

# ConfidenceIntervals\_LMER.R

*callinswitzer*

*Sat Feb 18 11:08:04 2017*

```
# Callin Switzer  
# 17 Feb 2017  
# LMER prediction intervals and CI's
```

```
library(lme4)
```

```
## Loading required package: Matrix
```

```
set.seed(271828)
```

```
data(sleepstudy)
```

```
library(magrittr)
```

```
sleepstudy %>% head
```

```
##   Reaction Days Subject  
## 1 249.5600    0    308  
## 2 258.7047    1    308  
## 3 250.8006    2    308  
## 4 321.4398    3    308  
## 5 356.8519    4    308  
## 6 414.6901    5    308
```

```
fm1 <- lmer(Reaction ~ Days + (1|Subject), data=sleepstudy)
```

```
summary(fm1)
```

```
## Linear mixed model fit by REML ['lmerMod']
```

```
## Formula: Reaction ~ Days + (1 | Subject)
```

```
## Data: sleepstudy
```

```
##
```

```
## REML criterion at convergence: 1786.5
```

```
##
```

```
## Scaled residuals:
```

```
##      Min       1Q   Median       3Q      Max
```

```
## -3.2257 -0.5529  0.0109  0.5188  4.2506
```

```
##
```

```
## Random effects:
```

```
## Groups   Name      Variance Std.Dev.
```

```
## Subject (Intercept) 1378.2   37.12
```

```
## Residual              960.5   30.99
```

```
## Number of obs: 180, groups: Subject, 18
```

```
##
```

```
## Fixed effects:
```

```
##              Estimate Std. Error t value
```

```
## (Intercept) 251.4051     9.7467   25.79
```

```
## Days         10.4673     0.8042   13.02
```

```
##
```

```
## Correlation of Fixed Effects:
```

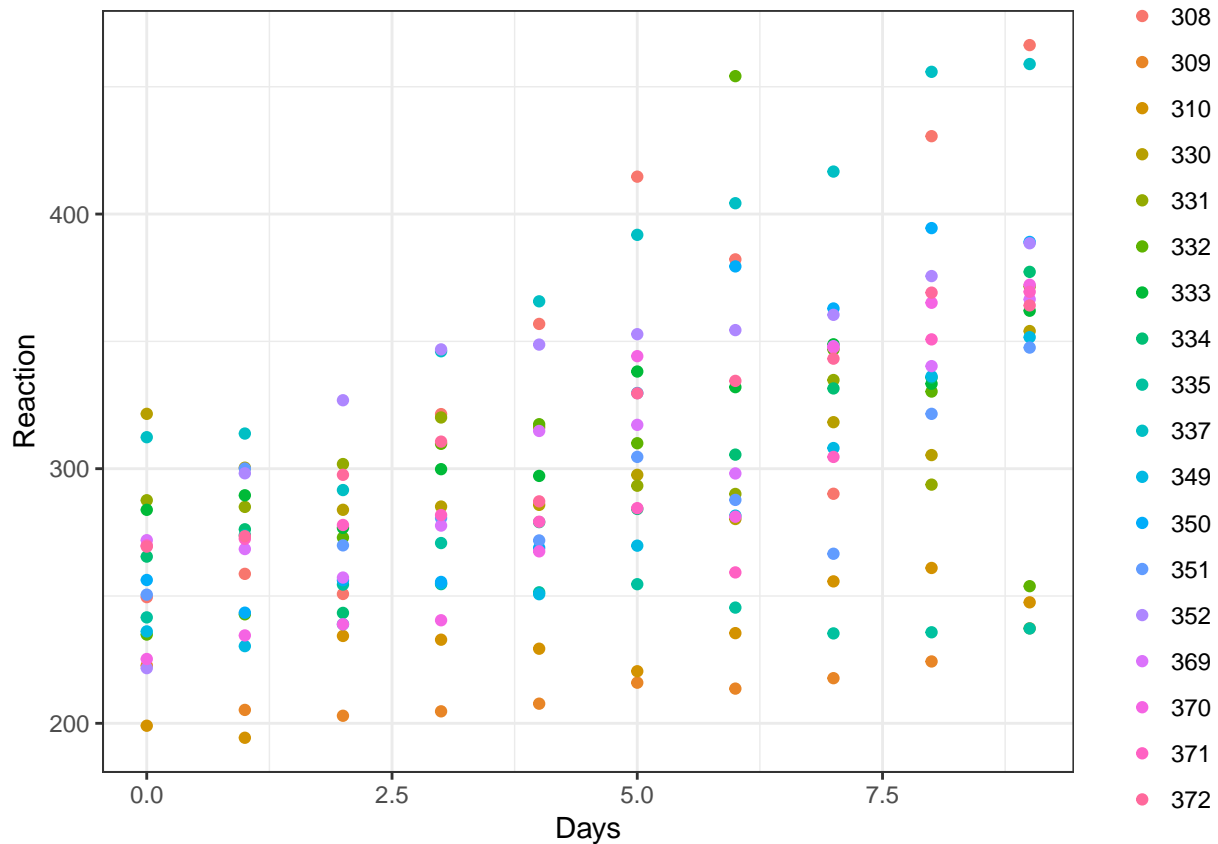
```
##      (Intr)
```

```
## Days -0.371
```

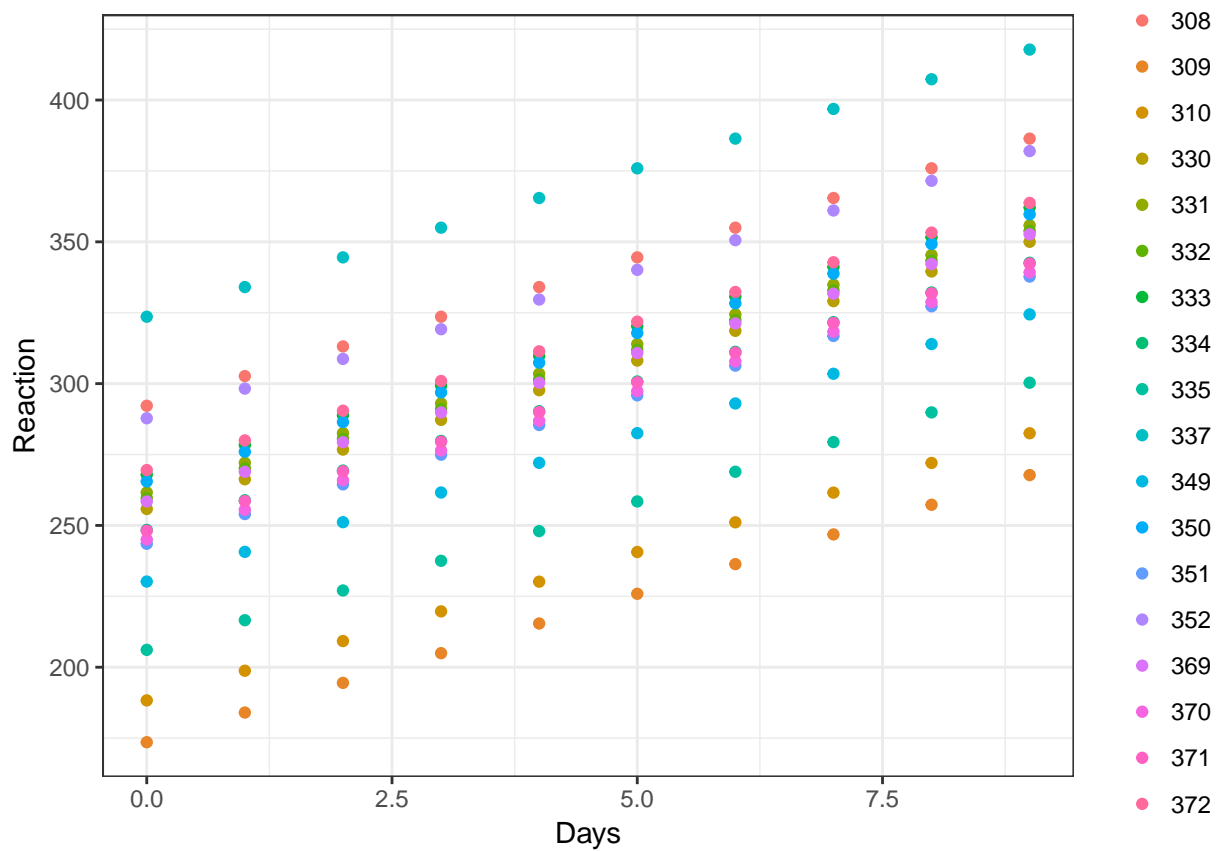
```
library(ggplot2)
theme_set(theme_bw())

pp = data.frame(preds = predict(fm1), Days = sleepstudy$Days, subj = sleepstudy$Subject)

