STAC32

Assignment 7

Due Thursday November 7 at 11:59pm

To begin:

```
library(tidyverse)
```

```
## -- Attaching packages --------------------------------- tidyverse 1.2.1 --
```

```
## v ggplot2 3.2.1     v purrr   0.3.2
## v tibble  2.1.3     v dplyr   0.8.3
## v tidyr   1.0.0     v stringr 1.4.0
## v readr   1.3.1     v forcats 0.4.0
```

```
## -- Conflicts ------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```
library(broom)
```

You may or may not need `broom`. If running `library(broom)` gives you an error like "no such package", you'll need to install it first, with `install.packages`.

1. Work through Chapter 14 of PASIAS. There are *lots* of questions there, so work through enough of them to get the idea.

   Hand in the next (rather long) question:

2. Are graduation rates better from colleges where students enter with higher SAT scores? How do student-related expenditures (tuition, textbooks etc.) come into the picture? A sample of US colleges and universities was taken (only of institutions with between 10,000 and 20,000 students). For each college or university, three things were recorded:

   - the median SAT score (of current students when they first enrolled)
   - student-related expenditure in $ per full-time student
   - six-year graduation rate, in percent. (This is the percentage of students who graduate within six years of first enrolling at the institution.)

   The data are in `http://www.utsc.utoronto.ca/~butler/assgt_data/graduation-rates.csv` as a CSV file.

   (a) (2 marks) Read in and display the data.

   Solution: The usual gimme first part:

```
my_url="http://www.utsc.utoronto.ca/~butler/assgt_data/graduation-rates.csv"
gradrate=read_csv(my_url)
```

```
## Parsed with column specification:
## cols(
##  med_sat = col_double(),
##  expenditure = col_double(),
##  grad_rate = col_double()
## )
```

```
gradrate
```

```
## # A tibble: 15 x 3
##    med_sat expenditure grad_rate
##      <dbl>       <dbl>     <dbl>
##  1    1065        7970        49
##  2     950        6401        33
##  3    1045        6285        37
##  4     990        6792        49
##  5     950        4541        22
##  6     970        7186        38
##  7     980        7736        39
##  8    1080        6382        52
##  9    1035        7323        53
## 10    1010        6531        41
## 11    1010        6216        38
## 12     930        7375        37
## 13    1005        7874        45
## 14    1090        6355        57
## 15    1085        6261        48
```

Note, at least for yourself, that you have the right three variables (and that they are quantitative).

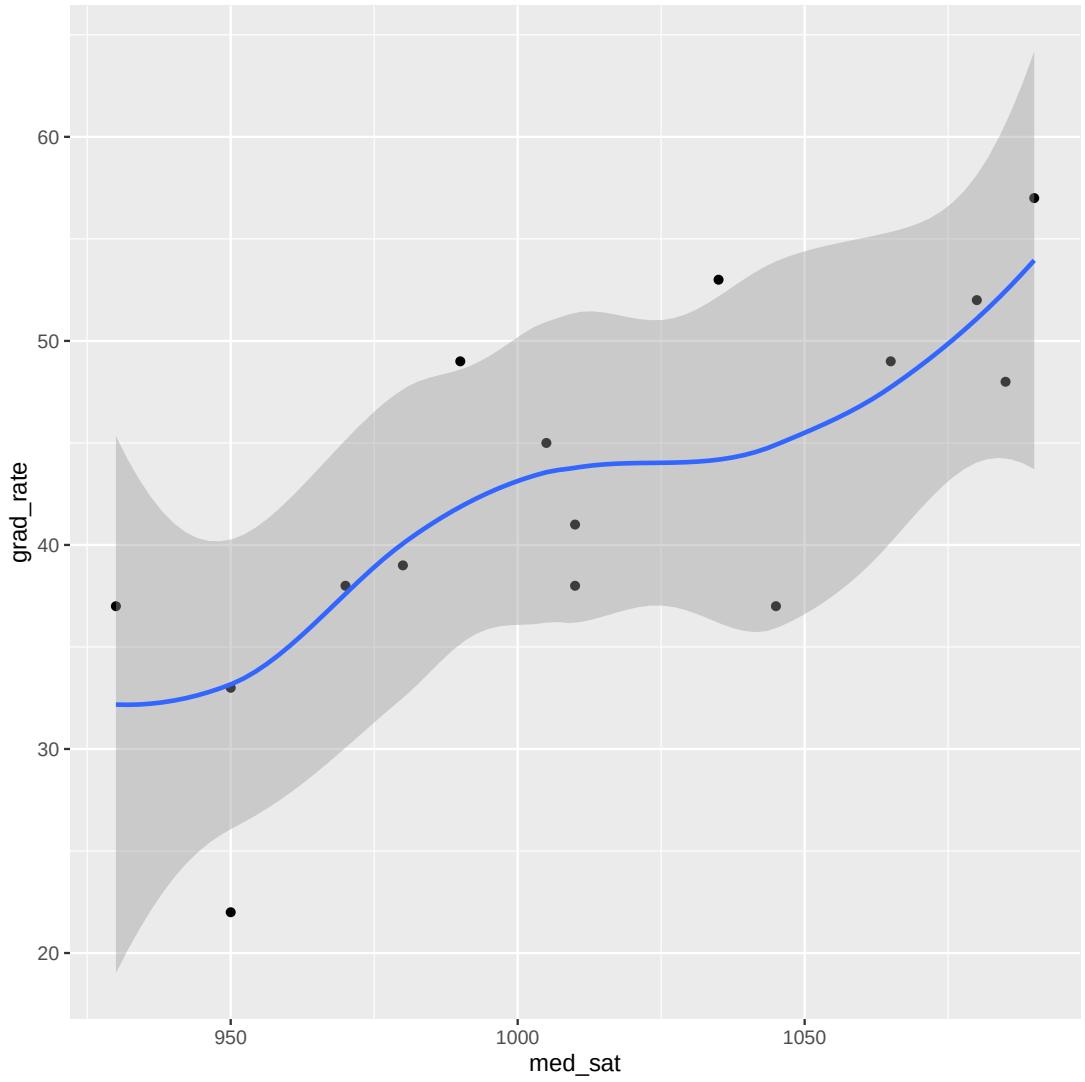
(b) (2 marks) Make a suitable plot of graduation rate and median SAT score.

