

Stat-340 – Assignment 2 – 2015 Spring Term

Part 1 - Breakfast cereals - Easy

In this part of the assignment, you will learn how to:

- importing a *.csv file using *Proc Import*;
- use Proc Tabulate;
- create dot, box, and notched-box plots in SGplot;
- improve plots through jittering;
- do a single factor CRD ANOVA using GLM;
- extract estimated marginal means (LSMEANS) from GLM and make a suitable plot.

We will start again with the cereal dataset.

1. Read the information about the dataset and breakfast cereals from
<http://www.stat.sfu.ca/~cschwarz/Stat-340/Assignments/Cereal/Cereal-description.pdf>
2. Download the cereal dataset from
<http://www.stat.sfu.ca/~cschwarz/Stat-340/Assignments/Cereal/cereal.csv>
and save it to your computer in an appropriate directory.
3. In many cases, *.csv files include the variable names in the first row of the data file. *SAS* has procedure *Proc Import* which can read both the variable names and data values, but at the loss of fine control in the operation.

Your code will look something like:

```
proc import file="cereal.csv" dbms=csv out=cereal replace;
run;
```

Note that if you are using the *SAS* University Editions, you will have to specify the *Path* variable and enclose the file name in DOUBLE quotes.

What does the *dbms=* option do? What does the *replace* option do?

4. Deal with the coding for missing values as was done in Assignment 1.
5. Manufacturers pay lots of money to supermarkets to place their product to increase sales. See http://en.wikipedia.org/wiki/Slotting_fee for details. For example, product placement charges are higher for shelves at eye-level rather than on the bottom level.

The variable *shelf* indicates the shelf (from the floor) where the product was placed. Notice that *shelf* is an **Ordinal** scaled categorical variable and should not be treated in the same way as, for example, grams of fat. While a shelf with value of 2 indicates that the shelf is higher than a shelf with the value 1, it does NOT mean that it is TWICE as high. Similarly, a shelf with a value of 3 indicates that it is higher than a shelf with a value of 1, but does not indicate that the shelf is 3× higher. It is important to examine data carefully to ensure that you understand if a variable is a nominal or ordinal scaled categorical variable or an interval or ratio scaled variable. Refer to Section 2.5 of my course notes for more details

The analyses involving nominal or ordinal scaled variables is DIFFERENT than the analyses using interval or ratio scaled variables. This is a VERY COMMON error made when using statistical packages.

A general rule of thumb is that nominal or ordinal variables should be coded using alphanumeric codes rather than numeric codes. For example, code the value of *Sex* as *m* or *f* rather than 0 or 1. In this dataset, the *display_shelf* variable should be coded using *low*, *mid*, *top* rather than 1, 2, and 3. In this way, you won't accidentally do a regression of calories vs. *low*, *mid*, *top* which makes no sense.

In the rest of this question, we will examine if there is evidence that the the average amount of sugar per serving varies by shelf height?

6. Use *Proc Tabulate* to get a summary table of the number of cereals by shelf, the mean amount of sugar per serving, and the standard deviation of the amount of sugar per serving.

Your code will look something like:

```
proc tabulate data=****;
  title2 'basic statistics';
  class shelf;
  var ****;
```

```
table shelf='Shelf', ***** (n*f=5.0 mean*f=7.1 std*f=7.1);
run;
```

where you need to replace the `*****` with appropriate code.

What does `*f=7.1` (and similar) do in *Proc Tabulate*? What does `= 'Shelf'` do in the *tables* statement? What is the function of the *class* and *var* statements?

What do you notice about the sample standard deviation of the amount of sugar per serving across the shelf heights? Most linear model procedures (such as regression or ANOVA) assume that the population standard deviations are equal across all treatment groups. Based on these summary statistics, is this a reasonable assumption? Why? At which point would you be concerned? Hint: refer to Section 5.3 in <http://www.stat.sfu.ca/~cschwarz/CourseNotes>.

The sample sizes in the three shelves is not equal – we say that the design is unbalanced. Normally, most computer packages can deal with unbalance rather easily. However, this will have an influence on the estimated standard errors (and confidence intervals) for the mean of each group. Can you predict the effect of the unequal sample size on the standard errors and confidence intervals for the mean number of calories per serving for each of the shelves? (A good exam question.)

7. Create a dot plot of sugars per serving vs shelf using *SGplot*. What variable should go on the Y-axis? What variable should go on the X-axis? Your code will look something like:

```
Proc SGplot data=****;
  title2 'basic plot WITHOUT jittering';
  scatter x=**** y=****;
  axis label='****' offsetmax=0.10 offsetmin=0.10;
  yaxis label="****";
run;
```

What does the 0.10 in the *offsetmax* and *offsetmin* clause do? Why do you want to do this for the plot? [Hint: do the plots with and without the offset arguments and see how the plot changes.] In general, try and avoid having data values plot at the edges of plots.

There are over 70 breakfast cereals in the dataset, but only about 30 dots are displayed. Why?

8. Note that the dot-plot is not totally satisfactory because many of the values are overprinted on top of each other. It is often recommended that you jitter the points by adding some (small) random noise before plotting. Of course you would still use the original data in any analysis. See <http://blogs.sas.com/content/iml/2011/07/05/jittering-to-prevent-overplotting-in-statistical-> for a discussion of jittering.

In SAS 9.4, the *SCATTER* statement has the *JITTER* option that will jitter the data for plotting purposes (it is possible to control the amount of jittering, but this is beyond this course). If you are using SAS 9.4, replot the above data using that option.