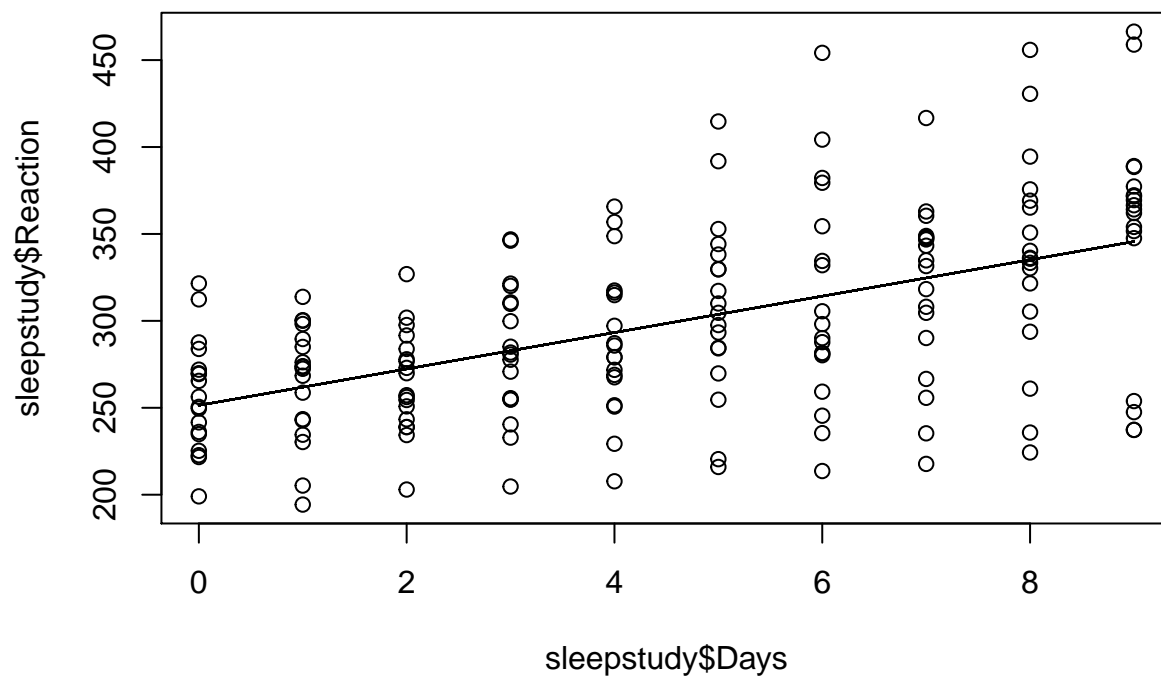
# look at data
ggplot(sleepstudy, aes(x = Days, y = Reaction, color = Subject)) +
  geom_point()
```



```
ggplot(sleepstudy, aes(x = Days, y = Reaction, color = Subject)) +
  # geom_point() +
  # geom_line() +
  geom_point(data = pp, aes(x = Days, y = preds, color = subj))
```



```
# base R plot
plot(sleepstudy$Reaction ~ sleepstudy$Days)
lines(predict(fm1, re.form = NA), x = sleepstudy$Days) # plot fitted line
curve(251.41 + x * 10.47, 0, 9, add = TRUE) # how that curve is predicted
```



```

# construct prediction intervals
mySumm <- function(.) {
  predict(., newdata=sleepstudy, re.form=NULL)
}

####Collapse bootstrap into median, 95% Prediction Interval
sumBoot <- function(merBoot) {
  return(
    data.frame(fit = apply(merBoot$t, 2, function(x) as.numeric(quantile(x, probs=.5, na.rm=TRUE))),
              lwr = apply(merBoot$t, 2, function(x) as.numeric(quantile(x, probs=.025, na.rm=TRUE))),
              upr = apply(merBoot$t, 2, function(x) as.numeric(quantile(x, probs=.975, na.rm=TRUE)))
    )
  )
}

##lme4::bootMer() two possible methods
# if use.u is FALSE, this generates new random effects and random errors
# if use.u is TRUE, then the levels of the random effects are essentially fixed
system.time(
  boot1 <- lme4::bootMer(fm1, mySumm, nsim=100, use.u=TRUE, type="parametric")
)

##      user  system elapsed
##    2.207    0.011    2.220

PI.boot1 <- sumBoot(boot1)
PI.boot1

```

```

##      fit      lwr      upr
## 1  288.3001 267.5070 310.9871
## 2  299.1074 279.0651 320.4913
## 3  309.5953 290.4834 330.0314
## 4  320.1899 301.9017 339.7453
## 5  330.0974 312.7747 350.0173
## 6  340.2590 322.7720 360.4192
## 7  350.9967 333.0110 370.8212
## 8  361.9927 