DefTypNorm

```
library(plyr)
library(tidyverse)
```

```
## -- Attaching core tidyverse packages ------------------- tidyverse 2.0.0.9000 --
## v dplyr     1.1.4     v readr     2.1.5
## v forcats   1.0.0     v stringr   1.5.1
## v ggplot2   3.5.1     v tibble    3.2.1
## v lubridate 1.9.3     v tidyr     1.3.1
## v purrr     1.0.2
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::arrange()   masks plyr::arrange()
## x purrr::compact()   masks plyr::compact()
## x dplyr::count()     masks plyr::count()
## x dplyr::desc()      masks plyr::desc()
## x dplyr::failwith()  masks plyr::failwith()
## x dplyr::filter()    masks stats::filter()
## x dplyr::id()        masks plyr::id()
## x dplyr::lag()       masks stats::lag()
## x dplyr::mutate()    masks plyr::mutate()
## x dplyr::rename()    masks plyr::rename()
## x dplyr::summarise() masks plyr::summarise()
## x dplyr::summarize() masks plyr::summarize()
## i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become error
```

```
library(lme4)
```

```
## Loading required package: Matrix
##
## Attaching package: 'Matrix'
##
## The following objects are masked from 'package:tidyr':
##
##     expand, pack, unpack
```

```
library(lmerTest)
```

```
##
## Attaching package: 'lmerTest'
##
## The following object is masked from 'package:lme4':
##
##     lmer
##
## The following object is masked from 'package:stats':
##
##     step
```

```r
library(emmeans)
```

```
