PSTAT 130

SAS BASE PROGRAMMING

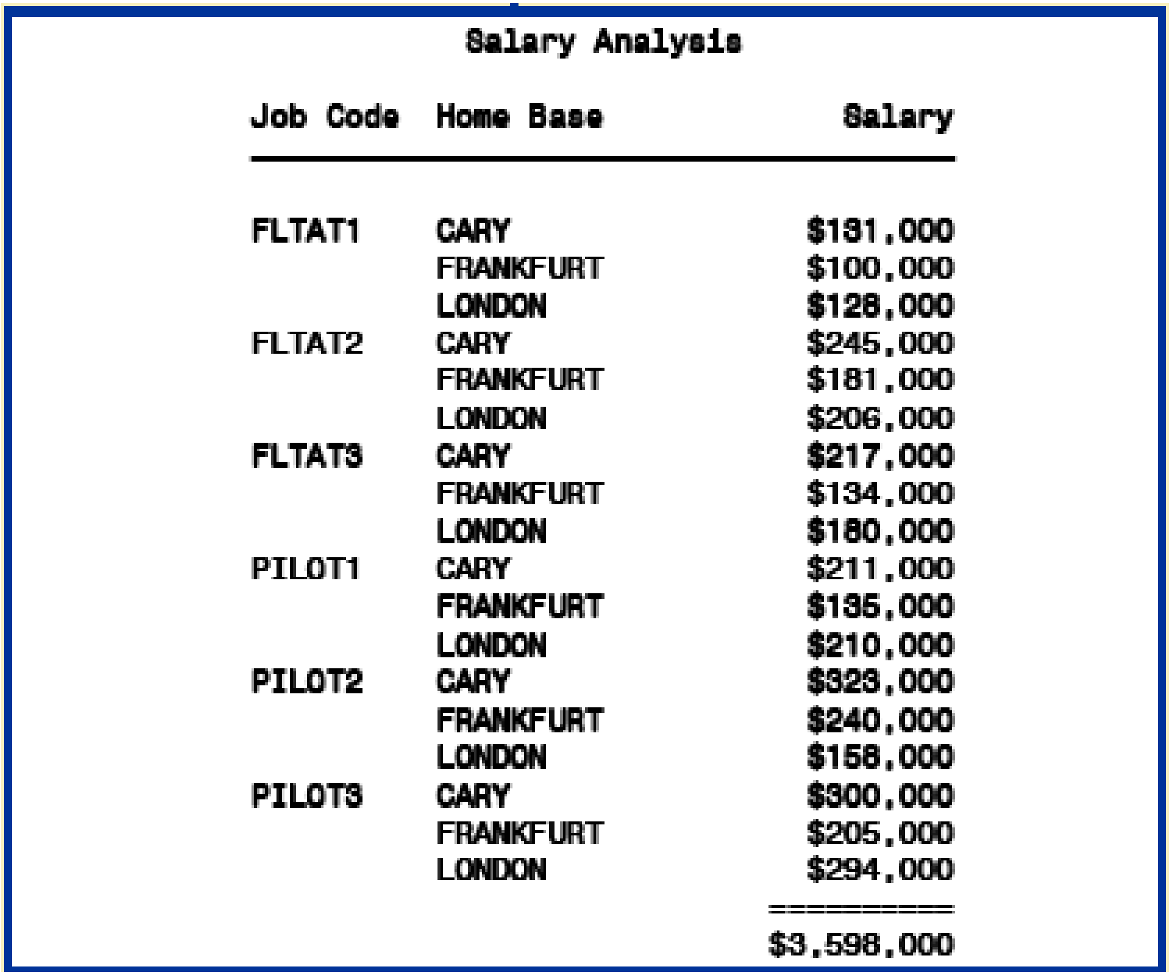
- Lecture 9 -

Dr. Julie Swenson

Objectives

- Summarize Data
  - PROC REPORT
    - Selecting variables
    - List reports
    - Define statement
    - Summary reports
  - PROC GCHART
    - Bar charts
    - Pie charts

PROC REPORT Output

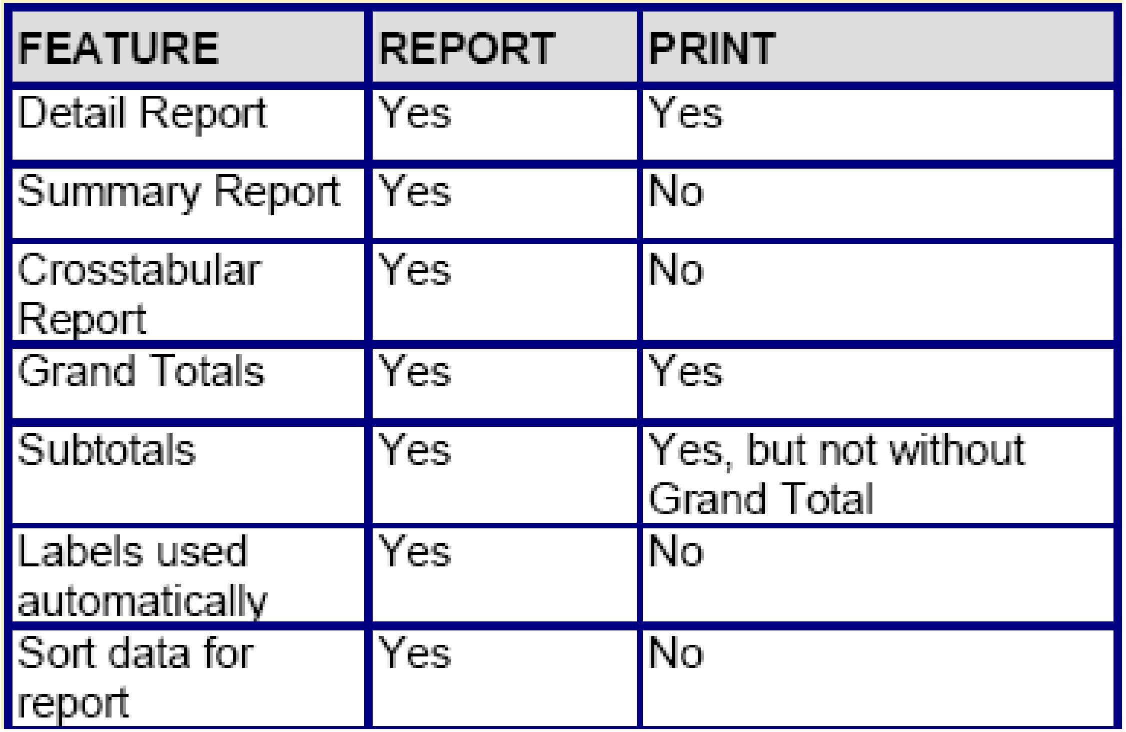


```
                 Salary Analysis

Job Code   Home Base                Salary
----------------------------------------------

FLTAT1     CARY                   $131,000
           FRANKFURT              $100,000
           LONDON                 $128,000
FLTAT2     CARY                   $245,000
           FRANKFURT              $181,000
           LONDON                 $206,000
FLTAT3     CARY                   $217,000
           FRANKFURT              $134,000
           LONDON                 $180,000
PILOT1     CARY                   $211,000
           FRANKFURT              $135,000
           LONDON                 $210,000
PILOT2     CARY                   $323,000
           FRANKFURT              $240,000
           LONDON                 $158,000
PILOT3     CARY                   $300,000
           FRANKFURT              $205,000
           LONDON                 $294,000
                                ==========
                                $3,598,000
```

The REPORT Procedure

- PROC REPORT allows you to
  - Create list reports
  - Create summary reports
  - Enhance reports
  - Request separate subtotals and grand totals

PROC REPORT vs. PROC PRINT



FEATURE	REPORT	PRINT
Detail Report	Yes	Yes
Summary Report	Yes	No
Crosstabular Report	Yes	No
Grand Totals	Yes	Yes
Subtotals	Yes	Yes, but not without Grand Total
Labels used automatically	Yes	No
Sort data for report	Yes	No

