Basic Concepts #10
Output Delivery System

JC Wang

Western Michigan University
Department of Statistics
Outline

1. Introduction
   - Overview
   - Uses of ODS
   - General Ideas

2. Select/Exclude Objects
   - ODS SELECT
   - ODS EXCLUDE

3. Output Data Sets
   - ODS OUTPUT
ODS Overview

- ODS = Output Delivery System
- ODS allows outputs from Data Step and Procedure steps to be presented in more “useful” ways
- ODS also allows outputs of Procedure steps to be stored in SAS data sets.
ODS Overview

- ODS = Output Delivery System
  - ODS allows outputs from Data Step and Procedure steps to be presented in more “useful” ways
  - ODS also allows outputs of Procedure steps to be stored in SAS data sets.
ODS Overview

- ODS = Output Delivery System
- ODS allows outputs from Data Step and Procedure steps to be presented in more “useful” ways
- ODS also allows outputs of Procedure steps to be stored in SAS data sets.
ODS Overview

- ODS = Output Delivery System
- ODS allows outputs from Data Step and Procedure steps to be presented in more “useful” ways
- ODS also allows outputs of Procedure steps to be stored in SAS data sets.
Uses of ODS

- ODS can arrange outputs in “prettier” ways
- It can also create outputs in a variety of formats such as: html, pdf, rtf, ps, etc.
- It can also create SAS data sets that generally can also be created within most of the SAS procedures.
Uses of ODS

- ODS can arrange outputs in “prettier” ways
- It can also create outputs in a variety of formats such as: html, pdf, rtf, ps, etc.
- It can also create SAS data sets that generally can also be created within most of the SAS procedures.
Uses of ODS

- ODS can arrange outputs in “prettier” ways
- It can also create outputs in a variety of formats such as: html, pdf, rtf, ps, etc.
- It can also create SAS data sets that generally can also be created within most of the SAS procedures.
Uses of ODS

- ODS can arrange outputs in “prettier” ways
- It can also create outputs in a variety of formats such as: html, pdf, rtf, ps, etc.
- It can also create SAS data sets that generally can also be created within most of the SAS procedures.
General Ideas of ODS

- Prior to version 7, SAS output was the results of a SAS procedure and was stored in a basic output file (or output window). In addition, some commonly used procedures do not produce output data sets and have to rely on Proc PrintTo and data steps for this purpose.

- ODS now creates output objects basically in 3 parts: data component, table definition (order of columns, rows, etc.), and an output destination (html, rtf, etc.)
Prior to version 7, SAS output was the results of a SAS procedure and was stored in a basic output file (or output window). In addition, some commonly used procedures do not produce output data sets and have to rely on PROC PRINTTO and data steps for this purpose.

ODS now creates output objects basically in 3 parts: data component, table definition (order of columns, rows, etc.), and an output destination (html, rtf, etc.)
General Ideas of ODS

- Prior to version 7, SAS output was the results of a SAS procedure and was stored in a basic output file (or output window). In addition, some commonly used procedures do not produce output data sets and have to rely on PROC PRINTTO and data steps for this purpose.

- ODS now creates output objects basically in 3 parts: data component, table definition (order of columns, rows, etc.), and an output destination (html, rtf, etc.)
To start output being delivered by ODS the general syntax is:
ods output-format <options>;

To end output being delivered by ODS:
ods output-format close;
General Ideas of ODS
continued

- To start output being delivered by ODS the general syntax is:
  ods output-format <options>;
- To end output being delivered by ODS:
  ods output-format close;
General Ideas of ODS
continued

- To start output being delivered by ODS the general syntax is:
  ods output-format <options> ;
- To end output being delivered by ODS:
  ods output-format close;
Example: html destination

To send the output of a procedure to html destination (in frame set):

```sas
ods html file = "body.html"
   contents = "contents.html"
   frame = "frame.html";
proc print data=mydata; run;
ods html close;
```

See example Array.sas.
Example: html destination

To send the output of a procedure to html destination (in frame set):

```sas
ods html file = "body.html"
    contents = "contents.html"
    frame = "frame.html";
    proc print data=mydata; run;
ods html close;
```

See example Array.sas.
Outline

1. Introduction
   - Overview
   - Uses of ODS
   - General Ideas

2. Select/Exclude Objects
   - ODS SELECT
   - ODS EXCLUDE

3. Output Data Sets
   - ODS OUTPUT
Selecting Tables

ODS <ODS-destination> SELECT table(s);
where table is one of

1. output-object or output-object (PERSIST)
2. keyword ALL
3. keyword NONE
Selecting Tables, continued
getting a list of output object

Use ODS TRACE statement. Example:

```
ODS TRACE ON; * ODS TRACE ON / LABEL LISTING;
PROC UNIVARIATE DATA=mydata;
   VAR x y;
RUN;
ODS TRACE OFF;
```

See examples ODS2.sas & ODS1.sas.
Selecting Tables, continued

getting a list of output object

Use ODS TRACE statement. Example:

```sas
ODS TRACE ON;   * ODS TRACE ON / LABEL LISTING;
   PROC UNIVARIATE DATA=mydata;
   VAR x y;
   RUN;
   ODS TRACE OFF;
```

See examples ODS2.sas & ODS1.sas.
Selecting Tables, continued

specifying output object by one of

- full/partial path
- label
- full/partial label path
- mixture of labels and paths
- partial-path\(n\) (e.g. TestsForLocation\#3 from PROC UNIVARIATE)

See examples ODS2.sas & ODS1.sas.
Selecting Tables, continued

specifying output object by one of

- full/partial path
- label
- full/partial label path
- mixture of labels and paths
- partial-path\(n\) (e.g. TestsForLocation\(n\) from PROC UNIVARIATE)

See examples ODS2.sas & ODS1.sas.
Selecting Tables, continued

specifying output object by one of

- full/partial path
- label
- full/partial label path
- mixture of labels and paths
- partial-path\#n (e.g. TestsForLocation\#3 from PROC UNIVARIATE)

See examples ODS2.sas & ODS1.sas.
Excluding Tables

ODS <ODS-destination> EXCLUDE table(s);
where table is one of

1. output-object or output-object (PERSIST)
2. keyword ALL
3. keyword NONE
Outline

1. Introduction
   - Overview
   - Uses of ODS
   - General Ideas

2. Select/Exclude Objects
   - ODS SELECT
   - ODS EXCLUDE

3. Output Data Sets
   - ODS OUTPUT
ODS OUTPUT

Use the following to produce output data set(s):

```
ODS OUTPUT table-name=SAS-dataset;
```

Example:

```
ODS _ALL_ CLOSE;
ODS OUTPUT Moments#1=x_mom Moments#2=y_mom;
   PROC UNIVARIATE DATA=mydata;
   VAR x y;
   RUN;
ODS LISTING;
```

See examples ODS2.sas & ODS1.sas.
ODS OUTPUT

Use the following to produce output data set(s):

ODS OUTPUT table-name=SAS-dataset;

Example:

```
ODS _ALL_ CLOSE;
ODS OUTPUT Moments#1=x_mom Moments#2=y_mom;
PROC UNIVARIATE DATA=mydata;
   VAR x y;
RUN;
ODS LISTING;
```

See examples ODS2.sas & ODS1.sas.