Homework 1

Leah Schultz

9/28/2017

Chapter 2: LDA Basics

```
library(tidyr)
library(ggplot2)
library(dplyr)
oysup <- read.csv("~/1-descriptives-and-graphs-leahschultz/oysup_teacher_self.csv")
purpose <- read.csv("~/Dropbox/Lab & Research/OYSUP Project/oysup_self.csv")
oysup <- oysup %>%
  dplyr::select(FAMID, neuro_7s:neuro_10s)
dems <- purpose %>%
  dplyr::select(SEX2)
oysup <- cbind(oysup, dems)
```

1. Move your data into a long format and a wide format. Did you have any specific challenges that you encountered? If so, discuss them.

```
oysup_long <- tbl_df(oysup) %>%
  gather(c(neuro_7s:neuro_10s), key = "grade", value = "value") %>%
  separate(grade, into = c("variable", "grade"), sep = "_", convert = T) %>%
  separate(grade, into = c("grade", "delete"), sep = "s") %>%
  dplyr::select(-delete) %>%
  spread(variable, value)
oysup_long
```

```
## # A tibble: 4,296 x 4
##    FAMID  SEX2 grade neuro
##  * <int> <int> <chr> <dbl>
##  1  1001     2    10   5.0
##  2  1001     2     7    NA
##  3  1001     2     8    NA
##  4  1001     2     9   3.5
##  5  1002     2    10   2.5
##  6  1002     2     7   3.5
##  7  1002     2     8   3.5
##  8  1002     2     9   2.0
##  9  1003     1    10   3.5
## 10  1003     1     7    NA
## # ... with 4,286 more rows
```

```
oysup_wide <- tbl_df(oysup_long) %>%
  gather(-c(FAMID, SEX2, grade), key = "variable", value = "value") %>%
  unite(VarG, variable, grade)  %>%
  spread(key = VarG, value = value) %>%
  select_if(~sum(!is.na(.)) > 0)
oysup_wide
```

```
## # A tibble: 1,074 x 6
##    FAMID  SEX2 neuro_10 neuro_7 neuro_8 neuro_9
##   * <int> <int>    <dbl>   <dbl>   <dbl>   <dbl>
##   1  1001     2      5.0      NA      NA     3.5
##   2  1002     2      2.5     3.5     3.5     2.0
##   3  1003     1      3.5      NA     4.0     3.5
##   4  1004     2      3.0     3.0     3.5     3.0
##   5  1005     1      2.5      NA      NA     2.5
##   6  1006     1      2.5      NA     2.0     1.5
##   7  1007     2      3.0     3.0     5.0     3.5
##   8  1008     2      3.5     5.0     3.5     4.0
##   9  1009     2      3.0     2.5     2.5     4.0
##  10  1010     1       NA      NA     4.0     3.0
## # ... with 1,064 more rows
```

Challenges: First I forgot to exclude the ID variable and stable demographics, so R tried to make it into a value. I had a lot of variables that had repeated measures, so I had to think about how to split them after I gathered everything. Also, my variables were not consistently named because I was mixing naming conventions (my preferred conventions, and then the ones that OPP used). I went in and cleaned up my file a lot more so that I could use the separate function easily in the next step.

Another thing that was difficult was I ended up with some NA columns when I spread my data back to wide format – the drop and fill arguments didn't seem to help, so I had to find a solution for how to drop the NA columns from the key-pair combinations that didn't exist (for example, oysup wasn't assessed at grade 1).

2. Create a wave variable and date variable (if applicable).

I already have a grade variable, which is equivalent to wave for my purposes, and do not have dates available beyond year, which is not very useful.

