

ALDA_SEM Week 2 MLM

Marilyn Piccirillo

10/19/2017

```
rm(list = ls())
library(foreign)
library(tidyverse)

## Loading tidyverse: ggplot2
## Loading tidyverse: tibble
## Loading tidyverse: tidyr
## Loading tidyverse: readr
## Loading tidyverse: purrr
## Loading tidyverse: dplyr

## Warning: package 'tidyr' was built under R version 3.4.2
## Warning: package 'purrr' was built under R version 3.4.2
## Warning: package 'dplyr' was built under R version 3.4.2
## Conflicts with tidy packages -----
## filter(): dplyr, stats
## lag():    dplyr, stats
library(dplyr)
library(tidyr)
library(broom)
library(ggplot2)
library(lubridate)

##
## Attaching package: 'lubridate'
## The following object is masked from 'package:base':
##
##     date
library(lme4)

## Loading required package: Matrix
##
## Attaching package: 'Matrix'
## The following object is masked from 'package:tidyr':
##
##     expand
library(lavaan)

## This is lavaan 0.5-23.1097
## lavaan is BETA software! Please report any bugs.
library(sjstats)

## Warning: package 'sjstats' was built under R version 3.4.2
```

```

## Warning in checkMatrixPackageVersion(): Package version inconsistency detected.
## TMB was built with Matrix version 1.2.10
## Current Matrix version is 1.2.11
## Please re-install 'TMB' from source or restore original 'Matrix' package

##
## Attaching package: 'sjstats'

## The following object is masked from 'package:broom':
##
##     bootstrap

library(sjPlot)

## Warning: package 'sjPlot' was built under R version 3.4.2

library(semPlot)

#Pull in all three timepoints
DepressionT1 <- read.spss("StressStudy_T1short.sav", use.value.labels = FALSE, to.data.frame = TRUE)

## Warning in read.spss("StressStudy_T1short.sav", use.value.labels = FALSE, :
## StressStudy_T1short.sav: Unrecognized record type 7, subtype 10 encountered
## in system file

## Warning in read.spss("StressStudy_T1short.sav", use.value.labels = FALSE, :
## StressStudy_T1short.sav: Long string missing values record found (record
## type 7, subtype 22), but ignored

DepressionT2 <- read.spss("StressStudy_T2short.sav", use.value.labels = FALSE, to.data.frame = TRUE)

## Warning in read.spss("StressStudy_T2short.sav", use.value.labels = FALSE, :
## StressStudy_T2short.sav: Unrecognized record type 7, subtype 10 encountered
## in system file

## Warning in read.spss("StressStudy_T2short.sav", use.value.labels = FALSE, :
## StressStudy_T2short.sav: Long string missing values record found (record
## type 7, subtype 22), but ignored

DepressionT3 <- read.spss("StressStudy_T3short.sav", use.value.labels = FALSE, to.data.frame = TRUE)

## Warning in read.spss("StressStudy_T3short.sav", use.value.labels = FALSE, :
## StressStudy_T3short.sav: Unrecognized record type 7, subtype 10 encountered
## in system file

## Warning in read.spss("StressStudy_T3short.sav", use.value.labels = FALSE, :
## StressStudy_T3short.sav: Long string missing values record found (record
## type 7, subtype 22), but ignored

#Merge all three timepoints
DepressionMerge <- left_join(DepressionT1, DepressionT2, by = "ID")
DepressionMerge <- left_join(DepressionMerge, DepressionT3, by = "ID")

#Separatedates
DepressionMerge$T1_Date <- as.Date(DepressionMerge$X._created_at.x)
DepressionMerge$T2_Date <- as.Date(DepressionMerge$X._created_at.y)
DepressionMerge$T3_Date <- as.Date(DepressionMerge$X._created_at)

#Make a shorter dataset
Depressionshort <- dplyr::select(DepressionMerge, ID, T1_Date, T2_Date, T3_Date, T1_BDI, T2_BDI, T3_BDI

```

```

#Converting into proper format
Depressionshort$q_gender_0_score <- as.factor(Depressionshort$q_gender_0_score)
Depressionshort$q_ethnicity_score <- as.factor(Depressionshort$q_ethnicity_score)
Depressionshort$q_sexuality_score <- as.factor(Depressionshort$q_sexuality_score)
Depressionshort$q_age_0 <- as.character(Depressionshort$q_age_0)
Depressionshort$q_age_0 <- as.numeric(Depressionshort$q_age_0)
Depressionshort <- as.tibble(Depressionshort)

#Make date variables
Depressionshort$T1_Day <- difftime(Depressionshort$T1_Date, Depressionshort$T1_Date, units = c("days"))
Depressionshort$T2_Day <- difftime(Depressionshort$T2_Date, Depressionshort$T1_Date, units = c("days"))
Depressionshort$T3_Day <- difftime(Depressionshort$T3_Date, Depressionshort$T1_Date, units = c("days"))

Depressionwide <- select(Depressionshort, everything())
#Selecting only people who completed all three time points
Depressionshort <- subset(Depressionshort, T1_BDI >= "0" & T2_BDI >= "0" & T3_BDI >= "0")

#Convert DepressionShort to longform
Depressionlong <- gather(Depressionshort, Var, Val, select = c("T1_ACSS", "T1_BDI", "T1_SIAS", "T2_ACSS", "T2_BDI", "T2_SIAS", "T3_ACSS", "T3_BDI", "T3_SIAS"))

## Warning: attributes are not identical across measure variables;
## they will be dropped

Depressionlong <- arrange(Depressionlong, .by_group = ID)
Depressionlong <- separate(Depressionlong, Var, c("Timepoint", "Variable"), sep = "_")
Depressionlong <- spread(Depressionlong, Variable, Val)
Depressionlong <- as.tibble(Depressionlong)
#Removing someone with bad T2 data
Depressionlong <- Depressionlong[-c(784:786), ]

Depressionlong <- dplyr::select(Depressionlong, -Timepoint)

#Create a wave variable...repeat 1,2,3 for each person
n<- c(1, 2, 3)
Depressionlong$Wave <- rep(n, 130)
#Make a shorter dataset to work with.
Depressionlong <- dplyr::select(Depressionlong, ID, Wave, Day, q_gender_0_score, q_age_0, q_ethnicity_score, q_sexuality_score)

#Make Wave variables
Depressionshort$Wave1 <- rep(1, 130)
Depressionshort$Wave2 <- rep(2, 130)
Depressionshort$Wave3 <- rep(3, 130)

#Shorten and rearrange Depressionshort
Depressionshort <- dplyr::select(Depressionshort, ID, Wave1, Wave2, Wave3, T1_Day, T2_Day, T3_Day, q_gender_0_score, q_age_0, q_ethnicity_score, q_sexuality_score)

#Clean up
rm(DepressionT1, DepressionT2, DepressionT3)

```

VISUALIZING THE DATA, DESCRIPTIVE OUTPUT INLCUDED

- 1) Visualize the data Let's first show the study variables with the entire dataset. Need to finish up the full dataset.

