ADDING BLANK ROWS AND COLUMNS TO TABULATE OUTPUT USING CLASSDATA AND FORMATS

Jonas V. Bilenas, SASATICIAN

June 8, 2020

Quick Tips

OUTLINE

- This Quick SAS® Tip will look at how to add Blank Rows and/or Blank Columns to PROC TABULATE Reports.
- If you are **not** a TABULATE user, geek or master I have added some updated TABULATE training slides in this presentation in the appendix.
  - TABULATE can be a bit confusing if you aren't a PROC TABULATE GEEK and/or not familiar with the POWER of PROC FORMAT.
  - Check out some of "The Battle of The Titans" papers which compare TABBULATE to REPORT. (Bruns, Pass, & Easton 2001,2002) and (Lafler, Cochran, & Pass 2017).
  - I will try to set up a You-Tube or other form of Presentation on How to become a TABULATE GEEK once I get a WEB CAM.
  - At my first job in the consumer banking industry a contractor we had hired was kind to train me in how to use PROC TABULATE. I have trained many colleagues and staff over the years in TABULATE. Sometimes a PROC FREQ, PROC MEANS, or PROC SQL can be quicker to generate simple reports but TABULATE masters love using PROC TABULATE.
  - Also take a look SESUG 2016 Paper with Kajal Tahiliani, :"Making Sense of PROC TABULATE" https://analytics.ncsu.edu/sesug/2016/HOW-138_Final_PDF.pdf
  - I will also update the code from the above paper to include some traffic lighting. Code links will be shown in the appendix of this document.
- I will include the code to create a simulated data set that I have used in many of my SAS USER GROUP (SUG) presentations titled "MAKING SENSE OF PROC TABULATE". The simulated data set is a simple banking example on reporting results for 2 different credit card balance transfer offers over 2 months. Good to test tabulate code on simulated or sampled data when you are generating new reports.
  - To run the code on this Tip or other presentations make sure you use the same seed used for generating the simulated data for each presentation to get the same results.
- To make this a Quick Tip we will look at a simple TABULATE report of Quantile and Percentile values of Balance Transfer amounts for simulated accounts that responded to the balance transfer offer.
- We will look at how using USER FORMAT CODE with the NOTSORTED option and CLASSDATA options in PROC TABULATE can *sometimes* make blank rows and/or columns in a Tabulate Report as is seen specifically for this Quick Tip.
- I will Include final code that dump output into ODS EXCEL in SAS 9.4 (more recent maintenance versions). I have found that it makes it easier to cut and paste results into other documents. Locations of the codes and excel output will be included in the appendix as well.

CODE FOR THE SIMULATED DATA

CODE FOR SIMULATED DATA USED IN THIS EXERCISE

```
proc format;
  value offer    /* Offer mix rates */
                 low    - 0.45 = 'A'
                 0.45 <- high = 'B'
  ;
  value $grroff /* Response Rate FORMATS TO use based on OFFER */
                 'A' = 'aoff'
                 'B' = 'boff'
  ;
  value aoff     /* Response rates for A-OFFER */
                 low - 0.10 = '1'
                 other      = '0'
  ;
  value boff     /* Response rates for B-OFFER */
                 low - 0.05 = '1'
                 other      = '0'
  ;
  value $ABT     /* Average Balance Transfer Amounts by Offer */
                 'A' = '6500'
                 'B' = '5400'
  ;
run;
```

- We will create simulated Balance Transfer offers for 2 offers for 2 different campaign periods in this code.
- Simulation code on the next page will set up offer A (a good offer for the consumer) 45% of the time and 55% of the time form offer B (not as good as offer A for the consumer).
- Offer A will get a simulated response rate of about 10% and offer B will get a response rate of about 5%.
- Average balance transfer amounts by responded customers will be approximately $6,500 for Offer A and about $5,400 for offer B. We will add some random normal variability (normal STD of $500) to the average balances.

