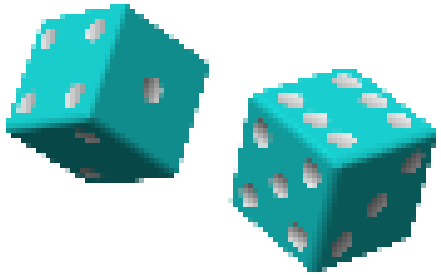
Brandy Sinco, Brief Bio

Brandy is a statistician and database programmer at the University of Michigan Medical School. She has worked on a wide variety of research projects. She holds a bachelors degree in engineering from UM, and masters degrees in mathematics (EMU) and statistics (Colorado State University). The American Public Health Association awarded Brandy a student research award for her statistics masters thesis on path analysis. She has also received "Advanced Analytics" and "Data for Good" awards from the Midwest SAS Users Group conference.

Outside of work, she enjoys playing the piano, composing music, singing in choir, yoga, tai chi, and martial arts aerobics.



Using SAS® to Generate p-Values with Monte Carlo Simulation

Brandy R. Sinco, BS, MA, MS, University of Michigan, Ann Arbor, MI

Statistician Senior

Center for Healthcare Outcomes and Policy (CHOP)

Outline

- Why is adjusting for multiple comparisons necessary?

- **Cool SAS code** for Procs FREQ, MULTTEST, TTEST, NPAR1WAY, LOGISTIC, GENMOD, MIXED, GLIMMIX, LIFETEST, PHREG.

- Examples from two unrelated studies.
  - Study on Lesbian, Gay, Bisexual, Transgender (LGBT) college students and mental health.
  - Study on Diabetes among inner city Detroit residents.

- Conclusions.

Why Multiple Comparisons Increase the Type 1 Error Rate

- Let α = probability of a type 1 error, i.e., probability of rejecting the null hypothesis when the null hypothesis is true. I.E., concluding that the test statistic indicates a significant result, such as a difference between two means, when the result is really insignificant.
- Let α' = probability of at least one type 1 error on m tests.
- Example of Type 1 Error: Concluding that there is a significant difference between groups 1 and 2, when no difference exists.
- If α = .05 and two comparisons are made, such as group 1 to control and group 2 to control, the probability of not making a type 1 error on either test = $(1 – \alpha)^2 = (1 - .05)^2 = .9025$. The probability of making a type 1 error on at least one of the two tests = $1-(1 – \alpha)^2 = .0975$.
- If m tests are done, the type 1 error, $\alpha' = 1- (1 – \alpha)^m$.
- For α = .05 and m = 10, $1- (1 – \alpha)^m = 0.5987$. α' = probability of at least one type one error, increase from .05 to .5987.
- Multiple comparisons increase α'. Goal of adjusting for multiple comparisons is to set α' = α.

Remedies for Multiple Comparisons in Statistical Literature

- 1) Smaller value of α. (Bonferroni, Sidak methods).
- 2) Tests with family-wise error rate of α. (Tukey, Scheffe, Dunnett methods.)
- 3) Monte Carlo simulation. (Bootstrap and Permutation tests.)

Smaller Value of Alpha, α.

- Use a smaller α, α_m, for each of the m hypothesis tests, so that overall type-1 error threshold for the m tests will be α.
- Bonferroni proposed that $\alpha_m = \alpha/m$. (Bonferroni, 1936).
- Sidák correction $\alpha_m = 1 - (1 - \alpha)^{1/m}$. (Sidák, 1967).
- If m = 10 and α = .05, Bonferroni's α_m = .005 and Sidák's α_m = .005116.
- Problem: α becomes small with increasing m and may cause type-2 error, which is failing to detect a significant result. Example, failing to detect a significant drop in blood pressure from an experimental medication.
- Simulation studies indicate that the Bonferroni and Sidák corrections are overly conservative. (Hsu and Nelson, 1998).
- Both options available in SAS on LSMEANS statement, Adjust = BON and Adjust = SIDAK. Available on Estimate statement of some procedures.

Family-Wise Error Rate of α, Alternative Test Statistics (Kutner, et al, 2005)

- Alternative test statistics to T and F.
- Tukey's Q statistic to compare m means, 2 at a time, total of m(m-1)/2 comparisons.
- Scheffe's method of comparing linear combinations of m means, using modified version of the F statistic.
- Dunnett's procedure to compare (m–1) treatment means to a control.

- Tukey's procedure is better for comparing multiple differences between means.
- Whereas, Scheffe's procedure is better for linear contrasts of multiple sample means.

- Available on LSMEANS statement, sometimes on Estimate statement with options Adjust=Tukey, Adjust=Scheffe, Adjust=Dunnett.

Adjustment with Monte Carlo Simulation

- Most reliable, efficient, state of the art method (Edwards and Berry, 1987).
- Involves repeated random sampling and estimation of confidence intervals from large number of random samples.
- Bootstrapping and Permutation tests.
- “When you have a choice of methods to adjust for multiple comparisons, Monte Carlo methods are the best choice. The reason they haven’t been used until recently is lack of computing power, which is no longer an issue.” Dr. Phil Chapman, Professor of Statistics at Colorado State University.
- Available in SAS with Adjust=Simulate option on LSMEANS or Estimate statements, Exact statement with MC option, or post-processing via Proc PLM. Syntax varies between SAS procedures.

Monte Carlo Simulation

- Multiple random samples from a population to estimate a parameter or a function of random variables (Sawilowsky, 2003).
- In theory, the grand mean from the collection of random samples will converge to the parameter estimate.
- Illustrate Monte Carlo simulation by examples rather than by complex equations.
- Modern version of the Monte Carlo simulation, known as the Markov Chain Monte Carlo method co-invented by Stanislaw Ulam in the 1940's, while working on nuclear weapons projects at the Los Alamos National Laboratory.
- Worked with John von Neumann to develop the algorithm.
- Due to the top secret nature of the work, this new method needed a “code” name.
- Nicholas Metropolis, a colleague of Ulam and Neumann, suggested using the name “Monte Carlo” after a casino where his uncle liked to gamble.

