IASCT
Indian Association for Statistics in Clinical Trials

Laxai
Aiding Research, Enhancing Life.

Indian Association for Statistics in Clinical Trials

&

Laxai Avanti Life Sciences Pvt. Ltd. Hyderabad

Welcome You

to

Clinical Trial Data Analysis and Reporting Using SAS Conference

3rd April 2009

www.iasct.net

IASCT

Indian Association for Statistics in Clinical Trials

Indian Association for Statistics in Clinical Trials (IASCT)

Bal Darekar (Quintiles)

April 2009

Background

- Growing interest of major pharma cos in diverting resource for stats-analytical projects to their Indian units.
  - India serves as an important knowledge hub for providing stat-analytics-programming-modeling activities in the clinical trials domain due to its vast talent pool and IT enabled culture
- In last few years, these activities have expanded substantially
  - Pharma MNCs - Pfizer, GSK, Novartis, BMS, Wyeth, Lilly, …
  - CROs/BPOs – Quintiles, Accenture, Parexel, Pharma Net, i3 Statprobe, ICON, PRA, PPD, TCS, Reliance, …
- Need for a platform for the statisticians & programmers to meet, share and learn.
  - Senior management of our parent organizations were supportive.
  - Representatives from Pfizer, BMS, Novartis, GSK, Quintiles and PharmaNet have taken the initiative to establish a framework that is being presented today.

Where are We now

- Indian Association for Statistics in Clinical Trials (IASCT) has been created.
- Objectives, Mission and Vision of this association have been agreed upon.
- Association's Bye-laws are completed and the IASCT is now a registered body in India.

Core Team

- Chitra Lele (ex-Pfizer, presently with Sciformix)
- Prashant Kirkire (ex-Pfizer, presently with i3 Statprobe)
- Ashwini Mathur (Novartis)
- Debjit Biswas (BMS)
- Suresh Bowalekar (PharmaNet)
- Bal Darekar (Quintiles)
- Amit Bhattacharya (GSK)

Mission

Core objectives

1. To enhance awareness about the role of statistics in clinical trials in the medical community, healthcare institutions, pharmaceutical and biotechnology firms, governmental organizations, and educational institutions in India.

2. To promote biostatistics and statistical programming in clinical research as career options for students of statistics and other technical disciplines in India.

3. To enable professional development of statisticians and statistical programmers by organizing training sessions, meetings and conferences relating to statistical techniques used in drug development.

Vision

The vision of IASCT is to grow to an organization that is recognized globally for its role in promoting statistical thinking, and use of appropriate statistical methods in pharmaceutical research and development programs in India and abroad.

Scope of Activities

- We plan activities (long-term) in three main areas
  - Newsletter
  - Events
    - Lecture Series
    - Seminars/ Webinars
    - Talent Showcasing/ Annual Events/ Sponsored Events
    - SAS schools and other training
  - Collaborating with other organizations
    - Statistics/Medical statistics/Biometrics societies in India on a program-by-program basis
    - Offering courses in collaboration with PSI, UK
    - Advisory Representations to Govt. bodies / Industry / Industry associations to influence policy

2007-2008 Events

- **SAS in Pharmaceutical Industry Workshop**

  November 23, 2007 – Bangalore

  December 10, 2007 – Mumbai

- One-day training course on “***Bayesian Biostatistics: An Introduction*** ” by Prof. Sujit Ghosh, NC State Univ. USA

  January 7, 2008 – Bangalore

- Statistics Workshop - “**Role of Statistician in Clinical Trials”**

  June 13, 2008 – Bangalore

  (Sponsor: Accenture & Wyeth)

  June 23, 2008 – Mumbai

  (Sponsor: Cytel India)

- Statistics Workshop - **“Statistics in Phase I”**

  November 14-15 - Pune

  (Sponsor: Cytel India)

IASCT

Indian Association for Statistics in Clinical Trials

IASCT Committee

Events committee	Admin committee
➢Amit Bhattarchayya (GSK) ➢ Debijit Biswas (BMS) ➢ Jagannatha P S (GSK) ➢ Varun Talwar (Sciformix) ➢ Tushar Sakpal (Pharmanet) ➢ Deepak Venkataramana (Wyeth) ➢ Ranganath Bandi (BMS)	➢ Bal Darekar (Quintiles) ➢Ashwini Mathur (Novartis) ➢Geethalakshmi Balakumar(Quintiles) ➢ Jayesh Natarajan(Quintiles) ➢ Shubharekha M.S (GSK) ➢ Vijaykeerthi S (GSK) ➢ Mihir Gandhi(BMS) ➢ Samrat Tatkare (BMS) ➢ Shailaja Chilappagari (Novartis)

IASCT
Indian Association for Statistics in Clinical Trials

Agenda – B'lore

3rd April'09

Laxai
Aiding Research, Enhancing Life.

Topics	Time (Hrs)	Speaker
Tea and Registration	9:30-10:00	
Welcome Note	10:00-10:15	Bal Darekar
Introduction of "Laxai Avanti Life Sciences Pvt. Ltd". Hyderabad	10:15-10:25	
Easy way to read data - Using functions	10:25-10:55	Sarath Surampudi
Effective Use of Proc Tabulate in Clinical Trials	10:55-11:25	Bharat Sharad Yadav
Manipulating Clinical Data with the Power of SAS Arrays.	11:30-12:00	Jino Joseph
Reporting of Adverse Events	12:00-12:30	Megha Kamani & Siva Prasad Mekala
Zero Rows: 5 Ways to Summarize Absolutely Nothing	12:30-13:00	Ramya Deepak
Lunch Break	13:00-14:00	
Safety data graphical displays	14:00-14:30	Vinay Mahajan
Graphs Made easy using SAS/GRAPH® SG Procedures	14:30-15:00	Kanimozhi A
Improving Graphics Using SAS/GRAPH Annotate Facility	15:00-15:30	
Tea Break	15:30-16:00	
Using the power of Regular Expressions to rationalize data and make it consistent	16:00-16:30	Anindita Bhattacharjee and Jayshree Garade
Importance and Methodologies of Validation in Clinical Trials Reporting	16:30-17:00	VijayKeerthi

IASCT
Indian Association for Statistics in Clinical Trials

Agenda – Mumbai

24th April'09

Laxai
Aiding Research, Enhancing Life.

Topics	Time (Hrs)	Speaker
Tea and Registration	9:30-10:00	
Welcome Note	10:00-10:15	Chitra Lele
Introduction of Laxai Avanti Life Sciences Pvt. Ltd. Hyderabad	10:15-10:25	
Data review made easy by Patient Summary listings	10:25-10:55	Murali Mareedu
Handling large datasets and improving the efficiency of SAS Programs	10:55-11:25	Jyoti Dialani
All About Alignment	11:30-12:00	Hemanth Padmakar
Data Validation Methodologies and Concepts Used in SAS in Clinical Trials	12:00-12:30	Bhargavi
Screen Control Language (SCL) Functions and it usage in missing data imputation	12:30-13:00	Jagannatha P S
Lunch Break	13:00-14:00	
Proc Tabulate Introduction	14:00-14:30	Suhas K R
Taking Advantage of Proc Prinito, Data Steps and Proc Gprint	14:30-15:00	Naga Deepthi Mungi
Tips and Tricks in SAS graphics	15:00-15:30	Nitin Pawar and Arvind Chaudhary
Tea Break	15:30-16:00	
Case Report Tabulations	16:00-16:30	Surendra.Kandregula
Application of Trend Analysis in Clinical Trials	16:30-17:00	Jagan Allu

A BIG THANK YOU TO THE EVENTS COMMITTEE

Sponsor: Debjit Biswas & Amit Bhattacharyya

Members: P.S. Jagannatha (Lead, GSK),
Varun Talwar (Sciformix),
Tushar Sakpal (Pharmanet),
Ranganath Bandi (BMS)

Thank you all for your participation today !!

High performance. Delivered.

FUNCTIONS

FUN-ACTIONS

Presented by SARATH

Different type of Functions

- Arithmetic functions (Ex: Max , Min , Sqrt …)
- **Character functions** (Ex: Compbel, Index ..)
- **Date and time functions** (Ex: Date, Mdy…)
- Quantile functions (Ex: Gaminv, Finv ….)
- **Special functions** (Ex: Smallest, Largest..)
- Zip code functions (Ex: Stname, Zipstate..)
- Trigonometric functions (Ex: Tan, Cos ..)
- **Length functions** (Ex: Length , Lengthm ..)
- **Variable information functions** (Ex: Varfmt, Varlen…)
- Non centrality functions (Ex: Fnonct, Cnonct …)
- Probability functions (Ex: Probf, Probbnml …)
- Mathematical functions (Ex: Exp, Log…)
- Statistical functions (Ex: Var, Stderr ,Std ..)
- Truncation functions (Ex: Ceil, Floor …)

How sas stores character data

```
Input name $ string $3.;

    left='x    ';     /* x  and 4 blanks*/
    right='    x';   /* 4 blanks and x */

  sub=substr(name,1,2);
  rep=repeat(name,1);

 Datalines;
 ABCDEFGH  123            (Two blanks)
 XXX               4      (19 blanks)
  Y              5         (20 blanks)
```

name?
string?
left?
right?
sub?
rep?

#	variable	Type	length
1	name	char	8
2	string	char	3
3	left	char	5
4	right	char	5
5	sub	char	8
6	rep	char	200

What's the difference between ?

1) INDE**X** – INDEX**C** – INDEX**W** **=?**

2) CAT – CAT**S** – CAT**T** – CAT**X = ?**

3) LENGTH**C** – LENGTH – LENGTH**M** – LENGTH**N =?**

Example

Obs	string
1	**There is a the in this line**
2	**Ends in the**
3	**Ends in the.**
4	**None here**

Values of :

- Indexc (string," the”);
- Index (string," the”);
- Indexw (string, “the”);

(position_**indexw)**	**(**position_**index)**	**(** position_**indexc)**
12	**1**	**1**
9	**9**	**1**
0	**9**	**1**
0	**0**	**4**

Useful   (CATS):-

A=“Bilbo”
B=“   Frodo   ”

CAT (A,B)	=	“Bilbo Frodo ”
CAT**S** (A,B)	=	“BilboFrodo”
CAT**T** (A,B)	=	“Bilbo Frodo”
CAT**X** (A,B)	=	“Bilbo Frodo”
CAT**X** (“:”,A,B)	=	“Bilbo:Frodo”
CAT**X** (“ Bilbo “)	=	“Bilbo”

LENGTH FUNCTIONS

LENGTH :

- Ex: length (‘ABC‘) = 3
- length (‘ABC ‘) =3
- **length(‘ ‘) =1**
-

LENGTHN :

Ex: lengthn (‘ABC‘) = 3

lengthn (‘ABC ‘) =3

lengthn(‘ ‘) =0

LENGTHC :

- Ex: lengthc (‘ABC‘) = 3
- **lengthc (‘ABC ‘) =6**
- lengthc(‘ ‘) =1

LENGTHM :

Ex: lengthm (‘ABC‘) = 3

lengthm (‘ABC ‘) =6

lengthm(‘ ‘) =1

Special functions : Ynot? (ANY/NOT)

- These functions return the location of the first alphanumeric, letter, digit, punctuation, or space in a character string.
- **ANY**ALPHA:
- Ex: STRING = "ABC 123 ?xyz_n_“

- **Function Returns**
- **ANY**ALPHA**(STRING) 1 (position of "A")**
- **ANY**ALPHA**("??$$%%") 0 (no alpha characters)**
- **ANY**ALPHA**(STRING,5) 10 (position of "x")**
- **ANY**ALPHA**(STRING,6) 10 (position of "x")**

Special functions : Ynot? (ANY/NOT) contin….

- **ANY**DIGIT :

  Ex: STRING = "ABC 123 ?xyz_n_“

- **Function Returns**
- **ANY**DIGIT**(STRING) 5 (position of "1")**
- **ANY**DIGIT**("??$$%%") 0 (no digits)**
- **ANY**DIGIT**(STRING,5) 5 (position of "1")**
- **ANY**DIGIT**(STRING,6) 6 (position of "2")**

Special functions : Ynot? (ANY/NOT) contin….

- **ANY**PUNCT :

  ! " # $ % & ‘ ( ) * + , - . / : ; < = > ? @ [\ ] ^ _ ` { | } ~
- **ANY**SPACE :

  Ex: STRING = "ABC 123 ?xyz_n_“

- **Function Returns**
- **ANY**SPACE**(STRING) 4 (position of the first blank)**
- **ANY**SPACE**("??$$%%") 0 (no spaces)**
- **ANY**SPACE**(STRING,5) 8 (position of the second blank)**

. **ANY**SPACE**(STRING,6) 8 (position of the second blank)**

Special functions : Ynot? (ANY/NOT) contin….

- **NOT**DIGIT :

  STRING = "ABC 123 ?xyz_n_“

- **Function Returns**
- **NOT**DIGIT (STRING) 1 (position of "A")
- **NOT**DIGIT ("123456") 0 (all digits)
- **NOT**DIGIT ("??$$%%") 1 (position of "?")
- **NOT**DIGIT (STRING,5) 8 (position of 2nd blank)

. **NOT**DIGIT (STRING,6) 8 (position of 2nd blank)

Special functions : Ynot? (ANY/NOT) contin….

- **NOT**UPPER:
- Ex: STRING = "ABC 123 ?xyz_n_“

- **Function Returns**
- **NOT**UPPER **("ABCDabcd") 5 (position of "a")**
- **NOT**UPPER**("ABCDEFG") 0 (all uppercase characters)**
- **NOT**UPPER**(STRING) 4 (position of 1st blank)**
- **NOT**UPPER**("??$$%%") 1 (position of "?")**
- **NOT**UPPER**(STRING,5) 5 (position of "1")**
- **NOT**UPPER**(STRING,6) 6 (position of "2")**

Special functions : Ynot? (ANY/NOT) contin….

- **NOT**ALPHA:
- Ex: STRING = "ABC 123 ?xyz_n_“

- **Function Returns**
- **NOT**ALPHA**(STRING) 4 (position of 1st blank)**
- **NOT**ALPHA **("ABCabc") 0 (all alpha characters)**
- **NOT**ALPHA**("??$$%%") 1 (position of first "?")**
- **NOT**ALPHA**(STRING,5) 5 (position of "1")**
- **NOT**ALPHA**(STRING,2) 4 (position of 1st blank)**

Variable information functions

- ***Varlen*(data-set,var)**
- **Use: returns the length of a sas data set variable**
- **Vformat(var)**
- **Use: returns the format associated with the given variable**
- **Vinformat(var)**
- **Use: returns the informat associated with the given variable**

Variable information functions CONTD…

- Vartype(dataset,var)
- Use: returns the data type of a sas dataset variable
- Varfmt(dataset,var)
- Use: returns the format assigned to a sas data set variable

WHAT ARE SAS DATE AND TIME VALUES?

- There are three primary ways of measuring time in the SAS System.
- These are known as DATE, TIME, and DATETIME values.
- DATE values are stored as the number of days that have elapsed since the start of time (January 1, 1960).
- TIME values are the number of seconds that have elapsed since midnight of the current day.

DATE AND TIME VALUES CONTD….

- On July 28, 2004 at 11:32 a.m. the SAS DATE value was 16,280 days since January 1, 1960.

- It was also 41,520 seconds since midnight (the TIME value), and the DATETIME value was 1,406,633,520 seconds since midnight January 1, 1960.

- The DATETIME value counts the number of seconds that have elapsed since midnight of January 1, 1960.

WHAT IS A SAS DATE AND TIME LITERAL?

```
data sampdate;
     sampdate = '28jul2004'd;
     samptime = '11:32't;
    sampdtime= '28jul2004:11:32'dt;

     put sampdate=;
     put samptime=;
     put sampdtime=;
run;
```

The LOG shows:

```
  sampdate=16280
  samptime=41520
  sampdtime=1406633520
```

WORKING WITH INTERVALS

-
  ```
  data age;
    dob = '04jun1975'd;
    age = yrdif(dob,'28jul2004'd,'act/act');
    put dob=;
    put age=;
   run;
  ```

- Log values:
  ```
     dob=5632
    age=29.151860169
  ```

INTNX -----------INTCK

- The INTNX and INTCK functions are also used to calculate intervals and are not limited to counting the number of elapsed years. Both use an argument to specify the type of date/time interval of interest. The INTCK function counts the number of intervals between two dates

EXAMPLE

-
  ```
  data period;
  sampdate = '28jul2004'd;
   yrstart = intnx('year',sampdate,1);
   yrstart2 = intnx('year2',sampdate,1);
  yrstart23 = intnx('year2.3',sampdate,1);
  run;
  ```

- The LOG shows:
  ```
  sampdate=July 28, 2004
  yrstart=01JAN2005
  yrstart2=01JAN2006
  yrstart23=01MAR2006
  ```

EXAMPLE

```
data ageint;
      dob = '04jun1975'd;
      yrs = intck('year',dob,'28jul2004'd);
      months = intck('month',dob,'28jul2004'd);
      weeks = intck('week',dob,'28jul2004'd);
      qtrs = intck('qtr',dob,'28jul2004'd);
run;
```

The LOG shows:

```
yrs=29
months=349
weeks=1521
qtrs=117
```

New functions:

- Small = **Smallest**(2, w,x,y,z);
- LARGE = **Largest**(2,w,x,y,z);

W=0
X=2
Y=7
Z=11

ANS:

- Small = 2
- LARGE = 7

QUESTIONS ?

THANQ

Effective use of PROC TABULATE in Clinical Trials

Bharat Yadav

Biostatistician

Manipal AcuNova Limited

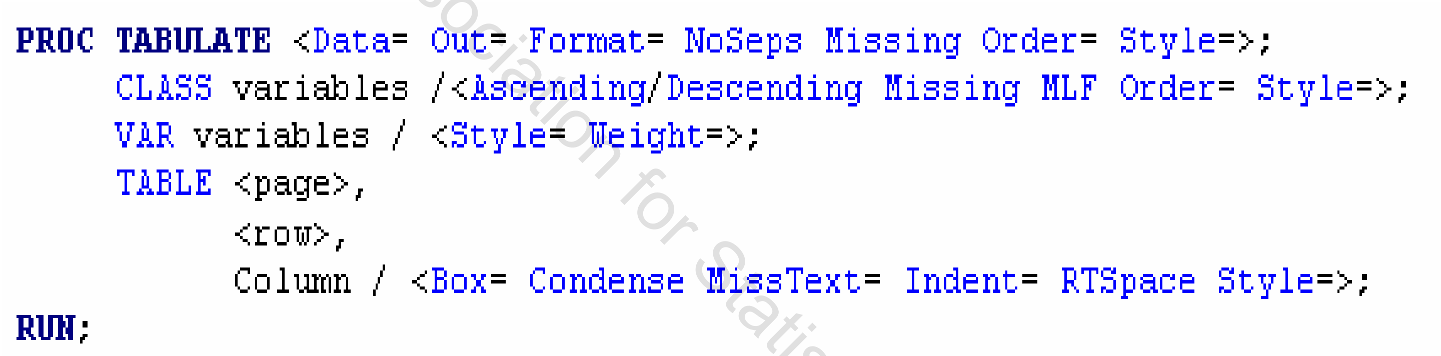
Objective

- This presentation starts with an introduction to PROC TABULATE.
- It looks at the basic syntax, and then builds on this syntax by using examples on how to produce one, two and three-dimensional tables using the TABLE statement.
- It covers how to choose statistics for the table, labeling variables and statistics, how to add totals, missing data and how to clean up the table.
- This presentation provides a simplified, step-by-step approach for coding PROC TABULATE.

Introduction

- PROC TABULATE is a procedure that displays descriptive statistics in tabular format.
- It computes many statistics that other procedures compute, such as MEANS, FREQ, and REPORT and displays them in a table format.
- PROC TABULATE will produce tables in up to three dimensions and allows, within each dimension, multiple variables to be reported one after another hierarchically.
- There are also some nice mechanisms that can be used to label and format the variables and the statistics produced.

Basic Syntax



```
PROC TABULATE <Data= Out= Format= NoSeps Missing Order= Style=>;
     CLASS variables /<Ascending/Descending Missing MLF Order= Style=>;
     VAR variables / <Style= Weight=>;
     TABLE <page>,
           <row>,
           Column / <Box= Condense MissText= Indent= RTSpace Style=>;
RUN;
```

Statistics options

❑ Descriptive statistic keywords

- COLPCTN
- COLPCTSUM
- CSS
- CV
- KURTOSIS |KURT
- LCLM
- MAX
- MEAN
- MIN
- N
- NMISS
- PAGEPCTN
- PAGEPCTSUM
- PCTN
- PCTSUM
- RANGE
- REPPCTN
- REPPCTSUM
- ROWPCTN
- ROWPCTSUM
- SKEWNESS|SKEW
- STDDEV|STD
- STDERR
- SUM
- SUMWGT
- UCLM
- USS
- VAR

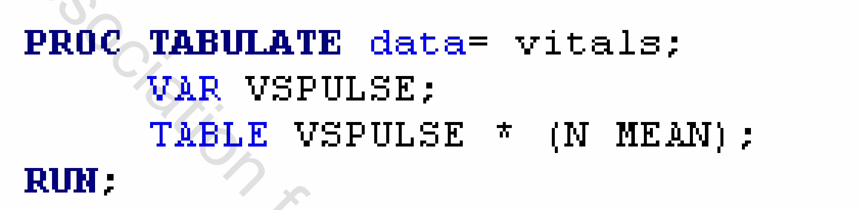
❑ Quantile statistic keywords

- MEDIAN|P50
- P1
- P5
- P10
- Q1|P25
- Q3|P75
- P90
- P95
- P99
- QRANGE

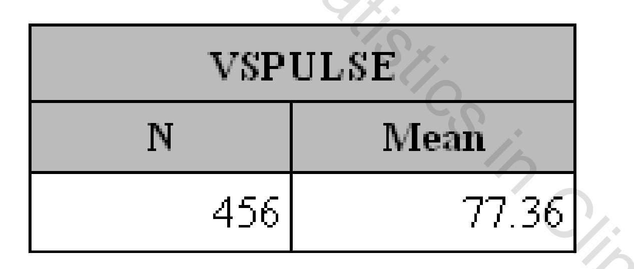
❑ Hypothesis testing keywords

- PROBT
- T

One-Dimensional Table

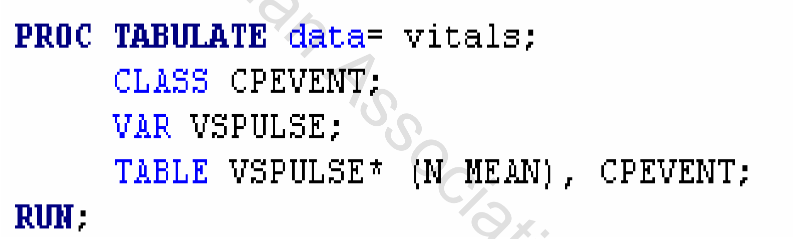


```
PROC TABULATE data= vitals;
     VAR VSPULSE;
     TABLE VSPULSE * (N MEAN);
RUN;
```

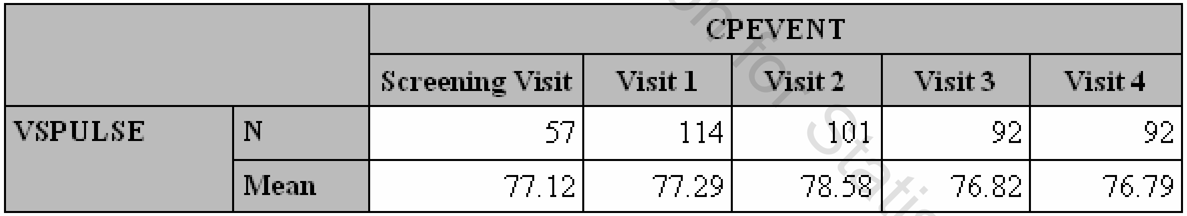


VSPULSE	
N	Mean
456	77.36

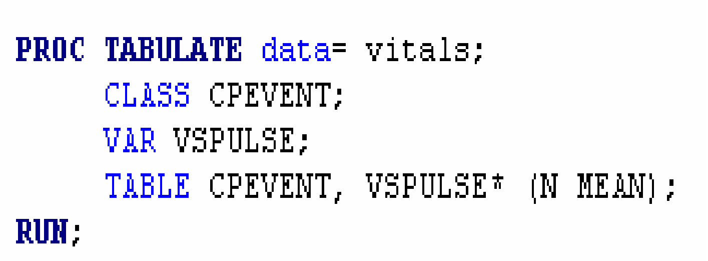
Two Dimensional Table



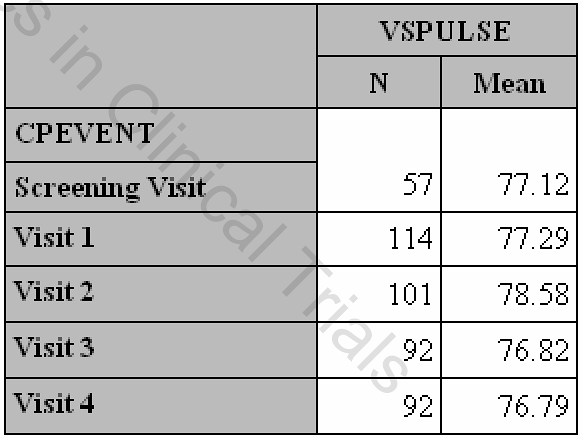
```
PROC TABULATE data= vitals;
     CLASS CPEVENT;
     VAR VSPULSE;
     TABLE VSPULSE* (N MEAN), CPEVENT;
RUN;
```



		CPEVENT				
		Screening Visit	Visit 1	Visit 2	Visit 3	Visit 4
VSPULSE	N	57	114	101	92	92
	Mean	77.12	77.29	78.58	76.82	76.79

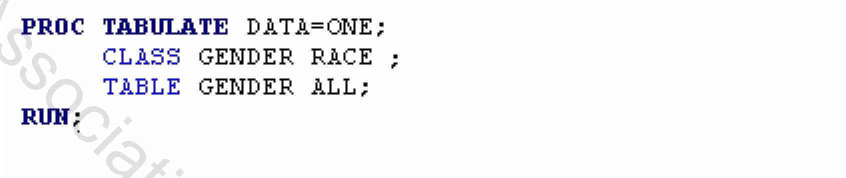


```
PROC TABULATE data= vitals;
     CLASS CPEVENT;
     VAR VSPULSE;
     TABLE CPEVENT, VSPULSE* (N MEAN);
RUN;
```

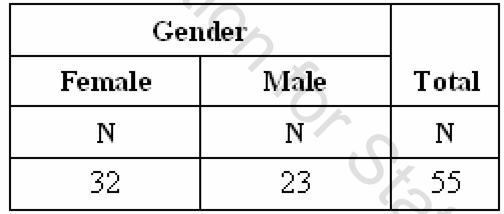


	VSPULSE	
	N	Mean
CPEVENT		
Screening Visit	57	77.12
Visit 1	114	77.29
Visit 2	101	78.58
Visit 3	92	76.82
Visit 4	92	76.79

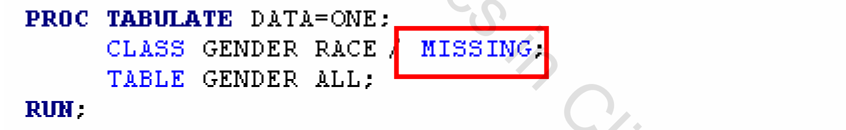
Missing Option



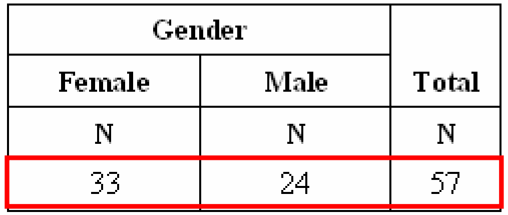
```
PROC TABULATE DATA=ONE;
     CLASS GENDER RACE ;
     TABLE GENDER ALL;
RUN;
```



Gender		Total
Female	Male	
N	N	N
32	23	55



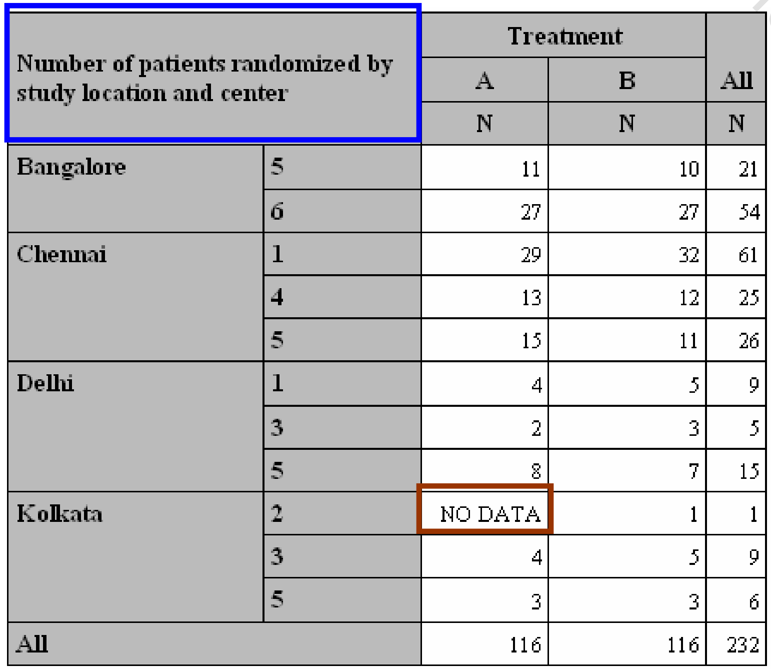
```
PROC TABULATE DATA=ONE;
     CLASS GENDER RACE / MISSING;
     TABLE GENDER ALL;
RUN;
```



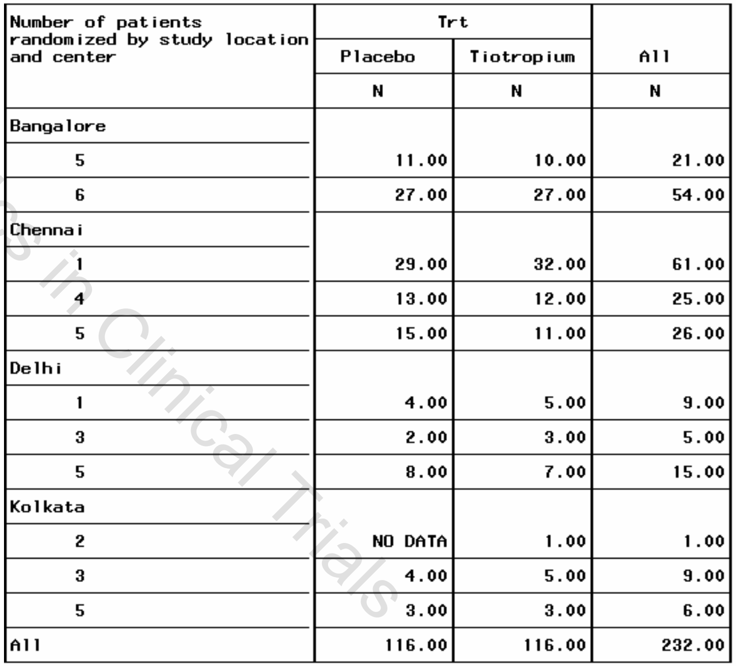
Gender		Total
Female	Male	
N	N	N
33	24	57

Table Options

```
PROC TABULATE DATA=ONE;
     CLASS CENTER LOCATION Treatment;
     TABLE LOCATION * CENTER ALL, Treatment ALL / BOX= "Number of patients randomized
     by study location and center" INDENT=6 CONDENSE RTS=30 MISSTEXT="NO DATA";
RUN;
```



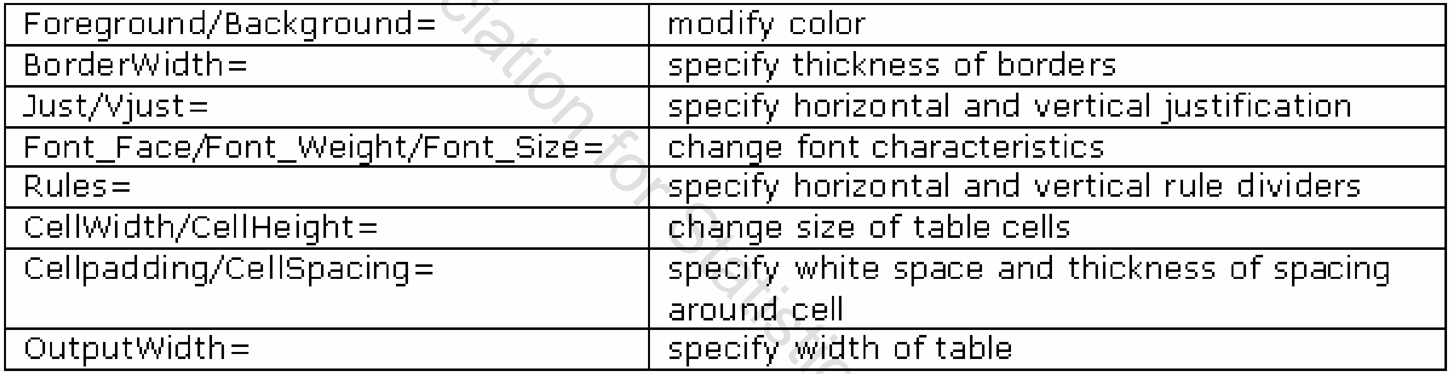
Number of patients randomized by study location and center		Treatment		All
		A	B	
		N	N	N
Bangalore	5	11	10	21
	6	27	27	54
Chennai	1	29	32	61
	4	13	12	25
	5	15	11	26
Delhi	1	4	5	9
	3	2	3	5
	5	8	7	15
Kolkata	2	NO DATA	1	1
	3	4	5	9
	5	3	3	6
All		116	116	232



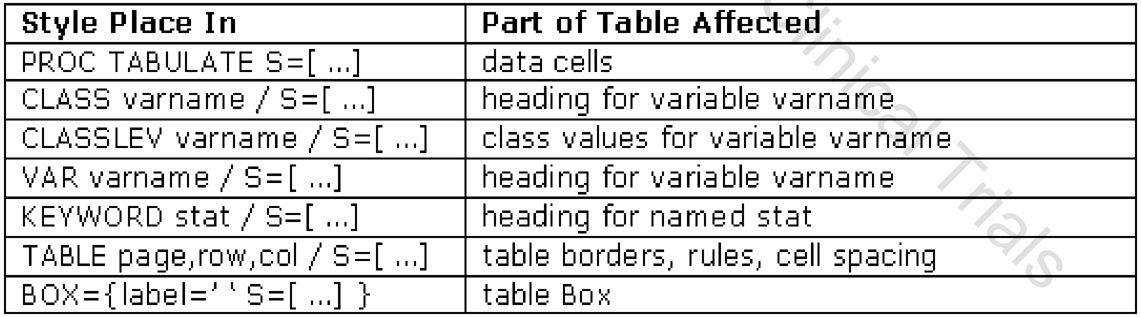
```
Number of patients           |        Trt        |
randomized by study location |  Placebo Tiotropium|     All
and center                   |      N        N    |       N
Bangalore
      5                      |  11.00     10.00   |   21.00
      6                      |  27.00     27.00   |   54.00
Chennai
      1                      |  29.00     32.00   |   61.00
      4                      |  13.00     12.00   |   25.00
      5                      |  15.00     11.00   |   26.00
Delhi
      1                      |   4.00      5.00   |    9.00
      3                      |   2.00      3.00   |    5.00
      5                      |   8.00      7.00   |   15.00
Kolkata
      2                      | NO DATA     1.00   |    1.00
      3                      |   4.00      5.00   |    9.00
      5                      |   3.00      3.00   |    6.00
All                          | 116.00    116.00   |  232.00
```

ODS Options

ODS Style elements to clean up the table:

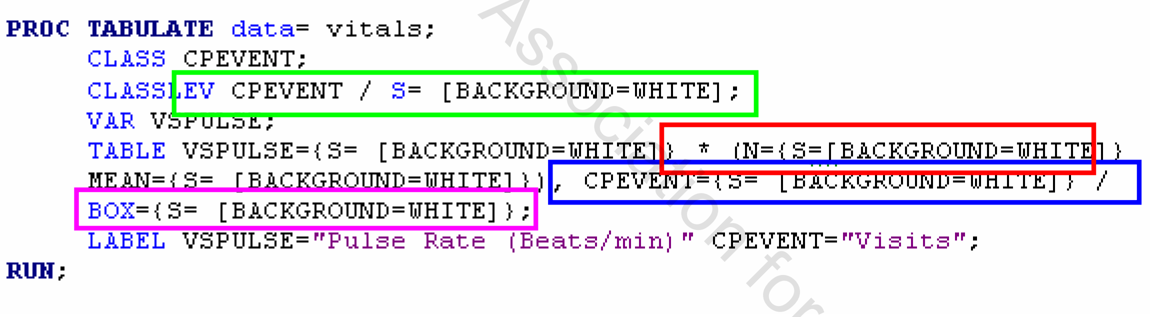


Foreground/Background=	modify color
BorderWidth=	specify thickness of borders
Just/Vjust=	specify horizontal and vertical justification
Font_Face/Font_Weight/Font_Size=	change font characteristics
Rules=	specify horizontal and vertical rule dividers
CellWidth/CellHeight=	change size of table cells
Cellpadding/CellSpacing=	specify white space and thickness of spacing around cell
OutputWidth=	specify width of table

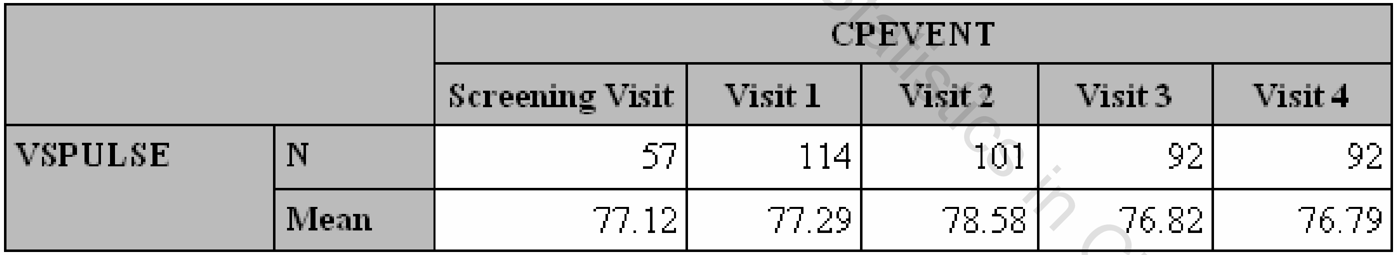


Style Place In	Part of Table Affected
PROC TABULATE S=[...]	data cells
CLASS varname / S=[...]	heading for variable varname
CLASSLEV varname / S=[...]	class values for variable varname
VAR varname / S=[...]	heading for variable varname
KEYWORD stat / S=[...]	heading for named stat
TABLE page,row,col / S=[...]	table borders, rules, cell spacing
BOX={label=‘ ‘ S=[...] }	table Box

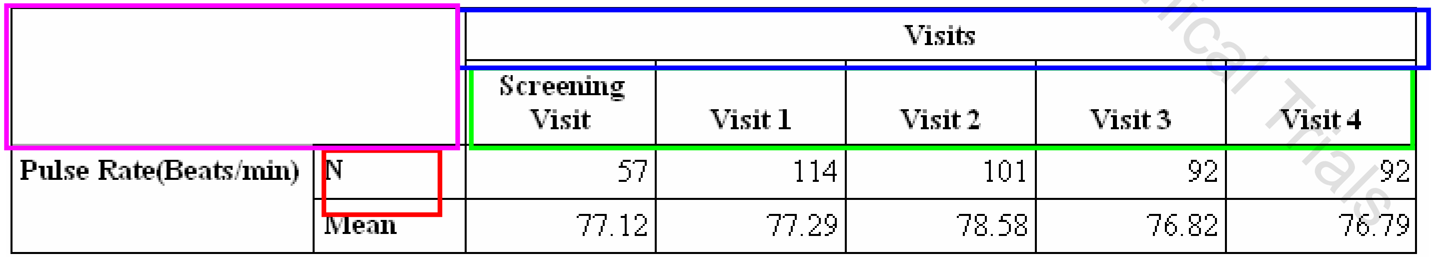
Style Options



```
PROC TABULATE data= vitals;
     CLASS CPEVENT;
     CLASSLEV CPEVENT / S= [BACKGROUND=WHITE];
     VAR VSPULSE;
     TABLE VSPULSE={S= [BACKGROUND=WHITE]} * (N={S=[BACKGROUND=WHITE]}
     MEAN={S= [BACKGROUND=WHITE]}), CPEVENT={S= [BACKGROUND=WHITE]} /
     BOX={S= [BACKGROUND=WHITE]};
     LABEL VSPULSE="Pulse Rate (Beats/min)" CPEVENT="Visits";
RUN;
```

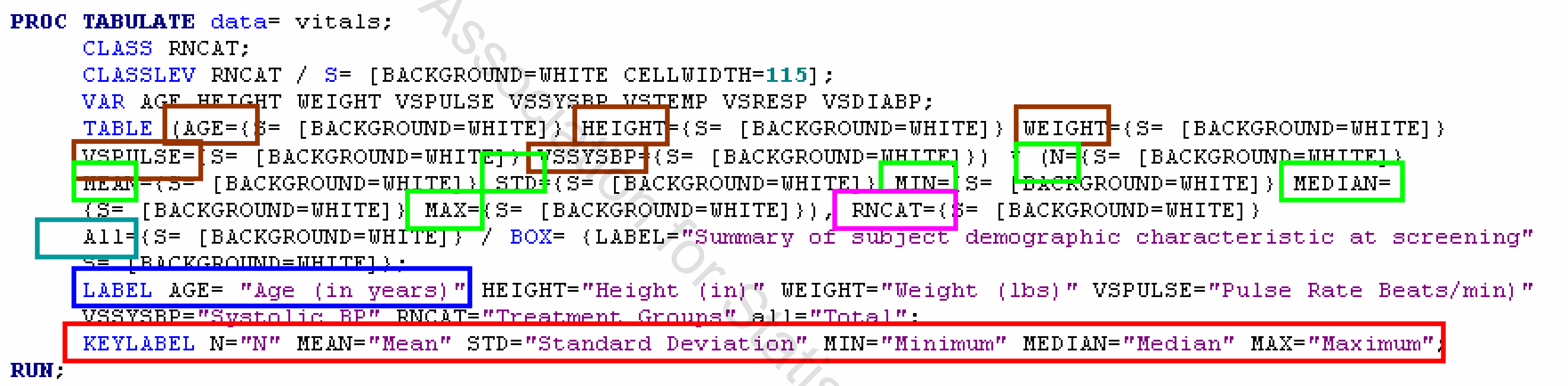


		CPEVENT				
		Screening Visit	Visit 1	Visit 2	Visit 3	Visit 4
VSPULSE	N	57	114	101	92	92
	Mean	77.12	77.29	78.58	76.82	76.79



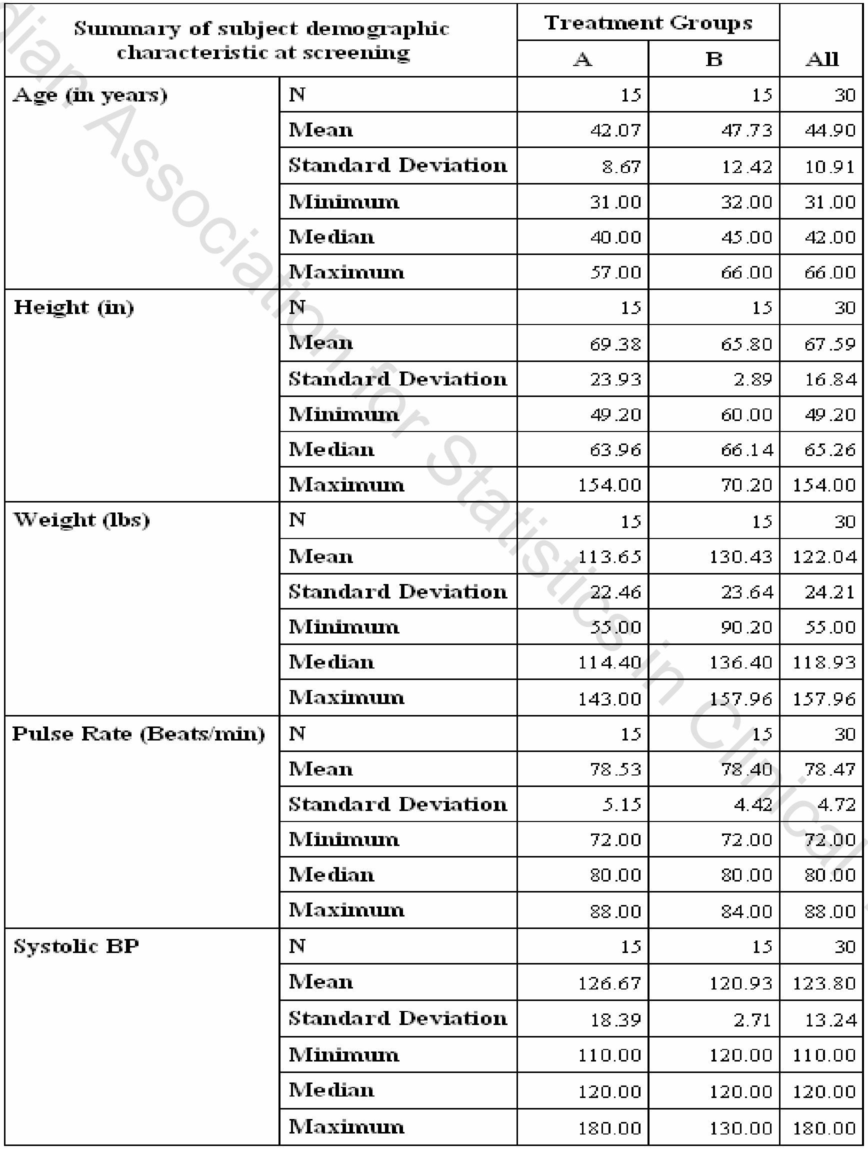
		Visits				
		Screening Visit	Visit 1	Visit 2	Visit 3	Visit 4
Pulse Rate(Beats/min)	N	57	114	101	92	92
	Mean	77.12	77.29	78.58	76.82	76.79

Example illustrating demography table



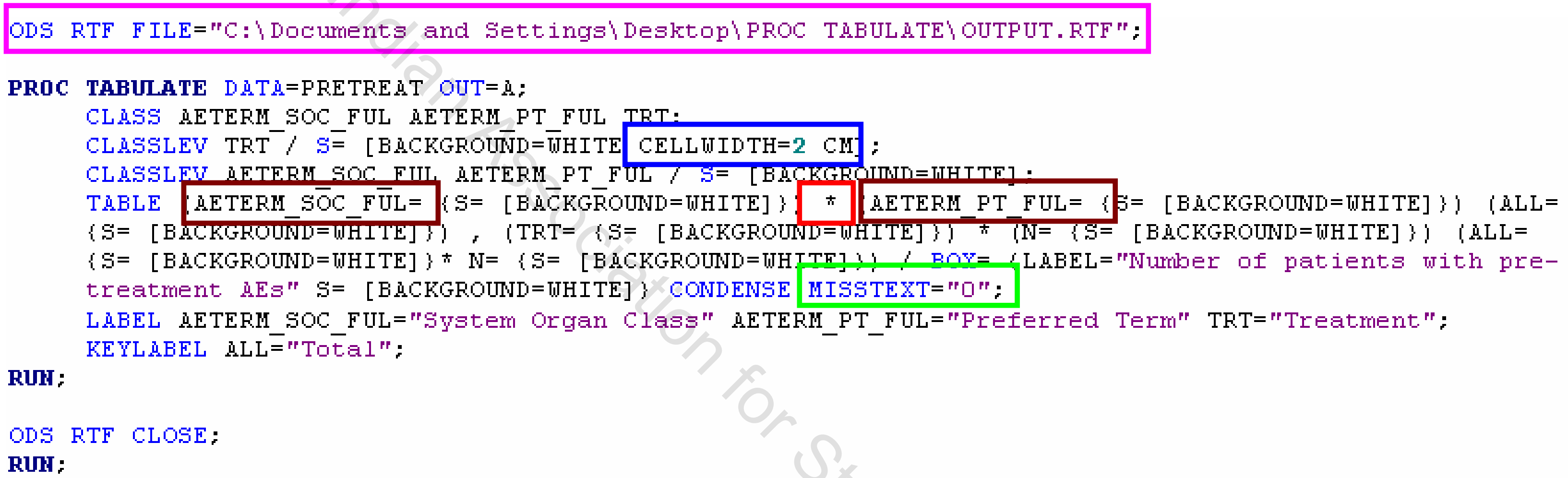
```
PROC TABULATE data= vitals;
      CLASS RNCAT;
      CLASSLEV RNCAT / S= [BACKGROUND=WHITE CELLWIDTH=115];
      VAR AGE HEIGHT WEIGHT VSPULSE VSSYSBP VSTEMP VSRESP VSDIABP;
      TABLE (AGE={S= [BACKGROUND=WHITE]} HEIGHT={S= [BACKGROUND=WHITE]} WEIGHT={S= [BACKGROUND=WHITE]}
      VSPULSE={S= [BACKGROUND=WHITE]} VSSYSBP={S= [BACKGROUND=WHITE]}) * (N={S= [BACKGROUND=WHITE]}
      MEAN={S= [BACKGROUND=WHITE]} STD={S= [BACKGROUND=WHITE]} MIN={S= [BACKGROUND=WHITE]} MEDIAN=
      {S= [BACKGROUND=WHITE]} MAX={S= [BACKGROUND=WHITE]}), RNCAT={S= [BACKGROUND=WHITE]}
      All={S= [BACKGROUND=WHITE]} / BOX= {LABEL="Summary of subject demographic characteristic at screening"
      S= [BACKGROUND=WHITE]};
      LABEL AGE= "Age (in years)" HEIGHT="Height (in)" WEIGHT="Weight (lbs)" VSPULSE="Pulse Rate Beats/min)"
      VSSYSBP="Systolic BP" RNCAT="Treatment Groups" all="Total";
      KEYLABEL N="N" MEAN="Mean" STD="Standard Deviation" MIN="Minimum" MEDIAN="Median" MAX="Maximum";
RUN;
```

