

Script to analyze the data for the Jarrett, Farwell and Herzberg Plaid design using asreml and asremlPlus

Data available from <https://doi.org/10.25909/13122095>

Initialize

```
library(knitr)
#knitr::spin("FHanal.r")
library(dae)
packageVersion("dae")

## [1] '3.1.38'

library(asreml)

## Loading required package: Matrix
packageVersion("asreml")

## [1] '4.1.0.130'

library(asremlPlus)

## The methods for alldiffs and data.frame do not require asreml
packageVersion("asremlPlus")

## [1] '4.2.32'

library(ggplot2)
library(qqplotr)

##
## Attaching package: 'qqplotr'

## The following objects are masked from 'package:ggplot2':
##
##   stat_qq_line, StatQqLine

options(width = 100)
opts_chunk$set("tidy" = FALSE, "dev" = "png", "dpi" = 300, fig.path = "figures/",
               "fig.width" = 7.5, "fig.height" = 7.5)
```

A set of colour-blind friendly colours

```
cbPalette <- c("#CC79A7", "#56B4E9", "#009E73", "#E69F00",
               "#F0E442", "#0072B2", "#D55E00", "#000000")
```

Load data

```
load("plaid.dat.rda")
```

Add factors for heterogeneous variances

```
plaid.dat <- within(plaid.dat,
                    {
```

```

        MotionsExpress <- fac.combine(list(Motions, Expressiveness),
                                         combine.levels = TRUE)
        WMotionsExpress <- fac.nested(MotionsExpress)
    })
plaid.dat <- with(plaid.dat, plaid.dat[order(MotionsExpress, WMotionsExpress), ])

```

Function to produce diagnostic plots

```

plotDiagnostics <- function(asrtests.obj, dat)
{
  asreml.obj <- asrtests.obj$asreml.obj
  dat <- within(dat,
    {
      res <- residuals(asreml.obj, type="stdCond")
      fit <- fitted(asreml.obj)
    })

  faceting <- list(facet_grid(rows = vars(Expressiveness), cols = vars(Motions)),
    aes(colour=Trainings),
    scale_colour_manual(values = cbPalette[3:4]))

  resfitplt <- ggplot(data=dat, aes_string(y="res", x="fit")) +
    labs(x = "Fitted values",
      y = "Standardized residuals") +
    geom_point(na.rm = TRUE, alpha=0.75, size=1.25) +
    geom_hline(yintercept=0, linetype = "solid", alpha=0.3) +
    geom_hline(yintercept=c(-3,3), colour = "maroon") +
    theme_bw() +
    theme(strip.text = element_text(size=10, face="bold")) +
    faceting

  qqplt <- ggplot(data=dat, mapping = aes_string(sample="res")) +
    stat_qq_point(alpha=1, size=1) +
    stat_qq_line(alpha=0.75, colour="maroon") + stat_qq_band(bandType = "ts", alpha = 0.2) +
    labs(x = "Theoretical Quantiles",
      y = "Sample Quantiles") +
    geom_hline(yintercept=0, linetype = "dashed", alpha=0.3) +
    geom_vline(xintercept=0, linetype = "dashed", alpha=0.3) +
    theme_bw() +
    theme(strip.text = element_text(size=10, face="bold")) +
    faceting

  print(resfitplt)
  print(qqplt)
  invisible(list(resfitplt = resfitplt, qqplt = qqplt))
}

```

Set asreml options so that a singular variance matrix is allowed and term order is retained

```
asreml.options(ai.sing = TRUE, keep.order = TRUE)
```

Use pstructure from dae to form a marginality matrix for the fixed effects to use in model selection

```
marg.fix <- pstructure(~ Trainings * Motions * Expressiveness, data = plaid.dat)
marg.fix <- marginality(marg.fix)
```

Set up to collect AIC values

```
AIC.tab <- list()
```

Fit model a) with no intertier variances and equal residual variances

```
ma.asr <- asreml(Y ~ Trainings * Motions * Expressiveness,
  random = ~ Raters + Viewings + Expressiveness:Patients,
  aom = TRUE, data = plaid.dat)
```

```
## License check Mon Mar 22 16:42:14 2021
## Model fitted using the gamma parameterization.
## ASReml 4.1.0 Mon Mar 22 16:42:15 2021
##          LogLik      Sigma2      DF      wall      cpu
##  1      -1573.378      4.74180     1176 16:42:15     0.0
##  2      -1556.653      4.62557     1176 16:42:15     0.0
##  3      -1539.037      4.49811     1176 16:42:15     0.0
##  4      -1527.011      4.39774     1176 16:42:15     0.0
##  5      -1520.116      4.32943     1176 16:42:15     0.0
##  6      -1517.532      4.29709     1176 16:42:15     0.0
##  7      -1516.781      4.28270     1176 16:42:15     0.0
##  8      -1516.660      4.27707     1176 16:42:15     0.0
##  9      -1516.654      4.27578     1176 16:42:15     0.0
## 10      -1516.654      4.27571     1176 16:42:15     0.0
```

Prepare to use asremIPlus

```
ma.asrt <- as.asrtests(ma.asr, NULL, NULL, ICLikelihood = "REML",
  label = "No intertier variance and equal residual variances")
```

Calculating denominator DF

```
(ma.asrt)
```

```
##
##
## ##### Summary of the fitting of the variance parameters
##
##          component  std.error  z.ratio bound %ch
## Viewings          1.4906360  0.89398174  1.667412    P    0
## Expressiveness:Patients 14.4822552  8.81950380  1.642071    P    0
## Raters             0.2777965  0.09155513  3.034199    P    0
## units!R            4.2757080  0.18298341 23.366642    P    0
##
##
## ##### Pseudo-anova table for fixed terms
##
##
```

```
## Wald tests for fixed effects.
## Response: Y
##
##              Df denDF  F.inc    Pr
## (Intercept)    1     6 18.910 0.0048
## Trainings      1    72 11.380 0.0012
## Motions        1     6 66.020 0.0002
## Trainings:Motions  1 1092 10.220 0.0014
## Expressiveness  1     6  7.355 0.0350
## Trainings:Expressiveness  1 1092  4.769 0.0292
## Motions:Expressiveness  1     6 37.770 0.0009
## Trainings:Motions:Expressiveness  1 1092  8.441 0.0037
##
##
## ##### Sequence of model investigations
##
## (If a row has NA for p but not denDF, DF and denDF relate to fixed and variance parameter numbers)
##
##              terms DF denDF  p      AIC      BIC      action
## 1 No intertier variance and equal residual variances  0      4 NA 3041.308 3061.588 Starting model
AIC.tab <- rbind(AIC.tab, infoCriteria(ma.asrt$asreml.obj))
```

Extract standardized residuals and fitted values and produce diagnostic plots

```
plotDiagnostics(ma.asrt, plaid.dat)
```

Get and plot predictions

```
ma.choice <- chooseModel(ma.asrt, terms.marginality = marg.fix,
                        denDF="numeric")

## Calculating denominator DF
chosen.terms <- unlist(ma.choice$sig.terms)
submod <- length(chosen.terms) != 1 && chosen.terms != "Trainings:Motions:Expressiveness"
ma.diffs <- predictPlus(ma.asr, classify = "Trainings:Motions:Expressiveness",
                      error.intervals = "half", meanLSD.type = "factor",
                      LSDby = c("Motions", "Expressiveness"),
                      wald = ma.asrt$wald.tab,
                      Vmatrix = TRUE, tables = "none")
```

If the chosen fixed model is a submodel of T:M:E, transform predictions to conform to the chosen model

```
if (submod)
{
  ma.TMEDiffs <- ma.diffs #save full diffs
  submod.ch <- paste("~", paste(chosen.terms, collapse = " + "))
  cat("\n\n#### Chosen model: ", submod.ch, "\n\n")
  ma.diffs <- linTransform(ma.diffs,
                        linear.transformation = as.formula(submod.ch),
                        classify = "Trainings:Motions:Expressiveness",
                        error.intervals = "half", meanLSD.type = "factor",
```