## Welcome to emmeans.
## Caution: You lose important information if you filter this package's results.
## See '? untidy'
```

```r
read.pcibex <- function(filepath, auto.colnames=TRUE, fun.col=function(col,cols){cols[cols==col]<-paste
  n.cols <- max(count.fields(filepath,sep=",",quote=NULL),na.rm=TRUE)
  if (auto.colnames){
    cols <- c()
    con <- file(filepath, "r")
    while ( TRUE ) {
      line <- readLines(con, n = 1, warn=FALSE)
      if ( length(line) == 0) {
        break
      }
      m <- regmatches(line,regexec("^# (\\d+)\\. (.+)\\.$",line))[[1]]
      if (length(m) == 3) {
        index <- as.numeric(m[2])
        value <- m[3]
        if (index < length(cols)){
          cols <- c()
        }
        if (is.function(fun.col)){
          cols <- fun.col(value,cols)
        }
        cols[index] <- value
        if (index == n.cols){
          break
        }
      }
    }
    close(con)
    return(read.csv(filepath, comment.char="#", header=FALSE, col.names=cols))
  }
  else{
    return(read.csv(filepath, comment.char="#", header=FALSE, col.names=seq(1:n.cols)))
  }
}

Data <- read.pcibex("results.csv", auto.colnames=FALSE)


colnames(Data) <- c("Time", "IP", "Controller", "Order", "ElementNum", "Label", "Group", "PennElementTy




#DefTypAnaphData <- Data %>%
#    filter(PennElementType=="Selector")

DTATRRT <- subset(Data, (PennElementType=="Selector"|Parameter=="play") & Type=="critical")

DTATRRT <- DTATRRT %>%
  select(ProlificID, Parameter, item, EventTime) %>%
