Describing and Plotting Longitudinal Data

Andrew Zieffler

Role of Description and Plotting

Examination of the data — In exploratory analysis this is more extensive; little known about structure of the data and effects.

Primarily we plot the change curves
- Individual change curves
- Aggregate (e.g., mean) change curve
- Condition on predictors/covariates of interest

Primarily we describe the change over time
- Aggregate (e.g., mean) change
- Variation in the change
- Correlation structure among the repeated measures

Example 1: A researcher wants to examine whether students' reading scores have changed over time. She has 5th–8th-grade reading score data for $n = 22$ students who were enrolled in Minneapolis public schools. She also has information on several covariates, including the students' sex, risk status, minority status, ELL status, special education status, and attendance rate.

```
# Read in minneapolis reading data
> mpls = read.csv(file = "~/epsy-8282/data/minneapolis.csv")
> head(mpls)

  studentID read.5 read.6 read.7 read.8 atRisk female minority ell sped  att
1         1    172    185    179    194      1      1        1   0    0 0.94
2         2    200    210    209     NA      1      1        1   0    0 0.91
3         3    191    199    203    215      1      0        1   0    0 0.97
4         4    200    195    194     NA      1      1        1   0    0 0.88
5         5    207    213    212    213      1      1        1   0    0 0.85
6         6    191    189    206    195      1      0        1   0    0 0.90


# Load the tidyr library
> library(tidyr)

# Convert to long data
> mpls_long = mpls %>% gather(grade, read, read.5:read.8)
```

Examining missing data patterns

To explore the missing data, we will first create a data frame that gathers the original data into three variables: (1) case/ID; (2) variable name; and (3) value for that variable. For example, the first student's data would look like this:

```
   studentID variable  value
1          1   read.5 172.00
2          1   read.6 185.00
3          1   read.7 179.00
4          1   read.8 194.00
5          1   atRisk   1.00
6          1   female   1.00
7          1 minority   1.00
8          1      ell   0.00
9          1     sped   0.00
10         1      att   0.94
```

We will gain use the gather() function to do this, except this time we will gather all the variables except studentID. We will also create a new variable that indicates whether the value is missing.

```
# Create data frame
> mpls_miss = mpls %>%
      gather(variable, value, read.5:att) %>%
      mutate(missing = na.omit(value))
```

```
> head(mpls_miss)
```

```
  studentID variable value missing
1         1   read.5   172   FALSE
2         2   read.5   200   FALSE
3         3   read.5   191   FALSE
4         4   read.5   200   FALSE
5         5   read.5   207   FALSE
6         6   read.5   191   FALSE
```

Now we can group by variable and count the number of missing cases. It is useful to know the counts and proportions of missing data for each variable.

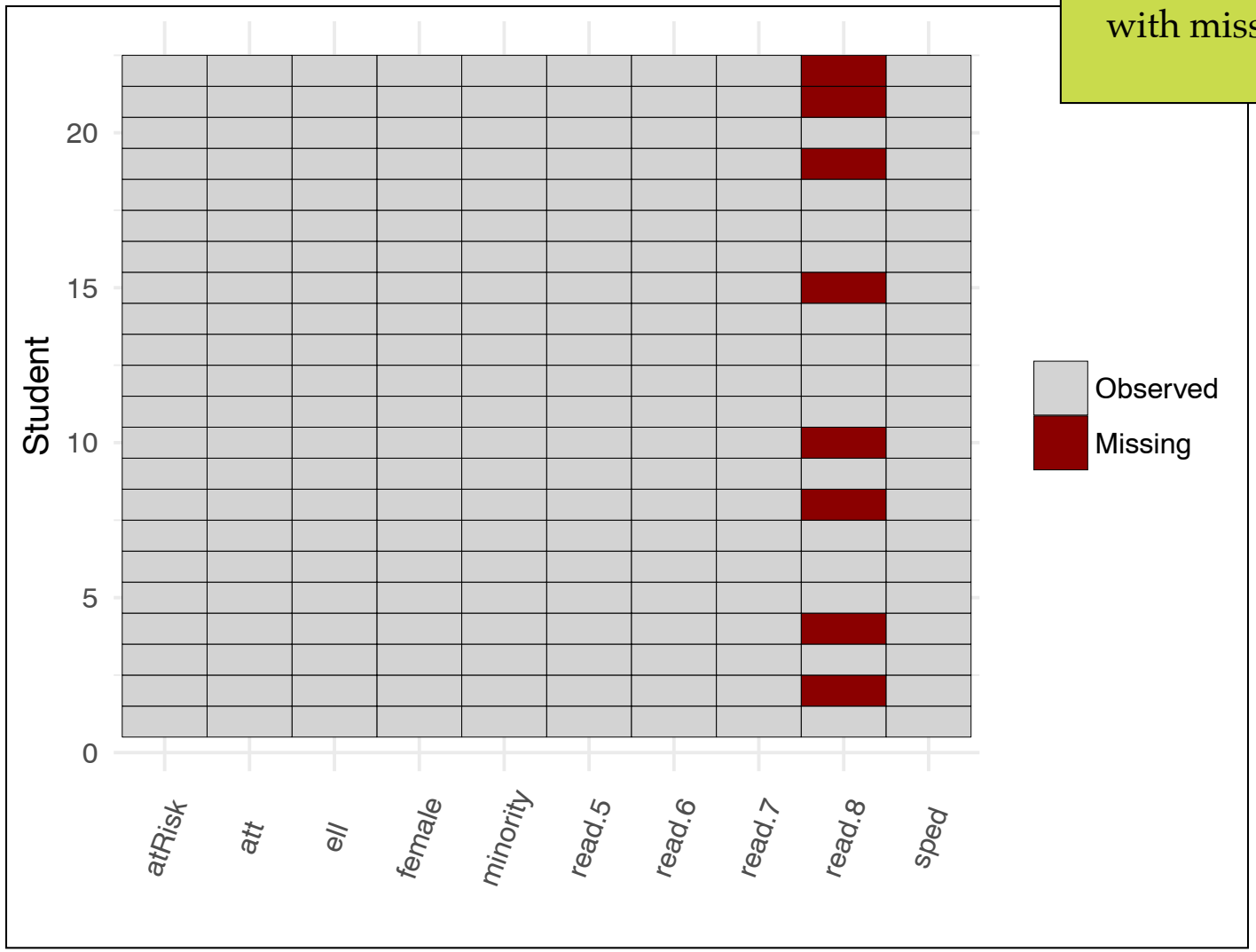
```
# Count missing data for each variable
> mpls_miss %>%
      group_by(variable) %>%
      summarize(
        n_miss = sum(missing),
        prop_miss = sum(missing) / length(missing)
      )
```

```
 1   atRisk      0 0.0000000
 2      att      0 0.0000000
 3      ell      0 0.0000000
 4   female      0 0.0000000
 5 minority      0 0.0000000
 6   read.5      0 0.0000000
 7   read.6      0 0.0000000
 8   read.7      0 0.0000000
 9   read.8      8 0.3636364
10     sped      0 0.0000000
```

The only variable with missing data is 8th-grade reading score (the last repeated measure). This is a common attrition pattern; later measurement waves tend to have higher rates of missing data.

When you have many variables, it can be better to plot the observed/missingness. To do this we plot a heatmap with case/ID on the y-axis and variables on the x-axis. We can fill the tiles by the missing value (TRUE/FALSE).

Visually we can see that the only variable with missing data is 8th-grade reading scores.



It can also be useful to see if the missingness is related to other covariates. To do this we add a column on to the original data that indicates whether the 8th-grade reading score is missing, then we can use functions from the corrr package to examine the correlations with each of the other variables.

```
> library(corrr)

# Explore whether missingness is related to covariates
> mpls %>%
      mutate(missing = na.omit(read.8)) %>%
      correlate() %>%
      focus(-studentID, -read.8, mirror = TRUE) %>%
      select(rowname, missing)
```

```
 1    read.5  0.18821475
 2    read.6  0.15602490
 3    read.7  0.10218489
 4    atRisk -0.06900656
 5    female  0.37500000
 6 minority  0.05241424
 7       ell -0.02503131
 8      sped -0.02503131
 9       att -0.07717393
10  missing           NA
```

The missingness is not too highly correlated with any of the repeated measures (students with higher 5th-grade scores tend to have higher missingness, but it is of rather small magnitude). The only covariate it correlates with is sex. Females are much more likely to have missingness. (In fact, of the 8 students with missingness, 7 are female.)

What does this indicate about whether the data are MCAR, MAR, or not MAR?

There are functions in several packages in R that are useful for examining missing data patterns. Here are some of those packages:

- VIM
- Amelia
- mice

From Obradović, Long, Cutuli, Chan, Hinz, Heistad,& Masten (2009).

Table 2. *Missing data frequencies (percentages) by GC and risk status across three assessment waves*

No. Missing Time Points	GC 2			GC 3			GC 4			GC 5		
	Adv.	Poverty	H/HM	Adv.	Poverty	H/HM	Adv.	Poverty	H/HM	Adv.	Poverty	H/HM
						Reading						
3[a]	2 (0.2)	12 (0.5)	2 (0.6)	5 (0.6)	32 (1.3)	5 (1.4)	11 (1.2)	33 (1.3)	4 (1.2)	6 (0.7)	40 (1.6)	10 (2.8)
2	161 (16.9)	634 (26.6)	125 (36.8)	155 (18.1)	586 (24.4)	126 (35.5)	170 (18.5)	591 (23.9)	116 (34.1)	179 (21.3)	691 (27.3)	137 (37.8)
1	117 (12.3)	480 (20.2)	97 (28.5)	98 (11.5)	486 (20.3)	113 (31.8)	149 (16.2)	512 (20.7)	108 (31.8)	80 (9.5)	468 (18.5)	86 (23.8)
0	675 (70.7)	1253 (52.7)	116 (34.1)	596 (69.8)	1295 (54.0)	111 (31.3)	588 (64.1)	1341 (54.1)	112 (32.9)	577 (68.5)	1334 (52.7)	129 (35.6)
Total[b]	955 (26.0)	2379 (64.8)	340 (9.3)	854 (23.7)	2399 (66.5)	355 (9.8)	918 (24.6)	2477 (66.3)	340 (9.1)	842 (22.5)	2533 (67.8)	362 (9.7)
						Math						
3[a]	1 (0.1)	10 (0.4)	4 (1.2)	5 (0.6)	28 (1.2)	3 (0.8)	11 (1.2)	31 (1.3)	3 (0.9)	5 (0.6)	42 (1.7)	11 (3.0)
2	163 (17.1)	632 (26.6)	123 (36.2)	154 (18.0)	588 (24.5)	126 (35.5)	169 (18.4)	595 (24.0)	117 (34.4)	178 (21.1)	689 (27.2)	134 (37.0)
1	119 (12.5)	491 (20.6)	100 (29.4)	98 (11.5)	483 (20.1)	115 (32.4)	154 (16.8)	515 (20.8)	110 (32.4)	84 (10.0)	477 (18.8)	90 (24.9)
0	672 (70.4)	1246 (52.4)	113 (33.2)	597 (69.9)	1300 (54.2)	111 (31.3)	584 (63.6)	1336 (53.9)	110 (32.4)	575 (68.3)	1325 (52.3)	127 (35.1)
Total[b]	955 (26.0)	2379 (64.8)	340 (9.3)	854 (23.7)	2399 (66.5)	355 (9.8)	918 (24.6)	2477 (66.3)	340 (9.1)	842 (22.5)	2533 (67.8)	362 (9.7)

Note: GC, grade cohort; Adv., advantaged students; H/HM, homeless/highly mobile.
[a]A very small percentage of students who are missing all three time points on one achievement outcome are included in the whole sample descriptives because they have data on the other achievement outcome. However, only students who had at least one time point were included in the analysis.
[b]The percentages for the total are based on the total frequency within the GC.

