Example CoDA Code

Dorothea Dumuid, 13th November 2017

The dataset used in these example analyses is freely available online, in a supplementary file attached to the Fairclough et al 2017 paper: Fitness, fatness and the reallocation of time between children's daily movement behaviours: an analysis of compositional data[1]

`https://ijbnpa.biomedcentral.com/articles/10.1186/s12966-017-0521-z`

```
#first, install and load the required R Packages.
install.packages(compositions)
library(compositions)

install.packages(car)

library(car)

#Download the dataset from https://ijbnpa.biomedcentral.com/articles/10
.1186/s12966-017-0521-z
#it is Additional File 7. Open in Excel and save as a csv file. I renam
ed it "dataset_example.csv"

#Import the data to R, and give the dataset a name (I will call it "dat
a1").

data1 <- read.csv("dataset_example.csv") #you can use this if you have
the dataset saved in the working directory.
#otherwise it might be easier to search for you dataset manually, using
the following command...
data1 <- read.csv(file.choose(), header=TRUE)

names(data1) #tells you the names of the variables

##  [1] "Child.ID"                  "School"
##  [3] "Sex"                       "Decimal.Age..y."
##  [5] "IMD.Decile"                "Height..cm."
##  [7] "Mass..cm."                 "BMI"
##  [9] "zBMI"                      "IOTF.grade"
## [11] "Waist.Circumference..cm." "WHtR"
## [13] "Total.20m.Shuttles"        "Wear.time"
## [15] "Sedentary"                 "Light"
## [17] "Moderate"                  "Vigorous"
## [19] "MVPA"                      "sleep"
## [21] "X24hours"                  "X"
## [23] "X.1"                       "X.2"
## [25] "X.3"
```

```
#there are some empty rows at the end of the dataset, so we get rid of
them here
data1 <- data1[1:169,]  #only keep the first 169 rows...

#get activity behaviours from the dataset, and give them new names
sleep <- data1$sleep
sb <- data1$Sedentary  #sedentary is "sb"
lpa <- data1$Light
mpa <- data1$Moderate
vpa <- data1$Vigorous

#amalgamate MVPA to one variable, call it "mvpa"
mvpa <- mpa+vpa

#make activity vectors- this puts all the behaviours together (the cbin
d() function binds the component columns together)
activity <- cbind(sleep,sb,lpa,mvpa)
head(activity) #this shows us some of the observations in our activity
variable.
```

```
##      sleep    sb   lpa mvpa
## [1,] 483.2 548.4 386.0 22.4
## [2,] 531.7 508.4 359.2 40.7
## [3,] 567.5 456.7 379.7 36.1
## [4,] 573.0 526.1 329.6 11.3
## [5,] 564.0 481.4 389.1  5.4
## [6,] 546.7 525.1 335.7 32.5
```

```
#Now we designate this variable of time-use behavoiurs as compositional
- using the acomp(activity) tells R that "activity" is a composition. T
he acomp() function also expresses the composition in proportions which
sum to 1.
#We call the composition "act.comp"
act.comp <- acomp(activity)

mean(act.comp) #this is the mean composition of the sample, expressed i
n proportions.
```

```
##     sleep        sb       lpa      mvpa
## 0.3809541 0.3543493 0.2463897 0.0183069
## attr(,"class")
## [1] acomp
```

```
clo(mean(act.comp),total=1440) #this is the mean composition, adjusted
to sum to 1440 min/day.
```

```
##     sleep        sb       lpa      mvpa
## 548.57385 510.26305 354.80116  26.36194
```

```
#get outcome measures from the dataset
bmi <- data1$zBMI #this is zBMI
```

```r
#get covariates
ses <- as.factor(data1$IMD.Decile) #tell R that this is a factor (might be better to treat as integer or continuous, but we'll just use factor in our example here)

age <- data1$Decimal.Age..y. # a continuous variable
sex <- factor(data1$Sex) # a factor (M/F)



#descriptive statistics
#center
(m=mean(act.comp)) #because R knows that act.comp is a composition, it gives you the geometric mean, in proportions (i.e., the values will all add to 1). This is called the compositional mean. If we are interested in the mean values in minutes, we can re-adjust the composiitonal mean to make the parts altogether sum to 1440 minutes). This does not change the composition, because the ratios between parts remain the same. We are simply zooming in/out but keeping the aspect ratio constant.