In SAS 9.3 (and other packages), you need to do the jittering yourself. Here is a code fragment that jitters the points – you may have to change the scaling factor applied to the random number to get a nice looking plot that isn't too jittered. You should also change the value of the seed (the 123434 value) used to initialize the random number generator. **Create new variables that contain the jittered values because you want to keep the original data for analyses.** Then create the dot plot using this new dataset.

```
data plotdata; /* add some noise for plotting */
  set new_cereal;
  jsugars = sugars + 2 *(ranuni(123434)-.5); /* add some random noise */
  jshelf = shelf + .1 *(ranuni(234323)-.5);
run;
```

Read the help file on the *ranuni()* function. What does the *ranuni()* function do? What does the 123434 in the *ranuni()* do? You should use a different value for your seed. Why? What does subtracting .5 do? What does multiplying by .1 or 2 do?

Redo the dot plot using the *jittered* values – don't forget to use the *plotdata* dataset because that is where you created the jittered values.

The dot plot is much more satisfactory. Why? Are there any obvious outliers?

9. Now that you've checked the data for outliers using the dot plots, use *SGplot* to create VERTICAL side-by-side BOX-plots of the data USING THE ORIGINAL DATA. See http://en.wikipedia.org/wiki/Box_plot or <http://web.pdx.edu/~stipakb/download/PA551/boxplot.html> in case you have forgotten what a box-plot shows. Unfortunately, you can't (easily yet) overlay the box plots on the original data but <http://www.scsug.org/wp-content/uploads/2012/11/SCSUG-2012-ScatterBox.pdf> shows how you could do this – this is beyond the scope of this course, and you are not expected to do this for the assignment.

The *VBOX* statement creates the vertical box plots. Your code will look something like:

```
Proc SGplot data=****;
  title2 'Vertical box-plots WITHOUT notches;
  vbox **** / category=**** ;
  xaxis label='****' offsetmax=0.10 offsetmin=0.10;
  yaxis label="****";
run;
```

Don't forget to use the original data (not the jittered data) when creating the box-plot.

What does the *category* option on the *vbox* statement do?

What do you conclude based on the box-plots. Hints – box plots ONLY show information about INDIVIDUAL data points and say nothing about the MEAN response.

10. Box-plots are often improved by adding NOTCHES to the plot. Read <http://www.cms.murdoch.edu.au/areas/maths/statsnotes/samplestats/boxplotmore.html> for details about the notches.

Repeat the side-by-side box plot and add the *NOTCHES* option using the code fragment:

```
Proc SGplot data=****;  
    title2 'Vertical box-plots WITH notches;  
    vbox **** / category=**** notches ;  
    xaxis label='****' offsetmax=0.10 offsetmin=0.10;  
    yaxis label="****";  
run;
```

Some of the notches look odd because the notch extends past the 25th or 75th percentiles. DO YOU UNDERSTAND what is happening?

What can you conclude based on the notched side-by-side box plot? CAREFUL, the information presented by the notches is VERY DIFFERENT than the information from the box-plot as a whole. BE SURE YOU UNDERSTAND the difference (this is a popular exam question).

11. We will now do a formal hypothesis test that the MEAN amount of sugars/serving is the same across the three shelves. What is the null and alternate hypotheses in words and symbols. (a good exam question).

PROC GLM is often used to test hypotheses about MEANS (using ANOVA) in simple experimental designs (such as a completely randomized design (CRD)). For more complex experimental designs you would use *PROC MIXED* - again beyond the scope of this course, but check my CourseNotes. Section 5.10 in <http://www.stat.sfu.ca/~cschwarz/CourseNotes> has an example of using *Proc GLM*.

Your code will look something like:

```
proc glm data=new_cereal PLOTS=(DIAGNOSTICS RESIDUALS);  
    title2 'does shelf height affect MEAN amount of sugar per serving';  
    class shelf;  
    model sugars = shelf;  
    lsmeans shelf / diff cl adjust=tukey lines;  
    ods output LSmeanCL = mymeansCL;  
run;
```

The *CLASS* statement specifies that shelf is a (nominal or ordinal) categorical variable. The *MODEL* statement specifies the *Y* variable (to the left of the = sign) and the factors (to the right of the = sign). This procedure is assuming that the data have been collected using a simple random sample or a completely randomized design – for more complex designs, you need to use *PROC MIXED*. Do you understand the features of a CRD design or SRS survey and how these relate to this dataset? [Another good exam question.]

Find the F-value for the Type III hypothesis test (You will always use the Type III tests – this will be explained in Stat-350). What is the *p*-value? What do you conclude about the MEAN sugar/serving over the three shelf locations? Note that hypotheses and conclusions are ALWAYS about the MEAN of the response variable. (Another good exam question.)

Following the omnibus hypothesis test (??? do you understand what I just said??), you need to do followup examinations to further investigate where the differences in group means may lie. As outlined in Section 5.13 of my CourseNotes, you should always do a multiple-comparison after ANY ANOVA and the Tukey-Kramer adjustment is the most common multiple comparison procedure used.

The *LSMEANS* statement in *Proc GLM* performs a multiple-comparison after the ANOVA. Find its output? [Hint, there are two sections, but the section labelled *Comparison lines* is easiest to interpret. What do you conclude based on the output from the *LSMEANS* statement? CAREFUL – remember that hypotheses are about the MEAN response.

We would like to extract the output from the *LSMEANS* statement to make a nice table and/or plot. As usual, we use the *ODS OUTPUT* statement to create an output dataset with the (partial) results from the *LSMeans* statement.

12. Print out the the *mymeansCL* dataset using a code fragment.

```
proc print data=mymeansCL;
    title2 'raw table';
run;
```

What information is presented? Can you match this from the output from the *GLM* procedure?

