ACB spreadsheet verification: bias estimate from EQA material

Ed Wilkes

This document describes the verification of the bias estimates from EQA material spreadsheet, written by Prof Anders Kallner, that performs calculations for the assessment of bias between a given material and EQA materials (July 2018 version). Calculations performed by these spreadsheets were verified in an independent statistical software (the R statistical computing environment v3.4.1) by the author of this document. The R packages required to run this code are shown below. This code can be copied and pasted into an instance of R and, given the test data as input, reproduce the analysis in this document.

Required packages:

```
require(dplyr)
require(knitr)
require(mcr)
require(reshape2)
require(VCA)
```

Reading group data into R:

```
# Read in csv file: "2018-07 ACB Trueness (bias) from EQA - group test
data.csv"
df_group <- read.csv(file.choose(), header = TRUE)
kable(df_group)
```

sample	mean	sd	n
1	3.4	0.3	23
2	1.9	0.3	24
3	6.6	0.5	23
4	10.4	0.6	24
5	10.9	0.8	23
6	13.1	1.0	23
7	20.2	1.5	24
8	24.8	1.5	23
9	31.2	1.6	22
10	35.0	6.0	23

Check calculation of SEMs from group EQA values:

These are the data shown in column F, from cell 10 onwards.

```
df_group %>%
```

```
  group_by(sample) %>%
  mutate(sem = sd / sqrt(n)) %>%
  kable
```

sample	mean	sd	n	sem
1	3.4	0.3	23	0.0625543
2	1.9	0.3	24	0.0612372
3	6.6	0.5	23	0.1042572
4	10.4	0.6	24	0.1224745
5	10.9	0.8	23	0.1668115
6	13.1	1.0	23	0.2085144
7	20.2	1.5	24	0.3061862
8	24.8	1.5	23	0.3127716
9	31.2	1.6	22	0.3411211
10	35.0	6.0	23	1.2510865

The calculated SEM values match those in the spreadsheet.

Reading sample data into R:

```
# Read in csv file: "2018-07 ACB Trueness (bias) from EQA - sample test
data.csv"
df_sample <- read.csv(file.choose(), header = TRUE)
kable(df_sample)
```

sample	result_1	result_2	result_3	result_4	result_5	result_6
1	3.3	3.5	3.7	4.0	3.2	2.5
2	1.6	1.7	1.9	1.5	2.0	2.1
3	6.9	7.2	6.6	6.7	7.0	7.1
4	10.0	8.2	6.9	7.2	9.5	9.1
5	8.7	9.8	8.5	8.9	9.5	9.2
6	12.5	12.6	12.2	12.8	12.7	12.6
7	18.8	19.7	18.7	19.0	19.5	19.2
8	21.3	25.6	20.7	26.5	27.6	21.1
9	29.2	32.6	29.9	30.0	34.0	32.5
10	32.0	33.0	33.5	36.0	35.6	37.0

Check calculation of sample means, SDs, Z-scores, absolute and relative differences:

These are the data shown in columns S:W, from cell 10 onwards.

```
df_sample_sum <- df_sample %>%
  melt(id.vars = "sample") %>%
  group_by(sample) %>%
    summarise(n_sample = n()
```

```
              ,mean_sample = mean(value)
              ,sd_sample = sd(value)
  ) %>%
  left_join(df_group, by = "sample") %>%
  mutate(z = (mean_sample - mean) / sd
         ,absolute_diff = mean_sample - mean
         ,relative_diff = (mean_sample - mean) / mean * 100
  )
  kable(df_sample_sum)
```

sample	n_sample	mean_sample	sd_sample	mean	sd	n	z	absolute_diff	relative_diff
1	6	3.366667	0.5125102	3.4	0.3	23	-0.1111111	-0.0333333	-0.9803922
2	6	1.800000	0.2366432	1.9	0.3	24	-0.3333333	-0.1000000	-5.2631579
3	6	6.916667	0.2316607	6.6	0.5	23	0.6333333	0.3166667	4.7979798
4	6	8.483333	1.2608198	10.4	0.6	24	-3.1944444	-1.9166667	-18.4294872
5	6	9.100000	0.4939636	10.9	0.8	23	-2.2500000	-1.8000000	-16.5137615
6	6	12.566667	0.2065591	13.1	1.0	23	-0.5333333	-0.5333333	-4.0712468
7	6	19.150000	0.3937004	20.2	1.5	24	-0.7000000	-1.0500000	-5.1980198
8	6	23.800000	3.1022572	24.8	1.5	23	-0.6666667	-1.0000000	-4.0322581
9	6	31.366667	1.9211108	31.2	1.6	22	0.1041667	0.1666667	0.5341880
10	6	34.516667	1.9600170	35.0	6.0	23	-0.0805556	-0.4833333	-1.3809524

All calculate values match those presented in the spreadsheet.