343.5822 381.6316
## 9  373.1264 354.1245 392.9056
## 10 383.3192 363.9899 404.2857
## 11 176.6688 157.7402 195.8020
## 12 186.8679 169.0716 205.8989
## 13 197.5350 180.0026 215.4802
## 14 208.6561 191.4322 224.9253
## 15 219.8715 202.9047 234.6256
## 16 230.6491 214.3393 245.1002
## 17 241.3153 223.3718 255.5794
## 18 251.3618 233.4455 266.1625
## 19 261.7539 244.0932 277.4080
## 20 272.3784 253.7674 288.2375
## 21 193.2201 175.1668 210.6219
## 22 203.6072 186.0943 220.7970
## 23 214.1691 197.5753 231.4158
## 24 224.9733 208.0303 242.0531
## 25 235.7487 218.1489 252.4229
## 26 245.6412 228.5147 262.5898
## 27 256.5786 239.2232 272.7711

```

##	28	266.6263	249.5860	282.9987
##	29	277.1092	259.3193	293.3891
##	30	287.4248	269.0233	304.4786
##	31	254.7499	239.0565	273.2669
##	32	265.5015	250.7797	282.8862
##	33	275.7903	261.0759	292.0821
##	34	286.2828	270.4739	302.4223
##	35	296.3405	279.8720	312.9344
##	36	306.6726	290.5096	323.5746
##	37	317.3348	300.8435	334.5253
##	38	327.6206	310.1625	345.7376
##	39	338.3879	319.4815	356.3223
##	40	348.7476	328.8005	366.6201
##	41	259.0067	241.0224	277.5792
##	42	269.4507	251.8120	287.6495
##	43	280.0843	262.5563	297.7199
##	44	290.1645	273.2985	307.7902
##	45	299.8852	284.0407	318.2134
##	46	310.2110	294.7829	329.2526
##	47	320.7349	305.4277	340.8931
##	48	331.0932	316.7965	352.5336
##	49	341.6500	328.1284	364.1741
##	50	352.5576	338.6048	375.8145
##	51	259.8077	238.2848	276.3829
##	52	270.1147	249.8718	286.2998
##	53	280.1694	261.3770	295.9713
##	54	290.4325	272.8066	307.0056
##	55	299.8518	283.4208	317.8075
##	56	309.5894	293.8374	328.5662
##	57	320.5088	304.2540	339.3305
##	58	331.8108	314.6706	350.2500
##	59	342.8319	325.0872	361.1161
##	60	352.8105	335.5038	371.9822
