Sheffield Hallam University

Mathematics and Statistics Support

https://maths.shu.ac.uk/mathshelp/

The following resources are associated:

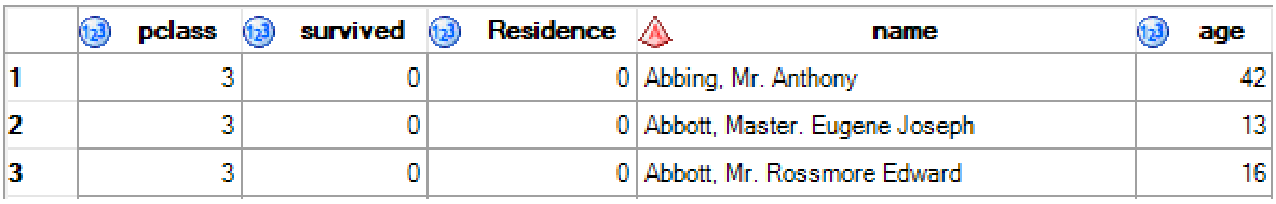
The dataset '*Titanic.csv*', 'Chi-squared test in SAS resource and 'Opening and labelling data in SAS'.

Summarising categorical variables in SAS

Dependent variable: Categorical (nominal)

Independent variable: Categorical (nominal)

Data: On April 14th 1912 the ship the Titanic sank. Information on 1309 of those on board will be used to demonstrate summarising categorical variables.



	pclass	survived	Residence	name	age
1	3	0	0	Abbing, Mr. Anthony	42
2	3	0	0	Abbott, Master. Eugene Joseph	13
3	3	0	0	Abbott, Mr. Rossmore Edward	16

Before carrying out analysis, set up formats to apply to numeric values for categorical data using the `proc format` command. Then apply these formats to any variable within procedure commands so that output displays labels rather than numbers. Here survival is coded as 0 and 1 but we want survived (1) and died (0) to appear in output. Country of Residence is American (0), British (1), Other(2).

Name for the format to be applied to variables later

```
proc format;
      value surv 1='Survived' 0='Died';
      value cla 1='1st' 2='2nd' 3='3rd';
      value nat 1='British' 0='American' 2='Other';
run;
```

Set up labels to give numbers when format used

.

Research question: Were Americans more likely to survive?

When summarising categorical data, percentages are usually preferable to frequencies although they can be misleading for very small sample sizes. Frequency tables can be produced using `proc freq` which is also used to carry out a chi-squared test, or more formatted tables can be produced using proc tabulate procedure.

`proc freq` means frequency procedure which is used to create a frequency table.
`data=Titanic` is used to determine which dataset is being used.
`Tables` creates a table with row variable before the *: `row*column` means residence (rows) compared against survived (columns). / is used to enable certain table options such as carrying out a chi-squared test (chisq).

By default, SAS will give row, column and total percentages but only one is needed. Here we want the row percentages (percentage surviving within each nationality) so use `nocol` to take out the column percentage and `nopercent` to remove the total percentages.

In the `tables` statement, put row variables before the *.

```
proc freq data=Titanic;
tables Residence*survived / nocol nopercent;
format survived surv. ;
format Residence nat. ;
```

Apply the `format nat.` to the variable `residence`
`format variableName formatName.`

This creates the summary table opposite, which looks like Americans were more likely to survive. 56% of Americans survived compared to 32% of British and 35% of other nationalities.