"As one would expect in a large urban district with high levels of mobility and poverty, many children had missing data for a specific test ad- ministration. Table 2 shows the patterns of missing data by GC and risk level across three waves of assessment. The majority of students in the poverty and advantaged groups had com- plete data across all four GCs. The majority of H/HM students did not have complete data. To maximize the information collected in this study, we used statistical procedures that al- lowed us to include all participants who had data for at least one time point …" (p. 498).

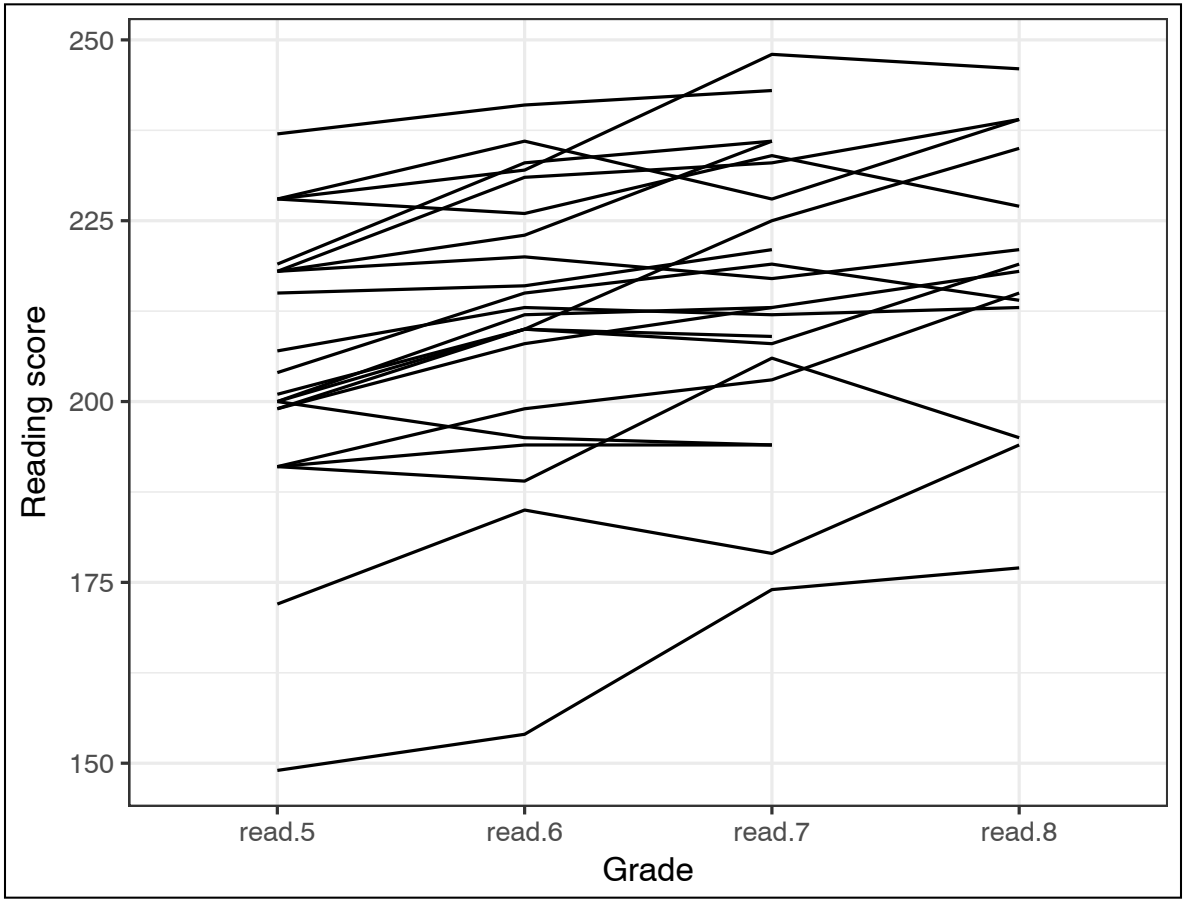
Examining individual change patterns

```
> library(ggplot2)

# Plot individual curves (spaghetti plot)
> ggplot(data = mpls_long, aes(x = grade, y = read)) +
      geom_line(aes(group = studentID)) +
      theme_bw() +
      xlab("Grade") +
      ylab("Reading score")
```

Use the long-formatted data for plotting with **ggplot2**.

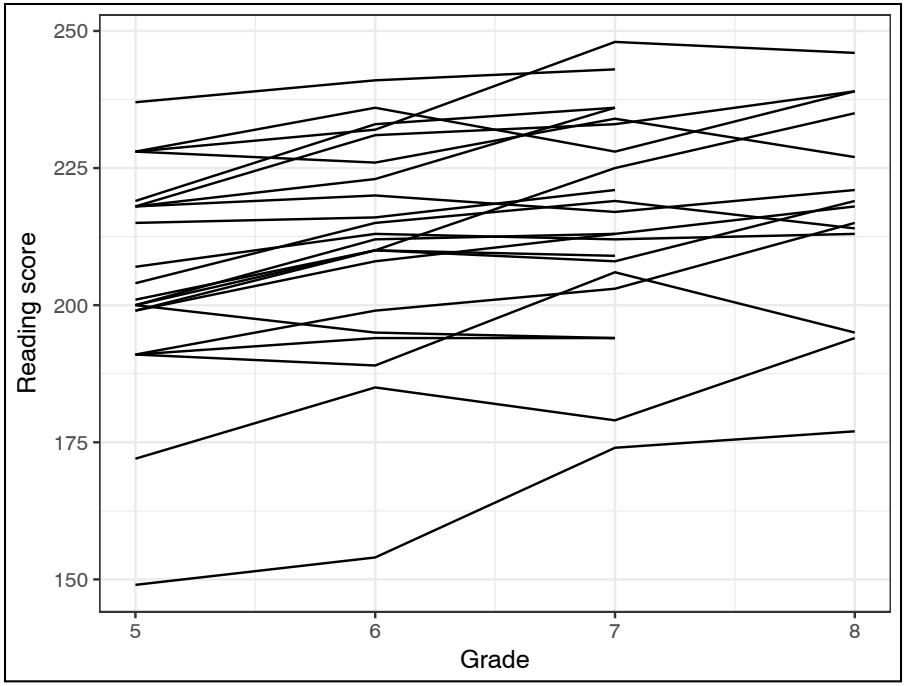
Note that grade is strings (not numbers) and is plotted alphabetically…which in this case happens to be in the right order.

Best to turn it into a number.



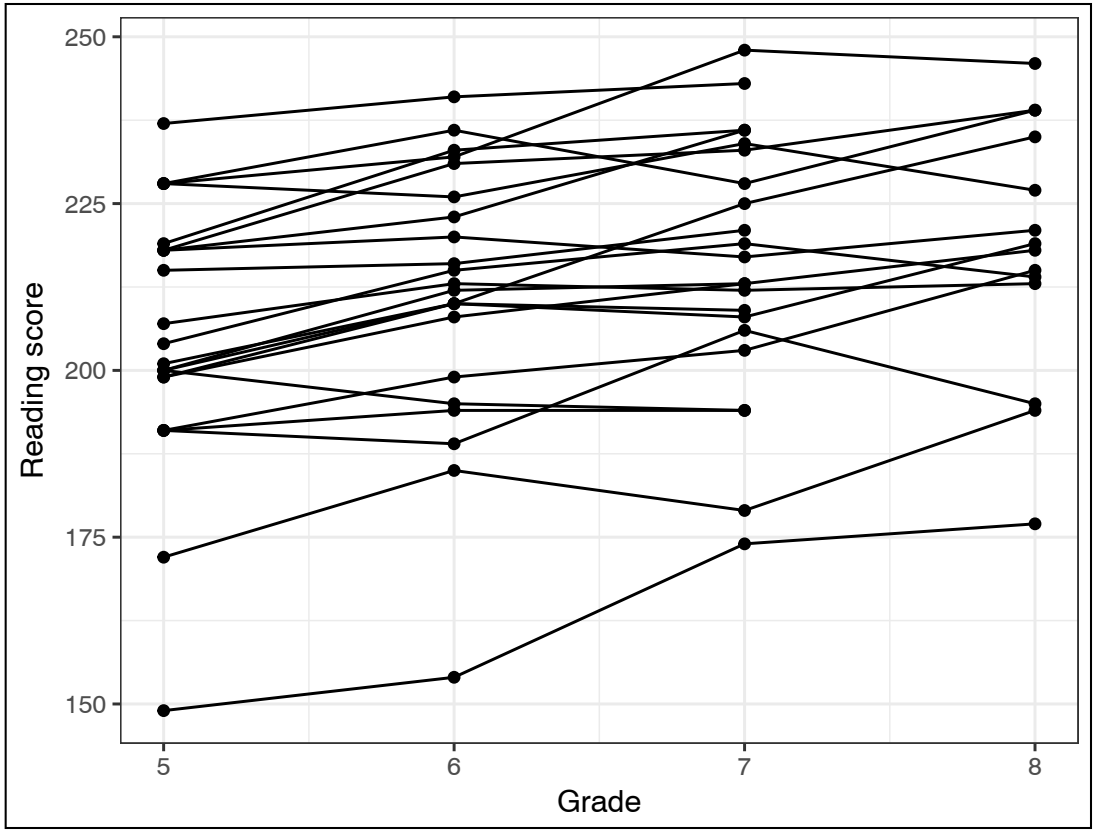
```
> mpls_long = mpls_long %>%
    mutate(grade2 = as.factor(as.integer(grade)) + 4)

> head(mpls_long)

  studentID atRisk female minority ell sped  att  grade read grade2
1         1      1      1        1   0    0 0.94 read.5  172      5
2         1      1      1        1   0    0 0.94 read.6  185      6
3         1      1      1        1   0    0 0.94 read.7  179      7
4         1      1      1        1   0    0 0.94 read.8  194      8
5         2      1      1        1   0    0 0.91 read.5  200      5
6         2      1      1        1   0    0 0.91 read.6  210      6


# Plot individual curves (spaghetti plot)
> ggplot(data = mpls_long, aes(x = grade2, y = read)) +
      geom_line(aes(group = studentID)) +
      theme_bw() +
      xlab("Grade") +
      ylab("Reading score")
```