3. What is your sample size for each wave of assessment?

```
oysup_long %>%
  group_by(grade) %>%
  filter(!is.na(neuro)) %>%
  count()
```

```
## # A tibble: 4 x 2
## # Groups:   grade [4]
##   grade     n
##   <chr> <int>
## 1    10   895
## 2     7   579
## 3     8   765
## 4     9   905
```

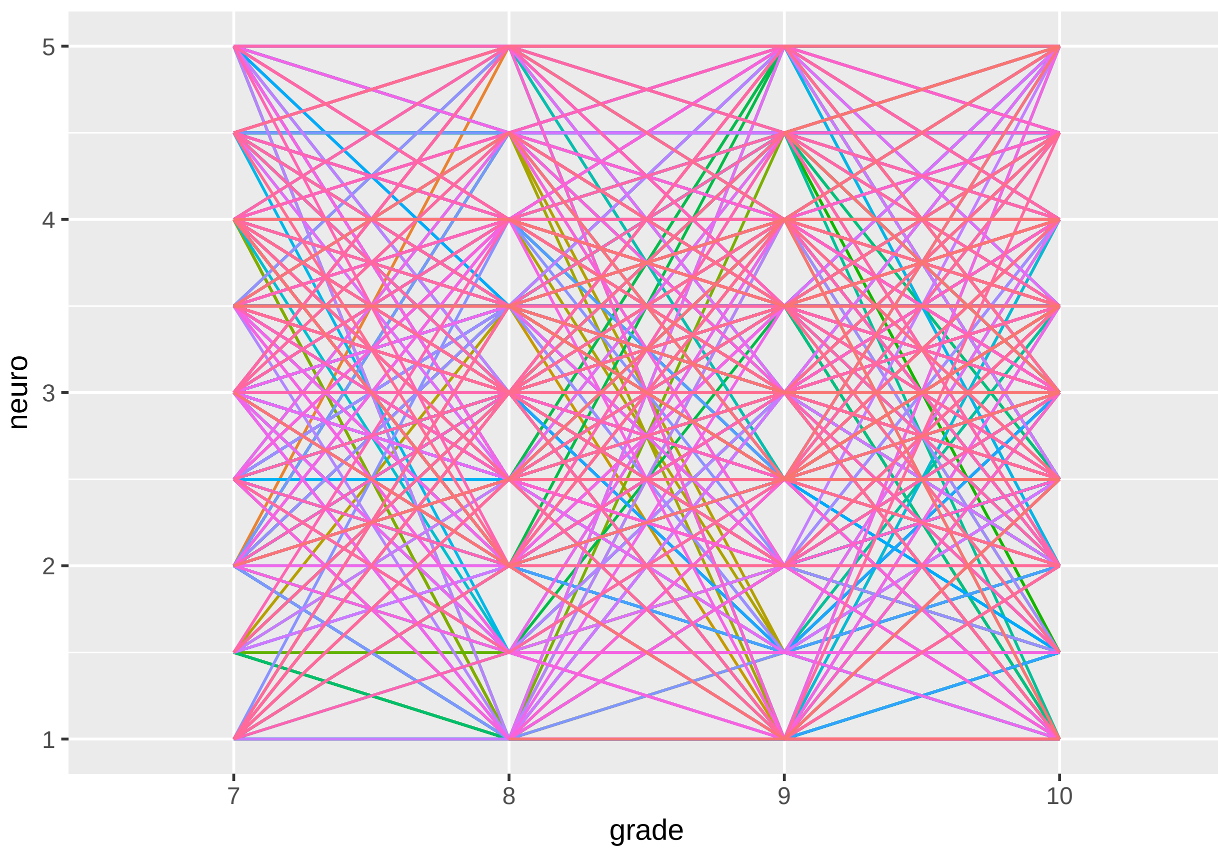
4. Take the date variable and convert it to a different date format such as time in study or age (if appropriate). What scale is most suitable for your analyses? (weeks/months/years?)

Not applicable for my analyses.

5. Graph your data using the different time metrics, fitting individual curves for each person.

Plotting individual curves for neuroticism over time:

```
gg2 <- ggplot(oysup_long, aes(x = grade, y = neuro, group=FAMID)) +
  geom_line() +
  aes(colour = factor(FAMID)) + guides(colour=FALSE) +
  scale_x_discrete(limits = c("7","8","9","10"))
gg2
```