13. Make a nicer table for inclusion in the report by changing the number of decimal places displayed using *FORMAT* statements, adding better labels, and changing the order of the presentation. Your code will look something like:

```
proc print data=mymeansCL label split=' ';
    title2 'nicer table';
```

```

var **** **** **** **** ****;
format **** 7.1;
label ****="A nicer label";
attrib **** label="A nicer label" format=7.1; /* label and format in one statement */
run;

```

You will likely want to dump this table out to an RTF file for use in the write-up later. [Hint: use ODS RTF]

14. Use *SGplot* to plot the estimated population means and lower and upper 95% confidence intervals for the population mean in much the same way you did in Assignment 1 for the odds ratio in the Titanic example. Your code fragment will look something like:

```

proc sgplot data=mymeansCL noautolegend;
  title2 'Estimated mean amount of sugar/serving by shelf location along with 95% ci f
  scatter x=shelf y=****;
  highlow x=shelf low=**** high=****;
  xaxis label='Shelf';
  yaxis label='Sugar/serving (g) - mean and 95% ci';
run;

```

What do you conclude from your plot? Is your conclusion consistent with the results from the multiple comparison procedure you ran earlier?

You will likely want to dump this graph out to an RTF file for use in the write-up later.

Hand in the following using the electronic assignment submission system:

- Your *SAS* code that did the above analysis.
- A PDF file containing all of your *SAS* output.
- A one page (maximum) double spaced PDF file containing a short write up on this analysis explaining the results of this analysis suitable for a manager who has had one course in statistics.

I strongly urge you read a similar report at <http://people.stat.sfu.ca/~cschwarz/Stat-300/Assignments/Assign01/assign01-cereal-writeup.pdf> to see you to structure your report! [Obviously don't copy word-for-word from this example!] Your results differ because your cereal data set is different.

You may find the other assignments in my Stat-300 course to be helpful for Stat-340 (hint, hint, hint).

Your handing should include the following:

-
- A (very) brief description of the dataset.
 - The results from the hypothesis testing about the amount of sugars/serving as a function of the shelf. It is not necessary to include the ANOVA table – just give the F-statistic, p-value, and your interpretation.
 - The estimated mean amount of sugar/serving for each shelf along with a 95% confidence interval for the population mean in a nicely formatted table. Be sure to ADD a legend to the table (you can't do this in *SAS*). You will need a table number and need refer to the table number in the text (see my sample write up above). Notice that the placement of the legend differs between figures and tables (see the example).
 - The plot of the mean amount of sugar/serving by shelf with the 95% confidence intervals. Normally you would not give both the table and the plot, but I want you to get practice in generating both types of output. Again, your figure needs a legend which you can't add in *SAS*. The placement of the legend differs for figures and legends.

You can adjust the size of the plot in your word-processor. To get the plot and figure side-by-side, you should create a 1×2 table in your document and then paste in the table or figure from the RTF file and adjust the size of the pasted table and/or figure.
 - A conclusion about the difference in mean amount of sugar/shelf across the 3 shelves, i.e. which shelf seems to have the highest mean amount of sugar per serving; which shelves seem to have the same mean amount of sugars/serving?

You will likely find it easiest to do the write up in a word processor and then print the result to a PDF file for submission. Pay careful attention to things like number of decimal places reported and don't just dump computer output into the report without thinking about what you want.

Part 2 - Road Accidents with Injury - Intermediate

In this assignment, we will examine the circumstances of personal injury road accidents in Great Britain in 2010. The statistics relate only to personal injury accidents on public roads that are reported to the police, and subsequently recorded, Information on damage-only accidents, with no human casualties or accidents on private roads or car parks are not included in this data.

Very few, if any, fatal accidents do not become known to the police although it is known that a considerable proportion of non-fatal injury accidents are not reported to the police.

In this part of the assignment, you will learn how to:

- import dates and times;
- format dates and time for display purposes;
- extract features of dates and times, such as the year, month, day, hour, minute;
- create summaries at a grosser level using *Proc Means* and related procedures;
- create histograms with a kernel smoother using SGplot;
- see the impact of data heaping.

This data was extracted from <http://www.data.gov.uk/dataset/road-accidents-safety-data> and is available at <http://www.stat.sfu.ca/~cschwarz/Stat-340/Assignments/Accidents/road-accidents-2010.csv>. A data guide is also available at: <http://www.stat.sfu.ca/~cschwarz/Stat-340/Assignments/Accidents/road-accidents-2010-data-guide.xls>.

1. Download the *.csv file into your directory. Also download the data guide. What convention is used for missing values (Hint: read the Introduction tab in the data guide)?

Open the *.csv file and have a look at the data values. Notice that the dates are in day/month/year format with varying number of digits for the year (but you can assume that they are all in 2010).

There are over 150,000 records, so manually scanning the entire data is simply not feasible, nor desirable.

-
2. Read the data into *SAS* using an input statements in a data step. Do NOT use *Proc Import* because I want you to get practice in reading date and time variables. Hints:

- Use the DLM, DSD, FIRSTOBS clauses on the INFILE statement as needed.
- The first column, accident id, should be defined as a character variable.
- Use the ddmmyy10. and hhmmss10. in-formats to read the date and time variables. Your code will look something like:

```
data accidents;
  infile .... dlm=... dsd firstobs=....;
  length AccidentId $20;
  input AccidentID ... Accident_Severity ... AccidentDate:ddmmyy10.
        AccidentTime:hhmmss10. ...;
  attrib AccidentDate label='Accident Date' format=yyymmdd10.;
  attrib AccidentTime label='Accident Time' format=hhmm5.;
```

You will have to specify all of the variable names on the *input* statement. Hint: Copy and paste the first line of the data file that contains all of the variable names. What does the colon (:) format modifier do?