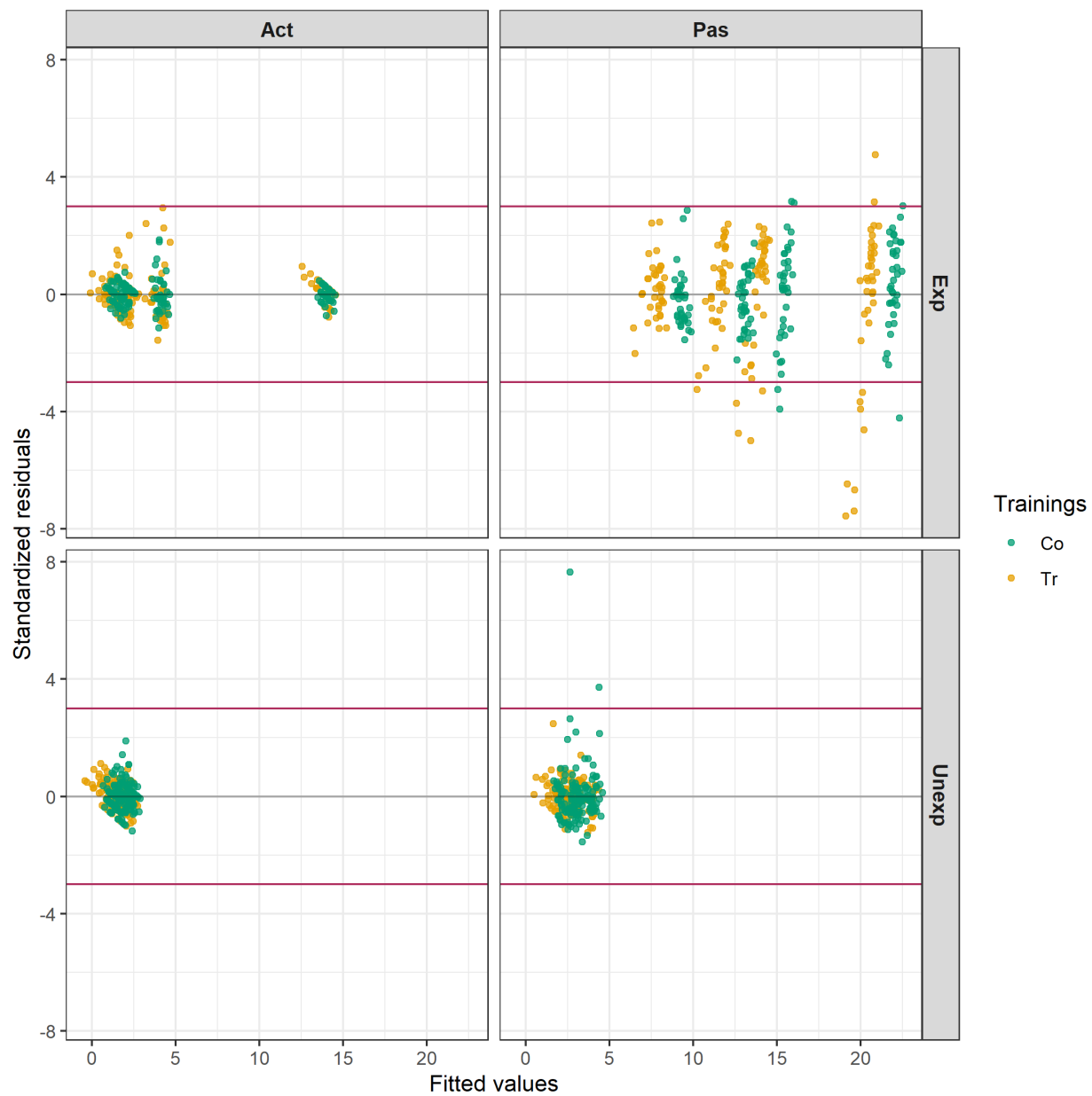


Figure 1: plot of chunk asr_ma

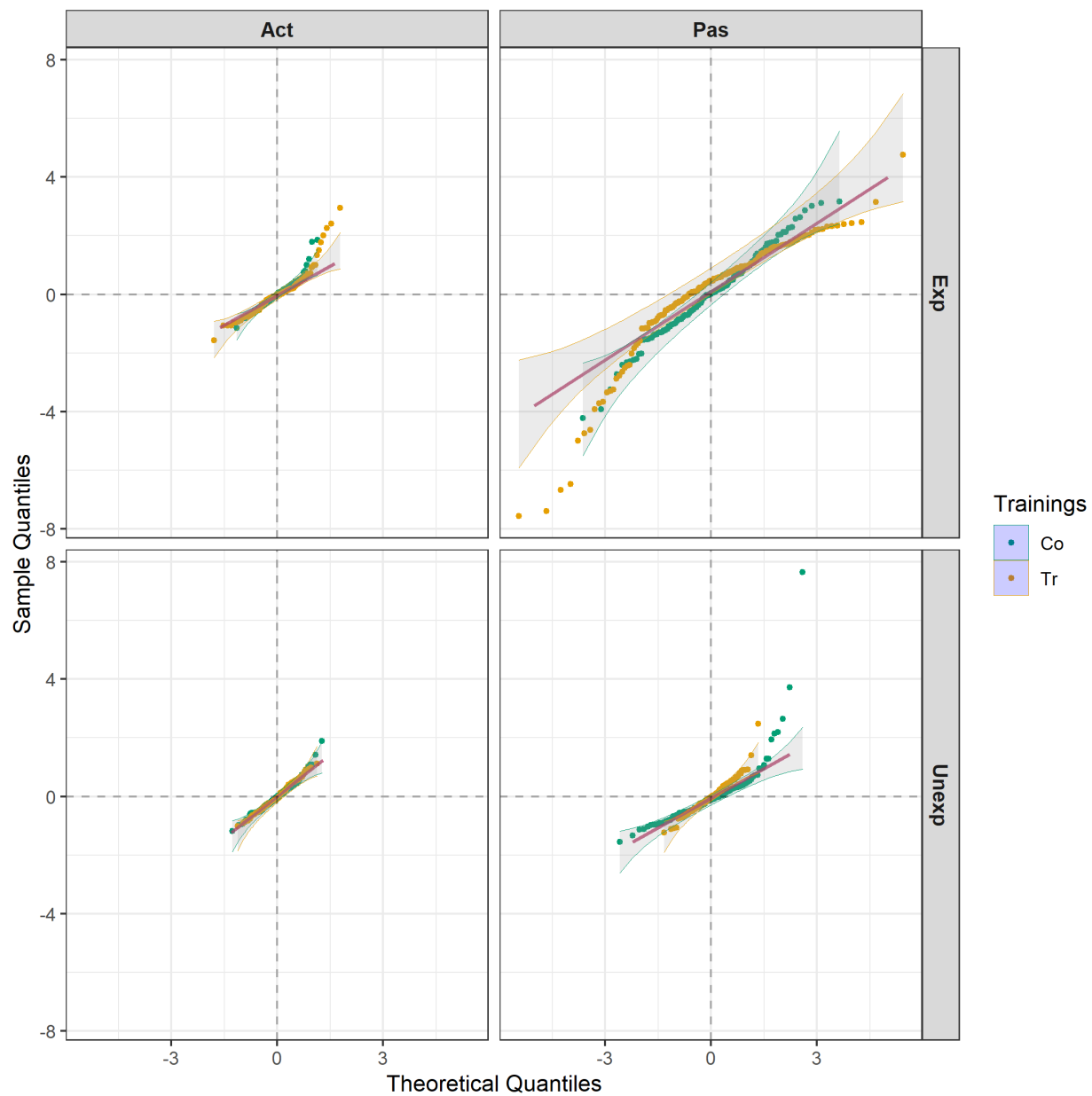


Figure 2: plot of chunk asr_ma

```

LSDby = c("Motions","Expressiveness"),
wald = ma.asrt$wald.tab, tables = "none")
}else
  cat("\n\n#### Chosen model: ~ Trainings:Motions:Expressiveness\n\n")

##
##
## #### Chosen model: ~ Trainings:Motions:Expressiveness
(ma.diffs$predictions)

##
##
## #### Predictions
##
##
## Notes:
## - The predictions are obtained by averaging across the hypertable
##   calculated from model terms constructed solely from factors in
##   the averaging and classify sets.
## - Use 'average' to move ignored factors into the averaging set.
## - The ignored set: Raters,Viewings,Patients
##
##
##   Trainings Motions Expressiveness predicted.value standard.error upper.halfLeastSignificant.limit
## 1      Co      Act      Exp      5.370101      2.007392      5.634801
## 2      Co      Act      Unexp     1.790563      2.007392      2.055263
## 3      Co      Pas      Exp     14.982800      2.007392     15.247499
## 4      Co      Pas      Unexp     3.057149      2.007392      3.321848
## 5      Tr      Act      Exp     5.261824      2.007392      5.526524
## 6      Tr      Act      Unexp     1.508846      2.007392      1.773545
## 7      Tr      Pas      Exp     13.407878      2.007392     13.672578
## 8      Tr      Pas      Unexp     2.705541      2.007392      2.970240
##   lower.halfLeastSignificant.limit est.status
## 1              5.105402   Estimable
## 2              1.525864   Estimable
## 3             14.718100   Estimable
## 4              2.792449   Estimable
## 5              4.997125   Estimable
## 6              1.244146   Estimable
## 7             13.143179   Estimable
## 8              2.440841   Estimable

plotPredictions(ma.diffs$predictions, classify = "Trainings:Motions:Expressiveness",
  y = "predicted.value",
  error.intervals = "half", interval.annotate = FALSE,
  ggplotFuncs = list(facet_grid(cols = vars(Expressiveness, Motions))))

plotPvalues(ma.diffs, factors.per.grid = 2, show.sig = TRUE,
  title = "Initial allocation model, \nequal variances")

```

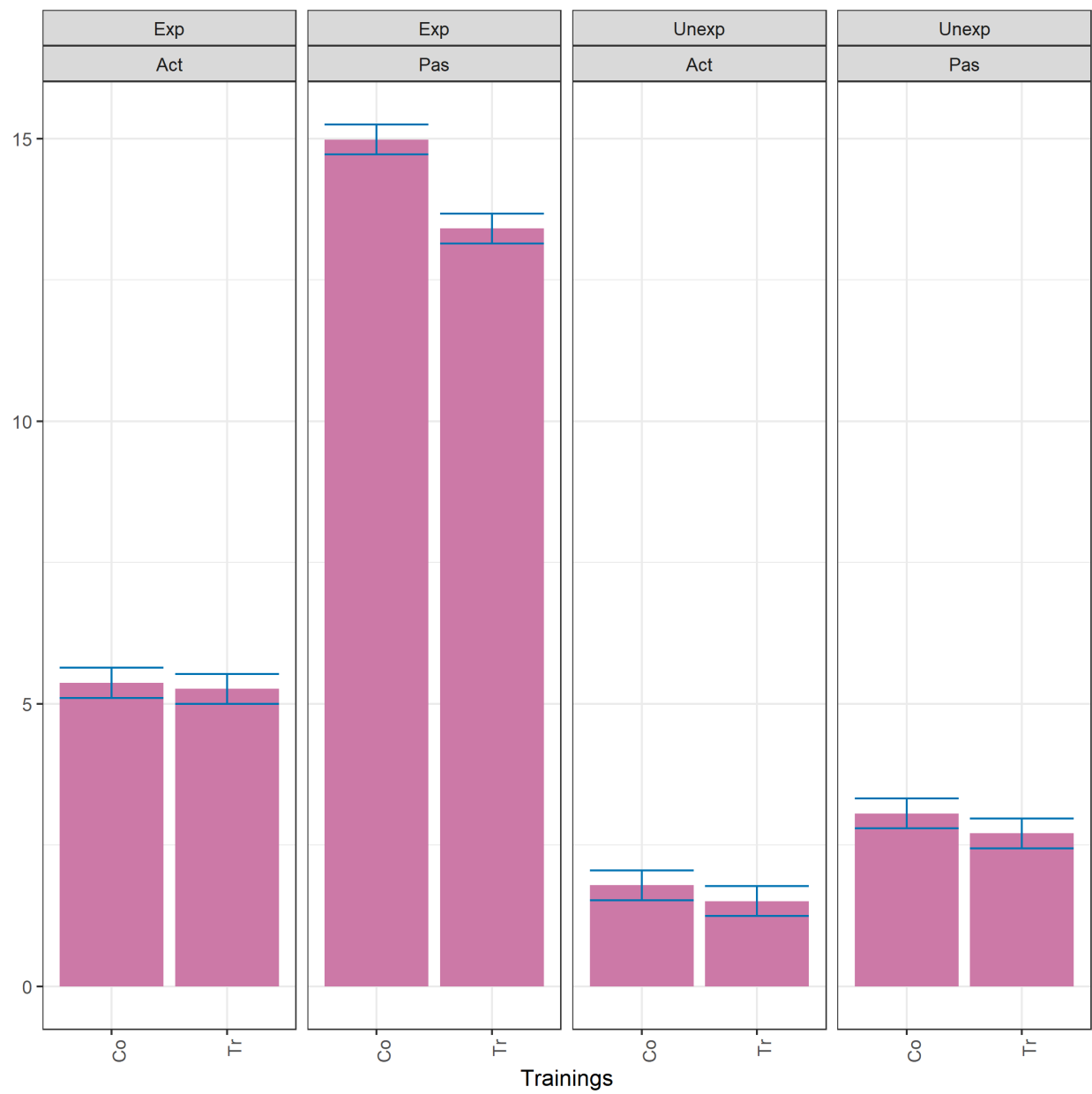


Figure 3: plot of chunk unnamed-chunk-11

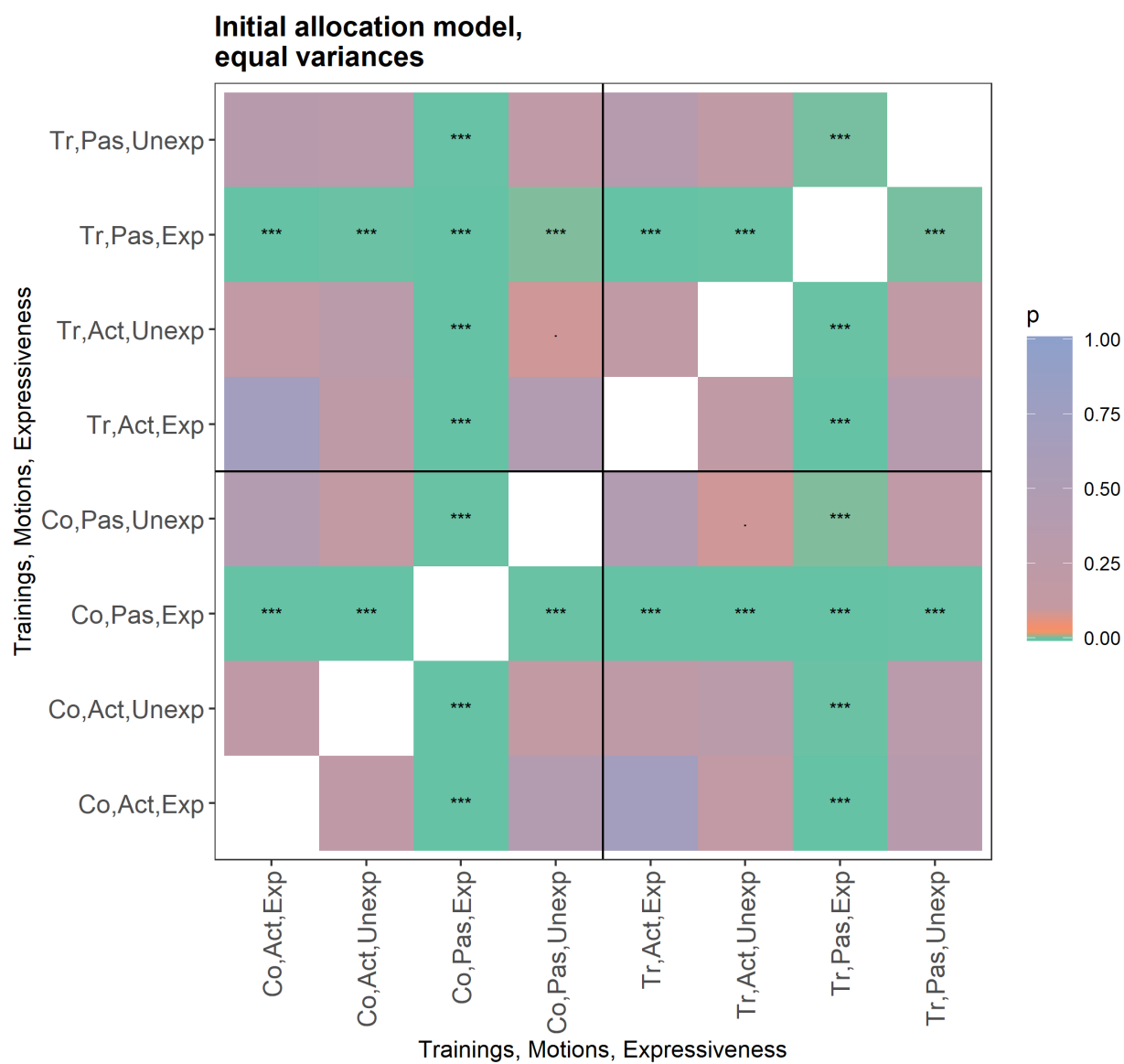


Figure 4: plot of chunk unnamed-chunk-11

Fit model b) no intertier variances and unequal residual variances to see whether heterogeneous variances improves on model a)

```
mb.asrt <- changeModelOnIC(ma.asrt,
  newResidual = "idh(MotionsExpress):WMotionsExpress",
  label = "Try heterogeneous variances",
  ICLikelihood = "REML", which.IC = "AIC")

## Calculating denominator DF
## Warning in infoCriteria.asreml(asreml.obj, ICLikelihood = ic.lik, bound.exclusions = bound.exclusions)
## MotionsExpress:WMotionsExpress!R
## Calculating denominator DF
## Warning in infoCriteria.asreml(new.asrtests.obj$asreml.obj, ICLikelihood = ic.lik, : The following b
## MotionsExpress:WMotionsExpress!R
(mb.asrt)

##
##
## ##### Summary of the fitting of the variance parameters
##
##
## component std.error z.ratio bound %ch
## Viewings 1.42818170 0.86894323 1.6435846 P 0
## Expressiveness:Patients 14.56566580 8.85614901 1.6446952 P 0
## Raters 0.01013582 0.01764610 0.5743944 P 0
## MotionsExpress:WMotionsExpress!R 1.00000000 NA NA F 0
## MotionsExpress:WMotionsExpress!MotionsExpress_Act,Exp 1.22114639 0.10279027 11.8799803 P 0
## MotionsExpress:WMotionsExpress!MotionsExpress_Act,Unexp 0.70625810 0.06064689 11.6454126 P 0
## MotionsExpress:WMotionsExpress!MotionsExpress_Pas,Exp 13.77992460 1.14420141 12.0432684 P 0
## MotionsExpress:WMotionsExpress!MotionsExpress_Pas,Unexp 2.43812281 0.20364308 11.9725294 P 0
##
##
## ##### Pseudo-anova table for fixed terms
##
##
## Wald tests for fixed effects.
## Response: Y
##
## Df denDF F.inc Pr
## (Intercept) 1 6.0 18.070 0.0054
## Trainings 1 71.9 13.650 0.0004
## Motions 1 5.9 65.700 0.0002
## Expressiveness 1 6.0 7.088 0.0374
## Trainings:Motions 1 544.6 2.958 0.0860
## Trainings:Expressiveness 1 624.1 0.027 0.8695
## Motions:Expressiveness 1 5.9 39.270 0.0008
## Trainings:Motions:Expressiveness 1 481.0 7.956 0.0050
##
##
## ##### Sequence of model investigations
##
## (If a row has NA for p but not denDF, DF and denDF relate to fixed and variance parameter numbers)
##
```

```
##
## 1 No intertier variance and equal residual variances 0 4 NA 3041.3081 3061.5876 Starting model
## 2 Try heterogeneous variances 0 3 NA -749.5992 -734.3896 Swapped
```

```
AIC.tab <- rbind(AIC.tab, infoCriteria(mb.asrt$asreml.obj))
```

```
## Warning in infoCriteria.asreml(mb.asrt$asreml.obj): The following bound terms were discounted:
## MotionsExpress:WMotionsExpress!R
```

Extract standardized residuals and fitted values and produce diagnostic plots

```
plotDiagnostics(mb.asrt, plaid.dat)
```