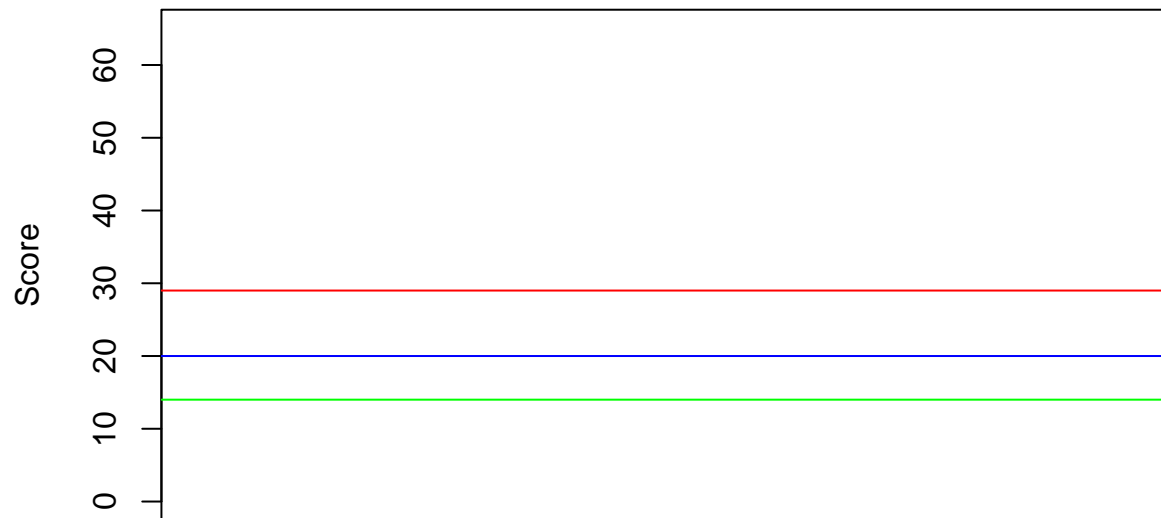
```
abline(h= c(14, 20, 29), col = c("green", "blue", "red"), boxplot(DepressionMerge$T1_BDITot,
  main = "Boxplot of Depression scores",
  xlab = "Depression",
  ylab = "Score",
  ylim = c(0, 65)))
```

Warning in is.na(x): is.na() applied to non-(list or vector) of type 'NULL'

Warning in is.na(x): is.na() applied to non-(list or vector) of type 'NULL'

Warning in is.na(x): is.na() applied to non-(list or vector) of type 'NULL'

Boxplot of Depression scores



Depression

```
boxplot(DepressionMerge$T1_ACSSTot,
  main = "Boxplot of Acquired Capability for Suicide scores",
  xlab = "Fear of Death",
  ylab = "Score",
  ylim = c(0, 30))
```

Warning in is.na(x): is.na() applied to non-(list or vector) of type 'NULL'

Warning in is.na(x): is.na() applied to non-(list or vector) of type 'NULL'

Warning in is.na(x): is.na() applied to non-(list or vector) of type 'NULL'

Boxplot of Acquired Capability for Suicide scores



Fear of Death

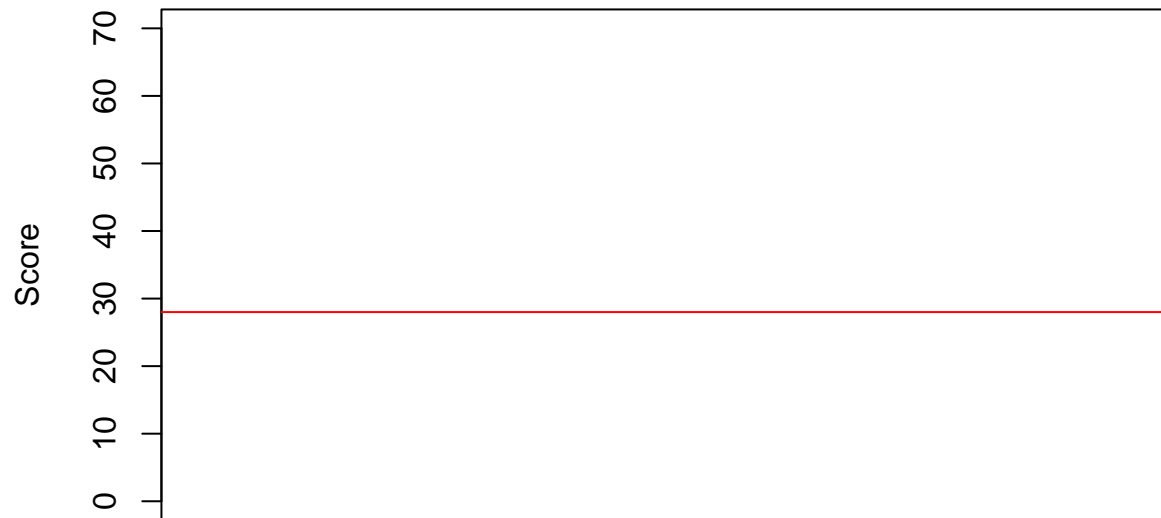
```
abline(h=28, col = "red", boxplot(DepressionMerge$T1_SIASTot, ylim = c(0, 70),
  main = "Boxplot of Social Anxiety scores",
  xlab = "Social Anxiety",
  ylab = "Score"))
```

```
## Warning in is.na(x): is.na() applied to non-(list or vector) of type 'NULL'
```

```
## Warning in is.na(x): is.na() applied to non-(list or vector) of type 'NULL'
```

```
## Warning in is.na(x): is.na() applied to non-(list or vector) of type 'NULL'
```

Boxplot of Social Anxiety scores



Social Anxiety

```
boxplot(DepressionMerge$T1_BSSTot,  
        main = "Boxplot of Suicide scores",  
        xlab = "Suicide",  
        ylab = "Score",  
        ylim = c(0, 20))
```

```
## Warning in is.na(x): is.na() applied to non-(list or vector) of type 'NULL'
```

```
## Warning in is.na(x): is.na() applied to non-(list or vector) of type 'NULL'
```

```
## Warning in is.na(x): is.na() applied to non-(list or vector) of type 'NULL'
```

Boxplot of Suicide scores



Suicide

We see

there's a pretty good range of scores in the full sample. For reference: 12% have mild depression, 11% have moderate depression, 8% have severe depression. 32% have clinical levels of social interaction anxiety.