Monte Carlo Simulation Example 1

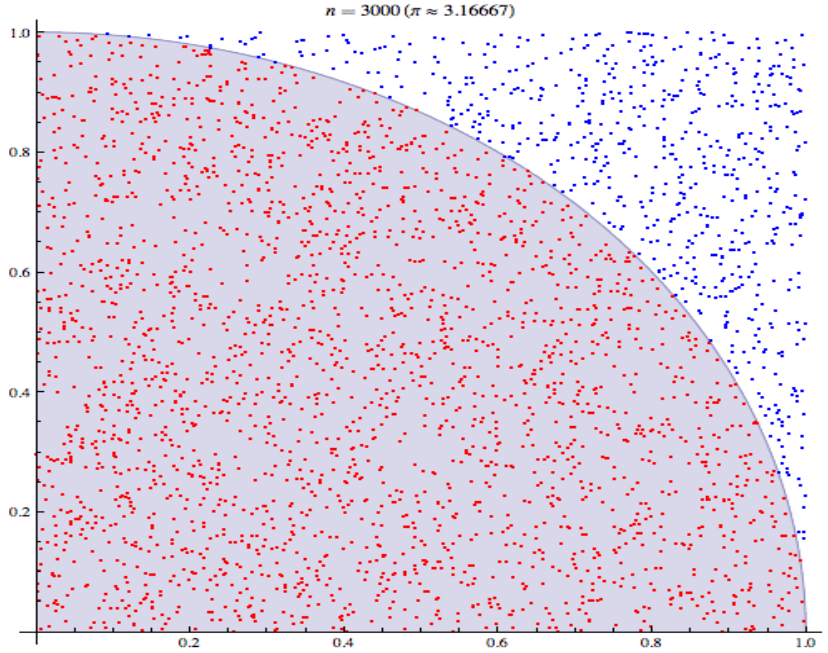
- Consider a coin toss.

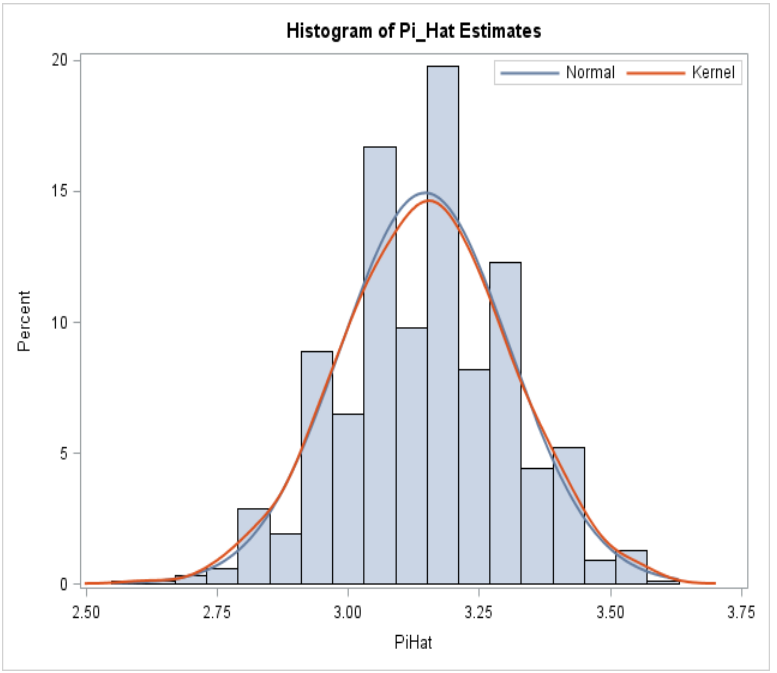
- To simulate the probability of the coin producing a head, we could draw one random sample from a uniform[0, 1] pseudo-random uniform variable. A value ≥ 0.50 would indicate a head and a value < .5 would indicate a tail. This would be a simulation, but not a Monte Carlo simulation.

- To make this a Monte Carlo simulation, draw a large number of pseudo-random samples from the interval [0,1]. Then, estimate the probability of head by taking the mean of the probabilities of rolling a head from all of the samples.

Monte Carlo Simulation Example 2

- For the second example, consider estimating the value of π from a circle of radius R and a square with each side of length 2R.
- Let C = area of the circle = πR^2 and S = area of the square = $4R^2$.
- Then, π = 4C/S, where C/S is the probability of a sample point being inside a circle.
- Random sample of 1000 Uniform(0,1) (X, Y) pairs; 1 if inside circle, 0 if outside circle.