  pivot_wider(names_from = Parameter, values_from = EventTime)




DTATR <- subset(Data, PennElementType=="Selector" & Type=="critical")

nrow(DTATR)
```

```
## [1] 736
```

```
nrow(left_join(DTATR, DTATRRT, by=c("ProlificID", "item")))
```

```
## [1] 736
```

```
DTATR <- left_join(DTATR, DTATRRT, by=c("ProlificID", "item"))
DTATR$DefType <- "TR"

length(unique(DTATR$ProlificID))
```

```
## [1] 46
```

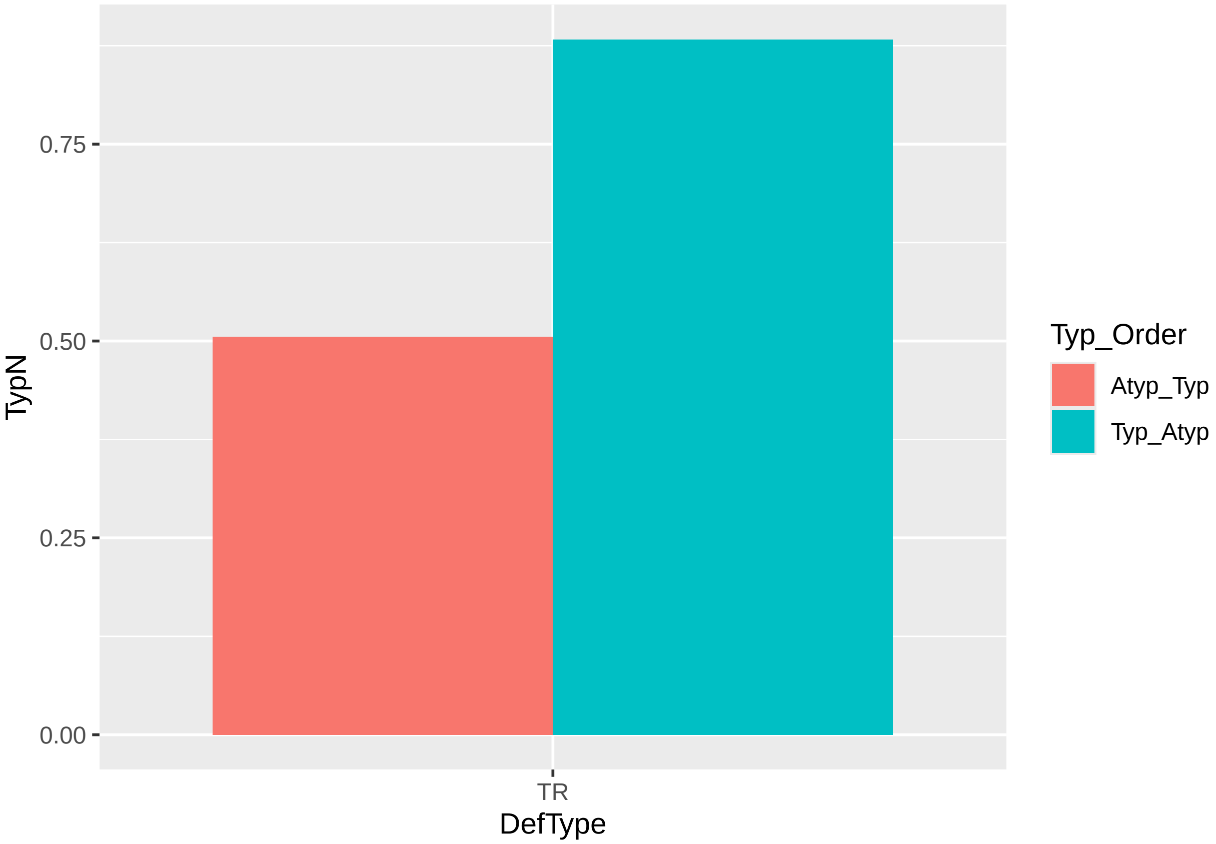
```
#46

nrow(subset(DTATR, !(Value=="I1" | Value=="I2")))
```

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

```
#1

DTATR <- subset(DTATR, Value=="I1" | Value=="I2")



DTATR$Typ <- (DTATR$Typ_Order=="Typ_Atyp" & DTATR$Value=="I1") | (DTATR$Typ_Order=="Atyp_Typ" & DTATR$Va


DTATR %>%
  group_by(Typ_Order) %>%
  summarize(Typ=mean(Typ=="TRUE"), n=length(Value))
```

```
## # A tibble: 2 x 3
##   Typ_Order   Typ     n
##   <chr>     <dbl> <int>
## 1 Atyp_Typ  0.505   368
## 2 Typ_Atyp  0.883   367
```

```
DTATR$TypN <- as.numeric(DTATR$Typ)

ggplot(DTATR, aes(DefType, TypN, fill=Typ_Order )) +
  stat_summary(fun="mean", geom="bar", position="dodge")
```



```
# RTs:

DTATR$Selection <- as.numeric(DTATR$Selection)
DTATR$play <- as.numeric(DTATR$play)
DTATR$RT <- DTATR$Selection-DTATR$play

nrow(subset(DTATR, RT > 10000))/nrow(DTATR)
```

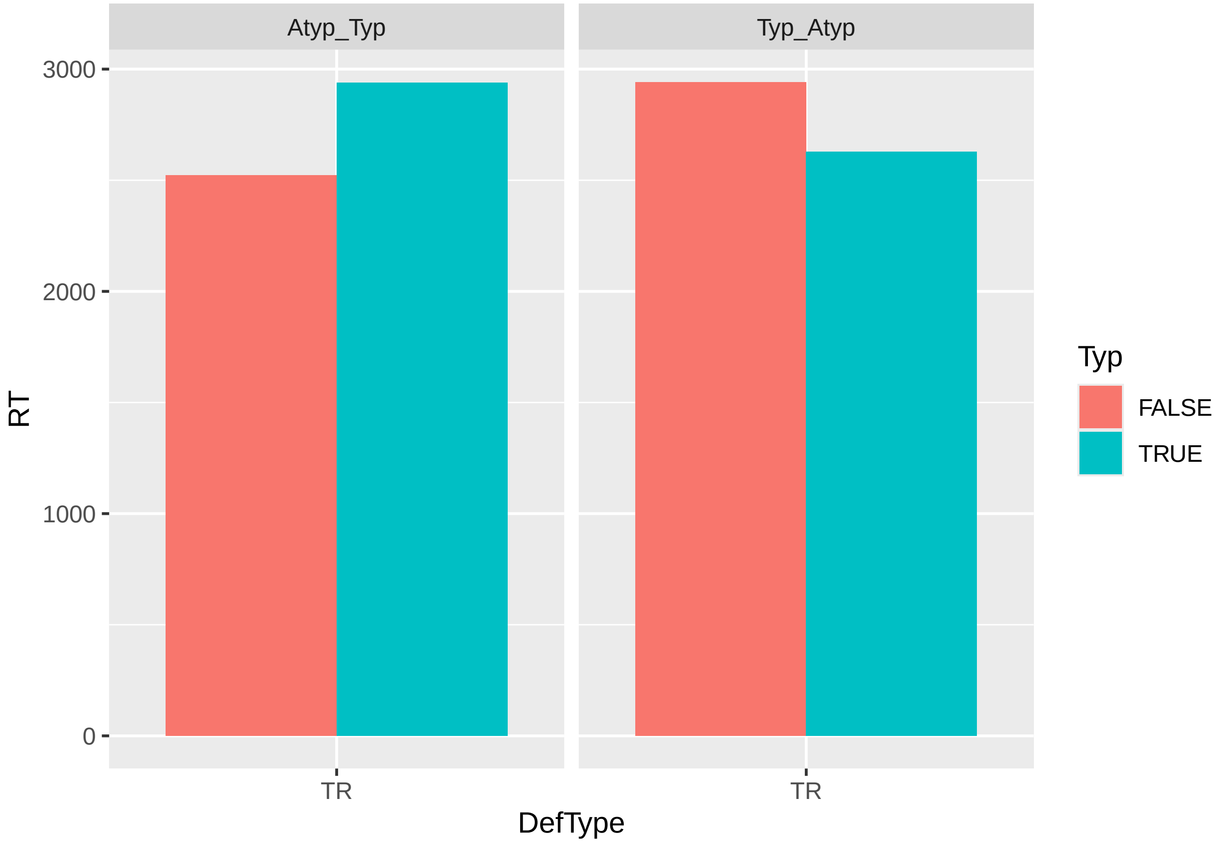
```
## [1] 0.01360544
```

```
DTATR %>%
  filter(RT < 10000) %>%
  group_by(DefType, Typ_Order, Typ) %>%
  summarize(RT=mean(RT), n=length(Value))
```