Now lets look at the descriptives for the subset of those who completed all three timepoints

```
abline(h= c(14, 20, 29), col = c("green", "blue", "red"), boxplot(Depressionshort$T1_BDITot,
  main = "Boxplot of Depression scores",
  xlab = "Depression",
  ylab = "Score",
  ylim = c(0, 65)))
```

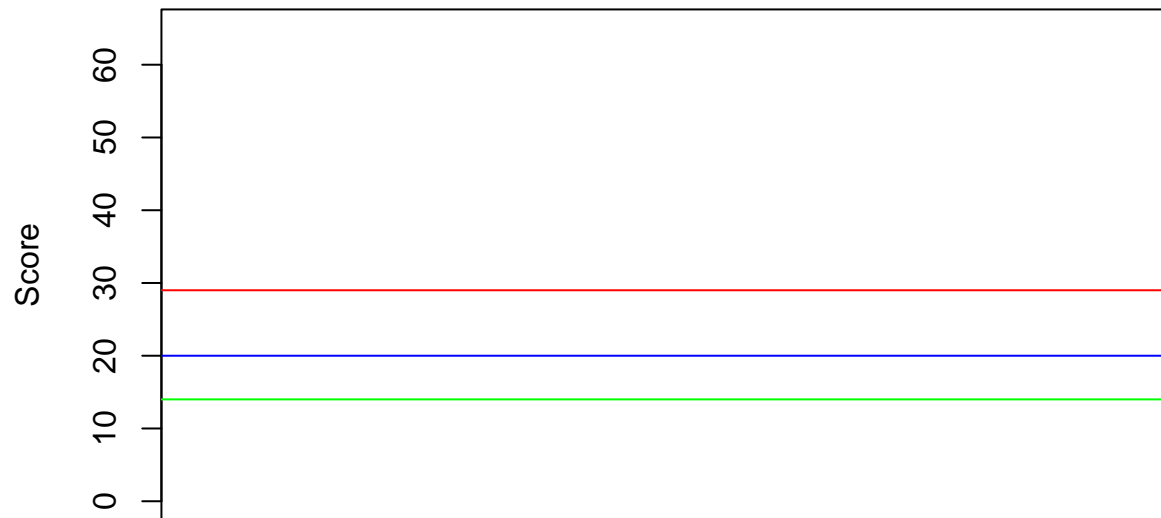
```
## Warning: Unknown or uninitialised column: 'T1_BDITot'.
```

```
## Warning in is.na(x): is.na() applied to non-(list or vector) of type 'NULL'
```

```
## Warning in is.na(x): is.na() applied to non-(list or vector) of type 'NULL'
```

```
## Warning in is.na(x): is.na() applied to non-(list or vector) of type 'NULL'
```

Boxplot of Depression scores



Depression

```
boxplot(Depressionshort$T1_ACSSTot,  
        main = "Boxplot of Acquired Capability for Suicide scores",  
        xlab = "Fear of Death",  
        ylab = "Score",  
        ylim = c(0, 20))
```

```
## Warning: Unknown or uninitialised column: 'T1_ACSSTot'.
```

```
## Warning: is.na() applied to non-(list or vector) of type 'NULL'
```

```
## Warning: is.na() applied to non-(list or vector) of type 'NULL'
```

```
## Warning: is.na() applied to non-(list or vector) of type 'NULL'
```


Boxplot of Acquired Capability for Suicide scores



Fear of Death

```
abline(h=28, col = "red", boxplot(Depressionshort$T1_SIASTot,  
  main = "Boxplot of Social Anxiety scores",  
  xlab = "Social Anxiety",  
  ylab = "Score",  
  ylim = c(0, 70)))
```

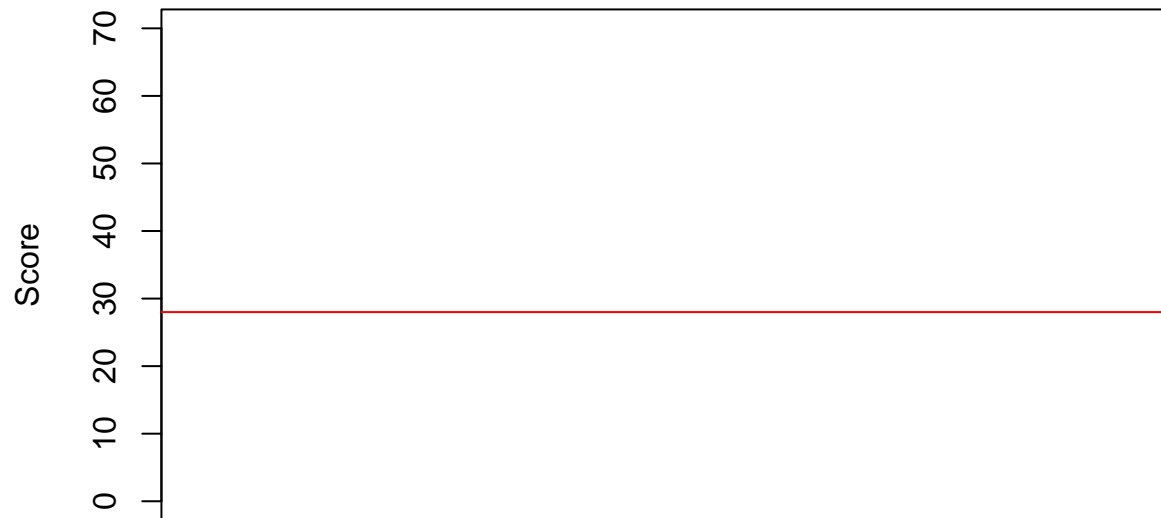
```
## Warning: Unknown or uninitialised column: 'T1_SIASTot'.
```

```
## Warning: is.na() applied to non-(list or vector) of type 'NULL'
```

```
## Warning: is.na() applied to non-(list or vector) of type 'NULL'
```

```
## Warning: is.na() applied to non-(list or vector) of type 'NULL'
```

Boxplot of Social Anxiety scores



Social Anxiety

```
boxplot(Depressionshort$T1_BSSTot,  
        main = "Boxplot of Suicide scores",  
        xlab = "Suicide",  
        ylab = "Score",  
        ylim = c(0, 30))
```

```
## Warning: Unknown or uninitialised column: 'T1_BSSTot'.
```

```
## Warning: is.na() applied to non-(list or vector) of type 'NULL'
```

```
## Warning: is.na() applied to non-(list or vector) of type 'NULL'
```

```
## Warning: is.na() applied to non-(list or vector) of type 'NULL'
```

Boxplot of Suicide scores



Suicide

Simi-

lar descriptives.

