Week 16

For the homework, you used PROC GLM just for its ability to test for equal variances. Today, we'll look at some more features of PROC GLM. This procedure is very old and fairly versatile. It can handle many general linear model problems.

PROC GLM

Here are some of the types of analyses that you can do using PROC GLM:

- ANOVA (balanced or unbalanced)
- ANCOVA (analysis of covariance – i.e., mixed categorical and numerical predictors)
- Simple or multiple regression
- Response surface models
- Weighted regression
- Polynomial regression
- Partial correlation
- MANOVA (multivariate ANOVA)
- repeated measures ANOVA
- estimates of linear combinations of coefficients

$$a_0\beta_0 + a_1\beta_1 + \cdots + a_k\beta_k$$

- multiple comparisons

PROC GLM

Although PROC GLM can do lots of different analyses, if a more specialized procedure can do the analysis, then this is usually considered more efficient. This might be important if you are doing many separate analyses such as in a simulation, genetic data, or large business applications.

For example, both PROC GLM and PROC TTEST can perform *t*-tests, but PROC TTEST would generally be more efficient. Similarly, if you have balanced data, then PROC ANOVA would be more efficient than PROC GLM according to the manuals (I haven't tested this...)

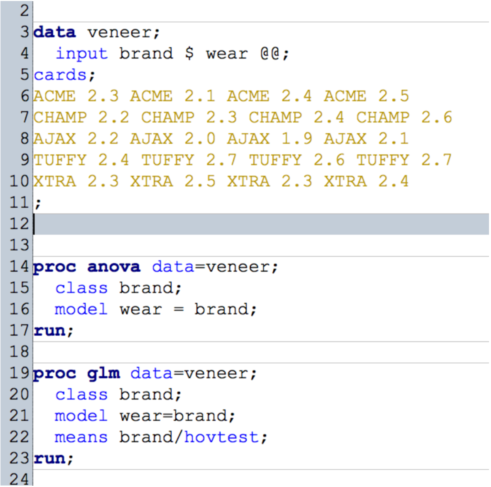
It's good to realize that all of these statistical approaches are related, though, and therefore it makes sense that the same procedure can handle so many: two sample *t*-tests are (generally) unbalanced ANOVAs with two categories, ANOVA is regression on dummy variables, and so forth.

PROC GLM

Similarly to what I said earlier about PROC IML being useful for understanding statistics, you can use different SAS procedures to run equivalent analyses, and figuring out what output from PROC X corresponds to in the output from PROC Y can help you to understand both the statistics and SAS.

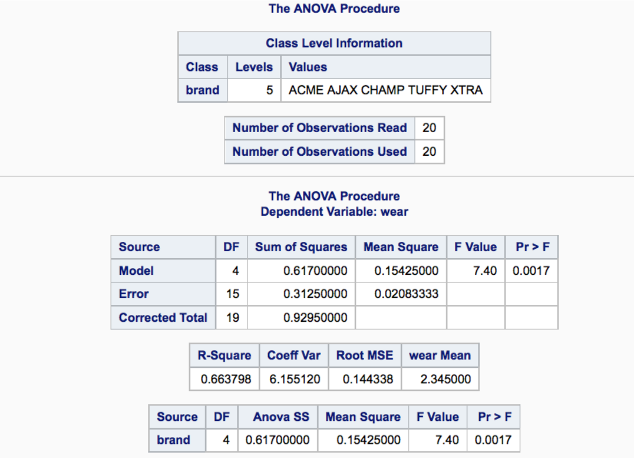
In some cases, there might also be discrepancies, such as if one procedure uses Least Squares while another procedure uses Maximum Likelihood to get estimates.