Same plot, but now the values on the x-axis are numeric. This plot, called a spaghetti plot, suggests that reading scores tend to increase over time. It also shows variation in students' 5th-grade reading scores (initial measurement).



```
# Plot individual curves and points (spaghetti plot)
> ggplot(data = mpls_long, aes(x = grade2, y = read)) +
      geom_line(aes(group = studentID)) +
      geom_point(aes(group = studentID)) +
      theme_bw() +
      xlab("Grade") +
      ylab("Reading score")
```

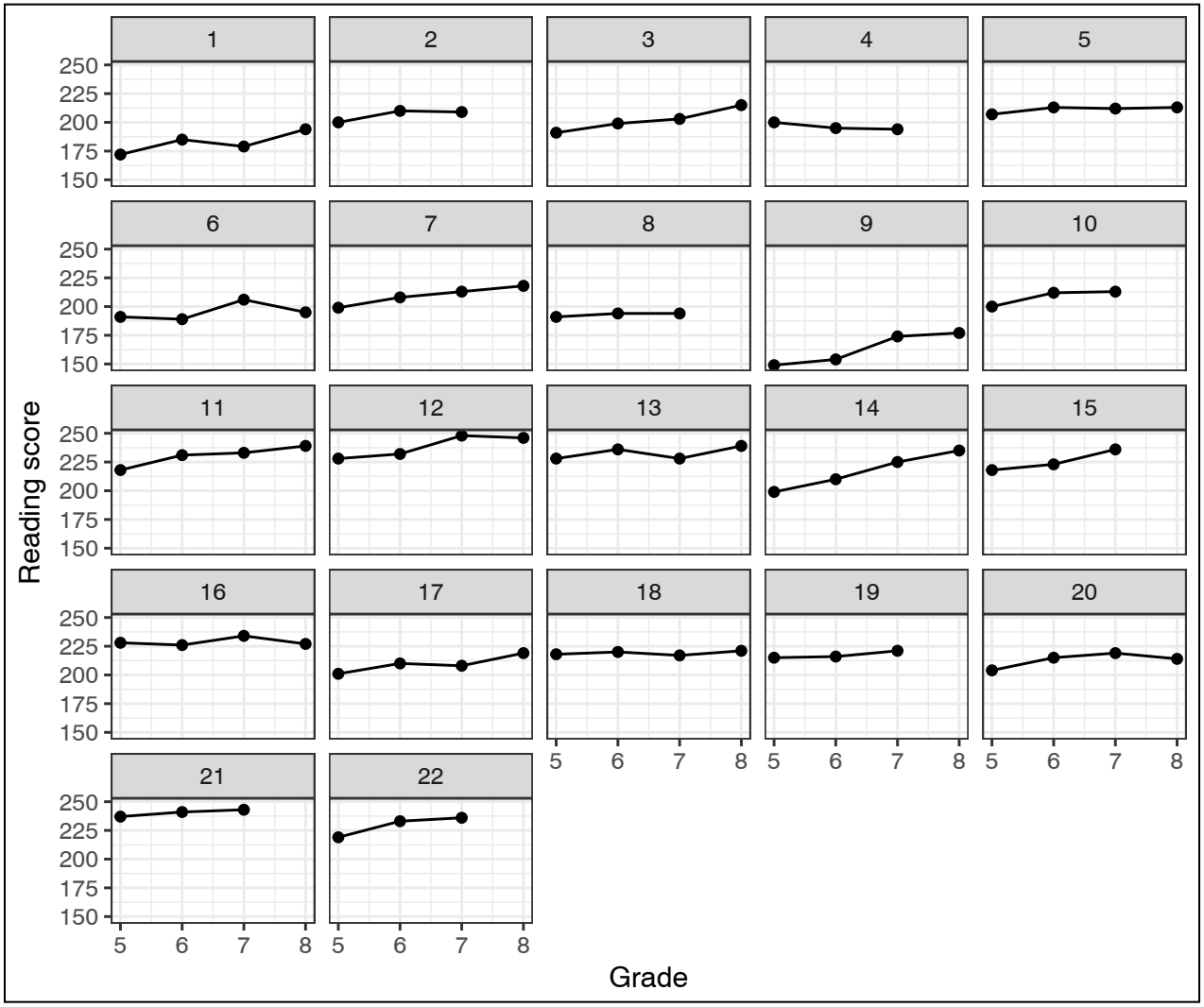
You can add the actual observed reading scores as points, but you don't need to.



It is often useful to show each individual curve in its own panel.

```
# Plot individual curves and points (spaghetti plot)
> ggplot(data = mpls_long, aes(x = grade2, y = read)) +
      geom_line(aes(group = studentID)) +
      geom_point(aes(group = studentID)) +
      theme_bw() +
      xlab("Grade") +
      ylab("Reading score") +
      facet_wrap(~studentID)
```

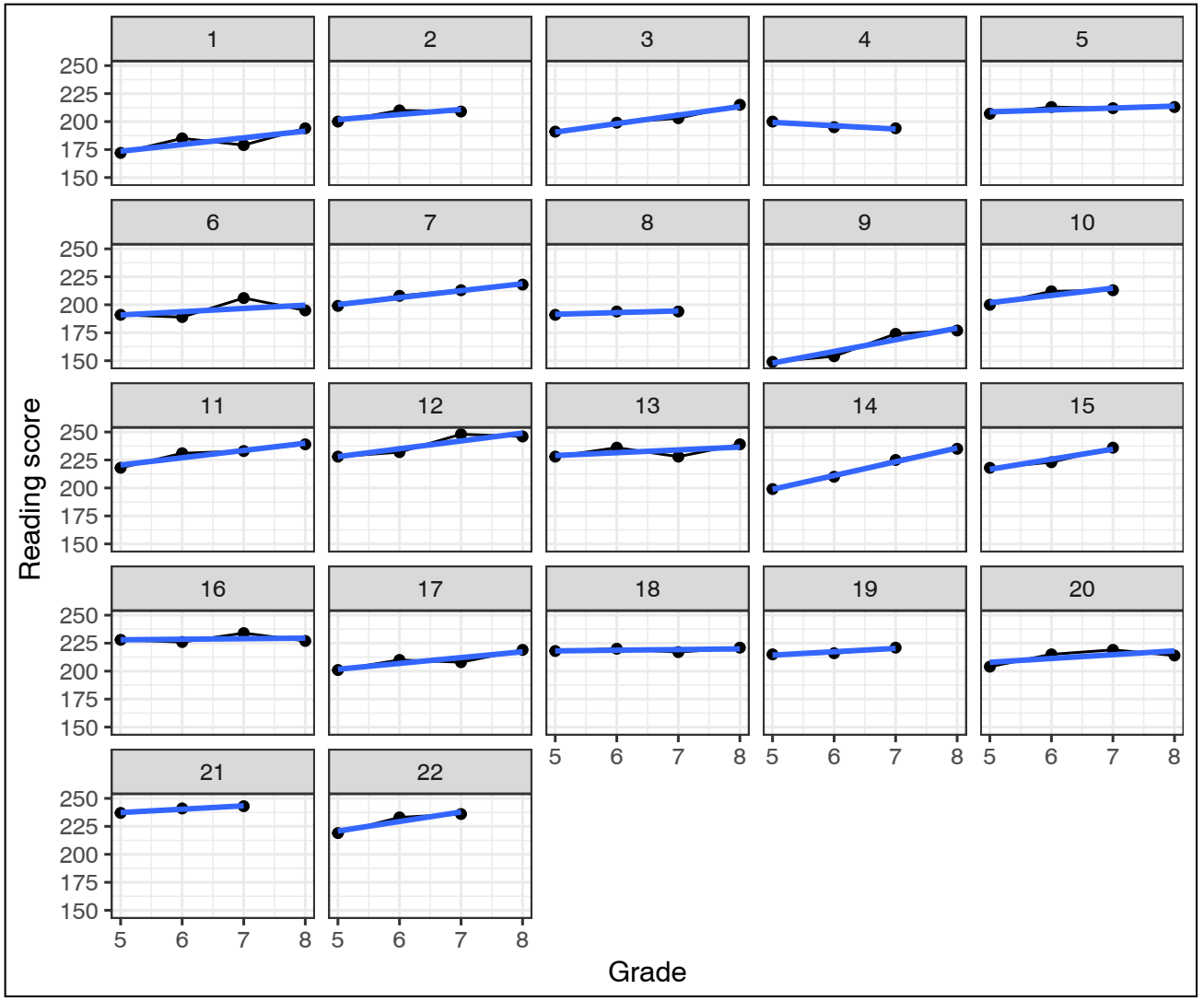
Here we can see that while most students' reading scores tend to increase, some are pretty flat (e.g., ID = 8). We can also see that some students are missing data for the 8th-grade (e.g., ID = 2) and that most change curves look relatively linear.



Add each student's regression line (based on her own data) to the individual panels.

