How Do I Clean My Data Using SAS Programming?

Ask the Expert

Jacqueline Johnson, Principal Analytical Training Consultant

Jacqueline Johnson

Principal Analytical Training Consultant

Jacqueline Johnson works with faculty at academic campuses around the country conducting software training to develop the future analytics workforce. Prior to joining SAS, she focused on statistical analyses of clinical trials data, including as a biostatistics faculty member in a medical school and a biostatistician in the pharmaceutical industry. Jacqueline has taught with SAS in commercial and academic settings for 15 years.

How Do I Clean My Data Using SAS Programming?

Ask the Expert

Jacqueline Johnson, Principal Analytical Training Consultant

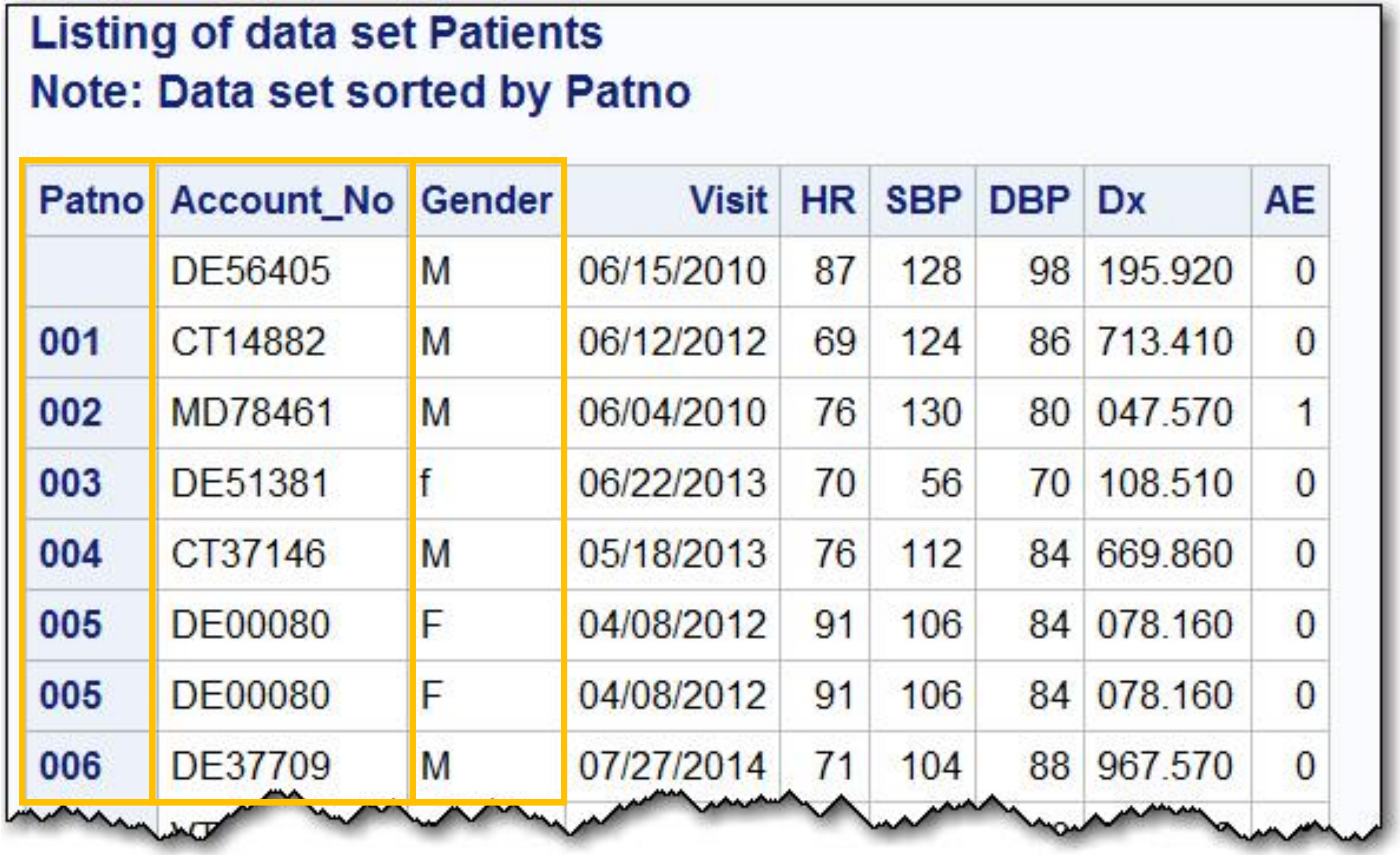
What Is Data Cleaning?

- Data cleaning involves:
  - verifying that the raw data was entered accurately into a dataset.
  - checking that character variables contain only valid values.
  - checking that numeric values are within predetermined ranges.
  - checking for missing values for variables where complete data is expected.
  - checking for duplicate data entries, eliminating duplicate data entries, or both.

What Is Data Cleaning? (cont.)

- Data cleaning involves:
  - checking for uniqueness of certain values such as patient IDs.
  - checking for invalid date values.
  - checking for a unique identifier (such as ID) in multiple SAS data sets.
  - standardizing character values such as company names or addresses.
  - ensuring that certain text value conform to a standard pattern (such as zip codes or phone numbers).
  - correcting errors that were found.
  - creating integrity constraints and audit trails.

Example: Checking Values of Character Variables



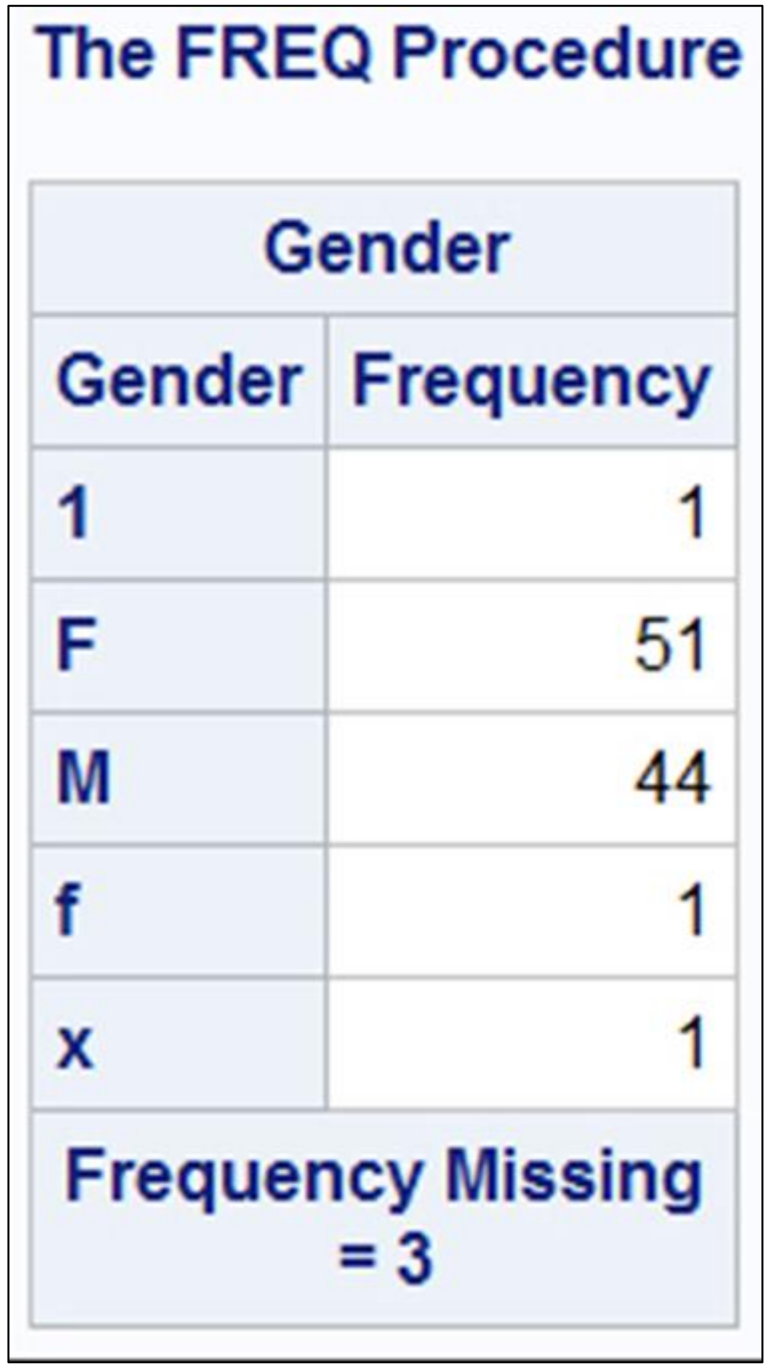
Listing of data set Patients
Note: Data set sorted by Patno

Patno	Account_No	Gender	Visit	HR	SBP	DBP	Dx	AE
	DE56405	M	06/15/2010	87	128	98	195.920	0
001	CT14882	M	06/12/2012	69	124	86	713.410	0
002	MD78461	M	06/04/2010	76	130	80	047.570	1
003	DE51381	f	06/22/2013	70	56	70	108.510	0
004	CT37146	M	05/18/2013	76	112	84	669.860	0
005	DE00080	F	04/08/2012	91	106	84	078.160	0
005	DE00080	F	04/08/2012	91	106	84	078.160	0
006	DE37709	M	07/27/2014	71	104	88	967.570	0

#1 Using PROC FREQ to Detect Character Data Errors

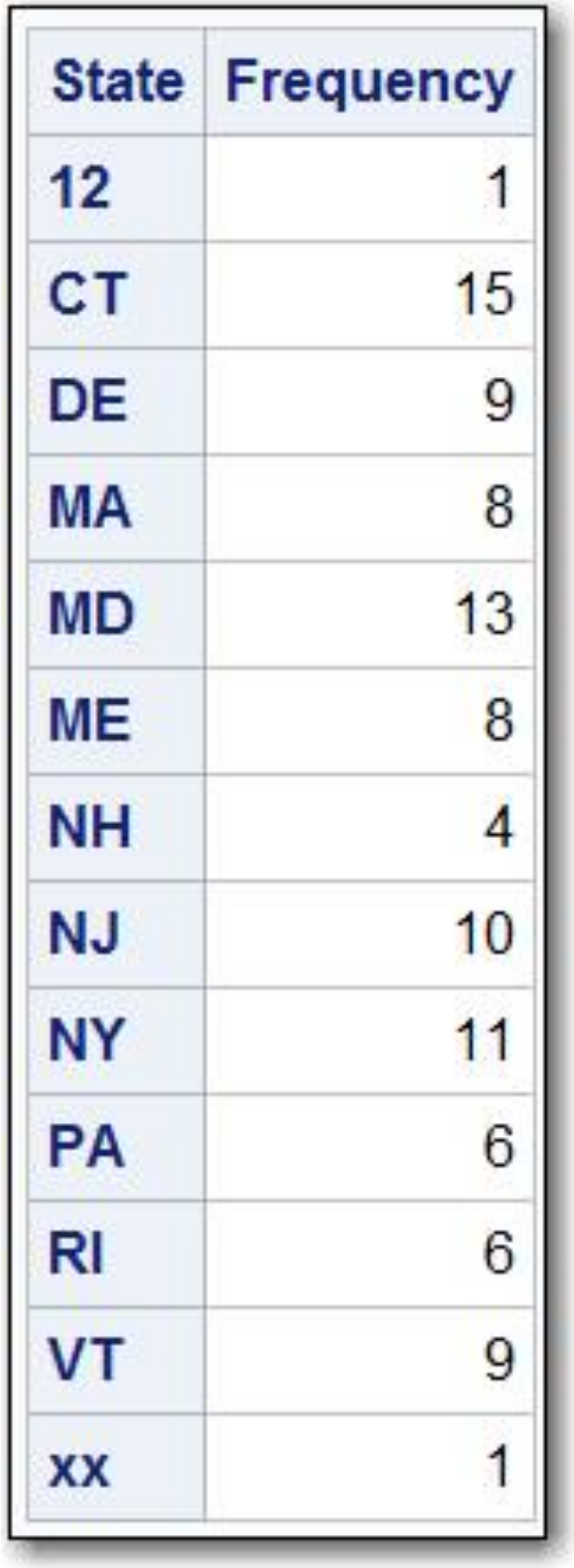
```
data Check_Char;
   set clean.Patients(keep=Patno Account_No Gender);
   length State $ 2;
   State = Account_No;
run;

proc freq data=Check_Char;
   tables Gender State / nocum nopercent;
run;
```