CODE FOR SIMULATED DATA USED IN THIS EXERCISE

```
data stuff.simdata;
  call streaminit(20160804);          /* Wicklin (2013) */
  do campaign = '2015/3', '2015/4'; /* campaign period */
    do i = 1 to 1e6;                  /* 1e6 = 1*10**6 = 1,000,000 */
      mailed=1;                       /* Is used in other presentations */
      offer=put(rand("Uniform"),offer.);
      fmtuse=put(offer,$grroff.);
      respond=input(putn(rand("Uniform"),fmtuse),best12.);
      if respond then
        baltran=rand("Normal")*500+input(put(offer,$ABT.),best12.);
      else baltran=.;
      output;
    end;
  end;
  drop i fmtuse;
run;
```

- See Rick Wicklin's 2013 Blog on "Six reasons you should stop using the RANUNI function to generate random numbers". I believe he has posted additional blogs regarding the new generation of random numbers.
- Use the same seed, 20160804, to get same result for this presentation.
- The PUTN second argument is **not** a FORMAT but a variable that directs to the FORMAT specified in the FMTUSE variable which, in this code, can take on 2 values using the $groff format; 'aoff' or 'boff'.
- PUT and PUTN always returns a character value. Use the INPUT function to convert to a numeric variable if required. You can go with generating a USER INFORMAT in PROC FORMAT but I used to have a few minor issues when using USER INFORMATS decades ago.

LET'S LOOK AT THE TABULATE OUTPUT WHERE WE WANT TO ADD A BLANK ROW

LET'S LOOK AT THE ORIGINAL TABULATE OUTPUT

- Only 1 comma. CAMPAIGN by OFFER will be the row elements.
- The column variable is only looking at the BALTRAN variable. We label that as 'Balance Transfer' in the report.
- We list all the stats to view specific percentile and quantiles.
- Run Time ran very quickly.
- If running time takes too long, try QMETHOD=P2 **but** "reliable estimations of some percentiles (P1,P5,P95,P99) might not be possible for some data sets".

Run Time on SAS On Demand For Academics was quick. Maybe everyone is on summer break:

```
NOTE: There were 2000000 observations read
from the data set STUFF.SIMDATA.
NOTE: PROCEDURE TABULATE used (Total process
time):
      real time           0.36 seconds
      user cpu time       0.66 seconds
      system cpu time     0.05 seconds
      memory              28212.42k
      OS Memory           59960.00k
```

```
title Tabulate Run with no Blank Rows and Columns;
proc tabulate data=stuff.simdata missing;
  class campaign offer;
  var baltran;
  table CAMPAIGN*OFFER
        ,
        baltran='Balance Transfer'*
        (min p1 p5 p25 median p75 p95 p99 max)*f=dollar8.2
        /rts=30 row=float misstext= ' '
run;
```

Tabulate Run with no Blank Rows and Columns

		Balance Transfer								
		Min	P1	P5	P25	Median	P75	P95	P99	Max
campaign	offer									
2015/3	A	$4,282.53	$5,324.69	$5,677.24	$6,161.11	$6,497.36	$6,833.45	$7,314.01	$7,646.90	$8,673.67
	B	$3,311.14	$4,236.64	$4,580.42	$5,059.80	$5,400.16	$5,739.53	$6,222.99	$6,550.34	$7,474.83
2015/4	A	$4,342.86	$5,355.04	$5,680.02	$6,156.57	$6,497.98	$6,838.17	$7,318.78	$7,654.10	$8,511.49
	B	$3,439.67	$4,242.70	$4,578.31	$5,064.28	$5,401.50	$5,735.58	$6,223.47	$6,573.54	$7,386.36

WE WANT TO ADD A BLANK ROW BETWEEN THE CAMPAIGNS

WE WANT TO ADD A BLANK ROW BETWEEN THE CAMPAIGNS

```
data classes;
  do campaign='2015/3', '2015/4';
    do offer = 'A', 'B';
      output;
    end;
  campaign=' '; offer=' ';
  output;
  end;
run;


proc format;
  value $classes (notsorted)
    '2015/3' = '2015/3'
    ' '      = ' '
    '2015/4' = '2015/4'
  ;
run;
```

- We are setting up a data sets with all the CLASS LEVELS we use and need to make the blank row. The data set classes will be referenced in the CLASSDATA option in PROC TABULATE on the next page.
- Blank Rows will be created by assigning blank spaces in classes data set and in the format $CLASSES.
- If the CLASS variable is numeric you can use numeric missing values (up to 53 missing values can be assigned to numeric variables). Not included in this presentation but I had examples over the years using TABULATE to generate P&L output by year and month using CLASSDATA option in tabulate and generating the NOTSORTED format to insert blank columns between years and months.
- The classes data set is used in the next TABULATE CODE.
- We set up a user format. By default, user formats are sorted by the **keys**; to the left of the =.
- We want to Follow the NOTSORTED FORMAT **Values**; to the right of the = . Try code without the NOTSORTED option to see what happens in the report. Note that we only want to include a blank row between campaign months but we still need all CLASS LEVELS specified in the CLASSES data set that will be specified in the CLASSDATA option of PROC TABULATE.