Output

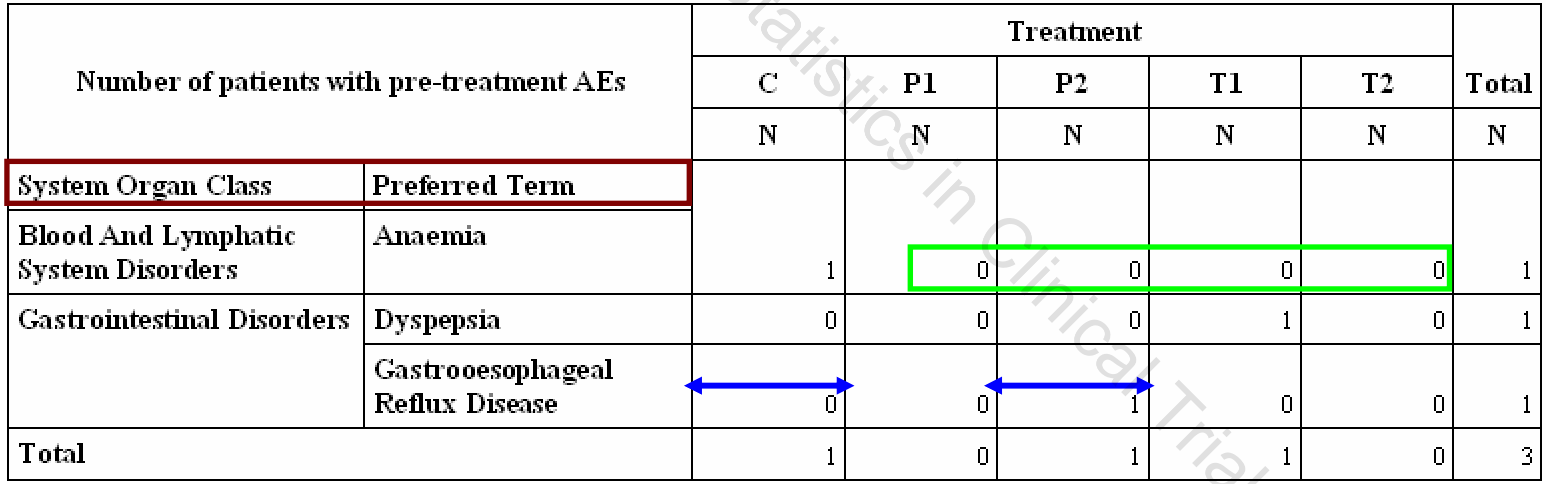


Summary of subject demographic characteristic at screening		Treatment Groups		All
		A	B	
Age (in years)	N	15	15	30
	Mean	42.07	47.73	44.90
	Standard Deviation	8.67	12.42	10.91
	Minimum	31.00	32.00	31.00
	Median	40.00	45.00	42.00
	Maximum	57.00	66.00	66.00
Height (in)	N	15	15	30
	Mean	69.38	65.80	67.59
	Standard Deviation	23.93	2.89	16.84
	Minimum	49.20	60.00	49.20
	Median	63.96	66.14	65.26
	Maximum	154.00	70.20	154.00
Weight (lbs)	N	15	15	30
	Mean	113.65	130.43	122.04
	Standard Deviation	22.46	23.64	24.21
	Minimum	55.00	90.20	55.00
	Median	114.40	136.40	118.93
	Maximum	143.00	157.96	157.96
Pulse Rate (Beats/min)	N	15	15	30
	Mean	78.53	78.40	78.47
	Standard Deviation	5.15	4.42	4.72
	Minimum	72.00	72.00	72.00
	Median	80.00	80.00	80.00
	Maximum	88.00	84.00	88.00
Systolic BP	N	15	15	30
	Mean	126.67	120.93	123.80
	Standard Deviation	18.39	2.71	13.24
	Minimum	110.00	120.00	110.00
	Median	120.00	120.00	120.00
	Maximum	180.00	130.00	180.00

Example illustrating AE table

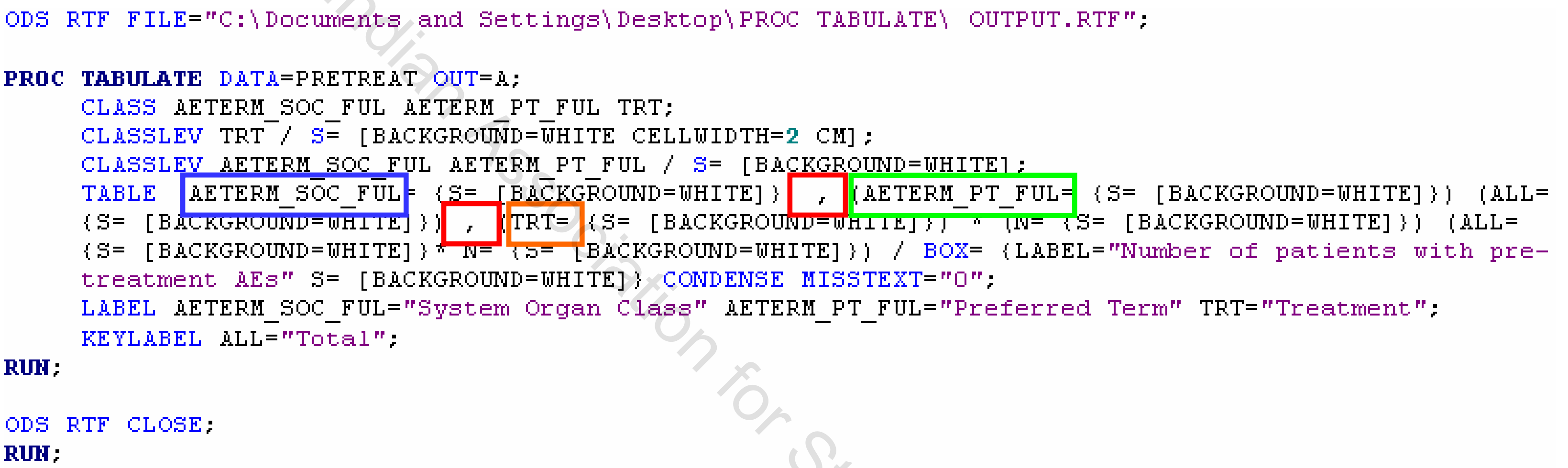


```
ODS RTF FILE="C:\Documents and Settings\Desktop\PROC TABULATE\OUTPUT.RTF";

PROC TABULATE DATA=PRETREAT OUT=A;
     CLASS AETERM_SOC_FUL AETERM_PT_FUL TRT;
     CLASSLEV TRT / S= [BACKGROUND=WHITE CELLWIDTH=2 CM];
     CLASSLEV AETERM_SOC_FUL AETERM_PT_FUL / S= [BACKGROUND=WHITE];
     TABLE (AETERM_SOC_FUL= {S= [BACKGROUND=WHITE]} * (AETERM_PT_FUL= {S= [BACKGROUND=WHITE]}) (ALL=
     {S= [BACKGROUND=WHITE]})) , (TRT= {S= [BACKGROUND=WHITE]}) * (N= {S= [BACKGROUND=WHITE]}) (ALL=
     {S= [BACKGROUND=WHITE]}* N= {S= [BACKGROUND=WHITE]}) / BOX= {LABEL="Number of patients with pre-
     treatment AEs" S= [BACKGROUND=WHITE]} CONDENSE MISSTEXT="0";
     LABEL AETERM_SOC_FUL="System Organ Class" AETERM_PT_FUL="Preferred Term" TRT="Treatment";
     KEYLABEL ALL="Total";
RUN;

ODS RTF CLOSE;
RUN;
```

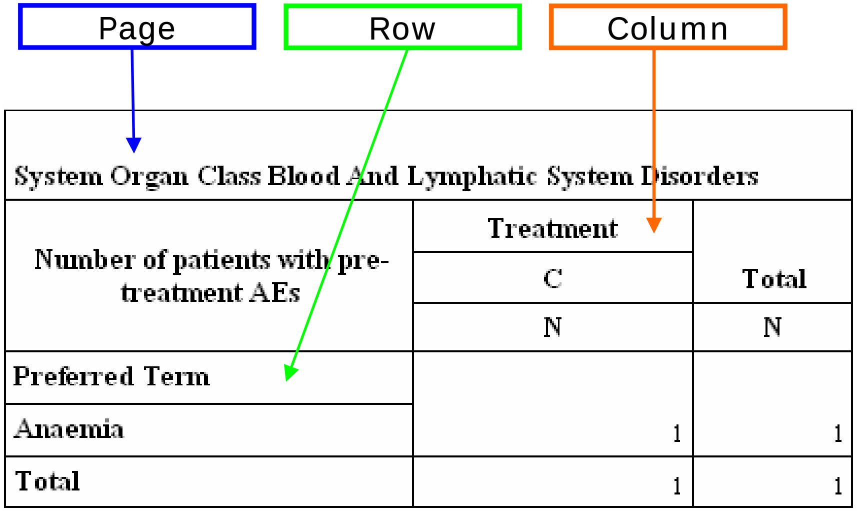


Number of patients with pre-treatment AEs		Treatment					Total
		C	P1	P2	T1	T2	
		N	N	N	N	N	N
System Organ Class	Preferred Term						
Blood And Lymphatic System Disorders	Anaemia	1	0	0	0	0	1
Gastrointestinal Disorders	Dyspepsia	0	0	0	1	0	1
	Gastrooesophageal Reflux Disease	0	0	1	0	0	1
Total		1	0	1	1	0	3

Three Dimensional Table

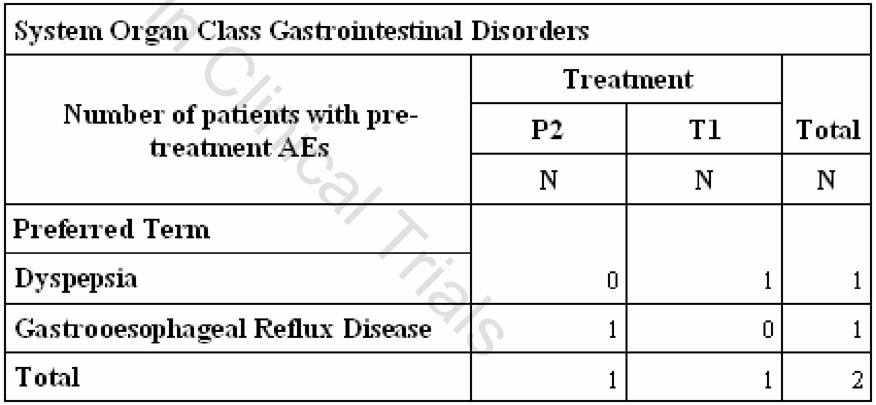


```
ODS RTF FILE="C:\Documents and Settings\Desktop\PROC TABULATE\ OUTPUT.RTF";

PROC TABULATE DATA=PRETREAT OUT=A;
     CLASS AETERM_SOC_FUL AETERM_PT_FUL TRT;
     CLASSLEV TRT / S= [BACKGROUND=WHITE CELLWIDTH=2 CM];
     CLASSLEV AETERM_SOC_FUL AETERM_PT_FUL / S= [BACKGROUND=WHITE];
     TABLE AETERM_SOC_FUL= {S= [BACKGROUND=WHITE]} , (AETERM_PT_FUL= {S= [BACKGROUND=WHITE]}) (ALL=
     {S= [BACKGROUND=WHITE]}) , (TRT= {S= [BACKGROUND=WHITE]}) * (N= {S= [BACKGROUND=WHITE]}) (ALL=
     {S= [BACKGROUND=WHITE]} * N= {S= [BACKGROUND=WHITE]}) / BOX= {LABEL="Number of patients with pre-
     treatment AEs" S= [BACKGROUND=WHITE]} CONDENSE MISSTEXT="0";
     LABEL AETERM_SOC_FUL="System Organ Class" AETERM_PT_FUL="Preferred Term" TRT="Treatment";
     KEYLABEL ALL="Total";
RUN;

ODS RTF CLOSE;
RUN;
```



System Organ Class Blood And Lymphatic System Disorders

Number of patients with pre-treatment AEs	Treatment	Total
	C	
	N	N
Preferred Term		
Anaemia	1	1
Total	1	1



System Organ Class Gastrointestinal Disorders

Number of patients with pre-treatment AEs	Treatment		Total
	P2	T1	
	N	N	N
Preferred Term			
Dyspepsia	0	1	1
Gastrooesophageal Reflux Disease	1	0	1
Total	1	1	2

Summary

- It computes many statistics and displays them in a table format.
- It will produce tables in up to three dimensions.
- Nice mechanism that can be used to label and format the variables and the statistics produced.
- Can be an efficient report writer, capable of producing a variety of displays.

References

- SAS online documentation
- SUGI papers on PROC TABULATE (www.lexjansen.com)

Thank You !

The Power of SAS Arrays in Clinical Trials

Jino Joseph
BDSI, Bangalore

Flow of talk

- Why do we need arrays?
- Basic Array concepts
  - Definition
  - Elements
  - Syntax
  - Rules
- Application in Clinical Trial Data Analysis
  - To search a specified Value
  - Count Consecutive days
  - Data LOCF
  - Data Merge
  - Convert missing to ‘0’
  - Data Concatenation

Why do we need arrays?

- A set of variables (of the same data type) grouped together for the duration of a data step by being given a name in an ARRAY statement
- Repetitious statements and redundant calculation codes reduced to few lines.
- A powerful tool to perform conditional and Iterative processing.
- Each variable can be identified by referring to the array by means of an index

Basic Array Concepts

- Two steps in the use of array are commonly involved:
  1. Array definition
  2. DO loop under optional IF-THEN-ELSE conditions
     - For example: if the temperature in ^{0}F at 5 different locations need to be converted to unit of ^{0}C, the following array codes may be used

```
array trs [5] t1 t2 t3 t4 t5;
      do  i  =1 to 5;
            If trs [i] ^= . Then trs[i] = (trs[i] -32)*5/9;
      End;
```

Contd..

- Types of Arrays
  - STATIC
    - Predefined Size
    - Simplest type of arrays
  - DYNAMIC
    - No fixed Size
    - Grow of shrink with different data automatically
    - * is used to represent the array size
    - The function DIM(array name) returns the number of elements in the array.

```
array trs [*] t: ;
do  i  =1 to DIM(trs);
        If trs[i]  ^= . Then trs[i] = (trs[i] -32)*5/9;
end;
```

Rules

- Scope confined to a data step
- Either character or numeric
  - _ALL_
  - _CHARACTER_
  - _NUMERIC_

Clinical Trial Applications

1. Search specified value

   – Very Efficient in finding a specified Target value.

     Eg: To find the day on which the maximum target value is reached

OBS	REGIMEN	SUBJECT	DAY1	DAY2	DAY3	DAY4
1	A	101	0.0	0.0	0.0	0.0
2	A	102	.	0.5	0.5	0.0
3	B	106	0.5	0.0	0.0	0.0
4	B	107	0.5	2.0	0.5	2.0
5	C	111	1.0	3.0	2.0	2.5
6	C	112	2.0	3.0	2.5	3.5

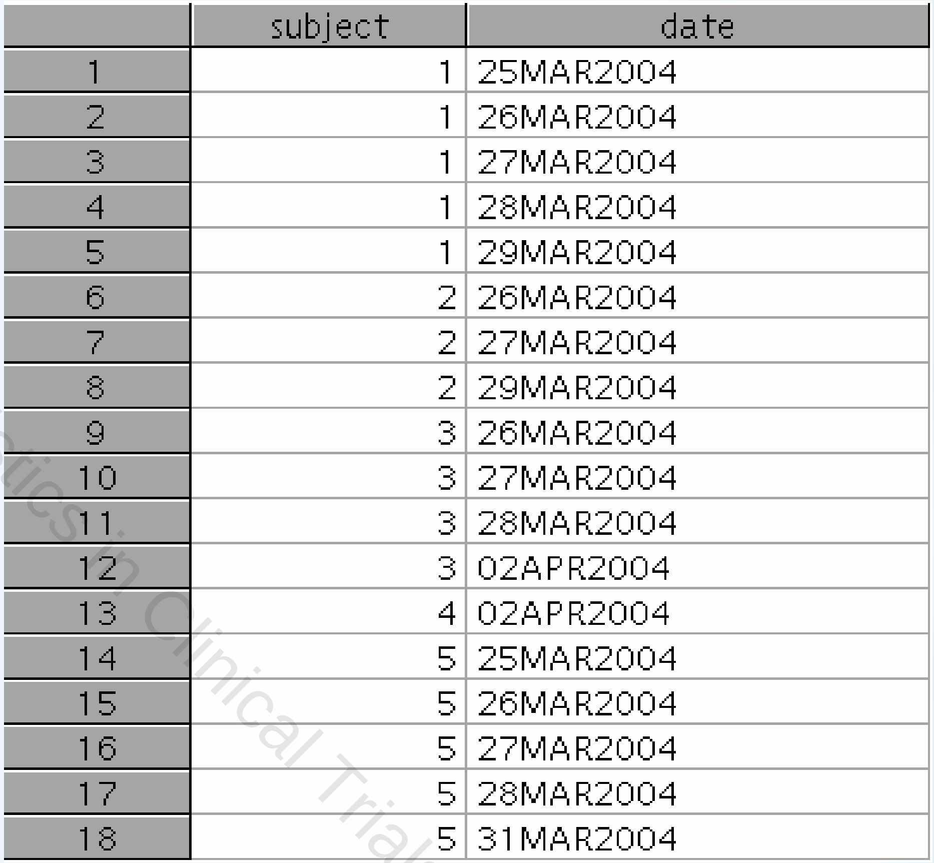
Code:

```
data eff2;
     set eff1;
     array days[4] day1 – day4;
     maxscore = max (of days [*]);
           do i = 1 to dim (days);
              if maxscore >0 and days[i] = maxscore then do;
                      tmax = i;
                      return;
              end;
           end;
     drop i maxscore;
run;
```

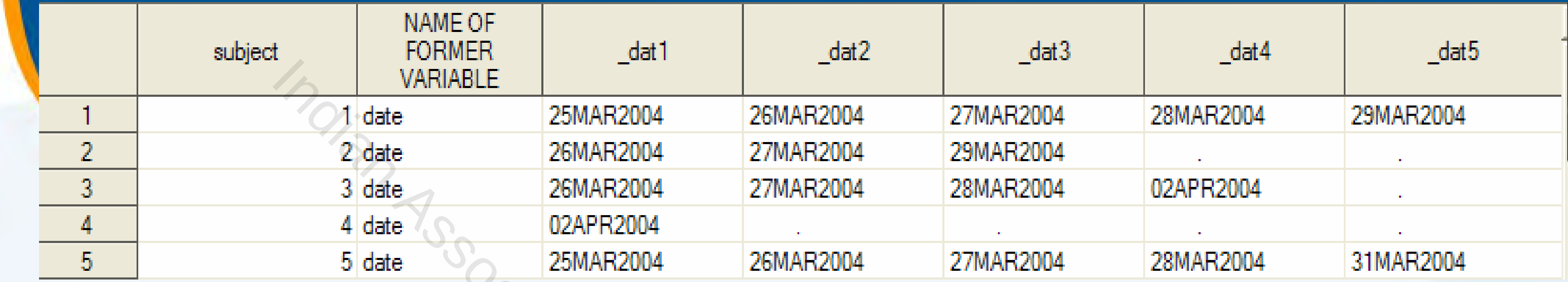
```
OBS  REGIMEN   SUBJECT    DAY1   DAY2   DAY3    DAY4   TMAX
1        A        101     0.0    0.0    0.0     0.0     .
2        A        102      .     0.5    0.5     0.0     2
3        B        106     0.5    0.0    0.0     0.0     1
4        B        107     0.5    2.0    05      2.0     2
5        C        111     1.0    3.0    2.0     2.5     2
6        C        112     2.0    3.0    2.5     3.5     4
```

2. Count consecutive days

- Patient diaries are used to collect important data, such as symptom scores, concomitant medication usages etc.
- Some analysis are based on diary data such as awakening-free nights, symptom free days.
- For Eg: We need to check whether a subject has not experienced night awakening for 3 consecutive days.



	subject	date
1	1	25MAR2004
2	1	26MAR2004
3	1	27MAR2004
4	1	28MAR2004
5	1	29MAR2004
6	2	26MAR2004
7	2	27MAR2004
8	2	29MAR2004
9	3	26MAR2004
10	3	27MAR2004
11	3	28MAR2004
12	3	02APR2004
13	4	02APR2004
14	5	25MAR2004
15	5	26MAR2004
16	5	27MAR2004
17	5	28MAR2004
18	5	31MAR2004



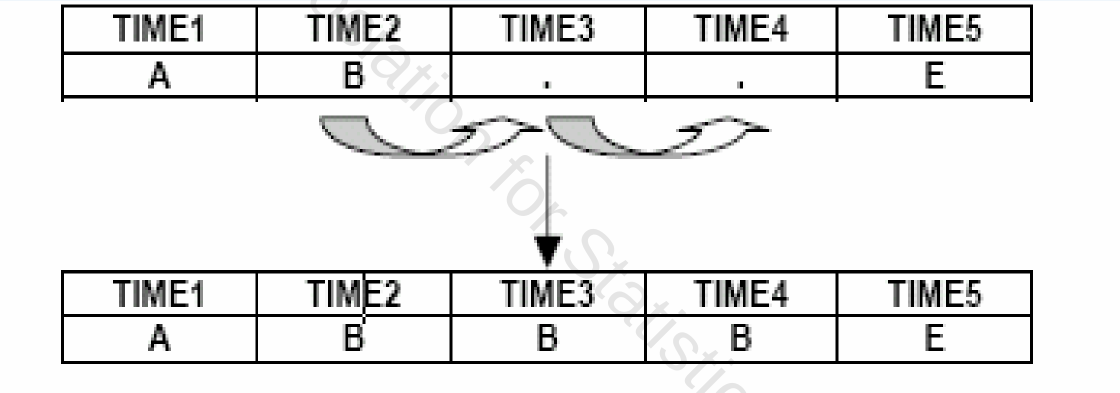
	subject	NAME OF FORMER VARIABLE	_dat1	_dat2	_dat3	_dat4	_dat5
1	1	date	25MAR2004	26MAR2004	27MAR2004	28MAR2004	29MAR2004
2	2	date	26MAR2004	27MAR2004	29MAR2004	.	.
3	3	date	26MAR2004	27MAR2004	28MAR2004	02APR2004	.
4	4	date	02APR2004	.	.	.	.
5	5	date	25MAR2004	26MAR2004	27MAR2004	28MAR2004	31MAR2004

```
proc transpose data = date prefix = _dat out = temp1;
by subject;
var date;
run;

data temp2 (keep=subject  count);
set temp1 ;
array dates {*} _dat: dummy;
retain  count 1;
   do i=1 to dim(dates)-1;
          if dates[i]^=. then do;
                    if dates[i] = dates[i+1]-1 then do; output; count=count + 1; end;
                       else do; output; count=1; end;
            end;
   end;
run;
```

Contd..

```
proc sort data = temp2;
by subject count;
run;

data temp3;
set temp2;
by subject;
if last.subject and count < = 3;
run;
```

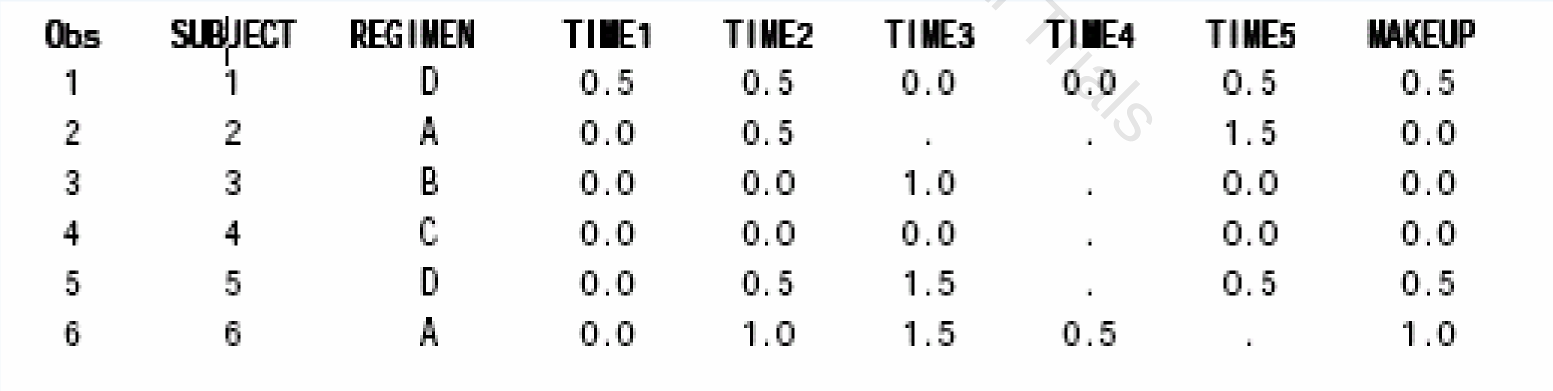
3. Data LOCF



TIME1	TIME2	TIME3	TIME4	TIME5
A	B	.	.	E

TIME1	TIME2	TIME3	TIME4	TIME5
A	B	B	B	E

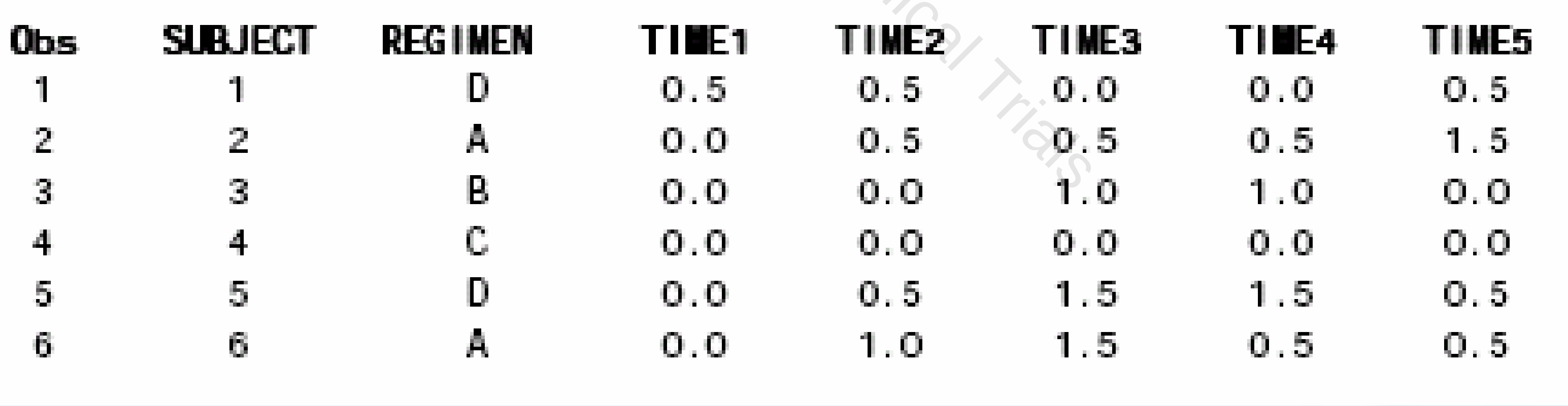
- Last non-missing value carried forward.
- The following data set called SCORE will be used as the example.



Obs	SUBJECT	REGIMEN	TIME1	TIME2	TIME3	TIME4	TIME5	MAKEUP
1	1	D	0.5	0.5	0.0	0.0	0.5	0.5
2	2	A	0.0	0.5	.	.	1.5	0.0
3	3	B	0.0	0.0	1.0	.	0.0	0.0
4	4	C	0.0	0.0	0.0	.	0.0	0.0
5	5	D	0.0	0.5	1.5	.	0.5	0.5
6	6	A	0.0	1.0	1.5	0.5	.	1.0

Code:

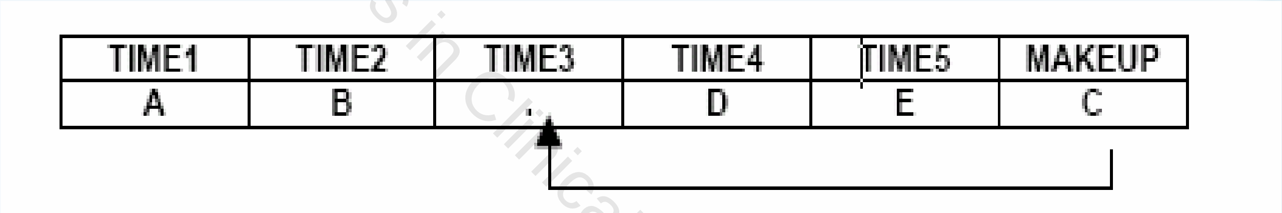
```
data locf;
    set score;
    array time [*] time: ;
            do i = 1 to dim(time);
                        if time[i] = . then time[i] = time[i -1];
              end;
    drop i makeup;
run;
```



```
Obs    SUBJECT    REGIMEN    TIME1    TIME2    TIME3    TIME4    TIME5
 1        1          D        0.5      0.5      0.0      0.0      0.5
 2        2          A        0.0      0.5      0.5      0.5      1.5
 3        3          B        0.0      0.0      1.0      1.0      0.0
 4        4          C        0.0      0.0      0.0      0.0      0.0
 5        5          D        0.0      0.5      1.5      1.5      0.5
 6        6          A        0.0      1.0      1.5      0.5      0.5
```

4. Find and Replace

- Exceptionally powerful and fast.
- Can replace the elements referred to by iterator I in the array with the new value when condition holds good, such as find and replace the missing data.
- Eg: In some cases experimental tests are not conducted continuously.



TIME1	TIME2	TIME3	TIME4	TIME5	MAKEUP
A	B	.	D	E	C

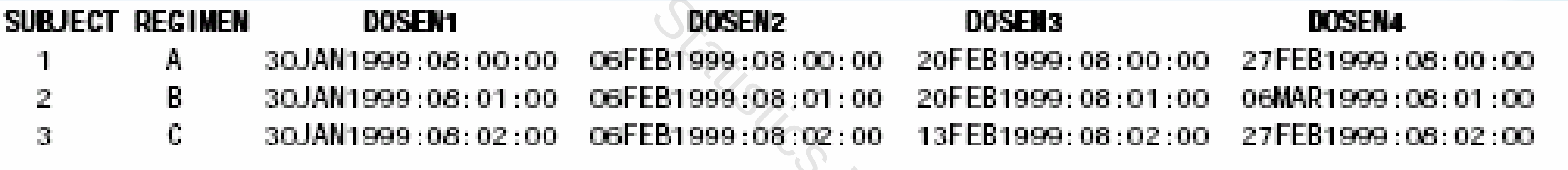
```
data replace;
set score;
array apps [5] time1- time5;
          do i=1 to dim(apps);
                    if apps[i] =. then apps[i]=makeup ;
          end;
drop i ;
run;
```

Code:

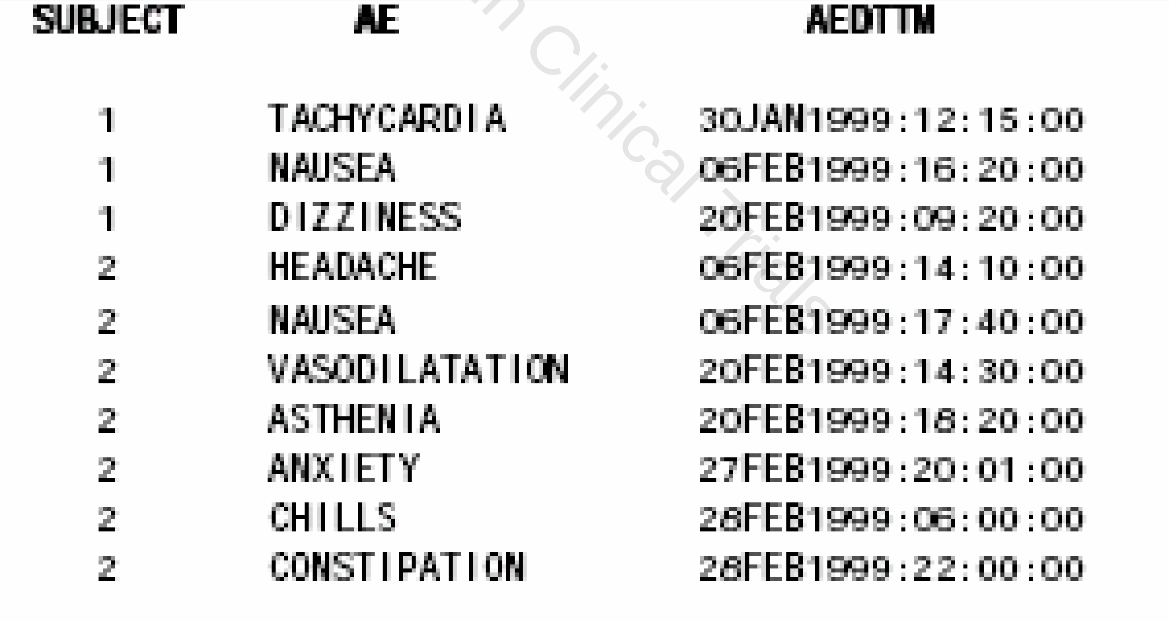
```
data replace2;
set score2;
array apps [5] time1- time5;
array makeups [2] makeup1 makeup2 ;
   j=1;
          do i=1 to dim(apps);
                    if apps[i] =. then do;
                              apps[i]=makeups[j] ;
                              j=j + 1;
                    end;
          end;
drop i j ;
run;
```

5. Data Merge

- It is often required to merge dose data with other safety data, such as AE’s , Vitals, Labs and locate the dose-related safety profiles.
  - Eg: A dose dataset and an AE dataset



SUBJECT	REGIMEN	DOSEN1	DOSEN2	DOSEN3	DOSEN4
1	A	30JAN1999:08:00:00	06FEB1999:08:00:00	20FEB1999:08:00:00	27FEB1999:08:00:00
2	B	30JAN1999:08:01:00	06FEB1999:08:01:00	20FEB1999:08:01:00	06MAR1999:08:01:00
3	C	30JAN1999:08:02:00	06FEB1999:08:02:00	13FEB1999:08:02:00	27FEB1999:08:02:00



SUBJECT	AE	AEDTTM
1	TACHYCARDIA	30JAN1999:12:15:00
1	NAUSEA	06FEB1999:16:20:00
1	DIZZINESS	20FEB1999:09:20:00
2	HEADACHE	06FEB1999:14:10:00
2	NAUSEA	06FEB1999:17:40:00
2	VASODILATATION	20FEB1999:14:30:00
2	ASTHENIA	20FEB1999:18:20:00
2	ANXIETY	27FEB1999:20:01:00
2	CHILLS	28FEB1999:06:00:00
2	CONSTIPATION	28FEB1999:22:00:00

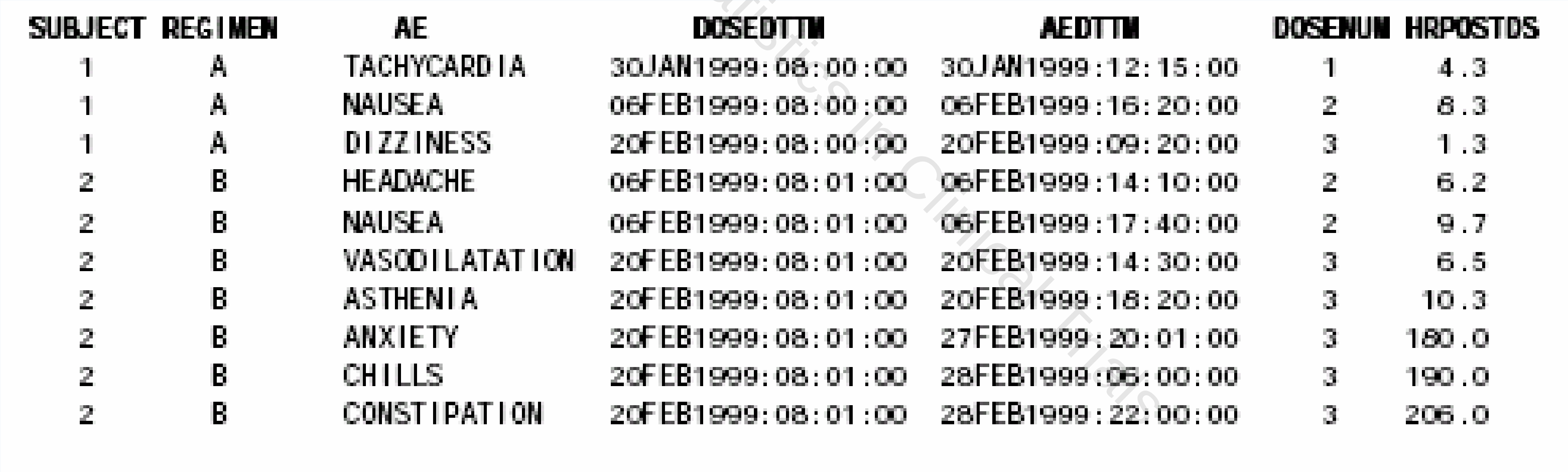
Code:

The code to merge these datasets is as following.

```
data dose_ae;
   merge dose ae;
   by subject;
   array dosen {*} dosen:;
   do I = 1 to dim(dosen);
          if dosen[i] ^= . And aedttm> dosen[i] then do;
                    dosedttm = dosen[i];
                    dosenum = I;
          end;
   end;
if dosenum^=.;
hrpostds = round(((aedttm-dosedttm)/3600),0.1);
format dosedttm datetime20.;
drop I dosen1 – dosen4;
run;
```

- Contd..

  The final result is as below, variable DOSENUM is the order of doses, and HRPOSTDS is time in hours after dosing.



```
SUBJECT  REGIMEN   AE              DOSEDTTM             AEDTTM               DOSENUM  HRPOSTDS
   1        A      TACHYCARDIA     30JAN1999:08:00:00   30JAN1999:12:15:00      1        4.3
   1        A      NAUSEA          06FEB1999:08:00:00   06FEB1999:16:20:00      2        8.3
   1        A      DIZZINESS       20FEB1999:08:00:00   20FEB1999:09:20:00      3        1.3
   2        B      HEADACHE        06FEB1999:08:01:00   06FEB1999:14:10:00      2        6.2
   2        B      NAUSEA          06FEB1999:08:01:00   06FEB1999:17:40:00      2        9.7
   2        B      VASODILATATION  20FEB1999:08:01:00   20FEB1999:14:30:00      3        6.5
   2        B      ASTHENIA        20FEB1999:08:01:00   20FEB1999:18:20:00      3       10.3
   2        B      ANXIETY         20FEB1999:08:01:00   27FEB1999:20:01:00      3      180.0
   2        B      CHILLS          20FEB1999:08:01:00   28FEB1999:06:00:00      3      190.0
   2        B      CONSTIPATION    20FEB1999:08:01:00   28FEB1999:22:00:00      3      206.0
```

6. To convert missing to ‘0’.

OBS	SUBJ	VAR1	VAR2	VAR3
1	101	24	.	7
2	102	20	8	.
3	106	.	.	6
4	107	7	9	8
5	111	.	18	.
6	112	17	.	.

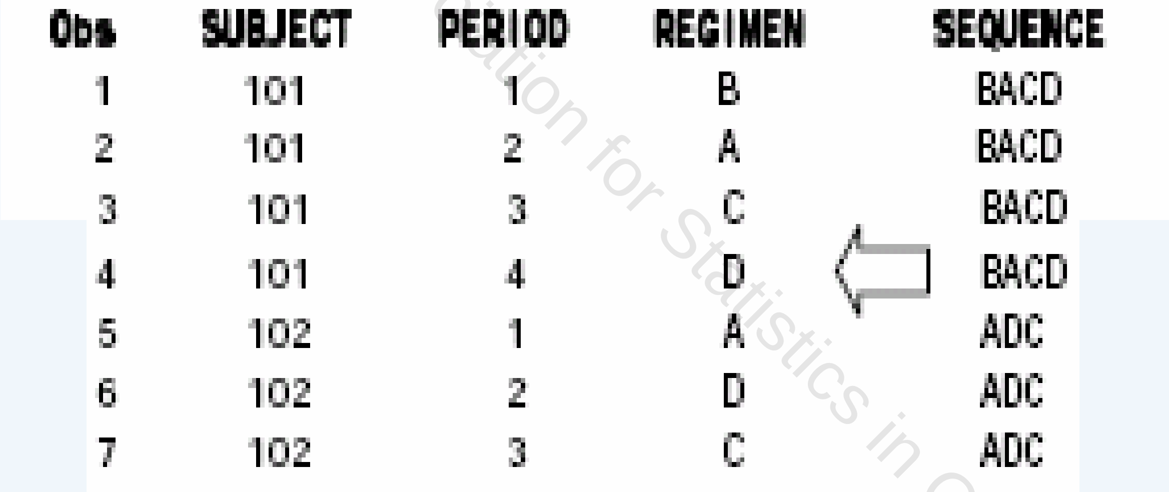
A → B

OBS	SUBJ	VAR1	VAR2	VAR3
1	101	24	0	7
2	102	20	8	0
3	106	0	0	6
4	107	7	9	8
5	111	0	18	0
6	112	17	0	0

Code:

```
data b;
        set a;
        array conv[*] var:;
                if conv[i] = . Then conv[i] = ‘0’;
run;
```

7. Data concatenation

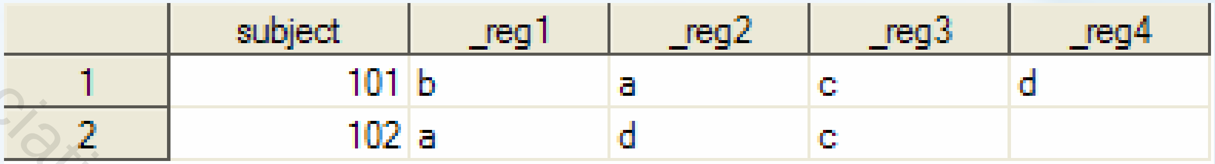


Obs	SUBJECT	PERIOD	REGIMEN	SEQUENCE
1	101	1	B	BACD
2	101	2	A	BACD
3	101	3	C	BACD
4	101	4	D	BACD
5	102	1	A	ADC
6	102	2	D	ADC
7	102	3	C	ADC

Code:

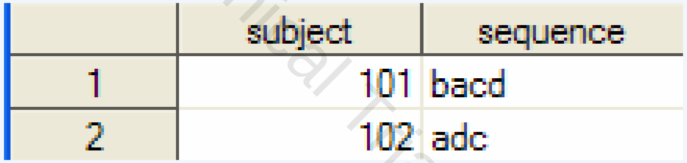
```
proc transpose data = regimen out = temp1 prefix = _reg;
     by subject;
     var regimen;
run;
```

Contd..



	subject	_reg1	_reg2	_reg3	_reg4
1	101	b	a	c	d
2	102	a	d	c	

```
data temp2;
   set temp1;
   length sequence $10;
   array regs [*] _reg:;
        do I = 1  to dim(regs);
                  sequence = compress(sequence || regs[i]);
        end;
   keep subject sequence;
run;
```



	subject	sequence
1	101	bacd
2	102	adc

```
data sequence;
   merge regimen temp2;
   by subject;
run;
```

Thank You

accenture

High performance. Delivered.

Reporting of Adverse Events

Megha Kamani
Sivaprasad Mekala

Objectives

- Overview
- Classification Dictionary
- Reporting
- Conclusion

Overview

- Adverse Event (AE) /Serious Adverse Event (SAE)
    - ➢ Significance
    - ➢ Intensity
    - ➢ Relatedness
    - ➢ Serious vs. Severe
- Detecting Adverse Events.
- Adverse Event Collection and Reporting.

Detecting Adverse Events

- ✓Symptoms (headache, nausea, etc…)
- ✓Physical findings (elevated BP, lump, etc…)
- ✓Abnormal lab values
- ✓Behavioral changes
- ✓Toxicity Grades

Detecting Adverse Events Contd.

- **Not limited to DRUG Side effects. Unfavorable deviation from BASELINE health, which includes:**
  - ✓ Worsening of conditions present at onset of the study
  - ✓ Patient deterioration due to primary disease
  - ✓ Intercurrent illness or event, i.e., flu, accident
  - ✓ Events related or possibly related to concomitant medications

Adverse Events Dictionary - MedDRA

- What is MedDRA?

  Medical **D**ictionary for **R**egulatory **A**ctivities is an electronic dictionary coding system, organized in a hierarchical structure from which terms are generated for use in classifying, analyzing and reporting adverse events

- How is MedDRA maintained?
- Benefits of MedDRA.
- Scope of MedDRA.
- Hierarchy.

Adverse Events Dictionary – MedDRA
Contd.

- Hierarchy
  - LLT (Low Level Term)
  - PT (Preferred Term)
  - HLT (High Level Term)
  - HLGT (High Level Group Term)
  - SOC (System Organ Class)

Reporting of Adverse Events

- CRF Page
- Data Structure
- Reporting

CRF

Reporting of Adverse Events Contd.

- Data Structure
  - ✓ AE Classifications
  - ✓ Severity / Toxicity
  - ✓ Relationship
  - ✓ Outcome
  - ✓ Visit Date
  - ✓ Start Date
  - ✓ Stop Date
  - ✓ Action Taken
  - ✓ SAE – Y/N (Form 7443)

Data Structure

Reporting of Adverse Events Contd.

- Listings & Tables
- Classification of Reports
  - ➢ Treatment Emergent
    - ✓By Severity
    - ✓By Drug Relationship
  - ➢ Serious Adverse Event
  - ➢ Withdrawal from study permanently
  - ➢ Discontinued from study temporarily

ae_smry

ae_smry_sev_rel

SAS Code

Conclusion

- Adverse Event
- Serious Adverse Event
- Classification Dictionary
- Reporting Structure

Questions?

accenture

THANK YOU!

Zero Rows: 5 Ways to Summarize Absolutely Nothing

Ramya Deepak

03 April 2009

INTRODUCTION

- SAS is wonderful at summarizing our data, including creating frequency counts and percentages
- However, sometimes, what *isn't* in the data is just as important as what *is* in the data
- Unfortunately, it is not so easy to get SAS to summarize what isn't there, e.g., how can a PROC FREQ count data points that do not exist in the data?

