Package Validation for “procs”

Background: The “procs” package simulates some SAS® procedures in R. The package was written to make it easier for SAS® programmers to work in R. The package was also written to align statistical output between SAS® and R, as the native R functions do not necessarily match SAS®. This “pre-validation” will reduce the number of discrepancies in statistical output, and speed the process of analysis.

Purpose: The purpose of this document is to validate the R “procs” package against equivalent SAS® output.

Methodology: For this validation, first a test was defined that could be accomplished in both SAS® and R. Then a programmer wrote a small script in each language that produced similar results. The results were compared visually to ensure a match. Any discrepancies would be investigated and resolved either by modifying the comparison scripts, or modifying the “procs” package code until it matched SAS®. Comparison code and results were documented.

SAS Version: SAS Studio 3.81
R Version: 4.3.2
Procs Version: 1.0.6
Author: David J. Bosak, Chief Software Architect, Archytas Clinical Solutions
Contributor: Yifei Chen
Last Modified Date: 3/02/2024

Change History

Date	Item	Procs Version
2023/06/23	Initial version.	0.0.9007
2023/06/24	Created first few tests for each function.	0.0.9007
2023/07/05	Discovered discrepancy in Procs-means-009 Upper 95% confidence limit. Had to change code to add alpha option and one-sided confidence limit calculation.	0.0.9007
2023/07/08	Resolved discrepancy on confidence limits. Added test means-010 to confirm two-sided CL.	0.0.9008
2023/07/10	Added Procs-means-011 tests for means t-test and p-value.	0.0.9008
2023/07/11	Added Procs-means-012 test for Uncorrected Sum of Squares.	0.0.9008
2023/07/20	Added Procs-means-013 test for skewness and kurtosis.	0.0.9008
2023/07/25	Added ability to use factors for sorting proc_freq output.	0.0.9008
2023/07/26	Fixed Fisher’s exact test for Cell 1.1. This was off when table was sorted properly.	0.0.9008
2023/07/29	Incremented to version 1.0.0 for CRAN submission.	1.0.0
2023/08/08	Incremented to version 1.0.2 for CRAN fixes	1.0.2

2023/08/10	Fixed zero-fill row on proc_freq() when user supplied a factor.	1.0.2
2023/08/11	Fixed n-way class summaries on proc_means() when multiple class vars are present.	1.0.2
2023/08/13	Fixed some labels on proc_means() report output.	1.0.2
2023/08/18	Add means-015 for more complicated statistics with by variable.	1.0.2
2023/09/04	Upgrade version and prepare for CRAN submission.	1.0.3
2023/12/17	Add validation for proc_ttest().	1.0.4
2023/12/20	Reran all comparisons and made a few fixes.	1.0.4
2024/01/20	Bug fixes. No changes to validation.	1.0.5
2024/03/02	Added validation for proc_reg().	1.0.6

Freq Data

SAS

```
data Color;
  input Region Eyes $ Hair $ Count @@;
  label Eyes  ='Eye Color'
        Hair  ='Hair Color'
        Region='Geographic Region';
  datalines;
1 blue  fair   23  1 blue  red    7  1 blue  medium 24
1 blue  dark   11  1 green fair   19  1 green red    7
1 green medium 18  1 green dark   14  1 brown fair   34
1 brown red    5  1 brown medium 41  1 brown dark   40
1 brown black  3  2 blue  fair   46  2 blue  red    21
2 blue  medium 44  2 blue  dark   40  2 blue  black  6
2 green fair   50  2 green red    31  2 green medium 37
2 green dark   23  2 brown fair   56  2 brown red    42
2 brown medium 53  2 brown dark   54  2 brown black  13
;
```

R

```
dat <- read.table(header = TRUE, text = '
 Region Eyes Hair Count
 1 blue  fair   23
 1 blue  dark   11
 1 green medium 18
 1 brown red    5
 1 brown black  3
 2 blue  medium 44
 2 green fair   50
 2 green dark   23
 2 brown medium 53
 1 blue  red    7
 1 green fair   19
 1 green dark   14
 1 brown medium 41
 2 blue  fair   46
 2 blue  dark   40
 2 green red    31
 2 brown fair   56
 2 brown dark   54
 1 blue  medium 24
 1 green red    7
 1 brown fair   34
 1 brown dark   40
 2 blue  red    21
 2 blue  black  6
 2 green medium 37
 2 brown red    42
 2 brown black  13
 ')
```

** Above input data will be used for all frequency comparisons unless data is provided in the script itself.

proc_freq-001

SAS Code and Output

```
proc freq data=Color;
  tables Eyes Hair Eyes*Hair / out=FreqCount outexpect sparse;
  weight Count;
  title 'Eye and Hair Color of European Children';
run;
```

Eye and Hair Color of European Children

The FREQ Procedure

Eye Color

Eyes	Frequency	Percent	Cumulative Frequency	Cumulative Percent
blue	222	29.13	222	29.13
brown	341	44.75	563	73.88
green	199	26.12	762	100.00

Hair Color

Hair	Frequency	Percent	Cumulative Frequency	Cumulative Percent
black	22	2.89	22	2.89
dark	182	23.88	204	26.77
fair	228	29.92	432	56.69
medium	217	28.48	649	85.17
red	113	14.83	762	100.00

Table of Eyes by Hair

Frequency Percent Row Pct Col Pct	Hair(Hair Color)					
Eyes(Eye Color)	black	dark	fair	medium	red	Total
blue	6 0.79 2.70 27.27	51 6.69 22.97 28.02	69 9.06 31.08 30.26	68 8.92 30.63 31.34	28 3.67 12.61 24.78	222 29.13
brown	16 2.10 4.69 72.73	94 12.34 27.57 51.65	90 11.81 26.39 39.47	94 12.34 27.57 43.32	47 6.17 13.78 41.59	341 44.75
green	0 0.00 0.00 0.00	37 4.86 18.59 20.33	69 9.06 34.67 30.26	55 7.22 27.64 25.35	38 4.99 19.10 33.63	199 26.12
Total	22 2.89	182 23.88	228 29.92	217 28.48	113 14.83	762 100.00

R Code and Output

```
library(procs)

proc_freq(dat, tables = v(Eyes, Hair, Eyes * Hair), weight = Count,
        title = "Eye and Hair Color of European Children")
```

Eye and Hair Color of European Children

Table of Eyes

Eyes	N	Frequency	Percent	Cumulative Frequency	Cumulative Percent
blue	762	222	29.13	222	29.13
brown	762	341	44.75	563	73.88
green	762	199	26.12	762	100.00

Table of Hair

Hair	N	Frequency	Percent	Cumulative Frequency	Cumulative Percent
black	762	22	2.89	22	2.89
dark	762	182	23.88	204	26.77
fair	762	228	29.92	432	56.69
medium	762	217	28.48	649	85.17
red	762	113	14.83	762	100.00

Table of Eyes by Hair

		Hair					
Eyes	Statistic	black	dark	fair	medium	red	Total
blue	Frequency	6	51	69	68	28	222
	Percent	0.79	6.69	9.06	8.92	3.67	29.13
	Row Pct	2.70	22.97	31.08	30.63	12.61	
	Col Pct	27.27	28.02	30.26	31.34	24.78	
brown	Frequency	16	94	90	94	47	341
	Percent	2.10	12.34	11.81	12.34	6.17	44.75
	Row Pct	4.69	27.57	26.39	27.57	13.78	
	Col Pct	72.73	51.65	39.47	43.32	41.59	
green	Frequency	0	37	69	55	38	199
	Percent	0.00	4.86	9.06	7.22	4.99	26.12
	Row Pct	0.00	18.59	34.67	27.64	19.10	
	Col Pct	0.00	20.33	30.26	25.35	33.63	
Total	Frequency	22	182	228	217	113	762
	Percent	2.89	23.88	29.92	28.48	14.83	100.00

Comparison	**Results**
Compare basic one and two way frequencies with weight option.	Pass

proc_freq-002

SAS Code and Output

```
proc freq data=Color nlevels;
  tables Eyes Hair Eyes*Hair / out=FreqCount outexpect sparse;
  title 'Eye and Hair Color of European Children';
run;
```

Eye and Hair Color of European Children

The FREQ Procedure

Number of Variable Levels		
Variable	Label	Levels
Eyes	Eye Color	3
Hair	Hair Color	5

Eye Color				
Eyes	Frequency	Percent	Cumulative Frequency	Cumulative Percent
blue	9	33.33	9	33.33
brown	10	37.04	19	70.37
green	8	29.63	27	100.00

Hair Color				
Hair	Frequency	Percent	Cumulative Frequency	Cumulative Percent
black	3	11.11	3	11.11
dark	6	22.22	9	33.33
fair	6	22.22	15	55.56
medium	6	22.22	21	77.78
red	6	22.22	27	100.00

Frequency
Percent
Row Pct
Col Pct

Table of Eyes by Hair						
	Hair(Hair Color)					
Eyes(Eye Color)	black	dark	fair	medium	red	Total
blue	1 3.70 11.11 33.33	2 7.41 22.22 33.33	2 7.41 22.22 33.33	2 7.41 22.22 33.33	2 7.41 22.22 33.33	9 33.33
brown	2 7.41 20.00 66.67	2 7.41 20.00 33.33	2 7.41 20.00 33.33	2 7.41 20.00 33.33	2 7.41 20.00 33.33	10 37.04
green	0 0.00 0.00 0.00	2 7.41 25.00 33.33	2 7.41 25.00 33.33	2 7.41 25.00 33.33	2 7.41 25.00 33.33	8 29.63
Total	3 11.11	6 22.22	6 22.22	6 22.22	6 22.22	27 100.00

R Code and Output	
proc_freq(dat, tables = v(Eyes, Hair, Eyes * Hair), title = "Eye and Hair Color of European Children", options = v(nlevels))	**Eye and Hair Color of European Children** Variable: Eyes, Levels: 3 **Table of Eyes** Eyes | N | Frequency | Percent | Cumulative Frequency | Cumulative Percent blue | 27 | 9 | 33.33 | 9 | 33.33 brown | 27 | 10 | 37.04 | 19 | 70.37 green | 27 | 8 | 29.63 | 27 | 100.00 Variable: Hair, Levels: 5 **Table of Hair** Hair | N | Frequency | Percent | Cumulative Frequency | Cumulative Percent black | 27 | 3 | 11.11 | 3 | 11.11 dark | 27 | 6 | 22.22 | 9 | 33.33 fair | 27 | 6 | 22.22 | 15 | 55.56 medium | 27 | 6 | 22.22 | 21 | 77.78 red | 27 | 6 | 22.22 | 27 | 100.00 Variable: Eyes, Levels: 3 Variable: Hair, Levels: 5 **Table of Eyes by Hair** Eyes | Statistic | Hair: black | dark | fair | medium | red | Total blue | Frequency | 1 | 2 | 2 | 2 | 2 | 9 Percent | 3.70 | 7.41 | 7.41 | 7.41 | 7.41 | 33.33 Row Pct | 11.11 | 22.22 | 22.22 | 22.22 | 22.22 | Col Pct | 33.33 | 33.33 | 33.33 | 33.33 | 33.33 | brown | Frequency | 2 | 2 | 2 | 2 | 2 | 10 Percent | 7.41 | 7.41 | 7.41 | 7.41 | 7.41 | 37.04 Row Pct | 20.00 | 20.00 | 20.00 | 20.00 | 20.00 | Col Pct | 66.67 | 33.33 | 33.33 | 33.33 | 33.33 | green | Frequency | 0 | 2 | 2 | 2 | 2 | 8 Percent | 0.00 | 7.41 | 7.41 | 7.41 | 7.41 | 29.63 Row Pct | 0.00 | 25.00 | 25.00 | 25.00 | 25.00 | Col Pct | 0.00 | 33.33 | 33.33 | 33.33 | 33.33 | Total | Frequency | 3 | 6 | 6 | 6 | 6 | 27 Percent | 11.11 | 22.22 | 22.22 | 22.22 | 22.22 | 100.00
Comparison	**Result**
Compare basic frequencies with no weight and nlevels option.	Pass

proc_freq-003

SAS Code and Output

```
data SummerSchool;
  input Gender $ Internship $ Enrollment $ Count @@;
  datalines;
boys  yes yes 35   boys   no no 27
boys   no yes 14   girls yes no 10
girls yes yes 32   girls  no no 23
girls  no yes 53   girls  . no 25
boys  yes . 29     girls  . no 29
;

proc freq data=SummerSchool order=data nlevels;
  tables Internship  / out = Fork missing;
  options missing=. mi;
run;
```

The FREQ Procedure

Number of Variable Levels			
Variable	Levels	Missing Levels	Nonmissing Levels
Internship	3	1	2

Internship	Frequency	Percent	Cumulative Frequency	Cumulative Percent
	2	20.00	2	20.00
yes	4	40.00	6	60.00
no	4	40.00	10	100.00

R Code and Output

```
prtm <- read.table(header = TRUE, text = '
 sex internship enrollment count
 1  boys       yes       yes    35
 2  boys       no        yes    14
 3 girls       yes       yes    32
 4 girls       no        yes    53
 5  boys       yes        NA    29
 6  boys       no         no    27
 7 girls       yes        no    10
 8 girls       no         no    23
 9 girls        NA        yes   25
 10 girls       NA        no    29')

res <- proc_freq(prtm, tables = v(internship),
                 options = v(nlevels, missing))
```

Variable	Levels	Missing Levels	Nonmissing Levels
internship	3	1	2

Table of internship					
internship	N	Frequency	Percent	Cumulative Frequency	Cumulative Percent
.	10	2	20.00	2	20.00
no	10	4	40.00	6	60.00
yes	10	4	40.00	10	100.00

Comparison	**Result**
Compare basic frequencies with nlevels and missing options.	Pass

proc_freq-004

SAS Code and Output

```
data SummerSchool;
  input Gender $ Internship $ Enrollment $ Count
@@;
  datalines;
boys  yes yes 35   boys  yes no 29
boys   no yes 14   boys   no no 27
girls yes yes 32   girls yes no 10
girls  no yes 53   girls  no no 23
;

proc freq data=SummerSchool order=data;
  tables Internship * Enrollment  / out = Fork chisq;
  by Gender;
  weight Count;
run;
```

The FREQ Procedure

Gender=boys

Frequency Percent Row Pct Col Pct	Table of Internship by Enrollment		
	Enrollment		
Internship	yes	no	Total
yes	35 33.33 54.69 71.43	29 27.62 45.31 51.79	64 60.95
no	14 13.33 34.15 28.57	27 25.71 65.85 48.21	41 39.05
Total	49 46.67	56 53.33	105 100.00

Statistics for Table of Internship by Enrollment

Statistic	DF	Value	Prob
Chi-Square	1	4.2366	0.0396
Likelihood Ratio Chi-Square	1	4.2903	0.0383
Continuity Adj. Chi-Square	1	3.4515	0.0632
Mantel-Haenszel Chi-Square	1	4.1963	0.0405
Phi Coefficient		0.2009	
Contingency Coefficient		0.1969	
Cramer's V		0.2009	