LET'S LOOK AT THE TABULATE CODE TO ADD A BLANK ROW

```
title Tabulate Run with Blank Rows;
proc tabulate data=stuff.simdata missing
              /*QMETHOD=P2*/ classdata=classes;
  class campaign/preloadfmt order=data;
  class offer;
  format campaign $classes.;
  var baltran;
  keylabel n=' ' sum=' ' mean=' ' std=' ' pctn=' ' pctsum=' ' ;
  table CAMPAIGN*OFFER
        ,
    baltran='Balance Transfer'*
        (min p1 p5 p25 median p75 p95 p99 max)*f=dollar8.2
        /rts=30 row=float misstext=' ';
run;
title;
```

- OK, you can add a row or column when you use ODS EXCEL to load the tabulate report to the specified Excel file and specified sheet. You can edit the Excel output using INSERT.
- But if you are copying the output directly from SAS/EG or SAS/STUDIO or using other ODS output such as RTF or HTML it maybe more difficult to insert blank rows or columns.

Tabulate Run with Blank Rows

		Balance Transfer								
		Min	P1	P5	P25	Median	P75	P95	P99	Max
campaign	offer									
2015/3	A	$4,282.53	$5,324.69	$5,677.24	$6,161.11	$6,497.36	$6,833.45	$7,314.01	$7,646.90	$8,673.67
	B	$3,311.14	$4,236.64	$4,580.42	$5,059.80	$5,400.16	$5,739.53	$6,222.99	$6,550.34	$7,474.83
2015/4	A	$4,342.86	$5,355.04	$5,680.02	$6,156.57	$6,497.98	$6,838.17	$7,318.78	$7,654.10	$8,511.49
	B	$3,439.67	$4,242.70	$4,578.31	$5,064.28	$5,401.50	$5,735.58	$6,223.47	$6,573.54	$7,386.36

LET'S LOOK AT A TABULATE CODE WHERE WE ALSO ADD SOME BLANK COLUMNS

LET'S LOOK AT A TABULATE CODE WHERE WE ALSO ADD SOME BLANK COLUMNS

```
data classes;
  do campaign='2015/3', '2015/4';
    do offer = 'A', 'B';
      output;
    end;
  campaign=' '; offer=' ';
  output;
  end;
run;


proc format;
  value $classes (notsorted)
    '2015/3' = '2015/3'
    ' '  = ' '
    '2015/4' = '2015/4'
  ;
  value blank_it_out
    low-high = '      '
    other    = '      '
  ;
  value accts /* set 0 account total to missing */
        . , 0  = ' '
        other  = [comma10.]
  ;
run;
```

- I want to add some additional statistics but want to separate new stats from percentiles using blank spaces.
- The Statistics are not numeric CLASS variables LEVELS so no need to change the CLASSES data set.
- So, for this example, I added 2 new USER FORMATS to blank out values when I repeat a STATISTIC in TABULATE. A cheating way of adding blank space. The added FORMATS:
  - BLANK_IT_OUT
  - ACCTS
- We still need the classes data set since we want to keep the blank row.

LET'S LOOK AT A TABULATE CODE WHERE WE ALSO ADD SOME BLANK COLUMNS

- The noseps and formchar are mostly used for LISTING output.
- MISSING option forces missing class levels to be included in the tabulate report.
- Make sure to preload the format with the order specified in USER FORMAT that has a NOTSORTED option.
  - **class campaign/preloadfmt order=data;**
- You can have more than 1 CLASS statement in PROC TABULATE.
- I always include **a keylabel** statement to blank out stat labels since they may change within the TABLE statements.
- MISSTEXT=' ' option specifies a blank value rather than a SAS missing value.
- I always forget why I always use the ROW=FLOAT option. From SAS 9.4 M7 documentation:
  - "specifies whether all title elements in a row crossing are allotted space even when they are blank."
- Be careful when copying the code from this presntation that includes quotes. Some don't look ok, even if I use the "courier new" font.

