PPD ISDP

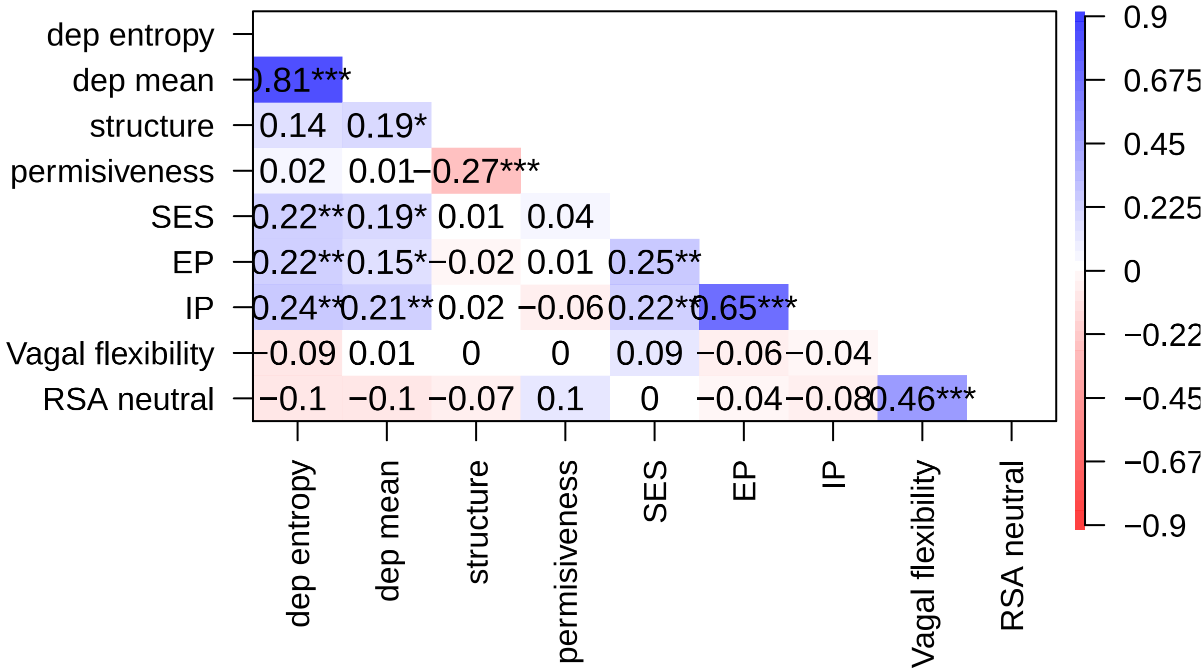
Elisa Ugarte

7/6/2021

Vars of interest

```
corr_par <-
  final %>%
   dplyr::select(
    `dep entropy`= scl1depent,
    `dep mean` = scl1dep,
    `structure` = f1custructure,
    `permisiveness` = f1cupermit,
    `SES` = f1RRses,
    `EP` = f1mextprb,
    `IP`= f1mintprb,
    `Vagal flexibility` = s,
    `RSA neutral` = i)

corr_par <- as.data.frame(corr_par)
#corr_par <- corr_par[,-1]

corPlot(corr_par, upper = F, diag = F, zlim = c(-0.9, 0.9), stars = T, cex = 1.1, pval=T,
       cuts=c(.001,.05), n.legend = 8, scale = F, ylas = 1, xlas = 2, main = "Correlations among variabl
```

Correlations among variables of interest



Descriptives

```
##                vars   n    mean     sd  median trimmed   mad     min     max
## fid               1 180 6227.83 315.27 6105.50 6166.70 80.80 6001.00 6947.00
## f1csex            2 180    1.47   0.50    1.00    1.47  0.00    1.00    2.00
## f1mintprb         3 173   52.61  10.49   52.00   52.81 11.86   29.00   75.00
## f1mextprb         4 173   53.45  11.61   54.00   53.44 14.83   32.00   80.00
## f1cage            5 180    5.37   1.10    4.79    5.36  1.15    3.18    6.92
## i                 6 154    0.00   1.15    0.16    0.05  1.29   -3.71    2.76
## s                 7 154    0.00   0.25   -0.01    0.01  0.22   -0.97    0.63
## t1baseline        8 154    6.84   1.11    6.87    6.87  1.05    3.67    9.63
## f1RRses           9 180   -0.01   0.87   -0.05    0.02  0.70   -2.42    1.81
## f1custructure    10 177    2.78   0.62    2.75    2.76  0.62    1.25    4.67
## f1cupermit       11 177    1.70   0.80    1.50    1.56  0.74    1.00    4.50
## f1cuwarm         12 177    1.85   0.59    1.75    1.81  0.62    1.00    3.60
## scl1depent       13 178    0.41   0.26    0.41    0.40  0.36    0.00    0.99
## scl1anxent       14 178   30.77  25.33   27.25   28.89 33.46    0.00   91.39
## scl1hosent       15 178   37.88  22.95   39.55   37.91 21.28    0.00   82.62
## scl1dep          16 178    0.53   0.61    0.31    0.42  0.34    0.00    3.38
## scl1anx          17 178    0.36   0.49    0.20    0.26  0.30    0.00    2.80
## scl1hos          18 178    0.50   0.51    0.33    0.42  0.25    0.00    3.50
## physent          19 176    0.48   0.15    0.47    0.50  0.14    0.00    0.77
##                range  skew kurtosis    se
## fid           946.00  1.69     1.00 23.50
## f1csex          1.00  0.11    -2.00  0.04
## f1mintprb      46.00 -0.10    -0.68  0.80
## f1mextprb      48.00  0.02    -0.95  0.88
## f1cage          3.74  0.13    -1.71  0.08
## i               6.47 -0.38    -0.08  0.09
```

```
## s                 1.60 -0.65     2.03  0.02
## t1baseline        5.96 -0.27     0.24  0.09
## f1RRses           4.23 -0.38     0.11  0.06
## f1custructure     3.42  0.23     0.09  0.05
## f1cupermit        3.50  1.51     1.87  0.06
## f1cuwarm          2.60  0.57    -0.23  0.04
## scl1depent        0.99  0.05    -0.95  0.02
## scl1anxent       91.39  0.38    -0.85  1.90
## scl1hosent       82.62 -0.11    -0.71  1.72
## scl1dep           3.38  2.02     4.93  0.05
## scl1anx           2.80  2.53     7.71  0.04
## scl1hos           3.50  2.06     7.10  0.04
## physent           0.77 -0.79     0.15  0.01
```

Convergent validity

```r
library(BayesFactor)
library(bayestestR)

#### CONVERGENT ####
#Neuroticsm - expecting a positive correlation

main_neuro <- correlationBF(final$scl1dep, final$neuro)
```

```
## Ignored 2 rows containing missing observations.
```

```r
describe_posterior(main_neuro)
```

```
## Summary of Posterior Distribution
##
## Parameter | Median |       95% CI |   pd |          ROPE | % in ROPE |     BF |         Prior
## ---------------------------------------------------------------------------------------------
## rho       |   0.63 | [0.53, 0.71] | 100% | [-0.05, 0.05] |        0% | > 1000 | Beta (3 +- 3)
```

```r
bayesfactor(main_neuro)
```

```
## Bayes Factors for Model Comparison
##
##     Model            BF
## [2] (rho != 0) 1.53e+19
##
## * Against Denominator: [1] (rho = 0)
## *   Bayes Factor Type: JZS (BayesFactor)
```

```r
ent_neuro <- correlationBF(final$scl1depent, final$neuro)
```

```
## Ignored 2 rows containing missing observations.
```

```r
describe_posterior(ent_neuro)
```

```
## Summary of Posterior Distribution
##
## Parameter | Median |       95% CI |   pd |          ROPE | % in ROPE |     BF |         Prior
## ---------------------------------------------------------------------------------------------
## rho       |   0.62 | [0.53, 0.71] | 100% | [-0.05, 0.05] |        0% | > 1000 | Beta (3 +- 3)
```

```r
bayesfactor(ent_neuro)
```

```
## Bayes Factors for Model Comparison
##
##     Model             BF
## [2] (rho != 0) 1.29e+18
##
## * Against Denominator: [1] (rho = 0)
## *   Bayes Factor Type: JZS (BayesFactor)
```

```r
# Conscientiousness - expecting a neg correlation
main_conc <- correlationBF(final$scl1dep, final$conc)
```

```
## Ignored 2 rows containing missing observations.
```

```r
describe_posterior(main_conc)
```

```
## Summary of Posterior Distribution
##
## Parameter | Median |          95% CI |     pd |          ROPE | % in ROPE |   BF |         Prior
## ---------------------------------------------------------------------------------------------------
## rho       |  -0.17 | [-0.31, -0.03] | 99.12% | [-0.05, 0.05] |     2.16% | 2.87 | Beta (3 +- 3)
```

```r
bayesfactor(main_conc)
```

```
## Bayes Factors for Model Comparison
##
##     Model         BF
## [2] (rho != 0) 2.87
##
## * Against Denominator: [1] (rho = 0)
## *   Bayes Factor Type: JZS (BayesFactor)
```

```r
ent_conc <- correlationBF(final$scl1depent, final$conc)
```

```
## Ignored 2 rows containing missing observations.
```

```r
describe_posterior(ent_conc)
```

```
## Summary of Posterior Distribution
##
## Parameter | Median |          95% CI |     pd |          ROPE | % in ROPE |    BF |         Prior
## ----------------------------------------------------------------------------------------------------
## rho       |  -0.24 | [-0.37, -0.09] | 99.95% | [-0.05, 0.05] |        0% | 30.85 | Beta (3 +- 3)
```

```r
bayesfactor(ent_conc)
```

```
## Bayes Factors for Model Comparison
##
##     Model          BF
## [2] (rho != 0) 30.85
##
## * Against Denominator: [1] (rho = 0)
## *   Bayes Factor Type: JZS (BayesFactor)
```

```r
#### DIVERGENT ####

# Entropy of a health and activities questionnaire
main_phys <- correlationBF(final$scl1dep, final$physent)
```

```
## Ignored 5 rows containing missing observations.
```

```
describe_posterior(main_phys)
```

```
## Summary of Posterior Distribution
##
## Parameter | Median |         95% CI |     pd |          ROPE | % in ROPE |    BF |          Prior
## ----------------------------------------------------------------------------------------------------
## rho       |  -0.06 | [-0.21, 0.09] | 79.03% | [-0.05, 0.05] |    39.41% | 0.243 | Beta (3 +- 3)
```

```
bayesfactor(main_phys)
```

```
## Bayes Factors for Model Comparison
##
##     Model          BF
## [2] (rho != 0) 0.243
##
## * Against Denominator: [1] (rho = 0)
## *   Bayes Factor Type: JZS (BayesFactor)
```

```
ent_phys <- correlationBF(final$scl1depent, final$physent)
```

```
## Ignored 5 rows containing missing observations.
```

```
describe_posterior(ent_phys)
```

```
## Summary of Posterior Distribution
##
## Parameter | Median |         95% CI |     pd |          ROPE | % in ROPE |    BF |          Prior
## ----------------------------------------------------------------------------------------------------
## rho       |   0.04 | [-0.11, 0.18] | 69.92% | [-0.05, 0.05] |    45.83% | 0.197 | Beta (3 +- 3)
```

```
bayesfactor(ent_phys)
```

```
## Bayes Factors for Model Comparison
##
##     Model          BF
## [2] (rho != 0) 0.197
##
## * Against Denominator: [1] (rho = 0)
## *   Bayes Factor Type: JZS (BayesFactor)
```

```
### PREDICTIVE ###

#Internalizing problems
main_int <- correlationBF(final$scl1dep, final$f1mintprb)
```

```
## Ignored 8 rows containing missing observations.
```

```
describe_posterior(main_int)
```

```
## Summary of Posterior Distribution
##
## Parameter | Median |        95% CI |     pd |          ROPE | % in ROPE |   BF |          Prior
## --------------------------------------------------------------------------------------------------
## rho       |   0.20 | [0.06, 0.34] | 99.50% | [-0.05, 0.05] |        0% | 6.23 | Beta (3 +- 3)
```

```r
bayesfactor(main_int)
```

```
## Bayes Factors for Model Comparison
##
##     Model        BF
## [2] (rho != 0) 6.23
##
## * Against Denominator: [1] (rho = 0)
## *   Bayes Factor Type: JZS (BayesFactor)
```

```r
ent_int <- correlationBF(final$scl1depent, final$f1mintprb)
```

```
## Ignored 8 rows containing missing observations.
```

```r
describe_posterior(ent_int)
```

```
## Summary of Posterior Distribution
##
## Parameter | Median |       95% CI |     pd |          ROPE | % in ROPE |    BF |         Prior
## ---------------------------------------------------------------------------------------------
## rho       |   0.23 | [0.10, 0.37] | 99.92% | [-0.05, 0.05] |        0% | 26.14 | Beta (3 +- 3)
```

```r
bayesfactor(ent_int)
```

```
## Bayes Factors for Model Comparison
##
##     Model         BF
## [2] (rho != 0) 26.14
##
## * Against Denominator: [1] (rho = 0)
## *   Bayes Factor Type: JZS (BayesFactor)
```

```r
#Inhibitory control
main_inc <- correlationBF(final$scl1dep, final$f1ctinc)
```

```
## Ignored 5 rows containing missing observations.
```

```r
describe_posterior(main_inc)
```

```
## Summary of Posterior Distribution
##
## Parameter | Median |         95% CI |     pd |          ROPE | % in ROPE |    BF |         Prior
## -----------------------------------------------------------------------------------------------
## rho       |  -0.21 | [-0.35, -0.07] | 99.72% | [-0.05, 0.05] |        0% | 11.15 | Beta (3 +- 3)
```

```r
bayesfactor(main_inc)
```

```
## Bayes Factors for Model Comparison
##
##     Model         BF
## [2] (rho != 0) 11.15
##
## * Against Denominator: [1] (rho = 0)
## *   Bayes Factor Type: JZS (BayesFactor)
```

```r
ent_inc <- correlationBF(final$scl1depent, final$f1ctinc)
```

```
## Ignored 5 rows containing missing observations.
```