Fisher's Exact Test	
Cell (1,1) Frequency (F)	35
Left-sided Pr <= F	0.9885
Right-sided Pr >= F	0.0311
Table Probability (P)	0.0196
Two-sided Pr <= P	0.0467

Sample Size = 105

The FREQ Procedure

Gender=girls

Frequency Percent Row Pct Col Pct	Table of Internship by Enrollment		
	Enrollment		
Internship	yes	no	Total
yes	32 27.12 76.19 37.65	10 8.47 23.81 30.30	42 35.59
no	53 44.92 69.74 62.35	23 19.49 30.26 69.70	76 64.41
Total	85 72.03	33 27.97	118 100.00

Statistics for Table of Internship by Enrollment

Statistic	DF	Value	Prob
Chi-Square	1	0.5593	0.4546
Likelihood Ratio Chi-Square	1	0.5681	0.4510
Continuity Adj. Chi-Square	1	0.2848	0.5936
Mantel-Haenszel Chi-Square	1	0.5545	0.4565
Phi Coefficient		0.0688	
Contingency Coefficient		0.0687	
Cramer's V		0.0688	

Fisher's Exact Test	
Cell (1,1) Frequency (F)	32
Left-sided Pr <= F	0.8317
Right-sided Pr >= F	0.2994
Table Probability (P)	0.1311
Two-sided Pr <= P	0.5245

Sample Size = 118

R Code and Output

```
prt <- read.table(header = TRUE, text = '
 sex internship enrollment count
 1  boys       yes       yes    35
 2  boys        no       yes    14
 3 girls       yes       yes    32
 4 girls        no       yes    53
 5  boys       yes        no    29
 6  boys        no        no    27
 7 girls       yes        no    10
 8 girls        no        no    23')

prt$enrollment <- factor(prt$enrollment, c("yes",
"no"))
prt$internship <- factor(prt$internship, c("yes", "no"))

proc_freq(prt, tables = "internship * enrollment",
        options = ChiSq,
        by = "sex",
        weight = "count")
```

sex=boys, Table of internship by enrollment

		enrollment		
internship	Statistic	yes	no	Total
yes	Frequency	35	29	64
	Percent	33.33	27.62	60.95
	Row Pct	54.69	45.31	
	Col Pct	71.43	51.79	
no	Frequency	14	27	41
	Percent	13.33	25.71	39.05
	Row Pct	34.15	65.85	
	Col Pct	28.57	48.21	
Total	Frequency	49	56	105
	Percent	46.67	53.33	100.00

sex=boys, Chi-Square Test

Measure	Value
Chi-Square	4.2366
DF	1
PR>ChiSq	0.0396

sex=girls, Table of internship by enrollment

		enrollment		
internship	Statistic	yes	no	Total
yes	Frequency	32	10	42
	Percent	27.12	8.47	35.59
	Row Pct	76.19	23.81	
	Col Pct	37.65	30.30	
no	Frequency	53	23	76
	Percent	44.92	19.49	64.41
	Row Pct	69.74	30.26	
	Col Pct	62.35	69.70	
Total	Frequency	85	33	118
	Percent	72.03	27.97	100.00

sex=girls, Chi-Square Test

Measure	Value
Chi-Square	0.5593
DF	1
PR>ChiSq	0.4546

Comparison	**Result**
Compare two way with chisquare statistic.	Pass.

proc_freq-005

SAS Code and Output

```
data SummerSchool;
  input Gender $ Internship $ Enrollment $ Count
@@;
  datalines;
boys  yes yes 35   boys  yes no 29
boys   no yes 14   boys   no no 27
girls yes yes 32   girls yes no 10
girls  no yes 53   girls  no no 23
;

proc freq data=SummerSchool order=data;
  tables Internship * Enrollment  / out = Fork chisq;
  by Gender;
  weight Count;
run;
```

The FREQ Procedure

Gender=boys

Frequency Percent Row Pct Col Pct	Table of Internship by Enrollment		
	Enrollment		
Internship	yes	no	Total
yes	35 33.33 54.69 71.43	29 27.62 45.31 51.79	64 60.95
no	14 13.33 34.15 28.57	27 25.71 65.85 48.21	41 39.05
Total	49 46.67	56 53.33	105 100.00

Statistics for Table of Internship by Enrollment

Statistic	DF	Value	Prob
Chi-Square	1	4.2366	0.0396
Likelihood Ratio Chi-Square	1	4.2903	0.0383
Continuity Adj. Chi-Square	1	3.4515	0.0632
Mantel-Haenszel Chi-Square	1	4.1963	0.0405
Phi Coefficient		0.2009	
Contingency Coefficient		0.1969	
Cramer's V		0.2009	

Fisher's Exact Test	
Cell (1,1) Frequency (F)	35
Left-sided Pr <= F	0.9885
Right-sided Pr >= F	0.0311
Table Probability (P)	0.0196
Two-sided Pr <= P	0.0467

Sample Size = 105

The FREQ Procedure

Gender=girls

Frequency Percent Row Pct Col Pct	Table of Internship by Enrollment		
	Enrollment		
Internship	yes	no	Total
yes	32 27.12 76.19 37.65	10 8.47 23.81 30.30	42 35.59
no	53 44.92 69.74 62.35	23 19.49 30.26 69.70	76 64.41
Total	85 72.03	33 27.97	118 100.00

Statistics for Table of Internship by Enrollment

Statistic	DF	Value	Prob
Chi-Square	1	0.5593	0.4546
Likelihood Ratio Chi-Square	1	0.5681	0.4510
Continuity Adj. Chi-Square	1	0.2848	0.5936
Mantel-Haenszel Chi-Square	1	0.5545	0.4565
Phi Coefficient		0.0688	
Contingency Coefficient		0.0687	
Cramer's V		0.0688	

Fisher's Exact Test	
Cell (1,1) Frequency (F)	32
Left-sided Pr <= F	0.8317
Right-sided Pr >= F	0.2994
Table Probability (P)	0.1311
Two-sided Pr <= P	0.5245

Sample Size = 118

R Code and Output

```
prt <- read.table(header = TRUE, text = '
 sex internship enrollment count
 1  boys       yes       yes    35
 2  boys       no        yes    14
 3 girls       yes       yes    32
 4 girls       no        yes    53
 5  boys       yes       no     29
 6  boys       no        no     27
 7 girls       yes       no     10
 8 girls       no        no     23')

prt$enrollment <- factor(prt$enrollment, c("yes",
"no"))
prt$internship <- factor(prt$internship, c("yes", "no"))

proc_freq(prt, tables = "internship * enrollment",
        options = Fisher,
        by = "sex",
        weight = "count")
```

sex=boys, Table of internship by enrollment

		enrollment		
internship	Statistic	yes	no	Total
yes	Frequency	35	29	64
	Percent	33.33	27.62	60.95
	Row Pct	54.69	45.31	
	Col Pct	71.43	51.79	
no	Frequency	14	27	41
	Percent	13.33	25.71	39.05
	Row Pct	34.15	65.85	
	Col Pct	28.57	48.21	
Total	Frequency	49	56	105
	Percent	46.67	53.33	100.00

sex=boys, Fisher's Exact Test

Measure	Value
Cell1.1	35.0000
Left.Sided	0.9885
Right.Sided	0.0311
Two.Sided	0.0467

sex=girls, Table of internship by enrollment

		enrollment		
internship	Statistic	yes	no	Total
yes	Frequency	32	10	42
	Percent	27.12	8.47	35.59
	Row Pct	76.19	23.81	
	Col Pct	37.65	30.30	
no	Frequency	53	23	76
	Percent	44.92	19.49	64.41
	Row Pct	69.74	30.26	
	Col Pct	62.35	69.70	
Total	Frequency	85	33	118
	Percent	72.03	27.97	100.00

sex=girls, Fisher's Exact Test

Measure	Value
Cell1.1	32.0000
Left.Sided	0.8317
Right.Sided	0.2994
Two.Sided	0.5245

Comparison	**Result**
Compare two way with Fisher’s Exact statistic.	Pass.

Means Data

SAS

```
data cake;
  input LastName $ 1-12 Age 13-14 PresentScore 16-17
      TasteScore 19-20 Flavor $ 23-32 Layers 34 ;
  datalines;
Orlando     27 93 80  Vanilla    1
Ramey       32 84 72  Rum        2
Goldston    46 68 75  Vanilla    1
Roe         38 79 73  Vanilla    2
Larsen      23 77 84  Chocolate  3
Davis       51 86 91  Spice      3
Strickland  19 82 79  Chocolate  1
Nguyen      57 77 84  Vanilla    3
Hildenbrand 33 81 83  Chocolate  1
Byron       62 72 87  Vanilla    2
Sanders     26 56 79  Chocolate  1
Jaeger      43 66 74             1
Davis       28 69 75  Chocolate  2
Conrad      69 85 94  Vanilla    1
Walters     55 67 72  Chocolate  2
Rossburger  28 78 81  Spice      2
Matthew     42 81 92  Chocolate  2
Becker      36 62 83  Spice      2
Anderson    27 87 85  Chocolate  1
Merritt     62 73 84  Chocolate  1
;
```

R

```
datm <- read.table(header = TRUE, text = '
LastName  Age PresentScore TasteScore Flavor Layers
Orlando     27 93 80  Vanilla    1
Ramey       32 84 72  Rum        2
Goldston    46 68 75  Vanilla    1
Roe         38 79 73  Vanilla    2
Larsen      23 77 84  Chocolate  3
Davis       51 86 91  Spice      3
Strickland  19 82 79  Chocolate  1
Nguyen      57 77 84  Vanilla    3
Hildenbrand 33 81 83  Chocolate  1
Byron       62 72 87  Vanilla    2
Sanders     26 56 79  Chocolate  1
Jaeger      43 66 74  NA         1
Davis       28 69 75  Chocolate  2
Conrad      69 85 94  Vanilla    1
Walters     55 67 72  Chocolate  2
Rossburger  28 78 81  Spice      2
Matthew     42 81 92  Chocolate  2
Becker      36 62 83  Spice      2
Anderson    27 87 85  Chocolate  1
Merritt     62 73 84  Chocolate  1
')
```

** Above input data will be used for all means comparisons unless data is provided in the script itself.

proc_means-001

SAS Code and Output

```
proc means data=cake  n mean median stddev min max
maxdec=4;
  var Age PresentScore TasteScore layers;
  title 'Summary of Presentation and Taste Scores';
  output out=outdata;
run;
```

Summary of Presentation and Taste Scores

The MEANS Procedure

Variable	N	Mean	Median	Std Dev	Minimum	Maximum
Age	20	40.2000	37.0000	14.8274	19.0000	69.0000
PresentScore	20	76.1500	77.5000	9.3768	56.0000	93.0000
TasteScore	20	81.3500	82.0000	6.6116	72.0000	94.0000
Layers	20	1.7000	2.0000	0.7327	1.0000	3.0000

R Code and Output

```
proc_means(datm, var = v(Age, PresentScore, TasteScore, Layers),
        stats = c("n", "mean", "median", "std", "min", "max"),
        options = v(maxdec = 4),
        titles = "Summary of Presentation and Taste Scores")
```

Summary of Presentation and Taste Scores

Variable	N	Mean	Median	Std Dev	Minimum	Maximum
Age	20	40.2000	37.0000	14.8274	19	69
PresentScore	20	76.1500	77.5000	9.3768	56	93
TasteScore	20	81.3500	82.0000	6.6116	72	94
Layers	20	1.7000	2.0000	0.7327	1	3

Comparison	**Result**
Simple Proc means on multiple variables.	Pass

proc_means-002

SAS Code and Output

```
proc means data=cake  nmiss median mode lclm uclm stderr
maxdec=4;
  var Age PresentScore TasteScore layers;
  title 'Summary of Presentation and Taste Scores';
  output out=outdata;
run;
```

Summary of Presentation and Taste Scores

The MEANS Procedure

Variable	N Miss	Median	Mode	Lower 95% CL for Mean	Upper 95% CL for Mean	Std Error
Age	0	37.0000	27.0000	33.2605	47.1395	3.3155
PresentScore	0	77.5000	77.0000	71.7615	80.5385	2.0967
TasteScore	0	82.0000	84.0000	78.2557	84.4443	1.4784
Layers	0	2.0000	1.0000	1.3571	2.0429	0.1638

R Code and Output

```
proc_means(datm, v(Age, PresentScore, TasteScore, Layers),
        stats = v(nmiss, median, mode, clm, stderr),
        options = v(maxdec = 4),
        titles = "Summary of Presentation and Taste Scores")
```

Summary of Presentation and Taste Scores

Variable	NMiss	Median	Mode	Lower Conf. Limit	Upper Conf. Limit	Std Err
Age	0	37.0000	27	33.2605	47.1395	3.3155
PresentScore	0	77.5000	77	71.7615	80.5385	2.0967
TasteScore	0	82.0000	84	78.2557	84.4443	1.4784
Layers	0	2.0000	1	1.3571	2.0429	0.1638

Comparison	**Result**
Comparison of more statistics options.	Pass

proc_means-003

SAS Code and Output

```
proc sort data=cake out=cake2;
by Layers;
run;

proc means data=cake2  n mean median stddev min max maxdec=4;
  var Age PresentScore TasteScore;
  title 'Summary of Presentation and Taste Scores';
  by Layers;
  output out=outdata;
run;
```

Summary of Presentation and Taste Scores

The MEANS Procedure

Layers=1

Variable	N	Mean	Median	Std Dev	Minimum	Maximum
Age	9	39.1111	33.0000	17.2731	19.0000	69.0000
PresentScore	9	76.7778	81.0000	11.8298	56.0000	93.0000
TasteScore	9	81.4444	80.0000	6.0231	74.0000	94.0000

Layers=2

Variable	N	Mean	Median	Std Dev	Minimum	Maximum
Age	8	40.1250	37.0000	12.4492	28.0000	62.0000
PresentScore	8	74.0000	75.0000	7.6718	62.0000	84.0000
TasteScore	8	79.3750	78.0000	7.5770	72.0000	92.0000

Layers=3

Variable	N	Mean	Median	Std Dev	Minimum	Maximum
Age	3	43.6667	51.0000	18.1475	23.0000	57.0000
PresentScore	3	80.0000	77.0000	5.1962	77.0000	86.0000
TasteScore	3	86.3333	84.0000	4.0415	84.0000	91.0000

R Code and Output

```
proc_means(datm, var = v(Age, PresentScore, TasteScore),
        stats = c("n", "mean", "median", "std", "min", "max"),
        by = "Layers",
        options = v(maxdec = 4),
        titles = "Summary of Presentation and Taste Scores")
```