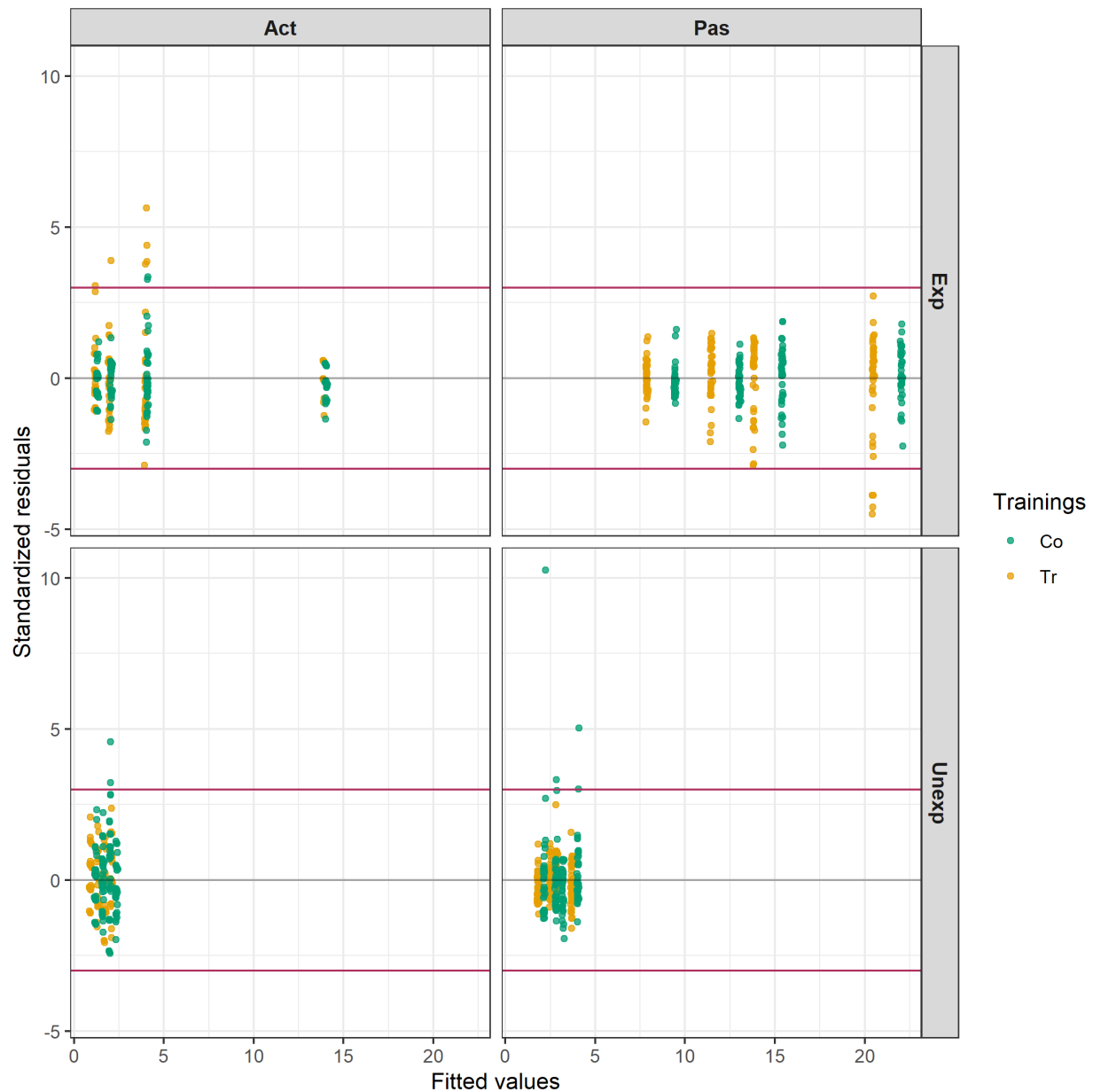


Figure 5: plot of chunk asr_mb

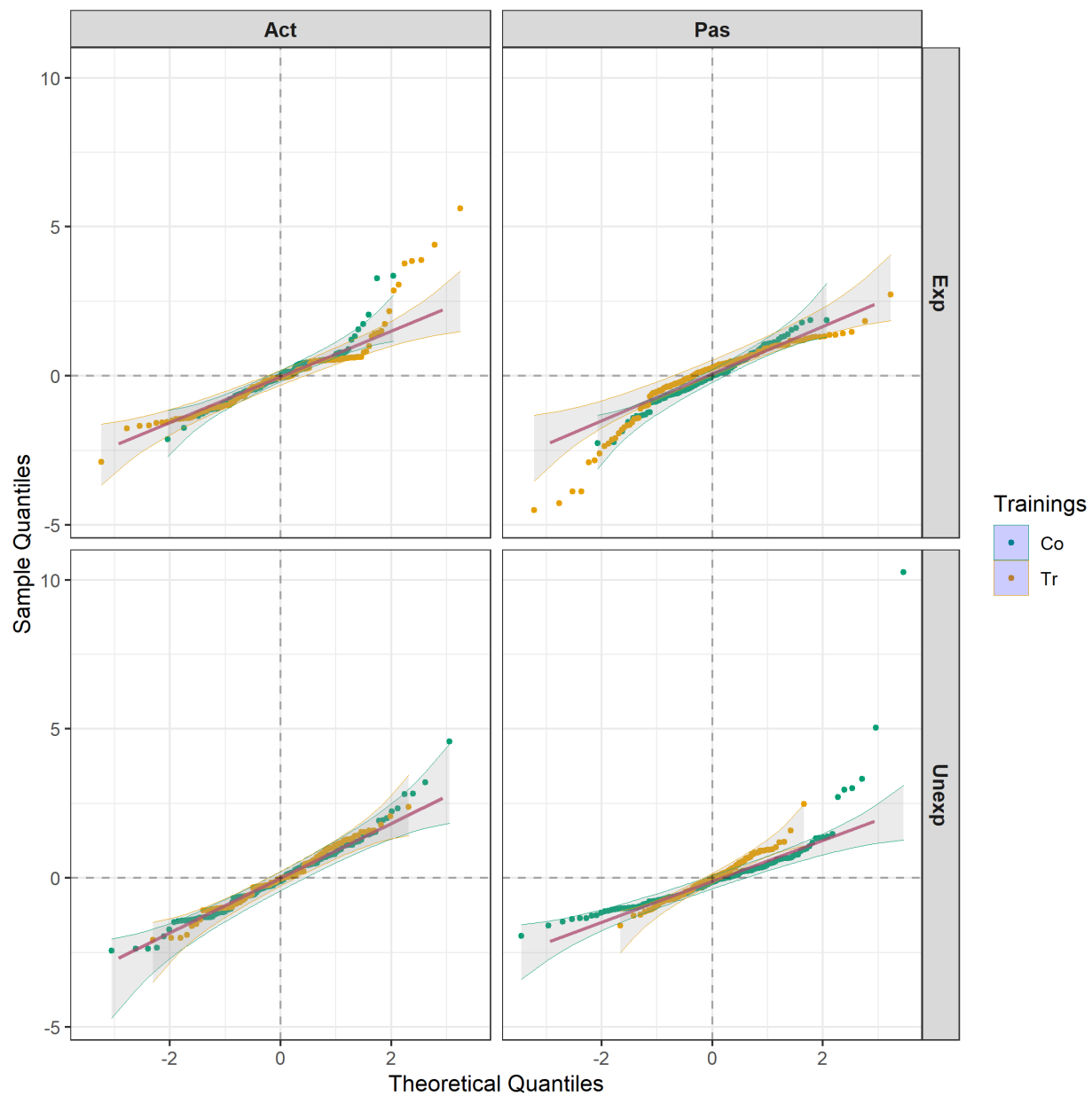


Figure 6: plot of chunk asr_mb

Get and plot predictions

```
mb.choice <- chooseModel(mb.asrt, terms.marginality = marg.fix,
                        denDF="numeric")

## Calculating denominator DF
chosen.terms <- unlist(mb.choice$sig.terms)
submod <- length(chosen.terms) != 1 && chosen.terms != "Trainings:Motions:Expressiveness"
mb.asr <- mb.asrt$asreml.obj
mb.diffs <- predictPlus(mb.asr, classify = "Trainings:Motions:Expressiveness",
                        error.intervals = "half", meanLSD.type = "factor",
                        LSDby = c("Motions","Expressiveness"), Vmatrix = TRUE,
                        wald = mb.asrt$wald.tab, tables = "none")
```

If the chosen fixed model is a submodel of T:M:E, transform predictions to conform to the chosen model

```
if (submod)
{
  mb.TMediffs <- mb.diffs #save full diffs
  submod.ch <- paste("~", paste(chosen.terms, collapse = " + "))
  cat("\n\n#### Chosen model: ", submod.ch, "\n\n")
  mb.diffs <- linTransform(mb.diffs,
                          linear.transformation = as.formula(submod.ch),
                          classify = "Trainings:Motions:Expressiveness",
                          error.intervals = "half", meanLSD.type = "factor",
                          LSDby = c("Motions","Expressiveness"),
                          wald = mb.asrt$wald.tab, tables = "none")
}else
  cat("\n\n#### Chosen model: ~ Trainings:Motions:Expressiveness\n\n")
```

```
##
##
## ##### Chosen model: ~ Trainings:Motions:Expressiveness
(mb.diffs$predictions)
```

```
##
##
## ##### Predictions
##
## Notes:
## - The predictions are obtained by averaging across the hypertextable
##   calculated from model terms constructed solely from factors in
##   the averaging and classify sets.
## - Use 'average' to move ignored factors into the averaging set.
## - The ignored set: Raters,Viewings,Patients
##
##
##   Trainings Motions Expressiveness predicted.value standard.error upper.halfLeastSignificant.limit
## 1      Co      Act      Exp      5.370101      2.001746      5.498385
## 2      Co      Act      Unexp     1.790563      2.000877      1.889259
## 3      Co      Pas      Exp     14.982800      2.022830     15.407378
```

```
## 4      Co      Pas      Unexp      3.057149      2.003799      3.236955
## 5      Tr      Act      Exp      5.261824      2.001746      5.390108
## 6      Tr      Act      Unexp      1.508846      2.000877      1.607542
## 7      Tr      Pas      Exp      13.407878      2.022830      13.832457
## 8      Tr      Pas      Unexp      2.705541      2.003799      2.885347
## lower.halfLeastSignificant.limit est.status
## 1              5.241817 Estimable
## 2              1.691867 Estimable
## 3             14.558221 Estimable
## 4              2.877342 Estimable
## 5              5.133540 Estimable
## 6              1.410150 Estimable
## 7             12.983300 Estimable
## 8              2.525734 Estimable

plotPredictions(mb.diffs$predictions, classify = "Trainings:Motions:Expressiveness",
  y = "predicted.value",
  error.intervals = "half", interval.annotate = FALSE,
  ggplotFuncs = list(facet_grid(cols = vars(Expressiveness, Motions))))

plotPvalues(mb.diffs, factors.per.grid = 2, show.sig = TRUE,
  title = "Initial allocation model, \nunequal variances")
```

Fit the full prior allocation model, except for identity terms, to show that it is nonsingular

```
mfull.asr <- asreml(Y ~ Trainings * Motions * Expressiveness,
  random = ~ Raters + Viewings +
    Expressiveness:Patients/(Occasions + Trainings*Motions) +
    Raters:(Motions*Expressiveness) + Raters:Expressiveness:Patients,
  aom = TRUE, data = plaid.dat)

## Model fitted using the gamma parameterization.
## ASReML 4.1.0 Mon Mar 22 16:42:34 2021
##          LogLik      Sigma2      DF      wall      cpu
## 1      -1521.361      3.35511     1176 16:42:34     0.0 (1 restrained)
## 2 singularities in the Average Information matrix.

## Warning in asreml(Y ~ Trainings * Motions * Expressiveness, random = ~Raters + : Singularities in
## the average information matrix.

## 2      -1503.359      3.41238     1176 16:42:34     0.0 (2 restrained)

## Warning in asreml(Y ~ Trainings * Motions * Expressiveness, random = ~Raters + : Singularities in
## the average information matrix.

## 3      -1487.165      3.42477     1176 16:42:34     0.0 (2 restrained)

## Warning in asreml(Y ~ Trainings * Motions * Expressiveness, random = ~Raters + : Singularities in
## the average information matrix.

## 4      -1479.065      3.34427     1176 16:42:34     0.0 (2 restrained)

## Warning in asreml(Y ~ Trainings * Motions * Expressiveness, random = ~Raters + : Singularities in
## the average information matrix.

## 5      -1474.862      3.29448     1176 16:42:34     0.0 (2 restrained)

## Warning in asreml(Y ~ Trainings * Motions * Expressiveness, random = ~Raters + : Singularities in
```