##	61	268.3413	247.9731	286.5905
##	62	278.9496	259.7006	296.7411
##	63	289.2907	270.3639	307.3104
##	64	299.6632	280.5748	317.9751
##	65	309.9963	290.9484	328.6433
##	66	320.6799	300.8024	339.3689
##	67	331.0326	310.8000	350.0245
##	68	341.6392	321.2663	360.3527
##	69	352.4812	331.7325	370.6809
##	70	363.1539	342.1987	381.4497
##	71	249.5196	229.5428	270.0556
##	72	260.2728	240.4265	279.7847
##	73	271.0738	251.2210	289.9440
##	74	282.0897	261.9058	300.5346
##	75	292.1061	272.0375	310.7966
##	76	302.6068	282.1693	321.7342
##	77	312.8439	292.5520	332.7265
##	78	323.2546	303.4034	344.1578
##	79	333.8525	314.2036	355.8557
##	80	344.2965	324.6068	367.6369
##	81	208.3159	191.2491	227.7707

```

## 82 219.1335 202.3509 238.4284
## 83 230.2651 213.2922 249.0006
## 84 240.5291 223.8206 259.6701
## 85 251.4647 233.9136 270.3396
## 86 261.5716 243.9711 281.0090
## 87 272.3934 254.4393 291.6785
## 88 283.1726 265.8723 302.2262
## 89 293.8687 277.2590 312.8176
## 90 304.3623 288.0826 324.1837
## 91 318.9951 301.7731 339.5297
## 92 329.5627 313.0376 349.7623
## 93 339.4251 323.5217 359.4651
## 94 349.7322 333.9399 369.1007
## 95 360.5790 344.8570 378.7699
## 96 370.8446 355.7697 388.4390
## 97 381.2475 365.6303 398.5288
## 98 392.1945 375.2871 409.4154
## 99 402.3154 385.0405 420.8749
## 100 413.0854 394.8336 432.3406
## 101 231.4636 214.7921 249.9369
## 102 242.8140 226.0427 260.3789
## 103 253.1943 236.1938 271.5470
## 104 263.8517 245.8498 282.4033
## 105 274.2634 255.7696 292.0009
## 106 285.0289 265.7473 302.2107
## 107 295.5803 275.7150 313.2794
## 108 306.1098 285.5819 323.7638
## 109 316.6148 294.7820 334.5054
## 110 326.9494 304.3706 346.3601
## 111 264.2665 245.8244 282.6892
## 112 275.0375 257.0064 293.5117
## 