Week 6 exercise solutions

Ali Movasati, Isabelle Cretton, Tristan Koning

Oct. 14th, 2024

```
# Set global code chunk options
knitr::opts_chunk$set(warning = FALSE)
```

```
# load required libraries
library(skimr)
library(ggplot2)
library(lme4)
```

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

```
library(dplyr)
```

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

## The following objects are masked from 'package:stats':
##
##     filter, lag

## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
library(reshape2)
library(lattice)
```

Exercise 1

(a)

```
# Load data
hearing <-  read.table(file = "data/hearing.txt", sep = "\t", header = TRUE)
hearing <- within(hearing, {
  ListID <- factor(ListID, levels = c("List1", "List2", "List3", "List4"))
})
skim(hearing)
```

Table 1: Data summary

Name	hearing
Number of rows	96
Number of columns	3
Column type frequency:	
factor	1
numeric	2
Group variables	None

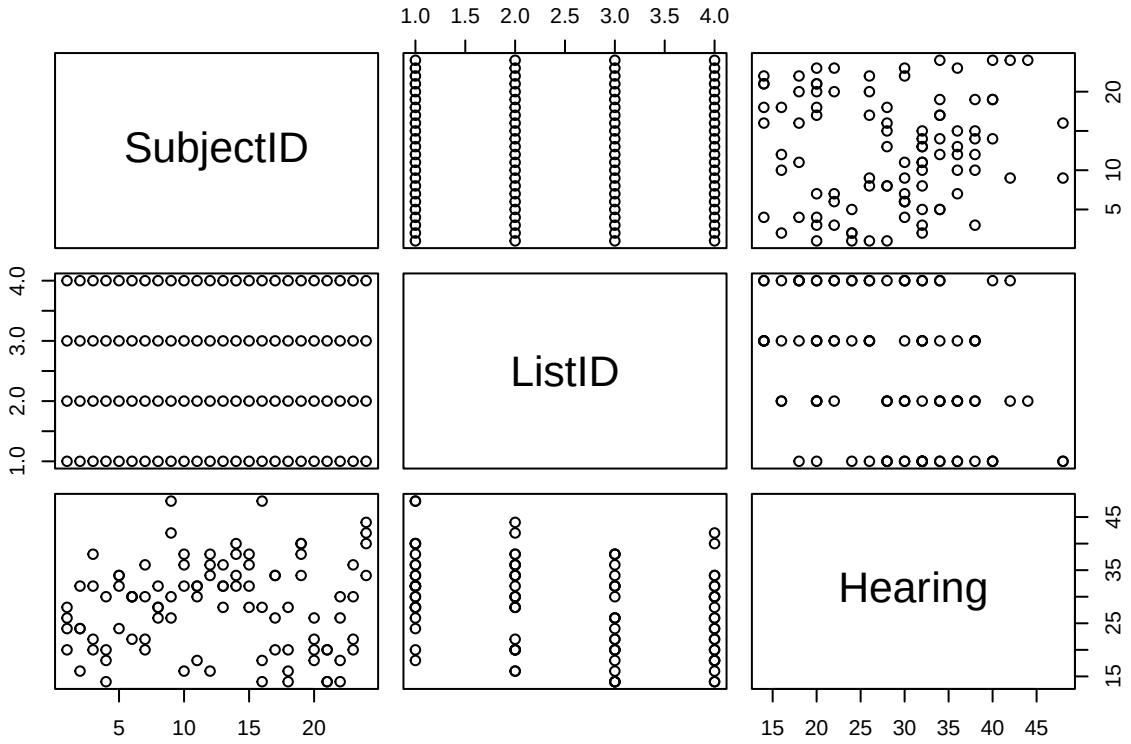
Variable type: factor

skim_variable	n_missing	complete_rate	ordered	n_unique	top_counts
ListID	0	1	FALSE	4	Lis: 24, Lis: 24, Lis: 24, Lis: 24

Variable type: numeric

skim_variable	n_missing	complete_rate	mean	sd	p0	p25	p50	p75	p100	hist
SubjectID	0	1	12.50	6.96	1	6.75	12.5	18.25	24	
Hearing	0	1	28.31	8.37	14	20.00	30.0	34.00	48	

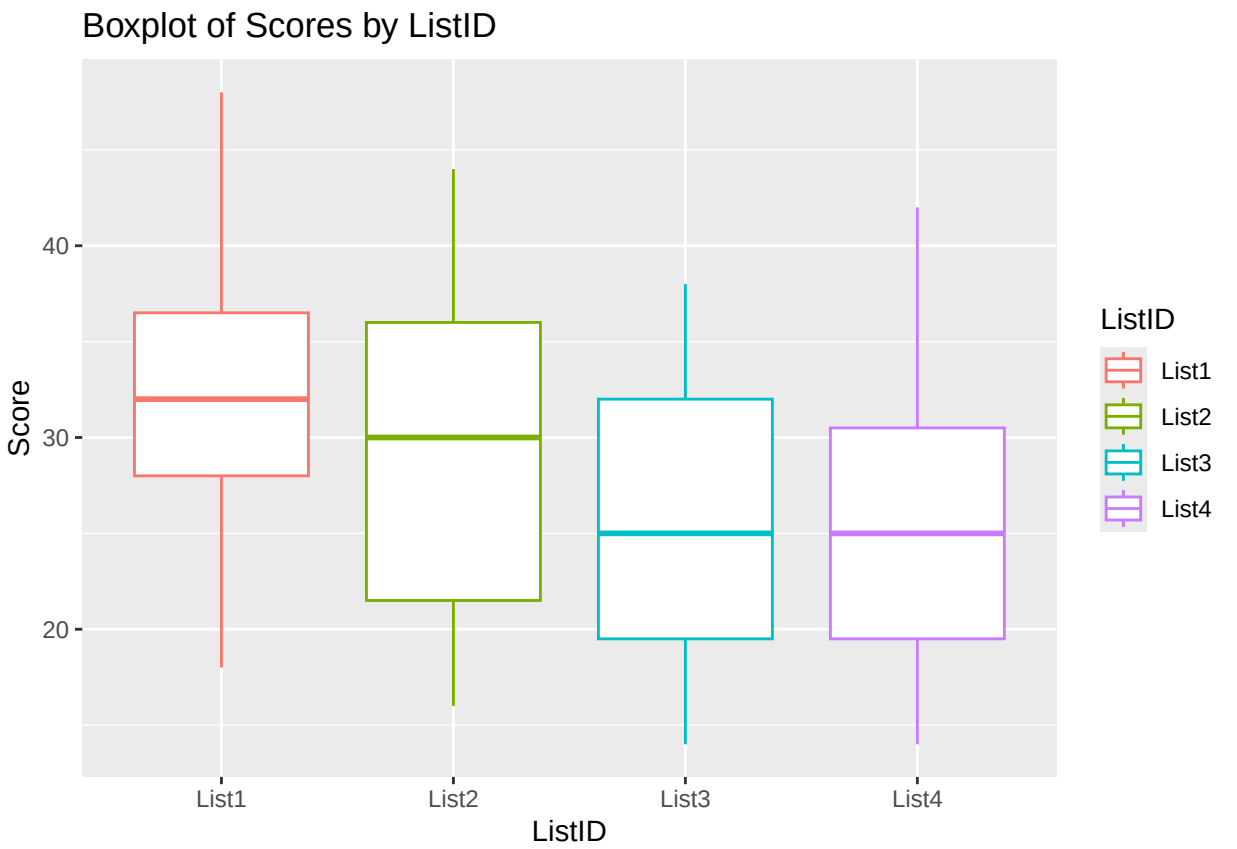
```
# Graphical summeries
pairs(hearing)
```



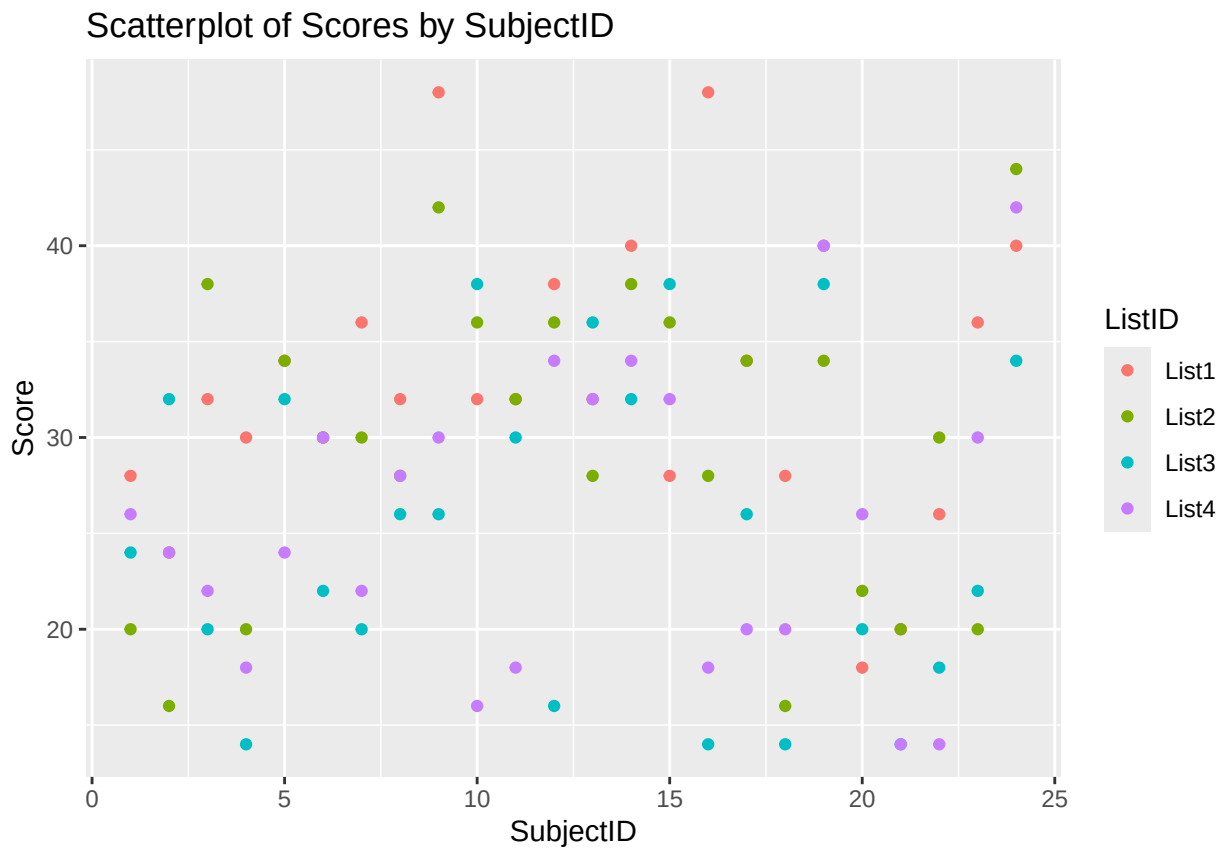

