

Some Goldilocks CoDA example codes

Dot Dumuid

19 August 2022

```
require(compositions)

## Loading required package: compositions

## Welcome to compositions, a package for compositional data analysis.
## Find an intro with "? compositions"

##
## Attaching package: 'compositions'

## The following objects are masked from 'package:stats':
##
##   anova, cor, cov, dist, var

## The following objects are masked from 'package:base':
##
##   %*%, norm, scale, scale.default

require(robustbase)

## Loading required package: robustbase

require(ggplot2)

## Loading required package: ggplot2

require(dplyr)

## Loading required package: dplyr

##
## Attaching package: 'dplyr'

## The following objects are masked from 'package:stats':
##
##   filter, lag

## The following objects are masked from 'package:base':
##
##   intersect, setdiff, setequal, union

require(lme4)

## Loading required package: lme4
```

```

## Loading required package: Matrix

require(lmerTest)

## Loading required package: lmerTest

##
## Attaching package: 'lmerTest'

## The following object is masked from 'package:lme4':
##
##   lmer

## The following object is masked from 'package:robustbase':
##
##   carrots

## The following object is masked from 'package:stats':
##
##   step

setwd("C:/Users/dumuid/OneDrive - University of South Australia/R club")

#get example dataset, using Stuart Fairclough's publicly available data from supplementary file of this paper
 #(Thank you Stuart)
## https://ijbnpa.biomedcentral.com/articles/10.1186/s12966-017-0521-z

#please see Dumuid, D., Olds, et al https://doi.org/10.1136/jech-2021-216686 for a published example of this kind of analysis

data = read.csv("Fairclough.csv")

#Look at the dataset
#top few rows
head(data)

##   Child.ID School Sex Decimal.Age..y. IMD.Decile Height..cm. Mass..cm.   BMI
## 1      NA     1   1      10.0           2      143.8     36.2 17.51
## 2      NA     1   1      10.6           2      146.6     36.1 16.80
## 3      NA     1   0       9.9           9      137.4     27.4 14.51
## 4      NA     1   0      10.5           1      142.8     54.7 26.82
## 5      NA     1   0      10.6           2      154.7     76.3 31.88
## 6      NA     1   1      10.7           9      144.9     32.7 15.57
##   zBMI IOTF.grade Waist.Circumference..cm. WHtR Total.20m.Shuttles Wear.time
## 1  0.55         1           63 0.44           21      24.0
## 2  0.05         1           64 0.44           45      23.9
## 3 -1.31         0           53 0.39           37      23.8
## 4  2.64         2           88 0.62            9      24.0
## 5  3.30         2           94 0.61           11      22.8
## 6 -0.69         1           58 0.40           27      24.0

```

```
## Sedentary Light Moderate Vigorous MVPA sleep X24hours
## 1 548.4 386.0 15.0 7.4 22.4 483.2 1440
## 2 508.4 359.2 23.7 17.0 40.7 531.7 1440
## 3 456.7 379.7 22.1 14.0 36.1 567.5 1440
## 4 526.1 329.6 7.1 4.2 11.3 573.0 1440
## 5 481.4 389.1 4.2 1.2 5.4 564.0 1440
## 6 525.1 335.7 21.5 11.0 32.5 546.7 1440
```

#select the variables that we might want (we will rename some of the complex names during the selection process)

```
d1 = data %>% dplyr::select(school=School, sex=Sex, age=Decimal.Age.y., ses=IMD.D
ecile,
                           height=Height..cm., mass=Mass..cm., bmi=BMI, zbmi=zBMI
,
                           owob=IOTF.grade, waist=Waist.Circumference..cm., wht=W
HtR,
                           shuttles=Total.20m.Shuttles, sb=Sedentary, lpa=Light,
mvpa=MVPA, sleep)
```

`head(d1)`

```
## school sex age ses height mass bmi zbmi owob waist wht shuttles sb
## 1 1 1 10.0 2 143.8 36.2 17.51 0.55 1 63 0.44 21 548.4
## 2 1 1 10.6 2 146.6 36.1 16.80 0.05 1 64 0.44 45 508.4
## 3 1 0 9.9 9 137.4 27.4 14.51 -1.31 0 53 0.39 37 456.7
## 4 1 0 10.5 1 142.8 54.7 26.82 2.64 2 88 0.62 9 526.1
## 5 1 0 10.6 2 154.7 76.3 31.88 3.30 2 94 0.61 11 481.4
## 6 1 1 10.7 9 144.9 32.7 15.57 -0.69 1 58 0.40 27 525.1
## lpa mvpa sleep
## 1 386.0 22.4 483.2
## 2 359.2 40.7 531.7
## 3 379.7 36.1 567.5
## 4 329.6 11.3 573.0
## 5 389.1 5.4 564.0
## 6 335.7 32.5 546.7
```