- *SAS* converts dates and time into internal values similar to how Excel stores dates and time. You need to format these values for display using the *ATTRIB* statement and a suitable out-format. What does the trailing 10 or 5 specify in the in-formats and out-formats? I recommend that dates ALWAYS be displayed in yyyy-mm-dd formats and times in 24-hr formats as there is no ambiguity about what the displayed values actually mean. For example 2015-02-03 is always interpreted as 3 February 2015, but 2/3/2015 or 3/2/2015 are BOTH commonly used in North America for this same date!
 - Note the use of CamelCase (<http://en.wikipedia.org/wiki/CamelCase>) to make variables easier to read. Note that *SAS* is NOT case sensitive (unlike many other programming languages), so the variable *camel case*, *CamelCase* and *CAMELCASE* all refer to the same variable and ALL three ways of referring to this variable can be used in the same program.
3. Print out the first 10 records. Check that the dates and times have been read properly and are displaying properly. Compare your printed data records with the actual file to verify that everything is ok.
4. **ALWAYS CHECK YOUR DATA FOR CODING ERRORS AND OTHER PROBLEMS!**

There are over 150,000 records, so manually scanning the entire data is simply not feasible, nor desirable. *Proc Tabulate* is your friend! Use *Proc Tabulate* to make a summary of the code values for several of the variables.

Your code will look something like:

```
proc tabulate data=accidents missing;
  class Accident_Severity v2 v3 v4 v5;
  table Accident_Severity ALL, n*f=7.0;
  table v2 ALL, n*f=7.0;
  ...
run;
```

Compare the codes that actually exist in the dataset to the data guide to see if any unusual codes exist representing missing value or if there are any outliers.

What does the *ALL* option do in the *table* statement?

As always, change any numeric/character codes representing missing values to actual missing values (as you did in Assignment 1 in the cereal dataset). Notice that because different variables have different codes for missing values, you CANNOT use the *array* method as was done in Assignment 1.

CAUTION: Look at the DataGuide for the *Road_Type* variable. What should you do about code 9?

5. Redo the *Proc Tabulate* to ensure that the coding has been done properly.

What does the *missing* option on the *Proc Tabulate* statement do? Hint: compare the tables with and without this options when the *class* variable has some missing values.

6. We will begin by looking at the number of (reported) daily accidents with personal injury by the day of the year and see if the average number of daily accidents varies by month.

A very common data summary method is to use *Proc Means* to compute summary statistics for subgroups and then plot these summary statistics in various ways to get a feel for your data. You will often see code fragments similar to below.

Use *Proc Means* to “count” the number of accidents by day of the year. Your code will look something like:

```
proc sort data=accidents; by accidentdate; run;
proc means data=accidents noprint;
  by accidentdate;
  var accidentdate;
  output out=dailysummary n=naccidents;
run;
```

Don't forget to sort the data prior to summarizing it. What does the *noprint* option do on the *Proc Means* statement? Why do you want to use this option? What does the *output* statement do:

You should print out the first 10 records of the the daily summary to see the names of the variables created by your *Proc Means*.

7. Plot the number of accidents by date using *SGplot*.

```
proc sgplot data=dailysummary;  
    title2 'number of accidents by date';  
    scatter y=**** x=AccidentDate;  
run;
```

Note that you will be using the daily summary dataset created above, and not the original data for the next few steps.

What is the general impression given by the plot? When is the number of accidents generally lowest and when are they highest?

8. You can improve the scatter plot by adding a smooth curve to the plot. One such smoother is the loess curve which is like a type of running average. The loess curve is a smoother that isn't restricted to straight lines. A brief discussion of loess smoothing is available at http://en.wikipedia.org/wiki/Local_regression.

A loess curve can be added using *Proc SGplot* using the *loess* statement. Read the help file and add the loess curve and create a second plot with the the points plotted and the loess curve also added.

What does the loess curve seem to show? Are you surprised?

Some of the days with unusually high number of accidents occurred in November 2010? Any guesses as to why?

9. I now want to compare the average number of daily accidents with personal injury across the 12 months. I need to "extract" the month from the *AccidentDate* variable.

Your code fragment will look something like:

```
data dailysummary;  
    set dailysummary;  
    month = month(*****);  
    day    = day(*****);  
run;  
proc print data=dailysummary(obs=10);  
run;
```

What is the purpose of the *month()* and *day()* functions?

-
10. Use *Proc Tabulate* to create a summary of the number of days, the mean accidents/day, and the std deviation of the accidents per day for each month in a similar fashion as you did in Part 1 of this assignment. Your code will look something like:

```
proc tabulate data=dailysummary;
  title2 'what is mean and std dev of daily accidents for each month';
  class ****;
  var naccidents;
  table ****, naccidents*(n*f=5.0 mean***** std*****);
run;
```

Think of what you want the table to look like to figure out what should be replaced in the **** (and similar text strings) above.

Notice how you can request multiple statistics from a single analysis variable.

Again check the std. deviations to see if the assumption of equal population standard deviations is tenable.

11. Use *Proc GLM* to test the hypothesis that the number of accidents/day is the same across all months. This will be done in a similar fashion as in Part 1.

```
proc glm data=**** PLOTS=(DIAGNOSTICS RESIDUALS);
  title2 'is there a difference in the mean number of accidents by month?';
  class ****;
  model **** = ****;
  lsmeans ****/ diff cl adjust=tukey lines;
  ods output lsmeanc1 = mymeansc1;
run;
```

What do you conclude from the omnibus test of the equality of the MEAN number of accidents/day?