```

```
##     sleep        sb       lpa      mvpa
## 0.3809541 0.3543493 0.2463897 0.0183069
## attr(,"class")
## [1] acomp
```

```r
round(clo(m, total=1440),digits=1)
```

```
## sleep    sb   lpa  mvpa
## 548.6 510.3 354.8  26.4
```

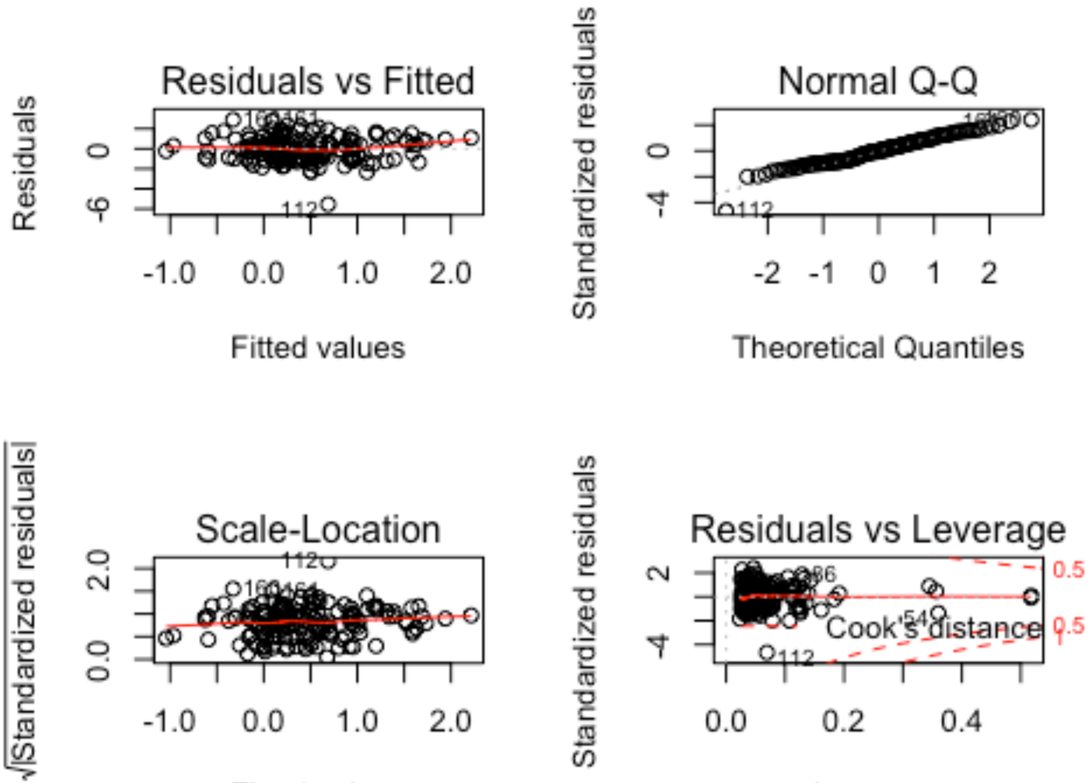
```r
#variation matrix, rounded to make it nicer: gives us an idea of the spread of the compositional parts, pairwise.
#values closer to zero are more dependent on each other. Zero=exactly proportional.
#this replaces univariate standard deviation.
round(variation(act.comp), digits=3)
```

```
##       sleep    sb   lpa  mvpa
## sleep 0.000 0.033 0.037 0.355
## sb    0.033 0.000 0.094 0.474
## lpa   0.037 0.094 0.000 0.272
## mvpa  0.355 0.474 0.272 0.000
```

```r
#Regression models

#The complete composition (i.e., all components) cannot be used as an explanatory variables because there is singularity or perfect multi-collinearity. Because all parts sum to 1 (or 24 h or 1440 minutes) one of the parts is always completely explained by the sum of the other parts (it will equal 1 - sum of remaining parts (if in proportions))
```

```
#This is why we use isometric log ratios to represent the composition i
n the linear model. The isometric log ratios (ilr) are created using an
orthonormal basis, therefore multi-collinearity is not an issue. The or
der of ratios and parts do not matter as the ratios are orthogonal rota
tions of each other.

#There are always 1 less ilr than there are parts in the composition (t
his is because one part in the composition can be always be explained b
y the remaining parts, therefore is superfluous, and the real dimension
of compositional data is one less than the number of parts)

#the set of ilr coordinates contains all the relative information about
the composition.