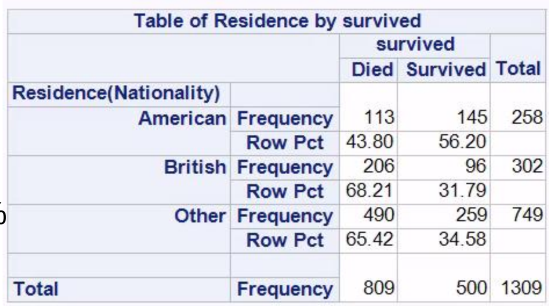


Table of Residence by survived

Residence(Nationality)		survived Died	 Survived	 Total
American	Frequency	113	145	258
	Row Pct	43.80	56.20	
British	Frequency	206	96	302
	Row Pct	68.21	31.79	
Other	Frequency	490	259	749
	Row Pct	65.42	34.58	
Total	Frequency	809	500	1309

PROC TABULATE

SAS has other functions which produce better formatted tables such as **`proc tabulate`** but there are also other procedures such as report and writing out to a file.

The `table` line uses a comma to sperate row and column variables. Either can be summarised using the * and summary stats to be used. Here a basic frequency table with column percentages (`COLPCTN`) and frequencies (`N`) requested.

```
table row*(N COLPCTN), column;
```

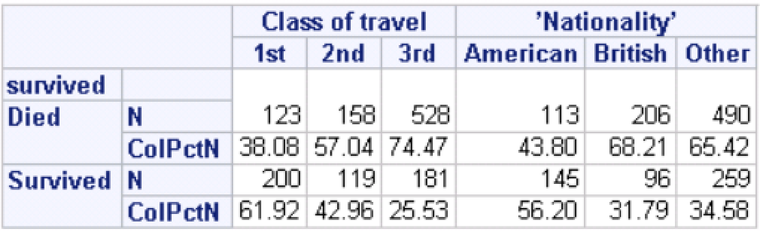
Put all categorical variables being used in the `class` line

```
proc tabulate data=Titanic;
class pclass residence survived;
table survived*(N COLPCTN), pclass residence;
format Residence nat. ;
```

Put variables to go across the columns after the comma

In the `table` statement, put row variables before the comma.

Specify the statistics required in brackets and multiply by either the row or column variables. Here we are requesting frequencies and column percentages



		Class of travel			'Nationality'		
		1st	2nd	3rd	American	British	Other
survived							
Died	N	123	158	528	113	206	490
	ColPctN	38.08	57.04	74.47	43.80	68.21	65.42
Survived	N	200	119	181	145	96	259
	ColPctN	61.92	42.96	25.53	56.20	31.79	34.58

The table can be rearranged, so nationality is in rows and row percentages are requested.

```
table residence, (pclass survived)*(N ROWPCTN) ;
```

To change the labels for the summary statistics, add labels after the summary statistics chosen. You can also add more information within the table through the BOX= command.

```
table survived*(N='No. of passengers' ROWPCTN='% within nationality'),
pclass residence/BOX='Survival rates by group';
```

Numeric variables can be formatted to a certain number of digits and decimals in a data step or procedure e.g. fare is formatted to 6 digits and one decimal: `format fare f6.1;`

Or by specifying a format for each statistic requested by using * after the statistic in the table line

```
table survived*(N COLPCTN*f=4.0), pclass residence;
```

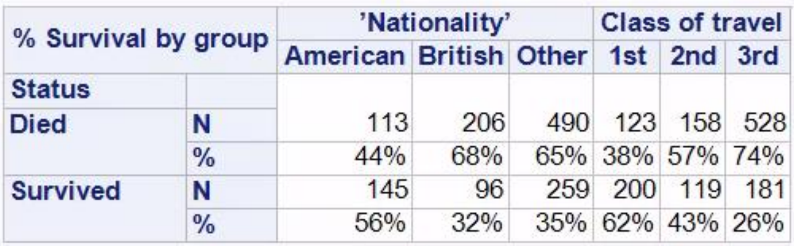
Ideally percentages should be reported to no decimals and with a percent sign to make it clear which are the frequencies and which are percentages but there is no automatic command so a new format needs to be created and applied. Here a new format we are calling `pctfmt` is being defined which allows up to 3 numbers, 0 decimal places and adds a % sign to whatever it is applied to.

```
proc format;
 picture pctfmt (round) other='009%';  run;
```