Solution:

Two quantitative variables, so a scatterplot. Graduation rate is the response (this is figured out at the end of a student's university career), and median SAT score is explanatory (this is something a student comes into university with). Hence graduation rate should be on the y-axis. Add a smooth trend if you like (optional, but it makes the next part easier):

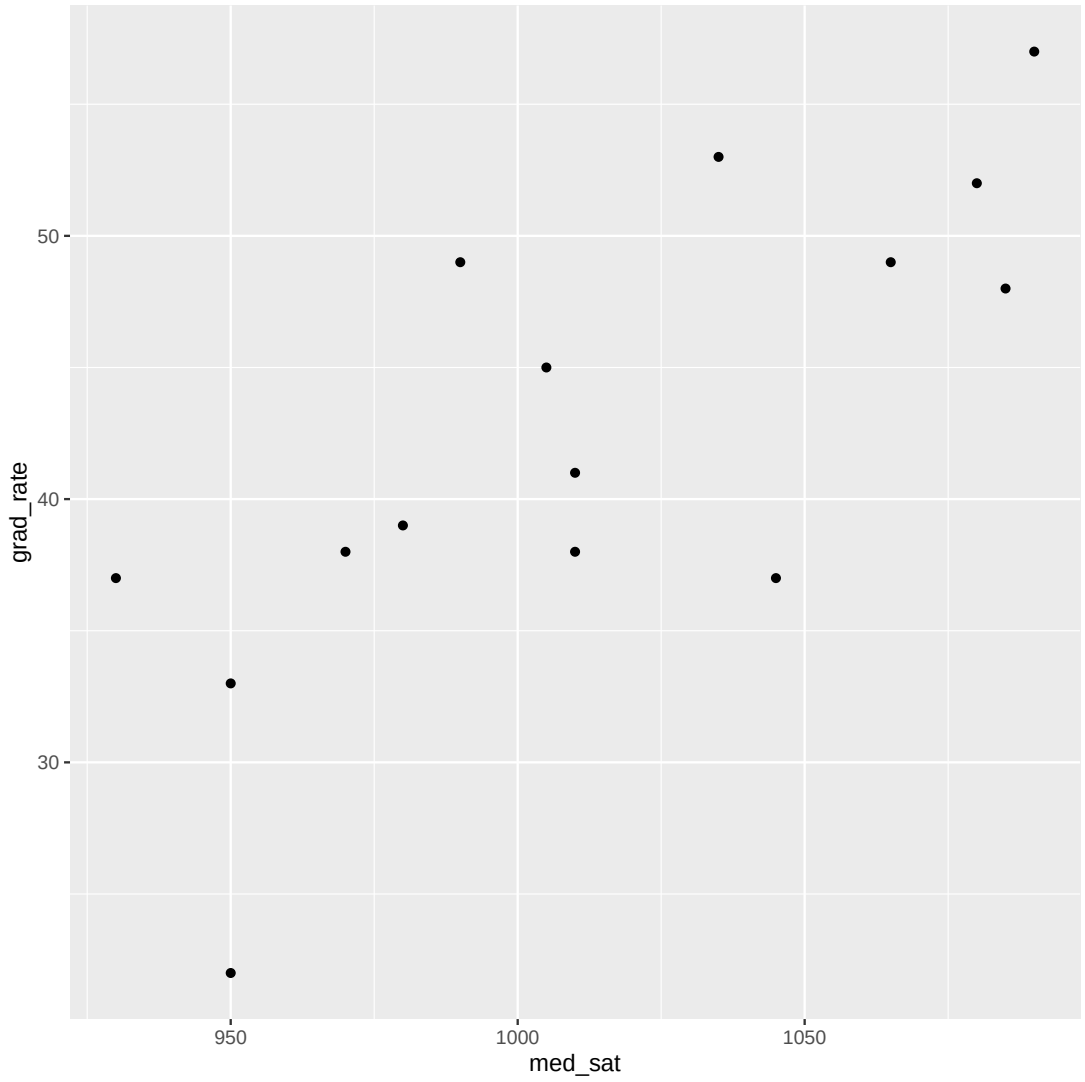
```
ggplot(gradrate, aes(x=med_sat, y=grad_rate)) + geom_point() + geom_smooth()
```

```
## `geom_smooth()` using method = 'loess' and formula 'y ~ x'
```