#the set of ilr coordinates are coordinates in real Euclidean space. Th
is is why we can use the set of ilr coordinates to represent the compos
ition in statistical models deisgned for read vectors in Euclidean spac
e (e.g. regression models)



#first make ilr coordinates using the ilr() command that is default in
the compositions package (call them "ilr.comp")
ilr.comp <- ilr(act.comp)
head(as.data.frame(ilr.comp)) #this is what the ilr coordinates look li
ke

##              V1         V2        V3
## 1  0.08950155 -0.2350534 -2.631589
## 2 -0.03168608 -0.3019361 -2.099402
## 3 -0.15359351 -0.2394395 -2.207144
## 4 -0.06038296 -0.4166653 -3.215797
## 5 -0.11197441 -0.2384497 -3.872979
## 6 -0.02850452 -0.3817338 -2.292077

#MODELS

#Explanatory compositions

#Now we use the ilr coordinates as explanatory variables in our linear
model.
#zbmi is the dependent variable, with covariates of SES, sex and age
lm.bmi=lm(bmi~ilr.comp+ses+sex+age)
(sum.bmi=car::Anova(lm.bmi))

## Anova Table (Type II tests)
##
## Response: bmi
##            Sum Sq  Df F value    Pr(>F)
## ilr.comp   31.953   3  7.0118 0.0001872 ***
```

```
## ses          9.872   7  0.9284 0.4862526
## sex          3.546   1  2.3341 0.1285942
## age          6.934   1  4.5647 0.0341964 *
## Residuals 236.969 156
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

#we use the car::Anova command because the Anova from the car package i
s a type II test- it adjusts each term in the model for each other term
. It does not do a step-wise adjustment as some of the other anova comm
ands do. This is important if you have a number of covariates you want
to adjust for.

#the ilr coordinates (ilr.comp) are significant predictors of zBMI.

#should check out the model fit
#model diagnostics
par(mfrow=c(2,2))
plot(lm.bmi) #model diagnostics look great
```



```
#Use the linear model to predict change in zBMI for change in compositi
on.
```

```
##MODEL ISOTEMPORAL 1:1 reallocations, pairwise...This analysis is desc
ribed in detail in Dumuid et al 2017b.[2]

# Firstly, predict zBMI for the mean composition (here we use the compo
sition in proportions, i.e., as proportions which sum to 1... it just m
akes it easier down the track)
# so the mean composition (in proportions) was labelled "m"

#we also need to find the mean values of the covariates so that these c
an be kept constant in the linear models. Then the only thing that chan
ges is time reallocation.
#some covariates were factors or integers, so the "arithmetic mean" wil
l not make sense. We will have to use the most common level, or round t
o the nearest whole number.

#to find average condition for integer covariate (SES). Must be a whole
number to match the original data.
as.integer(as.character(ses))

 (m.ses<-mean(as.integer(as.character(ses))))

## [1] 2.449704

as.character(round(m.ses,0))

## [1] "2"

m.ses=ses[ses==as.character(round(m.ses,0))][1] #round to nearest whole
number
m.ses

## [1] 2
## Levels: 1 2 3 4 5 6 7 9

(m.ses<-ses[ses=="2"][1]) # here we designate the median "m.ses" to be
"2".

## [1] 2
## Levels: 1 2 3 4 5 6 7 9

#to find average condition for factor coordinate (sex)
m.sex <- names(table(sex))[which.max(table(sex))] #average condition is
"0"

m.age <- mean(age) #this one is easy, because age is a continuous varia
ble. Just find the arithmetic mean.

#Predictions from model....
#Now we can predict zBMI for the "baseline" situation, we will call the
predicted zBMI "mean.pred"
```

```
(mean.pred <- predict(lm.bmi, newdata=list(ilr.comp=ilr(m), # "newdata"
is the data we will use to predict our new zBMI. We need to use the ilr
of the mean composition as that's what we used in the initial model.
                                       ses=m.ses,  #here the mean of th
e covariates...
                                       age=m.age,
                                       sex=m.sex)))
```

```
##         1
## 0.2202949
```

```
# Now predict zBMI for a new composition, where 15 minutes have been re
allocated between two activity behaviours, while the remaining behaviou
rs are held constant.