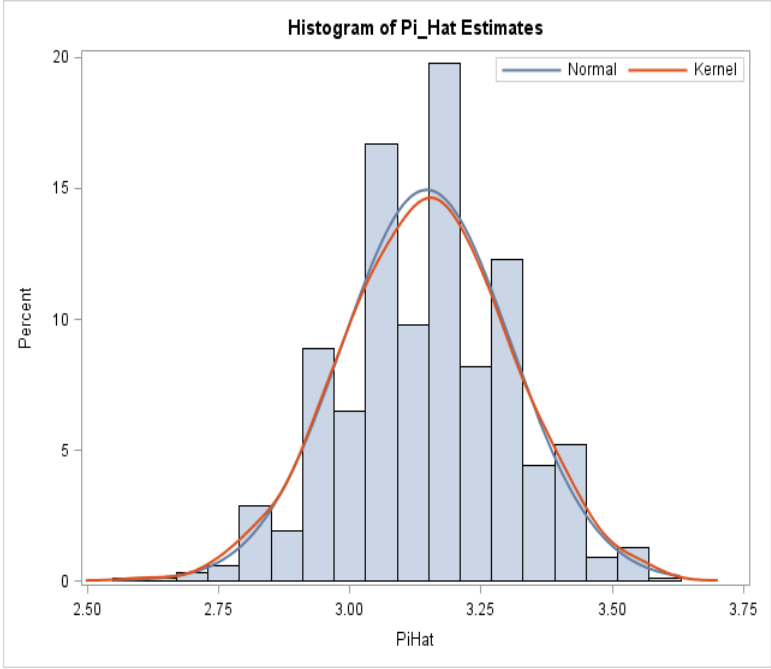


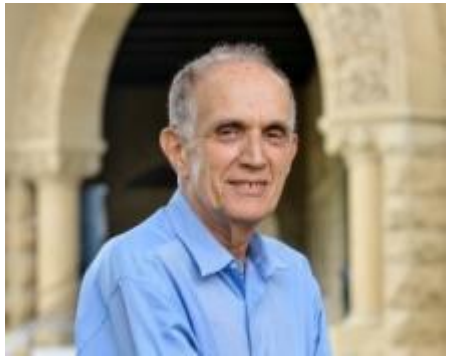
Monte Carlo Simulation Example 2 – SAS Code

```
/* SAS Program to Estimate π From 1000 Random Samples of 100 */
Data Circle;
Do B=1 to 1000; Do I=1 to 100;
 Call streaminit(B);
 X = Rand('UNIFORM', 0, 1);   Y = Rand('UNIFORM', 0, 1);
 R=Sqrt((X**2)+(Y**2));
 If R≤1 then InsideCircle=1;  else InsideCircle=0;  Output;
End;  End; Run;

PROC MEANS data=circle noprint;
Class B;
var InsideCircle;
Output out=circleby mean=PiQuarterHat;
run;
```

Monte Carlo Simulation Example 2 – SAS Code

- /* Delete first row because it contains grand mean */
- data circleby;
- set circleby;
- if _N_=1 then delete;
- PiHat=4*PiQuarterHat; run;
-
- Title 'Histogram of Pi_Hat Estimates';
- PROC SGPLOT data=CircleBy;
- histogram PiHat;
- density PiHat;
- density PiHat / type=kernel;
- keylegend / location=inside
- position=topright; run;





Bootstrapping and Permutation Tests (Efron and Tibshirani, 1993)

- Bootstrapping is used to estimate a confidence interval for a parameter estimate, such as a regresssion coefficient or a percentage or a mean. Can also be used for hypothesis tests.

- Permutation tests are used to assess differences between groups.

- Focus on bootstrapping first.
- Bootstrapping was invented by Bradley Efron and is a method for estimating the empirical distribution and confidence interval for a test statistic.
- Suppose we have a random sample of x, size n, with an empirical density function, f. $x_1, x_2, ..., x_n \sim f(x)$. $x_1, x_2, ..., x_n$ are assumed to be independent and identically distributed.
- Let T(x) be a test statistic for which we want to find a $(1 - \alpha)$ confidence interval.

Bootstrapping (Continued)

- To estimate the empirical distribution of T(x) by bootstrapping, we take B independent random samples with replacement of size n, and calculate T(x) for each replication. The steps for bootstrapping are as follows:

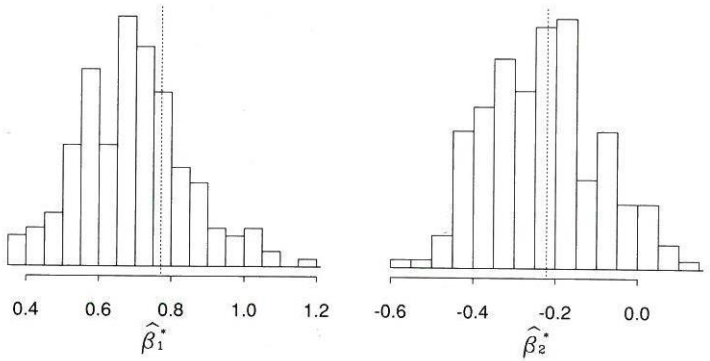
- 1. Take a random sample of size n with replacement from the sample set.
- 2. Calculate the statistic of interest, T(x), for each random sample.
- 3. Repeat steps 1 and 2 B times (B is usually ≥10,000).
- 4. Create a relative frequency histogram of the B statistics of interest, T(x), by placing a probability of 1/B at each estimated statistic.

- Point Estimate of T(x), $$T_B = \frac{1}{B}\sum_{b=1}^{B} T\left(x^*_b\right)$$
- where x^*_b = b^{th} bootstrap sample from $x_1, x_2, \ldots, x_n$.
- Standard error of T(x), $$se_B = T_B = \sqrt{\frac{1}{(B-1)}\sum_{b=1}^{B}\left(T\left(x^*_b\right) - T_B\right)^2}$$

Bootstrapping (Continued)

- Estimate (1 - α)*100% confidence interval for T(x) from the (α/2) and (1 - α/2) percentiles of the bootstrap density function.

- Bootstrapping can be used to estimate confidence intervals of statistics that are functions of more than one random variable, x, such as a correlation coefficient, regression coefficient, and even effect size. At the 2015 Conference on Statistical Practice, Erin Smith and Jason Osborne co-authored a presentation on bootstrapping confidence intervals for effect sizes.
- Histograms of Bootstrap Distributions for Regression Coefficient Estimates from Efron & Tibshirani (1993).