12. Use the LSMEANS to again extract the estimated mean number of accidents/day along with confidence limits and plot these values, as you did in Part 1 of this assignment. What do you conclude based on the output from GLM and the final scatter plot?

Send the plot of the mean number accidents to an RTF file for the report.

13. Now we will examine the number of accidents by hour of day and minute of the hour. Go back to the ORIGINAL data file, and create a derived variable for the hour and minute of the accidents. Your code fragment will look something like:

```
data accidents;
  set accidents;
  hour   = hour(*****);
  minute = minute(*****);
run;
```

What do the *hour()* and *minute()* functions do?

14. Print the first 10 records to verify that you have extracted the hour and minute correctly.
15. Make a histogram of times of accidents by hour of the day and superimpose a kernel-density estimate. Your code fragment will look something like:

```
proc sgplot data=accidents;
  title2 'histogram of hour of accidents';
  histogram hour / binwidth=1;
  density hour /type=kernel;
run;
```

What does the *binwidth* option do? What happens if the *binwidth* isn't specified – why does the plot look strange? Hint: what is the area under a pdf supposed to add up to? (In most cases, you can just let *SGplot* choose the bin widths for histograms, but sometimes you need to intervene.)

What does the distribution of accidents by the hour of the day indicate. Send the plot of the distribution of accidents by the hour of the day to an RTF file for the report.

16. Make a histogram of the minute of the hour for accidents using similar code. Set the bandwidth to again be 1 unit.
When is the most dangerous minute of an hour to be on the roads, as indicated by the histogram. Do you believe this? Why has this happened?
Why does the histogram have this odd shape?

Hand in the following using the online submission system:

- Your *SAS* code.
- A PDF file containing the the output from your *SAS* program.
- A one page (maximum) double spaced PDF file containing a short write up on this analysis suitable for a manager of traffic operations who has had one course in statistics. You should include:
 - A (very) brief description of the dataset.

-
- A description of the number of accidents per day as it varies over the month. In which months will you need to deploy more resources to deal with large number of accidents. A graph may be helpful.
 - At which hours of the day do accidents tend to occur. Use the plot to illustrate your argument. Are you surprised?

Part 3 - Cigarette Butts in Bird Nests - Challenging

Download and read the following paper where the effect of cigarette butts on the parasitic load in bird nests was investigated.

Suarez-Rodriguez, M., Lopez-Rull, I. and Garcia, C.M. (2013).
Incorporation of cigarette butts into nests reduces nest ectoparasite
load in urban birds: new ingredients for an old recipe?
Biological Letters 9, 20120931.
<http://dx.doi.org/10.1098/rsbl.2012.0931>

The authors have provided the raw data as an electronic supplement to this paper. I've downloaded the data and you can access it from <http://www.stat.sfu.ca/~cschwarz/Stat-340/Assignments/BirdButts/nests.xls>. This is an Excel workbook with TWO worksheets corresponding to the analyses in Figure 1 and Figure 2.

We will try and reproduce some of the results from this paper. You will find, rather surprisingly, that some of the results in the published paper are INCORRECT This is NOT an unusual occurrence in scientific paper – often authors use inappropriate statistical methods to analyze their data and this is not noticed in the peer review process.

In this part of the assignment, you will learn how to:

- use *Proc Import* to read Excel spreadsheets directly;
- use *Proc Tabulate* to examine standard deviations and balance in experimental designs;
- create categorical derived variables based on data in an experiment;
- estimate binomial proportions (and standard errors) using *Proc Freq* and create suitable plots;
- use *Proc Ttest*;
- use *Proc Reg* to perform a regression on the logarithmic scale and how to back-transform the results;
- extract information from the *Procs* for printing and plotting.

Let us begin.

1. Use *Proc Import* to import the data directly from the workbook. This requires that the worksheets have the variable names in the first row. Use code similar to:

```
proc import file="next.xls" dbms=xls out=nestinfo replace;
  sheet = 'Correlational';
  guessingrows = 9999;
run;
```

Note that if you are using the *SAS University Editions*, you will have to specify the *Path* variable and enclose the file name in DOUBLE quotes.

What is the difference between *dbms=xls* and *dbms=csv*. Why is *dbms=xls* used here?

What does the *sheet* statement do? What does the *guessingrows=9999* statement do? Why are these good programming practises?

2. Examine the log file to see how *SAS* creates the variable names. Print out the first 10 records to ensure that the data has been read properly.
3. The 3 R's (randomization, replication, blocking/stratification) are extremely important when running a survey or experiment.¹ How do the 3 R's apply to this data, i.e. what randomization scheme was used, how did they choose these sample sizes, and is there blocking/stratification?
4. There are two factors in this first study – host species and nest content. Use *Proc Tabulate* to create a simple summary table comparing the number of nests measured by host species and nest content. The code will look something like:

```
proc tabulate data=nestinfo;
  title2 'Summary of number of nests by species and nest content';
  class species nestcontent;
  table nestcontent, species*n*f=5.0;
run;
```

Is the design balanced? Will this be of concern in later analyzes?

¹Read Section 2.1 at <http://www.stat.sfu.ca/~cschwarz/CourseNotes> if you have forgotten the 3 R's.

-
5. The paper states: “Cellulose from cigarette butts was present in 89.29 per cent ...”. Reproduce this result using *SAS* and get a standard error for these values.

Create a derived variable for the present/absence of butts in a nest based on butt weight using a code fragment

```
data nestinfo;
  set nestinfo; /* access the previous dataset */
  length ButtsPresent $4;
  buttspresent = 'no';
  if ***** then buttspresent = 'yes';
run;
```

Why does this give presence/absence?

6. Print out the first 10 records after you created this derived variable to check that it has been done properly.
7. Use *Proc Freq* to generate a table showing the number and proportion present FOR EACH individual species, i.e. BY SPECIES.