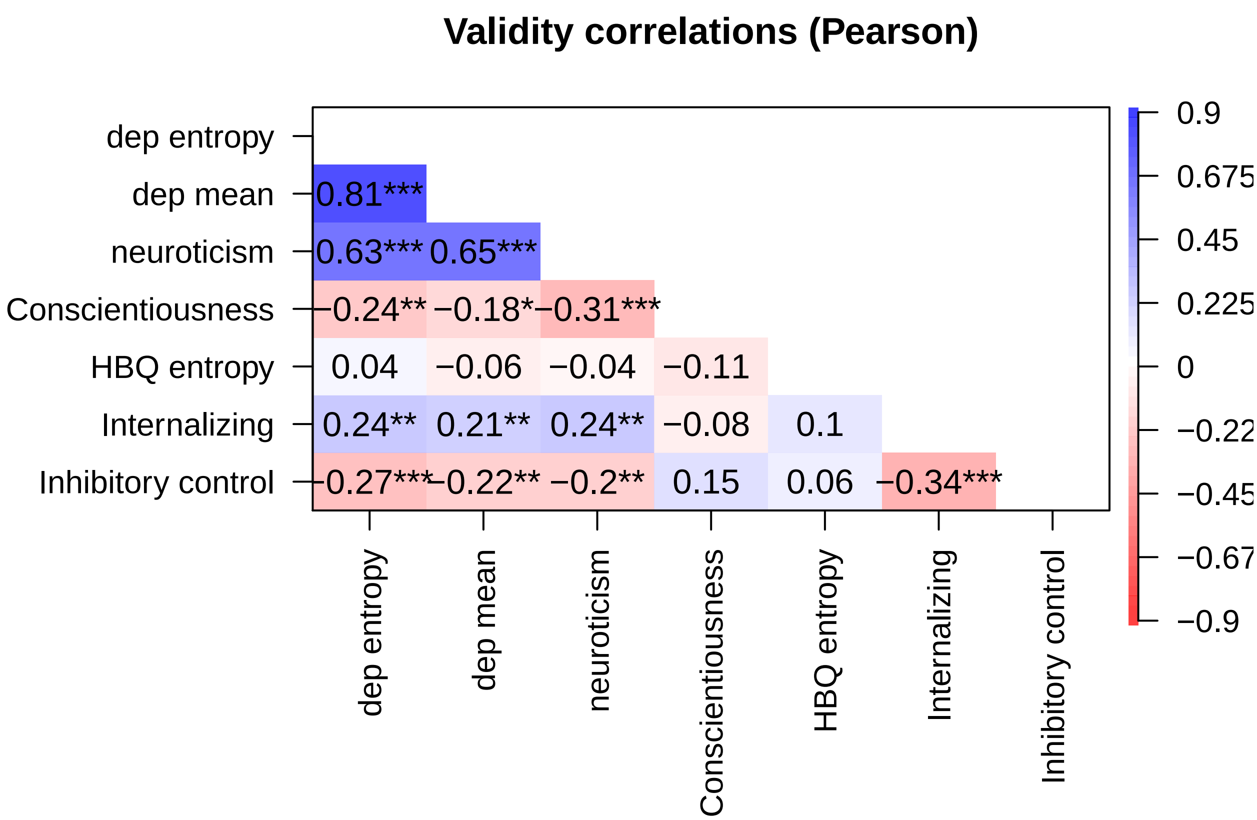
```r
describe_posterior(ent_inc)
```

```
## Summary of Posterior Distribution
##
## Parameter | Median |         95% CI |   pd |          ROPE | % in ROPE |     BF |         Prior
## ----------------------------------------------------------------------------------------------
## rho       |  -0.27 | [-0.40, -0.13] | 100% | [-0.05, 0.05] |        0% | 118.76 | Beta (3 +- 3)
```

```r
bayesfactor(ent_inc)
```

```
## Bayes Factors for Model Comparison
##
##     Model          BF
## [2] (rho != 0) 118.76
##
## * Against Denominator: [1] (rho = 0)
## *   Bayes Factor Type: JZS (BayesFactor)
```

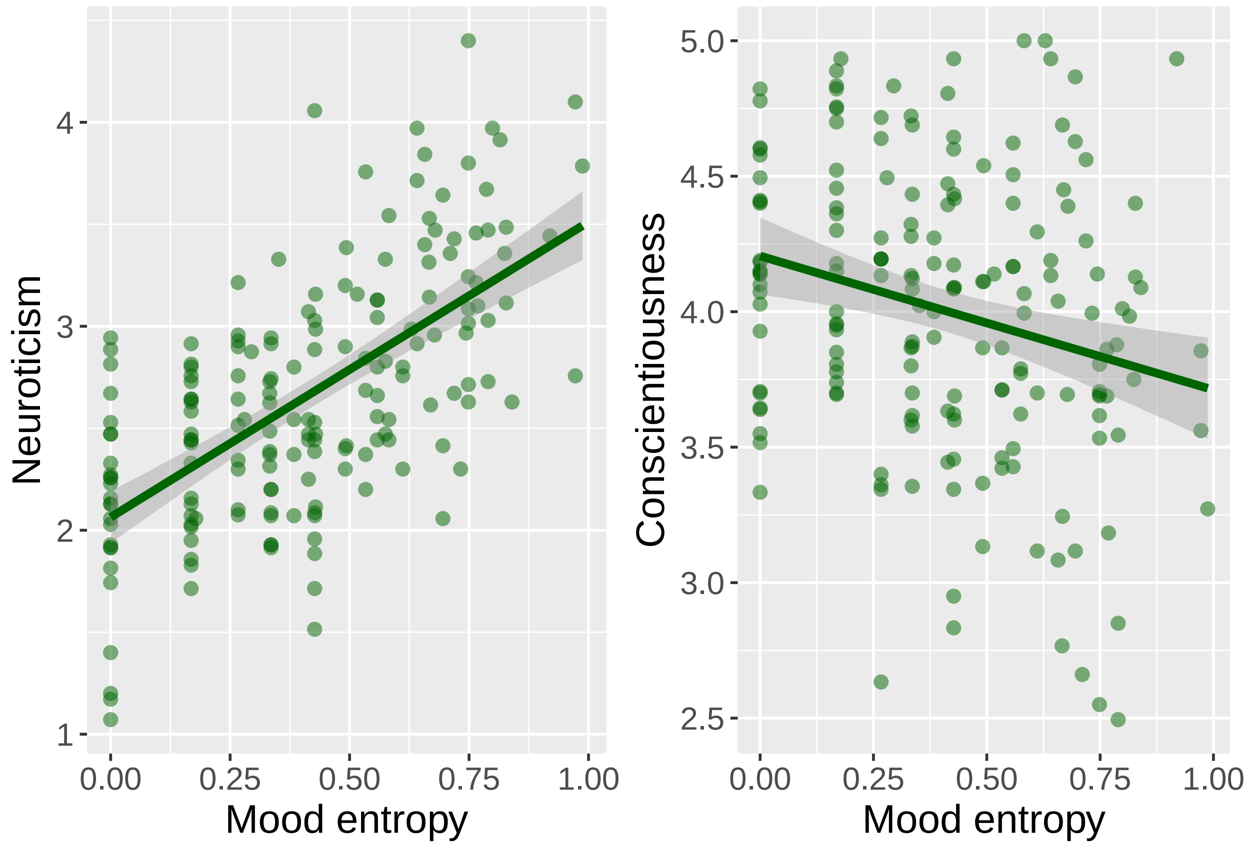
```r
#Validity correlation and plots

valid <-
  final %>%
   dplyr::select(
    `dep entropy`= scl1depent,
    `dep mean` = scl1dep,
    `neuroticism` = neuro,
    `Conscientiousness` = conc,
    `HBQ entropy` = physent,
    `Internalizing` = f1mintprb,
    `Inhibitory control` = f1ctinc)

corPlot(valid, upper = F, diag = F, zlim = c(-0.9, 0.9), stars = T, cex = 1.1, pval=T,
       cuts=c(.001,.05), n.legend = 8, scale = F, ylas = 1, xlas = 2, main = "Validity correlations (Pe
```




```
library(ggplot2)

p1 <- final %>%
  ggplot(aes(scl1depent, neuro)) +
  geom_point(size = 2, alpha = 1/2, colour = "darkgreen") +
  geom_smooth(method = "lm", size = 1.5, color = "darkgreen") + # +
  #ylim(-0.5,0.5)  +
  #xlim(-0.5,0.5)  +
  labs(
    x = "Mood entropy",
    y = "Neuroticism"
  ) + theme(text = element_text(size=15))

p2 <- final %>%
  ggplot(aes(scl1depent, conc)) +
  geom_point(size = 2, alpha = 1/2, colour = "darkgreen") +
  geom_smooth(method = "lm", size =1.5, color = "darkgreen") + # +
  #ylim(-0.5,0.5)  +
  #xlim(-0.5,0.5)  +
  labs(
    x = "Mood entropy",
    y = "Conscientiousness"
  ) + theme(text = element_text(size=15))

p3 <- final %>%
  ggplot(aes(scl1depent, physent)) +
  geom_point(size = 2, alpha = 1/2, colour = "red") +
  geom_smooth(method = "lm", size = 1.5, color = "darkred") + # +
```