```
# Boxplot of scores by ListID
bxp1 <- ggplot(hearing, aes(x = ListID, y = Hearing, color = ListID)) +
  geom_boxplot() +
  labs(title = "Boxplot of Scores by ListID", x = "ListID", y = "Score")

# Scatterplot of subjectids on hearing scores
scat1 <- ggplot(hearing, aes(x = SubjectID, y = Hearing, color = ListID)) +
  geom_point() +
  labs(title = "Scatterplot of Scores by SubjectID", x = "SubjectID", y = "Score")

print(bxp1)
```


Boxplot of Scores by ListID
Score
40
30
20
List1
List2
List3
List4
ListID
ListID
List1
List2
List3
List4

```
print(scat1)
```



(b)

```
model_A <- lm(Hearing ~ ListID, data = hearing)

summary(model_A)
```

```
##
## Call:
## lm(formula = Hearing ~ ListID, data = hearing)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -14.7500  -5.5833  -0.2083   6.3333  16.4167
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   32.750      1.612  20.315  < 2e-16 ***
## ListIDList2   -3.083      2.280  -1.352  0.17955
## ListIDList3   -7.500      2.280  -3.290  0.00142 **
## ListIDList4   -7.167      2.280  -3.144  0.00225 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
```

```
## Residual standard error: 7.898 on 92 degrees of freedom
## Multiple R-squared:  0.1382, Adjusted R-squared:  0.1101
## F-statistic: 4.919 on 3 and 92 DF,  p-value: 0.00325
```

```
anova(model_A)
```

```
## Analysis of Variance Table
##
## Response: Hearing
##           Df Sum Sq Mean Sq F value  Pr(>F)
## ListID     3  920.5 306.819  4.9192 0.00325 **
## Residuals 92 5738.2  62.371
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

We can explain 11% of the variance in hearing scores by the list ID variable (Adjusted R squared). Looking at the anova output, ListID has a significant p-value meaning that there is a significant difference in means between the lists.

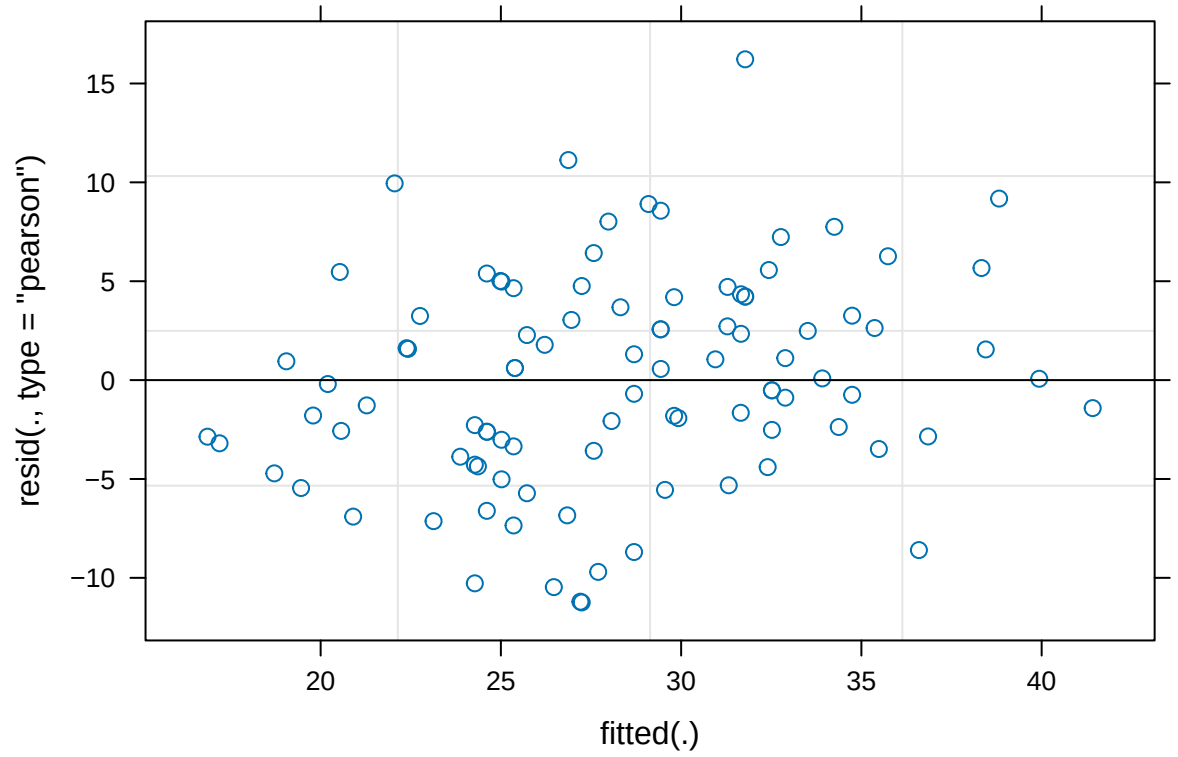
(c)

```
# Fit a linear mixed model
model_B <- lmer(Hearing ~ (1 | SubjectID) + ListID, data = hearing)
summary(model_B)
```

```
## Linear mixed model fit by REML ['lmerMod']
## Formula: Hearing ~ (1 | SubjectID) + ListID
##    Data: hearing
##
## REML criterion at convergence: 635.4
##
## Scaled residuals:
##      Min       1Q   Median       3Q      Max
## -1.86533 -0.56158 -0.01092  0.63222  2.69167
##
## Random effects:
##  Groups    Name        Variance Std.Dev.
##  SubjectID (Intercept) 26.04    5.103
##  Residual              36.33    6.027
## Number of obs: 96, groups:  SubjectID, 24
##
## Fixed effects:
##             Estimate Std. Error t value
## (Intercept)   32.750      1.612  20.315
## ListIDList2   -3.083      1.740  -1.772
## ListIDList3   -7.500      1.740  -4.311
## ListIDList4   -7.167      1.740  -4.119
##
## Correlation of Fixed Effects:
##             (Intr) LsIDL2 LsIDL3
## ListIDList2 -0.540
```

```
## ListIDList3 -0.540  0.500
## ListIDList4 -0.540  0.500  0.500
```

```
plot(model_B)
```




```
confint(model_B, method = "boot", nsim = 100, oldNames = FALSE)
```

```
## Computing bootstrap confidence intervals ...

##                                2.5 %     97.5 %
## sd_(Intercept)|SubjectID   3.148927  7.2032843
## sigma                      4.901444  6.8703745
## (Intercept)               29.081540 35.3216221
## ListIDList2               -6.341285  0.8156476
## ListIDList3              -10.790776 -2.6428658
## ListIDList4               -9.472148 -3.9235088
```