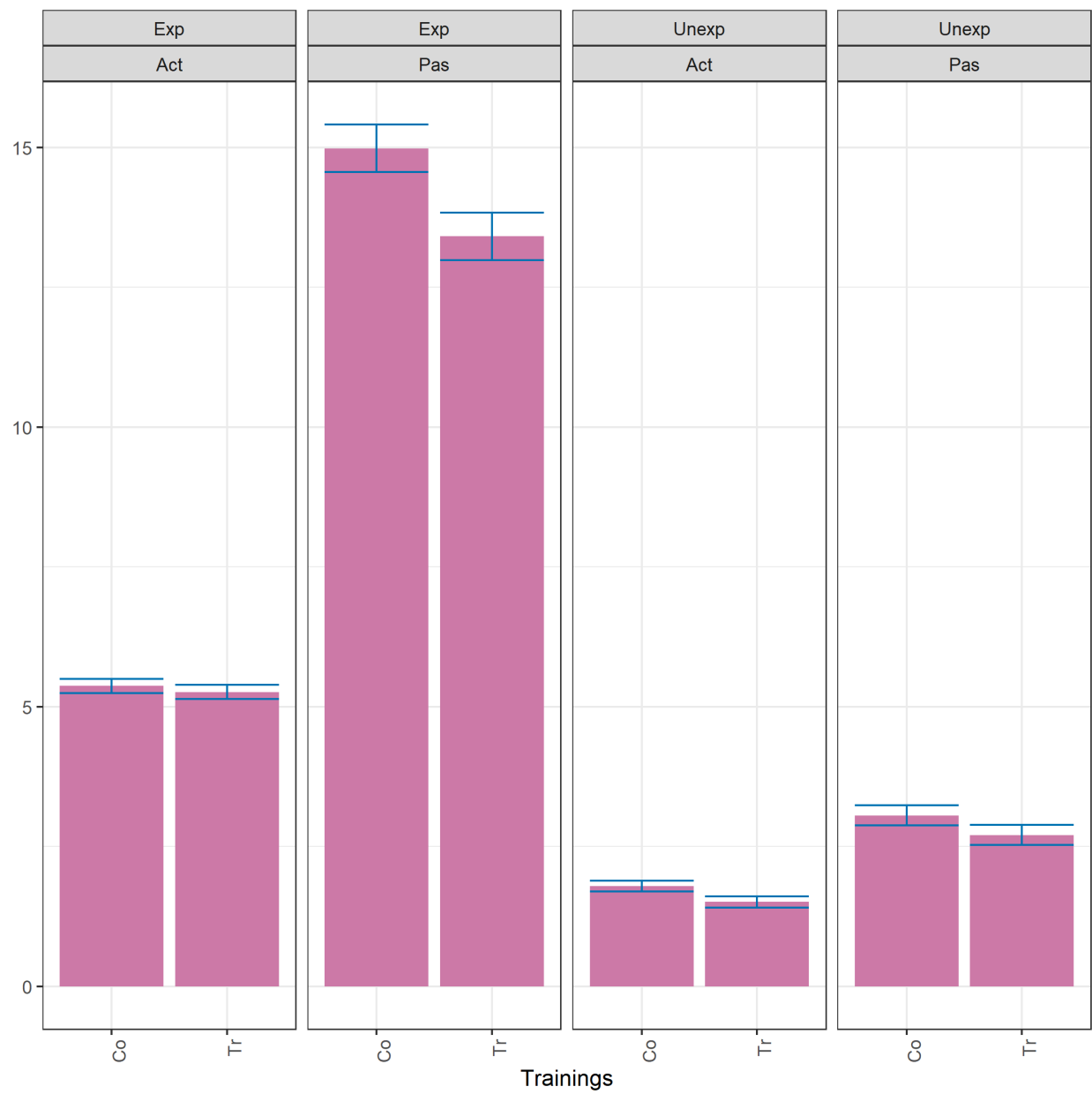


Figure 7: plot of chunk unnamed-chunk-13

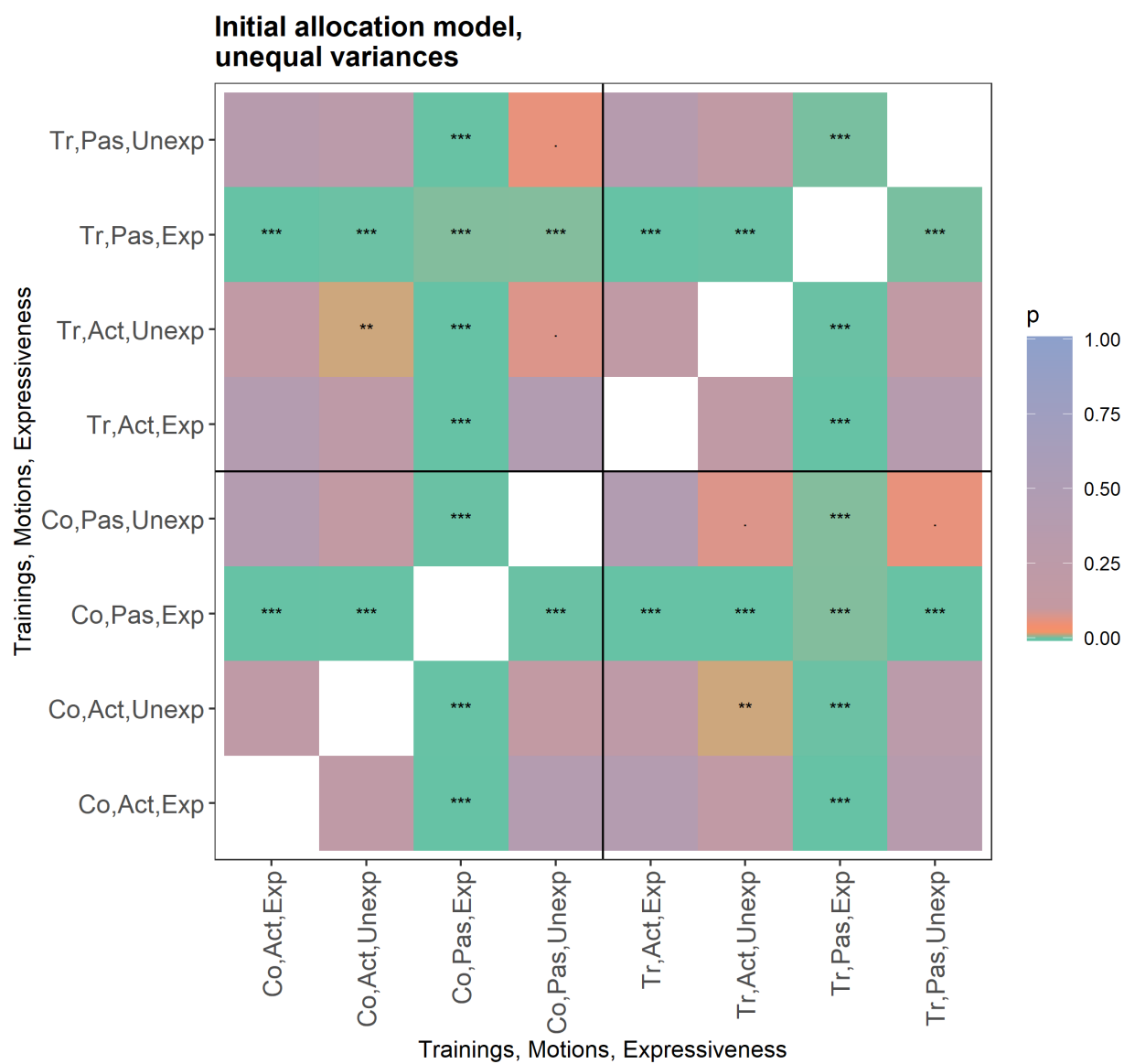


Figure 8: plot of chunk unnamed-chunk-13

```
## the average information matrix.
## 6      -1473.456      3.27809   1176 16:42:34    0.0 (1 restrained)
## Warning in asreml(Y ~ Trainings * Motions * Expressiveness, random = ~Raters + : Singularities in
## the average information matrix.
## 7      -1473.139      3.27117   1176 16:42:34    0.0
## Warning in asreml(Y ~ Trainings * Motions * Expressiveness, random = ~Raters + : Singularities in
## the average information matrix.
## 8      -1473.109      3.26898   1176 16:42:34    0.0
## Warning in asreml(Y ~ Trainings * Motions * Expressiveness, random = ~Raters + : Singularities in
## the average information matrix.
## 9      -1473.109      3.26870   1176 16:42:34    0.0
## Warning in asreml(Y ~ Trainings * Motions * Expressiveness, random = ~Raters + : Singularities in
## the average information matrix.
## 10     -1473.109      3.26870   1176 16:42:35    0.2
## Warning in asreml(Y ~ Trainings * Motions * Expressiveness, random = ~Raters + : Singularities in
## the average information matrix.
```

```
summary(mfull.asr)$varcomp
```

##	component	std.error	z.ratio	bound	%ch
## Viewings	7.609710e-01	0.89785598	0.847542398	P	0
## Expressiveness:Patients	1.448199e+01	8.81982538	1.641981274	P	0
## Expressiveness:Patients:Occasions	3.268700e-01	0.01572654	20.784609706	S	0
## Expressiveness:Patients:Trainings	5.324598e-04	0.10938702	0.004867669	P	0
## Expressiveness:Patients:Motions	3.268700e-01	0.01572654	20.784609706	S	0
## Expressiveness:Patients:Trainings:Motions	1.790661e-01	0.15444738	1.159399042	P	0
## Raters	1.280818e-06	NA	NA	B	0
## Raters:Motions	8.099187e-02	0.17004892	0.476285690	P	0
## Raters:Expressiveness	7.684186e-02	0.16999557	0.452022694	P	0
## Raters:Motions:Expressiveness	1.068292e+00	0.28250325	3.781522160	P	0
## Raters:Expressiveness:Patients	1.665475e-07	NA	NA	B	NA
## units!R	3.268700e+00	0.15726541	20.784609706	P	0

Change to model c), the nonsingular prior allocation model with intertier variances and equal residual variances

Drop singular terms OPE and MPE, identified using the EMS (and confirmed by asreml)

```
mc.asr <- asreml(Y ~ Trainings * Motions * Expressiveness,
  random = ~ Raters+ Viewings +
    Expressiveness:Patients/Trainings/Motions +
    Raters:(Occasions*Expressiveness) + Raters:Expressiveness:Patients,
  aom = TRUE, data = plaid.dat)
```

```
## Model fitted using the gamma parameterization.
## ASReml 4.1.0 Mon Mar 22 16:42:35 2021
##      LogLik      Sigma2      DF      wall      cpu
## 1     -1539.710      3.48193   1176 16:42:35    0.0 (1 restrained)
## 2     -1518.288      3.52565   1176 16:42:35    0.0 (2 restrained)
## 3     -1495.651      3.48752   1176 16:42:35    0.0 (2 restrained)
```

```
## 4      -1483.282      3.37667    1176 16:42:35    0.0 (2 restrained)
## 5      -1476.498      3.30912    1176 16:42:35    0.0 (2 restrained)
## 6      -1473.964      3.28475    1176 16:42:35    0.0 (1 restrained)
## 7      -1473.231      3.27393    1176 16:42:35    0.0
## 8      -1473.114      3.26970    1176 16:42:35    0.0
## 9      -1473.109      3.26875    1176 16:42:35    0.0
## 10     -1473.109      3.26870    1176 16:42:35    0.1
```

```
summary(mc.asr)$varcomp
```

```
##                                component std.error      z.ratio bound %ch
## Viewings                      1.414711e+00 0.8973050  1.57662240    P    0
## Expressiveness:Patients        1.448199e+01 8.8196925  1.64200609    P    0
## Expressiveness:Patients:Trainings 5.324600e-04 0.1093870  0.00486767    P    0
## Expressiveness:Patients:Trainings:Motions 1.790662e-01 0.1544474  1.15939901    P    0
## Raters                        1.131598e-06      NA      NA      B    0
## Raters:Occasions              8.099201e-02 0.1700489  0.47628649    P    0
## Raters:Expressiveness          7.684200e-02 0.1699956  0.45202350    P    0
## Raters:Occasions:Expressiveness 1.068292e+00 0.2825032  3.78152187    P    0
## Raters:Expressiveness:Patients  1.603566e-07      NA      NA      B   NA
## units!R                       3.268700e+00 0.1572654 20.78460917    P    0
```

```
printFormulae(mc.asr)
```

```
##
##
## ##### Formulae from asreml object
##
## fixed: Y ~ Trainings * Motions * Expressiveness
## random: ~ Raters + Viewings + Expressiveness:Patients/Trainings/Motions + Raters:(Occasions * Expressiveness)
## residual: NULL
```

Allow negative variance components so equivalent to ANOVA

```
terms <- names(mc.asr$vpparameters)
terms <- terms[-length(terms)]
mc.neg.asr <- setvarianceterms(mc.asr$call, terms = terms, bounds = "U")
```

```
## Model fitted using the gamma parameterization.
## ASReml 4.1.0 Mon Mar 22 16:42:35 2021
##                                LogLik      Sigma2      DF      wall      cpu
## 1      -1539.710      3.48193    1176 16:42:35    0.0 (1 restrained)
## 2      -1518.288      3.52565    1176 16:42:35    0.0 (2 restrained)
## 3      -1495.651      3.48752    1176 16:42:35    0.0 (2 restrained)
## 4      -1483.282      3.37667    1176 16:42:35    0.0 (2 restrained)
## 5      -1476.495      3.30977    1176 16:42:35    0.0 (2 restrained)
## 6      -1473.939      3.29094    1176 16:42:35    0.0 (2 restrained)
## 7      -1473.104      3.32005    1176 16:42:35    0.0
## 8      -1472.959      3.35597    1176 16:42:35    0.0
## 9      -1472.954      3.35462    1176 16:42:35    0.0
## 10     -1472.954      3.35457    1176 16:42:35    0.1
```

Prepare to use asremlPlus

```
mc.neg.asrt <- as.asrtests(mc.neg.asr, NULL, NULL, IClikelihood = "REML",
                          label = "Intertier interactions and unconstrained components")
```

```
## Calculating denominator DF
```

```
(mc.neg.asrt)
```

```
##
```

```
##
```

```
## ##### Summary of the fitting of the variance parameters
```

```
##
```

	component	std.error	z.ratio	bound	%ch
## Viewings	1.414711088	0.8973049	1.57662241	U	0
## Expressiveness:Patients	14.481988677	8.8197254	1.64199994	U	0
## Expressiveness:Patients:Trainings	0.002853368	0.1094696	0.02606538	U	0
## Expressiveness:Patients:Trainings:Motions	0.176745246	0.1545121	1.14389266	U	0
## Raters	-0.018172606	0.1705225	-0.10657012	U	0
## Raters:Occasions	0.097802882	0.2321986	0.42120354	U	0
## Raters:Expressiveness	0.115047665	0.2349919	0.48958140	U	0
## Raters:Occasions:Expressiveness	1.031395356	0.3168537	3.25511533	U	0
## Raters:Expressiveness:Patients	-0.085873419	0.1573197	-0.54585297	U	0
## units!R	3.354573808	0.2282498	14.69693817	P	0

```
##
```

```
##
```

```
## ##### Pseudo-anova table for fixed terms
```

```
##
```

```
##
```

```
## Wald tests for fixed effects.
```

```
## Response: Y
```

```
##
```

	Df	denDF	F.inc	Pr
## (Intercept)	1	6.0	18.900	0.0048
## Trainings	1	13.7	6.417	0.0242
## Motions	1	6.5	63.310	0.0001
## Trainings:Motions	1	12.7	2.951	0.1101
## Expressiveness	1	6.0	7.338	0.0352
## Trainings:Expressiveness	1	12.9	1.361	0.2645
## Motions:Expressiveness	1	6.4	36.460	0.0007
## Trainings:Motions:Expressiveness	1	11.5	2.574	0.1357

```
##
```

```
##
```

```
## ##### Sequence of model investigations
```

```
##
```

```
## (If a row has NA for p but not denDF, DF and denDF relate to fixed and variance parameter numbers)
```

```
##
```

	terms	DF	denDF	p	AIC	BIC	action
## 1	Intertier interactions and unconstrained components	0	10	NA	2965.908	3016.607	Starting model

```
printFormulae(mc.neg.asr)
```

```
##
```

```
##
```

```
## ##### Formulae from asreml object
```

```
##
```

```
## fixed: Y ~ Trainings * Motions * Expressiveness
```

```
## random: ~ Raters + Viewings + Expressiveness:Patients/Trainings/Motions + Raters:(Occasions * Expressiveness)
## residual: NULL
AIC.tab <- rbind(AIC.tab, infoCriteria(mc.neg.asrt$asreml.obj))
```

Extract standardized residuals and fitted values and produce diagnostic plots

```
plotDiagnostics(mc.neg.asrt, plaid.dat)
```