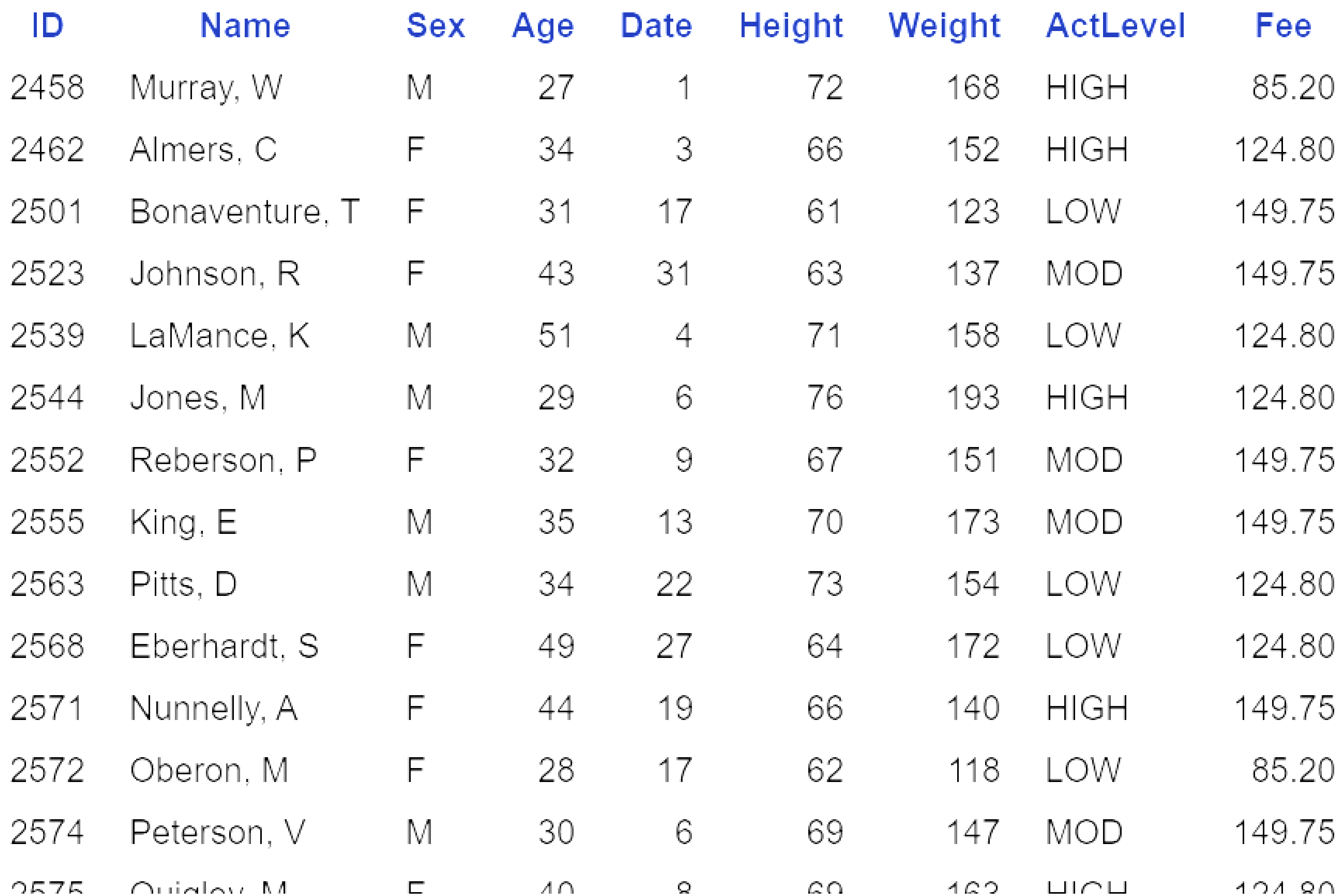
Create a List Report

- General form of a simple PROC REPORT step:

```
PROC REPORT DATA=SAS-data-set <options>;
RUN;
```

- Example:

```
proc report data=data1.admit;
run;
```



ID	Name	Sex	Age	Date	Height	Weight	ActLevel	Fee
2458	Murray, W	M	27	1	72	168	HIGH	85.20
2462	Almers, C	F	34	3	66	152	HIGH	124.80
2501	Bonaventure, T	F	31	17	61	123	LOW	149.75
2523	Johnson, R	F	43	31	63	137	MOD	149.75
2539	LaMance, K	M	51	4	71	158	LOW	124.80
2544	Jones, M	M	29	6	76	193	HIGH	124.80
2552	Reberson, P	F	32	9	67	151	MOD	149.75
2555	King, E	M	35	13	70	173	MOD	149.75
2563	Pitts, D	M	34	22	73	154	LOW	124.80
2568	Eberhardt, S	F	49	27	64	172	LOW	124.80
2571	Nunnelly, A	F	44	19	66	140	HIGH	149.75
2572	Oberon, M	F	28	17	62	118	LOW	85.20
2574	Peterson, V	M	30	6	69	147	MOD	149.75
2575	Quigley, M	F	40	8	69	163	HIGH	124.80

The REPORT Procedure

- The default list report displays
  - Each data value as it is stored in the data set, or formatted values if a format is stored with the data
  - Variable names or labels as report column headings
  - A default width for the report columns
  - Character values left-justified
  - Numeric values right-justified
  - Observations in the order in which they are stored in the data set

Print Selected Variables

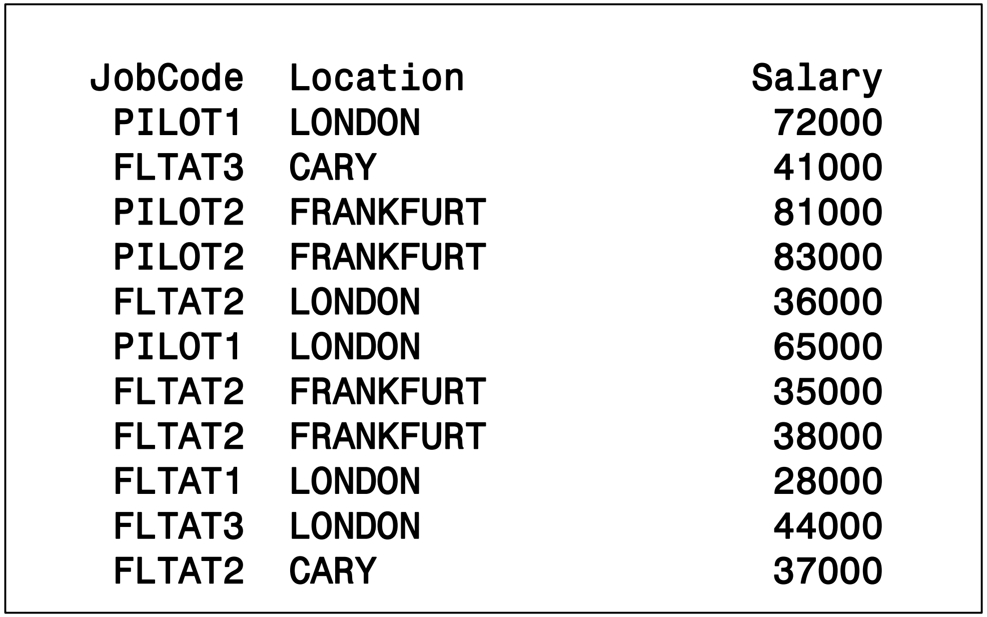
- General form of the COLUMN statement:

```
COLUMN SAS-variables;
```

Sample List Report

```
title 'Salary Analysis';
proc report data=data1.crew;
   column JobCode Location Salary;
run;
```

Partial SAS Output



```
JobCode  Location                    Salary
 PILOT1  LONDON                       72000
 FLTAT3  CARY                         41000
 PILOT2  FRANKFURT                    81000
 PILOT2  FRANKFURT                    83000
 FLTAT2  LONDON                       36000
 PILOT1  LONDON                       65000
 FLTAT2  FRANKFURT                    35000
 FLTAT2  FRANKFURT                    38000
 FLTAT1  LONDON                       28000
 FLTAT3  LONDON                       44000
 FLTAT2  CARY                         37000
```

The DEFINE Statement

- You can enhance the report by using DEFINE statements to
  - Define how each variable is used in the report
  - Assign formats to variables
  - Specify report column headers and column widths
  - Change the order of the rows in the report.

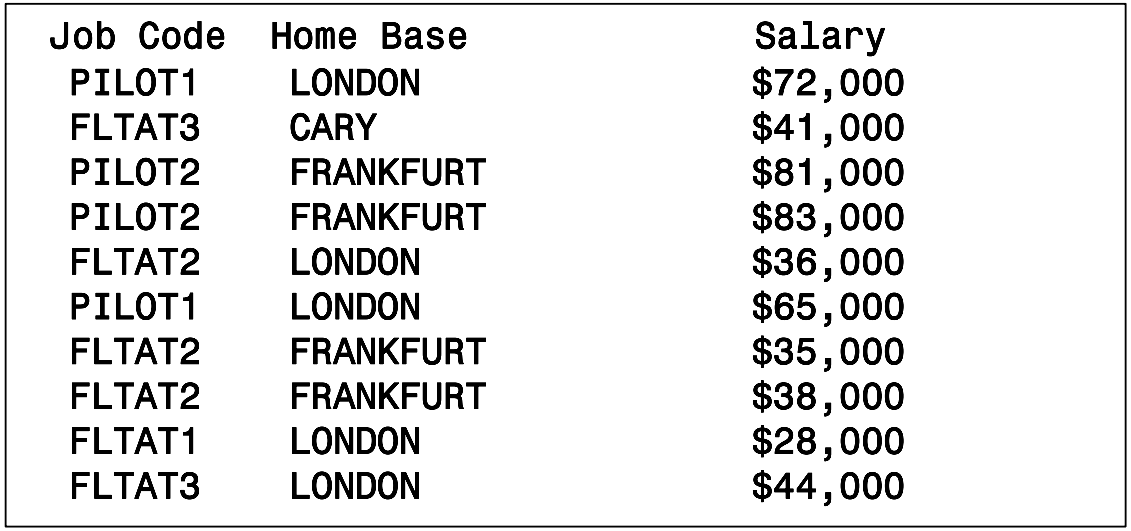
The DEFINE Statement: Options

- **DISPLAY** option specifies a variable to be a display variable.
  - Character variables are assumed to be display variables
- **ANALYSIS** option specifies a variable to be an analysis variable
  - Numeric variables are assumed to be analysis variables
- **FORMAT=** option assigns a format to a variable
- **WIDTH=** option controls width of report column
- **'report-column-header'** defines the column header for a variable (similar to a label)

Enhanced List Report

```
proc report data=data1.crew;
   column JobCode Location Salary;
   define JobCode / width=8 'Job Code';
   define Location / 'Home Base';
   define Salary / format=dollar10.;
run;
```

Partial SAS Output



```
Job Code  Home Base               Salary
 PILOT1    LONDON                $72,000
 FLTAT3    CARY                  $41,000
 PILOT2    FRANKFURT             $81,000
 PILOT2    FRANKFURT             $83,000
 FLTAT2    LONDON                $36,000
 PILOT1    LONDON                $65,000
 FLTAT2    FRANKFURT             $35,000
 FLTAT2    FRANKFURT             $38,000
 FLTAT1    LONDON                $28,000
 FLTAT3    LONDON                $44,000
```

ORDER Usage Type

- **ORDER** option identifies the variable(s) used to order the rows of the report

```
proc report data=data1.crew;
   column JobCode Location Salary;
   define JobCode / order width=8 'Job Code';
   define Location / 'Home Base';
   define Salary / format=dollar10.;
run;
```

ORDER Usage Type

Partial SAS Output

```
Job Code  Home Base           Salary
  FLTAT1    LONDON             $28,000
            FRANKFURT          $25,000
            CARY               $23,000
            CARY               $21,000
            CARY               $28,000
            FRANKFURT          $22,000
            LONDON             $29,000
  FLTAT2    LONDON             $36,000
            FRANKFURT          $35,000
            FRANKFURT          $38,000
            CARY               $37,000
            CARY               $34,000
            LONDON             $34,000
            CARY               $36,000
  FLTAT3    CARY               $41,000
            LONDON             $44,000
            FRANKFURT          $48,000
            FRANKFURT          $45,000
            CARY               $44,000
```

Individual Values (not Summary Stats)

Define Group Variables

- Use the REPORT procedure to create a summary report by defining variables as group variables.

