Solutions lab 1 - Clinical trials

Solutions 1 - Sample size estimation

The primary outcome measure is serum calcium level, which is a continuous (positive) measurement. The sample size formula we have discussed in lectures is

$$N \;=\; \frac{2\sigma^2}{(\mu_A - \mu_B)^2}[\phi^{-1}(1-\alpha/2) + \phi^{-1}(1-\beta)]^2$$

and can be computed in `R` using the function

```
samplesize.cts <- function(dif, std, siglev, pwr){
    2 * (std ^ 2) * ((qnorm(1 - siglev / 2) + qnorm(pwr)) ^ 2) / (dif ^ 2)
}
```

In this function `diff` is the difference in the treatments $\mu_A - \mu_B$, `std` is the outcome measure standard deviation σ, `siglev` is α and `pwr` is the power $1 - \beta$.

1.

```
samplesize.cts(0.25, 0.99, 0.05, 0.9)
```

```
## [1] 329.5464
```

2.

```
samplesize.cts(0.25, 0.99, 0.05, 0.8)
```

```
## [1] 246.166
```

3.

```
samplesize.cts(0.25, 1.5, 0.05, 0.9)
```

```
## [1] 756.5345
```

4.

```
samplesize.cts(0.2, 0.99, 0.05, 0.9)
```

```
## [1] 514.9163
```

5.

```
samplesize.bin <- function(theta.A, theta.B, siglev,pwr){
   (theta.A * (1 - theta.A) + theta.B * (1 - theta.B)) *
   ((qnorm(1 - siglev / 2) + qnorm(pwr)) ^ 2) / ((theta.A - theta.B) ^ 2)
}

samplesize.bin(0.15, 0.25, 0.05, 0.95)
```

```
## [1] 409.3334
```

Solutions 2 - Randomisation

1.

```
sample1 <- sample(c("A", "B"), 660, replace=TRUE)
```

2. The problem is an imbalance in the number of patients randomised to the two groups. In this example, we would ideally end up with 330 patients in either group. The sample function produces a **random** sample and therefore everytime it is run you observe a different number of patients in group A and group B. At the time of writing these solutions I observe 321 participants in group A and 339 participants in group B. Due to a large sample size of 660 patients the imbalance is not huge. The problem would likely be worse for a trial with a smaller sample size.

   ```
   table(sample1)
   ```

   ```
   ## sample1
   ##   A   B
   ## 321 339
   ```

3. The trick here is to realise that every patient has a probability of 0.5 of being randomised to a group and that there are $n = 660$ patients. That should sound like a familiar situation to you! We have 660 trials and a probability of 0.5 of success (success being allocated into one of the groups). You should immediately start thinking of a binomial distribution. The rest follows from your knowledge of probability theory. Note the last line utilises the trick

   $$P(\text{event happening}) = 1 - P(\text{all other events happening}).$$

   The probability the doctor looks for is approximately 67%. Additionally, the density plot illustrates that an imbalance of more than approximately 300/360 is highly unlikely.

   ```
   x <- 1:660
   px <- dbinom(x, 660, prob=0.5)
   1-sum(px[x %in% 325:335])
   ```

   ```
   ## [1] 0.6685564
   ```

   ```
   plot(x, px, main = "PDF of binomial distribution",
        xlab = "number of people assigned to one of the groups",
        ylab = "density")
   ```

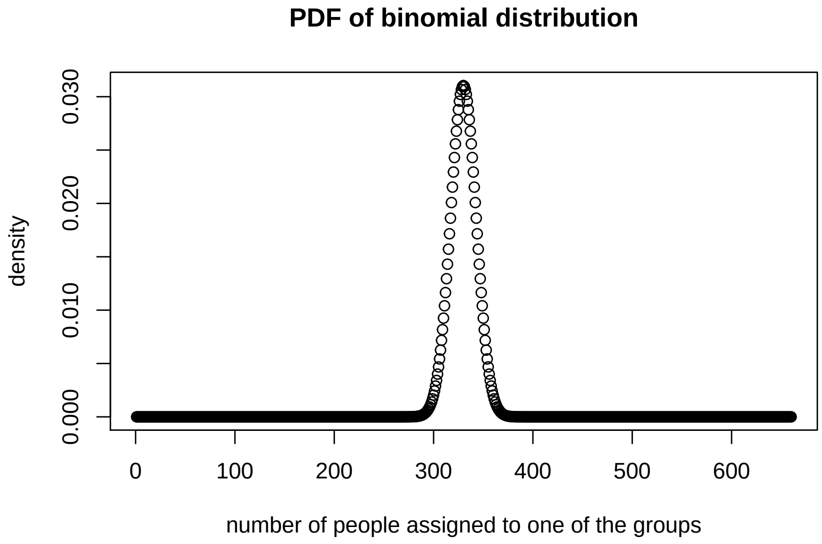

Figure 1: PDF of binomial distribution with parameters n = 660 and p = 0.5.