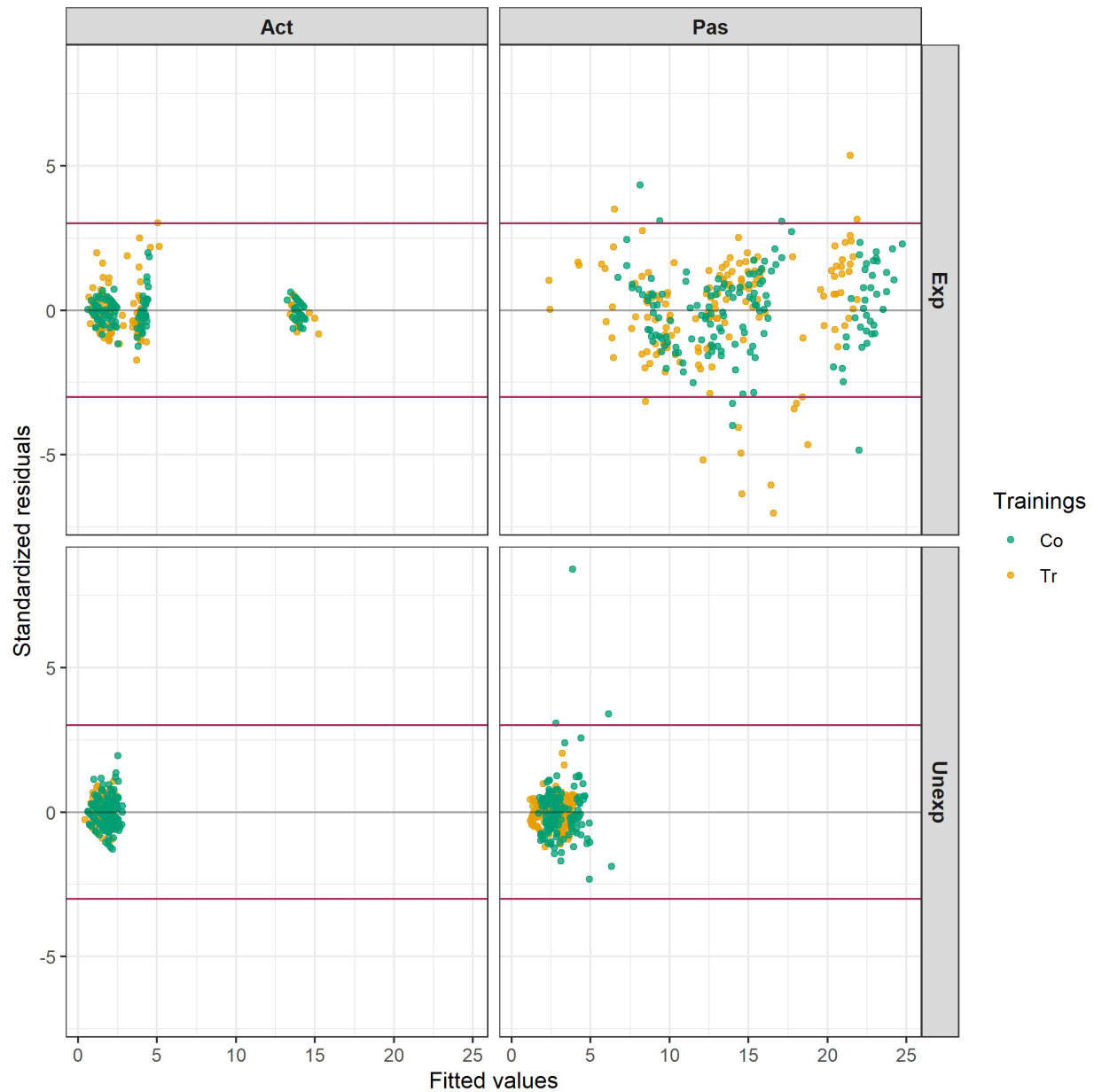


Figure 9: plot of chunk asr_mc

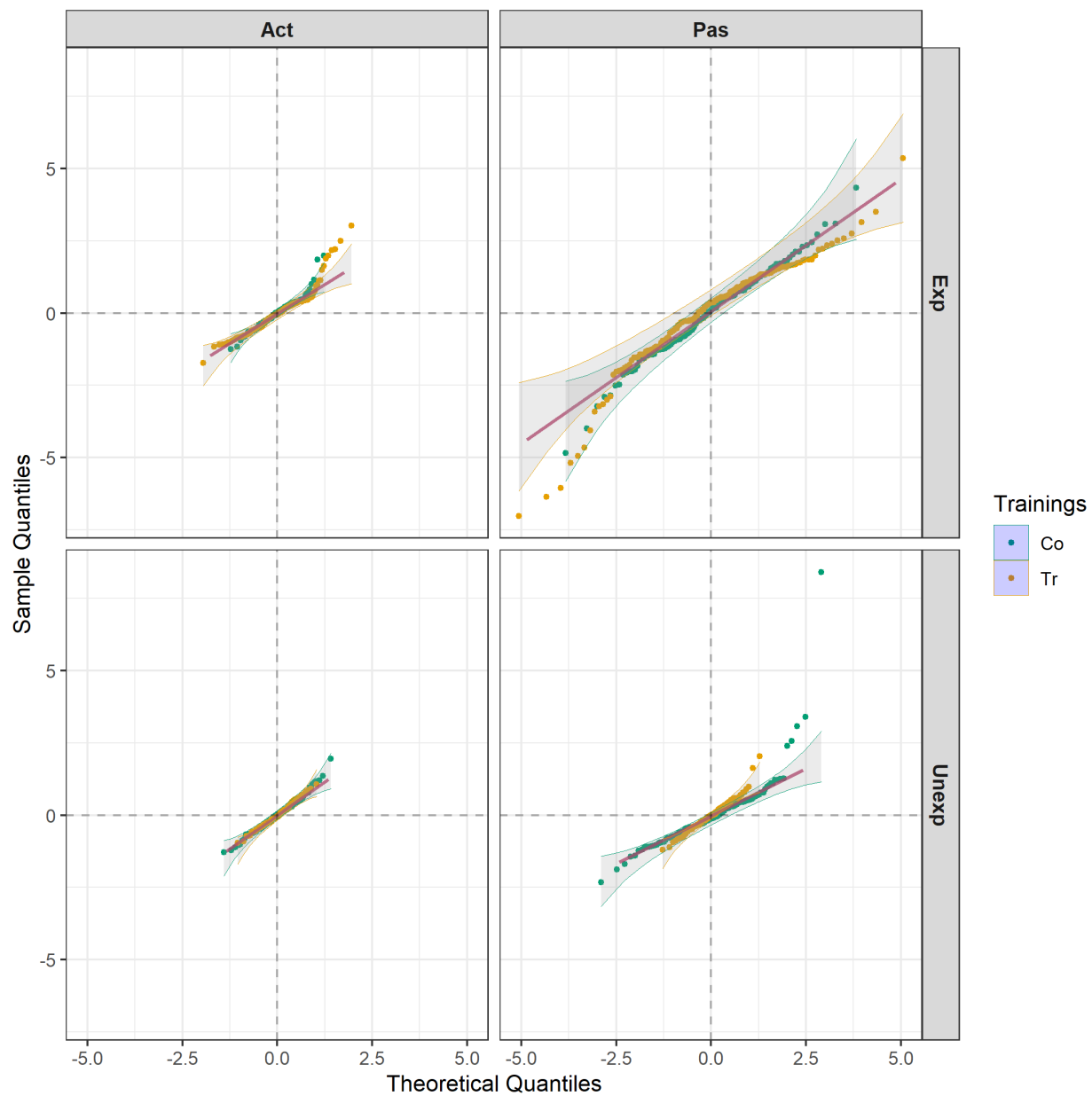


Figure 10: plot of chunk asr_mc

Get and plot predictions based on the nonsingular models with intertier interactions

```
mc.neg.choice <- chooseModel(mc.neg.asrt, terms.marginality = marg.fix,
                             denDF="numeric")

## Calculating denominator DF
## Calculating denominator DF
## Calculating denominator DF
## Calculating denominator DF
## Calculating denominator DF

chosen.terms <- unlist(mc.neg.choice$sig.terms)
submod <- length(chosen.terms) != 1 && chosen.terms != "Trainings:Motions:Expressiveness"
mc.neg.diffs <- predictPlus(mc.neg.asr, classify = "Trainings:Motions:Expressiveness",
                           error.intervals = "half", meanLSD.type = "factor",
                           LSDby = c("Motions","Expressiveness"), Vmatrix = TRUE,
                           wald = mc.neg.asrt$wald.tab, tables = "none")
```

If the chosen fixed model is a submodel of T:M:E, transform predictions to conform to the chosen model

```
if (submod)
{
  mc.neg.TMEDiffs <- mc.neg.diffs #save full diffs
  submod.ch <- paste("~", paste(chosen.terms, collapse = " + "))
  cat("\n\n#### Chosen model: ", submod.ch, "\n\n")
  mc.neg.diffs <- linTransform(mc.neg.diffs,
                              linear.transformation = as.formula(submod.ch),
                              classify = "Trainings:Motions:Expressiveness",
                              error.intervals = "half", meanLSD.type = "factor",
                              LSDby = c("Motions","Expressiveness"),
                              wald = mc.neg.asrt$wald.tab, tables = "none")
}else
  cat("\n\n#### Chosen model: ~ Trainings:Motions:Expressiveness\n\n")
```

```
##
##
## ##### Chosen model: ~ Motions:Expressiveness + Trainings
(mc.neg.diffs$predictions)
```

```
##
##
## ##### Predictions
##
## Notes:
## The original predictions, obtained as described below, have
## been linearly transformed to form estimated marginal means.
## - The predictions are obtained by averaging across the hypertable
##   calculated from model terms constructed solely from factors in
##   the averaging and classify sets.
## - Use 'average' to move ignored factors into the averaging set.
## - The ignored set: Raters,Viewings,Patients,Occasions
##
```

```
##
##   Trainings Motions Expressiveness predicted.value standard.error upper.halfLeastSignificant.limit
## 1      Co      Act      Exp      5.605528      2.009304      5.855796
## 2      Co      Act      Unexp     1.939270      2.009304      2.189537
## 3      Co      Pas      Exp     14.484904      2.009304     14.735172
## 4      Co      Pas      Unexp     3.170910      2.009304      3.421177
## 5      Tr      Act      Exp     5.026397      2.009304      5.276665
## 6      Tr      Act      Unexp     1.360139      2.009304      1.610406
## 7      Tr      Pas      Exp     13.905774      2.009304     14.156041
## 8      Tr      Pas      Unexp     2.591779      2.009304      2.842046
##   lower.halfLeastSignificant.limit est.status
## 1                5.355261   Estimable
## 2                1.689003   Estimable
## 3               14.234637   Estimable
## 4                2.920643   Estimable
## 5                4.776130   Estimable
## 6                1.109872   Estimable
## 7               13.655506   Estimable
## 8                2.341512   Estimable

plotPredictions(mc.neg.diffs$predictions, classify = "Trainings:Motions:Expressiveness",
  y = "predicted.value",
  error.intervals = "half", interval.annotate = FALSE,
  ggplotFuncs = list(facet_grid(cols = vars(Expressiveness, Motions))))

plotPvalues(mc.neg.diffs, factors.per.grid = 2, show.sig = TRUE,
  title = "Prior allocation model, \nequal variances")
```

Fit model d) with intertier variances and unequal residual variances

```
md.asr <- asreml(Y ~ Trainings * Motions * Expressiveness,
  random = ~ Raters+ Viewings +
    Expressiveness:Patients/Trainings/Motions +
    Raters:(Occasions*Expressiveness) + Raters:Expressiveness:Patients,
  residual = ~ idh(MotionsExpress):WMotionsExpress,
  maxit = 50, aom = TRUE,
  data = plaid.dat)
```

```
## Model fitted using the sigma parameterization.
## ASReml 4.1.0 Mon Mar 22 16:42:46 2021
##      LogLik      Sigma2      DF      wall      cpu
## 1    -1659.584        1.0    1176 16:42:47    0.0 (8 restrained)
## 2    -1194.037        1.0    1176 16:42:47    0.0 (4 restrained)
## 3    -1141.100        1.0    1176 16:42:47    0.0 (1 restrained)
## 4    -1134.016        1.0    1176 16:42:47    0.0 (3 restrained)
## 5    -1132.870        1.0    1176 16:42:47    0.0 (2 restrained)
## 6    -1132.700        1.0    1176 16:42:47    0.0 (4 restrained)
## 7    -1133.077        1.0    1176 16:42:47    0.0 (3 restrained)
## 8    -1132.688        1.0    1176 16:42:47    0.0 (2 restrained)
## 9    -1132.622        1.0    1176 16:42:47    0.0 (3 restrained)
## 10   -1133.282        1.0    1176 16:42:47    0.0 (2 restrained)
## 11   -1132.750        1.0    1176 16:42:47    0.0
## 12   -1132.629        1.0    1176 16:42:47    0.0
## 13   -1132.621        1.0    1176 16:42:47    0.0
```