- 1) Run linear models on all of your subjects (a basic regression). What is the average intercept, the average slope?

```
LinModel <- lm(BDI ~ Day, data = Depressionlong)
summary(LinModel)
```

```
##
## Call:
## lm(formula = BDI ~ Day, data = Depressionlong)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -10.140   -6.140   -3.134    2.933   35.587
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  10.14005    0.65912   15.384 < 2e-16 ***
## Day         -0.09317    0.02153   -4.328 1.92e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 8.647 on 388 degrees of freedom
## Multiple R-squared:  0.04605,    Adjusted R-squared:  0.0436
## F-statistic: 18.73 on 1 and 388 DF,  p-value: 1.917e-05
rse(LinModel)
```

```
## [1] 8.646629
```

The average intercept is: 10.14 BDI score (at baseline) The average slope is: -.09 Residual standard error = 8.65

- 2) Now run a mlm/lmer model with only a random intercept. What is the ICC? What does residual variance look like compared to linear model? Create a graph to show this effect.
- 3) Run MLM models: Running an empty (intercept only) model.

```
IntOnly <- lmer(BDI ~ 1 + (1 | ID), data = Depressionlong)
summary(IntOnly)
```

```
## Linear mixed model fit by REML ['lmerMod']
## Formula: BDI ~ 1 + (1 | ID)
## Data: Depressionlong
##
## REML criterion at convergence: 2672.1
##
## Scaled residuals:
##      Min       1Q   Median       3Q      Max
## -3.4186 -0.4155 -0.1431  0.2909  3.8962
##
## Random effects:
## Groups Name Variance Std.Dev.
## ID      (Intercept) 46.76   6.838
## Residual              31.65   5.626
## Number of obs: 390, groups: ID, 130
##
## Fixed effects:
##              Estimate Std. Error t value
## (Intercept)    8.008      0.664    12.06
```

```
icc(IntOnly)
```

```
## Linear mixed model
## Family: gaussian (identity)
## Formula: BDI ~ 1 + (1 | ID)
##
## ICC (ID): 0.596382
```

```
rse(IntOnly)
```

```
## [1] 4.817109
```

```
IntOnly.ci <- confint(IntOnly, level = .95, oldNames = F)
```

```
## Computing profile confidence intervals ...
```

```
## Warning in if (parm == "theta_") {: the condition has length > 1 and only
## the first element will be used
```

```
## Warning in if (parm == "beta_") {: the condition has length > 1 and only
## the first element will be used
```

```
IntOnly.ci
```

```
##              2.5 %   97.5 %
## sd_(Intercept)|ID 5.857098 7.924990
## sigma            5.174753 6.146196
## (Intercept)      6.701688 9.313696
```

```
summary1 <- broom::tidy(IntOnly)
summary1
```

```
##               term estimate std.error statistic    group
## 1      (Intercept) 8.007692 0.6639825  12.06009    fixed
## 2      sd_(Intercept).ID 6.838413      NA      NA      ID
## 3 sd_Observation.Residual 5.625719      NA      NA Residual

#Plotting residuals
IntOnly.aug <- abs(broom::augment(IntOnly)[,c(4)])

## Warning: Deprecated: please use `purrr::possibly()` instead
## Warning: Deprecated: please use `purrr::possibly()` instead
## Warning: Deprecated: please use `purrr::possibly()` instead
## Warning: Deprecated: please use `purrr::possibly()` instead
## Warning: Deprecated: please use `purrr::possibly()` instead
head(IntOnly.aug)

## [1]  1.289892  1.289892  1.289892 18.550788 10.449212  6.449212

mean.mlm = mean(IntOnly.aug)
sd.mlm = sd(IntOnly.aug)

LinMod.aug <- abs(broom::augment(LinModel)[,5])

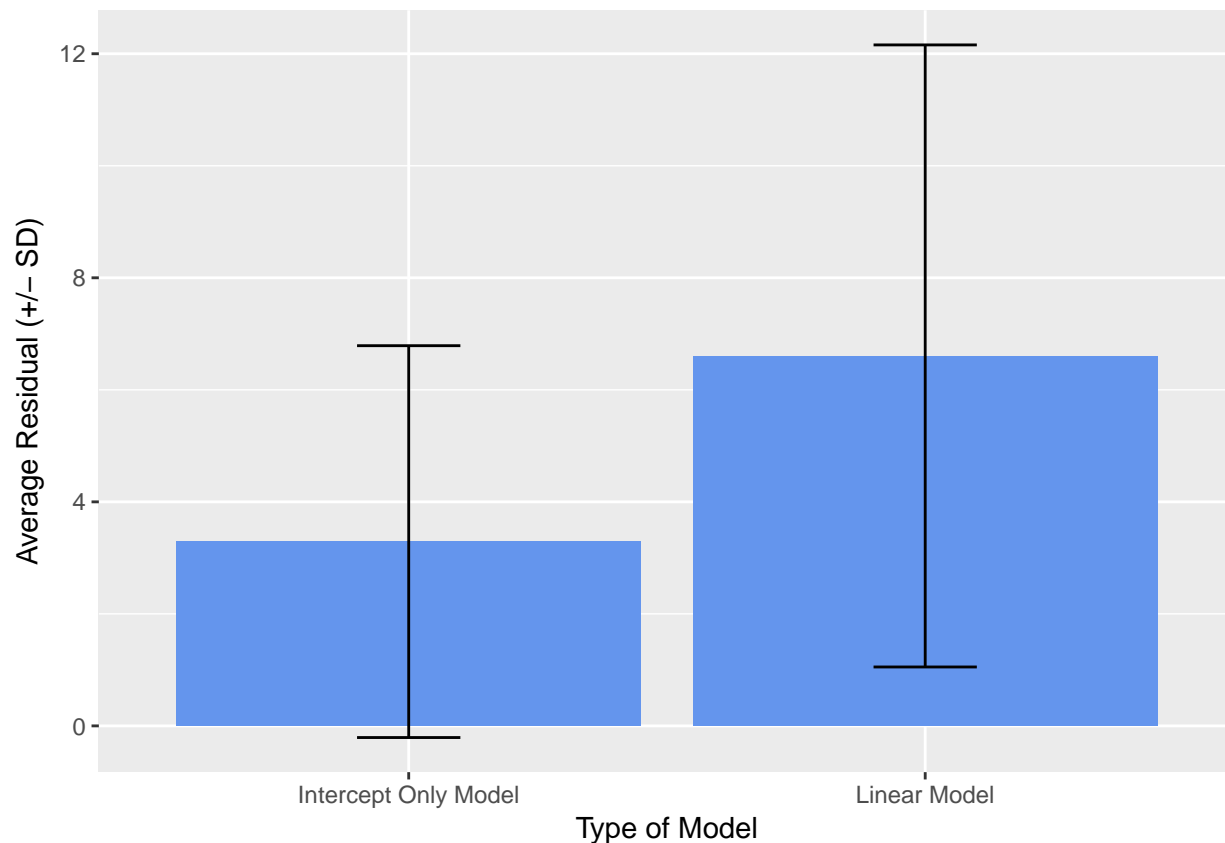
## Warning: Deprecated: please use `purrr::possibly()` instead
## Warning: Deprecated: please use `purrr::possibly()` instead
## Warning: Deprecated: please use `purrr::possibly()` instead
## Warning: Deprecated: please use `purrr::possibly()` instead
## Warning: Deprecated: please use `purrr::possibly()` instead
head(LinMod.aug)

## [1]  9.140049  7.090349  4.947481 18.859951  8.183517  2.226986

mean.lin <- mean(LinMod.aug)
sd.lin <- sd(LinMod.aug)

dataframe <- data.frame("Type" = c("Linear Model", "Intercept Only Model"),
                        "Mean" = c(mean.lin, mean.mlm), "SD" = c(sd.lin, sd.mlm))

ResidPlot <- ggplot(dataframe, aes(Type, Mean)) +
  geom_col(fill = "cornflowerblue") +
  geom_errorbar(aes(ymin = Mean - SD, ymax = Mean + SD), width=0.2)
ResidPlot + labs(y="Average Residual (+/- SD)", x = "Type of Model")
```