#see what sort of variables R thinks these are

`str(d1)`

```
## 'data.frame': 169 obs. of 16 variables:
## $ school : int 1 1 1 1 1 1 1 1 1 1 ...
## $ sex : int 1 1 0 0 0 1 1 1 0 1 ...
## $ age : num 10 10.6 9.9 10.5 10.6 10.7 9.9 10.5 10 9.8 ...
## $ ses : int 2 2 9 1 2 9 3 3 3 2 ...
## $ height : num 144 147 137 143 155 ...
## $ mass : num 36.2 36.1 27.4 54.7 76.3 32.7 32.8 37.3 46.5 36.8 ...
## $ bmi : num 17.5 16.8 14.5 26.8 31.9 ...
## $ zbmi : num 0.55 0.05 -1.31 2.64 3.3 -0.69 0.2 1.61 1.63 0.38 ...
## $ owob : int 1 1 0 2 2 1 1 2 2 1 ...
## $ waist : int 63 64 53 88 94 58 60 67 70 67 ...
## $ wht : num 0.44 0.44 0.39 0.62 0.61 0.4 0.43 0.5 0.48 0.46 ...
```

```
## $ shuttles: int 21 45 37 9 11 27 13 13 22 19 ...
## $ sb : num 548 508 457 526 481 ...
## $ lpa : num 386 359 380 330 389 ...
## $ mvpa : num 22.4 40.7 36.1 11.3 5.4 32.5 24.5 15.8 44.5 13.3 ...
## $ sleep : num 483 532 568 573 564 ...
```

#will need to change some of these variables to factors.

```
d1$school = factor(d1$school)
d1$sex = factor(d1$sex)
d1$owob = factor(d1$owob)
```

#compositional variables

```
d1$comp = d1 %>% select(sleep, sb, lpa, mvpa)
head(d1$comp)
```

```
## 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
```

#are there any zeros?

```
missingSummary(d1$comp)
```

```
## missingType
## variable NMV BDL MAR MNAR SZ Err
## sleep 169 0 0 0 0 0
## sb 169 0 0 0 0 0
## lpa 169 0 0 0 0 0
## mvpa 169 0 0 0 0 0
```

#No.

#Make ilrs - use default ilr transformation as we are not going to interpret individual regression coefficients.

```
d1$ilrs = ilr(d1$comp)
```

#will save them as separate variables rather than a set of ilrs (helps with prediction later)

```
d1$ilr1 = d1$ilrs[,1]
d1$ilr2 = d1$ilrs[,2]
d1$ilr3 = d1$ilrs[,3]
```

```
#####
```

#now that data is set up, start making the analytical models.

#Research Q - what time-use compositions are associated with the best fitness (highest number of shuttles)

#and fatness (lowest WHT, or we could use a target WHT ratio)