INTRODUCTION

- Example 1: In the pharmaceutical industry, the programmer may have to summarize all of the demographics that appear on a case report form.
  - However, when the data contains a small population or there is something obscure on the CRF which no subject in the data fulfills, the summarization of all the points on the CRF becomes difficult

- Example 2: A statistician may want to see all of the values on the CRF in a table even if no subject in the data reported that characteristic

- In these cases we are interested in the fact that no one is actually in the data---or as we call it here, a zero row

- The goal of this presentation is to present five different examples of how to get SAS to summarize those zero rows for us, that is, summarize records that aren't there

INITIAL DATA

- For our examples, we will consider an ECG dataset with some ECG interpretations missing that we will need to summarize later
- The expected results are *Normal, Abnormal – CS, Abnormal – NCS, No Result*
- We will count the number of subjects in each treatment group with the different ECG Interpretations

Subj ID	Treatment Group	ECG Interpretation
001	1	Abnormal - NCS
002	1	Normal
003	1	Abnormal - NCS
004	1	Abnormal - NCS
001	2	Normal
008	2	No Result

- It is apparent that the combination of any *Treatment Group* and *ECG Interpretation “Abnormal – CS”* is missing
- As well as the combination of *Treatment Group 1* and *“No Result”* and even the combination of *Treatment Group 2* and *“Abnormal – NCS”*

TECHNIQUES TO SUMMARIZE MISSING DATA

METHOD 1 – PROC FREQ USING A DUMMY HARD-CODED DATASET

- In this example, we use simple OUTPUT statement to create a blank record for each possible combination of treatment group and ECG interpretation

- Using FREQ procedure a dataset with the counts of actual data is created

Treatment Group	ECG Interpretation	Frequency Count
1	Abnormal - NCS	3
1	Normal	1
2	Normal	1
2	No Result	1

METHOD 1 – PROC FREQ USING A DUMMY HARD-CODED DATASET

- A dummy dataset having every possible combination of treatment group and ECG interpretation is created using the OUTPUT statement

```
data egtemp;
   do trtgrp = 1, 2;
     do egintp = ‘Normal’, ‘Abnormal – NCS’,
                 ‘Abnormal – CS’, ‘No Result’;
        output;
     end;
   end;
run;
```

Treatment Group	ECG Interpretation
1	Abnormal - CS
1	Abnormal - NCS
1	Normal
1	No Result
2	Abnormal - CS
2	Abnormal - NCS
2	Normal
2	No Result

METHOD 1 – PROC FREQ USING A DUMMY HARD-CODED DATASET

- On merging the dataset of frequency counts with the dummy dataset we get a complete set of frequencies for every possible combination of Treatment Group and ECG Interpretations

Treatment Group	ECG Interpretation	Frequency Count
1	Abnormal - CS	0
1	Abnormal - NCS	3
1	Normal	1
1	No Result	0
2	Abnormal - CS	0
2	Abnormal - NCS	0
2	Normal	1
2	No Result	1

METHOD 1 – PROC FREQ USING A DUMMY HARD-CODED DATASET

- The biggest advantage to this method is that it is simple and requires no formats
- The disadvantage, however, is that the programmer needs to be aware of all of the possible combinations before programming
- It could become a maintenance nightmare if the possible values change

METHOD 2 – PROC FREQ USING THE SPARSE OPTION

- In this example we use the FREQ procedure with the SPARSE option to create a dataset that includes the frequencies of the various combinations of Treatment Group and ECG Interpretation

- Using the sparse option in PROC FREQ, SAS outputs a record for every possible combination that could potentially occur in the data rather than just the combinations that do occur

```
proc freq data=ecg noprint;
     table trtgrp * egintp /out=egfrq(drop=percent) sparse;
run;
```

METHOD 2 – PROC FREQ USING THE SPARSE OPTION

- There is no record of *Treatment Group 1* and *“No Result”* and *Treatment Group 2* and *“Abnormal – NCS”* in the data, but SAS lists it as a possible combination because “*No Result”* and *“Abnormal – NCS”* occur in other data points

Treatment Group	ECG Interpretation	Frequency Count
1	Abnormal - NCS	3
1	Normal	1
1	No Result	0
2	Abnormal - NCS	0
2	Normal	1
2	No Result	1

- The sparse option is convenient to use and allows for simpler code
- A glaring limitation is that the sparse option will only summarize what it sees in the data. So although we know “*Abnormal – CS”* option from the CRF, SAS does not know this and it is left off of the frequency counts

METHOD 3 – PROC FREQ USING AN AUTOMATED DUMMY DATASET

- In this example we use the SQL procedure to automatically create a dummy dataset based on the values of formats specified by the programmer

- Using FREQ procedure a dataset with the counts of actual data is created

Treatment Group	ECG Interpretation	Frequency Count
1	Abnormal - NCS	3
1	Normal	1
2	Normal	1
2	No Result	1

METHOD 3 – PROC FREQ USING AN AUTOMATED DUMMY DATASET

- Then, using a combination of PROC SQL and the “coalesce” function, SAS joins the dataset with the counts (created from the PROC FREQ) with the dummy dataset and fills in a count of zero where an actual count from the data does not exist.

```
proc sql;
   create table egtemp as
     select a.start label=“Treatment Group" format=$trt. as trtgrp,
            b.start label="ECG Interpretation" format=$egintp. as
   egintp
     from formats(where=(fmtname=‘TREATMENT GROUP')) as a,
           formats(where=(fmtname='ECG INTERPRETATION')) as b;
   create table egfrq as
     select a.trtgrp,
            a.egintp,
            coalesce(b.count, 0) as count
     from egtemp as a left join egfrq as b
     on a.trtgrp = b.trtgrp and a.egintp = b.egintp;
quit;
```

METHOD 3 – PROC FREQ USING AN AUTOMATED DUMMY DATASET

Treatment Group	ECG Interpretation	Frequency Count
1	Abnormal - CS	0
1	Abnormal - NCS	3
1	Normal	1
1	No Result	0
2	Abnormal - CS	0
2	Abnormal - NCS	0
2	Normal	1
2	No Result	1

- This method is great because it is automatic and based on the formats
- The disadvantage is the code is rather complicated

METHOD 4 – PROC MEANS USING “COMPLETETYPES” OPTION

- This example uses a method that is very similar to using the sparse option in PROC FREQ but instead this time with PROC MEANS

- Using PROC MEANS on the initial data and the COMPLETETYPES option, we get an output dataset that includes all possible combinations that could potentially occur in the data in addition to combinations that do occur

```
proc means data=egtemp completetypes noprint nway;
   class trtgrp egintp;
   output out=egfrq(rename=(_freq_=count) drop=_type_);
run;
```

METHOD 4 – PROC MEANS USING “COMPLETETYPES” OPTION

- So, even though there is no record with *Treatment Group 1* and *No Result* and *Treatment Group 2* and *Abnormal - NCS* in the data, the completetypes option includes this combination because *No Result* and *Abnormal - NCS* occur in other data points

Treatment Group	ECG Interpretation	Frequency Count
1	Abnormal - NCS	3
1	Normal	1
1	No Result	0
2	Abnormal - NCS	0
2	Normal	1
2	No Result	1

- Like the sparse option in PROC FREQ, the completetypes option is very simple to use
- Similar again to sparse, there must be at least one occurrence of a value for completetypes to summarize appropriately

METHOD 5 – PROC MEANS USING “COMPLETETYPES” AND THE “PRELOADFMT” OPTION

- In this example we use PROC MEANS with COMPLETETYPES and PRELOADFMT option to create a dataset with all possible combination of Treatment Group and ECG Interpretation

- The PRELOADFMT option in PROC MEANS specifies that all formats are preloaded to the CLASS variables

- In the initial ECG dataset, the formats for both treatment group and ECG Interpretation are assigned using ATTRIB statements

METHOD 5 – PROC MEANS USING “COMPLETETYPES” AND THE “PRELOADFMT” OPTION

- The PRELOADFMT option in PROC MEANS uses these assigned formats to determine what the possible combinations of values could be

```
proc means data=egtemp completetypes noprint nway;
     class trtgrp egintp /PRELOADFMT ;
     output out=egfrq(rename=(_freq_=count) drop=_type_);
run;
```

Treatment Group	ECG Interpretation	Frequency Count
1	Abnormal - CS	0
1	Abnormal - NCS	3
1	Normal	1
1	No Result	0
2	Abnormal - CS	0
2	Abnormal - NCS	0
2	Normal	1
2	No Result	1

METHOD 5 – PROC MEANS USING “COMPLETETYPES” AND THE “PRELOADFMT” OPTION

- Advantages to this method include simplicity of use and the fact there is no requirement to have at least one occurrence of a value in the data

- A disadvantage is that this method only works when formats are used in combination with the input data

CONCLUSION

- When producing summary tables in the pharmaceutical industry, it is frequently important to summarize what is not there as well as what is there.

- In this presentation we have discussed five separate ways to accomplish this task and which method we choose depends on the complexity and characteristics of the data

- Whichever method you choose, you should now be armed with the knowledge and the ability to summarize nothing!

QUESTIONS??

Thank You

Safety data graphical display

Vinay Mahajan

Novartis Pharmaceuticals

April 2009

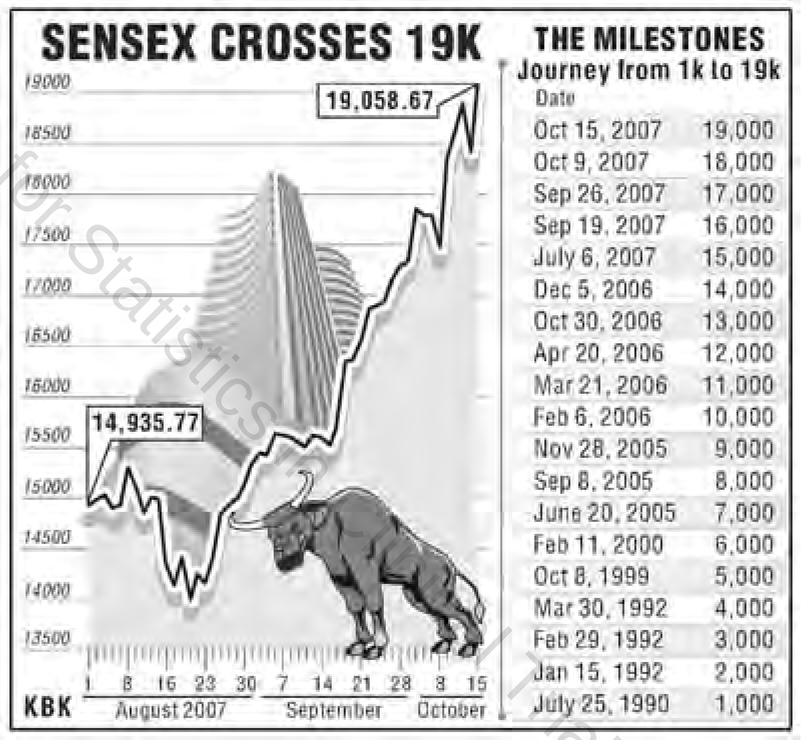
Introduction

Red

Yellow

Green

Sensex zooms and reaches astronomical levels



THE MILESTONES

Journey from 1k to 19k

Date	
Oct 15, 2007	19,000
Oct 9, 2007	18,000
Sep 26, 2007	17,000
Sep 19, 2007	16,000
July 6, 2007	15,000
Dec 5, 2006	14,000
Oct 30, 2006	13,000
Apr 20, 2006	12,000
Mar 21, 2006	11,000
Feb 6, 2006	10,000
Nov 28, 2005	9,000
Sep 8, 2005	8,000
June 20, 2005	7,000
Feb 11, 2000	6,000
Oct 8, 1999	5,000
Mar 30, 1992	4,000
Feb 29, 1992	3,000
Jan 15, 1992	2,000
July 25, 1990	1,000

Introduction

Mumbai local trains:

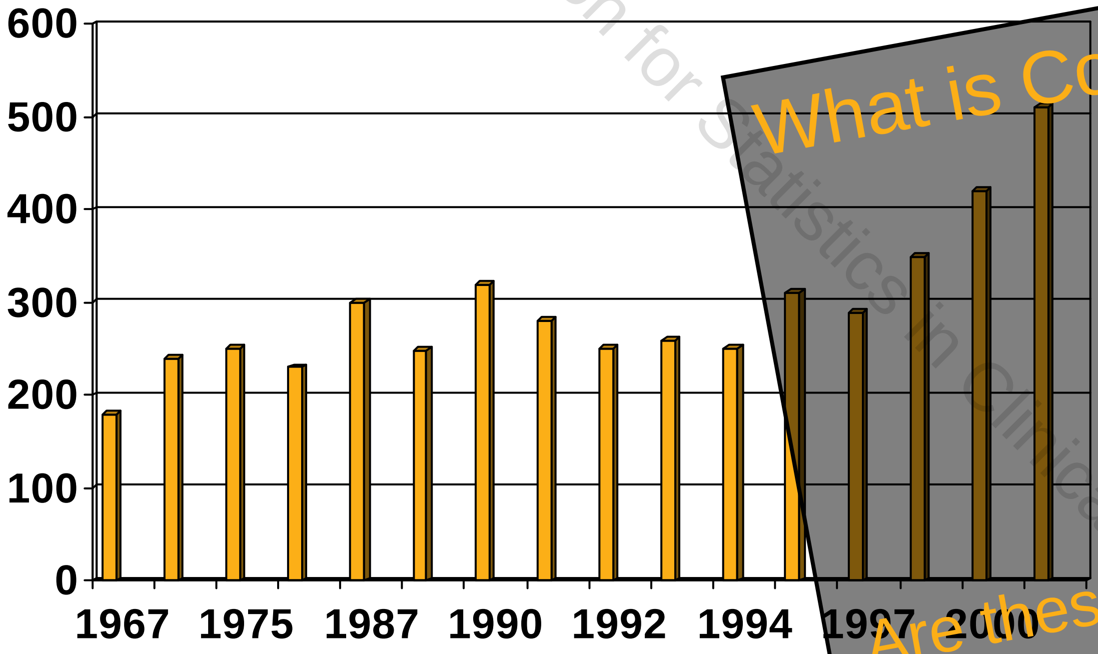
For the timid, getting into and off a Mumbai train is close to a life altering experience. The hapless commuter just flows with the tide.

What has got this to do with Safety Reporting ???

Introduction

Some data ...

1967: 180; 1973: 240; 1975: 250; 1978: 230; 1987: 300; 1989: 248; 1990: 320; 1991: 280; 1992: 250; 1993: 260; 1994: 250; 1996: 310; 1997: 290; 1999: 350; 2000: 420; 2001: 510



What is Common ?

Picture,

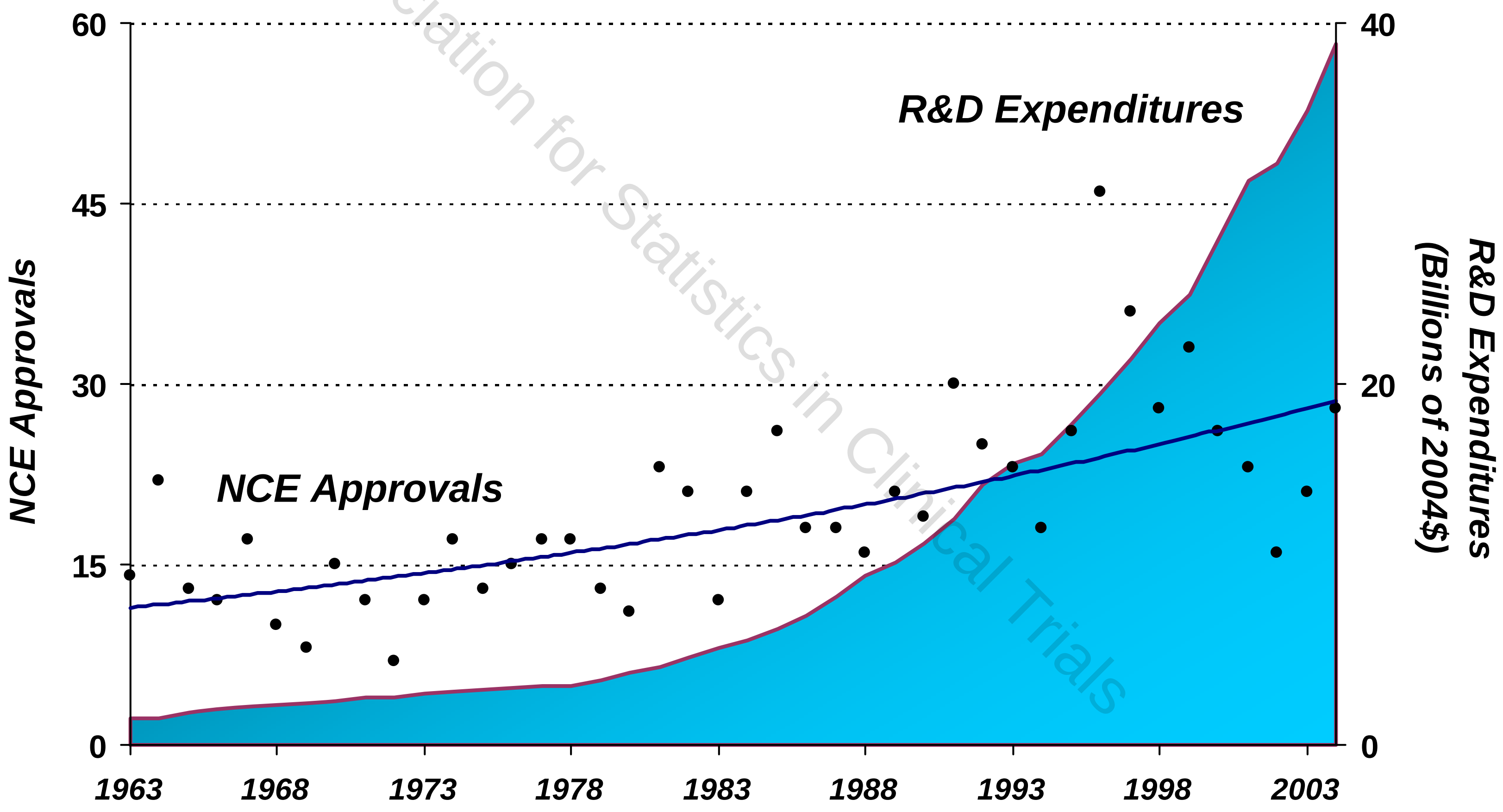
graphic,

chart !!!

Are these more appealing than the words, numbers ?

Pharmaceutical industry: current situation

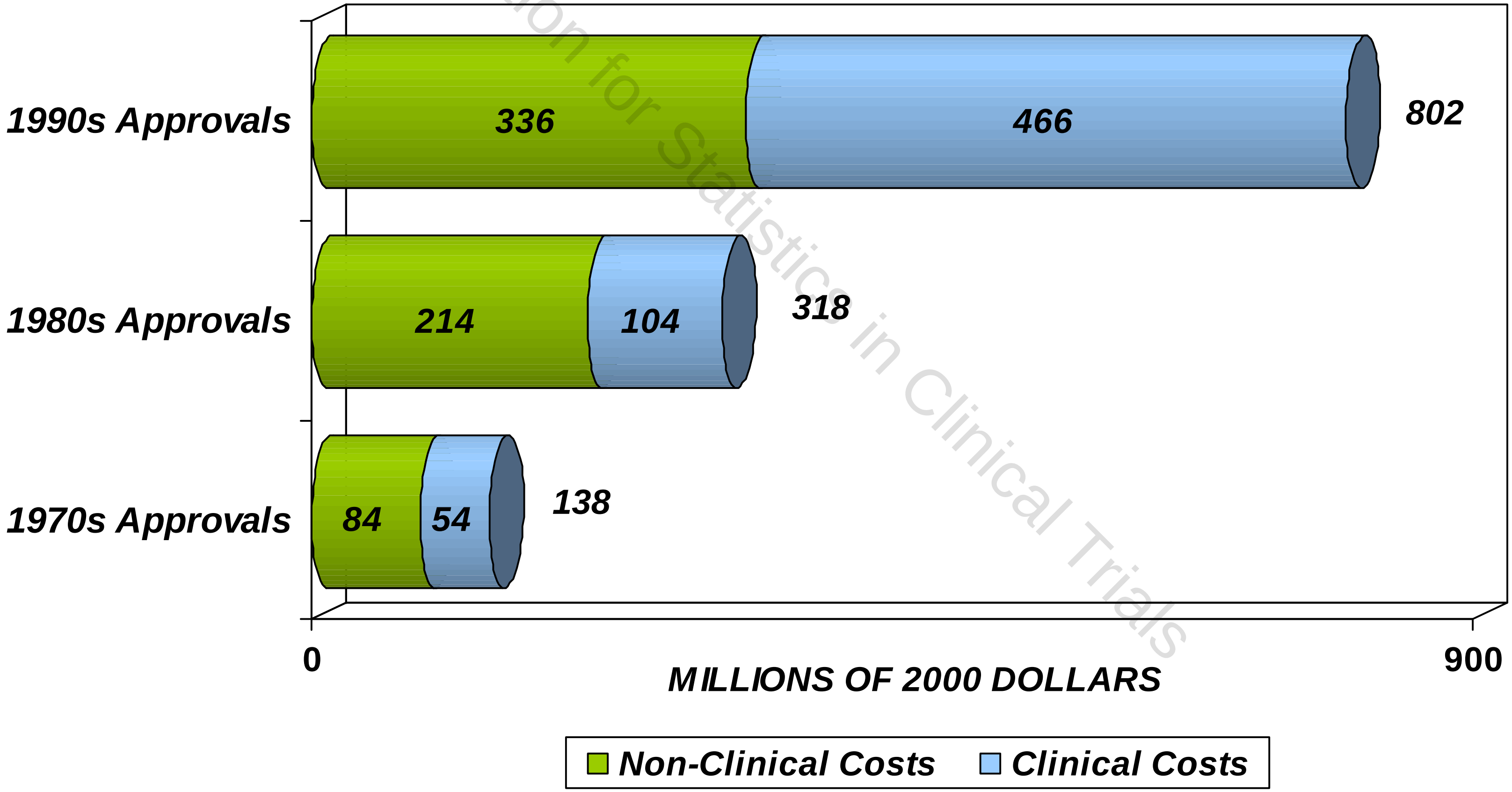
New Drug Approvals Are Not Keeping Pace with Rising R&D Spending



R&D expenditures are adjusted for inflation Source: Tufts CSDD Approved NCE Database, PhRMA, 2005

Pharmaceutical industry: current situation, contd.

Capitalized Costs have Increased 481% from the 1970s to the 1990s



Source: DiMasi et al., J Health Econ, 2003;22:151-185

Pharmaceutical industry: current situation, contd.

Possible reasons for non approvals

Adverse drug reactions are believed to cause over **100,000 deaths per year in the U.S.**

Serious adverse events are among the top 5 causes of death

Drug-related mortality and morbidity

- estimated to cost U.S. health care system > $150Bn in 2000 dollars
- could represents > 5-10% of total U.S. health care spending

19 drugs have been withdrawn from the market since 1998

- Withdrawals ranged 3-7 years from introduction
- 26% of drugs introduced 1980-2006 have black box warnings

Drugs: ARE THEY SAFE ?

Source: FDA/CDER/PhRMA/AASLD Meeting Arthur Holden, Chairman, SAEC Ltd. , 27 March 2007

Pharmaceutical industry: current situation, contd.

Who defines: “drug is safe” & who approves them ?

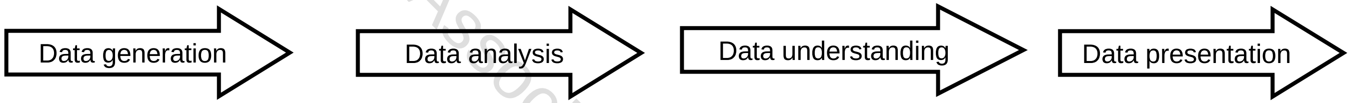
- Health Authorities: USFDA, EMEA, PMDA, etc.
- Approval based on clinical trial data (Safety & Efficacy)
  - CSR based on ICH E3: Appendix 14, Appendix 16 Tables/Listings/Figures
  - New standards for Safety Review, February 2005
  - Clinical Review Template – annotated Safety Section

Good Review Practices

Review Guidance: Conducting a Clinical Safety Review of a New Product Application and Preparing a Report on the Review, February 2005 84 pages

Data

Creation, analysis, representation



	Tables	Graphs
Industry	Health Authorities Journals Documented evidence	Meetings Illustrations Exploration
In general	Organize and document Communication Research	Structure and pattern Communication Research Hidden relationships Target audience: Unfamiliarity with data Less skilled quantitatively / statistically

Take a look at some of the commonly used graphs

Commonly used graphs

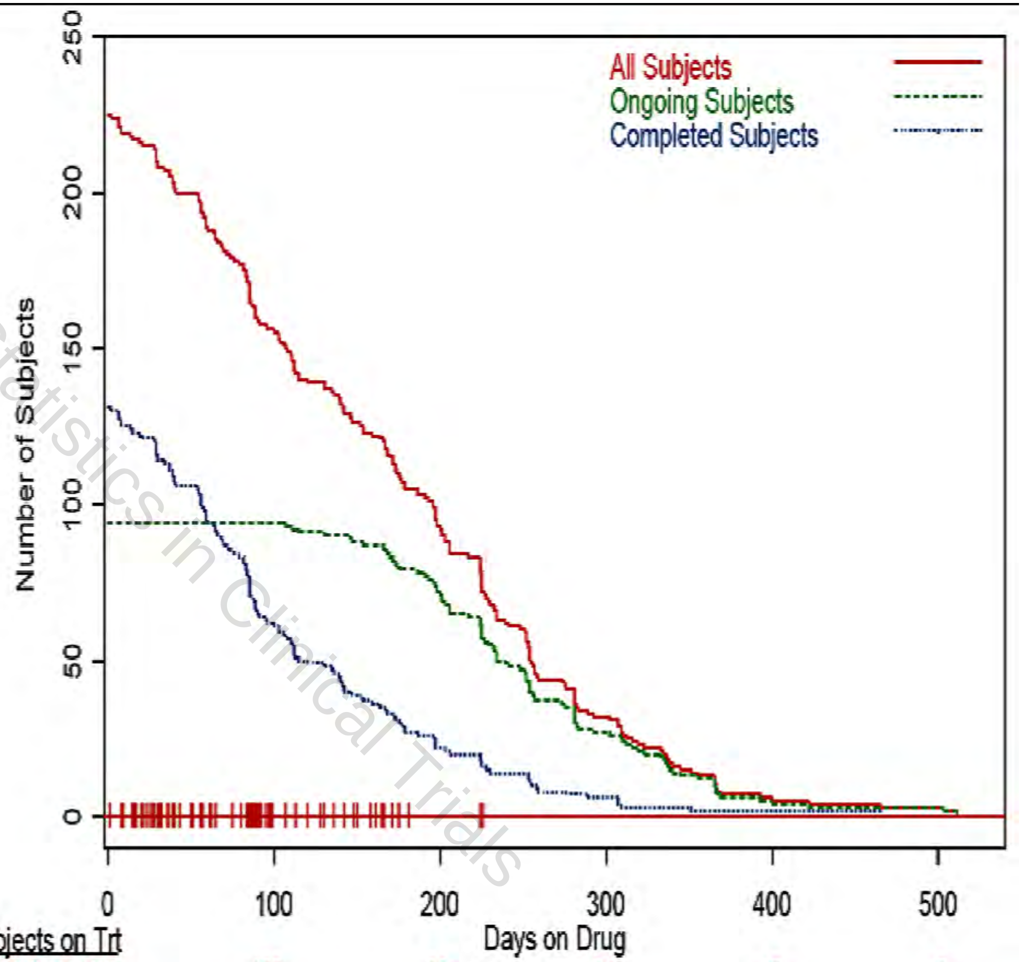
Summary of exposure

	Exposure (days)
N	225
Mean	198
Median	187
Min	7
Max	500

Exposure to drug:

How much drug and How long ?

Are there any AE's ?



Subjects on Trt	0	100	200	300	400	500
All Subjects	225	155	92	31	5	2
Ongoing Subjects	94	94	71	26	4	2
Completed Subjects	131	61	21	5	1	0

NOVARTIS

Commonly used graphs

Adverse events

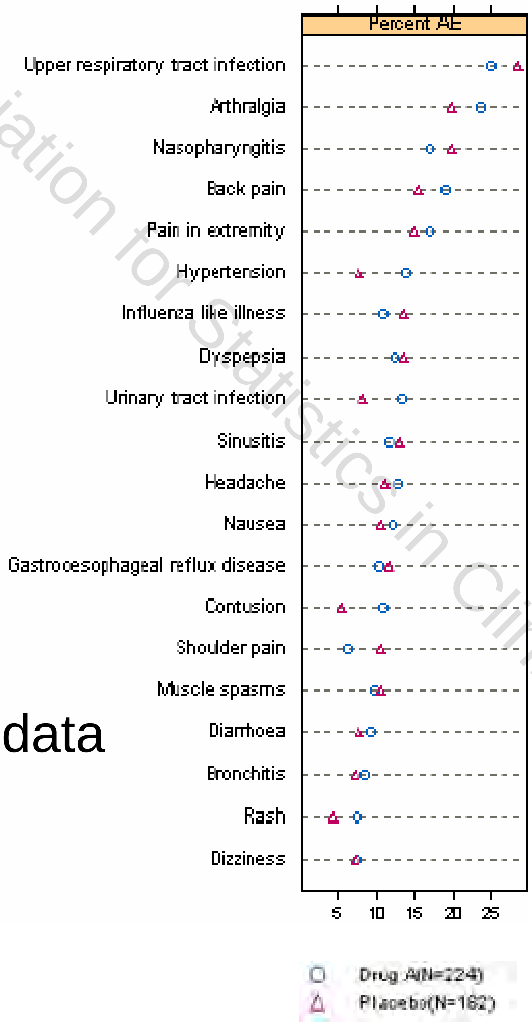
PREFERRED TERM	Placebo (N = 184) n (%)	Drug A (N = 224) n (%)
-Total	146 (79.3)	195 (87.0)
CONSTIPATION	43 (23.4)	59 (26.3)
ASTHENIA	32 (17.4)	39 (17.4)
BACK PAIN	27 (14.7)	37 (16.5)
BONE PAIN	23 (12.5)	34 (15.1)
FATIGUE	22 (12.0)	29 (12.9)
HYPOCALCAEMIA	5 (2.7)	16 (7.1)
INSOMNIA	22 (12.0)	10 (4.5)

Any signals in the safety data

Labs

Vital signs

ECGs

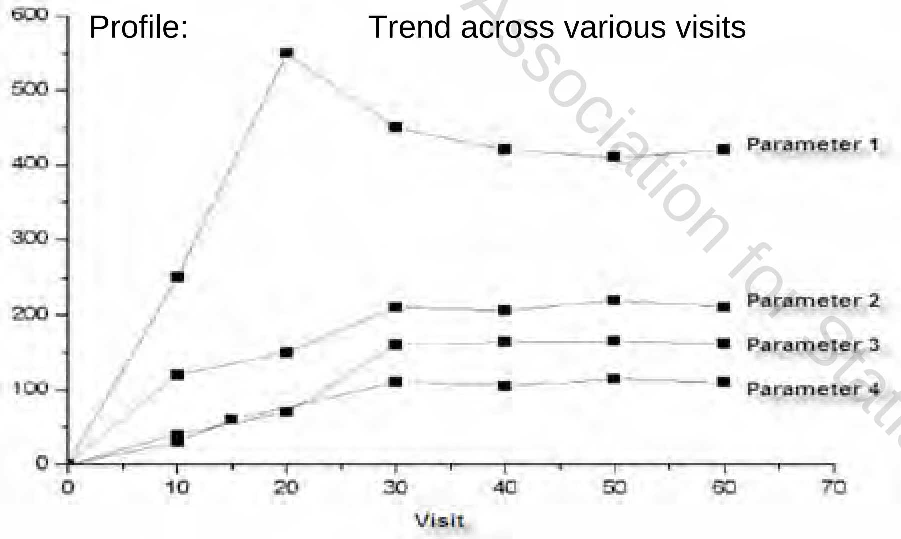


Some other graphs for AE's:

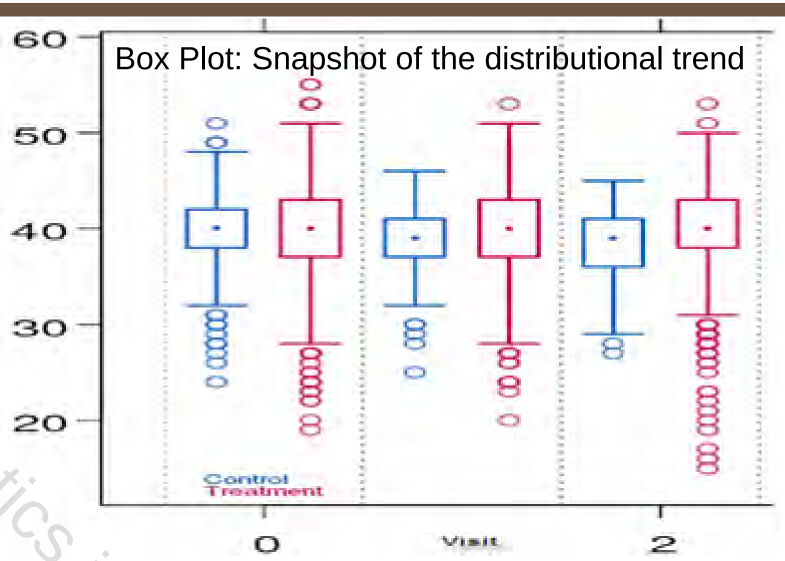
- •Treatment
- •System organ class
- •Preferred term
- •Severity / CTC grade
- •Relationship with drug
- •Special interest
- •Time to event

Commonly used graphs

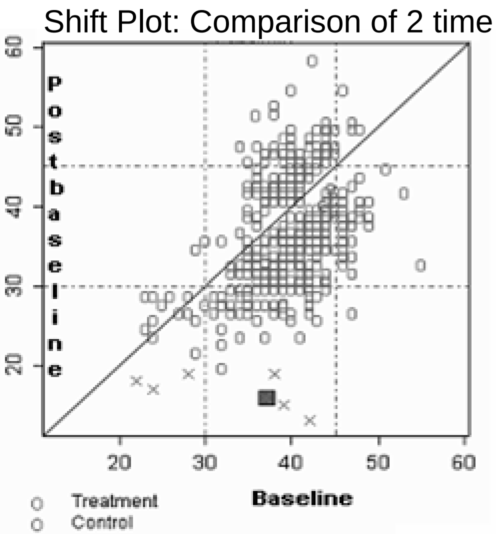
Lab / ECG / Vital sign reports: Profile, Shift Plots, Box Plot, Mean (SD)



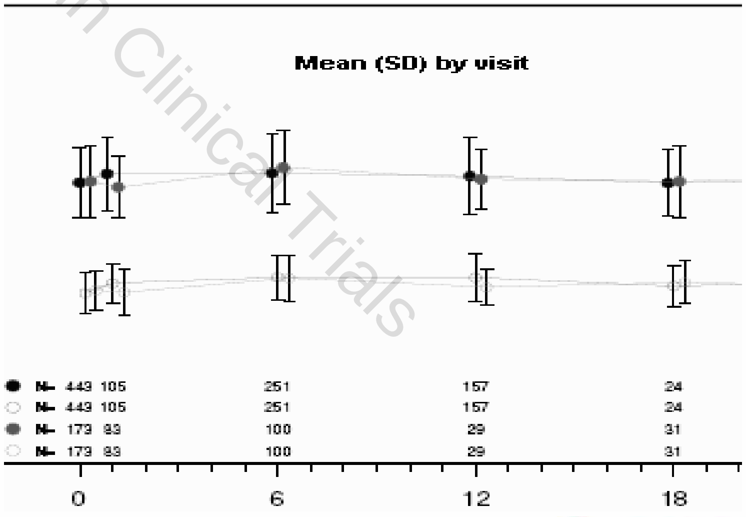
Profile: Trend across various visits



Box Plot: Snapshot of the distributional trend

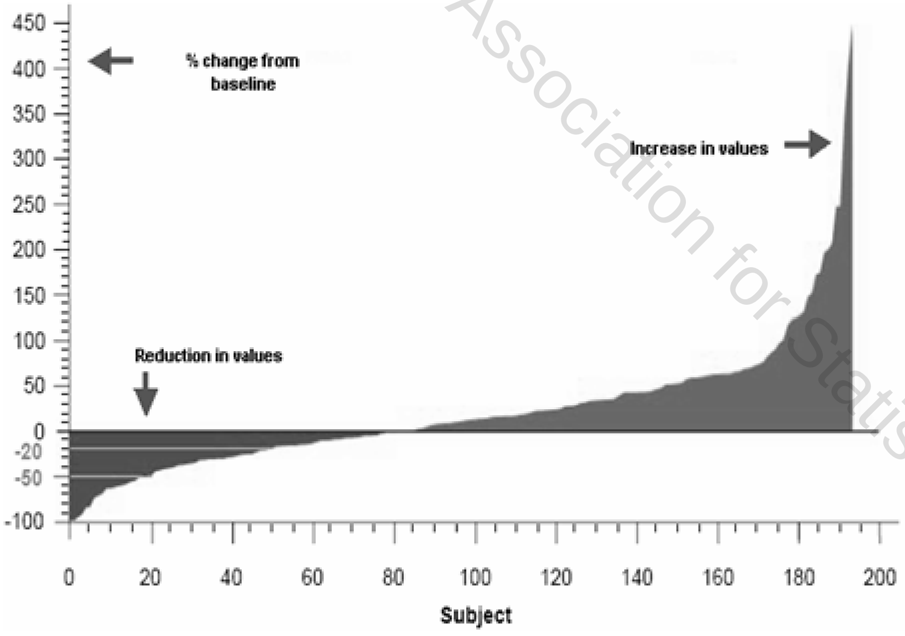


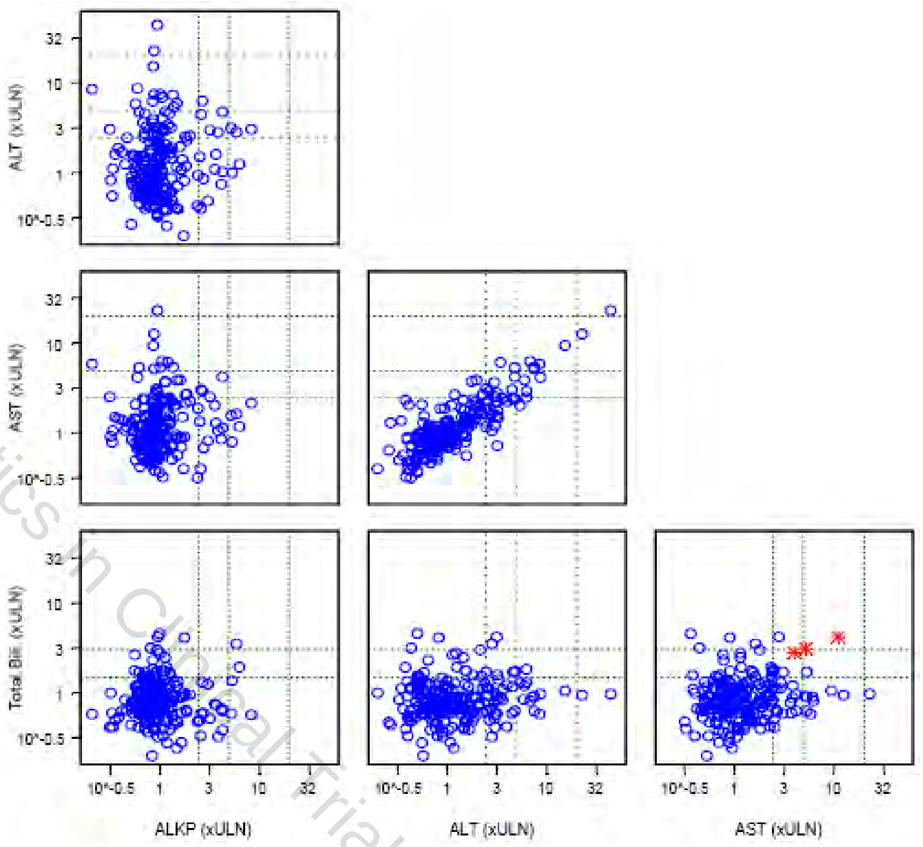
Shift Plot: Comparison of 2 time points



Commonly used graphs

Waterfall Plot, Hy’s law (Liver toxicity)



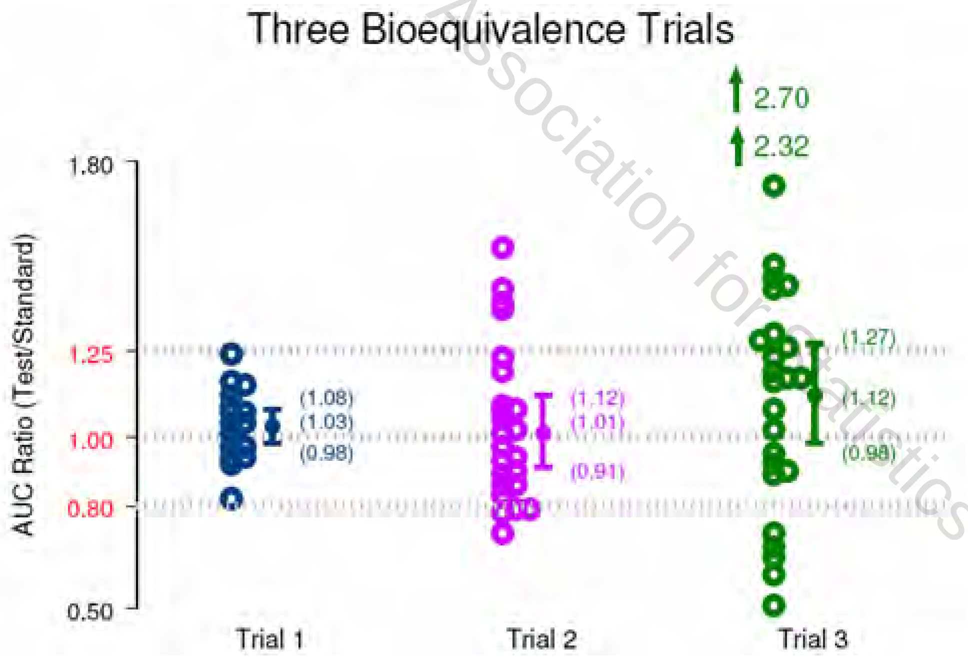


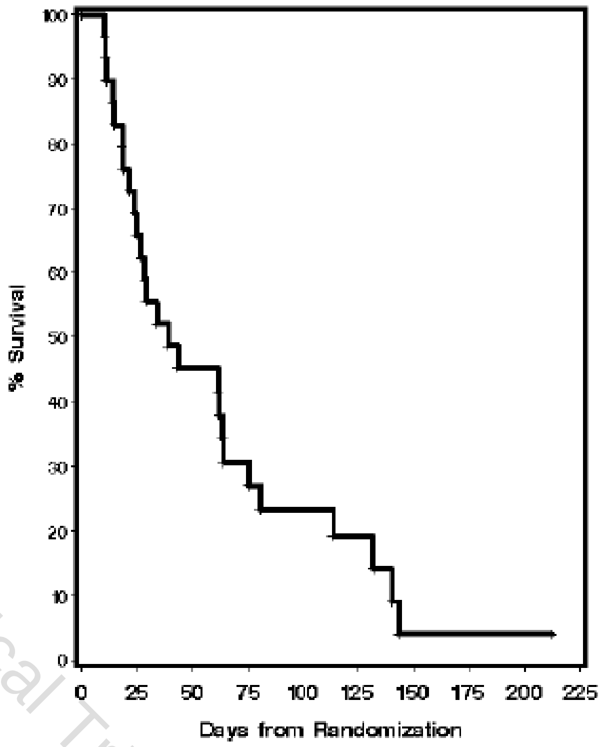
Displays the distribution by looking at the order statistics

Traditionally used to identify the shrinkage in tumor size

Commonly used graphs

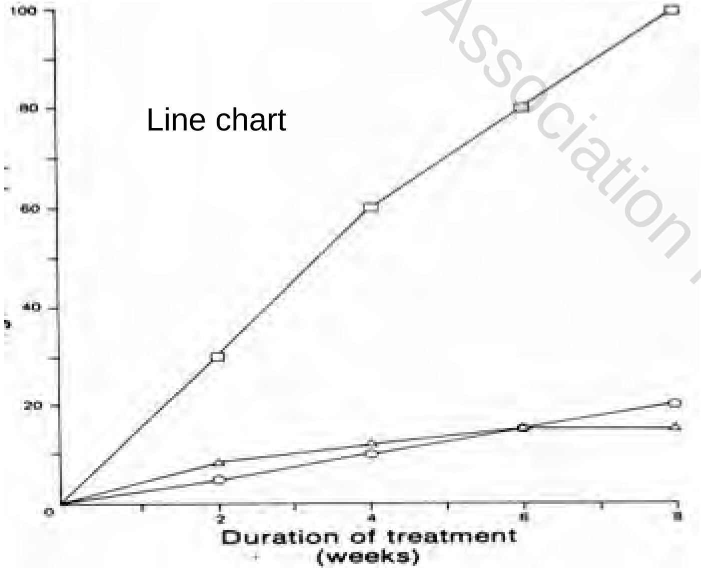
Bioequivalence Trials, Survival curves

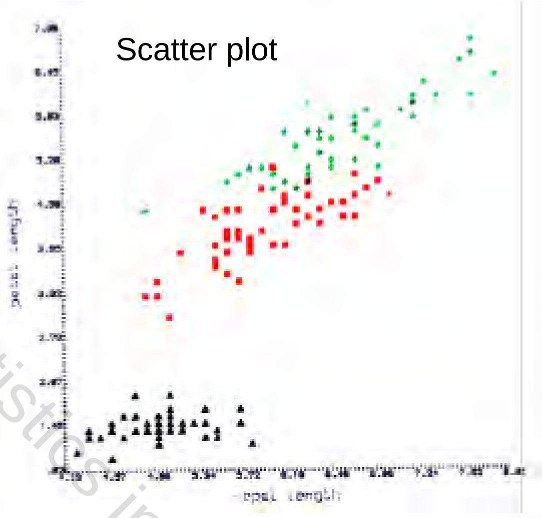


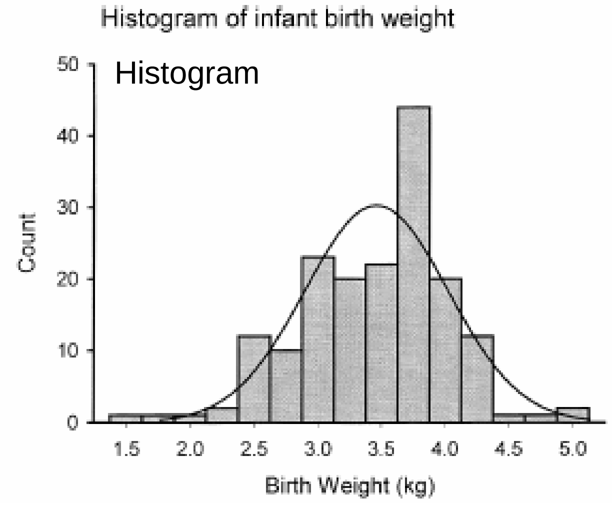


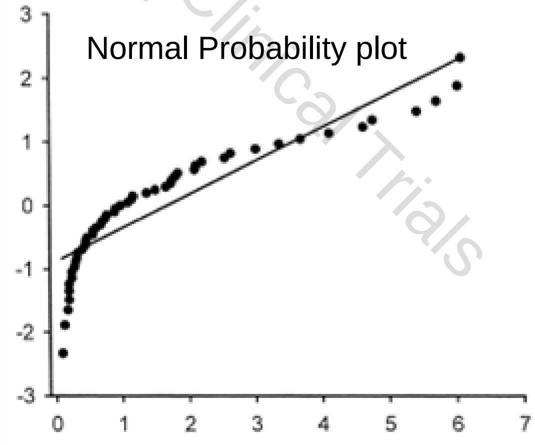
Commonly used graphs

Different types





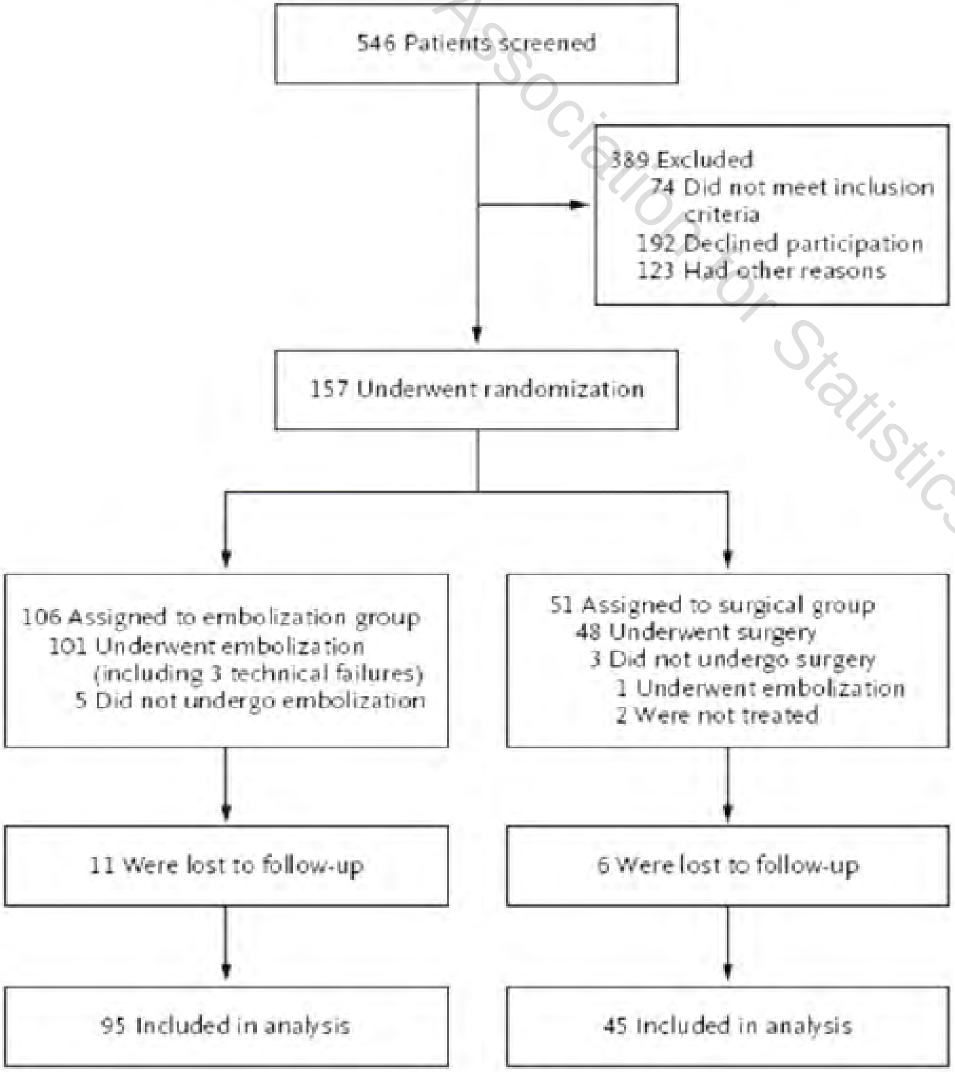




Graphs: that can be used

Graphs: that can be used

(1) Clinical trial overview: Trial profile



546 Patients screened

389 Excluded
74 Did not meet inclusion criteria
192 Declined participation
123 Had other reasons

157 Underwent randomization

106 Assigned to embolization group
101 Underwent embolization (including 3 technical failures)
5 Did not undergo embolization

51 Assigned to surgical group
48 Underwent surgery
3 Did not undergo surgery
1 Underwent embolization
2 Were not treated

11 Were lost to follow-up

6 Were lost to follow-up

95 Included in analysis

45 Included in analysis

Integral part of CONSORT statement: Flow diagram

1 page high level summary

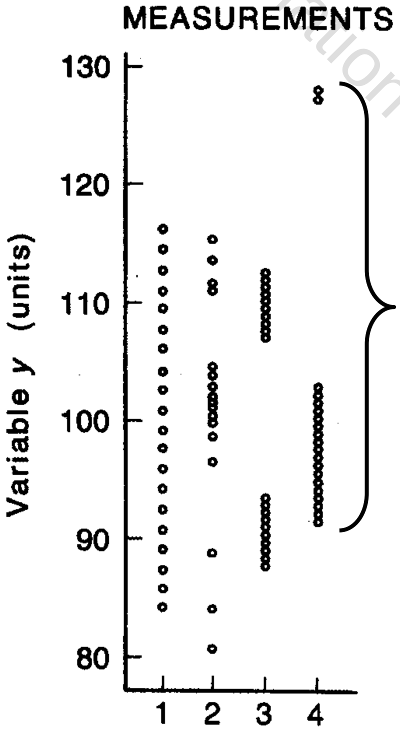
Very easy to understand

Should this be included in CSRs ?

