Repeatability and reproducubility in ultrasound measurement

APhi, SDC 469
December 2014
Version 4

Compiled Tuesday 9th December, 2014, 13:55
from: C:/Bendix/Steno/APhi/repro.tex

Bendix Carstensen Steno Diabetes Center, Gentofte, Denmark
& Department of Biostatistics, University of Copenhagen
bxc@steno.dk
http://BendixCarstensen.com

Contents

Chapter 1

Intra-observer variability

1.1 The data set

The dataset consists of measurements of subcutaneous and visceral fat on 33 patients. Each patient has been measured 4 times by each observer:

Twice lying in the same position; average of two such measurements constitute the standard for an ultrasound measurement of fat thickness. Between these sessions patients are asked to get up and lie down again for a repeat measurement by the same or the other observer.

Hence the relevant measurements to consider from a *clinical* point of view are the averages of the measurements taken in the same lie-down round.

So measurements are classified by:

- Observer (referred to as "methods")
- Patient (referred to as "items")
- Replicates (between which the patient has been standing)

The replicate measurements are *exchangeable* within observers in the sense that a measurement by one observer is not connected to any particular of the two measurements by the other observer.

1.1.1 Initial plots and analyses

The dataset is acquired from a SAS-dataset as shown in the relvant section in chapter **??**

First we load the dataset and inspect the first 10 observations in order to get an overview of the structure of the dataset:

```
> library( MethComp )
> aphi <- read.xport( "./data/inter.xpt" )
> names( aphi ) <- tolower( names(aphi) )
> str( aphi )
  'data.frame':        240 obs. of  7 variables:
   $ ptno    : num  1 1 1 1 1 1 1 1 2 2 ...
   $ subkutan: num  3.77 3.85 3.56 3.82 3.72 3.41 3.52 4 2.32 2.49 ...
   $ viseral : num  9.48 9.33 9.12 8.74 8.89 8.85 8.9 8.85 8.36 9.45 ...
   $ operator: Factor w/ 2 levels "APHI","NABJ": 1 1 1 1 2 2 2 2 1 1 ...
   $ skema   : num  1 1 2 2 1 1 2 2 1 1 ...
   $ maaling : num  1 2 1 2 1 2 1 2 1 2 ...
   $ sex     : num  2 2 2 2 2 2 2 2 1 1 ...
```

We rename operators (`APHI` → `Son.1` and `NABJ` → `Son.2`):

```
> levels( aphi$operator ) <- c("Son.1","Son.2")
> aphi$sex <- factor( aphi$sex, labels=c("M","F") )
> head( aphi, 10 )
```

	ptno	subkutan	viseral	operator	skema	maaling	sex
1	1	3.77	9.48	Son.1	1	1	F
2	1	3.85	9.33	Son.1	1	2	F
3	1	3.56	9.12	Son.1	2	1	F
4	1	3.82	8.74	Son.1	2	2	F
5	1	3.72	8.89	Son.2	1	1	F
6	1	3.41	8.85	Son.2	1	2	F
7	1	3.52	8.90	Son.2	2	1	F
8	1	4.00	8.85	Son.2	2	2	F
9	2	2.32	8.36	Son.1	1	1	M
10	2	2.49	9.45	Son.1	1	2	M

Now take the mean over the two replicate measurements, and list the observations for the first patient from the two datasets.

```
> apm <- aggregate( aphi, by=list(aphi$skema,aphi$operator,aphi$ptno), FUN="mean" )
> names(apm)
 [1] "Group.1"  "Group.2"  "Group.3"  "ptno"     "subkutan" "viseral"
 [7] "operator" "skema"    "maaling"  "sex"

> names(apm)[1:3] <- c("skema","operator","ptno")
> apm <- apm[,c(1:3,5:6)]
> subset( apm, ptno==4 )
```

	skema	operator	ptno	subkutan	viseral
13	1	Son.1	4	1.665	4.600
14	2	Son.1	4	1.780	4.855
15	1	Son.2	4	1.640	5.175
16	2	Son.2	4	1.700	4.960

```
> subset( aphi, ptno==4 )
```

	ptno	subkutan	viseral	operator	skema	maaling	sex
25	4	1.59	4.49	Son.1	1	1	M
26	4	1.74	4.71	Son.1	1	2	M
27	4	1.76	4.79	Son.1	2	1	M
28	4	1.80	4.92	Son.1	2	2	M
29	4	1.76	5.24	Son.2	1	1	M
30	4	1.52	5.11	Son.2	1	2	M
31	4	1.57	4.91	Son.2	2	1	M
32	4	1.83	5.01	Son.2	2	2	M

Then we can set up two different datasets, one for each type of measurement:

```
> sub <- Meth( apm, meth="operator", item="ptno", repl="skema", y="subkutan" )
  The following variables from the dataframe
  "apm" are used as the Meth variables:
  meth: operator
  item: ptno
  repl: skema
     y: subkutan
        #Replicates
  Method           2 #Items #Obs: 120 Values:  min    med   max
    Son.1         30     30        60        0.375 2.2250 5.000
    Son.2         30     30        60        0.485 2.3275 5.355

> vis <- Meth( apm, meth="operator", item="ptno", repl="skema", y="viseral" )
```

```
The following variables from the dataframe
"apm" are used as the Meth variables:
meth: operator
item: ptno
repl: skema
   y: viseral
        #Replicates
Method      1     2 #Items #Obs: 119 Values:  min  med   max
  Son.1     1    29     30        59        2.910 6.11 12.09
  Son.2     0    30     30        60        3.015 7.15 13.38
```

For colorcoding by sex we also need a small dataframe with itemnumber and sex together to get the color sequence right:

```
> item2sex <- aphi[match(unique(aphi$ptno),aphi$ptno),c("ptno","sex")]
> names(item2sex)[1] <- "item"
> item2sex <- item2sex[order(item2sex$item),]
> sxclr <- c("blue","red")[item2sex$sex]
```

1.2 Limits of Agreement (LoA)

Based on these datasets we can now make Bland-Altman plots and compute limits of agreement separately for the two types of measurement. But in order to plot measurements we need to make some pairing of the replicates from the two observers. In figure 1.1 are given Bland-Altman plots both for the pairing in the dataset and for a random pairing.

```
> par( mfcol=c(2,2), mar=c(3,3,3,3), oma=c(0,0,0,0), mgp=c(3,1,0)/1.6, las=1 )
> BA.plot( sub, eqax=T, repl.conn=TRUE )
> BA.plot( vis, eqax=T, repl.conn=TRUE )
> BA.plot( perm.repl(sub), eqax=T, repl.conn=TRUE )
> BA.plot( perm.repl(vis), eqax=T, repl.conn=TRUE )
```

We can also consider to log-transform data, looking at relative differences and using the residual sds as coefficients of variation:

```
> par( mfcol=c(2,2), mar=c(3,3,3,3), oma=c(0,0,0,0), mgp=c(3,1,0)/1.6, las=1 )
> BA.plot( sub, mult=T, diflim=c(0.5,2) )
> BA.plot( vis, mult=T, diflim=c(0.5,2) )
> BA.plot( perm.repl(sub), mult=T, diflim=c(0.5,2) )
> BA.plot( perm.repl(vis), mult=T, diflim=c(0.5,2) )
```

Here is the code for the plots in the article:

```
> for( i in 1:3 )
+    {
+ if( i==1 )   postscript( "inter.eps", width=8, height=4, pointsize=12 )
+ if( i==2 ) win.metafile( "inter.emf", width=8, height=4, pointsize=12 )
+ if( i==3 )          pdf( "inter.pdf", width=8, height=4, pointsize=12 )
+ par( mfrow=c(1,2), mar=c(3,3,1,3), mgp=c(3,1,0)/1.6, las=1,
+      oma=c(0,0,0,0), cex=1 )
+ BA.plot( mean(sub), mult=T, diflim=c(0.5,2), axlim=c(0,6) )
+ mtext( "Subcutaneous fat", side=3, line=0.1, at=0, adj=0 )
+ BA.plot( mean(vis), mult=T, diflim=c(0.5,2), axlim=c(3,16) )
+ mtext( "Visceral fat", side=3, line=0.1, at=3, adj=0 )
+ text( "Figure 2", side=1, adj=0.5, outer=TRUE, line=-1 )
+ dev.off()
+     }
```

And here is the code for the corresponding plots in the dissertation colorcoded by sex:

```
> for( i in 1:2 )
+    {
+ if( i==1 )   postscript( "dis-inter.eps", width=8, height=4, pointsize=12 )
+ if( i==2 ) win.metafile( "dis-inter.emf", width=8, height=4, pointsize=12 )
+ if( i==3 )          pdf( "dis-inter.pdf", width=8, height=4, pointsize=12 )
+ par( mfrow=c(1,2), mar=c(3,3,1,3), mgp=c(3,1,0)/1.6, las=1,
+      oma=c(0,0,0,0), cex=1 )
+ BA.plot( mean(sub), mult=T, diflim=c(0.5,2), axlim=c(0,6),
+          col.lines="black", col.points=sxclr )
+ mtext( "Subcutaneous fat", side=3, line=0.1, at=0, adj=0 )
+ BA.plot( mean(vis), mult=T, diflim=c(0.5,2), axlim=c(3,16),
```

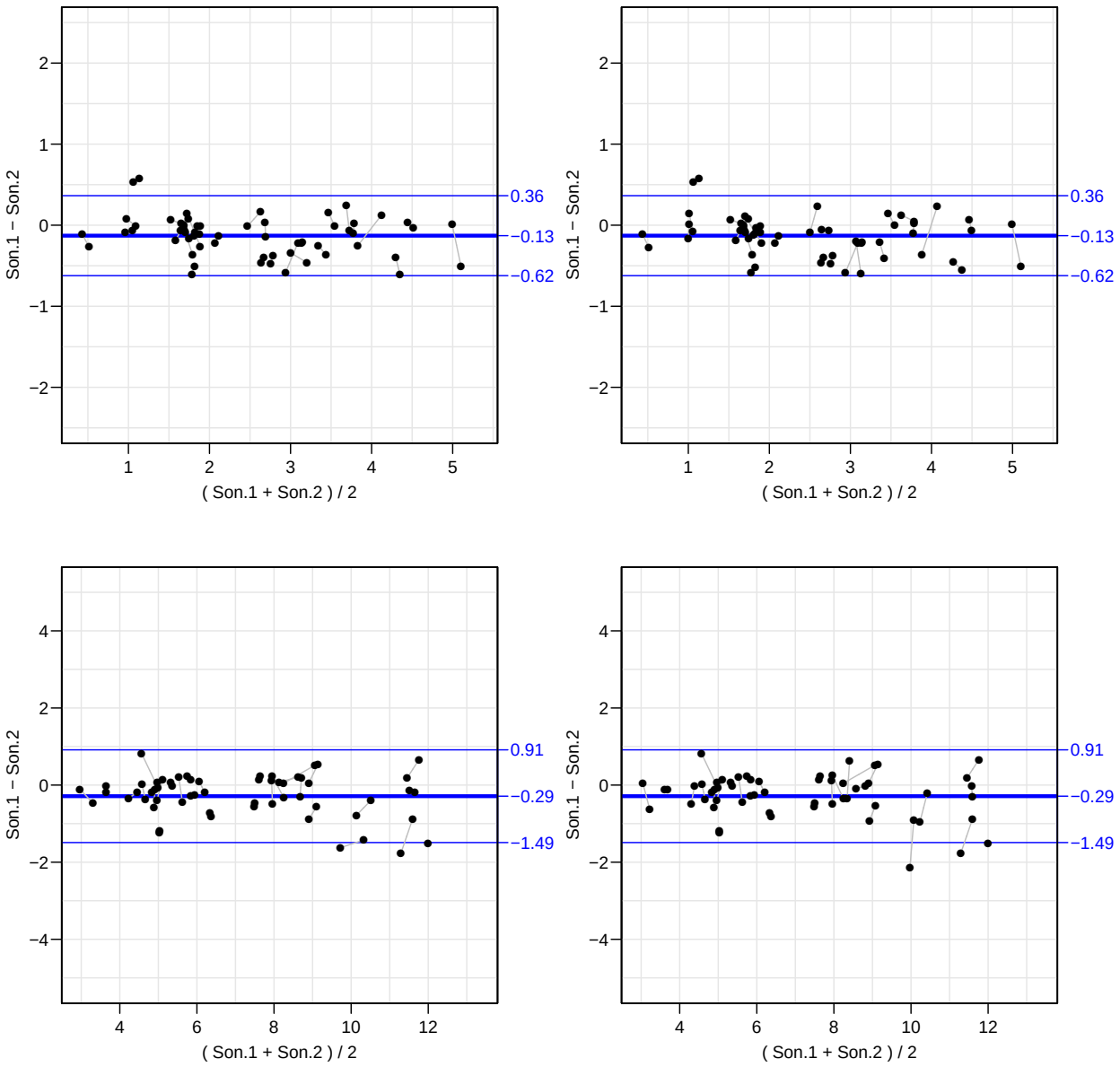

Figure 1.1: *Bland-Altman plots for subcutaneous (left) and visceral (right) fat. The left plots are based on the numbering of replicates in the dataset, the two right ones are based on a random pairing of the replicates. The y-axes are scaled to have the same extent as the x-axes, which in turn is scaled by the range of means.*

```
+          col.lines="black", col.points=sxclr )
+ mtext( "Visceral fat", side=3, line=0.1, at=3, adj=0 )
+ # mtext( "Figure 2", side=1, adj=0.5, outer=TRUE, line=-1 )
+ dev.off()
+     }
```

We also have the possibility of overlaying the two plots, both on the original scale (units are cm) as in figure 1.3 or using the log-transform as in figure 1.4.

```
> par( mfrow=c(1,1), mar=c(3,3,1,3), mgp=c(3,1,0)/1.6 )
> BA.plot( sub, axlim=c(0,14), diflim=c(-3,3), col.points="blue",
```

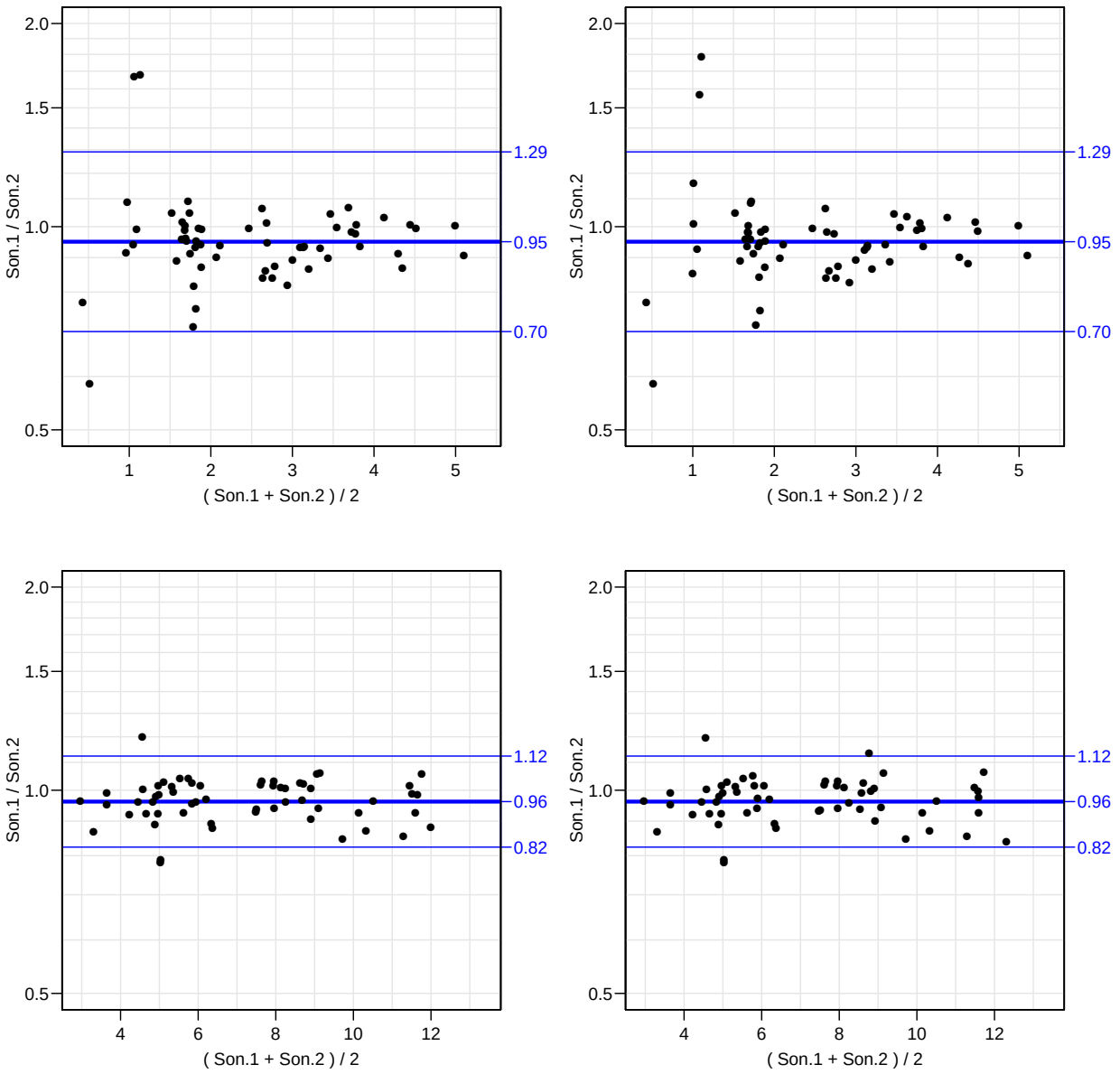

Figure 1.2: *Bland-Altman plots for subcutaneous (left) and visceral (right) fat, using the log-transformed data corresponding to relative differences between sonographers. The left plots are based on the numbering of replicates in the dataset, the two right ones are based on a random pairing of the replicates.*

```
+          col.lines="blue", xaxs="i" )
> par( new=T )
> BA.plot( vis, axlim=c(0,14), diflim=c(-3,3), col.points="red",
+          col.lines="red", xaxs="i", grid=FALSE )


> par( mfrow=c(1,1), mar=c(3,3,1,3), mgp=c(3,1,0)/1.6 )
> BA.plot( sub, axlim=c(0,14), diflim=c(0.5,2), col.points="blue",
+          col.lines="blue", mult=TRUE, xaxs="i" )
> par( new=T )
> BA.plot( vis, axlim=c(0,14), diflim=c(0.5,2), col.points="red",
+          col.lines="red", mult=TRUE, xaxs="i", grid=FALSE )
```

In order to assess whether the original data or the log-transformed data are the better we make a quck overview of the extent to which the assumptions about constant difference and constant variation are fulfilled:

```
> DA.reg( sub )
   Conversion between methods:
                 alpha   beta sd.pred beta=1 int(t-f) slope(t-f) sd(t-f) int(sd) slope(sd)   sd=K
To:   From:
Son.1 Son.1   0.000  1.000      NA     NA    0.000       0.000      NA      NA         NA     NA
      Son.2  -0.003  0.951   0.233  0.070   -0.003      -0.050   0.239   0.156      0.029  0.179
Son.2 Son.1   0.003  1.052   0.245  0.070    0.003       0.050  -0.239   0.156      0.029  0.179
      Son.2   0.000  1.000      NA     NA    0.000       0.000      NA      NA         NA     NA

> DA.reg( sub, Trans="log" )
```