Bootstrapping in SAS Example – Sample Mean

- Data VerizonSubset;
- Call streaminit(10192026);
- Do I=1 to 25;
- X = Rand('NORMAL', 45, 5);   output;
- End;
- Drop I;
- Run;

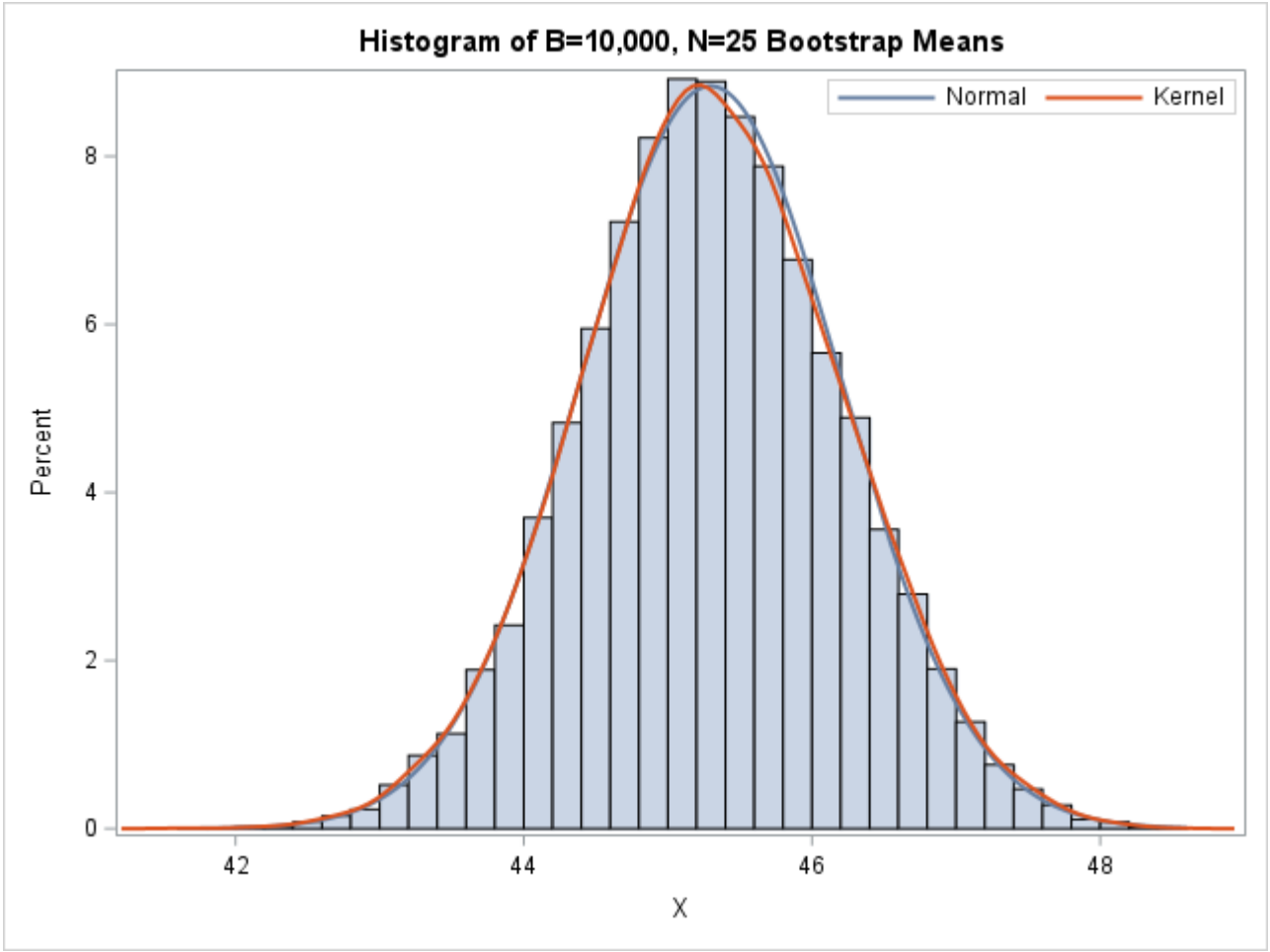
- proc surveyselect data=VerizonSubset Seed=7252016 method=urs n=25 reps=10000 out=bootsamp25 outhits;
- run;

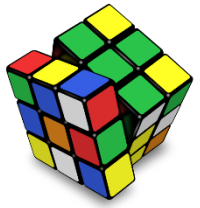
Bootstrapping in SAS Example – Sample Mean

```
Title 'Bootstrap Means';
ods html path="c:\temp";
proc means data=bootsamp25;
Class Replicate;
var  X;
Output Out=BootMeans25;
run;
ods html close;

Data BootMeans25;
Set BootMeans25;
Keep Replicate _STAT_ X;
if Replicate NE . and  _STAT_='MEAN';
Run;
/* xbar=45.3, sigma=4.6, min=35.7, max=52.8  */
```

Figure: Bootstrapped Means Histogram and SAS Code

```
Title 'Histogram of B=10,000 Bootstrap Means, N=25';
ods html path="c:\temp";
PROC SGPLOT data=BootMeans25;
    histogram X;
    density X;
    density X / type=kernel;
     keylegend / location=inside
     position=topright;
run;
```



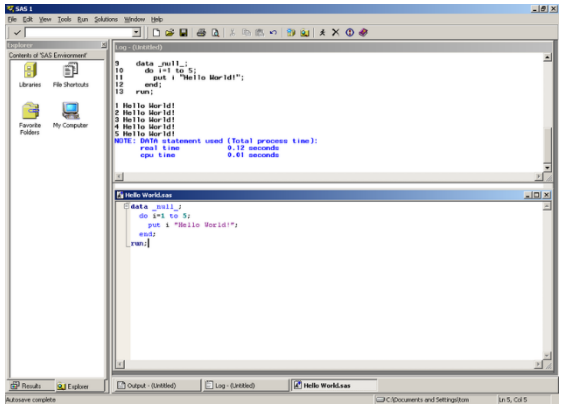


Permutation Tests (Efron and Tibshirani, 1993)

- Let x be a random sample of size n, with n_1 in group 1 and n_2 in group 2.

- Let T(x) be a test statistic, such as a T for a mean test or a X^2 for a percentage.
- Then, randomly divide x into groups of n_1 and n_2.
- Re-compute the test statistic, $T_p(x)$, for each combination of x, randomly subsetted into sizes of n_1 and n_2.
- The total number of subsets will be $n!/(n_1!n_2!)$.