Summary of Presentation and Taste Scores

Layers=1

Variable	N	Mean	Median	Std Dev	Minimum	Maximum
Age	9	39.1111	33	17.2731	19	69
PresentScore	9	76.7778	81	11.8298	56	93
TasteScore	9	81.4444	80	6.0231	74	94

Layers=2

Variable	N	Mean	Median	Std Dev	Minimum	Maximum
Age	8	40.1250	37.0000	12.4492	28	62
PresentScore	8	74.0000	75.0000	7.6718	62	84
TasteScore	8	79.3750	78.0000	7.5770	72	92

Layers=3

Variable	N	Mean	Median	Std Dev	Minimum	Maximum
Age	3	43.6667	51	18.1475	23	57
PresentScore	3	80.0000	77	5.1962	77	86
TasteScore	3	86.3333	84	4.0415	84	91

Comparison	**Result**
Comparison of by variable.	Pass

proc_means-004

SAS Code and Output

```
proc sort data=cake out=cake2;
by Layers;
run;

proc means data=cake2  n mean median stddev min max
maxdec=4;
  var Age PresentScore TasteScore;
  title 'Summary of Presentation and Taste Scores';
  class Layers;
  output out=outdata;
run;
```

Summary of Presentation and Taste Scores

The MEANS Procedure

Layers	N Obs	Variable	N	Mean	Median	Std Dev	Minimum	Maximum
1	9	Age	9	39.1111	33.0000	17.2731	19.0000	69.0000
		PresentScore	9	76.7778	81.0000	11.8298	56.0000	93.0000
		TasteScore	9	81.4444	80.0000	6.0231	74.0000	94.0000
2	8	Age	8	40.1250	37.0000	12.4492	28.0000	62.0000
		PresentScore	8	74.0000	75.0000	7.6718	62.0000	84.0000
		TasteScore	8	79.3750	78.0000	7.5770	72.0000	92.0000
3	3	Age	3	43.6667	51.0000	18.1475	23.0000	57.0000
		PresentScore	3	80.0000	77.0000	5.1962	77.0000	86.0000
		TasteScore	3	86.3333	84.0000	4.0415	84.0000	91.0000

R Code and Output

```
proc_means(datm, var = v(Age, PresentScore, TasteScore),
        stats = c("n", "mean", "median", "std", "min",
"max"),
        class = "Layers",
        options = v(maxdec = 4),
        titles = "Summary of Presentation and Taste Scores")
```

Summary of Presentation and Taste Scores

Layers	Variable	N	Mean	Median	Std Dev	Minimum	Maximum
1	Age	9	39.1111	33.0000	17.2731	19	69
1	PresentScore	9	76.7778	81.0000	11.8298	56	93
1	TasteScore	9	81.4444	80.0000	6.0231	74	94
2	Age	8	40.1250	37.0000	12.4492	28	62
2	PresentScore	8	74.0000	75.0000	7.6718	62	84
2	TasteScore	8	79.3750	78.0000	7.5770	72	92
3	Age	3	43.6667	51.0000	18.1475	23	57
3	PresentScore	3	80.0000	77.0000	5.1962	77	86
3	TasteScore	3	86.3333	84.0000	4.0415	84	91

Comparison	**Result**
Comparison of class variable.	Pass

proc_means-005

SAS Code and Output

```
data cake2;
set cake;
if (_n_ = 5) then do;
PresentScore = .;
end;
run;

proc means data=cake2  n nmiss mean median lclm uclm stddev
maxdec=4;
  var PresentScore TasteScore;
  title 'Summary of Presentation and Taste Scores';
  output out=outdata;
run;
```

Summary of Presentation and Taste Scores

The MEANS Procedure

Variable	N	N Miss	Mean	Median	Lower 95% CL for Mean	Upper 95% CL for Mean	Std Dev
PresentScore	19	1	76.1053	78.0000	71.4630	80.7475	9.6315
TasteScore	20	0	81.3500	82.0000	78.2557	84.4443	6.6116

R Code and Output

```
datm2 <- datm

datm2[5, "PresentScore"] <- NA

proc_means(datm2, var = c("PresentScore", "TasteScore"),
        stats = c("n", "nmiss", "mean", "median", "mode", "clm",
"std"),
        titles = "My first title for Means",
        options = v(notype, nonobs, maxdec=4))
```

My first title for Means

Variable	N	NMiss	Mean	Median	Mode	Lower Conf. Limit	Upper Conf. Limit	Std Dev
PresentScore	19	1	76.1053	78.0000	81	71.4630	80.7475	9.6315
TasteScore	20	0	81.3500	82.0000	84	78.2557	84.4443	6.6116

Comparison	**Result**
Comparison of class variable.	Pass

proc_means-006

SAS Code and Output	

proc means data=cake css cv lclm mode n maxdec=4;
  var Age PresentScore TasteScore layers;
  title 'Summary of Presentation and Taste Scores';
  output out=outdata;
run;
	Summary of Presentation and Taste Scores
	The MEANS Procedure
	Variable | Corrected SS | Coeff of Variation | Lower 95% CL for Mean | Mode | N
	Age | 4177.2 | 36.8842 | 34.4670 | 27.0000 | 20
	PresentScore | 1670.6 | 12.3135 | 72.5245 | 77.0000 | 20
	TasteScore | 830.6 | 8.1273 | 78.7937 | 84.0000 | 20
	Layers | 10.2000 | 43.0997 | 1.4167 | 1.0000 | 20

R Code and Output	

proc_means(datm, var = v(Age, PresentScore, TasteScore, Layers),
  stats = c("css", "cv", "lclm", "mode", "nobs"),
  options = v(maxdec = 4),
  titles = "Summary of Presentation and Taste Scores")
	Summary of Presentation and Taste Scores
	Variable | Corrected SS | Coeff of Variation | Lower 95% CL for Mean | Mode | NOBS
	Age | 4177.2000 | 36.8842 | 34.4670 | 27 | 20
	PresentScore | 1670.5500 | 12.3135 | 72.5245 | 77 | 20
	TasteScore | 830.5500 | 8.1273 | 78.7937 | 84 | 20
	Layers | 10.2000 | 43.0997 | 1.4167 | 1 | 20

Comparison	**Result**
Comparison of more statistics options.	Pass, except SAS doesn't have NOBS keyword.

proc_means-007

SAS Code and Output

```
proc means data=cake p1 p5 p10 p20 p25 p30 p40 p50
maxdec=4;
  var Age PresentScore TasteScore layers;
  title 'Summary of Presentation and Taste Scores';
  output out=outdata;
run;
```

Summary of Presentation and Taste Scores

The MEANS Procedure

Variable	1st Pctl	5th Pctl	10th Pctl	20th Pctl	25th Pctl	30th Pctl	40th Pctl	50th Pctl
Age	19.0000	21.0000	24.5000	27.0000	27.5000	28.0000	32.5000	37.0000
PresentScore	56.0000	59.0000	64.0000	67.5000	68.5000	70.5000	75.0000	77.5000
TasteScore	72.0000	72.0000	72.5000	74.5000	75.0000	77.0000	79.5000	82.0000
Layers	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	2.0000

R Code and Output

```
proc_means(datm, var = v(Age, PresentScore, TasteScore,
Layers),
        stats = c("p1", "p5", "p10", "p20", "p25", "p30", "p40",
"p50"),
        options = v(maxdec = 4),
        titles = "Summary of Presentation and Taste Scores")
```

Summary of Presentation and Taste Scores

Variable	1st Pctl	5th Pctl	10th Pctl	20th Pctl	25th Pctl	30th Pctl	40th Pctl	50th Pctl
Age	19.0000	21.0000	24.5000	27.0000	27.5000	28.0000	32.5000	37.0000
PresentScore	56.0000	59.0000	64.0000	67.5000	68.5000	70.5000	75.0000	77.5000
TasteScore	72.0000	72.0000	72.5000	74.5000	75.0000	77.0000	79.5000	82.0000
Layers	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	2.0000

Comparison	**Result**
Comparison of more statistics options.	Pass

proc_means-008

SAS Code and Output

```
proc means data=cake  p60 p70 p75 p80 p90 p95 p99 maxdec=4;
  var Age PresentScore TasteScore layers;
  title 'Summary of Presentation and Taste Scores';
  output out=outdata;
run;
```

Summary of Presentation and Taste Scores

The MEANS Procedure

Variable	60th Pctl	70th Pctl	75th Pctl	80th Pctl	90th Pctl	95th Pctl	99th Pctl
Age	42.5000	48.5000	53.0000	56.0000	62.0000	65.5000	69.0000
PresentScore	80.0000	81.5000	83.0000	84.5000	86.5000	90.0000	93.0000
TasteScore	83.5000	84.0000	84.5000	86.0000	91.5000	93.0000	94.0000
Layers	2.0000	2.0000	2.0000	2.0000	3.0000	3.0000	3.0000

R Code and Output

```
proc_means(datm, var = v(Age, PresentScore, TasteScore, Layers),
        stats = c("p60", "p70", "p75", "p80", "p90", "p95", "p99"),
        options = v(maxdec = 4),
        titles = "Summary of Presentation and Taste Scores")
```

Summary of Presentation and Taste Scores

Variable	60th Pctl	70th Pctl	75th Pctl	80th Pctl	90th Pctl	95th Pctl	99th Pctl
Age	42.5000	48.5000	53.0000	56.0000	62.0000	65.5000	69.0000
PresentScore	80.0000	81.5000	83.0000	84.5000	86.5000	90.0000	93.0000
TasteScore	83.5000	84.0000	84.5000	86.0000	91.5000	93.0000	94.0000
Layers	2.0000	2.0000	2.0000	2.0000	3.0000	3.0000	3.0000

Comparison	**Result**
Comparison of more statistics options.	Pass

proc_means-009

SAS Code and Output	

proc means data=cake q1 q3 qrange range sum uclm var maxdec=4;
  var Age PresentScore TasteScore layers;
  title 'Summary of Presentation and Taste Scores';
  output out=outdata;
run;
	Summary of Presentation and Taste Scores
	The MEANS Procedure
	Variable | Lower Quartile | Upper Quartile | Quartile Range | Range | Sum | Upper 95% CL for Mean | Variance
	Age | 27.5000 | 53.0000 | 25.5000 | 50.0000 | 804.0 | 45.9330 | 219.9
	PresentScore | 68.5000 | 83.0000 | 14.5000 | 37.0000 | 1523.0 | 79.7755 | 87.9237
	TasteScore | 75.0000 | 84.5000 | 9.5000 | 22.0000 | 1627.0 | 83.9063 | 43.7132
	Layers | 1.0000 | 2.0000 | 1.0000 | 2.0000 | 34.0000 | 1.9833 | 0.5368

R Code and Output	

proc_means(datm, var = v(Age, PresentScore, TasteScore, Layers),
  stats = c("q1", "q3", "qrange", "range", "sum", "uclm", "vari"),
  options = v(maxdec = 4),
  titles = "Summary of Presentation and Taste Scores")
	Summary of Presentation and Taste Scores
	Variable | Lower Quartile | Upper Quartile | Quartile Range | Range | Sum | Upper 95% CL for Mean | Variance
	Age | 27.5000 | 53.0000 | 25.5000 | 50 | 804 | 45.9330 | 219.8526
	PresentScore | 68.5000 | 83.0000 | 14.5000 | 37 | 1523 | 79.7755 | 87.9237
	TasteScore | 75.0000 | 84.5000 | 9.5000 | 22 | 1627 | 83.9063 | 43.7132
	Layers | 1.0000 | 2.0000 | 1.0000 | 2 | 34 | 1.9833 | 0.5368

Comparison	**Result**
Comparison of more statistics options.	Pass

proc_means-010

SAS Code and Output	

proc means data=cake mean clm alpha = 0.1 maxdec=4;
  var PresentScore TasteScore;
  title 'Summary of Presentation and Taste Scores';
  output out=outdata;
run;
	Summary of Presentation and Taste Scores
	The MEANS Procedure
	Variable | Mean | Lower 90% CL for Mean | Upper 90% CL for Mean
	PresentScore | 76.1500 | 72.5245 | 79.7755
	TasteScore | 81.3500 | 78.7937 | 83.9063

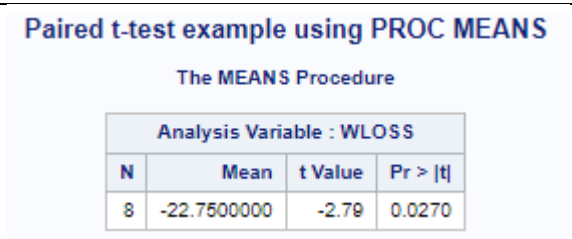
R Code and Output	
proc_means(datm, var = v(PresentScore, TasteScore), stats = v(mean, clm), options = v(alpha = 0.1, maxdec=4), titles = "Summary of Presentation and Taste Scores")	**Summary of Presentation and Taste Scores** Variable | Mean | Lower 90% CL for Mean | Upper 90% CL for Mean PresentScore | 76.1500 | 72.5245 | 79.7755 TasteScore | 81.3500 | 78.7937 | 83.9063
Comparison	**Result**
Comparison of 90% confidence limit with alpha option.	Pass

proc_means-011

SAS Code and Output

```
DATA WEIGHT;
      INPUT WBEFORE WAFTER;
      * Calculate WLOSS in the DATA step *;
      WLOSS=WAFTER-WBEFORE;
      DATALINES;
      200 190
      175 154
      188 176
      198 193
      197 198
      310 240
      245 204
      202 178
      ;
run;

PROC MEANS N MEAN T PRT;
  VAR WLOSS;
  TITLE 'Paired t-test example using PROC MEANS';
RUN;
```



Paired t-test example using PROC MEANS

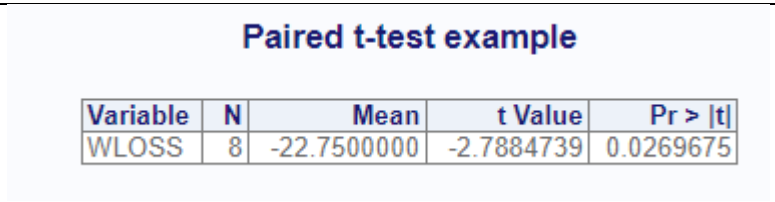
The MEANS Procedure

Analysis Variable : WLOSS			
N	Mean	t Value	Pr > |t|
8	-22.7500000	-2.79	0.0270

R Code and Output

```
datp <- read.table(header = TRUE, text = '
  WBEFORE WAFTER
  200 190
  175 154
  188 176
  198 193
  197 198
  310 240
  245 204
  202 178')
datp$WLOSS <- datp$WAFTER - datp$WBEFORE

res <- proc_means(datp, var = WLOSS,
                  stats = v(n, mean, t, prt),
                  titles = c("Paired t-test example"))
```