PROC ANOVA



```
data veneer;
  input brand $ wear @@;
cards;
ACME 2.3 ACME 2.1 ACME 2.4 ACME 2.5
CHAMP 2.2 CHAMP 2.3 CHAMP 2.4 CHAMP 2.6
AJAX 2.2 AJAX 2.0 AJAX 1.9 AJAX 2.1
TUFFY 2.4 TUFFY 2.7 TUFFY 2.6 TUFFY 2.7
XTRA 2.3 XTRA 2.5 XTRA 2.3 XTRA 2.4
;


proc anova data=veneer;
  class brand;
  model wear = brand;
run;

proc glm data=veneer;
  class brand;
  model wear=brand;
  means brand/hovtest;
run;
```

PROC ANOVA



The ANOVA Procedure

Class Level Information		
Class	Levels	Values
brand	5	ACME AJAX CHAMP TUFFY XTRA

Number of Observations Read	20
Number of Observations Used	20

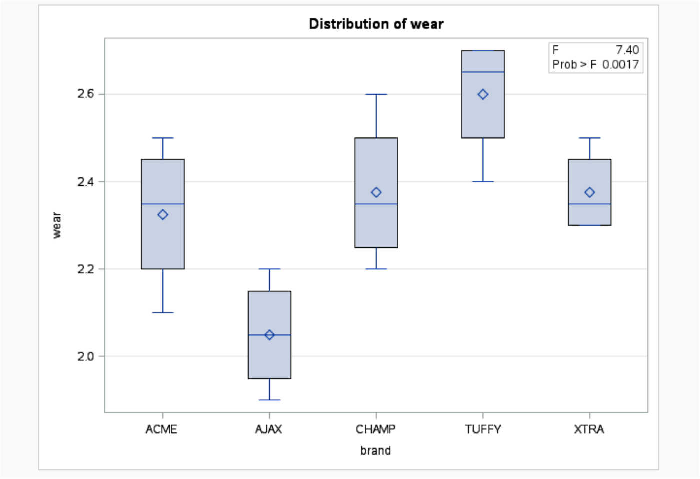
The ANOVA Procedure
Dependent Variable: wear

Source	DF	Sum of Squares	Mean Square	F Value	Pr > F
Model	4	0.61700000	0.15425000	7.40	0.0017
Error	15	0.31250000	0.02083333		
Corrected Total	19	0.92950000			

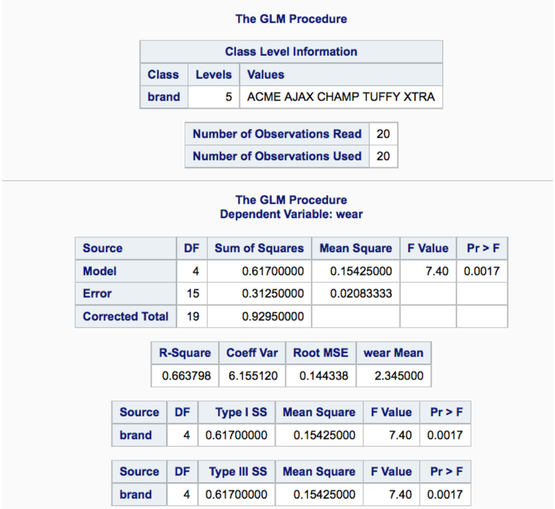
R-Square	Coeff Var	Root MSE	wear Mean
0.663798	6.155120	0.144338	2.345000

Source	DF	Anova SS	Mean Square	F Value	Pr > F
brand	4	0.61700000	0.15425000	7.40	0.0017

PROC ANOVA


Distribution of wear
F 7.40
Prob > F 0.0017
wear
2.0
2.2
2.4
2.6
ACME
AJAX
CHAMP
TUFFY
XTRA
brand

PROC GLM



The GLM Procedure

Class Level Information		
Class	**Levels**	**Values**
brand	5	ACME AJAX CHAMP TUFFY XTRA

Number of Observations Read	20
Number of Observations Used	20

The GLM Procedure
Dependent Variable: wear

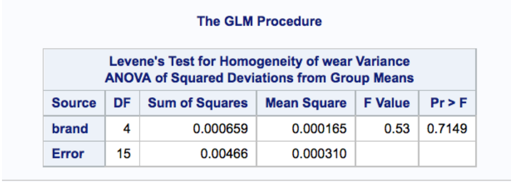
Source	DF	Sum of Squares	Mean Square	F Value	Pr > F
Model	4	0.61700000	0.15425000	7.40	0.0017
Error	15	0.31250000	0.02083333		
Corrected Total	19	0.92950000			

R-Square	Coeff Var	Root MSE	wear Mean
0.663798	6.155120	0.144338	2.345000

Source	DF	Type I SS	Mean Square	F Value	Pr > F
brand	4	0.61700000	0.15425000	7.40	0.0017