we cannot entirely conclude that the hearing scores differ for different lists, as there does exist an overlap in the confidence intervals of the estimates.

(d)

```
# TODO
```

Exercise 2

(a)

```
termites <- read.table(file = "data/termites.txt", sep = " ", header = TRUE)
# Remove NA entries
termites <- termites %>% select_if(~ !any(is.na(.)))

# EDA
skim(termites)
```

Table 4: Data summary

Name	termites
Number of rows	16
Number of columns	15
Column type frequency:	
numeric	15
Group variables	None

Variable type: numeric

skim_variable	n_missing	complete_rate	mean	sd	p0	p25	p50	p75	p100	hist
dish	0	1	8.50	4.76	1	4.75	8.5	12.25	16	
dose	0	1	7.50	2.58	5	5.00	7.5	10.00	10	
day1	0	1	25.00	0.00	25	25.00	25.0	25.00	25	
day2	0	1	23.94	1.88	18	24.00	24.5	25.00	25	
day4	0	1	19.00	6.63	4	15.50	22.5	23.25	25	
day5	0	1	15.56	7.51	3	9.00	19.0	21.25	24	
day6	0	1	13.81	7.79	1	6.75	17.0	20.25	22	
day7	0	1	12.62	7.49	1	4.75	16.0	18.25	22	
day8	0	1	11.31	7.60	0	2.75	14.0	18.00	21	
day10	0	1	10.31	7.27	0	2.75	13.0	16.00	20	
day11	0	1	9.50	7.00	0	1.75	12.5	15.00	18	
day12	0	1	8.38	6.23	0	1.75	11.0	12.25	18	
day13	0	1	7.81	5.88	0	1.75	10.0	11.50	17	
day14	0	1	7.50	5.83	0	1.00	9.0	11.50	17	
day15	0	1	6.81	5.56	0	0.75	7.5	11.25	16	

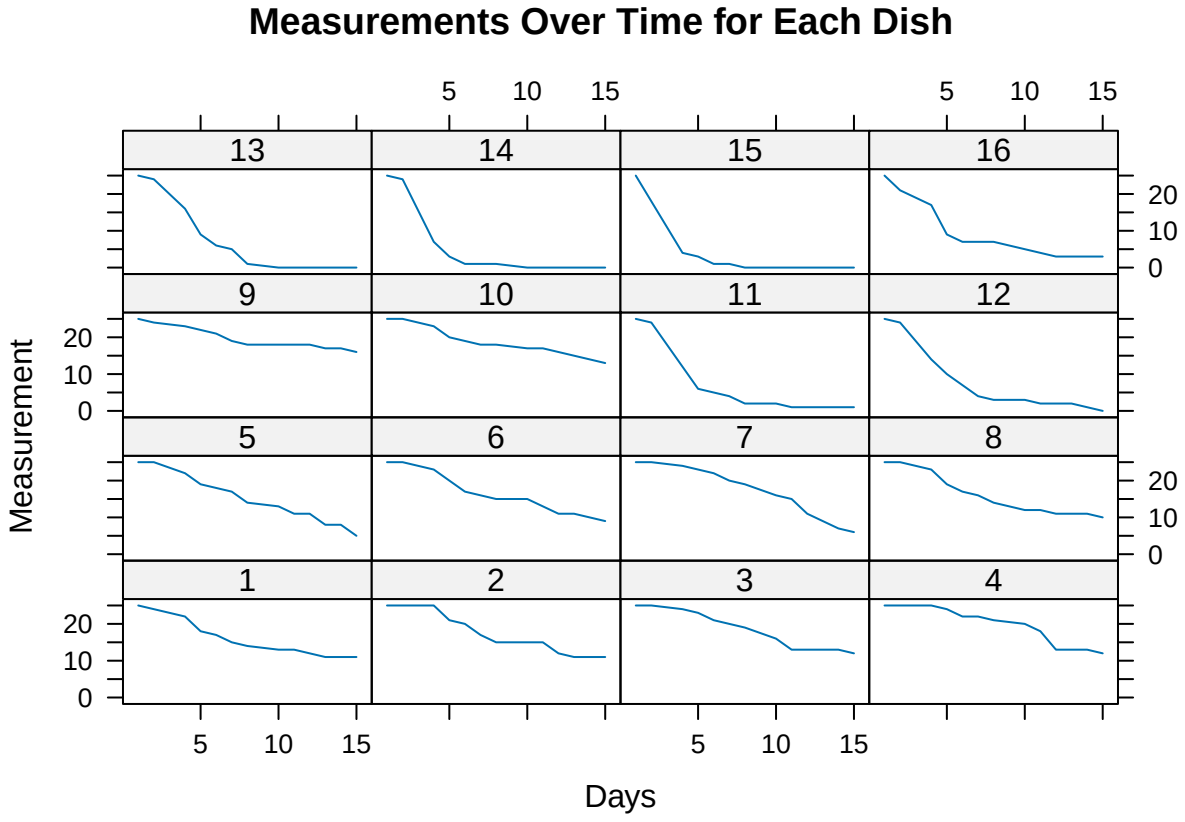
(b)

```r
# Reshape data
termites <- melt(termites, id.vars = c("dish", "dose"),
                 variable.name = "day",
                 value.name = "measurement")
str(termites)
```