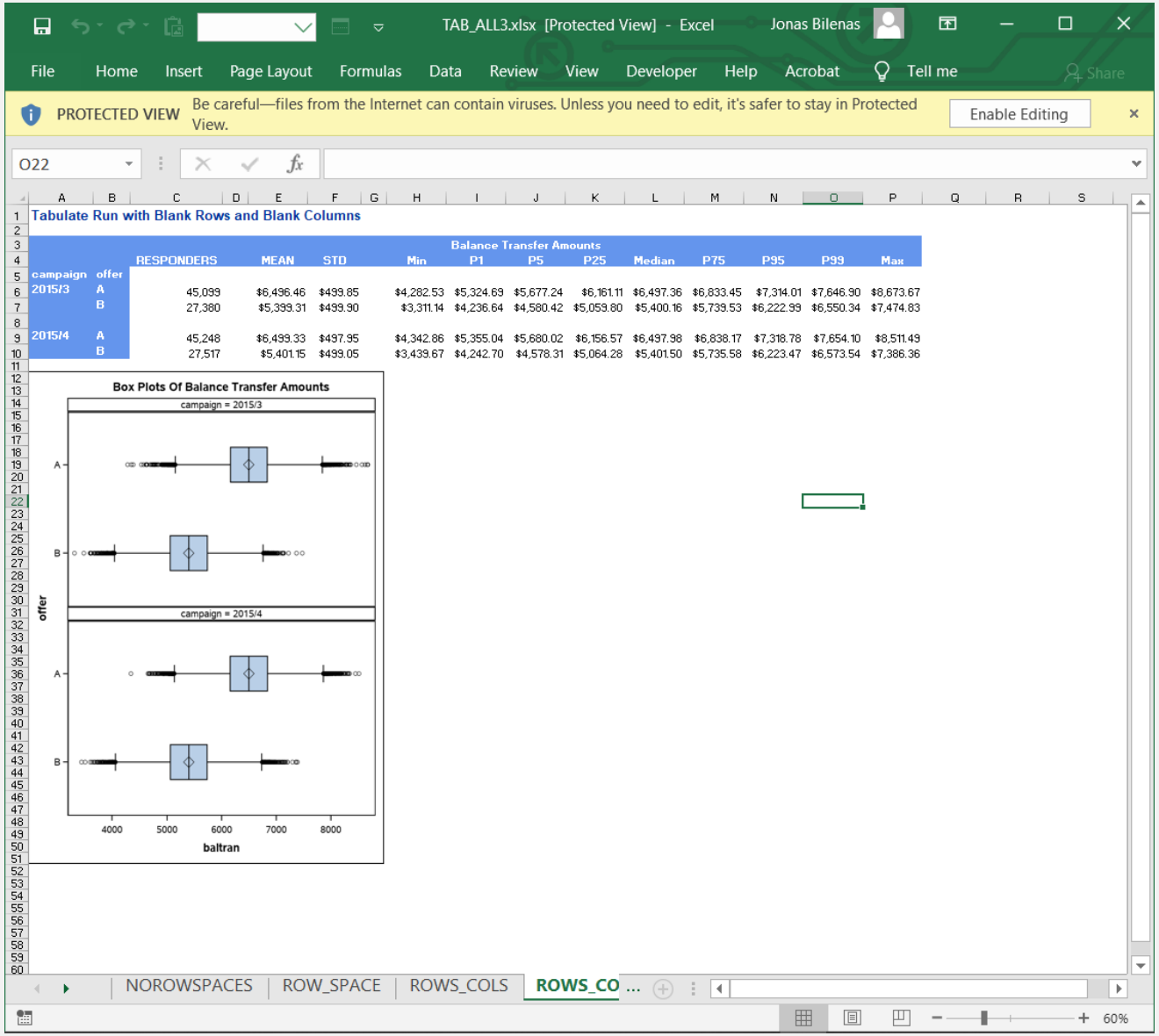
```
title Tabulate Run with Blank Rows and Blank Columns;
proc tabulate data=stuff.simdata noseps formchar='            ' missing
              /*QMETHOD=P2*/ classdata=classes;
  class campaign/preloadfmt order=data;
  class offer;
  format campaign $classes.;
  var baltran;
  keylabel n=' ' sum=' ' mean=' ' std=' ' pctn=' ' pctsum= ' ';
  TABLE CAMPAIGN*OFFER
        ,
    baltran='Balance Transfer Amounts'*
        (
          (N = 'RESPONDERS'*f=accts.
           N = '        '*f=blank_it_out. /* 1st blank column */
          )
          (MEAN = 'MEAN'*f=dollar8.2
           STD  = 'STD'*f=dollar8.2
           N ='        '*f=blank_it_out.  /* 2nd blank column */
          )
          (min p1 p5 p25 median p75 p95 p99 max)*f=dollar8.2
        )
        /rts=30 row=float misstext=' ';
run;
```