#the first step is to get the best fitting models - I'm using sample specific z-scores for the outcomes (scale()) to make interpretation easier

```

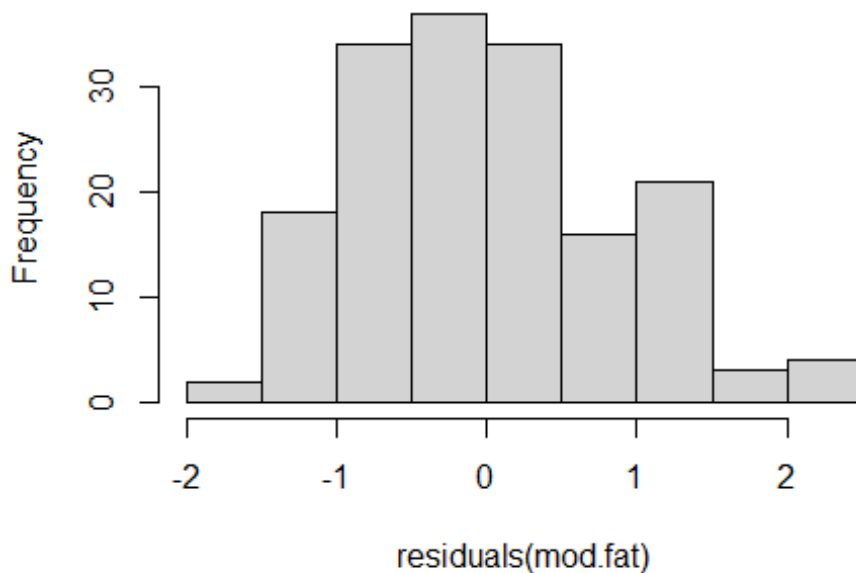
#model for fatness
mod.fat = lmer(scale(log(wht)) ~ cbind(ilr1, ilr2, ilr3) + sex + age + ses + (1|school) , data=d1) #observations are nested in schools, school is a random intercept
summary(mod.fat)

## Linear mixed model fit by REML. t-tests use Satterthwaite's method [
## lmerModLmerTest]
## Formula: scale(log(wht)) ~ cbind(ilr1, ilr2, ilr3) + sex + age + ses +
## (1 | school)
## Data: d1
##
## REML criterion at convergence: 453.4
##
## Scaled residuals:
##   Min       1Q   Median       3Q      Max
## -1.7842 -0.7111 -0.1301  0.6050  2.7289
##
## Random effects:
##   Groups Name          Variance Std.Dev.
##   school (Intercept) 0.06663  0.2581
##   Residual              0.80356  0.8964
## Number of obs: 169, groups: school, 7
##
## Fixed effects:
##              Estimate Std. Error      df t value Pr(>|t|)
## (Intercept)   -5.70964    2.48133 161.85151  -2.301    0.0227 *
## cbind(ilr1, ilr2, ilr3)ilr1  -0.68665    0.73565 158.94634  -0.933    0.3520
## cbind(ilr1, ilr2, ilr3)ilr2   1.15277    0.53209 161.87790   2.166    0.0317 *
## cbind(ilr1, ilr2, ilr3)ilr3  -1.02957    0.19180 161.83046  -5.368 2.72e-07 ***
## sex1              0.38943    0.16010 160.68316   2.432    0.0161 *
## age                0.33140    0.23824 161.84999   1.391    0.1661
## ses               -0.04033    0.04079  68.12136  -0.989    0.3263
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Correlation of Fixed Effects:
##              (Intr) c(1,i2,i3)1 c(1,i2,i3)2 c(1,i2,i3)3 sex1  age
## c(1,i2,i3)1  0.112
## c(1,i2,i3)2 -0.067  0.457
## c(1,i2,i3)3  0.171  0.209  -0.428
## sex1         -0.073 -0.100   0.290   -0.481
## age          -0.979 -0.021   0.051   0.008  -0.033
## ses           0.006 -0.009   0.011   0.046   0.006 -0.040

hist(residuals(mod.fat))#Looks fairly good

```

Histogram of residuals(mod.fat)



```
car::Anova(mod.fat, test.statistic= "F") #composition is a significant predictor of zBMI.
```

```
## Analysis of Deviance Table (Type II Wald F tests with Kenward-Roger df)
##
```

```
## Response: scale(log(wht))
```

```
##
```

	F	Df	Df.res	Pr(>F)	
## cbind(ilor1, ilr2, ilr3)	9.3743	3	161.360	9.522e-06	***
## sex	5.8389	1	160.544	0.0168	*
## age	1.8778	1	161.834	0.1725	
## ses	0.8725	1	64.144	0.3538	

```
## ---
```

```
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
#have a look to see if a quadratic term is indicated for the ilrs. Can use the poly(,2) function
```

```
mod.fat2 = lmer(scale(log(wht)) ~ poly(cbind(ilor1, ilr2, ilr3),2) + sex + age + ses + (1|school) , data=d1) #observations are nested in schools, school is a random intercept
```

```
anova(mod.fat, mod.fat2) #compare the two models
```

```
## refitting model(s) with ML (instead of REML)
```

```
## Data: d1
```

```
## Models:
```

```
## ..1: scale(log(wht)) ~ cbind(ilor1, ilr2, ilr3) + sex + age + ses + (1 | school)
```

```
## ..2: scale(log(wht)) ~ poly(cbind(ilor1, ilr2, ilr3), 2) + sex + age + ses + (1
```

```
| school)
##      npar    AIC    BIC logLik deviance Chisq Df Pr(>Chisq)
## ..1     9 460.64 488.81 -221.32  442.64
## ..2    15 463.57 510.52 -216.78  433.57 9.0741  6    0.1695
```

#the squared model is not indicated (at $p < 0.05$ or at $p < 0.1$), it's not significantly better than the non-squared model.