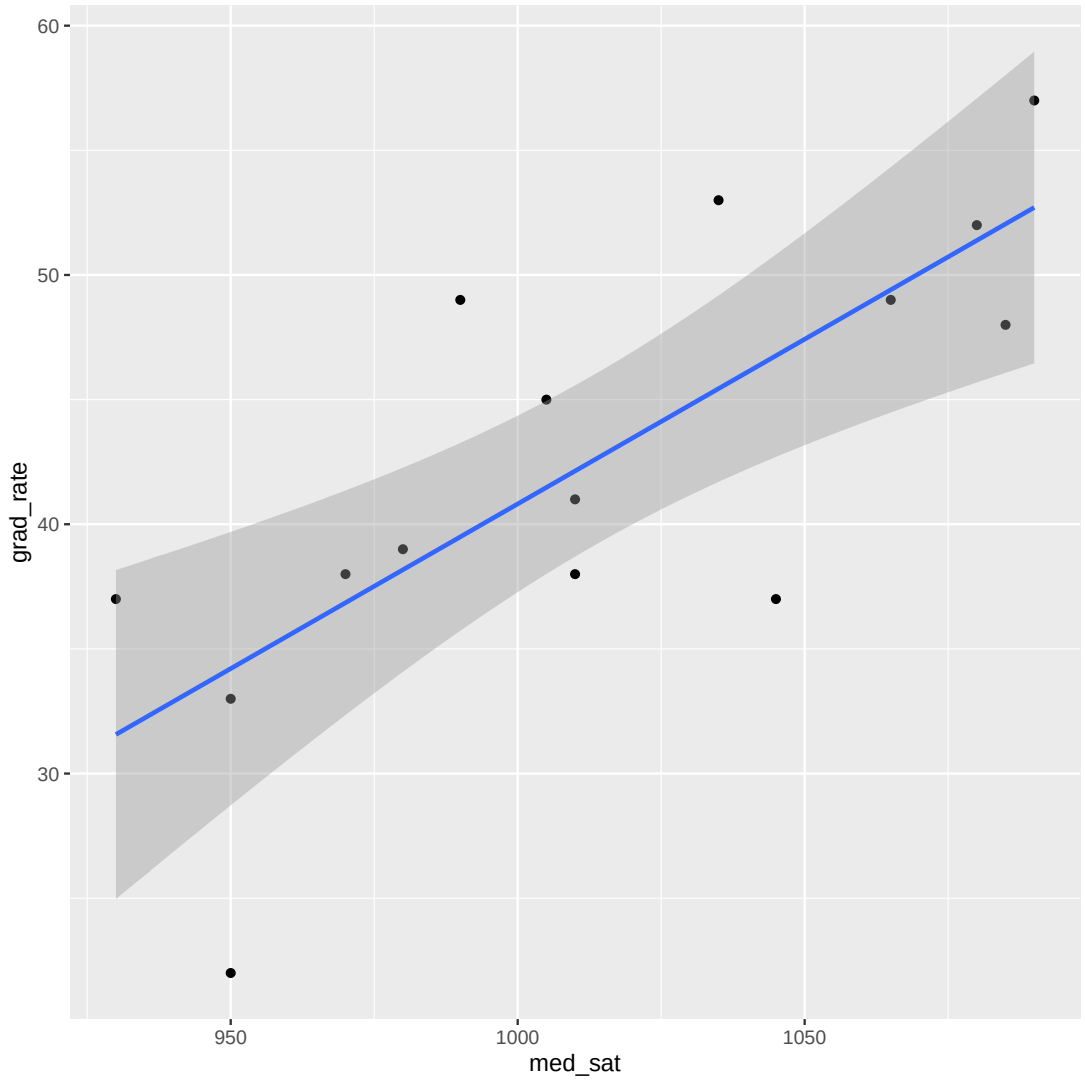
or without the smooth trend (also good):

```
ggplot(gradrate, aes(x=med_sat, y=grad_rate)) + geom_point()
```



I'm not so keen about putting a linear trend on, at this stage:

```
ggplot(gradrate, aes(x=med_sat, y=grad_rate)) + geom_point() +
    geom_smooth(method="lm")
```



I'm not so keen because we are *looking to see whether the trend is linear*, rather than *assessing a linear trend to see how well it fits*, at this stage anyway (we are still in exploratory mode). If you had, say, an upward-sloping curve (like, say, the windmill example in class), the points would be pretty close to a straight line, but a curve would still be better. If you put a linear fit on there, you might conclude that a straight line fits well, but that is not the point: you are trying to find the most appropriate form of relationship.

More discussion on this below.

(c) (2 marks) Comment briefly on what you learn from your plot. (Hint: linear or curved? Up or down? Strong or weak?)

Solution: The trend is more or less linear, upward, strong or moderately strong.

Don't be too picky about (especially) linearity; if it's "not obviously nonlinear" in a way that you could describe, that is fine. Something like down-up-down would be a problem, but something

that wiggles a bit like this one is not a problem at all. (This is like assessing normality in a t-test; as long as things are "not obviously non-normal", all is OK.)