LET'S LOOK AT A TABULATE OUTPUT WHERE WE ALSO ADD SOME BLANK COLUMNS

Tabulate Run with Blank Rows and Blank Columns

		Balance Transfer Amounts											
		RESPONDERS	MEAN	STD	Min	P1	P5	P25	Median	P75	P95	P99	Max
campaign	offer												
2015/3	A	45,099	$6,496.46	$499.85	$4,282.53	$5,324.69	$5,677.24	$6,161.11	$6,497.36	$6,833.45	$7,314.01	$7,646.90	$8,673.67
	B	27,380	$5,399.31	$499.90	$3,311.14	$4,236.64	$4,580.42	$5,059.80	$5,400.16	$5,739.53	$6,222.99	$6,550.34	$7,474.83
2015/4	A	45,248	$6,499.33	$497.95	$4,342.86	$5,355.04	$5,680.02	$6,156.57	$6,497.98	$6,838.17	$7,318.78	$7,654.10	$8,511.49
	B	27,517	$5,401.15	$499.05	$3,439.67	$4,242.70	$4,578.31	$5,064.28	$5,401.50	$5,735.58	$6,223.47	$6,573.54	$7,386.36

USING SAS9.4 YOU CAN ALSO DUMP GRAPHS IN SPECIFIED EXCEL FILES USING ODS EXCEL



- Full code will be provided to test if you can get the same code. If code does not work you may not the have a newer maintenance version of SAS 9.4 or using earlier versions of SAS such as 9.1, 9.2, or 9.3.
- Also note that if you "Enable Editing" you may get a warning that the numeric data is in character format. Highlight the region of numeric fields and left click on the '!' and select "convert to numeric".

APPENDIX

- **Appendix 1. My updated Rules to become a TABULATE MASTER.**
- **Appendix 2. Some Tabulate Syntax.**
- **Appendix 3. Some Tabulate statistics SAS 9.4 / M7.**
  - **Some stats may not work in earlier versions of SAS 9.4.**
  - **Extracted from https://documentation.sas.com/?docsetId=proc&docsetTarget=p0n4welprckk8yn1ro9swaef6x0n.htm&docsetVersion=9.4&locale=en**
- **Appendix 4. Code. Note, may not work if you are running an old version of SAS. Code:**
  - **tab1_data.sas**
    - **Generates the simulated data set used for this presentation**
  - **tab_all.sas**
    - **Single program to run the ODS ECEL output generating 5 sheet names.**
- **Appendix 5. The spreadsheet created from tab_all.sas**
  - **TAB_ALL3.xlsx**

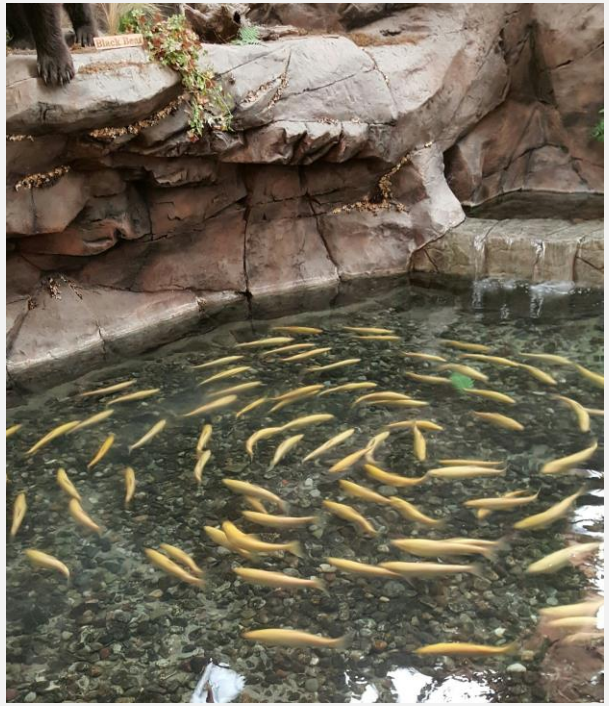
1. MY UPDATED RULES TO BECOME A TABULATE MASTER