#model for fitness

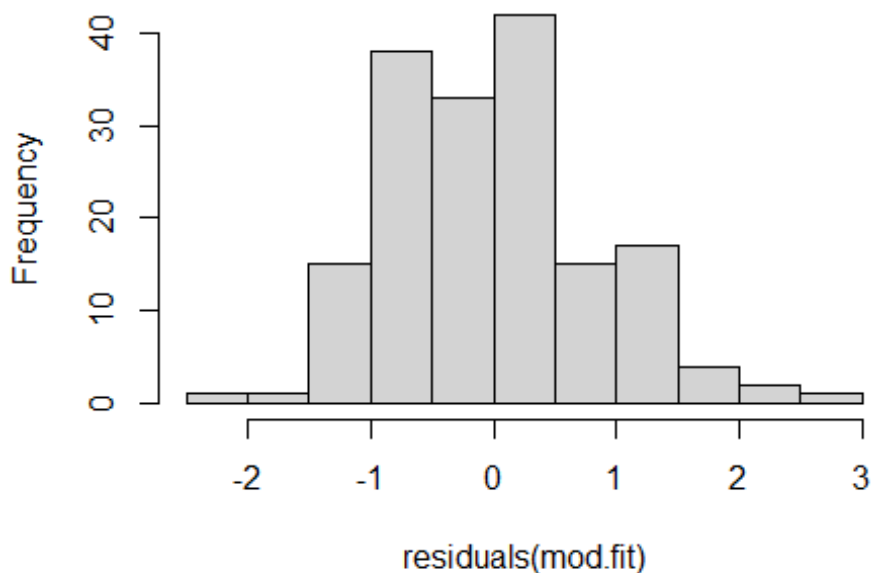
```
mod.fit = lmer(scale(shuttles) ~ cbind(ilr1, ilr2, ilr3) + sex + age + ses + (1|school), data=d1) #observations are nested in schools, school is a random intercept
summary(mod.fit)
```

```
## Linear mixed model fit by REML. t-tests use Satterthwaite's method [
## lmerModLmerTest]
## Formula: scale(shuttles) ~ cbind(ilr1, ilr2, ilr3) + sex + age + ses +
##      (1 | school)
##      Data: d1
##
## REML criterion at convergence: 434.2
##
## Scaled residuals:
##      Min       1Q   Median       3Q      Max
## -2.3941 -0.7097 -0.0335  0.5581  3.0242
##
## Random effects:
## Groups   Name                Variance Std.Dev.
## school  (Intercept)  0.05074  0.2253
## Residual                    0.71629  0.8463
## Number of obs: 169, groups: school, 7
##
## Fixed effects:
##
##              Estimate Std. Error    df t value Pr(>|t|)
## (Intercept)    0.280101  2.338022 161.574254  0.120  0.905
## cbind(ilr1, ilr2, ilr3)ilr1  0.148200  0.694273 158.996542  0.213  0.831
## cbind(ilr1, ilr2, ilr3)ilr2  0.001363  0.501639 161.967349  0.003  0.998
## cbind(ilr1, ilr2, ilr3)ilr3  1.032934  0.180723 161.523710  5.716 5.15e-08
## sex1              -0.021485  0.151023 160.905584 -0.142  0.887
## age                0.216627  0.224480 161.524911  0.965  0.336
## ses                0.034483  0.038035  62.159962  0.907  0.368
##
## (Intercept)
## cbind(ilr1, ilr2, ilr3)ilr1
## cbind(ilr1, ilr2, ilr3)ilr2
## cbind(ilr1, ilr2, ilr3)ilr3 ***
## sex1
## age
## ses
## ---
```

```
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Correlation of Fixed Effects:
##          (Intr) c(1,i2,i3)1 c(1,i2,i3)2 c(1,i2,i3)3 sex1  age
## c(1,i2,i3)1  0.113
## c(1,i2,i3)2 -0.065  0.458
## c(1,i2,i3)3  0.172  0.209    -0.428
## sex1         -0.073 -0.100    0.290    -0.480
## age         -0.979 -0.022    0.050    0.007    -0.033
## ses         0.008 -0.010    0.014    0.045    0.007 -0.041
```

```
hist(residuals(mod.fit))#Looks pretty good
```