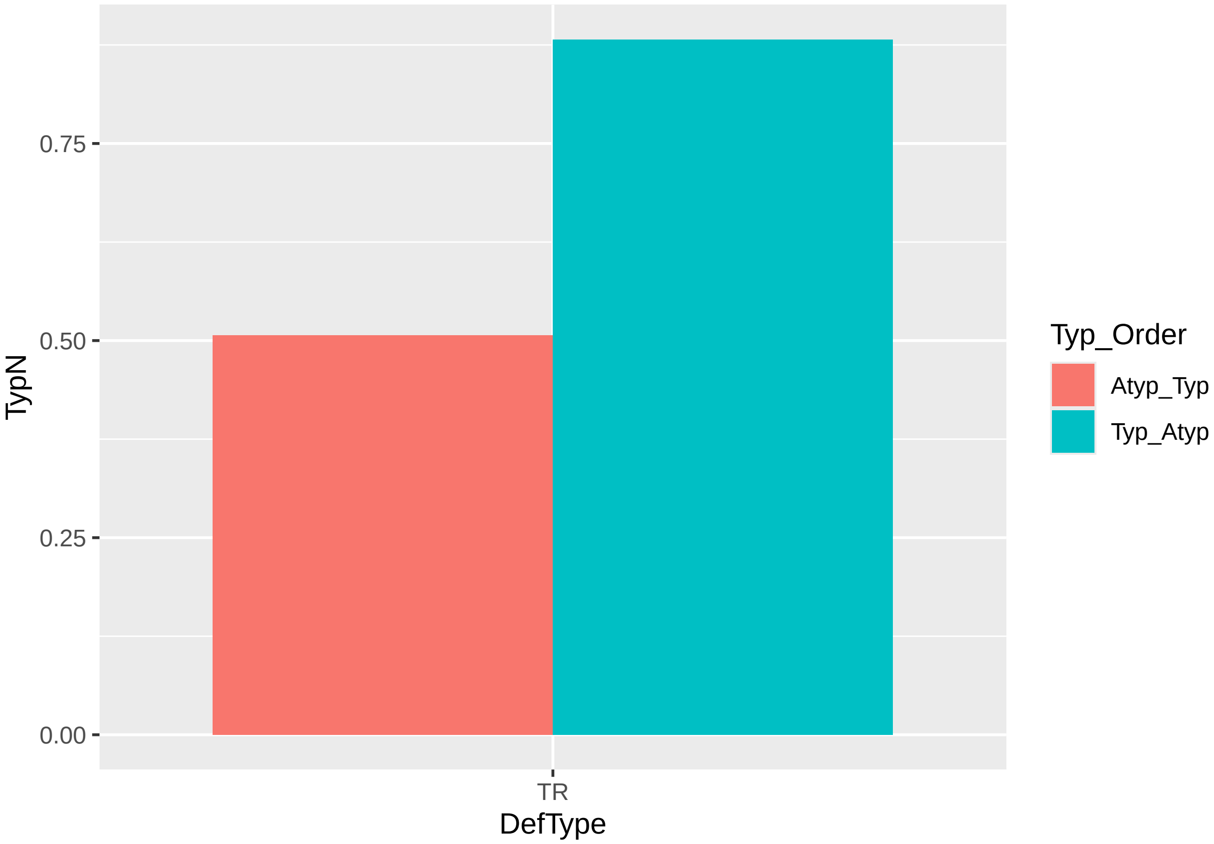
```
## `summarise()` has grouped output by 'DefType', 'Typ_Order'. You can override
## using the `.groups` argument.
```

```
## # A tibble: 4 x 5
## # Groups:   DefType, Typ_Order [2]
##   DefType Typ_Order Typ      RT     n
##   <chr>   <chr>     <lgl> <dbl> <int>
## 1 TR      Atyp_Typ  FALSE 2521.   178
## 2 TR      Atyp_Typ  TRUE  2939.   183
## 3 TR      Typ_Atyp  FALSE 2940.    43
## 4 TR      Typ_Atyp  TRUE  2628.   321
```

```
ggplot(subset(DTATR, RT < 10000), aes(DefType, RT, fill= Typ )) +
  stat_summary(fun="mean", geom="bar", position="dodge") +
  facet_wrap(~Typ_Order)
```



```
ggplot(subset(DTATR, RT < 10000), aes(DefType, TypN, fill=Typ_Order )) +
  stat_summary(fun="mean", geom="bar", position="dodge")
```



```
myCenter= function(x) {
  if (is.numeric(x)) { return(x - mean(x, na.rm=T)) }
  if (is.factor(x)) {
    x= as.numeric(x)
    return(x - mean(x, na.rm=T))
  }
  if (is.data.frame(x) || is.matrix(x)) {
    m= matrix(nrow=nrow(x), ncol=ncol(x))
    colnames(m)= paste("c", colnames(x), sep="")
    for (i in 1:ncol(x)) {
      m[,i]= myCenter(x[,i])
    }
    return(as.data.frame(m))
  }
}

DTATR$DefTypeC <- 0
DTATR$Typ_OrderC <- myCenter(as.factor(as.character(DTATR$Typ_Order)))
```

```
nrow(DTATR)
```

```
## [1] 735
```

```
TypDataItemDiff <- read_csv("TypDataItemDiff.csv")
```

```
## Rows: 16 Columns: 5
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr (1): noun_en
## dbl (4): item, 1, 2, Diff
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```
TypDataItemDiff$item <- as.factor(TypDataItemDiff$item)
DTATR$item <- as.factor(DTATR$item)
nrow(left_join(DTATR, TypDataItemDiff, by="item"))
```

```
## [1] 735
```

```
DTATR <- left_join(DTATR, TypDataItemDiff, by="item")