#1 Output from PROC FREQ



The FREQ Procedure

Gender	
Gender	**Frequency**
1	1
F	51
M	44
f	1
x	1
Frequency Missing = 3	

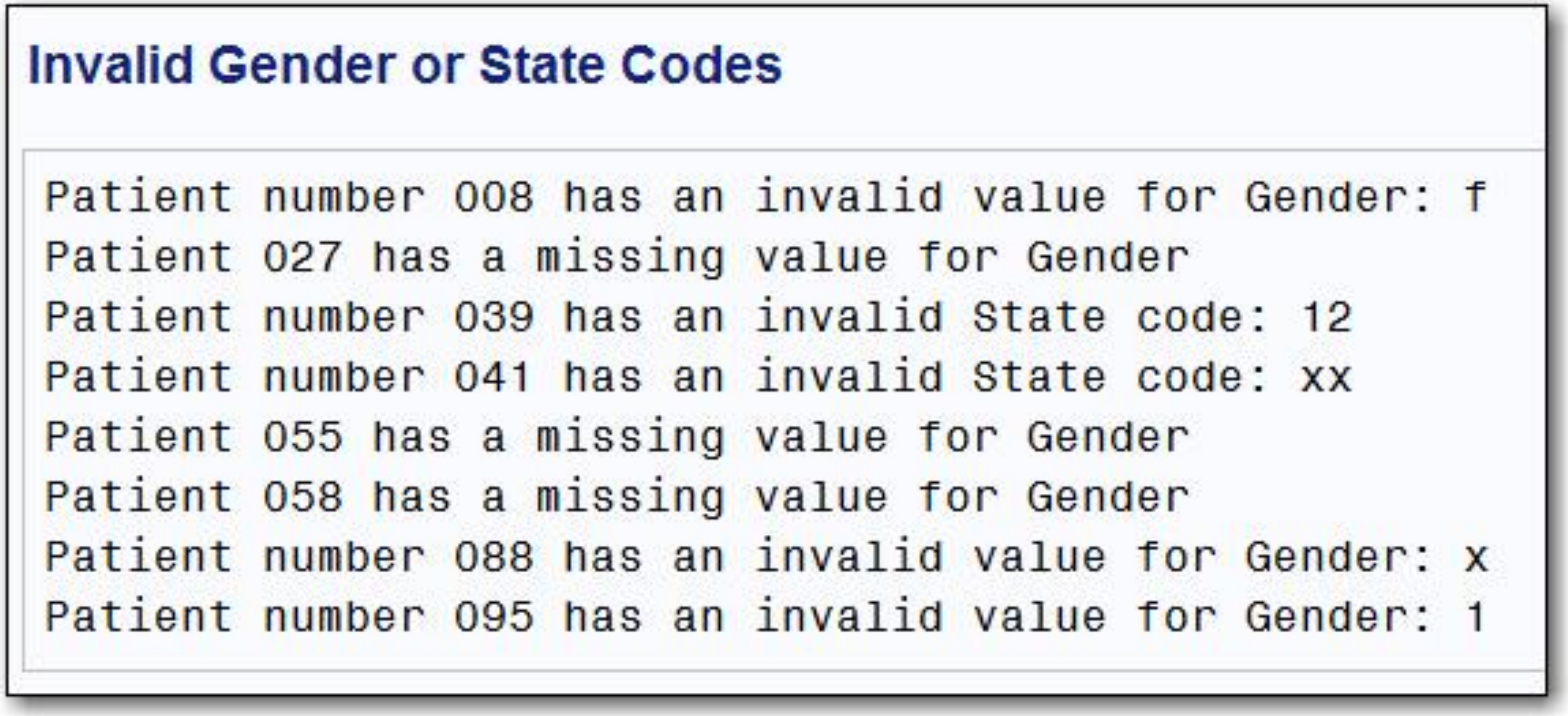


State	Frequency
12	1
CT	15
DE	9
MA	8
MD	13
ME	8
NH	4
NJ	10
NY	11
PA	6
RI	6
VT	9
xx	1

#2 Using the DATA Step to Detect Character Data Errors

```
data _null_;

   file print;
   set clean.Patients(keep=Patno Gender Account_No);
   length State $ 2;
   State = Account_No;

   *Checking value of Gender;
   if missing(Gender) then put
      "Patient " Patno "has a missing value for Gender";
   else if Gender not in ('M','F') then put "Patient number "
       Patno "has an invalid value for Gender: " Gender;

   *Checking for invalid State abbreviations;
   if State not in ('NJ','NY','PA','CT','DE','VT','NH',
       'ME','RI','MA','MD') then put
       "Patient number " Patno "has an invalid State code: " State;
run;
```

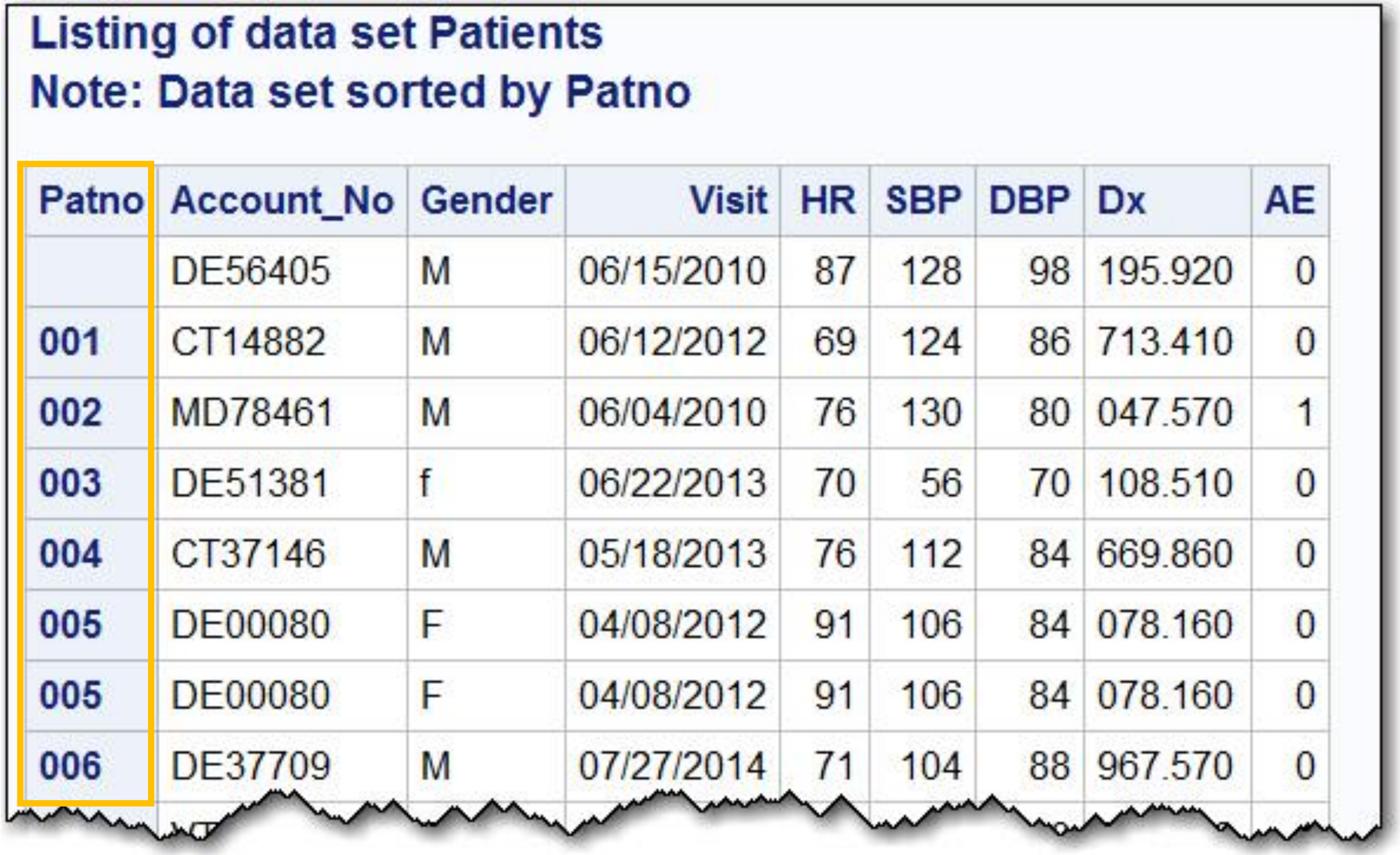
#2 Using the DATA Step to Detect Character Data Errors



Invalid Gender or State Codes

```
Patient number 008 has an invalid value for Gender: f
Patient 027 has a missing value for Gender
Patient number 039 has an invalid State code: 12
Patient number 041 has an invalid State code: xx
Patient 055 has a missing value for Gender
Patient 058 has a missing value for Gender
Patient number 088 has an invalid value for Gender: x
Patient number 095 has an invalid value for Gender: 1
```

#3 Using the PRINT Procedure to List Invalid Values



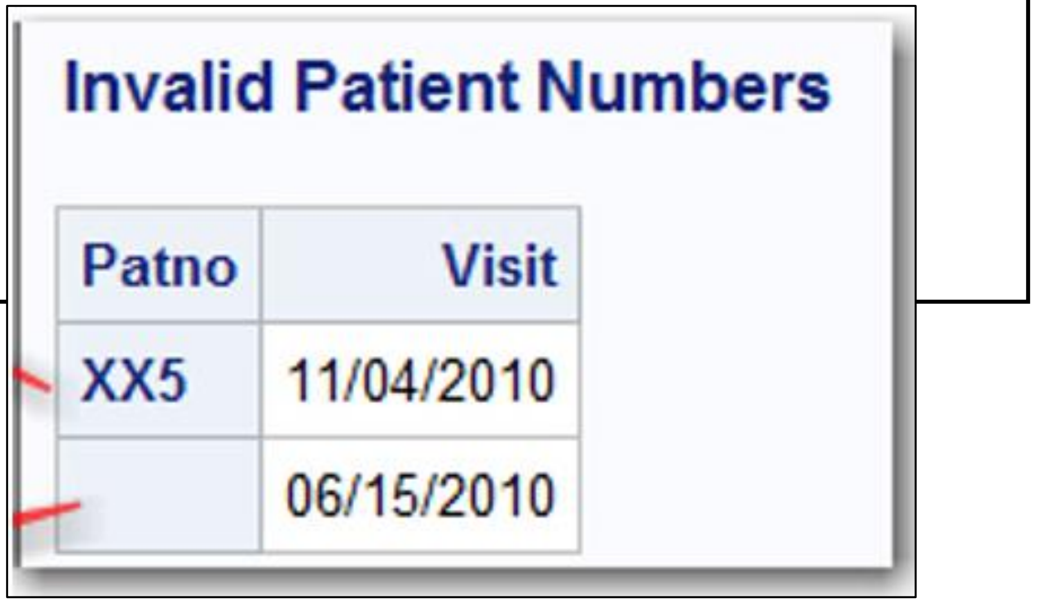
Listing of data set Patients
Note: Data set sorted by Patno