Average intercept = 8.008 (at baseline) Slope = 0 (empty model) ICC = .60. Looks like there's roughly an equal distribution between the within and between person levels (slightly more on the between-person level). Residual variance = 31.65 Residual standard error = 4.82

The residual standard error for the intercept only model is about half of the residual variance of the linear model. This is interesting because the Linear model uses time (Day) as a predictor, whereas this empty, intercept only model does not account for time, but still does a better job of explaining more variance.

- 3) Introduce a fixed slope term. What is the difference in terms of the fixed effects estimates between this estimate and the previous? Of the residual standard error? Create a graph to show both fixed effects estimates and the CIs around them.

Running a fixed slopes model.

```
library(MuMIn)

## Warning: package 'MuMIn' was built under R version 3.4.2
FixSlopes <- lmer(BDI ~ Day + (1 | ID), data = Depressionlong)
summary(FixSlopes)

## Linear mixed model fit by REML ['lmerMod']
## Formula: BDI ~ Day + (1 | ID)
## Data: Depressionlong
##
## REML criterion at convergence: 2627.7
##
## Scaled residuals:
##      Min       1Q   Median       3Q      Max
## -3.5218 -0.4515 -0.0954  0.3479  3.7281
```

```

##
## Random effects:
##   Groups   Name                Variance Std.Dev.
##   ID        (Intercept) 48.88      6.991
##   Residual                26.03      5.102
## Number of obs: 390, groups: ID, 130
##
## Fixed effects:
##               Estimate Std. Error t value
## (Intercept) 10.27890    0.73074 14.066
## Day         -0.09923    0.01320 -7.518
##
## Correlation of Fixed Effects:
##      (Intr)
## Day -0.413
rse(FixSlopes)

## [1] 4.333794
anova(IntOnly, FixSlopes)

## refitting model(s) with ML (instead of REML)
## Data: Depressionlong
## Models:
## IntOnly: BDI ~ 1 + (1 | ID)
## FixSlopes: BDI ~ Day + (1 | ID)
##           Df      AIC      BIC logLik deviance Chisq Chi Df Pr(>Chisq)
## IntOnly    3 2679.1 2691.0 -1336.6 2673.1
## FixSlopes  4 2629.9 2645.8 -1311.0 2621.9 51.225      1 8.238e-13 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
FixSlopes.ci <- confint(FixSlopes, level = .95, oldNames = F)

## Computing profile confidence intervals ...

## Warning in if (parm == "theta_") {: the condition has length > 1 and only
## the first element will be used

## Warning in if (parm == "beta_") {: the condition has length > 1 and only
## the first element will be used
FixSlopes.ci

##               2.5 %      97.5 %
## sd_(Intercept)|ID 6.0373075 8.06121964
## sigma           4.6845057 5.56394438
## (Intercept)      8.8447432 11.71401436
## Day              -0.1251292 -0.07329785
summary1 <- broom::tidy(FixSlopes)
r.squaredGLMM(FixSlopes)

##           R2m           R2c
## 0.05170327 0.67042396
#Graphing fixed effects
dataframe <- data.frame("Type" = c("Linear Model_Intercept", "MLMod_Intercept", "MLMod_Day"),

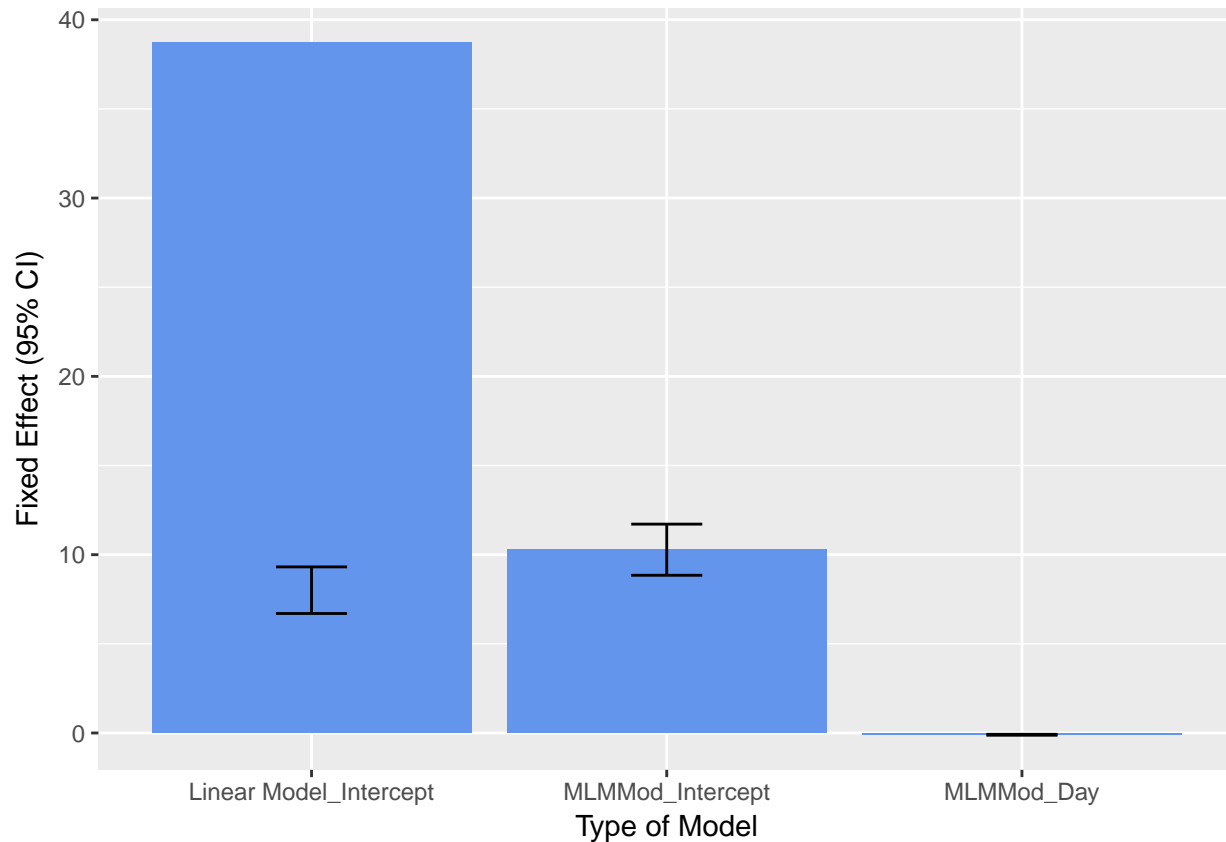
```

```

"Mean" = c(38.72, summary1[1,2], summary1[2,2]),
"CILower" = c(IntOnly.ci[3,1], FixSlopes.ci[3,1], FixSlopes.ci[4,1]),
"CILower" = c(IntOnly.ci[3,2], FixSlopes.ci[3,2], FixSlopes.ci[4,2]))

FixedPlot <- ggplot(dataframe, aes(x = reorder(Type, -Mean), Mean)) +
  geom_col(fill = "cornflowerblue") +
  geom_errorbar(aes(ymin = CILower, ymax = CIUpper), width=0.2)
FixedPlot + labs(y="Fixed Effect (95% CI)", x = "Type of Model")