Your code fragment will look something like:

```
Proc Freq data=nestinfo;
  by *****;
  table ButtsPresent / binomial(level='yes');
  output out=nestprop2 binomial;
  ods output BinomialProp=nestprop;
run;
```

Don't forget to sort the data by species before using a *By* statement.

You want the proportion of nests with butts present, but *SAS* may compute the proportion of nests WITHOUT butts present. Refer to the hints email from assignment 1 on how to fix this problem.

In *Proc Freq*, the LEVELS options on the BINOMIAL option of the TABLES statement allows you specify the level of the factor for which the binomial proportion is computed.

Some procedures have multiple ways to send information from the procedure to dataset. In *Proc Freq* the *OUTPUT* statement and the *ODS OUTPUT* statements can both be used (as seen above). Print out both created datasets to see that the same information is available in both datasets but in different formats. For example, one method gives you the information on one line per species, while the other method gives you several lines per species. Sometimes, one method creates a dataset that is more convenient for further processing than the other method.

-
8. Create suitable plot showing for each species, the estimated proportion of nests with butts along with the 95% confidence intervals. Hint: Use *Proc SGplot* and the *HighLow* statement as you did in the previous assignment. What variables do you want to plot? [Hint: Which variable shows the proportion of butts present, and which variables how the upper and lower bound of the confidence interval?]

The paper just reported the naked estimate with far too many significant figures. Why are naked estimates a bad thing to do; what would be a suitable number of significant figures to use for percentages and proportion.

What do you conclude about the relative proportions of nests with cigarette butts in the two species based on the confidence intervals. What assumptions are you making about the sampling plan when you compute a standard error and confidence intervals. What assumptions are you making about the other factor (nest content) when you ignore it during the analysis?

9. Conduct a formal test that the proportion of nests with butts is the same across both host species using *Proc Freq* using a code fragment similar to:

```
proc freq data=nestinfo;
    title2 'chi square test for equal proportions';
    table ****x*buttspresent / chisq nocol nopercnt;
run;
```

Notice you DO NOT USE A *BY* statement here. Why?

What is the hypothesis of interest? What is the p-value. Interpret the p-value. What p-value is reported in the paper? Is your results consistent with the paper's result?

One of the marks of a good statistician is that you turn off extraneous output in the contingency table that is created using the *nocol*, *norow*, or *nopercent* options in the *TABLES* statement.

SAS dumps out several test statistics and p-values (a case of computer diarrhea). Which test statistic and which p-value is appropriate for this hypothesis? Why?

10. The paper also reports "... and weighed on average 2.45 ± 3.34 g (range 0 - 11.75) and 3.06 ± 4.15 g (range 0 - 14.86) in HOSP and HOFI nests, respectively...'. '

Use *Proc Univariate* to find the mean butt weight in nest of each species. Hint: use a *BY* statement. To get confidence intervals for the mean, you will need to specify the *CIBASIC* option on the *Proc Univariate* statement. Your code fragment will look something like:

```
proc univariate data=nestinfo cibasic;
    by ****;
```

```

var ****;
ods output basicintervals=mycibuttweight;
run;

```

The *ODS OUTPUT* statement extracts the estimated mean and the 95% confidence interval. The *basicintervals* is the ods table name. Print out the dataset and you will see that it contains too much information (e.g. it contains information on confidence intervals for the variance which is not of interest).

11. Similar to what you did in Assignment 1 with the Titanic example, you need to extract only those rows that contain information about the mean number of butts. Use a *Data* step to extract the relevant information. For example, to select records that only contains record from females, the code might look like:

```

data femaleonly;
  set bothsexes;
  if sex = 'f';
run;

```

Notice the form of the selecting *if* statement. In general, use the *where* statements to select observations WITHIN procedure but leave the dataset unchanged; use a selection *if* statement to create a NEW dataset with a subset of records. More information is available at <http://support.sas.com/kb/24/286.html>.

12. Create a plot of the confidence interval for the mean butt weight by species in a similar way as in previous questions.

Do your results match those reported in the paper? What do you conclude based on the plot you created?

13. In the previous two questions, we conducted an ANOVA to compare the MEAN response across groups. If there are only two groups, a special case of ANOVA is the two-sample t-test.

You guessed it, *Proc Ttest* is the procedure to use in this case. Your code will look something like:

```

proc ttest data=nestinfo;
  title2 'comparison of mean butt weights';
  class ****;
  var ****;
run;

```

As in *GLM*, the *CLASS* statement defines the factors (groups) whose means are to be compared. The *VAR* statement defines the response *Y* variable.

What is the null and alternate hypothesis? What is the test statistic? What is the p-value. Interpret the p-value.

Note that *SAS* give you both the equal- and unequal-variance t-test. The latter is also known as the Welch-test or the Satterthwaite-test. Which should be used in this case? Refer to

Ruxton, G.D. (2006).

The unequal variance *t*-test is an underused alternative to Student's *t*-test and the Mann-Whitney *U* test.

Behavioral Ecology 17, 688-690.

<http://dx.doi.org/10.1093/beheco/ark016>

What is the estimated difference in mean weight (along with a 95% confidence interval for the difference). Interpret this confidence interval. Is the interval consistent with the p-value from the hypothesis test?

What is the difference between the confidence interval for the mean and the ranges reported in the paper. What assumptions are you making for these measures of variability to be sensible?

Why the actual phrasing used in the paper ("Neither the presence nor the amount of cellulose per nest differed between species") incorrect? Hint. Suppose you are comparing the heights of males and female students. Which is a better phrase to summarize the results:

- Male heights are larger than female heights. [Are all males taller than all females?]
- Mean height for males is larger than mean height for females.