Patno	Account_No	Gender	Visit	HR	SBP	DBP	Dx	AE
	DE56405	M	06/15/2010	87	128	98	195.920	0
001	CT14882	M	06/12/2012	69	124	86	713.410	0
002	MD78461	M	06/04/2010	76	130	80	047.570	1
003	DE51381	f	06/22/2013	70	56	70	108.510	0
004	CT37146	M	05/18/2013	76	112	84	669.860	0
005	DE00080	F	04/08/2012	91	106	84	078.160	0
005	DE00080	F	04/08/2012	91	106	84	078.160	0
006	DE37709	M	07/27/2014	71	104	88	967.570	0

#3 Checking the Patient Numbers

- The patient numbers are three digits and are stored as character data.
- You can use the NOTDIGIT function to test for invalid patient numbers.

```
title "Invalid Patient Numbers";
proc print data=Clean.Patients;
   where notdigit(Patno);
   id Patno;
   var Visit;
run;
```



Invalid Patient Numbers

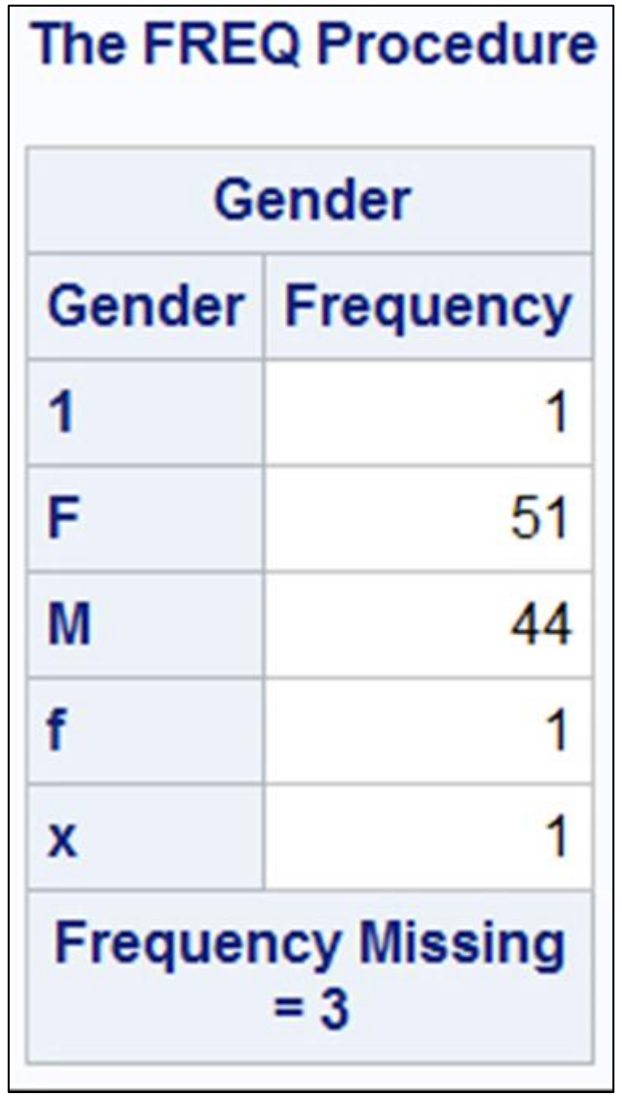
Patno	Visit
XX5	11/04/2010
	06/15/2010

Other NOT Functions

Function
Notalpha
Notalnum
Notpunct
Notspace
Notalpha

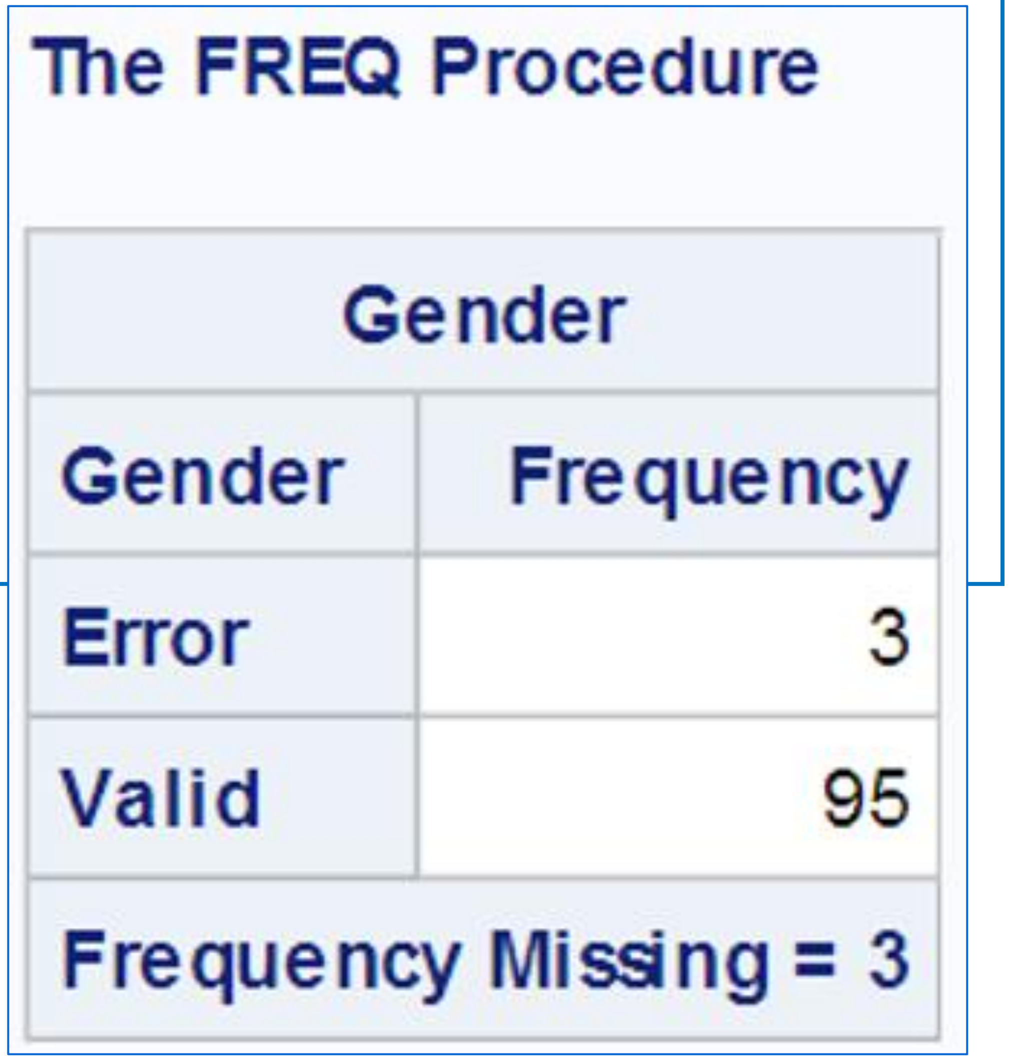
#4 Using PROC FORMAT to Check for Invalid Values

```
proc format;
   value $Gender_Check 'M','F' = 'Valid'
                       ' '     = 'Missing'
                       other   = 'Error';
run;

proc freq data=Clean.Patients;
   tables Gender / nocum nopercent;
   format Gender $Gender_Check.;
run;
```



The FREQ Procedure

Gender	
Gender	Frequency
1	1
F	51
M	44
f	1
x	1
Frequency Missing = 3	



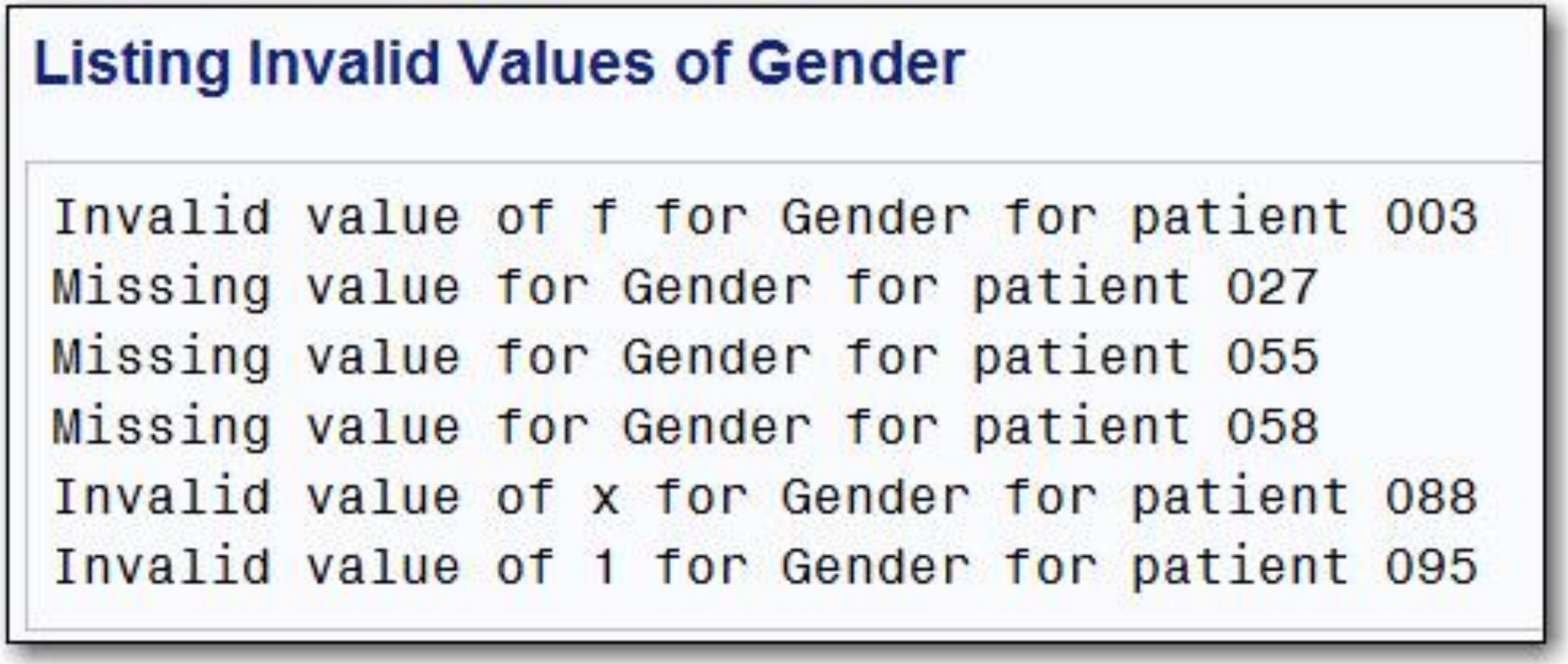
The FREQ Procedure

Gender	
Gender	Frequency
Error	3
Valid	95
Frequency Missing = 3	

#4 Using PROC FORMAT to Check for Invalid Values

```
proc format;
   value $Gender_Check 'M','F' = 'Valid'
                       ' '     = 'Missing'
                       other   = 'Error';
run;
data _null_;
   set Clean.Patients(keep=Patno Gender);
   file print;
   if put(Gender,$Gender_Check.) = 'Missing' then put
      "Missing value for Gender for patient " Patno;
   else if put(Gender,$Gender_Check.) = 'Error' then put
      "Invalid value of " Gender "for Gender for patient " Patno;
run;
```