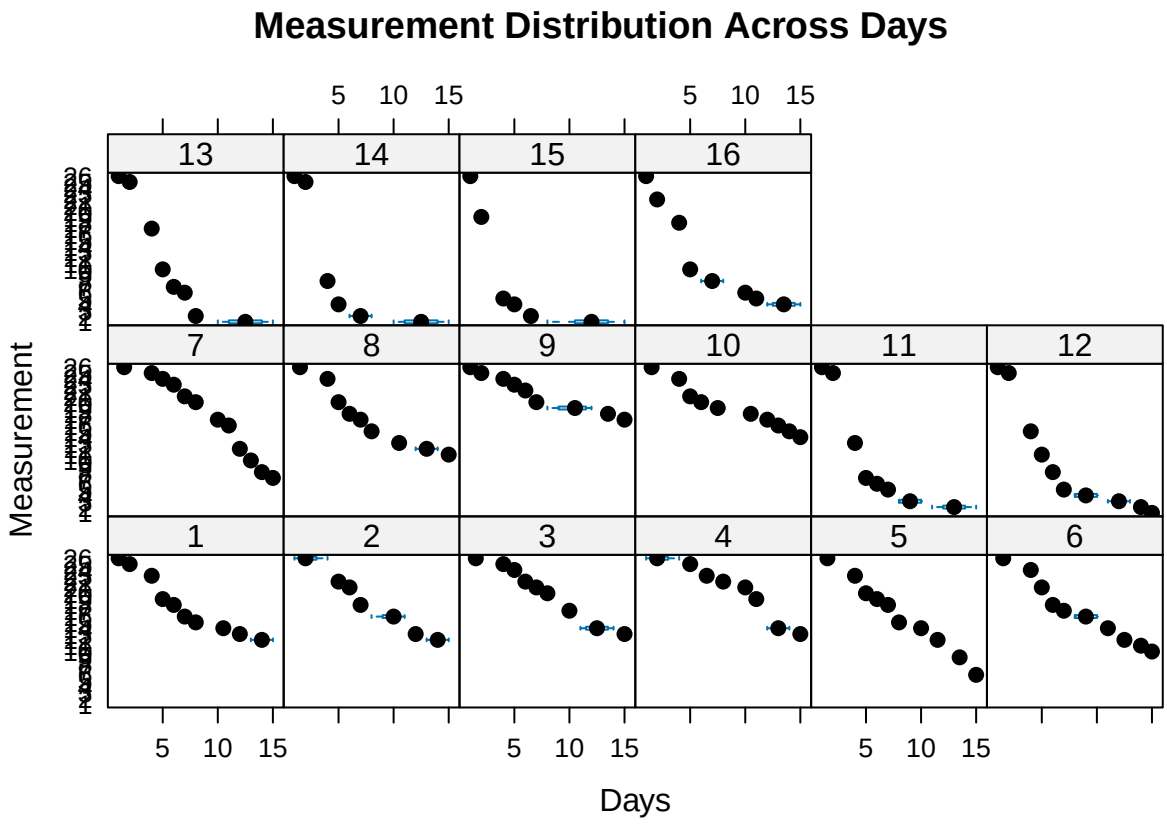
```
## 'data.frame':    208 obs. of  4 variables:
##  $ dish       : int  1 2 3 4 5 6 7 8 9 10 ...
##  $ dose       : int  5 5 5 5 5 5 5 5 10 10 ...
##  $ day        : Factor w/ 13 levels "day1","day2",..: 1 1 1 1 1 1 1 1 1 1 ...
##  $ measurement: int  25 25 25 25 25 25 25 25 25 25 ...
```

```r
# Convert day to numeric
termites$day <- as.numeric(gsub("day", "", as.character(termites$day)))

# Graphical EDA
lattice::xyplot(measurement ~ as.numeric(gsub("day", "", day)) | as.factor(dish),
       data = termites, type = "l",
       layout = c(4, 4),
       main = "Measurements Over Time for Each Dish",
       xlab = "Days", ylab = "Measurement")
```



```r
# bwplot of measurements distribution across days
lattice::bwplot(measurement ~ as.numeric(gsub("day", "", day)) | as.factor(dish),
       data = termites,
       main = "Measurement Distribution Across Days",
       xlab = "Days", ylab = "Measurement")
```



(c)