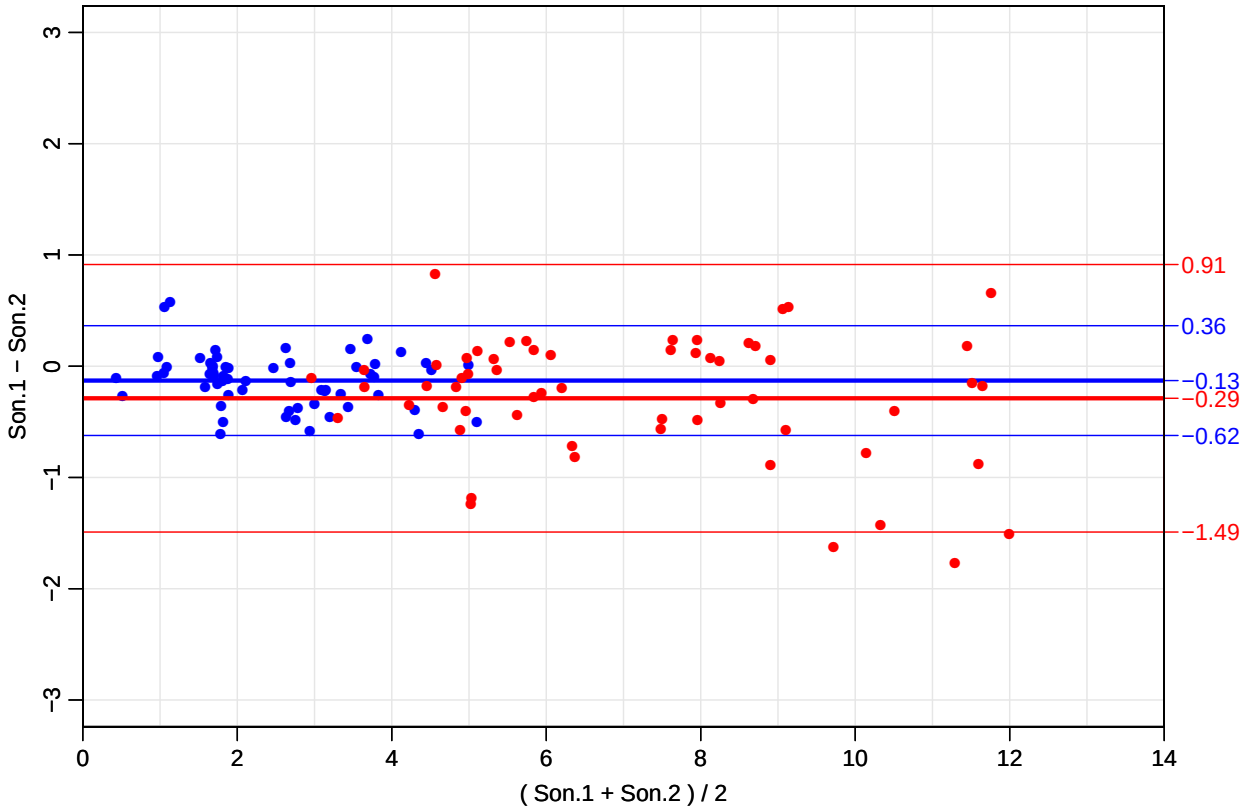

Figure 1.3: *Bland-Altman plots for subcutaneous (blue) and visceral (red) fat. Scale is the observed scale (cm).*

```
  Note: Response transformed by:  .Primitive("log")


   Conversion between methods:
                alpha   beta sd.pred beta=1 int(t-f) slope(t-f) sd(t-f) int(sd) slope(sd)   sd=K
  To:   From:
  Son.1 Son.1   0.000  1.000      NA     NA    0.000      0.000      NA      NA         NA     NA
        Son.2  -0.061  1.012   0.152  0.750   -0.061      0.012   0.151   0.216     -0.127  0.000
  Son.2 Son.1   0.061  0.988   0.150  0.750    0.061     -0.012  -0.151   0.216     -0.127  0.000
        Son.2   0.000  1.000      NA     NA    0.000      0.000      NA      NA         NA     NA
> DA.reg( vis )
   Conversion between methods:
                alpha   beta sd.pred beta=1 int(t-f) slope(t-f) sd(t-f) int(sd) slope(sd)   sd=K
  To:   From:
  Son.1 Son.1   0.000  1.000      NA     NA    0.000      0.000      NA      NA         NA     NA
        Son.2   0.064  0.955   0.528  0.112    0.065     -0.046   0.540  -0.022      0.075  0.001
  Son.2 Son.1  -0.067  1.047   0.553  0.112   -0.065      0.046  -0.540  -0.022      0.075  0.001
        Son.2   0.000  1.000      NA     NA    0.000      0.000      NA      NA         NA     NA
> DA.reg( vis, Trans="log" )
  Note: Response transformed by:  .Primitive("log")


   Conversion between methods:
                alpha   beta sd.pred beta=1 int(t-f) slope(t-f) sd(t-f) int(sd) slope(sd)   sd=K
  To:   From:
  Son.1 Son.1   0.000  1.000      NA     NA    0.000      0.000      NA      NA         NA     NA
        Son.2  -0.051  1.007   0.075  0.787   -0.051      0.007   0.075   0.069      0.001  0.955
  Son.2 Son.1   0.051  0.993   0.075  0.787    0.051     -0.007  -0.075   0.069      0.001  0.955
        Son.2   0.000  1.000      NA     NA    0.000      0.000      NA      NA         NA     NA
```

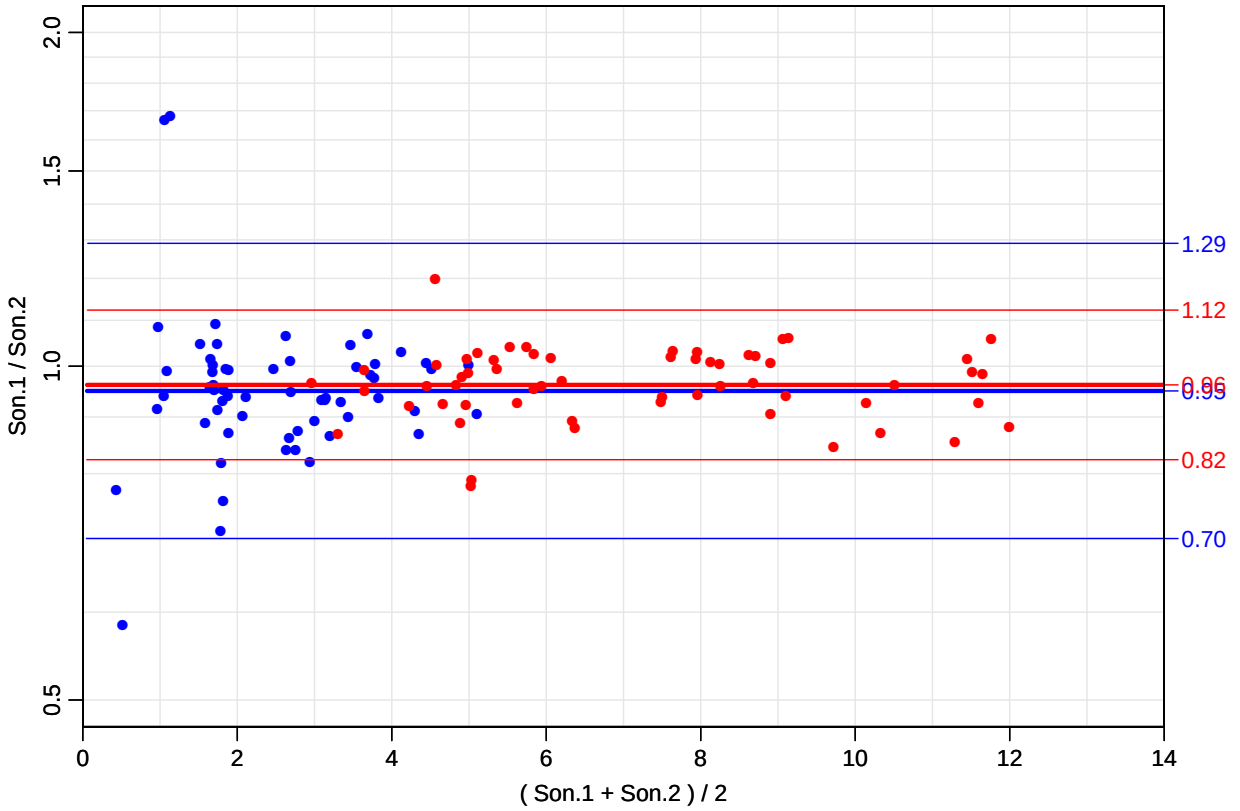

Figure 1.4: *Bland-Altman plots for subcutaneous (blue) and visceral (red) fat. Scale is the the log-scale, i.e. relative measurements between the two observers.*

The columns "`slope(sd)`" and "`sd=K`" gives the estimated slope of the standard deviation of the differences as a function of the averages and p-value for the test of wheter this is 0.

For subcutaneous data there is a significant decrease in the sd. for log-transformed data clearly owing to the few observations with very small values.

For visceral data there is a significant *increase* in sd for data on the original scale.

So either way, we violate the assumptions. But for the subcutaneous measurements the decreasing variation is clearly associated with the very small values which drives the relattive precision down. If we try to exclude patients with the smallest subcutaneous measurements we can see this:

```
> with( sub, sort( tapply( y, item, mean ) ) )
      8      27      28      17      29      19      18       4       6       3
0.47250 1.00500 1.03250 1.09625 1.60125 1.61375 1.69125 1.69625 1.74125 1.80000
     15      12      21      16      10      13       2      30       7      22
1.80250 1.81500 1.81625 1.86375 2.09000 2.54750 2.65250 2.68875 2.76875 3.01500
     24      25       9      20       1      23      11      26      14       5
3.10000 3.14000 3.38875 3.50500 3.70625 3.77875 3.97500 4.32375 4.48000 5.04750
> DA.reg( subset( sub, !(item %in% c(8,27,28,17) ) ), Tr="log" )
  Note: Response transformed by:  .Primitive("log")


   Conversion between methods:
                alpha   beta sd.pred beta=1 int(t-f) slope(t-f) sd(t-f) int(sd) slope(sd)   sd=K
  To:   From:
  Son.1 Son.1   0.000  1.000      NA     NA    0.000      0.000      NA      NA        NA     NA
        Son.2  -0.085  1.023   0.088  0.503   -0.084      0.023   0.087   0.117    -0.034  0.189
  Son.2 Son.1   0.083  0.978   0.086  0.503    0.084     -0.023  -0.087   0.117    -0.034  0.189
        Son.2   0.000  1.000      NA     NA    0.000      0.000      NA      NA        NA     NA
> DA.reg( subset( sub, !(item %in% c(8,17) ) ), Tr="log" )
  Note: Response transformed by:  .Primitive("log")


   Conversion between methods:
                alpha   beta sd.pred beta=1 int(t-f) slope(t-f) sd(t-f) int(sd) slope(sd)   sd=K
  To:   From:
  Son.1 Son.1   0.000  1.000      NA     NA    0.000      0.000      NA      NA        NA     NA
        Son.2  -0.060  1.000   0.087  0.988   -0.060      0.000   0.087   0.095    -0.014  0.527
  Son.2 Son.1   0.060  1.000   0.087  0.988    0.060      0.000  -0.087   0.095    -0.014  0.527
        Son.2   0.000  1.000      NA     NA    0.000      0.000      NA      NA        NA     NA
```

— so essentially it hinges on persons number 8 and 17:

```
> print( subset( sub, item==8 ) )
    meth item repl     y viseral
1 Son.1    8    1 0.375   4.975
2 Son.1    8    2 0.380   4.955
3 Son.2    8    1 0.485   4.150
4 Son.2    8    2 0.650   5.025
```

. . . who seems to have larger values both sor subcutaneous and visceral fat after standing up (at least according to sonographer 2).

1.3 Variance components model

The Bland-Altman plots are a bit deceptive, because we have chosen to use the replicate measurements as separate items. So in order to get the proper variance components estimated we set up a model for data along the lines set out in [2]. This will give us the residual variation for each observer and the allocation of the variation to the interaction between observers and patients (`MxI` — method×item where method is observer and item is patient) and to the residual variation — the variation between replicates *within* each observer. In the current experimental setup this is the variation induced by the patinets standing up between measurements.

Formally the model for a measurement (subcutaneous or visceral) y_{opr} by observer o on patient p, replicate r is:

$$y_{opr} = \mu_p + \alpha_o + a_{op} + e_{opr}, \quad a_{op} \sim \mathcal{N}(0, \tau^2), \quad e_{opr} \sim \mathcal{N}(0, \sigma_o^2)$$

Note that the residual variance depends on observer, allowing the measurements to have different precision between observers.

This model is a standard variance component model, see[2]. The fitting of this is implemented in the `MethComp` package that we are using here.

The residual variation estimated separately for each observer gives the uncertainty with which each observer can reproduce a measurement on a particular patient. The observer by patient variation represents how much variation there is between the observers across the patient population. The variance components are estimated in a proper model for the data where the *exchangeability* of the replicates are taken into account.

```
> ( BA.sub <- BA.est( sub, linked=FALSE ) )
   Conversion between methods:
                alpha   beta sd.pred LoA-lo LoA-up
  To:   From:
  Son.1 Son.1   0.000  1.000   0.145 -0.290  0.290
        Son.2  -0.129  1.000   0.247 -0.623  0.364
  Son.2 Son.1   0.129  1.000   0.247 -0.364  0.623
        Son.2   0.000  1.000   0.149 -0.299  0.299

   Variance components (sd):
        IxR  MxI   res
  Son.1   0 0.14 0.103
  Son.2   0 0.14 0.106
> ( BA.vis <- BA.est( vis, linked=FALSE ) )
   Conversion between methods:
                alpha   beta sd.pred LoA-lo LoA-up
  To:   From:
  Son.1 Son.1   0.000  1.000   0.383 -0.766  0.766
        Son.2  -0.288  1.000   0.601 -1.491  0.914
  Son.2 Son.1   0.288  1.000   0.601 -0.914  1.491
        Son.2   0.000  1.000   0.330 -0.659  0.659

   Variance components (sd):
        IxR   MxI   res
  Son.1   0 0.342 0.271
  Son.2   0 0.342 0.233
```

The variation in the visceral measurements are much larger than in the subcutaneous when measured on the absolute scale — clearly because of the different range of the absolute measurements (figure 1.3). But we also see that the residual variation (the variation *within*

observer) is slightly smaller than the observer by patient variation (the *between* observer variation).

These characteristics are the same when we use the (natural) log-transform on the original data. In this case the sd.s are interpretable as coefficients of variation; see *e.g.* chapter 9 of [1]. By that token we can compute the LoA and transform to the percentage scale:

```
> ( BA.lsub <- BA.est( sub, linked=FALSE, Transform="log" ) )

  Note: Response transformed by:  .Primitive("log")


   Conversion between methods:
               alpha   beta sd.pred LoA-lo LoA-up
  To:   From:
  Son.1 Son.1   0.000  1.000   0.059 -0.118  0.118
        Son.2  -0.051  1.000   0.153 -0.358  0.255
  Son.2 Son.1   0.051  1.000   0.153 -0.255  0.358
        Son.2   0.000  1.000   0.086 -0.172  0.172

   Variance components (sd):
        IxR   MxI   res
  Son.1   0 0.095 0.042
  Son.2   0 0.095 0.061

> (exp( BA.lsub$LoA )-1)*100

                     Mean    Lower    Upper       SD
  Son.2 - Son.1   5.269925 -22.5214 43.02991 16.56315

> ( BA.lvis <- BA.est( vis, linked=FALSE, Transform="log" ) )

  Note: Response transformed by:  .Primitive("log")


   Conversion between methods:
               alpha   beta sd.pred LoA-lo LoA-up
  To:   From:
  Son.1 Son.1   0.000  1.000   0.048 -0.096  0.096
        Son.2  -0.039  1.000   0.078 -0.194  0.117
  Son.2 Son.1   0.039  1.000   0.078 -0.117  0.194
        Son.2   0.000  1.000   0.057 -0.114  0.114

   Variance components (sd):
        IxR  MxI   res
  Son.1   0 0.04 0.034
  Son.2   0 0.04 0.040

> (exp( BA.lvis$LoA )-1)*100

                     Mean     Lower    Upper       SD
  Son.2 - Son.1   3.95672 -10.99977 21.42664 8.076361
```

Thus the percentwise limits of agreement for subcutaneous fat is $(-22.5; 43.0)\%$ and for visceral fat $(-11.0; 21.4)\%$. Thus the relative precision in visceral fat determination is better than for subcutaneous fat.

1.3.1 Precision of the variance estimates

1.3.2 Confidence intervals for the variance components

We can assess the precision of the variance components approximately by using the degrees of freedom. First take a look at the AOV-diagram from Carstensen (2010):

$$\begin{array}{ccccc} & & [M\times I]^{MI}_{MI-2M-I+2} & \longrightarrow & (\alpha_m+\beta_m\mu_i)^{2M+I-2}_{2M-2} \\ & \nearrow & & & \downarrow \\ [M\times I\times R]^{MIR}_{MIR-MI-IR} & & & & \\ & \searrow & & & \\ & & [I\times R]^{IR}_{IR-I} & \longrightarrow & (\mu_i)^{I}_{I} \end{array}$$

This is only valid for $M > 2$ (for $M = 2$ and $R = 2$ the residual d.f. is 0, which is clearly wrong). The catch is that for $M = 2$ there are not MI levels of the $M \times I$-effect, but only I (because the variation on both sides of the I-men is the same). Moreover, we are here concerned with a model with exchangeable replicates, so the $I \times R$-effect is not present. So the diagram is modified to:

$$\begin{array}{ccccc} & & [M\times I]^{2I}_{I-1} & \longrightarrow & (\alpha_m+\mu_i)^{I+1}_{1} \\ & \nearrow & & & \downarrow \\ [M\times I\times R]^{2IR}_{2I(R-1)+1} & & & & \\ & \searrow & & & \\ & & & \longrightarrow & (\mu_i)^{I}_{I} \end{array}$$

Thus with the setup where we have $I = 30$ and $R = 2$, the $M \times I$-effect has 29 d.f. and the residual has 61 d.f., that is 30.5 for each of the two residual variances.

A rough estimate of a confidence interval for a sd with f d.f. is the sd estimate multiplied by $(\sqrt{f/\chi^2_{0.025}(f)}, \sqrt{f/\chi^2_{0.975}(f)})$[1]

Inspection of the variance components (the lower part of the output from each type of measurement), we see that the coefficient of variation for measurements of subcutaneous fat is 4.2% and 6.1% for the two sonographers, whereas they for visceral fat are 3.4% and 4.0%.

```
> ( erf.mxi <- c( 1, sqrt( 29 / qchisq(c(0.975,0.025),29) ) ) )
  [1] 1.0000000 0.7964069 1.3443152
> ( erf.res <- c( 1, sqrt( 30.5 / qchisq(c(0.975,0.025),30.5) ) ) )
  [1] 1.0000000 0.8004213 1.3330231
```

Thus approximate confidence intervals for the c.v. of the subcutaneous measurements is (4.2;3.3;5.6)% and (6.1;4.9;8.1)% and for the visceral measurements (3.4;2.7;4.5)% and (4.0;3.2;5.4)% for sonographer 1 and 2 respectively.