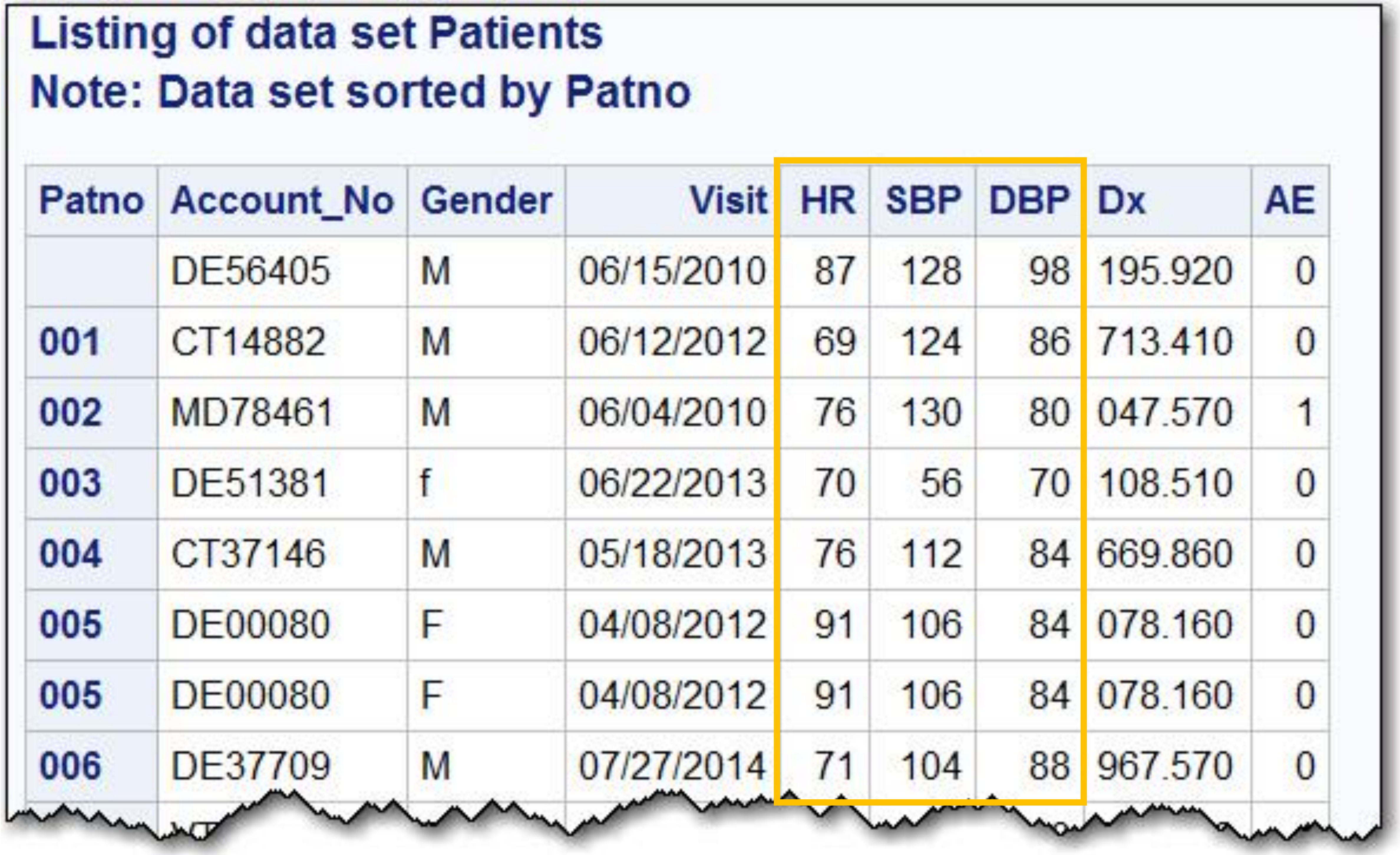
#4 Using PROC FORMAT to Check for Invalid Values



Listing Invalid Values of Gender

```
Invalid value of f for Gender for patient 003
Missing value for Gender for patient 027
Missing value for Gender for patient 055
Missing value for Gender for patient 058
Invalid value of x for Gender for patient 088
Invalid value of 1 for Gender for patient 095
```

Example: Checking Values of Numeric Variables



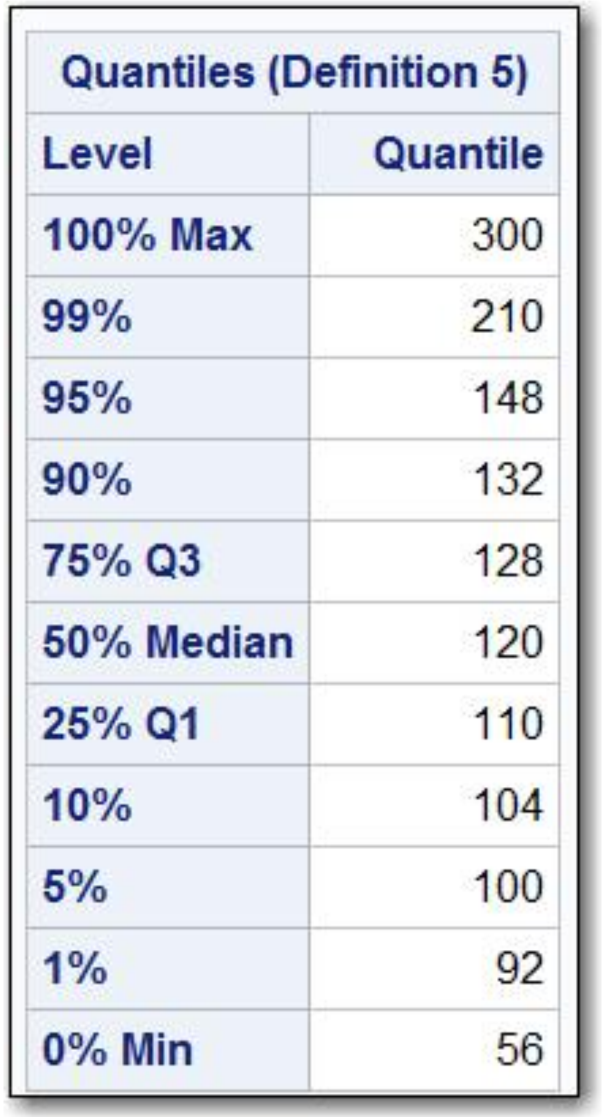
Listing of data set Patients

Note: Data set sorted by Patno

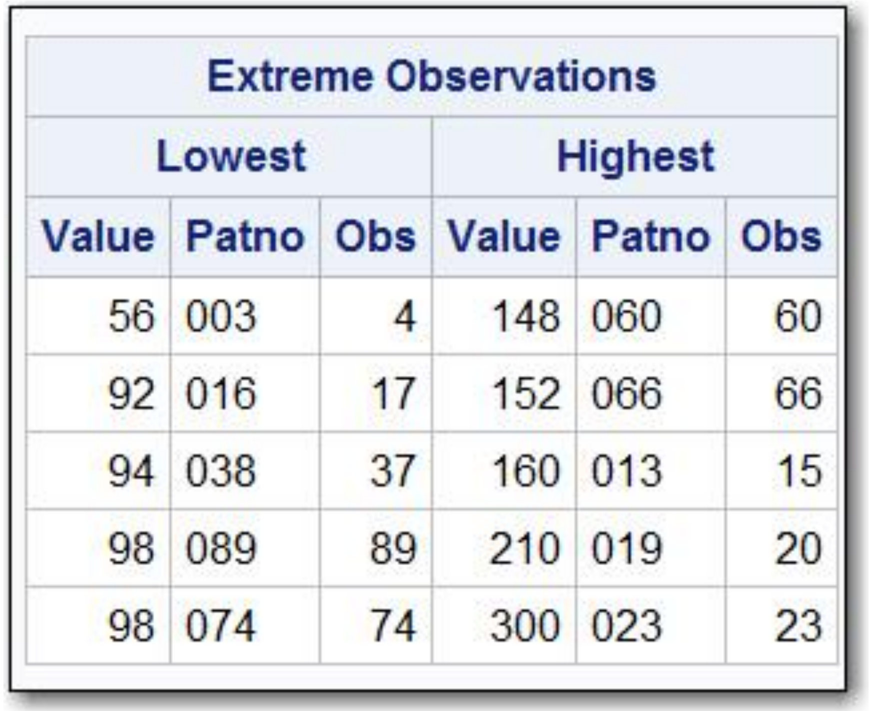
Patno	Account_No	Gender	Visit	HR	SBP	DBP	Dx	AE
	DE56405	M	06/15/2010	87	128	98	195.920	0
001	CT14882	M	06/12/2012	69	124	86	713.410	0
002	MD78461	M	06/04/2010	76	130	80	047.570	1
003	DE51381	f	06/22/2013	70	56	70	108.510	0
004	CT37146	M	05/18/2013	76	112	84	669.860	0
005	DE00080	F	04/08/2012	91	106	84	078.160	0
005	DE00080	F	04/08/2012	91	106	84	078.160	0
006	DE37709	M	07/27/2014	71	104	88	967.570	0