Check calculations of mean Z-scores and *t*-tests of results:

These are the data shown in cells R31:R38 and V31:39.

```
df_sample_sum %>%
```

```
  summarise(n_z = n()
            ,mean_z = mean(z)
            ,sd_z = sd(z)
            ,sem_z = sd(z) / sqrt(n_z)
            ,t_stat_z = t.test(x = z)$statistic
            ,p_value_z = t.test(x = z)$p.value
            ,total_mean_group = mean(mean)
            ,total_sem_group = sd(mean) / sqrt(n_z)
            ,total_mean_sample = mean(mean_sample)
            ,t_stat = t.test(x = mean
                             ,y = mean_sample
                             ,paired = TRUE)$statistic
            ,p_value = t.test(x = mean
                              ,y = mean_sample
                              ,paired = TRUE)$p.value

            ) %>%
  kable
```

n_z	mean_z	sd_z	sem_z	t_stat_z	p_value_z	total_mean_group	total_sem_group	total_mean_sample	t_stat	p_value
10	-0.7131944	1.151873	0.3642543	-1.957957	0.0819129	15.75	3.648752	15.10667	2.594275	0.0290094

All calculated values match those in the spreadsheet.

Fit regression models and check results:

These are the data shown in cells B32:F33 and C34.

Deming:

```
# Fit Deming regression and get parameters
mcreg(x = df_sample_sum$mean
      ,y = df_sample_sum$mean_sample
      ,error.ratio = 1
      ,alpha = 0.05
      ,method.reg = "Deming"
      ,method.ci = "analytical")@para
##                  EST         SE        LCI      UCI
## Intercept -0.6645609 0.46189767 -1.7296988 0.400577
## Slope      1.0013478 0.02408191  0.9458148 1.056881
# EST = estimate of parameter
# SE = standard error of parameter
# LCI = lower confidence interval
# UCI = upper confidence interval
```

OLR:

```
# Fit OLR model
lm(df_sample_sum$mean_sample ~ df_sample_sum$mean)
##
## Call:
## lm(formula = df_sample_sum$mean_sample ~ df_sample_sum$mean)
```

```
##
## Coefficients:
##       (Intercept)  df_sample_sum$mean
##           -0.6282              0.9990
# Get confidence intervals for above model
confint(lm(df_sample_sum$mean_sample ~ df_sample_sum$mean))
##                         2.5 %    97.5 %
## (Intercept)        -1.6908302 0.4345277
## df_sample_sum$mean  0.9436313 1.0544408
```

Correlation coefficients:

```
# Calculate Pearson's r and its confidence intervals
cor.test(x = df_sample_sum$mean
        ,y = df_sample_sum$mean_sample
        ,method = "pearson"
        ,conf.level = 0.95)
##
##  Pearson's product-moment correlation
##
## data:  df_sample_sum$mean and df_sample_sum$mean_sample
## t = 41.581, df = 8, p-value = 1.233e-10
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  0.9898954 0.9994755
## sample estimates:
##       cor
## 0.9976945
```

All calculated regression coefficients match those in the spreadsheet.

Perform ANOVA for estimating differences between samples:

These are the data shown in cells P42:V51. The ANOVA is fitted with the following equation, where y_i represents the Z-score of the i^{th} sample in the j^{th} repeat measurement; μ represents the grand mean of all measurements; α_i represents the effect of the i^{th} sample; and ϵ_{ij} represents the residual error (i.e., error between repeats) for the i^{th} sample and the j^{th} repeat.

$$y_{ij} = \mu + \alpha_i + \epsilon_{ij}$$

```
# Transform data
df_anova <- left_join(df_group, df_sample, by = "sample") %>%
  melt(id.vars = c("sample", "mean", "sd", "n")) %>%
  group_by(sample) %>%
  mutate(sample_z = (value - mean) / sd) %>%
  as.data.frame

# Perform ANOVA
anova_model <- anovaVCA(sample_z ~ sample, df_anova)
anova_model$aov.tab[,1:6]
##              DF       SS       MS       VC    %Total       SD
## total  28.57495       NA       NA 2.534284 100.00000 1.591944
## sample  9.00000 71.64785 7.960873 1.085318  42.82541 1.041786
```

```
## error  50.00000 72.44833 1.448967 1.448967  57.17459 1.203730
cat("F =", round(anova_model$aov.tab[2,3] / anova_model$aov.tab[3,3], 4))
## F = 5.4942
```

The calculated values match those shown in the spreadsheet.

Conclusions:

1. Calculations of SEMs from the group means and SDs matched those in the spreadsheet
2. Calculations of sample means, SDs, differences, and Z-scores matched those in the spreadsheet
3. Calculations of mean Z-scores and statistical tests between the differences in values produced identical results to those in the spreadsheet
4. Regression analyses produced identical results to those in the spreadsheet
5. The ANOVA matched that presented in the spreadsheet