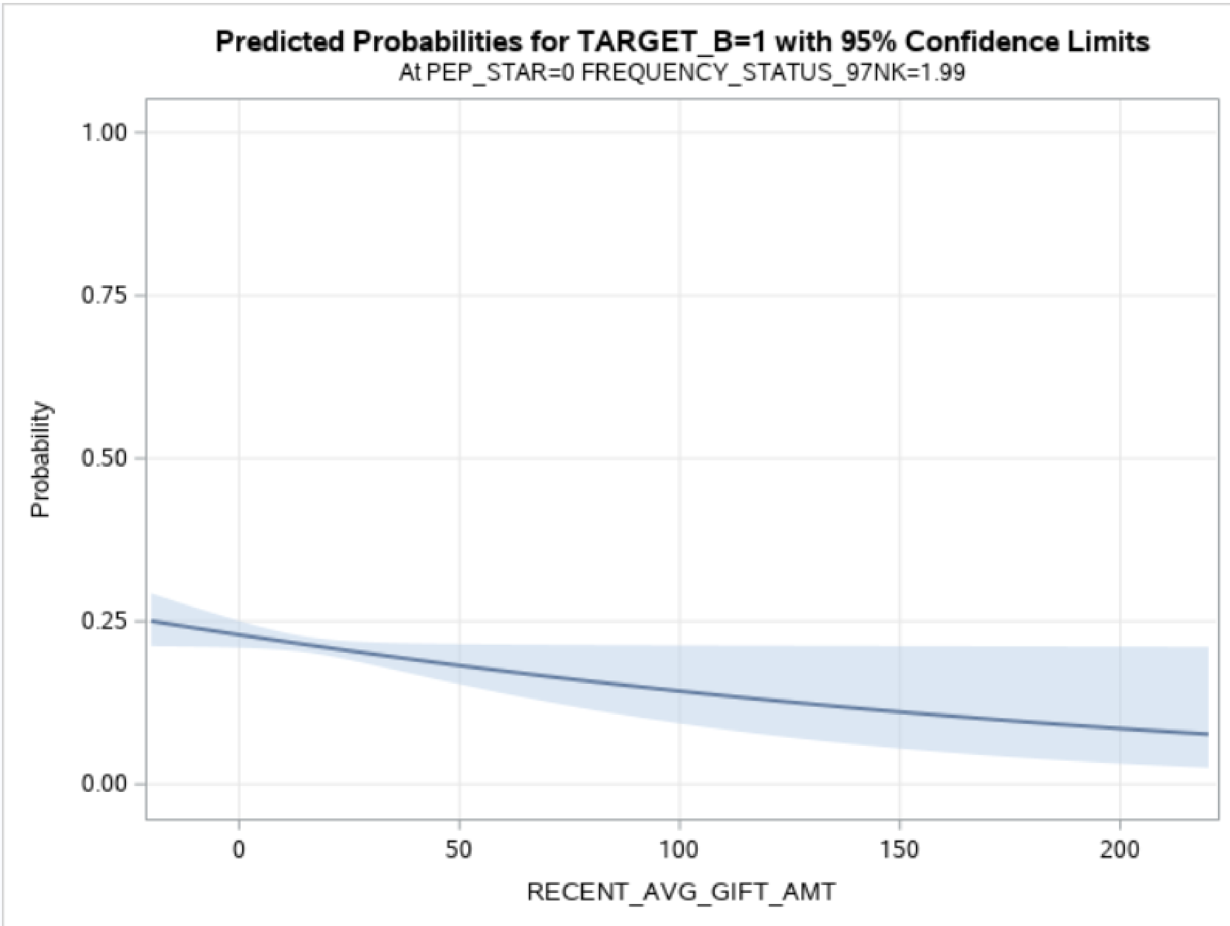
Odds Ratio Estimates and Profile-Likelihood Confidence Intervals				
Effect	Unit	Estimate	95% Confidence Limits	
PEP_STAR 1 vs 0	1.0000	1.401	1.267	1.549
RECENT_AVG_GIFT_AMT	1.0000	0.994	0.988	1.000
FREQUENCY_STATUS_97N	1.0000	1.243	1.187	1.303