Running PROC UNIVARIATE on HR, SBP, and DBP

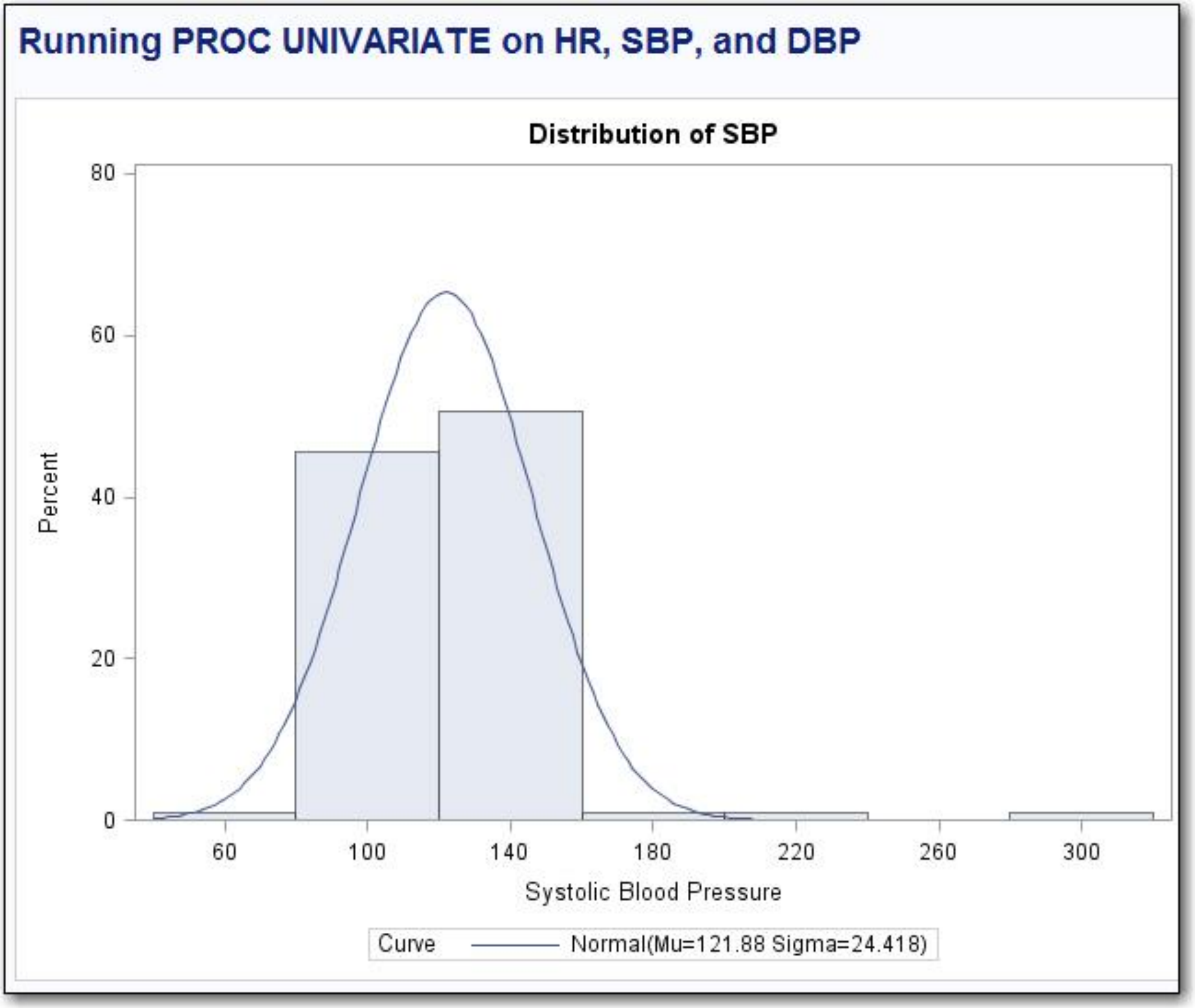
```
proc univariate data=Clean.Patients;
   id Patno;
   var HR SBP DBP;
   histogram / normal;
run;
```



Quantiles (Definition 5)	
Level	Quantile
100% Max	300
99%	210
95%	148
90%	132
75% Q3	128
50% Median	120
25% Q1	110
10%	104
5%	100
1%	92
0% Min	56



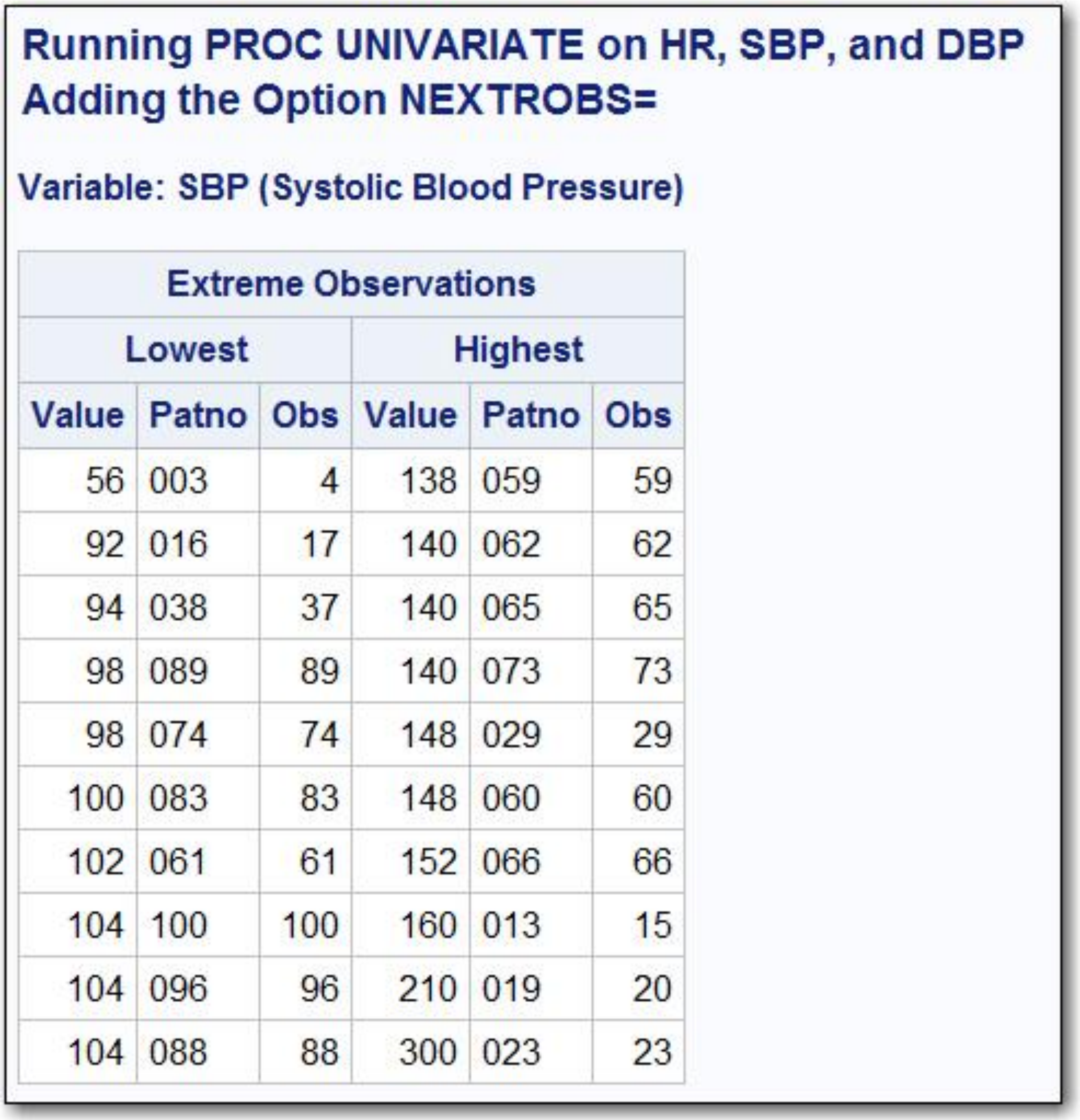
Extreme Observations					
Lowest			Highest		
Value	Patno	Obs	Value	Patno	Obs
56	003	4	148	060	60
92	016	17	152	066	66
94	038	37	160	013	15
98	089	89	210	019	20
98	074	74	300	023	23



Running PROC UNIVARIATE on HR, SBP, and DBP

Running PROC UNIVARIATE on HR, SBP, and DBP

```
ods select ExtremeObs;
proc univariate data=Clean.Patients nextrobs=10;
   id Patno;
   var HR SBP DBP;
   histogram / normal;
run;
```



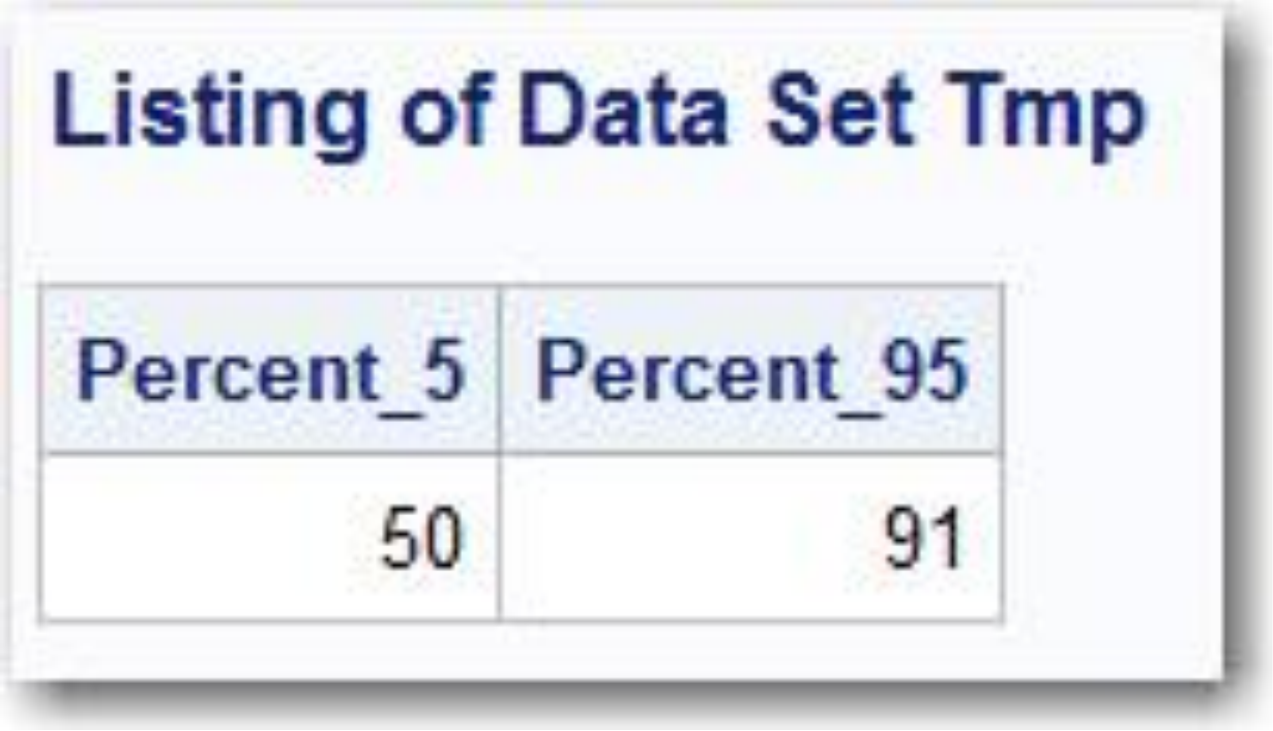
Running PROC UNIVARIATE on HR, SBP, and DBP
Adding the Option NEXTROBS=

Variable: SBP (Systolic Blood Pressure)

Extreme Observations					
Lowest			Highest		
Value	Patno	Obs	Value	Patno	Obs
56	003	4	138	059	59
92	016	17	140	062	62
94	038	37	140	065	65
98	089	89	140	073	73
98	074	74	148	029	29
100	083	83	148	060	60
102	061	61	152	066	66
104	100	100	160	013	15
104	096	96	210	019	20
104	088	88	300	023	23

Listing the Top and Bottom 5%

```
proc univariate data=Clean.Patients noprint;
   var HR;
   id Patno;
   output out=Tmp pctlpts=5 95
                  pctlpre = Percent_;
run;
```