- All observations whose group variables have the same values are collapsed into a single row in the report.

Group Variables

- If you have a group variable, there must be no display or order variables.
  - Group variables produce summary reports (observations collapsed into groups).
  - Display and order variables produce list reports (one row for each observation).

Analysis Variables

- Default usage for numeric variables is ANALYSIS with a default statistic of SUM.
  - If the report contains group variables, the report displays the sum of the analysis variables' values for each group.
  - If the report contains at least one display or order variable and no group variables, the report displays all of the values of the analysis variables.
  - If the report contains only numeric variables, the report displays grand totals for the numeric variables.

Define Analysis Variables

- Selected statistics include:

SUM	sum (default)
N	number of non-missing values
MEAN	Average
MAX	Maximum Value
MIN	Minimum Value

- Example:

```
define Salary / mean format=dollar10.;
```

Summarize the Data

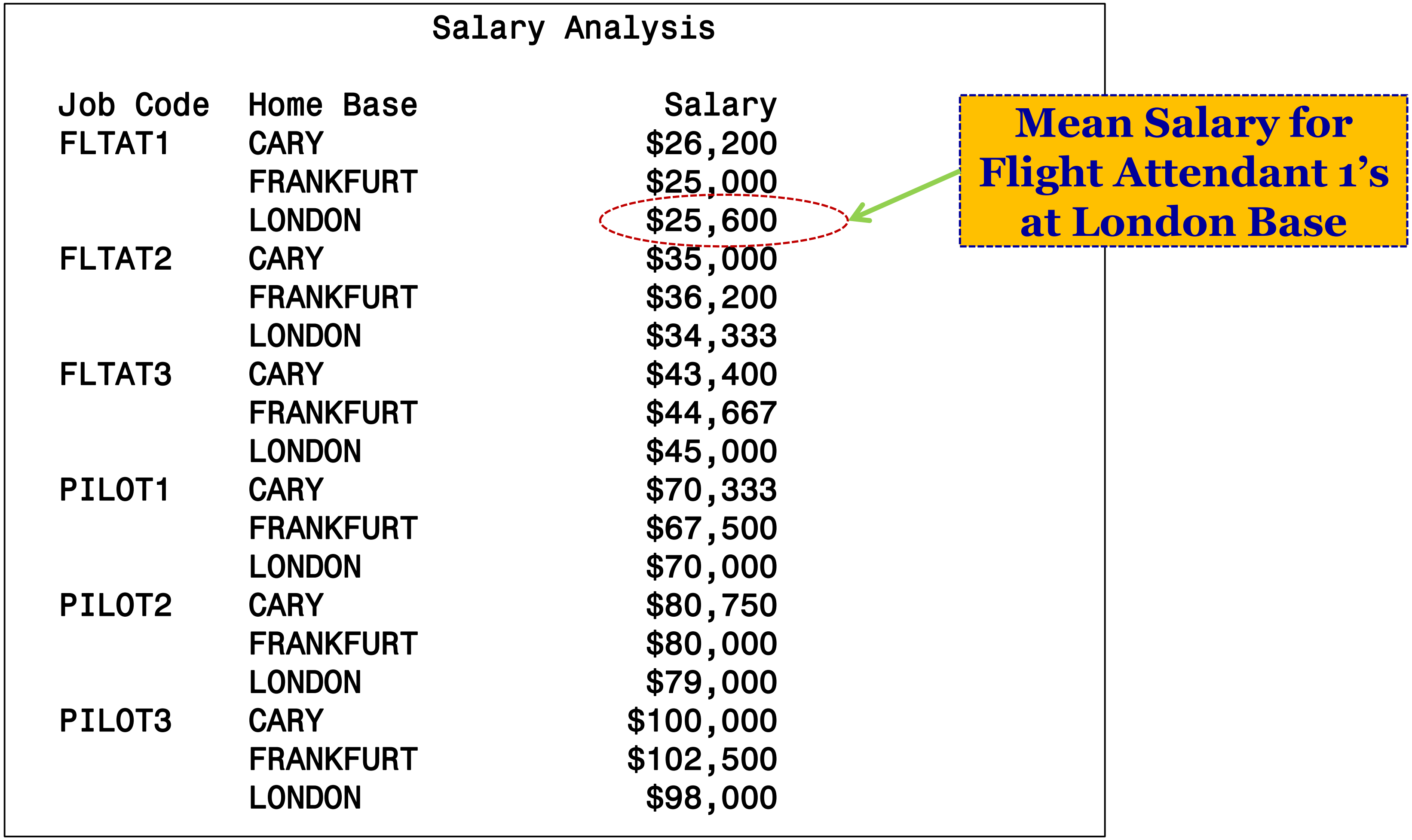
- Use the GROUP usage in the DEFINE statement to specify the variables that define groups.

- Example

```
proc report data=data1.crew;
   title 'Salary Analysis';
   column JobCode Location Salary;
   define JobCode / group width=8 'Job Code';
   define Location / group 'Home Base';
   define Salary / analysis mean format=dollar10.;
run;
```

Optional specification

Summarize the Data



```
                      Salary Analysis

Job Code   Home Base                 Salary
FLTAT1     CARY                     $26,200
           FRANKFURT                $25,000
           LONDON                   $25,600
FLTAT2     CARY                     $35,000
           FRANKFURT                $36,200
           LONDON                   $34,333
FLTAT3     CARY                     $43,400
           FRANKFURT                $44,667
           LONDON                   $45,000
PILOT1     CARY                     $70,333
           FRANKFURT                $67,500
           LONDON                   $70,000
PILOT2     CARY                     $80,750
           FRANKFURT                $80,000
           LONDON                   $79,000
PILOT3     CARY                    $100,000
           FRANKFURT               $102,500
           LONDON                   $98,000
```

Mean Salary for Flight Attendant 1's at London Base

BREAK & RBREAK Statements

- Use the BREAK statement to display a total at the beginning or end of a Group.
- Use the RBREAK statement to display the grand total at the beginning or end of the report.

Print Subtotals

- General form of the BREAK statement:

```
BREAK BEFORE|AFTER <BREAK-VARIABLE>/ <options>;
```

- Selected options:

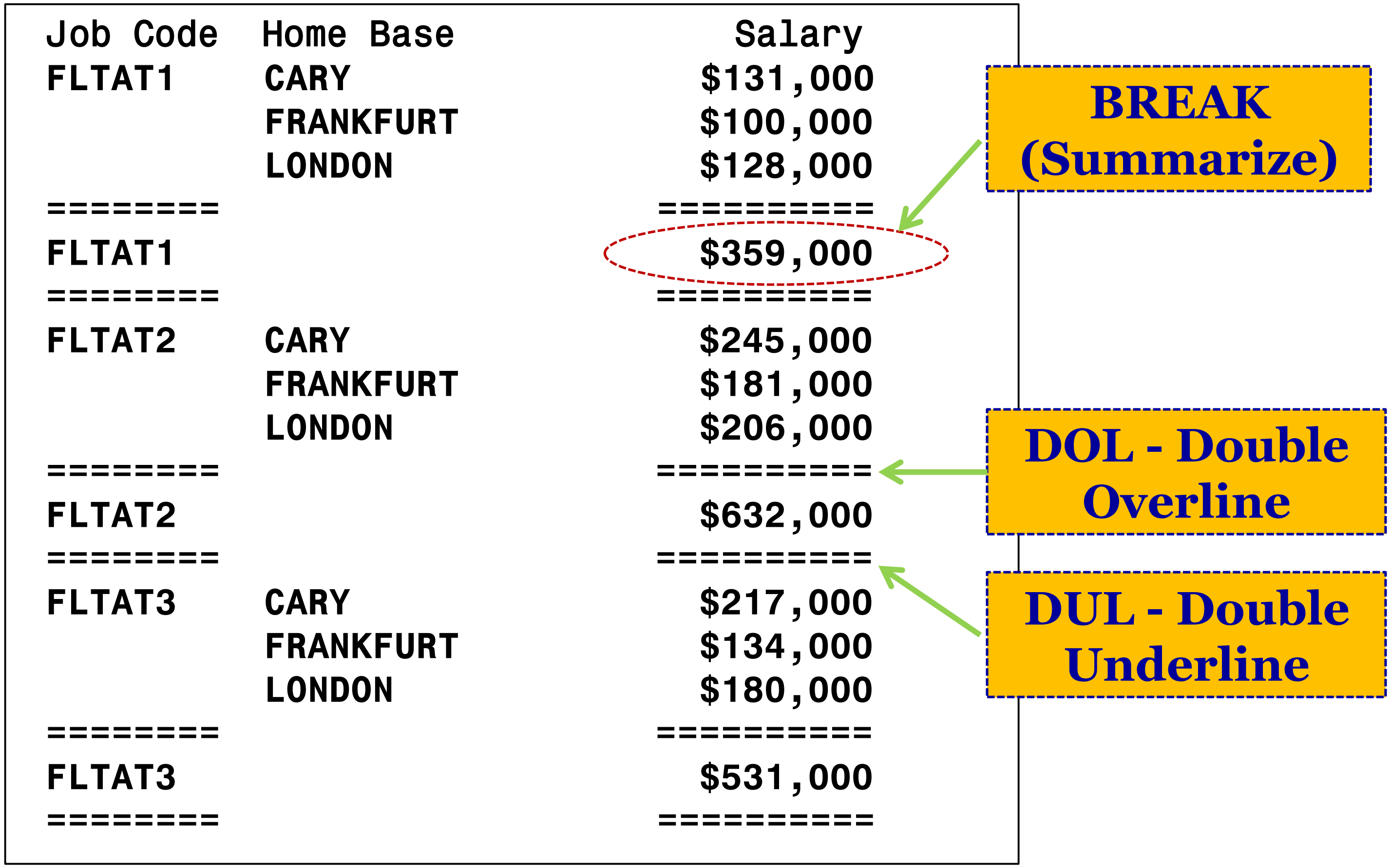
SUMMARIZE	Prints the total.
OL	Prints a single line above the total.
DOL	Prints a double line above the total.
UL	Prints a single line below the total.
DUL	Prints a double line below the total.