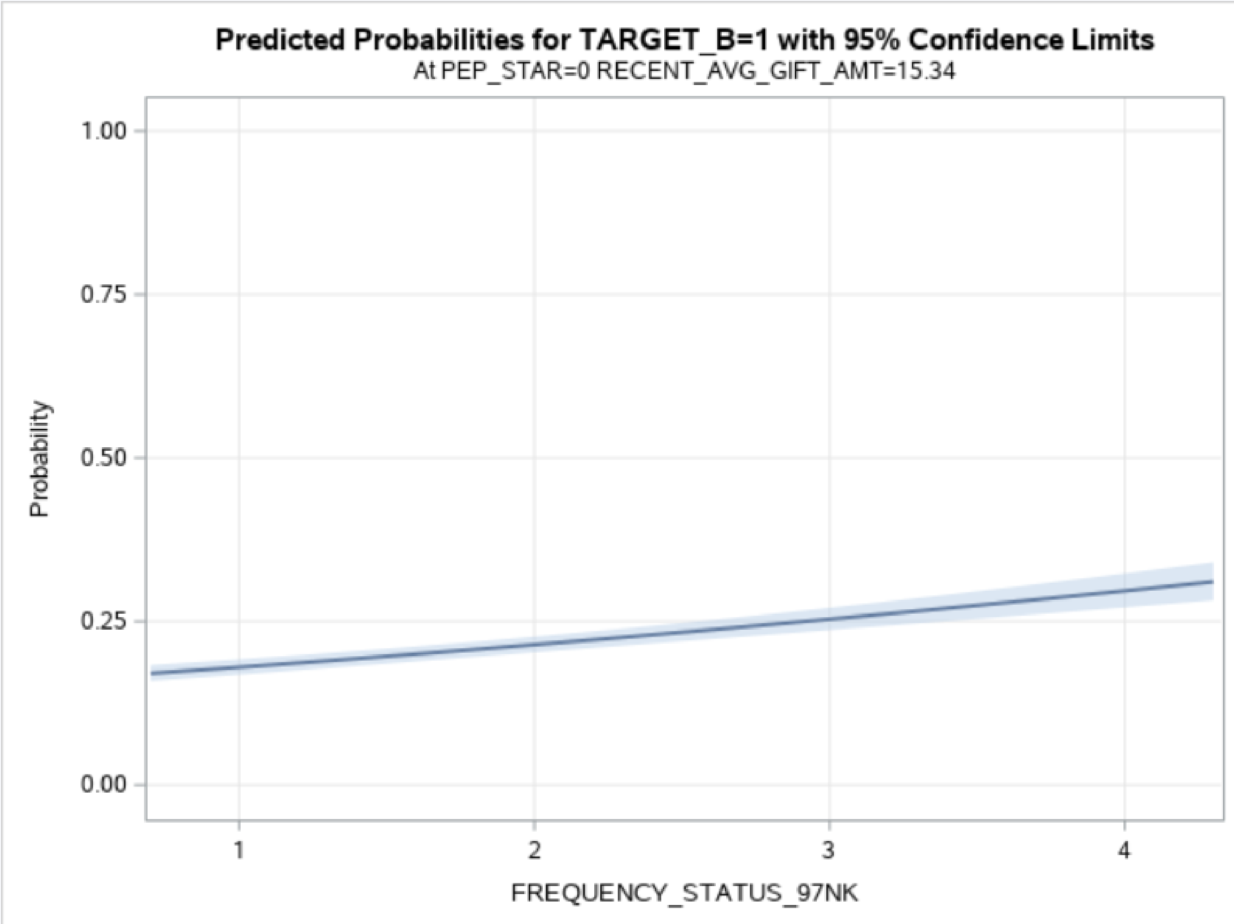



Predicted Probabilities for TARGET_B = 1
Probability
1.0
0.8
0.6
0.4
0.2
0.0
0
50
100
150
200
RECENT_AVG_GIFT_AMT
PEP_STAR
0
1
Fit computed at FREQUENCY_STATUS_97NK=1.99


Predicted Probabilities for TARGET_B = 1
Probability
1.0
0.8
0.6
0.4
0.2
0.0
1.0
1.5
2.0
2.5
3.0
3.5
4.0
FREQUENCY_STATUS_97NK
PEP_STAR
0
1
Fit computed at RECENT_AVG_GIFT_AMT=15.34


Predicted Probabilities for TARGET_B=1 with 95% Confidence Limits
At RECENT_AVG_GIFT_AMT=15.34 FREQUENCY_STATUS_97NK=1.99
1.00
0.75
0.50
0.25
0.00
Probability
0
1
PEP_STAR


Predicted Probabilities for TARGET_B=1 with 95% Confidence Limits
At PEP_STAR=0 FREQUENCY_STATUS_97NK=1.99
1.00
0.75
0.50
0.25
0.00
Probability
0
50
100
150
200
RECENT_AVG_GIFT_AMT



