Sassafras Manual

Contents

1 Introduction

Sassafras is a shell mode program for statistical analysis.

To build and run:

```
make
./sassafras infile
```

Infile is a text file that tells the program what to do. The syntax is a subset of SAS-Language. There are “data steps” and “procedure steps.” Data steps get data into the program and procedure steps compute the results. A data step begins with the keyword *data* and a procedure step begins with the keyword *proc*.

Example

A die, which may be loaded, is tossed six times. The observed point values are one to six. Compute a 95% confidence interval for the true mean μ given the observed data.

```
data
input y
datalines
1
2
3
4
5
6
;
proc means clm
```

The following result is displayed.

```
Variable     95% CLM MIN     95% CLM MAX
Y                  1.537           5.463
```

Here is the same result using R.

```
> y = c(1,2,3,4,5,6)
> t.test(y)

        One Sample t-test

data:  y
t = 4.5826, df = 5, p-value = 0.005934
alternative hypothesis: true mean is not equal to 0
95 percent confidence interval:
 1.536686 5.463314
```

2 Data Step

A data step is used to get data into the program.

`data` *name*

`infile "`*filename*`" dlm="`*delims*`" firstobs=`*n*

`input` *list*

var `=` *expression*

`datalines`

Notes

1. *name* is optional.
2. The `dlm` and `firstobs` settings are optional.
3. *delims* is a sequence of delimiter characters. The default is tab, comma, and space.
4. *n* is the starting input record number. Use `firstobs=2` to skip a header in the data file.
5. *list* is a list of variable names separated by spaces. For each categorical variable place a `$` after the variable name.
6. Optional *var* = *expression* statements create new vectors in the data set.
7. The `datalines` statement is followed by observational data. At the end of the data a blank line or a semicolon terminates the statement.

Example 1

The following example is a minimalist data step with in-line data.

```
data
input y
datalines
1
2
3
4
5
6
```

Example 2

Use `@@` at the end of an input statement to allow multiple values on an input line.

```
data
input y @@
datalines
1 2 3
4 5 6
```

Example 3

A dollar sign after an input variable indicates that the variable is categorical instead of numerical.

```
data
input trt $ y @@
datalines
A 6    A 0    A 2    A 8   A 11
A 4    A 13   A 1    A 8   A 0
B 0    B 2    B 3    B 1   B 18
B 4    B 14   B 9    B 1   B 9
C 13   C 10   C 18   C 5   C 23
C 12   C 5    C 16   C 1   C 20
```

Example 4

An infile statement is used to read data from a file.

```
data
input color $ x1 x2 x3 x4 x5 x6 x7 x8 x9 x10 x11 y
infile "wine-data"
```

Example 5

Expressions in a data step create new data vectors. The following example creates Y2 which is the input vector Y squared element-wise.

```
data
input color $ x1 x2 x3 x4 x5 x6 x7 x8 x9 x10 x11 y
infile "wine-data"
y2 = y ** 2
```

3 Anova Procedure

The anova procedure fits a classification model to data using ordinary least squares. The response variable must be numeric and the explanatory variables must be categorical.

> `proc anova data=`*name*
>
> `model` y `=` *list*
>
> `means` *list*
>
> `means` *list* `/ lsd ttest alpha=`*value*

Notes

1. `data=`*name* is optional. The default is data from the most recent data step.
2. y is the response variable which must be numeric.
3. *list* is one or more explanatory variables separated by spaces. The explanatory variables must be categorical. Interaction terms are specified using the syntax `A*B`.
4. The means statement can include one or more of the following options.

`lsd`	Compare treatment means using least significance difference
`ttest`	Compare treatment means using two sample t-test
`alpha`	Set the level of significance. Default is 0.05.