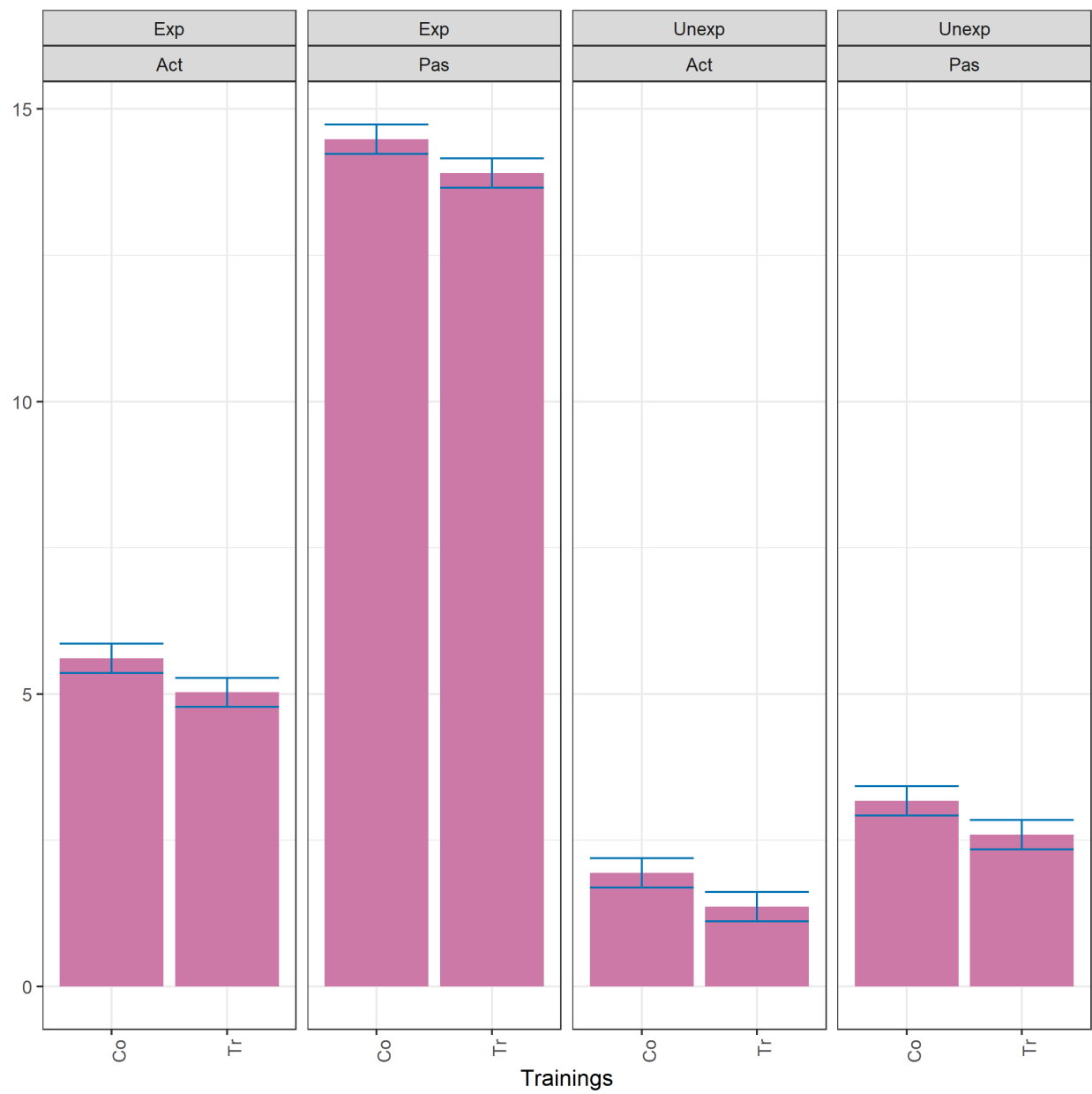


Figure 11: plot of chunk unnamed-chunk-18

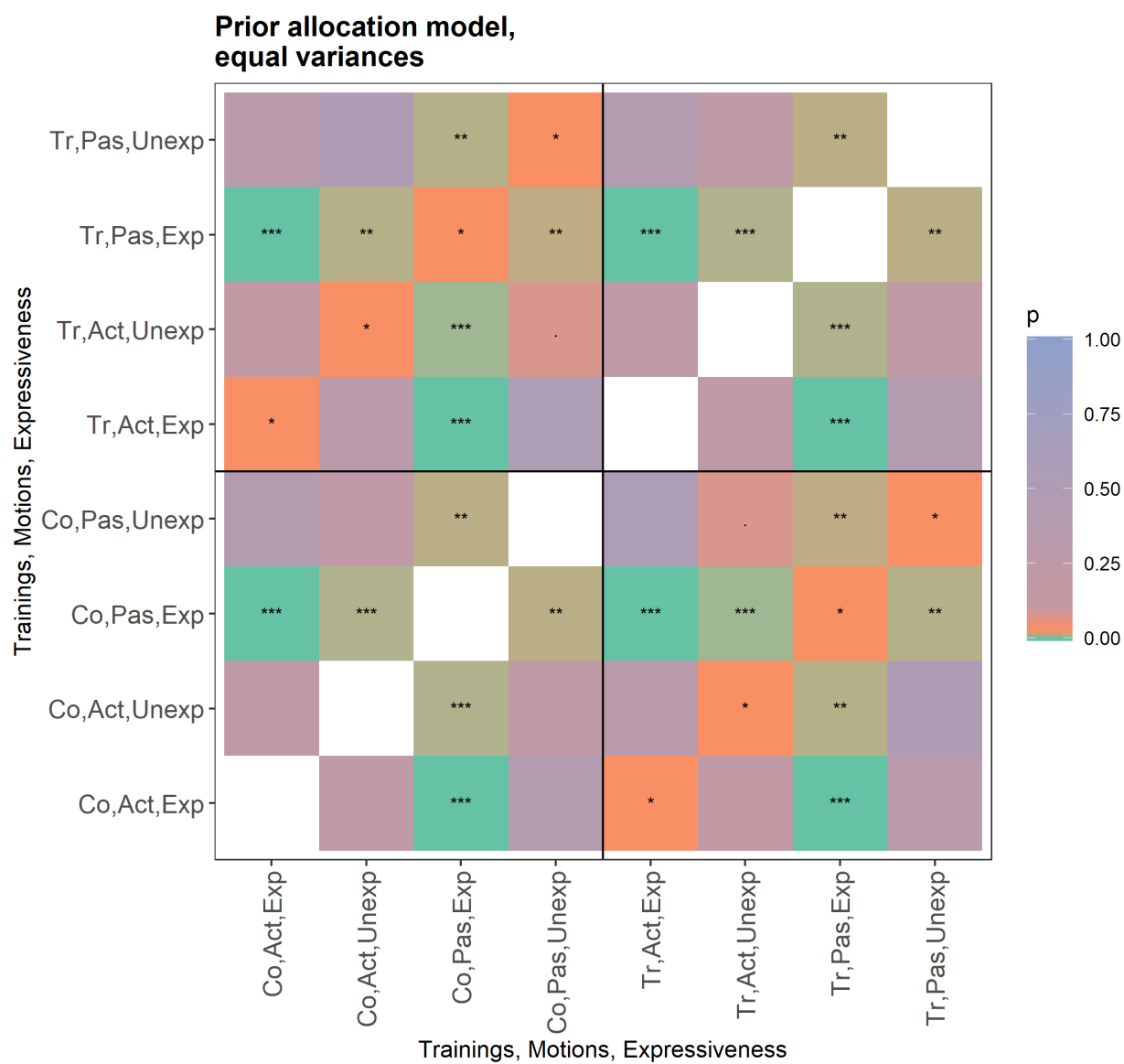


Figure 12: plot of chunk unnamed-chunk-18

```
## 14      -1132.621          1.0   1176 16:42:47    0.1
```

Prepare to use asremlPlus

```
md.asrt <- as.asrtests(md.asr, NULL, NULL, ICLikelihood = "REML",
                      label = "Intertier variance and unequal residual variances")
```

```
## Warning in infoCriteria.asreml(asreml.obj, ICLikelihood = ic.lik, bound.exclusions = bound.exclusion
```

```
## Raters:Occasions, Raters:Expressiveness, Raters:Occasions:Expressiveness, MotionsExpress:WMotionsExp
```

```
## Calculating denominator DF
```

```
(md.asrt)
```

```
##
```

```
##
```

```
## ##### Summary of the fitting of the variance parameters
```

```
##
```

	component	std.error	z.ratio	bound
## Viewings	1.400908e+00	0.86897501	1.6121380	P
## Expressiveness:Patients	1.456116e+01	8.85623965	1.6441697	P
## Expressiveness:Patients:Trainings	5.040427e-03	0.04463139	0.1129346	P
## Expressiveness:Patients:Trainings:Motions	5.717214e-02	0.06081453	0.9401066	P
## Raters	1.840418e-03	0.01793428	0.1026201	P
## Raters:Occasions	1.304646e-07	NA	NA	B
## Raters:Expressiveness	1.304646e-07	NA	NA	B
## Raters:Occasions:Expressiveness	1.304646e-07	NA	NA	B
## Raters:Expressiveness:Patients	1.862729e-01	0.07477739	2.4910326	P
## MotionsExpress:WMotionsExpress!R	1.000000e+00	NA	NA	F
## MotionsExpress:WMotionsExpress!MotionsExpress_Act,Exp	1.054088e+00	0.12381945	8.5131048	P
## MotionsExpress:WMotionsExpress!MotionsExpress_Act,Unexp	5.034771e-01	0.08401565	5.9926589	P
## MotionsExpress:WMotionsExpress!MotionsExpress_Pas,Exp	1.368750e+01	1.15101141	11.8917194	P
## MotionsExpress:WMotionsExpress!MotionsExpress_Pas,Unexp	2.176451e+00	0.19672484	11.0634281	P

```
##
```

	%ch
## Viewings	0.0
## Expressiveness:Patients	0.0
## Expressiveness:Patients:Trainings	0.0
## Expressiveness:Patients:Trainings:Motions	0.0
## Raters	0.1
## Raters:Occasions	0.0
## Raters:Expressiveness	0.0
## Raters:Occasions:Expressiveness	0.0
## Raters:Expressiveness:Patients	0.0
## MotionsExpress:WMotionsExpress!R	0.0
## MotionsExpress:WMotionsExpress!MotionsExpress_Act,Exp	0.0
## MotionsExpress:WMotionsExpress!MotionsExpress_Act,Unexp	0.0
## MotionsExpress:WMotionsExpress!MotionsExpress_Pas,Exp	0.0
## MotionsExpress:WMotionsExpress!MotionsExpress_Pas,Unexp	0.0

```
##
```

```
##
```

```
## ##### Pseudo-anova table for fixed terms
```

```
##
```

```
##
```

```
## Wald tests for fixed effects.
```

```
## Response: Y
```

```
##
##
##      Df denDF  F.inc    Pr
## (Intercept)      1   6.0 18.070 0.0054
## Trainings      1   6.1  6.512 0.0428
## Motions        1   5.9 65.700 0.0002
## Trainings:Motions      1   7.3  2.622 0.1477
## Expressiveness      1   6.0  7.088 0.0374
## Trainings:Expressiveness      1   7.3  0.273 0.6171
## Motions:Expressiveness      1   5.9 39.300 0.0008
## Trainings:Motions:Expressiveness      1  16.4  5.578 0.0309
##
##
## ##### Sequence of model investigations
##
## (If a row has NA for p but not denDF, DF and denDF relate to fixed and variance parameter numbers)
##
##      terms DF denDF  p      AIC      BIC      action
## 1 Intertier variance and unequal residual variances  0      10 NA 2285.241 2335.94 Starting model
printFormulae(md.asr)

##
##
## ##### Formulae from asreml object
##
## fixed: Y ~ Trainings * Motions * Expressiveness
## random: ~ Raters + Viewings + Expressiveness:Patients/Trainings/Motions + Raters:(Occasions * Expressiveness)
## residual: ~ idh(MotionsExpress):WMotionsExpress
```

Extract standardized residuals and fitted values and produce diagnostic plots

```
plotDiagnostics(md.asrt, plaid.dat)
```

Remove the bound terms

```
md.asrt <- rmboundary(md.asrt)
print(md.asrt, which = "test")
```

```
##
##
## ##### Sequence of model investigations
##
## (If a row has NA for p but not denDF, DF and denDF relate to fixed and variance parameter numbers)
##
##      terms DF denDF  p      AIC      BIC      action
## 1 Intertier variance and unequal residual variances  0      10 NA 2285.241 2335.94 Starting model
## 2      Raters:Expressiveness      1      NA NA      NA      NA      Boundary
## 3      Raters:Occasions      1      NA NA      NA      NA      Boundary
## 4      Raters:Expressiveness:Occasions      1      NA NA      NA      NA      Boundary
```

Test the remaining components for intertier interactions using AIC

```
intvcomps <- md.asrt$asreml.obj$vparameters[c(3:4,6)]
terms <- names(intvcomps)
```