In tabular form we have for the variance components:

```
> dnam <- list( resp = c("Subc","Visc"),
+                 tr = c("Ident","Log"),
+              VCmp = c("MxI","Res1","Res2"),
+                     c("Est","lo","hi") )
> VC <- array( NA, dimnames=dnam, dim=sapply(dnam,length) )
> VC["Subc","Ident","MxI" ,] <- BA.sub$VarComp[1,"MxI"] * erf.mxi
```

[1] This is based on a heuristic argument that a (simpel) variance estimate is distributed as $\sigma^2\chi^2(f)/f$ where f is the degrees of freedom and σ is the true standard deviation.

```
> VC["Subc","Ident","Res1",] <- BA.sub$VarComp[1,"res"] * erf.res
> VC["Subc","Ident","Res2",] <- BA.sub$VarComp[2,"res"] * erf.res
> VC["Subc","Log"  ,"MxI" ,] <- BA.lsub$VarComp[1,"MxI"] * erf.mxi
> VC["Subc","Log"  ,"Res1",] <- BA.lsub$VarComp[1,"res"] * erf.res
> VC["Subc","Log"  ,"Res2",] <- BA.lsub$VarComp[2,"res"] * erf.res
> VC["Visc","Ident","MxI" ,] <- BA.vis$VarComp[1,"MxI"] * erf.mxi
> VC["Visc","Ident","Res1",] <- BA.vis$VarComp[1,"res"] * erf.res
> VC["Visc","Ident","Res2",] <- BA.vis$VarComp[2,"res"] * erf.res
> VC["Visc","Log"  ,"MxI" ,] <- BA.lvis$VarComp[1,"MxI"] * erf.mxi
> VC["Visc","Log"  ,"Res1",] <- BA.lvis$VarComp[1,"res"] * erf.res
> VC["Visc","Log"  ,"Res2",] <- BA.lvis$VarComp[2,"res"] * erf.res
> round( ftable( VC, row.vars=c(1,3) ), 3 )
            tr Ident               Log
                Est    lo    hi   Est    lo    hi
  resp VCmp
  Subc MxI    0.140 0.111 0.188 0.095 0.076 0.128
       Res1   0.103 0.082 0.137 0.042 0.033 0.056
       Res2   0.106 0.085 0.141 0.061 0.049 0.081
  Visc MxI    0.342 0.272 0.459 0.040 0.032 0.054
       Res1   0.271 0.217 0.361 0.034 0.027 0.045
       Res2   0.233 0.187 0.311 0.040 0.032 0.054
```

1.4 Modelling the variation between original replicates

Recall that the method of measurement is *defined* as taking the average of two measurements on the patient, lying in the same position (i.e. without getting up). These avarages formed the basis for the previous analyses.

If we consider the original data, these will be classified additionally by a repeat measurement of the patient in the same position, say s, so our model for these data would be:

$$y_{oprs} = \mu_p + \alpha_o + a_{op} + e_{opr} + f_{oprs}, \quad a_{op} \sim \mathcal{N}(0, \tau^2), \quad e_{opr} \sim \mathcal{N}(0, \sigma_o^2), \quad f_{oprs} \sim \mathcal{N}(0, \omega_o^2)$$

In this notation r changes whenever the person (examininee, p) stands up and lies down again (`Skema`), whereas s changes when a repeat measurement is taken on the person lying in the same position (`Maaling`).

In the original dataset we do not have the replicate measurements as a separate variable it is merely the interaction (cross-classification) of the variable `Skema` and `Maaling`.

The model is fitted using the original dataset with 8 measurements for each persons. The original dataset is first made into two `Meth` objects, where we keep the original variables too, specifically we need the `Skema`, which is the original replication (between stand-ups), whereas the new extra replication is the interaction between `Skema` and `Maaling` (strictly speaking `Maaling` *within* `Skema`).

```
> str( aphi )
  'data.frame':        240 obs. of  7 variables:
   $ ptno    : num  1 1 1 1 1 1 1 1 2 2 ...
   $ subkutan: num  3.77 3.85 3.56 3.82 3.72 3.41 3.52 4 2.32 2.49 ...
   $ viseral : num  9.48 9.33 9.12 8.74 8.89 8.85 8.9 8.85 8.36 9.45 ...
   $ operator: Factor w/ 2 levels "Son.1","Son.2": 1 1 1 1 2 2 2 2 1 1 ...
   $ skema   : num  1 1 2 2 1 1 2 2 1 1 ...
   $ maaling : num  1 2 1 2 1 2 1 2 1 2 ...
   $ sex     : Factor w/ 2 levels "M","F": 2 2 2 2 2 2 2 2 1 1 ...
```

```
> subs <- Meth( transform( aphi, repl=interaction(skema,maaling) ),
+               meth="operator", item="ptno", repl="repl", y="subkutan", keep.vars=T )
  The following variables from the dataframe
  "transform(aphi, repl = interaction(skema, maaling))" are used as the Meth variables:
  meth: operator
  item: ptno
  repl: repl
     y: subkutan
          #Replicates
  Method           4 #Items #Obs: 240 Values:  min   med  max
    Son.1         30     30       120         0.36 2.240 5.05
    Son.2         30     30       120         0.43 2.295 5.44
> vics <- Meth( transform( aphi, repl=interaction(skema,maaling) ),
+               meth="operator", item="ptno", repl="repl", y="viseral", keep.vars=T )
  The following variables from the dataframe
  "transform(aphi, repl = interaction(skema, maaling))" are used as the Meth variables:
  meth: operator
  item: ptno
  repl: repl
     y: viseral
          #Replicates
  Method      3     4 #Items #Obs: 239 Values:  min   med   max
    Son.1     1    29     30       119         2.84 6.550 12.44
    Son.2     0    30     30       120         2.98 7.055 13.46
```

This would be a model where the extra residual variance components, ω_o represented the observers' ability to reproduce measurements on a patient lying in the same position (*i.e.* without getting up). Also note that the variation between two measurements by observer o on a person between getting up has a standard deviation of $\sqrt{\sigma_o^2 + \omega_o^2}$.

This model is not nicely packed in the `MethComp` package, so we must do it the hard way; first with the subcutaneous measurements:

```
> msub <- lme( y ~ item - 1 + meth,
+                random = list( item = pdIdent( ~ meth-1 ),
+                               skema = pdDiag ( ~ meth-1 ) ),
+               weights = varIdent( form = ~1 | meth ),
+               control = lmeControl(returnObject=TRUE),
+                  data = subs )
> msub
  Linear mixed-effects model fit by REML
    Data: subs
    Log-restricted-likelihood: 65.94016
    Fixed: y ~ item - 1 + meth
      item1     item2     item3     item4     item5     item6     item7     item8
  3.6423310 2.5869942 1.7341109 1.6322237 4.9825615 1.6764111 2.7032519 0.4077301
      item9    item10    item11    item12    item13    item14    item15    item16
  3.3236198 2.0252761 3.9106134 1.7504754 2.4835427 4.4158127 1.7381441 1.7993711
     item17    item18    item19    item20    item21    item22    item23    item24
  1.0337569 1.6269017 1.5491718 3.4410427 1.7517944 2.9495862 3.7144477 3.0345862
     item25    item26    item27    item28    item29    item30 methSon.2
  3.0751535 4.2580219 0.9405828 0.9684200 1.5371777 2.6243864 0.1291667

  Random effects:
   Formula: ~meth - 1 | item
   Structure: Multiple of an Identity
          methSon.1 methSon.2
  StdDev: 0.1399351 0.1399351

   Formula: ~meth - 1 | skema %in% item
   Structure: Diagonal
           methSon.1  methSon.2  Residual
  StdDev: 0.04406451 0.05936473 0.1311011
```

```
  Variance function:
   Structure: Different standard deviations per stratum
   Formula: ~1 | meth
   Parameter estimates:
     Son.1    Son.2
  1.000000 0.941926
  Number of Observations: 240
  Number of Groups:
             item skema %in% item
               30               60
```

This output is a bit confusing and difficult to use, so instead we fish out the relevant variance components using `VarCorr` and a somewhat arcane machinery to get the residual variances by method. We also need the `ci.lin` function from the `Epi` package to extract the bias between the observers (which is bound to be almost the same as from the mean-based analyses):

```
> library( Epi )
> ci.lin( msub, subset="meth" )[,1]
  [1] 0.1291667
> ( vc.msub <- VarCorr( msub ) )
            Variance          StdDev
  item =    pdIdent(meth - 1)
  methSon.1 0.019581828       0.13993509
  methSon.2 0.019581828       0.13993509
  skema =   pdDiag(meth - 1)
  methSon.1 0.001941681       0.04406451
  methSon.2 0.003524171       0.05936473
  Residual  0.017187491       0.13110107
> vc.sub <- as.numeric(vc.msub[grep("methSon",rownames(vc.msub)),"StdDev"])
> om.sub <- attr(msub$residuals,"std")
> ( om.sub <- tapply( om.sub, names(om.sub), unique ) )
      Son.1     Son.2
  0.1311011 0.1234875
```

Actually, we may package it all in a convenience function:

```
> vc.get <-
+ function( obj )
+ {
+ vc <- VarCorr( obj )
+ vc <- as.numeric(vc[grep("methSon",rownames(vc)),"StdDev"])
+ om <- attr(obj$residuals,"std")
+ om <- tapply( om, names(om), unique )
+ res <- c( vc[2:4], om, sqrt( vc[3:4]^2 + om^2 ) )
+ names( res ) <- c("tau","sig1","sig2","om1","om2","rep1","rep2")
+ res
+ }
> vc.get( msub )
        tau       sig1       sig2        om1        om2       rep1       rep2
  0.13993509 0.04406451 0.05936473 0.13110107 0.12348751 0.13830825 0.13701583
> round( vc.sub <- vc.get(msub), 3 )
    tau  sig1  sig2   om1   om2  rep1  rep2
  0.140 0.044 0.059 0.131 0.123 0.138 0.137
```

We then do the same with the visceral data, using the update mechanism and the extraction tools:

```
> mvis <- update( msub, data = vics )
> round( vc.vis <- vc.get(mvis), 3 )
    tau  sig1  sig2   om1   om2  rep1  rep2
  0.333 0.146 0.170 0.327 0.225 0.359 0.282
```

We can easily expand to the log-scale as well with the update mechanism too:

```
> lsub <- update( msub, log(y) ~ . )
> round( vc.lsub <- vc.get(lsub), 3 )
    tau  sig1  sig2   om1   om2  rep1  rep2
  0.094 0.000 0.046 0.060 0.058 0.060 0.074
```

```
> lvis <- update( mvis, log(y) ~ . )
> round( vc.lvis <- vc.get(lvis), 3 )
    tau  sig1  sig2   om1   om2  rep1  rep2
  0.040 0.012 0.035 0.045 0.030 0.046 0.046
```

All these variance component estimates can now be arranged in a table, see table 1.1. There is quite substantial difference in the residual coefficient of variation as estimated from the original data and that from the mean data. This is because the analyses of the mean data are based on logarithms of the means, and not the means of the logarithmns

The general conclusion is based on these analyses that on the absolute scale the subcutaneous measurements are more reproducible, with a coefficient of repeatability of

Table 1.1: *Table of variance components estimated from the original data and from the mean data. "Mean" refer to analyses of the averages of two measurements (without stand-up), "Original" to analysis of the singe measurements. The residual variation for the "Original" data is computed as the square root of the sum of the between and within stand-ups variances.*

Measurement		Linear scale (cm)		Log scale (c.v. (%))	
Variance component		Mean	Original	Mean	Original
Subcutaneous					
τ	observer×patient	0.140	0.140	9.5	9.4
σ_1	between stand-ups (son.1)		0.044		0.0
σ_N	between stand-ups (son.2)		0.059		4.6
ω_1	within stand-ups (son.1)		0.131		6.0
ω_2	within stand-ups (son.2)		0.123		5.8
$\sqrt{\sigma_1^2+\omega_1^2}$	replication (son.1)	0.103	0.138	4.2	6.0
$\sqrt{\sigma_2^2+\omega_2^2}$	replication (son.2)	0.106	0.137	6.1	13.7
Visceral					
τ	observer×patient	0.342	0.333	4.0	4.0
σ_1	between stand-ups (son.1)		0.146		1.2
σ_2	between stand-ups (son.2)		0.170		3.5
ω_1	within stand-ups (son.1)		0.327		4.5
ω_2	within stand-ups (son.2)		0.225		3.0
$\sqrt{\sigma_1^2+\omega_1^2}$	replication (son.1)	0.271	0.359	3.4	4.6
$\sqrt{\sigma_2^2+\omega_2^2}$	replication (son.2)	0.233	0.282	4.0	4.6

$2.8 \times \sqrt{0.138^2 + 0.137^2} = 0.55$ cm, i.e. the prediction interval for the difference between two measurements of subcutaneous fat is ± 0.55 cm. For Visceral fat the coefficient of reproducibility is 1.42 cm.

1.5 Conclusion

For subcutaneous fat, the within sonographer c.v. was about 5% and for visceral fat measurements it was about 3.5%. When we include the variation around person-means too (the `MxI`-variance component), we obtain limits of agreement on the relative scale between sonographer 1 and 2 of $(-22.5; 43.0)$% for subcutaneous measurements and $(-11.0; 21.4)$% for visceral measurements. In broad terms, subcutaneous measurements are within 40% of each other and visceral within 20%.

For the absolute measurements the picture was the opposite: The variations for the subcutaneous fat were smaller than for the visceral fat.

Finally we produce the Bland-Altman plots for the two types of measurement side-by-side:

```
> par( mfrow=c(1,2), mar=c(3,3,1,3), mgp=c(3,1,0)/1.6 )
> BA.plot( mean(sub), mult=TRUE )
> BA.plot( mean(vis), mult=TRUE )
```

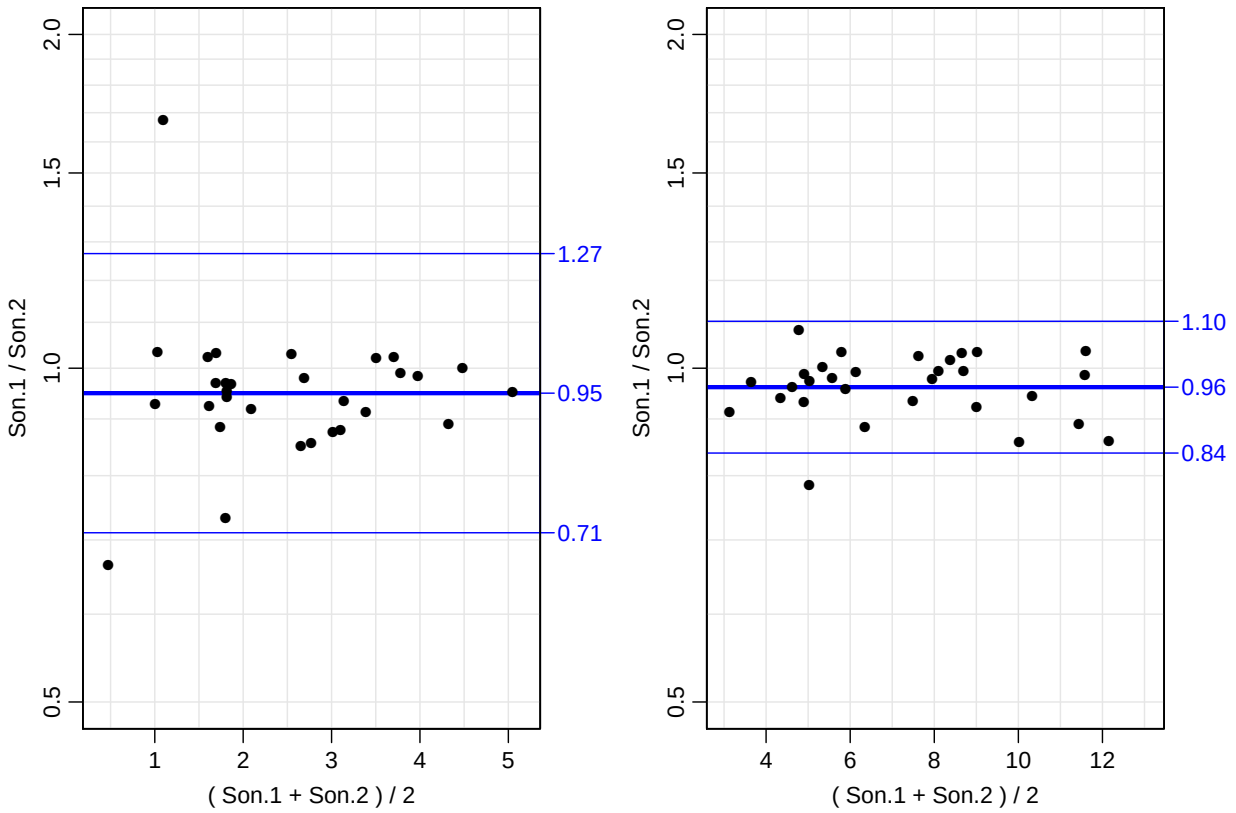

Figure 1.5: *Bland-Altman plots of the ratios (that is, log-transformed data) for subcutaneous (left) and visceral (right) fat measurements.*

1.6 Comparative measure

As a comparative measure, we will take a look at the ratio of the visceral to the subcutaneous fat measurements.

Since this is a relative measure, it is only meaningful to to analyze the ratio of these between meals — if we looked at the differences we would get substantially different results if we considered the subcutaneous to visceral ratio. Analysis on the log scale basically only changes sign if the ratio is inverted.