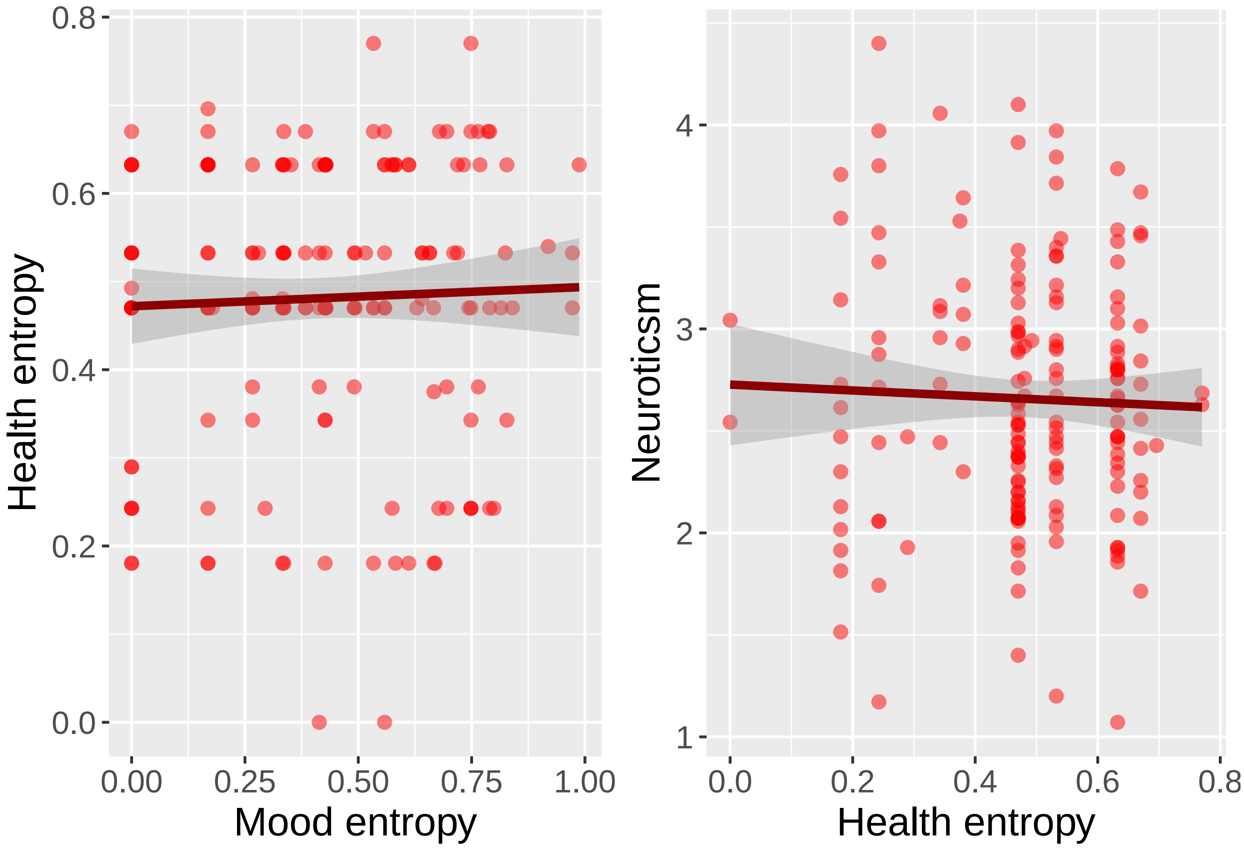
```r
  #ylim(-0.5,0.5)  +
  #xlim(-0.5,0.5)  +
  labs(
    x = "Mood entropy",
    y = "Health entropy"
  ) + theme(text = element_text(size=15))

p4 <- final %>%
  ggplot(aes(physent, neuro)) +
  geom_point(size = 2, alpha = 1/2, colour = "red") +
  geom_smooth(method = "lm", size = 1.5, color = "darkred") + # +
  #ylim(-0.5,0.5)  +
  #xlim(-0.5,0.5)  +
  labs(
    x = "Health entropy",
    y = "Neuroticsm"
  ) + theme(text = element_text(size=15))

p5 <- final %>%
  ggplot(aes(scl1depent, f1mintprb)) +
  geom_point(size = 2, alpha = 1/2, colour = "blue") +
  geom_smooth(method = "lm", size = 1.5, color = "darkblue") + # +
  #ylim(-0.5,0.5)  +
  #xlim(-0.5,0.5)  +
  labs(
    x = "Mood entropy",
    y = "Child internalizing problems"
  ) + theme(text = element_text(size=15))

p6 <- final %>%
  ggplot(aes(scl1depent, f1ctinc)) +
  geom_point(size = 2, alpha = 1/2, colour = "blue") +
  geom_smooth(method = "lm", size = 1.5, color = "darkblue") + # +
  #ylim(-0.5,0.5)  +
  #xlim(-0.5,0.5)  +
  labs(
    x = "Mood entropy",
    y = "Child Inhibitory control"
  ) + theme(text = element_text(size=15))

library(gridExtra)
p <- grid.arrange(p1, p2, nrow=1, ncol=2)
```

```
## `geom_smooth()` using formula 'y ~ x'

## `geom_smooth()` using formula 'y ~ x'
```

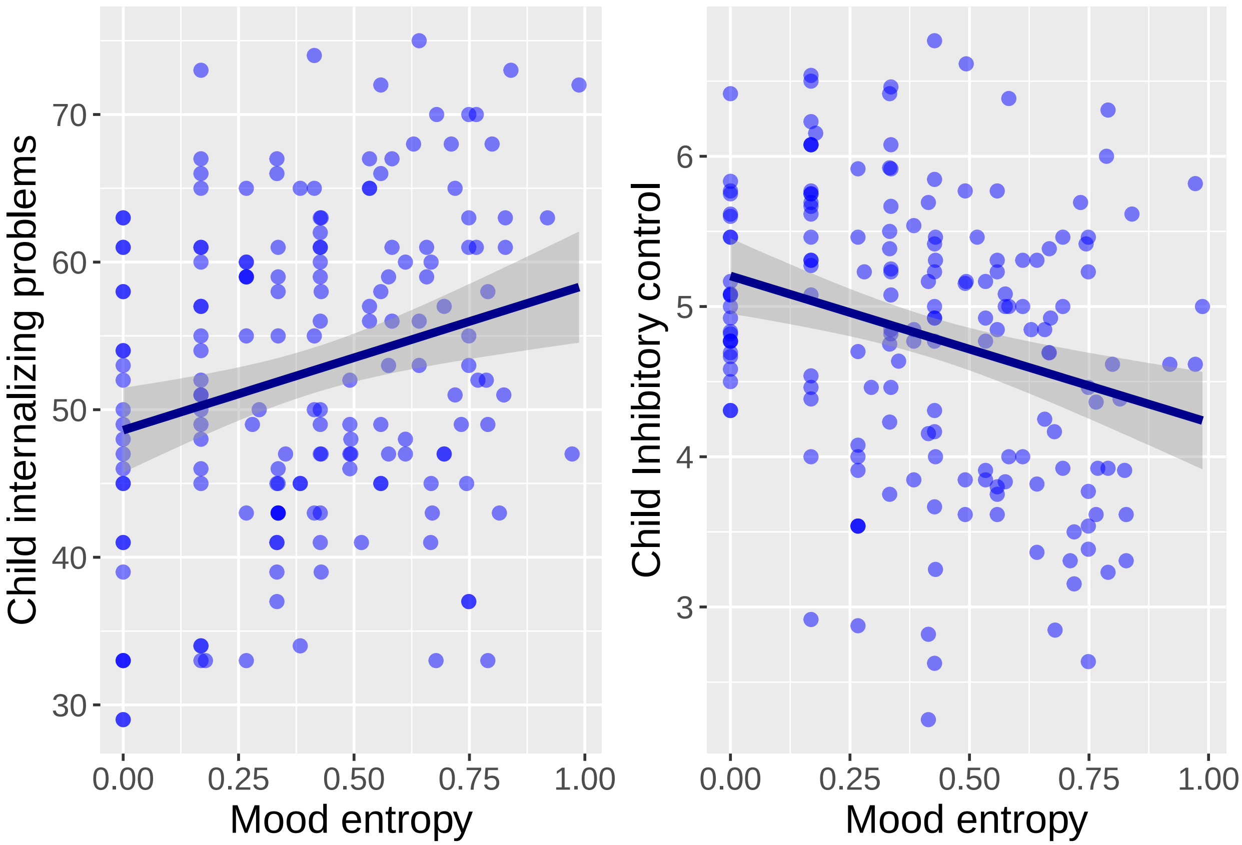



```
a <- grid.arrange(p3,p4, nrow = 1, ncol=2)
```

```
## `geom_smooth()` using formula 'y ~ x'
## `geom_smooth()` using formula 'y ~ x'
```




```
s <- grid.arrange(p5,p6, nrow = 1, ncol=2)

## `geom_smooth()` using formula 'y ~ x'
## `geom_smooth()` using formula 'y ~ x'
```



RQ 1: Controlling for child baseline, intercept, sex, age, observed parenting, and mood levels

is maternal mood entropy related to children's Physio (Reactivity (only at T1))