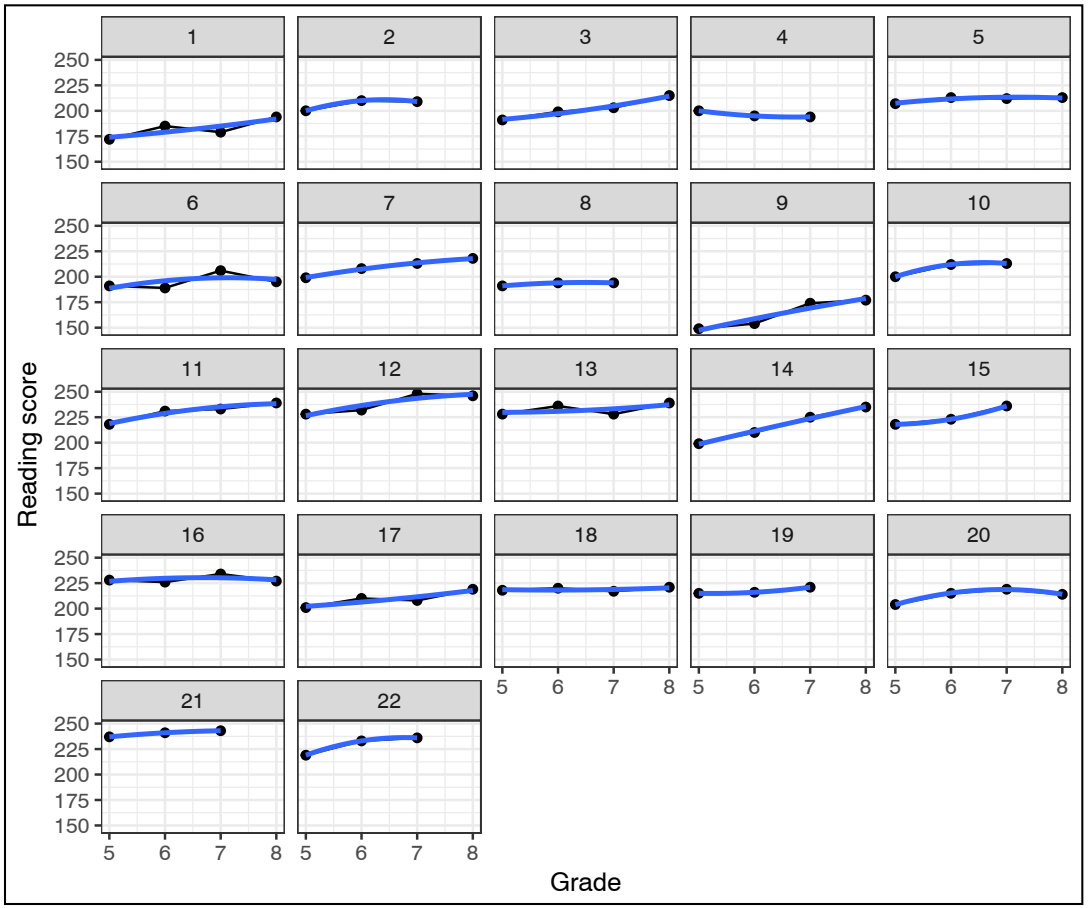
```
# Plot individual curves and points (spaghetti plot)
> ggplot(data = mpls_long, aes(x = grade2, y = read)) +
      geom_line(aes(group = studentID)) +
      geom_point(aes(group = studentID)) +
      geom_smooth(aes(group = studentID), method = "lm", se = FALSE) +
      theme_bw() +
      xlab("Grade") +
      ylab("Reading score") +
      facet_wrap(~studentID)
```

The linear trend seems to be a good fit for most students.



If you are unsure whether the trend is linear, you can change the formula to a higher order polynomial.

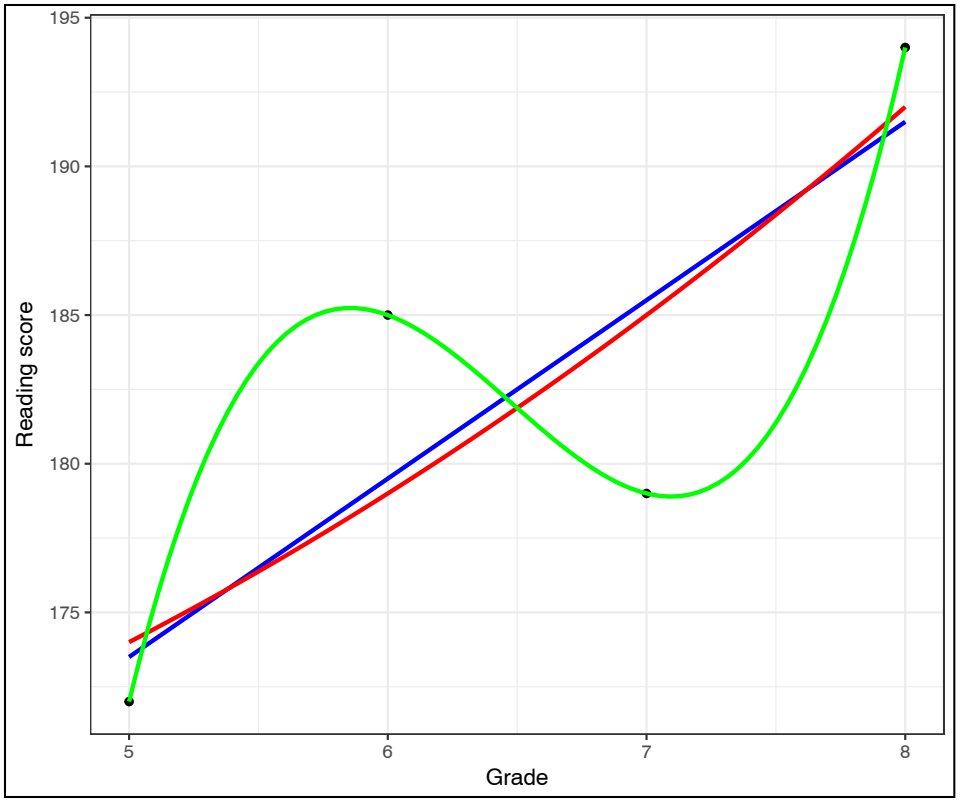
```
# Plot individual curves and points (spaghetti plot)
> ggplot(data = mpls_long, aes(x = grade2, y = read)) +
      geom_line(aes(group = studentID)) +
      geom_point(aes(group = studentID)) +
      geom_smooth(aes(group = studentID), method = "lm", se = FALSE,
        formula = y~poly(x, 2)) +
      theme_bw() +
      xlab("Grade") +
      ylab("Reading score") +
      facet_wrap(~studentID)
```

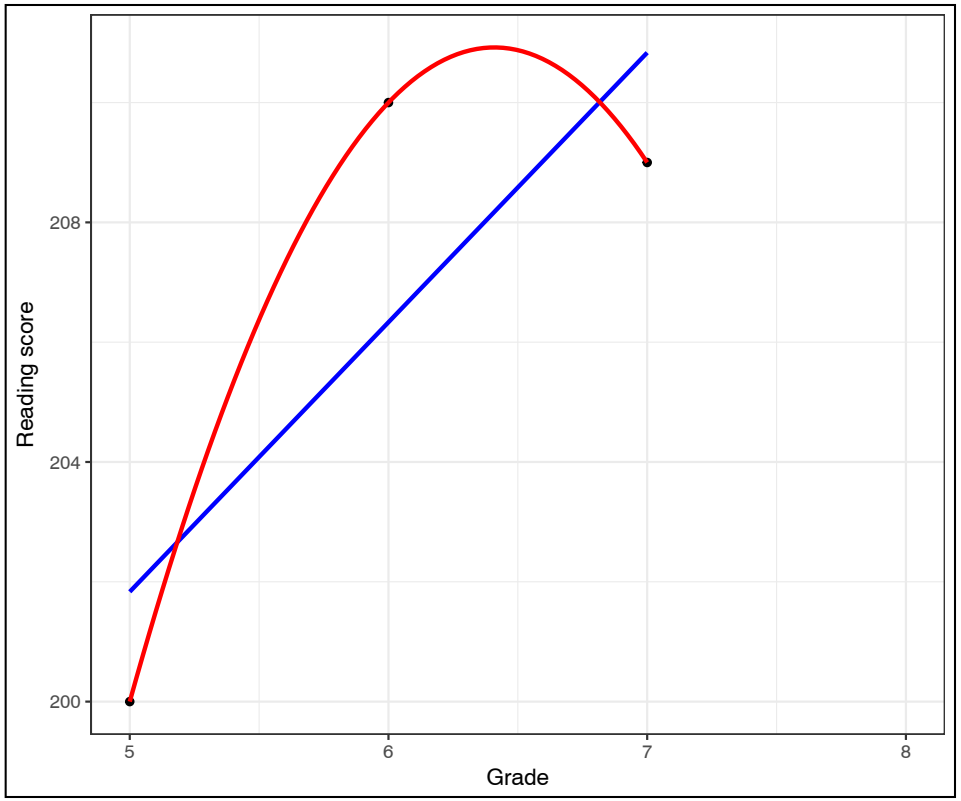


`y~poly(x, 2)` fits a quadratic (second-order) polynomial to the data.

In this case, it looks like the quadratic and linear trends fit about equally well.

Note: The order of the polynomial must be less than the number of measurement waves. With four measurement waves we could fit up to a third-degree polynomial (cubic).