DTATR$ProlificID <- as.factor(DTATR$ProlificID)

summary(glmer(Typ ~   Diff + (1|ProlificID) + (1|item), data=subset(DTATR, Typ_Order=="Atyp_Typ"), famil
```

```
## Generalized linear mixed model fit by maximum likelihood (Laplace
##   Approximation) [glmerMod]
##  Family: binomial  ( logit )
## Formula: Typ ~ Diff + (1 | ProlificID) + (1 | item)
##    Data: subset(DTATR, Typ_Order == "Atyp_Typ")
##
##      AIC      BIC   logLik deviance df.resid
##    453.5    469.1   -222.7    445.5      364
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -2.0643 -0.6416  0.3237  0.6466  2.1453
##
## Random effects:
##  Groups     Name        Variance  Std.Dev.
##  ProlificID (Intercept) 2.380e+00 1.5427796
##  item       (Intercept) 3.642e-08 0.0001909
## Number of obs: 368, groups:  ProlificID, 46; item, 16
##
## Fixed effects:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)  -0.3199     0.7249  -0.441    0.659
## Diff          0.1116     0.2088   0.534    0.593
##
## Correlation of Fixed Effects:
##      (Intr)
## Diff -0.933
```

```
DefTypENDiff.lmer <- glmer(Typ ~  Typ_OrderC + Diff  + (1|ProlificID) +(1|item) , data=DTATR, family="bi
```

```
## Warning in checkConv(attr(opt, "derivs"), opt$par, ctrl = control$checkConv, :
## Model failed to converge with max|grad| = 0.036936 (tol = 0.002, component 1)
```

```
summary(DefTypENDiff.lmer)
```

```
## Generalized linear mixed model fit by maximum likelihood (Laplace
##   Approximation) [glmerMod]
##  Family: binomial  ( logit )
## Formula: Typ ~ Typ_OrderC + Diff + (1 | ProlificID) + (1 | item)
##    Data: DTATR
##
##      AIC      BIC   logLik deviance df.resid
##    746.3    769.3   -368.1    736.3      730
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -4.3851 -0.6799  0.2924  0.5005  2.1345