```



Average intercept = 10.28 (At baseline) Slope = -0.10 depression scores/day Residual variance = 26.03
Residual standard error = 4.33

The intercept (i.e., depression at baseline) score is larger in the fixed effects model than in the intercept only. Additionally, the residual variance is smaller as compared the intercept only model. This is in keeping with the fact that we have added more predictors to the model, which should help measure and capture more variance. This is supported by the results of the ANOVA, which suggests that the fixed effects model exhibits better fit.

Furthermore, it looks like intercept and slope are moderately correlated. That is, we see that people start out at different levels of depression (significant, positive intercept), and they tend to decrease over time (significant, negative slope). (Unlikely that these were seasonal or semester effects as the study ran over the entire year).

- 4) Run an additional model with a random slope. How does this change compare to the previous model? Should you keep the random slope or not?
- 5) Interpret the correlation between the slope and the intercept.

Now let's try to model a random slope


```

RandSlope <- lmer(BDI ~ Day + (Day|ID), data = Depressionlong)

## Warning in checkConv(attr(opt, "derivs"), opt$par, ctrl = control
## $checkConv, : Model failed to converge with max|grad| = 0.00297997 (tol =
## 0.002, component 1)

summary(RandSlope)

## Linear mixed model fit by REML ['lmerMod']
## Formula: BDI ~ Day + (Day | ID)
## Data: Depressionlong
##
## REML criterion at convergence: 2610.6
##
## Scaled residuals:
##      Min       1Q   Median       3Q      Max
## -3.2490 -0.3907 -0.1057  0.3332  3.2699
##
## Random effects:
## Groups Name Variance Std.Dev. Corr
## ID      (Intercept) 71.907371 8.47982
## Day      Day      0.007298 0.08543 -0.78
## Residual      22.021751 4.69273
## Number of obs: 390, groups: ID, 130
##
## Fixed effects:
##              Estimate Std. Error t value
## (Intercept) 10.31919    0.82912 12.446
## Day         -0.10169    0.01445 -7.036
##
## Correlation of Fixed Effects:
##      (Intr)
## Day -0.649
## convergence code: 0
## Model failed to converge with max|grad| = 0.00297997 (tol = 0.002, component 1)

anova(FixSlopes, RandSlope)

## refitting model(s) with ML (instead of REML)
## Data: Depressionlong
## Models:
## FixSlopes: BDI ~ Day + (1 | ID)
## RandSlope: BDI ~ Day + (Day | ID)
##      Df    AIC    BIC logLik deviance Chisq Chi Df Pr(>Chisq)
## FixSlopes  4 2629.9 2645.8 -1311.0 2621.9
## RandSlope  6 2616.8 2640.6 -1302.4 2604.8 17.065      2 0.0001969 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

```

With the random slope model, we see a *stronger* correlation between intercept and slope. This means that the higher you start out in depression, the more you are expected to decrease over time (i.e., regression to the mean). We also see in this model that the intercept is still positive and statistically significant. When we model a random slope that accounts for individual variability, we see that people on average tend to decrease in their level of depression and this is statistically significant. Comparing the random slopes model to the fixed slopes model, we see that the Fixed slopes model exhibits better fit. Additionally, it looks like we ran

into convergence issues with the random slopes model.