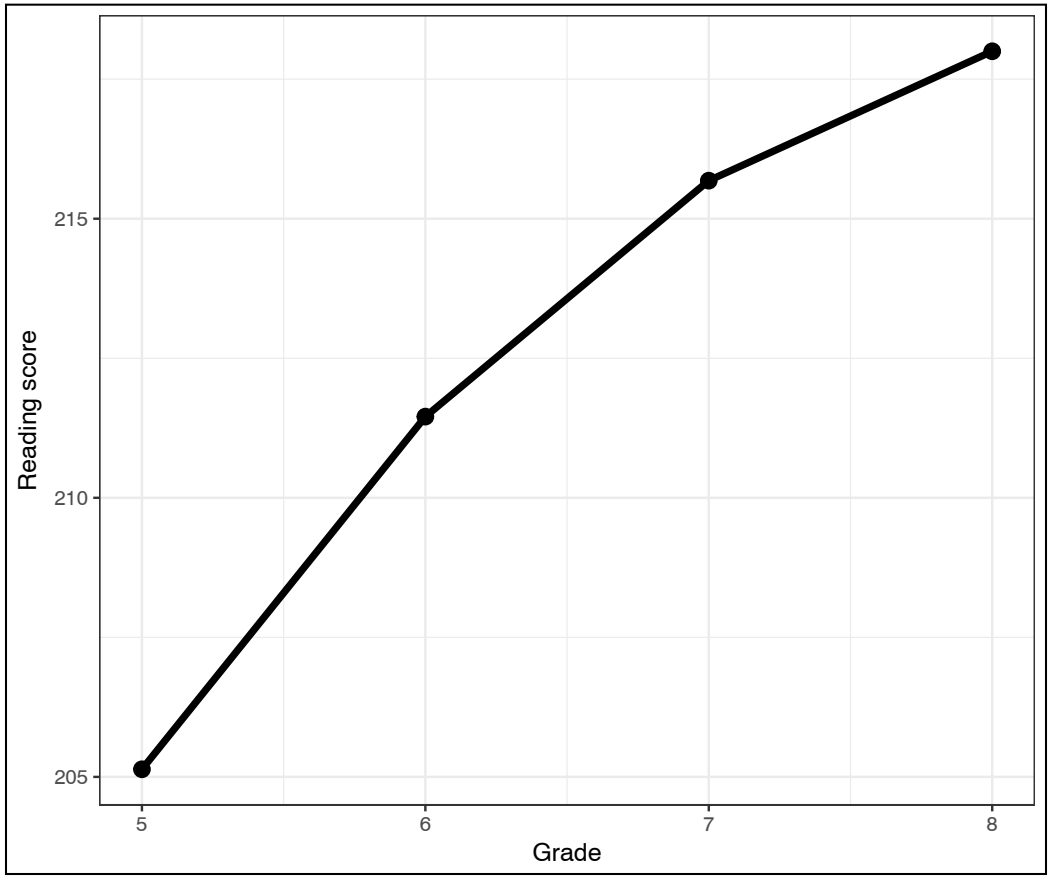
The plots show the linear (blue), quadratic (red), and cubic (green) fit curves for Student 1 (LEFT) and Student 2 (RIGHT). Student 2 did not have an 8th-grade measurement so the cubic could not be fitted.

Also note that for *k* measurment waves, if we fit the polynomial curve with degree *k*–1 the curve fits the data perfectly (no error). This is a saturated model. You have used up all possible *df*. While this sounds attractive, you have probably overfitted the data (less generalizable). Since the *df* are used up, you can also not include any covariates in the model —- there is no residual error to explain.

Examining the mean change pattern

To plot the observed mean change pattern, we use the `stat_summary()` function. We need to consider all students in the mean computation, so we cannot group by studentID. We instead use `group=1` to use the whole dataset.

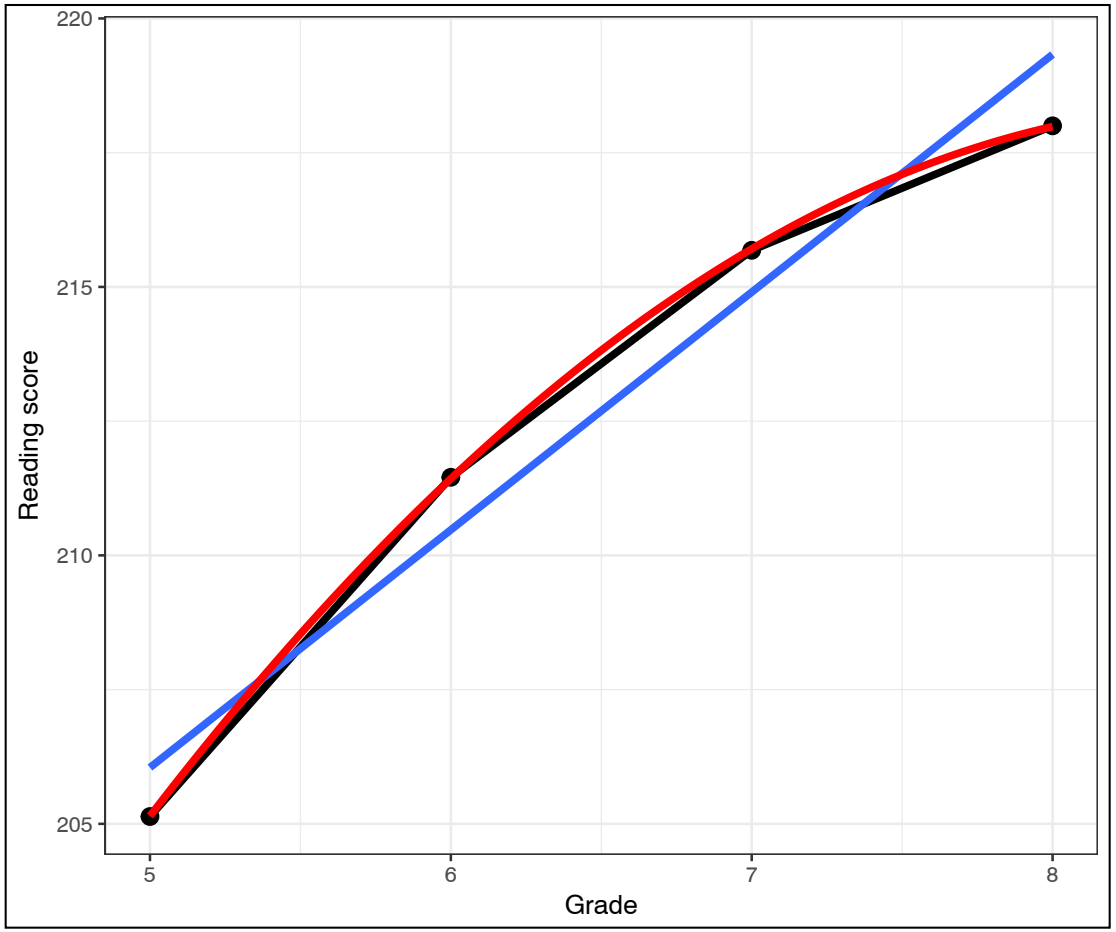
```
# Plot observed mean pattern
> ggplot(data = mpls_long, aes(x = grade2, y = read)) +
      stat_summary(group = 1, fun.y = mean, geom = "line", lwd = 1.5) +
      stat_summary(group = 1, fun.y = mean, geom = "point", size = 3) +
      theme_bw() +
      xlab("Grade") +
      ylab("Reading score")
```



This plot suggests that the mean reading score increases over time. Is the increase linear? Quadratic? It looks like it might be potentially quadratic.

Here we fit the same observed mean change curve, but add the linear and quadratic regression smoothers.

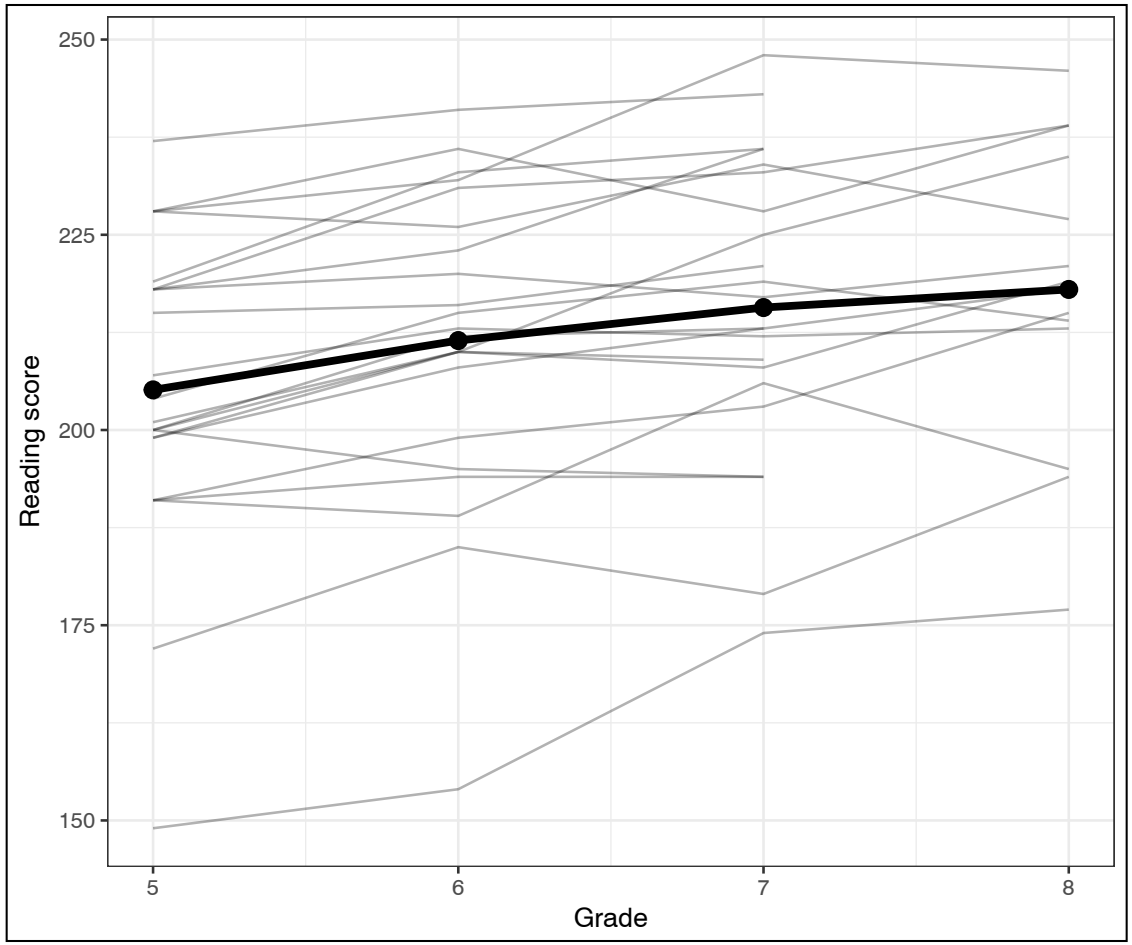
```
# Plot observed mean pattern and regression smoothers
> ggplot(data = mpls_long, aes(x = grade2, y = read)) +
      stat_summary(group = 1, fun.y = mean, geom = "line", lwd = 1.5) +
      stat_summary(group = 1, fun.y = mean, geom = "point", size = 3) +
      geom_smooth(group = 1, method = "lm", se = FALSE) +
      geom_smooth(group = 1, method = "lm", se = FALSE,
        formula = y~poly(x, 2), color = "red") +
      theme_bw() +
      xlab("Grade") +
      ylab("Reading score")
```



It appears as though the data fit the quadratic mean change curve better than the linear mean change curve.