##Controls

```
sm <- dat[c(1:12,15)]
sm <-
  sm %>%
  mutate(f1csex= f1csex-1) %>%
  mutate(f1csex = as.factor(f1csex))
contrasts(sm$f1csex) <- c(-.5, .5)

d0_lm <-
na.omit(sm)


mod_0 <- lm(scale(s)~
              1, data=d0_lm)

mod_i<-
  lm(
    scale(s) ~
      scale(i),
    data = d0_lm
```

```r
  )
anova(mod_0, mod_i)
```

```
## Analysis of Variance Table
##
## Model 1: scale(s) ~ 1
## Model 2: scale(s) ~ scale(i)
##   Res.Df    RSS Df Sum of Sq      F    Pr(>F)
## 1    140 140.00
## 2    139 111.22  1    28.776 35.963 1.647e-08 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```r
model_parameters(mod_i)
```

```
## Parameter   | Coefficient |   SE |        95% CI |    t(139) |      p
## ---------------------------------------------------------------------
## (Intercept) |   -2.50e-18 | 0.08 | [-0.15, 0.15] | -3.31e-17 | > .999
## i           |        0.45 | 0.08 | [ 0.30, 0.60] |      6.00 | < .001
```

```r
mod_base<-
  lm(
    scale(s) ~
      scale(i) +
      scale(t1baseline),
    data = d0_lm
  )
anova(mod_i, mod_base)
```

```
## Analysis of Variance Table
##
## Model 1: scale(s) ~ scale(i)
## Model 2: scale(s) ~ scale(i) + scale(t1baseline)
##   Res.Df    RSS Df Sum of Sq      F Pr(>F)
## 1    139 111.22
## 2    138 111.22  1 0.0039492 0.0049 0.9443
```

```r
model_parameters(mod_base)
```

```
## Parameter   | Coefficient |   SE |        95% CI |    t(138) |      p
## ---------------------------------------------------------------------
## (Intercept) |   -4.90e-18 | 0.08 | [-0.15, 0.15] | -6.48e-17 | > .999
## i           |        0.46 | 0.13 | [ 0.21, 0.71] |      3.58 | < .001
## t1baseline  |   -9.00e-03 | 0.13 | [-0.26, 0.25] |     -0.07 | 0.944
```

```r
#Not necessary to have baseline & i. Model with i fits better? But people are more familiar with baseli

mod_age <-
  lm(
    scale(s) ~
      scale(i) +
      scale(f1cage),
    data = d0_lm
  )
anova(mod_i, mod_age)
```

```
## Analysis of Variance Table
```

```
##
## Model 1: scale(s) ~ scale(i)
## Model 2: scale(s) ~ scale(i) + scale(f1cage)
##   Res.Df    RSS Df Sum of Sq      F   Pr(>F)
## 1    139 111.22
## 2    138 105.28  1    5.9437 7.7909 0.005996 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
model_parameters(mod_age)
```

```
## Parameter   | Coefficient |   SE |        95% CI |   t(138) |      p
## --------------------------------------------------------------------
## (Intercept) |    1.66e-17 | 0.07 | [-0.15, 0.15] | 2.26e-16 | > .999
## i           |        0.46 | 0.07 | [ 0.31, 0.60] |     6.18 | < .001
## f1cage      |        0.21 | 0.07 | [ 0.06, 0.35] |     2.79 | 0.006
```

```
#Age is a significant covariate

mod_sex <-
  lm(
    scale(s) ~
      scale(i) +
      scale(f1cage) +
      f1csex,
    data = d0_lm
  )
anova(mod_age, mod_sex)
```

```
## Analysis of Variance Table
##
## Model 1: scale(s) ~ scale(i) + scale(f1cage)
## Model 2: scale(s) ~ scale(i) + scale(f1cage) + f1csex
##   Res.Df    RSS Df Sum of Sq      F  Pr(>F)
## 1    138 105.28
## 2    137 101.39  1    3.8884 5.2541 0.02342 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
model_parameters(mod_sex)
```

```
## Parameter   | Coefficient |   SE |        95% CI | t(137) |      p
## ------------------------------------------------------------------
## (Intercept) |        0.01 | 0.07 | [-0.13, 0.16] |   0.18 | 0.858
## i           |        0.47 | 0.07 | [ 0.32, 0.61] |   6.42 | < .001
## f1cage      |        0.22 | 0.07 | [ 0.07, 0.36] |   2.99 | 0.003
## f1csex [1]  |        0.33 | 0.15 | [ 0.05, 0.62] |   2.29 | 0.023
```

```
#Sex is also relevant

mod_ep <-
  lm(
    scale(s) ~
      scale(i) +
      scale(f1cage) +
      f1csex +
      scale(f1mextprb),
```

```r
    data = d0_lm
  )
anova(mod_sex, mod_ep)
```

```
## Analysis of Variance Table
##
## Model 1: scale(s) ~ scale(i) + scale(f1cage) + f1csex
## Model 2: scale(s) ~ scale(i) + scale(f1cage) + f1csex + scale(f1mextprb)
##   Res.Df    RSS Df Sum of Sq      F Pr(>F)
## 1    137 101.39
## 2    136 100.51  1   0.88543 1.1981 0.2756
```

```r
model_parameters(mod_ep)
```

```
## Parameter   | Coefficient |   SE |         95% CI | t(136) |      p
## -------------------------------------------------------------------
## (Intercept) |        0.01 | 0.07 | [-0.13, 0.16] |   0.17 | 0.864
## i           |        0.47 | 0.07 | [ 0.32, 0.61] |   6.39 | < .001
## f1cage      |        0.23 | 0.07 | [ 0.09, 0.38] |   3.13 | 0.002
## f1csex [1]  |        0.32 | 0.15 | [ 0.03, 0.61] |   2.19 | 0.031
## f1mextprb   |       -0.08 | 0.07 | [-0.23, 0.07] |  -1.09 | 0.276
```

```r
mod_ip <-
  lm(
    scale(s) ~
      scale(i) +
      scale(f1cage) +
      f1csex +
      scale(f1mintprb),
    data = d0_lm
  )
anova(mod_sex, mod_ip)
```

```
## Analysis of Variance Table
##
## Model 1: scale(s) ~ scale(i) + scale(f1cage) + f1csex
## Model 2: scale(s) ~ scale(i) + scale(f1cage) + f1csex + scale(f1mintprb)
##   Res.Df    RSS Df Sum of Sq      F Pr(>F)
## 1    137 101.39
## 2    136 101.34  1  0.051855 0.0696 0.7923
```

```r
model_parameters(mod_ip)
```

```
## Parameter   | Coefficient |   SE |         95% CI | t(136) |      p
## -------------------------------------------------------------------
## (Intercept) |        0.01 | 0.07 | [-0.13, 0.16] |   0.17 | 0.862
## i           |        0.47 | 0.07 | [ 0.32, 0.61] |   6.35 | < .001
## f1cage      |        0.22 | 0.07 | [ 0.07, 0.37] |   2.99 | 0.003
## f1csex [1]  |        0.33 | 0.15 | [ 0.03, 0.62] |   2.17 | 0.032
## f1mintprb   |       -0.02 | 0.08 | [-0.17, 0.13] |  -0.26 | 0.792
```

```r
mod_ses <-
  lm(
    scale(s) ~
      scale(i) +
      scale(f1cage) +
```

```
      f1csex +
      scale(f1RRses),
    data = d0_lm
  )
anova(mod_sex, mod_ses)
```

```
## Analysis of Variance Table
##
## Model 1: scale(s) ~ scale(i) + scale(f1cage) + f1csex
## Model 2: scale(s) ~ scale(i) + scale(f1cage) + f1csex + scale(f1RRses)
##   Res.Df    RSS Df Sum of Sq      F Pr(>F)
## 1    137 101.39
## 2    136 100.22  1    1.1757 1.5955 0.2087
```

```
model_parameters(mod_ses)
```

```
## Parameter   | Coefficient |   SE |        95% CI | t(136) |      p
## ------------------------------------------------------------------
## (Intercept) |        0.01 | 0.07 | [-0.13, 0.16] |   0.20 | 0.843
## i           |        0.47 | 0.07 | [ 0.32, 0.61] |   6.43 | < .001
## f1cage      |        0.22 | 0.07 | [ 0.07, 0.36] |   2.97 | 0.004
## f1csex [1]  |        0.37 | 0.15 | [ 0.08, 0.66] |   2.49 | 0.014
## f1RRses     |        0.09 | 0.07 | [-0.05, 0.24] |   1.26 | 0.209
```

```
#EP is not related to slope but is related to entropy, leave it
#SES is related to entropy, leave it


mod_str <-
  lm(
    scale(s) ~
      scale(i) +
      scale(f1cage) +
      f1csex +
      scale(f1custructure),
    data = d0_lm
  )
anova(mod_sex, mod_str)
```

```
## Analysis of Variance Table
##
## Model 1: scale(s) ~ scale(i) + scale(f1cage) + f1csex
## Model 2: scale(s) ~ scale(i) + scale(f1cage) + f1csex + scale(f1custructure)
##   Res.Df     RSS Df Sum of Sq     F Pr(>F)
## 1    137 101.391
## 2    136  99.887  1    1.5042 2.048 0.1547
```

```
model_parameters(mod_str)
```

```
## Parameter     | Coefficient |   SE |        95% CI | t(136) |      p
## --------------------------------------------------------------------
## (Intercept)   |        0.01 | 0.07 | [-0.13, 0.16] |   0.18 | 0.855
## i             |        0.48 | 0.07 | [ 0.33, 0.62] |   6.53 | < .001
## f1cage        |        0.25 | 0.08 | [ 0.10, 0.40] |   3.28 | 0.001
## f1csex [1]    |        0.34 | 0.15 | [ 0.05, 0.63] |   2.33 | 0.021
## f1custructure |        0.11 | 0.08 | [-0.04, 0.26] |   1.43 | 0.155
```

```r
mod_per <-
  lm(
    scale(s) ~
      scale(i) +
      scale(f1cage) +
      f1csex +
      scale(f1cupermit),
    data = d0_lm
  )
anova(mod_sex, mod_per)
```

```
## Analysis of Variance Table
##
## Model 1: scale(s) ~ scale(i) + scale(f1cage) + f1csex
## Model 2: scale(s) ~ scale(i) + scale(f1cage) + f1csex + scale(f1cupermit)
##   Res.Df    RSS Df Sum of Sq      F Pr(>F)
## 1    137 101.39
## 2    136 101.14  1   0.25394 0.3415 0.5599
```

```r
model_parameters(mod_per)
```

```
## Parameter   | Coefficient |   SE |        95% CI | t(136) |      p
## -------------------------------------------------------------------
## (Intercept) |        0.01 | 0.07 | [-0.13, 0.16] |   0.18 | 0.857
## i           |        0.47 | 0.07 | [ 0.33, 0.62] |   6.43 | < .001
## f1cage      |        0.22 | 0.07 | [ 0.07, 0.36] |   2.98 | 0.003
## f1csex [1]  |        0.34 | 0.15 | [ 0.05, 0.63] |   2.30 | 0.023
## f1cupermit  |       -0.04 | 0.07 | [-0.19, 0.10] |  -0.58 | 0.560
```

```r
#Parenting does not improve model fit
```

#SEM models using structure + permisiveness // MLR - bootstrap

```r
#Main model

model <- '
s ~ f1csexc + f1cagec + i + f1RRses + f1cupermitc + f1custructurec + scl1dep + scl1depent + f1mextprbc
f1cupermitc ~~ f1custructurec + 0*f1mextprbc
scl1dep ~~ scl1depent + f1mextprbc
scl1depent ~~ f1mextprbc
f1csexc ~~ 0*f1cagec + f1RRses + 0*f1mextprbc
f1cagec ~~ 0*f1RRses  + f1mextprbc + f1custructure c
f1RRses ~~  scl1dep + scl1depent + f1mextprbc
f1RRses ~ 0*1
i ~ 0*1
f1csexc ~ 0*1
f1cagec ~ 0*1
f1cupermitc ~ 0*1
f1custructurec ~ 0*1
'
fit <- sem(model, data = dat, estimator = "MLR", missing = "FIML.x")
summary(fit, standardized=T, fit.measures = T, rsquare=T, ci = T)
```

```
## lavaan 0.6-8 ended normally after 90 iterations
##
##   Estimator                                         ML
```

```
##   Optimization method                           NLMINB
##   Number of model parameters                        33
##
##   Number of observations                           180
##   Number of missing patterns                         6
##
## Model Test User Model:
##                                               Standard      Robust
##   Test Statistic                                20.553      20.722
##   Degrees of freedom                                32          32
##   P-value (Chi-square)                           0.941       0.938
##   Scaling correction factor                                  0.992
##        Yuan-Bentler correction (Mplus variant)
##
## Model Test Baseline Model:
##
##   Test statistic                               330.224     291.585
##   Degrees of freedom                                45          45
##   P-value                                        0.000       0.000
##   Scaling correction factor                                  1.133
##
## User Model versus Baseline Model:
##
##   Comparative Fit Index (CFI)                    1.000       1.000
##   Tucker-Lewis Index (TLI)                       1.056       1.064
##
##   Robust Comparative Fit Index (CFI)                         1.000
##   Robust Tucker-Lewis Index (TLI)                            1.056
##
## Loglikelihood and Information Criteria:
##
##   Loglikelihood user model (H0)              -1949.455   -1949.455
##   Scaling correction factor                                  1.245
##       for the MLR correction
##   Loglikelihood unrestricted model (H1)      -1939.179   -1939.179
##   Scaling correction factor                                  1.121
##       for the MLR correction
##
##   Akaike (AIC)                                3964.911    3964.911
##   Bayesian (BIC)                              4070.278    4070.278
##   Sample-size adjusted Bayesian (BIC)         3965.767    3965.767
##
## Root Mean Square Error of Approximation:
##
##   RMSEA                                          0.000       0.000
##   90 Percent confidence interval - lower         0.000       0.000
##   90 Percent confidence interval - upper         0.012       0.014
##   P-value RMSEA <= 0.05                          0.998       0.998
##
##   Robust RMSEA                                               0.000
##   90 Percent confidence interval - lower                     0.000
##   90 Percent confidence interval - upper                     0.013
##
## Standardized Root Mean Square Residual:
```

```
##
##   SRMR                                          0.048       0.048
##
## Parameter Estimates:
##
##   Standard errors                             Sandwich
##   Information bread                           Observed
##   Observed information based on                Hessian
##
## Regressions:
##                    Estimate  Std.Err  z-value  P(>|z|) ci.lower ci.upper
##   s ~
##     f1csexc           0.101    0.033    3.108    0.002    0.037    0.165
##     f1cagec           0.047    0.018    2.617    0.009    0.012    0.082
##     i                 0.106    0.019    5.534    0.000    0.068    0.143
##     f1RRses           0.044    0.021    2.051    0.040    0.002    0.085
##     f1cupermitc      -0.017    0.020   -0.814    0.416   -0.056    0.023
##     f1custructurec    0.027    0.028    0.971    0.332   -0.027    0.081
##     scl1dep           0.090    0.047    1.903    0.057   -0.003    0.182
##     scl1depent       -0.250    0.111   -2.253    0.024   -0.467   -0.032
##     f1mextprbc       -0.001    0.002   -0.831    0.406   -0.005    0.002
##    Std.lv  Std.all
##
##     0.101    0.197
##     0.047    0.201
##     0.106    0.471
##     0.044    0.147
##    -0.017   -0.051
##     0.027    0.065
##     0.090    0.213
##    -0.250   -0.255
##    -0.001   -0.068
##
## Covariances:
##                    Estimate  Std.Err  z-value  P(>|z|) ci.lower ci.upper
##   f1cupermitc ~~
##     f1custructurec   -0.128    0.037   -3.496    0.000   -0.200   -0.056
##     f1mextprbc        0.000                               0.000    0.000
##   scl1dep ~~
##     scl1depent        0.128    0.013    9.664    0.000    0.102    0.154
##     f1mextprbc        1.043    0.558    1.871    0.061   -0.050    2.137
##   scl1depent ~~
##     f1mextprbc        0.683    0.221    3.093    0.002    0.250    1.116
##   f1csexc ~~
##     f1cagec           0.000                               0.000    0.000
##     f1RRses          -0.064    0.031   -2.079    0.038   -0.124   -0.004
##     f1mextprbc        0.000                               0.000    0.000
##   f1cagec ~~
##     f1RRses           0.000                               0.000    0.000
##     f1mextprbc        1.904    0.911    2.090    0.037    0.118    3.691
##     f1custructurec   -0.186    0.045   -4.144    0.000   -0.274   -0.098
##   f1RRses ~~
##     scl1dep           0.098    0.033    2.978    0.003    0.034    0.163
##     scl1depent        0.050    0.016    3.013    0.003    0.017    0.082
```