If you wanted decimals to be displayed (unlikely unless they are very small), you would adjust the `other='009%'` to `other='009.0%'`

Then apply the format to the requested percentages in PROC TABULATE.

```
table survived*(N
COLPCTN*f=pctfmt.), pclass
residence;
```

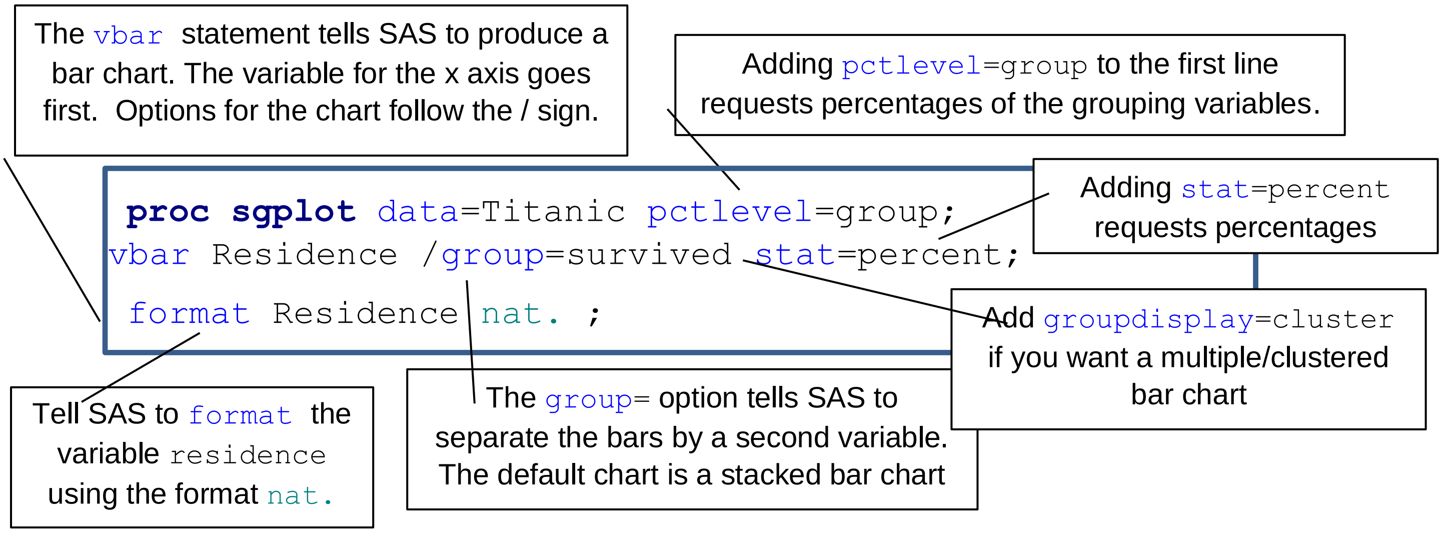


% Survival by group		'Nationality'			Class of travel		
		American	British	Other	1st	2nd	3rd
Status							
Died	N	113	206	490	123	158	528
	%	44%	68%	65%	38%	57%	74%
Survived	N	145	96	259	200	119	181
	%	56%	32%	35%	62%	43%	26%

Bar Charts

PROC SGPLOT can be used to produce a range of charts. The type of chart you choose is dependent on the data types of your variables. For categorical data, bar charts and percentages are most appropriate which can produced using the `vbar` statement in PROC SGPLOT followed by the variable to be plotted on the x-axis (the independent variable if you have one).

If you wish to split by a second variable, use the `/group=` command. The default chart will show frequencies, but you can request percentages by adding `stat=percent` to the vbar line. To ensure percentages of survival out of each nationality, add `pctlevel=group` to the top row.



The `vbar` statement tells SAS to produce a bar chart. The variable for the x axis goes first. Options for the chart follow the / sign.