Paired t-test example

Variable	N	Mean	t Value	Pr > |t|
WLOSS	8	-22.7500000	-2.7884739	0.0269675

Comparison	**Result**
Comparison of hypothesis tests.	Pass

proc_means-012

SAS Code and Output	

proc means data=cake n mean uss;
  var Age PresentScore TasteScore Layers;
  title 'Summary of Presentation and Taste Scores';
  output out=outdata;
run;
	Summary of Presentation and Taste Scores
	The MEANS Procedure
	Variable | N | Mean | Uncorrected SS
	Age | 20 | 40.2000000 | 36498.00
	PresentScore | 20 | 76.1500000 | 117647.00
	TasteScore | 20 | 81.3500000 | 133187.00
	Layers | 20 | 1.7000000 | 68.0000000

R Code and Output	
proc_means(datm, var = v(Age, PresentScore, TasteScore, Layers), stats = v(n, mean, uss), titles = c("Test"))	**Test** Variable | N | Mean | Uncorrected SS Age | 20 | 40.2000000 | 36498.0000000 PresentScore | 20 | 76.1500000 | 117647.0000000 TasteScore | 20 | 81.3500000 | 133187.0000000 Layers | 20 | 1.7000000 | 68.0000000
Comparison	**Result**
Test USS statistics.	Pass

proc_means-013

SAS Code and Output

```
proc means data=cake  n mean skewness kurtosis maxdec = 8;
  var  Age PresentScore TasteScore Layers;
  title 'Summary of Presentation and Taste Scores';
  output out=outdata;
run;
```

Summary of Presentation and Taste Scores

The MEANS Procedure

Variable	N	Mean	Skewness	Kurtosis
Age	20	40.20000000	0.49070401	-0.96131045
PresentScore	20	76.15000000	-0.36762986	-0.30448351
TasteScore	20	81.35000000	0.25698724	-0.70821594
Layers	20	1.70000000	0.55306602	-0.83361603

R Code and Output

```
proc_means(datm, var = v(Age, PresentScore, TasteScore, Layers),
                stats = v(n, mean,  skew, kurt),
                titles = c("Test"),
                options = c(maxdec = 8))
```

Test

Variable	N	Mean	Skewness	Kurtosis
Age	20	40.20000000	0.49070401	-0.96131045
PresentScore	20	76.15000000	-0.36762986	-0.30448351
TasteScore	20	81.35000000	0.25698724	-0.70821594
Layers	20	1.70000000	0.55306602	-0.83361603

Comparison	Result
Test Skewness and Kurtosis	Pass

proc_means-014

SAS Data

```
data cake2;
  input LastName $ 1-12 Age 13-14 PresentScore 16-17
      TasteScore 19-20 Flavor $ 23-32 Layers 34 Region 36;
  datalines;
Orlando     27 93 80  Vanilla    1 1
Ramey       32 84 72  Rum        2 1
Goldston    46 68 75  Vanilla    1 1
Roe         38 79 73  Vanilla    2 1
Larsen      23 77 84  Chocolate  3 1
Davis       51 86 91  Spice      3 1
Strickland  19 82 79  Chocolate  1 1
Nguyen      57 77 84  Vanilla    3 1
Hildenbrand 33 81 83  Chocolate  1 1
Byron       62 72 87  Vanilla    2 1
Sanders     26 56 79  Chocolate  1 2
Jaeger      43 66 74             1 2
Davis       28 69 75  Chocolate  2 2
Conrad      69 85 94  Vanilla    1 2
Walters     55 67 72  Chocolate  2 2
Rossburger  28 78 81  Spice      2 2
Matthew     42 81 92  Chocolate  2 2
Becker      36 62 83  Spice      2 2
Anderson    27 87 85  Chocolate  1 2
Merritt     62 73 84  Chocolate  1 2
;
```

SAS Code and Output

R Data

```
datmr <- read.table(header = TRUE, text = '
 LastName  Age PresentScore TasteScore Flavor Layers Region
 Orlando     27 93 80  Vanilla    1 1
 Ramey       32 84 72  Rum        2 1
 Goldston    46 68 75  Vanilla    1 1
 Roe         38 79 73  Vanilla    2 1
 Larsen      23 77 84  Chocolate  3 1
 Davis       51 86 91  Spice      3 1
 Strickland  19 82 79  Chocolate  1 1
 Nguyen      57 77 84  Vanilla    3 1
 Hildenbrand 33 81 83  Chocolate  1 1
 Byron       62 72 87  Vanilla    2 1
 Sanders     26 56 79  Chocolate  1 2
 Jaeger      43 66 74  NA         1 2
 Davis       28 69 75  Chocolate  2 2
 Conrad      69 85 94  Vanilla    1 2
 Walters     55 67 72  Chocolate  2 2
 Rossburger  28 78 81  Spice      2 2
 Matthew     42 81 92  Chocolate  2 2
 Becker      36 62 83  Spice      2 2
 Anderson    27 87 85  Chocolate  1 2
 Merritt     62 73 84  Chocolate  1 2
 ')
```

```
proc means data=cake2  n min max mean stddev maxdec=4;
  var Age PresentScore TasteScore;
  title 'Summary of Presentation and Taste Scores';
  class Region Layers;
  output out=mdata;
run;
```

Summary of Presentation and Taste Scores

The MEANS Procedure

Region	Layers	N Obs	Variable	N	Minimum	Maximum	Mean	Std Dev
1	1	4	Age	4	19.0000	46.0000	31.2500	11.3835
			PresentScore	4	68.0000	93.0000	81.0000	10.2307
			TasteScore	4	75.0000	83.0000	79.2500	3.3040
	2	3	Age	3	32.0000	62.0000	44.0000	15.8745
			PresentScore	3	72.0000	84.0000	78.3333	6.0277
			TasteScore	3	72.0000	87.0000	77.3333	8.3865
	3	3	Age	3	23.0000	57.0000	43.6667	18.1475
			PresentScore	3	77.0000	86.0000	80.0000	5.1962
			TasteScore	3	84.0000	91.0000	86.3333	4.0415
2	1	5	Age	5	26.0000	69.0000	45.4000	19.7053
			PresentScore	5	56.0000	87.0000	73.4000	13.0115
			TasteScore	5	74.0000	94.0000	83.2000	7.4632
	2	5	Age	5	28.0000	55.0000	37.8000	11.2783
			PresentScore	5	62.0000	81.0000	71.4000	7.8930
			TasteScore	5	72.0000	92.0000	80.6000	7.7653

R Code and Output

```
res <- proc_means(datmr,
            var = c("Age", "PresentScore", "TasteScore"),
            stats = c("n", "min", "max","mean", "std"),
            output = c("out", "report"),
            class = c("Region", "Layers"),
            titles = "My first title for Means",
            options = c("long", maxdec = 4))
```

My first title for Means

Region	Layers	Variable	N	Minimum	Maximum	Mean	Std Dev
1	1	Age	4	19	46	31.2500	11.3835
1	1	PresentScore	4	68	93	81.0000	10.2307
1	1	TasteScore	4	75	83	79.2500	3.3040
1	2	Age	3	32	62	44.0000	15.8745
1	2	PresentScore	3	72	84	78.3333	6.0277
1	2	TasteScore	3	72	87	77.3333	8.3865
1	3	Age	3	23	57	43.6667	18.1475
1	3	PresentScore	3	77	86	80.0000	5.1962
1	3	TasteScore	3	84	91	86.3333	4.0415
2	1	Age	5	26	69	45.4000	19.7053
2	1	PresentScore	5	56	87	73.4000	13.0115
2	1	TasteScore	5	74	94	83.2000	7.4632
2	2	Age	5	28	55	37.8000	11.2783
2	2	PresentScore	5	62	81	71.4000	7.8930
2	2	TasteScore	5	72	92	80.6000	7.7653

Comparison	**Result**
Test multiple class variables	Pass

proc_means-015

SAS Data	R Data

data cake;
  input LastName $ 1-12 Age 13-14 PresentScore 16-17
  TasteScore 19-20 Flavor $ 23-32 Layers 34 Region 36;
  datalines;
Orlando 27 93 80 Vanilla 3 1
Ramey 32 84 72 Rum 2 1
Goldston 46 68 75 Vanilla 1 1
Roe 38 79 73 Vanilla 2 1
Larsen 23 77 84 Chocolate 3 1
Davis 51 86 91 Spice 3 1
Strickland 19 82 79 Chocolate 1 1
Nguyen 57 77 84 Vanilla 3 1
Hildenbrand 33 81 83 Chocolate 1 1
Byron 62 72 87 Vanilla 2 1
Sanders 26 56 79 Chocolate 1 1
Jaeger 43 66 74 1 2
Davis 28 69 75 Chocolate 2 2
Conrad 69 85 94 Vanilla 1 2
Walters 55 67 72 Chocolate 2 2
Rossburger 28 78 81 Spice 2 2
Matthew 42 81 92 Chocolate 2 2
Becker 36 62 83 Spice 2 2
Anderson 27 87 85 Chocolate 1 2
Merritt 62 73 84 Chocolate 1 2
;
	datsp <- datm
	datsp$Layers[1] <- 3

SAS Code and Output	

```
proc sort data=cake out=cake2;
by Layers;
run;

proc means data=cake2  kurt skew cv clm maxdec = 8;
  var Age PresentScore TasteScore;
  title 'Summary of Presentation and Taste Scores';
  by Layers;

  output out=mdata;
run;
```

Summary of Presentation and Taste Scores

The MEANS Procedure

Layers=1

Variable	Kurtosis	Skewness	Coeff of Variation	Lower 95% CL for Mean	Upper 95% CL for Mean
Age	-0.95194732	0.55802418	43.85482852	25.73043112	55.51956888
PresentScore	-0.70369508	-0.60221998	14.51013940	65.68224186	83.81775814
TasteScore	0.98276993	0.85172466	7.85651169	76.26370010	86.98629990

Layers=2

Variable	Kurtosis	Skewness	Coeff of Variation	Lower 95% CL for Mean	Upper 95% CL for Mean
Age	-0.24886914	0.94617635	31.02599979	29.71722304	50.53277696
PresentScore	-1.23207654	-0.28347268	10.36735257	67.58618050	80.41381950
TasteScore	-1.06156373	0.59659254	9.54580892	73.04048132	85.70951868

Layers=3

Variable	Kurtosis	Skewness	Coeff of Variation	Lower 95% CL for Mean	Upper 95% CL for Mean
Age	-5.12632751	0.05902707	43.03797468	12.44920641	66.55079359
PresentScore	-2.34699816	0.65484802	9.32382865	70.89878690	95.60121310
TasteScore	2.00892049	0.94865417	5.39642979	77.47258189	92.02741811

R Code and Output

```
proc_means(datsp, var = c("Age", "PresentScore", "TasteScore"),
           stats = c("kurtosis", "skew", "cv", "clm"),
           by = c("Layers"),
           options = c(maxdec = 8)
           )
```

Layers=1

Variable	Kurtosis	Skewness	Coeff of Variation	Lower 95% CL for Mean	Upper 95% CL for Mean
Age	-0.95194732	0.55802418	43.85482852	25.73043112	55.51956888
PresentScore	-0.70369508	-0.60221998	14.51013940	65.68224186	83.81775814
TasteScore	0.98276993	0.85172466	7.85651169	76.26370010	86.98629990

Layers=2

Variable	Kurtosis	Skewness	Coeff of Variation	Lower 95% CL for Mean	Upper 95% CL for Mean
Age	-0.24886914	0.94617635	31.02599979	29.71722304	50.53277696
PresentScore	-1.23207654	-0.28347268	10.36735257	67.58618050	80.41381950
TasteScore	-1.06156373	0.59659254	9.54580892	73.04048132	85.70951868

Layers=3

Variable	Kurtosis	Skewness	Coeff of Variation	Lower 95% CL for Mean	Upper 95% CL for Mean
Age	-5.12632751	0.05902707	43.03797468	12.44920641	66.55079359
PresentScore	-2.34699816	0.65484802	9.32382865	70.89878690	95.60121310
TasteScore	2.00892049	0.94865417	5.39642979	77.47258189	92.02741811

Comparison	**Result**
Test more complicated statistics with by variables and low sample sizes.	Pass

Transpose Data

SAS	R

data score;
  input Student $9. +1 StudentID $ Section $ Test1 Test2 Final;
  datalines;
Capalleti 0545 1 94 91 87
Dubose 1252 2 51 65 91
Engles 1167 1 95 97 97
Grant 1230 2 63 75 80
Krupski 2527 2 80 76 71
Lundsford 4860 1 92 40 86
McBane 0674 1 75 78 72
;
	score <- read.table(header = TRUE,
	colClasses = c(Student = "character", StudentID = "character",
	Section = "character"), text = '
	Student StudentID Section Test1 Test2 Final
	Capalleti "0545" "1" 94 91 87
	Dubose "1252" "2" 51 65 91
	Engles "1167" "1" 95 97 97
	Grant "1230" "2" 63 75 80
	Krupski "2527" "2" 80 76 71
	Lundsford "4860" "1" 92 40 86
	McBane "0674" "1" 75 78 72
	', stringsAsFactors = FALSE)

proc_transpose-001

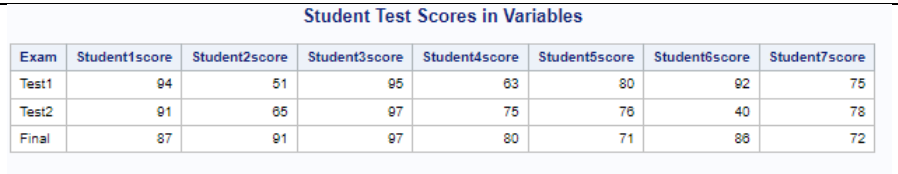
SAS Code and Output	

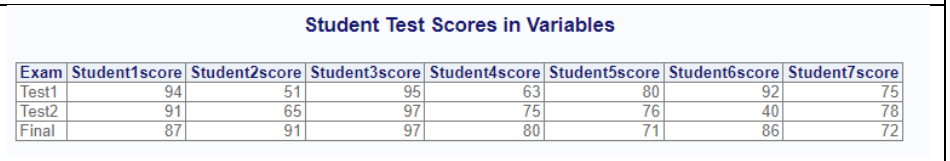
proc transpose data=score out=score_transposed name = Exam;
id Student;
run;
proc print data=score_transposed noobs;
  title 'Student Test Scores in Variables';
run;
	Student Test Scores in Variables
	
	Exam | Capalleti | Dubose | Engles | Grant | Krupski | Lundsford | McBane
	Test1 | 94 | 51 | 95 | 63 | 80 | 92 | 75
	Test2 | 91 | 65 | 97 | 75 | 76 | 40 | 78
	Final | 87 | 91 | 97 | 80 | 71 | 86 | 72