Here we again display same observed mean change curve (without smoothers), and we add the individual change curves (with transparency).

```
# Plot individual change curves and observed mean pattern
> ggplot(data = mpls_long, aes(x = grade2, y = read)) +
      geom_line(aes(group = studentID), alpha = 0.3) +
      stat_summary(group = 1, fun.y = mean, geom = "line", lwd = 1.5) +
      stat_summary(group = 1, fun.y = mean, geom = "point", size = 3) +
      theme_bw() +
      xlab("Grade") +
      ylab("Reading score")
```



Once we add the individual change curves, the scale on the y-axis changes and the mean change curve appears to flatten out. Perhaps that quadratic relationship is really linear?

We will hold off on this decision for now, but just note that perspective matters.

Examining the mean change pattern conditioned on covariates

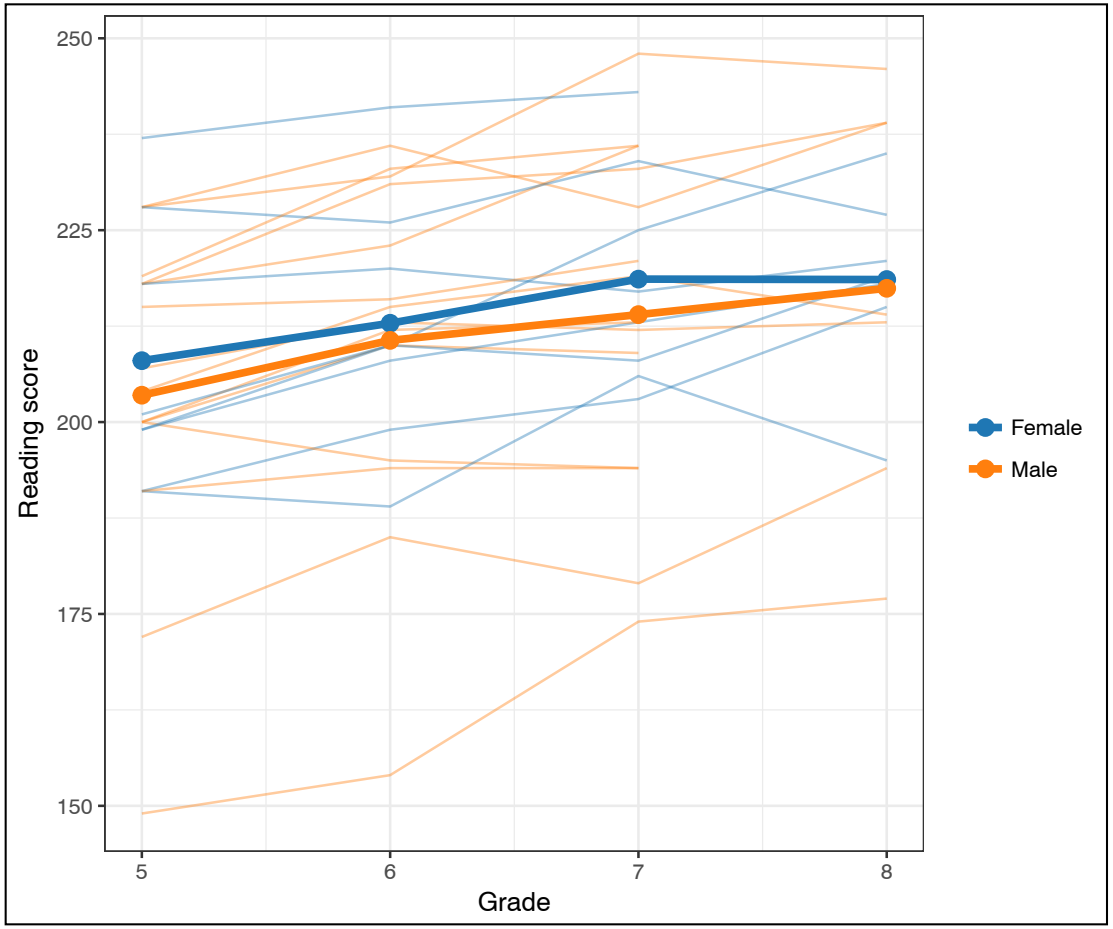
We can also condition these plots on any of the covariates to see if the individual and mean change curves differ across values of the covariate. There are two common ways of doing this: (1) use color to differentiate covariate values; and (2) use faceting to differentiate covariate values.

To create more aesthetically-pleasing plots, it is best to first coerce any categorical covariates into factors. For example our sex covariate (female) is 0s and 1s. R sees this as an integer. To have R see this as categorical, we use the `factor()` function to create a new sex variable in the dataset.

```
# Create discrete (categorical) sex variable
> mpls_long = mpls_long %>%
    mutate(sex = factor(female, levels = c(0, 1), labels = c("Male", "Female")))
> head(mpls_long)

  studentID atRisk female minority ell sped  att  grade read grade2  sex
1         1      1      1        1   0    0 0.94 read.5  172      5 Male
2         1      1      1        1   0    0 0.94 read.6  185      6 Male
3         1      1      1        1   0    0 0.94 read.7  179      7 Male
4         1      1      1        1   0    0 0.94 read.8  194      8 Male
5         2      1      1        1   0    0 0.91 read.5  200      5 Male
6         2      1      1        1   0    0 0.91 read.6  210      6 Male
```

We need to add `color=sex` to the global aesthetic. We also need to include `group=sex` in a local aesthetic for the `stat_summary()` layers. We also load the **ggsci** library to get the d3 color palette.

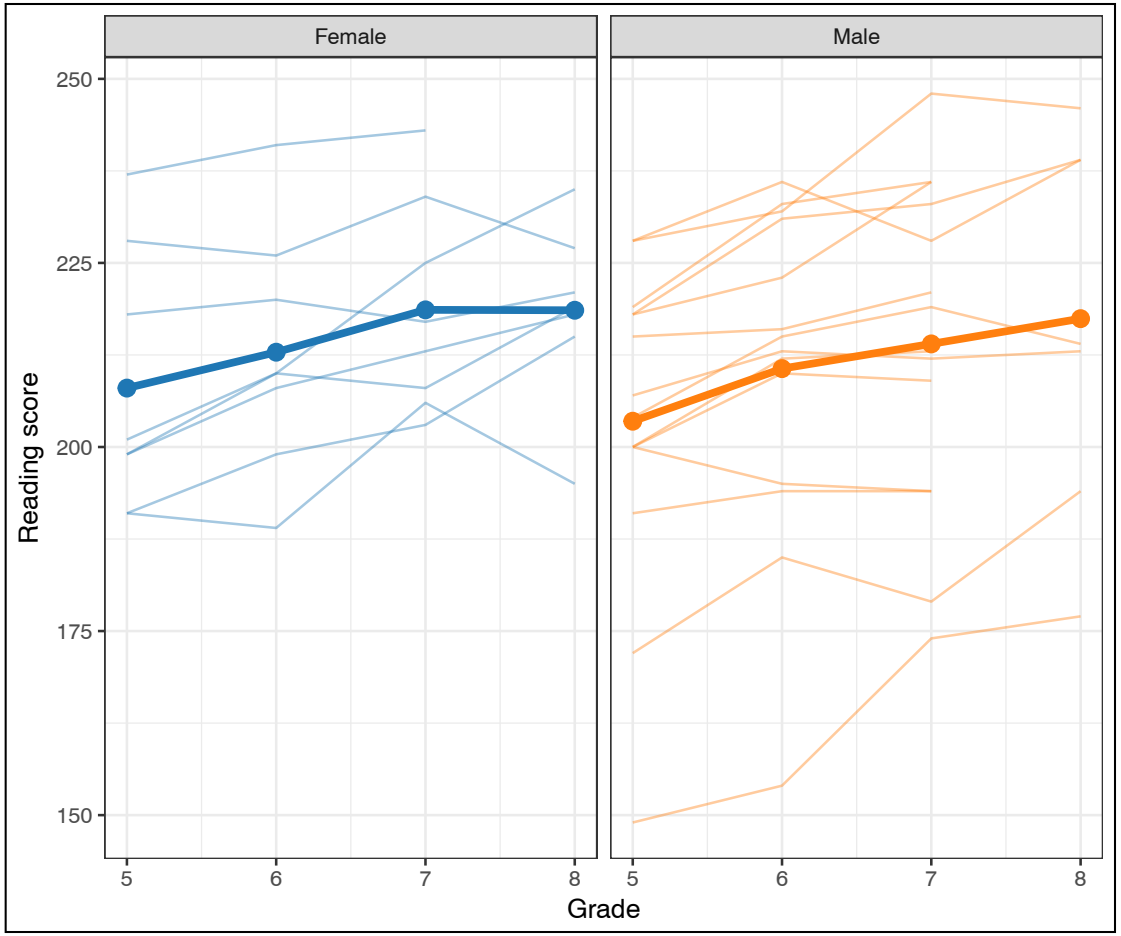
```
# Plot observed individual change curves and mean pattern by sex using color
> library(ggsci)
> ggplot(data = mpls_long, aes(x = grade2, y = read, color = sex)) +
      geom_line(aes(group = studentID), alpha = 0.3) +
      stat_summary(aes(group = sex), fun.y = mean, geom = "line", lwd = 1.5) +
      stat_summary(aes(group = sex), fun.y = mean, geom = "point", size = 3) +
      theme_bw() +
      xlab("Grade") +
      ylab("Reading score") +
      scale_color_d3(name = "")
```



This plot suggests that there are slight differences in the observed mean change in reading scores between females and males. Females have a higher mean reading score than males at all measurement waves. This difference, however, seems to be decreasing over time…a sex by time interaction.

Faceting can better differentiate the individual curves by sex, however, since males and females will be in different panels, the comparison of mean change patterns can be more difficult.