Histogram of residuals(mod.fit)



```
car::Anova(mod.fit, test.statistic= "F") #composition is a significant predictor of zbmi.
```

```
## Analysis of Deviance Table (Type II Wald F tests with Kenward-Roger df)
##
## Response: scale(shuttles)
##
##          F Df  Df.res  Pr(>F)
## cbind(ilor1, ilor2, ilor3) 16.5577  3 161.296 1.99e-09 ***
## sex          0.0199  1 160.852  0.8879
## age          0.9009  1 161.502  0.3440
## ses          0.7332  1  60.331  0.3952
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```



```

#have a look to see if a quadratic term is indicated for the ilrs. Can use the poly(,2) function
mod.fit2 = lmer(scale(shuttles) ~ poly(cbind(ilr1, ilr2, ilr3),2) + sex + age + ses + (1|school) , data=d1) #observations are nested in schools, school is a random intercept
anova(mod.fit, mod.fit2) #compare the two models

## refitting model(s) with ML (instead of REML)

## Data: d1
## Models:
## ..1: scale(shuttles) ~ cbind(ilr1, ilr2, ilr3) + sex + age + ses + (1 | school)
## ..2: scale(shuttles) ~ poly(cbind(ilr1, ilr2, ilr3), 2) + sex + age + ses + (1 | school)
##      npar    AIC    BIC  logLik deviance  Chisq Df Pr(>Chisq)
## ..1     9 440.53 468.7 -211.26   422.53
## ..2    15 442.95 489.9 -206.47   412.95 9.5797 6      0.1435

#the squared model is not indicated (at p<0.05 or at p<0.1, it's not significantly better than the non-squared model.

#will use the two linear models, no squared terms.

#####
##Use the above models to make predictions across the empirical time-use footprint
#(the feasible range of time-use compositions observed within our sample)

#there are lots of ways of making hypothetical time-use compositions,
#one way is to create an evenly spaced "grid" of data points, for example
#to create every possible time-use composition in 10-minute increments.

#I'll start by making a grid that is larger than current min/max time spent in each variable
range(d1$sleep)
## [1] 393.0 624.7

range(d1$sb)
## [1] 299.6 694.8

range(d1$lpa)
## [1] 205.2 501.8

range(d1$mvpa)
## [1] 4.9 91.5

mygrid=rbind(
  expand.grid(
    sl=seq(min(380), max(640),10),

```

```

sb=seq(min(280), max(700), 10),
lpa=seq(min(200), max(510), 10),
mvpa=seq(min(0), max(100), 10))
head(mygrid)

##    sl  sb lpa mvpa
## 1 380 280 200    0
## 2 390 280 200    0
## 3 400 280 200    0
## 4 410 280 200    0
## 5 420 280 200    0
## 6 430 280 200    0

#only keep compositions in this grid that sum to 1440
mg=subset(mygrid, rowSums(mygrid)==1440)

#can truncate at +/- 3SD if you want to get rid of outliers in univariate activities
#get highest/lowest bounds of activity behaviours at +/- 3SD from the sample mean for predictive grid
mvpa1=pmax(pmin(d1$mvpa, mean(d1$ mvpa)+sd(d1$ mvpa)*3),0)
lpa1=pmax(pmin(d1$lpa, mean(d1$lpa)+sd(d1$lpa)*3),mean(d1$lpa)-sd(d1$lpa)*3)
sb1=pmax(pmin(d1$sb, mean(d1$sb)+sd(d1$sb)*3),mean(d1$sb)-sd(d1$sb)*3)
sl1=pmax(pmin(d1$sleep, mean(d1$ sleep)+sd(d1$ sleep)*3),mean(d1$ sleep)-sd(d1$ sleep)*3)

#can use these bounds to trim the predictive grid to values observed in sample
mg1=subset(mg, mg$sl<max(sl1) & mg$sl>min(sl1))
mg2=subset(mg1, mg1$sb<max(sb1) & mg1$sb>min(sb1))
mg3=subset(mg2, mg2$lpa<max(lpa1) & mg2$lpa>min(lpa1))
mg4=subset(mg3, mg3$mvpa<max(mvpa1) & mg3$mvpa>min(mvpa1))

#make the grid a compositional object
griddata=acomp(mg4)
min(griddata)