Since the dataset already consistes of means over the replicates we can directly compute the ratio of these:

```
> head( vis )
```

```
    meth item repl     y subkutan
1 Son.1    1    1 9.405    3.810
2 Son.1    1    2 8.930    3.690
3 Son.2    1    1 8.870    3.565
4 Son.2    1    2 8.875    3.760
5 Son.1    2    1 8.905    2.405
6 Son.1    2    2 9.615    2.470
```

```
> vs <- transform( vis, y = y/subkutan )
> vs <- Meth( vs )
```

```
  The following variables from the dataframe
  "vs" are used as the Meth variables:
  meth: meth
  item: item
  repl: repl
     y: y
          #Replicates
  Method      1     2 #Items #Obs: 119 Values:  min      med      max
    Son.1     1    29     30        59     1.362007 2.937710 13.26667
    Son.2     0    30     30        60     1.361878 2.910514 13.31343
```

Once we have created a `Meth` object with the ratio as measurement we can make the simple analyses to see if the assumptions behind LoA are fulfilled.

```
> DA.reg( vs, Tr="log" )
```

```
  Note: Response transformed by:  .Primitive("log")


   Conversion between methods:
                alpha   beta sd.pred beta=1 int(t-f) slope(t-f) sd(t-f) int(sd) slope(sd)   sd=K
  To:   From:
  Son.1 Son.1   0.000  1.000      NA     NA    0.000      0.000      NA      NA        NA     NA
        Son.2  -0.012  1.024   0.177  0.573   -0.012      0.024   0.175   0.027     0.106  0.005
  Son.2 Son.1   0.012  0.977   0.173  0.573    0.012     -0.024  -0.175   0.027     0.106  0.005
        Son.2   0.000  1.000      NA     NA    0.000      0.000      NA      NA        NA     NA
```

We see that the basic assumptions behind the LoA seem not to be fulfilled, as the SD is increasing, which is also apparent from the Bland-Altman plots:

```
> BA.plot( vs, mult=TRUE )
```

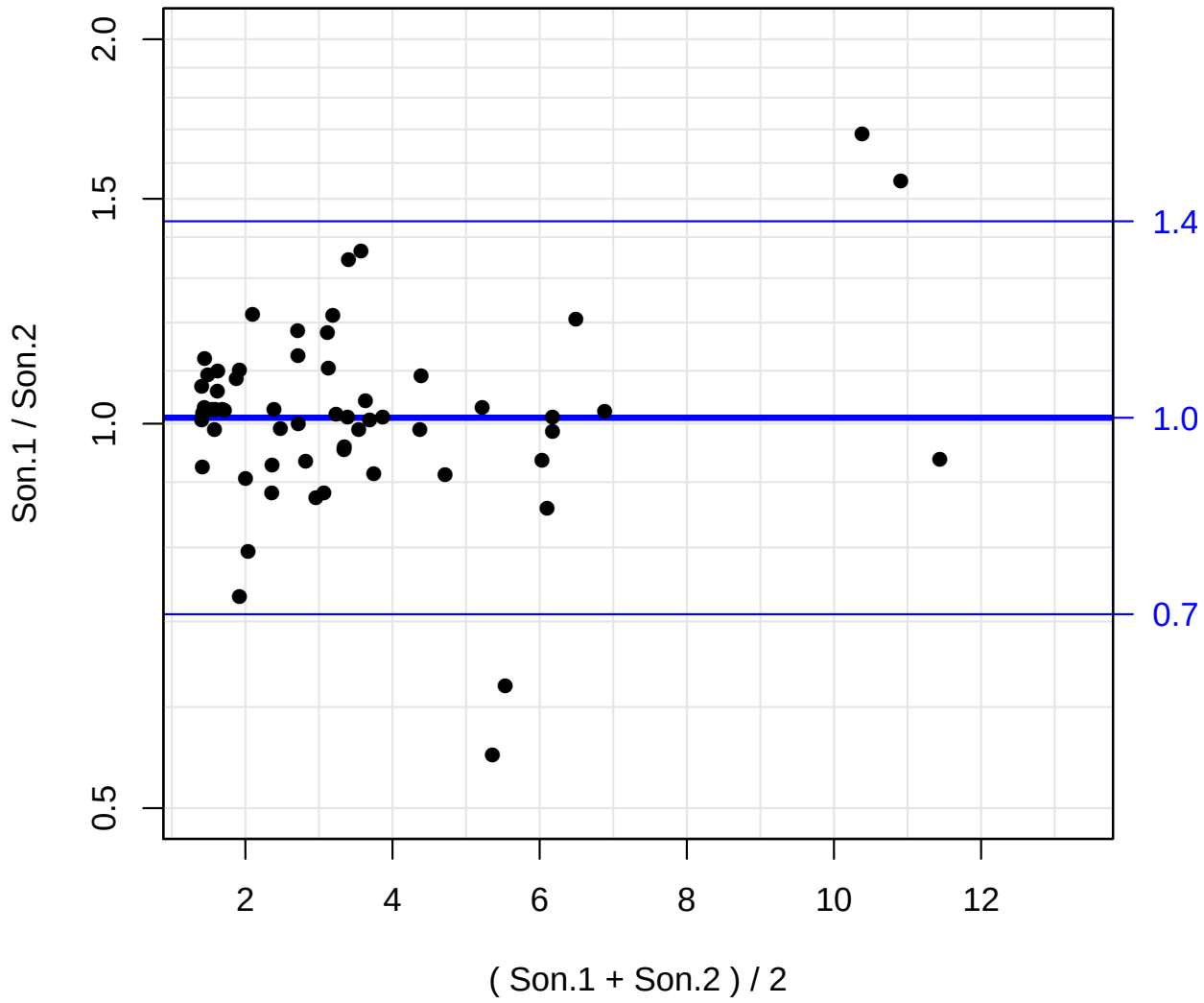

Figure 1.6: *Bland-Altman plots based on the visceral to subcutaneous ratio. The plots assess the reproducibility of the V/S ratio. Clearly, the increasing SD is attributabe to a few outliers.*

Chapter 2

Day to day variation

2.1 Data input

Data is prepared from SAS-files from the Addition project; the printout of the sas-log and -output that generated the data is in the data section **??**

Having created the SAS-export dataset, we read this by R:

```
> options( width=120 )
> library( foreign )
> library( MethComp )
> print( sessionInfo(), l=F )
  R version 3.1.2 (2014-10-31)
  Platform: i386-w64-mingw32/i386 (32-bit)

  attached base packages:
  [1] utils     datasets  graphics  grDevices stats     methods   base

  other attached packages:
  [1] Epi_1.1.68     MethComp_1.22  nlme_3.1-118   foreign_0.8-61

  loaded via a namespace (and not attached):
  [1] grid_3.1.2      lattice_0.20-29
> d2d <- read.xport( "./data/d2d.xpt" )
> names( d2d ) <- tolower( names( d2d ) )
> d2d <- transform( d2d, gap=ave( gap,
+                                 id,
+                                FUN=function(x) mean(x,na.rm=TRUE) ) )
> d2d <- transform( d2d, lng = factor( gap>35, labels=c("<=35",">35") ) )
> str( d2d )
  'data.frame':        66 obs. of  38 variables:
   $ id      : num  3465 3465 4212 4212 5014 ...
   $ valideri: num  0 1 0 1 0 1 0 1 0 1 ...
   $ label_no: num  10336 10417 10408 10446 10354 ...
   $ date_fil: num  18274 18323 18316 18337 18288 ...
   $ waist1  : num  94.5 93 70.5 71.5 97.5 92.7 97.5 93 99 99 ...
   $ waist2  : num  94 93.5 71 71.5 97 93 98 94 99 99 ...
   $ waist3  : num  NA NA NA NA NA NA NA NA NA NA ...
   $ hip1    : num  98 94.2 98 90 99 95.7 99 93 103 103 ...
   $ hip2    : num  99 94 98.5 90 49 ...
   $ hip3    : num  NA NA NA NA NA NA NA NA NA NA ...
   $ height  : num  166 166 160 160 179 ...
   $ weight  : num  71.1 72 51 50.4 81.8 81.8 76.7 75.5 94 95.2 ...
   $ impedanc: num  527 527 669 740 542 548 464 444 388 375 ...
   $ fat_perc: num  27.1 29.3 28.7 30.5 27.1 28 25.3 24 23.9 23.7 ...
   $ au_m1_vi: num  1 1 1 1 1 1 1 1 1 1 ...
   $ au_m1_v2: num  9.46 8.24 5.49 4.38 8.89 ...
```

```
 $ au_m1_v3: num  NA NA NA NA NA NA NA NA NA NA ...
 $ au_m1_su: num  1 1 1 1 1 1 1 1 1 1 ...
 $ au_m1_s2: num  1.59 0.94 1.46 1.46 2.54 2.49 3.58 3.55 1.77 2.06 ...
 $ au_m1_s3: num  NA NA NA NA NA NA NA NA NA NA ...
 $ au_m1_li: num  1 1 1 1 1 1 1 1 1 1 ...
 $ au_m1_l2: num  NA NA NA NA NA NA NA NA NA NA ...
 $ au_m2_vi: num  1 1 1 1 1 1 1 1 1 1 ...
 $ au_m2_v2: num  8.99 8.3 5.16 4.54 8.83 ...
 $ au_m2_v3: num  NA NA NA NA NA NA NA NA NA NA ...
 $ au_m2_su: num  1 1 1 1 1 1 1 1 1 1 ...
 $ au_m2_s2: num  1.22 0.97 1.59 1.3 2.68 2.27 3.59 3.27 1.59 2.05 ...
 $ au_m2_s3: num  NA NA NA NA NA NA NA NA NA NA ...
 $ au_m2_li: num  1 1 1 1 1 1 1 1 1 1 ...
 $ au_m2_l2: num  NA NA NA NA NA NA NA NA NA NA ...
 $ au_initi: Factor w/ 3 levels "","APHI","NABJ": 3 1 3 3 2 1 3 1 2 1 ...
 $ repeat_v: num  1 1 1 1 1 1 1 1 1 1 ...
 $ repeat_2: num  1 1 1 1 1 1 1 1 1 1 ...
 $ gap     : num  49 49 21 21 35 35 14 14 35 35 ...
 $ sex     : num  1 1 0 0 1 1 1 1 1 1 ...
 $ letterty: Factor w/ 2 levels "Invitation of IFG/IGT",..: 2 2 2 2 2 2 2 2 2 2 ...
 $ age_scre: num  67 67 58 58 59 59 66 66 60 60 ...
 $ lng     : Factor w/ 2 levels "<=35",">35": 2 2 1 1 1 1 1 1 1 1 ...
```

```
> head( d2d )
```

```
     id valideri label_no date_fil waist1 waist2 waist3 hip1 hip2 hip3 height weight impedanc fat_p
1 3465        0    10336    18274   94.5   94.0      NA 98.0 99.0   NA  166.5   71.1      527     2
2 3465        1    10417    18323   93.0   93.5      NA 94.2 94.0   NA  165.6   72.0      527     2
3 4212        0    10408    18316   70.5   71.0      NA 98.0 98.5   NA  159.9   51.0      669     2
4 4212        1    10446    18337   71.5   71.5      NA 90.0 90.0   NA  160.3   50.4      740     3
5 5014        0    10354    18288   97.5   97.0      NA 99.0 49.0   NA  179.1   81.8      542     2
6 5014        1    10418    18323   92.7   93.0      NA 95.7 95.5   NA  178.0   81.8      548     2
  au_m1_v3 au_m1_su au_m1_s2 au_m1_s3 au_m1_li au_m1_l2 au_m2_vi au_m2_v2 au_m2_v3 au_m2_su au_m2_
1       NA        1     1.59       NA        1       NA        1     8.99       NA        1     1.
2       NA        1     0.94       NA        1       NA        1     8.30       NA        1     0.
3       NA        1     1.46       NA        1       NA        1     5.16       NA        1     1.
4       NA        1     1.46       NA        1       NA        1     4.54       NA        1     1.
5       NA        1     2.54       NA        1       NA        1     8.83       NA        1     2.
6       NA        1     2.49       NA        1       NA        1     9.21       NA        1     2.
  au_m2_l2 au_initi repeat_v repeat_2 gap sex                   letterty age_scre  lng
1       NA     NABJ        1        1  49   1 Invitation of NGT/Low risk       67  >35
2       NA                 1        1  49   1 Invitation of NGT/Low risk       67  >35
3       NA     NABJ        1        1  21   0 Invitation of NGT/Low risk       58 <=35
4       NA     NABJ        1        1  21   0 Invitation of NGT/Low risk       58 <=35
5       NA     APHI        1        1  35   1 Invitation of NGT/Low risk       59 <=35
6       NA                 1        1  35   1 Invitation of NGT/Low risk       59 <=35
```

There are two types of measurement of interest, subcutaneous and visceral.

2.2 Subcutaneous

We first make a `Meth` object with subcutaneous fat as response variable:

```
> subc <- with( d2d, rbind( data.frame( item = id,
+                                       meth = valideri,
+                                       repl = 1,
+                                          y = au_m1_s2,
+                                        sex = sex,
+                                        gap = gap,
+                                        lng = lng ),
+                           data.frame( item = id,
+                                       meth = valideri,
+                                       repl = 2,
+                                          y = au_m2_s2,
```

```
+                                          sex = sex,
+                                          gap = gap,
+                                          lng = lng ) ) )
> subc$sex <- factor( 2-subc$sex, labels=c("M","F") )
> subc <- Meth( subc, keep.var=TRUE )
  The following variables from the dataframe
  "subc" are used as the Meth variables:
  meth: meth
  item: item
  repl: repl
     y: y
         #Replicates
  Method          2 #Items #Obs: 132 Values:  min   med  max
       0         33     33        66         0.66 2.465 8.41
       1         33     33        66         0.84 2.245 5.71

> levels( subc$meth ) <- paste( "Day", 1:2 )
> str( subc )

  Classes 'Meth' and 'data.frame':        132 obs. of  7 variables:
   $ meth: Factor w/ 2 levels "Day 1","Day 2": 1 2 1 2 1 2 1 2 1 2 ...
   $ item: Factor w/ 33 levels "3465","4212",..: 1 1 2 2 3 3 4 4 5 5 ...
   $ repl: Factor w/ 2 levels "1","2": 1 1 1 1 1 1 1 1 1 1 ...
   $ y   : num  1.59 0.94 1.46 1.46 2.54 2.49 3.58 3.55 1.77 2.06 ...
   $ sex : Factor w/ 2 levels "M","F": 1 1 2 2 1 1 1 1 1 1 ...
   $ gap : num  49 49 21 21 35 35 14 14 35 35 ...
   $ lng : Factor w/ 2 levels "<=35",">35": 2 2 1 1 1 1 1 1 1 1 ...

> summary( subc )

         #Replicates
  Method          2 #Items #Obs: 132 Values:  min   med  max
    Day 1        33     33        66         0.66 2.465 8.41
    Day 2        33     33        66         0.84 2.245 5.71

> # Fix a blooper in data:
> subset.data.frame( subc, item=="59662" )

      meth  item repl    y sex gap lng
  47  Day 1 59662    1 8.41   M  42 >35
  48  Day 2 59662    1 3.41   M  42 >35
  113 Day 1 59662    2 3.43   M  42 >35
  114 Day 2 59662    2 3.39   M  42 >35

> subc$y[abs(subc$y-8.41)<0.001] <- NA
> subc <- Meth( subc )

  The following variables from the dataframe
  "subc" are used as the Meth variables:
  meth: meth
  item: item
  repl: repl
     y: y
         #Replicates
  Method      1     2 #Items #Obs: 131 Values:  min   med  max
    Day 1     1    32     33        65         0.66 2.460 6.10
    Day 2     0    33     33        66         0.84 2.245 5.71

> subset.data.frame( subc, item=="59662" )

      meth  item repl    y sex gap lng
  47  Day 2 59662    1 3.41   M  42 >35
  112 Day 1 59662    2 3.43   M  42 >35
  113 Day 2 59662    2 3.39   M  42 >35
```

We can the make Bland-Altman plots, both on the original scale and on the log-scale. However we use the mean of the two replicate measurements, as obtained by `mean.Meth`:

```
> par( mfrow=c(1,2), mar=c(5,5,1,1) )
> BA.plot( subc, repl.conn=TRUE, col.points="gray",
+          axlim=c(0.5,6.5), eqax=TRUE, col.lines="gray" )
> par( new=TRUE )
> BA.plot( mean(subc,simplify=TRUE),
+          axlim=c(0.5,6.5), eqax=TRUE )
> BA.plot( subc, repl.conn=TRUE, col.points="gray",
+          axlim=c(0.5,6.5), eqax=TRUE, col.lines="gray",
+          Transform="log" )
> par( new=TRUE )
> BA.plot( mean(subc,simplify=TRUE),
+          axlim=c(0.5,6.5), eqax=TRUE,
+          Transform="log" )
```

A final quick overview of data is obtained from regressing the differences on the means:

```
> summary( subc )
         #Replicates
Method       1     2 #Items #Obs: 131 Values:  min   med  max
  Day 1      1    32     33        65         0.66 2.460 6.10
  Day 2      0    33     33        66         0.84 2.245 5.71

> summary( mean(subc) )
         #Replicates
Method           1 #Items #Obs: 66 Values:  min   med   max
  Day 1         33     33       33         0.74 2.465 6.060
  Day 2         33     33       33         0.87 2.225 5.615

> DA.reg( subc )
 Conversion between methods:
              alpha   beta sd.pred beta=1 int(t-f) slope(t-f) sd(t-f) int(sd) slope(sd)   sd=K
To:   From:
Day 1 Day 1   0.000  1.000      NA     NA    0.000      0.000      NA      NA        NA     NA
      Day 2  -0.048  1.067   0.359  0.095   -0.046      0.065   0.348   0.287     0.023  0.410
Day 2 Day 1   0.045  0.937   0.337  0.095    0.046     -0.065  -0.348   0.287     0.023  0.410
      Day 2   0.000  1.000      NA     NA    0.000      0.000      NA      NA        NA     NA

> DA.reg( mean(subc) )
```

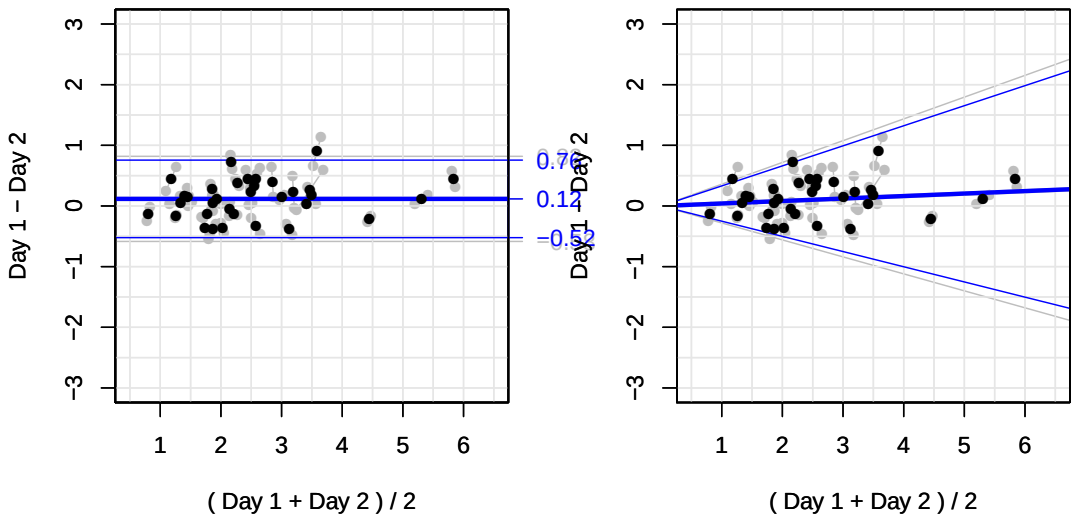

Figure 2.1: *Bland-Altman plots for original scale and for relative scale for replicates and means of the subcutaneous measurements*

```
   Conversion between methods:
                alpha   beta sd.pred beta=1 int(t-f) slope(t-f) sd(t-f) int(sd) slope(sd)   sd=K
  To:   From:
  Day 1 Day 1   0.000  1.000      NA     NA    0.000      0.000      NA      NA        NA     NA
        Day 2  -0.040  1.063   0.327  0.223   -0.039      0.062   0.317   0.290     0.006  0.864
  Day 2 Day 1   0.038  0.940   0.307  0.223    0.039     -0.062  -0.317   0.290     0.006  0.864
        Day 2   0.000  1.000      NA     NA    0.000      0.000      NA      NA        NA     NA
> DA.reg( mean(subc), Transform="log" )
  Note: Response transformed by:  .Primitive("log")


   Conversion between methods:
                alpha   beta sd.pred beta=1 int(t-f) slope(t-f) sd(t-f) int(sd) slope(sd)   sd=K
  To:   From:
  Day 1 Day 1   0.000  1.000      NA     NA    0.000      0.000      NA      NA        NA     NA
        Day 2   0.010  1.038   0.151  0.542    0.010      0.037   0.148   0.225    -0.101  0.025
  Day 2 Day 1  -0.010  0.964   0.145  0.542   -0.010     -0.037  -0.148   0.225    -0.101  0.025
        Day 2   0.000  1.000      NA     NA    0.000      0.000      NA      NA        NA     NA
```