BREAK Example

- Example

```
proc report data=data1.crew;
   column JobCode Location Salary;
   define JobCode / group width=8 'Job Code';
   define Location / group 'Home Base';
   define Salary / format=dollar10.;
   break after JobCode / summarize dol dul;
run;
```

BREAK Example Output



```
Job Code   Home Base               Salary
FLTAT1     CARY                  $131,000
           FRANKFURT             $100,000
           LONDON                $128,000
========                       ==========
FLTAT1                           $359,000
========                       ==========
FLTAT2     CARY                  $245,000
           FRANKFURT             $181,000
           LONDON                $206,000
========                       ==========
FLTAT2                           $632,000
========                       ==========
FLTAT3     CARY                  $217,000
           FRANKFURT             $134,000
           LONDON                $180,000
========                       ==========
FLTAT3                           $531,000
========                       ==========
```

BREAK (Summarize)

DOL - Double Overline

DUL - Double Underline

Print Grand Totals

- General form of the RBREAK statement:

```
RBREAK BEFORE|AFTER / <options>;
```

- Selected options:

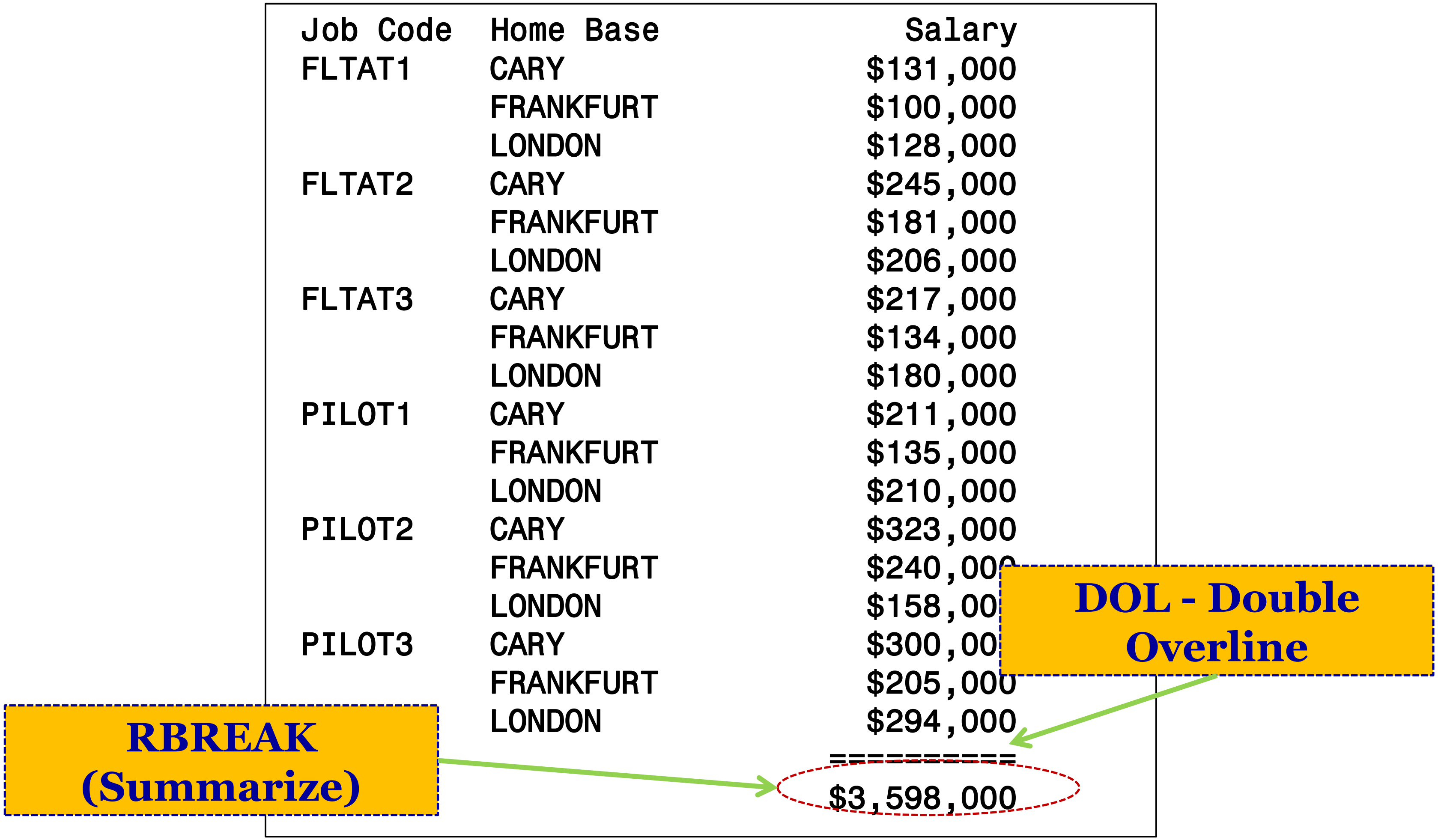
SUMMARIZE	Prints the total.
OL	Prints a single line above the total.
DOL	Prints a double line above the total.
UL	Prints a single line below the total.
DUL	Prints a double line below the total.

RBREAK Example

- Example

```
proc report data=data1.crew;
   column JobCode Location Salary;
   define JobCode / group width=8 'Job Code';
   define Location / group 'Home Base';
   define Salary / format=dollar10.;
   rbreak after / summarize dol;
run;
```

RBREAK Example Output



```
Job Code   Home Base            Salary
FLTAT1     CARY               $131,000
           FRANKFURT          $100,000
           LONDON             $128,000
FLTAT2     CARY               $245,000
           FRANKFURT          $181,000
           LONDON             $206,000
FLTAT3     CARY               $217,000
           FRANKFURT          $134,000
           LONDON             $180,000
PILOT1     CARY               $211,000
           FRANKFURT          $135,000
           LONDON             $210,000
PILOT2     CARY               $323,000
           FRANKFURT          $240,000
           LONDON             $158,00
PILOT3     CARY               $300,00
           FRANKFURT          $205,000
           LONDON             $294,000
                            ==========
                            $3,598,000
```

DOL - Double Overline
RBREAK (Summarize)

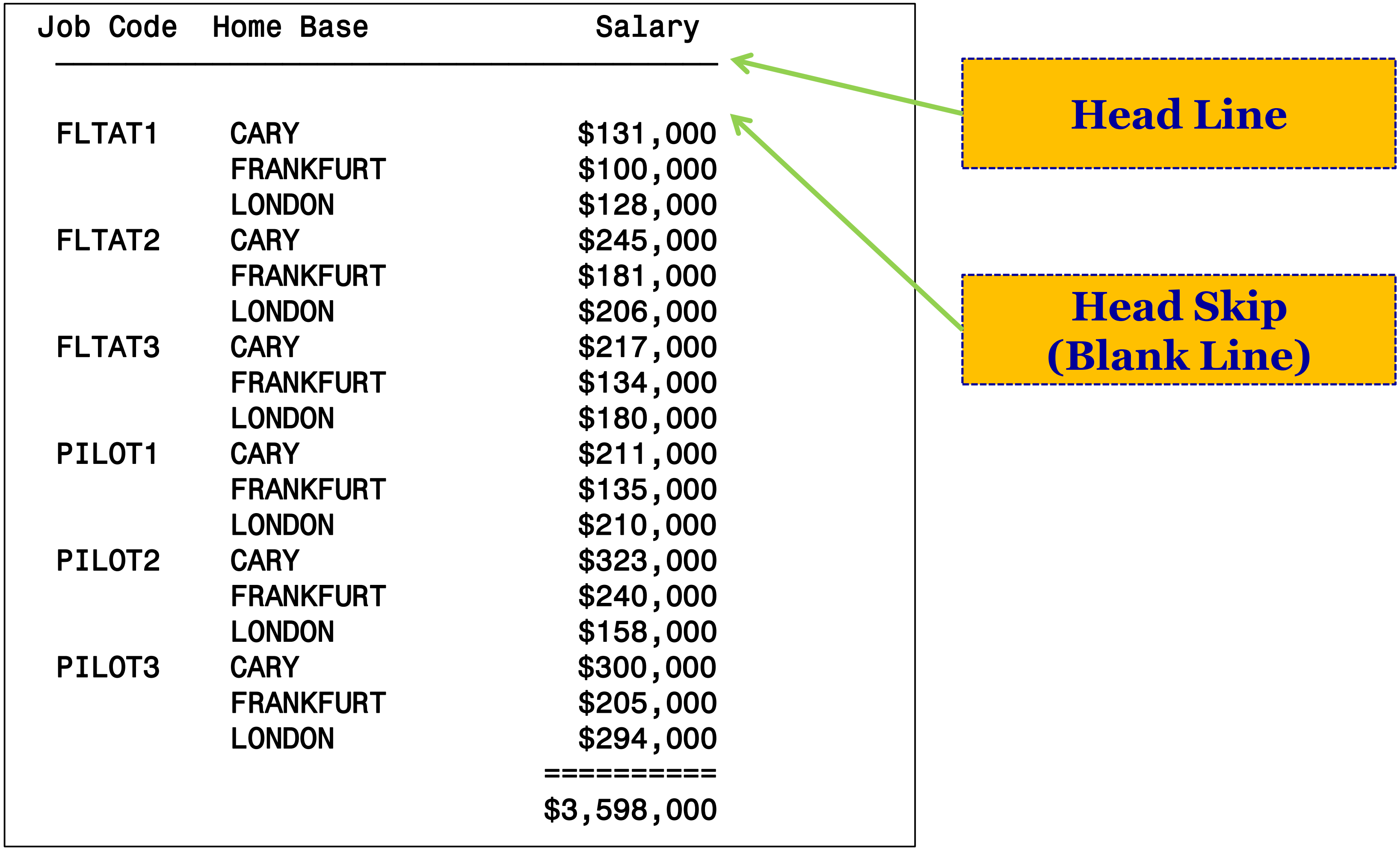
Enhance the Report

- We can further enhance our reports with the `headline` and `headskip` options
  - `Headline` places a line underneath the column headers
  - `Headskip` places a blank line underneath the column headers