##
## Random effects:
##  Groups     Name        Variance Std.Dev.
##  ProlificID (Intercept) 0.93093  0.9648
##  item       (Intercept) 0.04897  0.2213
## Number of obs: 735, groups:  ProlificID, 46; item, 16
##
## Fixed effects:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept) 0.711114   0.001915  371.29   <2e-16 ***
## Typ_OrderC  2.317826   0.001916 1209.79   <2e-16 ***
## Diff        0.150691   0.001915   78.67   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Correlation of Fixed Effects:
##            (Intr) Typ_OC
## Typ_OrderC 0.000
## Diff       0.000  0.001
## optimizer (Nelder_Mead) convergence code: 0 (OK)
## Model failed to converge with max|grad| = 0.036936 (tol = 0.002, component 1)
```

```
DefTypENDiffAtypTyp.lmer1 <- glmer(Typ ~ Typ_OrderC * Diff  + (1|ProlificID) , data=DTATR, family="bino
summary(DefTypENDiffAtypTyp.lmer1)
```

```
## Generalized linear mixed model fit by maximum likelihood (Laplace
##   Approximation) [glmerMod]
##  Family: binomial  ( logit )
## Formula: Typ ~ Typ_OrderC * Diff + (1 | ProlificID)
##    Data: DTATR
##
##      AIC      BIC   logLik deviance df.resid
##    746.8    769.8   -368.4    736.8      730
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -4.7298 -0.6955  0.2927  0.4989  2.2241
##
## Random effects:
##  Groups     Name        Variance Std.Dev.
##  ProlificID (Intercept) 0.9047   0.9512
## Number of obs: 735, groups:  ProlificID, 46
##
## Fixed effects:
##                 Estimate Std. Error z value Pr(>|z|)
## (Intercept)       0.6176     0.5674   1.088    0.276
## Typ_OrderC        1.7794     1.1235   1.584    0.113
## Diff              0.1764     0.1692   1.043    0.297
## Typ_OrderC:Diff   0.1602     0.3459   0.463    0.643
##
## Correlation of Fixed Effects:
##             (Intr) Typ_OC Diff
## Typ_OrderC   0.360
## Diff        -0.949 -0.372
## Typ_OrdrC:D -0.361 -0.982  0.390
```

```
DefTypENDiffAtypTyp.lmer2 <- glmer(Typ ~ Typ_OrderC + Diff  + (1|ProlificID) , data=DTATR, family="bino
summary(DefTypENDiffAtypTyp.lmer2)
```

```
## Generalized linear mixed model fit by maximum likelihood (Laplace
##   Approximation) [glmerMod]
##  Family: binomial  ( logit )
## Formula: Typ ~ Typ_OrderC + Diff + (1 | ProlificID)
##    Data: DTATR
##
##      AIC      BIC   logLik deviance df.resid
##    745.0    763.4   -368.5    737.0      731
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -4.6827 -0.6999  0.2916  0.4992  2.2208