Source	DF	Type III SS	Mean Square	F Value	Pr > F
brand	4	0.61700000	0.15425000	7.40	0.0017

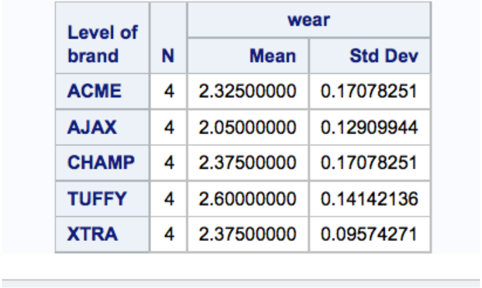
PROC GLM



The GLM Procedure

Levene's Test for Homogeneity of wear Variance
ANOVA of Squared Deviations from Group Means

Source	DF	Sum of Squares	Mean Square	F Value	Pr > F
brand	4	0.000659	0.000165	0.53	0.7149
Error	15	0.00466	0.000310		



Level of brand	N	wear	
		Mean	Std Dev
ACME	4	2.32500000	0.17078251
AJAX	4	2.05000000	0.12909944
CHAMP	4	2.37500000	0.17078251
TUFFY	4	2.60000000	0.14142136
XTRA	4	2.37500000	0.09574271

Comparison of PROC GLM and PROC ANOVA

Basically all of the tables produced by PROC ANOVA are produced by PROC GLM. Both procedures also produce box plots showing the distribution of `wear` values for each brand.

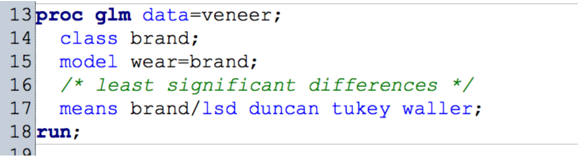
PROC GLM also did a formal significance test for equal variances using Levene's test (due to the HOVTEST option) and gives a table with the means and standard deviations for each group

PROC GLM also refers to type I and type III SS (sums of squares) whereas PROC ANOVA just refers to ANOVA SS. These are all identical for one-way (one factor) balanced ANOVA, but type I and type III are generally different for unbalanced ANOVA.

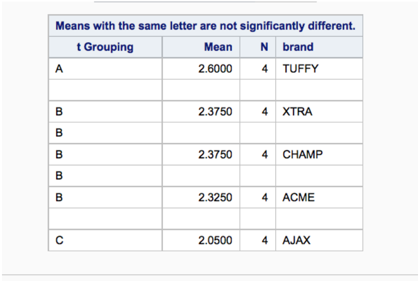
Multiple comparisons in PROC GLM

There are several things you can do in PROC GLM that you can't do in PROC ANOVA, even for balanced data.

PROC GLM: multiple comparisons



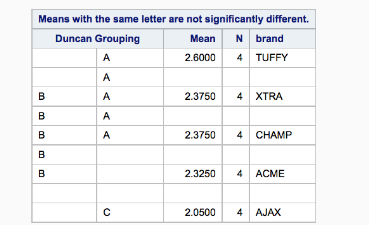
```
proc glm data=veneer;
  class brand;
  model wear=brand;
  /* least significant differences */
  means brand/lsd duncan tukey waller;
run;
```



Means with the same letter are not significantly different.

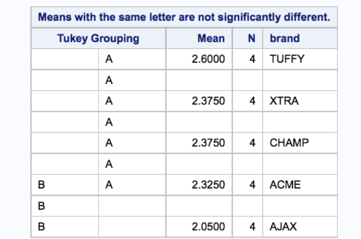
t Grouping	Mean	N	brand
A	2.6000	4	TUFFY
B	2.3750	4	XTRA
B			
B	2.3750	4	CHAMP
B			
B	2.3250	4	ACME
C	2.0500	4	AJAX

PROC GLM: multiple comparisons



Means with the same letter are not significantly different.

Duncan Grouping		Mean	N	brand
	A	2.6000	4	TUFFY
	A			
B	A	2.3750	4	XTRA
B	A			
B	A	2.3750	4	CHAMP
B				
B		2.3250	4	ACME
	C	2.0500	4	AJAX



Means with the same letter are not significantly different.