Example

```
data
input trt $ y @@
datalines
A 6    A 0    A 2    A 8    A 11
A 4    A 13   A 1    A 8    A 0
B 0    B 2    B 3    B 1    B 18
B 4    B 14   B 9    B 1    B 9
C 13   C 10   C 18   C 5    C 23
C 12   C 5    C 16   C 1    C 20

proc anova
model y = trt
means trt / lsd ttest
```

The following result is displayed.

```
                              Analysis of Variance

    Source      DF      Sum of Squares      Mean Square      F Value      Pr > F
    Model        2        293.60000000     146.80000000         3.98      0.0305
```

```
Error       27       995.10000000       36.85555556
Total       29      1288.70000000

            R-Square     Coeff Var     Root MSE       Y Mean
            0.227826     76.846553     6.070878     7.900000

 Source      DF         Anova SS      Mean Square      F Value      Pr > F
 TRT          2     293.60000000     146.80000000         3.98      0.0305
```

Mean Response

TRT	N	Mean Y	95% CI MIN	95% CI MAX
A	10	5.300000	1.360938	9.239062
B	10	6.100000	2.160938	10.039062
C	10	12.300000	8.360938	16.239062

Least Significant Difference Test

TRT	TRT	Delta Y	95% CI MIN	95% CI MAX	t Value	Pr > |t|	
A	B	-0.800000	-6.370676	4.770676	-0.29	0.7705	
A	C	-7.000000	-12.570676	-1.429324	-2.58	0.0157	*
B	A	0.800000	-4.770676	6.370676	0.29	0.7705	
B	C	-6.200000	-11.770676	-0.629324	-2.28	0.0305	*
C	A	7.000000	1.429324	12.570676	2.58	0.0157	*
C	B	6.200000	0.629324	11.770676	2.28	0.0305	*

Two Sample t-Test

TRT	TRT	Delta Y	95% CI MIN	95% CI MAX	t Value	Pr > |t|	
A	B	-0.800000	-5.922306	4.322306	-0.33	0.7466	
A	C	-7.000000	-12.664270	-1.335730	-2.60	0.0182	*
B	A	0.800000	-4.322306	5.922306	0.33	0.7466	
B	C	-6.200000	-12.467653	0.067653	-2.08	0.0523	
C	A	7.000000	1.335730	12.664270	2.60	0.0182	*
C	B	6.200000	-0.067653	12.467653	2.08	0.0523	

Mean response table

The confidence interval for a treatment mean is computed as follows.

$$\bar{y}_i \pm t(1-\alpha/2, dfe) \cdot \sqrt{\frac{MSE}{n_i}}$$

Recall that MSE is an estimate of model variance. From the anova table

```
Error       27       995.10000000       36.85555556
```

we obtain

$$
\begin{aligned}
MSE &= 36.85555556 \\
dfe &= 27
\end{aligned}
$$

Using R, the confidence interval for the mean of treatment A can be checked as follows.

```
> MSE = 36.8556
> dfe = 27
> t = qt(0.975,dfe)
> 5.3 - t * sqrt(MSE/10)
[1] 1.360934
> 5.3 + t * sqrt(MSE/10)
[1] 9.239066
```

Least significant difference test

The least significant difference of two means $\bar{y}_i$ and $\bar{y}_j$ is

$$LSD_{ij} = t(1 - \alpha/2, dfe) \cdot \sqrt{MSE \cdot \left(\frac{1}{n_i} + \frac{1}{n_j}\right)}$$

The corresponding confidence interval is

$$\bar{y}_i - \bar{y}_j \pm LSD_{ij}$$

Two sample t-test

The two sample t-test is computed as follows.

$$
\begin{aligned}
SSE &= \widehat{Var}_i \cdot (n_i - 1) + \widehat{Var}_j \cdot (n_j - 1) \\
dfe &= n_i + n_j - 2 \\
MSE &= \frac{SSE}{dfe} \\
SE &= \sqrt{MSE \cdot \left(\frac{1}{n_i} + \frac{1}{n_j}\right)} \\
t^* &= \frac{\bar{y}_i - \bar{y}_j}{SE}
\end{aligned}
$$

SSE is the sum of squares error recovered from variance estimates, dfe is the degrees of freedom error, MSE is mean square error, SE is the standard error, and t^* is the test statistic. The confidence interval is

$$\bar{y}_i - \bar{y}_j \pm t(1 - \alpha/2, dfe) \cdot SE$$

The null hypothesis is that the two treatment means are equal.