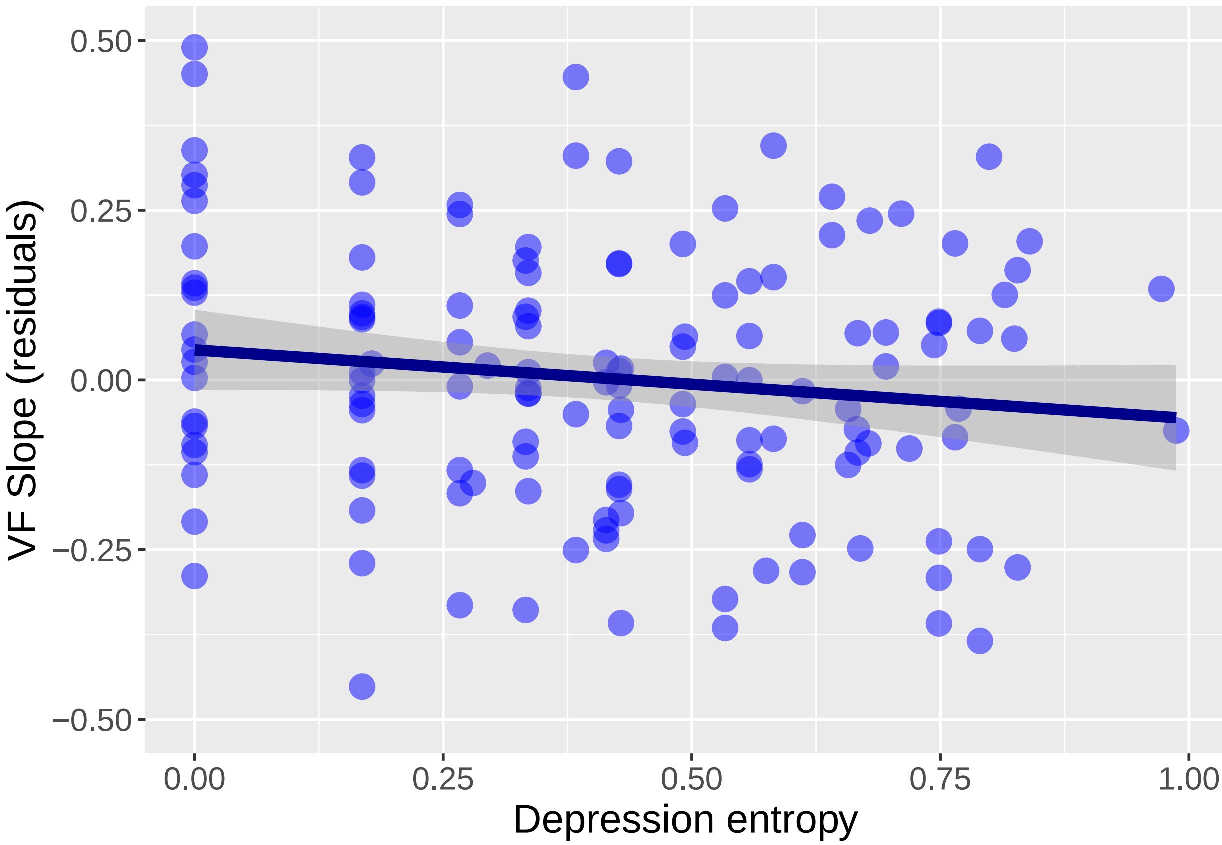
```
##     f1mextprbc        2.294    0.749    3.061    0.002    0.825    3.762
##    Std.lv  Std.all
##
##    -0.128   -0.259
##     0.000    0.000
##
##     0.128    0.806
##     1.043    0.148
##
##     0.683    0.225
##
##     0.000    0.000
##    -0.064   -0.148
##     0.000    0.000
##
##     0.000    0.000
##     1.904    0.150
##    -0.186   -0.273
##
##     0.098    0.186
##     0.050    0.219
##     2.294    0.229
##
## Intercepts:
##                    Estimate  Std.Err  z-value  P(>|z|) ci.lower ci.upper
##     f1RRses           0.000                               0.000    0.000
##     i                 0.000                               0.000    0.000
##     f1csexc           0.000                               0.000    0.000
##     f1cagec           0.000                               0.000    0.000
##     f1cupermitc       0.000                               0.000    0.000
##     f1custructurec    0.000                               0.000    0.000
##    .s                 0.053    0.036    1.468    0.142   -0.018    0.125
##     scl1dep           0.537    0.045   11.909    0.000    0.449    0.625
##     scl1depent        0.409    0.019   21.383    0.000    0.371    0.446
##     f1mextprbc        0.112    0.842    0.134    0.894   -1.538    1.763
##    Std.lv  Std.all
##     0.000    0.000
##     0.000    0.000
##     0.000    0.000
##     0.000    0.000
##     0.000    0.000
##     0.000    0.000
##     0.053    0.208
##     0.537    0.882
##     0.409    1.562
##     0.112    0.010
##
## Variances:
##                    Estimate  Std.Err  z-value  P(>|z|) ci.lower ci.upper
##    .s                 0.044    0.006    7.513    0.000    0.032    0.055
##     f1csexc           0.249    0.002  120.561    0.000    0.245    0.253
##     f1cagec           1.203    0.050   24.034    0.000    1.105    1.301
##     i                 1.309    0.148    8.864    0.000    1.019    1.598
##     f1RRses           0.749    0.082    9.174    0.000    0.589    0.909
```

```
##     f1cupermitc       0.632    0.094    6.712    0.000    0.447    0.816
##     f1custructurec    0.385    0.042    9.270    0.000    0.303    0.466
##     scl1dep           0.370    0.074    5.036    0.000    0.226    0.514
##     scl1depent        0.068    0.005   12.872    0.000    0.058    0.079
##     f1mextprbc      134.345   10.499   12.796    0.000  113.767  154.923
##    Std.lv  Std.all
##     0.044    0.667
##     0.249    1.000
##     1.203    1.000
##     1.309    1.000
##     0.749    1.000
##     0.632    1.000
##     0.385    1.000
##     0.370    1.000
##     0.068    1.000
##   134.345    1.000
##
## R-Square:
##                    Estimate
##     s                 0.333
```

##Creating graphs

```
####

library(modelr)
library(ggplot2)

bestcov <- lm(
s ~ f1csexc + f1cagec + i + f1RRses + f1cupermitc + f1custructurec + scl1dep + f1mextprbc,
  data = dat
)

dat <-
  dat %>%
  add_residuals(bestcov, var = "resid_s")

dat %>%
  ggplot(aes(scl1depent, resid_s)) +
  geom_point(size = 4, alpha = 1/2, colour = "blue") +
  geom_smooth(method = "lm", size = 2, color = "darkblue") + # +
  ylim(-0.5,0.5)  +
  #xlim(-0.5,0.5)  +
  labs(
    x = "Depression entropy",
    y = "VF Slope (residuals)"
  ) + theme(text = element_text(size=15))
```

```
## `geom_smooth()` using formula 'y ~ x'
```



Extra 1: Entropy calculations

```
scl1 <- read.csv("F1 PPD SCL-90.csv")

#scl1 <- scl1 %>% rename(fid = ?..fid)

scl1$VAR00001 <- NULL

#Entropy function can be found in the Conte Center website

entropy <- function(p){
  iz <- which(p==0)
  z <- -p * log2(p)
  if (length(iz))
    z[iz] <- 0
  sum(z) }
norm_entropy_E <-
  function(x,E=0:4,minnum=floor(length(x)/2)){
    if(sum(is.na(x)) > minnum) return(NA)
    xf <- factor(x,levels=E)
    p <- prop.table(table(xf))
    k <- length(E)
    nent <- 100*entropy(p)/log2(k)
    nent
  }
```

```
####SCL 1 #####

scl1 <- scl1[c(1,6,15,16,21,23,27,30:33,55,72,80,
               3,18,24,34,40,58,73,79,81,87,
               12,25,64,68,75,82,
               2,5,13,28,41,43,49,50,53,54,57,59,
               4,10,11,29,39,46,47,52,56,66,
               7,22,35,37,38,42,62,70,74,
               14,26,48,51,71,76,83,
               9,19,44,69,77,84,
               8,17,36,63,78,85,86,88,89,91)]


scl1$scl1depent<- apply(scl1[,2:14],1,FUN=norm_entropy_E)
scl1$scl1dep <- apply(scl1[,2:14], 1, mean, na.rm = T)
corr.test(scl1$scl1depent, scl1$scl1dep)
```

```
## Call:corr.test(x = scl1$scl1depent, y = scl1$scl1dep)
## Correlation matrix
## [1] 0.81
## Sample Size
## [1] 180
## Probability values  adjusted for multiple tests.
## [1] 0
##
##  To see confidence intervals of the correlations, print with the short=FALSE option
```

```
#write.table(scl, "scl_t1.csv", na=".", sep=",", col.names = T, row.names = F)

#######HBQ#######
entropy <- function(p){
  iz <- which(p==0)
  z <- -p * log2(p)
  if (length(iz))
    z[iz] <- 0
  sum(z) }
norm_entropy_E2 <-
  function(x,E=0:2,minnum=floor(length(x)/2)){
    if(sum(is.na(x)) > minnum) return(NA)
    xf <- factor(x,levels=E)
    p <- prop.table(table(xf))
    k <- length(E)
    nent <- 100*entropy(p)/log2(k)
    nent
  }

entropy <- function(p){
  iz <- which(p==0)
  z <- -p * log2(p)
  if (length(iz))
    z[iz] <- 0
  sum(z) }
norm_entropy_E3 <-
```