ch.time <- 15/1440 #ch.time is the amount of time reallocated (divide b
y 1440 to put 15 minutes in to the proportional scale- remember that "m
", our mean composition is in this scale: the components add to 1)


#Now we some new compositions where 15 minutes are reallocated from one
component to another. In the code, m[1] refers to the fist component of
our composition, from above, this is sleep. m[2] is sb, m[3] is lpa and
m[4] is mvpa. So comp1 is a composition where 15 minutes have been adde
d to sleep and 15 minutes have been subtracted from sb. Lpa and MVPA ha
ve been kept constant at their mean value.

comp1 <- acomp(c(m[1]+ch.time, m[2]-ch.time, m[3], m[4]))

#and now we predict zBMI for the new composition....

(comp1.pred <- predict(lm.bmi, newdata=list(ilr.comp=ilr(comp1),
                                        ses=m.ses,  #here the covariate
s are kept constant at the mean...
                                       age=m.age,
                                       sex=m.sex)))
```

```
##         1
## 0.2715709
```

```
#So to find the difference in zBMI for the reallocation of 15 minutes f
rom sleep to sb, we subtract the estimated zBMI from comp1 from the ref
erence comp.

(sleep.to.sb <- comp1.pred- mean.pred)
```

```
##          1
## 0.05127603
```

```
#this is predicted increase in zBMI when 15 miutes are reallocated from
sleep to sb, at the mean composition of the sample.
```

```
###What about increasing one part, at the equal expense of all remaining parts?  "1:many reallocation". This is described in Dumuid et al. 2017a[3]

#We can use the same model as above, but have to change our new predictive composition.

m #(this is the mean composition, in proportions)

##     sleep        sb       lpa      mvpa
## 0.3809541 0.3543493 0.2463897 0.0183069
## attr(,"class")
## [1] acomp

#so increase sleep by 60 minutes, getting time from the remaining behaviours at equal proportions looks like this:
r <- 60/1440/m[1] #this is what we have to multiply sleep by to increase it by 60 minutes at the mean.
s <- r* m[1]/(1-m[1]) #this is what we have to multiply each of the remaining parts by, to reduce them evenly so that the daily total of 1440 minutes is maintained.
c1 <- cbind(m[1]*(1+r),m[2]*(1-s),m[3]*(1-s),m[4]*(1-s))  #create the new composition, c1
#just check that the minutes are what we want
clo(c1, total=1440)  #the new composition

##           [,1]     [,2]     [,3]     [,4]
## sleep 608.5738 475.9183 330.9203 24.58757

clo(m, total=1440)  #the mean composition. We see that sleep has been increased by 60 minutes, and the remaining have been decreased.

##     sleep        sb       lpa      mvpa
## 548.57385 510.26305 354.80116  26.36194

#Now use c1 in the predictive model
comp.c1.pred <- predict(lm.bmi, newdata=list(ilr.comp=ilr(c1),
                                         ses=m.ses,
                                         age=m.age,
                                         sex=m.sex))

comp.c1.pred - mean.pred  #estimated increase in zBMI when sleep is increased by 60 mins, at the expense of the remaining components equally.

##         1
## 0.1703519

#You can change the predictive composition to be whatever interests you, then put it into the predictive ilr linear model.
```

```
#For the above 1:many reallocations, you can obtain a p-value which describes the significance of this reallocation. To do this, you need to use a specific type of ilr transformation. This is described in the Chastin et al.(2015) paper[4]


#This tells R how you want the isometric log ratios to be set up. (sbp = sequential binary partition)


sbp  <-  matrix(c( 1, -1, -1,-1, #the sign matrix for the ilr
                   0, 1, -1, -1,
                   0, 0, 1, -1),
                ncol=4, byrow=TRUE)
colnames(sbp)  <- c("Sleep", "Sed", "LPA", "MVPA")
rownames(sbp) <- c("ilr1", "ilr2", "ilr3")
sbp #we can see that each row corresponds to an isometric log ratio coordinate, and each column to a component.
```

```
##      Sleep Sed LPA MVPA
## ilr1     1  -1  -1   -1
## ilr2     0   1  -1   -1
## ilr3     0   0   1   -1
```

```
#the first ilr has sleep in its numerator and the remaining components (i.e., the geometric mean of the remaining components) in its denominator. This is the ilr of interest to us, because it captures all relative information regarding sleep.