- Let M = number of trials for which $|T_p(x)| \geq |T(x)|$.
- Let N = $n!/(n_1!n_2!)$.
- Then, the p-value from the permutation test, also known as an “exact test” = M/N.
-
- If the dataset is too large to calculate the test statistic for N subsets, Monte Carlo sampling will be used.



Part 2: Focus on Applications in SAS Procedures: Procs FREQ, MULTTEST, GLM, NPAR1WAY, MIXED, GLIMMIX, LIFETEST, PHREG

Proc Freq for Percentages and Count

- In PROC FREQ, p-values with Monte Carlo simulation can be created by using the Exact statement with the MC option.

- The exact statement tells SAS to use a permutation test, which also known as the Fisher Exact Test and is based on the hypergeometric distribution.

- When the MC option is included on the Exact statement, SAS computes Monte Carlo estimates regardless of the sample size.

- PROC FREQ will not do any Monte Carlo simulations unless the MC option is included.

Proc Freq for Percentages and Count

According to the PROC FREQ documentation, the following equations are used to compute the Monte Carlo p-value.

- t = observed test statistic
- M = number of samples with Test Statistic >= t
- N = total number of samples
- P_{MC} = p-value adjusted with Monte Carlo simulation = M/N.
- se(P_{MC}) =standard error of the Monte Carlo p-value = sqrt(P_{MC}(1-P_{MC})/(N-1)).

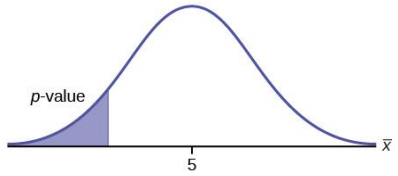
For an example, I will use PROC FREQ to compare depression, suicide, and demographics between cisgender and transgender college students. Cisgender refers to a person having the gender identity that was assigned at birth and transgender refers to seeing one's gender identity in a different way. In the PROC FREQ code below, Monte Carlo simulation is invoked by the MC option.

Proc Freq for Percentages and Count

```
PROC FREQ Data=intersect;
  Tables (PHQ3Cat Suicide_NY Audit_BRS NHW disability_diagnosis_NY
SES_pell_NY conference_attend) * gender_trans / ChiSq Exact NoCum;
  Exact Fisher / MC seed=03202015;
```

- For depression, p < .0001 without Monte Carlo Simulation and .0018 with Monte Carlo.

Monte Carlo Estimate for the Exact Test p-Value	
Pr ≤ P	0.0018
99% Lower Conf Limit	0.0007
99% Upper Conf Limit	0.0029
Number of Samples	10000
Initial Seed	3202015

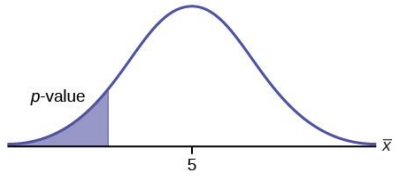


Proc MULTTEST for T-Tests

- Use PROC MULTTEST or TTEST to do a T-Test with Monte Carlo adjustment.
- In PROC MULTTEST, using bootstrap sampling within-strata is a more time efficient method of doing t-tests, rather than doing permutation tests.
- Monte Carlo p-value will be based on bootstrap samples within the groups to be tested.

The PByTest plot is highly recommended and illustrates the effect of multiple comparison adjustment. The example below is from a diabetes study.

```
PROC MULTTEST data=across bootstrap nsample=10000 seed=4102015
outsamp=res plots=PByTest(vref=0.05 0.1);
 test mean(BL_HbA1c BL_LDLCholesterol BL_HDLCholesterol
BL_TotalCholesterol BL_SystolicBPAve BL_DiastolicBPAve
BL_WaistInches);
 class RandomizationBin;
```