Tukey Grouping		Mean	N	brand
	A	2.6000	4	TUFFY
	A			
	A	2.3750	4	XTRA
	A			
	A	2.3750	4	CHAMP
	A			
B	A	2.3250	4	ACME
B				
B		2.0500	4	AJAX

PROC GLM: contrasts

Sometimes we want to test hypotheses about some of the effects in a model. The null hypothesis that we are working with is

$$H_0 : \mu_{\text{ACME}} = \mu_{\text{AJAX}} = \mu_{\text{CHAMP}} = \mu_{\text{TUFFY}} = \mu_{\text{XTRA}}$$

We might also test smaller null hypotheses such as

$$H_0 : \mu_{\text{ACME}} = \mu_{\text{AJAX}}$$

or

$$H_0 : \frac{\mu_{\text{ACME}} + \mu_{\text{AJAX}} + \mu_{\text{CHAMP}}}{3} = \frac{\mu_{\text{TUFFY}} + \mu_{\text{XTRA}}}{2}$$

which is equivalent to

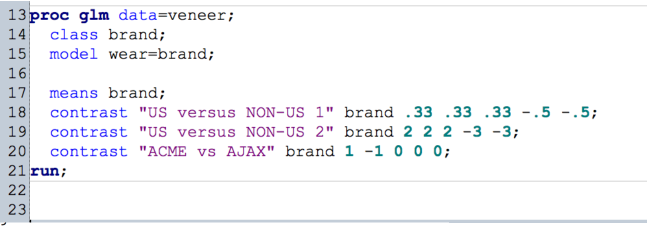
$$H_0 : 2(\mu_{\text{ACME}} + \mu_{\text{AJAX}} + \mu_{\text{CHAMP}}) - 3(\mu_{\text{TUFFY}} + \mu_{\text{XTRA}}) = 0$$

PROC GLM: contrasts

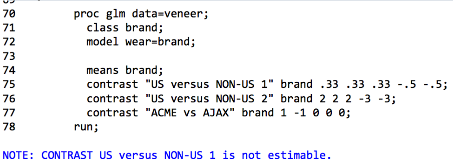
Contrasts are linear combinations of the parameters that are equal to 0. In addition, you can use the ESTIMATE statement to estimate linear combinations of parameters that are estimable (you'll need a linear model theory course for a more in-depth discussion of which functions are estimable, but SAS will not compute something if you ask you to compute a combination that is not estimable).

```
 69
70 proc glm data=veneer;
71 class brand;
72 model wear=brand;
73 means brand;
74 estimate "ACME/AJAX v. CHAMP" brand 1 1 1 0 0 / divisor=3;
75 run;
NOTE: ACME/AJAX v. CHAMP is not estimable.
```

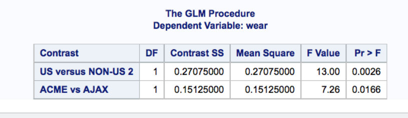
PROC GLM: contrasts



```
proc glm data=veneer;
  class brand;
  model wear=brand;

  means brand;
  contrast "US versus NON-US 1" brand .33 .33 .33 -.5 -.5;
  contrast "US versus NON-US 2" brand 2 2 2 -3 -3;
  contrast "ACME vs AJAX" brand 1 -1 0 0 0;
run;
```



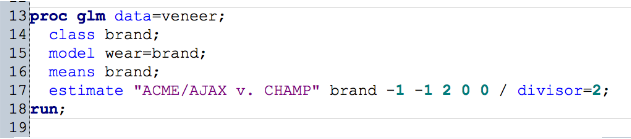
```
70         proc glm data=veneer;
71           class brand;
72           model wear=brand;
73
74           means brand;
75           contrast "US versus NON-US 1" brand .33 .33 .33 -.5 -.5;
76           contrast "US versus NON-US 2" brand 2 2 2 -3 -3;
77           contrast "ACME vs AJAX" brand 1 -1 0 0 0;
78         run;

NOTE: CONTRAST US versus NON-US 1 is not estimable.
79
```



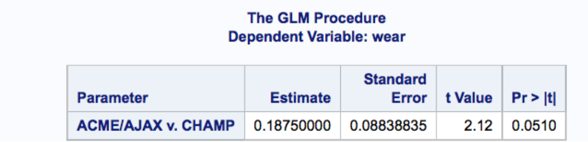
The GLM Procedure
Dependent Variable: wear