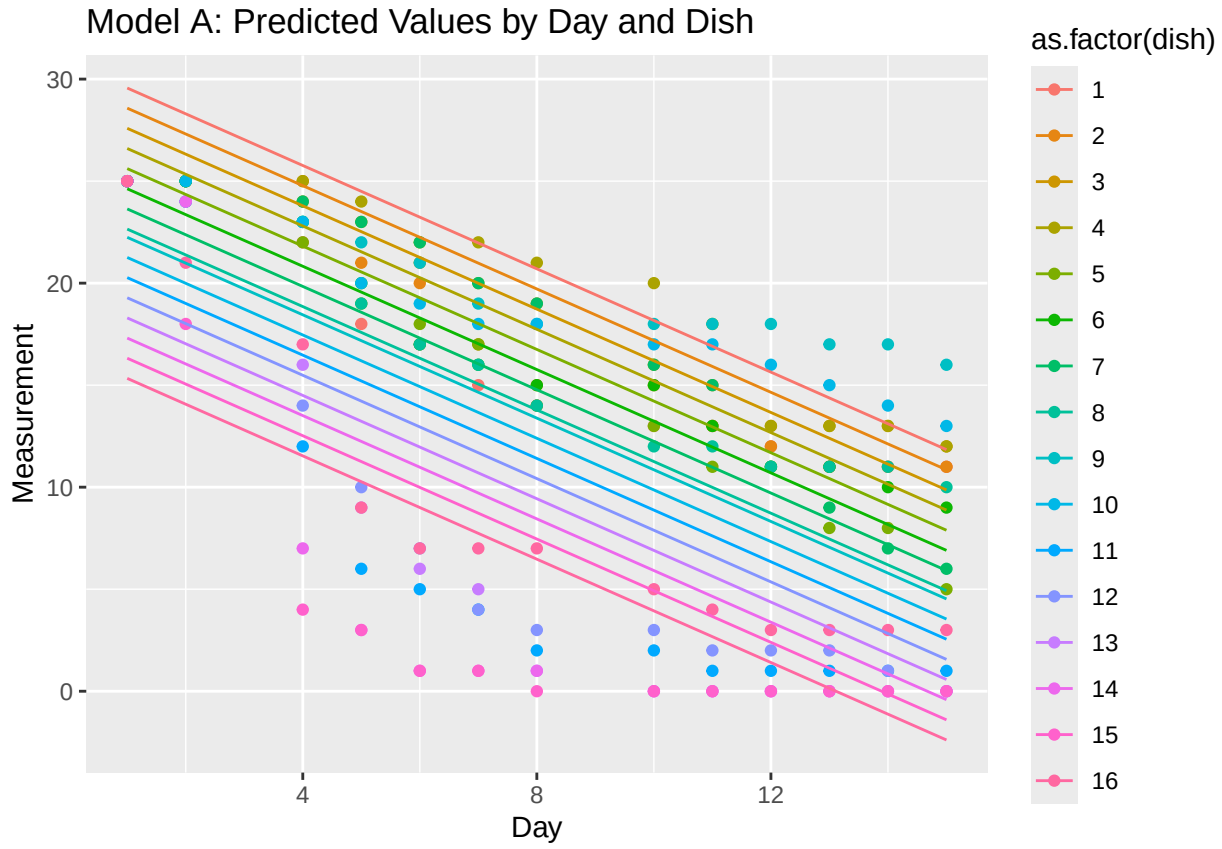
```r
# Fit a linear model
model_A <- lm(measurement ~ dose + day + dish, data = termites)
summary(model_A)
```

```
##
## Call:
## lm(formula = measurement ~ dose + day + dish, data = termites)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -9.9783 -3.1681  0.1416  3.3309 11.4765
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 31.23290    1.30944  23.852  < 2e-16 ***
```

```
## dose         0.11676    0.26441   0.442    0.659
## day         -1.26588    0.07451 -16.989  < 2e-16 ***
## dish        -0.98764    0.14340  -6.887  6.9e-11 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 4.739 on 204 degrees of freedom
## Multiple R-squared:  0.6928, Adjusted R-squared:  0.6883
## F-statistic: 153.4 on 3 and 204 DF,  p-value: < 2.2e-16
```

```r
termites$predicted_A <- predict(model_A)

# Plot model
ggplot(termites, aes(x = as.numeric(gsub("day", "", day)), y = measurement, color = as.factor(dish))) +
  geom_point() +
  geom_line(aes(y = predicted_A)) +
  labs(title = "Model A: Predicted Values by Day and Dish", x = "Day", y = "Measurement")
```



It is problematic to use dish as a dependent variable, as repeated measurements are taken on the same dish. This violates the assumption of independence of observations.

(d)

```r
# Fit a linear mixed model
model_B <- lmer(measurement ~ dose + day + (1 | dish), data = termites)
summary(model_B)
```

```
## Linear mixed model fit by REML ['lmerMod']
## Formula: measurement ~ dose + day + (1 | dish)
##    Data: termites
##
## REML criterion at convergence: 1137.6
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -2.0822 -0.5606 -0.0373  0.4065  3.4749
##
## Random effects:
##  Groups   Name        Variance Std.Dev.
##  dish     (Intercept) 18.66    4.320
##  Residual             10.97    3.312
## Number of obs: 208, groups:  dish, 16
##
## Fixed effects:
##             Estimate Std. Error t value
## (Intercept) 34.68963    3.51844   9.859
## dose        -1.46346    0.44167  -3.313
## day         -1.26588    0.05208 -24.305
##
## Correlation of Fixed Effects:
##      (Intr) dose
## dose -0.941
## day  -0.123  0.000
```