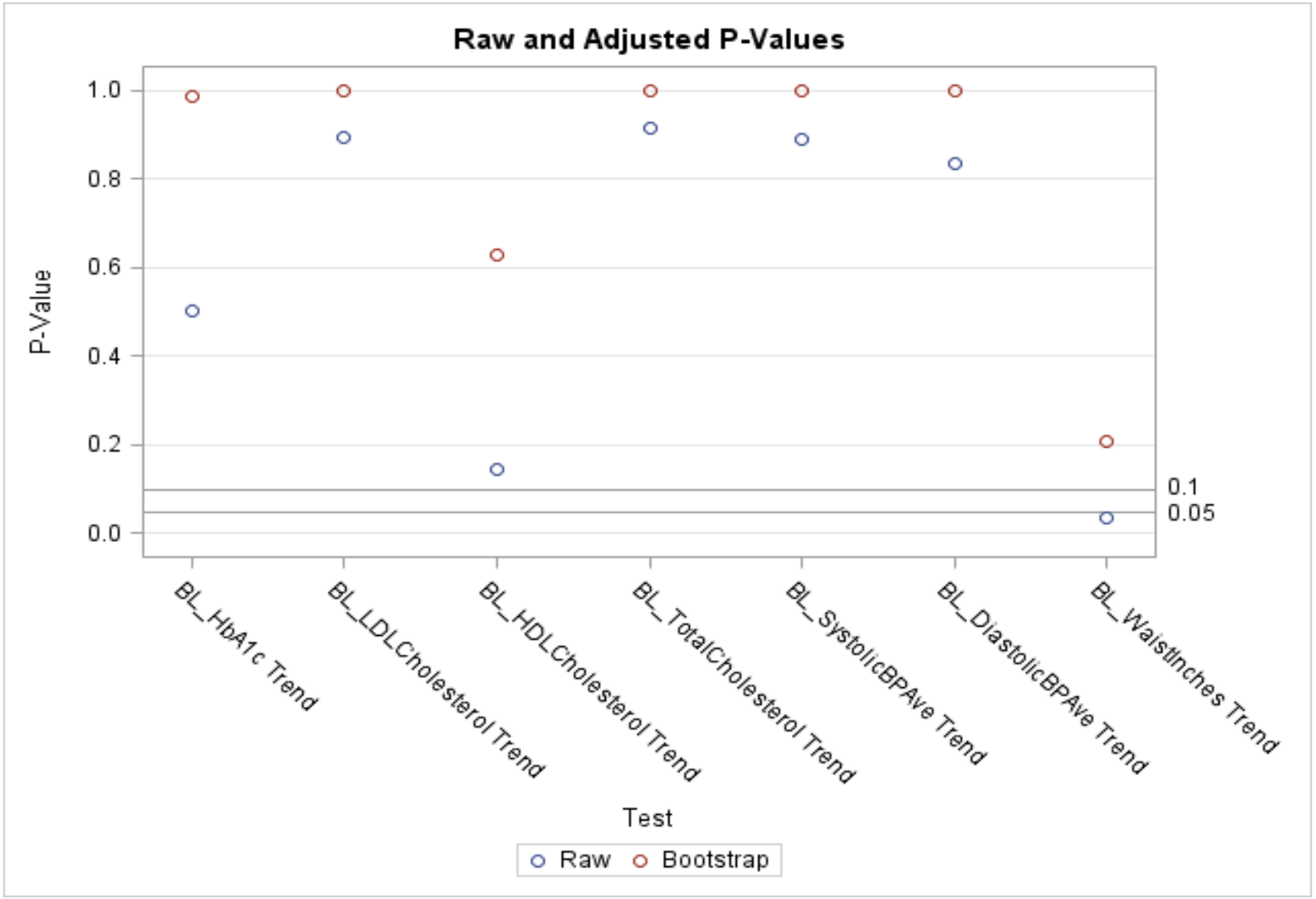
Proc MULTTEST for T-Tests

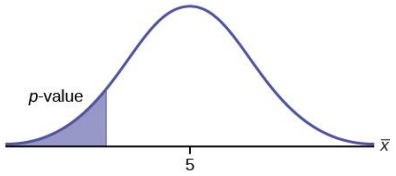
- The output from PROC MULTTEST will include both the raw and bootstrapped p-values.
- The plot and table indicate that the p-value for waist circumference (BL_WaistInches) is significant without Monte Carlo adjustment, but loses significance after adjustment.

Table of Raw and Bootstrapped p-values

Variable	Raw p-Value	Bootstrap p-Value
BL_HbA1c	0.5044	0.9877
BL_LDLCholesterol	0.8950	1.0000
BL_HDLCholesterol	0.1454	0.6284
BL_TotalCholesterol	0.9158	1.0000
BL_SystolicBPAve	0.8921	1.0000
BL_DiastolicBPAve	0.8367	1.0000
BL_WaistInches	**0.0358**	**0.2086**

Raw and Adjusted P-Values from PROC MULTTEST





Proc TTEST

Add “plots = bootstrap” to proc statement and the bootstrap statement to Proc TTest.

- PROC TTEST data=across **plots = bootstrap**;
- var BL_HbA1c BL_LDLCholesterol BL_HDLCholesterol BL_TotalCholesterol BL_SystolicBPAve BL_DiastolicBPAve BL_WaistInches;
- class RandomizationBin;
- **bootstrap / seed=4102015 nsamples = 10000;**
- run;

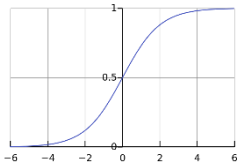
PROC NPar1Way for Wilcoxon or Kruskal-Wallis Tests

- The Wilcoxon test is used to compare non-normal, but ordinal, data between two groups. Nonparametric version of the T-test.
- The Kruskal-Wallis test is an extension of the Wilcoxon test for more than two groups.
- In SAS, both tests are implemented in PROC NPar1Way with the same syntax.
- If there more than two groups, SAS will perform a Kruskal-Wallis test with the Wilcoxon statement.
- The syntax is similar to PROC FREQ and the equations for p_{MC} are the same as for PROC FREQ.

PROC NPar1Way for Wilcoxon or Kruskal-Wallis Tests

```
PROC NPAR1WAY wilcoxon data=fromacc.mainvar_cohort3;
 class Randomization;  /* Class is treatment group */
 var TotalClasses;  /* Compare attendance between groups. */
 exact wilcoxon/ MC n=100000 seed=102752011;
```

Monte Carlo Estimates for the Exact Test	
One-Sided Pr >= S	
Estimate	0.2350
99% Lower Conf Limit	0.2315
99% Upper Conf Limit	0.2385



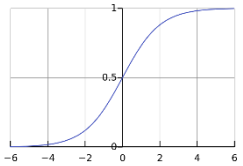
PROCS LOGISTIC & GENMOD for Category & Count Regressions

- In Proc Logistic, the Estimate statement has an “Adjust=“ option and Adjust=Simulate is one of the options.

- Use a seed, so that each run of the procedure will produce the same results.

- While NSAMP=10000 could be added to the estimate statement, the SAS default is 12,604. SAS produces the confidence interval for the estimate by bootstrapping.

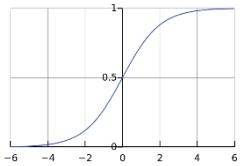
- Proc GENMOD has the Adjust=Simulate option on the LSMEANS and LSMEstimate statements, but not on the Estimate statement. To generate boostrapped p-values for an Estimate produced by Proc GenMod, need to use Proc PLM after Proc GenMod.



PROC LOGISTIC for Category & Count Regressions

- In the example below, I am estimating the odds of depression, predicted by outness, among transgender college students, separate estimates for People of Color and for non-Hispanic Whites. The model indicated that outness, i.e. feeling safe disclosing one's identity significantly reduced the odds of depression among People of Color.