113 285.2453 267.8093 304.3562
## 114 296.1515 278.2276 315.2041
## 115 307.0640 288.6242 326.0777
## 116 318.0558 299.0110 336.0692
## 117 328.3143 309.3977 346.0862
## 118 338.5712 319.7845 357.3023
## 119 348.9433 330.2166 368.0179
## 120 359.4170 341.0291 378.4413
## 121 242.5173 229.6342 257.6616
## 122 253.1551 240.4725 267.8094
## 123 263.8475 251.2946 278.1165
## 124 274.3750 261.8780 288.9509
## 125 285.4840 272.0306 299.7853
## 126 296.7419 281.9882 310.6197
## 127 307.1620 291.3019 321.4541
## 128 317.6068 300.6088 332.2885
## 129 328.4189 310.4347 343.1089
## 130 339.2119 320.0878 353.9574
## 131 284.0114 263.5722 308.0350
## 132 294.4056 275.3729 318.3704
## 133 304.5931 286.6922 328.1724
## 134 314.8044 297.8298 338.1803
## 135 325.4688 308.5160 348.2759

```

```

## 136 336.2458 319.6797 357.7583
## 137 346.5709 330.0959 366.7867
## 138 357.2723 340.8240 376.5192
## 139 368.1773 351.5520 386.8881
## 140 378.7508 361.5490 397.2571
## 141 255.6372 236.2824 276.9691
## 142 266.5882 248.2456 286.2428
## 143 277.2009 260.2088 295.5516
## 144 287.5572 272.1325 305.3964
## 145 297.7109 283.7237 316.1980
## 146 308.3730 294.1707 326.9996
## 147 319.2987 303.2385 337.5414
## 148 329.7047 312.9195 348.4266
## 149 340.3639 323.6819 359.9495
## 150 351.2631 335.0285 371.0515
## 151 246.2085 227.3720 263.6377
## 152 256.8433 237.8749 273.3870
## 153 267.7712 248.8591 283.7110
## 154 277.9846 259.8432 294.2804
## 155 288.4036 270.7825 305.3218
## 156 298.7415 281.5052 316.3493
## 157 309.4836 292.2090 326.8963
## 158 319.8556 302.8535 337.9806
## 159 330.2871 312.8472 349.0648
## 160 340.7165 322.8342 360.1932
## 161 247.8213 225.2348 265.9190
## 162 258.9231 236.7291 275.7340
## 163 269.0969 248.2234 285.7422