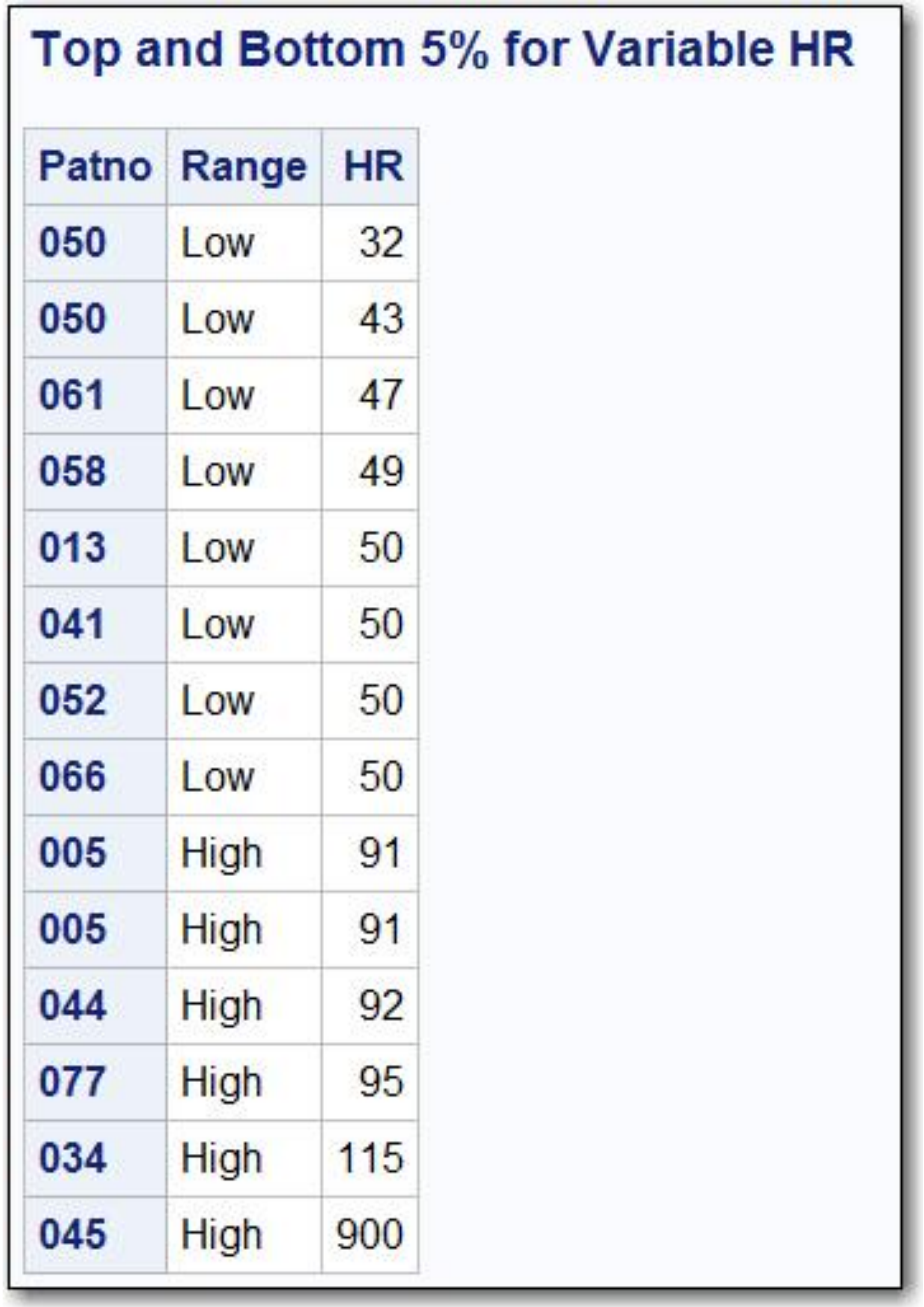
Listing of Data Set Tmp

Percent_5	Percent_95
50	91

Listing the Top and Bottom 5%

```
data HighLowPercent;
   set Clean.Patients(keep=Patno HR);
   *Bring in upper and lower cutoffs for variable;
   if _n_ = 1 then set Tmp;

   if HR le Percent_5 and not missing(HR) then do;
      Range = 'Low ';
      output;
   end;

   else if HR ge Percent_95 then do;
      Range = 'High';
      output;
   end;
run;

proc sort data=HighLowPercent;
   by HR;
run;
```

Listing the Top and Bottom 5%



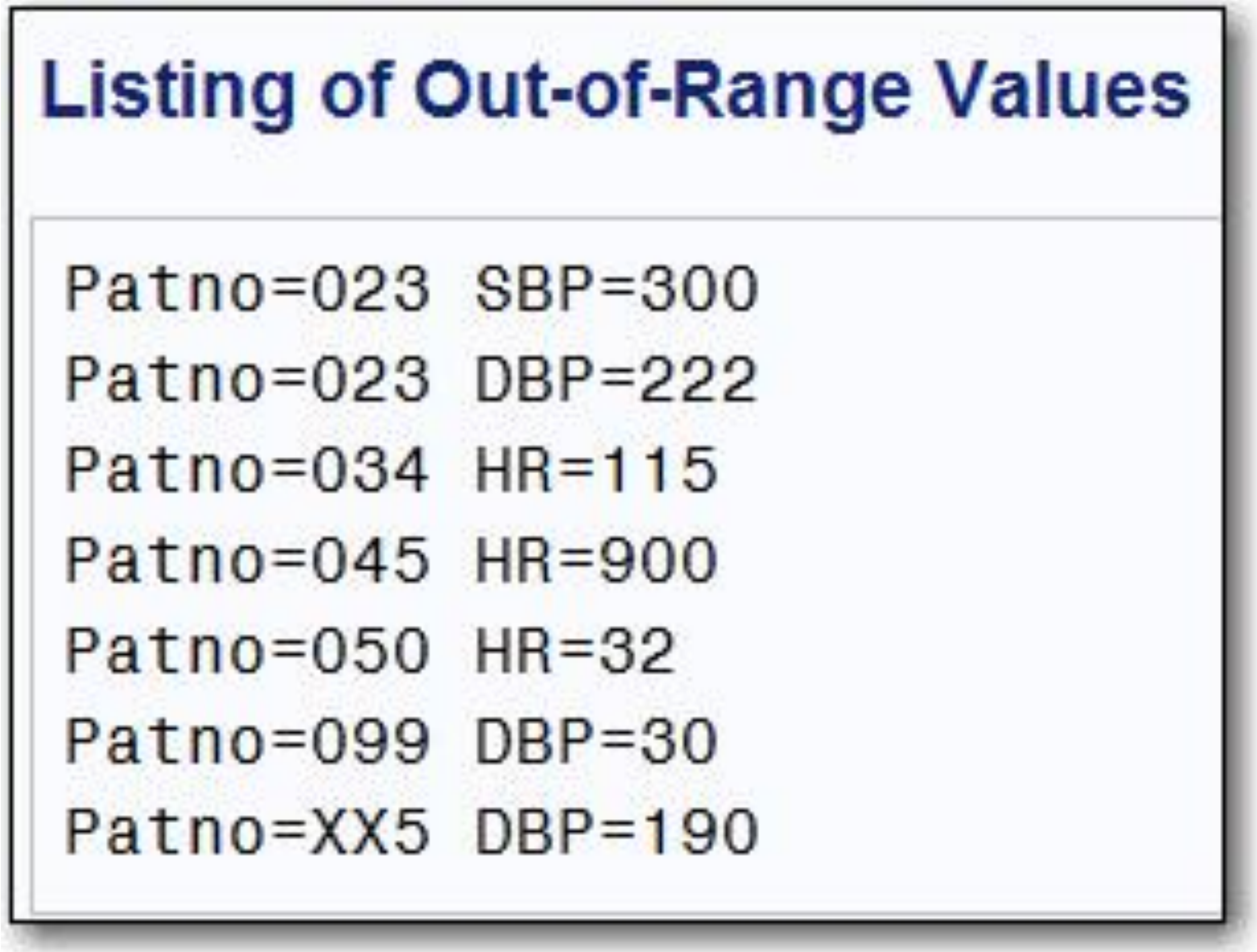
Top and Bottom 5% for Variable HR

Patno	Range	HR
050	Low	32
050	Low	43
061	Low	47
058	Low	49
013	Low	50
041	Low	50
052	Low	50
066	Low	50
005	High	91
005	High	91
044	High	92
077	High	95
034	High	115
045	High	900

Listing Out-of-Range Values Using a DATA Step

```
data _null_;
   file print;
   set Clean.Patients(keep=Patno HR SBP DBP);

   *Check HR;
   if (HR lt 40 and not missing(HR)) or
      HR gt 100 then put Patno= HR=;

   *Check SBP;
   if (SBP lt 50 and not missing (SBP)) or
      SBP gt 240 then put Patno= SBP=;

   *Check DBP;
   if (DBP lt 35 and not missing (DBP)) or
      DBP gt 130 then put Patno= DBP=;
run;
```

Listing Out-of-Range Values Using a DATA Step



Listing of Out-of-Range Values

```
Patno=023 SBP=300
Patno=023 DBP=222
Patno=034 HR=115
Patno=045 HR=900
Patno=050 HR=32
Patno=099 DBP=30
Patno=XX5 DBP=190
```

A Caution about Missing Values

- Remember that SAS missing values are logically treated as smaller than any non-missing value.
- Thus, the following statement will list all values below 40, including missing values:

```
if HR lt 40 or HR gt 100 then put Patno= HR=;
```

- If you do not want to include missing values, use this:

```
if (HR lt 40 and not missing(HR)) or
   HR gt 100 then put Patno= HR=;

if (HR ge 0 and HR lt 40) or HR gt 100
   then put Patno= HR=;

if 0 le HR lt 40 or HR gt 100 then
   put Patno= HR=;
```

How Macros Work

Sample Macro

```
%macro demo(Dsn=, Number=)
   title "Listing of Data Set &Dsn";
   proc print data=&Dsn (obs=&Number);
   run;
%mend demo;
```

Calling the Macro

```
%demo(Dsn=Clean.Patients, Number=15)
```

Generated Code

```
title "Listing of Data Set Clean.Patients";
proc print data=Clean.Patients (obs=15);
run;
```

A Macro to List Out-of-Range Data Values

```
%macro range(Dsn=,   /* data set name       */
             Var=,   /* variable to display */
             Low=,   /* low value           */
             High=,  /* high value          */
             Idvar=  /* ID variable         */);

   data _null_;
      set &Dsn(keep=&Idvar &Var);
      file print;

      if (&Var lt &Low and not missing(&Var)) then
         put "The value of &Var for &Idvar " &Idvar
             "is below &Low.";

      else if &Var gt &High then
         put "The value of &Var for &Idvar " &Idvar
             "is above &High.";
   run;

%mend range;
```