$$H_0 : \bar{y}_i = \bar{y}_j$$

If $|t^*|$ is greater than the critical value $t(1-\alpha/2, dfe)$, or equivalently, if the confidence interval does not cross zero, then reject H_0 and conclude that the treatment means are not equal. The following R session uses the above equations to duplicate the Sassafras result for treatments A and B.

```
> YA = c(6,0,2,8,11,4,13,1,8,0)
> YB = c(0,2,3,1,18,4,14,9,1,9)
> sse = var(YA) * (length(YA) - 1) + var(YB) * (length(YB) - 1)
> dfe = length(YA) + length(YB) - 2
> mse = sse / dfe
> se = sqrt(mse * (1 / length(YA) + 1 / length(YB)))
> t = (mean(YA) - mean(YB)) / se
> mean(YA) - mean(YB) - qt(0.975,dfe) * se
[1] -5.922307
> mean(YA) - mean(YB) + qt(0.975,dfe) * se
[1] 4.322307
> 2 * (1 - pt(abs(t),dfe))
[1] 0.746606
```

The same result is obtained with the t-test function.

```
> t.test(YA,YB,var.equal=TRUE)

        Two Sample t-test

data:  YA and YB
t = -0.3281, df = 18, p-value = 0.7466
alternative hypothesis: true difference in means is not equal to 0
95 percent confidence interval:
 -5.922307  4.322307
sample estimates:
mean of x mean of y
      5.3       6.1
```

4 Means Procedure

The means procedure prints statistics about a data set.

`proc means data=`*name* `alpha=`*value* `maxdec=`*n* *stats*

`var` *list*

`class` *list*

Notes

1. The settings that follow the `means` keyword are optional. The settings can appear in any order.
2. If `data` is not specified then the default is data from the most recent data step.
3. `alpha` sets the level of significance. The default is 0.05.
4. `maxdec` sets the decimal precision in the output. n ranges from 0 to 8. The default is 3.
5. *stats* is a list of statistics keywords from the following table.

`clm`	Confidence limits of the mean
`max`	Maximum value
`mean`	Mean value
`min`	Minimum value
`n`	Number of observations
`range`	max − min
`std`	Standard deviation s
`stddev`	Another keyword for s
`stderr`	Standard error $s/\sqrt{n}$
`var`	Variance s^2

   If *stats* is not specified then the default list is `n mean std min max`.
6. The optional `var` statement specifies which variables to print. The default is all variables. Variable names in *list* are separated by spaces.
7. The optional `class` statement prints statistics for each level of the categorical variables in *list*. Variable names in *list* are separated by spaces.

Example 1

The following example reads in the wine[1] data set and shows the default action of proc means.

```
data wine
input color $ x1 x2 x3 x4 x5 x6 x7 x8 x9 x10 x11 y
```

[1] P. Cortez, A. Cerdeira, F. Almeida, T. Matos and J. Reis. *Modeling wine preferences by data mining from physicochemical properties.* In Decision Support Systems, Elsevier, 47(4):547-553, 2009.

```
infile "wine-data"

proc means
```

The following result is displayed.

Variable	N	Mean	Std Dev	Minimum	Maximum
X1	6497	7.215	1.296	3.800	15.900
X2	6497	0.340	0.165	0.080	1.580
X3	6497	0.319	0.145	0.000	1.660
X4	6497	5.443	4.758	0.600	65.800
X5	6497	0.056	0.035	0.009	0.611
X6	6497	30.525	17.749	1.000	289.000
X7	6497	115.745	56.522	6.000	440.000
X8	6497	0.995	0.003	0.987	1.039
X9	6497	3.219	0.161	2.720	4.010
X10	6497	0.531	0.149	0.220	2.000
X11	6497	10.492	1.193	8.000	14.900
Y	6497	5.818	0.873	3.000	9.000

Example 2

The following example adds a var statement to show Y by itself. Also, the desired statistics are specified.

```
data wine
input color $ x1 x2 x3 x4 x5 x6 x7 x8 x9 x10 x11 y
infile "wine-data"

proc means n mean clm
var y
```

The following result is displayed.