4. One possible solution is to use the `sample()` function to create 6 sets of integers from 1 to 110 in a random order. Then for each set simply allocate all those integers between 1 and 55 to treatment A and the integers between 56 and 110 to treatment B. The output of the first of the 6 blocks is displayed - the remaining output is omitted to save space.

```
permutation1 <- sample(1:110, size=110, replace = FALSE)
permutation1
```

```
##   [1]  35  91  11  13  45  59  94  42  52  81  29  31 103  67  84  69  21  92
##  [19]  76   6  85  33  41   4  88  99  96  54 102  75  83  22  62  93  57 108
##  [37]  78  56  40  87  71  73  86  37   9 101   8  55   1 104  97  47  49  74
##  [55]  14  32   3  17 109  23  36  20  82  60  79  12  58  70  90  77  10 110
##  [73]  80   5  27 107  18  34  44 100  19  98  50  25  72  89  63  39  30   7
##  [91]  24  38  15  61  46  68 106  43  65  28  16  51   2  48 105  26  66  64
## [109]  95  53
```

```
permutation2 <- sample(1:110, size=110, replace = FALSE)
permutation3 <- sample(1:110, size=110, replace = FALSE)
permutation4 <- sample(1:110, size=110, replace = FALSE)
permutation5 <- sample(1:110, size=110, replace = FALSE)
permutation6 <- sample(1:110, size=110, replace = FALSE)

block1 <- rep("A",110)
block1
```

```
##   [1] "A" "A" "A" "A" "A" "A" "A" "A" "A" "A" "A" "A" "A" "A" "A" "A" "A" "A"
##  [19] "A" "A" "A" "A" "A" "A" "A" "A" "A" "A" "A" "A" "A" "A" "A" "A" "A" "A"
##  [37] "A" "A" "A" "A" "A" "A" "A" "A" "A" "A" "A" "A" "A" "A" "A" "A" "A" "A"
##  [55] "A" "A" "A" "A" "A" "A" "A" "A" "A" "A" "A" "A" "A" "A" "A" "A" "A" "A"
##  [73] "A" "A" "A" "A" "A" "A" "A" "A" "A" "A" "A" "A" "A" "A" "A" "A" "A" "A"
##  [91] "A" "A" "A" "A" "A" "A" "A" "A" "A" "A" "A" "A" "A" "A" "A" "A" "A" "A"
## [109] "A" "A"
```

```
block2 <- rep("A",110)
block3 <-rep("A",110)
block4 <- rep("A",110)
block5 <- rep("A",110)
block6 <-rep("A",110)

block1[permutation1 > 55] <- "B"
block1
```

```
##   [1] "A" "B" "A" "A" "A" "B" "B" "A" "A" "B" "A" "A" "B" "B" "B" "B" "A" "B"
##  [19] "B" "A" "B" "A" "A" "A" "B" "B" "B" "A" "B" "B" "B" "A" "B" "B" "B" "B"
##  [37] "B" "B" "A" "B" "B" "B" "B" "A" "A" "B" "A" "A" "A" "B" "B" "A" "A" "B"
##  [55] "A" "A" "A" "A" "B" "A" "A" "A" "B" "B" "B" "A" "B" "B" "B" "B" "A" "B"
##  [73] "B" "A" "A" "B" "A" "A" "A" "B" "A" "B" "A" "A" "B" "B" "B" "A" "A" "A"
##  [91] "A" "A" "A" "B" "A" "B" "B" "A" "B" "A" "A" "A" "A" "A" "B" "A" "B" "B"
## [109] "B" "A"
```

```
block2[permutation2 > 55] <- "B"
block3[permutation3 > 55] <- "B"
block4[permutation1 > 55] <- "B"
block5[permutation2 > 55] <- "B"
block6[permutation3 > 55] <- "B"

sample2 <- c(block1, block2, block3, block4, block5, block6)
table(sample2)
```

```
## sample2
##   A   B
## 330 330
```

5. You can set the seed by using the `set.seed()` function. The seed is the initial value that is fed into the random number generator. For example, let's revisit the random sample from 1. and 2. Running the code twice gives different results:

```
table(sample(c("A", "B"), 660, replace=TRUE))
```

```
##
##   A   B
## 321 339
```

```
table(sample(c("A", "B"), 660, replace=TRUE))
```

```
##
##   A   B
## 332 328
```

With the `set.seed()` function the results are always the same, in this example with a seed of *1234* group A contains 331 participants and group B contains 329.

```
set.seed("1234")
table(sample(c("A", "B"), 660, replace=TRUE))
```

```
##
##   A   B
## 331 329
```

```
set.seed("1234")
table(sample(c("A", "B"), 660, replace=TRUE))
```

```
##
##   A   B
## 331 329
```