A Macro to List Out-of-Range Data Values

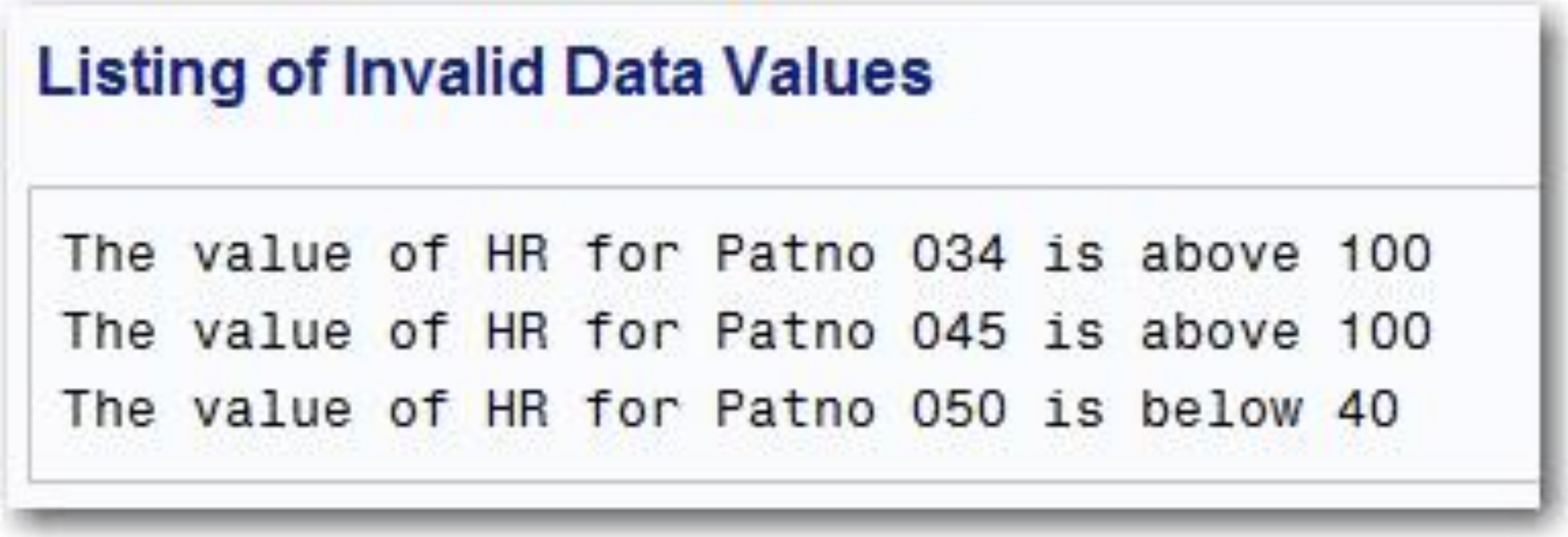
```
%range(Dsn=Clean.Patients, Var=HR,
       Low=40, High=100, Idvar=Patno)
```

After Substitution

```
data _null_;
   set Clean.Patients(keep=Patno HR);
   file print;
   if (HR lt 40 and not missing(HR)) then
      put "The value of HR for Patno " Patno
          "is below 40.";
   else if HR gt 100 then
      put "The value of HR for Patno " Patno
          "is above 100.";
 run;
```

A Macro to List Out-of-Range Data Values

```
%range(Dsn=clean.patients,
       Var=HR,
       Low=40,
       High=100,
       Idvar=Patno)
```

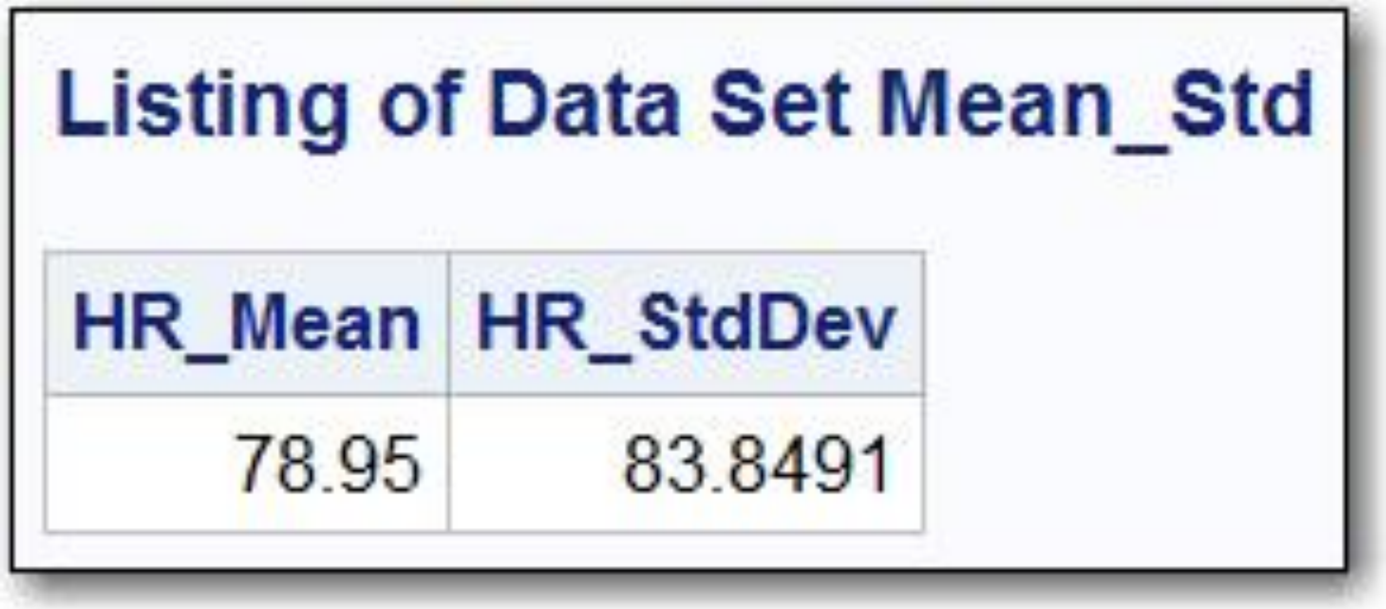


Listing of Invalid Data Values

```
The value of HR for Patno 034 is above 100
The value of HR for Patno 045 is above 100
The value of HR for Patno 050 is below 40
```

Automatic Outlier Detection

```
proc means data=Clean.Patients noprint;
   var HR;
   output out=Mean_Std(drop=_type_ _freq_)
          mean=
          std= / autoname;
run;
```

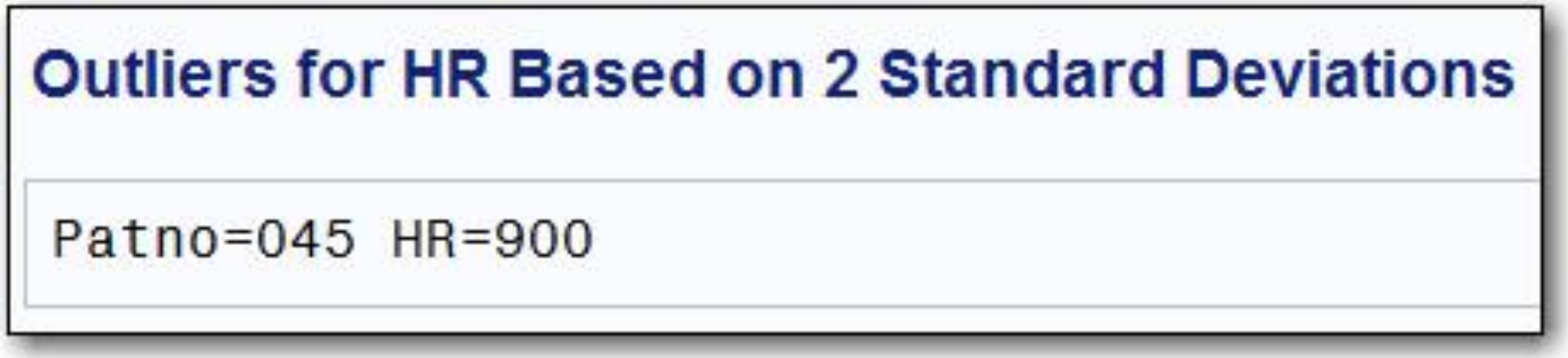


Listing of Data Set Mean_Std

HR_Mean	HR_StdDev
78.95	83.8491

Automatic Outlier Detection

```
data _null_;
   file print;
   set Clean.Patients(keep=Patno HR);

   ***bring in the means and standard deviations;
   if _n_ = 1 then set Mean_Std;

   if (HR lt (HR_Mean - 2*HR_StdDev)) and
      (not missing(HR)) or
      (HR gt (HR_Mean + 2*HR_StdDev)) then

      put Patno= HR=;
run;
```