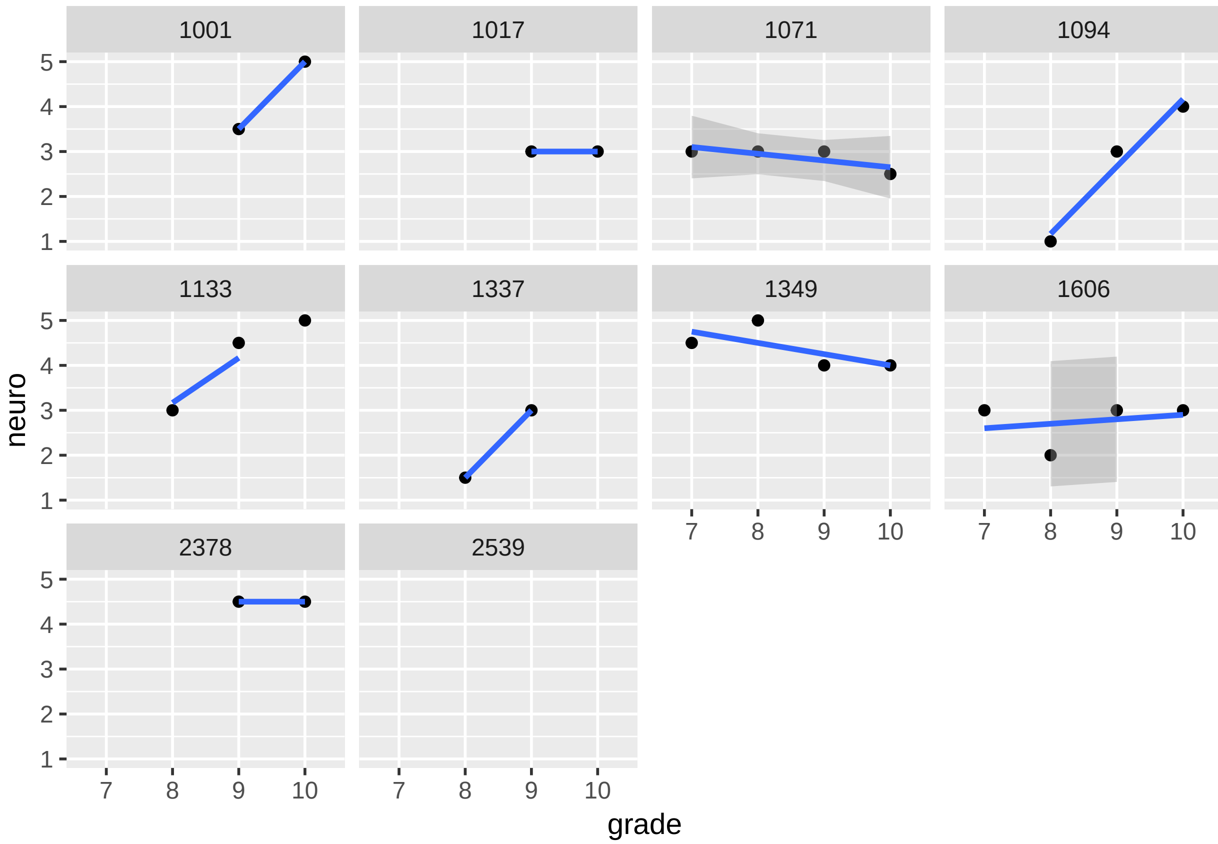
Predicted values should look a little better, because each time point is an average of just two values…

```
## Subset of 10 curves
set.seed(11)
ex.random <- oysup_long %>%
  dplyr::select(FAMID) %>%
  distinct %>%
  sample_n(10)
```

```
example <-
  left_join(ex.random, oysup_long)
```

```
## Joining, by = "FAMID"
```