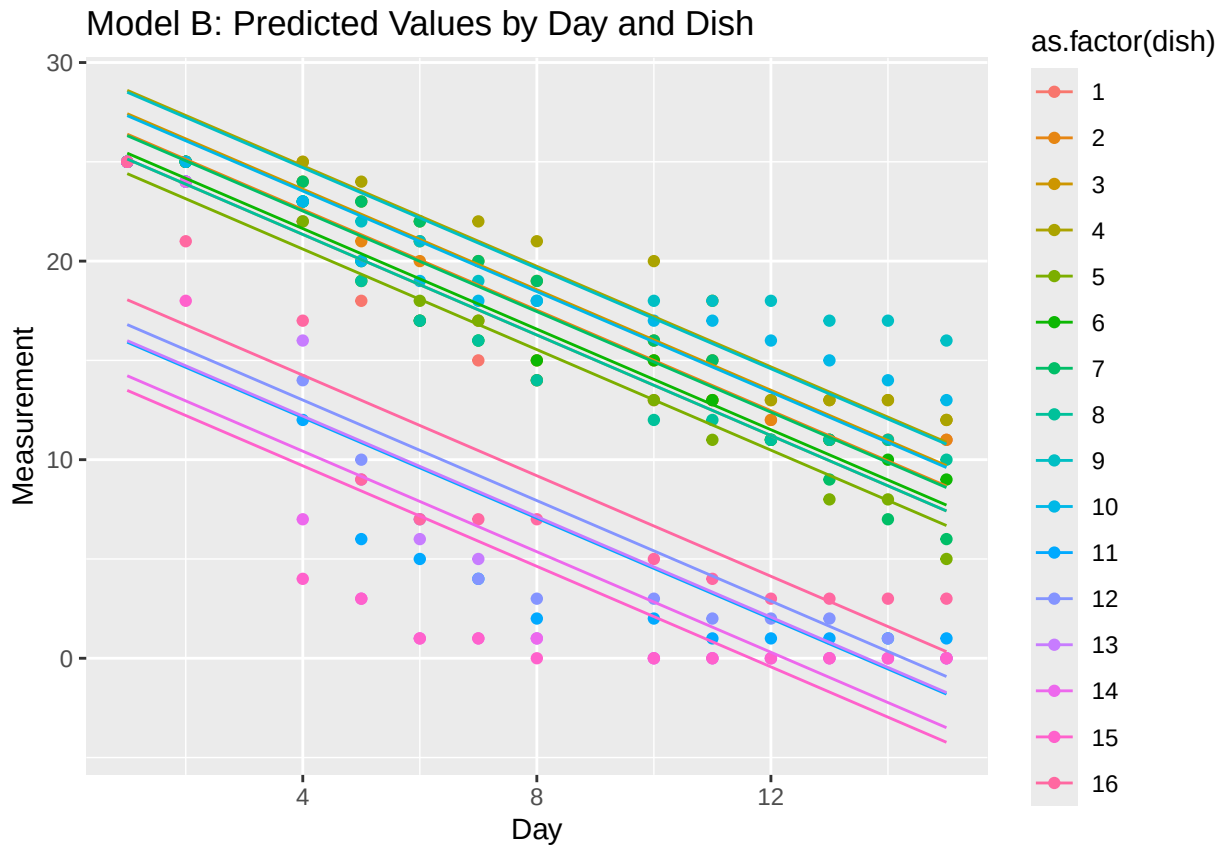
```r
confint(model_B, method = "boot", nsim = 100, oldNames = FALSE)
```

```
## Computing bootstrap confidence intervals ...
```

```
##                          2.5 %     97.5 %
## sd_(Intercept)|dish  2.944326  6.1177049
## sigma                3.038919  3.6519322
## (Intercept)         27.465606 41.2047480
## dose                -2.352257 -0.5650528
## day                 -1.371263 -1.1520227
```

```r
termites$predicted_B <- predict(model_B)

# Plot model
ggplot(termites, aes(x = as.numeric(gsub("day", "", day)), y = measurement, color = as.factor(dish))) +
  geom_point() +
  geom_line(aes(y = predicted_B)) +
  labs(title = "Model B: Predicted Values by Day and Dish", x = "Day", y = "Measurement")
```



The linear mixed model is more appropriate for this data as it accounts for the repeated measurements taken on the same dish, this can also be seen visually in the plots, where each prediction in model_B seem to be closer to the actual measurements compared to model_A. Dose now seems highly correlated with the measurements whereas in model_A it was not significant.

(e)

```
# Fit a linear mixed model
model_C <- lmer(measurement ~ dose + day + (day | dish) + day, data = termites)
```

```
## boundary (singular) fit: see help('isSingular')
```

```
summary(model_C)
```

```
## Linear mixed model fit by REML ['lmerMod']
## Formula: measurement ~ dose + day + (day | dish) + day
##    Data: termites
##
## REML criterion at convergence: 1115.1
##
## Scaled residuals:
##      Min       1Q   Median       3Q      Max
## -2.47711 -0.51473  0.04884  0.42996  2.89506
```

```
##
## Random effects:
##  Groups   Name        Variance Std.Dev. Corr
##  dish     (Intercept) 4.77731  2.1857
##           day         0.06521  0.2554   1.00
##  Residual             9.66642  3.1091
## Number of obs: 208, groups:  dish, 16
##
## Fixed effects:
##             Estimate Std. Error t value
## (Intercept) 32.49049    2.18765  14.852
## dose        -1.17024    0.27571  -4.245
## day         -1.26588    0.08041 -15.743
##
## Correlation of Fixed Effects:
##      (Intr) dose
## dose -0.945
## day   0.085  0.000
## optimizer (nloptwrap) convergence code: 0 (OK)
## boundary (singular) fit: see help('isSingular')
```

```
confint(model_C, method = "boot", nsim = 100, oldNames = FALSE)
```