Contrast	DF	Contrast SS	Mean Square	F Value	Pr > F
US versus NON-US 2	1	0.27075000	0.27075000	13.00	0.0026
ACME vs AJAX	1	0.15125000	0.15125000	7.26	0.0166

PROC GLM: estimates of linear functions of parameters

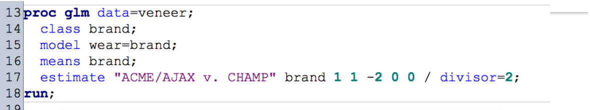


```
proc glm data=veneer;
  class brand;
  model wear=brand;
  means brand;
  estimate "ACME/AJAX v. CHAMP" brand -1 -1 2 0 0 / divisor=2;
run;
```

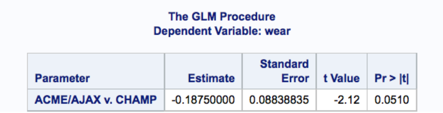


The GLM Procedure
Dependent Variable: wear

Parameter	Estimate	Standard Error	t Value	Pr > |t|
ACME/AJAX v. CHAMP	0.18750000	0.08838835	2.12	0.0510



```
proc glm data=veneer;
  class brand;
  model wear=brand;
  means brand;
  estimate "ACME/AJAX v. CHAMP" brand 1 1 -2 0 0 / divisor=2;
run;
```



The GLM Procedure
Dependent Variable: wear

Parameter	Estimate	Standard Error	t Value	Pr > |t|
ACME/AJAX v. CHAMP	-0.18750000	0.08838835	-2.12	0.0510

Review

Instead of reviewing week by week what we've done, I thought it would be more interesting to review some SAS by looking at interview questions for jobs that require SAS programming.

There are tons of examples to find online, and they can help you find holes in your SAS knowledge. Different kinds of jobs require SAS programming, not all of them for statisticians. As mentioned before, pharmaceuticals employ PhD Statisticians and Master's level SAS programmers who might have different backgrounds, not necessarily statistics. Another major area for SAS is in business analytics, where you might model customer behavior (for example, predicting how many customers will respond to a credit card promotion, what the terms of the promotion should be, etc.)

Interview questions related to SAS

Here are some questions posted to the following website of actual interview questions related to SAS.

`http://www.sconsig.com/tipscons/list_sas_tech_questions.htm`

SAS interview questions

`List differences between SAS PROCs and the SAS DATA STEP.`

SAS interview questions

`List differences between SAS PROCs and the SAS DATA STEP.`

Answer: Procs are sub-routines with a specific purpose in mind and the data step is designed to read in and manipulate data

SAS interview questions

Does SAS do power calculations via a PROC?

SAS interview questions

`Does SAS do power calculations via a PROC?`

Answer. No, but there are macros out there written by other users.

SAS interview questions

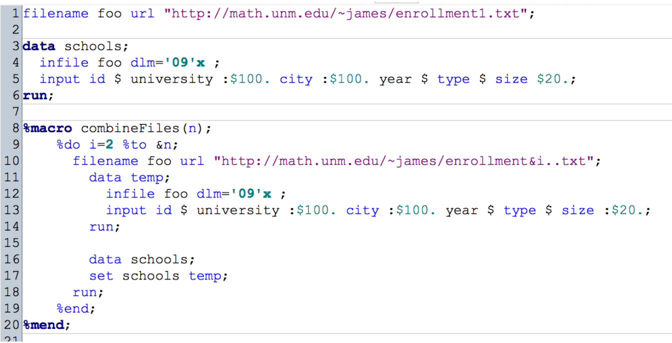
`How can you write a SAS data set to a comma delimited file?`

Answer. PUT (formatted) statement in data step.

But why would you convert a file from say, tab delimited to comma delimited in SAS? Why not just do it in Excel? Usually this would be easier in Excel, but if you had many files, such as our homework with the 60 university files, it would be easier to automate this in SAS where you put a macro over the data step that uses the PUT statement and you convert 60 files from tab delimited to comma delimited by executing one SAS program instead of opening and closing 60 Excel files and using menus to change them to comma-delimited.