If you put a linear fit on your graph with `geom_smooth`, to be convincing about the *relationship* being linear, you need to say that the points are randomly distributed about the line that you drew. (For example, if it was really a curve, you'd see several consecutive points below the line, several consecutive below, and so on, which doesn't seem to be happening here. Compare the windmill data from lecture, in particular the fit of the line to those data.) This is hard, and using a plain `geom_smooth` to put a smooth trend on the graph is much easier to work with. What you want is for your smooth to *tell* you that the relationship is a line, rather than putting a line on there and trying to decide whether it works.

Extra: this is about what you would expect from a practical point of view: students entering a university with higher SAT scores are (in theory) smarter, and thus would be better placed to handle whatever university life throws at them (and therefore more likely to graduate).

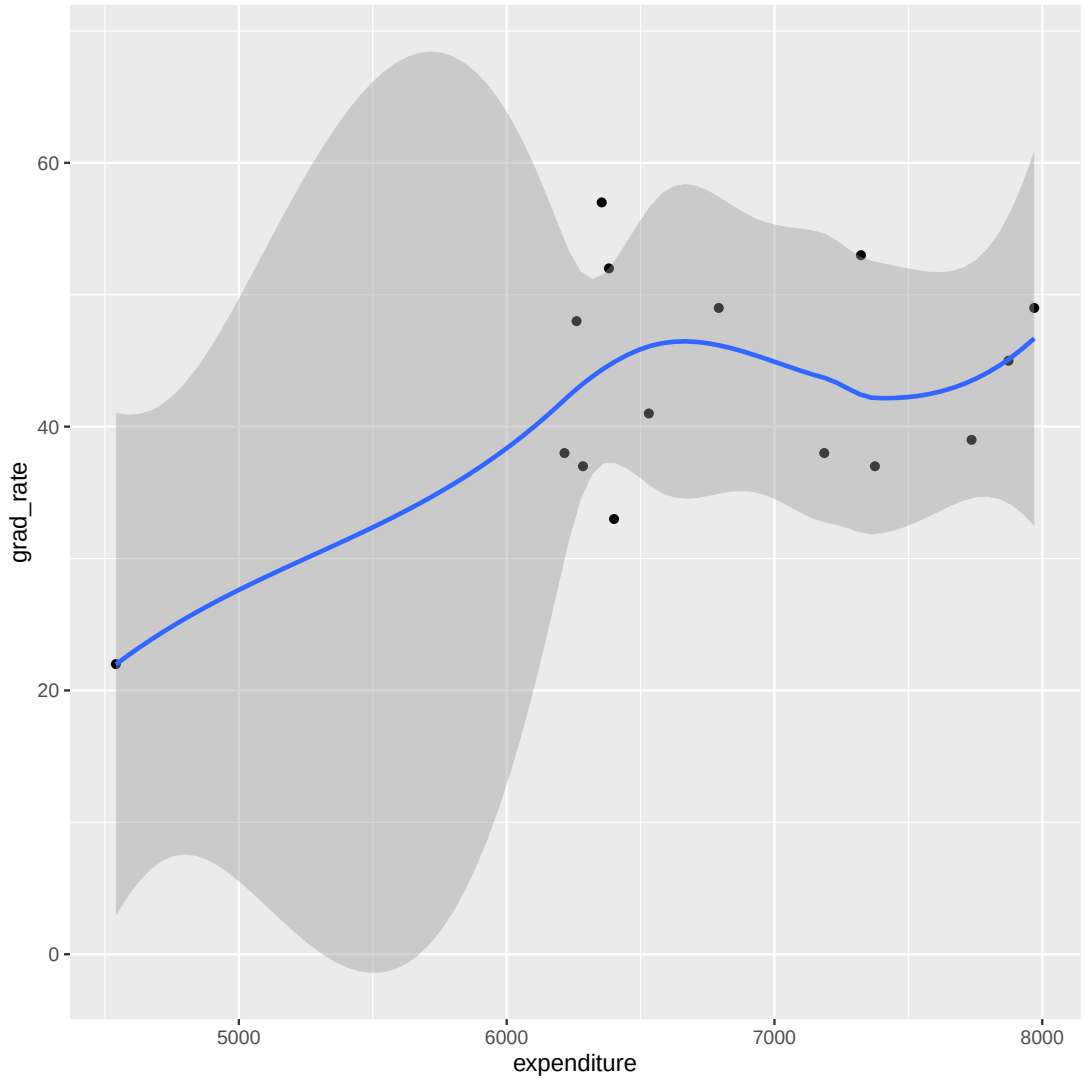
(d) (3 marks) Make a (similar) suitable plot for graduation rate against expenditure. Comment briefly on what you see.

Solution:

Again, a scatterplot. Copy, paste, and edit:

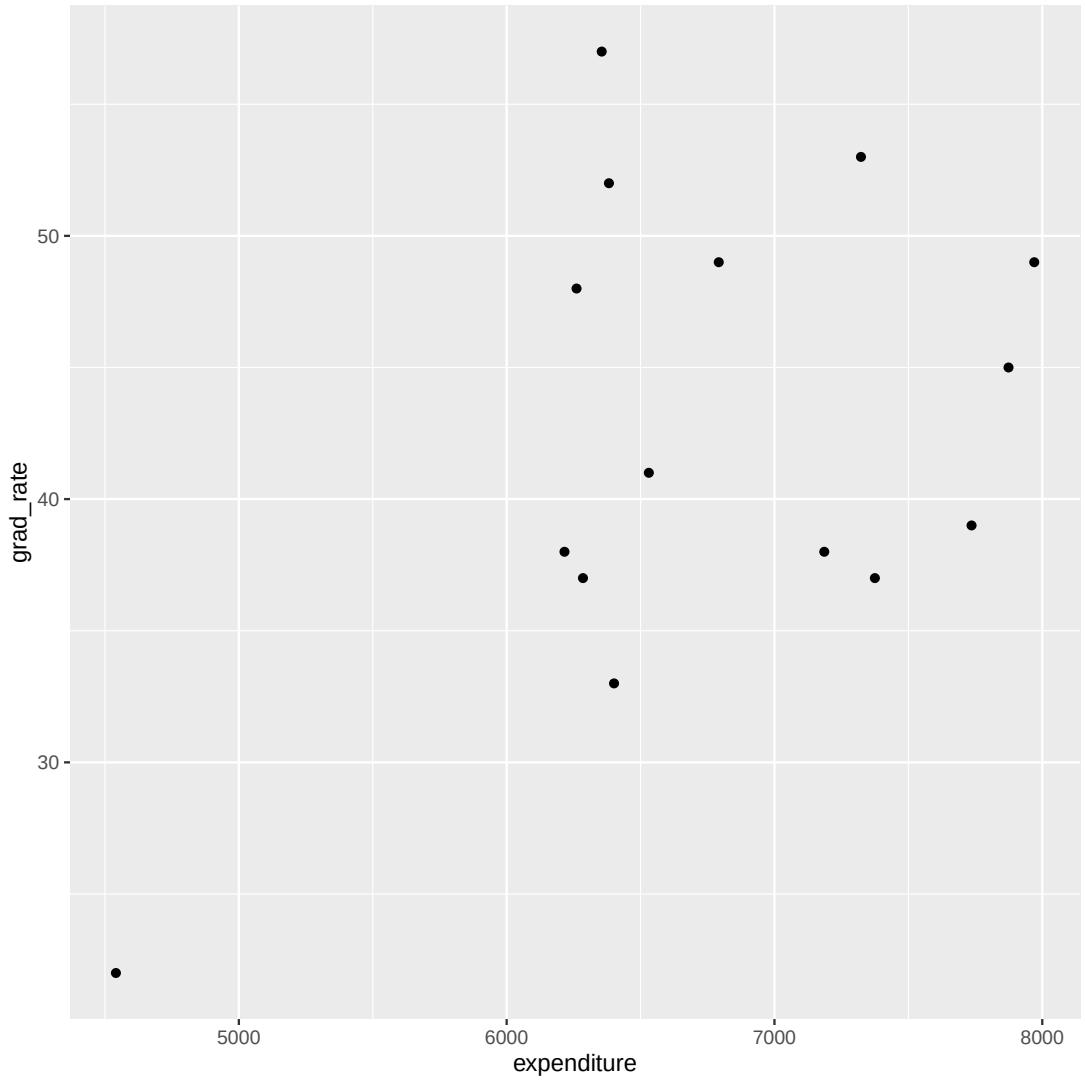
```
ggplot(gradrate, aes(x=expenditure, y=grad_rate)) + geom_point() + geom_smooth()
```

```
## `geom_smooth()` using method = 'loess' and formula 'y ~ x'
```



or without the smooth trend:

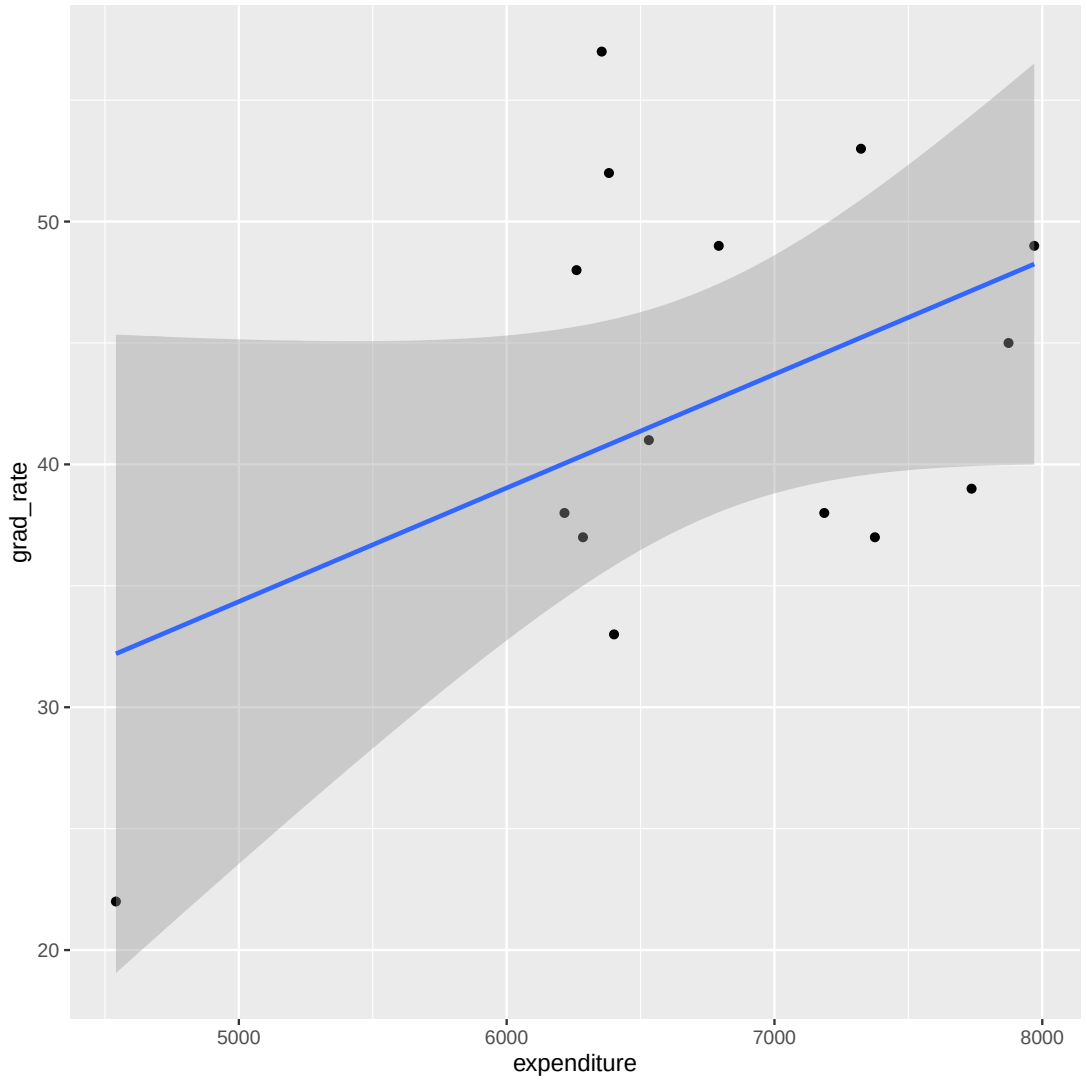
```
ggplot(gradrate, aes(x=expenditure, y=grad_rate)) + geom_point()
```