On the surface, both *Proc Ttest* and *Proc GLM* could be used to compare the means of two groups when the data are collected using a Completely Randomized Design (CRD). View <http://www.youtube.com/watch?v=jj8BX3g4JeA> for a video on comparing the usage of the two procedures. However, only *Proc Ttest* can deal with the unequal variance across groups through the Welch (unequal-variance) t-test.

14. Repeat the *Proc Univariate*, *SGplot*, and *Proc Univariate* for the NUMBER OF MITES. What do you conclude?

Note that you won't get the results in the paper (!) because they did more than a simple t-test. They actually fit an ANCOVA model (we will cover this later in the course).

15. The above is useful, but not very interesting in answering the question of interest.

Plot the number of parasites vs. the weight of butts (similar to Figure 1) with a different plotting symbol for each host species. Your code will look similar to:

```
proc sgplot data=nestinfo;
  title2 'number of mites vs weight of butts';
  scatter x=**** y=**** / group=****;
run;
```

Label the axes appropriately using the *Xaxis* and *Yaxis* statements in *SGplot*.

A good exploratory tool is to fit a smooth curve to the data. You previously used the *LOESS* statement in *Proc SGplot* to overlay a smooth curve. Another smoother is a spline. Use the *PBSPLINE* statement in *Proc SGplot* to overlay a smooth curve between the number of parasites and the weight of butts. What appears to be happening?

16. The authors claim to have fit a regression line of $\log(\text{parasites})$ vs. butt weight. Create a suitable derived variable in a new dataset, i.e. create variable for the logarithm (base e) of the number of parasites.
17. Use *Proc Reg* to fit the regression to the LOG of the number of parasites. Your code will look something like:

```
proc reg data=nestinfo;
  title2 'regression of log(number of mites) vs butt weight';
  model **** = ****;
  output out=modelfit pred=estmean_log lclm=lclm_log uclm=uclm_log;
run;
```

Here we use the *Output* statement in *Proc Reg* (NOTE IT IS NOT ODS OUTPUT!) to create new dataset that includes all of the original variables plus the fitted mean (on the log-scale) and the lower and upper confidence limits for the MEAN response on the LOG-SCALE.

What is the fitted line? Interpret the coefficients – hint: because the *Y* variable is on the log scale, what is the interpretation of $\exp(\text{slope})$? Hint: read the hints at the end of the assignment.

Perform a suitable hypothesis test. What is the null and alternate hypothesis, the appropriate test-statistic, and the p-value. Interpret the p-value.

Find a 95% confidence interval for the slope. Is this interval consistent with the p-value?

Look at the diagnostic plots – do you notice any problems? Aside from a single potential outlier the model seem adequate.² In actual fact, because the response variable (number of mites) are smallish counts, a poisson regression (generalized linear model) is likely a better choice for this analysis. But a close approximation is obtained by modeling $\log(\text{count})$ vs. *X* variables as long as there are no 0 counts where $\log(0)$ is undefined!

²A good statistician would rerun the model removing this outlier to see if the results are materially different.

Look at the fitted regression line with the confidence interval for the mean and the prediction interval for the individual responses. Note the outlier. Do you understand the difference between these two intervals and when you would be interested in using either interval?

18. We need now to ANTI-LOG the results. Do an anti-log transform of the predicted mean and the confidence limits using a code fragment similar to

```
data modelfit;
  set modelfit;
  estmean      = exp(****);
  estmean_lcl  = exp(****);
  estmean_ucl  = exp(****);
run;
```

Print out a few records to verify that you've done the anti-log transform correctly.

19. Finally use *Proc SGplot* to create a scatterplot, with the fitted line, and confidence limits for the mean number of mites. Your code will use a combination of the *Scatter* (with the *Group* option), the *Series* statement to draw the mean response, and the *Band* statement to draw confidence bands. Notice that the order of statements within *Proc Sgplot* affects the output with later statements “hiding” the results from earlier statements, so you may have to try several different orders to get a suitable plot. **You should sort the data by the weight of butts in the nests before plotting – otherwise your plot will look funny.** Your code fragment will look something like:

```
proc sort data=modelfit; by butts_weight; run;
proc sgplot data=modelfit;
  title2 'Fitted regression line of log(number mites) vs
        butt weight on the ANTI-log scale';
  band    x=****    upper=estmean_ucl lower=****;
  scatter x=****    y=number_of_mites / group=species;
  series  x=****    y=estmean;
run;
```

Compare your final plot to Figure 1 of the published paper.

Hand in the following using the online submission system:

- Your *SAS* code.
- A PDF files containing the output from your *SAS* program.

-
- A one page (maximum) double spaced PDF file containing a short write up of the results. The only computer output that I want is the final plot (similar to Figure 1 of the published paper) from above (use ODS RTF) to send it to an RTF file.

Your report should contain

- A (very) brief description of the data set (2 or 3 sentences),
- The fitted regression line (with measures of precision) on the log-scale, an interpretation of the slope (caution: because it is on the log-scale, what is the interpretation of $\exp(\textit{estimated slope})$ (5 or 6 sentences).
- Finally, what is your overall conclusion about the effect of the weight of cigarette butts on the number of ectoparasites in nests? Presumably because birds only would gather “used” butts (why?), the weight of butts serves a surrogate for the amount of nicotine (and other chemicals) in the nest.

Whew!