##
## Random effects:
##  Groups     Name        Variance Std.Dev.
##  ProlificID (Intercept) 0.9034   0.9505
## Number of obs: 735, groups:  ProlificID, 46
##
## Fixed effects:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)   0.7125     0.5294   1.346    0.178
## Typ_OrderC    2.2917     0.2144  10.689   <2e-16 ***
## Diff          0.1461     0.1554   0.941    0.347
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Correlation of Fixed Effects:
##            (Intr) Typ_OC
## Typ_OrderC  0.047
## Diff       -0.942  0.046
```

```
#Smaller BIC/AIC = Better
anova(DefTypENDiffAtypTyp.lmer1,DefTypENDiffAtypTyp.lmer2)
```

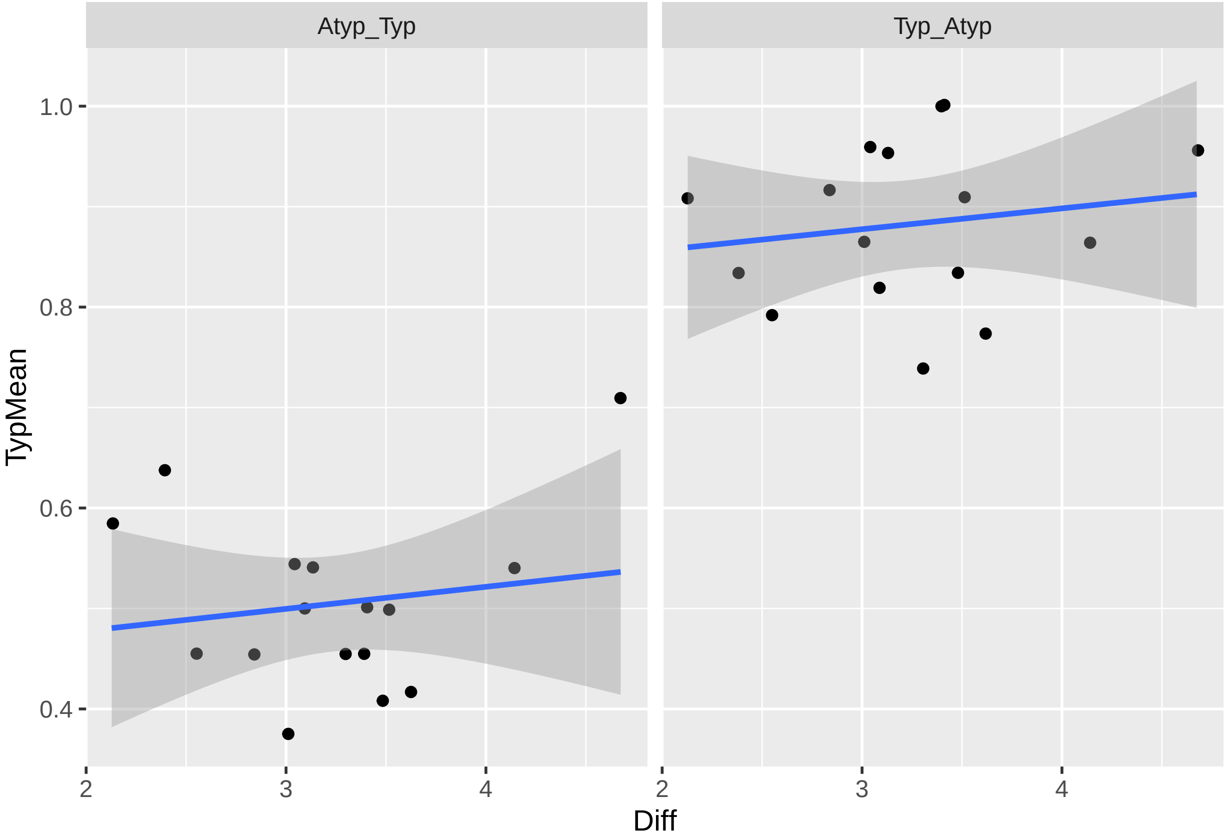
```
## Data: DTATR
## Models:
## DefTypENDiffAtypTyp.lmer2: Typ ~ Typ_OrderC + Diff + (1 | ProlificID)
## DefTypENDiffAtypTyp.lmer1: Typ ~ Typ_OrderC * Diff + (1 | ProlificID)
##                           npar    AIC    BIC  logLik deviance  Chisq Df
## DefTypENDiffAtypTyp.lmer2    4 744.99 763.39 -368.49   736.99
## DefTypENDiffAtypTyp.lmer1    5 746.77 769.77 -368.39   736.77 0.2133  1
##                           Pr(>Chisq)
## DefTypENDiffAtypTyp.lmer2
## DefTypENDiffAtypTyp.lmer1     0.6442
```

```
# ( seems like non-interaction model is better )


DTATRItem <- DTATR %>%
  select(item, DefType, Typ_Order, Typ, Diff) %>%
  group_by(item, DefType, Typ_Order) %>%
  summarize(TypMean=mean(Typ), Diff=mean(Diff))
```