I think the most obvious thing here is the one unusually low expenditure (below $5,000): an outlier. This seems to me a better answer than "upward trend" (see below for discussion).

If you put a line on your graph in this one, what you need to observe (at least for yourself) is that the line goes through the outlier but not very convincingly through anything else. You might remember from your regression course that outliers tend to pull the line close to them, which is what is happening here:

```
ggplot(gradrate, aes(x=expenditure, y=grad_rate)) + geom_point() +
    geom_smooth(method="lm")
```



Regressions are based on means, so that in the same way that outliers distort the mean, outliers can also distort a regression, which is what is happening here, at least somewhat. (The regression line doesn't go closer to the outlier because of the non-trend in the rest of the points.) It looks again like an upward trend, but if you (mentally) omit the outlier, there appears to be very little trend of graduation rate with expenditure among the other points. That is to say, the apparent upward trend is almost entirely driven by the one outlying point. This is a very fragile thing to base a conclusion upon.

I don't know what kind of relationship I was expecting here.

(e) (2 marks) Fit a (multiple) regression predicting graduation rate from the other two variables and display the results.

Solution:

This is the standard thing. The hardest part is choosing a name for it. Pick your own:

```
gradrate.1=lm(grad_rate~med_sat+expenditure, data=gradrate)
summary(gradrate.1)
```

```
##
## Call:
## lm(formula = grad_rate ~ med_sat + expenditure, data = gradrate)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -7.6322 -2.1929 -0.8927  2.4291  9.2569
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) -1.187e+02  2.675e+01  -4.436 0.000812 ***
## med_sat      1.297e-01  2.480e-02   5.229 0.000211 ***
## expenditure  4.424e-03  1.480e-03   2.989 0.011305 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 4.843 on 12 degrees of freedom
## Multiple R-squared:  0.7569,Adjusted R-squared:  0.7163
## F-statistic: 18.68 on 2 and 12 DF,  p-value: 0.0002066
```