R Code and Output	
res <- proc_transpose(score, var = c("Test1", "Test2", "Final"), id = Student, name = "Exam") proc_print(res, titles = "Student Test Scores in Variables")	**Student Test Scores in Variables** Exam | Capalleti | Dubose | Engles | Grant | Krupski | Lundsford | McBane Test1 | 94 | 51 | 95 | 63 | 80 | 92 | 75 Test2 | 91 | 65 | 97 | 75 | 76 | 40 | 78 Final | 87 | 91 | 97 | 80 | 71 | 86 | 72
Comparison	**Results**
Perform basic transpose with ID variable and name.	Pass

proc_transpose-002

SAS Code and Output	

proc transpose data=score out=score_transposed
  prefix = Student suffix= score name = Exam;
run;
proc print data=score_transposed noobs;
  title 'Student Test Scores in Variables';
 	
run;
	Student Test Scores in Variables
	 
	Exam | Student1score | Student2score | Student3score | Student4score | Student5score | Student6score | Student7score
	Test1 | 94 | 51 | 95 | 63 | 80 | 92 | 75
	Test2 | 91 | 65 | 97 | 75 | 76 | 40 | 78
	Final | 87 | 91 | 97 | 80 | 71 | 86 | 72

R Code and Output	
res <- proc_transpose(score, var = c("Test1", "Test2", "Final"), name = "Exam", prefix = "Student", suffix = "score") proc_print(res, titles = "Student Test Scores in Variables")	**Student Test Scores in Variables**  Exam | Student1score | Student2score | Student3score | Student4score | Student5score | Student6score | Student7score Test1 | 94 | 51 | 95 | 63 | 80 | 92 | 75 Test2 | 91 | 65 | 97 | 75 | 76 | 40 | 78 Final | 87 | 91 | 97 | 80 | 71 | 86 | 72
Comparison	**Results**
Perform basic transpose with prefix and suffix.	Pass

proc_transpose-003

SAS Code and Output

```
proc sort data = score out = score_sorted;
by Section;
run;

proc transpose data=score_sorted out=score_transposed name =
Exam;
  id Student;
  by Section;
run;
proc print data=score_transposed noobs;
   title 'Student Test Scores in Variables';

run;
```

Student Test Scores in Variables

Section	Exam	Capalleti	Engles	Lundsford	McBane	Dubose	Grant	Krupski
1	Test1	94	95	92	75	.	.	.
1	Test2	91	97	40	78	.	.	.
1	Final	87	97	86	72	.	.	.
2	Test1	.	.	.	.	51	63	80
2	Test2	.	.	.	.	65	75	76
2	Final	.	.	.	.	91	80	71

R Code and Output

```
res <- proc_transpose(score, var = c("Test1", "Test2", "Final"),
                      name = "Exam", id = "Student", by = "Section")

proc_print(res, titles = "Student Test Scores in Variables")
```

Student Test Scores in Variables

Section	Exam	Capalleti	Engles	Lundsford	McBane	Dubose	Grant	Krupski
1	Test1	94	95	92	75			
1	Test2	91	97	40	78			
1	Final	87	97	86	72			
2	Test1					51	63	80
2	Test2					65	75	76
2	Final					91	80	71

Comparison	**Results**
Perform basic transpose with by group.	Pass

proc_transpose-004

SAS Code and Output

```
proc means data = score nonobs;
output out = stats;
run;

data score2 (keep = Group _STAT_ Test1 Test2 Final) ;
set stats (drop =_TYPE_ _FREQ_);
Group = "Group1";
run;

proc print data=score2 noobs;
   title 'Student Test Scores in Variables';

run;
```

Student Test Scores in Variables

STAT	Test1	Test2	Final	Group
N	7.0000	7.0000	7.0000	Group1
MIN	51.0000	40.0000	71.0000	Group1
MAX	95.0000	97.0000	97.0000	Group1
MEAN	78.5714	74.5714	83.4286	Group1
STD	16.8608	18.5908	9.6412	Group1

R Code and Output

```
stats <- proc_means(score, options = v(notype, nonobs))


res1 <- data.frame(Group = "Group1", stats)

res2 <- proc_transpose(res1, copy = "Group",
                       name = "STAT", id = "VAR")
proc_print(res2, titles = "Student Test Scores in Variables")
```

Student Test Scores in Variables

Group	STAT	Test1	Test2	Final
Group1	N	7	7	7
Group1	MEAN	78.5714285714286	74.5714285714286	83.4285714285714
Group1	STD	16.8607744272235	18.5908323541214	9.64118151226192
Group1	MIN	51	40	71
Group1	MAX	95	97	97

Comparison	**Results**
Perform basic transpose with copy parameter.	Pass. Differences that SAS proc means puts stats in rows anyway, as doesn't need to be transformed. Also, data step does not reorder columns. Procs package tries to improve on both of these issues.

T-Test Data

SAS	R
data cls; set sashelp.class; if _n_ < 10 then do; region = 'A'; end; else do; region = 'B'; end; run;	cls <- read.table(header = TRUE, text = ' Name Sex Age Height Weight region Alfred M 14 69.0 112.5 A Alice F 13 56.5 84.0 A Barbara F 13 65.3 98.0 A Carol F 14 62.8 102.5 A Henry M 14 63.5 102.5 A James M 12 57.3 83.0 A Jane F 12 59.8 84.5 A Janet F 15 62.5 112.5 A Jeffrey M 13 62.5 84.0 A John M 12 59.0 99.5 B Joyce F 11 51.3 50.5 B Judy F 14 64.3 90.0 B Louise F 12 56.3 77.0 B Mary F 15 66.5 112.0 B Philip M 16 72.0 150.0 B Robert M 12 64.8 128.0 B Ronald M 15 67.0 133.0 B Thomas M 11 57.5 85.0 B William M 15 66.5 112.0 B')

proc_ttest-001

SAS Code and Output

```
proc ttest data=cls
      h0=65
      alpha=0.05;
      var Height;
run;
```

The TTEST Procedure

Variable: Height

N	Mean	Std Dev	Std Err	Minimum	Maximum
19	62.3368	5.1271	1.1762	51.3000	72.0000

Mean	95% CL Mean		Std Dev	95% CL Std Dev	
62.3368	59.8657	64.8080	5.1271	3.8741	7.5820

DF	t Value	Pr > |t|
18	-2.26	0.0362

R Code and Output

```
proc_ttest(cls,
      var = c("Height"),
      options = c("h0" = 65, "alpha" = 0.05))
```

The TTEST Function

Variable: Height

N	Mean	Std Dev	Std Err	Minimum	Maximum
19	62.3368	5.1271	1.1762	51.3000	72.0000

Mean	Lower 95% CL for Mean	Upper 95% CL for Mean	Std Dev	Lower 95% CL for Std Dev	Upper 95% CL for Std Dev
62.3368	59.8657	64.8080	5.1271	3.8741	7.5820

DF	t Value	Pr > |t|
18.000	-2.26	0.0362

Comparison	**Results**
Perform basic T-Test on single variable.	Pass.

proc_ttest-002

SAS Code and Output

```
proc ttest data=cls alpha=0.05;
  paired weight * height;
run;
```

The TTEST Procedure

Difference: Weight - Height

N	Mean	Std Dev	Std Err	Minimum	Maximum
19	37.6895	18.4378	4.2299	-0.8000	78.0000

Mean	95% CL Mean		Std Dev	95% CL Std Dev	
37.6895	28.8027	46.5762	18.4378	13.9318	27.2663

DF	t Value	Pr > |t|
18	8.91	<.0001

R Code and Output

```
proc_ttest(cls,
        paired = "Weight * Height",
        options = c("alpha" = 0.05))
```

The TTEST Function
Difference: Weight-Height

N	Mean	Std Dev	Std Err	Minimum	Maximum
19	37.6895	18.4378	4.2299	-0.8000	78.0000

Mean	Lower 95% CL for Mean	Upper 95% CL for Mean	Std Dev	Lower 95% CL for Std Dev	Upper 95% CL for Std Dev
37.6895	28.8027	46.5762	18.4378	13.9318	27.2663

DF	t Value	Pr > |t|
18.000	8.91	0.0000

Comparison	**Results**
Perform basic T-Test on paired variables.	Pass.

proc_ttest-003

SAS Code and Output

```
proc ttest data=cls alpha=0.05;
  class sex;
  var height;
run;
```

The TTEST Procedure

Variable: Height

Sex	Method	N	Mean	Std Dev	Std Err	Minimum	Maximum
F		9	60.5889	5.0183	1.6728	51.3000	66.5000
M		10	63.9100	4.9379	1.5615	57.3000	72.0000
Diff (1-2)	Pooled		-3.3211	4.9759	2.2863		
Diff (1-2)	Satterthwaite		-3.3211		2.2883		

Sex	Method	Mean	95% CL Mean		Std Dev	95% CL Std Dev	
F		60.5889	56.7315	64.4463	5.0183	3.3897	9.6140
M		63.9100	60.3776	67.4424	4.9379	3.3965	9.0147
Diff (1-2)	Pooled	-3.3211	-8.1447	1.5025	4.9759	3.7339	7.4596
Diff (1-2)	Satterthwaite	-3.3211	-8.1551	1.5129			

Method	Variances	DF	t Value	Pr > |t|
Pooled	Equal	17	-1.45	0.1645
Satterthwaite	Unequal	16.727	-1.45	0.1652

Equality of Variances

Method	Num DF	Den DF	F Value	Pr > F
Folded F	8	9	1.03	0.9527

R Code and Output

```
proc_ttest(cls,
        var = "Height",
        class = "Sex",
        options = c("alpha" = 0.05))
```

The TTEST Function

Variable: Height

Sex	Method	N	Mean	Std Dev	Std Err	Minimum	Maximum
F		9	60.5889	5.0183	1.6728	51.3000	66.5000
M		10	63.9100	4.9379	1.5615	57.3000	72.0000
Diff (1-2)	Pooled		-3.3211		2.2863		
Diff (1-2)	Satterthwaite		-3.3211		2.2883		

Sex	Method	Mean	Lower 95% CL for Mean	Upper 95% CL for Mean	Std Dev	Lower 95% CL for Std Dev	Upper 95% CL for Std Dev
F		60.5889	56.7315	64.4463	5.0183	3.3897	9.6140
M		63.9100	60.3776	67.4424	4.9379	3.3965	9.0147
Diff (1-2)	Pooled	-3.3211	-8.1447	1.5025			
Diff (1-2)	Satterthwaite	-3.3211	-8.1551	1.5129			

Method	Variances	DF	t Value	Pr > |t|
Pooled	Equal	17.000	-1.45	0.1645
Satterthwaite	Unequal	16.727	-1.45	0.1652

Method	Num DF	Den DF	F Value	Pr > F
Folded F	8.000	9.000	1.03	0.9527

Comparison	**Results**
Perform basic T-Test on variable with class.	Pass, except std and std CL not available for Pooled CL.

proc_ttest-004

SAS Code and Output

```
proc ttest data=cls
        h0=65
        alpha=0.1;
        var Height;
        by region;
run;
```

The TTEST Procedure

Variable: Height

region=A

N	Mean	Std Dev	Std Err	Minimum	Maximum
9	62.1333	3.8778	1.2926	56.5000	69.0000

Mean	90% CL Mean		Std Dev	90% CL Std Dev	
62.1333	59.7297	64.5370	3.8778	2.7853	6.6350

DF	t Value	Pr > |t|
8	-2.22	0.0574

The TTEST Procedure

Variable: Height

region=B

N	Mean	Std Dev	Std Err	Minimum	Maximum
10	62.5200	6.2553	1.9781	51.3000	72.0000

Mean	90% CL Mean		Std Dev	90% CL Std Dev	
62.5200	58.8939	66.1461	6.2553	4.5623	10.2912

DF	t Value	Pr > |t|
9	-1.25	0.2415

R Code and Output

```
proc_ttest(cls,
        var = c("Height"),
        by = "region",
        options = c("h0" = 65, "alpha" = 0.05))
```

The TTEST Function
Variable: Height
region=A

N	Mean	Std Dev	Std Err	Minimum	Maximum
9	62.1333	3.8778	1.2926	56.5000	69.0000

Mean	Lower 90% CL for Mean	Upper 90% CL for Mean	Std Dev	Lower 90% CL for Std Dev	Upper 90% CL for Std Dev
62.1333	59.7297	64.5370	3.8778	2.7853	6.6350

DF	t Value	Pr > |t|
8.000	-2.22	0.0574

The TTEST Function
Variable: Height
region=B

N	Mean	Std Dev	Std Err	Minimum	Maximum
10	62.5200	6.2553	1.9781	51.3000	72.0000

Mean	Lower 90% CL for Mean	Upper 90% CL for Mean	Std Dev	Lower 90% CL for Std Dev	Upper 90% CL for Std Dev
62.5200	58.8939	66.1461	6.2553	4.5623	10.2912

DF	t Value	Pr > |t|
9.000	-1.25	0.2415

Comparison	**Results**
Perform T-Test on single variable with by.	Pass.

proc_ttest-005

SAS Code and Output

```
proc ttest data=cls alpha=0.1;
  paired weight * height;
  by region;
run;
```

The TTEST Procedure

Difference: Weight - Height

region=A

N	Mean	Std Dev	Std Err	Minimum	Maximum
9	33.8111	9.7403	3.2468	21.5000	50.0000

Mean	90% CL Mean		Std Dev	90% CL Std Dev	
33.8111	27.7736	39.8486	9.7403	6.9960	16.6658

DF	t Value	Pr > |t|
8	10.41	<.0001

The TTEST Procedure

Difference: Weight - Height

region=B

N	Mean	Std Dev	Std Err	Minimum	Maximum
10	41.1800	23.8116	7.5299	-0.8000	78.0000

Mean	90% CL Mean		Std Dev	90% CL Std Dev	
41.1800	27.3769	54.9831	23.8116	17.3669	39.1748

DF	t Value	Pr > |t|
9	5.47	0.0004

R Code and Output

```
proc_ttest(cls,
        paired = "Weight * Height",
        by = "region",
        options = c("alpha" = 0.1))
```

The TTEST Function
Difference: Weight-Height
region=A

N	Mean	Std Dev	Std Err	Minimum	Maximum
9	33.8111	9.7403	3.2468	21.5000	50.0000

Mean	Lower 90% CL for Mean	Upper 90% CL for Mean	Std Dev	Lower 90% CL for Std Dev	Upper 90% CL for Std Dev
33.8111	27.7736	39.8486	9.7403	6.9960	16.6658

DF	t Value	Pr > |t|
8.000	10.41	0.0000

The TTEST Function
Difference: Weight-Height
region=B

N	Mean	Std Dev	Std Err	Minimum	Maximum
10	41.1800	23.8116	7.5299	-0.8000	78.0000

Mean	Lower 90% CL for Mean	Upper 90% CL for Mean	Std Dev	Lower 90% CL for Std Dev	Upper 90% CL for Std Dev
41.1800	27.3769	54.9831	23.8116	17.3669	39.1748