Solutions 3 - Analysis of the data

1. First things first - let's plot the data! Initially, there does not appear to be a big difference between the plots, however the small difference we can observe suggests a slightly higher calcium level among the vitamin D group. That difference seems to be more pronounced among artificially fed babies.

```
dat <- read.csv("vitaminD.csv")
caption <- c("Boxplots of artifically fed babies.",
             "Boxplots of breast fed babies.")
boxplot(calcium ~ treatment,
        data = dat[dat$feeding == "Art", ],
        main = "Artifically fed babies")
```

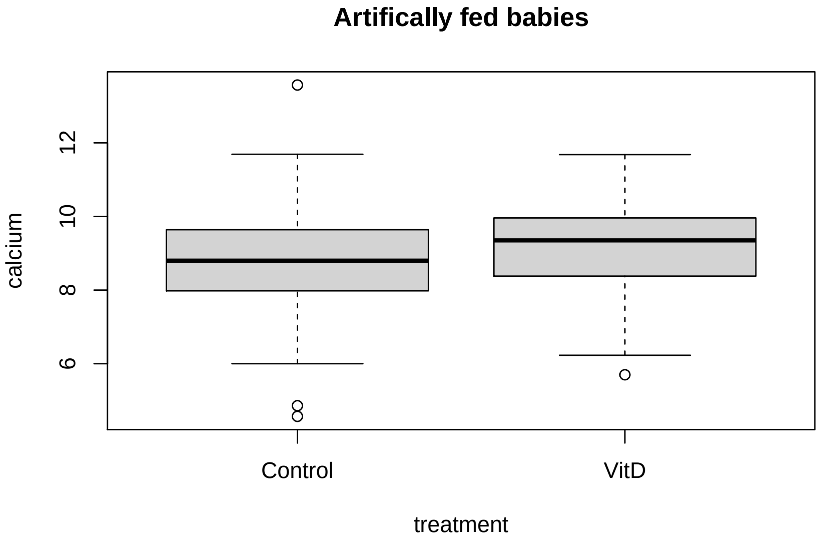

Figure 2: Boxplots of artifically fed babies.

```
boxplot(calcium ~ treatment,
        data = dat[dat$feeding == "Br", ],
        main = "Breast fed babies")
```

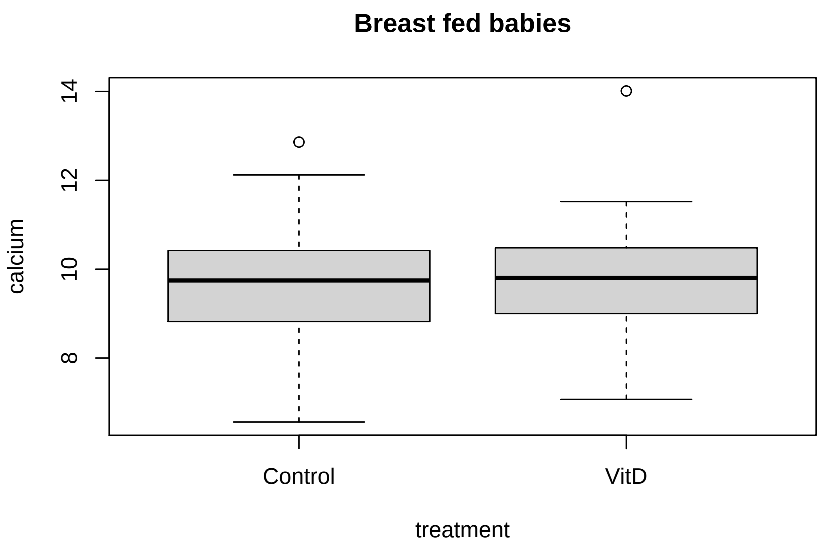

Figure 3: Boxplots of breast fed babies.

Based on the t-tests below, there appears to be a significant difference in calcium level between the control and treatment group among artificially fed babies. The 95% CI is entirely negative indicating that the calcium level in the control group is less than in the treatment group. That result cannot be observed among breast fed babies.

```
t.test(calcium ~ treatment,data = dat[dat$feeding == "Art", ])
```

```
##
##   Welch Two Sample t-test
##
## data:  calcium by treatment
## t = -3.607, df = 364.55, p-value = 0.0003529
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -0.6470332 -0.1904505
## sample estimates:
## mean in group Control    mean in group VitD
##              8.783684              9.202426
```

```
t.test(calcium ~ treatment,data = dat[dat$feeding == "Br", ])
```

```
##
##   Welch Two Sample t-test
##
## data:  calcium by treatment
## t = -0.7963, df = 137.91, p-value = 0.4272
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -0.5238766  0.2230678
## sample estimates:
## mean in group Control    mean in group VitD
##              9.636471              9.786875
```