Graphs: that can be used

(2) Ways to represent data sets (1/3) : data points

Listings run into 100’s of pages



Range

As good as listing all the data on one page

Very helpful if the number of patients is small.

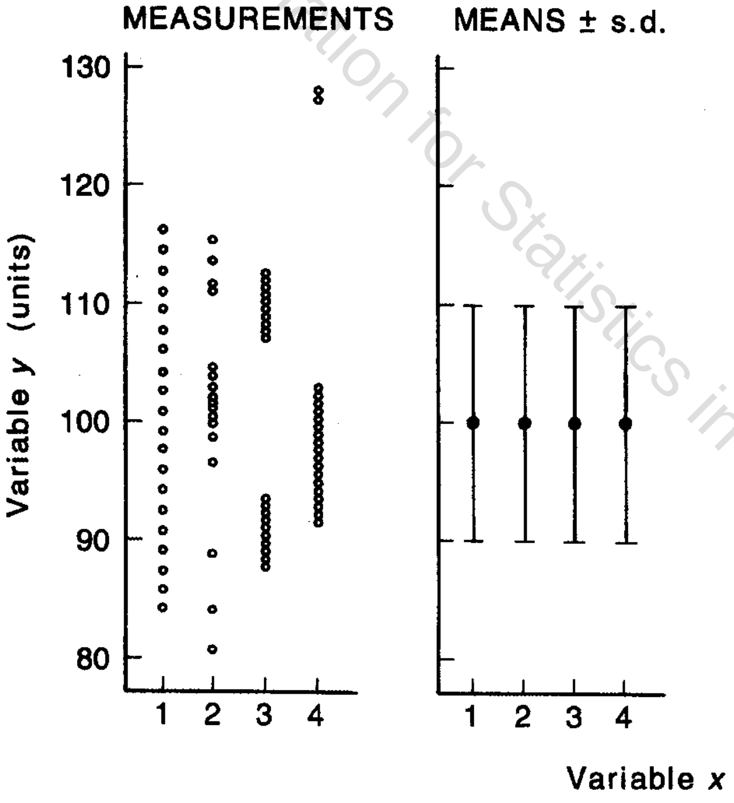
Too cluttered in case there are a lot of patients

Valiela (2001) Doing Science: Design, Analysis, & Communication of Scientific Research. New York: Oxford University Press.

Graphs: that can be used

(2) Ways to represent data sets (2/3) : data points, Mean +/- SD

Display of individual values and summaries together side by side

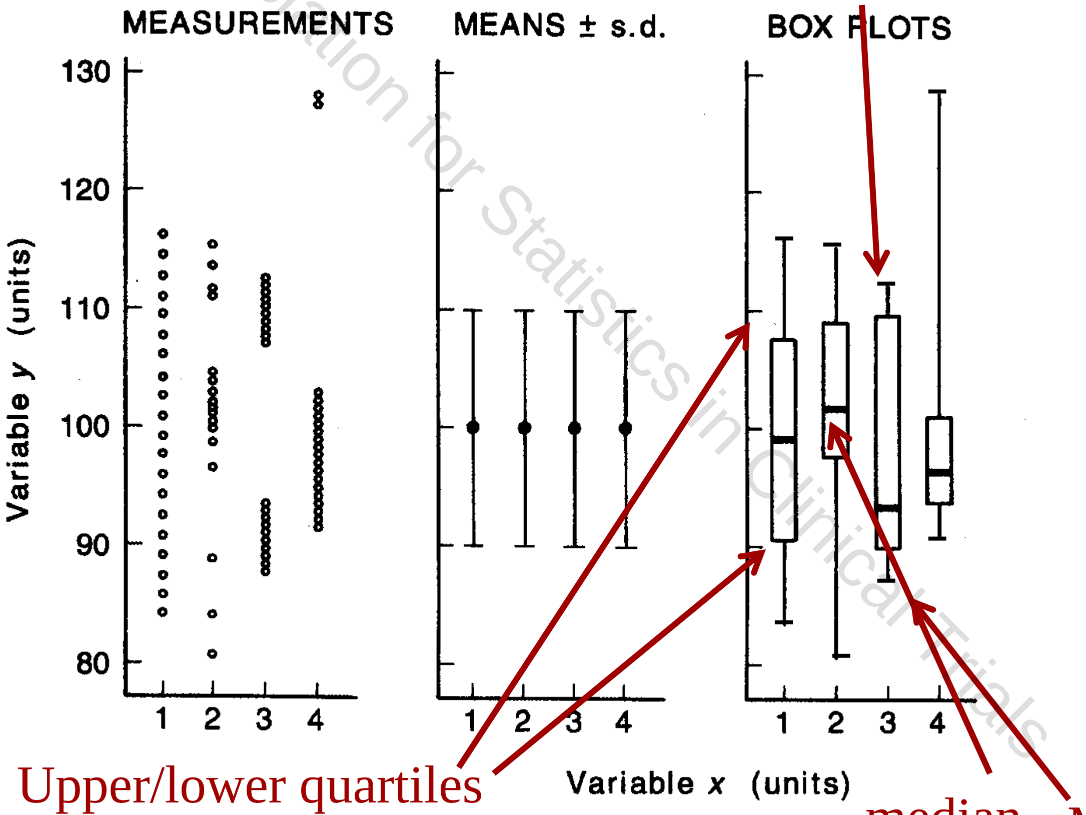


Valiela (2001) Doing Science: Design, Analysis, & Communication of Scientific Research. New York: Oxford University Press.

Graphs: that can be used

(2) Ways to represent data sets (3/3) : data points, Mean +/- SD, Box Plots

Display of individual values and descriptive statistics together side by side



Max or 1.5 IQR

Upper/lower quartiles

median

Min, 1.5 IQR

Listing

+

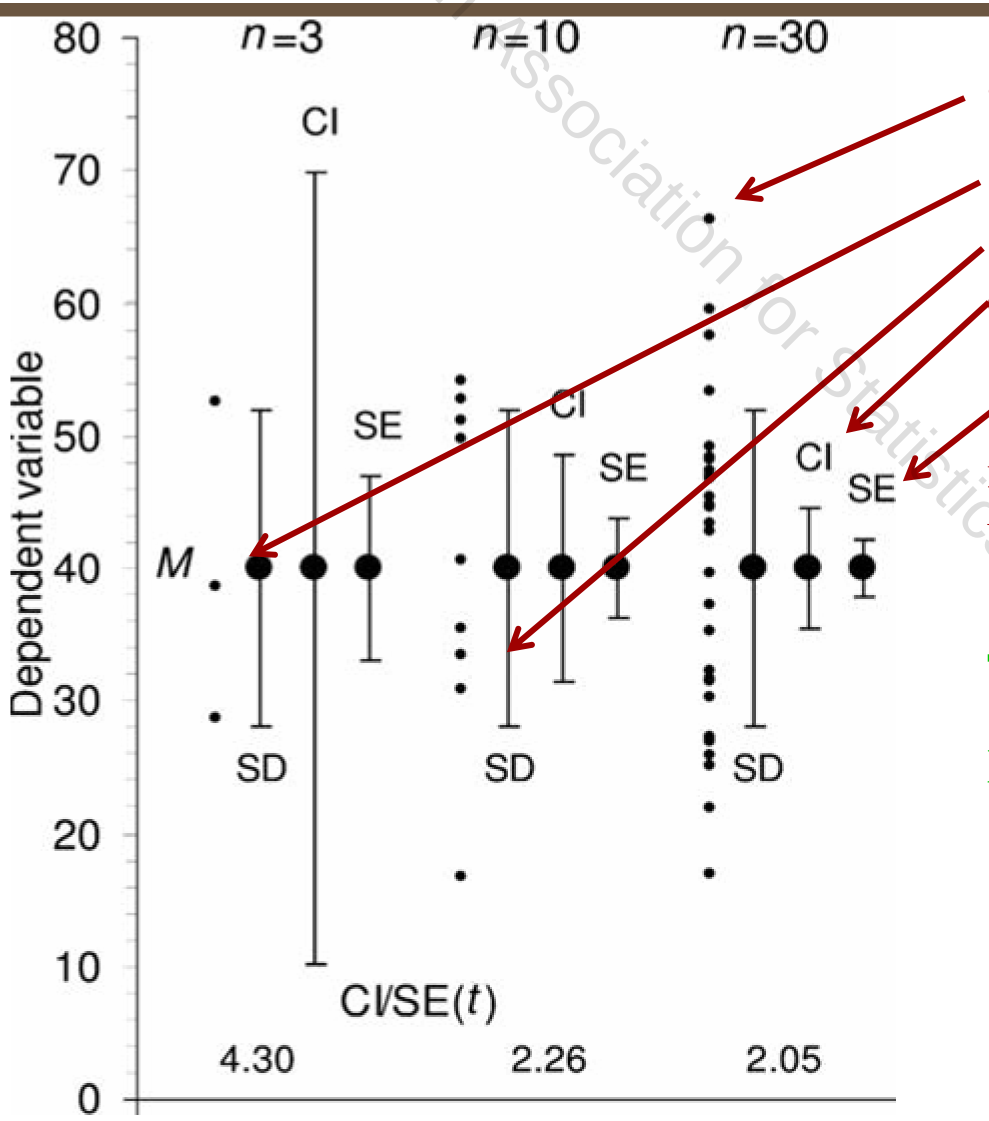
Table

Valiela (2001) Doing Science: Design, Analysis, & Communication of Scientific Research. New York: Oxford University Press.

NOVARTIS

Graphs: that can be used

(3) Inferential error bars



• = data points

● = data mean M

SD = Error bars

CI = 95% Confidence intervals

SE = Standard Error

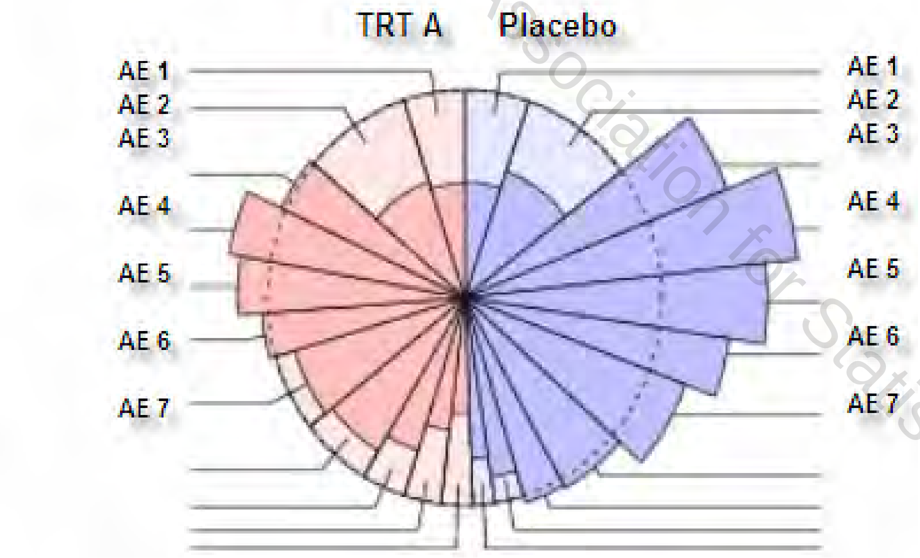
Ratio of CI / SE = t-test, for that specific n.

Values of t are shown at the bottom.

To find sig difference between 2 treatments: plot the differences.

Graphs: that can be used

(4) A modified Pie chart: Spie chart



A Spie chart combines two pie charts to compare partitions.

One pie chart is drawn as-is, and serves as the basis for comparison.

The other is superimposed on the first, using the same angles for the slices,

but different radii, so as to achieve the desired areas.

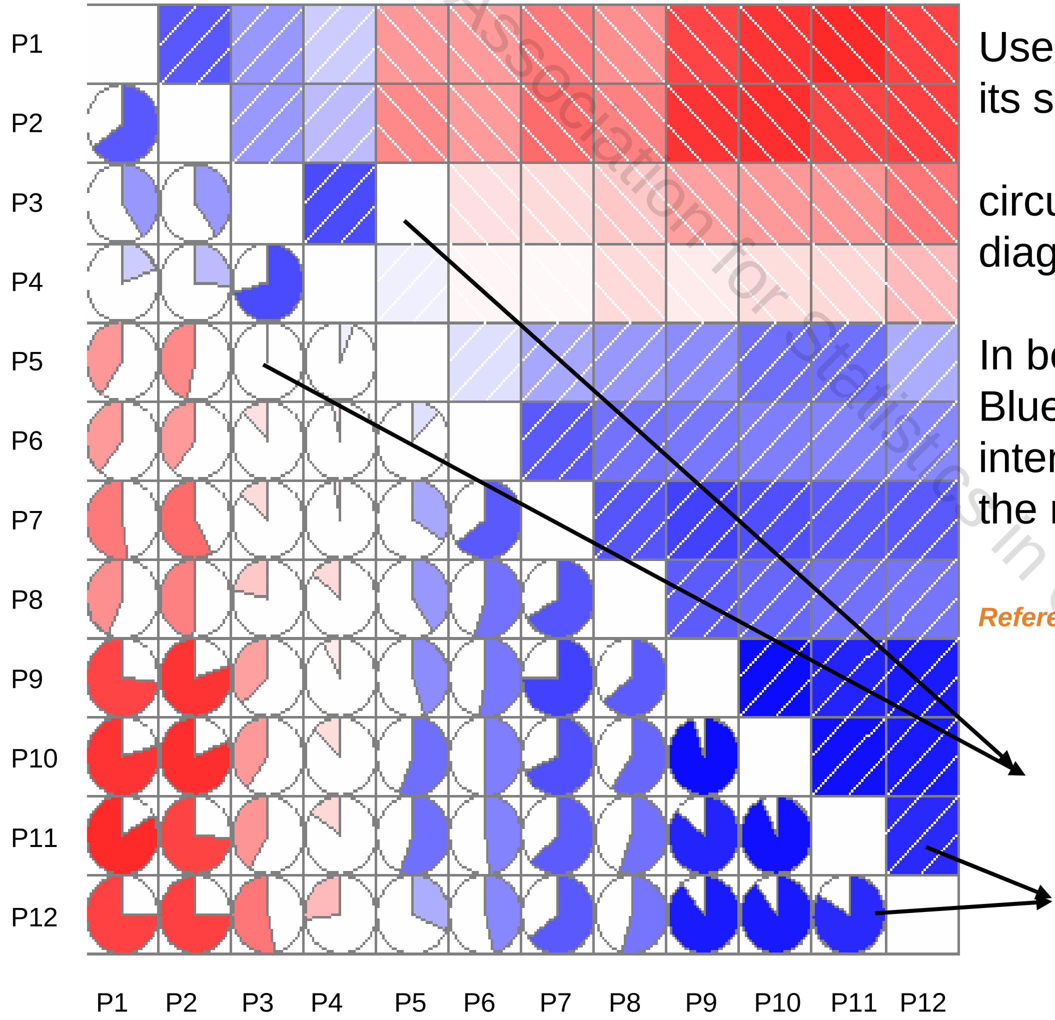
Any frequency for any variable can be plotted.

E.g. AE's are plotted. AE 4 is seen more in Placebo than TRT A.

Reference: D. G. Feitelson, "Comparing Partitions with Spie Charts". Technical Report 2003-87, School of Computer Science and Engineering, The Hebrew University of Jerusalem, Dec 2003. URL: http://www.cs.huji.ac.il/~feit/papers/Spie03TR.pdf

Graphs: that can be used

(5) Corrgrams: useful in multivariate analysis



Use the value of a correlation to depict its sign and magnitude.

circular `”pac-man" pies, and shading, with diagonal stripes indicating the direction.

In both,
Blue is positive correlations, Red for negative,
intensity of shading proportional to
the magnitude of the correlation.

Reference: Michael Friendly

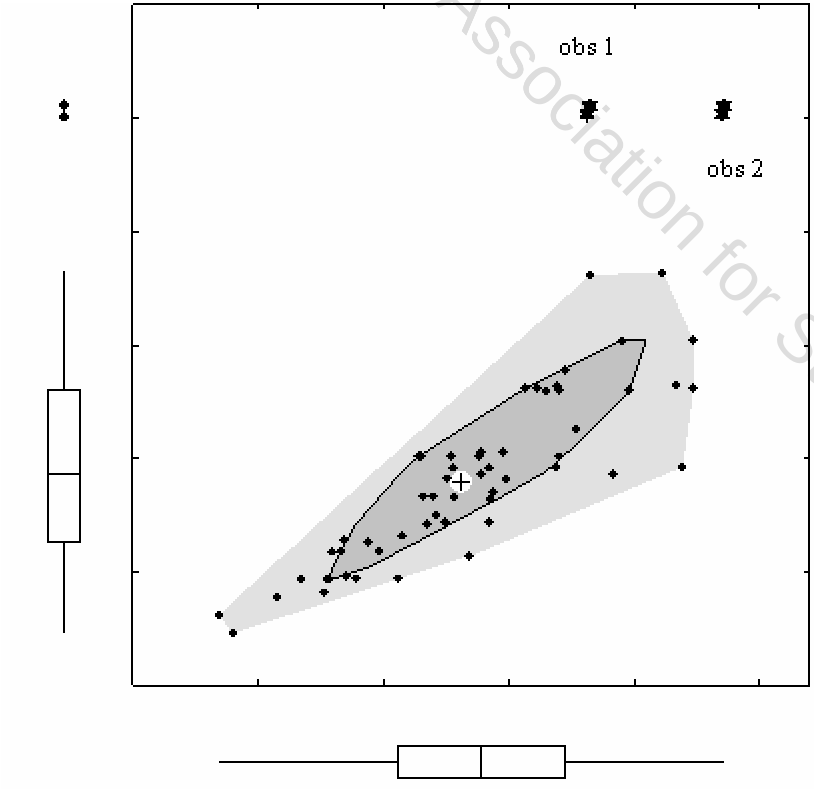
Nearly no Correlation between P3 and P5

Very high positive Correlation between P11 and P12

Change from baseline values can be plotted, 1 plot for each treatment.

Graphs: that can be used

(6) Bagplot: Tukey (1975), Peter Rousseeuw and Ida Ruts



The large + marks the bivariate median. The dark inner region (the “bag”) contains the 50% of the observations with greatest bivariate depth.

The lighter surrounding “loop” marks the observations within the bivariate fences.

Observations outside the loop are plotted individually and labeled.

Location: the depth of median

Spread: the size of the bag

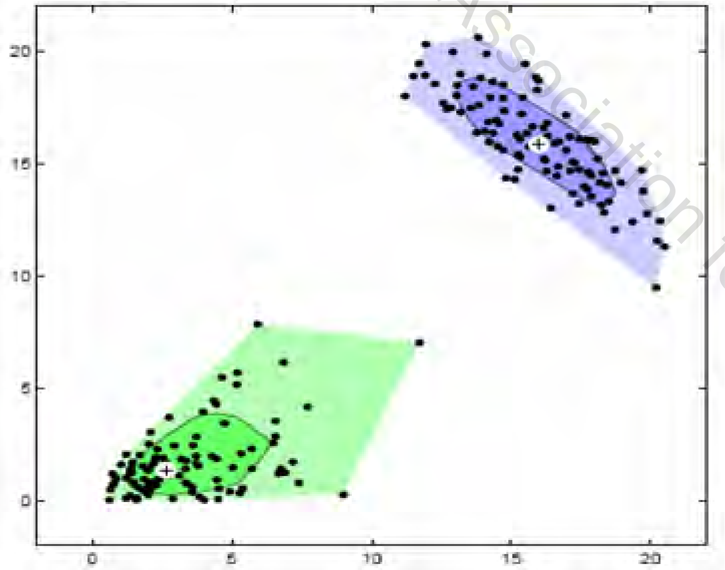
Correlation: the orientation of the bag

Skewness: the shape of the bag and the loop

Tails: the points near the boundary of the loop and the outliers

Graphs: that can be used

(6) Bagplot: Tukey (1975), Peter Rousseeuw and Ida Ruts

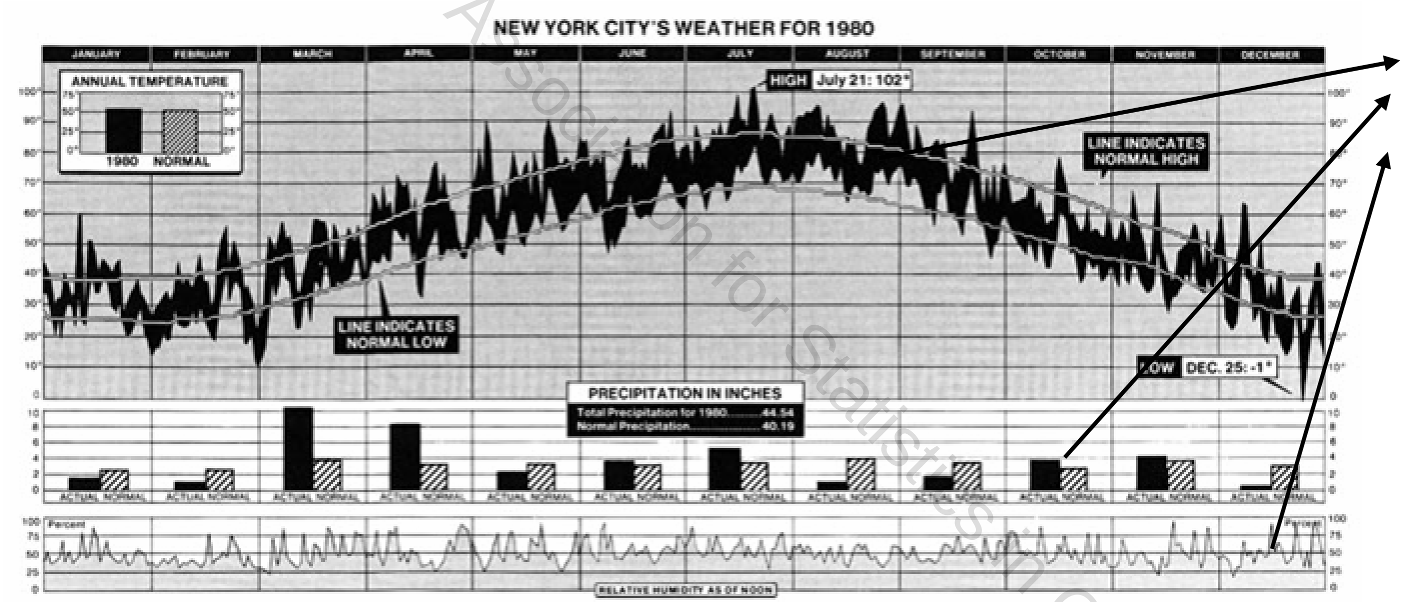


100 points in each dataset

Location: the depth of median	Median lies in the lower part of the bag	Median is in the middle of the bag
Spread: the size of the bag	Roughly similar	Roughly similar
Correlation: the orientation of the bag	Positive	Negative
Skewness: the shape of the bag and the loop	Very skewed: median as it lies in the lower part of the bag where the loop is narrow and right part is wider	Data is nicely balanced
Tails: the points near the boundary of the loop and the outliers	Medium tailed and no outliers	Medium tailed and no outliers

Graphs: that can be used

(7) Chart: New York weather in 1980



Temperature,
Precipitation,
Relative humidity

2200 numbers summarize the trends and patterns

In the graph of temperature, the area is filled between the daily low and daily high.

What makes this graph successful, in spite of the large amount of information presented are

(a) clear visual comparisons between the 1980 data and the long-run average,

(b) clear textual labels,

(c) visual segregation between the three series.

For example, it is easy to see that March and April were about of normal temperature, but a lot wetter.

Months = Visits

1980 = Treatment A

Average = Placebo

Low/High= Min/Max placebo per visit

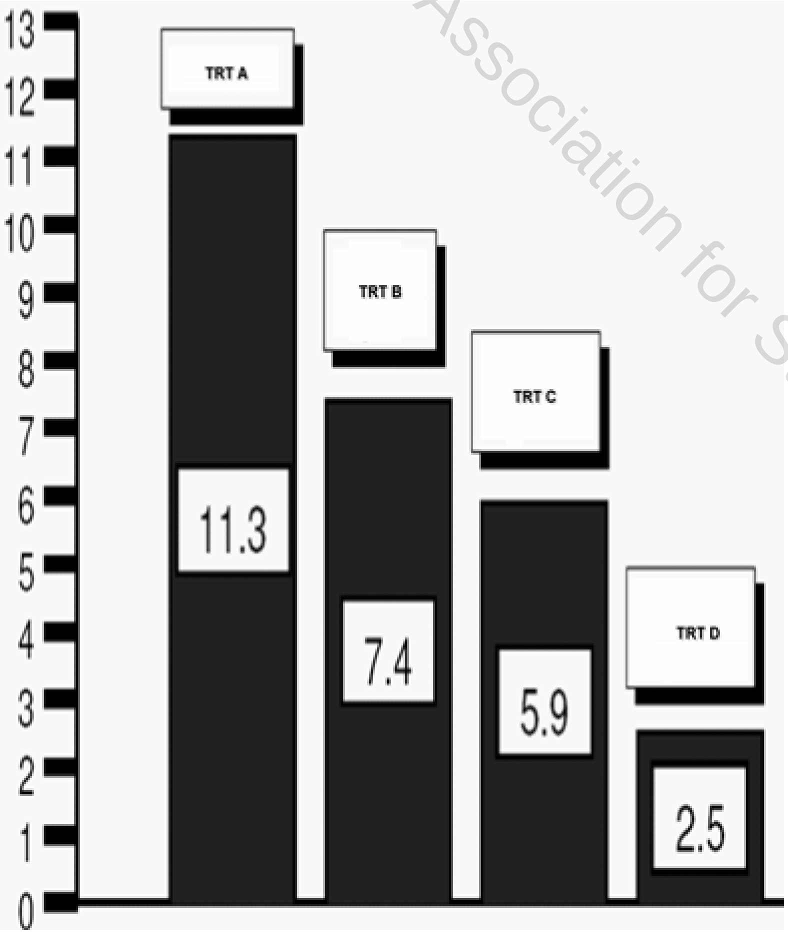
3 parameters

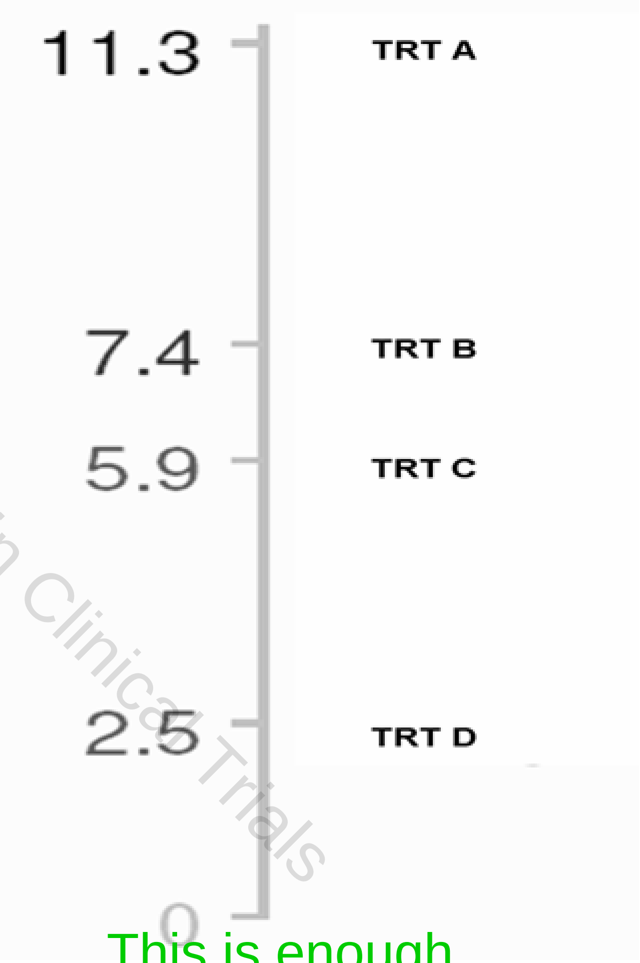
Source: New York Times (Jan. 11, 1981, p. 32; Tufte (1983), p. 30)

Examples of good and bad graphs

Examples of good and bad graphs

(1) Too much ink





Not needed

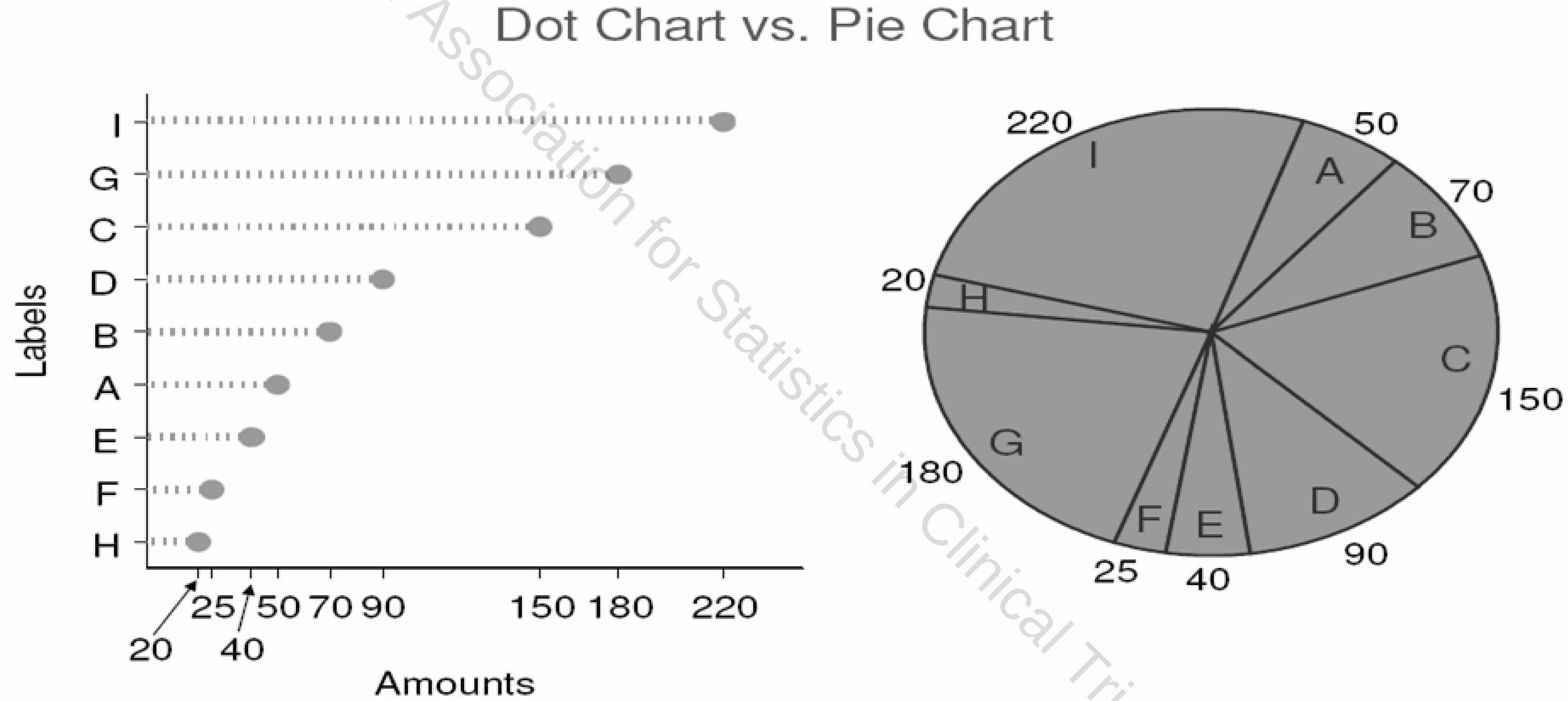
This is enough

Too much ink

Emphasis on data

Examples of good and bad graphs

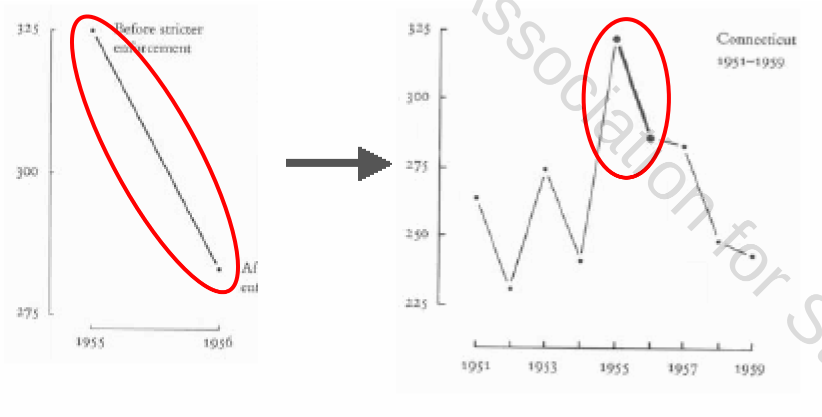
(2) Combine dot and Pie chart



Plot a dot chart to better comprehend a pie chart.

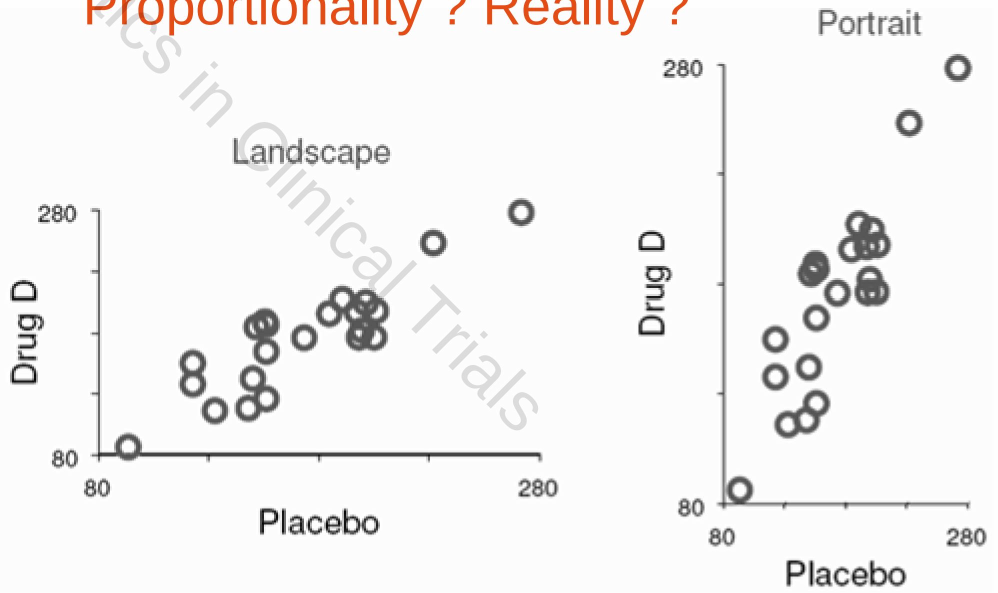
Examples of good and bad graphs

(3), (4) Show context



Is something hidden ?

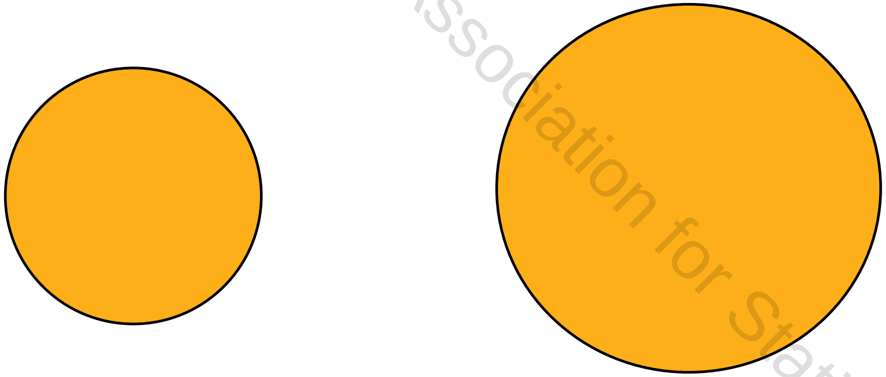
Proportionality ? Reality ?



NOVARTIS

Examples of good and bad graphs

(5), (6) Distortion

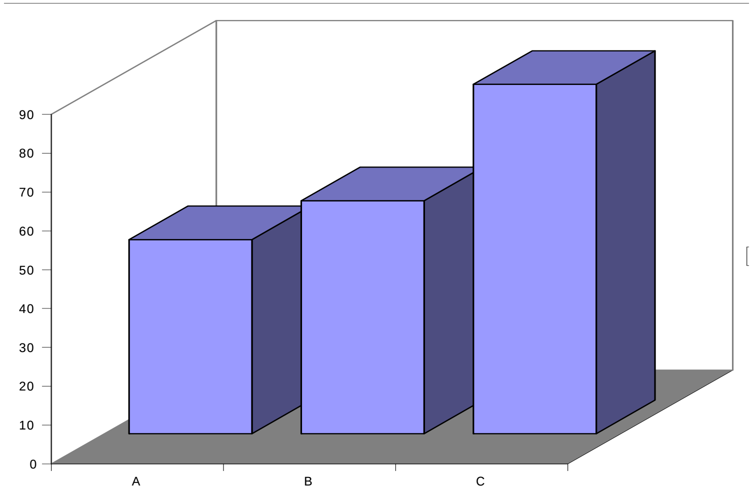


Number of people on Drug A

Number of people on Drug B

Readers do not compare areas in circles correctly

(larger circle does not *appear* to have the increased area it actually does)



3-dimensional graphs may fool the eye

Source: How to Effectively Communicate Your Findings Mary Purugganan, Ph.D. Leadership & Professional Development Workshop March 23, 2007

Do’s

Some points to keep in mind

Good graphic

- Terms are spelled out
- Text runs left to right
- Data are clarified with small notes
- Legends vs. labels –decide which one is appropriate
- Graphic attracts viewer
- Color choices (blue – good)
- Font type is clear, precise, modest
- Upper & lower case, with serifs
- Graphics should tend toward the horizontal, greater in length than height.

Bad graphic

- Excessive abbreviations to decode
- Text in vertical or multiple directions
- Graphic requires repeated references to scattered text
- Repeated back & forth between legend & graphic
- Graphic is repellent, filled with chart junk
- Dark letters on dark contrast (Red & green)
- Type is dense, heavy, overbearing
- All upper case, sans serif

Source: Summary (adapted from Tufte, pg 183)

Do's

Accuracy in perceiving graphical cues, Cleveland's experiments (1985)

Position along axis ↓ Most accurate perception, use more

Length

Angle / slope

Area

Volume

Color / shade ↓ Least accurate perception, use less

Show the data

Avoid distorting

Present many numbers in small region

Encourage thinking

Make it attractive

Reveal data at various levels

Make large datasets readable

SAS codes

Refer

www.math.yorku.caSCS.sas

www.sas.com

www.gsociology.icaap.org/methods/presenting.htm

Thank you !!!

Back up slides

Why improve data presentation?

- To draw accurate conclusions
- To demonstrate professionalism
- To increase your credibility
- To better analyze, synthesize, and understand your data
  - To see hidden relationships
  - To appreciate limitations, gaps
  - To formulate new questions

USFDA New Standards for Safety Review (2)

- AE incidence by interaction (cont.)
  - Relative risks and attributable risks for subgroup differences
  - Life table/ time-to-event analyses/ cumulative incidence anlayses
  - Hazard rates – risk over time estimation
- Less common AEs
  - Identify and group by body system for rates
- Laboratories
  - Overview of testing methodology
  - Analysis of measures of central tendency
  - Analysis of outliers or shifts to abnormal
  - Marked outliters and dropouts due to lab abn
  - Dose dependency
  - Time dependency
  - Demographic interactions
  - Drug-drug interactions
  - Underlying medical condition interactions
  - Special section on Liver laboratory abn
  - Shift tables
  - Scatter plots
  - Box plots
  - Cumulative distribution displays
  - Tables of deviation in >1 parameter
- Vital signs
  - Overview of testing
  - Analysis of measures of central tendency
  - Analysis of outliers or shifts to abnormal
  - Marked outliters and dropouts due to lab abn
- ECG's
  - Describe baseline and number of on-study ECGs
  - Analysis of measures of central tendency
  - Analysis of outliers or shifts to abnormal
  - Marked outliters and dropouts due to lab abn

- Immunogenicity
  - Summarize and assess available data
- Carcinogenicity
  - Summarize and assess
- Special Safety Studies
  - Summarize any such studies
  - Similar to other drugs in pharmacological class?
  - Studies on cumulative irritancy, sensitizing potential
  - Photosensitivity, photoallergenicity
  - Special Thorough QT study
    - To be done on all NMEs
  - Studies to demonstrate a safety advantage over existing therapeutics
- Withdrawal phenomenon or Abuse potential
  - Reivew/summary of relevant studies
  - Scheduling recommendations
- Human Repro and Pregnancy data
- Assessment of Effect on Growth
- Overdose Experience
- Post-marketing experience
- Causality determination
- Adequacy of patient exposure and Safety assessments
  - Refer to ICH
  - Adequate numbers of various demogrpahic subsets
  - Doses and durations of exposrue were adequate to assess safety for intended use
  - Were study designs adequate to answer critical questions
  - Were potential class effects evaluated
  - Did patient exclusions from studies limit relevance of satey assessments
- Review of secondary clinical data sources
  - IND data
  - Post-marketing data
    - Literature reports

Source: DIA 2005, Cooper

USFDA New Standards for Safety Review (3)

- Additional Clinical Issues
  - Level of confidence for dose/regimen
  - Dose-toxicity and dose response relationships
  - Dose modification for special populations
- General assessment of adequacy of Special Animal and/or In Vitro testing
  - Pre-clinical animal models
  - QT studies
- Adequacy of routine clinical testing
  - Labs, vital signs, ECGs, assessment of certain events
- Adequacy of metabolic, clearance, and interaction workup
  - P450 and p-glycoprotien pathways
  - Other drug-drug interaction studies
  - Specify potential safety consequences
- Adequacy of evaluation for potentially problematic AEs that might be expected for a new drug
  - Assess adequacy and note pertinant negative findings (absences of findings)
- Assessment of Quality and completeness of data
  - Generall overall assessment of the quality an dcompleteness of data with a description of the basis for this assessment
- Additional submissions, including safety update
  - Particularly those submission whose data were not incorporated into the rest of the review

- Summary assessment of important identified adverse events
  - Not important limitations of data and make conclusions
- General Methodology
  - Discussion of general methodological issues
- Pooled data vs. individual study data
- Causality determination
- Exploration of predictive factors
  - Plasma levels, duration of treatment, concom meds, concom illnesses, age, sex, race
- Special populations
- Pediatrics
- AC meeting
- Literature review
- Post-marketing Risk management plan
- Other relevant materials
  - Result of consultations with DDMAC, ODS reviews, actual use and labeling comprehension studies, marketing studies
- Overall assessment
  - Conclusions
  - Recommendation (regulatory)
  - Recommendations on post-marketing actions
- Risk management activity
  - Include all such recommended activity with rationale
- Required phase 4 commitments
  - Include the agreed upon studies, the timeline for submission, and basis for each phase 4 commitment
- Labeling review

Source: DIA 2005, Cooper

USFDA New Standards for Safety Review (1)

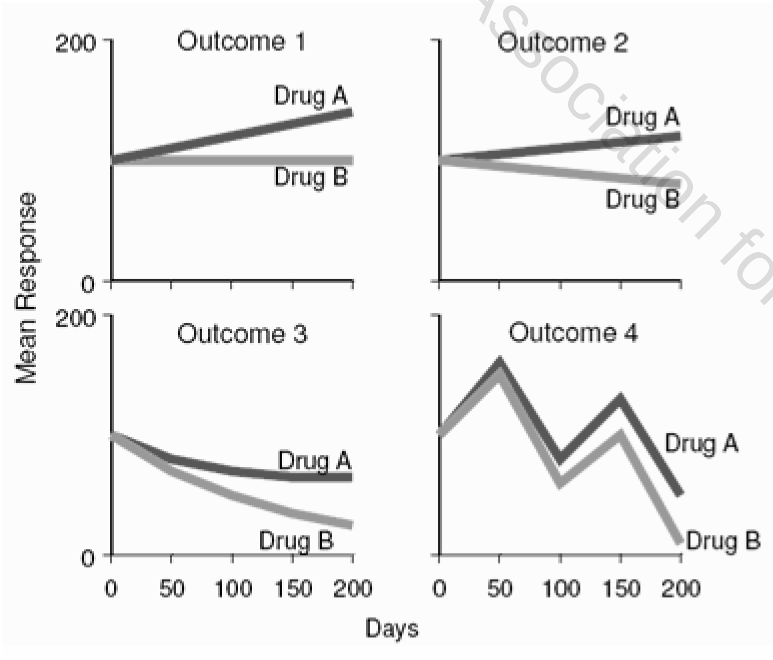
- Deaths
  - Overall mortality
  - Cause specific
  - Expected vs unexpected
  - Dose response
  - Time to death analysis
  - Subgroup analysis
  - Interaction analysis
- SAEs
  - Overall rates
  - Rates by event
  - Dose response
  - By duration of exposure
  - By person-time exposure as denominator
  - Assessment according to alternative explanation
  - Assessment of interaction by subgroup
- Dropouts and other SAEs
  - Overall rates
  - Profile of dropouts (by reason)
  - AEs associated with Dropouts
  - Exposure response
  - Time dependency
- Other significant AEs as defined by ICH
  - Marked lab abnormalities
  - Any AE leading to dropout or intervention
  - Potentially important abnormalities not meeting above definition
- Construct of algorithms of combo's of clinical findings
  - Identify possible combinations of clinical findings that may be a marker for a particular toxicity

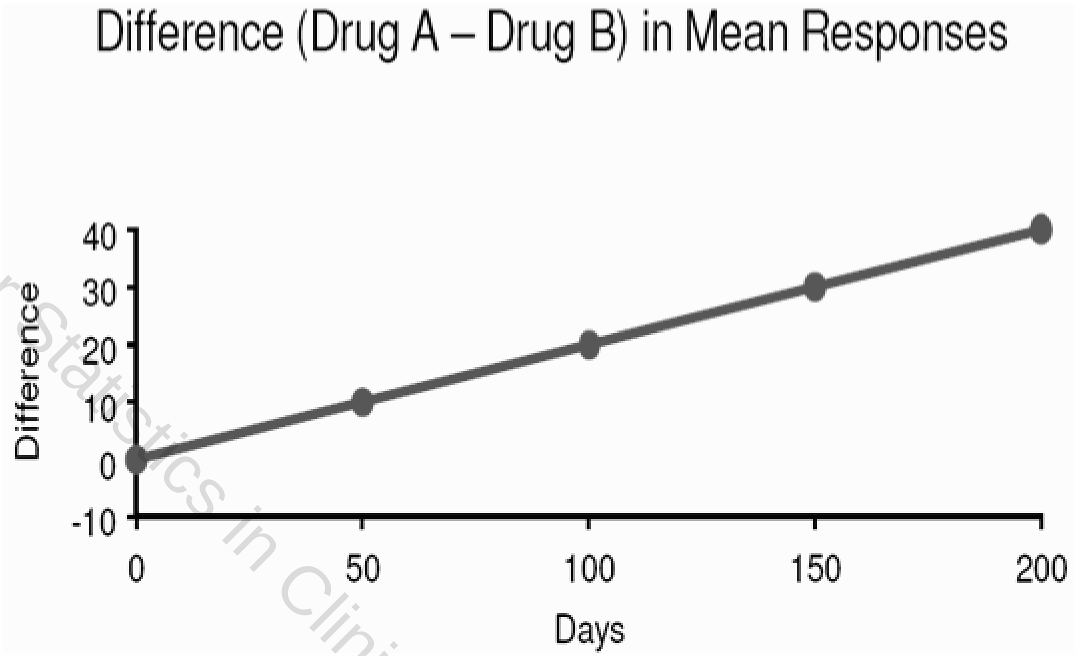
- Identify possible consequences of a safety signal from any source
- Common AEs
  - Incidence for subsets -controlled studies
  - LLT's should be compared to mapped PT's
  - Assess for causality
  - Comparison of severity between treatment arms
- Dose dependency for AEs
  - Titration studies
- Time to onset for AEs
  - Particularly for events that occur commonly
- AE incidence by interaction
  - demographic
    - race, gender, age
  - Drug-drug interaction
  - Underlying medical problems such as DM or renal disease
  - Dose response
    - body weight-adjustted dose
    - cumulative dose
    - Body surface area-adjusted dose
    - dosing schedule
  - Exposure adjusted event rates "person-time approach"
    - When hazard rate is constant over time
    - Break observation period into intervals