To see if we have non-constant bias or non-constant SD we make plots using these options:

```
> par( mfcol=c(3,2), mar=c(5,5,1,1) )
> BA.plot( mean(subc),
+          axlim=c(0.5,6.5), eqax=TRUE, eqn=TRUE, pl.type="conv" )
  Relationships between methods:
   Day 1-Day 2 = 0.12 (0.32)
   Day 1 = 0.12+Day 2 (0.32)
   Day 2 = -0.12+Day 1 (0.32)
> BA.plot( mean(subc),
+          axlim=c(0.5,6.5), eqax=TRUE,
+          dif.type="lin", eqn=TRUE, pl.type="conv" )
  Relationships between methods:
   Day 1-Day 2 = -0.04+0.06(Day 1+Day 2)/2 (0.32)
   Day 1 = -0.04+1.06Day 2 (0.33)
   Day 2 = 0.04+0.94Day 1 (0.31)
> BA.plot( mean(subc),
+          axlim=c(0.5,6.5), eqax=TRUE,
+          dif.type="lin", sd.type="lin", eqn=TRUE, pl.type="conv" )
  Relationships between methods:
   Day 1-Day 2 = -0.04+0.06(Day 1+Day 2)/2 (0.29+0.01Avg.)
   Day 1 = -0.04+1.06Day 2 (0.30+0.01Day 2)
   Day 2 = 0.04+0.94Day 1 (-0.28-0.01Day 1)
> BA.plot( mean(subc),
+          axlim=c(0.5,6.5), eqax=TRUE, eqn=TRUE )
  Relationships between methods:
   Day 1-Day 2 = 0.12 (0.32)
   Day 1 = 0.12+Day 2 (0.32)
   Day 2 = -0.12+Day 1 (0.32)
> BA.plot( mean(subc),
+          axlim=c(0.5,6.5), eqax=TRUE,
+          dif.type="lin", eqn=TRUE )
  Relationships between methods:
   Day 1-Day 2 = -0.04+0.06(Day 1+Day 2)/2 (0.32)
   Day 1 = -0.04+1.06Day 2 (0.33)
   Day 2 = 0.04+0.94Day 1 (0.31)
> BA.plot( mean(subc),
+          axlim=c(0.5,6.5), eqax=TRUE,
+          dif.type="lin", sd.type="lin", eqn=TRUE )
  Relationships between methods:
   Day 1-Day 2 = -0.04+0.06(Day 1+Day 2)/2 (0.29+0.01Avg.)
   Day 1 = -0.04+1.06Day 2 (0.30+0.01Day 2)
   Day 2 = 0.04+0.94Day 1 (-0.28-0.01Day 1)
```

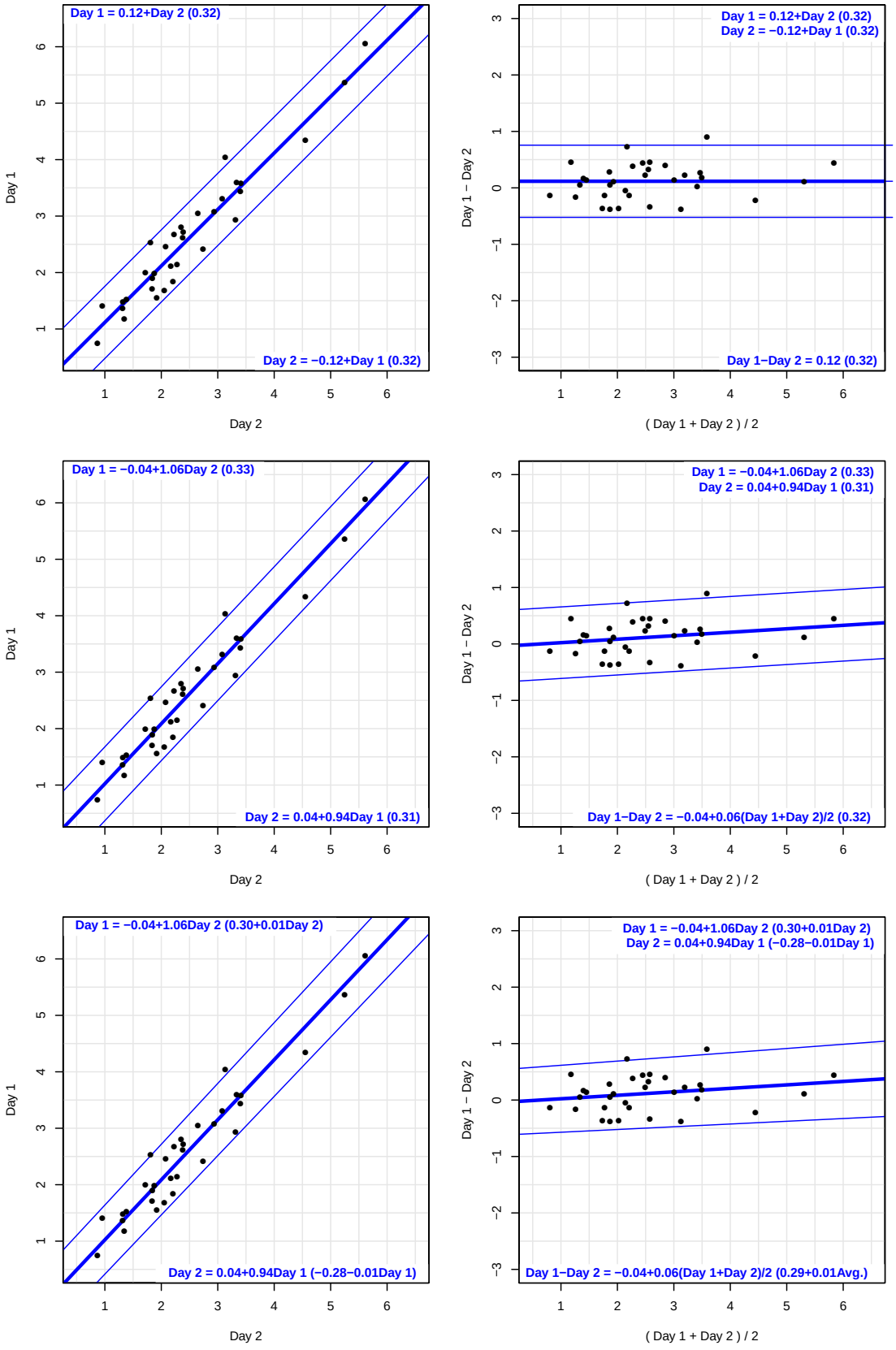

Figure 2.2: *Bland-Altman and conversion plots for mean of the subcutaneous measurements. All models are based on regression of differences on means. Top to bottom is a model with constant difference and constant s.d.; linear difference and constant s.d. ; linear difference, linear s.d.*

2.3 Visceral

This is basically a repeat of the subcutaneous exercise:

```
> visc <- with( d2d, rbind( data.frame( item = id,
+                                       meth = valideri,
+                                       repl = 1,
+                                          y = au_m1_v2,
+                                        sex = sex,
+                                        gap = gap,
+                                        lng = lng ),
+                           data.frame( item = id,
+                                       meth = valideri,
+                                       repl = 2,
+                                          y = au_m2_v2,
+                                        sex = sex,
+                                        gap = gap,
+                                        lng = lng ) ) )
> visc$sex <- factor( 2-visc$sex, labels=c("M","F") )
> str( visc )
  'data.frame':        132 obs. of  7 variables:
   $ item: num  3465 3465 4212 4212 5014 ...
   $ meth: num  0 1 0 1 0 1 0 1 0 1 ...
   $ repl: num  1 1 1 1 1 1 1 1 1 1 ...
   $ y   : num  9.46 8.24 5.49 4.38 8.89 ...
   $ sex : Factor w/ 2 levels "M","F": 1 1 2 2 1 1 1 1 1 1 ...
   $ gap : num  49 49 21 21 35 35 14 14 35 35 ...
   $ lng : Factor w/ 2 levels "<=35",">35": 2 2 1 1 1 1 1 1 1 1 ...

> visc <- Meth( visc )

  The following variables from the dataframe
  "visc" are used as the Meth variables:
  meth: meth
  item: item
  repl: repl
     y: y
         #Replicates
  Method          2 #Items #Obs: 132 Values:  min   med   max
       0         33     33        66         3.12 8.165 14.77
       1         33     33        66         4.08 7.700 13.07

> levels( visc$meth ) <- paste( "Day", 1:2 )
> summary( visc )

           #Replicates
  Method           2 #Items #Obs: 132 Values:  min   med   max
    Day 1         33     33        66         3.12 8.165 14.77
    Day 2         33     33        66         4.08 7.700 13.07

> head( visc )

     meth item repl    y sex gap  lng
  1 Day 1 3465    1 9.46   M  49  >35
  2 Day 2 3465    1 8.24   M  49  >35
  3 Day 1 4212    1 5.49   F  21 <=35
  4 Day 2 4212    1 4.38   F  21 <=35
  5 Day 1 5014    1 8.89   M  35 <=35
  6 Day 2 5014    1 9.43   M  35 <=35
```

We can then make a Bland-Altman plot:

```
> par( mfrow=c(1,2), mar=c(5,5,1,1) )
> BA.plot( visc, repl.conn=TRUE, col.points="gray",
+          axlim=c(2,14), eqax=TRUE, col.lines="gray" )
> par( new=TRUE )
> BA.plot( mean.Meth(visc,simplify=TRUE),
+          axlim=c(2,14), eqax=TRUE )
```

```
> BA.plot( visc, repl.conn=TRUE, col.points="gray",
+          axlim=c(2,14), eqax=TRUE, col.lines="gray",
+          Transform="log")
> par( new=TRUE )
> BA.plot( mean.Meth(visc,simplify=TRUE),
+          axlim=c(2,14), eqax=TRUE,
+          Transform="log" )
```

```
> par( mfrow=c(1,2), mar=c(3,3,1,3), mgp=c(3,1,0)/1.6, las=1 )
> BA.plot( mean(visc), pl.type="BA",
+          axlim=c(2,14), eqax=TRUE,
+          Transform="log" )
> BA.plot( mean(visc), pl.type="BA",
+          axlim=c(2,14), eqax=TRUE,
+          mult=TRUE, diflim=2 )
```

```
> DA.reg( visc )
```

```
   Conversion between methods:
                alpha   beta sd.pred beta=1 int(t-f) slope(t-f) sd(t-f) int(sd) slope(sd)   sd=K
  To:   From:
  Day 1 Day 1   0.000  1.000      NA     NA    0.000       0.000      NA      NA         NA     NA
        Day 2  -0.904  1.140   1.172  0.020   -0.845       0.131   1.096   0.578      0.062  0.127
  Day 2 Day 1   0.793  0.877   1.028  0.020    0.845      -0.131  -1.096   0.578      0.062  0.127
        Day 2   0.000  1.000      NA     NA    0.000       0.000      NA      NA         NA     NA
```

```
> DA.reg( mean(visc) )
```

```
   Conversion between methods:
                alpha   beta sd.pred beta=1 int(t-f) slope(t-f) sd(t-f) int(sd) slope(sd)   sd=K
  To:   From:
  Day 1 Day 1   0.000  1.000      NA     NA    0.000       0.000      NA      NA         NA     NA
        Day 2  -0.904  1.140   1.155  0.098   -0.845       0.131   1.080   0.588      0.058  0.302
  Day 2 Day 1   0.793  0.877   1.013  0.098    0.845      -0.131  -1.080   0.588      0.058  0.302
        Day 2   0.000  1.000      NA     NA    0.000       0.000      NA      NA         NA     NA
```

```
> DA.reg( mean(visc), Transform="log" )
```

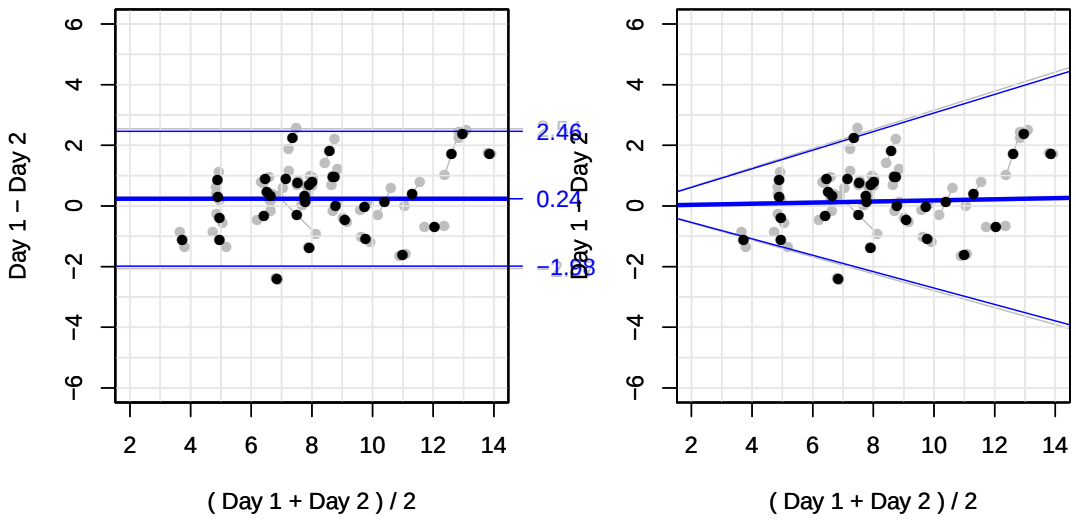

Figure 2.3: *Bland-Altman and conversion plots for original replicates and means of the visceral measurements*

```
  Note: Response transformed by:  .Primitive("log")


   Conversion between methods:
                alpha   beta sd.pred beta=1 int(t-f) slope(t-f) sd(t-f) int(sd) slope(sd)   sd=K
  To:   From:
  Day 1 Day 1   0.000  1.000      NA     NA    0.000      0.000      NA      NA        NA     NA
        Day 2  -0.275  1.143   0.152  0.108   -0.256      0.133   0.142   0.351    -0.103  0.082
  Day 2 Day 1   0.240  0.875   0.133  0.108    0.256     -0.133  -0.142   0.351    -0.103  0.082
        Day 2   0.000  1.000      NA     NA    0.000      0.000      NA      NA        NA     NA



> par( mfcol=c(3,2), mar=c(5,5,1,1) )
> BA.plot( mean(visc,simplify=TRUE),
+          axlim=c(2,14), eqax=TRUE, eqn=TRUE )

  Relationships between methods:
   Day 1-Day 2 = 0.24 (1.11)
   Day 1 = 0.24+Day 2 (1.11)
   Day 2 = -0.24+Day 1 (1.11)

> BA.plot( mean(visc,simplify=TRUE),
+          axlim=c(2,14), eqax=TRUE,
+          dif.type="lin", eqn=TRUE )

  Relationships between methods:
   Day 1-Day 2 = -0.84+0.13(Day 1+Day 2)/2 (1.08)
   Day 1 = -0.90+1.14Day 2 (1.16)
   Day 2 = 0.79+0.88Day 1 (1.01)

> BA.plot( mean(visc,simplify=TRUE),
+          axlim=c(2,14), eqax=TRUE,
+          dif.type="lin", sd.type="lin", eqn=TRUE )

  Relationships between methods:
   Day 1-Day 2 = -0.84+0.13(Day 1+Day 2)/2 (0.59+0.06Avg.)
   Day 1 = -0.90+1.14Day 2 (0.62+0.07Day 2)
   Day 2 = 0.79+0.88Day 1 (-0.56-0.05Day 1)

> BA.plot( mean(visc,simplify=TRUE),
+          axlim=c(2,14), eqax=TRUE, eqn=TRUE, pl.type="conv" )

  Relationships between methods:
   Day 1-Day 2 = 0.24 (1.11)
   Day 1 = 0.24+Day 2 (1.11)
   Day 2 = -0.24+Day 1 (1.11)

> BA.plot( mean(visc,simplify=TRUE),
+          axlim=c(2,14), eqax=TRUE,
+          dif.type="lin", eqn=TRUE, pl.type="conv" )

  Relationships between methods:
   Day 1-Day 2 = -0.84+0.13(Day 1+Day 2)/2 (1.08)
   Day 1 = -0.90+1.14Day 2 (1.16)
   Day 2 = 0.79+0.88Day 1 (1.01)

> BA.plot( mean(visc,simplify=TRUE),
+          axlim=c(2,14), eqax=TRUE,
+          dif.type="lin", sd.type="lin", eqn=TRUE, pl.type="conv" )

  Relationships between methods:
   Day 1-Day 2 = -0.84+0.13(Day 1+Day 2)/2 (0.59+0.06Avg.)
   Day 1 = -0.90+1.14Day 2 (0.62+0.07Day 2)
   Day 2 = 0.79+0.88Day 1 (-0.56-0.05Day 1)
```

2.4 Analysis of coefficients of variation

First we make an analysis that allows us to see how the coefficient of variation looks for the two types of measurement:

```
> BA.est( subc, linked=FALSE )
   Conversion between methods:
                alpha   beta sd.pred LoA-lo LoA-up
  To:   From:
  Day 1 Day 1   0.000  1.000   0.230 -0.460  0.460
        Day 2   0.117  1.000   0.351 -0.585  0.819
  Day 2 Day 1  -0.117  1.000   0.351 -0.819  0.585
        Day 2   0.000  1.000   0.172 -0.345  0.345

   Variance components (sd):
        IxR   MxI   res
  Day 1   0 0.202 0.163
  Day 2   0 0.202 0.122
> BA.est( visc, linked=FALSE )
   Conversion between methods:
                alpha   beta sd.pred LoA-lo LoA-up
  To:   From:
  Day 1 Day 1   0.000  1.000   0.452 -0.903  0.903
        Day 2   0.239  1.000   1.153 -2.066  2.544
  Day 2 Day 1  -0.239  1.000   1.153 -2.544  2.066
        Day 2   0.000  1.000   0.412 -0.824  0.824

   Variance components (sd):
        IxR   MxI   res
  Day 1   0 0.755 0.319
  Day 2   0 0.755 0.291
> BA.est( subc, linked=FALSE, Tr="log" )
  Note: Response transformed by:  .Primitive("log")


   Conversion between methods:
                alpha   beta sd.pred LoA-lo LoA-up
  To:   From:
  Day 1 Day 1   0.000  1.000   0.106 -0.212  0.212
        Day 2   0.041  1.000   0.161 -0.282  0.363
  Day 2 Day 1  -0.041  1.000   0.161 -0.363  0.282
        Day 2   0.000  1.000   0.085 -0.170  0.170

   Variance components (sd):
        IxR   MxI   res
  Day 1   0 0.091 0.075
  Day 2   0 0.091 0.060
> BA.est( visc, linked=FALSE, Tr="log" )
  Note: Response transformed by:  .Primitive("log")


   Conversion between methods:
                alpha   beta sd.pred LoA-lo LoA-up
  To:   From:
  Day 1 Day 1   0.000  1.000   0.049 -0.098  0.098
        Day 2   0.019  1.000   0.150 -0.281  0.318
  Day 2 Day 1  -0.019  1.000   0.150 -0.318  0.281
        Day 2   0.000  1.000   0.058 -0.116  0.116

   Variance components (sd):
        IxR   MxI   res
  Day 1   0 0.099 0.035
  Day 2   0 0.099 0.041
```

From the analyses of the log-transformed data it is seen that the coefficient of variation between individuals within method (the `MxI`-effect) is about 9–10%. This is basically how much the means over replicates withing each method varies around the average between the methods. The residual coefficients of variation are as in the intra-observer experiment somewhat larger for the subcutaneous measurements than for the visceral measurements.