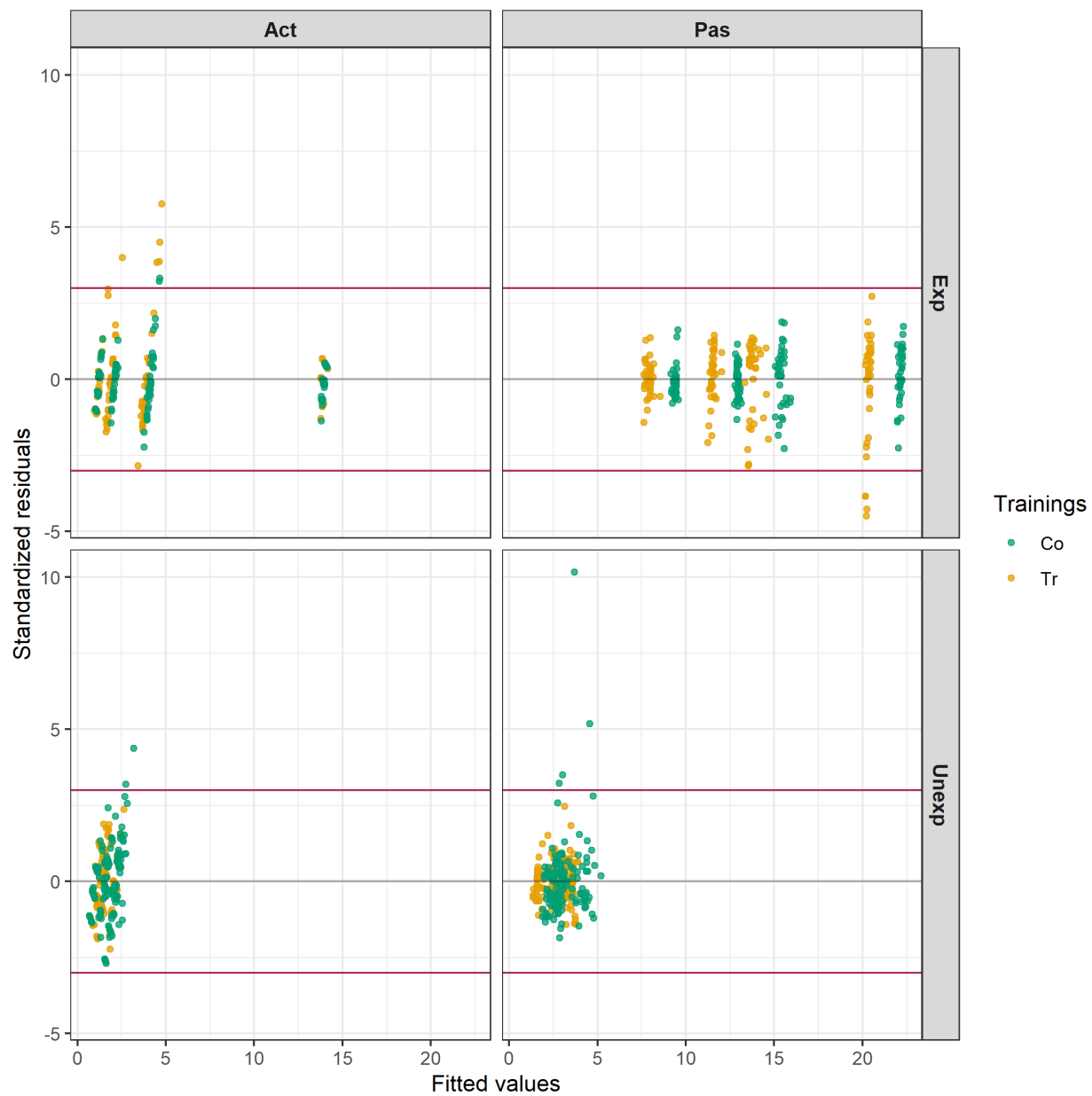


Figure 13: plot of chunk asr_md

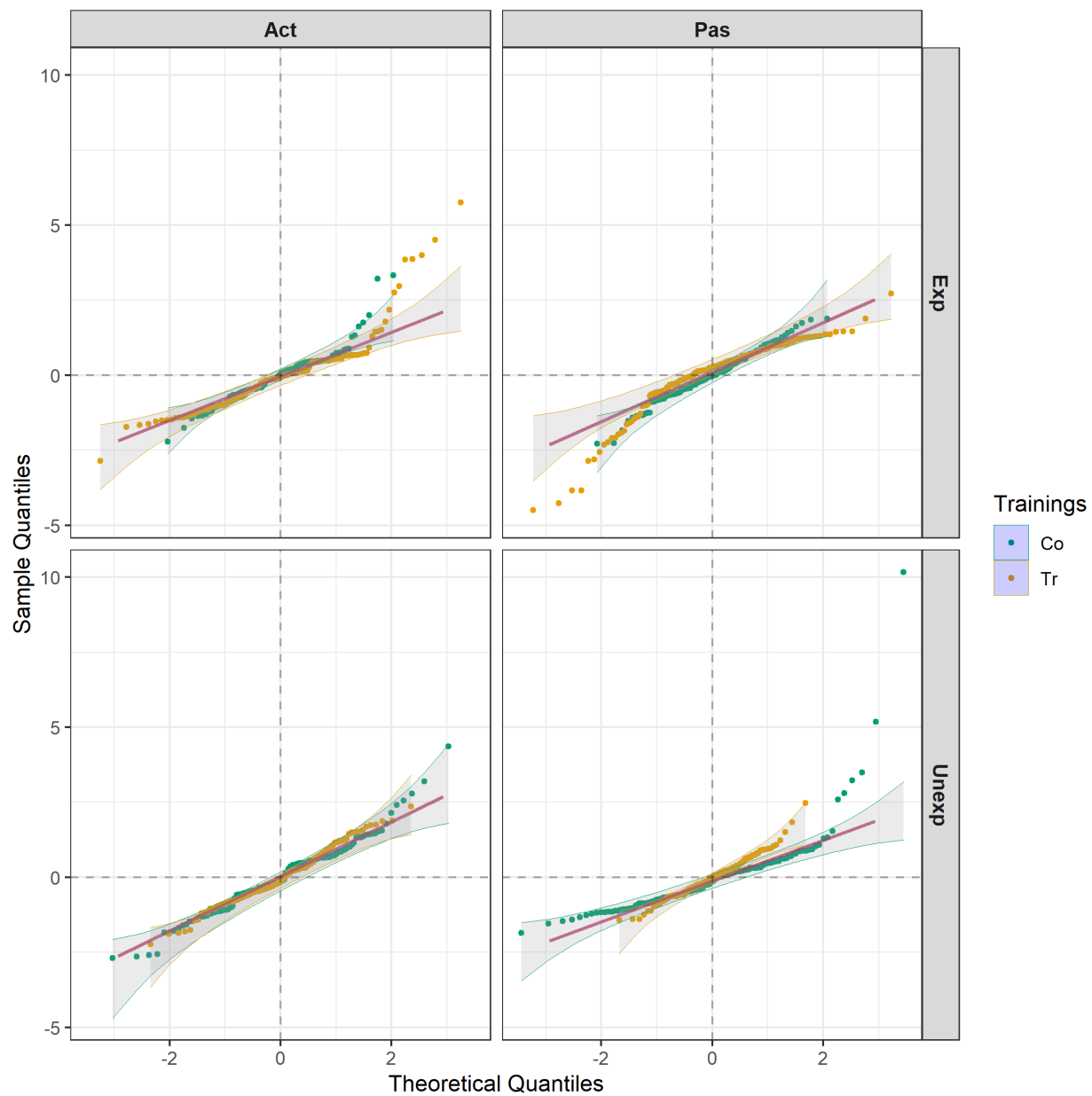


Figure 14: plot of chunk asr_md

```

md.AIC.asrt <- md.asrt
for (term in terms)
  md.AIC.asrt <- changeModelOnIC(md.AIC.asrt, dropRandom = term,
                                which.IC = "AIC",
                                label = paste("Try dropping", term))

## Warning in infoCriteria.asreml(asrtests.obj$asreml.obj, ICLikelihood = ic.lik, : The following bound
## MotionsExpress:WMotionsExpress!R
## Calculating denominator DF
## Warning in infoCriteria.asreml(asreml.obj, ICLikelihood = ic.lik, bound.exclusions = bound.exclusion
## MotionsExpress:WMotionsExpress!R
## Calculating denominator DF
## Warning in infoCriteria.asreml(new.asrtests.obj$asreml.obj, ICLikelihood = ic.lik, : The following b
## MotionsExpress:WMotionsExpress!R
## Calculating denominator DF
## Warning in infoCriteria.asreml(new.asrtests.obj$asreml.obj, ICLikelihood = ic.lik, : The following b
## MotionsExpress:WMotionsExpress!R
## Calculating denominator DF
## Warning in infoCriteria.asreml(asreml.obj, ICLikelihood = ic.lik, bound.exclusions = bound.exclusion
## MotionsExpress:WMotionsExpress!R
## Calculating denominator DF
## Warning in infoCriteria.asreml(new.asrtests.obj$asreml.obj, ICLikelihood = ic.lik, : The following b
## MotionsExpress:WMotionsExpress!R
## Calculating denominator DF
## Warning in infoCriteria.asreml(new.asrtests.obj$asreml.obj, ICLikelihood = ic.lik, : The following b
## MotionsExpress:WMotionsExpress!R
## Calculating denominator DF
## Warning in infoCriteria.asreml(asreml.obj, ICLikelihood = ic.lik, bound.exclusions = bound.exclusion
## MotionsExpress:WMotionsExpress!R
## Calculating denominator DF
## Warning in infoCriteria.asreml(new.asrtests.obj$asreml.obj, ICLikelihood = ic.lik, : The following b
## MotionsExpress:WMotionsExpress!R
(md.AIC.asrt)

##
##
## #### Summary of the fitting of the variance parameters
##
##
## component std.error z.ratio bound
## Viewings 1.398494383 0.86877041 1.6097399 P
## Expressiveness:Patients 14.563670533 8.85621923 1.6444569 P
## Expressiveness:Patients:Trainings:Motions 0.062024263 0.04797916 1.2927335 P
## Raters 0.001810873 0.01793291 0.1009804 P
## Raters:Expressiveness:Patients 0.186764054 0.07471863 2.4995646 P
## MotionsExpress:WMotionsExpress!R 1.000000000 NA NA F
## MotionsExpress:WMotionsExpress!MotionsExpress_Act,Exp 1.053617727 0.12377800 8.5121564 P
## MotionsExpress:WMotionsExpress!MotionsExpress_Act,Unexp 0.503082797 0.08395296 5.9924365 P
## MotionsExpress:WMotionsExpress!MotionsExpress_Pas,Exp 13.686303631 1.15093950 11.8914188 P

```

```

## MotionsExpress:WMotionsExpress!MotionsExpress_Pas,Unexp 2.176228975 0.19670439 11.0634491 P
## %ch
## Viewings 0.0
## Expressiveness:Patients 0.0
## Expressiveness:Patients:Trainings:Motions 0.1
## Raters 0.1
## Raters:Expressiveness:Patients 0.0
## MotionsExpress:WMotionsExpress!R 0.0
## MotionsExpress:WMotionsExpress!MotionsExpress_Act,Exp 0.0
## MotionsExpress:WMotionsExpress!MotionsExpress_Act,Unexp 0.0
## MotionsExpress:WMotionsExpress!MotionsExpress_Pas,Exp 0.0
## MotionsExpress:WMotionsExpress!MotionsExpress_Pas,Unexp 0.0
##
##
## ##### Pseudo-anova table for fixed terms
##
##
## Wald tests for fixed effects.
## Response: Y
##
##
## Df denDF F.inc Pr
## (Intercept) 1 6.0 18.070 0.0054
## Trainings 1 9.2 6.895 0.0271
## Motions 1 5.9 65.700 0.0002
## Expressiveness 1 6.0 7.088 0.0374
## Trainings:Motions 1 11.9 2.577 0.1346
## Trainings:Expressiveness 1 9.9 0.324 0.5821
## Motions:Expressiveness 1 5.9 39.300 0.0008
## Trainings:Motions:Expressiveness 1 27.5 5.427 0.0274
##
##
## ##### Sequence of model investigations
##
## (If a row has NA for p but not denDF, DF and denDF relate to fixed and variance parameter numbers)
##
##
## terms DF denDF p AIC BIC
## 1 Intertier variance and unequal residual variances 0 10 NA 2285.241005 2335.9397464
## 2 Raters:Expressiveness 1 NA NA NA NA
## 3 Raters:Occasions 1 NA NA NA NA
## 4 Raters:Expressiveness:Occasions 1 NA NA NA NA
## 5 Try dropping Expressiveness:Patients:Trainings 0 -1 NA -1.983624 -7.0534978
## 6 Try dropping Expressiveness:Patients:Trainings:Motions 0 -1 NA 3.903492 -1.1663822
## 7 Try dropping Raters:Expressiveness:Patients 0 -1 NA 4.365970 -0.7039041
## action
## 1 Starting model
## 2 Boundary
## 3 Boundary
## 4 Boundary
## 5 Swapped
## 6 Unswapped
## 7 Unswapped
printFormulae(md.AIC.asrt$asreml.obj)
##

```

```
##
## ##### Formulae from asreml object
##
## fixed: Y ~ Trainings + Motions + Expressiveness + Trainings:Motions + Trainings:Expressiveness + Mot
## random: ~ Raters + Viewings + Expressiveness:Patients + Raters:Expressiveness:Patients + Expressiven
## residual: ~ idh(MotionsExpress):WMotionsExpress

AIC.tab <- rbind(AIC.tab, infoCriteria(md.AIC.asrt$asreml.obj))

## Warning in infoCriteria.asreml(md.AIC.asrt$asreml.obj): The following bound terms were discounted:
## MotionsExpress:WMotionsExpress!R
```

Extract standardized residuals and fitted values and produce diagnostic plots

```
plotDiagnostics(md.AIC.asrt, plaid.dat)
```

Get and plot predictions based on models chosen using AIC

```
md.AIC.choice <- chooseModel(md.AIC.asrt, terms.marginality = marg.fix,
                             denDF="numeric")

## Calculating denominator DF
chosen.terms <- unlist(md.AIC.choice$sig.terms)
submod <- length(chosen.terms) != 1 && chosen.terms != "Trainings:Motions:Expressiveness"
current.asr <- md.AIC.asrt$asreml.obj
md.AIC.diffs <- predictPlus(current.asr, classify = "Trainings:Motions:Expressiveness",
                            error.intervals = "half", meanLSD.type = "factor",
                            LSDby = c("Motions","Expressiveness"),
                            wald = md.AIC.asrt$wald.tab,
                            Vmatrix = TRUE, tables = "none")
```

If the chosen fixed model is a submodel of T:M:E, transform predictions to conform to the chosen model

```
if (submod)
{
  md.AIC.TMEDiffs <- md.AIC.diffs #save full diffs
  submod.ch <- paste("~", paste(chosen.terms, collapse = " + "))
  cat("\n\n#### Chosen model: ", submod.ch, "\n\n")
  md.AIC.subdiffs <- linTransform(md.AIC.diffs,
                                linear.transformation = as.formula(submod.ch),
                                classify = "Trainings:Motions:Expressiveness",
                                error.intervals = "half", meanLSD.type = "factor",
                                LSDby = c("Motions","Expressiveness"),
                                wald = md.AIC.asrt$wald.tab, tables = "none")

  (md.AIC.subdiffs$predictions)
} else
cat("\n\n#### Chosen model: ~ Trainings:Motions:Expressiveness\n\n")

##
##
## ##### Chosen model: ~ Trainings:Motions:Expressiveness
```