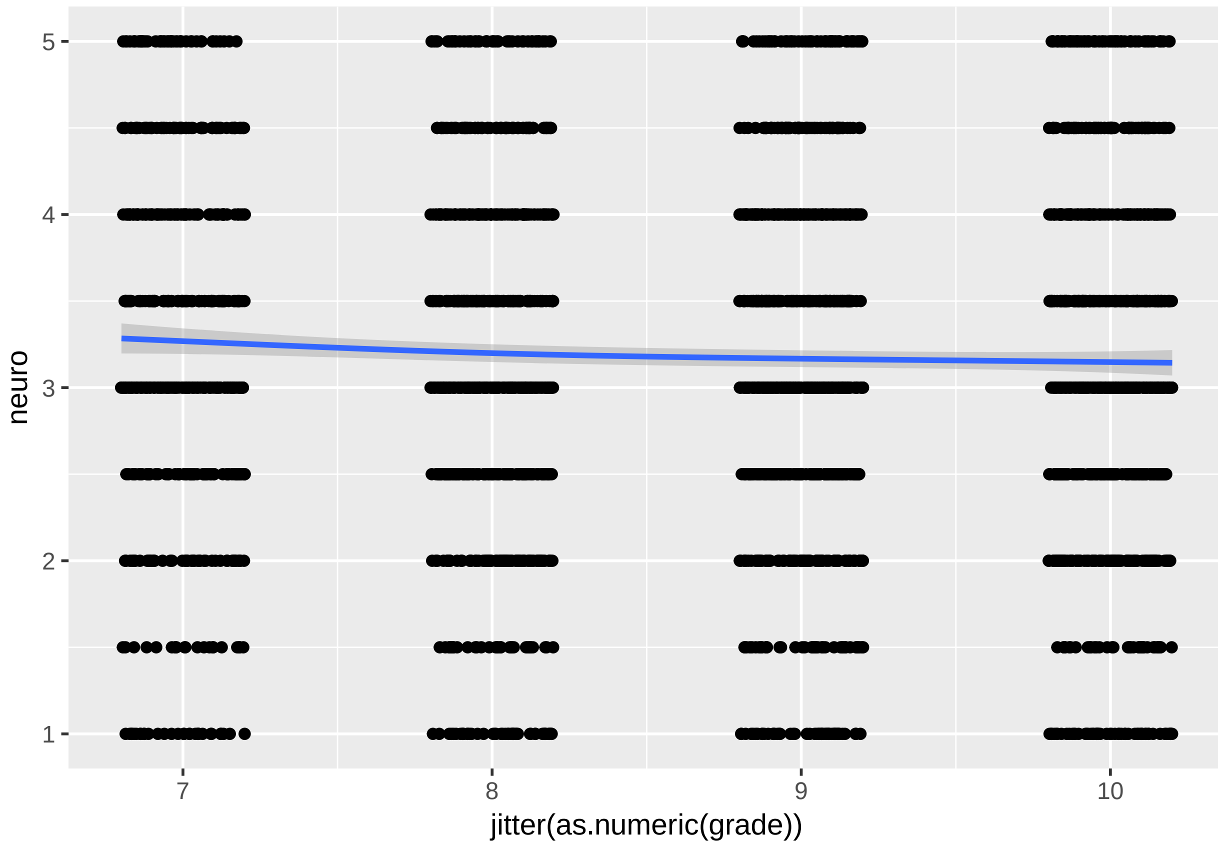
```
gg4 <- ggplot(example, aes(x = grade, y = neuro, group = FAMID)) +
  geom_point() + stat_smooth(method="lm") + facet_wrap(~FAMID) +
  ylim(1,5)+
  scale_x_discrete(limits = c("7","8","9","10"))
gg4
```



6. Create an overall average trend of your data (split up into groups if appropriate). Attempt to color your individual data points and/or shade different lines (highlight some participants, highlight the average trend line but not the individual level lines).

```
gg5 <- ggplot(oysup_long, aes(x = jitter(as.numeric(grade)), y = neuro)) +
  geom_point() + stat_smooth()
gg5
```

```
## `geom_smooth()` using method = 'gam'
```



7. Look at the correlations of your DV across time.

```
neuro <- oysup_wide %>%
  select(neuro_10:neuro_9)
cor(neuro, use = "complete.obs")
```

```
##           neuro_10   neuro_7   neuro_8   neuro_9
## neuro_10 1.0000000 0.3841322 0.3918898 0.4872357
## neuro_7  0.3841322 1.0000000 0.4245971 0.4349766
## neuro_8  0.3918898 0.4245971 1.0000000 0.4717272
## neuro_9  0.4872357 0.4349766 0.4717272 1.0000000
```