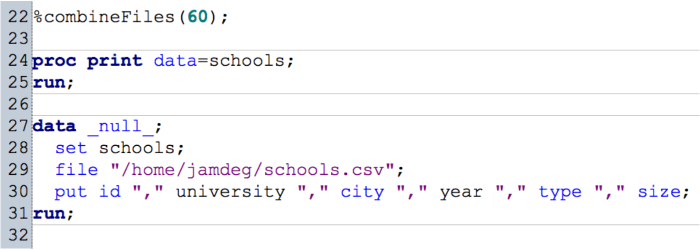
Another reason you might need to do this in SAS is that some files are too big for Excel (this has improved for Excel in recent years, and the practical limits for Excel depend on your system).

Creating a comma-delimited file



```
filename foo url "http://math.unm.edu/~james/enrollment1.txt";

data schools;
  infile foo dlm='09'x ;
  input id $ university :$100. city :$100. year $ type $ size $20.;
run;

%macro combineFiles(n);
    %do i=2 %to &n;
      filename foo url "http://math.unm.edu/~james/enrollment&i..txt";
        data temp;
          infile foo dlm='09'x ;
          input id $ university :$100. city :$100. year $ type $ size :$20.;
        run;

        data schools;
        set schools temp;
      run;
    %end;
%mend;
```

Creating a comma-delimited file



```
%combineFiles(60);

proc print data=schools;
run;

data _null_;
  set schools;
  file "/home/jamdeg/schools.csv";
  put id "," university "," city "," year "," type "," size;
run;

```

Creating a comma-delimited file

For the previous example, I probably should have created a semi-colon delimited file (or some other delimiter), since the original data has commas...

A challenge

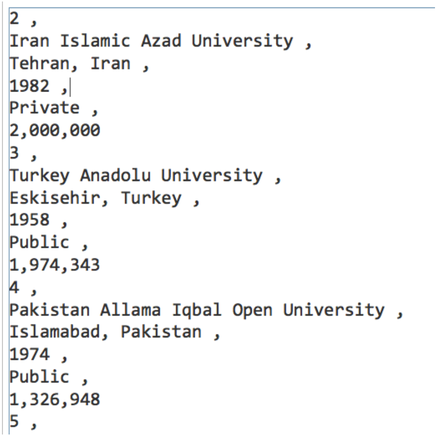
A challenging extension of the previous problem (which I spent some time on, but didn't solve) is to generate a comma-delimited file using arrays instead of specifying all of the variables. If you have a large number of variables, it is a pain to specify them all.

Here is my attempt, which sort of works except that it puts newlines between variables, which I don't really want.

A challenge

```
%macro AllVar;
data _null_;
  set schools;
   file "/home/jamdeg/schools3.csv";
   array vars{*} _all_;
   %do i=1 %to 5;
     put vars{&i} ",";
   %end;
   put vars{6};
run;

%mend;
%AllVar;
```

Here is the result



```
2 ,
Iran Islamic Azad University ,
Tehran, Iran ,
1982 ,
Private ,
2,000,000
3 ,
Turkey Anadolu University ,
Eskisehir, Turkey ,
1958 ,
Public ,
1,974,343
4 ,
Pakistan Allama Iqbal Open University ,
Islamabad, Pakistan ,
1974 ,
Public ,
1,326,948
5 ,
```

A challenge

An alternate solution might be to concatenate all variables (read in as strings) and put string functions in between them. This works fine if you specify all of the variable names, such as

```
newvar = vars{1}||","||vars{2}||","||vars{3}
```

However, it should be possible to generate a loop to do this, but I couldn't get it to work (after trying for an hour or so...)

Converting files from tab-delimited to comma delimited

So this is a good place to review the use of macros and PUT statements. We'll convert the 60 university files from tab-delimited to comma delimited.

SAS interview questions

`Have you ever been to any user conferences? If so, which ones? Do you remember any paper/presentation which stood out? If so, why?`

SAS has a lot of user support groups and conferences that publish articles on how to use SAS. I've used their publications extensively in looking up examples for teaching the class. Often, just googling a particular topic results in a SAS user's group publication.

Many states have their own local user's group, and there are regional groups as well, which usually put on annual conferences. This might be a way to attend a more local conference instead of a large (expensive) national conference.