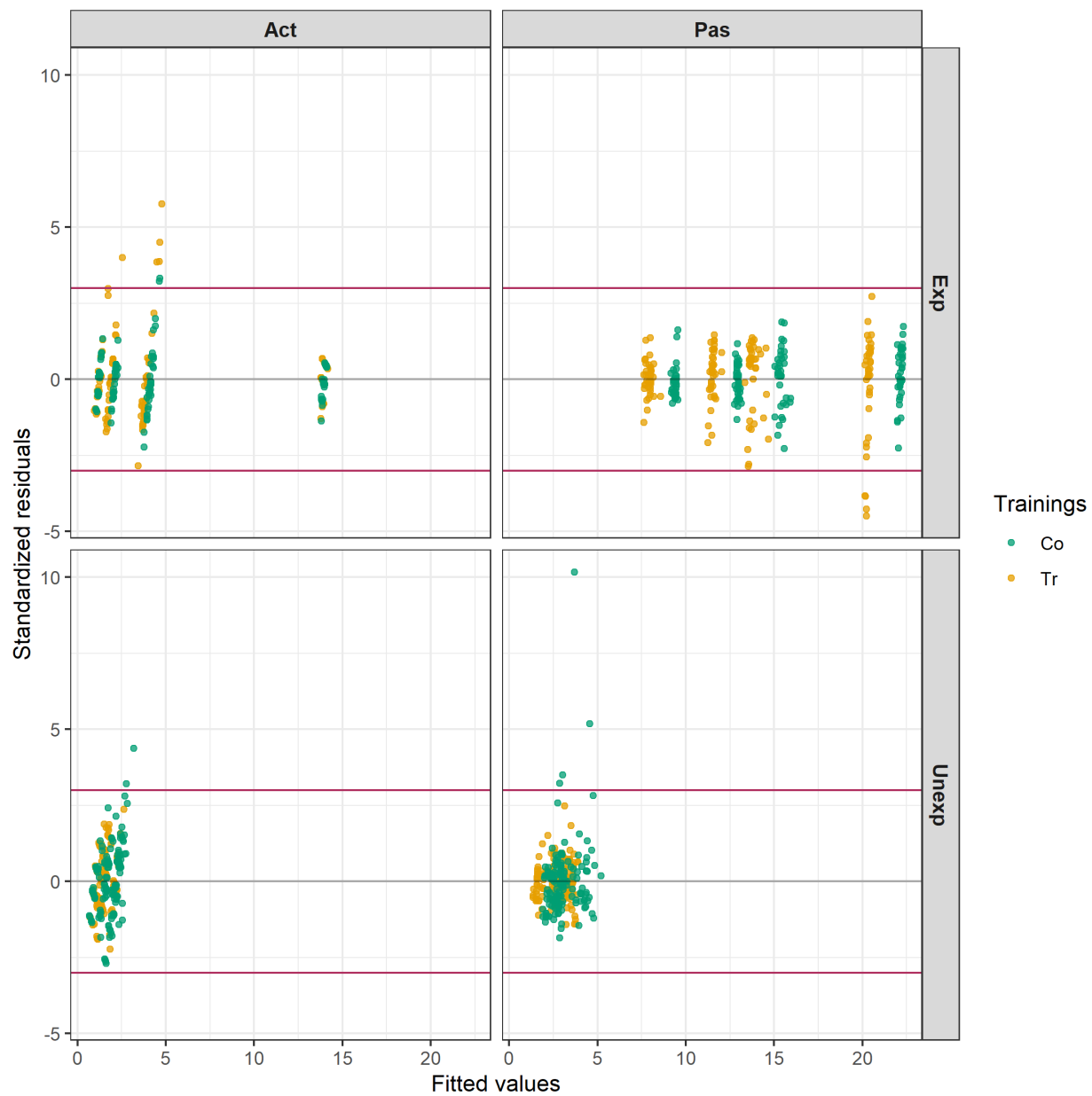


Figure 15: plot of chunk asr_mdAIC

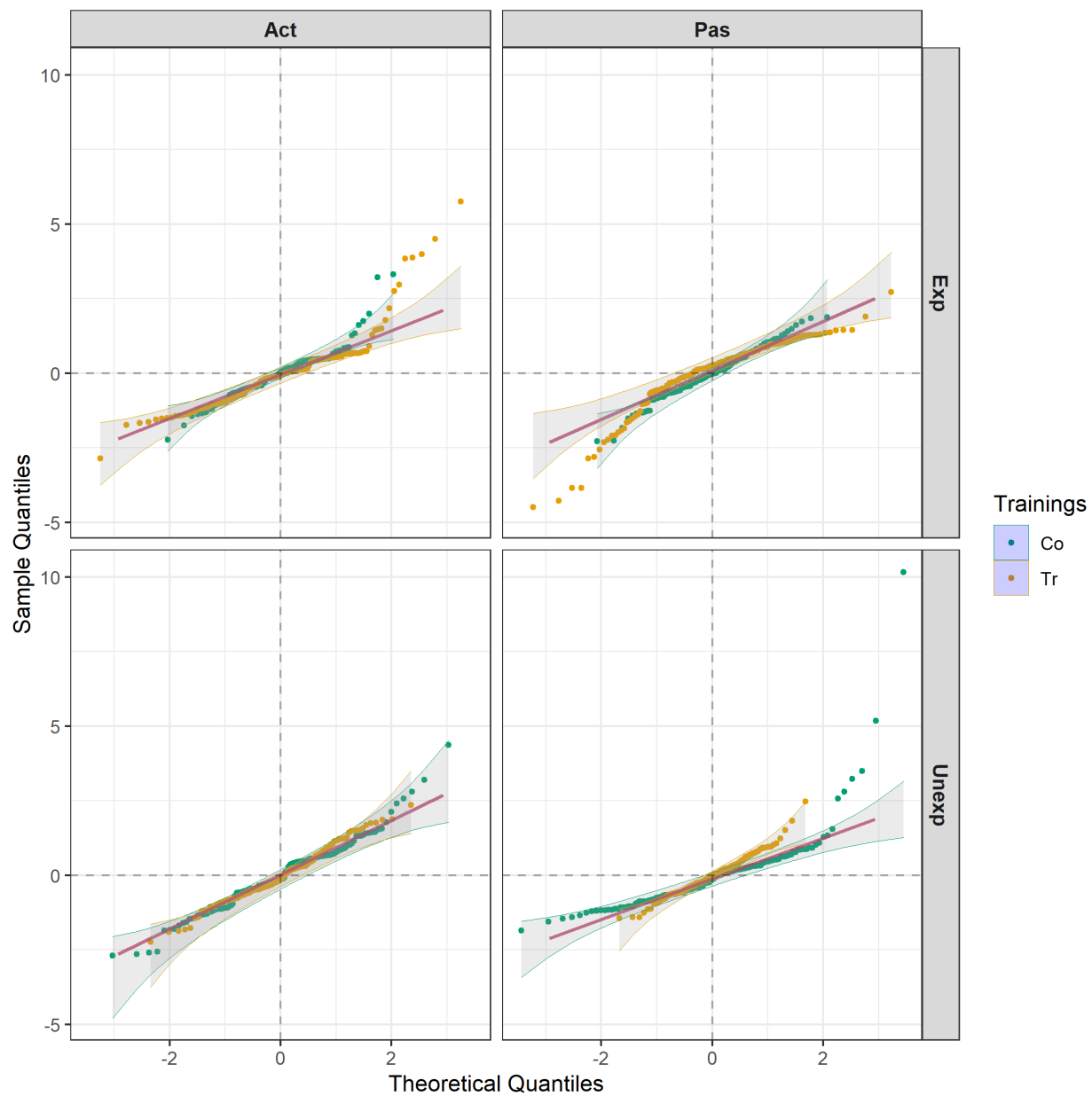


Figure 16: plot of chunk asr_mdAIC

```
(md.AIC.diffs$predictions)
```

```
##
##
## #### Predictions
##
## Notes:
## - The predictions are obtained by averaging across the hypertable
##   calculated from model terms constructed solely from factors in
##   the averaging and classify sets.
## - Use 'average' to move ignored factors into the averaging set.
## - The ignored set: Raters,Viewings,Patients
##
##
##   Trainings Motions Expressiveness predicted.value standard.error upper.halfLeastSignificant.limit
## 1      Co      Act      Exp      5.370101      2.003617      5.594396
## 2      Co      Act      Unexp     1.790563      2.002688      1.996696
## 3      Co      Pas      Exp     14.982800      2.024805     15.462046
## 4      Co      Pas      Unexp     3.057149      2.005508      3.314534
## 5      Tr      Act      Exp     5.261824      2.003617      5.486119
## 6      Tr      Act      Unexp     1.508846      2.002688      1.714979
## 7      Tr      Pas      Exp     13.407878      2.024805     13.887125
## 8      Tr      Pas      Unexp     2.705541      2.005508      2.962926
##   lower.halfLeastSignificant.limit est.status
## 1                                5.145807 Estimable
## 2                                1.584430 Estimable
## 3                               14.503553 Estimable
## 4                                2.799763 Estimable
## 5                                5.037530 Estimable
## 6                                1.302712 Estimable
## 7                               12.928632 Estimable
## 8                                2.448155 Estimable

plotPredictions(md.AIC.diffs$predictions, classify = "Trainings:Motions:Expressiveness",
  y = "predicted.value",
  error.intervals = "half", interval.annotate = FALSE,
  ggplotFuncs = list(facet_grid(cols = vars(Expressiveness, Motions))))

plotPvalues(md.AIC.diffs, factors.per.grid = 2, show.sig = TRUE,
  title = "Prior allocation model, \nunequal variances")
```

Print out AIC values

```
(AIC.tab <- cbind(Model = letters[1:4], AIC.tab))
```

```
##   Model fixedDF varDF NBound      AIC      BIC    loglik
## 1    a        0     4      0 3041.308 3061.588 -1516.654
## 2    b        0     7      1 2291.709 2327.198 -1138.854
## 3    c        0    10      0 2965.908 3016.607 -1472.954
## 4    d        0     9      1 2283.257 2328.886 -1132.629
```

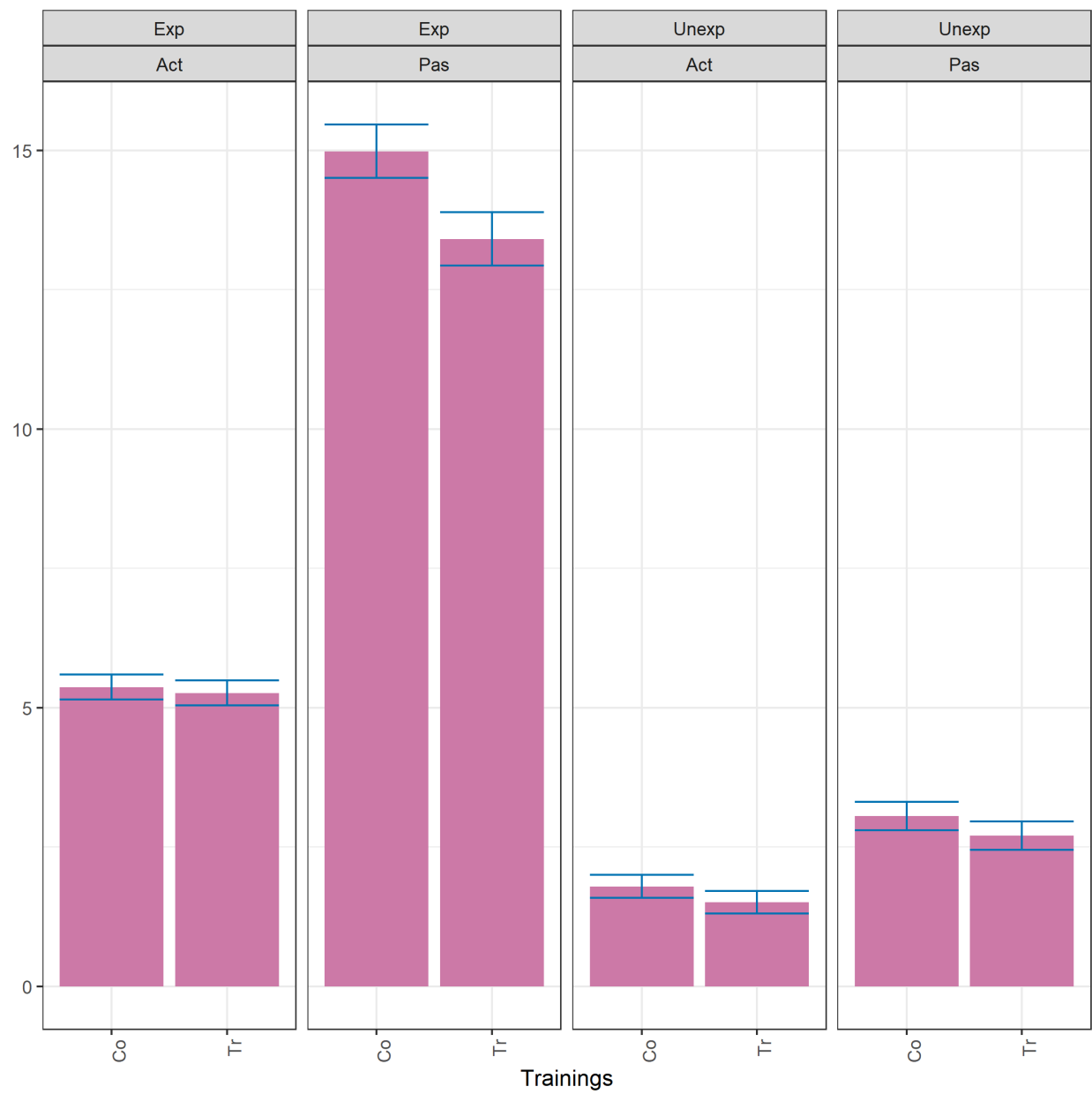


Figure 17: plot of chunk unnamed-chunk-23

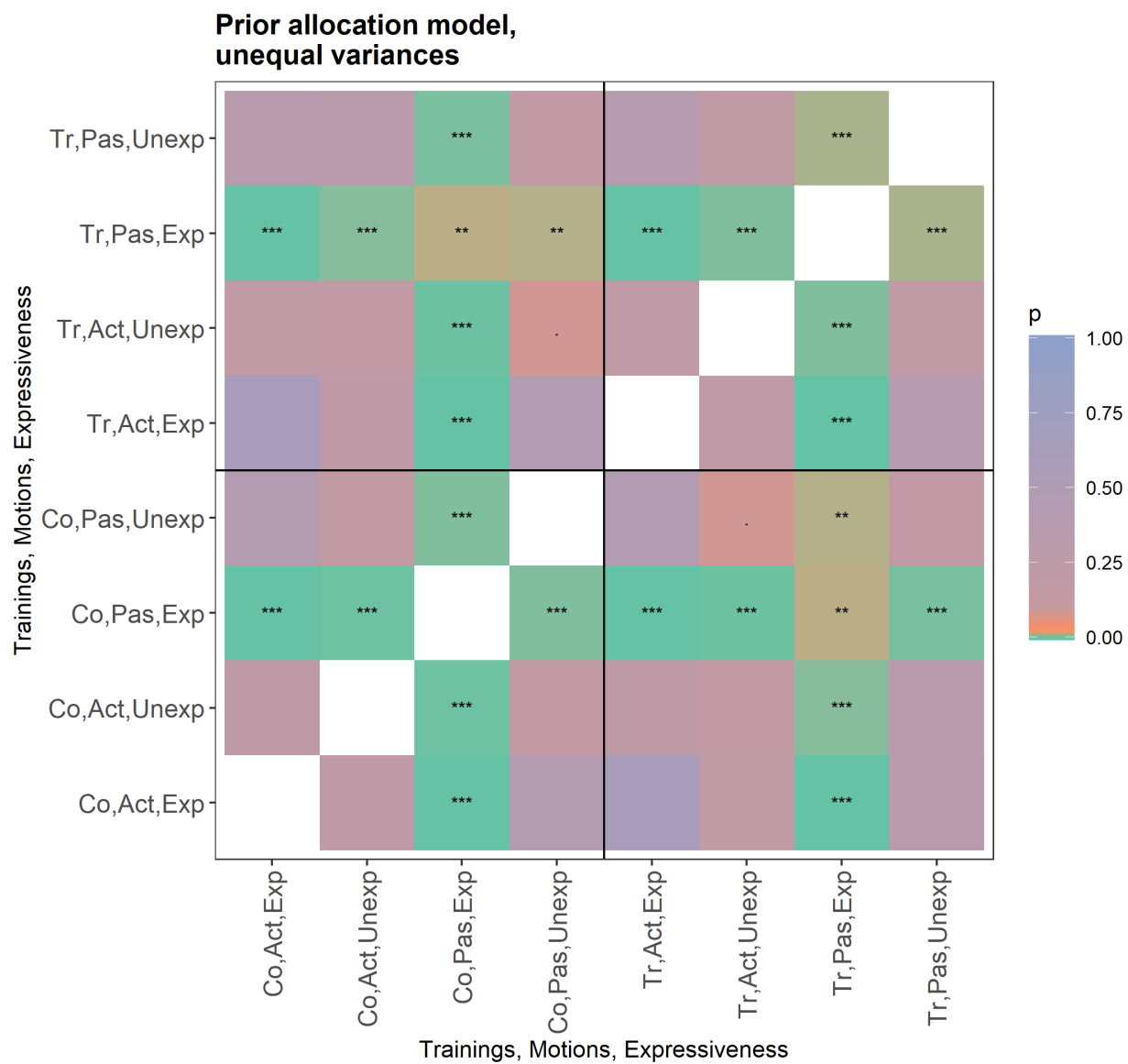


Figure 18: plot of chunk unnamed-chunk-23

Save the workspace

```
save.image("FHanal.RData")
```