Variable	N	Mean	95% CLM MIN	95% CLM MAX
Y	6497	5.818	5.797	5.840

Example 3

The following example adds a class statement to show statistics for each wine color.

```
data wine
input color $ x1 x2 x3 x4 x5 x6 x7 x8 x9 x10 x11 y
infile "wine-data"

proc means n mean clm
var y
class color
```

The following result is displayed.

COLOR	Variable	N	Mean	95% CLM MIN	95% CLM MAX
red	Y	1599	5.636	5.596	5.676
white	Y	4898	5.878	5.853	5.903

5 Print Procedure

The print procedure prints data in a data set.

> `proc print data=`*name*
>
> `var` *list*

Notes

1. `data=`*name* is optional. The default is data from the most recent data step.
2. The optional `var` statement specifies which variables to print. The default is all variables. Variable names in *list* are separated by spaces.

Example

The following example reads a data set and prints it.

```
data
input trt $ y @@
datalines
A 6    A 0    A 2    A 8   A 11
A 4    A 13   A 1    A 8   A 0
B 0    B 2    B 3    B 1   B 18
B 4    B 14   B 9    B 1   B 9
```

```
proc print
```

The following result is displayed.

Obs	TRT	Y
1	A	6
2	A	0
3	A	2
4	A	8
5	A	11
6	A	4
7	A	13
8	A	1
9	A	8
10	A	0
11	B	0
12	B	2
13	B	3
14	B	1
15	B	18
16	B	4

17	B	14
18	B	9
19	B	1
20	B	9

6 Reg Procedure

The reg procedure fits a linear model to data using ordinary least squares. The response variable must be numeric. For models with no intercept, anova results will differ from R. This is because R switches to uncorrected sums of squares for models with no intercept.

`proc reg data=`*name*

`model` *y* `=` *list*

`model` *y* `=` *list* `/ noint`

Notes

1. `data=`*name* is optional. The default is data from the most recent data step.
2. *y* is the response variable which must be numeric.
3. *list* is a list of explanatory variables separated by spaces. If functions of explanatory variables are required then they must be defined in the data step.
4. The `noint` option fits a linear model with no intercept term.

Example 1

The following example reads in the wine data set and fits a linear model with no intercept term.

```
data
input color $ x1 x2 x3 x4 x5 x6 x7 x8 x9 x10 x11 y
infile "wine-data"

proc reg
model y = color x1 / noint
```

The following result is displayed.

```
                         Analysis of Variance

            DF     Sum of Squares     Mean Square     F Value     Pr > F
Model        2           72.79210        36.39605       48.42     0.0000
Error     6494         4880.89360         0.75160
Total     6496         4953.68570

          Root MSE              0.86695     R-Square     0.0147
          Dependent Mean        5.81838     Adj R-Sq     0.0144
          Coeff Var            14.90018

                         Parameter Estimates
```

```
                 Estimate     Std Err     t Value     Pr > |t|
COLOR red         5.77309     0.08194       70.45       0.0000
COLOR white       5.99084     0.06628       90.39       0.0000
X1               -0.01647     0.00950       -1.73       0.0829
```

Example 2

The following exercise is from *Econometrics*[2]. Using data from a 1963 paper by Marc Nerlove, estimate parameters for the model

$$\log(\text{COST}) = \beta_0 + \beta_1 \log(\text{KWH}) + \beta_2 \log(\text{PL}) + \beta_3 \log(\text{PF}) + \beta_4 \log(\text{PK}) + \varepsilon$$

where COST is production cost, KWH is kilowatt hours, PL is price of labor, PF is price of fuel, and PK is price of capital.

```
data
infile "nerlove-data"
input COST KWH PL PF PK
LCOST = log(COST)
LKWH = log(KWH)
LPL = log(PL)
LPF = log(PF)
LPK = log(PK)

proc reg
model LCOST = LKWH LPL LPF LPK
```