`http://support.sas.com/usergroups/us-groups.html`

SAS interview questions

`What editor do you use?`
Answers: Emacs - hired on the spot. Kedit, UltarEdit, Xedit, vi, ISPF are also acceptable answers. What I'm looking for is a sense that the candidate can interact with the OS beyond SAS. Of course, [name deleted] is a DM bigot, but he has good reasons for doing so. With that exception, I feel just using DM displays a lack of intellectual curiosity about the rest of the OS which, to me, results in low scores.

I found this amusing since I normally use Emacs at least when on campus, and I also use it for writing R and C code or in manipulating plain text files. When I connect remotely to UNM from home, I find Emacs is sluggish so I use pico, which is much more limited. It is good to be aware of at least one plain text editor (not Word, Wordpad, or Notepad) that will not insert special characters, different styles of newlines, etc. into your files. Some editors such as Emacs have some "intelligence" that helps you to balance parentheses or use proper indentation for some programs.

SAS interview questions

`What is the difference between a FUNCTION and a PROC?`
`Example:  MEAN function and PROC MEANS`

Answer: One will give an average across an observation (a row) and the other will give an average across many observations (a column)

SAS interview questions

What are some of the differences between a WHERE and an IF statement?

SAS interview questions

`What are some of the differences between a WHERE and an IF statement?`

Answer: Major differences include:

- IF can only be used in a DATA step
- Many IF statements can be used in one DATA step
- Must read the record into the program data vector to perform selection with IF
- WHERE can be used in a DATA step as well as a PROC.
- A second WHERE will replace the first unless the option ALSO is used
- Data subset prior to reading record into PDV with WHERE

SAS interview questions

```
How would you check the uniqueness of a data set?  i.e.
that the data set was unique on its primary key (ID).
suppose there were two Identifier variables:  ID1, ID2
```

SAS interview questions

How would you check the uniqueness of a data set? i.e. that the data set was unique on its primary key (ID). suppose there were two Identifier variables: ID1, ID2

Answer.

```
    proc FREQ order = FREQ;
    tables ID;
    shows multiples at top of listing

    Or, compare number of obs of original data set
    and data after
    proc SORT nodups
    or
    proc SORT nodupkey

    use first.ID in data step to write non-unique to
```

SAS interview questions

`Name some of the ways to create a macro variable`

SAS interview questions

Name some of the ways to create a macro variable

Answer.

```
%let
CALL SYMPUT(....)
when creating a macro example:
%macro mymacro(a=,b=);
in SQL using INTO
```

SAS interview questions

Here are some questions suggested by SAS itself:

- What is the significance of January 1st, 1960?
- Which SAS Procedures (PROCs) do you use and for what purpose?
- What is your preferred reporting tool? Why? How does (insert preferred tool) differ from (insert other tool)?
- List the SAS functions you generally use and how you use them.
- Describe your familiarity with SAS Formats/Informats.
    - a. Do you create formats? Yes, How?
    - b. What other uses of formats are there?
- Describe your familiarity with SAS macros and the macro language.
- What SAS system options are you familiar with and what for what purpose do you use them?
- Describe your familiarity with SAS/INTRNET.
- What have you heard about version 9 that you are looking forward to?
- For what purposes do you use DATA _NULL_?

http://www.globalstatements.com/sas/jobs/technicalinterview.html

A SAS technical interview typically starts with a few of the key concepts that are essential in SAS programming. These questions are intended to separate those who have actual substantive experience with SAS from those who have used in only a very limited or superficial way. If you have spent more than a hundred hours reading and writing SAS programs, it is safe to assume that you are familiar with topics such as these:

- SORT procedure
- Data step logic
- KEEP=, DROP= dataset options
- Missing values
- Reset to missing, or the RETAIN statement
- Log
- Data types
- FORMAT procedure for creating value formats
- IN= dataset option

Additional questions

- When is it useful to use a RETAIN statement?
- When is it useful to use PROC TRANSPOSE?
- How does SAS represent missing values? What is the difference (in terms of representation) between a missing character value versus a missing numeric value?
- Suppose you had a dataset with 100 columns. How could you get SAS to alphabetize the variable names so that they show up in alphabetical order?
- How would you read in a file one character at a time? (Note that this could be useful for removing bad characters from a file)