Source: DIA 2005, Cooper

Examples of good and bad graphs

(2) Trap too simple to fall in

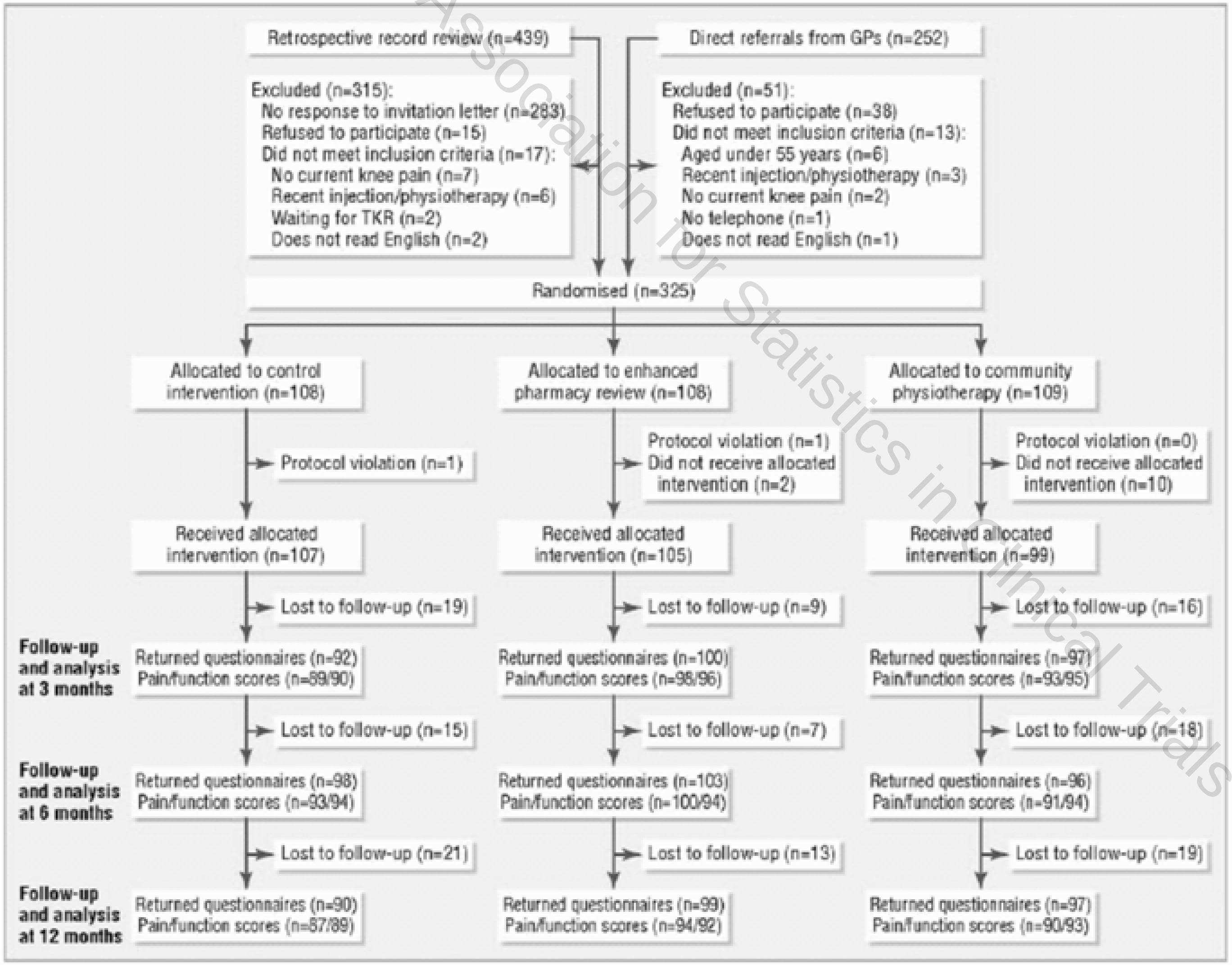




Avoid mental subtraction

New type of graphs

(1a) Clinical trial overview: Trial profile



Retrospective record review (n=439)

Direct referrals from GPs (n=252)

Excluded (n=315):
No response to invitation letter (n=283)
Refused to participate (n=15)
Did not meet inclusion criteria (n=17):
No current knee pain (n=7)
Recent injection/physiotherapy (n=6)
Waiting for TKR (n=2)
Does not read English (n=2)

Excluded (n=51):
Refused to participate (n=38)
Did not meet inclusion criteria (n=13):
Aged under 55 years (n=6)
Recent injection/physiotherapy (n=3)
No current knee pain (n=2)
No telephone (n=1)
Does not read English (n=1)

Randomised (n=325)

Allocated to control intervention (n=108)

Allocated to enhanced pharmacy review (n=108)

Allocated to community physiotherapy (n=109)

Protocol violation (n=1)

Protocol violation (n=1)
Did not receive allocated intervention (n=2)

Protocol violation (n=0)
Did not receive allocated intervention (n=10)

Received allocated intervention (n=107)

Received allocated intervention (n=105)

Received allocated intervention (n=99)

Lost to follow-up (n=19)

Lost to follow-up (n=9)

Lost to follow-up (n=16)

Follow-up and analysis at 3 months

Returned questionnaires (n=92)
Pain/function scores (n=89/90)

Returned questionnaires (n=100)
Pain/function scores (n=98/96)

Returned questionnaires (n=97)
Pain/function scores (n=93/95)

Lost to follow-up (n=15)

Lost to follow-up (n=7)

Lost to follow-up (n=18)

Follow-up and analysis at 6 months

Returned questionnaires (n=98)
Pain/function scores (n=93/94)

Returned questionnaires (n=103)
Pain/function scores (n=100/94)

Returned questionnaires (n=96)
Pain/function scores (n=91/94)

Lost to follow-up (n=21)

Lost to follow-up (n=13)

Lost to follow-up (n=19)

Follow-up and analysis at 12 months

Returned questionnaires (n=90)
Pain/function scores (n=87/89)

Returned questionnaires (n=99)
Pain/function scores (n=94/92)

Returned questionnaires (n=97)
Pain/function scores (n=90/93)

Too much text

Repetition

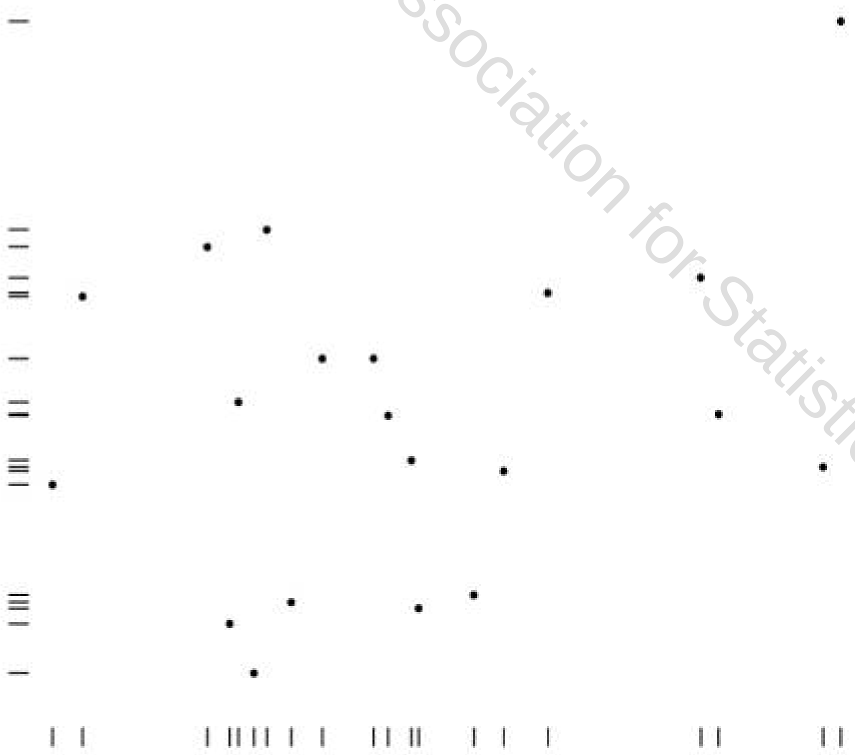
Do not use flow chart

Instead

Use a table

Graphs: that can be used

(8) Dash-dot-plot



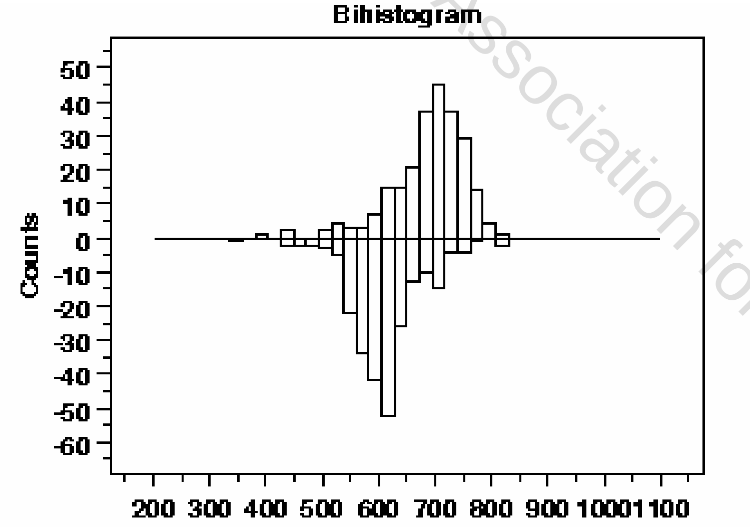
A type of scatter plot which lets you see the marginal distribution of each axis

Due to the scatter plot: marginal and joint distribution are displayed together

Source: Edward Tufte in 'The Visual Display of Quantitative Information' (Second Edition, Graphic Press, 2001 P.133).

Graphs: that can be used

(9) Bihistogram : graphical alternative to the two-sample t-test



Graphs Made easy using SAS/GRAPH SG procedure

Kanimozhi A

Overview

- What is SG procedure
- Syntax
- Statements
- Examples
- Traditional SAS/Graph Vs SG Procedure
- Pros and Cons of SG Procedure
- Summary

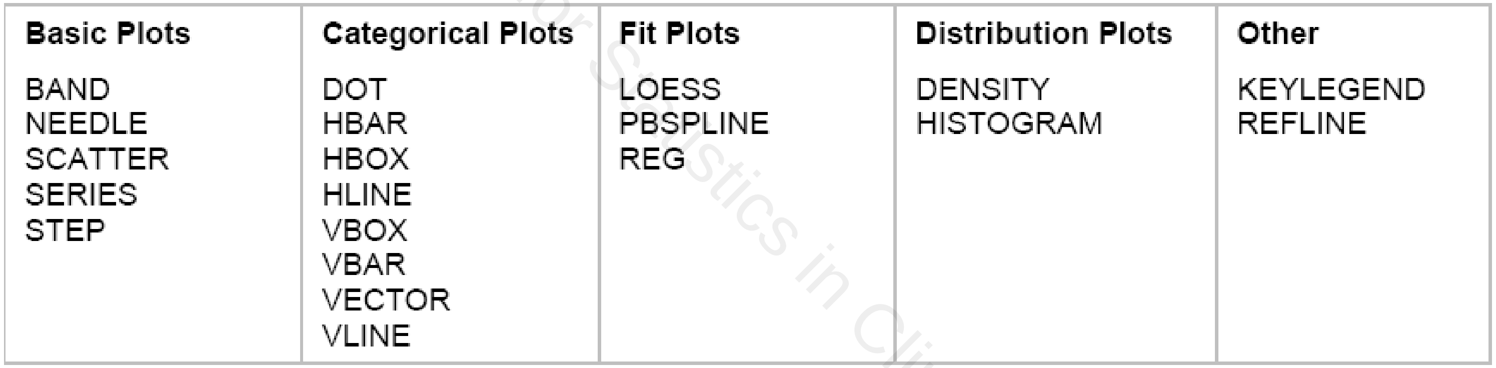
What is SG Proc

- Making a plot of a data is often the first step in data analysis or statistical analysis
- SAS 9.2 introduces the first installment of new family of procedures designed to create statistical graphics to assist in data analysis
- The names of the new procedures all begin with “SG” to differentiate them from traditional SAS/GRAPH procedure
- Are inbuilt on top of the ODS GRAPHICS system
- Facilitate to create graphs quickly and efficiently, with simple coding
- Can create effective and attractive graphics that can be as simple as scatter plots to paneled displays with classifications , all with the syntax clear and concise
- SG procedures includes **SGPLOT**, **SGPANEL** and **SGSCATTER**

SGPLOT

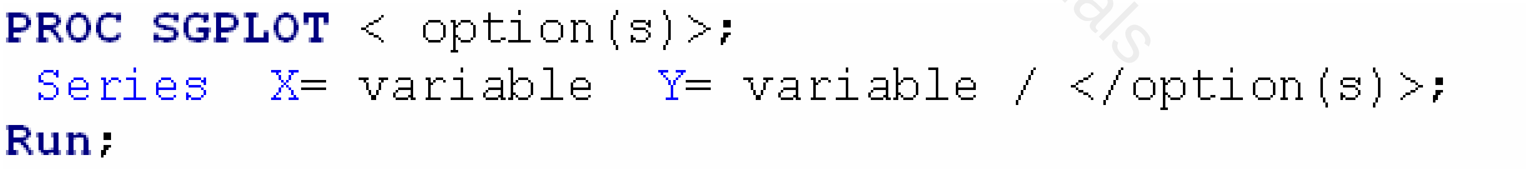
PROC SGPLOT is designed to create individual plots and charts with powerful overlaying capabilities

A variety of plot types are supported:



Basic Plots	Categorical Plots	Fit Plots	Distribution Plots	Other
BAND NEEDLE SCATTER SERIES STEP	DOT HBAR HBOX HLINE VBOX VBAR VECTOR VLINE	LOESS PBSPLINE REG	DENSITY HISTOGRAM	KEYLEGEND REFLINE

Syntax:

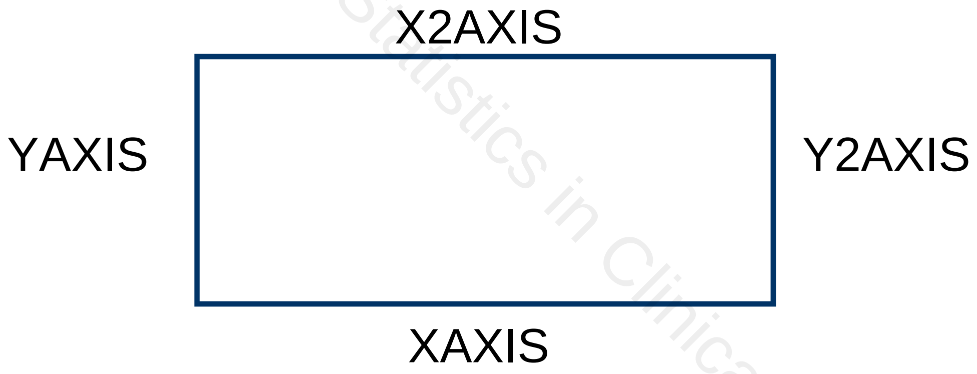


```
PROC SGPLOT < option(s)>;
 Series  X= variable  Y= variable / </option(s)>;
Run;
```

Plot Axes

The SGPLOT procedure contains statements that enables us to change the type and appearance of the axes:

XAXIS, X2AXIS, YAXIS, and Y2AXIS.



X2AXIS

YAXIS

Y2AXIS

XAXIS

By default, the type of each axis is determined by the types of plots that use the axis and the data that is applied to the axis.

Axis types

Discrete

Discrete is the default axis type for character data.

Linear

Linear is the default axis type for numeric data.

Logarithmic

The axis contains a logarithmic range of values. The logarithmic axis type is not used as a default.

Time

The axis contains a range of time values. Time is the default axis type for data that uses a SAS time, date, or datetime format.

Legends

It creates a legend automatically based on the plot statements and options that are specified

The automatic legend functionality can be overruled by defining legend with the KEYLEGEND statement or by specifying the NOAUTOLEGEND option

We can create customized legends by using one or more KEYLEGEND statements.

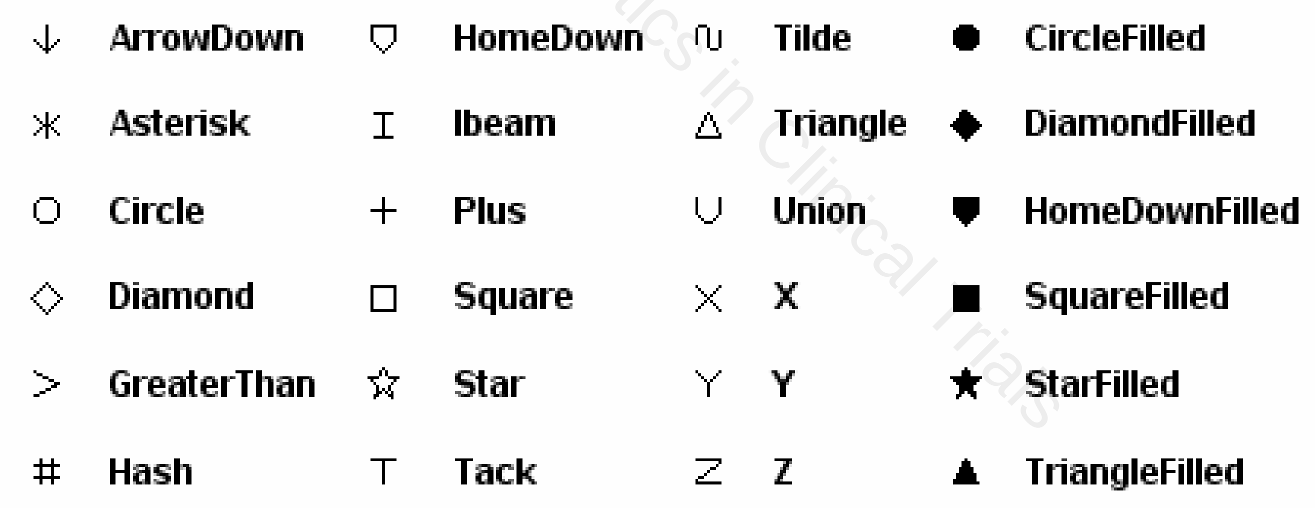
we can use the KEYLEGEND statement to control the contents, title, location, and border of the legend

Marker Symbols

The marker option can be used for automatic marker symbols

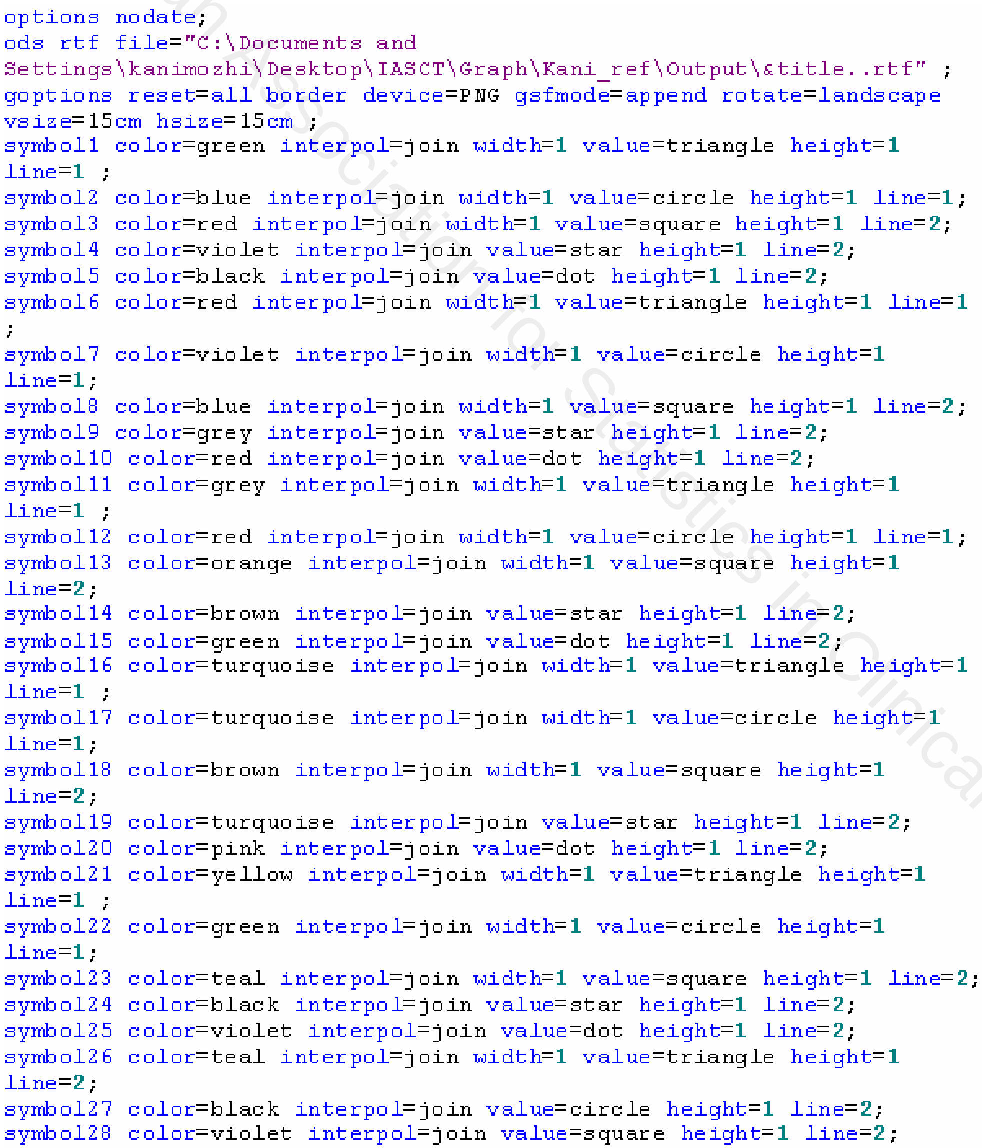
The MARKERATTRS= option on some of the plot statements enables to specify the marker symbol that is used to represent the data according to our wish.

List of Marker symbols



↓	ArrowDown	⛉	HomeDown	∿	Tilde	●	CircleFilled
⁎	Asterisk	I	Ibeam	△	Triangle	◆	DiamondFilled
○	Circle	+	Plus	∪	Union	⛊	HomeDownFilled
◇	Diamond	□	Square	×	X	■	SquareFilled
>	GreaterThan	☆	Star	Y	Y	★	StarFilled
#	Hash	⊤	Tack	Z	Z	▲	TriangleFilled

Example1 (Line chart from 9.1.3)

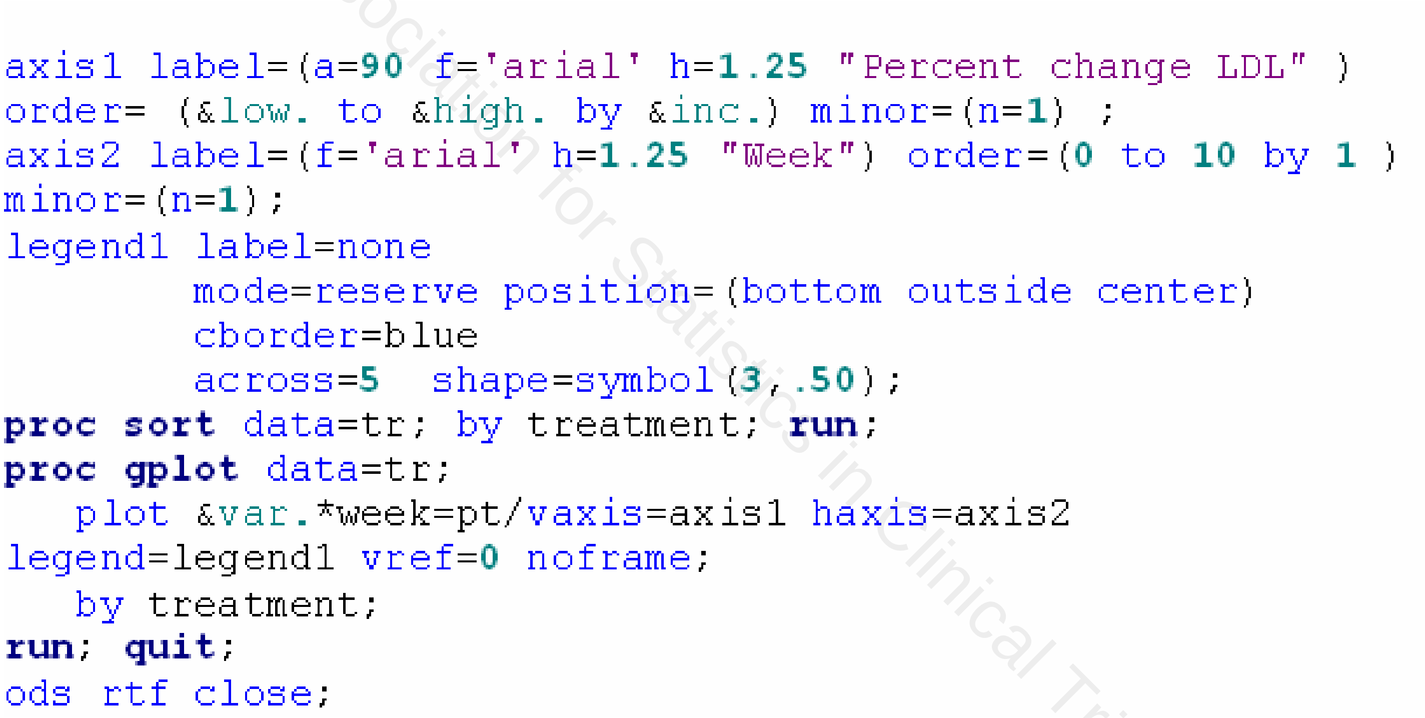


```
options nodate;
ods rtf file="C:\Documents and
Settings\kanimozhi\Desktop\IASCT\Graph\Kani_ref\Output\&title..rtf" ;
goptions reset=all border device=PNG gsfmode=append rotate=landscape
vsize=15cm hsize=15cm ;
symbol1 color=green interpol=join width=1 value=triangle height=1
line=1 ;
symbol2 color=blue interpol=join width=1 value=circle height=1 line=1;
symbol3 color=red interpol=join width=1 value=square height=1 line=2;
symbol4 color=violet interpol=join value=star height=1 line=2;
symbol5 color=black interpol=join value=dot height=1 line=2;
symbol6 color=red interpol=join width=1 value=triangle height=1 line=1
;
symbol7 color=violet interpol=join width=1 value=circle height=1
line=1;
symbol8 color=blue interpol=join width=1 value=square height=1 line=2;
symbol9 color=grey interpol=join value=star height=1 line=2;
symbol10 color=red interpol=join value=dot height=1 line=2;
symbol11 color=grey interpol=join width=1 value=triangle height=1
line=1 ;
symbol12 color=red interpol=join width=1 value=circle height=1 line=1;
symbol13 color=orange interpol=join width=1 value=square height=1
line=2;
symbol14 color=brown interpol=join value=star height=1 line=2;
symbol15 color=green interpol=join value=dot height=1 line=2;
symbol16 color=turquoise interpol=join width=1 value=triangle height=1
line=1 ;
symbol17 color=turquoise interpol=join width=1 value=circle height=1
line=1;
symbol18 color=brown interpol=join width=1 value=square height=1
line=2;
symbol19 color=turquoise interpol=join value=star height=1 line=2;
symbol20 color=pink interpol=join value=dot height=1 line=2;
symbol21 color=yellow interpol=join width=1 value=triangle height=1
line=1 ;
symbol22 color=green interpol=join width=1 value=circle height=1
line=1;
symbol23 color=teal interpol=join width=1 value=square height=1 line=2;
symbol24 color=black interpol=join value=star height=1 line=2;
symbol25 color=violet interpol=join value=dot height=1 line=2;
symbol26 color=teal interpol=join width=1 value=triangle height=1
line=2;
symbol27 color=black interpol=join value=circle height=1 line=2;
symbol28 color=violet interpol=join value=square height=1 line=2;
```

Contd.

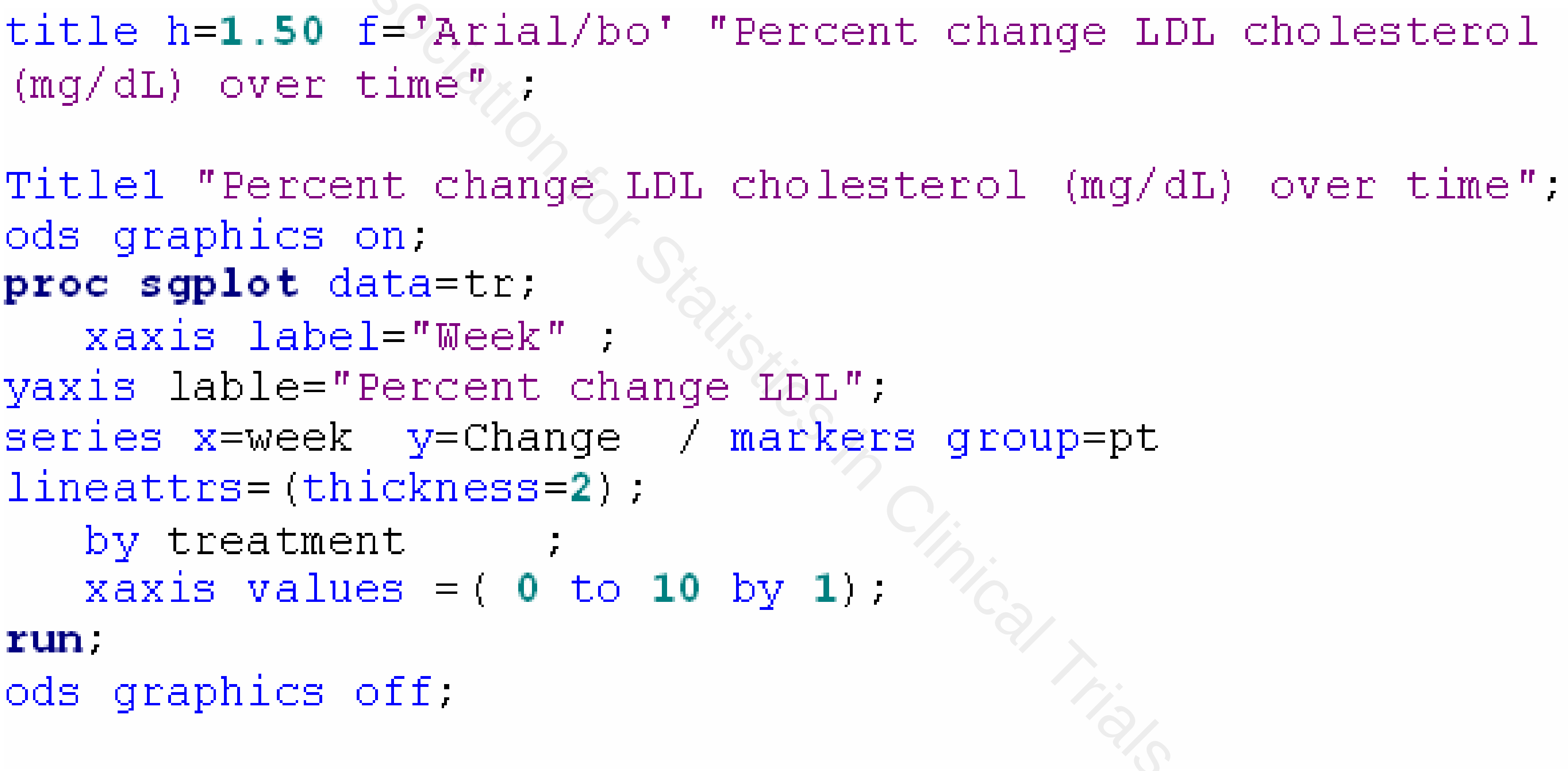
Example1 (Line chart from 9.1.3)

Contd.



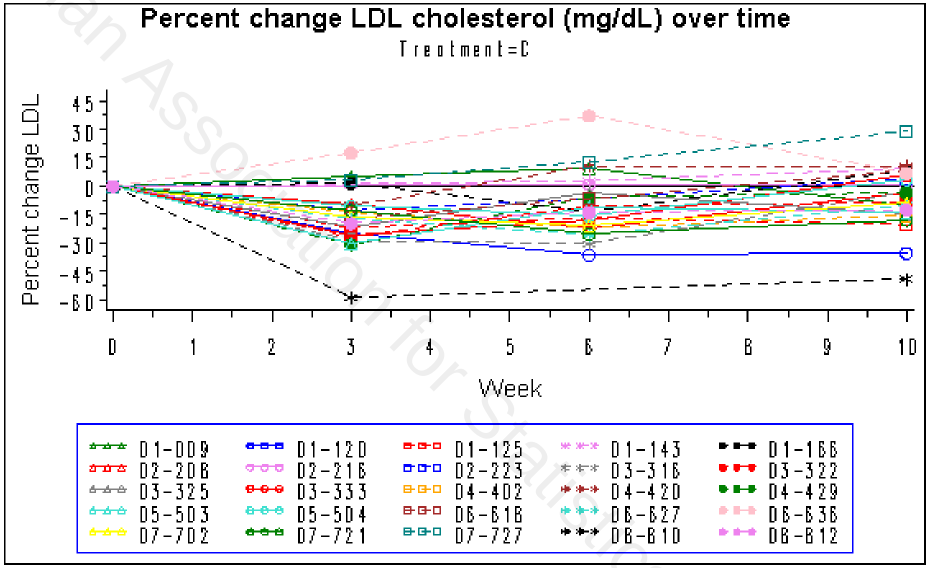
```
axis1 label=(a=90 f='arial' h=1.25 "Percent change LDL" )
order= (&low. to &high. by &inc.) minor=(n=1) ;
axis2 label=(f='arial' h=1.25 "Week") order=(0 to 10 by 1 )
minor=(n=1);
legend1 label=none
        mode=reserve position=(bottom outside center)
        cborder=blue
        across=5  shape=symbol(3,.50);
proc sort data=tr; by treatment; run;
proc gplot data=tr;
   plot &var.*week=pt/vaxis=axis1 haxis=axis2
legend=legend1 vref=0 noframe;
   by treatment;
run; quit;
ods rtf close;
```

Line chart from 9.2

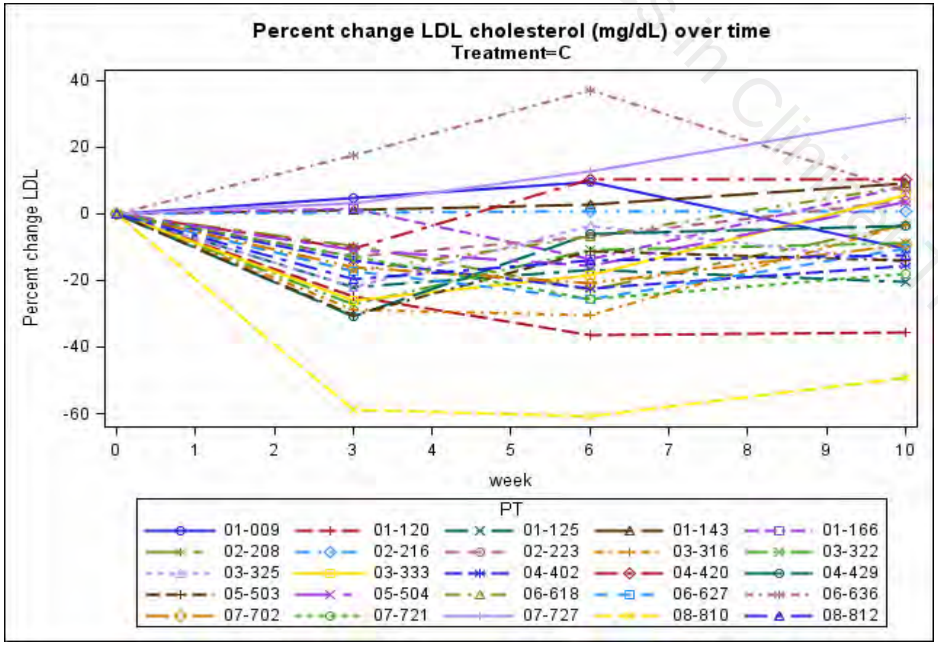


```
title h=1.50 f='Arial/bo' "Percent change LDL cholesterol
(mg/dL) over time" ;

Title1 "Percent change LDL cholesterol (mg/dL) over time";
ods graphics on;
proc sgplot data=tr;
   xaxis label="Week" ;
yaxis lable="Percent change LDL";
series x=week  y=Change  / markers group=pt
lineattrs=(thickness=2);
   by treatment      ;
   xaxis values =( 0 to 10 by 1);
run;
ods graphics off;
```

OUTPUTS



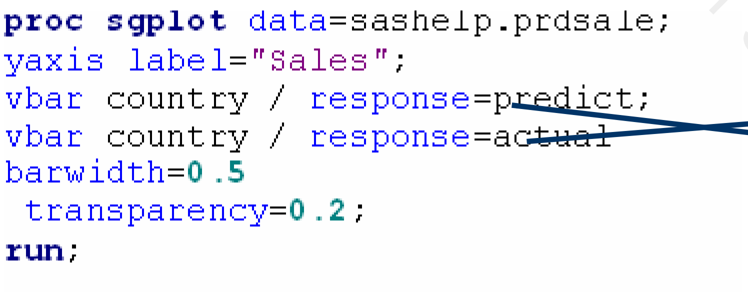
SAS 9.1.3



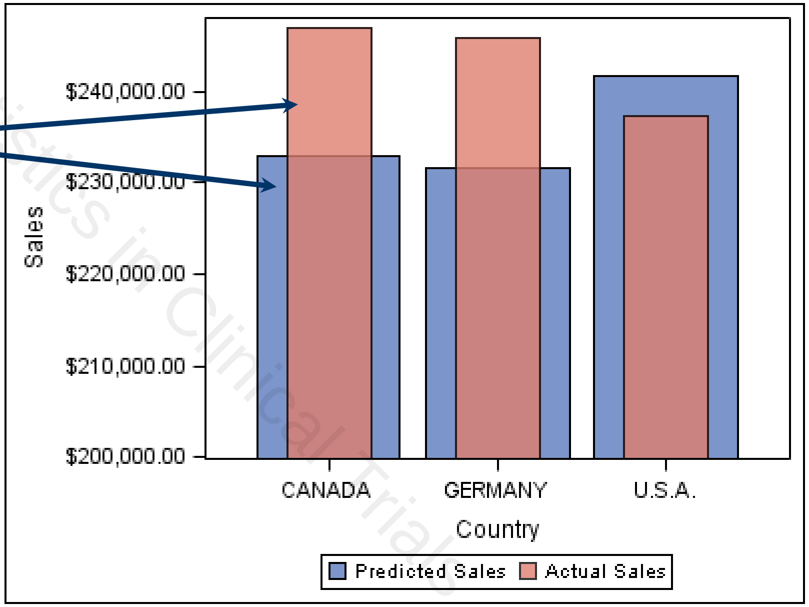
SAS 9.2

Example2

The following code creates a graph with two bar charts:



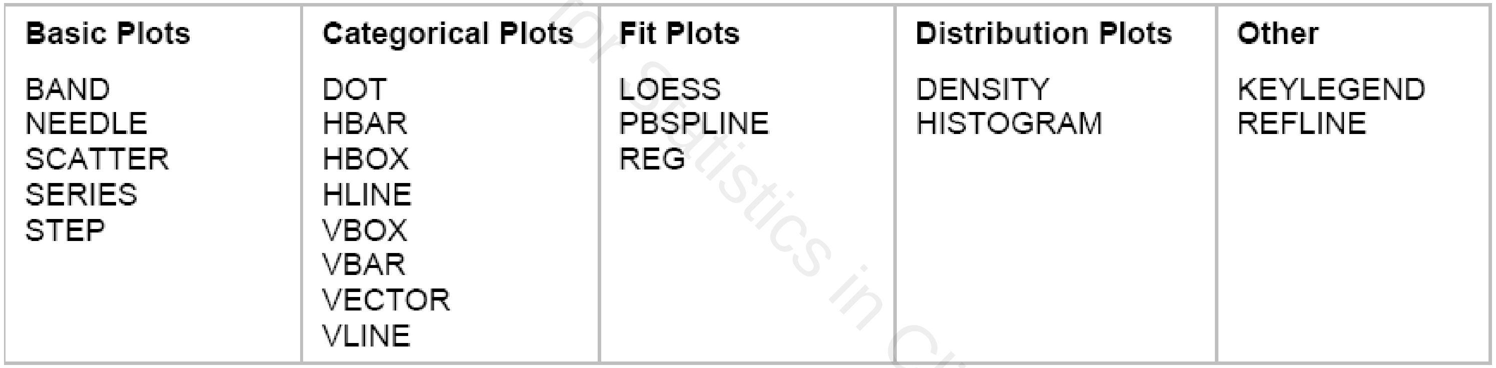
```
proc sgplot data=sashelp.prdsale;
yaxis label="Sales";
vbar country / response=predict;
vbar country / response=actual
barwidth=0.5
 transparency=0.2;
run;
```



SGPANEL

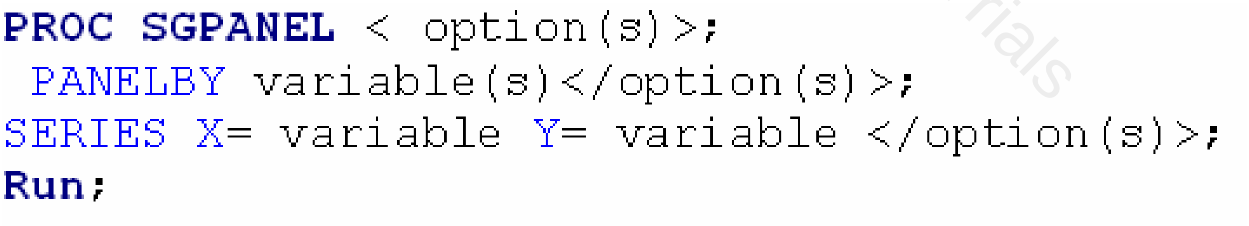
Is designed to produce the paneled graphs based on classification variables.

A variety of plot types are supported:



Basic Plots	Categorical Plots	Fit Plots	Distribution Plots	Other
BAND NEEDLE SCATTER SERIES STEP	DOT HBAR HBOX HLINE VBOX VBAR VECTOR VLINE	LOESS PBSPLINE REG	DENSITY HISTOGRAM	KEYLEGEND REFLINE

Syntax:



```
PROC SGPANEL < option(s)>;
 PANELBY variable(s)</option(s)>;
SERIES X= variable Y= variable </option(s)>;
Run;
```

Plot Axes

It contains two statements that enable us to change the type and appearance for the axes of the graph cells in the panel:

COLAXIS and ROWAXIS.

By default, the type of each axis is determined by the types of plots that use the axis and the data that is applied to the axis.

- The axis types are same as SGPLOT:

Discrete , Linear , Logarithmic and Time

- The legend and the marker remains as same in SGPLOT

Panelby Statement

- It is the key statement in SGPanel PROCEDURE
- Two different Layout styles can be considered on Panelby statement

  1. Panel and 2. Lattice

- The default layout style is PANEL.

  1. We can specify any number of classifier variables.

  2. The graph cells in the panel are arranged automatically, and the classifier values are displayed above each graph cell in the panel.

- The Lattice layout style requires exactly two classifier variables.

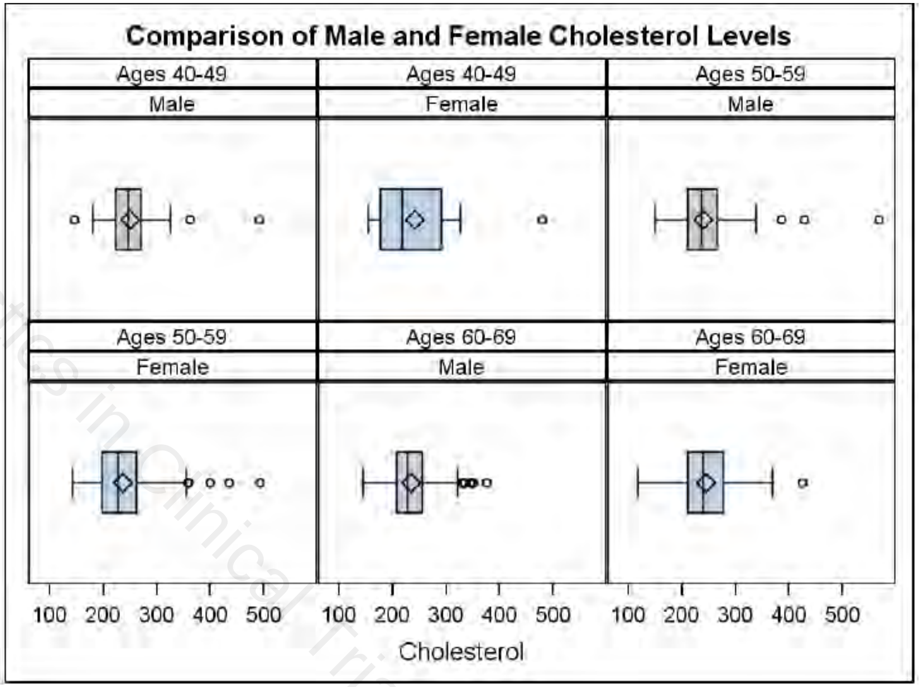
  1. The values of the first variable are assigned as columns, and the values of the second variable are assigned as rows.