```
## `summarise()` has grouped output by 'item', 'DefType'. You can override using
## the `.groups` argument.
```

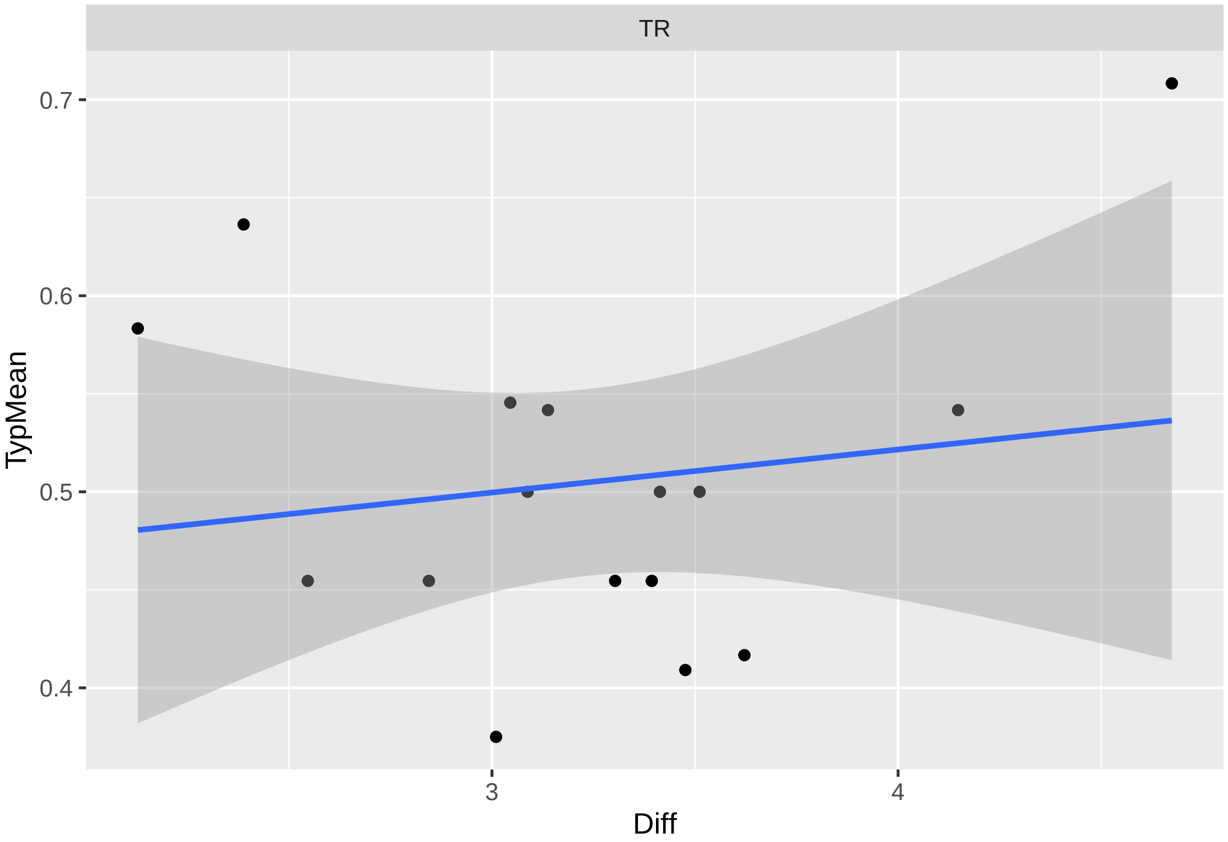
```
ggplot(DTATRItem, aes(Diff, TypMean)) +
  geom_jitter() +
  geom_smooth(method="lm") +
  facet_wrap(~Typ_Order)
```

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



```
ggplot(subset(DTATRItem, Typ_Order=="Atyp_Typ"), aes(Diff, TypMean)) +
  geom_point() +
  geom_smooth(method="lm") +
  facet_wrap(~DefType)
```

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


TR
TypMean
0.7
0.6
0.5
0.4
Diff
3
4