The following result is displayed.

```
                           Analysis of Variance

          DF     Sum of Squares     Mean Square     F Value     Pr > F
Model      4          269.51482        67.37870      437.69     0.0000
Error    140           21.55201         0.15394
Total    144          291.06683

          Root MSE             0.39236     R-Square      0.9260
          Dependent Mean       1.72466     Adj R-Sq      0.9238
          Coeff Var           22.74969

                           Parameter Estimates

                Estimate     Std Err     t Value     Pr > |t|
Intercept       -3.52650     1.77437       -1.99       0.0488
LKWH             0.72039     0.01747       41.24       0.0000
LPL              0.43634     0.29105        1.50       0.1361
```

[2] Hansen, Bruce E. *Econometrics.* www.ssc.wisc.edu/~bhansen

```
        LPF             0.42652     0.10037         4.25        0.0000
        LPK            -0.21989     0.33943        -0.65        0.5182
```

The following code can be pasted into R to obtain a similar result.

```
d = read.table("nerlove-data")
lcost = log(d[,1])
lkwh = log(d[,2])
lpl = log(d[,3])
lpf = log(d[,4])
lpk = log(d[,5])
m = lm(lcost ~ lkwh + lpl + lpf + lpk)
summary(m)
```

The following result is displayed in R.

```
Call:
lm(formula = lcost ~ lkwh + lpl + lpf + lpk)

Residuals:
     Min       1Q   Median       3Q      Max
-0.97784 -0.23817 -0.01372  0.16031  1.81751

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept) -3.52650    1.77437  -1.987   0.0488 *
lkwh         0.72039    0.01747  41.244  < 2e-16 ***
lpl          0.43634    0.29105   1.499   0.1361
lpf          0.42652    0.10037   4.249 3.89e-05 ***
lpk         -0.21989    0.33943  -0.648   0.5182
---
Signif. codes:  0 *** 0.001 ** 0.01 * 0.05 . 0.1   1

Residual standard error: 0.3924 on 140 degrees of freedom
Multiple R-squared:  0.926,        Adjusted R-squared:  0.9238
F-statistic: 437.7 on 4 and 140 DF,  p-value: < 2.2e-16
```

Example 3

The following model uses the “trees” data set from R.

```
data
input Girth Height Volume
LG = log(Girth)
LH = log(Height)
LV = log(Volume)
datalines
  8.3    70   10.3
```

```
 8.6     65   10.3
 8.8     63   10.2
10.5     72   16.4
10.7     81   18.8
10.8     83   19.7
11.0     66   15.6
11.0     75   18.2
11.1     80   22.6
11.2     75   19.9
11.3     79   24.2
11.4     76   21.0
11.4     76   21.4
11.7     69   21.3
12.0     75   19.1
12.9     74   22.2
12.9     85   33.8
13.3     86   27.4
13.7     71   25.7
13.8     64   24.9
14.0     78   34.5
14.2     80   31.7
14.5     74   36.3
16.0     72   38.3
16.3     77   42.6
17.3     81   55.4
17.5     82   55.7
17.9     80   58.3
18.0     80   51.5
18.0     80   51.0
20.6     87   77.0

proc reg
model LV = LG LH
```

The following result is displayed.

```
                              Analysis of Variance

    Source      DF     Sum of Squares      Mean Square      F Value      Pr > F
    Model        2         1.53213547       0.76606773       613.19      0.0000
    Error       28         0.03498056       0.00124931
    Total       30         1.56711603

                Root MSE            0.03535     R-Square      0.9777
                Dependent Mean      1.42133     Adj R-Sq      0.9761
                Coeff Var           2.48679

                              Parameter Estimates
```

```
Parameter        Estimate     Std Err     t Value     Pr > |t|
(Intercept)      -2.88007     0.34734       -8.29       0.0000
log(Girth)        1.98265     0.07501       26.43       0.0000
log(Height)       1.11712     0.20444        5.46       0.0000
```

Let us see if the above parameters correspond to the volume of a cone given by

$$V = \frac{\pi}{12} d^2 h$$

where d is the diameter (girth) and h is the height of the cone. The model from the regression is

$$\log V = -2.88 + 1.98 \log d + 1.12 \log h$$

Take the antilog of both sides and obtain

$$V = 0.00132 \times d^{1.98} \times h^{1.12}$$

The exponents resemble the volume formula but the overall coefficient 0.00132 is two orders of magnitude smaller than $\pi/12 \approx 0.262$. It turns out the discrepancy is due to the units of measure. Girth is measured in inches while height and volume are measured in feet. To convert girth from inches to feet requires a factor of 1/12. Hence the leading coefficient should be

$$\frac{\pi}{12} \times \frac{1}{144} \approx 0.00182$$

which is in the ballpark of 0.00132 from the regression model.