## [1] 0.006944444

head(as.data.frame(griddata)) #these are the first few rows of the grid

##          sl          sb          lpa          mvpa
## 39202 0.4305556 0.4166667 0.1458333 0.006944444
## 39228 0.4236111 0.4236111 0.1458333 0.006944444
## 39254 0.4166667 0.4305556 0.1458333 0.006944444
## 39280 0.4097222 0.4375000 0.1458333 0.006944444
## 39306 0.4027778 0.4444444 0.1458333 0.006944444
## 39332 0.3958333 0.4513889 0.1458333 0.006944444

nrow(as.data.frame(griddata))

## [1] 4340

```

```
#make ilrs from the griddata.
```

```
#check for zeros first
```

```
missingSummary(griddata)
```

```
##           missingType
## variable  NMV  BDL  MAR  MNAR  SZ  Err
##      sl   4340   0    0    0    0   0
##      sb   4340   0    0    0    0   0
##      lpa  4340   0    0    0    0   0
##      mvpa 4340   0    0    0    0   0
```

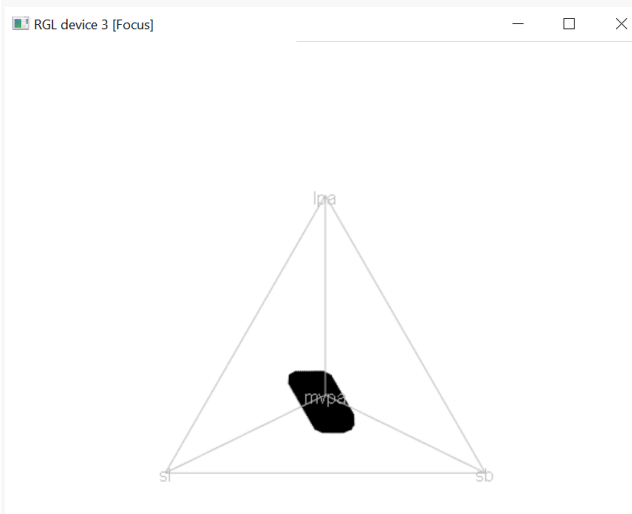
```
ilr.grid = ilr(griddata)
```

```
#can have a look at the predictive grid in 3-D
```

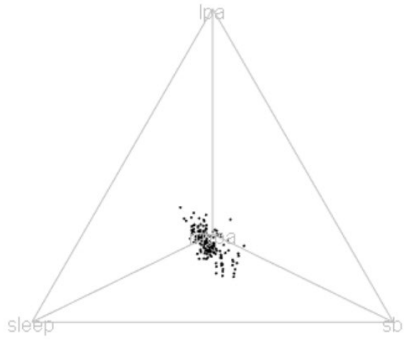
```
require(rgl)
```

```
## Loading required package: rgl
```

```
plot3D(griddata) #produces an interactive plot that you can zoom in, drag around to look at it.
```



```
plot3D(acompl(d1$comp)) #these are the actual observed data.
```



```

##      Sleep      Sedentary          LPA          MVPA
## "0.35733928" "0.43031618" "0.16143279" "0.05091175"
## attr(,"class")
## [1] "acomp"

#put activity variables in minutes/day
df.fatness$Sleep=df.fatness$Sleep*1440
df.fatness$Sedentary=df.fatness$Sedentary*1440
df.fatness$LPA=df.fatness$LPA*1440
df.fatness$MVPA=df.fatness$MVPA*1440

#get range of best 5%
range.best.sl=range(as.data.frame(best.fatness5[,1]))
range.best.sb=range(as.data.frame(best.fatness5[,2]))
range.best.lpa=range(as.data.frame(best.fatness5[,3]))
range.best.mvpa=range(as.data.frame(best.fatness5[,4]))

#make nice results summary to look like mean (range low:high)
sum.sl=c(b[1], range.best.sl, b[2], range.best.sb, b[3], range.best.lpa, b[4], ran
ge.best.mvpa)
fat.sum=round(sum.sl*1440)
fat.sum #in minutes

##      Sleep      low      high      Sedentary      low      high      LPA      low
##      515      440      620      620      510      690      232      210
##      high      MVPA      low      high
##      290      73      50      80

round(fat.sum/60, 1) #in hours

##      Sleep      low      high      Sedentary      low      high      LPA      low
##      8.6      7.3      10.3      10.3      8.5      11.5      3.9      3.5
##      high      MVPA      low      high
##      4.8      1.2      0.8      1.3