Hints and comments from the marker from past years

Some general hints:

- Again, there were many submission without names. If there is no name on ALL parts of the assignments, we can't assign grades. See my standards for code and writeup for submitting code and assignments on the Assignment page.
- Another general error was submitting the log file instead of the script. You should NOT get a log file if you are running the assignments using the *SAS* interactive editor. If you are using the batch submission system, you are doing things the hard way. Please see me about this.
- The style of the *SAS* scripts was much better this time, but there is still room for improvement in terms of spacing, indents, comments and headers. I am trying to prepare you for real work experiences. In the real world, you need to document code, and make your code look “nice” to read, and review. In many cases, you will write some code, leave the project for several months, and then come back to it. You need to be able to figure out what was done.

Part I

***** When do you put a variable in the *CLASS* or *VAR* statement of *Proc Tabulate*?

The *CLASS* statement is used to define grouping variables, e.g. factors that typically define the margins of the table. Any alphanumeric variable must appear in a *CLASS* statement as you can't compute any statistics (other than *n* on it).

If you want to compute means and other summary statistics, the variable must appear in the *VAR* statement.

***** Dot plots.

You will find it easiest to use *Proc SGplot* and the *SCATTER* statement to get the dot plots rather than the *DOT* statement, i.e. use

```
proc sgplot ...  
  scatter x=... y=...;
```

***** Box plots.

Look at example 9 of the SGplot documentation to see how to use SGplot to get box plots. This is a newer (and better) method than using Proc Boxplot (which is an older procedure).

***** ODS table names.

In my example in my course notes for *Proc GLM*, I used several ODS output statements for several of the tables. You will only need to use ONE of these (check the assignment for details).

***** What is the difference between a box-plot and a confidence interval?
This is discussed in many statistics textbooks.

***** What do the notches on a box-plot tell me?
Refer to the weblinks in the assignment for an explanation.

***** (Marker). Don't copy Carl's solutions word-for-word.

Part 2

Some hints from the marker about common errors:

- no caption on plots.
- not reporting an F-test result and p -value which provides evidence that the monthly means are not all the same.
- not including confidence intervals in the plot of the mean number of accidents per month.
- writeups that exceeded one page (not penalized, but will be next time).
- plotting histogram of accidents by their minute of occurrence instead of by hour (not very common).

All plots (figures) must have a caption (legend) at the BOTTOM of the figure. All tables must have a caption (legend) at the TOP of the table. You can't add these using *SAS*, but MUST add these after the fact using your word processor. See the solutions for examples of this.

***** My dates and number values don't look like dates or numbers. *SAS* converts dates and number into internal values. If you don't associate an out-format (e.g. *yymmdd10.*) with a date variable, it won't display the dates in a meaning format.

***** Don't tabulate variables with many thousands of levels!
Don't use *Proc Tabulate* on the variables in the *AccidentData* set that have several THOUSAND values!

For example, every accident has a different Accident ID and so you should not use *Proc Tabulate* on this variable.

***** (Marker). Be sure that conclusions in ANOVA are about MEANS and conclusions from χ^2 tests are about PROPORTIONS.

Part 3

Common errors noted by the marker in the write up and code:

- There was much confusion about the interpretation of the regression for this question. Many students said that a one unit increase in buttweight led to an $\exp(0.2)$ decrease in number of mites.
- Along those lines, it was common to see the regression expressed as

$$\text{number of mites} = \exp(3.49) - \exp(0.2) * \text{buttweight}$$

Please see below.

- No regression line reported at all
- Regression line reported without standard errors
- No concrete interpretation of the regression line (i.e. saying 'there is a negative relationship' but not elaborating)
- Reporting too many (i.e. 7) digits
- Plotting errors, like producing the plot with the *y*-axis still on the log scale, plotting unsorted data, plotting something else entirely.
- General style errors: spaces, indents, comments, and headers in code.
- Generating plots with the wrong title or axis labels in code.

There are NO jobs available for people who cannot explain what they are doing with statistical packages. There are MANY jobs available for people with quantitative skills who can explain the results clearly and correctly.

In the first 2 years at University, we cram as many facts as possible into you. In 3rd and 4th year, we are now interested in integrating the results.

- What happens when you do the regression on the log scale? Suppose that the final regression line is

$$\log(\text{mites}) = 3.0 - 0.2 * \text{buttweight}$$

The interpretation of the slope (-0.2) implies that for every increase by 1 g for the buttweight, the $\log(\text{mites})$, on average, declines by 0.2. But what does this mean on the anti-log scale?

The anti-log of this above equation is

$$\text{mites} = \exp(3)\exp(-0.2\text{buttweight})$$

You may need review your knowledge about logarithms if you don't understand why this is the equation on the anti-log scale.

So now consider a buttweight of 10 g vs. 11 g (any 2 values will give the same results as below)

$$\text{mites}(10 \text{ g}) = \exp(3)\exp(-0.2 * 10)$$

$$\text{mites}(11 \text{ g}) = \exp(3)\exp(-0.2 * 11)$$

Now take the ratio of the second equation to the first equation. This gives

$$\frac{\text{mites}(11 \text{ g})}{\text{mites}(10 \text{ g})} = \exp(-0.2 * 11 - (-0.2 * 10))$$

Again review the properties of logarithms if you don't understand this. This reduced to

$$\exp(-0.2) = 0.81$$

So every time, the butt weight goes up by 1 g, the number of mites, on average, is MULTIPLIED by 0.81, i.e. a 19% decrease.

Look at the graph of mites vs. butts weight with the fitted line. At butts weight = 0, the average number of mites is about 30. At butts weight = 1, the average number of mites is about 25. This is a multiplier of $25/30 = 0.82$ or about a 20% reduction. The same MULTIPLICATIVE effect continues over the entire curve.

***** I don't understand why we can't use the *REG* statement in *SGplot* directly?

The problem is that the regression is done on the log-scale, but you want to display the results on the anti-log scale.