- **Rule #1: Don’t panic**
  - **You Need to be in a Certain Frame of Mind to do TABULATE code. Relax and focus on the task. Do not multitask if possible. If you have many projects to work on set aside 1 hour to focus on each of the tasks. There are a large number of research studies that suggest multitasking will make you less productive and may cause brain damage.**
- **Rule #2: Design the Report on Paper with Pencil or Pen**
  - **MOST IMPORTANT RULE! If you don’t know how your report should look like you are stuck in limbo. TABULATE is very flexible.**
- **Rule #3: If building a complicated report, build components of your report one at a time and verify that the results make sense. Build the report in stages.**
- **Rule #4: Try the ALL Statement in your report. Used to total up CLASS levels.**
  - **This presentation will not include an ALL statement but you can try it.**
- **Rule #5: Try some PCTN and/or PCTSUM options to get percentages or ratios.**
  - **Not included in this exercise. But needs to be tested to make sure you get the expected results. I also use PICTURE FORMATS with a MULT option to get Ratios as opposed to percentages.**

1. MY UPDATED RULES TO BECOME A TABULATE MASTER

- **Rule #6: use PICTURE FORMATS and be careful to test out the results.**
  - **This will not be covered here but is handy:**
    - **Add a percent sign to the output result via a user defined PICTURE FORMAT. TABULATE percentage stats (PCTN, PCTSUM) cannot use the PERCENT format since the values are already multiplied by 100. Maybe there is a SAS FORMAT that will handle that but I am not sure.**
    - **Remember the ROUND option for PICTURE FORMATS if you want rounded results. Default is to truncate.**
    - **You can also get RATIO results, as opposed to PERCENT stats using the MULT option for USER DEFINED PICTURE FORMATS.**
- **Rule #7: (new rule). Use NOTSORTED option in your USER FORMATS to follow the output in the order specified by the NOTSORETED FORMAT.**
  - **We will require this rule in this example.**

1. MY UPDATED RULES TO BECOME A TABULATE MASTER



Maybe the bear ate the rare fish?

- **Rule #8: Try the CLASSDATA option in TABULATE that specifies exact combinations of CLASS LEVELS to include even if a CLASS LEVEL is missing in the data set. The first time I used it was when I was asked a question by an attendee on TABULATE the day before I was presenting my "Making Sense of Proc Tabulate Paper" at PNWSUG (Pacific North West SAS Users Group) Conference in 2006.**
  - **The issue was with missing fish species in their monthly tracking of fish in specific streams.**
  - **On occasion, a species may have not be seen but she wanted to add the missing fish CLASS LEVEL to the TABULATE report since they are tracked in the monthly reports over time.**
  - **CLASSDATA was not in my PNWSUG paper but was added to the presentation. Luckily that I found a paper by Wright(2002) that explained the CLASSDATA option.**
  - **Also check Cerussi (2011)**
- **Rule #9: Try ODS EXCEL to send TABULATE output along with SAS GRAPHS directly to an Excel file. Code will be shared. Code may not work if you have an older version of SAS.**
- **Rule #10: If your boss wants changes to the report, don't worry. Show him/her you can do it as a TABULATE master.**

2. SOME TABULATE SYNTAX

- **CLASS: categories (numeric or character variables)**
- **VAR: analysis variables (numeric variables)**
- **TABLE: Specifies table design.**
  - **Table dimensions (page, row, column) are separated by commas “,”**
  - *** : Use to specify any of the following:**
    - **Another CLASS variable Split**
    - **A VAR variable**
    - **A statistic**
    - **A format**
  - **<> : Specifies the denominator CLASS dimensions for PCTN or PCTSUM. Also used to specify the denominator VAR variable.**
  - **(): To group CLASS and/or VAR variables**
  - **= : For FORMAT and label specification**