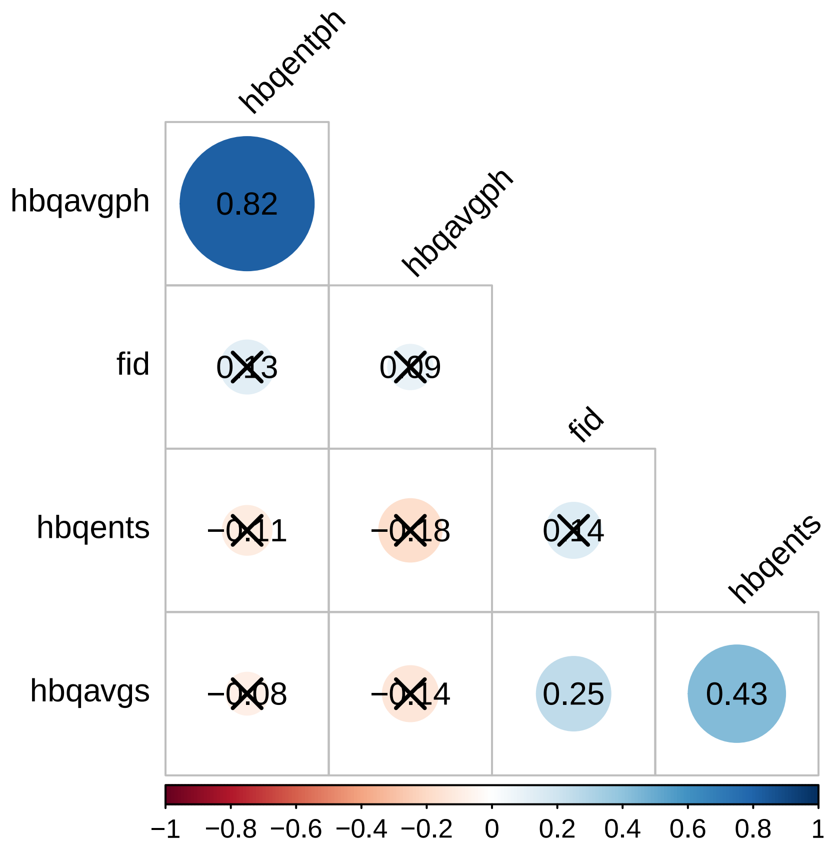
```
  function(x,E=0:3,minnum=floor(length(x)/2)){
    if(sum(is.na(x)) > minnum) return(NA)
    xf <- factor(x,levels=E)
    p <- prop.table(table(xf))
    k <- length(E)
    nent <- 100*entropy(p)/log2(k)
    nent
  }

## HBQ 4 yr old children (1/2 sample)

hbq4 <- read.csv("F1 PPD hbq4.csv")
hbq4 <- hbq4[c(1,3:7,9,10,190,14, 44,47,50)]
hbq4$hbqentph <- apply(hbq4[,2:6],1,FUN=norm_entropy_E3)
hbq4$hbqavgph <- apply(hbq4[,2:6], 1, mean, na.rm = T)
hbq4$hbqents <- apply(hbq4[,11:13],1,FUN=norm_entropy_E2)
hbq4$hbqavgs <- apply(hbq4[,11:13], 1, mean, na.rm = T)
hbq4 <- hbq4[c(1,14:17)]
library(rstatix)
cor.mat <- cor_mat(hbq4)
cor.mat
```

```
## # A tibble: 5 x 6
##   rowname    fid hbqentph hbqavgph hbqents hbqavgs
## * <chr>    <dbl>    <dbl>    <dbl>   <dbl>   <dbl>
## 1 fid      1         0.13     0.091    0.14    0.25
## 2 hbqentph 0.13      1        0.82    -0.11  -0.081
## 3 hbqavgph 0.091     0.82     1       -0.18  -0.14
## 4 hbqents  0.14    -0.11    -0.18      1       0.43
## 5 hbqavgs  0.25    -0.081   -0.14      0.43    1
```

```
cor.mat %>% cor_get_pval()
```

```
## # A tibble: 5 x 6
##   rowname     fid hbqentph hbqavgph   hbqents   hbqavgs
##   <chr>     <dbl>    <dbl>    <dbl>     <dbl>     <dbl>
## 1 fid      0      1.97e- 1 3.78e- 1 0.181     0.0122
## 2 hbqentph 0.197  0        3.77e-25 0.302     0.431
## 3 hbqavgph 0.378  3.77e-25 0        0.0853    0.172
## 4 hbqents  0.181  3.02e- 1 8.53e- 2 0         0.0000118
## 5 hbqavgs  0.0122 4.31e- 1 1.72e- 1 0.0000118 0
```

```
cor.mat %>%
  cor_reorder() %>%
  pull_lower_triangle() %>%
  cor_plot(label = TRUE)
```




```
## HBQ 6 yr old children (1/2 sample)

hbq6 <- read.csv("F1 PPD hbq6.csv")
hbq6 <- hbq6[c(1,3:7,9,10,178,14,44,47,50)]
hbq6$hbqentph <- apply(hbq6[,2:6],1,FUN=norm_entropy_E3)
hbq6$hbqavgph <- apply(hbq6[,2:6], 1, mean, na.rm = T)
hbq6$hbqents <- apply(hbq6[,11:13],1,FUN=norm_entropy_E2)
hbq6$hbqavgs <- apply(hbq6[,11:13], 1, mean, na.rm = T)
hbq6 <- hbq6[c(1,14:17)]
library(rstatix)
cor.mat <- cor_mat(hbq6)
cor.mat
```

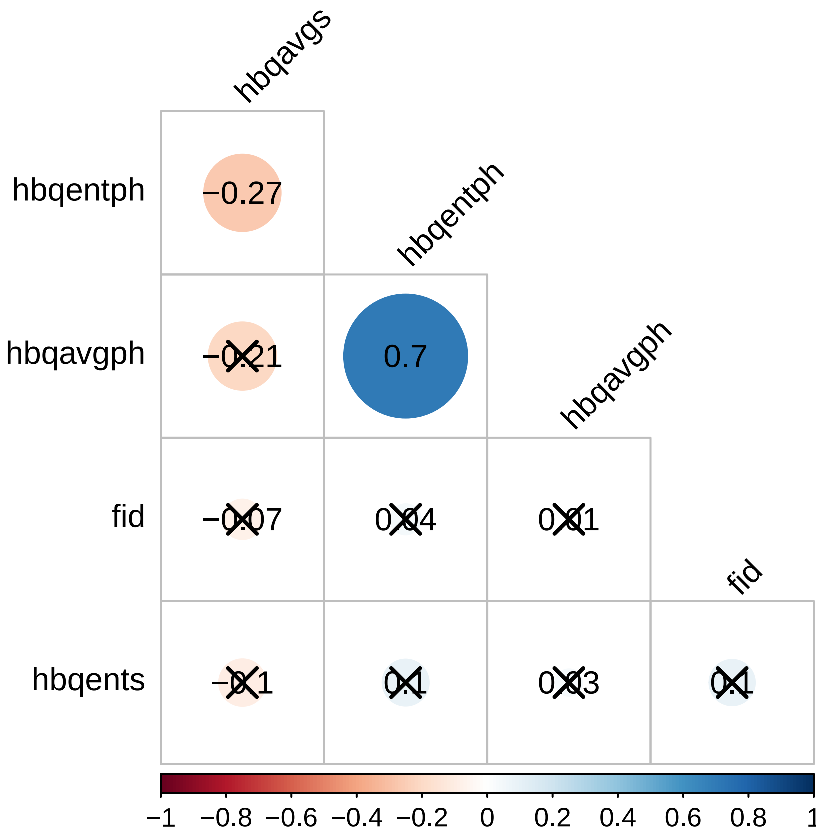
```
## # A tibble: 5 x 6
##   rowname      fid hbqentph hbqavgph hbqents hbqavgs
## * <chr>      <dbl>    <dbl>    <dbl>   <dbl>   <dbl>
## 1 fid       1         0.042   0.0054   0.095  -0.072
## 2 hbqentph  0.042     1       0.7      0.098  -0.27
## 3 hbqavgph  0.0054    0.7     1        0.032  -0.21
## 4 hbqents   0.095     0.098   0.032    1      -0.1
## 5 hbqavgs  -0.072    -0.27   -0.21    -0.1     1
```

```
cor.mat %>% cor_get_pval()
```

```
## # A tibble: 5 x 6
##   rowname    fid hbqentph hbqavgph hbqents hbqavgs
##   <chr>    <dbl>    <dbl>    <dbl>   <dbl>   <dbl>
## 1 fid      0      7.1 e- 1 9.62e- 1   0.402   0.528
```

```
## 2 hbqentph 0.71  0         5.63e-13   0.387  0.0147
## 3 hbqavgph 0.962 5.63e-13 0          0.78   0.062
## 4 hbqents  0.402 3.87e- 1 7.8 e- 1   0      0.369
## 5 hbqavgs  0.528 1.47e- 2 6.2 e- 2   0.369  0
```

```
cor.mat %>%
  cor_reorder() %>%
  pull_lower_triangle() %>%
  cor_plot(label = TRUE)
```




```
hbq <- rbind(hbq4,hbq6)
#library(data.table)
#write.table(hbq, "hbq_t1.csv", na=".", sep=",",row.names = F)
```