2. The interaction plot can help us informally assess if an interaction is likely to be present in the data. The lines in the plot below are parallel indicating no interaction between what group a patient is assigned to and the method of feeding.

```r
interaction.plot(dat$treatment,
                 dat$feeding,dat$calcium,
                 main = "Interaction plot of group and type of feed",
                 xlab = "Treatment",
                 ylab = "Mean calcium level")
```

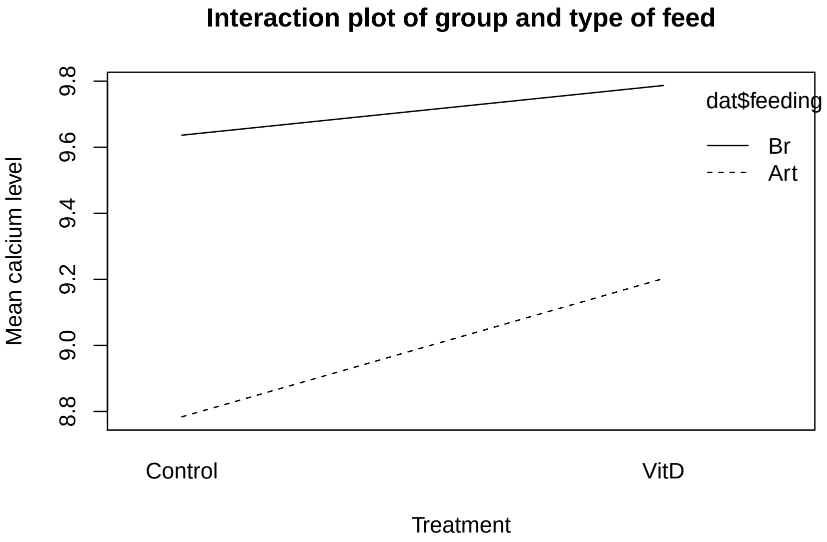

Figure 4: Interactions plot for vitamin D study.

The linear model confirms above graph with a non-significant p-value. We therefore refit the model without an interaction.

```r
model1 <- lm(calcium~feeding*treatment,dat=dat)
summary(model1)
```

```
##
## Call:
## lm(formula = calcium ~ feeding * treatment, data = dat)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -4.2137 -0.8134  0.0563  0.7901  4.7863
##
## Coefficients:
##                          Estimate Std. Error t value Pr(>|t|)
## (Intercept)               8.78368    0.07137 123.071  < 2e-16 ***
## feedingBr                 0.85279    0.13902   6.134 1.53e-09 ***
## treatmentVitD             0.41874    0.11698   3.580 0.000371 ***
## feedingBr:treatmentVitD -0.26834    0.22494  -1.193 0.233364
## ---
## Signif. codes:  0 ’***’ 0.001 ’**’ 0.01 ’*’ 0.05 ’.’ 0.1 ’ ’ 1
##
## Residual standard error: 1.205 on 616 degrees of freedom
## Multiple R-squared:  0.0903,  Adjusted R-squared:  0.08586
## F-statistic: 20.38 on 3 and 616 DF,  p-value: 1.328e-12
```

All terms in model 2 are significant.

```
model2 <- lm(calcium ~ feeding+treatment, data=dat)
summary(model2)
```

```
##
## Call:
## lm(formula = calcium ~ feeding + treatment, data = dat)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -4.2407 -0.7980  0.0680  0.8193  4.7593
##
## Coefficients:
##                Estimate Std. Error t value Pr(>|t|)
## (Intercept)     8.81070    0.06771 130.131  < 2e-16 ***
## feedingBr       0.75030    0.10933   6.863 1.65e-11 ***
## treatmentVitD  0.34617    0.09995   3.463  0.00057 ***
## ---
## Signif. codes:  0 ’***’ 0.001 ’**’ 0.01 ’*’ 0.05 ’.’ 0.1 ’ ’ 1
##
## Residual standard error: 1.205 on 617 degrees of freedom
## Multiple R-squared:  0.08819, Adjusted R-squared:  0.08524
## F-statistic: 29.84 on 2 and 617 DF,  p-value: 4.265e-13
```

The confidence interval of the parameters is given below. Both breast feeding and vitamin D appear to have a positive effect on mean calcium level since both parameters are positive with CIs that don’t include 0. We conclude that babies who are breast fed have a calcium level that is on average 0.75 mg per 100ml higher than artificially fed babies. A similar conclusion can be made for babies in the vitamin D group.

```
cbind(model2$coeff, confint(model2))
```

```
##                             2.5 %    97.5 %
## (Intercept)   8.8106972 8.6777346 8.9436599
## feedingBr     0.7502958 0.5355929 0.9649987
## treatmentVitD 0.3461743 0.1498896 0.5424590
```