DF	t Value	Pr > |t|
9.000	5.47	0.0004

Comparison	**Results**
Perform T-Test on paired variables with by.	Pass.

proc_ttest-006

SAS Code and Output

```
proc ttest data=cls alpha=0.1;
  class sex;
  var height;
  by region;
run;
```

The TTEST Procedure

Variable: Height

region=A

Sex	Method	N	Mean	Std Dev	Std Err	Minimum	Maximum
F		5	61.3800	3.3522	1.4991	56.5000	65.3000
M		4	63.0750	4.7947	2.3974	57.3000	69.0000
Diff (1-2)	Pooled		-1.6950	4.0341	2.7061		
Diff (1-2)	Satterthwaite		-1.6950		2.8275		

Sex	Method	Mean	90% CL Mean		Std Dev	90% CL Std Dev	
F		61.3800	58.1841	64.5759	3.3522	2.1766	7.9525
M		63.0750	57.4332	68.7168	4.7947	2.9707	14.0006
Diff (1-2)	Pooled	-1.6950	-6.8220	3.4320	4.0341	2.8457	7.2498
Diff (1-2)	Satterthwaite	-1.6950	-7.3427	3.9527			

Method	Variances	DF	t Value	Pr > |t|
Pooled	Equal	7	-0.63	0.5510
Satterthwaite	Unequal	5.2077	-0.60	0.5740

Equality of Variances

Method	Num DF	Den DF	F Value	Pr > F
Folded F	3	4	2.05	0.5002

The TTEST Procedure

Variable: Height

region=B

Sex	Method	N	Mean	Std Dev	Std Err	Minimum	Maximum
F		4	59.6000	7.0588	3.5294	51.3000	66.5000
M		6	64.4667	5.4006	2.2048	57.5000	72.0000
Diff (1-2)	Pooled		-4.8667	6.0757	3.9218		
Diff (1-2)	Satterthwaite		-4.8667		4.1615		

Sex	Method	Mean	90% CL Mean		Std Dev	90% CL Std Dev	
F		59.6000	51.2940	67.9060	7.0588	4.3736	20.6118
M		64.4667	60.0239	68.9094	5.4006	3.6295	11.2833
Diff (1-2)	Pooled	-4.8667	-12.1595	2.4262	6.0757	4.3639	10.3956
Diff (1-2)	Satterthwaite	-4.8667	-13.1444	3.4110			

Method	Variances	DF	t Value	Pr > |t|
Pooled	Equal	8	-1.24	0.2498
Satterthwaite	Unequal	5.3129	-1.17	0.2920

Equality of Variances

Method	Num DF	Den DF	F Value	Pr > F
Folded F	3	5	1.71	0.5600

R Code and Output

```
proc_ttest(cls,
        var = "Height",
        class = "Sex",
        by = "region",
        options = c("alpha" = 0.1))
```

The TTEST Function
Variable: Height
region=A

Sex	Method	N	Mean	Std Dev	Std Err	Minimum	Maximum
F		5	61.3800	3.3522	1.4991	56.5000	65.3000
M		4	63.0750	4.7947	2.3974	57.3000	69.0000
Diff (1-2)	Pooled		-1.6950		2.7061		
Diff (1-2)	Satterthwaite		-1.6950		2.8275		

Sex	Method	Mean	Lower 90% CL for Mean	Upper 90% CL for Mean	Std Dev	Lower 90% CL for Std Dev	Upper 90% CL for Std Dev
F		61.3800	58.1841	64.5759	3.3522	2.1766	7.9525
M		63.0750	57.4332	68.7168	4.7947	2.9707	14.0006
Diff (1-2)	Pooled	-1.6950	-6.8220	3.4320			
Diff (1-2)	Satterthwaite	-1.6950	-7.3427	3.9527			

Method	Variances	DF	t Value	Pr > |t|
Pooled	Equal	7.000	-0.63	0.5510
Satterthwaite	Unequal	5.208	-0.60	0.5740

Method	Num DF	Den DF	F Value	Pr > F
Folded F	3.000	4.000	2.05	0.5002

The TTEST Function
Variable: Height
region=B

Sex	Method	N	Mean	Std Dev	Std Err	Minimum	Maximum
F		4	59.6000	7.0588	3.5294	51.3000	66.5000
M		6	64.4667	5.4006	2.2048	57.5000	72.0000
Diff (1-2)	Pooled		-4.8667		3.9218		
Diff (1-2)	Satterthwaite		-4.8667		4.1615		

Sex	Method	Mean	Lower 90% CL for Mean	Upper 90% CL for Mean	Std Dev	Lower 90% CL for Std Dev	Upper 90% CL for Std Dev
F		59.6000	51.2940	67.9060	7.0588	4.3736	20.6118
M		64.4667	60.0239	68.9094	5.4006	3.6295	11.2833
Diff (1-2)	Pooled	-4.8667	-12.1595	2.4262			
Diff (1-2)	Satterthwaite	-4.8667	-13.1444	3.4110			

Method	Variances	DF	t Value	Pr > |t|
Pooled	Equal	8.000	-1.24	0.2498
Satterthwaite	Unequal	5.313	-1.17	0.2920

Method	Num DF	Den DF	F Value	Pr > F
Folded F	3.000	5.000	1.71	0.5600

Comparison	**Results**
Perform T-Test on unpaired variable with by.	Pass. Initially F Value and CL values were not coming out.

proc_ttest-007

SAS Code and Output

```
proc ttest data=cls alpha=0.1;
  class sex;
  var Weight;
  by region;
run;
```

The TTEST Procedure

Variable: Weight

region=A

Sex	Method	N	Mean	Std Dev	Std Err	Minimum	Maximum
F		5	96.3000	12.1891	5.4511	84.0000	112.5
M		4	95.5000	14.4511	7.2255	83.0000	112.5
Diff (1-2)	Pooled		0.8000	13.2061	8.8589		
Diff (1-2)	Satterthwaite		0.8000		9.0512		

Sex	Method	Mean	90% CL Mean		Std Dev	90% CL Std Dev	
F		96.3000	84.6790	107.9	12.1891	7.9145	28.9170
M		95.5000	78.4957	112.5	14.4511	8.9537	42.1972
Diff (1-2)	Pooled	0.8000	-15.9839	17.5839	13.2061	9.3158	23.7333
Diff (1-2)	Satterthwaite	0.8000	-16.8183	18.4183			

Method	Variances	DF	t Value	Pr > |t|
Pooled	Equal	7	0.09	0.9306
Satterthwaite	Unequal	5.9429	0.09	0.9325

Equality of Variances

Method	Num DF	Den DF	F Value	Pr > F
Folded F	3	4	1.41	0.7278

The TTEST Procedure

Variable: Weight

region=B

Sex	Method	N	Mean	Std Dev	Std Err	Minimum	Maximum
F		4	82.3750	25.6949	12.8475	50.5000	112.0
M		6	117.9	23.7327	9.6888	85.0000	150.0
Diff (1-2)	Pooled		-35.5417	24.4870	15.8063		
Diff (1-2)	Satterthwaite		-35.5417		16.0913		

Sex	Method	Mean	90% CL Mean		Std Dev	90% CL Std Dev	
F		82.3750	52.1403	112.6	25.6949	15.9203	75.0294
M		117.9	98.3932	137.4	23.7327	15.9496	49.5837
Diff (1-2)	Pooled	-35.5417	-64.9342	-6.1491	24.4870	17.5878	41.8976
Diff (1-2)	Satterthwaite	-35.5417	-66.6452	-4.4381			

Method	Variances	DF	t Value	Pr > |t|
Pooled	Equal	8	-2.25	0.0547
Satterthwaite	Unequal	6.1828	-2.21	0.0679

Equality of Variances

Method	Num DF	Den DF	F Value	Pr > F
Folded F	3	5	1.17	0.8154

R Code and Output

```
proc_ttest(cls,
        var = "Weight",
        class = "Sex",
        by = "region",
        options = c("alpha" = 0.1))
```

The TTEST Function
Variable: Weight
region=A

Sex	Method	N	Mean	Std Dev	Std Err	Minimum	Maximum
F		5	96.3000	12.1891	5.4511	84.0000	112.5000
M		4	95.5000	14.4511	7.2255	83.0000	112.5000
Diff (1-2)	Pooled		0.8000		8.8589		
Diff (1-2)	Satterthwaite		0.8000		9.0512		

Sex	Method	Mean	Lower 90% CL for Mean	Upper 90% CL for Mean	Std Dev	Lower 90% CL for Std Dev	Upper 90% CL for Std Dev
F		96.3000	84.6790	107.9210	12.1891	7.9145	28.9170
M		95.5000	78.4957	112.5043	14.4511	8.9537	42.1972
Diff (1-2)	Pooled	0.8000	-15.9839	17.5839			
Diff (1-2)	Satterthwaite	0.8000	-16.8183	18.4183			

Method	Variances	DF	t Value	Pr > |t|
Pooled	Equal	7.000	0.09	0.9306
Satterthwaite	Unequal	5.943	0.09	0.9325

Method	Num DF	Den DF	F Value	Pr > F
Folded F	3.000	4.000	1.41	0.7278

The TTEST Function
Variable: Weight
region=B

Sex	Method	N	Mean	Std Dev	Std Err	Minimum	Maximum
F		4	82.3750	25.6949	12.8475	50.5000	112.0000
M		6	117.9167	23.7327	9.6888	85.0000	150.0000
Diff (1-2)	Pooled		-35.5417		15.8063		
Diff (1-2)	Satterthwaite		-35.5417		16.0913		

Sex	Method	Mean	Lower 90% CL for Mean	Upper 90% CL for Mean	Std Dev	Lower 90% CL for Std Dev	Upper 90% CL for Std Dev
F		82.3750	52.1403	112.6097	25.6949	15.9203	75.0294
M		117.9167	98.3932	137.4401	23.7327	15.9496	49.5837
Diff (1-2)	Pooled	-35.5417	-64.9342	-6.1491			
Diff (1-2)	Satterthwaite	-35.5417	-66.6452	-4.4381			

Method	Variances	DF	t Value	Pr > |t|
Pooled	Equal	8.000	-2.25	0.0547
Satterthwaite	Unequal	6.183	-2.21	0.0679

Method	Num DF	Den DF	F Value	Pr > F
Folded F	3.000	5.000	1.17	0.8154

Comparison	**Results**
Perform T-Test on unpaired variable with by.	Pass. More testing to ensure it is coming out with different variables.

proc_ttest-008

SAS Code and Output

```
proc ttest data=cls alpha=0.1;
  class sex;
  var Age;
  by region;
run;
```

The TTEST Procedure

Variable: Age

region=A

Sex	Method	N	Mean	Std Dev	Std Err	Minimum	Maximum
F		5	13.4000	1.1402	0.5099	12.0000	15.0000
M		4	13.2500	0.9574	0.4787	12.0000	14.0000
Diff (1-2)	Pooled		0.1500	1.0657	0.7149		
Diff (1-2)	Satterthwaite		0.1500		0.6994		

Sex	Method	Mean	90% CL Mean		Std Dev	90% CL Std Dev	
F		13.4000	12.3130	14.4870	1.1402	0.7403	2.7049
M		13.2500	12.1234	14.3766	0.9574	0.5932	2.7957
Diff (1-2)	Pooled	0.1500	-1.2044	1.5044	1.0657	0.7518	1.9152
Diff (1-2)	Satterthwaite	0.1500	-1.1764	1.4764			

Method	Variances	DF	t Value	Pr > |t|
Pooled	Equal	7	0.21	0.8398
Satterthwaite	Unequal	6.9548	0.21	0.8363

Equality of Variances

Method	Num DF	Den DF	F Value	Pr > F
Folded F	4	3	1.42	0.8061

The TTEST Procedure

Variable: Age

region=B

Sex	Method	N	Mean	Std Dev	Std Err	Minimum	Maximum
F		4	13.0000	1.8257	0.9129	11.0000	15.0000
M		6	13.5000	2.0736	0.8466	11.0000	16.0000
Diff (1-2)	Pooled		-0.5000	1.9843	1.2809		
Diff (1-2)	Satterthwaite		-0.5000		1.2450		

Sex	Method	Mean	90% CL Mean		Std Dev	90% CL Std Dev	
F		13.0000	10.8517	15.1483	1.8257	1.1312	5.3312
M		13.5000	11.7941	15.2059	2.0736	1.3936	4.3324
Diff (1-2)	Pooled	-0.5000	-2.8818	1.8818	1.9843	1.4252	3.3952
Diff (1-2)	Satterthwaite	-0.5000	-2.8494	1.8494			

Method	Variances	DF	t Value	Pr > |t|
Pooled	Equal	8	-0.39	0.7065
Satterthwaite	Unequal	7.1887	-0.40	0.6996

Equality of Variances

Method	Num DF	Den DF	F Value	Pr > F
Folded F	5	3	1.29	0.8882

R Code and Output

```
proc_ttest(cls,
        var = "Age",
        class = "Sex",
        by = "region",
        options = c("alpha" = 0.1))
```

The TTEST Function
Variable: Age
region=A

Sex	Method	N	Mean	Std Dev	Std Err	Minimum	Maximum
F		5	13.4000	1.1402	0.5099	12.0000	15.0000
M		4	13.2500	0.9574	0.4787	12.0000	14.0000
Diff (1-2)	Pooled		0.1500		0.7149		
Diff (1-2)	Satterthwaite		0.1500		0.6994		

Sex	Method	Mean	Lower 90% CL for Mean	Upper 90% CL for Mean	Std Dev	Lower 90% CL for Std Dev	Upper 90% CL for Std Dev
F		13.4000	12.3130	14.4870	1.1402	0.7403	2.7049
M		13.2500	12.1234	14.3766	0.9574	0.5932	2.7957
Diff (1-2)	Pooled	0.1500	-1.2044	1.5044			
Diff (1-2)	Satterthwaite	0.1500	-1.1764	1.4764			

Method	Variances	DF	t Value	Pr > |t|
Pooled	Equal	7.000	0.21	0.8398
Satterthwaite	Unequal	6.955	0.21	0.8363

Method	Num DF	Den DF	F Value	Pr > F
Folded F	4.000	3.000	1.42	0.8061

The TTEST Function
Variable: Age
region=B

Sex	Method	N	Mean	Std Dev	Std Err	Minimum	Maximum
F		4	13.0000	1.8257	0.9129	11.0000	15.0000
M		6	13.5000	2.0736	0.8466	11.0000	16.0000
Diff (1-2)	Pooled		-0.5000		1.2809		
Diff (1-2)	Satterthwaite		-0.5000		1.2450		