Extra 2: Vagal flexibility

```
library(foreign)
library(lavaan)
library(psych)

rsa.data = read.spss('f1 ppd cardio zudoc clean 02112020.sav', to.data.frame = TRUE)

names(rsa.data)
```

```
##   [1] "fid"       "f1zanote"  "f1zmnote"  "f1ztnote"  "f1za1vt"   "f1za2vt"
##   [7] "f1za3vt"   "f1za4vt"   "f1zm1vt"   "f1zm2vt"   "f1zm3vt"   "f1zm4vt"
##  [13] "f1zt1vt"   "f1zt2vt"   "f1zt3vt"   "f1zt4vt"   "f1ziedt"   "f1zini"
```

```
##  [19] "f1zidur"   "f1zifvt"   "f1zievt"   "f1zivs"    "f1zinote"  "f1zs1edt"
##  [25] "f1zs1ni"   "f1zs1dur"  "f1zs1fvt"  "f1zws1not" "f1zaedt"   "f1zani"
##  [31] "f1zadur"   "f1zafvt"   "f1zaevt"   "f1zavs"    "f1za1ni"   "f1za1dur"
##  [37] "f1za2ni"   "f1za2dur"  "f1za3ni"   "f1za3dur"  "f1za4ni"   "f1za4dur"
##  [43] "f1zs2edt"  "f1zs2ni"   "f1zs2dur"  "f1zs2fvt"  "f1zs2not"  "f1zmedt"
##  [49] "f1zmni"    "f1zmdur"   "f1zmfvt"   "f1zmevt"   "f1zmvs"    "f1zm1ni"
##  [55] "f1zm1dur"  "f1zm2ni"   "f1zm2dur"  "f1zm3ni"   "f1zm3dur"  "f1zm4ni"
##  [61] "f1zm4dur"  "f1zs3edt"  "f1zs3ni"   "f1zs3dur"  "f1zs3fvt"  "f1zs3not"
##  [67] "f1ztedt"   "f1ztni"    "f1ztdur"   "f1ztfvt"   "f1ztevt"   "f1ztvs"
##  [73] "f1zt1ni"   "f1zt1dur"  "f1zt2ni"   "f1zt2dur"  "f1zt3ni"   "f1zt3dur"
##  [79] "f1zt4ni"   "f1zt4dur"  "f1zs4edt"  "f1zs4ni"   "f1zs4dur"  "f1zs4fvt"
##  [85] "f1zs4not"  "f1zgedt"   "f1zgni"    "f1zgdur"   "f1zgfvt"   "f1zgevt"
##  [91] "f1zgvs"    "f1zgnote"  "f1zg1ni"   "f1zg1dur"  "f1zg1vt"   "f1zg2ni"
##  [97] "f1zg2dur"  "f1zg2vt"   "f1zg3ni"   "f1zg3dur"  "f1zg3vt"   "f1zg4ni"
## [103] "f1zg4dur"  "f1zg4vt"   "f1zs5edt"  "f1zs5ni"   "f1zs5dur"  "f1zs5fvt"
## [109] "f1zs5not"  "f1zcedt"   "f1zcni"    "f1zcdur"   "f1zcfvt"   "f1zcevt"
## [115] "f1zcvs"    "f1zcnote"  "f1zc1ni"   "f1zc1dur"  "f1zc1vt"   "f1zc2ni"
## [121] "f1zc2dur"  "f1zc2vt"   "f1zc3ni"   "f1zc3dur"  "f1zc3vt"   "f1zc4ni"
## [127] "f1zc4dur"  "f1zc4vt"   "f1zs6edt"  "f1zs6ni"   "f1zs6dur"  "f1zs6fvt"
## [133] "f1zs6note" "ZRE_1"     "ZRE_2"     "f1savf"    "ZRE_3"     "ZRE_4"
## [139] "f1fevf"    "ZRE_5"     "ZRE_6"     "f1anvf"
```