## 164 280.2187 260.1808 295.5137
## 165 290.8680 271.7340 305.2905
## 166 301.5598 282.8388 315.1521
## 167 312.3659 293.9436 324.9993
## 168 322.7694 305.0481 334.9346
## 169 333.7239 316.0560 345.4590
## 170 344.0699 326.0671 357.2263
## 171 268.5083 250.4957 288.0249
## 172 278.8126 261.3130 298.1422
## 173 289.1427 272.5436 308.3276
## 174 299.7059 283.9939 318.7212
## 175 309.8127 295.2226 329.1149
## 176 320.6968 306.0267 339.5085
## 177 331.3434 316.2092 349.9022
## 178 341.8191 326.2144 360.5312
## 179 352.0534 336.4674 371.5129
## 180 362.4635 346.8157 382.4983

## bootstrap confidence interval, subtracting variation due to individuals
pframe <- data.frame(Days = unique(sleepstudy$Days))
pframe$Reaction <- 0
pp <- predict(fm1, newdata = pframe, re.form=NA, type = 'response')

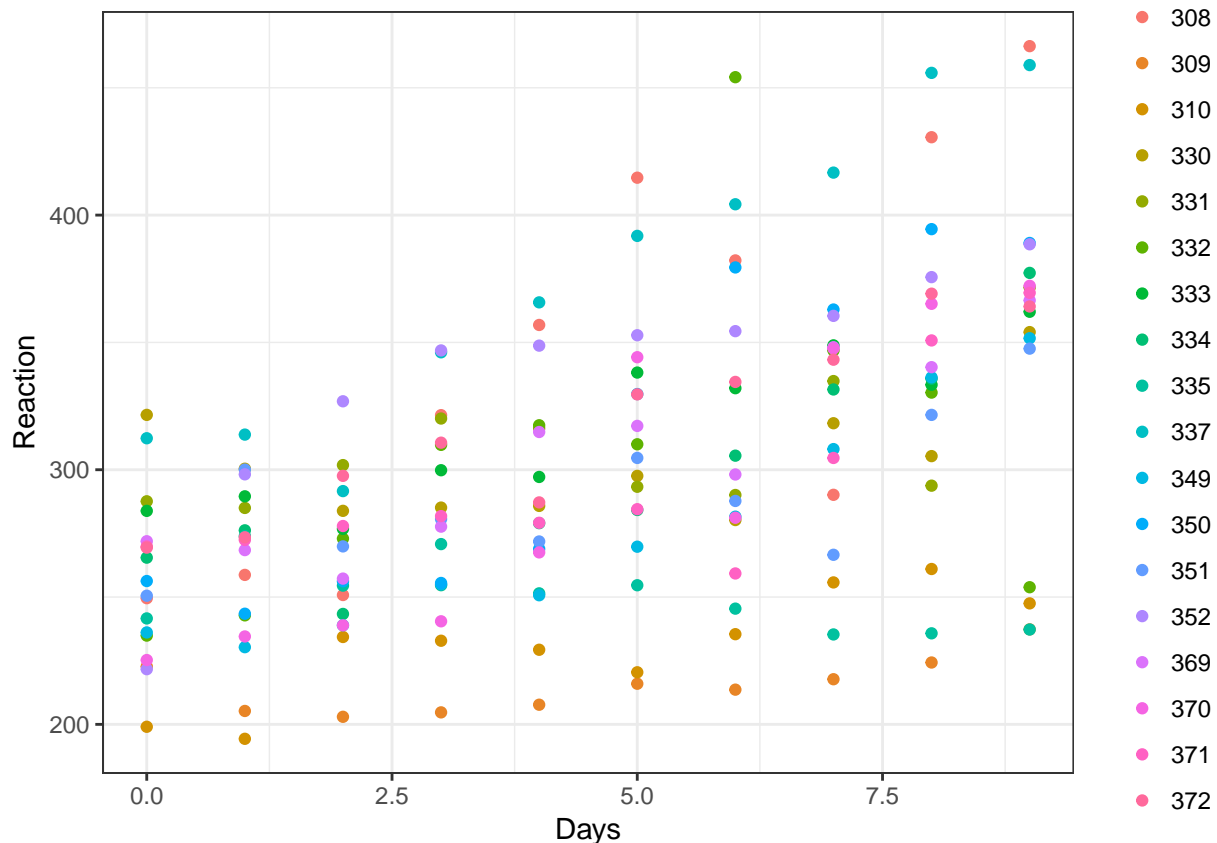
mm <- model.matrix(terms(fm1), pframe)
predFun<-function(.) (mm%*%fixef(.) )
bb<-bootMer(fm1,FUN=predFun,nsim=200) #do this 200 times

```

```
bb_se<-apply(bb$t,2,function(x) quantile(x, probs = c(0.025, 0.975)))
pframe$blo<-bb_se[1,]
pframe$bhi<-bb_se[2,]
pframe$predMean <- pp

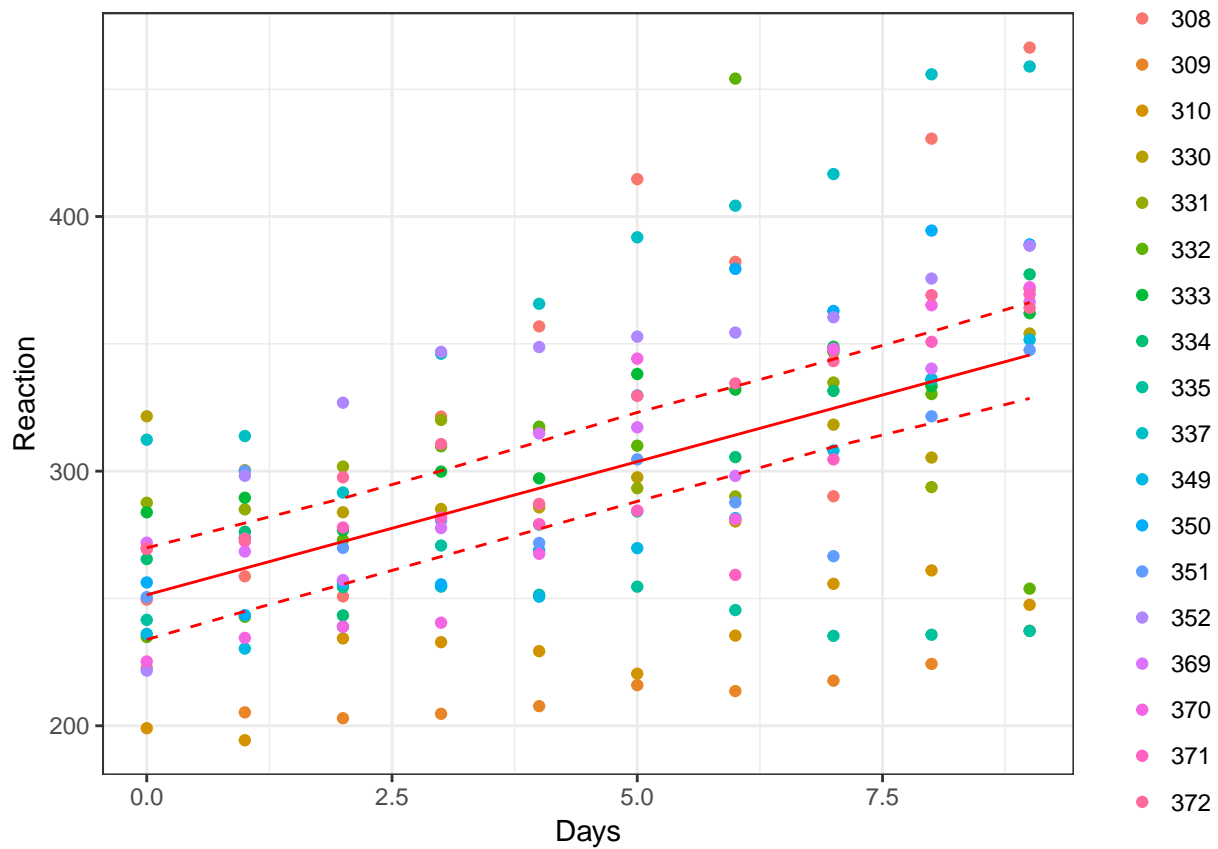
# calculate an "average" prediction interval, based on all individuals
predMeans = data.frame(predMean = tapply(PI.boot1$fit, sleepstudy$Days, mean ),
  predUp = tapply(PI.boot1$upr, sleepstudy$Days, mean ),
  predLwr = tapply(PI.boot1$lwr, sleepstudy$Days, mean ),
  Days = unique(sleepstudy$Days))

# plot raw data, colored by subject
ggplot(sleepstudy, aes(x = Days, y = Reaction, color = Subject)) +
  geom_point()
```

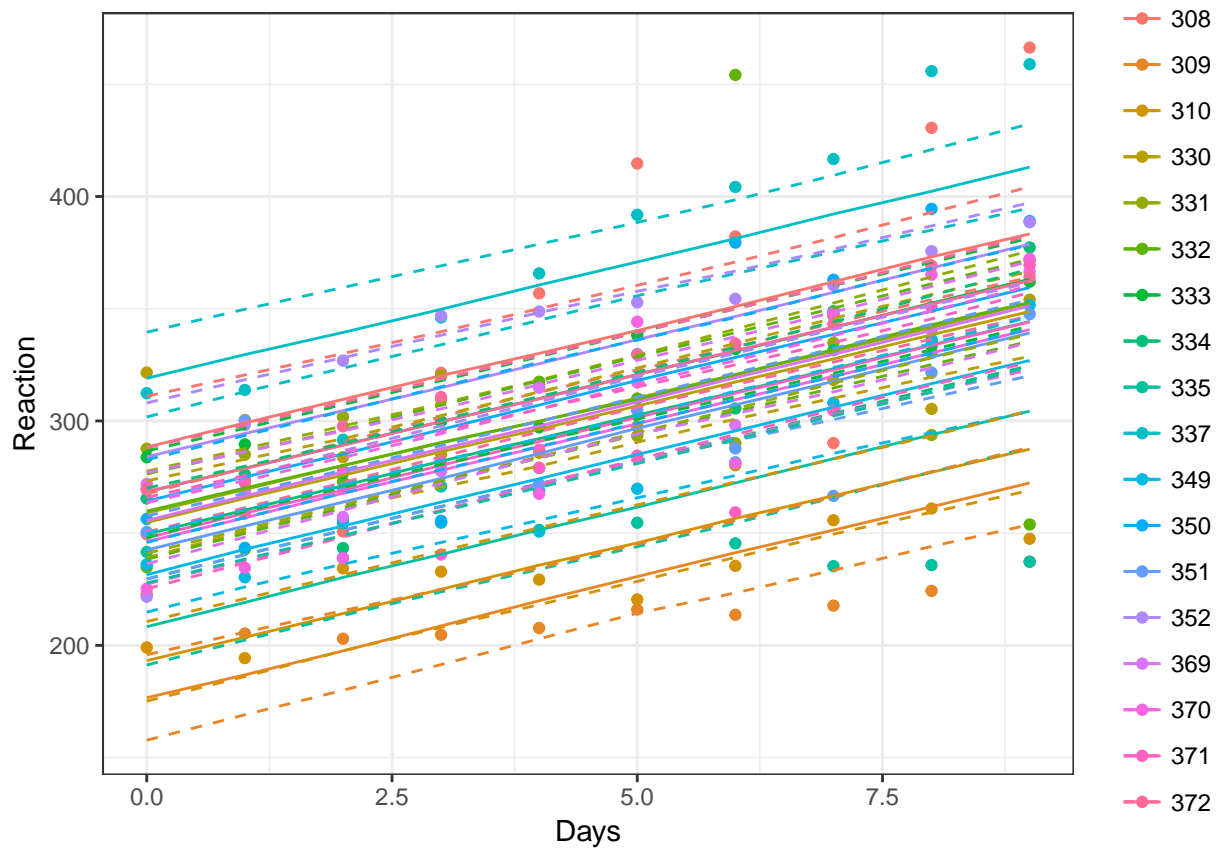


```
# plot confidence interval for an average individual (i.e. not taking random effects into account)
ggplot(sleepstudy, aes(x = Days, y = Reaction, color = Subject)) +