Adjusted Predicted Probabilities of the Veteran's Organization Data

Obs	P_1	PEP_STAR	RECENT_AVG_GIFT_AMT	FREQUENCY_STATUS_97NK
1	0.046390	1	15.00	1
2	0.033094	0	17.50	1
3	0.064890	0	8.33	4
4	0.090167	1	5.00	4
5	0.059152	1	8.33	2
6	0.058117	1	11.57	2
7	0.046941	1	12.86	1
8	0.031733	0	25.00	1
9	0.045126	1	20.00	1
10	0.032091	0	23.00	1

```
/* Solution for l3p1 */

/* step 2 */

data pmlr.pva_train_mi(drop=i);
  set pmlr.pva_train;
  /* name the missing indicator variables */
  array mi{*} mi_DONOR_AGE mi_INCOME_GROUP
          mi_WEALTH_RATING;
  /* select variables with missing values */
  array x{*} DONOR_AGE INCOME_GROUP WEALTH_RATING;
  do i=1 to dim(mi);
    mi{i}=(x{i}=.);
    nummiss+mi{i};
  end;
run;


/* step 3 */

proc rank data=pmlr.pva_train_mi out=work.pva_train_rank groups=3;
  var recent_response_prop recent_avg_gift_amt;
  ranks grp_resp grp_amt;
run;


/* step 4 */

proc sort data=work.pva_train_rank out=work.pva_train_rank_sort;
```

```
  by grp_resp grp_amt;
run;


/* step 5 */

proc stdize data=work.pva_train_rank_sort method=median
        reponly out=pmlr.pva_train_imputed;
  by grp_resp grp_amt;
  var DONOR_AGE INCOME_GROUP WEALTH_RATING;
run;


/* step 6 */

options nolabel;
proc means data=pmlr.pva_train_imputed median;
  class grp_resp grp_amt;
  var DONOR_AGE INCOME_GROUP WEALTH_RATING;
run;
options label;
```

The MEANS Procedure

grp_resp	grp_amt	N Obs	Variable	Median
0	0	487	DONOR_AGE INCOME_GROUP WEALTH_RATING	65.0000000 4.0000000 5.0000000
	1	1147	DONOR_AGE INCOME_GROUP WEALTH_RATING	58.0000000 4.0000000 5.0000000
	2	1612	DONOR_AGE INCOME_GROUP WEALTH_RATING	58.0000000 4.0000000 6.0000000
1	0	671	DONOR_AGE INCOME_GROUP WEALTH_RATING	65.0000000 4.0000000 4.5000000
	1	1270	DONOR_AGE INCOME_GROUP WEALTH_RATING	59.0000000 4.0000000 5.0000000
	2	1202	DONOR_AGE INCOME_GROUP WEALTH_RATING	57.0000000 4.0000000 5.0000000
2	0	2155	DONOR_AGE INCOME_GROUP WEALTH_RATING	63.0000000 4.0000000 5.0000000
	1	733	DONOR_AGE INCOME_GROUP WEALTH_RATING	61.0000000 4.0000000 6.0000000
	2	410	DONOR_AGE INCOME_GROUP WEALTH_RATING	58.5000000 4.0000000 6.0000000

Practice: Imputing Missing Values

For the veterans' organization project, impute missing values for several variables in the **pmlr.pva_train** data set.

Reminder: If you started a new SAS session, you must run **setup.sas** to define the **pmlr** library before you do this practice.

<u>**Step 1**</u>: Open **l3p01_runFirst.sas** from the **practices** folder and run the code.

<u>**Step 2**</u>: Open **l3p01.sas** in your SAS software. Write a DATA step that creates missing value indicators for the following inputs in the **pmlr.pva_train** data set: **Donor_Age**, **Income_Group**, and **Wealth_Rating**. Also add a cumulative count of the missing values. Name the output data set **pmlr.pva_train_mi**. Highlight and submit the DATA step you wrote and check the log.

<u>Step 3</u>: In your program, view the code for step 3. This program uses PROC RANK to group the values of the variables **Recent_Response_Prop** and **Recent_Avg_Gift_Amt** into three groups each. Note that this code creates an output data set named **work.pva_train_rank**. Highlight and submit the step 3 code and check the log.

<u>Step 4</u>: Sort the **work.pva_train_rank** data set by **Grp_Resp** and **Grp_Amt**. Name the output data set **work.pva_train_rank_sort**. Submit the code and check the log to verify that the code ran without errors.

<u>Step 5</u>: To impute missing values in the **work.pva_train_rank_sort** data set for each BY group and create an output data set named **pmlr.pva_train_imputed**, add a PROC STDIZE step with a BY statement. Submit the code and check the log.

<u>Step 6</u>: Use PROC MEANS to determine the values that were used to replace the missing values in the **pmlr.pva_train_imputed** data set. Add OPTIONS statements to display variable names instead of labels in the output from PROC MEANS (using the NOLABEL option) and then to reset the display of labels. Submit the code and look at the results.

For **Grp_Resp**=0 and **Grp_Amt**=0, what value replaced the missing value of **Donor_Age**?

The results indicate that, for **Grp_Resp**=0 and **Grp_Amt**=0, the missing value for **Donor_Age** was replaced with the value *65*.

For the complete solution code, open **l3p1_s.sas** from the **practices/solutions** folder.