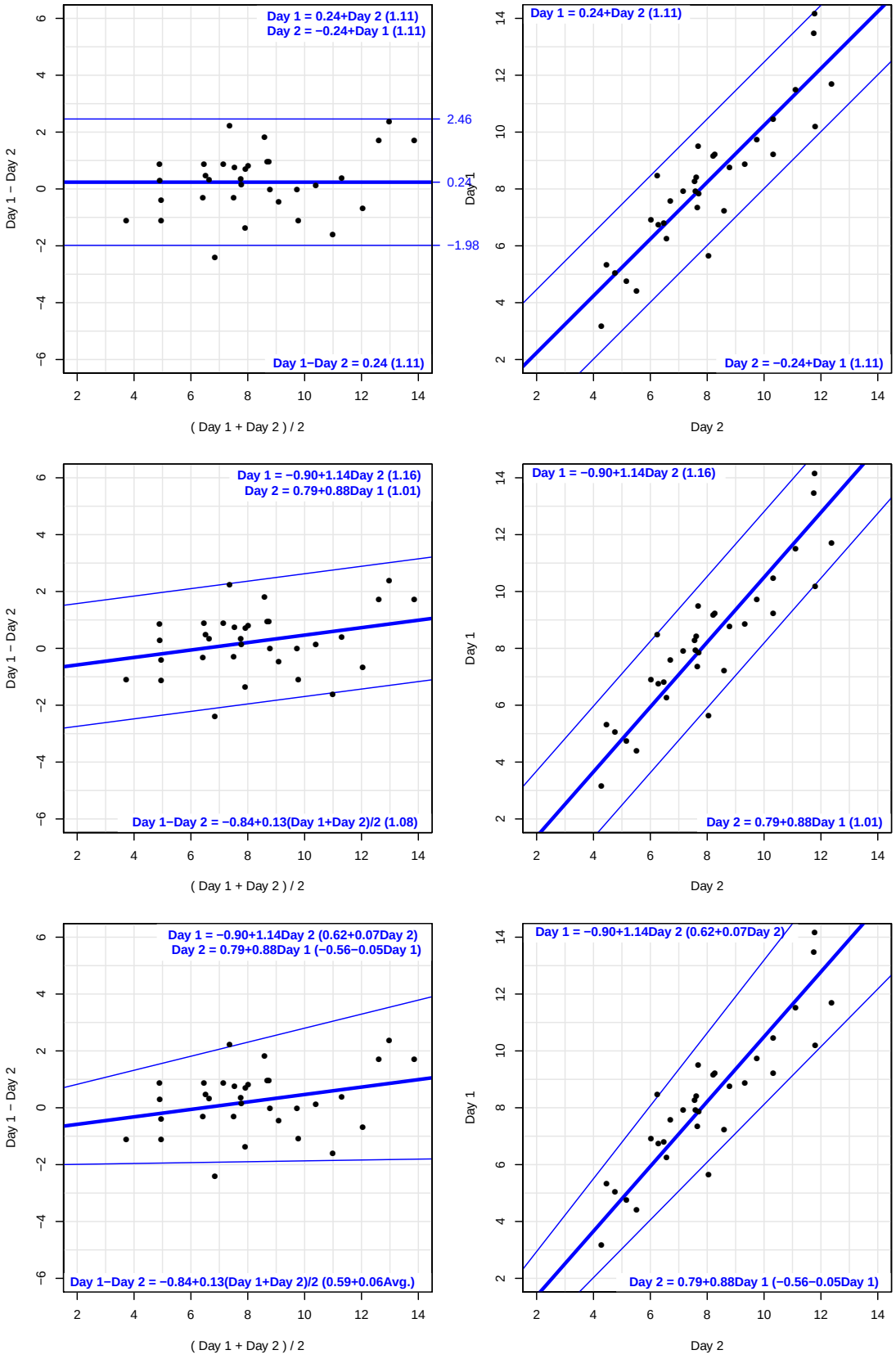

Figure 2.4: *Bland-Altman and conversion plots for mean of the visceral measurements*

2.5 Comparison of relative differences

Finally we produce the Bland-Altman plots for the two types of measurement side-by-side:

```
> par( mfrow=c(1,2), mar=c(3,3,1,3), mgp=c(3,1,0)/1.6 )
> BA.plot( mean(subc), mult=TRUE )
> BA.plot( mean(visc), mult=TRUE )
```

Here is the code for the plots in the article:

```
> for( i in 1:3 )
+    {
+ if( i==1 )   postscript( "day2d.eps", width=8, height=4, pointsize=12 )
+ if( i==2 ) win.metafile( "day2d.emf", width=8, height=4, pointsize=12 )
+ if( i==3 )          pdf( "day2d.pdf", width=8, height=4, pointsize=12 )
+ par( mfrow=c(1,2), mar=c(3,3,1,3), mgp=c(3,1,0)/1.6, las=1,
+      oma=c(0,0,0,0), cex=1 )
+ BA.plot( mean(subc), mult=T, diflim=c(0.5,2), axlim=c(0,6) )
+ mtext( "Subcutaneous fat", side=3, line=0.1, at=0, adj=0 )
+ BA.plot( mean(visc), mult=T, diflim=c(0.5,2), axlim=c(3,16) )
+ mtext( "Visceral fat", side=3, line=0.1, at=3, adj=0 )
+ mtext( "Figure 3", side=1, adj=0.5, outer=TRUE, line=-1 )
+ dev.off()
+     }
```

Here is the code for the plots in the dissertation (color-coded with sex):

```
> clr <- c("blue","red")
> item2sex <- subc[match(unique(subc$item),subc$item),c("item","sex")]
> item2sex <- item2sex[order(item2sex$item),]
> sxclr <- clr[item2sex$sex]
```

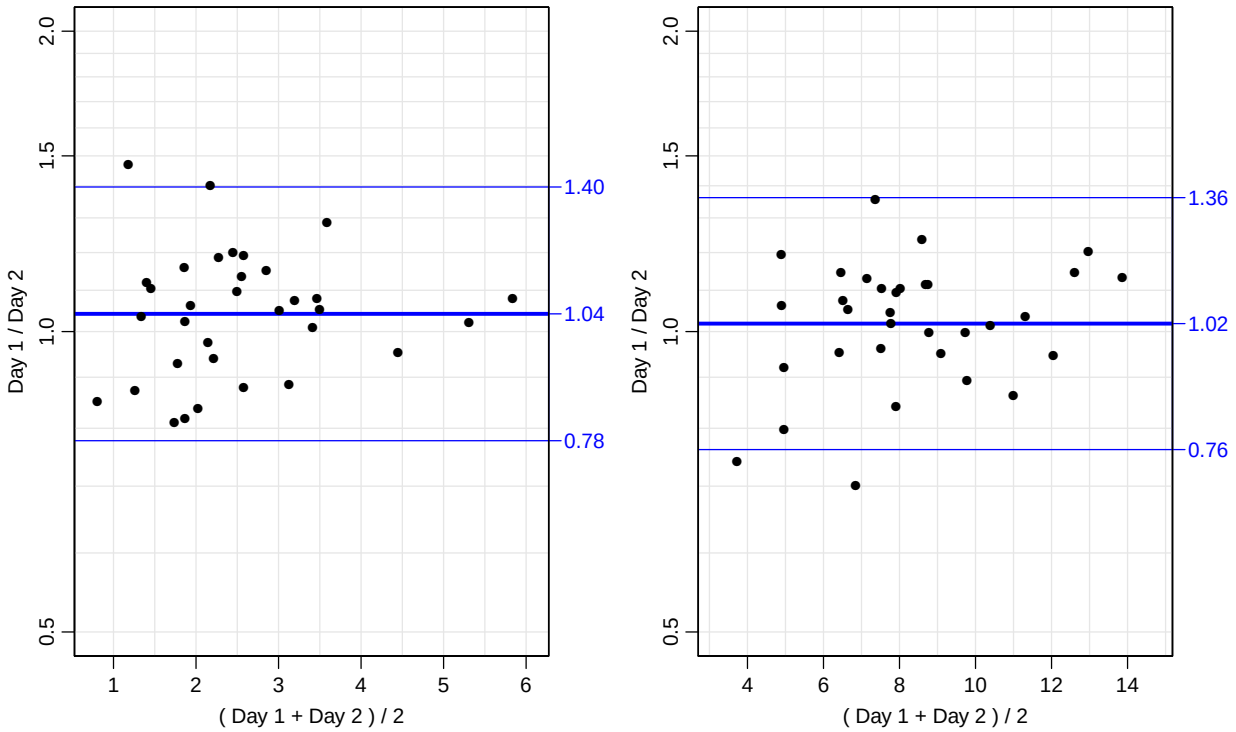

Figure 2.5: *Bland-Altman plots of the ratios (that is, log-transformed data) for subcutaneous (left) and visceral (right) fat measurements.*

```
> for( i in 1:3 )
+    {
+ if( i==1 )   postscript( "dis-day2d.eps", width=8, height=4, pointsize=12 )
+ if( i==2 ) win.metafile( "dis-day2d.emf", width=8, height=4, pointsize=12 )
+ if( i==3 )          pdf( "dis-day2d.pdf", width=8, height=4, pointsize=12 )
+ par( mfrow=c(1,2), mar=c(3,3,1,3), mgp=c(3,1,0)/1.6, las=1,
+      oma=c(0,0,0,0), cex=1 )
+ BA.plot( mean(subc), mult=T, diflim=c(0.5,2), axlim=c(0, 6), col.lines="black", col.points=sxclr )
+ mtext( "Subcutaneous fat", side=3, line=0.1, at=0, adj=0 )
+ BA.plot( mean(visc), mult=T, diflim=c(0.5,2), axlim=c(3,16), col.lines="black", col.points=sxclr )
+ mtext( "Visceral fat", side=3, line=0.1, at=3, adj=0 )
+ dev.off()
+     }
```

And here is the code for the plots in the dissertation (color-coded with difference between day 1 and 2 (`gap`) more or less than 35 days):

```
> clr <- c("limegreen","maroon")
> item2lng <- subc[match(unique(subc$item),subc$item),c("item","lng")]
> item2lng <- item2lng[order(item2lng$item),]
> lgclr <- clr[item2lng$lng]
> for( i in 1:3 )
+    {
+ if( i==1 )   postscript( "dig-day2d.eps", width=8, height=4, pointsize=12 )
+ if( i==2 ) win.metafile( "dig-day2d.emf", width=8, height=4, pointsize=12 )
+ if( i==3 )          pdf( "dig-day2d.pdf", width=8, height=4, pointsize=12 )
+ par( mfrow=c(1,2), mar=c(3,3,1,3), mgp=c(3,1,0)/1.6, las=1,
+      oma=c(0,0,0,0), cex=1 )
+ BA.plot( mean(subc), mult=T, diflim=c(0.5,2), axlim=c(0, 6),
+          col.lines="black", col.points=lgclr )
+ mtext( "Subcutaneous fat", side=3, line=0.1, at=0, adj=0 )
+ BA.plot( mean(visc), mult=T, diflim=c(0.5,2), axlim=c(3,16),
+          col.lines="black", col.points=lgclr )
+ mtext( "Visceral fat", side=3, line=0.1, at=3, adj=0 )
+ dev.off()
+     }
```

2.6 Comparative measure

As a comparative measure, we will take a look at the ratio of the visceral to the subcutaneous fat measurements.

Since this is a relative measure, it is only meaningful to to analyze the ratio of these between meals — if we looked at the differences we would get substantially different results if we considered the subcutaneous to visceral ratio. Analysis on the log scale basically only changes sign if the ratio is inverted.

So we first create the means over the replicates and then compute the ratio of these:

```
> mv <- mean( visc )
> ms <- mean( subc )
> names( mv )[4] <- "v"
> names( ms )[4] <- "s"
> vs <- merge( mv, ms )
> vs <- transform( vs, y = v/s )
> head(vs)
     meth  item repl     v     s         y
  1 Day 1 10761    1  4.40 6.060 0.7260726
  2 Day 1 11634    1  7.91 1.845 4.2872629
  3 Day 1 12076    1 10.46 2.115 4.9456265
  4 Day 1 13626    1  6.75 2.465 2.7383367
  5 Day 1 15401    1  5.64 4.035 1.3977695
  6 Day 1 16731    1  6.81 1.990 3.4221106
```

```
> vs <- Meth( vs )
  The following variables from the dataframe
  "vs" are used as the Meth variables:
  meth: meth
  item: item
  repl: repl
     y: y
          #Replicates
  Method           1 #Items #Obs: 66 Values:  min      med       max
    Day 1         33     33       33    0.7260726 3.257261 12.051064
    Day 2         33     33       33    0.9821906 3.244898  8.973384
```

Once we have created a `Meth` object with the ratio as measurement we can make the simple analyses to see if the assumptions behind LoA are fulfilled.

```
> DA.reg( vs, Tr="log" )
  Note: Response transformed by:  .Primitive("log")


   Conversion between methods:
                alpha   beta sd.pred beta=1 int(t-f) slope(t-f) sd(t-f) int(sd) slope(sd)   sd=K
  To:   From:
  Day 1 Day 1   0.000  1.000      NA     NA    0.000      0.000      NA      NA        NA     NA
        Day 2  -0.132  1.088   0.233  0.258   -0.126      0.085   0.223   0.218     0.002  0.974
  Day 2 Day 1   0.121  0.919   0.214  0.258    0.126     -0.085  -0.223   0.218     0.002  0.974
        Day 2   0.000  1.000      NA     NA    0.000      0.000      NA      NA        NA     NA
```

Clearly, the assumptions are fulfilled; there is no evidence of non-constant ratio nor monotonously varying variance.

Finally we show the Bland-Altman plot for the ratio measure:

```
> BA.plot( vs, mult=TRUE )
```

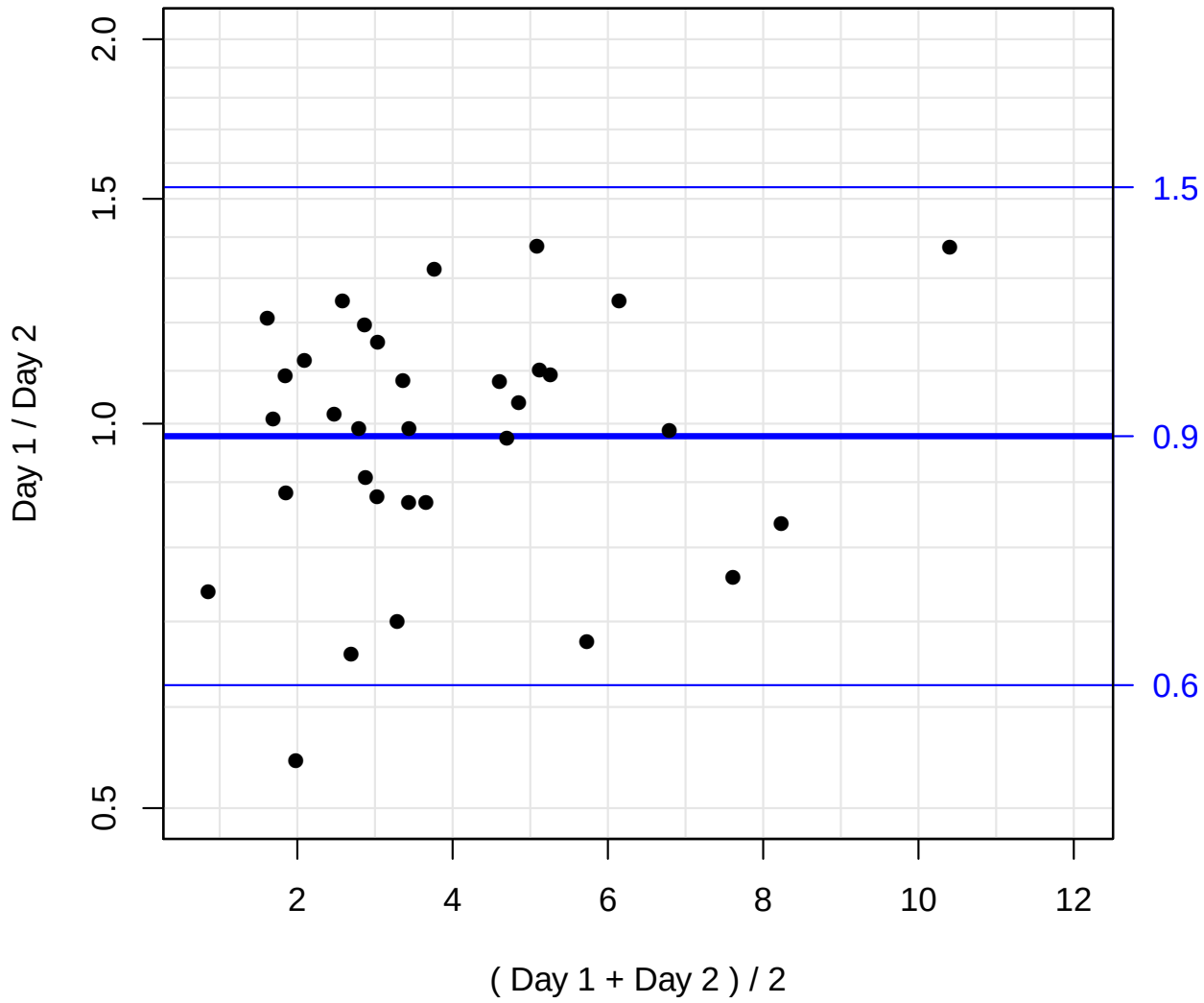

Figure 2.6: *Bland-Altman plot based on the visceral to subcutaneous ratio. The plots assesses the reproducibility of the V/S ratio from day to day.*

Chapter 3

Meal variation

3.1 Data

Patients (23) were measured for subcutaneous and visceral fat in a fasting state, and 1 and 2 hours after a meal. At each occasion, duplicate measurements were made; the average of these is *defined* as the measurement.