  geom_point() +
  geom_line(data = pframe, aes(x = Days, y = predMean), color = 'red' ) +
  geom_line(data = pframe, aes(x = Days, y = bhi), color = 'red', lty = 2 ) +
  geom_line(data = pframe, aes(x = Days, y = blo), color = 'red', lty = 2)
```

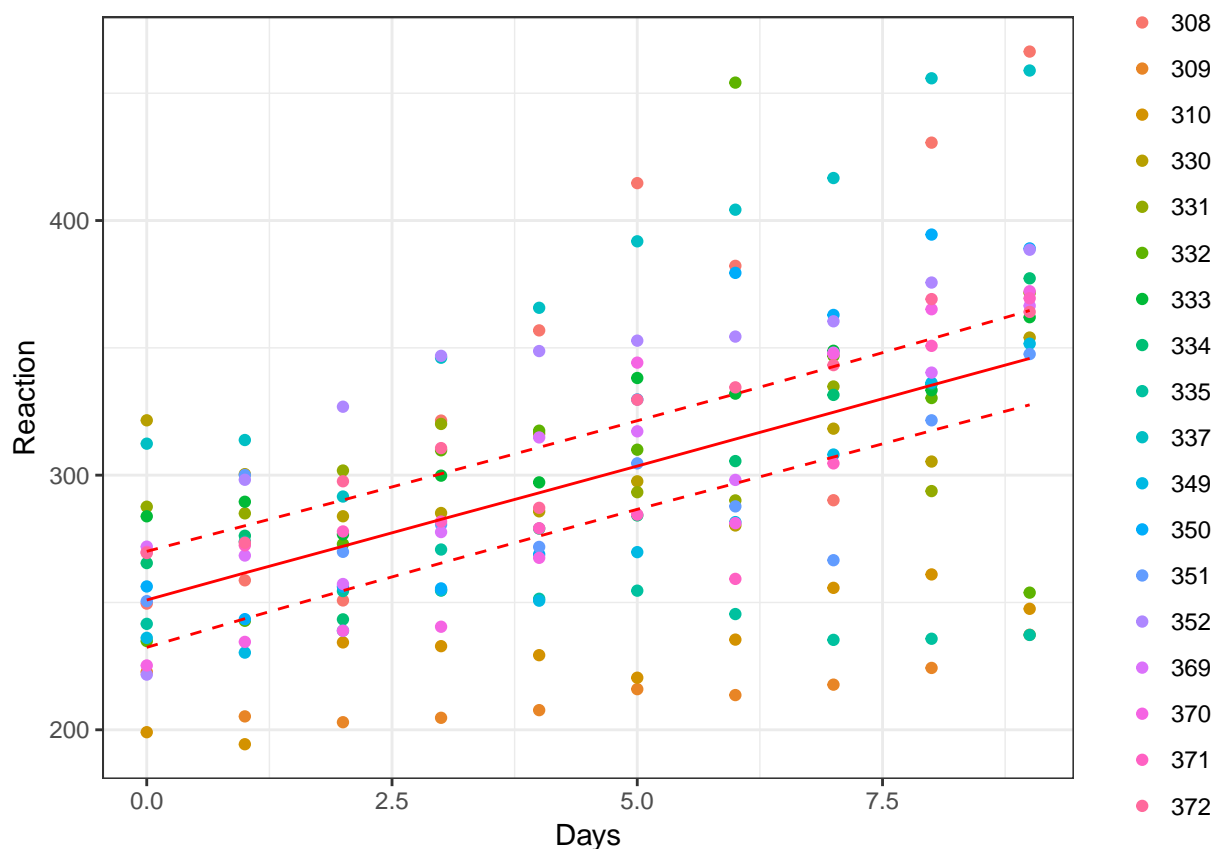




```
# plot showing predictions and CI's for each individual
ggplot(sleepstudy, aes(x = Days, y = Reaction, color = Subject)) +
  geom_point() +
  geom_line(data = PI.boot1, aes(x = sleepstudy$Days, y = fit, color = sleepstudy$Subject)) +
  geom_line(data = PI.boot1, aes(x = sleepstudy$Days, y = lwr,
                                color = sleepstudy$Subject), lty = 2) +
  geom_line(data = PI.boot1, aes(x = sleepstudy$Days, y = upr,
                                color = sleepstudy$Subject), lty = 2)
```



```
# plot showing average predictions and intervals (accounting for random effects)
ggplot(sleepstudy, aes(x = Days, y = Reaction, color = Subject)) +
  geom_point() +
  geom_line(data = predMeans, aes(x = Days, y = predMean), color = 'red') +
  geom_line(data = predMeans, aes(x = Days, y = predLwr), color = 'red', lty = 2) +
  geom_line(data = predMeans, aes(x = Days, y = predUpr), color = 'red', lty = 2)
```



```
### another method for fixed effects only
library(merTools)
```

```
## Loading required package: arm
```

```
## Loading required package: MASS
```

```
##
```

```
## arm (Version 1.9-3, built: 2016-11-21)
```

```
## Working directory is /Users/callinswitzer/Documents/GitRepos/GarbageCollector
```

```
## Loading required package: dplyr
```

```
##
```

```
## Attaching package: 'dplyr'
```

```
## The following object is masked from 'package:MASS':
```