#####
####
#####Predict fitness values for all the hypothetical compositions in our gridda
ta
predboys=predict(mod.fit, newdata=data.frame(ilr1 = ilr.grid[,1],
                                             ilr2 = ilr.grid[,2],
                                             ilr3 = ilr.grid[,3],
                                             ses=rep(mean(d1$ses),nrow(griddata)),
                                             age=rep(mean(d1$age),nrow(griddata)),
                                             sex=rep("0", nrow(griddata)) ), re.fo

rm = NA)
#then for girls
predgirls=predict(mod.fit, newdata=data.frame(ilr1 = ilr.grid[,1],
                                             ilr2 = ilr.grid[,2],
                                             ilr3 = ilr.grid[,3],
                                             ses=rep(mean(d1$ses),nrow(griddata))

```

```

,
                                age=rep(mean(d1$age),nrow(griddata))
,
                                sex=rep("1", nrow(griddata)) ), re.f
orm = NA)

#and then find the average of the two (population has about 50:50 boys:girls)
pred=rowMeans(cbind(predboys, predgirls))

#this makes a dataframe with the predictive grid and the outcome score
df.fit=data.frame(Sleep=griddata[,1],Sedentary=griddata[,2], LPA=griddata[,3], MVP
A=griddata[,4], Predicted_Outcome_Zscore=pred)

#extract the compositiona associated with the Lowest 5% of predicted bmi.
best.fitness5=df.fit%>% top_frac(.05, Predicted_Outcome_Zscore)

#this is the centre of the lowest 5%
(b=mean(acomp(best.fitness5[,1:4])))

##      Sleep      Sedentary      LPA      MVPA
## "0.3559981" "0.4015253" "0.1866027" "0.0558739"
## attr(,"class")
## [1] "acomp"

#put activity variables in minutes/day
df.fit$Sleep=df.fit$Sleep*1440
df.fit$Sedentary=df.fit$Sedentary*1440
df.fit$LPA=df.fit$LPA*1440
df.fit$MVPA=df.fit$MVPA*1440

#get range of best 5%
range.best.sl=range(as.data.frame(best.fitness5[,1]))
range.best.sb=range(as.data.frame(best.fitness5[,2]))
range.best.lpa=range(as.data.frame(best.fitness5[,3]))
range.best.mvpa=range(as.data.frame(best.fitness5[,4]))

#make nice results summary to look like mean (range low:high)
sum.sl=c(b[1], range.best.sl, b[2], range.best.sb, b[3], range.best.lpa, b[4], ran
ge.best.mvpa)
fit.sum=round(sum.sl*1440)
fit.sum #in minutes

##      Sleep      Sedentary      LPA      MVPA
##      513      440      620      578      480      690      269      210
##      400      80      80      80

round(fit.sum/60, 1) #in hours

```

```

##      Sleep              Sedentary              LPA              MVPA
##      8.6              7.3              10.3              9.6              8.0              11.5              4.5              3.5
##      MVPA
##      6.7              1.3              1.3              1.3

#####
####

#how much overlap with fitness and fatness best zones?
ff5=rbind(best.fitness5[,1:4], best.fatness5[,1:4])
#which predictive compositions are overlapping (not unique)
head(ff5)

##      Sleep Sedentary      LPA      MVPA
## 299077 0.4305556 0.3680556 0.1458333 0.05555556
## 299103 0.4236111 0.3750000 0.1458333 0.05555556
## 299129 0.4166667 0.3819444 0.1458333 0.05555556
## 299155 0.4097222 0.3888889 0.1458333 0.05555556
## 299181 0.4027778 0.3958333 0.1458333 0.05555556
## 299207 0.3958333 0.4027778 0.1458333 0.05555556

table(duplicated(ff5))

##
## FALSE  TRUE
##   324   110

#110 duplicated composition - these compositions are in both the fitness and fatness best zones

best.fitness5=cbind.data.frame(best.fitness5, measure=rep("Fitness",nrow(best.fitness5)))
best.fitness5=best.fitness5[,1:6]
best.fatness5=cbind.data.frame(best.fatness5, measure=rep("Fatness",nrow(best.fatness5)))
ff5=rbind(best.fitness5, best.fatness5)

#these are the overlapping compositions
dups5=ff5%>% group_by(Sleep, Sedentary, LPA,MVPA) %>% filter(n(>1)) %>% ungroup()

#centre of best fit and best fat means
mm=mean(acomp(dups5[,1:4]))
round(clo(mm, total=24),1)