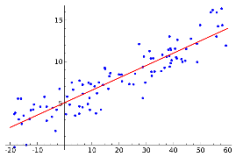
```
PROC LOGISTIC Descending Data=Intersect;
Model PHQ3Cat=age  SoCat4_BRSN NHW disability_diagnosis_NY SES_pell_NY
... /link=cumlogit aggregate CLOdds=Both;

Estimate 'Outness POC' LGBTQ_outness_cen 1 /adjust=simulate seed=2052015;

Estimate 'Outness NHW' LGBTQ_outness_cen 1 LGBTQ_outness_cen*NHW 1
/adjust=simulate seed=2052015;

where gender_trans=1;
run;
```



PROC LOGISTIC for Category & Count Regressions

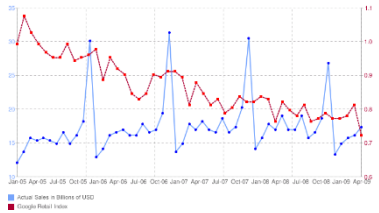
In this example, the raw p-value and bootstrapped p-values are similar. Boostrapping does not always inflate the p-value.

Depression 3 categories	Estimate	Standard Error	z Value	Raw p-value Pr > |z|	Bootstrap p Adj P
2	-1.4649	0.4738	-3.09	0.0020	0.0017



PROCS GLM and PLM for Linear Regression

- For linear regression, PROC REG does not have an “Estimate” statement.
- Although, PROC GLM does have an estimate statement, only LSMEANS has the “adjust=” option.
- Like PROC GENMOD, the only way to currently get Monte Carlo p-values for linear regression estimates is to route the results from the “estimate” statement in PROC GLM to PROC PLM or to use the LSMeans or LSMEstimate statements.
- In the next section on PROC MIXED, the syntax for using PROC PLM will be displayed and the same syntax can be used with PROCS GLM and GENMOD. In Proc Mixed, the Adjust = Simulate option is available on the LSMeans and LSMEstimate statements.

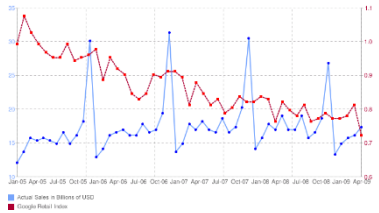


PROCS MIXED and PLM for Linear Mixed Models

- Write Model statement in Proc Mixed. **Use the Store statement.** Then, write the Estimate statement in Proc PLM.
- Proc PLM does post-processing of linear models, including multiplicity-adjusted estimates.

/* Usual Proc Mixed Code. Add the Store statement. */

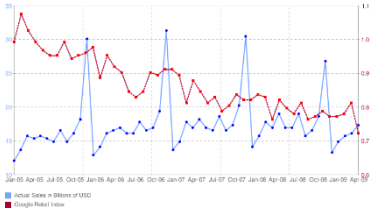
```
PROC MIXED Data= A_Long Method=REML NOCLPRINT
plots(only)=(StudentPanel(conditional box));
Class REACHID TimepointN RandomizationN;
Model HbA1c=BL_HbA1c*RandomizationN TimepointN RandomizationN
TimepointN*RandomizationN HSGrad MoreDiabMeds / Solution
Influence(effect=REACHID Est) ddfm=KR;
Repeated TimepointN / Type= CS Subject=REACHID R RCorr;
Store MixRes;
run;
```



PROCS MIXED and PLM for Linear Mixed Models

Then, Write the estimate statements in PROC PLM with the “adjust=simulate” option and PROC PLM produces the estimates with p-values adjusted for Monte Carlo simulation. Last, route the PLM output to PROC Print to display.

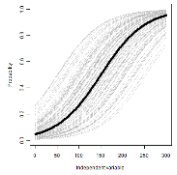
```
PROC PLM Restore=MixRes;
Estimate 'Ref M6 - BL'  Intercept 1  RandomizationBinN 0 1
TimePointN 0 0 1 TimepointN*RandomizationBinN 0 0 0 0 0 1
BL_HbA1c*RandomizationBinN 0 &BLRef
  MoreDiabMeds .3588 HSGrad .306 / adjust=simulate(NSAMP=10000
SEED=3282016);

Estimate 'Trt M6 - BL'  Intercept 1  RandomizationBinN 1 0
TimePointN 0 0 1 TimepointN*RandomizationBinN 0 0 0 0 1 0
BL_HbA1c*RandomizationBinN &BLGrp 0
  MoreDiabMeds .3588 HSGrad .306 / adjust=simulate(NSAMP=10000
SEED=3282016);
 ODS Output Estimates=PLMEst;
```



PROCS MIXED and PLM for Linear Mixed Models

The output from PROC PLM will produce the following table, indicating the p-values for the HbA1c (hemoglobin A1c) difference estimates between time points and treatment group are not being distorted by multiple comparisons.

Label	Estimate	StdErr	Probt	Adjustment	Adjusted p
Ref M6 - BL	-0.103	0.1533	0.5025	Simulated	0.4951
Trt1 M6 - BL	**-0.38**	**0.1377**	**0.0065**	**Simulated**	**0.0077**
Trt2 M6 - BL	**-0.789**	**0.175**	**<.0001**	**Simulated**	**<.0001**
Trt2-Trt1 M6-BL	-0.408	0.2226	0.0685	Simulated	0.0659
AveTrt1,2- RefM6-BL	**-0.471**	**0.1899**	**0.0142**	**Simulated**	**0.0146**
Int Effect1 M6 - BL	-0.277	0.2067	0.182	Simulated	0.1897
Int Effect2 M6 - BL	**-0.686**	**0.2326**	**0.0037**	**Simulated**	**0.0039**

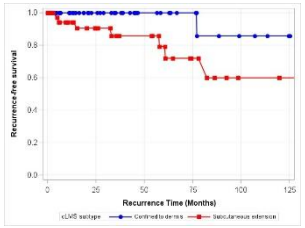


PROC GLIMMIX FOR GENERALIZED LINEAR MIXED MODELS

- While I expected PROC GLIMMIX to have the same syntax for Monte Carlo p-values on Estimate statements as PROC MIXED, I was surprised to learn that PROC GLIMMIX has the same syntax as PROC LOGISTIC.
- I.E., `Estimate / Adjust=Simulate Seed = …;`

- Why do PROCS LOGISTIC and GLIMMIX have the Adujst=Simulate option on the Estimate statement, but not PROCS GLM and MIXED?