Sex	Method	Mean	Lower 90% CL for Mean	Upper 90% CL for Mean	Std Dev	Lower 90% CL for Std Dev	Upper 90% CL for Std Dev
F		13.0000	10.8517	15.1483	1.8257	1.1312	5.3312
M		13.5000	11.7941	15.2059	2.0736	1.3936	4.3324
Diff (1-2)	Pooled	-0.5000	-2.8818	1.8818			
Diff (1-2)	Satterthwaite	-0.5000	-2.8494	1.8494			

Method	Variances	DF	t Value	Pr > |t|
Pooled	Equal	8.000	-0.39	0.7065
Satterthwaite	Unequal	7.189	-0.40	0.6996

Method	Num DF	Den DF	F Value	Pr > F
Folded F	5.000	3.000	1.29	0.8882

Comparison	**Results**
Perform T-Test on unpaired variable with by.	Pass. Another test with a different variable.

proc_ttest-009A

SAS Code and Output

```
proc ttest data=cls
alpha=0.1;
  class sex;
  var Height Weight
Age;
  by region;
run;
```

The TTEST Procedure

Variable: Height

region=A

Sex	Method	N	Mean	Std Dev	Std Err	Minimum	Maximum
F		5	61.3800	3.3522	1.4991	56.5000	65.3000
M		4	63.0750	4.7947	2.3974	57.3000	69.0000
Diff (1-2)	Pooled		-1.6950	4.0341	2.7061		
Diff (1-2)	Satterthwaite		-1.6950		2.8275		

Sex	Method	Mean	90% CL Mean		Std Dev	90% CL Std Dev	
F		61.3800	58.1841	64.5759	3.3522	2.1766	7.9525
M		63.0750	57.4332	68.7168	4.7947	2.9707	14.0006
Diff (1-2)	Pooled	-1.6950	-6.8220	3.4320	4.0341	2.8457	7.2498
Diff (1-2)	Satterthwaite	-1.6950	-7.3427	3.9527			

Method	Variances	DF	t Value	Pr > |t|
Pooled	Equal	7	-0.63	0.5510
Satterthwaite	Unequal	5.2077	-0.60	0.5740

Equality of Variances

Method	Num DF	Den DF	F Value	Pr > F
Folded F	3	4	2.05	0.5002

Variable: Weight

Sex	Method	N	Mean	Std Dev	Std Err	Minimum	Maximum
F		5	96.3000	12.1891	5.4511	84.0000	112.5
M		4	95.5000	14.4511	7.2255	83.0000	112.5
Diff (1-2)	Pooled		0.8000	13.2061	8.8589		
Diff (1-2)	Satterthwaite		0.8000		9.0512		

Sex	Method	Mean	90% CL Mean		Std Dev	90% CL Std Dev	
F		96.3000	84.6790	107.9	12.1891	7.9145	28.9170
M		95.5000	78.4957	112.5	14.4511	8.9537	42.1972
Diff (1-2)	Pooled	0.8000	-15.9839	17.5839	13.2061	9.3158	23.7333
Diff (1-2)	Satterthwaite	0.8000	-16.8183	18.4183			

Method	Variances	DF	t Value	Pr > |t|
Pooled	Equal	7	0.09	0.9306
Satterthwaite	Unequal	5.9429	0.09	0.9325

Equality of Variances

Method	Num DF	Den DF	F Value	Pr > F
Folded F	3	4	1.41	0.7278

Variable: Age

Sex	Method	N	Mean	Std Dev	Std Err	Minimum	Maximum
F		5	13.4000	1.1402	0.5099	12.0000	15.0000
M		4	13.2500	0.9574	0.4787	12.0000	14.0000
Diff (1-2)	Pooled		0.1500	1.0657	0.7149		
Diff (1-2)	Satterthwaite		0.1500		0.6994		

Sex	Method	Mean	90% CL Mean		Std Dev	90% CL Std Dev	
F		13.4000	12.3130	14.4870	1.1402	0.7403	2.7049
M		13.2500	12.1234	14.3766	0.9574	0.5932	2.7957
Diff (1-2)	Pooled	0.1500	-1.2044	1.5044	1.0657	0.7518	1.9152
Diff (1-2)	Satterthwaite	0.1500	-1.1764	1.4764			

Method	Variances	DF	t Value	Pr > |t|
Pooled	Equal	7	0.21	0.8398
Satterthwaite	Unequal	6.9548	0.21	0.8363

Equality of Variances

Method	Num DF	Den DF	F Value	Pr > F
Folded F	4	3	1.42	0.8061

R Code and Output

```
proc_ttest(cls,
        var = c("Height",
"Weight", "Age"),
        class = "Sex",
        by = "region",
        options = c("alpha" =
0.1))
```

The TTEST Function
Variable: Height
region=A

Sex	Method	N	Mean	Std Dev	Std Err	Minimum	Maximum
F		5	61.3800	3.3522	1.4991	56.5000	65.3000
M		4	63.0750	4.7947	2.3974	57.3000	69.0000
Diff (1-2)	Pooled		-1.6950		2.7061		
Diff (1-2)	Satterthwaite		-1.6950		2.8275		

Sex	Method	Mean	Lower 90% CL for Mean	Upper 90% CL for Mean	Std Dev	Lower 90% CL for Std Dev	Upper 90% CL for Std Dev
F		61.3800	58.1841	64.5759	3.3522	2.1766	7.9525
M		63.0750	57.4332	68.7168	4.7947	2.9707	14.0006
Diff (1-2)	Pooled	-1.6950	-6.8220	3.4320			
Diff (1-2)	Satterthwaite	-1.6950	-7.3427	3.9527			

Method	Variances	DF	t Value	Pr > |t|
Pooled	Equal	7.000	-0.63	0.5510
Satterthwaite	Unequal	5.208	-0.60	0.5740

Method	Num DF	Den DF	F Value	Pr > F
Folded F	3.000	4.000	2.05	0.5002

Variable: Weight
region=A

Sex	Method	N	Mean	Std Dev	Std Err	Minimum	Maximum
F		5	96.3000	12.1891	5.4511	84.0000	112.5000
M		4	95.5000	14.4511	7.2255	83.0000	112.5000
Diff (1-2)	Pooled		0.8000		8.8589		
Diff (1-2)	Satterthwaite		0.8000		9.0512		

Sex	Method	Mean	Lower 90% CL for Mean	Upper 90% CL for Mean	Std Dev	Lower 90% CL for Std Dev	Upper 90% CL for Std Dev
F		96.3000	84.6790	107.9210	12.1891	7.9145	28.9170
M		95.5000	78.4957	112.5043	14.4511	8.9537	42.1972
Diff (1-2)	Pooled	0.8000	-15.9839	17.5839			
Diff (1-2)	Satterthwaite	0.8000	-16.8183	18.4183			

Method	Variances	DF	t Value	Pr > |t|
Pooled	Equal	7.000	0.09	0.9306
Satterthwaite	Unequal	5.943	0.09	0.9325

Method	Num DF	Den DF	F Value	Pr > F
Folded F	3.000	4.000	1.41	0.7278

Variable: Age
region=A

Sex	Method	N	Mean	Std Dev	Std Err	Minimum	Maximum
F		5	13.4000	1.1402	0.5099	12.0000	15.0000
M		4	13.2500	0.9574	0.4787	12.0000	14.0000
Diff (1-2)	Pooled		0.1500		0.7149		
Diff (1-2)	Satterthwaite		0.1500		0.6994		

Sex	Method	Mean	Lower 90% CL for Mean	Upper 90% CL for Mean	Std Dev	Lower 90% CL for Std Dev	Upper 90% CL for Std Dev
F		13.4000	12.3130	14.4870	1.1402	0.7403	2.7049
M		13.2500	12.1234	14.3766	0.9574	0.5932	2.7957
Diff (1-2)	Pooled	0.1500	-1.2044	1.5044			
Diff (1-2)	Satterthwaite	0.1500	-1.1764	1.4764			

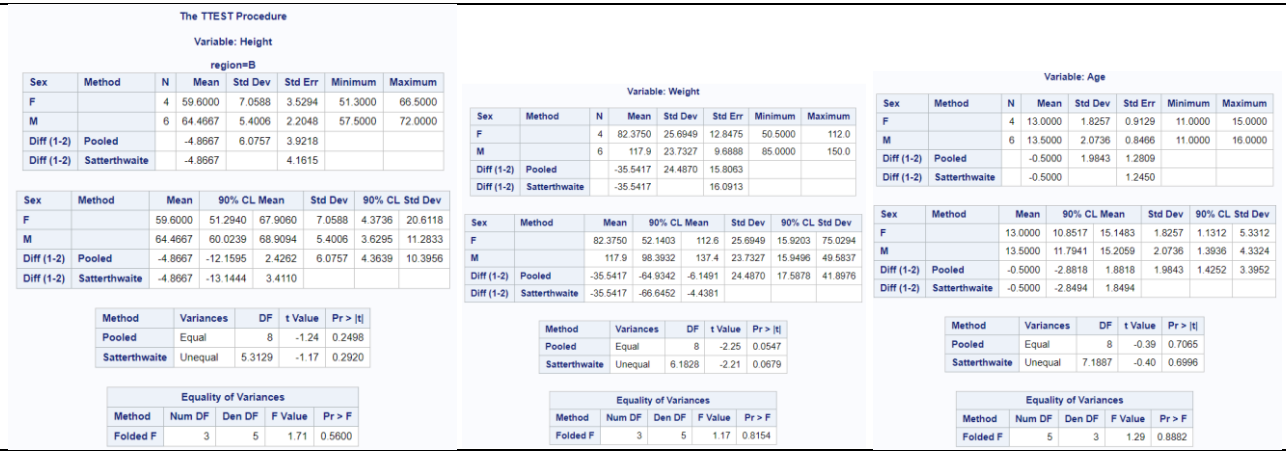
Method	Variances	DF	t Value	Pr > |t|
Pooled	Equal	7.000	0.21	0.8398
Satterthwaite	Unequal	6.955	0.21	0.8363

Method	Num DF	Den DF	F Value	Pr > F
Folded F	4.000	3.000	1.42	0.8061

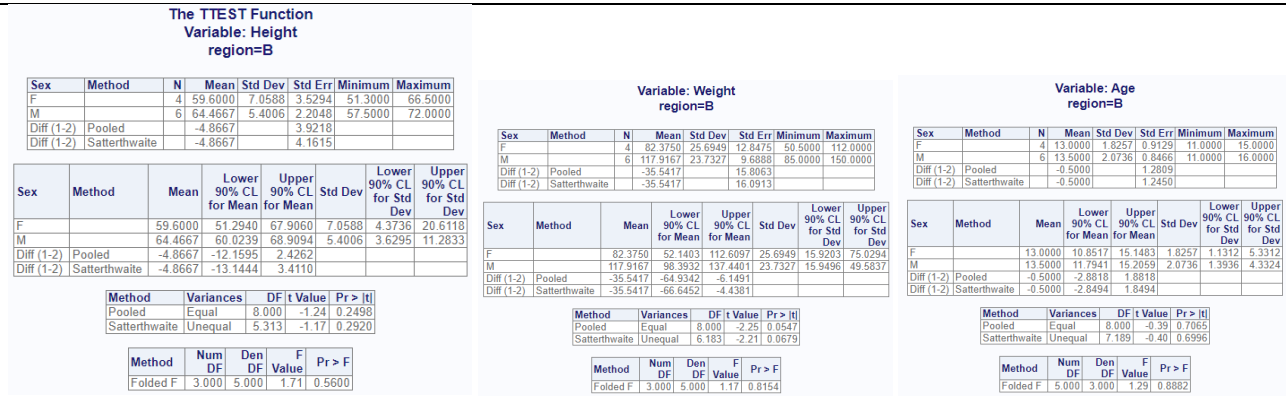
Comparison	**Results**
Perform T-Test on multiple unpaired variable with by.	Pass.

proc_ttest-009B

SAS Code and Output	

proc ttest data=cls alpha=0.1;
  class sex;
  var Height Weight Age;
  by region;
run;
	

R Code and Output	

proc_ttest(cls,
  var = c("Height", "Weight", "Age")
  class = "Sex",
  by = "region",
  options = c("alpha" = 0.1))
	

Comparison	**Results**
Perform T-Test on multiple unpaired variable with by.	Pass.

Reg Data

SAS

```
datalines;
44 89.47 44.609 11.37   40 75.07 45.313 10.07
44 85.84 54.297  8.65   42 68.15 59.571  8.17
38 89.02 49.874  9.22   47 77.45 44.811 11.63
40 75.98 45.681 11.95   43 81.19 49.091 10.85
44 81.42 39.442 13.08   38 81.87 60.055  8.63
44 73.03 50.541 10.13   45 87.66 37.388 14.03
45 66.45 44.754 11.12   47 79.15 47.273 10.60
54 83.12 51.855 10.33   49 81.42 49.156  8.95
51 69.63 40.836 10.95   51 77.91 46.672 10.00
48 91.63 46.774 10.25   49 73.37 50.388 10.08
57 73.37 39.407 12.63   54 79.38 46.080 11.17
52 76.32 45.441  9.63   50 70.87 54.625  8.92
51 67.25 45.118 11.08   54 91.63 39.203 12.88
51 73.71 45.790 10.47   57 59.08 50.545  9.93
49 76.32 48.673  9.40   48 61.24 47.920 11.50
52 82.78 47.467 10.50
;
```

R

```
fitness <- read.table(header = TRUE, text = '
Age Weight Oxygen RunTime
44 89.47 44.609 11.37
40 75.07 45.313 10.07
44 85.84 54.297  8.65
42 68.15 59.571  8.17
38 89.02 49.874  9.22
47 77.45 44.811 11.63
40 75.98 45.681 11.95
43 81.19 49.091 10.85
44 81.42 39.442 13.08
38 81.87 60.055  8.63
44 73.03 50.541 10.13
45 87.66 37.388 14.03
45 66.45 44.754 11.12
47 79.15 47.273 10.60
54 83.12 51.855 10.33
49 81.42 49.156  8.95
51 69.63 40.836 10.95
51 77.91 46.672 10.00
48 91.63 46.774 10.25
49 73.37 50.388 10.08
57 73.37 39.407 12.63
54 79.38 46.080 11.17
52 76.32 45.441  9.63
50 70.87 54.625  8.92
51 67.25 45.118 11.08
54 91.63 39.203 12.88
51 73.71 45.790 10.47
57 59.08 50.545  9.93
49 76.32 48.673  9.40
48 61.24 47.920 11.50
52 82.78 47.467 10.50')
```

proc_reg-001

SAS Code and Output

```
proc reg data = fitness;
model Oxygen = RunTime;
run;
```

Number of Observations Read	31
Number of Observations Used	31

Analysis of Variance

Source	DF	Sum of Squares	Mean Square	F Value	Pr > F
Model	1	632.90010	632.90010	84.01	<.0001
Error	29	218.48144	7.53384		
Corrected Total	30	851.38154			

Root MSE	2.74478	R-Square	0.7434
Dependent Mean	47.37581	Adj R-Sq	0.7345
Coeff Var	5.79364		