##      Sleep Sedentary      LPA      MVPA
##      8.6      10.1      4.0      1.3

round(clo(mm, total=1440))

##      Sleep Sedentary      LPA      MVPA
##      516      606      238      80

```

```

range(dups5[,1]*24)#range for sleep bests
## [1] 7.333333 10.333333
range(dups5[,2]*24) #range for SED bests
## [1] 8.5 11.5
range(dups5[,3]*24) #range for LPA bests
## [1] 3.500000 4.833333
range(dups5[,4]*24) #range for MVPA bests
## [1] 1.333333 1.333333

#####
#####
#####
#####

#Some plots to explore
#centre of best fit and best fat means
meandups=mean(acom(dups5[,1:4]))
mm=mean(acom(rbind(meandups)))
round(clo(mm, total=24),1)

##      Sleep Sedentary      LPA      MVPA
##      8.6      10.1      4.0      1.3

round(clo(mm, total=1440))

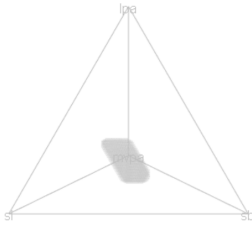
##      Sleep Sedentary      LPA      MVPA
##      516      606      238      80

mm1=rbind(mm,mm)
meanfifa=cbind.data.frame(mm1, measure=rep("mean",2))

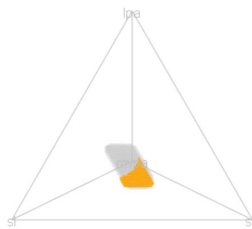
# plot the whole grid of hypothetical compositions
plot3D(acom(griddata[,1:4]), cex=0.5,color="grey", alpha=0.6)

```


RGL device 3 [Focus]



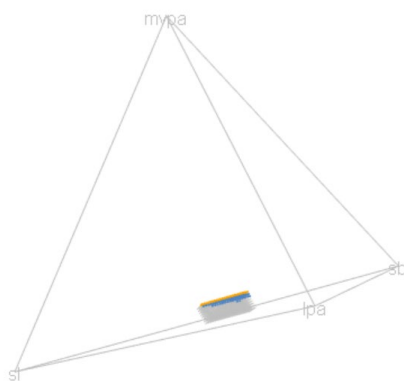
RGL device 3 [Focus]



superimpose the best 5% fitness (blue)

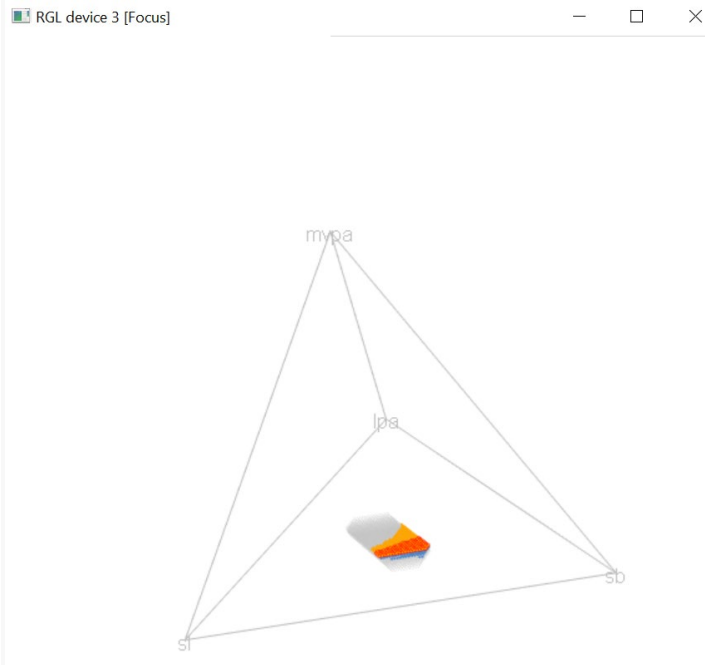
```
plot3D(acompl(best.fitness5[,1:4]), cex=2,color="#4E84C4", alpha=1, add=TRUE)
```

RGL device 3 [Focus]



#show where best 5% fitness and fitness overlap (red)

```
plot3D(acompl(dups5[,1:4]), cex=2,color="red", alpha=1, add=TRUE)
```



#plot centre of best fitness/fatness overlapping area

```
plot3D(acom(meanfifa[,1:4]), cex=5,color="black", alpha=1, add=TRUE)
```