  2. The classifier values are displayed above the columns and to the right side of the rows.

Example

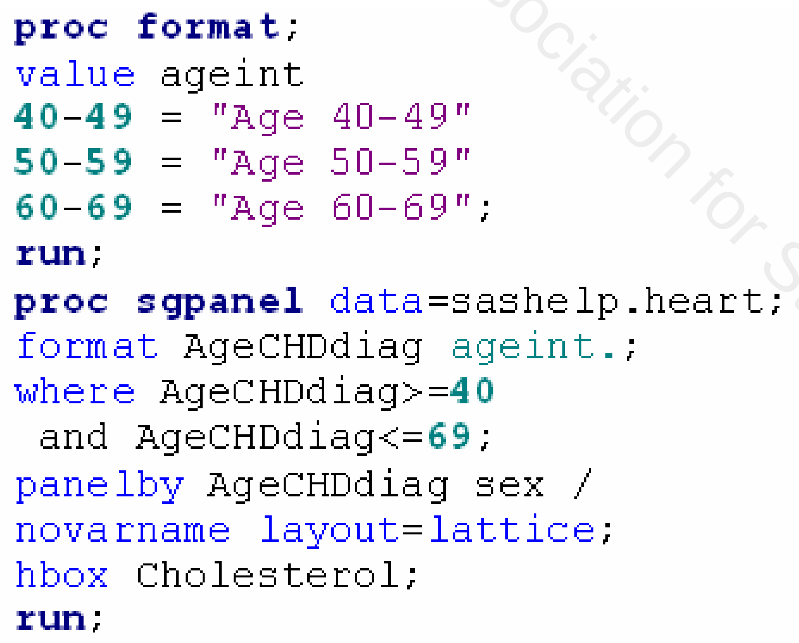
We need to compare the cholesterol levels between males and females by age who have been diagnosed with coronary heart disease in a heart study

```
proc format;
value ageint
40-49 = "Age 40-49"
50-59 = "Age 50-59"
60-69 = "Age 60-69";
run;

proc sgpanel data=sashelp.heart;
format AgeCHDdiag ageint.;
where AgeCHDdiag>=40
 and AgeCHDdiag<=69;
panelby AgeCHDdiag sex
/novarname;
hbox Cholesterol;
run;
```

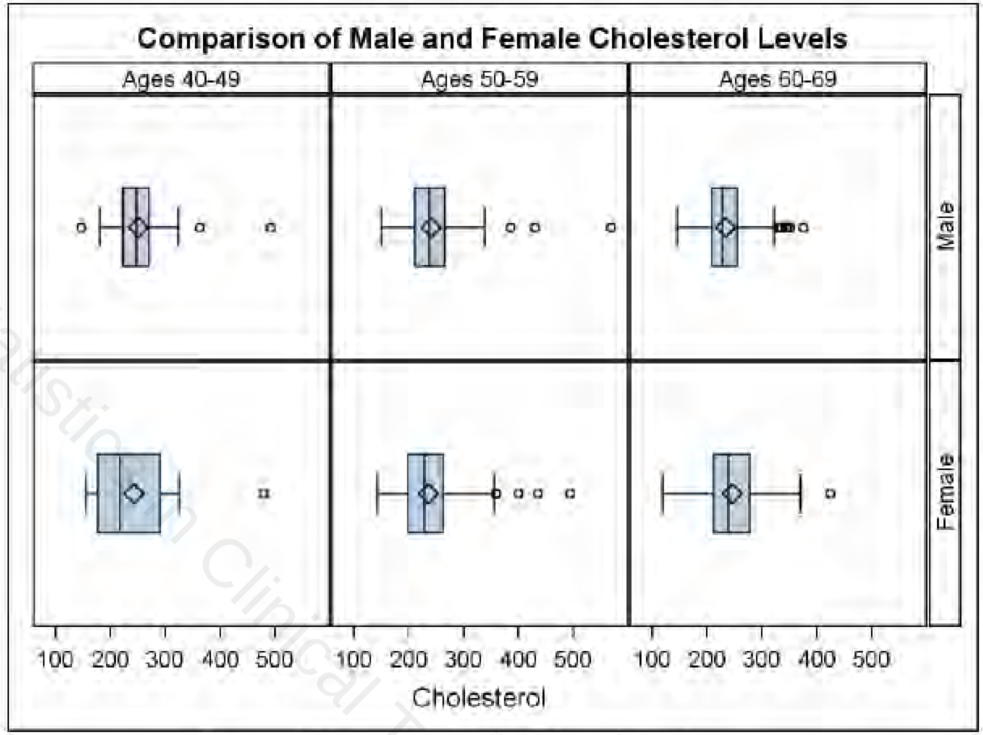


Better display is to use lattice layout instead of Panel

Example



```
proc format;
value ageint
40-49 = "Age 40-49"
50-59 = "Age 50-59"
60-69 = "Age 60-69";
run;
proc sgpanel data=sashelp.heart;
format AgeCHDdiag ageint.;
where AgeCHDdiag>=40
 and AgeCHDdiag<=69;
panelby AgeCHDdiag sex /
novarname layout=lattice;
hbox Cholesterol;
run;
```



The first panelby variable is used as column value and the second one is used as a row value

SGSCATTER

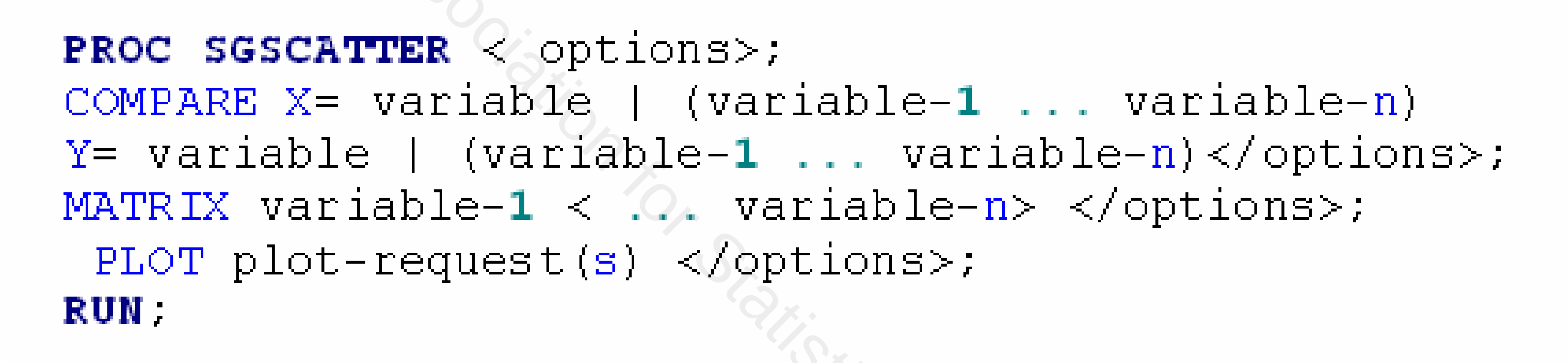
It is designed to create panels of scatter plots and scatter plot matrices

It contains three statements that can be used to create a paneled graph of scatter plots:

- PLOT
- COMPARE
- MATRIX

Each of the statements are specialized for creating different types of paneled graphs.

SGSCATTER SYNTAX



```
PROC SGSCATTER < options>;
COMPARE X= variable | (variable-1 ... variable-n)
Y= variable | (variable-1 ... variable-n)</options>;
MATRIX variable-1 < ... variable-n> </options>;
 PLOT plot-request(s) </options>;
RUN;
```

Plot Statement

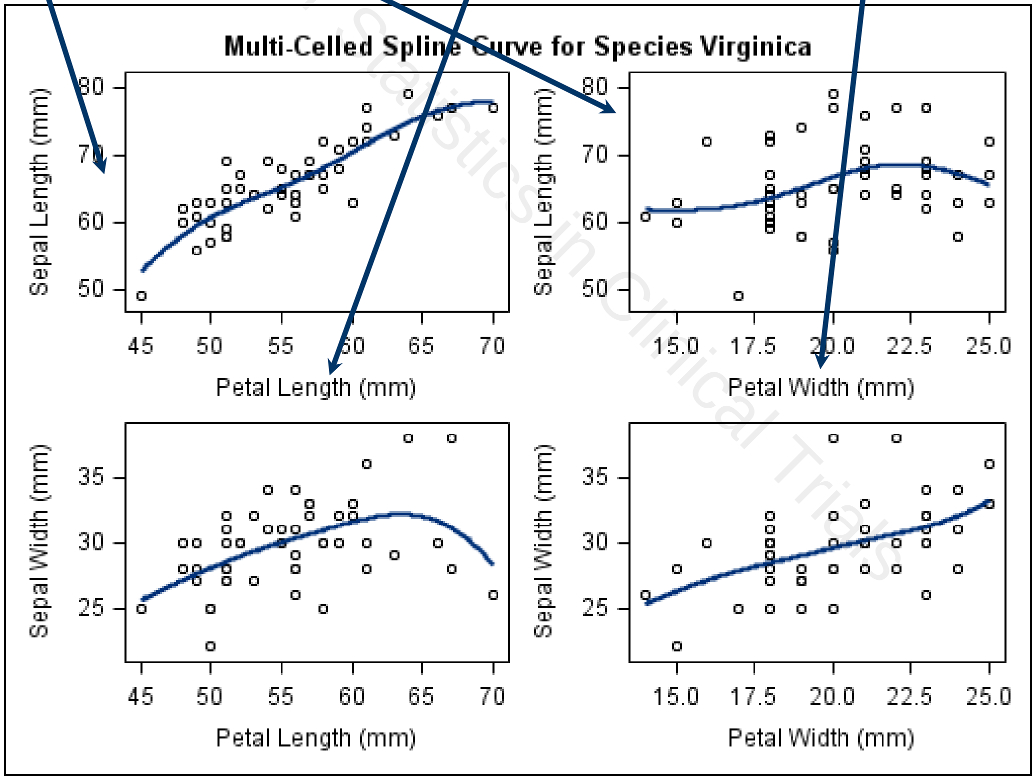
- It is best used when there is a relationship between the variables that we want to plot , but the data ranges are different.
- The method of specifying the Y*X pairs can be any of the following form:

Y_0*X0 ……Y_n*X_n , Y*(X_0…X_n) , (Y_0…..Y_n)*X and (Y_0….Y_n)*(X_0….X_n)

- Each variable pair that specified in the PLOT statement creates an independent graph cell.
- we can also overlay fit plots and ellipses on each cell by using options.
- By default, the axis ranges of each cell are independent from the other cells. However, we can use the UNISCALE= option to specify that all of the cells use the same axis ranges for the X axis, the Y axis, or both axes.
- It is possible to create a single scatter cell with the PLOT statement, but the SGPLOT procedure is better suited to creating a single-celled graph.

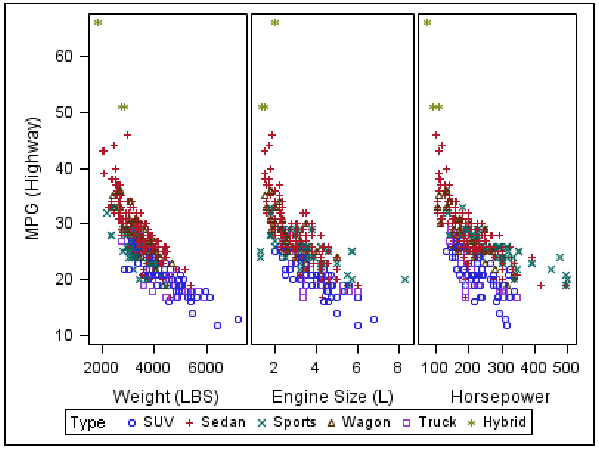
Example

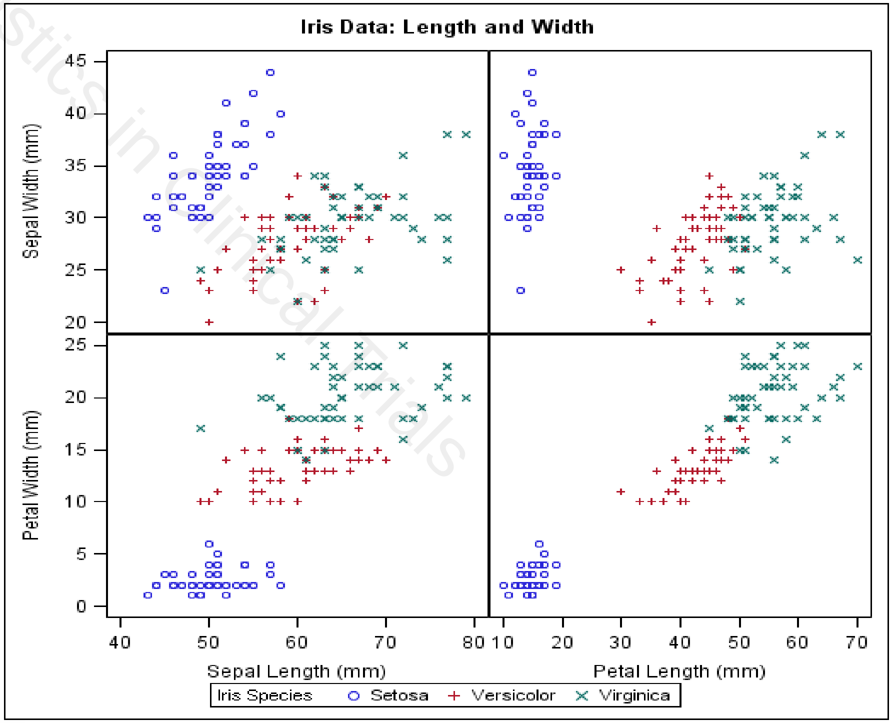
```
proc sgscatter
data=sashelp.iris(where=(species="Virginica"));
  title "Multi-Celled Spline Curve for Species Virginica";
  plot (sepallength sepalwidth)*(petallength petalwidth)
       / pbspline;
run;
```

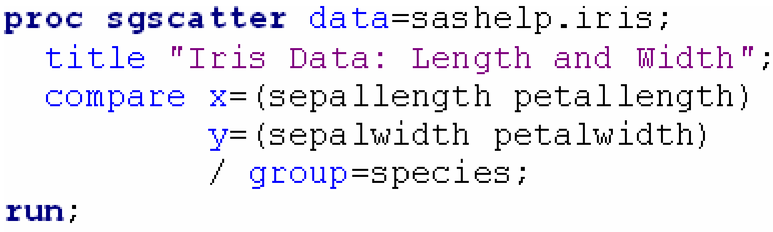


COMPARE Statement

- It is used to create a shared axis panel, also called an MxN matrix.
- The list of X and Y variables are crossed to create each cell in the graph.
- All cells in a row share the same row axis range.
- All cells in a column share the same column axis range.
- we can add fit plots and confidence ellipses to each cell in the panel by using options.
- can also be used to do simple X or Y axis sharing by specifying only one X or Y variable.





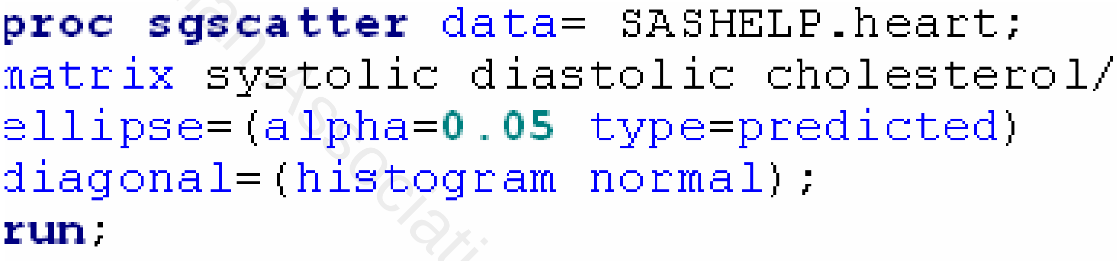


```
proc sgscatter data=sashelp.iris;
  title "Iris Data: Length and Width";
  compare x=(sepallength petallength)
          y=(sepalwidth petalwidth)
          / group=species;
run;
```

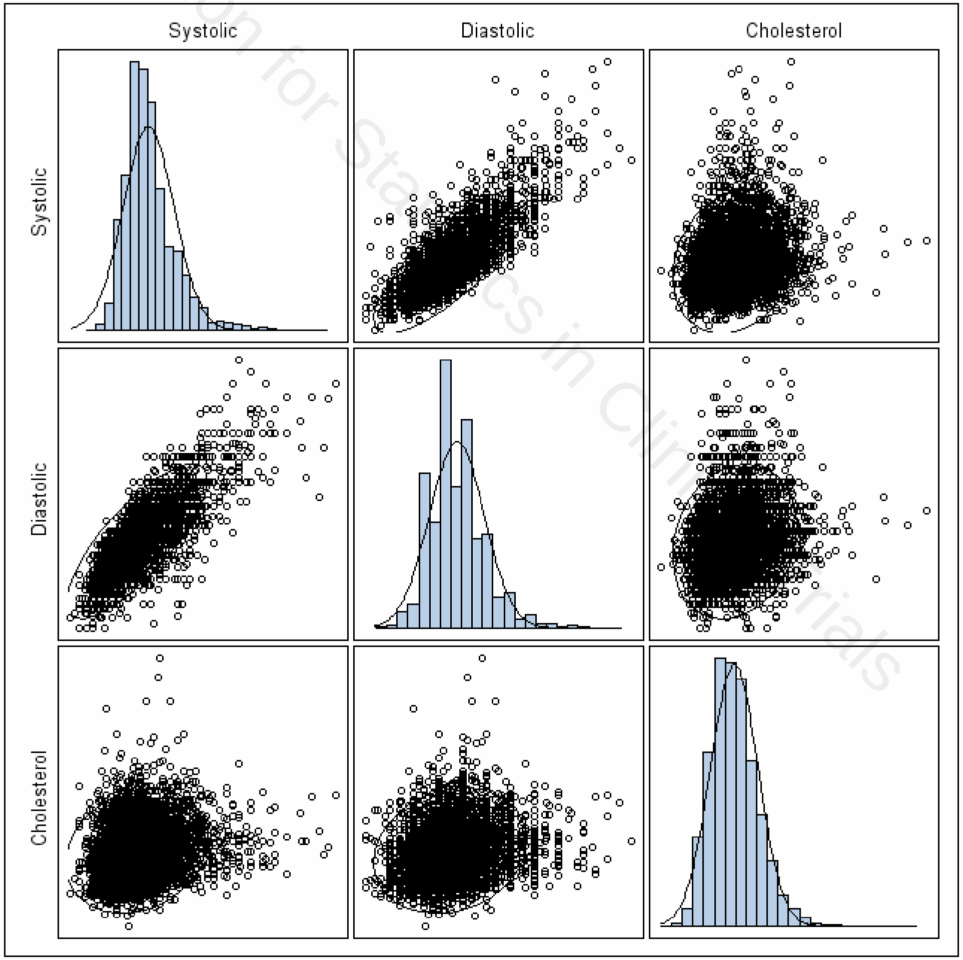
MATRIX Statement

- It is used to create scatter plot matrices of a list of variables
- It can be used for finding possible trends or correlations in different pairs
- The list of variables specified in on the statement is crossed to create an N*N matrix
- It also supports computed ellipses and a DIAGONAL option for adding plots in the diagonal

Example



```
proc sgscatter data= SASHELP.heart;
matrix systolic diastolic cholesterol/
ellipse=(alpha=0.05 type=predicted)
diagonal=(histogram normal);
run;
```



9.1.3 vs. 9.2

SAS/GRAPH 9.1.3	SAS/GRAPH 9.2
Global statements like: Goptions, AXIS, LEGEND , PATTERN, NOTE are used	All these attributes are derived either from the active ODS style or from the syntax with in the procedure
TITLE , FOOTNOTE, FOMAT and LABEL are used	TITLE , FOOTNOTE, FOMAT and LABEL are used. Justify option: justify two strings in the same location in the statement , the append instead of moving to the next line.
For some graphs, the plot type is determined by global options. For example, the INTERPOLATION= option on the SYMBOL statement might determine whether a graph is a scatter plot or a box plot.	The plot type is determined by the plot statement only.
Transparency is not supported.	can specify the degree of transparency for many graphics elements

9.1.3 vs. 9.2

SAS/GRAPH 9.1.3	SAS/GRAPH 9.2
Scaling of fonts and markers is not supported.	Scaling of fonts and markers is on by default. This means that the sizes of fonts and markers are adjusted as appropriate to the size of your graph. You can disable scaling by using the NOSCALE option on the ODS GRAPHICS statement.
the NOTE statement or Annotate is typically used to insert additional information, such as statistics, directly into a graph	information can be added using the procedure's INSET statement.

Pros

- Less coding
- Consistent appearance for reporting, generation of publication ready graphs in color, black and white.
- Statistical styling

because these procedures use the ODS style for default graph appearance attributes, it not only reduces the coding effect, but it also eliminates the need for determining the color and the attributes.

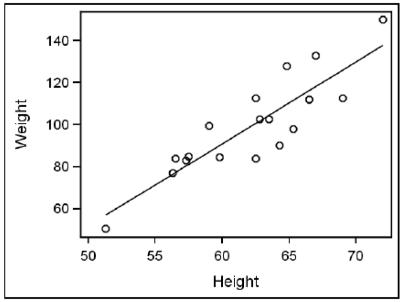


Figure 4a: Default styling for SERIES line

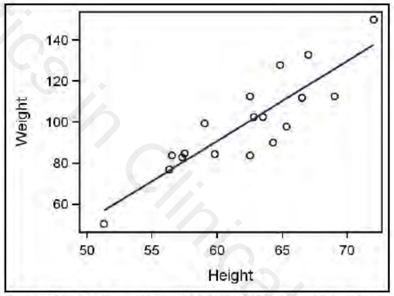


Figure 4b: SERIES line using the GraphFit style element

- Image Quality

the ODS GRAPHICS system allows to create the high resolution graphics without having to adjust any features in the graph

Cons

- It dose not replace traditional SAS/GRAPH ,for few Graphs we need to use 9.1.3 example: Counter plot
- SAS Help does not have clear cut examples for better understanding
- Yet another language to learn

Summary

- Facilitate to create graphs quickly and efficiently, with simple coding.
- SGPLOT helps to create individual plots and charts with powerful overlaying capabilities.
- SGPANEL can be used when we need to compare the values between two or more groups.
- SGSCATTER can be used when there is a relationship or trend between the variables that we want to plot, but the data ranges are different.

Conclusion

The concept behind the SG Procedures are simple in theory, yet powerful in execution

References

- SUGI Papers **http://www.lexjansen.com/**

Thank you

GlaxoSmithKline

Improving Graphics Using SAS/GRAPH Annotate Facility

Deepak Sriramulu

03 Apr 2009

IASCT

Introduction

- Have you ever created a graph with SAS/GRAPH and really liked it… except for one little thing?

- Often when creating graphs using SAS, you find one little part that you wish you could change or add that would make your output perfect

- The Annotate facility acts as a bridge between the procedure selected by the user and the user’s desire to customize the graphics output

- This presentation covers the concepts of the Annotate facility, followed by some examples that will be very useful for producing customized graphics.

Annotation Steps

A good annotation strategy begins with questions like:

1) *What* part of the graphics area will be used?

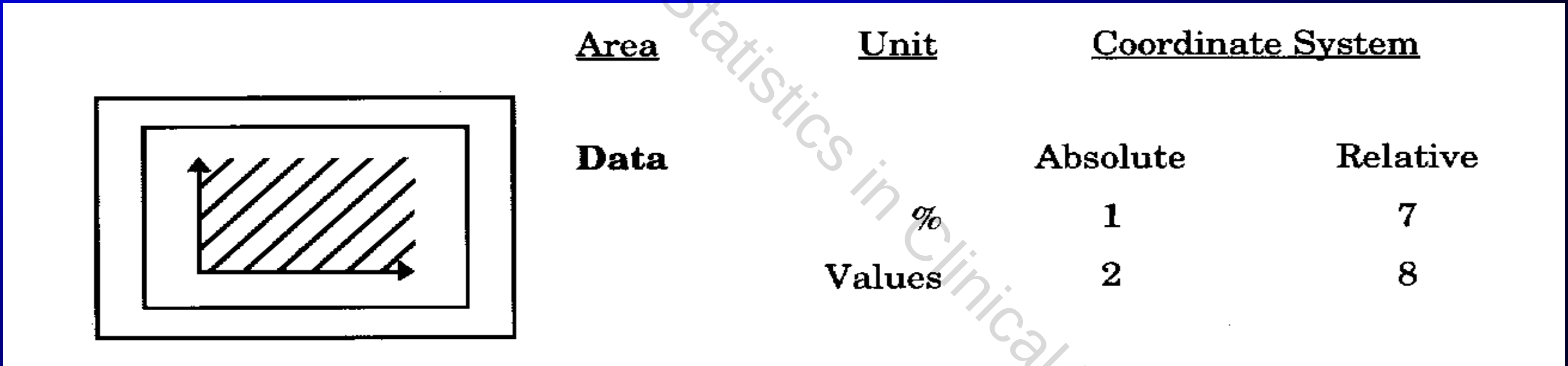
2) *Where* will the annotation element be put?

3) *What* should be done?

(4) *How* should this be done?

What part of Graphics area

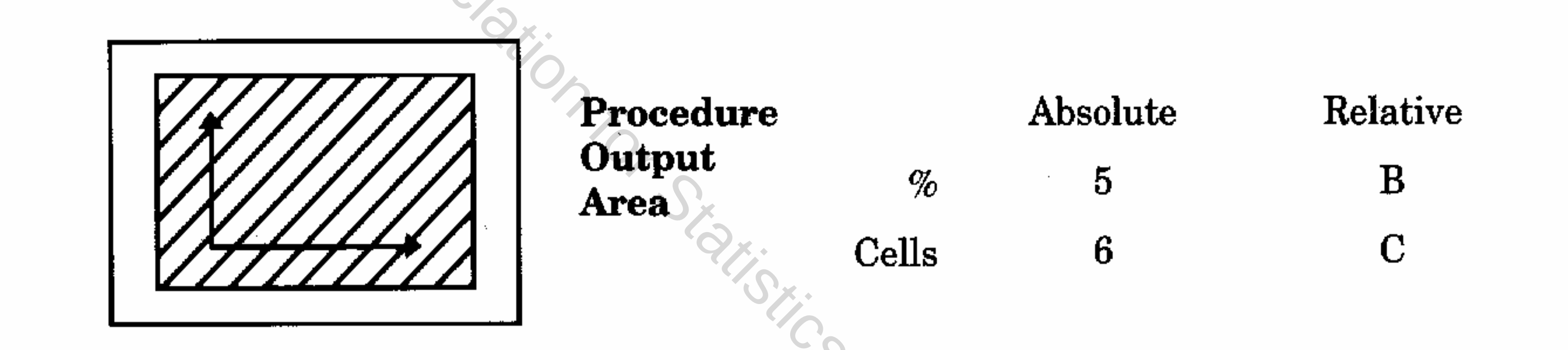
1) Data Area which represents only the space within the graph axes



Area	Unit	Coordinate System	
Data		Absolute	Relative
	%	1	7
	Values	2	8

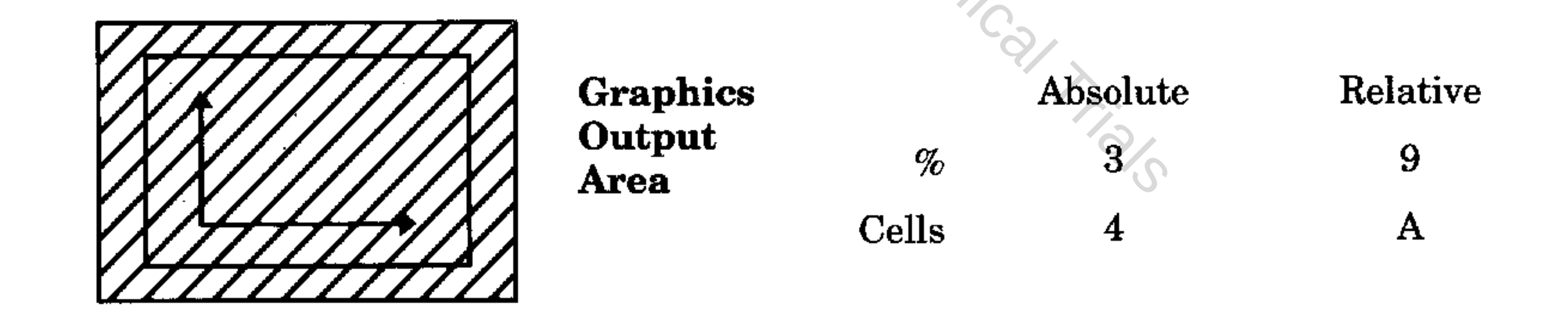
What part of Graphics area

2) Procedure Output Area or the area taken up by the graphic object



Procedure Output Area		Absolute	Relative
	%	5	B
	Cells	6	C

3) Graphics Output Area which is the entire writable page of output

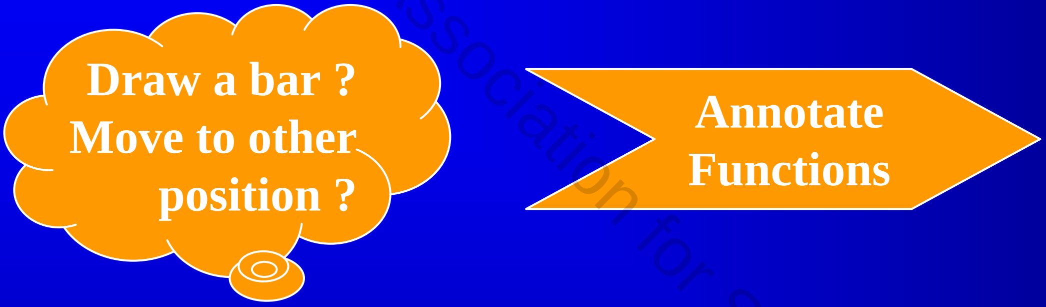


Graphics Output Area		Absolute	Relative
	%	3	9
	Cells	4	A

Where will the annotation element be put

X	The numeric horizontal coordinate.
Y	The numeric vertical coordinate.
Z	For three-dimensional graphs specifies the coordinate for the 3rd dimension.
HSYS	The type of units for the size (height) variable.
XSYS	The coordinate system for the X variable.
YSYS	The coordinate system for the Y variable.
ZSYS	The coordinate system for the Z variable (for three-dimensional graphs).

What should be done ?



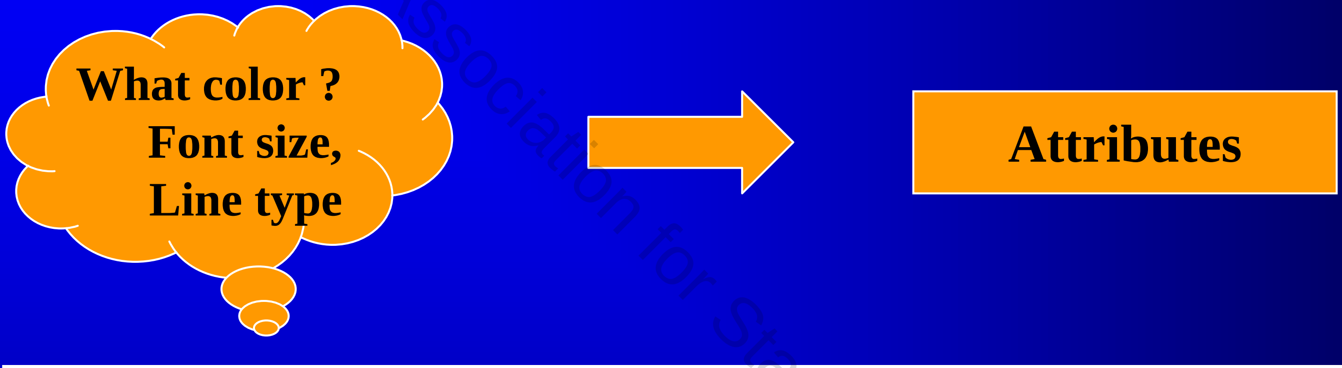
Specifies the Annotate drawing action.

LABEL	Adds a text
MOVE	Moves to a specific point
DRAW	Draws a line from the current position to a specified position

What should be done

POLY	Specifies the starting point of a polygon
POLYCONT	Continues drawing the polygon
BAR	Draws a rectangle from the current position to a specified position
SYMBOL	Draws a symbol.
PIE	Draws a pie slice, circle or arc.
MOVE	Move to the new x,y coordinates

How should be done



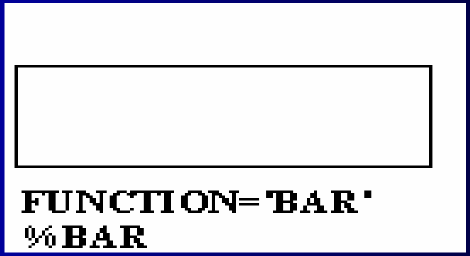
What color ?
Font size,
Line type

Attributes

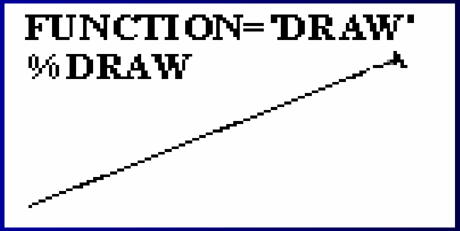
COLOR	Color of graphics item.
LINE	Line type of graphics item.
SIZE	Size of the graphics item. Specific to the function. For example size is the height of the character for a label function.
STYLE	Font/pattern of a graphics item.
TEXT	Text to use in a label symbol or comment.

Annotate macro actions

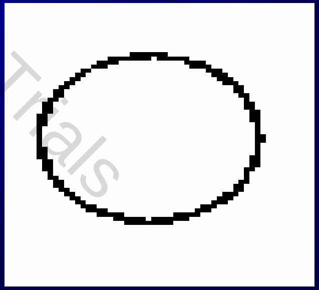
%BAR(x1, y1, x2, y2, color, line, style);



%DRAW(x, y, color, line, size);

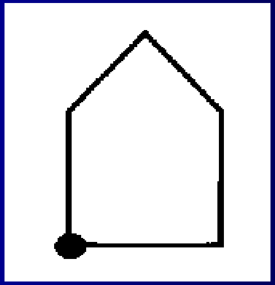


%CIRCLE(x, y, size, color);

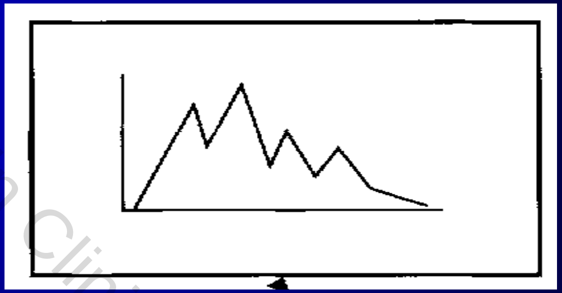


Annotate macro actions

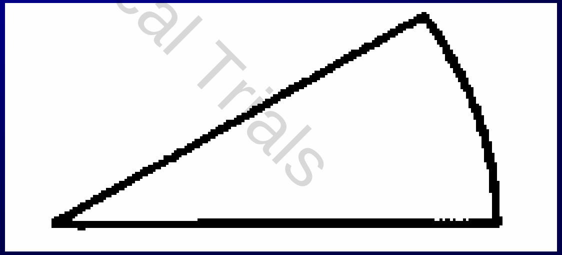
%POLY(x, y, color, style, line);



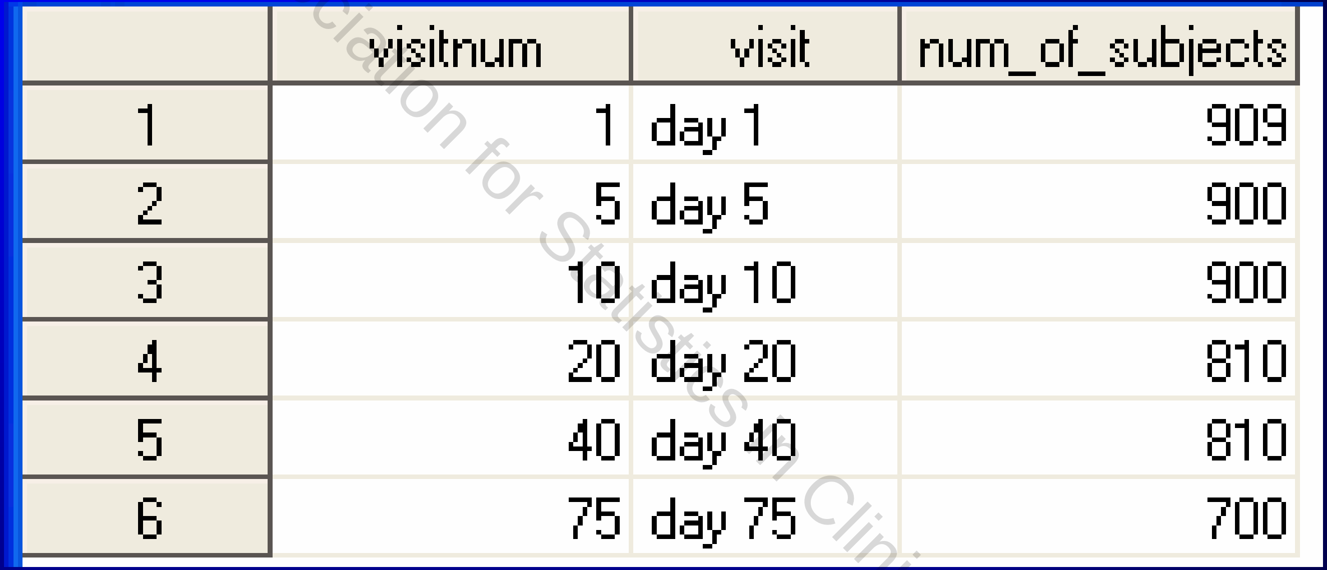
%FRAME(color, line, size, style);



%SLICE(x1, y1,angle, rotate, size, color, style, line);

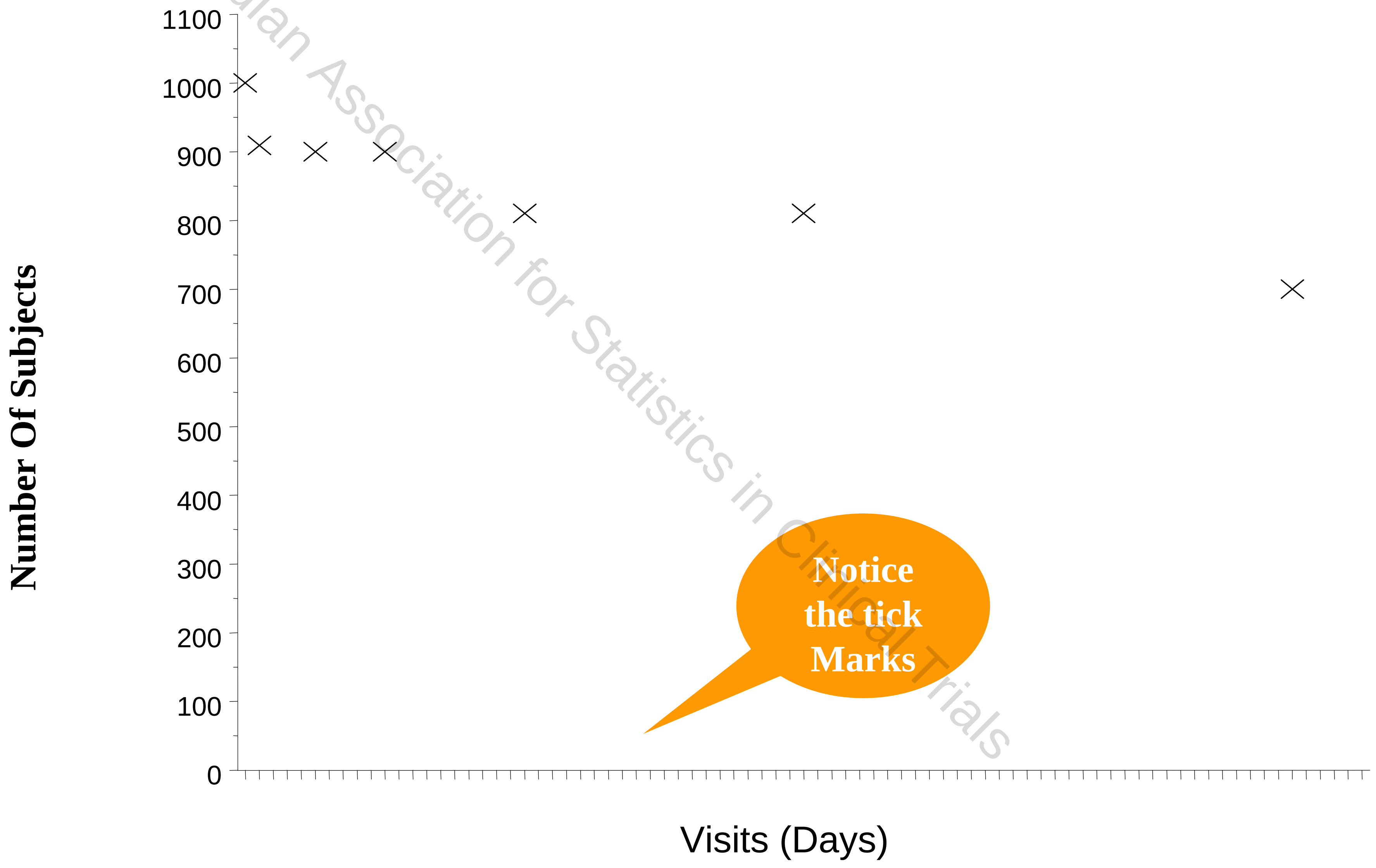


A look at our sample data for the graphs



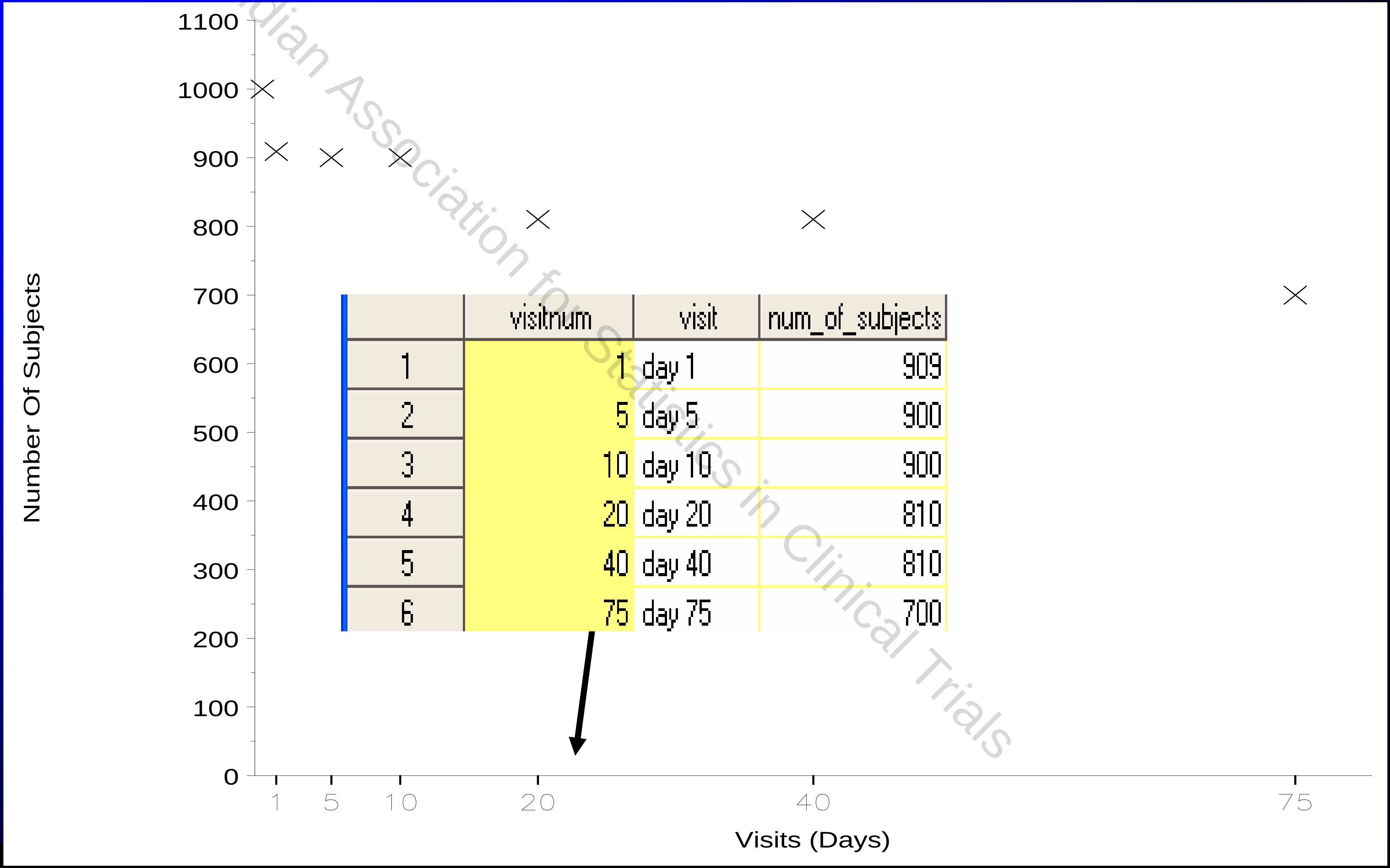
	visitnum	visit	num_of_subjects
1	1	day 1	909
2	5	day 5	900
3	10	day 10	900
4	20	day 20	810
5	40	day 40	810
6	75	day 75	700

Plot of No. of Subjects Vs Visits



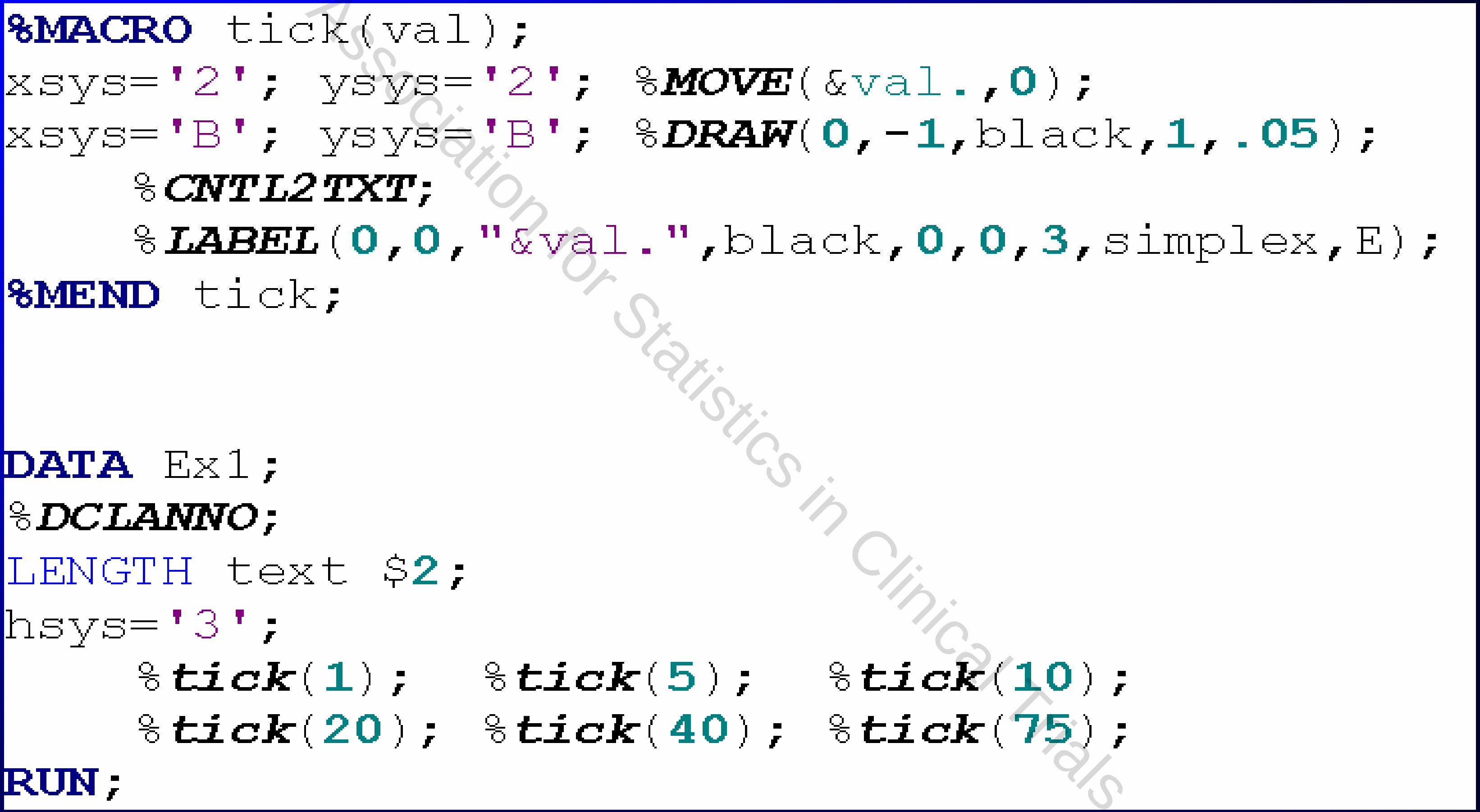
Notice the tick Marks

Example1: Relabeling Axis



	visitnum	visit	num_of_subjects
1	1	day 1	909
2	5	day 5	900
3	10	day 10	900
4	20	day 20	810
5	40	day 40	810
6	75	day 75	700

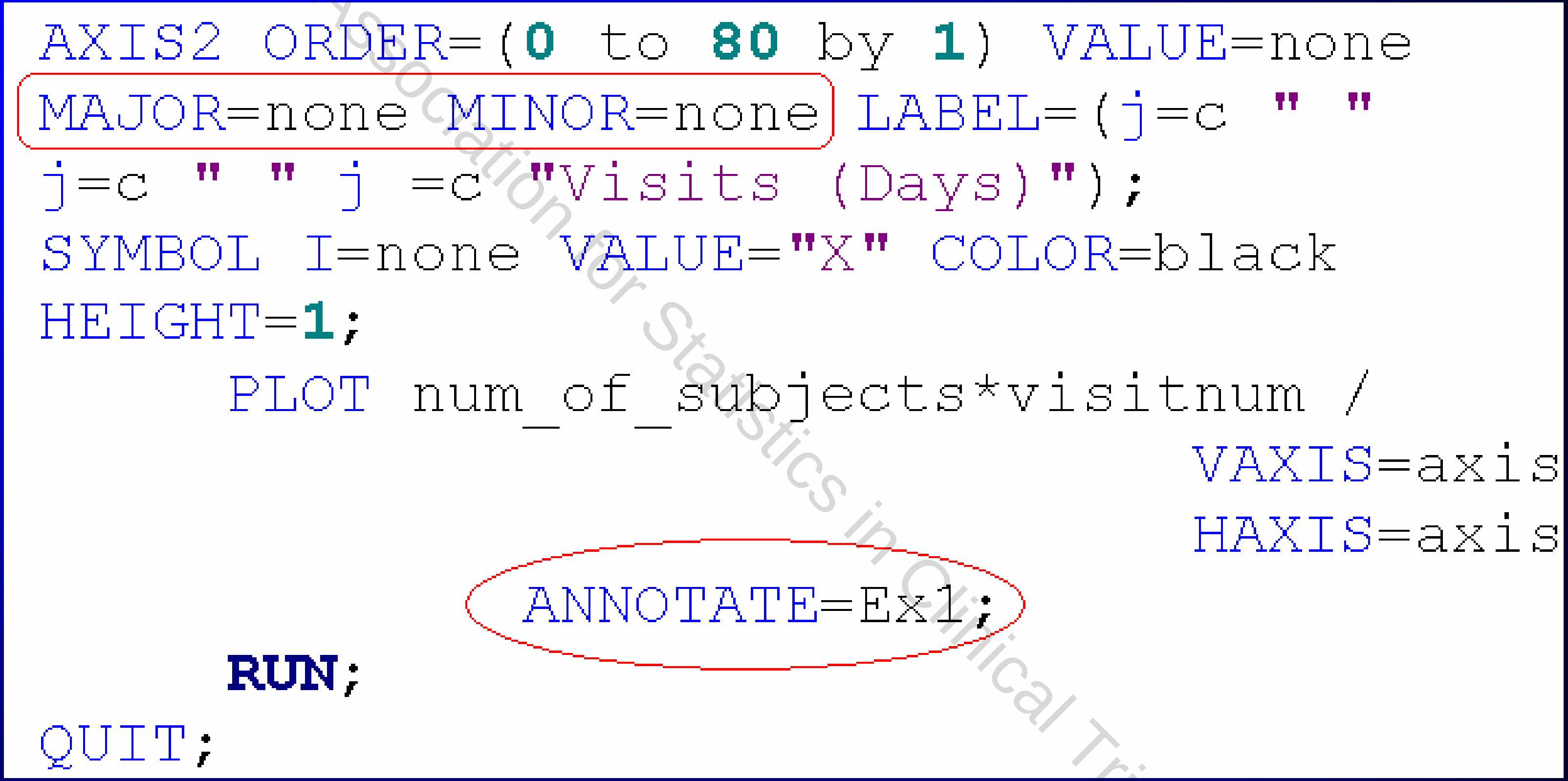
Example 1 : Relabeling axis



```
%MACRO tick(val);
xsys='2'; ysys='2'; %MOVE(&val.,0);
xsys='B'; ysys='B'; %DRAW(0,-1,black,1,.05);
    %CNTL2TXT;
    %LABEL(0,0,"&val.",black,0,0,3,simplex,E);
%MEND tick;


DATA Ex1;
%DCLANNO;
LENGTH text $2;
hsys='3';
    %tick(1);  %tick(5);  %tick(10);
    %tick(20); %tick(40); %tick(75);
RUN;
```