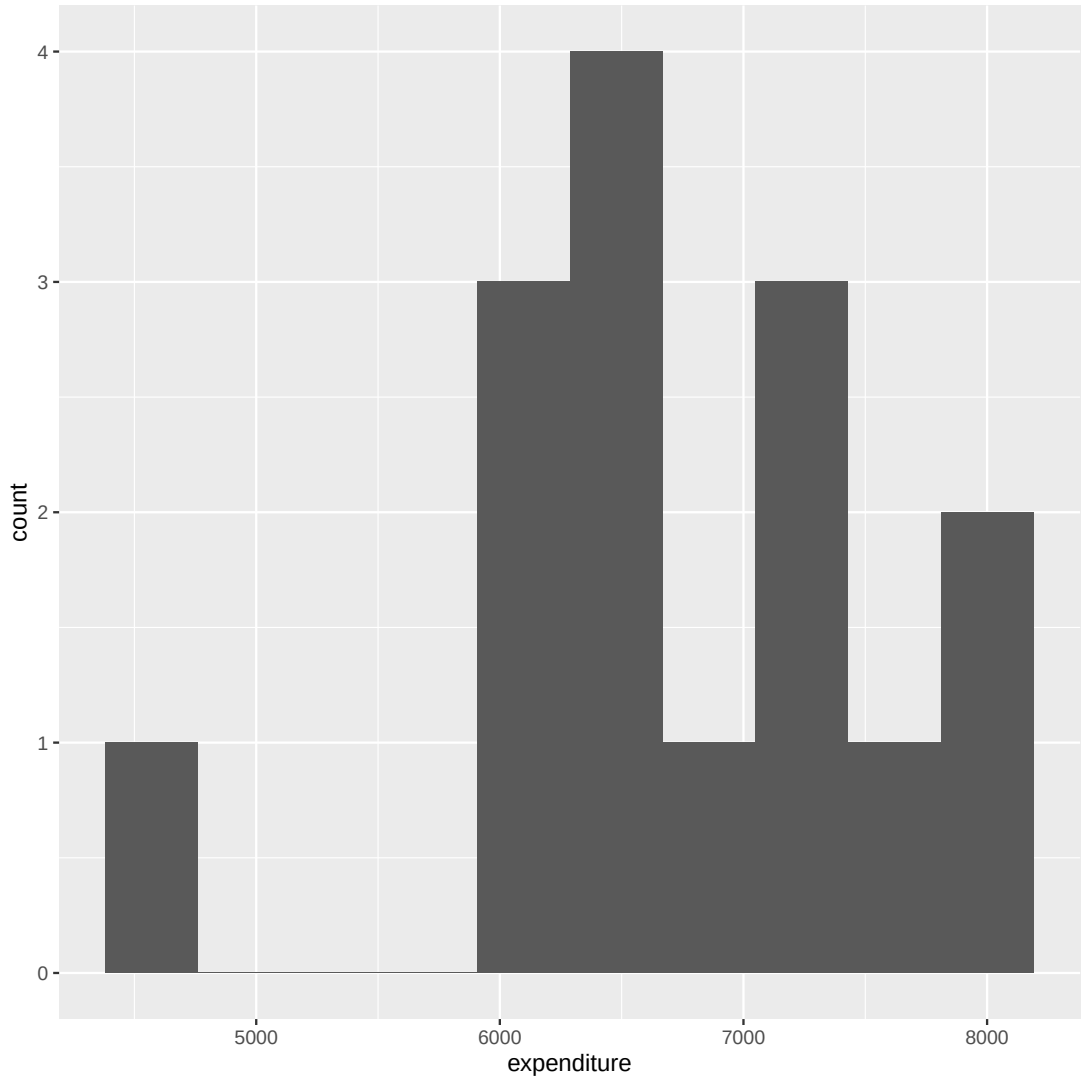
No comment needed here, but you might observe that both explanatory variables are significant and therefore you (appear to) need both of them to predict graduation rate. Observing this now will get you half the answer for a later part.

(f) (3 marks) One of the `expenditure` values was much lower than the others. It turns out that this value was an error. Create a new data frame that *excludes* this observation, and give the new data frame a name.

Solution: We are omitting a row for which something is true, which (correctly) suggests `filter`. But how to specify that lowest expenditure value. There is no need to be clever here; anything that gets the job done is fine.

I think the easiest way is to go back to the scatterplot of graduation rate and expenditure and note that the smallest expenditure is less than \$5,000 (or \$5,500 or something like that) and the others are all greater. If you didn't think to look back at a previous plot, you could also draw a one-variable plot of expenditures, like a histogram:

```
ggplot(gradrate, aes(x=expenditure)) + geom_histogram(bins=10)
```



You can get away with more bins than usual here, since you're not interested in the shape, just what that lowest value *is*.

The story here is the same: for the values you want to keep, they all have an expenditure greater than 5000 (5500, whatever).

So now we can run the `filter`:

```
gradrate %>% filter(expenditure > 5000) -> gradrate_x
summary(gradrate_x)
```

```
##     med_sat        expenditure     grad_rate
##  Min.   : 930.0   Min.   :6216   Min.   :33
##  1st Qu.: 982.5   1st Qu.:6362   1st Qu.:38
##  Median :1010.0   Median :6662   Median :43
##  Mean   :1017.5   Mean   :6906   Mean   :44
##  3rd Qu.:1060.0   3rd Qu.:7362   3rd Qu.:49
##  Max.   :1090.0   Max.   :7970   Max.   :57
```

The smallest expenditure is now over 6,000; also, if you check, there are now 14 rows instead of 15:

```
nrow(gradrate_x)
```

```
## [1] 14
```

Feel free to use any value in the `filter` that will omit the lowest one, for example `expenditure` strictly greater than the minimum expenditure. If it works, it's good. Get it done.

I chose the name `gradrate_x` to suggest that something had been excluded; also, I avoided using a number because I am using them for models rather than data frames. If you use numbers for everything, you will have to keep straight in your head whether it's a data frame or a regression model, and thus what you will need to do with each one.

(g) (3 marks) Re-run your model of (e), but on your new data set. What would you say is the most important difference between the output of the two models? Explain briefly.

Solution:

Change the name of the data frame to the one with the errant expenditure removed, and (probably) change the number of the model also:

```
gradrate.2=lm(grad_rate~med_sat+expenditure, data=gradrate_x)
summary(gradrate.2)
```

```
##
## Call:
## lm(formula = grad_rate ~ med_sat + expenditure, data = gradrate_x)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -8.378 -2.761 -1.070  2.111  8.634
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) -98.667141  35.868589  -2.751  0.01886 *
## med_sat       0.119397   0.027847   4.288  0.00128 **
## expenditure   0.003067   0.002189   1.401  0.18868
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 4.9 on 11 degrees of freedom
## Multiple R-squared:  0.6259,Adjusted R-squared:  0.5579
## F-statistic: 9.201 on 2 and 11 DF,  p-value: 0.004483
```

I think the major change is that `expenditure` is no longer significant. I guess you can also say that R-squared has gone down by a lot, given that all we did was to take out one observation.