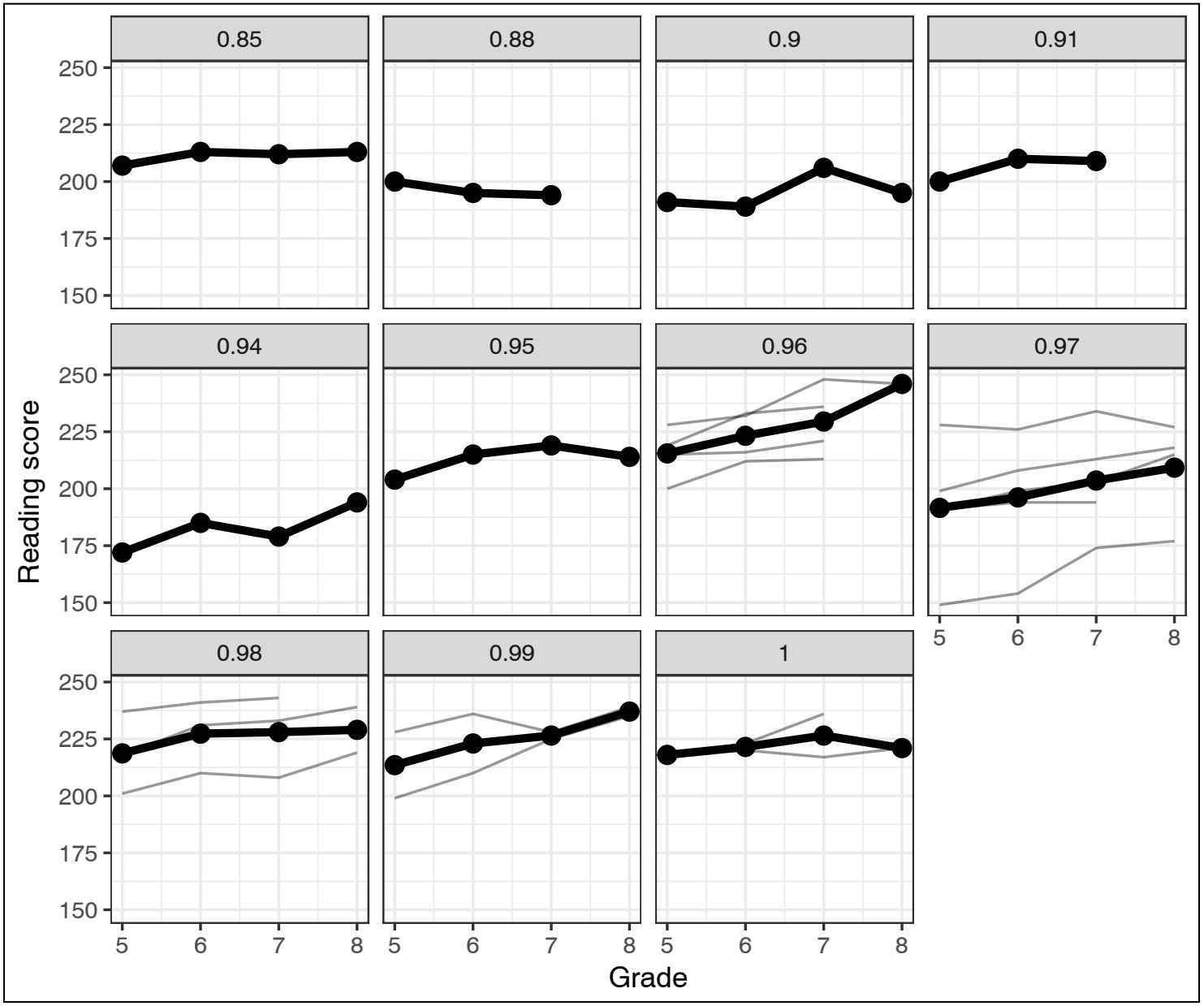
```
# Plot observed individual change curves and mean pattern by sex using faceting
> ggplot(data = mpls_long, aes(x = grade2, y = read, color = sex)) +
      geom_line(aes(group = studentID), alpha = 0.3) +
      stat_summary(aes(group = sex), fun.y = mean, geom = "line", lwd = 1.5) +
      stat_summary(aes(group = sex), fun.y = mean, geom = "point", size = 3) +
      theme_bw() +
      xlab("Grade") +
      ylab("Reading score") +
      scale_color_d3() +
      facet_wrap(~sex) +
      guides(color = FALSE)
```



This plot shows more clearly why females have a higher observed mean change curve. There are male students who have individual curves at the same level as females, but there are also males who have change curves that are lower on the scale…more variation in baseline scores. The male curves also seem to have a larger positive slope than the female curves, in general.

When covariates are continuous (e.g., attendance) plotting their effects is more difficult. Turning a continuous covariate in to a factor often results in many levels with scarce numbers of cases in each level. For example, here is what that would look like if we turned attendance into a factor.

Note that many attendance rates only include data from one student (e.g., att = 0.85). It is also difficult to think about how the mean change curve is different across this many levels of attendance.



It is often useful to split continuous variables into three or four discrete ordered categories. Limiting it to around three categories makes understanding the relationship between the covariate and the mean change pattern more understandable from a plot.

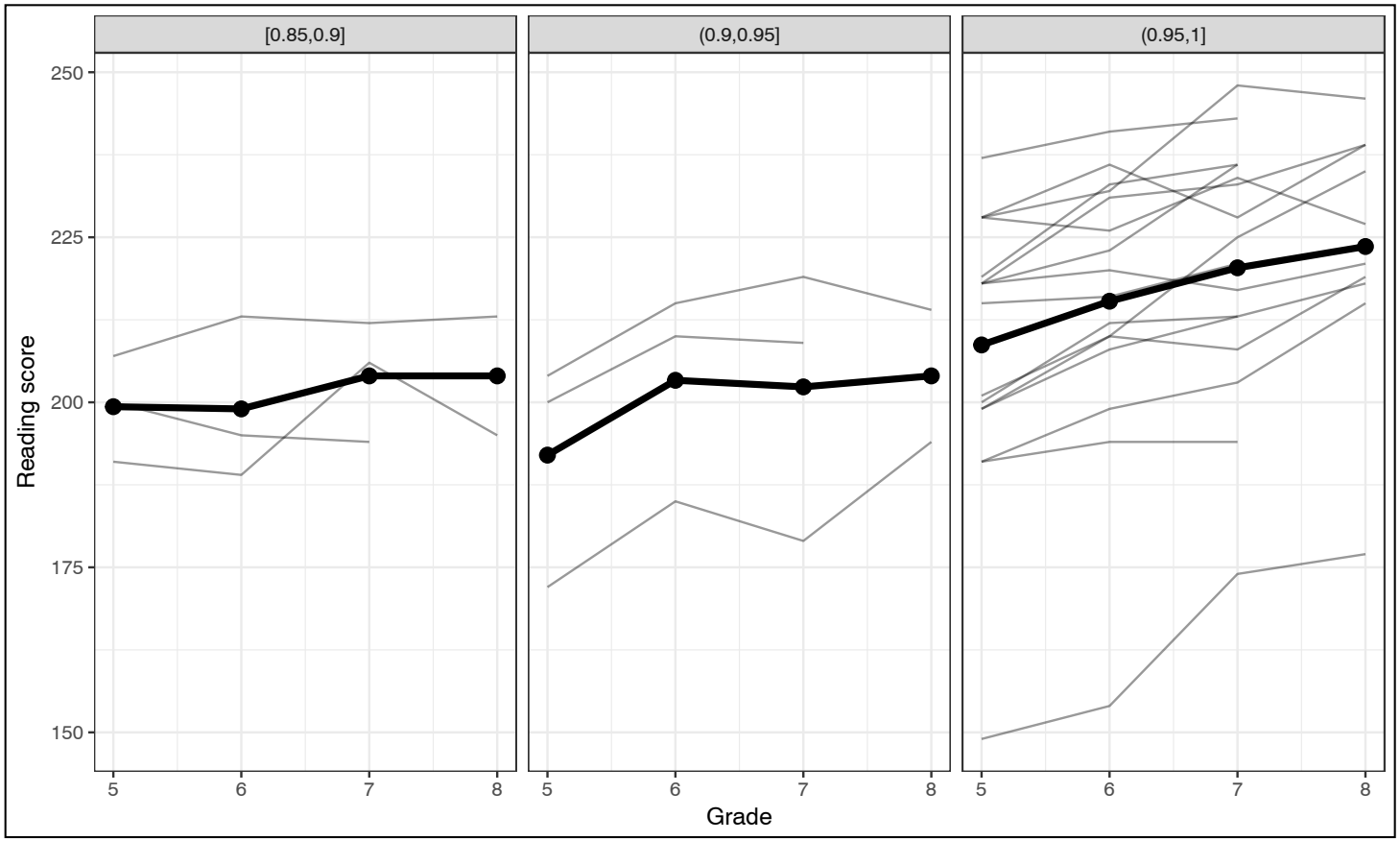
There are several ways to do this in R. Below I use the `cut_interval()` function from the **ggplot2** library.

```
# Discretize attendance variable
> mpls_long = mpls_long %>%
    mutate(att_category = cut_interval(att, n = 3))
> head(mpls_long)

  studentID atRisk female minority ell sped  att  grade read grade2  sex
1         1      1      1        1   0    0 0.94 read.5  172      5 Male
2         1      1      1        1   0    0 0.94 read.6  185      6 Male
3         1      1      1        1   0    0 0.94 read.7  179      7 Male
4         1      1      1        1   0    0 0.94 read.8  194      8 Male
5         2      1      1        1   0    0 0.91 read.5  200      5 Male
6         2      1      1        1   0    0 0.91 read.6  210      6 Male
  att_category
1   (0.9,0.95]
2   (0.9,0.95]
3   (0.9,0.95]
4   (0.9,0.95]
5   (0.9,0.95]
6   (0.9,0.95]
```

```
# Plot observed individual change curves and mean pattern by discretized attendance
using faceting
> ggplot(data = mpls_long, aes(x = grade2, y = read)) +
      geom_line(aes(group = studentID), alpha = 0.3) +
      stat_summary(aes(group = att_category), fun.y = mean,
        geom = "line", lwd = 1.5) +
      stat_summary(aes(group = att_category), fun.y = mean,
        geom = "point", size = 3) +
      theme_bw() +
      xlab("Grade") +
      ylab("Reading score") +
      facet_wrap(~att_category, nrow = 1)
```

It looks as though the students in the two lowest attendance categories have very little change in mean reading score over time. For students who have the highest attendance rates, they not only have a higher average 5th-grade reading score, but the mean reading score increases over time.