Let us compare the Reg results to R. The following block of code can be pasted directly into the R shell prompt.

```
d=log10(trees[,1])
h=log10(trees[,2])
V=log10(trees[,3])
m=lm(V~d+h)
summary(m)
```

This is the R result, which matches Reg.

```
Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept) -2.88007    0.34734  -8.292 5.06e-09 ***
d            1.98265    0.07501  26.432  < 2e-16 ***
h            1.11712    0.20444   5.464 7.81e-06 ***
---
Signif. codes:  0 *** 0.001 ** 0.01 * 0.05 . 0.1   1

Residual standard error: 0.03535 on 28 degrees of freedom
Multiple R-squared:  0.9777,       Adjusted R-squared:  0.9761
F-statistic: 613.2 on 2 and 28 DF,  p-value: < 2.2e-16
```

7 Review

Analysis of Variance

The components of an analysis of variance table are computed as follows.

	DF	SS	Mean Square	F-value	p-value
Model	$p-1$	SSR	$MSR = SSR/(p-1)$	$F^* = MSR/MSE$	$1 - F(F^*, p-1, n-p)$
Error	$n-p$	SSE	$MSE = SSE/(n-p)$		
Total	$n-1$	SST			

In the table, n is the number of observations and p is the number of model parameters including the intercept term if there is one. The sums of squares are computed as follows.

$$SSR = \sum (\hat{y}_i - \bar{y})^2$$
$$SSE = \sum (y_i - \hat{y}_i)^2$$
$$SST = \sum (y_i - \bar{y})^2$$

Recall that MSE is an estimate of model variance.

$$MSE = \hat{\sigma}^2$$

A simple way to model the response variable is to use the average $\bar{y}$. The p-value above indicates whether or not the regression model is better than $\bar{y}$. The null hypothesis is that the regression model is no better than the average, that is

$$H_0 : SST = SSE$$

The test for H_0 is known as an omnibus test because an equivalent hypothesis is

$$H_0 : \beta_1 = \beta_2 = \cdots = \beta_{p-1} = 0$$

Under H_0 we have $SSR = 0$ hence another equivalent hypothesis is

$$H_0 : F^* = 0$$

The test statistic F^* is used because it has a well-known distribution. Recall that the p-value is (loosely) the probability that H_0 is true. Hence for small p-values, reject H_0 and conclude that the regression model is better than $\bar{y}$.

Confidence interval of the mean

The confidence interval of the mean is

$$\bar{x} \pm t_{1-\alpha/2,n-1} \frac{s}{\sqrt{n}}$$

where $\bar{x}$ is the observed mean, s is the observed standard deviation, n is the number of observations, and $t_{1-\alpha/2,n-1}$ is the quantile function. In R, the confidence interval of the mean of `1:10` can be computed as follows.

```
> x = 1:10
> n = length(x)
> alpha = 0.05
> mean(x) - qt(1-alpha/2,n-1) * sd(x)/sqrt(n)
[1] 3.334149
> mean(x) + qt(1-alpha/2,n-1) * sd(x)/sqrt(n)
[1] 7.665851
```

Alternatively, the `t.test` function can be used.

```
> t.test(1:10)

        One Sample t-test

data:  1:10
t = 5.7446, df = 9, p-value = 0.0002782
alternative hypothesis: true mean is not equal to 0
95 percent confidence interval:
 3.334149 7.665851
sample estimates:
mean of x
      5.5
```

Recall that the quantile function is the inverse of the cumulative distribution function. Let F be the cumulative distribution function. Then

$$F(t_{1-\alpha/2,n-1}) = 1 - \alpha/2$$

For example, in R we have

```
> t = qt(0.975,8)
> t
[1] 2.306004
> pt(t,8)
[1] 0.975
```