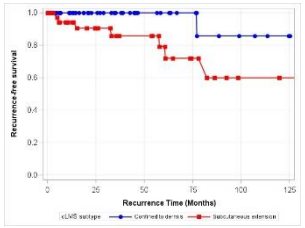
- According Cyrus Bradford, SAS Technical Support, regarding PROC GLIMMIX:
- “PROC GLIMMIX is a newer procedure than PROC MIXED. It introduced expanded functionality, including the ADJUST=SIMULATE Option in the ESTIMATE Statement, which was not available in PROC MIXED.”



PROC LIFETEST FOR TESTING SURVIVAL TIMES BETWEEN STRATA

- Adjust = simulate is an option on the Strata statement in Proc LifeTest.

- ods graphics on;
- proc lifetest data=sashelp.BMT plots=survival(atrisk=0 to 2500 by 500);
- time T * Status(0);
- strata Group / test=logrank adjust = simulate(SEED = 10122026 NSAMP = 10000);
- run;
- ods graphics off;



PROC PHREG FOR COX PROPORTIONAL HAZARD REGRESSION

- Adjust = simulate is an option on the Estimate, LSMeans, and LSMEstimate statements in Proc PHReg.

- ods graphics on;
- proc phreg data = Lms_jessie2025;
- class primary_subtype3 (ref = ‘1’);
- model RecurTime_Months*CensorRecur(1) = Age Female primary_subtype3 UpperLowerSite GradeCat3/ ties=breslow firth;
- assessvar=(Age);
- **Estimate ‘Subtype 1 vs 2' primary_subtype3 1 -1 Age 45.6 Female 0.5 /**
- **Adjust = Simulate(seed = 10132026 NSAMP = 10000);**
- run;
- ods graphics off;

Conclusions

- Monte Carlo simulation is an effective method of adjusting for multiple comparisons and is available in SAS for comparisons of percentages and means, linear and logistic regression, and mixed models.
- **Syntax varies between SAS procedures, always check documentation.**
- When comparing percentages, use PROC FREQ with the Exact statement and the MC option for Monte Carlo simulation. PROC NPar1Way has same syntax as PROC FREQ.
- For the T-Test, use PROC MULTTEST with the “test mean” option and remember the “plots=PByTest” option on the PROC MULTTEST statement.
- Can also use Proc TTest with the Bootstrap option..
- For linear regression and linear mixed models, use PROCS GLM and MIXED, respectively, and then PROC PLM for post-processing of estimates.
- For logistic regression,generalized linear mixed models, or survival analysis, PROCS LOGISTIC, GLIMMIX, and PHREG already have an “Adjust=Simulate” option on the estimate statement.

Acknowledgements

- Peers for Progress grant from the American Association of Family Physicians Foundation (grant P30DK092926) (Michigan Center for Diabetes Translational Research), by the National Institute of Diabetes and Digestive and Kidney (NIDDK) Diseases (grant R18DK0785501A1), and by the Centers for Disease Control and Prevention (cooperative agreement no. U50/CCU417409).

- Centering Gender Identity, Gender Expression, and Race in Research, Policy, and Programs to Support LGBTQ College Students through Intersectional Research.

- Faculty Seed Grant, Small Grant Awards, Institute for Research on Women & Gender, University of Michigan; Critical Intersectionality Group, School of Social Work, University of Michigan.

References

1. Bonferroni CE. Teoria statistica delle classi e calcolo delle probabilità. Pubblicazioni del R Istituto Superiore di Scienze Economiche e Commerciali di Firenze. 1936.

2. Šidák ZK. Rectangular confidence regions for the means of multivariate normal distributions): 626–633. JASA. 1967;62(318):626-633.

3. Hsu JC, Nelson B. Multiple comparisons in the general linear model. Journal of Computational and Graphical Statistics. 1998;7(1):23-41.

4. Kutner MH. Applied linear statistical models. Boston: McGraw-Hill Irwin; 2005:xxviii, 1396 p.

5. Edwards D, Berry JJ. The efficiency of simulation-based multiple comparisons. Biometrics. 1987;43:913-928.

6. Sawilowsky SS. Target article: You think you've got trivials? Journal of modern applied statistical methods. 2003;2(1):218-225.

7. Efron B. Bootstrap methods: Another look at the jackknife. The Annals of Statistics. 1979;7(1):1-26.

8. Efron B, Tibshirani R. An introduction to the bootstrap. New York: Chapman & Hall; 1993:xvi, 436 p.

9. Smith E, Osborne JW. Bootstrapping confidence intervals for effect sizes (and other weird things). . 2015;Conference on Statistical Practice, New Orleans.

10. Agresti A. A survey of exact inference for contingency tables. Statistical Science. 1992;7(1):131-153.

11. Mehta CR, Patel NR. A network algorithm for performing fisher's exact test in r × c contingency tables. Journal of the American Statistical Association. 1983;78(382):427-434.

12. Westfall PH, Young SS. Resampling-based multiple testing: Examples and methods for P-value adjustment. New York: Wiley; 1993:xvii, 340 p.

13. Mehta CR, Patel NR, Senchaudhuri P. Efficient monte carlo methods for conditional logistic regression. Journal of the American Statistical Association. 2000;95(449):99-108.

Contact Information

- Brandy R. Sinco, Statistician and Programmer/Analyst
- University of Michigan
- Michigan Medicine
- Ann Arbor, MI 48109-2800
- E-Mail: brsinco@umich.edu