The data were initially pre-processed by SAS, and then we read and inspect the data for analysis:

```
> library( foreign )
> meal <- read.xport( "./data/time.xpt" )
> names( meal ) <- tolower( names(meal) )
> meal <- transform( meal,
+                    meth = factor( meal$meth,
+                                   labels=c("Fasting","1 hour","2 hours") ),
+                       sex = factor( 2-sex, labels=c("M","F") ) )
> str( meal )
  'data.frame':        265 obs. of  6 variables:
   $ item: num  5064 5064 5064 5064 5064 ...
   $ y   : num  4.62 2.96 4.89 2.97 5.32 2.53 5.3 2.26 4.46 2.54 ...
   $ type: Factor w/ 2 levels "Sub","Vis": 2 1 2 1 2 1 2 1 2 1 ...
   $ meth: Factor w/ 3 levels "Fasting","1 hour",..: 1 1 1 1 2 2 2 2 3 3 ...
   $ repl: num  1 1 2 2 1 1 2 2 1 1 ...
   $ sex : Factor w/ 2 levels "M","F": 2 2 2 2 2 2 2 2 2 2 ...

> head( meal, 30 )

     item     y type    meth repl sex
  1  5064  4.62  Vis Fasting    1   F
  2  5064  2.96  Sub Fasting    1   F
  3  5064  4.89  Vis Fasting    2   F
  4  5064  2.97  Sub Fasting    2   F
  5  5064  5.32  Vis  1 hour    1   F
  6  5064  2.53  Sub  1 hour    1   F
  7  5064  5.30  Vis  1 hour    2   F
  8  5064  2.26  Sub  1 hour    2   F
  9  5064  4.46  Vis 2 hours    1   F
  10 5064  2.54  Sub 2 hours    1   F
  11 5064  5.01  Vis 2 hours    2   F
  12 5064  2.81  Sub 2 hours    2   F
  13 5315  8.23  Vis Fasting    1   M
  14 5315  2.31  Sub Fasting    1   M
  15 5315  7.58  Vis Fasting    2   M
  16 5315  2.97  Sub Fasting    2   M
  17 5315  9.12  Vis  1 hour    1   M
  18 5315  2.99  Sub  1 hour    1   M
  19 5315  9.03  Vis  1 hour    2   M
```

```
  20 5315  2.92  Sub  1 hour    2   M
  21 5315  8.13  Vis 2 hours    1   M
  22 5315  2.76  Sub 2 hours    1   M
  23 5315  9.45  Vis 2 hours    2   M
  24 5315  2.85  Sub 2 hours    2   M
  25 7248 11.01  Vis Fasting    1   M
  26 7248  3.12  Sub Fasting    1   M
  27 7248 11.54  Vis Fasting    2   M
  28 7248  3.16  Sub Fasting    2   M
  29 7248 12.41  Vis  1 hour    1   M
  30 7248  3.11  Sub  1 hour    1   M
```

```
> addmargins( with( meal, table(item,sex) ), 1,
+             FUN=list( N=function(x) sum(x>0) ) )
```

```
        sex
item     M  F
  5064   0 12
  5315  12  0
  7248  12  0
  7298  12  0
  9632  12  0
  10558 12  0
  13836 12  0
  14236  7  0
  15854  0 11
  16427  0 12
  17091  0 12
  18666  0  9
  22251 11  0
  55865  0 11
  57205 12  0
  57849 12  0
  57935 12  0
  58172 12  0
  59724  0 12
  61194  0 12
  64152 12  0
  65544 12  0
  66709  0 12
  N     14  9
```

Then we inspect the two types of outcome separately, by forming `Meth` objects for subcutaneous and visceral fat respectively:

```
> library( MethComp )
> Sub <- Meth( subset(meal,type=="Sub") )
  The following variables from the dataframe
  "subset(meal, type == "Sub")" are used as the Meth variables:
  meth: meth
  item: item
  repl: repl
     y: y
            #Replicates
  Method             2 #Items #Obs: 138 Values:  min   med  max
    Fasting         23     23        46         0.93 2.440 4.91
    1 hour          23     23        46         0.92 2.280 4.80
    2 hours         23     23        46         0.80 2.355 4.91
```

```
> addmargins( with( Sub, table(item,sex) ), 1,
+             FUN=list( N=function(x) sum(x>0) ) )
```

```
        sex
item     M  F
  5064   0  6
  5315   6  0
  7248   6  0
```

```
    7298   6  0
    9632   6  0
    10558  6  0
    13836  6  0
    14236  6  0
    15854  0  6
    16427  0  6
    17091  0  6
    18666  0  6
    22251  6  0
    55865  0  6
    57205  6  0
    57849  6  0
    57935  6  0
    58172  6  0
    59724  0  6
    61194  0  6
    64152  6  0
    65544  6  0
    66709  0  6
    N     14  9
```

```
> Vis <- Meth( subset(meal,type=="Vis") )
```

```
  The following variables from the dataframe
  "subset(meal, type == "Vis")" are used as the Meth variables:
  meth: meth
  item: item
  repl: repl
     y: y
            #Replicates
  Method         1     2 #Items #Obs: 127 Values:  min  med   max
    Fasting      1    21     22         43         4.29 7.94 14.94
    1 hour       1    21     22         43         5.03 8.97 15.86
    2 hours      3    19     22         41         4.46 8.77 15.65
```

```
> addmargins( with( Vis, table(item,sex) ), 1,
+             FUN=list( N=function(x) sum(x>0) ) )
```

```
         sex
  item     M  F
    5064   0  6
    5315   6  0
    7248   6  0
    7298   6  0
    9632   6  0
    10558  6  0
    13836  6  0
    14236  1  0
    15854  0  5
    16427  0  6
    17091  0  6
    18666  0  3
    22251  5  0
    55865  0  5
    57205  6  0
    57849  6  0
    57935  6  0
    58172  6  0
    59724  0  6
    61194  0  6
    64152  6  0
    65544  6  0
    66709  0  6
    N     14  9
```

The method of measuring thickness of fat is to use the mean of two measurements

3.1.1 Subcutaneous fat

For an initial overview we make pairwise comparisons of the three time-points using `DA.reg`, both for the original data and for the log-transformed:

```
> DA.reg( mean(Sub) )
   Conversion between methods:
                  alpha   beta sd.pred beta=1 int(t-f) slope(t-f) sd(t-f) int(sd) slope(sd)   sd=
To:     From:
Fasting Fasting   0.000  1.000      NA     NA    0.000      0.000      NA      NA         NA     NA
        1 hour    0.101  0.995   0.244  0.929    0.101     -0.005   0.244   0.200      0.006  0.91
        2 hours   0.157  0.950   0.176  0.240    0.161     -0.051   0.181   0.173      0.007  0.773
1 hour  Fasting  -0.101  1.005   0.245  0.929   -0.101      0.005  -0.244   0.200      0.006  0.91
        1 hour    0.000  1.000      NA     NA    0.000      0.000      NA      NA         NA     NA
        2 hours   0.057  0.955   0.163  0.250    0.058     -0.046   0.166   0.133      0.001  0.972
2 hours Fasting  -0.166  1.052   0.186  0.240   -0.161      0.051  -0.181   0.173      0.007  0.773
        1 hour   -0.059  1.047   0.170  0.250   -0.058      0.046  -0.166   0.133      0.001  0.972
        2 hours   0.000  1.000      NA     NA    0.000      0.000      NA      NA         NA     NA

> DA.reg( mean(Sub), Trans="log" )
  Note: Response transformed by:  .Primitive("log")


   Conversion between methods:
                  alpha   beta sd.pred beta=1 int(t-f) slope(t-f) sd(t-f) int(sd) slope(sd)   sd=
To:     From:
Fasting Fasting   0.000  1.000      NA     NA    0.000      0.000      NA      NA         NA     NA
        1 hour    0.091  0.940   0.106  0.298    0.094     -0.062   0.109   0.128     -0.038  0.432
        2 hours   0.085  0.926   0.082  0.100    0.089     -0.077   0.085   0.135     -0.060  0.037
1 hour  Fasting  -0.097  1.064   0.113  0.298   -0.094      0.062  -0.109   0.128     -0.038  0.432
        1 hour    0.000  1.000      NA     NA    0.000      0.000      NA      NA         NA     NA
        2 hours  -0.006  0.985   0.076  0.696   -0.006     -0.015   0.077   0.077     -0.019  0.597
2 hours Fasting  -0.092  1.080   0.089  0.100   -0.089      0.077  -0.085   0.135     -0.060  0.037
        1 hour    0.006  1.016   0.078  0.696    0.006      0.015  -0.077   0.077     -0.019  0.597
        2 hours   0.000  1.000      NA     NA    0.000      0.000      NA      NA         NA     NA
```

We see no evidence of non-constant bias or s.d. except possibly for the difference between fasting and 2 hours past meal, where the sd. of the differences seems to be decreasing.

This is also apparent from the overall plot of the readings from different times:

```
> plot.Meth( mean(Sub), cex=1.2, var.names=TRUE )
```

Finally, we estimate the variance components from the complete set with replicates for the subcutaneous measurements:

```
> BA.est( Sub, linked=FALSE )
   Conversion between methods:
                  alpha   beta sd.pred LoA-lo LoA-up
To:     From:
Fasting Fasting   0.000  1.000   0.196 -0.391  0.391
        1 hour    0.088  1.000   0.259 -0.429  0.606
        2 hours   0.037  1.000   0.238 -0.438  0.513
1 hour  Fasting  -0.088  1.000   0.259 -0.606  0.429
        1 hour    0.000  1.000   0.172 -0.344  0.344
        2 hours  -0.051  1.000   0.220 -0.492  0.390
2 hours Fasting  -0.037  1.000   0.238 -0.513  0.438
        1 hour    0.051  1.000   0.220 -0.390  0.492
        2 hours   0.000  1.000   0.195 -0.391  0.391

   Variance components (sd):
        IxR   MxI   res
Fasting   0 0.136 0.138
1 hour    0 0.121 0.121
2 hours   0 0.000 0.138
```

```
> BA.est( Sub, linked=FALSE, Tr="log" )
  Note: Response transformed by:  .Primitive("log")


   Conversion between methods:
                    alpha   beta sd.pred LoA-lo LoA-up
  To:     From:
  Fasting Fasting   0.000  1.000   0.082 -0.165  0.165
          1 hour    0.044  1.000   0.119 -0.193  0.282
          2 hours   0.027  1.000   0.118 -0.209  0.263
  1 hour  Fasting  -0.044  1.000   0.119 -0.282  0.193
          1 hour    0.000  1.000   0.078 -0.155  0.155
          2 hours  -0.018  1.000   0.105 -0.227  0.192
  2 hours Fasting  -0.027  1.000   0.118 -0.263  0.209
          1 hour    0.018  1.000   0.105 -0.192  0.227
          2 hours   0.000  1.000   0.104 -0.208  0.208

   Variance components (sd):
          IxR   MxI   res
  Fasting   0 0.072 0.058
```

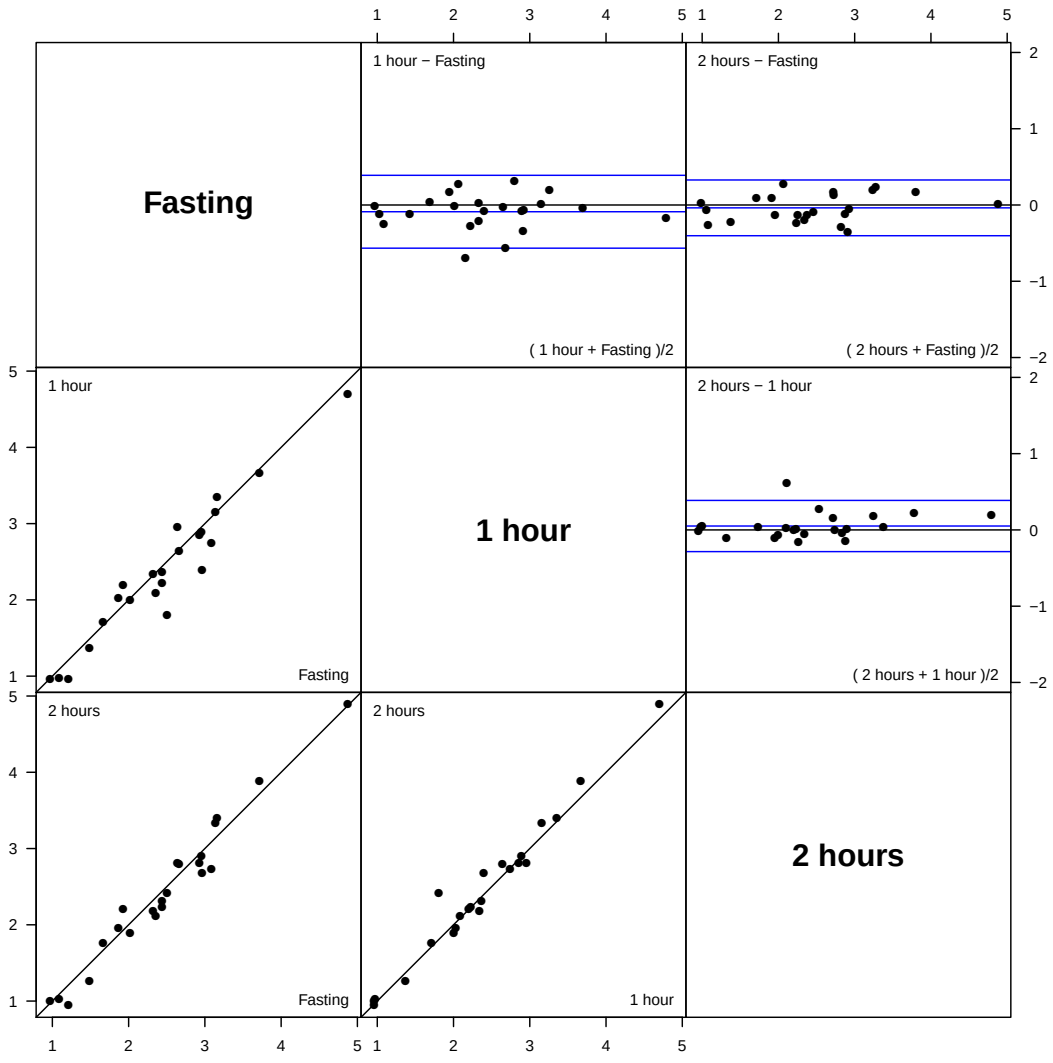

Figure 3.1: *Subcutaneous measurements, averages over replicates.*

```
  1 hour    0 0.051 0.055
  2 hours   0 0.000 0.073
```

```
> BA.est( Sub, linked=FALSE, Tr="log", varMxI=FALSE )
```

```
  Note: Response transformed by:  .Primitive("log")


   Conversion between methods:
                   alpha   beta sd.pred LoA-lo LoA-up
  To:     From:
  Fasting Fasting   0.000  1.000   0.086 -0.172  0.172
          1 hour    0.044  1.000   0.110 -0.175  0.264
          2 hours   0.027  1.000   0.119 -0.212  0.265
  1 hour  Fasting  -0.044  1.000   0.110 -0.264  0.175
          1 hour    0.000  1.000   0.078 -0.157  0.157
          2 hours  -0.018  1.000   0.117 -0.251  0.216
  2 hours Fasting  -0.027  1.000   0.119 -0.265  0.212
          1 hour    0.018  1.000   0.117 -0.216  0.251
          2 hours   0.000  1.000   0.103 -0.205  0.205

   Variance components (sd):
          IxR   MxI   res
  Fasting   0 0.051 0.061
  1 hour    0 0.051 0.055
  2 hours   0 0.051 0.073
```

3.1.2 Visceral fat

For an initial overview we make pairwise comparisons of the three time-points using `DA.reg`:

```
> DA.reg( Vis )
```

```
   Conversion between methods:
                   alpha   beta sd.pred beta=1 int(t-f) slope(t-f) sd(t-f) int(sd) slope(sd)   sd=k
  To:     From:
  Fasting Fasting   0.000  1.000      NA     NA    0.000      0.000      NA      NA         NA     NA
          1 hour   -0.573  0.947   0.497  0.044   -0.589     -0.055   0.511   0.342     0.020  0.258
          2 hours  -0.338  0.952   0.558  0.101   -0.346     -0.049   0.572   0.563     0.000  0.985
  1 hour  Fasting   0.605  1.056   0.525  0.044    0.589      0.055  -0.511   0.342     0.020  0.258
          1 hour    0.000  1.000      NA     NA    0.000      0.000      NA      NA         NA     NA
          2 hours   0.336  1.001   0.484  0.970    0.336      0.001   0.484   0.695    -0.023  0.196
  2 hours Fasting   0.355  1.051   0.586  0.101    0.346      0.049  -0.572   0.563     0.000  0.985
          1 hour   -0.336  0.999   0.483  0.970   -0.336     -0.001  -0.484   0.695    -0.023  0.196
          2 hours   0.000  1.000      NA     NA    0.000      0.000      NA      NA         NA     NA
```

```
> DA.reg( Vis, Tr="log" )
```

```
  Note: Response transformed by:  .Primitive("log")


   Conversion between methods:
                   alpha   beta sd.pred beta=1 int(t-f) slope(t-f) sd(t-f) int(sd) slope(sd)   sd=k
  To:     From:
  Fasting Fasting   0.000  1.000      NA     NA    0.000      0.000      NA      NA         NA     NA
          1 hour   -0.295  1.076   0.063  0.013   -0.285      0.073   0.061   0.117    -0.026  0.196
          2 hours  -0.148  1.026   0.074  0.443   -0.146      0.026   0.073   0.180    -0.052  0.042
  1 hour  Fasting   0.275  0.929   0.059  0.013    0.285     -0.073  -0.061   0.117    -0.026  0.196
          1 hour    0.000  1.000      NA     NA    0.000      0.000      NA      NA         NA     NA
          2 hours   0.147  0.952   0.061  0.096    0.150     -0.050   0.062   0.189    -0.060  0.009
  2 hours Fasting   0.145  0.975   0.072  0.443    0.146     -0.026  -0.073   0.180    -0.052  0.042
          1 hour   -0.154  1.051   0.064  0.096   -0.150      0.050  -0.062   0.189    -0.060  0.009
          2 hours   0.000  1.000      NA     NA    0.000      0.000      NA      NA         NA     NA
```

We see no evidence of non-constant bias or s.d. which is also apparent from the overall plot of the readings from different times:

```
> plot.Meth( Vis, cex=1.2 )
```

```
> plot( mean(Vis,sim=T), cex=1.2, var.names=TRUE )
```

Finally we estimate the variance components from the complete set with replicates for the visceral measurements:

```
> BA.est( Vis )
   Conversion between methods:
                    alpha    beta sd.pred LoA-lo LoA-up
```