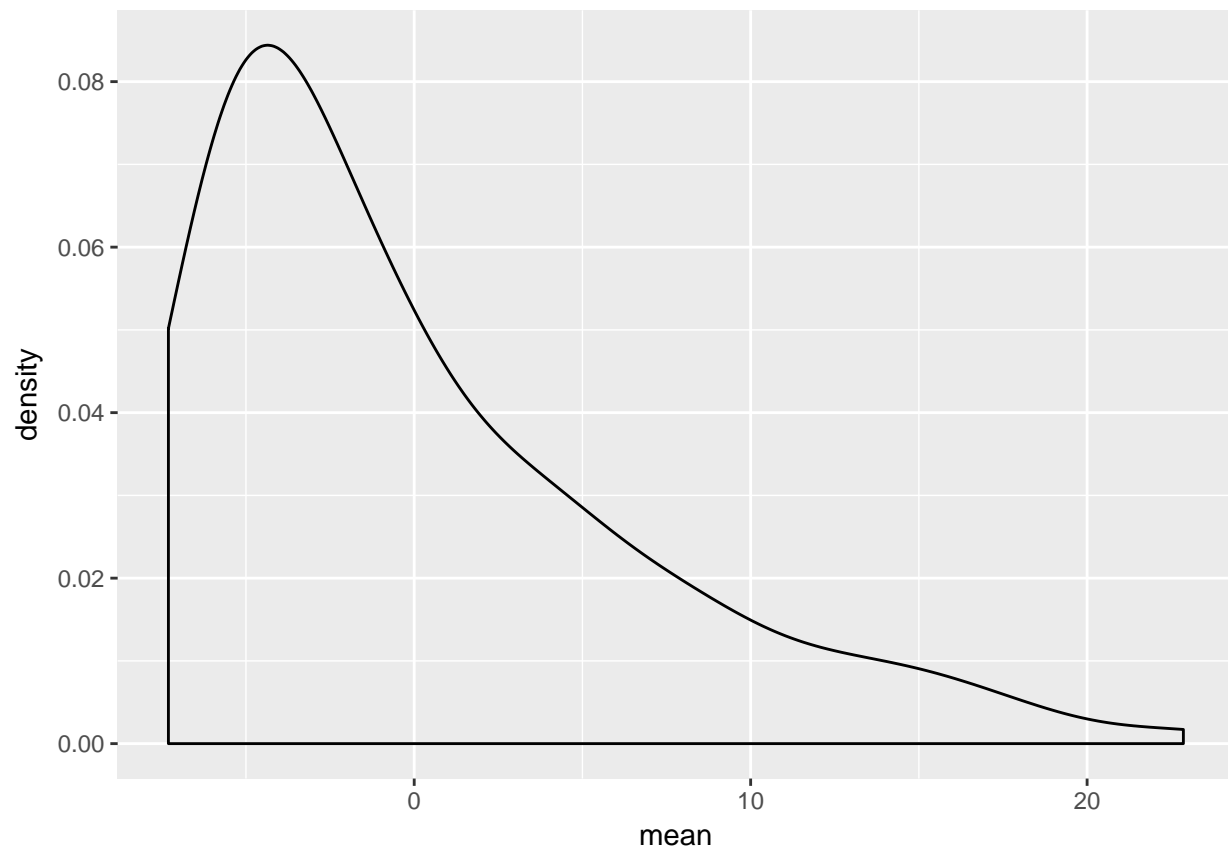
6. Create a density plot of the random effects from your final model.

```
library(merTools)

## Loading required package: arm
## Loading required package: MASS
##
## Attaching package: 'MASS'
## The following object is masked from 'package:dplyr':
##
##      select
##
## arm (Version 1.9-3, built: 2016-11-21)
## Working directory is /Users/marilynpiccirillo/1-descriptives-and-graphs-piccirillom
re.sim <- REsim(FixSlopes)
head(re.sim)

##   groupFctr groupID      term      mean      median      sd
## 1      ID    1350 (Intercept) -5.9979466 -5.8598704 2.724990
## 2      ID    1354 (Intercept)  2.2410955  2.4878656 2.623114
## 3      ID    1358 (Intercept)  6.0513401  5.9599882 2.908588
## 4      ID    1364 (Intercept) -4.3280662 -4.6155357 2.877545
## 5      ID    1376 (Intercept) -6.7095837 -6.9524215 2.657779
## 6      ID    1378 (Intercept) -0.5196041 -0.2460209 2.802529

#Density Intercept
graph1 <- re.sim %>%
  filter(term == "(Intercept)")
ggplot(graph1, aes(mean)) +
  geom_density()
```



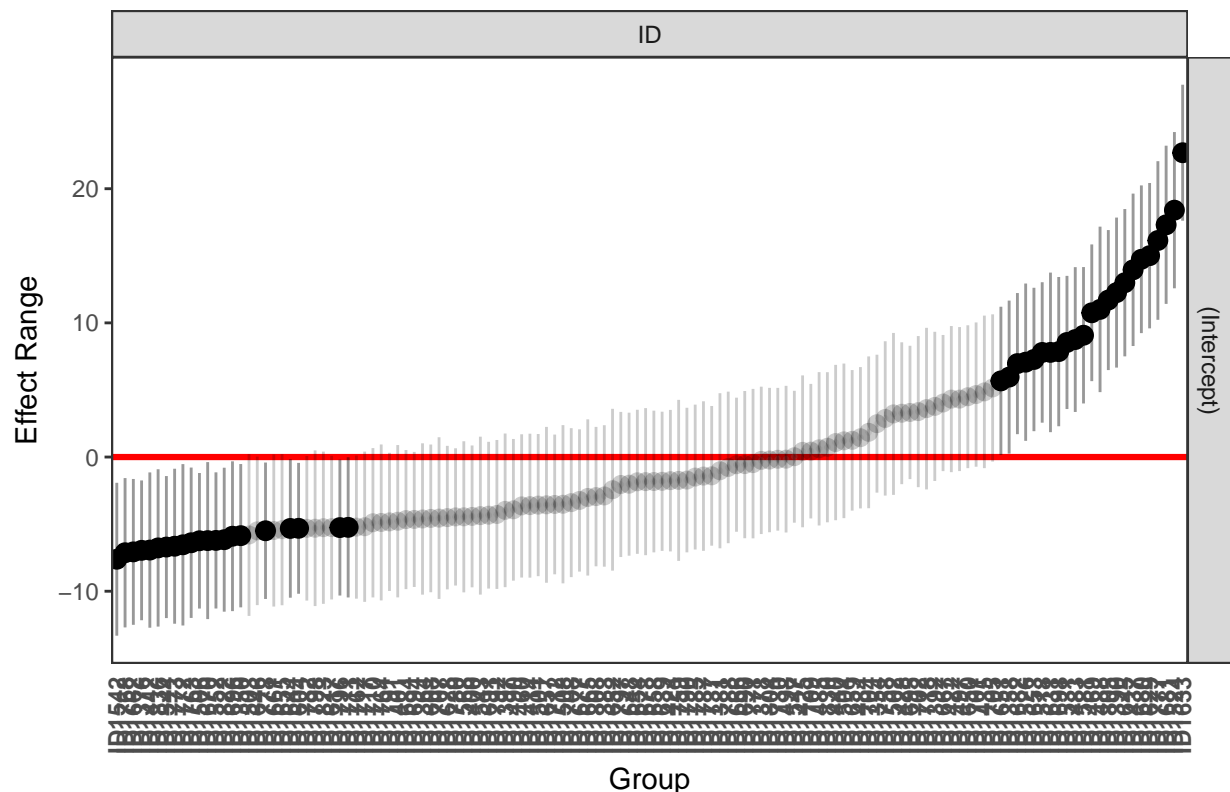
```
#Density Day  
graph2 <- re.sim %>%  
  filter(term == "Day")  
ggplot(graph2, aes(mean)) +  
  geom_density()
```



7. Create a caterpillar plot of the random effects. Is there any person that seems odd in terms of a large standard errors around intercept and slope estimates?

```
graph3 <- plotREsim(re.sim, labs = T)
graph3
```

Effect Ranges



The caterpillar plot suggests that there may be a few individuals whose depression scores are significantly different than the group average.