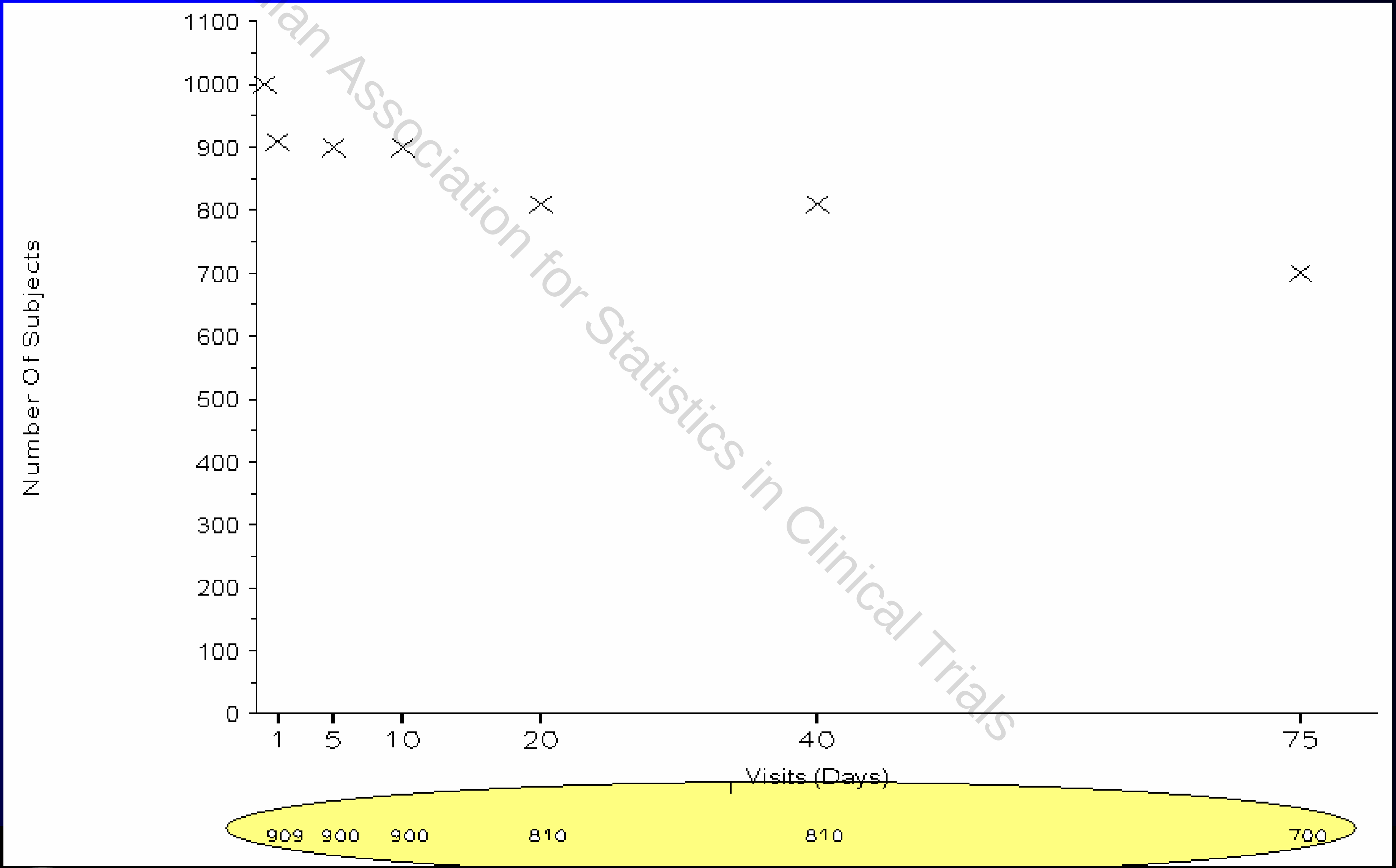
GlaxoSmithKline

Example1: Relabeling axis

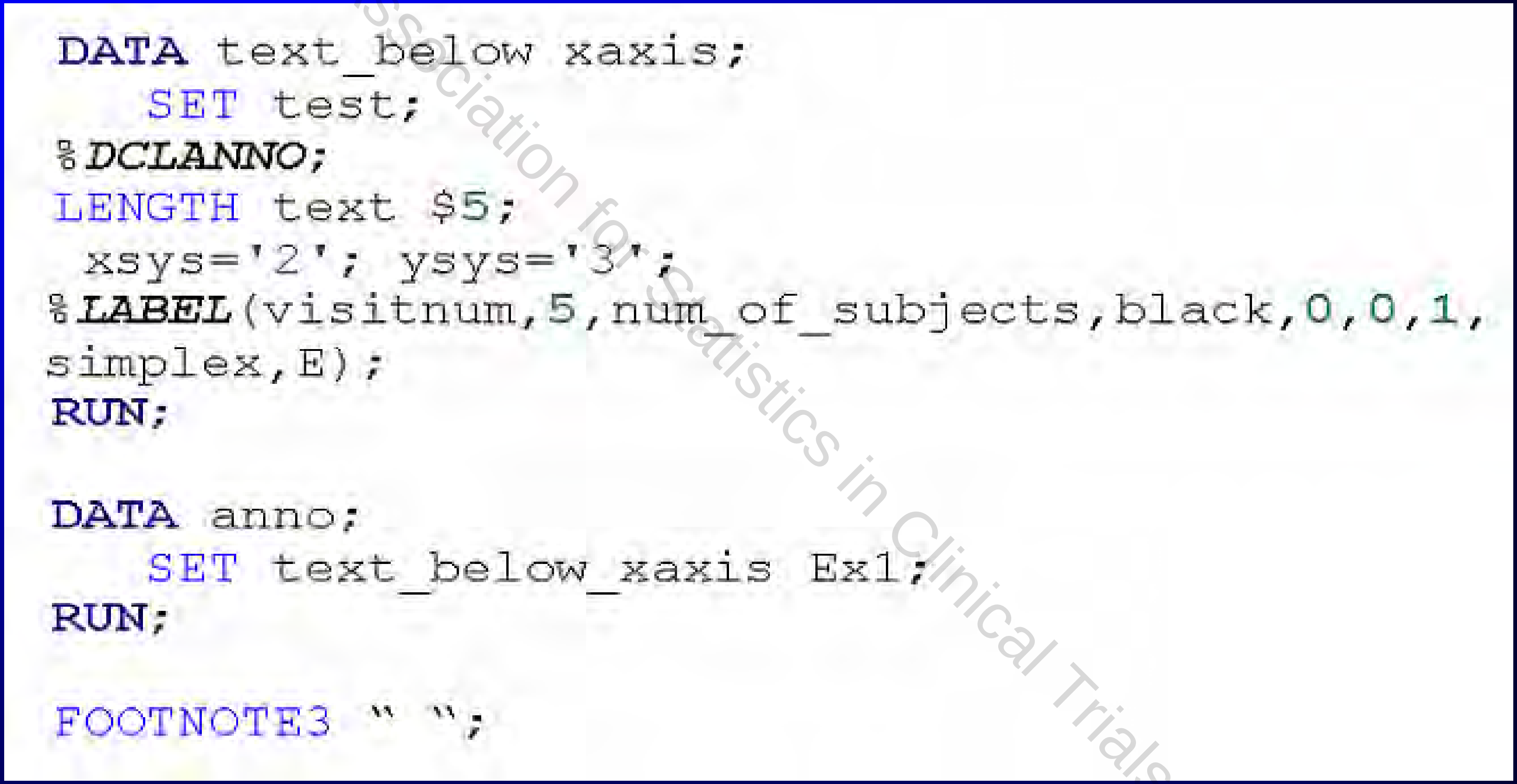


```
AXIS2 ORDER=(0 to 80 by 1) VALUE=none
MAJOR=none MINOR=none LABEL=(j=c " "
j=c " " j =c "Visits (Days)");
SYMBOL I=none VALUE="X" COLOR=black
HEIGHT=1;
     PLOT num_of_subjects*visitnum /
                          VAXIS=axis
                          HAXIS=axis
            ANNOTATE=Ex1;
     RUN;
QUIT;
```

Example 2: Display text below X-axis

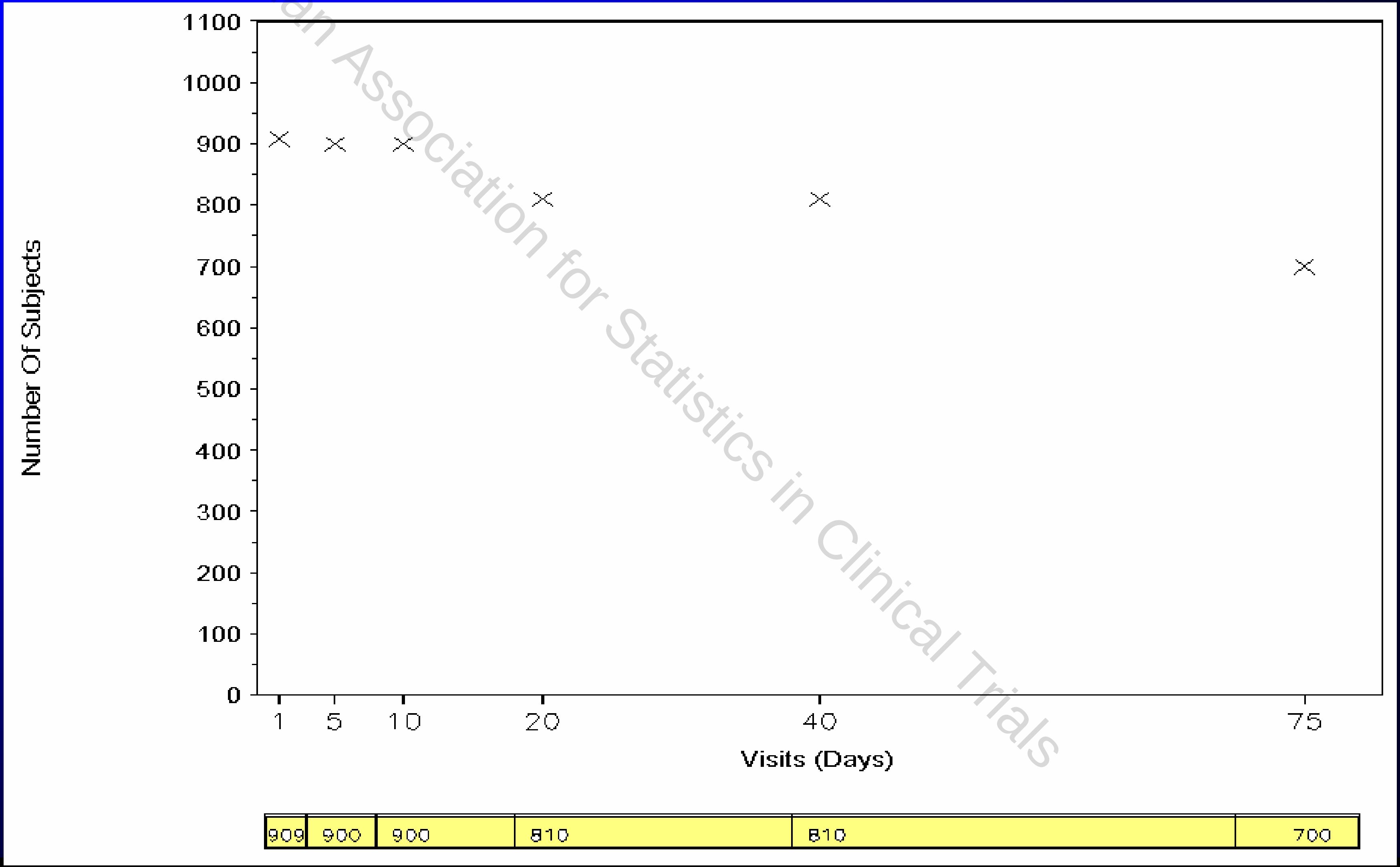


Example2: Display text below X-axis

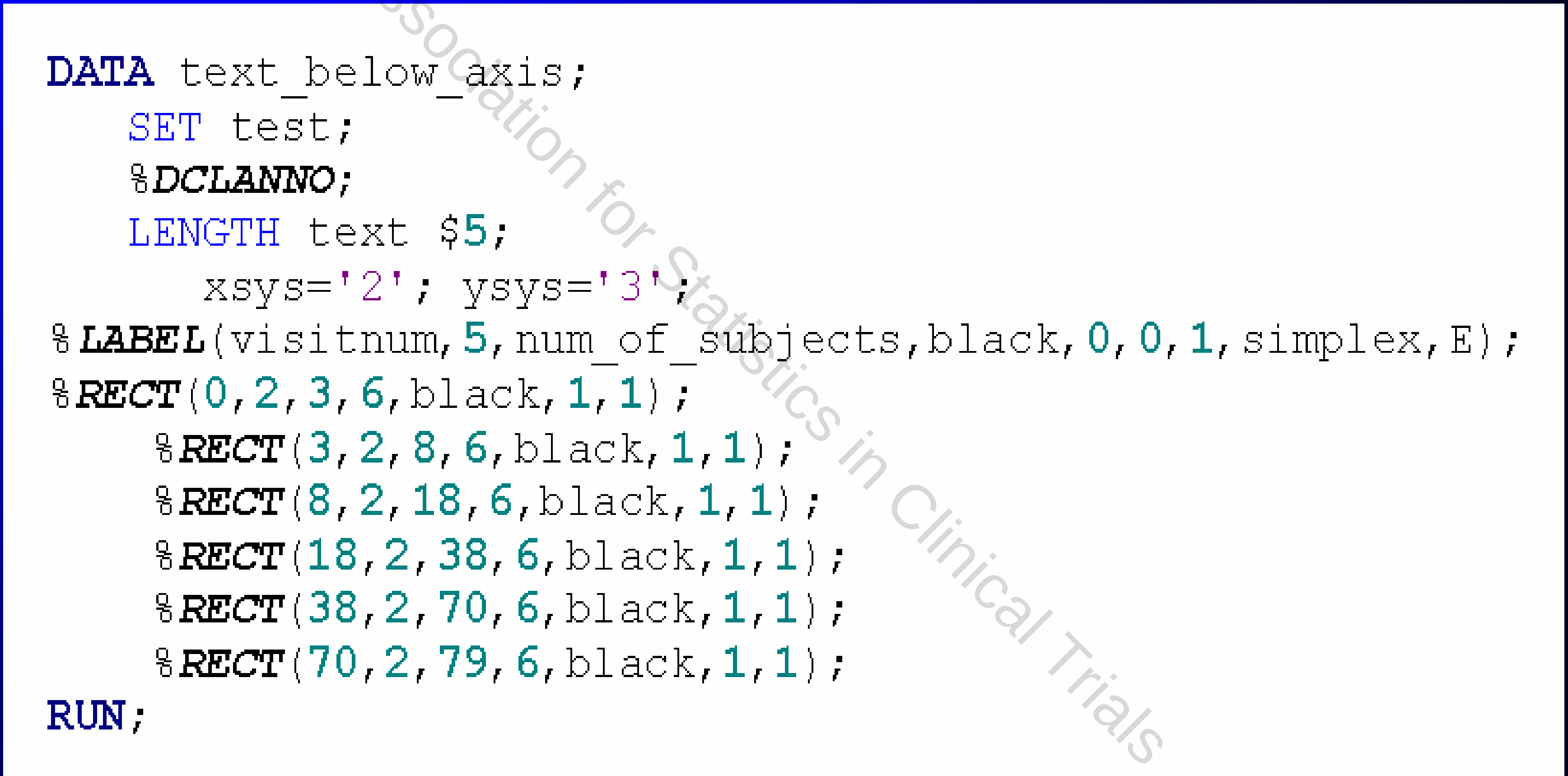


```
DATA text_below xaxis;
    SET test;
%DCLANNO;
LENGTH text $5;
 xsys='2'; ysys='3';
%LABEL(visitnum,5,num_of_subjects,black,0,0,1,
simplex,E);
RUN;

DATA anno;
    SET text_below_xaxis Ex1;
RUN;

FOOTNOTE3 “ “;
```

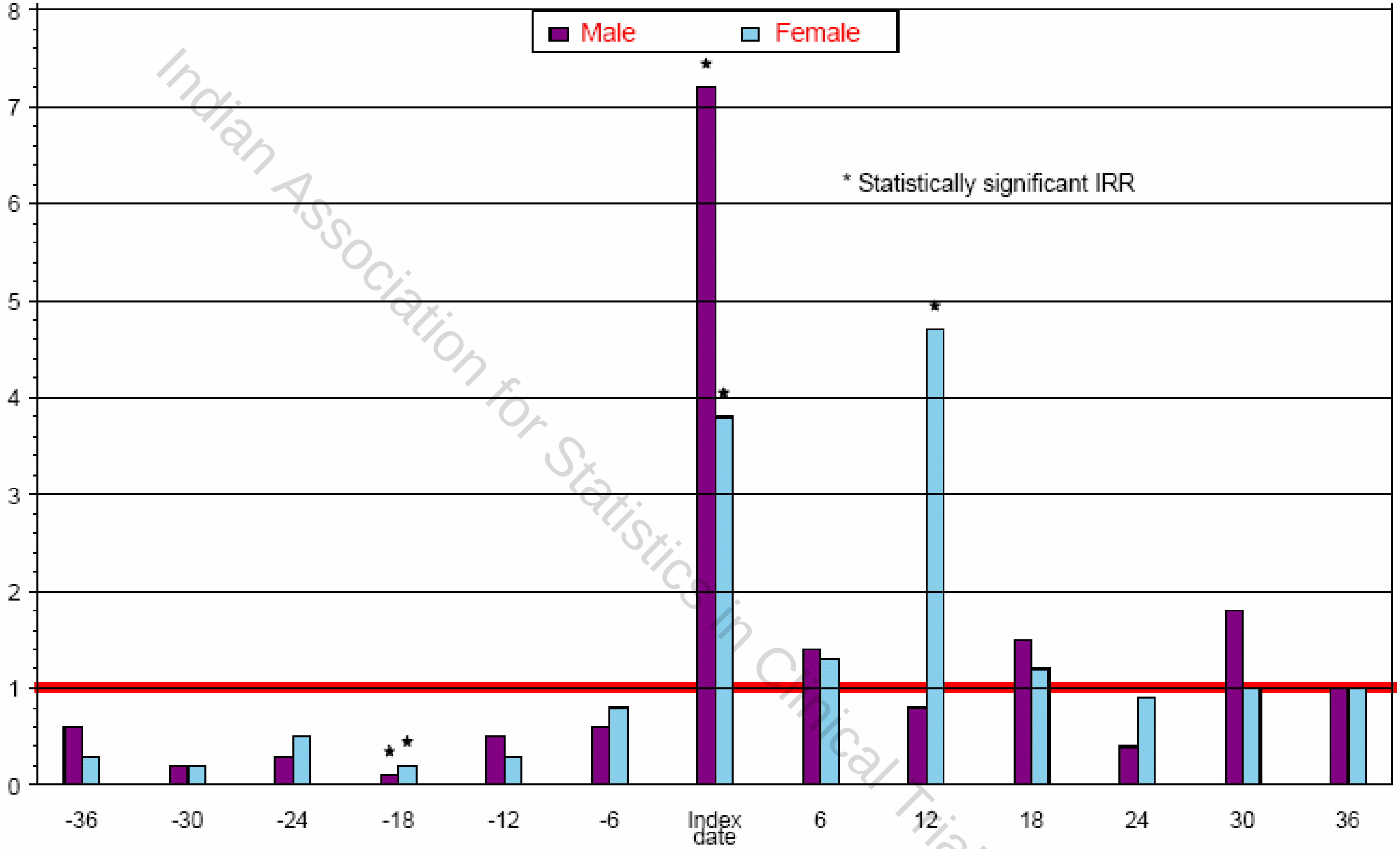
Example3: Put box below X-axis values



Example3: Put box below X-axis values



```
DATA text_below_axis;
   SET test;
   %DCLANNO;
   LENGTH text $5;
      xsys='2'; ysys='3';
%LABEL(visitnum,5,num_of_subjects,black,0,0,1,simplex,E);
%RECT(0,2,3,6,black,1,1);
    %RECT(3,2,8,6,black,1,1);
    %RECT(8,2,18,6,black,1,1);
    %RECT(18,2,38,6,black,1,1);
    %RECT(38,2,70,6,black,1,1);
    %RECT(70,2,79,6,black,1,1);
RUN;
```



Time (months)	-36	-30	-24	-18	-12	-6	Index date	6	12	18	24	30	36
Male	68.18	47.06	48.19	24.69	75.00	285.7	1061	516.1	434.8	777.8	363.6	1778	2000
Female	22.73	22.99	69.77	48.19	98.77	233.8	970.6	571.4	880.0	428.6	727.3	1143	2000

Advantages

1. **Can do anything to everything using annotate facility, the whole graph can be drawn without using procedures like GPLOT, GCHART…. (using PROC GANNO)**

2. **Macro functions available for performing same action, use them in the data step code for the annotate statement instead of writing the individual steps**

   E.g. drawing a line involves function=move, function=draw etc) but only needs one call with macro %LINE (x1, y1, x2, y2, color, line, size);

3. **X & Y axis variables for both numeric and character are available**

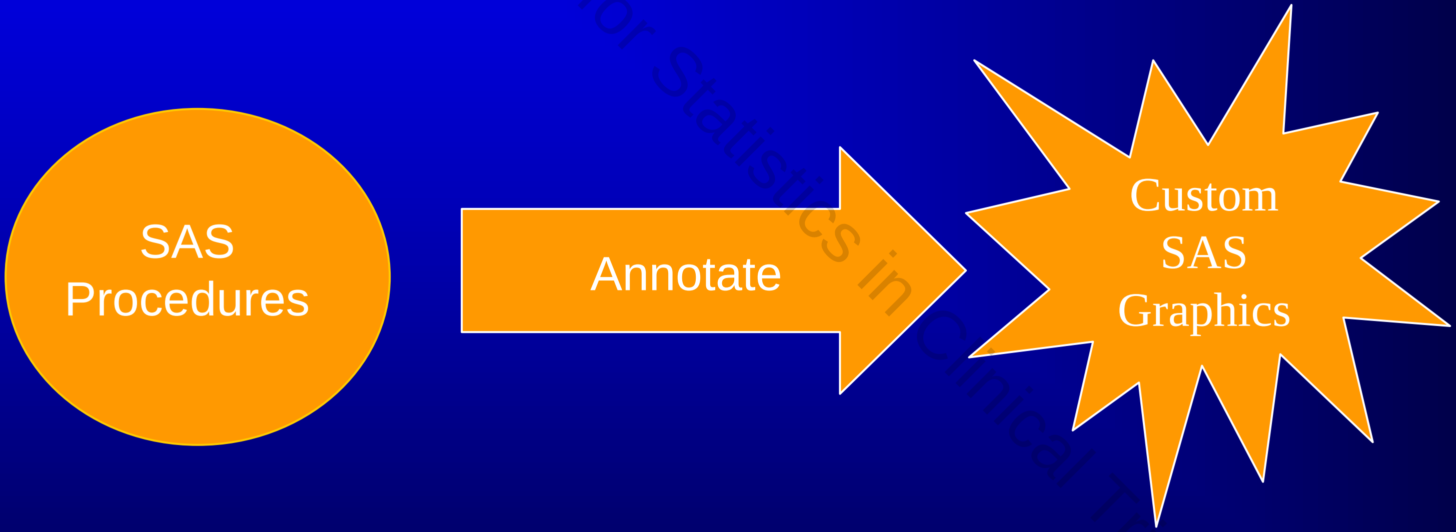
4. **Code can be made generic by using different functions and options available in Annotate facility**

Disadvantages

1. **Code can be complicated when trying to plot graphs using annotate facility only**

Conclusion

Annotate facility can be used as a powerful tool, when used along with SAS/GRAPH procedures



Contact Information

Your comments and questions are valued and encouraged.
Deepak Sriramulu
GlaxoSmithKline Pharmaceuticals Ltd.
Embassy Links, #5 S.R.T Road,
(Cunningham road)
Bangalore 560052
deepak.s.sriramulu@gsk.com

Thank You

GlaxoSmithKline

Cytel

STATISTICAL SOFTWARE & SERVICES

vîslatiφn

Regular Expressions for IrRegular Data!

Jayshree Garade
Anindita Bhattacharjee

www.cytel.com

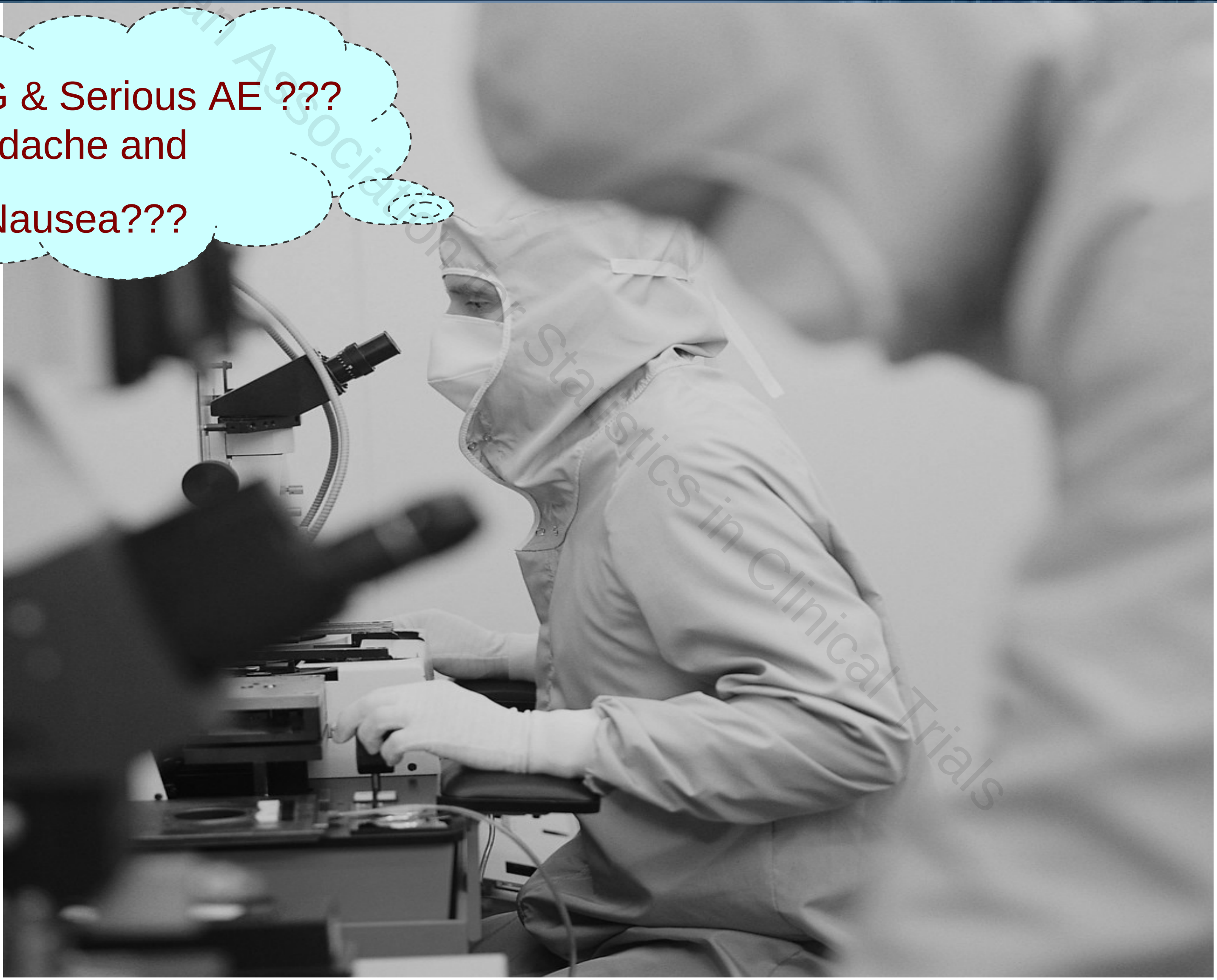
We will go through…

- Background
- Introducing Regular Expressions
- Advantages over SAS String Functions
- Points to note while using Regular Expressions
- References

Background

USUBJID	TRT	SAE	COMMENT
1	TRT A	Y	Headache and nausea
2	Placebo	N	Nausae and headache
3	TRT A	Y	Patient reported headache and nausea
4	TRT A	Y	Pt. Rptd. Head ache and nausea
5	TRT A	Y	Naus. And hdache reported
6	Placebo	N	Pt reported headache at admission; patient later reported nausea.
…	…	…	…

Background

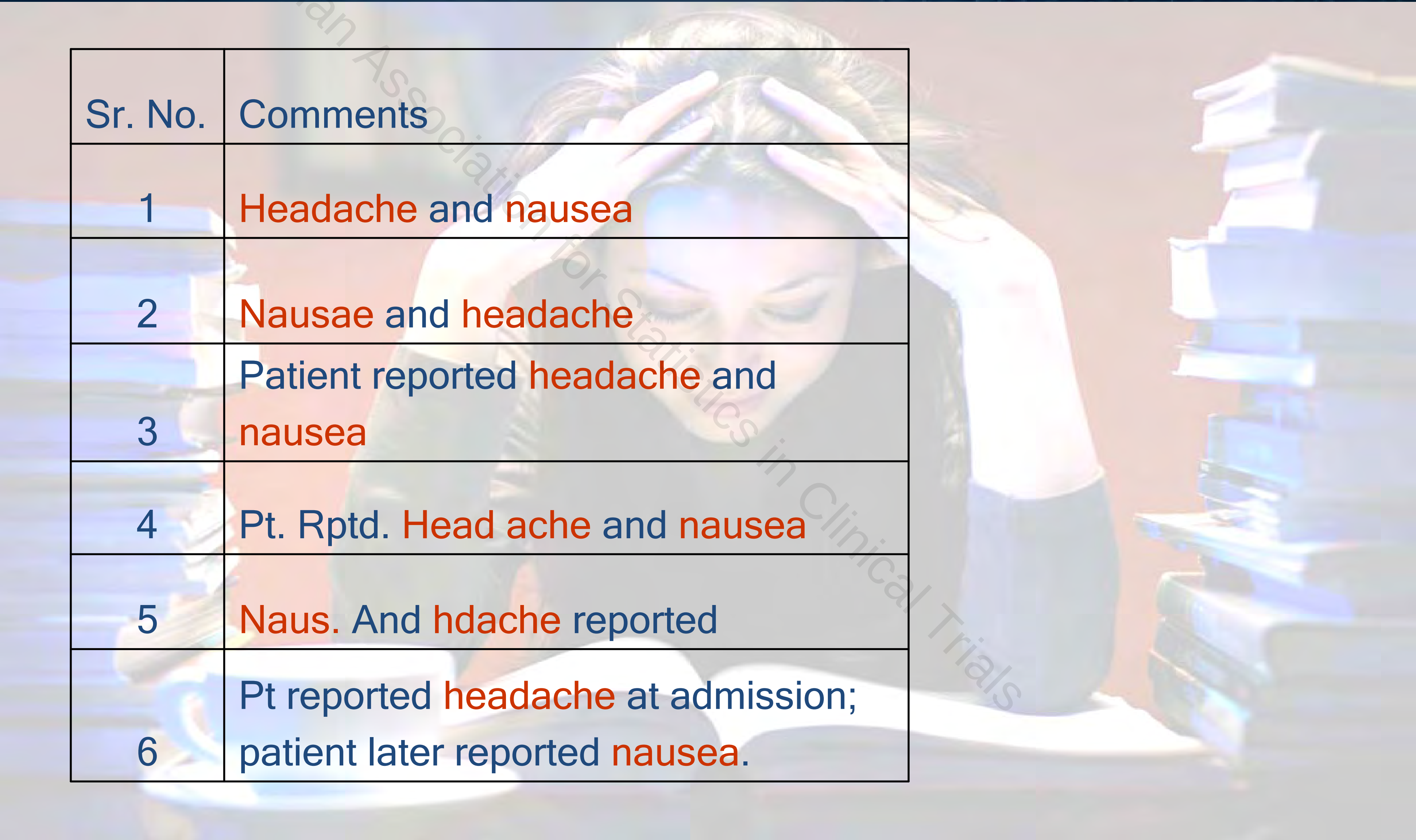


DRUG & Serious AE ??? – Headache and Nausea???

Background

COMMENT
Headache and nausea
Nausae and headache
Patient reported headache and nausea
Pt. Rptd. Head ache and nausea
Naus. And hdache reported
Pt reported headache at admission; patient later reported nausea.

…Inconsistent data



Sr. No.	Comments
1	Headache and nausea
2	Nausae and headache
3	Patient reported headache and nausea
4	Pt. Rptd. Head ache and nausea
5	Naus. And hdache reported
6	Pt reported headache at admission; patient later reported nausea.

Let us start with a Problem…

USUBJID	VISIT	VSDT	PRSDTLTM	VNTR_RT	VNTRTUN
1	1	17-Oct-08	Per 1 D01 Predose	47	/min
1	2	3-Nov-08	Per 1 D01	58	/min
1	2	3-Nov-08	Per 1 D 01 01 hr 30 min	51	/min
1	2	3-Nov-08	Per 1d01 02 hr	49	/min
1	3	4-Nov-08	Day2	53	/min
1	90	3-Feb-09	Poststudy	56	/min
….	….	….	….	….	….

…Timepoint Variable

USUBJID	VISIT	VSDT	PRSDTLTM	VNTR_RT	VNTRTUN
1	1	17-Oct-08	Per 1 D01 Predose	47	/min
1	2	3-Nov-08	Per 1 D01	58	/min
1	2	3-Nov-08	Per 1 D 01 01 hr 30 min	51	/min
1	2	3-Nov-08	Per 1d01 02 hr	49	/min
1	3	4-Nov-08	Day2	53	/min
1	90	3-Feb-09	Poststudy	56	/min
….	….	….	….	….	….

…New Time Description Variable

USUBJID	VISIT	VSDT	PRSDTLTM	VNTR_RT	VNTRTUN	time_desc
1	1	17-Oct-08	Per 1 D01 Predose	47	/min	Predose
1	2	3-Nov-08	Per 1 D01	58	/min	Day 1
1	2	3-Nov-08	Per 1 D 01 01 hr 30 min	51	/min	Day 1, 1 Hour, 30 Minutes
1	2	3-Nov-08	Per 1d01 02 hr	49	/min	Day 1, 2 Hours, 0 Minutes
1	3	4-Nov-08	Day2	53	/min	Day 2
1	90	3-Feb-09	Poststudy	56	/min	Poststudy
….	….	….	….	….	….	….

Problem - Extract and Format

USUBJID	VISIT	VSDT	PRSDTLTM	VNTR_RT	VNTRTUN	time_desc
1	1	17-Oct-08	Per 1 D01 Predose	47	/min	Predose
1	2	3-Nov-08	Per 1 D01	58	/min	Day 1
1	2	3-Nov-08	Per 1 D 01 01 hr 30 min	51	/min	Day 1, 1 Hour, 30 Minutes
1	2	3-Nov-08	Per 1d01 02 hr	49	/min	Day 1, 2 Hours, 0 Minutes
1	3	4-Nov-08	Day2	53	/min	Day 2
1	90	3-Feb-09	Poststudy	56	/min	Poststudy
….	….	….	….	….	….	….

…Ways to approach the problem

- Traditional --- Using SAS String Functions

INDEX TRANWRD SUBSTR ANYALNUM ANYALPHA
ANYDIGIT ANYSPACE NOTALNUM NOTALPHA
ANYALNUM NOTUPPER ANYALPHA FIND ANYDIGIT
FINDC ANYPUNCT ANYSPACE INDEXC NOTALNUM
INDEXW NOTALPHA VERIFY NOTDIGIT CALL CATS
CALL CATT CALL CATX TRANSLATE SCAN SCANQ CALL
SCAN CALL SCANQ COMPARE COMPLEV CALL
COMPCOST SOUNDEX COMPGED SPEDIS MISSING
RANK REPEAT REVERSE…………

… Why traditional method may not work

- Complex patterned text data
- Inconsistent data
- Free Text fields
- Highly unstructured data streams

Using SAS String functions in above cases may be inefficient or impractical if not impossible

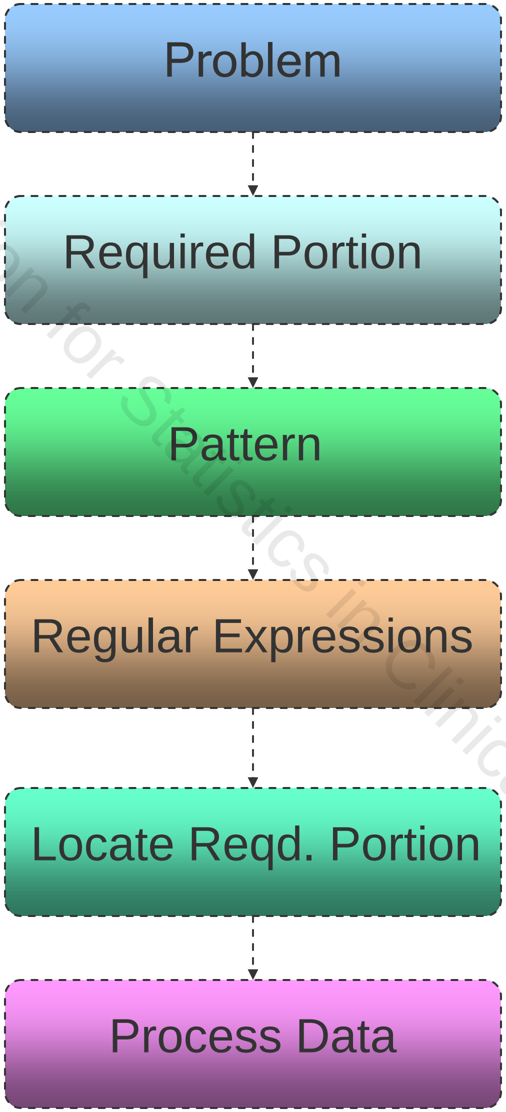
Alternative Approach to Problem…

Introducing REGULAR EXPRESSIONS!!

Introduction - Regular Expressions

- Powerful technique for searching and manipulating text data.
- A mini programming language - pattern matching.
- 2 types – pattern matching functions in SAS
    - ➢SAS Regular Expressions – SAS Version 6.12
    - ➢PERL Regular Expressions - SAS Version 9

Steps to use Regular Expressions…



Problem

Required Portion

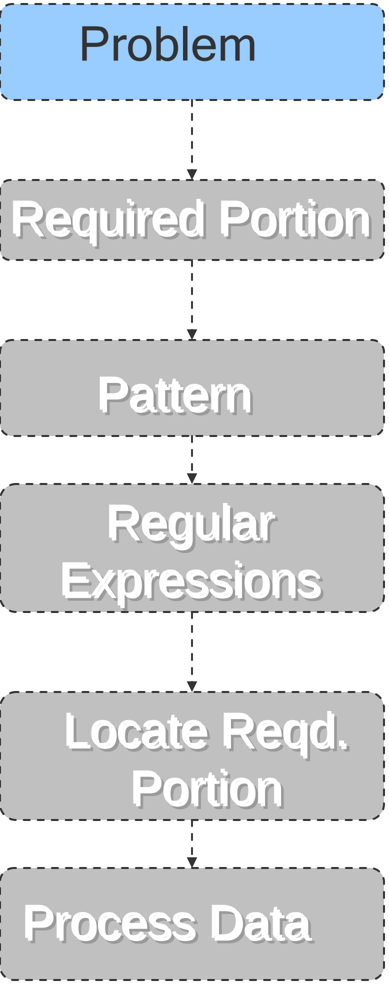
Pattern

Regular Expressions

Locate Reqd. Portion

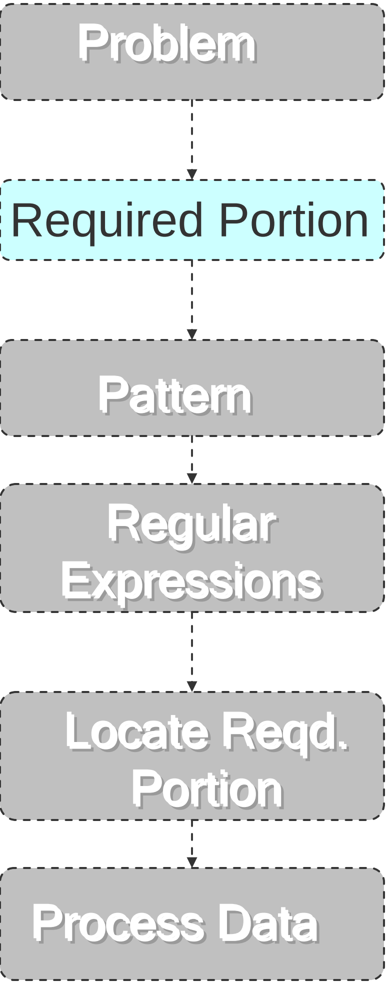
Process Data

Step1 - Identify the problem ...



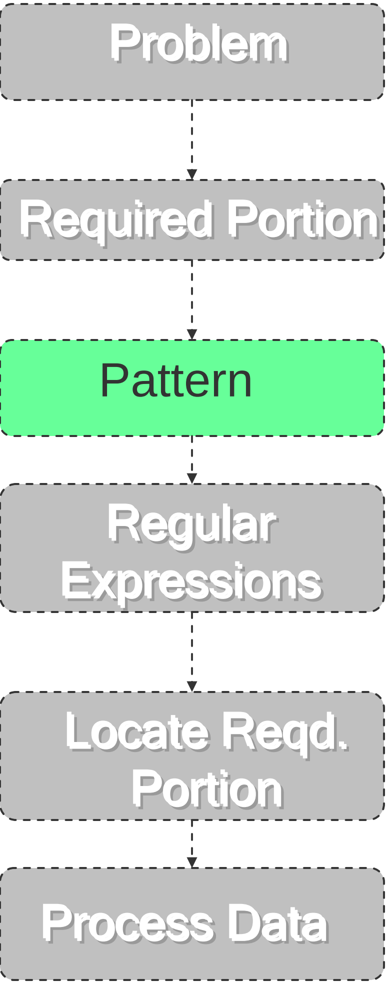
USUBJID	VISIT	VSDT	PRSDTLTM	VNTR_RT	VNTRTUN	time_desc
1	1	17-Oct-08	Per 1 D01 Predose	47	/min	Predose
1	2	3-Nov-08	Per 1 D01	58	/min	Day 1
1	2	3-Nov-08	Per 1 D 01 01 hr 30 min	51	/min	Day 1, 1 Hour, 30 Minutes
1	2	3-Nov-08	Per 1d01 02 hr	49	/min	Day 1, 2 Hours, 0 Minutes
1	3	4-Nov-08	Day2	53	/min	Day 2
1	90	3-Feb-09	Poststudy	56	/min	Poststudy
....	....	....	....	....	....	....