This may be evidence of a time by attendance interaction effect.



From Obradović, et al. (2009).

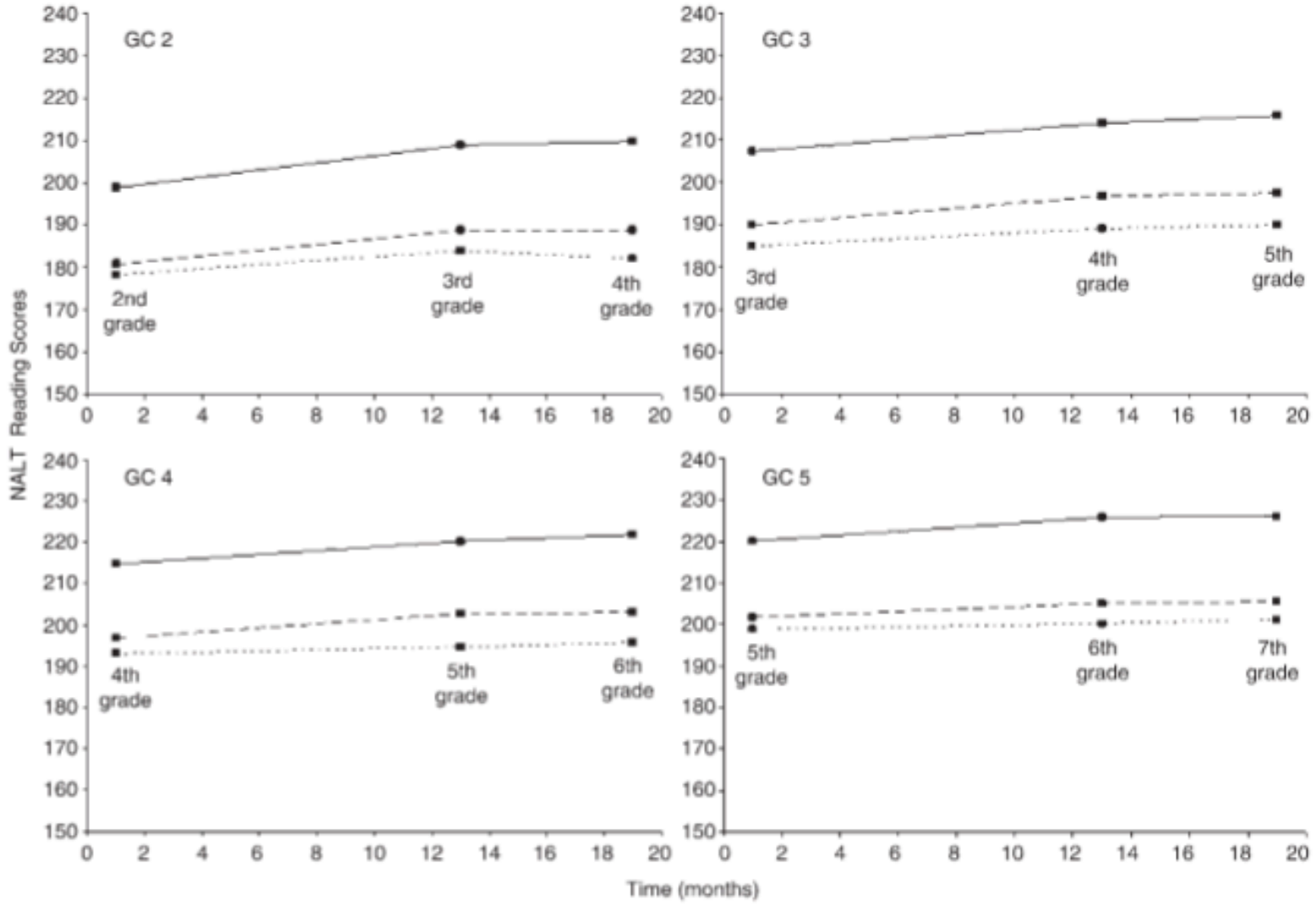


Figure 1. The observed means of reading achievement for (—) advantaged, (– – –) poverty, and (- - -) homeless and highly mobile groups over time by grade cohort (GC).

Computing summaries to describe the change pattern

We can actually compute summary statistics to further describe the change pattern. In general, it is useful to compute (1) the mean at each measurement wave, and (2) the SD or variance at each measurement wave. If you have important covariates, you can compute the mean and SD at each measurement wave conditioned on those covariates as well.

```
# Compute the mean and sd for each measurement wave
> mpls_long %>%
    group_by(grade2) %>%
    summarize(M = mean(read), SD = sd(read))

  grade2        M       SD
   <dbl>    <dbl>    <dbl>
1      5 205.1364 19.99356
2      6 211.4545 20.06116
3      7 215.6818 19.44562
4      8       NA       NA
```

Many computations, including `mean()` and `sd()` return NA if there are any NA values in the data.

To remedy this, only compute on the complete data.

```
# Compute the mean and sd for each measurement wave
> mpls_long %>%
    na.omit() %>%
    group_by(grade2) %>%
    summarize(M = mean(read), SD = sd(read))

  grade2        M       SD
   <dbl>    <dbl>    <dbl>
1      5 205.1364 19.99356
2      6 211.4545 20.06116
3      7 215.6818 19.44562
4      8 218.0000 19.37881
```

The mean values show an increase over time. The variation in reading scores seems fairly constant over time (homogeneity of variance).

What if we conditioned on sex?

```
# Compute the mean and sd for each measurement wave
> mpls_long %>%
    na.omit() %>%
    group_by(grade2, sex) %>%
    summarize(M = mean(read), SD = sd(read)) %>%
    arrange(sex)
```

grade2	sex	M	SD
<dbl>	<fctr>	<dbl>	<dbl>
1	5	Female	208.0000	17.44379
2	6	Female	212.8750	16.11067
3	7	Female	218.6250	14.25219
4	8	Female	218.5714	12.35391
5	5	Male	203.5000	21.77066
6	6	Male	210.6429	22.54775
7	7	Male	214.0000	22.20534
8	8	Male	217.4286	25.69603

The mean values show an increase over time for both males and females. Consistent with the plot we created earlier, females have a higher mean 5th-grade reading score than males, but both groups end up at about the same average score (males have a higher slope). The variation in reading scores seems fairly constant over time (homogeneity of variance) within sex, but male reading scores seem to have more variation than female scores.

We also want to compute the correlations (or covariances) between the repeated measures. To do this we use the `correlate()` function from the **corrr** package. This function operates on the WIDE data.

```
# Load corrr library (you may need to install it)
> library(corrr)

# Compute the correlations on the wide data
> mpls %>%
    select(read.5:read.8) %>%
    correlate() %>%
    shave() %>%
    fashion(decimals = 3)

  rowname read.5 read.6 read.7 read.8
1  read.5
2  read.6   .965
3  read.7   .914   .926
4  read.8   .883   .929   .923
```

The correlations are quite high between the repeated measures (as expected). We also see that they might be diminishing as the measurement waves are further apart. Note: By default the `correlation()` function uses *pairwise complete observations*.

If you prefer listwise deletion use `na.omit()` prior to `correlate()`.

```
# Compute the correlations - listwise deletion
> mpls %>%
    select(read.5:read.8) %>%
    na.omit() %>%
    correlate() %>%
    shave() %>%
    fashion(decimals = 3)

  rowname read.5 read.6 read.7 read.8
1  read.5
2  read.6   .976
3  read.7   .928   .913
4  read.8   .883   .929   .923
```

While the actual correlations are different, the same general pattern is observed.
How you delete is up to you, but REPORT IT!!!

You can also compute the covariances if you prefer. Remember that correlations are just standardized covariances. To compute the covariances, we use the `cov()` function. We have to tell this function how to treat missingness by employing the use= argument.

```
# Compute the variance-covariance matrix - pairwise deletion
> mpls %>%
    select(read.5:read.8) %>%
    cov(., use = "pairwise.complete.obs")

          read.5   read.6   read.7   read.8
read.5 399.7424 386.8874 355.3788 383.5385
read.6 386.8874 402.4502 361.1039 396.4615
read.7 355.3788 361.1039 378.1320 361.6923
read.8 383.5385 396.4615 361.6923 375.5385


# Compute the variance-covariance matrix - listwise deletion
> mpls %>%
    select(read.5:read.8) %>%
    cov(., use = "complete.obs")

          read.5   read.6   read.7   read.8
read.5 502.8626 481.9451 421.0714 383.5385
read.6 481.9451 485.2088 406.8132 396.4615
read.7 421.0714 406.8132 409.1044 361.6923
read.8 383.5385 396.4615 361.6923 375.5385
```

The output is the variance–covariance matrix.

	read.5	read.6	read.7	read.8
read.5	399.7424	386.8874	355.3788	383.5385
read.6	386.8874	402.4502	361.1039	396.4615
read.7	355.3788	361.1039	378.1320	361.6923
read.8	383.5385	396.4615	361.6923	375.5385

The values on the main diagonal are the variances at each measurement wave.

	read.5	read.6	read.7	read.8
read.5	399.7424	386.8874	355.3788	383.5385
read.6	386.8874	402.4502	361.1039	396.4615
read.7	355.3788	361.1039	378.1320	361.6923
read.8	383.5385	396.4615	361.6923	375.5385

The values on the off-diagonal are covariances. These are difficult to interpret directly. They are positive, implying the correlations are positive. Their magnitudes are based on the scales of the variables.

While they are less interpretable than correlations, this is the matrix that will be modeled in the analysis.

References and Source Material

- Long, J. D. (2012). *Longitudinal data analysis for the behavioral sciences using R*. Thousand Oaks, CA: Sage. (Minneapolis reading data)
- Obradović, J., Long, J., Cutuli, J., Chan, C., Hinz, E., Heistad, D., & Masten, A. (2009). Academic achievement of homeless and highly mobile children in an urban school district: Longitudinal evidence on risk, growth, and resilience. *Development and Psychopathology, 21*(2), 493-518. doi: 10.1017/S0954579409000273

- Graphics used from Open Clip Art Library (https://openclipart.org/)