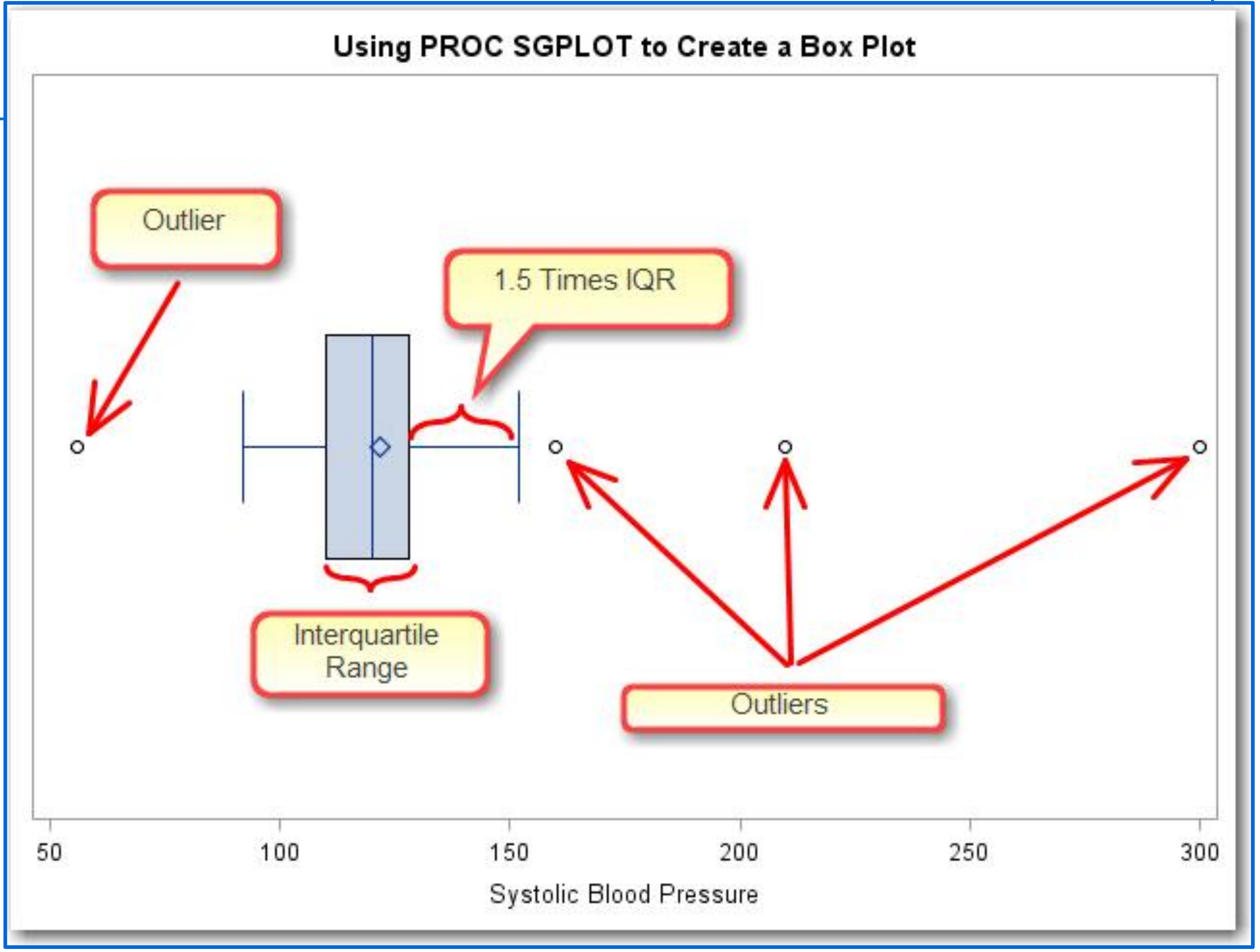


Outliers for HR Based on 2 Standard Deviations

```
Patno=045 HR=900
```

Detecting Outliers Based on the Interquartile Range

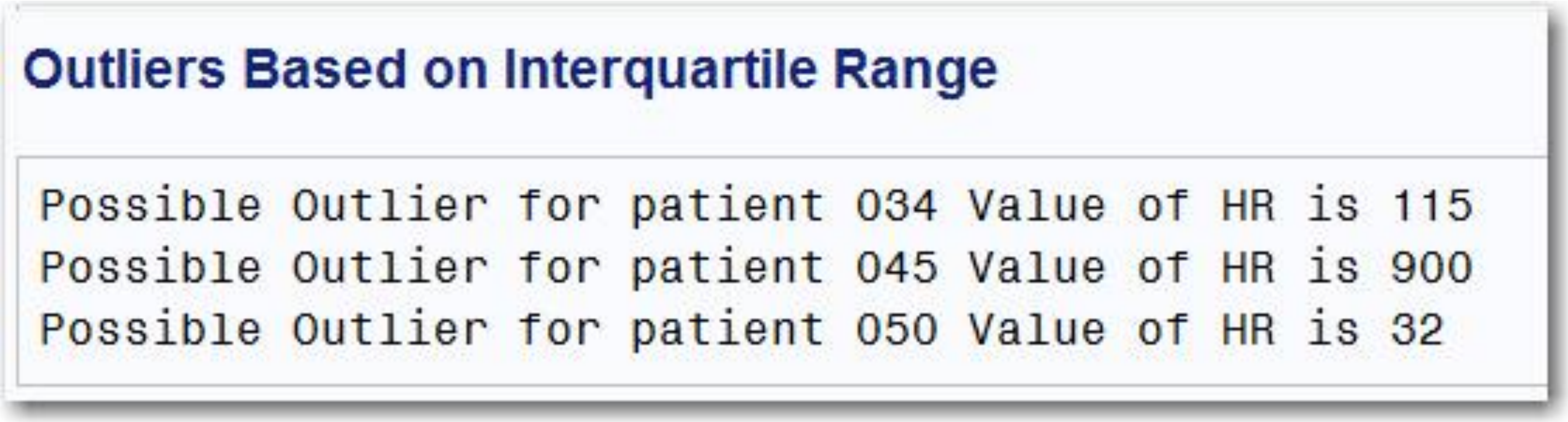
```
proc sgplot data=Clean.Patients(keep=Patno SBP);
   hbox SBP;
run;
```



Detecting Outliers Using the Interquartile Range

```
proc means data=Clean.Patients noprint;
   var HR;
   output out=Tmp Q1= Q3= QRange= / autoname;
run;

data _null_;
   file print;
   set Clean.Patients(keep=Patno HR);

   if _n_ = 1 then set Tmp;

   if (HR le (HR_Q1 - 1.5*HR_Qrange)) and
      (not missing(HR)) or
      (HR ge (HR_Q3 + 1.5*HR_Qrange)) then

      put "Possible Outlier for patient "
      Patno "Value of HR is " HR;
run;
```

Detecting Outliers Using the Interquartile Range



Outliers Based on Interquartile Range

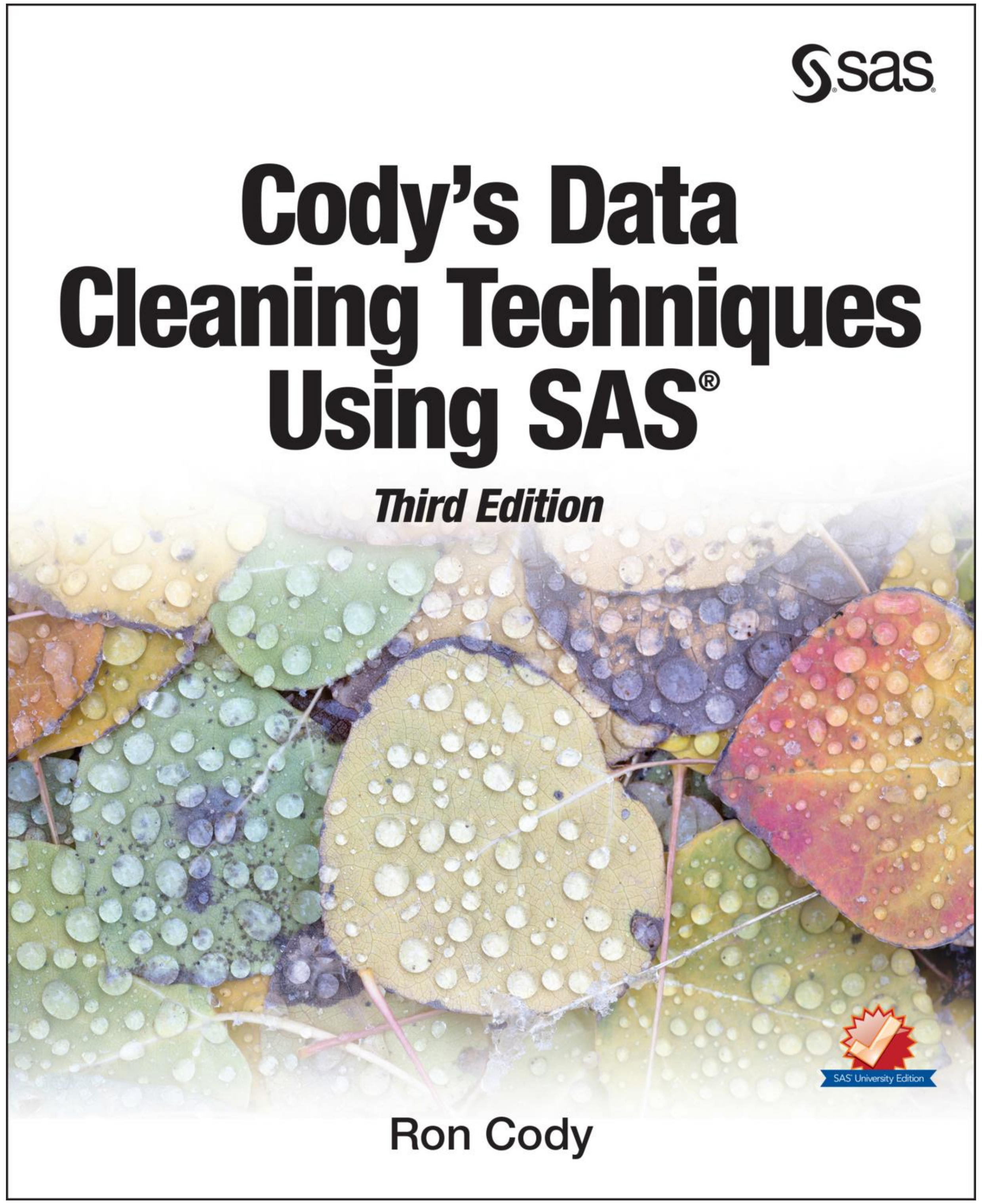
```
Possible Outlier for patient 034 Value of HR is 115
Possible Outlier for patient 045 Value of HR is 900
Possible Outlier for patient 050 Value of HR is 32
```

Additional Data Cleaning Topics

- Using regular expressions to look for character patterns
- Identifying missing values
- Checking dates, in standard and non-standard formats
- Detecting duplicate observations
- Checking for an ID in multiple files
- Comparing datasets
- Adding integrity constants
- Creating an audit trail as changes are made

Continue Your Learning

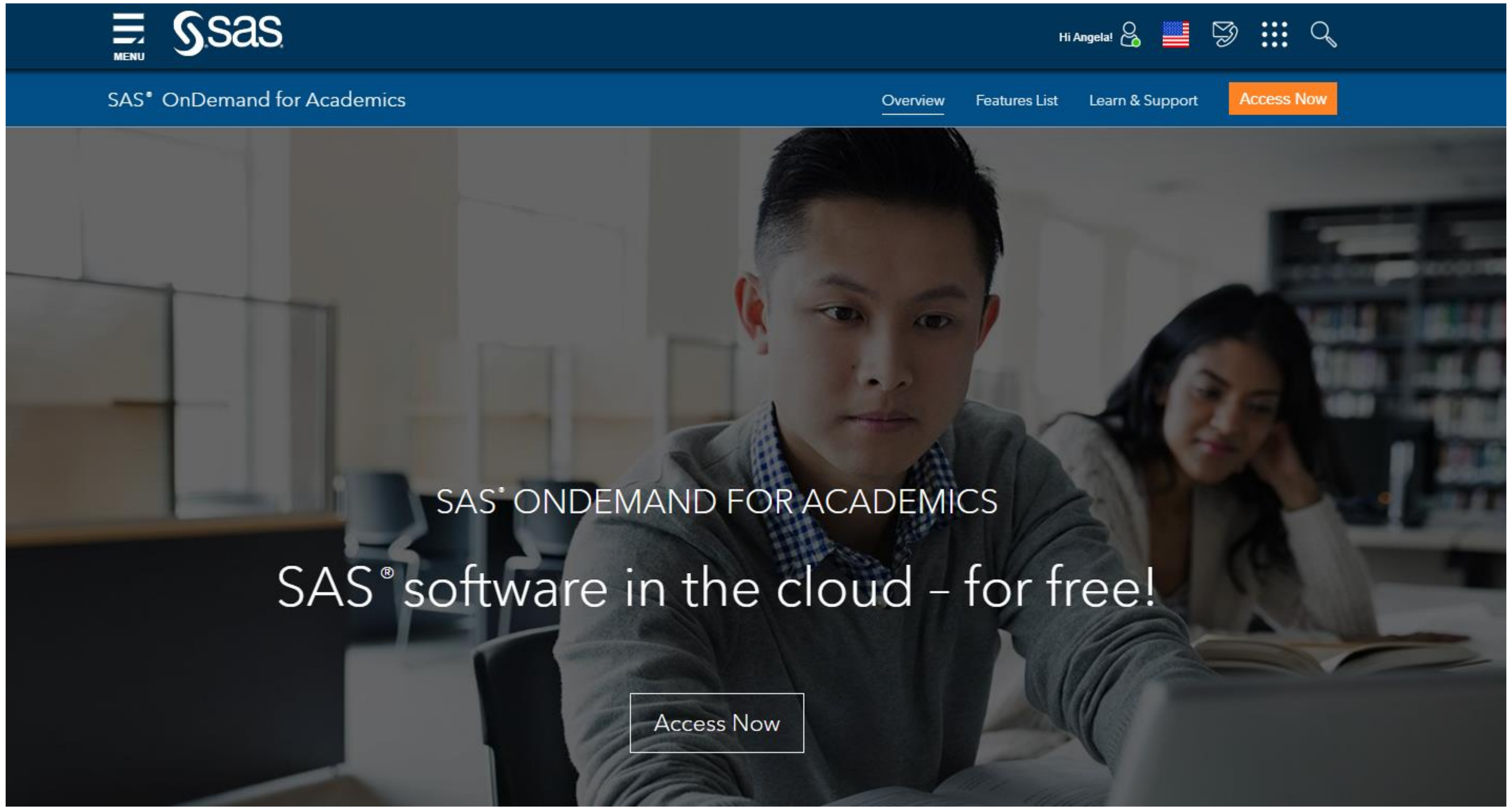
SAS Press Book



- Webinar material comes from this book!
- Available on Redshelf and Amazon.
- Programs and datasets are downloadable for free from the Ron Cody SAS Author Page. Includes several helpful macros!

Continue Your Learning

SAS OnDemand for Academics



SAS® ONDEMAND FOR ACADEMICS

SAS® software in the cloud – for free!

- Free SAS software for students, educators, and independent learners.
- Register at: www.sas.com/ondemand
- Launch at: welcome.oda.sas.com

Q&A