- Example

```
proc report data=data1.crew headline headskip;
   column JobCode Location Salary;
   define JobCode / group width=8 'Job Code';
   define Location / group 'Home Base';
   define Salary / format=dollar10.;
   rbreak after / summarize dol;
run;
```

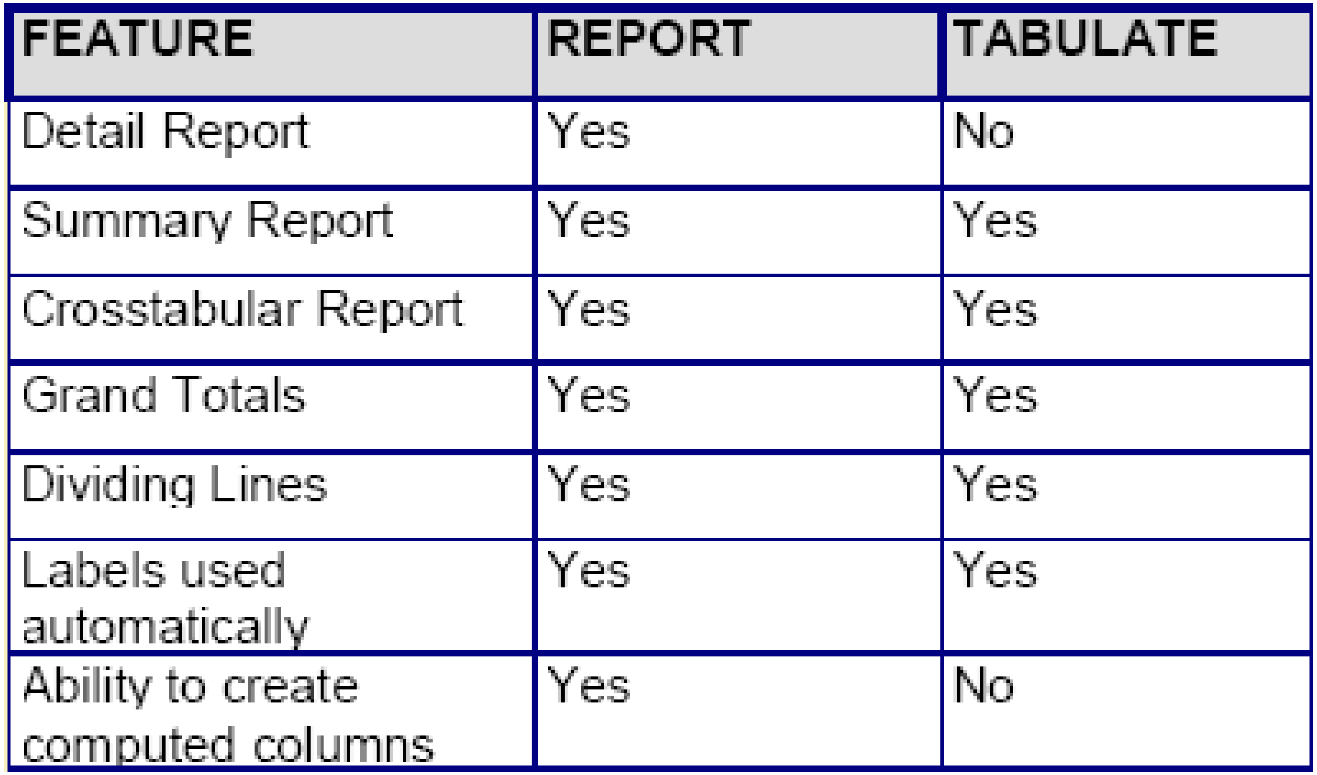
Example Output



```
 Job Code   Home Base             Salary
 ---------------------------------------


  FLTAT1     CARY                $131,000
             FRANKFURT           $100,000
             LONDON              $128,000
  FLTAT2     CARY                $245,000
             FRANKFURT           $181,000
             LONDON              $206,000
  FLTAT3     CARY                $217,000
             FRANKFURT           $134,000
             LONDON              $180,000
  PILOT1     CARY                $211,000
             FRANKFURT           $135,000
             LONDON              $210,000
  PILOT2     CARY                $323,000
             FRANKFURT           $240,000
             LONDON              $158,000
  PILOT3     CARY                $300,000
             FRANKFURT           $205,000
             LONDON              $294,000
                               ==========
                               $3,598,000
```

Head Line
Head Skip
(Blank Line)

PROC REPORT vs. PROC TABULATE



FEATURE	REPORT	TABULATE
Detail Report	Yes	No
Summary Report	Yes	Yes
Crosstabular Report	Yes	Yes
Grand Totals	Yes	Yes
Dividing Lines	Yes	Yes
Labels used automatically	Yes	Yes
Ability to create computed columns	Yes	No

Summarize with Charts

- In the GCHART procedure, you can do the following:
  - Specify the physical form of the chart.
  - Specify a chart variable that determines the number of bars or pie slices to create.
  - Optionally specify an analysis variable to use for calculating statistics that determine the height (or length) of the bar or the size of the slice.

The GCHART Procedure

- General form of the PROC GCHART statement:

```
PROC GCHART DATA=SAS-data-set;
RUN;
```

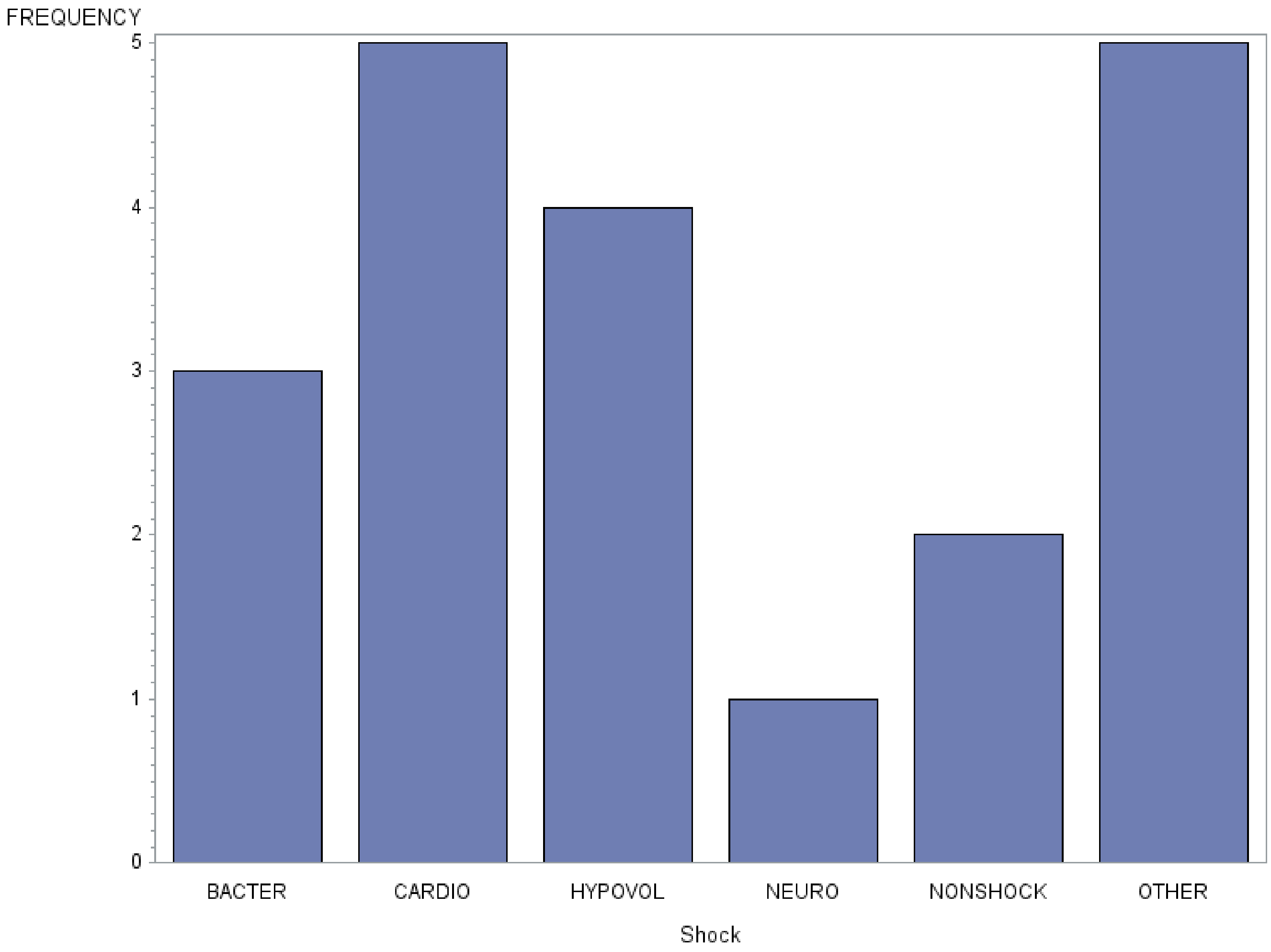
- Use one of these statements to specify the desired type of chart:

```
VBAR chart-variable . . . </options>;

HBAR chart-variable . . . </options>;

PIE chart-variable . . . </options>;
```

Chart Variable

- The chart variable
  - Determines the number of bars or slices produced within a graph
  - Can be character or numeric