Step2 - Visualize the "Required Portion" within the source text



PRSDTLTM
Per 1 D01 Predose
Per 1 D01
Per 1 D 01 01 hr 30 min
Per 1d01 02 hr
Day2
Poststudy

Step 3 - Identify a pattern



Problem

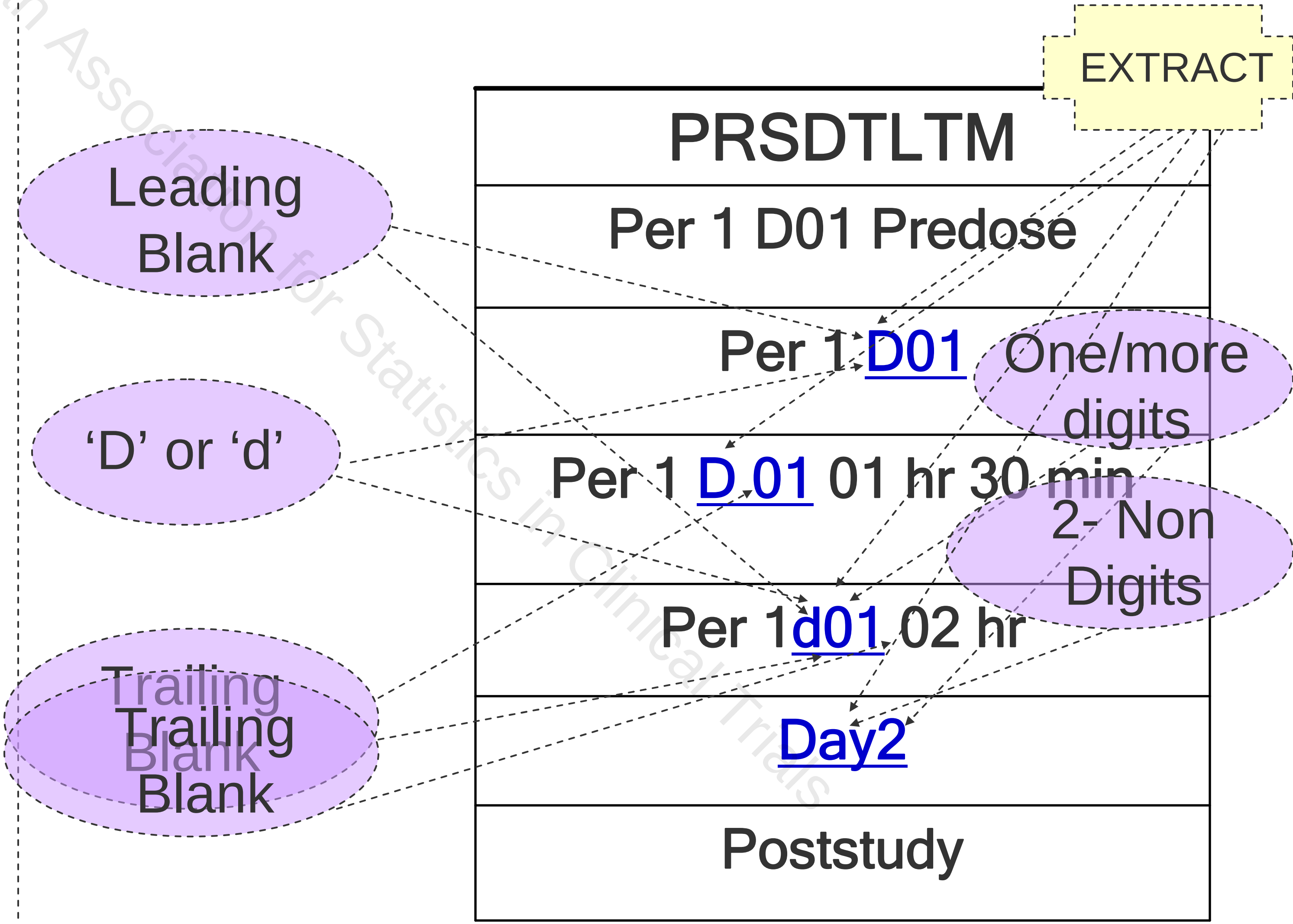
Required Portion

Pattern

Regular Expressions

Locate Reqd. Portion

Process Data



EXTRACT

PRSDTLTM
Per 1 D01 Predose
Per 1 D01
Per 1 D 01 01 hr 30 min
Per 1 d01 02 hr
Day2
Poststudy

Leading Blank

'D' or 'd'

Trailing Blank

Trailing Blank

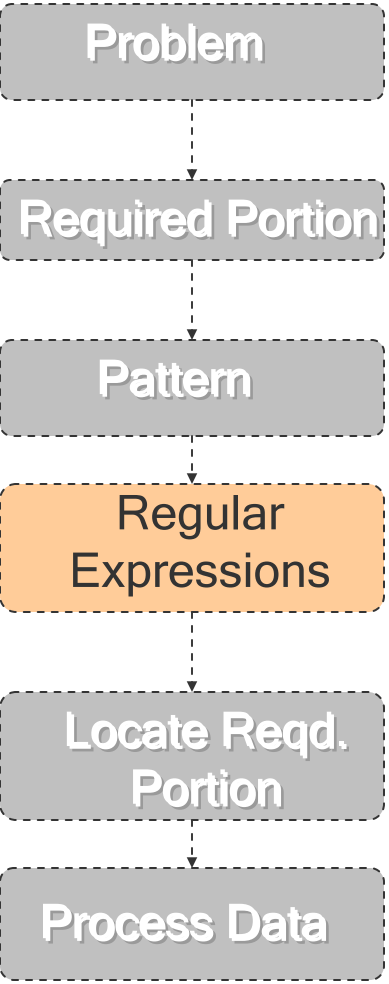
One/more digits

2- Non Digits

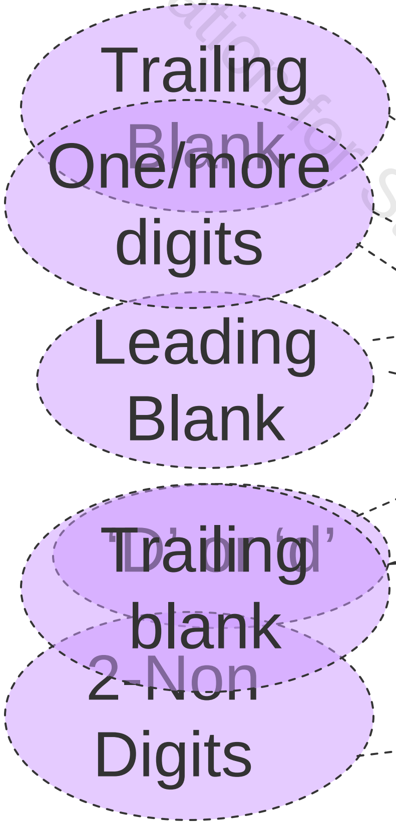
Regular Expressions Syntax…at a glance

Metacharacter	Description
*	Matches the previous sub expression zero or more times
+	Matches the previous sub expression one or more times
?	Matches the previous sub expression zero or one times
\d	Matches a digit (0-9)
\D	Matches a non-digit
\w	Matches a word character (upper or lower case letter, blank, or underscore)
[abc]	Matches any of the characters in the brackets
\(	Matches (

Step 4 - Write the Regular Expression for the pattern

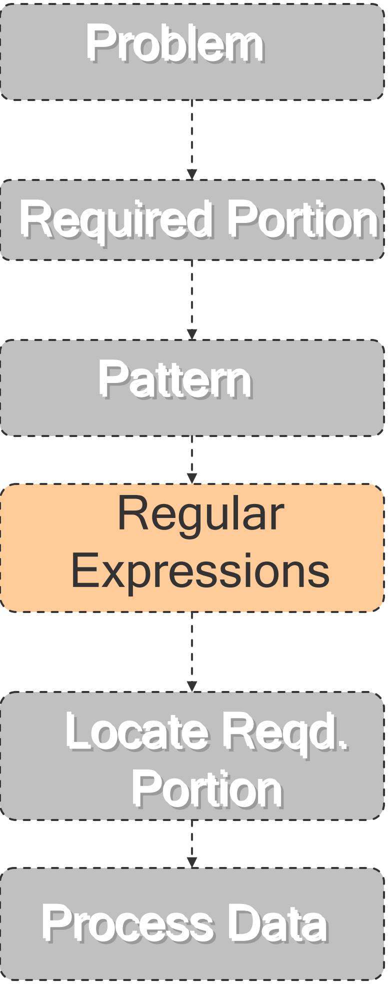


`("/ ? [Dd] (\D\D)?  ?\d+  + /")`



PRSDTLTM
Per 1 D01 Predose
Per 1 D01
Per 1 D 01 01 hr 30 min
Per 1 d01 02 hr
Day2
Poststudy

Step 4 - Write the Regular Expression for the pattern



Problem

Required Portion

Pattern

Regular Expressions

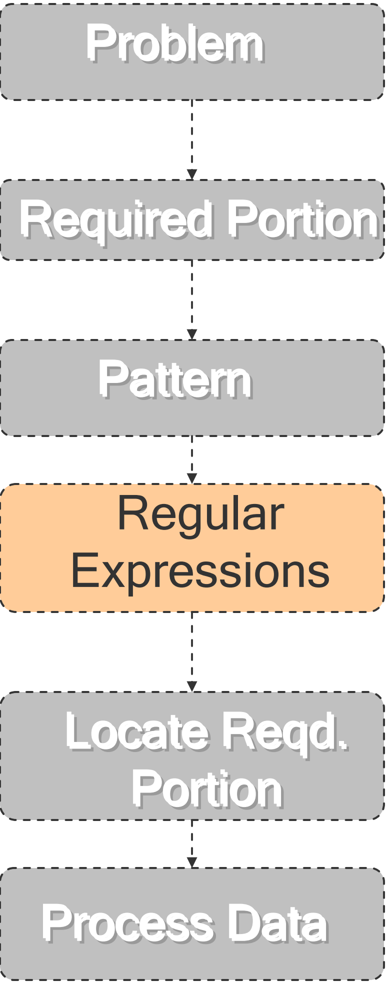
Locate Reqd. Portion

Process Data

```
("/ ?[Dd](\D\D)? ?\d+ +/")
```

PRSDTLTM
Per 1 D01 Predose
Per 1 D01
Per 1 D 01 01 hr 30 min
Per 1d01 02 hr
Day2
Poststudy

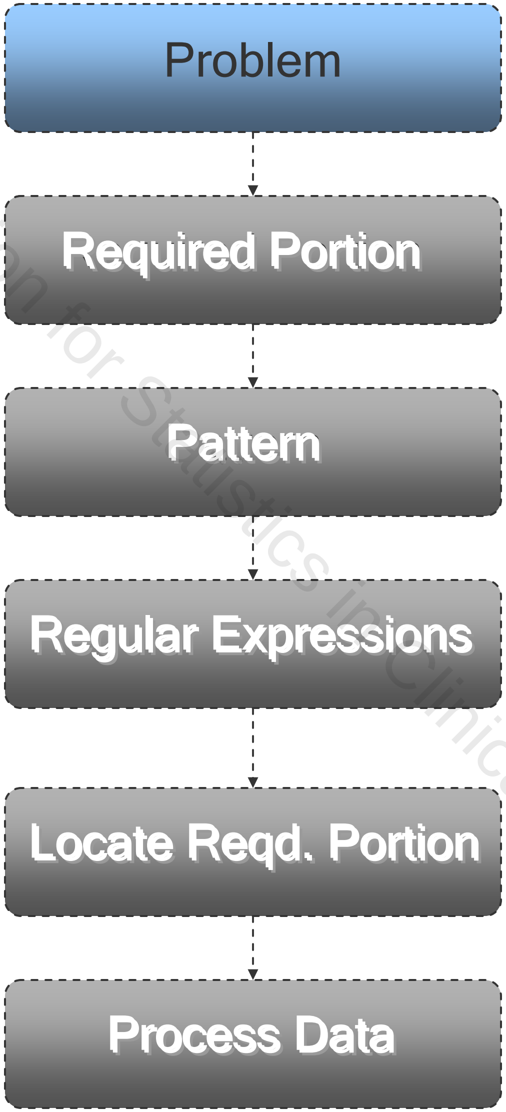
Step 4 - Write the Regular Expression for the pattern



```
/* Extracting the Day Text portion*/
data day_txt;

set lb.ecg(keep = PRSDTLTM);
retain day_exp day_nexp;
 if _n_ = 1 then
 do ;

* defined to describe the day text
pattern;
day_exp= PRXPARSE("/ ?[Dd](\D\D)? ?\d+ +/");


end;


run;
```

Metacharacters

Recap... Steps to use Regular Expressions...



Problem

Required Portion

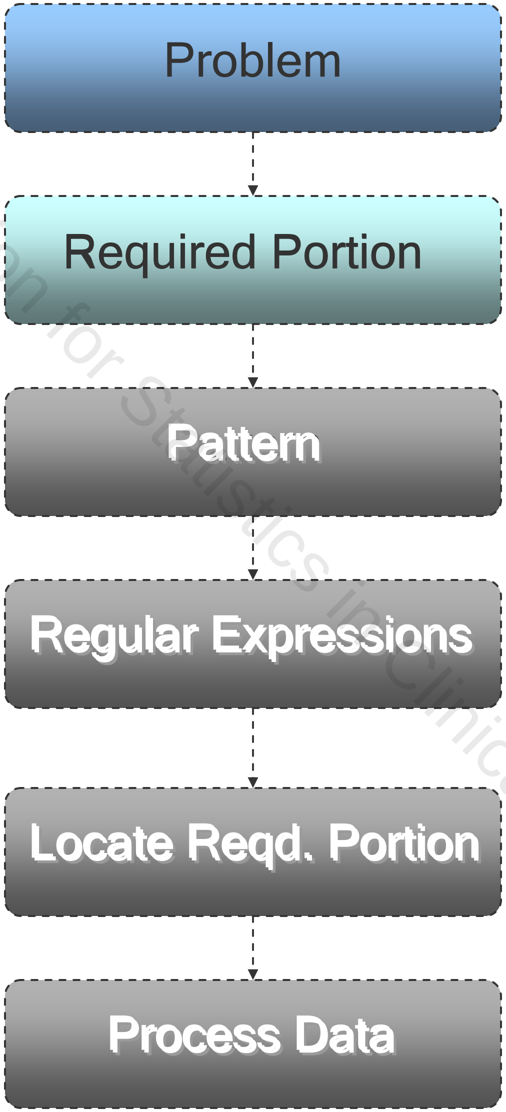
Pattern

Regular Expressions

Locate Reqd. Portion

Process Data

Recap… Steps to use Regular Expressions…



Problem

Required Portion

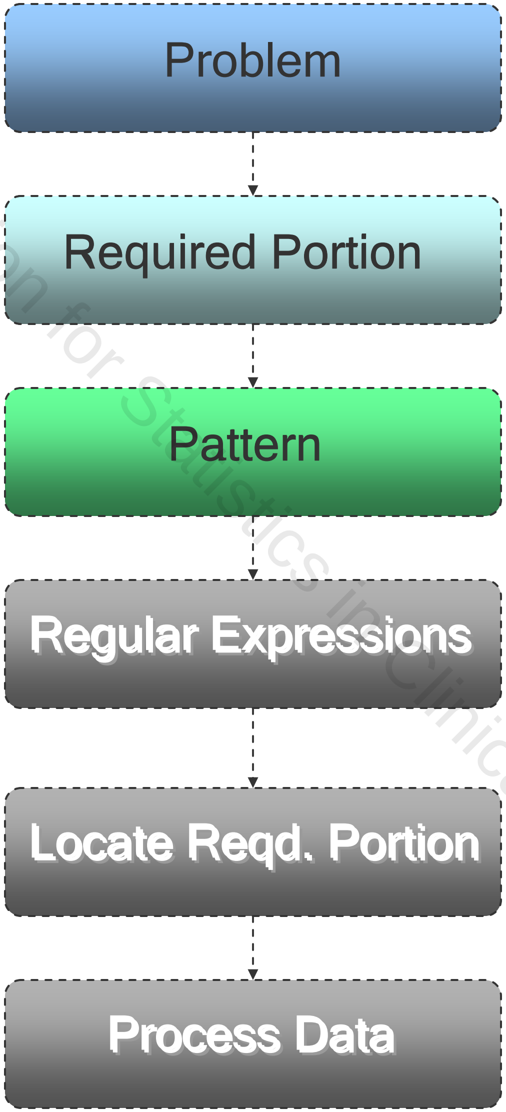
Pattern

Regular Expressions

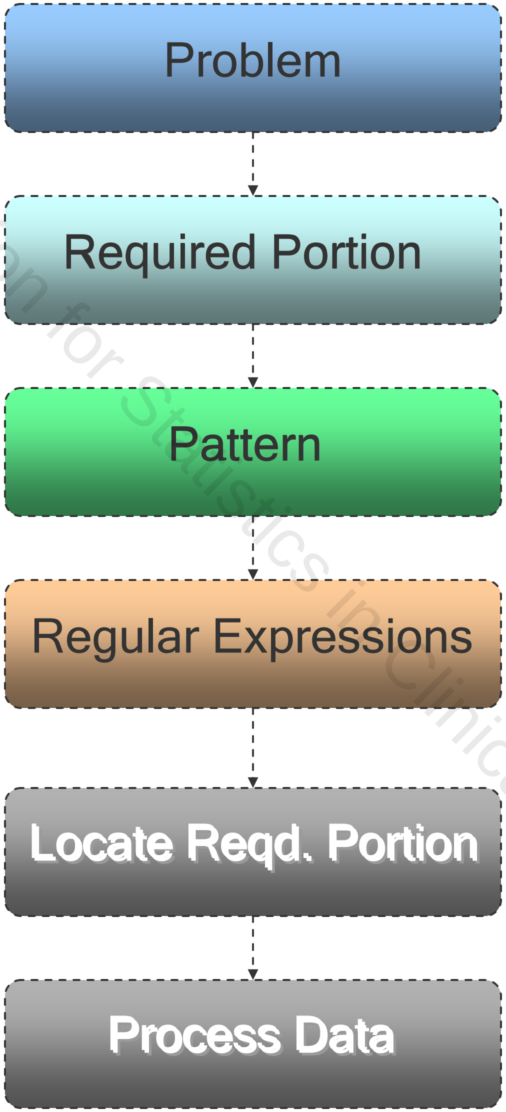
Locate Reqd. Portion

Process Data

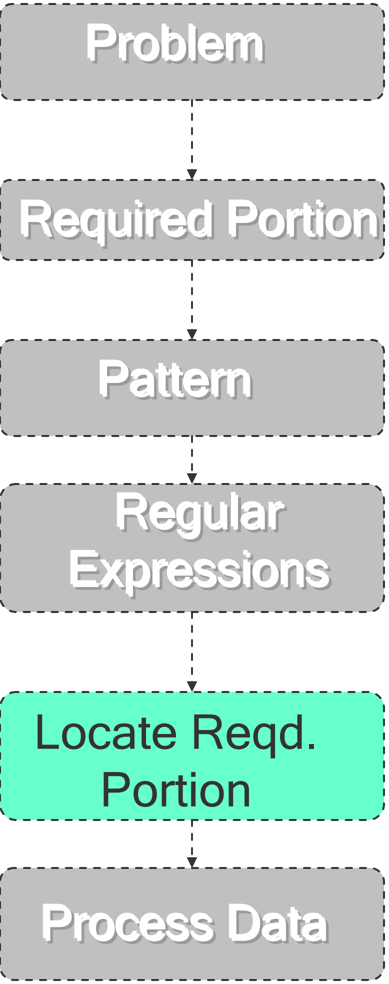
Recap… Steps to use Regular Expressions…



Recap… Steps to use Regular Expressions…



Step 5 - Locate the "Required Portion"



```
/* Extracting the Day Text portion*/
data day_txt;
    set lb.ecg(keep = PRSDTLTM);
    retain day_exp day_nexp;
    if _n_ = 1 then
    do ;
      * defined to describe the day text pattern;
      day_exp = PRXPARSE("/ ?[Dd](\D\D)? ?\d+ +/");
end;


*Locating the day text pattern in PRSDTLTM
var;
CALL
PRXSUBSTR(day_exp,PRSDTLTM,dayst,dayln);

run;
```

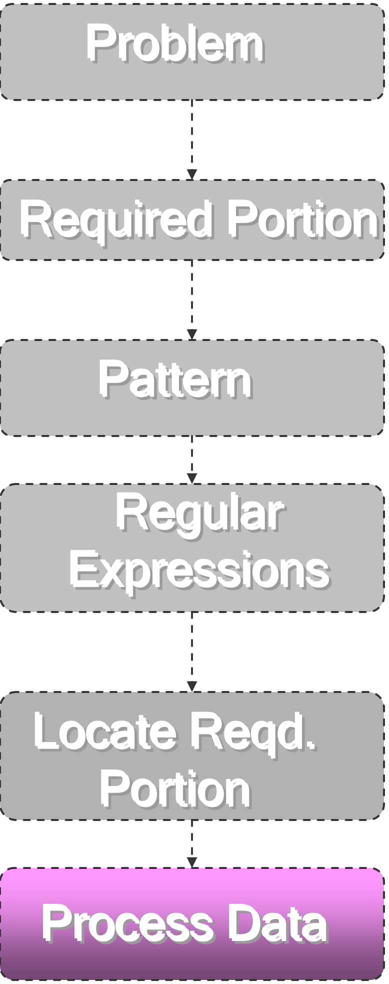
Pattern defn

Source Variable

Stores Start position of matched string

Stores length of matched string

Step 6 - Use other SAS text functions to further process data



```
/* Extracting the Day Text portion*/
data day_txt;

    set lb.ecg(keep = PRSDTLTM);
    retain day_exp day_nexp;

    if _n_ = 1 then
    do ;
      * defined to describe the day text pattern;
      day_exp = PRXPARSE("/ ?[Dd](\D\D)? ?\d+ +/");
     end;

     * Locating the day text pattern in the PRSDTLTM var;
    CALL PRXSUBSTR(day_exp,PRSDTLTM, dayst, dayln);

* Extracting
day_txt =
    substrn(PRSDTLTM,dayst,dayln);


run;
```

Source Variable

Starting Position

Length of matched pattern

…Output

PRSDTLTM	day_txt
Per 1 D01 Predose	
Per 1 D01	D01
Per 1 D 01 01 hr 30 min	D 01
Per 1d01 02 hr	d01
Day2	Day2
Poststudy	

Extracted string

Advantages…

- Compact solution

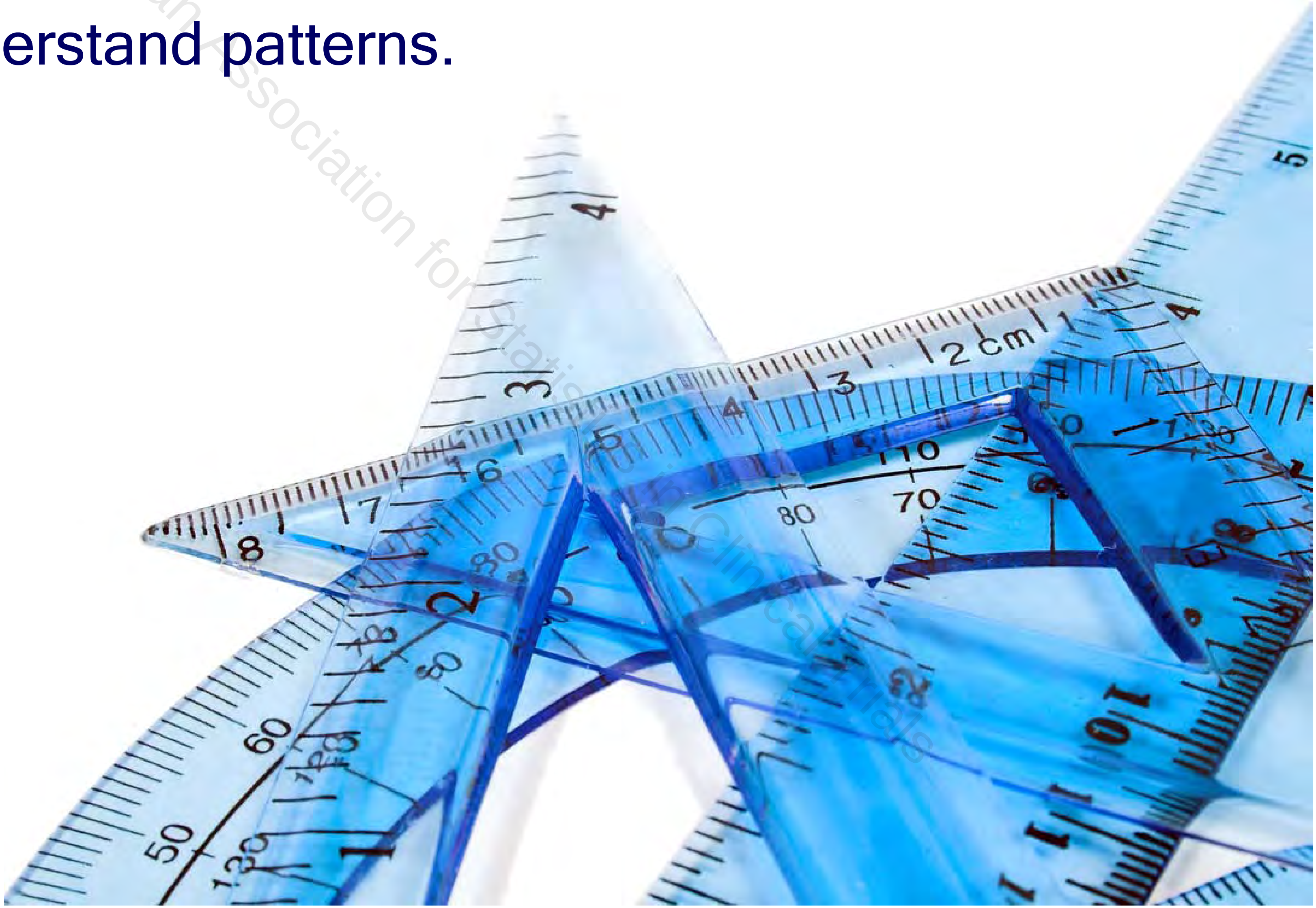
- Tremendous flexibility
  - ➢Concise description.
  - ➢Highly unstructured data streams.
  - ➢Multiple matching patterns in one step.

Know before you leap

Document thoroughly.

…Know before you leap

Understand patterns.

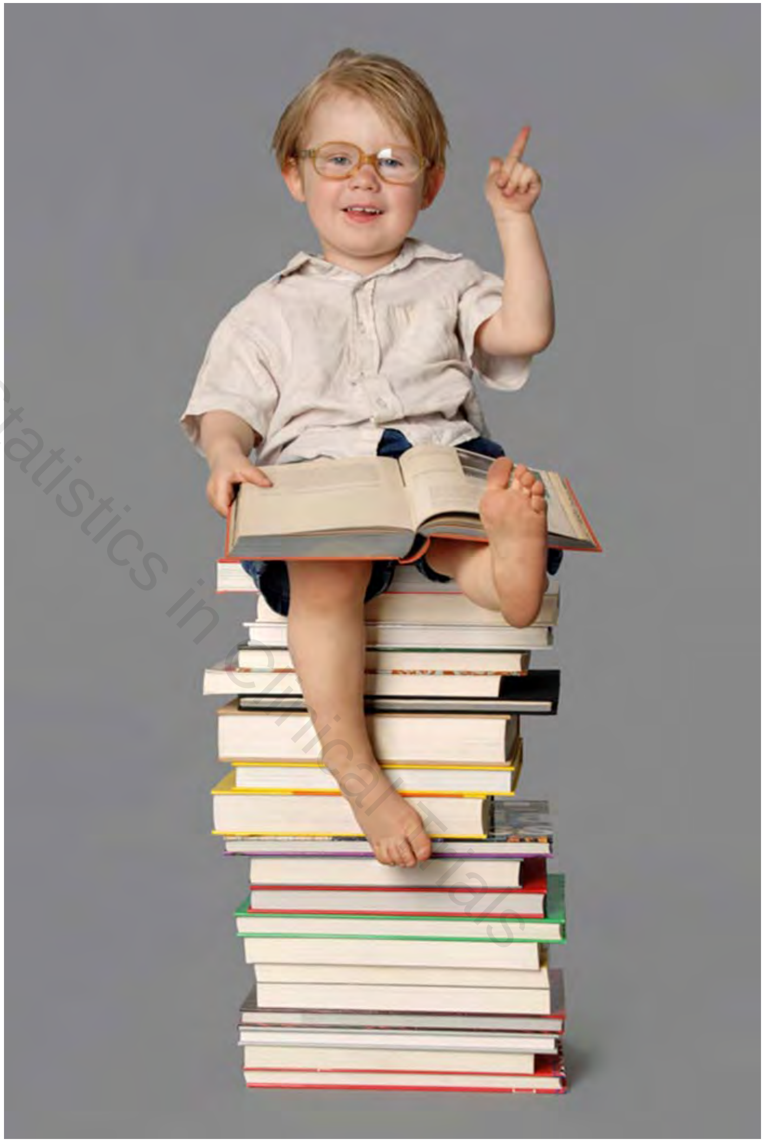


…Know before you leap

Define before use.

…Know before you leap

Define only once.



Take away statement…

Programmers who process text data on a regular basis should strongly consider adding regular expressions to their programming tool bag.

References…

□ Perl Regular Expressions in SAS

Kevin McGowan , Constella Group, Durham, NC

□ Using Regular Expressions with SAS®

Brian Conley, Prevision Marketing, MA

References…

- □ Paper TU02-

  An Introduction to Regular Expressions with Examples from Clinical Data - Richard F. Pless, Ovation Research Group, Highland Park, IL

- □ SUGI 29-Tutorials - Paper 265-29

  An Introduction to Perl Regular Expressions in SAS 9

  Ron Cody, Robert Wood Johnson Medical School, Piscataway, NJ

- □ An Introduction to PERL Regular Expression in SAS®

  James J. Van Campen, SRI International, Menlo Park, CA

Questions??

Email Address

Jayshree Garade - jayshree.garade@vislation.com

Anindita Bhattacharjee - anindita.b@vislation.com

Importance and Methodologies of Validation in Clinical Trials Reporting

Vijay Keerthi S

03-Apr-2009

Agenda

- What is Validation
- Why is Validation needed
- How do you approach Validation
- Independent programming
- Use of Validation dataset
- General Techniques to Facilitate Validation

What is Validation?

Validation is the act, or process, of proving the accuracy and integrity of the output of the programming being performed.

Why is Validation needed?

- Reporting accuracy is crucial because these data represent people, the patients or subjects of the trials
- Validation is a regulatory requirement
- Developing a positive relationship with clients

How do you approach Validation?

- Start with all the information
- Have a Validation plan
- Make the code Do the work
- Ask questions
- Be proactive
- Validating early saves time
- Validation must come first

Independent Programming

- One of the standard validation methods in which two independent programmers program and then compare the output

- Principles:
  - The job at hand is to find as many bugs or errors in the result as possible
  - Put your trust away when you validate his/her output
  - Also, the program developers should not look upon the independent testing process as criticism, nor should they perceive it as developer testing of the program.

Validation Dataset

- In general, a random subset of the data will be taken from the listing and check by hand to make sure the results are correctly portrayed in the output.
- This labor intensive process looks for inconsistencies by visually examining the outputs.
- This is very time consuming and prone to errors

Validation Dataset (Cont…)

Preferred Approach

- First Step: Source programmer to create a SAS dataset that will be used in creating the report. This SAS dataset is termed as Validation dataset
- Second Step: QC programmer need to independently program the same information that is in the Validation dataset.
- Third Step: Use PROC COMPARE to let SAS do the Comparisons

```
proc compare base=ORIGINAL
                        comp=VALIDATE;

run;
```

- This procedure can be applied for the validation of tables and graphs also.
- Additionally we should conduct a visual check of the graphs for correctness and analyze the outliners.

Validation Dataset (Cont…)

What to look for in the PROC COMPARE output:

- To be confident that your ORIGINAL and VALIDATE data sets are similar be sure to check that the number of variables (Nvar) and observations (Nobs) are the same (see the top of the comparison output).
- It is also a good idea to check that the format of the variables is the same, the output will indicate if the formats are not the same.
- Finally look for this message at the bottom of the output *"NOTE: No unequal values were found. All values compared are exactly equal."* When you see this message in addition to matching Nvar and Nobs values then your job of validation is done!

General Techniques to Facilitate Validation

- Using PROC FREQ for Validation
- MSGLEVEL=I in MERGE statement
- MERGE statement with IN= option
- Using Macros Effectively and Judiciously
- Maintain a clean log
- Flagging Problem Data
- Don’t Delete
- Drop the duplicates
- The Essential Checklist

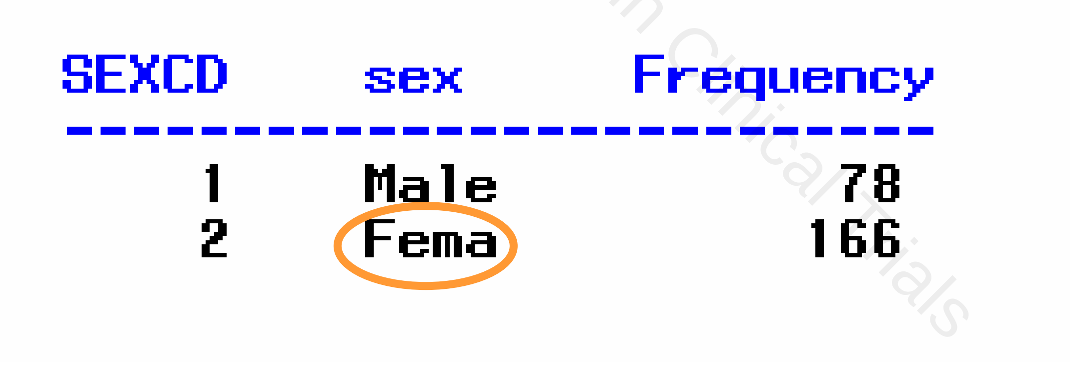
Using PROC FREQ for Validation

- Most commonly used in Clinical Data Validation
- Helpful when performing cross-variable checks
- An Example:

```
data demo;
     set orglib.demo;
               if sexcd eq 1 then sex = "Male";
               else if sexcd eq 2 then sex = "Female";
run;

proc freq data=demo;
     tables sexcd*sex / list missing;
     title 'CHECK RECODES';
run;
```

Using PROC FREQ for Validation (Cont…)

- In this case it is creating a character version of a variable that was originally collected as a numeric variable
- The code needs to prove that the meaning of the variables being transformed has not changed
- In this output, it is easy to spot the error in reformatting the SEXCD variable



```
SEXCD     sex       Frequency
-----------------------------
    1     Male             78
    2     Fema            166
```

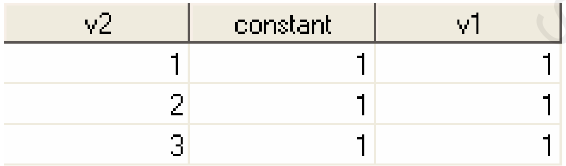
General Techniques to Facilitate Validation

- Using PROC FREQ for Validation
- MSGLEVEL=I in MERGE statement
- MERGE statement with IN= option
- Using Macros Effectively and Judiciously
- Maintain a clean log
- Flagging Problem Data
- Don’t Delete
- Drop the duplicates
- The Essential Checklist

MSGLEVEL=I in MERGE statement

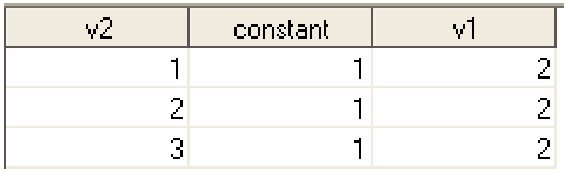
- MERGE is a very effective and powerful tool
- Can give surprising and undesirable results

out1



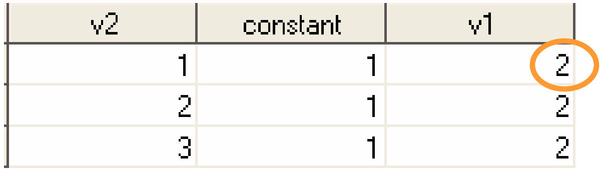
v2	constant	v1
1	1	1
2	1	1
3	1	1

main



v2	constant	v1
1	1	2
2	1	2
3	1	2

```
data out3;
        merge out1 main;
        by constant v2;
run;
```

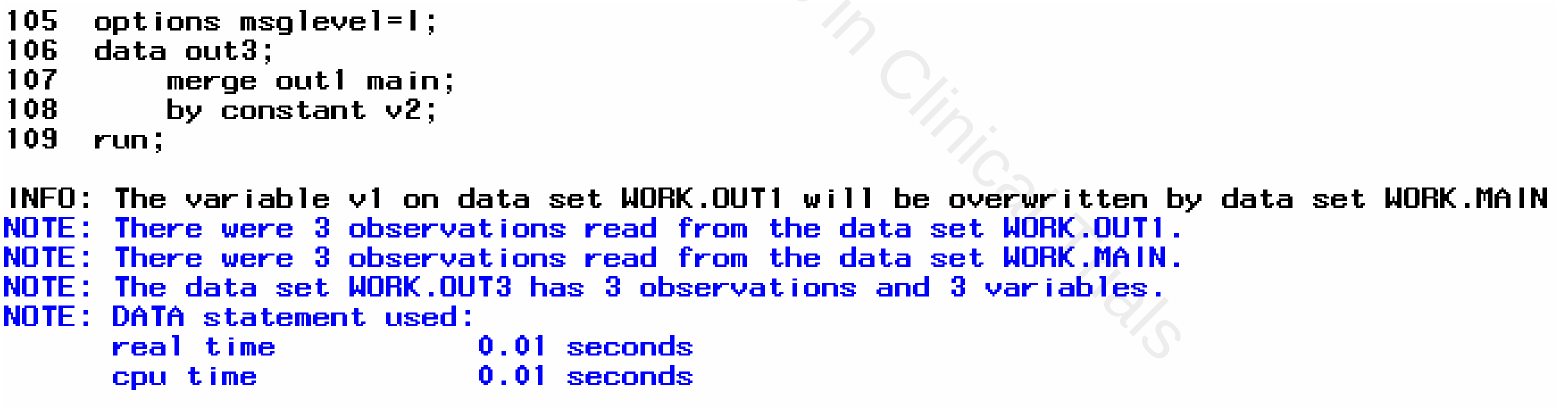


v2	constant	v1
1	1	2
2	1	2
3	1	2

MSGLEVEL=I in MERGE statement (Cont…)

```
options msglevel=I;
data out3;
        merge out1 main;
        by constant v2;
run;
```

- The MSGLEVEL system option gives additional information in the log when merging the datasets



```
105   options msglevel=I;
106   data out3;
107       merge out1 main;
108       by constant v2;
109   run;

INFO: The variable v1 on data set WORK.OUT1 will be overwritten by data set WORK.MAIN
NOTE: There were 3 observations read from the data set WORK.OUT1.
NOTE: There were 3 observations read from the data set WORK.MAIN.
NOTE: The data set WORK.OUT3 has 3 observations and 3 variables.
NOTE: DATA statement used:
      real time           0.01 seconds
      cpu time            0.01 seconds
```

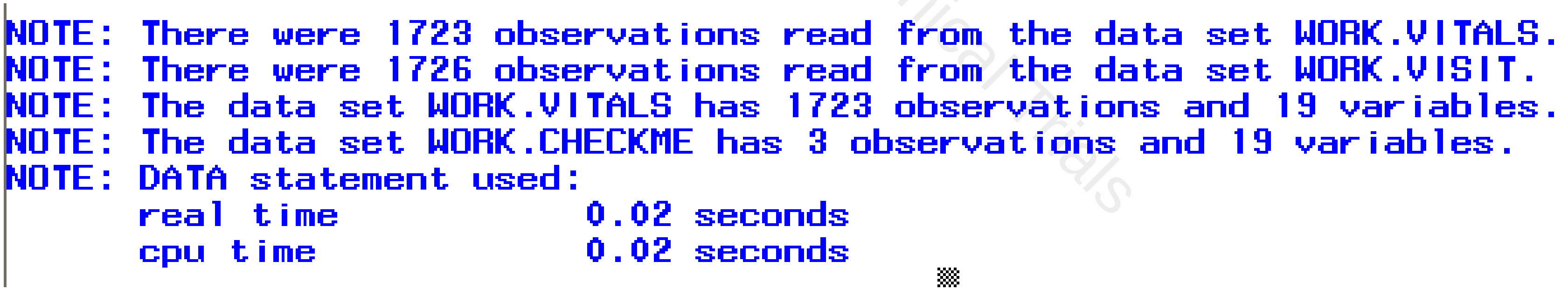
General Techniques to Facilitate Validation

- Using PROC FREQ for Validation
- MSGLEVEL=I in MERGE statement
- MERGE statement with IN= option
- Using Macros Effectively and Judiciously
- Maintain a clean log
- Flagging Problem Data
- Don’t Delete
- Drop the duplicates
- The Essential Checklist

MERGE statement with IN= option

- MERGE statement with IN= option can be great tool for validation

```
data vitals checkme;
            merge vitals(in=invl) visit (in=invt);
            by inv_no patid visit ;
            if invl and invt then output vitals ;
            else output checkme;
run ;
```



```
NOTE: There were 1723 observations read from the data set WORK.VITALS.
NOTE: There were 1726 observations read from the data set WORK.VISIT.
NOTE: The data set WORK.VITALS has 1723 observations and 19 variables.
NOTE: The data set WORK.CHECKME has 3 observations and 19 variables.
NOTE: DATA statement used:
      real time           0.02 seconds
      cpu time            0.02 seconds
```

General Techniques to Facilitate Validation

- Using PROC FREQ for Validation
- MSGLEVEL=I in MERGE statement
- MERGE statement with IN= option
- Using Macros Effectively and Judiciously
- Maintain a clean log
- Flagging Problem Data
- Don’t Delete
- Drop the duplicates
- The Essential Checklist

Using Macros Effectively and Judiciously

- General rule for truly efficient programming is to add macros only when they add significantly to the process
- Macros can also create validation nightmares if used in excess
- Important to consider the cost-benefit ratio
- Use mprint, mlogic and symbolgen for macros validation

General Techniques to Facilitate Validation

- Using PROC FREQ for Validation
- MSGLEVEL=I in MERGE statement
- MERGE statement with IN= option
- Using Macros Effectively and Judiciously
- Maintain a clean log
- Flagging Problem Data
- Don’t Delete
- Drop the duplicates
- The Essential Checklist

Maintain a clean log

- Log not only be free of error but also free of warnings and some of the notes
    - NOTE: Numeric values have been converted to character values at the places given by: (Line):(Column)
    - NOTE: Character values have been converted to numeric values at the places given by: (Line):(Column)
- It is much easier to notice real issues if they arise
- Any issues caused by new data are easy to see as you skim through the file

General Techniques to Facilitate Validation

- Using PROC FREQ for Validation
- MSGLEVEL=I in MERGE statement
- MERGE statement with IN= option
- Using Macros Effectively and Judiciously
- Maintain a clean log
- Flagging Problem Data
- Don’t Delete
- Drop the duplicates
- The Essential Checklist

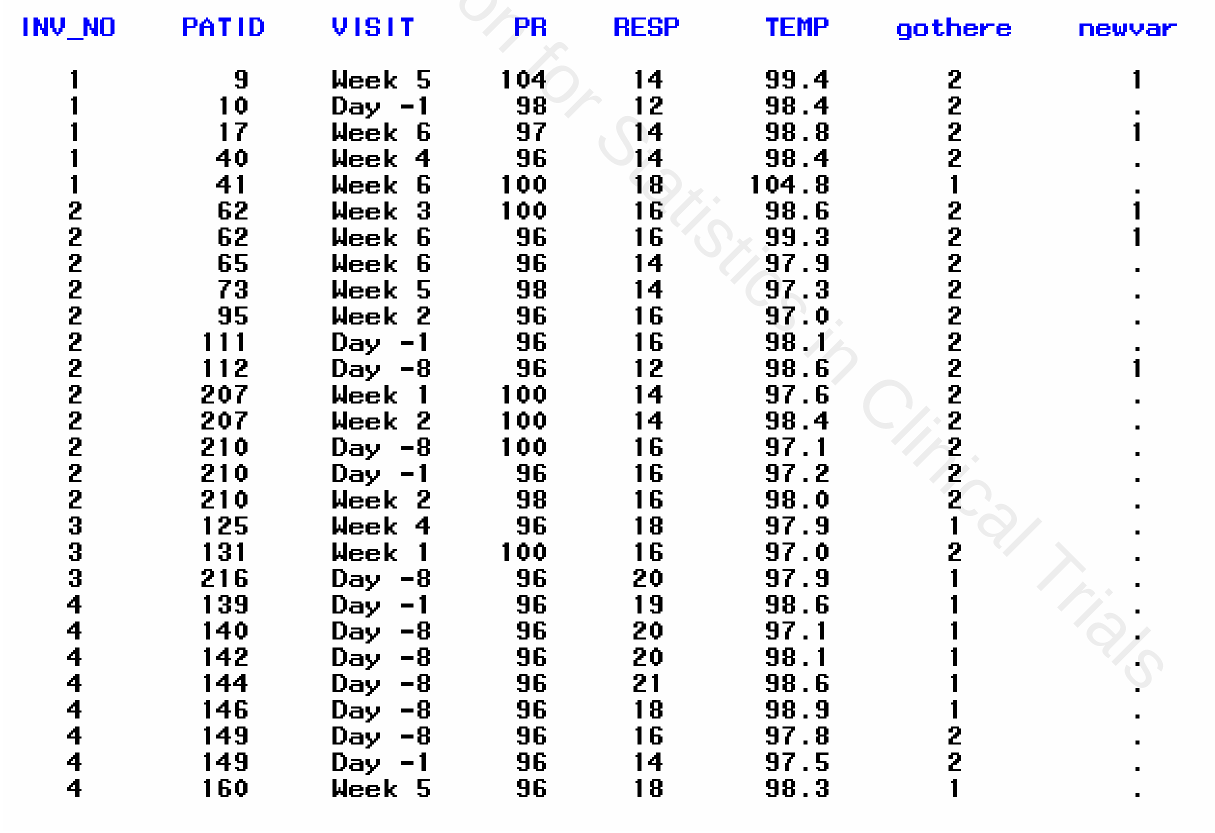
Flagging Problem Data

- Useful when tracking how data is moving through complicated logic statements

```
data flags ;
    set orglib.vitals ;
    if pr gt 95 then do ;
      gothere = 1 ;
      if resp le 16 then do ;
                  gothere = 2 ;
                  if temp ge 99 then newvar = 1 ;
      end ;
    end ;
run ;
```

Flagging Problem Data (Cont…)

```
proc print data=flags (where=(gothere ne .)) ;
    var inv_no patid visit pr resp temp gothere newvar ;
    title "CHECK LOGIC FOR NEWVAR" ;
run ;
```



```
INV_NO    PATID    VISIT      PR    RESP    TEMP    gothere    newvar

   1         9     Week 5    104     14      99.4       2          1
   1        10     Day -1     98     12      98.4       2          .
   1        17     Week 6     97     14      98.8       2          1
   1        40     Week 4     96     14      98.4       2          .
   1        41     Week 6    100     18     104.8       1          .
   2        62     Week 3    100     16      98.6       2          1
   2        62     Week 6     96     16      99.3       2          1
   2        65     Week 6     96     14      97.9       2          .
   2        73     Week 5     98     14      97.3       2          .
   2        95     Week 2     96     16      97.0       2          .
   2       111     Day -1     96     16      98.1       2          .
   2       112     Day -8     96     12      98.6       2          1
   2       207     Week 1    100     14      97.6       2          .
   2       207     Week 2    100     14      98.4       2          .
   2       210     Day -8    100     16      97.1       2          .
   2       210     Day -1     96     16      97.2       2          .
   2       210     Week 2     98     16      98.0       2          .
   3       125     Week 4     96     18      97.9       1          .
   3       131     Week 1    100     16      97.0       2          .
   3       216     Day -8     96     20      97.9       1          .
   4       139     Day -1     96     19      98.6       1          .
   4       140     Day -8     96     20      97.1       1          .
   4       142     Day -8     96     20      98.1       1          .
   4       144     Day -8     96     21      98.6       1          .
   4       146     Day -8     96     18      98.9       1          .
   4       149     Day -8     96     16      97.8       2          .
   4       149     Day -1     96     14      97.5       2          .
   4       160     Week 5     96     18      98.3       1          .
```

General Techniques to Facilitate Validation

- Using PROC FREQ for Validation
- MSGLEVEL=I in MERGE statement
- MERGE statement with IN= option
- Using Macros Effectively and Judiciously
- Maintain a clean log
- Flagging Problem Data
- Don’t Delete
- Drop the duplicates
- The Essential Checklist

Don’t Delete

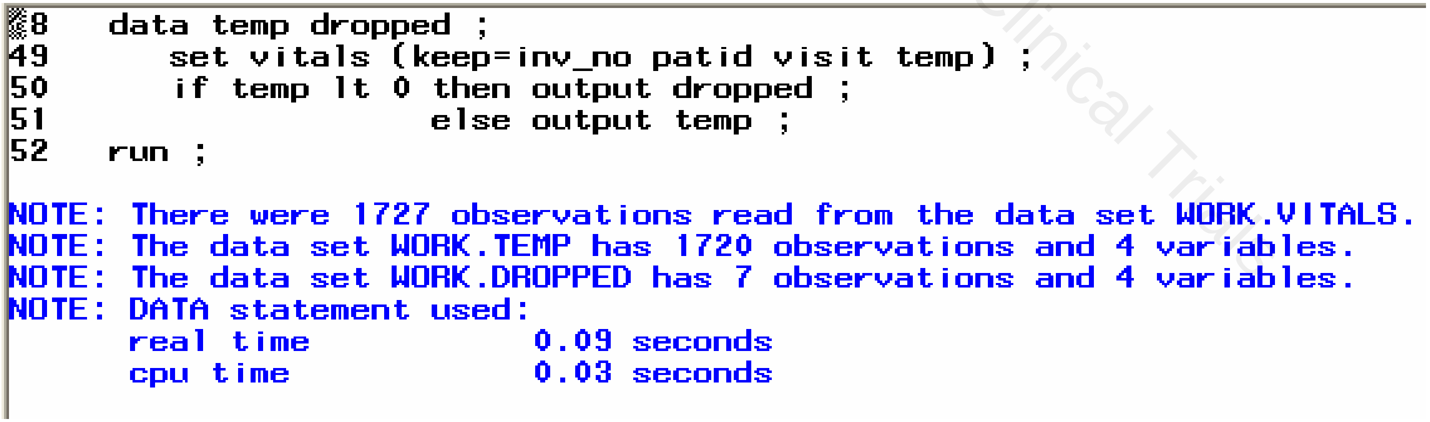
- Often you might want to remove unnecessary records from a dataset
- Generally tempted to code a simple statement like:

  If temp lt 0 then delete;

- This does not allow to check the deleted records

Don’t Delete (Cont…)

```
data temp dropped ;
   set vitals (keep=inv_no patid visit temp) ;
   if temp lt 0 then output dropped ;
                 else output temp ;
run ;

proc print data=dropped ;
   title 'TEMP LESS THAN 0 SO DROPPED FROM DATA SET' ;
run ;
```



```
     data temp dropped ;
        set vitals (keep=inv_no patid visit temp) ;
        if temp lt 0 then output dropped ;
                          else output temp ;
     run ;

NOTE: There were 1727 observations read from the data set WORK.VITALS.
NOTE: The data set WORK.TEMP has 1720 observations and 4 variables.
NOTE: The data set WORK.DROPPED has 7 observations and 4 variables.
NOTE: DATA statement used:
      real time           0.09 seconds
      cpu time            0.03 seconds
```

General Techniques to Facilitate Validation

- Using PROC FREQ for Validation
- MSGLEVEL=I in MERGE statement
- MERGE statement with IN= option
- Using Macros Effectively and Judiciously
- Maintain a clean log
- Flagging Problem Data
- Don’t Delete
- Drop the duplicates
- The Essential Checklist

Drop the duplicates

- Often datasets contain duplicate records that needs to be removed
- This can be done using Proc Sort and options NODUPKEY or NODUPREC
- To check the dropped duplicated records use the DUPOUT option in SAS 9 PROC SORT

Drop the duplicates (Cont…)

```
proc sort data=vitals (where=(pr gt 90))
          out=htemp
          nodupkey
          dupout=dropped ;
   by inv_no patid ;
run ;

proc print data=htemp;
     title ‘PATIENTS WITH PULSE RATE OVER 90' ;
run ;

proc print data=dropped ;
     title 'DUPLICATES DROPPED FROM PROC SORT' ;
run ;
```

General Techniques to Facilitate Validation

- Using PROC FREQ for Validation
- MSGLEVEL=I in MERGE statement
- MERGE statement with IN= option
- Using Macros Effectively and Judiciously
- Maintain a clean log
- Flagging Problem Data
- Don’t Delete
- Drop the duplicates
- The Essential Checklist

The Essential Checklist

- Layout and format of the displays is consistent with the RAP.
- N' to be consistent across tables
- Check the population label.
- Units have to be displayed.
- Add footnotes whenever necessary.
- Check for truncation.
- Check the decimal places for summary statistics.
- Percentages should be checked manually. (Random)
- Proc compare output for QC of Analysis datasets should have no message other than “No unequal values found”.
- Estimate should lie within confidence interval.
- P value should lie between 0 and 1.
- Reconcile Graphs with corresponding Tables.

Thank You!