8. Create a plot of the trajectory, along with a spaghetti plot of each person's individual slope. Set the alpha level (transparency) on the individual slopes to make them easier to see.

```
Predict <- predictInterval(merMod = IntOnly, newdata = Depressionlong, level = 0.9, n.sims = 100,
                           stat = "median", include.resid.var = TRUE)

## Warning in predictInterval(merMod = IntOnly, newdata = Depressionlong, level = 0.9, : newdata is tbl
##               coerced to a data.frame

head(Predict)

##      fit      upr      lwr
## 1  1.483964 12.16680 -7.8103357
## 2  2.178596 14.63820 -8.1630198
## 3  3.128564 10.33119 -7.6956120
## 4 10.904725 22.03305 -0.2537284
## 5  9.872702 19.12052  1.0229465
## 6 10.008696 21.58746 -0.8410469

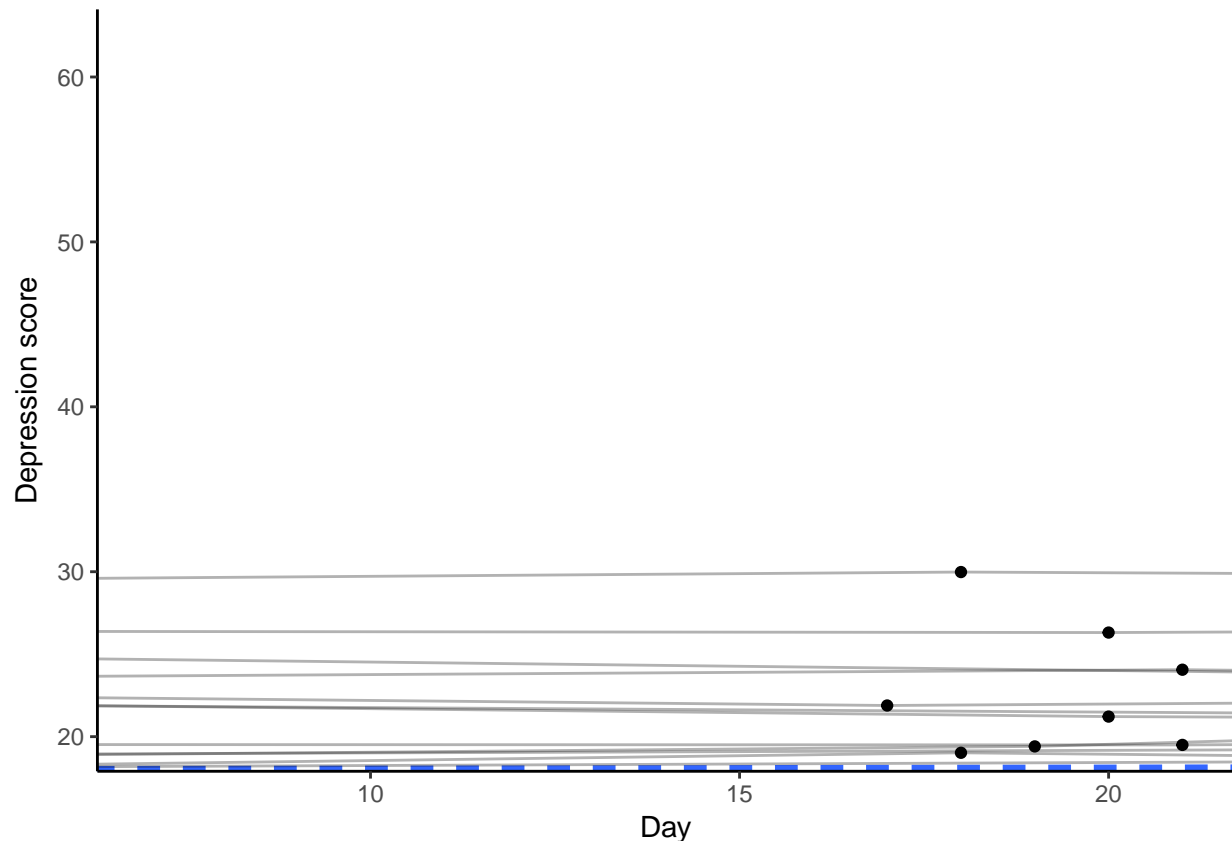
Depressionlong$fit <- Predict$fit
Depressionlong$upr <- Predict$upr
Depressionlong$lwr <- Predict$lwr

#Spaghetti plot 1
ggplot(aes(x=Day, y=fit), data=Depressionlong) +
  geom_point() +
```

```

stat_smooth(method = lm, se = F) +
labs(x="Day", y="Depression score") + theme_bw() +
stat_smooth(aes(x = Day, y = upr), method = lm, se = F, alpha = .3, linetype = "dashed") +
stat_smooth(aes(x = Day, y = lwr), method = lm, se = F, alpha = .3, linetype = "dashed")+
geom_line(data = Depressionlong, aes(x = Day, y = fit, group=ID), alpha = .3) +
coord_cartesian(xlim = c(7,21), ylim = c(20,62)) +
theme_classic()

```



```

#Spaghetti plot 2
ggplot(aes(x=Day, y=fit), data=Depressionlong) +
  #geom_point() +
  stat_smooth(method = lm, se = F, aes(x = Day, y = fit, group=ID), alpha = .3,
    color = "lightsteelblue3") +
  stat_smooth(method = lm, se = F, color = "royalblue4") +
  labs(x="Day", y="Depression score") + theme_bw() +
  stat_smooth(aes(x = Day, y = upr), method = lm, se = F, alpha = .3, linetype = "dashed",
    color = "royalblue4") +
  stat_smooth(aes(x = Day, y = lwr), method = lm, se = F, alpha = .3, linetype = "dashed",
    color = "royalblue4")+
  coord_cartesian(xlim = c(7,21), ylim = c(20,62)) +
  theme_classic()

```