Vertical Bar Chart

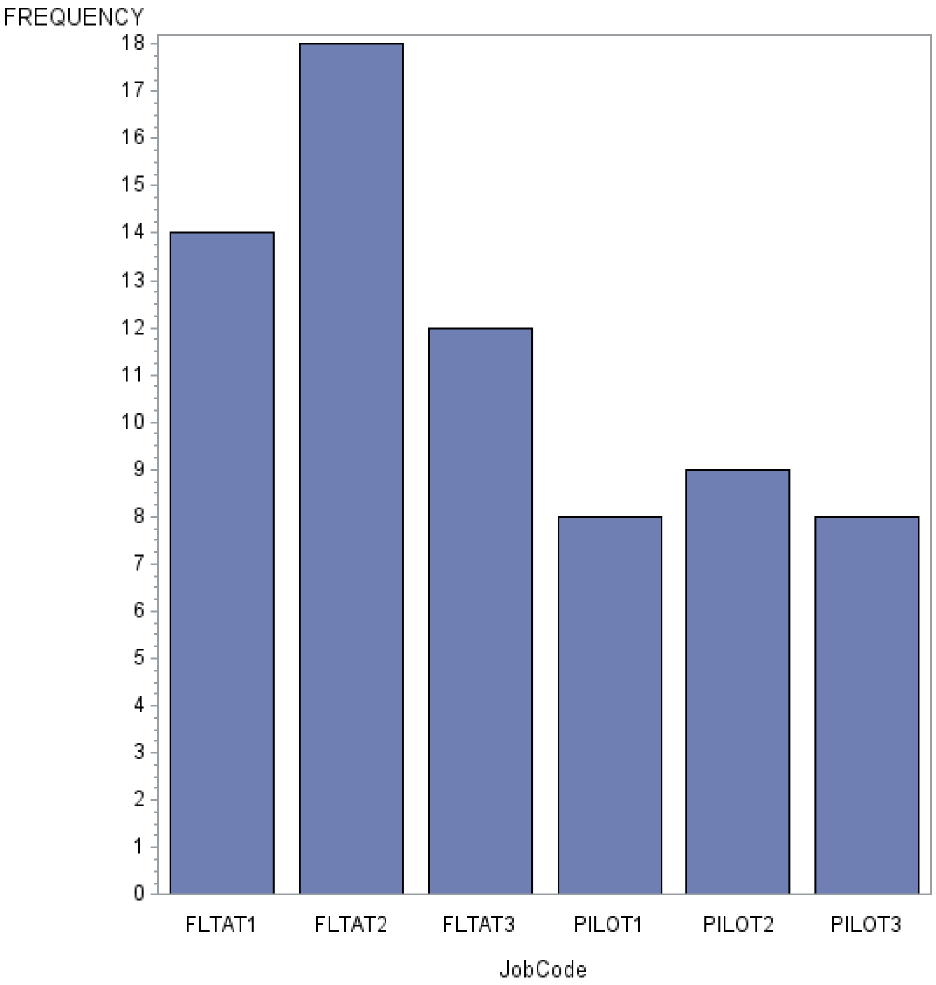


Vertical Bar Chart Example

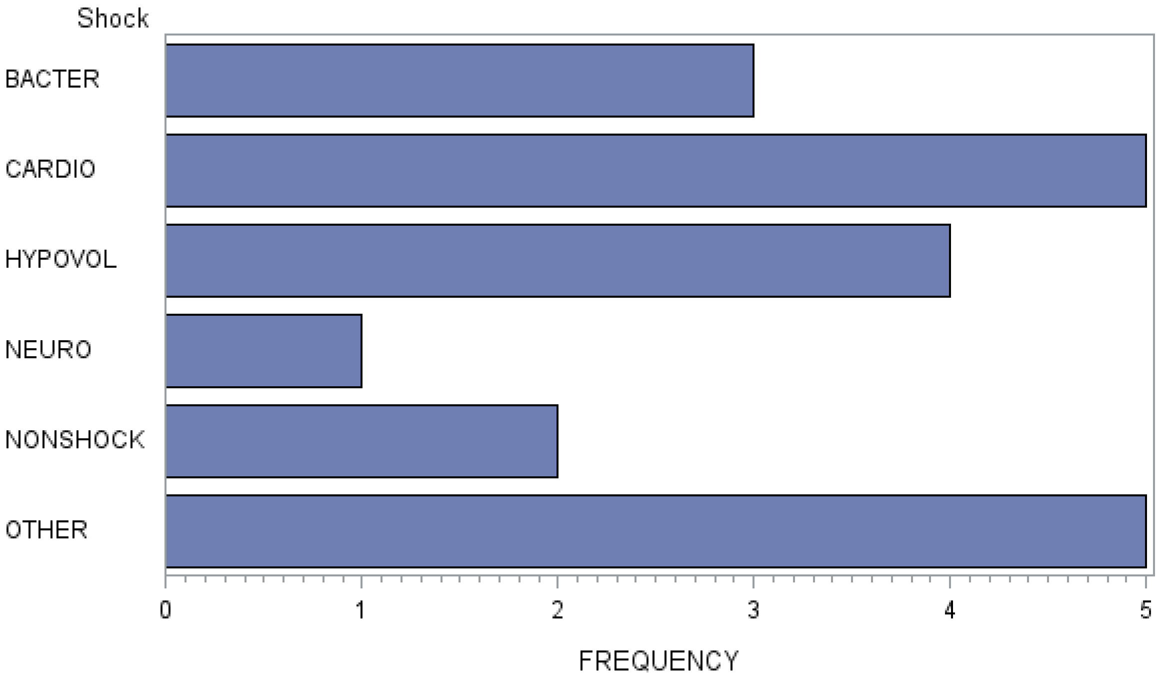
- Example
  - Using the `crew` data set in data1, produce a vertical bar chart that displays the number of employees in each job code.
  - Solution

```
proc gchart data=data1.crew;
   vbar JobCode;
run;
```

Vertical Bar Chart Example Output



Horizontal Bar Chart



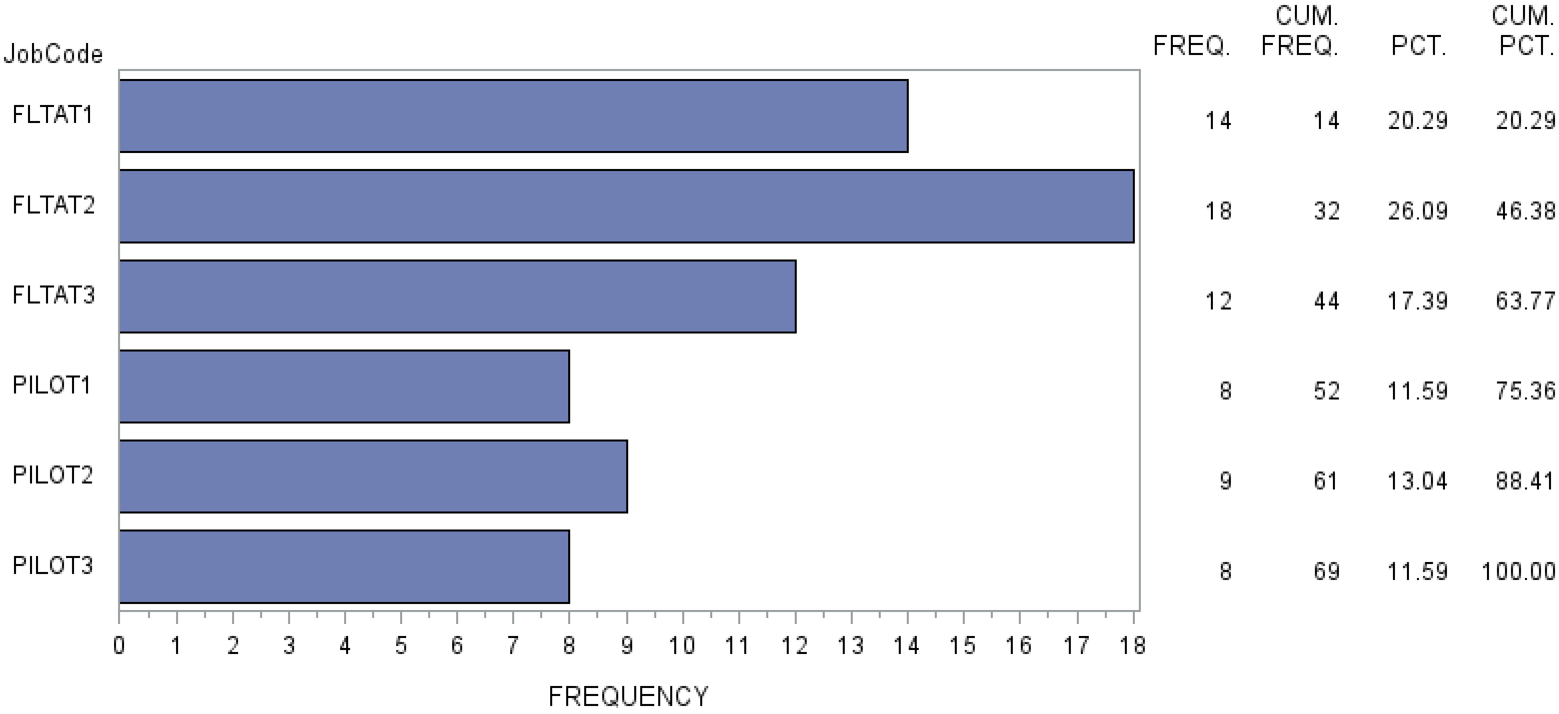
FREQ.	CUM. FREQ.	PCT.	CUM. PCT.
3	3	15	15
5	8	25	40
4	12	20	60
1	13	5	65
2	15	10	75
5	20	25	100

Horizontal Bar Chart

- Example
  - Using the `crew` data set in data1, produce a horizontal bar chart that displays the number of employees in each job code.
  - Solution

```
proc gchart data=data1.crew;
   hbar JobCode;
run;
```