psi  <-  gsi.buildilrBase(t(sbp)) #this makes the above sbp into something useable for R

ilr.comp1 <- ilr(act.comp, V=psi) # V=psi speficies that we want to use the sbp from above, to make the special type of ilr where the first coordinate is the first component relative to the geometric mean of the remaining components
```

$$ilr1 = \sqrt{\frac{3}{4}}\ln\frac{Sleep}{\sqrt[3]{Sed \cdot LPA \cdot MVPA}}\,;\ ilr2 = \sqrt{\frac{2}{3}}\ln\frac{Sed}{\sqrt{LPA \cdot MVPA}}\,;\ ilr3 = \frac{1}{\sqrt{2}}\ln\frac{LPA}{MVPA}$$

```
#Now we can use this set of ilrs in our model
lm.bmi1 <- lm(bmi~ilr.comp1+ses+sex+age)
summary(lm.bmi1)  #here we see the relationship between the first ilr coordinate and sleep is not significant, but it is positive. Therefore we can say that, as sleep increases at the expense of equal decrease in sed, lpa and mvpa, zbmi is predicted to increase. This supports what we saw when we performed our predictions.
```

```
## 
## Call:
## lm(formula = bmi ~ ilr.comp1 + ses + sex + age)
## 
## Residuals:
##     Min      1Q  Median      3Q     Max 
## -5.5479 -0.9345  0.0000  0.8399  2.8970 
## 
## Coefficients:
##                Estimate Std. Error t value Pr(>|t|)   
## (Intercept)   -9.530806   3.327566  -2.864  0.00476 **
## ilr.comp1ilr1  1.133903   1.053249   1.077  0.28333   
## ilr.comp1ilr2 -0.616482   0.492951  -1.251  0.21295   
## ilr.comp1ilr3  1.322203   0.547279   2.416  0.01685 * 
## ses2          -0.005087   0.228894  -0.022  0.98230   
## ses3           0.431123   0.307571   1.402  0.16299   
## ses4          -0.391475   0.440406  -0.889  0.37543   
## ses5          -0.258360   0.892037  -0.290  0.77248   
## ses6           1.092290   0.734886   1.486  0.13921   
## ses7          -0.616311   1.253007  -0.492  0.62351   
## ses9          -0.161242   0.426155  -0.378  0.70567   
## sex1           0.341942   0.223817   1.528  0.12859   
## age            0.683116   0.319733   2.137  0.03420 * 
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## 
## Residual standard error: 1.232 on 156 degrees of freedom
## Multiple R-squared:  0.1856, Adjusted R-squared:  0.1229 
## F-statistic: 2.962 on 12 and 156 DF,  p-value: 0.0009588
```

```r
#Let's try MVPA, relative to the remainin components.
#we could re-order our components to put MVPA first, and use the same sign matrix (sbp) as before, or we could keep the same order of components and modify the sign matrix. Let's change the sign matrix, remembering MVPA is the fourth column and we want it as the numerator of the first ilr (give it a positive sign). We also want only the first ilr to contain all the information about MVPA, so we will not have MVPA in any of the remaining ilr coordinates.
sbp2  <-  matrix(c( -1, -1, -1,1, #the sign matrix for the ilr
                -1, -1, 1, 0,
                -1, 1, 0, 0),
             ncol=4, byrow=TRUE)
colnames(sbp2)  <- c("Sleep", "Sed", "LPA", "MVPA")
rownames(sbp2) <- c("ilr1", "ilr2", "ilr3")
sbp2
```

```
##      Sleep Sed LPA MVPA
## ilr1    -1  -1  -1    1
## ilr2    -1  -1   1    0
## ilr3    -1   1   0    0
```

```r
psi2  <-  gsi.buildilrBase(t(sbp2))

#Now make the ilrs using this new sign matrix (sbp)
ilr.comp2 <- ilr(act.comp, V=psi2) #this set of ilrs has MVPA:remaining as the first coordinate
```

$$ilr1 = \sqrt{\frac{3}{4}}\ln\frac{MVPA}{\sqrt[3]{LPA\cdot Sed\cdot Sleep}};\ ilr2 = \sqrt{\frac{2}{3}}\ln\frac{LPA}{\sqrt{Sed\cdot Sleep}};\ ilr3 = \frac{1}{\sqrt{2}}\ln\frac{Sed}{Sleep}$$

```r
#And run the linear model again, with our new ilrs.