Adding `pctlevel=group` to the first line requests percentages of the grouping variables.

```
 proc sgplot data=Titanic pctlevel=group;
vbar Residence /group=survived stat=percent;
 format Residence nat. ;
```

Adding `stat=percent` requests percentages

Add `groupdisplay=cluster` if you want a multiple/clustered bar chart

Tell SAS to `format` the variable `residence` using the format `nat.`

The `group=` option tells SAS to separate the bars by a second variable. The default chart is a stacked bar chart

To change from a stacked to a clustered/multiple bar chart add `groupdisplay=cluster` after the / in the vbar line `vbar Residence /group=survived stat=percent;`

When producing charts for reports, it is important that all axes are labelled and font for axes titles and values are made larger than the standard in SAS.

The charts should appear with labels attached to the data if you have set up formats correctly but if not, specify axes labels using `label=` and control the size of the text (here it is 15pt) using `labelattrs=(size=15pt)`

The legend is formatted in a similar way to the axes, but you can also adjust the position within the chart by adding `position=top` for example.

```
keylegend/ title='Survived or Died' titleattrs=(size=15pt)
valueattrs=(size=15pt)position=top;
```

The size of the value labels can be adjusted using `valueattrs=(size=15pt).` Each aspect is addressed in its own line, e.g., `xaxis` to change the variable and value attributes of the x-axis.

```
xaxis label='Class of passenger' labelattrs=(size=15pt)
valueattrs=(size=15pt);
```

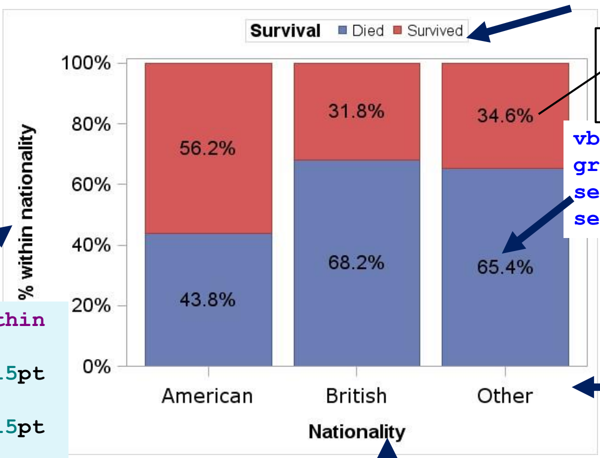
To display the percentages on the chart itself, add `seglabel` to the `vbar` line and then make the font larger in the same way as adjusting the axes using `seglabelattrs`.

Use `pctndec=0` in the **`proc sgplot`** line to reduce percentages displayed on the chart to 0 d.p.

```
proc sgplot data=titanic pctndec=0;
```

```
vbar Residence /group=survived stat=percent seglabel
seglabelattrs=(size=15);
```

```
keylegend/title='Survival'
titleattrs=(size=13pt weight=bold)
valueattrs=(size=12pt) position=top;
```



Reduce to 0 d.p. `proc sgplot data=titanic pctndec=0;`

```
vbar residence /
group=survived stat=pct
seglabel
seglabelattrs=(size=15);
```

```
yaxis label='% within
nationality'
labelattrs=(size=15pt
weight=bold)
valueattrs=(size=15pt
color=black);
```

```
xaxis
valueattrs=(size=15pt);
```

```
xaxis label='Nationality'
labelattrs=(size=15pt weight=bold)
```

Reporting charts

Always give the chart a figure number and title e.g. Figure 1: Stacked bar chart of nationality and survival, ensure the font is large enough to read and all axes and values are labelled.

Add a brief summary of the chart choosing the more important differences and in context of the research question the chart is addressing.

Do not include both a crosstabulation table with percentages and a bar chart of the same percentages. Perhaps use a table with several variables and bar charts where there are significant differences. Never produce a bar chart when there are only binary variables as percentages within the text are sufficient e.g. 80% of males died compared to 27% of women.