Horizontal Bar Chart



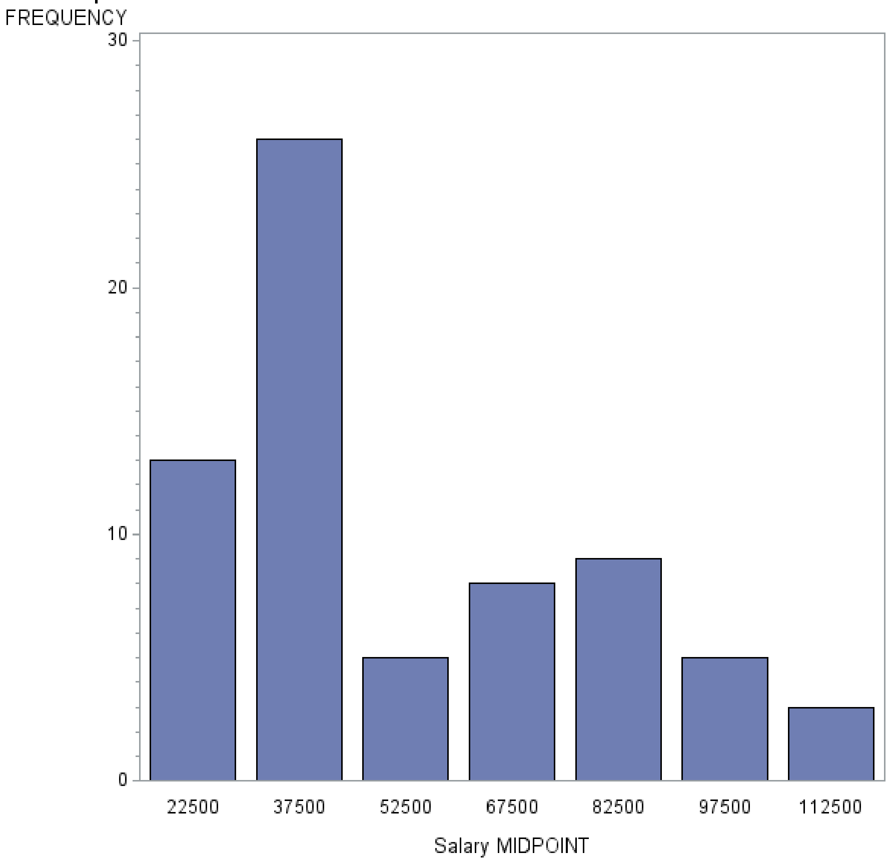
JobCode	FREQ.	CUM. FREQ.	PCT.	CUM. PCT.
FLTAT1	14	14	20.29	20.29
FLTAT2	18	32	26.09	46.38
FLTAT3	12	44	17.39	63.77
PILOT1	8	52	11.59	75.36
PILOT2	9	61	13.04	88.41
PILOT3	8	69	11.59	100.00

Chart Numeric Variables

- GCHART can graph character or numeric variables.
- When graphing numeric variables, by default, the values are grouped together to create a smaller number of lines.
- To override the default behavior for numeric chart variables, use the DISCRETE option in the VBAR, HBAR, or PIE statement.

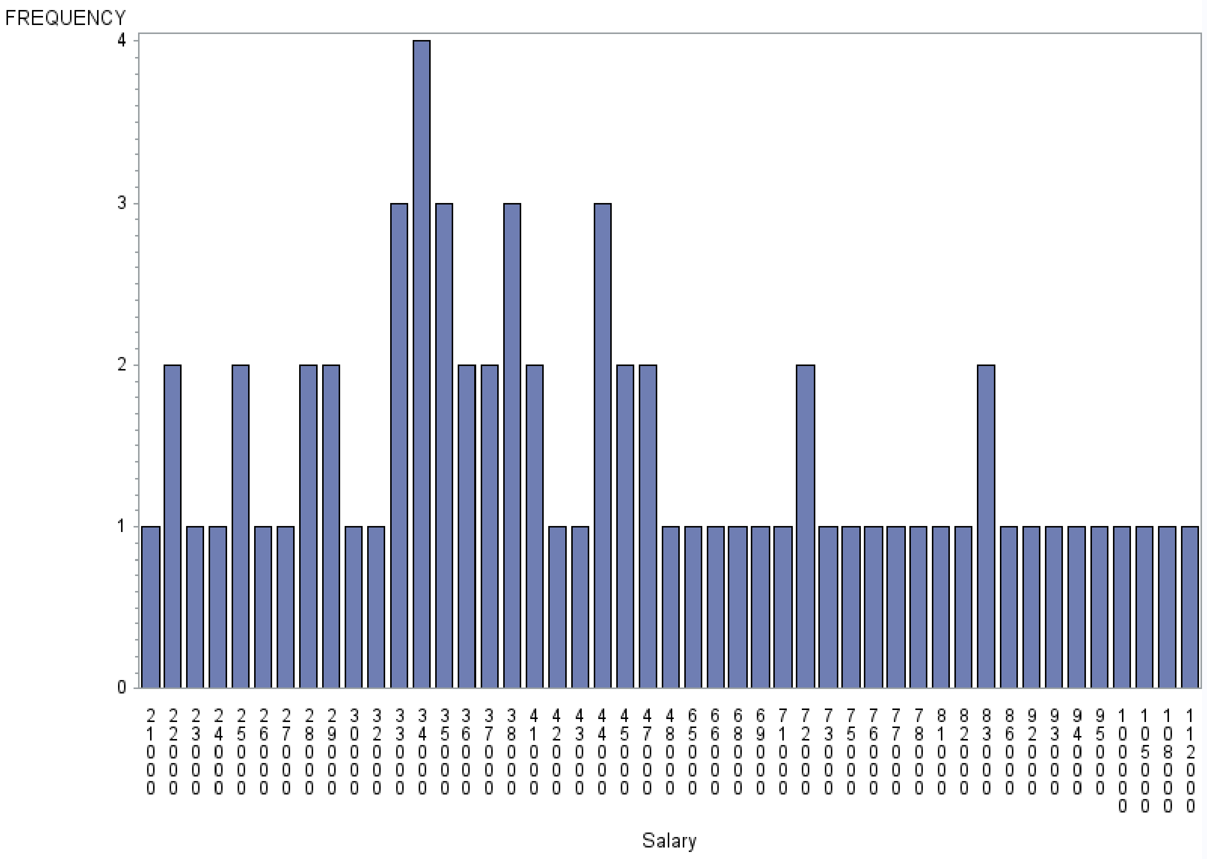
Numeric Chart: Default Grouping

```
proc gchart data=data1.crew;
  vbar salary;
run;
```



Numeric Chart: DISCRETE Option

```
proc gchart data=data1.crew;
  vbar salary / discrete;
run;
```



Analysis Variables

- An analysis variable can be used to chart summary statistics.

- Example:
  - Produce a vertical bar chart that displays the average salary of employees in each job code.

```
proc gchart data=data1.crew;
   vbar JobCode / sumvar=Salary type=mean;
run;
```

Chart An Analysis Variable

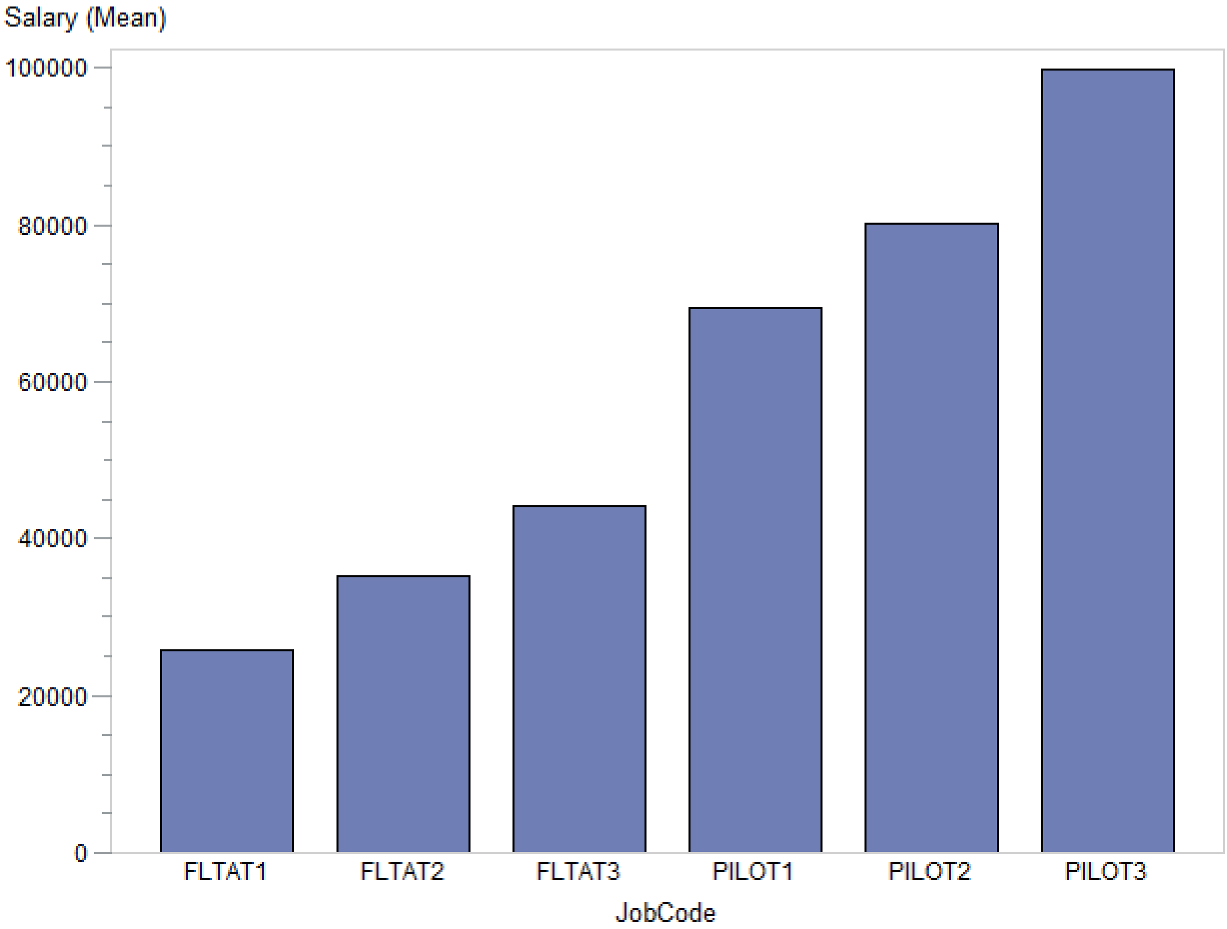
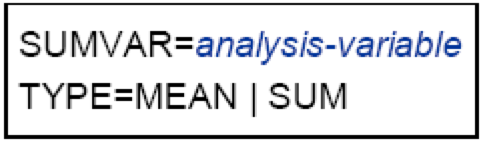


Chart Summary Statistics

- You can chart a summary statistic (e.g., mean) using the SUMVAR= and TYPE= options

- General form:

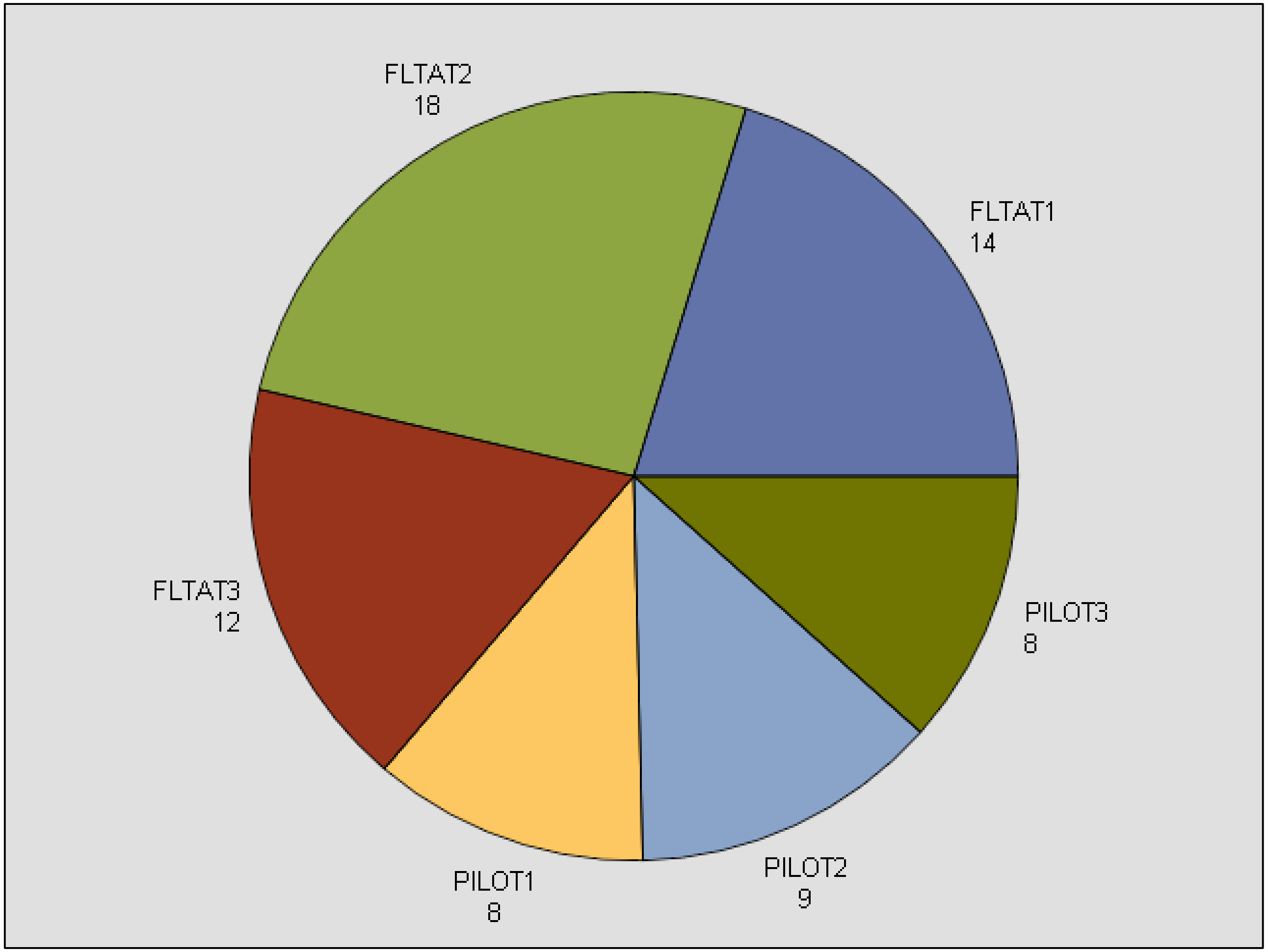


```
SUMVAR=analysis-variable
TYPE=MEAN | SUM
```

- Description:

SUMVAR =	specifies the analysis variable to use for the sum or mean calculation.
TYPE=	specifies that the height or length of the bar or size of the slice represents a mean or sum of the *analysis-variable values*.

Pie Chart

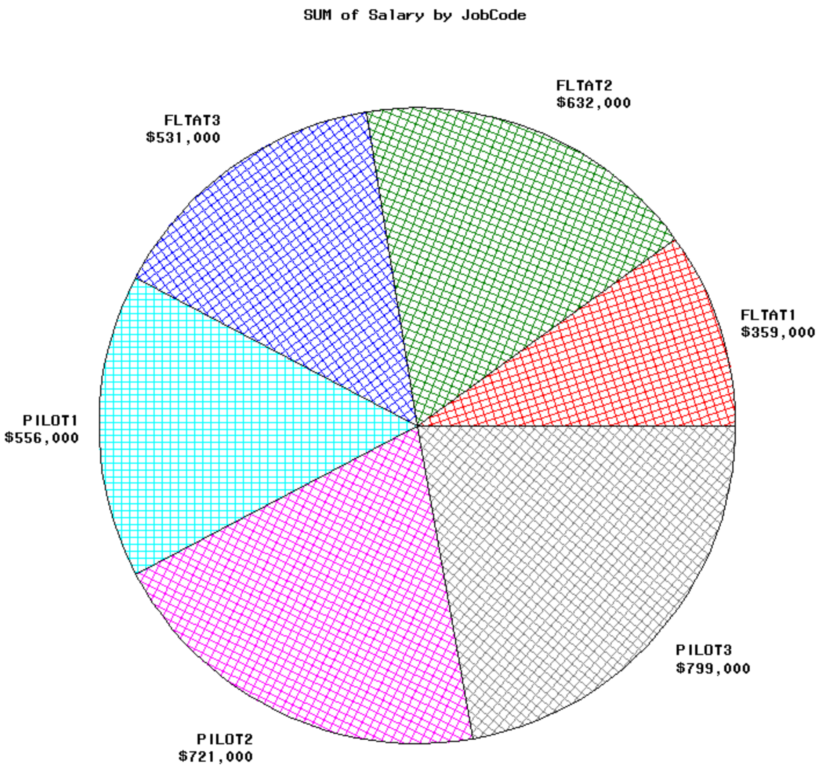


Pie Chart Example

- Example
  - Use the `crew` data set in the data1 folder
  - Produce a pie chart that displays the sum of the salaries for each job code
  - Format the salaries appropriately
  - Fill the pie chart with a grid pattern

Fill Option

```
proc gchart data=data1.crew;
   pie JobCode / sumvar=Salary
                 type=sum
                 fill=x;
   format Salary dollar8.;
run;
```

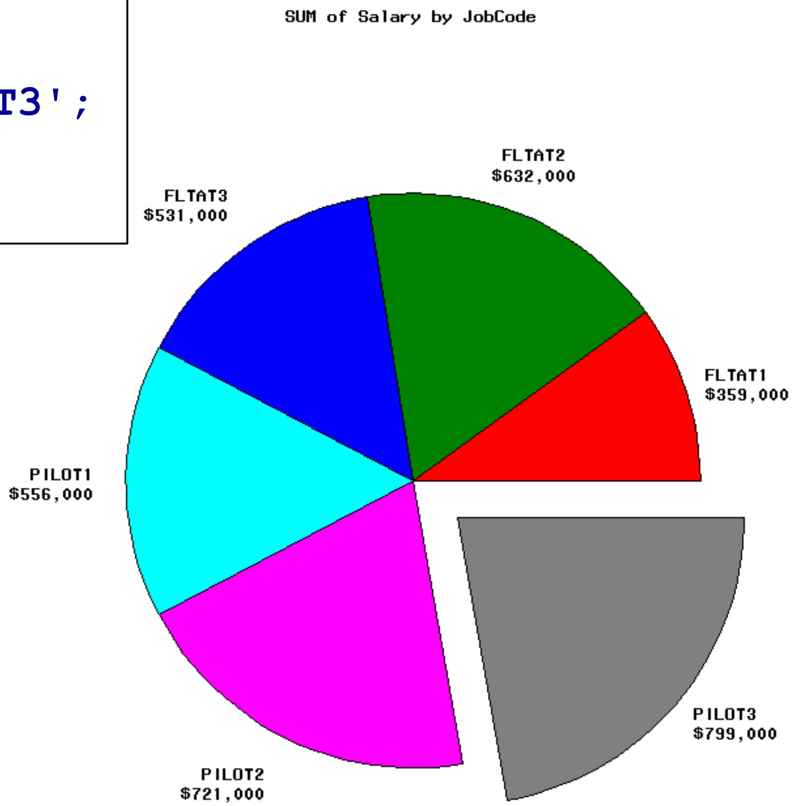


Pie Chart Example

- Example
  - Use the `crew` data set in the data1 folder
  - Produce a pie chart that displays sum of the salaries for each job code
  - Format the salaries appropriately
  - Fill the pie chart with a solid pattern
  - Bring the slice for 'PILOT3' out to emphasize it

Explode Option

```
proc gchart data=data1.crew;
   pie JobCode / sumvar=Salary
                 type=sum
                 explode = 'PILOT3';
   format Salary dollar8.;
run;
```



Class Exercise 1

- Use the `employees` data set in the data1 folder
  - Create a report using the REPORT procedure
    - Display the following variables
      - `Country`
      - `LastName`
      - `Salary`
    - Display the observations sorted by `Country`
    - Change the column header of `LastName` to 'Last Name'
    - Format the `Salary` appropriately

Class Exercise 2

- Use the `heart` data set in the data1 folder
  - Use the REPORT procedure
  - Display the following variables:
    - `Survive`
    - `Sex`
    - `Arterial`
  - Create a summary report grouped by `Survive` and `Sex` for the mean `Arterial` values
  - Use the following column headers
    - Survived
    - Gender
    - Mean Arterial Pressure

Class Exercise 2 - continued

- Format the `Sex` variable as follows:
  - 1=Female
  - 2=Male
- Keep 2 decimal places for the mean arterial values
- Specify a column width of 8 for `Survive` and `Arterial`
- Place a header line below the column headers
- Display the subtotal mean `Arterial` value at the end of each `Survive` group; place a double line above this value and a single line below this value
- Display the total mean `Arterial` value at the end of the report