lm.bmi2 <- lm(bmi~ilr.comp2+ses+sex+age)
summary(lm.bmi2)
```

```
##
## Call:
## lm(formula = bmi ~ ilr.comp2 + ses + sex + age)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -5.5479 -0.9345  0.0000  0.8399  2.8970
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)   -9.530806   3.327566  -2.864  0.00476 **
## ilr.comp2ilr1 -1.166929   0.263004  -4.437 1.72e-05 ***
## ilr.comp2ilr2  0.639835   0.734245   0.871  0.38486
## ilr.comp2ilr3 -1.281754   1.021549  -1.255  0.21146
## ses2          -0.005087   0.228894  -0.022  0.98230
## ses3           0.431123   0.307571   1.402  0.16299
## ses4          -0.391475   0.440406  -0.889  0.37543
## ses5          -0.258360   0.892037  -0.290  0.77248
## ses6           1.092290   0.734886   1.486  0.13921
## ses7          -0.616311   1.253007  -0.492  0.62351
## ses9          -0.161242   0.426155  -0.378  0.70567
## sex1           0.341942   0.223817   1.528  0.12859
## age            0.683116   0.319733   2.137  0.03420 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.232 on 156 degrees of freedom
## Multiple R-squared:  0.1856, Adjusted R-squared:  0.1229
## F-statistic: 2.962 on 12 and 156 DF,  p-value: 0.0009588
```

```r
#You will notice the regression beta for the first ilr is significant, and negative. Therefore, as MVPA is increased at the equal expense of the remaining activities, zBMI is predicted to decrease.
```

```
#why not make a change-prediction for MVPA (remember, MVPA is m[4], the
fourth component)
r <- 60/1440/m[4] #this is what we have to multiply mvpa by to increase
it by 60 minutes at the mean.
s <- r* m[4]/(1-m[4]) #this is what we have to multiply each of the rem
aining parts by, to reduce them evenly so that the daily total of 1440
minutes is maintained.
c2 <- cbind(m[1]*(1-s),m[2]*(1-s),m[3]*(1-s),m[4]*(1+r))  #create the n
ew composition where mvpa is increased, c2
#just check that the minutes are what we want
clo(c2, total=1440)  #the new composition

##              [,1]     [,2]     [,3]     [,4]
## sleep 525.2904 488.6056 339.7421 86.36194

clo(m, total=1440)  #the mean composition. We see that mvpa has been in
creased by 60 minutes, and the remaining have been decreased.

##     sleep        sb       lpa      mvpa
## 548.57385 510.26305 354.80116  26.36194

#Now use c2 in the predictive model. Here I am using the original model
to predict, lm.bmi, which has the default ilr transformation. We must b
e consistent with which ilr transformation is being used and ensure the
new ilr set we are using to predict zBMI is created by the same transfo
rmation (sbp) as the ilr set used in the original linear model).
comp.c2.pred <- predict(lm.bmi, newdata=list(ilr.comp=ilr(c2), #the mod
el lm.bmi contained ilrs created by the default ilr transformation in R
compositionsm so can use the default ilr transformation for our new c2.
                                         ses=m.ses,
                                         age=m.age,
                                         sex=m.sex))

comp.c2.pred - mean.pred

##         1
## -1.243022

#so zBMI is predicted to decrease by 1.24 when MVPA is increased from t
he mean by 60 min at the expense of the remaining activities equally.

#Just to show that you can use any model for these predictions....here
using the lm.bmi2 with specific ilr transformation...
lm.bmi2 <- lm(bmi~ilr.comp2+ses+sex+age) #ilr.comp2 is made by sbp of V
=psi2)
comp.c2.pred <- predict(lm.bmi2, newdata=list(ilr.comp2=ilr(c2, V=psi2)
, #here must specify the sbp as V=psi2)
                                         ses=m.ses,
                                         age=m.age,
                                         sex=m.sex))
```

```
comp.c2.pred - mean.pred

##         1
## -1.243022

#Notice this is the same predicted difference as we got before.


#LINEAR MODELS :: COMPOSITION IS THE DEPENDENT
#let's see if zbmi predicts the composition.

lmodel=lm(ilr.comp~bmi)
car::Anova(lmodel)

##
## Type II MANOVA Tests: Pillai test statistic
##     Df test stat approx F num Df den Df    Pr(>F)
## bmi  1   0.11349   7.0409      3    165 0.0001751 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

#this means that zBMI is a significant predictor of the composition. We
use the car::Anova command because the Anova from the car package is a
type II test- it adjusts each term in the model for each other term. It
does not do a step-wise adjustment as some of the other anova commands
do. This is important if you have a number of covariates you want to ad
just for.