Parameter Estimates

Variable	DF	Parameter Estimate	Standard Error	t Value	Pr > |t|
Intercept	1	82.42177	3.85530	21.38	<.0001
RunTime	1	-3.31056	0.36119	-9.17	<.0001

R Code and Output

```
proc_reg(fitness,
      model = Oxygen ~ RunTime)
```

	NOBS
Number of Observations Read	31
Number of Observations Used	31

Analysis of Variance

Source	DF	Sum of Squares	Mean Square	F Value	Pr > F
Model	1	632.90010	632.90010	84.01	<.0001
Error	29	218.48144	7.53384		
Corrected Total	30	851.38154			

Root MSE	Dependent Mean	Coeff Var	R-Square	Adj R-Sq
2.74478	47.37581	5.79364	0.7434	0.7345

Parameter Estimates

Variable	DF	Parameter Estimate	Std Err	t Value	Pr > |t|
Intercept	1	82.42177	3.85530	21.38	<.0001
RunTime	1	-3.31056	0.36119	-9.17	<.0001

Comparison	**Results**
Perform basic Reg on single variable.	Pass.

proc_reg-002

SAS Code and Output

```
proc reg data = fitness;
model Oxygen = RunTime / clb;
run;
```

Parameter Estimates

Variable	DF	Parameter Estimate	Standard Error	t Value	Pr > |t|	95% Confidence Limits	
Intercept	1	82.42177	3.85530	21.38	<.0001	74.53679	90.30675
RunTime	1	-3.31056	0.36119	-9.17	<.0001	-4.04928	-2.57183

R Code and Output

```
proc_reg(fitness,
      model = Oxygen ~ RunTime,
      stats = clb)
```

Parameter Estimates

Variable	DF	Parameter Estimate	Std Err	t Value	Pr > |t|	Lower 95% CL for Mean	Upper 95% CL for Mean
Intercept	1	82.42177	3.85530	21.38	<.0001	74.53679	90.30675
RunTime	1	-3.31056	0.36119	-9.17	<.0001	-4.04928	-2.57183

Comparison	**Results**
Perform basic Reg on single variable with clb option.	Pass.

proc_reg-003

SAS Code and Output

```
proc reg data = fitness;
model Oxygen = RunTime / hcc;
run;
```

Parameter Estimates

Variable	DF	Parameter Estimate	Standard Error	t Value	Pr > |t|	Heteroscedasticity Consistent: Standard Error	Heteroscedasticity Consistent: t Value	Heteroscedasticity Consistent: Pr > |t|
Intercept	1	82.42177	3.85530	21.38	<.0001	3.97197	20.75	<.0001
RunTime	1	-3.31056	0.36119	-9.17	<.0001	0.34976	-9.47	<.0001

R Code and Output

```
proc_reg(fitness,
      model = Oxygen ~ RunTime)
```

Parameter Estimates

Variable	DF	Parameter Estimate	Std Err	t Value	Pr > |t|	Heteroscedasticity Consistent: Std Error	Heteroscedasticity Consistent: t Value	Heteroscedasticity Consistent: Pr>|t|
Intercept	1	82.42177	3.85530	21.38	<.0001	3.97197	20.75	<.0001
RunTime	1	-3.31056	0.36119	-9.17	<.0001	0.34976	-9.47	<.0001

Comparison	**Results**
Perform basic Reg on single variable with hcc option.	Pass.

proc_reg-004

SAS Code and Output

```
proc reg data = fitness;
model Oxygen = RunTime / hcc
hccmethod=3;
run;
```

Parameter Estimates

Variable	DF	Parameter Estimate	Standard Error	t Value	Pr > |t|	Heteroscedasticity Consistent: Standard Error	Heteroscedasticity Consistent: t Value	Heteroscedasticity Consistent: Pr > |t|
Intercept	1	82.42177	3.85530	21.38	<.0001	4.43888	18.57	<.0001
RunTime	1	-3.31056	0.36119	-9.17	<.0001	0.39199	-8.45	<.0001

R Code and Output

```
proc_reg(fitness,
      model = Oxygen ~ RunTime)
```

Parameter Estimates

Variable	DF	Parameter Estimate	Std Err	t Value	Pr > |t|	Heteroscedasticity Consistent: Std Error	Heteroscedasticity Consistent: t Value	Heteroscedasticity Consistent: Pr>|t|
Intercept	1	82.42177	3.85530	21.38	<.0001	4.43888	18.57	<.0001
RunTime	1	-3.31056	0.36119	-9.17	<.0001	0.39199	-8.45	<.0001

Comparison	**Results**
Perform basic Reg on single variable with hcc type 3.	Pass.

proc_reg-005

SAS Code and Output

```
proc reg data = fitness;
model Oxygen = RunTime / p;
run;
```

Output Statistics

Obs	Dependent Variable	Predicted Value	Residual
1	44.6	44.7808	-0.1718
2	45.3	49.0845	-3.7715
3	54.3	53.7855	0.5115
4	59.6	55.3745	4.1965
5	49.9	51.8985	-2.0245
6	44.8	43.9200	0.8910
7	45.7	42.8606	2.8204
8	49.1	46.5022	2.5888
9	39.4	39.1197	0.3223
10	60.1	53.8517	6.2033
11	50.5	48.8858	1.6552
12	37.4	35.9747	1.4133
13	44.8	45.6084	-0.8544
14	47.3	47.3299	-0.0569
15	51.9	48.2237	3.6313
16	49.2	52.7923	-3.6363
17	40.8	46.1712	-5.3352
18	46.7	49.3162	-2.6442
19	46.8	48.4886	-1.7146
20	50.4	49.0514	1.3366
21	39.4	40.6095	-1.2025
22	46.1	45.4429	0.6371
23	45.4	50.5411	-5.1001
24	54.6	52.8916	1.7334
25	45.1	45.7408	-0.6228
26	39.2	39.7818	-0.5788
27	45.8	47.7603	-1.9703
28	50.5	49.5480	0.9970
29	48.7	51.3026	-2.6296
30	47.9	44.3504	3.5696
31	47.5	47.6609	-0.1939

Sum of Residuals	0
Sum of Squared Residuals	218.48144
Predicted Residual SS (PRESS)	250.90084

R Code and Output

proc_reg(fitness,
model = Oxygen ~ RunTime,
stats = p)

Output Statistics			
Obs	Dependant Variable	Predicted Value	Residual
1	44.6090	44.7808	-0.1718
2	45.3130	49.0845	-3.7715
3	54.2970	53.7855	0.5115
4	59.5710	55.3745	4.1965
5	49.8740	51.8985	-2.0245
6	44.8110	43.9200	0.8910
7	45.6810	42.8606	2.8204
8	49.0910	46.5022	2.5888
9	39.4420	39.1197	0.3223
10	60.0550	53.8517	6.2033
11	50.5410	48.8858	1.6552
12	37.3880	35.9747	1.4133
13	44.7540	45.6084	-0.8544
14	47.2730	47.3299	-0.0569
15	51.8550	48.2237	3.6313
16	49.1560	52.7923	-3.6363
17	40.8360	46.1712	-5.3352
18	46.6720	49.3162	-2.6442
19	46.7740	48.4886	-1.7146
20	50.3880	49.0514	1.3366
21	39.4070	40.6095	-1.2025
22	46.0800	45.4429	0.6371
23	45.4410	50.5411	-5.1001
24	54.6250	52.8916	1.7334
25	45.1180	45.7408	-0.6228
26	39.2030	39.7818	-0.5788
27	45.7900	47.7603	-1.9703
28	50.5450	49.5480	0.9970
29	48.6730	51.3026	-2.6296
30	47.9200	44.3504	3.5696
31	47.4670	47.6609	-0.1939

	Value
Sum of Residuals	-0.00000
Sum of Squared Residuals	218.48144
Predicted Residual SS (PRESS)	250.90084

Comparison	**Results**
Perform basic Reg on single variable and p option	Pass. Decimal difference on dep variable OK.

proc_reg-006

SAS Code and Output

```
data fitness2;
set fitness;
if (Age < 50) then do;
AgeCat = "<50";
end;
else do;
AgeCat = ">=50";
end;
run;

proc sort data = fitness2 out = fitness2;
by AgeCat;
run;


proc reg data = fitness2;
model Oxygen = RunTime;
by AgeCat;
run;
```

Number of Observations Read	19
Number of Observations Used	19

Analysis of Variance

Source	DF	Sum of Squares	Mean Square	F Value	Pr > F
Model	1	442.58253	442.58253	59.27	<.0001
Error	17	126.93987	7.46705		
Corrected Total	18	569.52240			

Root MSE	2.73259	R-Square	0.7771
Dependent Mean	48.19005	Adj R-Sq	0.7640
Coeff Var	5.67045		

Parameter Estimates

Variable	DF	Parameter Estimate	Standard Error	t Value	Pr > |t|
Intercept	1	81.93465	4.42771	18.50	<.0001
RunTime	1	-3.21087	0.41706	-7.70	<.0001

Number of Observations Read	12
Number of Observations Used	12

Analysis of Variance

Source	DF	Sum of Squares	Mean Square	F Value	Pr > F
Model	1	173.94484	173.94484	23.08	0.0007
Error	10	75.37221	7.53722		
Corrected Total	11	249.31705			

Root MSE	2.74540	R-Square	0.6977
Dependent Mean	46.08658	Adj R-Sq	0.6675
Coeff Var	5.95705		

Parameter Estimates

Variable	DF	Parameter Estimate	Standard Error	t Value	Pr > |t|
Intercept	1	83.13372	7.75240	10.72	<.0001
RunTime	1	-3.45992	0.72022	-4.80	0.0007

R Code and Output

```
fitness2 <- fitness
fitness2$AgeCat <- ifelse(fitness$Age < 50,
                  "<50", ">=50")

proc_reg(fitness2,
      model = Oxygen ~ RunTime,
      by = AgeCat)
```

	NOBS
Number of Observations Read	19
Number of Observations Used	19

Source	DF	Sum of Squares	Mean Square	F Value	Pr > F
Model	1	442.58253	442.58253	59.27	<.0001
Error	17	126.93987	7.46705		
Corrected Total	18	569.52240			

Root MSE	Dependent Mean	Coeff Var	R-Square	Adj R-Sq
2.73259	48.19005	5.67045	0.7771	0.7640

Variable	DF	Parameter Estimate	Std Err	t Value	Pr > |t|
Intercept	1	81.93465	4.42771	18.50	<.0001
RunTime	1	-3.21087	0.41706	-7.70	<.0001

	NOBS
Number of Observations Read	12
Number of Observations Used	12

Source	DF	Sum of Squares	Mean Square	F Value	Pr > F
Model	1	173.94484	173.94484	23.08	0.0007
Error	10	75.37221	7.53722		
Corrected Total	11	249.31705			

Root MSE	Dependent Mean	Coeff Var	R-Square	Adj R-Sq
2.74540	46.08658	5.95705	0.6977	0.6675

Variable	DF	Parameter Estimate	Std Err	t Value	Pr > |t|
Intercept	1	83.13372	7.75240	10.72	<.0001
RunTime	1	-3.45992	0.72022	-4.80	0.0007

Comparison	**Results**
Perform basic Reg on single variable with by groups.	Pass.

proc_reg-007

SAS Code and Output

```
proc reg data = fitness;
model Oxygen = RunTime Age;
run;
```

Number of Observations Read	31
Number of Observations Used	31

Analysis of Variance

Source	DF	Sum of Squares	Mean Square	F Value	Pr > F
Model	2	650.66573	325.33287	45.38	<.0001
Error	28	200.71581	7.16842		
Corrected Total	30	851.38154			

Root MSE	2.67739	R-Square	0.7642
Dependent Mean	47.37581	Adj R-Sq	0.7474
Coeff Var	5.65139		

Parameter Estimates

Variable	DF	Parameter Estimate	Standard Error	t Value	Pr > |t|
Intercept	1	88.46229	5.37264	16.47	<.0001
RunTime	1	-3.20395	0.35877	-8.93	<.0001
Age	1	-0.15037	0.09551	-1.57	0.1267

R Code and Output

```
proc_reg(fitness,
      model = Oxygen ~ RunTime + Age)
```

	NOBS
Number of Observations Read	31
Number of Observations Used	31

Analysis of Variance

Source	DF	Sum of Squares	Mean Square	F Value	Pr > F
Model	2	650.66573	325.33287	45.38	<.0001
Error	28	200.71581	7.16842		
Corrected Total	30	851.38154			

Root MSE	Dependent Mean	Coeff Var	R-Square	Adj R-Sq
2.67739	47.37581	5.65139	0.7642	0.7474

Parameter Estimates

Variable	DF	Parameter Estimate	Std Err	t Value	Pr > |t|
Intercept	1	88.46229	5.37264	16.47	<.0001
RunTime	1	-3.20395	0.35877	-8.93	<.0001
Age	1	-0.15037	0.09551	-1.57	0.1267

Comparison	**Results**
Perform basic Reg on single variable with two ind variables	Pass.

proc_reg-008

SAS Code and Output

```
proc reg data = fitness;
model Oxygen = RunTime Age Weight;
run;
```

Analysis of Variance

Source	DF	Sum of Squares	Mean Square	F Value	Pr > F
Model	3	656.27095	218.75698	30.27	<.0001
Error	27	195.11060	7.22632		
Corrected Total	30	851.38154			

Root MSE	2.68818	R-Square	0.7708
Dependent Mean	47.37581	Adj R-Sq	0.7454
Coeff Var	5.67416		

Parameter Estimates

Variable	DF	Parameter Estimate	Standard Error	t Value	Pr > |t|
Intercept	1	93.12615	7.55916	12.32	<.0001
RunTime	1	-3.14039	0.36738	-8.55	<.0001
Age	1	-0.17388	0.09955	-1.75	0.0921
Weight	1	-0.05444	0.06181	-0.88	0.3862

R Code and Output

```
proc_reg(fitness,
      model = Oxygen ~ RunTime + Age + Weight )
```

Analysis of Variance

Source	DF	Sum of Squares	Mean Square	F Value	Pr > F
Model	3	656.27095	218.75698	30.27	<.0001
Error	27	195.11060	7.22632		
Corrected Total	30	851.38154			

Root MSE	Dependent Mean	Coeff Var	R-Square	Adj R-Sq
2.68818	47.37581	5.67416	0.7708	0.7454

Parameter Estimates

Variable	DF	Parameter Estimate	Std Err	t Value	Pr > |t|
Intercept	1	93.12615	7.55916	12.32	<.0001
RunTime	1	-3.14039	0.36738	-8.55	<.0001
Age	1	-0.17388	0.09955	-1.75	0.0921
Weight	1	-0.05444	0.06181	-0.88	0.3862

Comparison	**Results**
Perform basic Reg on single variable with three ind variables.	Pass.