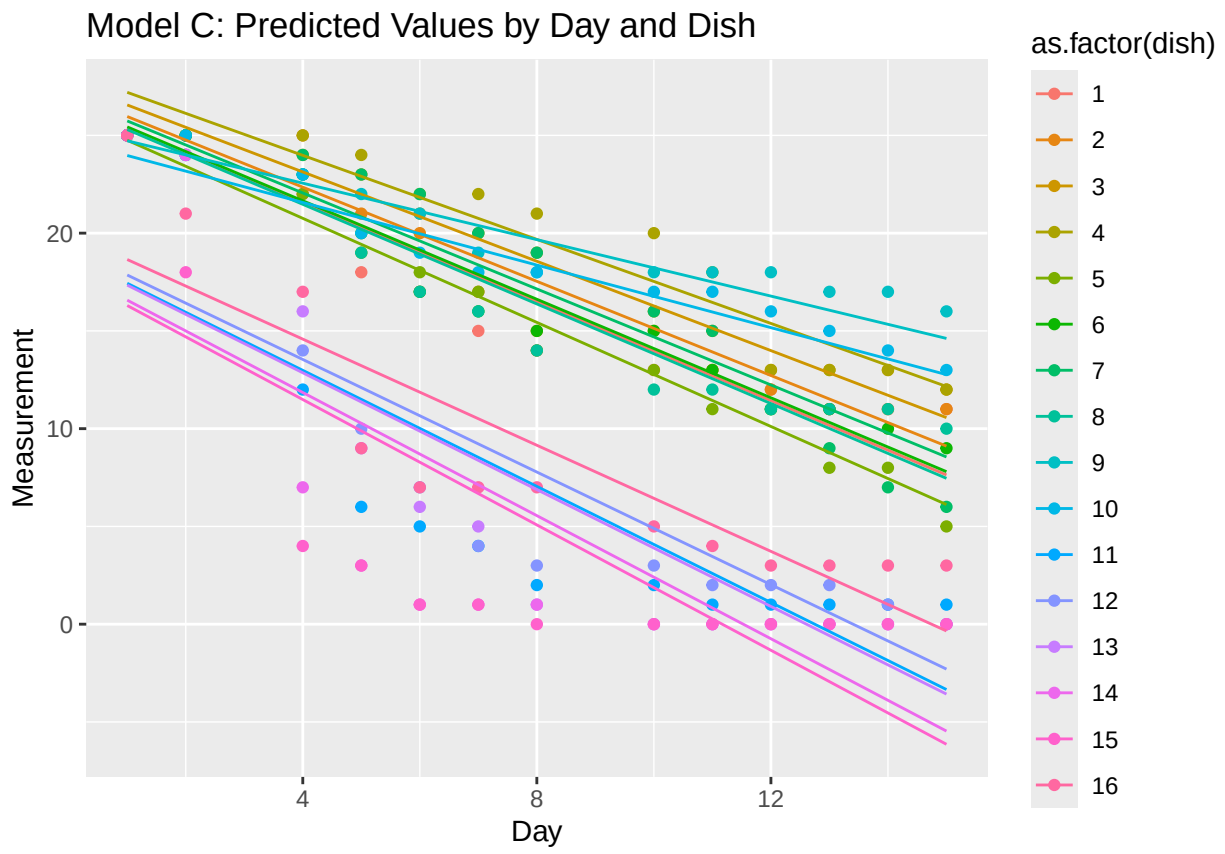
```
## Computing bootstrap confidence intervals ...
```

```
##
## 57 message(s): boundary (singular) fit: see help('isSingular')
## 8 warning(s): Model failed to converge with max|grad| = 0.00201939 (tol = 0.002, component 1) (and o
```

```
##                                2.5 %     97.5 %
## sd_(Intercept)|dish       1.0057294  3.7244811
## cor_day.(Intercept)|dish  0.1108457  1.0000000
## sd_day|dish               0.1301194  0.3788158
## sigma                     2.8301218  3.4344205
## (Intercept)              28.3973339 36.4064658
## dose                     -1.7257402 -0.5898666
## day                      -1.4435524 -1.1113893
```

```
termites$predicted_C <- predict(model_C)

# Plot model
ggplot(termites, aes(x = as.numeric(gsub("day", "", day)), y = measurement, color = as.factor(dish))) +
  geom_point() +
  geom_line(aes(y = predicted_C)) +
  labs(title = "Model C: Predicted Values by Day and Dish", x = "Day", y = "Measurement")
```



Again, this model seems to be more accurate than the previous models.

(f)

```
boostrap_ci <- function(data, formula, parameter, N = 1000, conf = 0.90) {
  estimates <- numeric(N)

  for (i in 1:N) {
    resample <- data[sample(nrow(data), replace = TRUE), ]
    model <- lmer(formula, data = resample)
    estimates[i] <- fixef(model)[[parameter]]
  }

  return(quantile(estimates, c((1 - conf) / 2, 1 - (1 - conf) / 2)))
}

ci <- boostrap_ci(termites, measurement ~ dose + day + (1 | dish) + day, "dose")
ci
```

```
##        5%       95%
## -1.615396 -1.306670
```

The estimate of model_B where we have only a random intercept for dish falls within the 90% confidence interval, therefore we can conclude that the estimate is significant.