#to get an idea of what the relationship looks like, we can make some p
redictions using the linear model.

#what is the time-use composition at varying points of zBMI?
#let's predict time-use composition for zBMI = -3, and the continue to
predict for each level of zBMI, until zBMI = 3

(ilr.pred.bmi3a <- predict(lmodel, newdata=list(bmi=-3))) #the predicte
d composition is in ilr coordinates

##           [,1]       [,2]      [,3]
## 1 -0.06864288 -0.2765817 -2.059827

(comp.bmi3a <- ilrInv(ilr.pred.bmi3a)) #we express the ilr coordinates
as a composition (i.e., in proportions)

##   1         2         3         4
## 1 0.3751979 0.3404874 0.2547223 0.02959243
## attr(,"class")
## [1] acomp
```

```
(c1 <- clo(comp.bmi3a, total=1440)) #we linearly adjust the proportions
(zoom in), to sum to 1440 minutes, so we can consider minutes/day

##         1        2        3       4
## 1 540.285 490.3018 366.8001 42.6131

ilr.pred.bmi2a <- predict(lmodel, newdata=list(bmi=-2))
comp.bmi2a <- ilrInv(ilr.pred.bmi2a)
c2 <- clo(comp.bmi2a, total=1440)

ilr.pred.bmi1a <- predict(lmodel, newdata=list(bmi=-1))
comp.bmi1a <- ilrInv(ilr.pred.bmi1a)
c3 <- clo(comp.bmi1a, total=1440)

ilr.pred.bmi0 <- predict(lmodel, newdata=list(bmi=0))
comp.bmi0 <- ilrInv(ilr.pred.bmi0)
c4 <- clo(comp.bmi0, total=1440)

ilr.pred.bmi1=predict(lmodel, newdata=list(bmi=1))
comp.bmi1=ilrInv(ilr.pred.bmi1)
c5 <- clo(comp.bmi1, total=1440)

ilr.pred.bmi2=predict(lmodel, newdata=list(bmi=2))
comp.bmi2=ilrInv(ilr.pred.bmi2)
c6 <- clo(comp.bmi2, total=1440)

ilr.pred.bmi3=predict(lmodel, newdata=list(bmi=3))
comp.bmi3=ilrInv(ilr.pred.bmi3)
c7 <- clo(comp.bmi3, total=1440)


 #lets bind all the predicted time-use compositions together into a tab
le- use the rbind() command to rowbind.
ans <- rbind(c1,c2,c3,c4,c5,c6,c7)
(ans <- data.frame(ans)) #this is the table of predicted time-use compo
sitions

## zBMI    Sleep     Sed      LPA      MVPA
## -3    540.2850 490.3018 366.8001 42.61310
## -2    543.0512 496.3685 363.5065 37.07378
## -1    545.5137 502.2177 360.0328 32.23574
## 0     547.7041 507.8733 356.4080 28.01457
## 1     549.6508 513.3566 352.6576 24.33497
## 2     551.3789 518.6870 348.8041 21.13002
## 3     552.9109 523.8817 344.8669 18.34049

#looking at the predicted compositions we see that as zBMI increases, s
leep & sedentary are predictd to increase, whereas LPA and MVPA are pre
dicted to decrease.
```

1. Fairclough SJ, Dumuid D, Taylor S, et al. Fitness, fatness and the reallocation of time between children's daily movement behaviours: an analysis of compositional data. *Int J Behav Nutr Phys Act*. 2017; 14: DOI: 10.1186/s12966-017-0521-z.
2. Dumuid D, Pedisic Z, Stanford T, et al. The compositonal isotemporal substiution model: a method for estimating changes in a health outcome for reallocation of time between sleep, sedentary behaviour and physical activity *Stat Methods Med Res*. 2017: in press.
3. Dumuid D, Stanford T, Martín-Fernández J, et al. Compositional data analysis for physical activity, sedentary time and sleep research. *Stat Methods Med Res*. 2017: doi: 10.1177/0962280217710835.
4. Chastin SF, Palarea-Albaladejo J, Dontje ML, et al. Combined effects of time spent in physical activity, sedentary behaviors and sleep on obesity and cardio-metabolic health markers: A novel compositional data analysis approach. *PLoS ONE*. 2015; 10: e0139984.