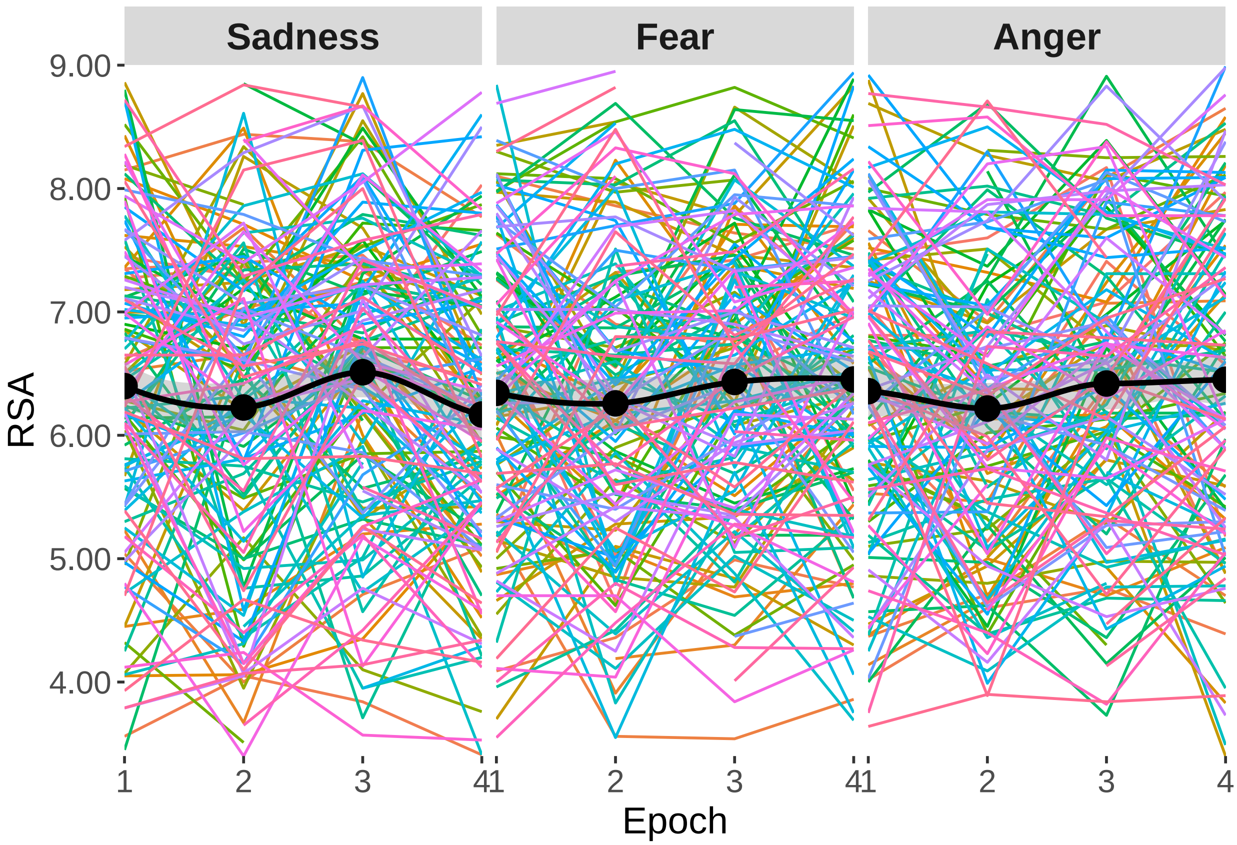
```
#Creating spaghetti plots
fig <- rsa.data[c(1,5:16)]
#fig$emo <- c('Sadness','Sadness','Sadness','Sadness','Fear','Fear', 'Fear', 'Fear', 'Anger','Anger','A
#'Anger')
head(fig)
```

```
##    fid f1za1vt f1za2vt f1za3vt f1za4vt f1zm1vt f1zm2vt f1zm3vt f1zm4vt f1zt1vt
## 1 6001    7.45    7.28    6.73    8.03    6.78    5.03    6.53    6.99    6.81
## 2 6003    7.35    6.58    7.33    6.73    6.89    4.65    7.29    5.55    7.46
## 3 6004    7.27    7.37    6.68    6.45    7.26    6.55    7.86    6.95    7.05
## 4 6006    7.35    7.40    7.46    6.58    5.05    7.32    7.42    7.71    6.11
## 5 6007    6.62    7.03    7.24    6.61    7.63    6.90    6.72    7.72    6.67
## 6 6008    3.56    4.05    3.84    3.41    4.09    4.36    4.99    4.77    4.02
##   f1zt2vt f1zt3vt f1zt4vt
## 1    6.81    7.08    7.97
## 2    7.60    8.17    5.18
## 3    5.89    7.41    6.57
## 4    5.14    6.41    7.92
## 5    5.93    6.95    4.88
## 6    4.60    4.76    4.39
```

```
library(tidyr)
fig <- gather(fig, condition, RSA, f1za1vt:f1zt4vt, factor_key=FALSE)
fig$emo <- NA
fig$emo[fig$condition == "f1za1vt"] <- "Sadness"
fig$emo[fig$condition == "f1za2vt"] <- "Sadness"
fig$emo[fig$condition == "f1za3vt"] <- "Sadness"
fig$emo[fig$condition == "f1za4vt"] <- "Sadness"

fig$emo[fig$condition == "f1zm1vt"] <- "Fear"
fig$emo[fig$condition == "f1zm2vt"] <- "Fear"
fig$emo[fig$condition == "f1zm3vt"] <- "Fear"
fig$emo[fig$condition == "f1zm4vt"] <- "Fear"
```

```
fig$emo[fig$condition == "f1zt1vt"] <- "Anger"
fig$emo[fig$condition == "f1zt2vt"] <- "Anger"
fig$emo[fig$condition == "f1zt3vt"] <- "Anger"
fig$emo[fig$condition == "f1zt4vt"] <- "Anger"


fig$time <- NA
fig$time[fig$condition == "f1za1vt"] <- "1"
fig$time[fig$condition == "f1za2vt"] <- "2"
fig$time[fig$condition == "f1za3vt"] <- "3"
fig$time[fig$condition == "f1za4vt"] <- "4"

fig$time[fig$condition == "f1zm1vt"] <- "1"
fig$time[fig$condition == "f1zm2vt"] <- "2"
fig$time[fig$condition == "f1zm3vt"] <- "3"
fig$time[fig$condition == "f1zm4vt"] <- "4"

fig$time[fig$condition == "f1zt1vt"] <- "1"
fig$time[fig$condition == "f1zt2vt"] <- "2"
fig$time[fig$condition == "f1zt3vt"] <- "3"
fig$time[fig$condition == "f1zt4vt"] <- "4"

library(ggplot2)

fig$emo_f = factor(fig$emo, levels=c('Sadness','Fear','Anger'))
fig$time <- as.numeric(fig$time)

fmt_dcimals <- function(decimals=0){
  function(x) format(x,nsmall = decimals,scientific = FALSE)
}

p <- ggplot(data = fig, aes(x = time, y = RSA,group = fid, colour = factor(fid)))
p + geom_line() + stat_smooth(aes(group = 1), colour = "black") + stat_summary(aes(group = 1),
 geom = "point", fun.y = mean, colour = "black", size = 4) + facet_grid(. ~ emo_f) +
  theme(legend.position="none") + scale_y_continuous(labels = fmt_dcimals(2), limits = c(3.40,9), expan
  scale_x_continuous(name="Epoch", limits = c(1,4), expand = c(0, 0)) +
  theme(panel.background =  element_blank()) + theme(axis.text=element_text(size=12),axis.title=element_
  theme(strip.text.x = element_text(size = 14, face = "bold"))
```

```
## `geom_smooth()` using method = 'loess' and formula 'y ~ x'
```




```
#Modeling vagal flexibility
vf.model <-'
i1 =~ 1* f1za1vt + 1* f1za2vt + 1* f1za3vt + 1* f1za4vt
s1 =~ 0* f1za1vt + -1*f1za2vt + f1za3vt + f1za4vt
i2 =~ 1* f1zm1vt + 1* f1zm2vt + 1* f1zm3vt + 1* f1zm4vt
s2 =~ 0* f1zm1vt + -1*f1zm2vt + f1zm3vt + f1zm4vt
i3 =~ 1* f1zt1vt + 1* f1zt2vt + 1* f1zt3vt + 1* f1zt4vt
s3 =~ 0* f1zt1vt + -1*f1zt2vt + f1zt3vt + f1zt4vt
#residual variances
f1za1vt~~r1*f1za1vt
f1za2vt~~r1*f1za2vt
f1za3vt~~r1*f1za3vt
f1za4vt~~r1*f1za4vt
#residual variances
f1zm1vt~~r2*f1zm1vt
f1zm2vt~~r2*f1zm2vt
f1zm3vt~~r2*f1zm3vt
f1zm4vt~~r2*f1zm4vt
#residual variances
f1zt1vt~~r3*f1zt1vt
f1zt2vt~~0*f1zt2vt
f1zt3vt~~r3*f1zt3vt
f1zt4vt~~r3*f1zt4vt
#second order growth factors
i =~  i1 + i2 + i3
s =~  1*s1 + 1*s2 + 1*s3
#means
```

```
i ~ 0
s ~ 0
i1~ 1
i2~ 1
i3~ 1
s1~ 1
s2~ 1
s3~ 1
#residual variances
i1~~r*i1
i2~~r*i2
i3~~r*i3
s1~~t*s1
s2~~t*s2
s3~~t*s3
i ~~ i
s ~~ s
i ~~ s
'
vf.fit <- growth(vf.model, data = rsa.data, missing = 'FIMl')
summary(vf.fit, fit.measures=TRUE, standardized=TRUE)
```

```
## lavaan 0.6-8 ended normally after 60 iterations
##
##   Estimator                                         ML
##   Optimization method                           NLMINB
##   Number of model parameters                        34
##   Number of equality constraints                    12
##
##                                                   Used       Total
##   Number of observations                           154         180
##   Number of missing patterns                         4
##
## Model Test User Model:
##
##   Test statistic                                78.300
##   Degrees of freedom                                68
##   P-value (Chi-square)                           0.184
##
## Model Test Baseline Model:
##
##   Test statistic                              1574.955
##   Degrees of freedom                                66
##   P-value                                        0.000
##
## User Model versus Baseline Model:
##
##   Comparative Fit Index (CFI)                    0.993
##   Tucker-Lewis Index (TLI)                       0.993
##
## Loglikelihood and Information Criteria:
##
##   Loglikelihood user model (H0)              -2541.380
##   Loglikelihood unrestricted model (H1)      -2502.230
```

```
##
##   Akaike (AIC)                                5126.760
##   Bayesian (BIC)                              5193.573
##   Sample-size adjusted Bayesian (BIC)         5123.940
##
## Root Mean Square Error of Approximation:
##
##   RMSEA                                          0.031
##   90 Percent confidence interval - lower         0.000
##   90 Percent confidence interval - upper         0.059
##   P-value RMSEA <= 0.05                          0.849
##
## Standardized Root Mean Square Residual:
##
##   SRMR                                           0.041
##
## Parameter Estimates:
##
##   Standard errors                             Standard
##   Information                                 Observed
##   Observed information based on                Hessian
##
## Latent Variables:
##                    Estimate  Std.Err  z-value  P(>|z|)   Std.lv  Std.all
##   i1 =~
##     f1za1vt           1.000                               1.178    0.835
##     f1za2vt           1.000                               1.178    0.768
##     f1za3vt           1.000                               1.178    0.832
##     f1za4vt           1.000                               1.178    0.843
##   s1 =~
##     f1za1vt           0.000                               0.000    0.000
##     f1za2vt          -1.000                              -0.834   -0.544
##     f1za3vt           0.042    0.116    0.363    0.716    0.035    0.025
##     f1za4vt          -0.185    0.111   -1.670    0.095   -0.154   -0.110
##   i2 =~
##     f1zm1vt           1.000                               1.211    0.851
##     f1zm2vt           1.000                               1.211    0.785
##     f1zm3vt           1.000                               1.211    0.858
##     f1zm4vt           1.000                               1.211    0.858
##   s2 =~
##     f1zm1vt           0.000                               0.000    0.000
##     f1zm2vt          -1.000                              -0.834   -0.541
##     f1zm3vt          -0.142    0.102   -1.397    0.162   -0.119   -0.084
##     f1zm4vt          -0.357    0.104   -3.426    0.001   -0.298   -0.211
##   i3 =~
##     f1zt1vt           1.000                               1.265    0.825
##     f1zt2vt           1.000                               1.265    0.908
##     f1zt3vt           1.000                               1.265    0.829
##     f1zt4vt           1.000                               1.265    0.823
##   s3 =~
##     f1zt1vt           0.000                               0.000    0.000
##     f1zt2vt          -1.000                              -0.834   -0.599
##     f1zt3vt          -0.091    0.091   -0.993    0.321   -0.076   -0.050
##     f1zt4vt           0.019    0.096    0.200    0.841    0.016    0.010
```

```
##   i =~
##     i1                1.000                               0.996    0.996
##     i2                1.028    0.045   22.798    0.000    0.996    0.996
##     i3                1.075    0.046   23.331    0.000    0.996    0.996
##   s =~
##     s1                1.000                               0.510    0.510
##     s2                1.000                               0.510    0.510
##     s3                1.000                               0.510    0.510
##
## Covariances:
##                    Estimate  Std.Err  z-value  P(>|z|)   Std.lv  Std.all
##   i ~~
##     s                 0.166    0.078    2.129    0.033    0.332    0.332
##
## Intercepts:
##                    Estimate  Std.Err  z-value  P(>|z|)   Std.lv  Std.all
##     i                 0.000                               0.000    0.000
##     s                 0.000                               0.000    0.000
##    .i1                6.373    0.104   61.567    0.000    5.410    5.410
##    .i2                6.407    0.107   60.124    0.000    5.290    5.290
##    .i3                6.396    0.111   57.759    0.000    5.056    5.056
##    .s1                0.281    0.102    2.743    0.006    0.336    0.336
##    .s2                0.263    0.104    2.523    0.012    0.315    0.315
##    .s3                0.247    0.080    3.079    0.002    0.297    0.297
##    .f1za1vt           0.000                               0.000    0.000
##    .f1za2vt           0.000                               0.000    0.000
##    .f1za3vt           0.000                               0.000    0.000
##    .f1za4vt           0.000                               0.000    0.000
##    .f1zm1vt           0.000                               0.000    0.000
##    .f1zm2vt           0.000                               0.000    0.000
##    .f1zm3vt           0.000                               0.000    0.000
##    .f1zm4vt           0.000                               0.000    0.000
##    .f1zt1vt           0.000                               0.000    0.000
##    .f1zt2vt           0.000                               0.000    0.000
##    .f1zt3vt           0.000                               0.000    0.000
##    .f1zt4vt           0.000                               0.000    0.000
##
## Variances:
##                    Estimate  Std.Err  z-value  P(>|z|)   Std.lv  Std.all
##    .f1za1vt   (r1)    0.602    0.044   13.723    0.000    0.602    0.303
##    .f1za2vt   (r1)    0.602    0.044   13.723    0.000    0.602    0.256
##    .f1za3vt   (r1)    0.602    0.044   13.723    0.000    0.602    0.300
##    .f1za4vt   (r1)    0.602    0.044   13.723    0.000    0.602    0.308
##    .f1zm1vt   (r2)    0.559    0.040   13.963    0.000    0.559    0.276
##    .f1zm2vt   (r2)    0.559    0.040   13.963    0.000    0.559    0.235
##    .f1zm3vt   (r2)    0.559    0.040   13.963    0.000    0.559    0.281
##    .f1zm4vt   (r2)    0.559    0.040   13.963    0.000    0.559    0.281
##    .f1zt1vt   (r3)    0.753    0.055   13.574    0.000    0.753    0.320
##    .f1zt2vt           0.000                               0.000    0.000
##    .f1zt3vt   (r3)    0.753    0.055   13.574    0.000    0.753    0.324
##    .f1zt4vt   (r3)    0.753    0.055   13.574    0.000    0.753    0.319
##    .i1         (r)    0.012    0.019    0.642    0.521    0.009    0.009
##    .i2         (r)    0.012    0.019    0.642    0.521    0.008    0.008
##    .i3         (r)    0.012    0.019    0.642    0.521    0.008    0.008
```

```
##    .s1         (t)    0.516    0.069    7.524    0.000    0.740    0.740
##    .s2         (t)    0.516    0.069    7.524    0.000    0.740    0.740
##    .s3         (t)    0.516    0.069    7.524    0.000    0.740    0.740
##     i                 1.375    0.180    7.626    0.000    1.000    1.000
##     s                 0.181    0.072    2.526    0.012    1.000    1.000
```

```
fitmeasures(vf.fit)
```

```
##                npar                fmin               chisq                  df
##              22.000               0.254              78.300              68.000
##              pvalue      baseline.chisq         baseline.df     baseline.pvalue
##               0.184            1574.955              66.000               0.000
##                 cfi                 tli                nnfi                 rfi
##               0.993               0.993               0.993                  NA
##                 nfi                pnfi                 ifi                 rni
##                  NA               0.979               0.993               0.993
##                logl   unrestricted.logl                 aic                 bic
##           -2541.380           -2502.230            5126.760            5193.573
##              ntotal                bic2               rmsea      rmsea.ci.lower
##             154.000            5123.940               0.031               0.000
##      rmsea.ci.upper        rmsea.pvalue                 rmr          rmr_nomean
##               0.059               0.849               0.085               0.088
##                srmr        srmr_bentler srmr_bentler_nomean                crmr
##               0.041               0.041               0.040               0.058
##         crmr_nomean          srmr_mplus   srmr_mplus_nomean               cn_05
##               0.030               0.057               0.034             174.570
##               cn_01                 gfi                agfi                pgfi
##             193.802               0.986               0.982               0.745
##                 mfi                ecvi
##               0.967               0.794
```

```
head(lavPredict(vf.fit))
```

```
##            i1          s1       i2          s2       i3         s3          i
## [1,] 7.094498  0.07188357 7.096149  1.19111647 7.154874  0.3448742  0.6973987
## [2,] 6.925475  0.37928184 6.932469  1.39406290 6.991670 -0.6083300  0.5380583
## [3,] 6.973171  0.09857066 7.047401  0.39818849 7.024596  1.1345963  0.6007449
## [4,] 6.790471 -0.03458508 6.801922 -0.23162686 6.783897  1.6438966  0.3852910
## [5,] 6.755073  0.06361373 6.824583 -0.02620553 6.753383  0.8233830  0.3711224
## [6,] 4.316871  0.17699304 4.352381 -0.12978594 4.248804 -0.3511958 -2.0105788
##                s
## [1,]  0.17507993
## [2,]  0.09380286
## [3,]  0.17284817
## [4,]  0.11856371
## [5,]  0.03439217
## [6,] -0.30152441
```

```
#Creating new data set
rsafinal = rsa.data[c("fid")]

## merge factor scores to original data.frame
fid <- lavInspect(vf.fit, "case.idx")
fscores <- predict(vf.fit)
## loop over factors
for (fs in colnames(fscores)) {
```

```r
  rsafinal[fid, fs] <- fscores[ , fs]
}
head(rsafinal)
```

```
##    fid       i1          s1       i2          s2       i3         s3          i
## 1 6001 7.094498  0.07188357 7.096149  1.19111647 7.154874  0.3448742  0.6973987
## 2 6003 6.925475  0.37928184 6.932469  1.39406290 6.991670 -0.6083300  0.5380583
## 3 6004 6.973171  0.09857066 7.047401  0.39818849 7.024596  1.1345963  0.6007449
## 4 6006 6.790471 -0.03458508 6.801922 -0.23162686 6.783897  1.6438966  0.3852910
## 5 6007 6.755073  0.06361373 6.824583 -0.02620553 6.753383  0.8233830  0.3711224
## 6 6008 4.316871  0.17699304 4.352381 -0.12978594 4.248804 -0.3511958 -2.0105788
##             s
## 1  0.17507993
## 2  0.09380286
## 3  0.17284817
## 4  0.11856371
## 5  0.03439217
## 6 -0.30152441
```

```r
names(rsafinal) <- c("fid", "isad", "sad", "ifear", "sfear", "ianger", "sanger", "i", "s" )
head(rsafinal)
```

```
##    fid     isad         sad    ifear       sfear   ianger     sanger          i
## 1 6001 7.094498  0.07188357 7.096149  1.19111647 7.154874  0.3448742  0.6973987
## 2 6003 6.925475  0.37928184 6.932469  1.39406290 6.991670 -0.6083300  0.5380583
## 3 6004 6.973171  0.09857066 7.047401  0.39818849 7.024596  1.1345963  0.6007449
## 4 6006 6.790471 -0.03458508 6.801922 -0.23162686 6.783897  1.6438966  0.3852910
## 5 6007 6.755073  0.06361373 6.824583 -0.02620553 6.753383  0.8233830  0.3711224
## 6 6008 4.316871  0.17699304 4.352381 -0.12978594 4.248804 -0.3511958 -2.0105788
##             s
## 1  0.17507993
## 2  0.09380286
## 3  0.17284817
## 4  0.11856371
## 5  0.03439217
## 6 -0.30152441
```