```
##
```

```
##      select
```

```
## The following objects are masked from 'package:stats':
```

```
##
```

```
##      filter, lag
```

```
## The following objects are masked from 'package:base':
```

```
##
```

```
##      intersect, setdiff, setequal, union
```

```
pframe$Subject = 999 # not a subject from our dataset
```

```
# can change indlue.resid.var
```

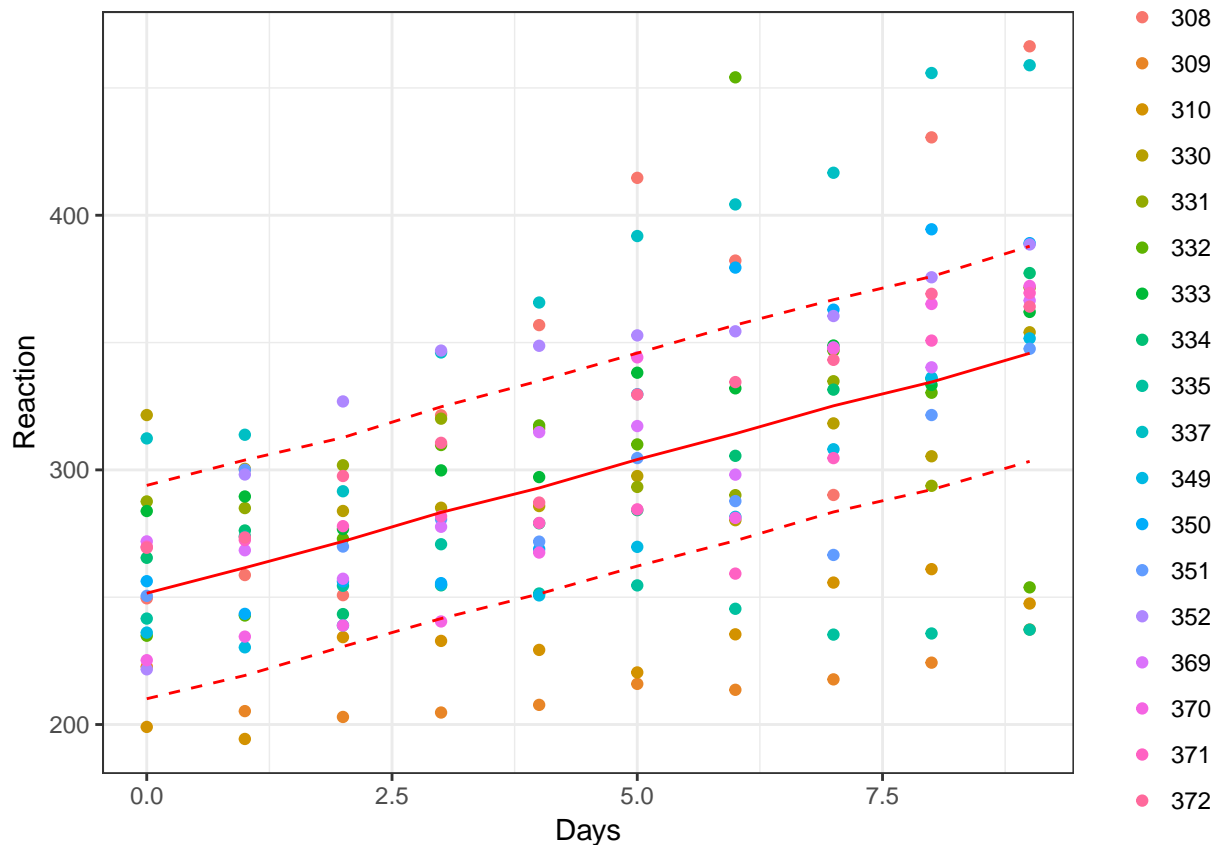
```
preds <- predictInterval(fm1, newdata = pframe, n.sims = 10000, stat = "mean", include.resid.var = TRUE)
```

```
## Warning:      The following levels of Subject from newdata
## -- 999 -- are not in the model data.
##      Currently, predictions for these values are based only on the
##      fixed coefficients and the observation-level error.

## Warning: executing %dopar% sequentially: no parallel backend registered

preds$Days = unique(sleepstudy$Days)
```

```
# plot showing average predictions and intervals (including residual variance)
ggplot(sleepstudy, aes(x = Days, y = Reaction, color = Subject)) +
  geom_point() +
  geom_line(data = preds, aes(x = Days, y = fit), color = 'red') +
  geom_line(data = preds, aes(x = Days, y = upr), color = 'red', lty = 2) +
  geom_line(data = preds, aes(x = Days, y = lwr), color = 'red', lty = 2)
```



```
fm1
```

```
## Linear mixed model fit by REML ['lmerMod']
## Formula: Reaction ~ Days + (1 | Subject)
##      Data: sleepstudy
## REML criterion at convergence: 1786.465
## Random effects:
##      Groups   Name      Std.Dev.
##      Subject  (Intercept) 37.12
##      Residual                      30.99
```

```
## Number of obs: 180, groups:  Subject, 18
## Fixed Effects:
## (Intercept)          Days
##      251.41          10.47
```