3. SOME TABULATE STATISTICS SAS 9.4 / M7

Descriptive statistic keywords

```
COLPCTN          PCTSUM
COLPCTSUM        RANGE
CSS              REPPCTN
CV               REPPCTSUM
KURTOSIS | KURT
ROWPCTN          ROWPCTSUM

LCLM
MAX              SKEWNESS | SKEW
MEAN             STDDEV | STD
MIN              STDERR
MODE             SUM
N                SUMWGT
NMISS            UCLM
PAGEPCTN         USS
PAGEPCTSUM       VAR
PCTN
```

Quantile statistic keywords

```
MEDIAN | P50    Q3 | P75
P1     P70
P5     P80
P10    P90
P20    P95
P30    P99
P40
P60
Q1|P25     QRANGE
```

Hypothesis testing keywords

```
PROBT | PRT     T
```

Source:
https://documentation.sas.com/?docsetId=proc&docsetTarget=p0n4welprckk8yn1ro9swaef6x0n.htm&docsetVersion=9.4&locale=en

4. LOCATION OF THE CODE

- Code is stored in GitHub Directory. Your best bet is to open in google chrome website.
  - https://github.com/jonasbilenas/Short-Topics/blob/master/tab1_data.sas
  - https://github.com/jonasbilenas/Short-Topics/blob/master/tab_all.sas

5. LOCATION OF THE ODS EXCEL SPREADSHEET FILE

- https://jonasbilenascom.wpcomstaging.com/wp-content/uploads/2020/06/TAB_ALL3.xlsx

REFERENCES

- Bilenas, J. (2005) "The Power of PROC FORMAT", SAS Press, Cary, NC, SAS Institute Inc.
- Bilenas, J. (2014) "Then Power of PROC FORMAT", SAS GLOBAL FORUM, SGF 2014.
- Bilenas, J. & Tahiliani, K. (2016) "Then Power of PROC FORMAT", SESUG 2016.
- Bilenas, J. & Tahiliani, K. (2016) "Making Sense of PROC Tabulate", SESUG 2016. https://analytics.ncsu.edu/sesug/2016/HOW-138_Final_PDF.pdf
- Bruns, D., Pass,R., & Easton, A. (2002) "Battle of the Titans: REPORT vs. Tabulate", SUGI 27.
- Bruns, D., Pass,R., & Easton, A. (2001) "Battle of the Titans: REPORT vs. Tabulate", NESUG 2001.
- Cerussi, P. (2011) "Force Missing Rows and Columns with CLASSDATA in PROC TABULATE", SAS GLOBAL FORUM 2011. https://support.sas.com/resources/papers/proceedings11/087-2011.pdf
- Haworth, L. & McKnight, J. (2015) "PROC TABULATE by Example, Second Edition" SAS Press, Cary, NC, SAS Institute Inc.
- Lafler, K.P, Cochran, B., & Pass,R. (MWSUG 2017 - Paper SA14) "The Battle of the Titans (Part II): PROC TABULATE versus PROC REPORT"
- Watts, P. (2006) "Using Database Principles to Optimize SAS® Format Construction from Tabular Data", SUGI31.
- WICKLIN, RICK (2013) "SIX REASONS YOU SHOULD STOP USING THE RANUNI FUNCTION TO GENERATE RANDOM NUMBERS." AVAILABLE AT https://blogs.sas.com/content/iml/2013/07/10/stop-using-ranuni.html
- Wright, W. (2002 "Introduction to PROC TABULATE" NESUG 2006

Note: papers from SGF, SESUG, and SUGI can be quickly located at http://lexjansen.com/

DISCLAIMERS

- **SAS® and all other SAS Institute Inc. product or service names are registered trademarks or trademarks of SAS Institute Inc. in the USA and other countries. ® indicates USA registration.**
- **Other brand and product names are trademarks of their respective companies.**
- **The contents of this paper are the work of the author and do not necessarily represent the opinions, recommendations, or practices of any company that I worked for.**
- **CODE was tested on SAS® OnDemand for Academics**
- **No warranty for any code in this presentation. Use at your own risk.**