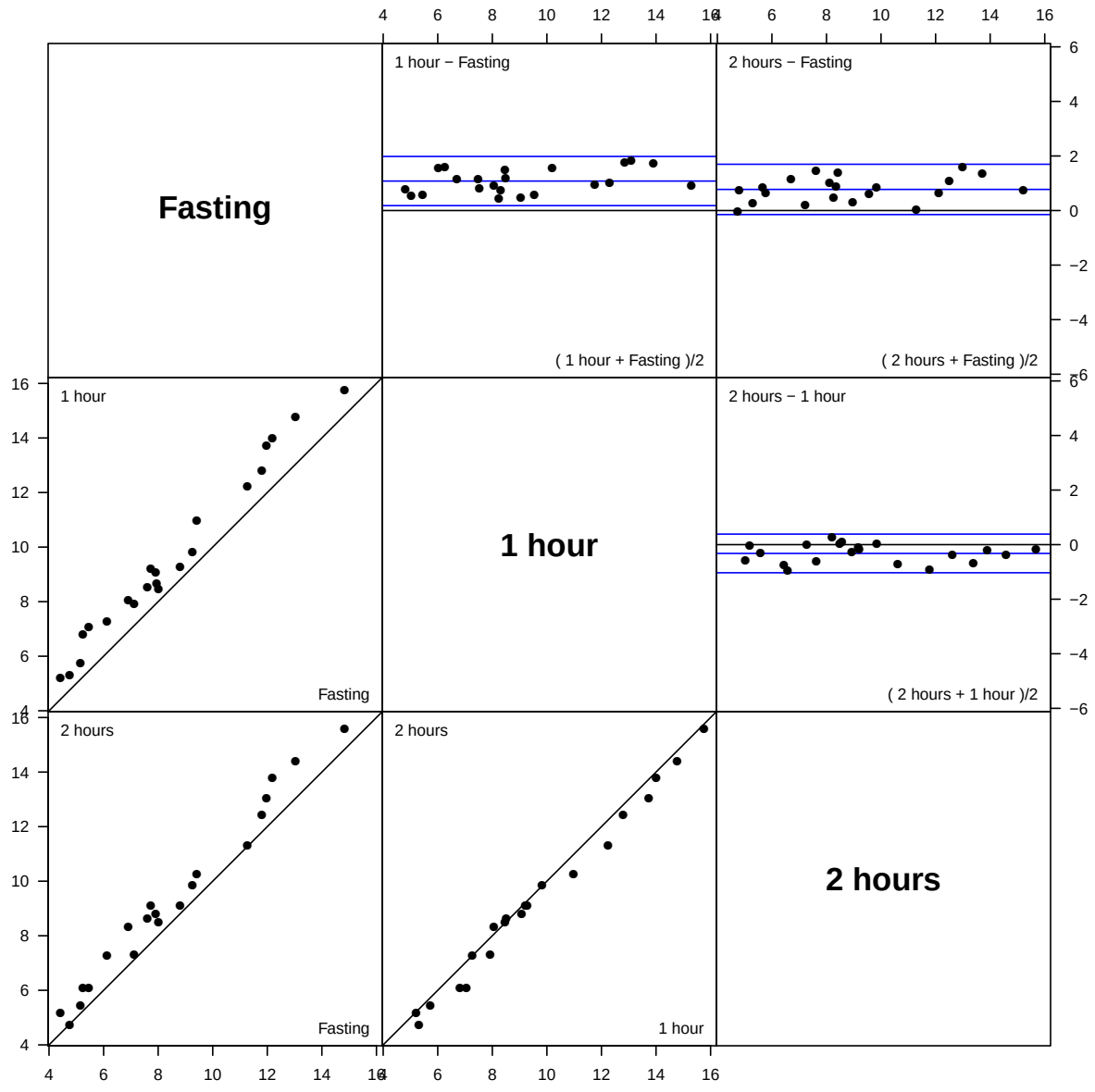

Figure 3.2: *Visceral measurements, averages over replicates.*

```
  To:     From:
  Fasting Fasting   0.000  1.000   0.429 -0.857  0.857
          1 hour   -1.087  1.000   0.527 -2.141 -0.034
          2 hours  -0.770  1.000   0.574 -1.918  0.378
  1 hour  Fasting   1.087  1.000   0.527  0.034  2.141
          1 hour    0.000  1.000   0.329 -0.659  0.659
          2 hours   0.318  1.000   0.465 -0.613  1.248
  2 hours Fasting   0.770  1.000   0.574 -0.378  1.918
          1 hour   -0.318  1.000   0.465 -1.248  0.613
          2 hours   0.000  1.000   0.518 -1.036  1.036

   Variance components (sd):
          IxR   MxI   res
  Fasting   0 0.322 0.303
  1 hour    0 0.167 0.233
  2 hours   0 0.000 0.366
```

```
> BA.est( Vis, Tr='log', linked=FALSE )
```

```
  Note: Response transformed by:  .Primitive("log")


   Conversion between methods:
                   alpha   beta sd.pred LoA-lo LoA-up
  To:     From:
  Fasting Fasting   0.000  1.000   0.052 -0.103  0.103
          1 hour   -0.128  1.000   0.065 -0.259  0.003
          2 hours  -0.090  1.000   0.071 -0.232  0.052
  1 hour  Fasting   0.128  1.000   0.065 -0.003  0.259
          1 hour    0.000  1.000   0.037 -0.074  0.074
          2 hours   0.038  1.000   0.062 -0.086  0.161
  2 hours Fasting   0.090  1.000   0.071 -0.052  0.232
          1 hour   -0.038  1.000   0.062 -0.161  0.086
          2 hours   0.000  1.000   0.068 -0.135  0.135

   Variance components (sd):
          IxR   MxI   res
  Fasting   0 0.038 0.036
  1 hour    0 0.029 0.026
  2 hours   0 0.000 0.048
```

```
> BA.est( Vis, Tr='log', linked=FALSE, varMxI=FALSE )
```

```
  Note: Response transformed by:  .Primitive("log")


   Conversion between methods:
                   alpha   beta sd.pred LoA-lo LoA-up
  To:     From:
  Fasting Fasting   0.000  1.000   0.053 -0.106  0.106
          1 hour   -0.128  1.000   0.061 -0.250 -0.005
          2 hours  -0.090  1.000   0.072 -0.234  0.054
  1 hour  Fasting   0.128  1.000   0.061  0.005  0.250
          1 hour    0.000  1.000   0.037 -0.074  0.074
          2 hours   0.038  1.000   0.067 -0.096  0.171
  2 hours Fasting   0.090  1.000   0.072 -0.054  0.234
          1 hour   -0.038  1.000   0.067 -0.171  0.096
          2 hours   0.000  1.000   0.065 -0.131  0.131

   Variance components (sd):
          IxR   MxI   res
  Fasting   0 0.029 0.038
  1 hour    0 0.029 0.026
  2 hours   0 0.029 0.046
```

It is seen that the coefficient of variation (that is the "irrelevant" variation) is smallest (c.v. 2.5%) for the fasting measurements, whereas it is about 4% for the non-fasting.

Furthermore there is little information about the separate methods' variation around the mean for each person, the overall c.v. is about 3%.

3.2 Comparison of relative differences

We also want to see the relative differences between the means, so that we can get a feel for how the differences are relative to measurements themselves. Hence we take the averages over the replicates and make Bland-Altman plots color-coded by sex.

A little fidgeting is required to get the sex variable across to the datasets of means, and a lot of fidgeting is required to get the colors as a vector correctly, since the plotting of the colors is actually done on the basis of a wide-transformed dataset:

```
> clr <- c("blue","red")
> item2sex <- Sub[match(unique(Sub$item),Sub$item),c("item","sex")]
> item2sex <- item2sex[order(item2sex$item),]
> sxclr <- clr[item2sex$sex]
> mSub <- merge( mean(Sub), item2sex, all=FALSE )
> mVis <- merge( mean(Vis), item2sex, all=FALSE )
> par( mfrow=c(3,2), mar=c(3,3,1,2.5), mgp=c(3,1,0)/1.6, oma=c(0,0,2,0) )
> BA.plot( mSub, wh.comp=1:2, mult=TRUE, diflim=1.5,
+          col.lines="black", col.points=sxclr )
> BA.plot( mVis, wh.comp=1:2, mult=TRUE, diflim=1.5,
+          col.lines="black", col.points=sxclr )
> BA.plot( mSub, wh.comp=2:3, mult=TRUE, diflim=1.5,
+          col.lines="black", col.points=sxclr )
> BA.plot( mVis, wh.comp=2:3, mult=TRUE, diflim=1.5,
+          col.lines="black", col.points=sxclr )
> BA.plot( mSub, wh.comp=c(1,3), mult=TRUE, diflim=1.5,
+          col.lines="black", col.points=sxclr )
> BA.plot( mVis, wh.comp=c(1,3), mult=TRUE, diflim=1.5,
+          col.lines="black", col.points=sxclr )
> mtext( c("Subcutaneous","Visceral"),line=0,outer=TRUE, side=3,
+        at=c(1,3)/4, cex=0.7 )
```

From the figure 3.3 we see that the relative limits of agreement on the multiplicative scale for subcutaneous measurements are about $\overset{\times}{\div}$ 1.15 a bit larger between the fasting and 1 hour reading, but largely centered around 1.

For the visceral measurements the width of relative LoA are about $\overset{\times}{\div}$ 1.10, but also we see that the fasting values are some 10% smaller than the both of the non-fasting values.

Here is the code for the plots in the article:

```
> for( i in 1:3 )
+    {
+ if( i==1 )   postscript( "meals.eps", width=8, height=8, pointsize=12 )
+ if( i==2 ) win.metafile( "meals.emf", width=8, height=8, pointsize=12 )
+ if( i==3 )          pdf( "meals.pdf", width=8, height=8, pointsize=12 )
+ par( mfrow=c(2,2), mar=c(3,3,1,3), mgp=c(3,1,0)/1.6, las=1,
+      oma=c(0,0,0,0), cex=1 )
+ BA.plot( mSub, wh.comp=1:2, mult=TRUE, diflim=1.5, col.lines="black" )
+ mtext( "Subcutaneous fat", side=3, line=0.1, at=0, adj=0 )
+ BA.plot( mVis, wh.comp=1:2, mult=TRUE, diflim=1.5, col.lines="black" )
+ mtext( "Visceral fat", side=3, line=0.1, at=3, adj=0 )
+ BA.plot( mSub, wh.comp=c(1,3), mult=TRUE, diflim=1.5, col.lines="black" )
+ BA.plot( mVis, wh.comp=c(1,3), mult=TRUE, diflim=1.5, col.lines="black" )
+ mtext( "Figure 4", side=1, adj=0.5, outer=TRUE, line=-1 )
+ dev.off()
+    }
```

Here is the code for the plots in the disseration (using colors):

```
> for( i in 1:3 )
+    {
+ if( i==1 )   postscript( "dis-meals.eps", width=8, height=8, pointsize=12 )
+ if( i==2 ) win.metafile( "dis-meals.emf", width=8, height=8, pointsize=12 )
```

```
+ if( i==3 )          pdf( "dis-meals.pdf", width=8, height=8, pointsize=12 )
+ par( mfrow=c(2,2), mar=c(3,3,1,3), mgp=c(3,1,0)/1.6, las=1,
+      oma=c(0,0,0,0), cex=1 )
+ BA.plot( mSub, wh.comp=1:2, mult=TRUE, diflim=1.5,
+           col.lines="black", col.points=sxclr )
+ mtext( "Subcutaneous fat", side=3, line=0.1, at=0, adj=0 )
+ BA.plot( mVis, wh.comp=1:2, mult=TRUE, diflim=1.5,
+           col.lines="black", col.points=sxclr )
+ mtext( "Visceral fat", side=3, line=0.1, at=3, adj=0 )
+ BA.plot( mSub, wh.comp=c(1,3), mult=TRUE, diflim=1.5,
```

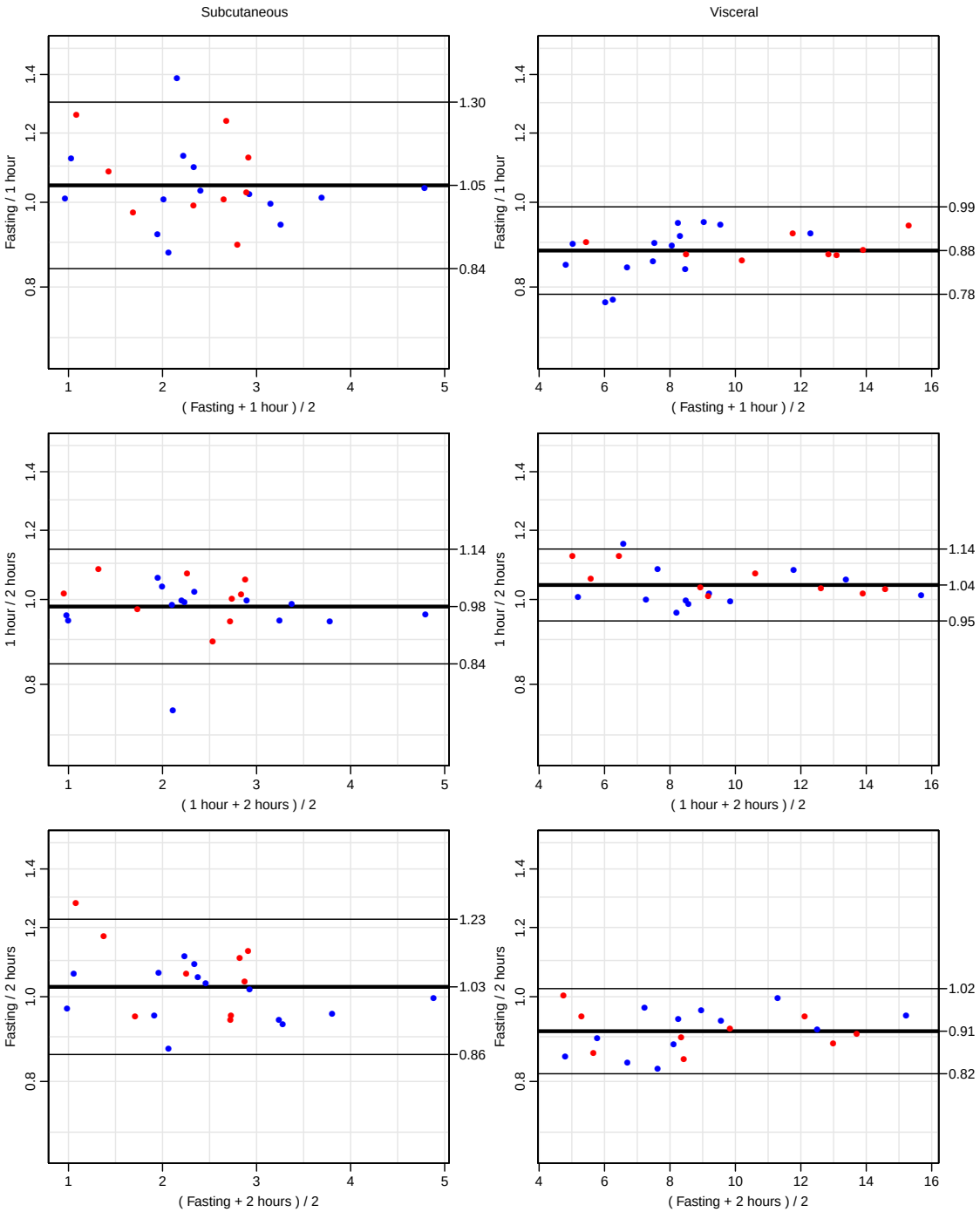

Figure 3.3: *Bland-Altman plots for subcutaneous (left) and visceral (right) measurements.*

```
+          col.lines="black", col.points=sxclr )
+ BA.plot( mVis, wh.comp=c(1,3), mult=TRUE, diflim=1.5,
+          col.lines="black", col.points=sxclr )
+ # mtext( "Figure 4", side=1, adj=0.5, outer=TRUE, line=-1 )
+ dev.off()
+    }
```

3.3 Comparative measure

As a comparative measure, we will take a look at the ratio of the visceral to the subcutaneous fat measurements.

Since this is a relative measure, it is only meaningful to to analyze the ratio of these between meals — if we looked at the differences we would get substantially different results if we considered the subcutaneous to visceral ratio. Analysis on the log scale basically only changes sign if the ratio is inverted.

So we first create the means over the replicates and then compute the ratio of these:

```
> mv <- mean( Vis )
> ms <- mean( Sub )
> names( mv )[4] <- "v"
> names( ms )[4] <- "s"
> vs <- merge( mv, ms )
> vs <- transform( vs, y = v/s )
> head(vs)
     meth  item repl      v     s        y
1 1 hour 10558    1 14.005 2.640 5.304924
2 1 hour 13836    1  7.270 2.225 3.267416
3 1 hour 15854    1  8.060 4.700 1.714894
4 1 hour 16427    1  7.055 2.005 3.518703
5 1 hour 17091    1  5.210 2.890 1.802768
6 1 hour 18666    1  8.680 1.805 4.808864

> vs <- Meth( vs )

  The following variables from the dataframe
  "vs" are used as the Meth variables:
  meth: meth
  item: item
  repl: repl
     y: y
           #Replicates
  Method             1 #Items #Obs: 66 Values:  min      med      max
    Fasting         22     22       22     1.415385 3.380213 10.97706
    1 hour          22     22       22     1.714894 3.900666 14.15464
    2 hours         22     22       22     1.705521 3.801295 12.71707
```

Once we have created a `Meth` object with the ratio as measurement we can make the simple analyses to see if the assumptions behind LoA are fulfilled.

```
> DA.reg( vs, Tr="log" )
  Note: Response transformed by:  .Primitive("log")


   Conversion between methods:
                    alpha   beta sd.pred beta=1 int(t-f) slope(t-f) sd(t-f) int(sd) slope(sd)   sd=k
  To:     From:
  Fasting Fasting   0.000  1.000      NA     NA    0.000      0.000      NA      NA         NA     NA
          1 hour   -0.157  0.985   0.137  0.794   -0.158     -0.015   0.138   0.120      0.009  0.831
          2 hours  -0.089  0.977   0.121  0.643   -0.090     -0.023   0.122   0.107      0.005  0.890
  1 hour  Fasting   0.159  1.015   0.139  0.794    0.158      0.015  -0.138   0.120      0.009  0.831
```

```
          1 hour    0.000  1.000      NA      NA    0.000      0.000      NA      NA        NA     NA
          2 hours   0.059  0.991   0.069   0.735    0.060     -0.009   0.069   0.105    -0.026  0.164
2 hours Fasting     0.091  1.023   0.124   0.643    0.090      0.023  -0.122   0.107     0.005  0.890
          1 hour   -0.060  1.010   0.070   0.735   -0.060      0.009  -0.069   0.105    -0.026  0.164
          2 hours   0.000  1.000      NA      NA    0.000      0.000      NA      NA        NA     NA
```

We see that the basic assumptions behind the LoA seem to be fulfilled.

Finally we show the Bland-Altman plots:

```
> par(mfrow=c(1,3),mar=c(4,4,1,3),mgp=c(3,1,0)/1.6)
> BA.plot( vs, wh.comp=c(1,2), mult=TRUE )
> BA.plot( vs, wh.comp=c(1,3), mult=TRUE )
> BA.plot( vs, wh.comp=c(2,3), mult=TRUE )
```

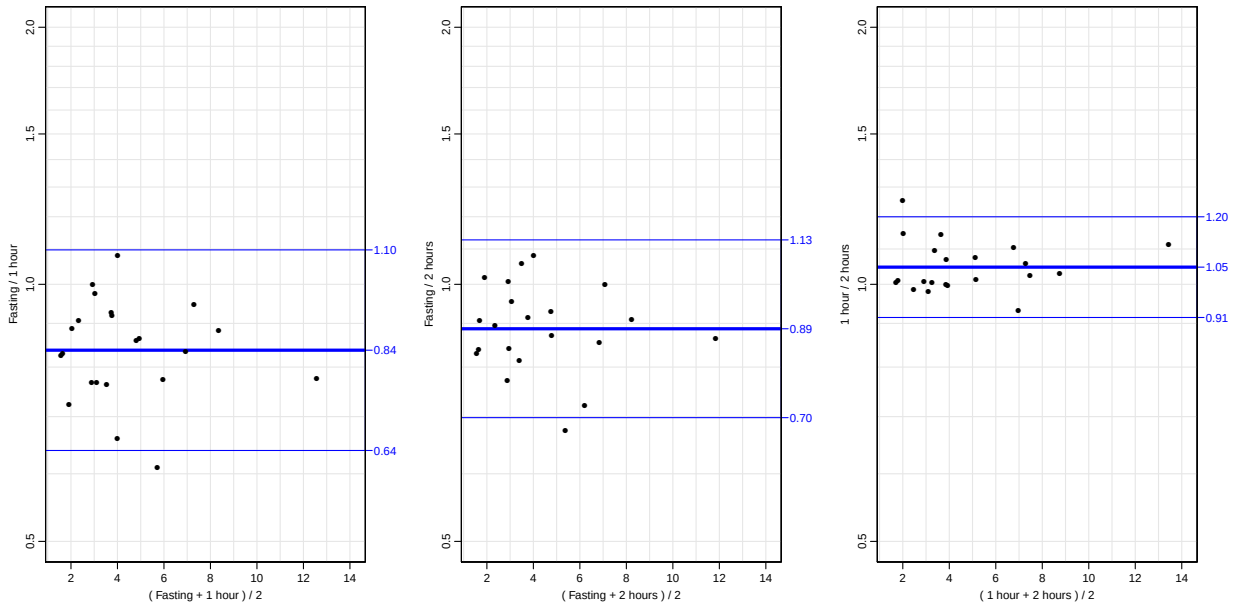

Figure 3.4: *Bland-Altman plots based on the visceral to subcutaneous ratio. The plots assess the reproducibility of the V/S ratio.*

Bibliography

[1] B. Carstensen. *Comparing Clinical Measurement Methods: A practical guide*. Wiley, 2010.

[2] B Carstensen, J Simpson, and LC Gurrin. Statistical models for assessing agreement in method comparison studies with replicate measurements. *International Journal of Biostatistics*, 4(1):Article 16, 2008.