This says to me that the significance of `expenditure` was an illusion. If you go back to the plot of graduation rate vs. expenditure, you'll see that basically *all* the evidence of an upward trend was coming from that observation with very small expenditure. Take that out, as we did, and the evidence for the trend goes away. (In less statistical terms, there was one college with a very small expenditure and also a very small graduation rate, which was making it look as if

increasing expenditure also increased graduation rate, which is not true for the other colleges.)

Extra: there is more to the relationship with expenditure, however, as we are about to explore.

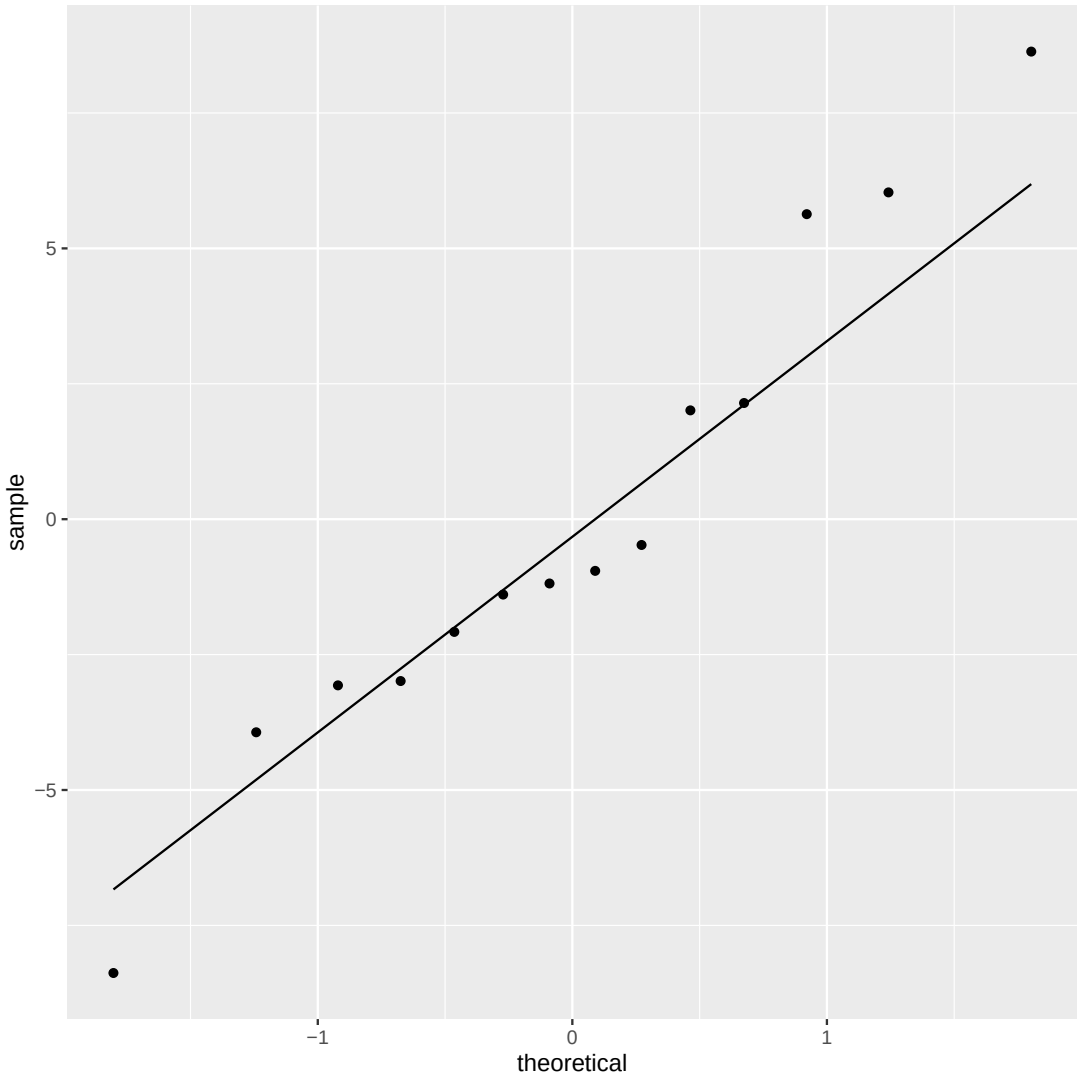
(h) (3 marks) Produce the one of the standard residual plots that most clearly indicates a problem with the most recent regression, and describe the problem it indicates. (Hint: look at plots of residuals: normal quantile plot, against fitted values, against each of the explanatory variables including non-significant ones. Also note that you may need to do some extra work to obtain a data frame with the data and the stuff from the regression in it.)

Solution:

The hint basically tells you what to do. Work with the regression that I called `gradrate.2`, produce the plots one by one (for yourself), and hand in the one that looks most problematic (or, maybe, stop when you get one that indicates a serious problem, but you really ought to look at them all and grab the one that looks like the biggest problem).

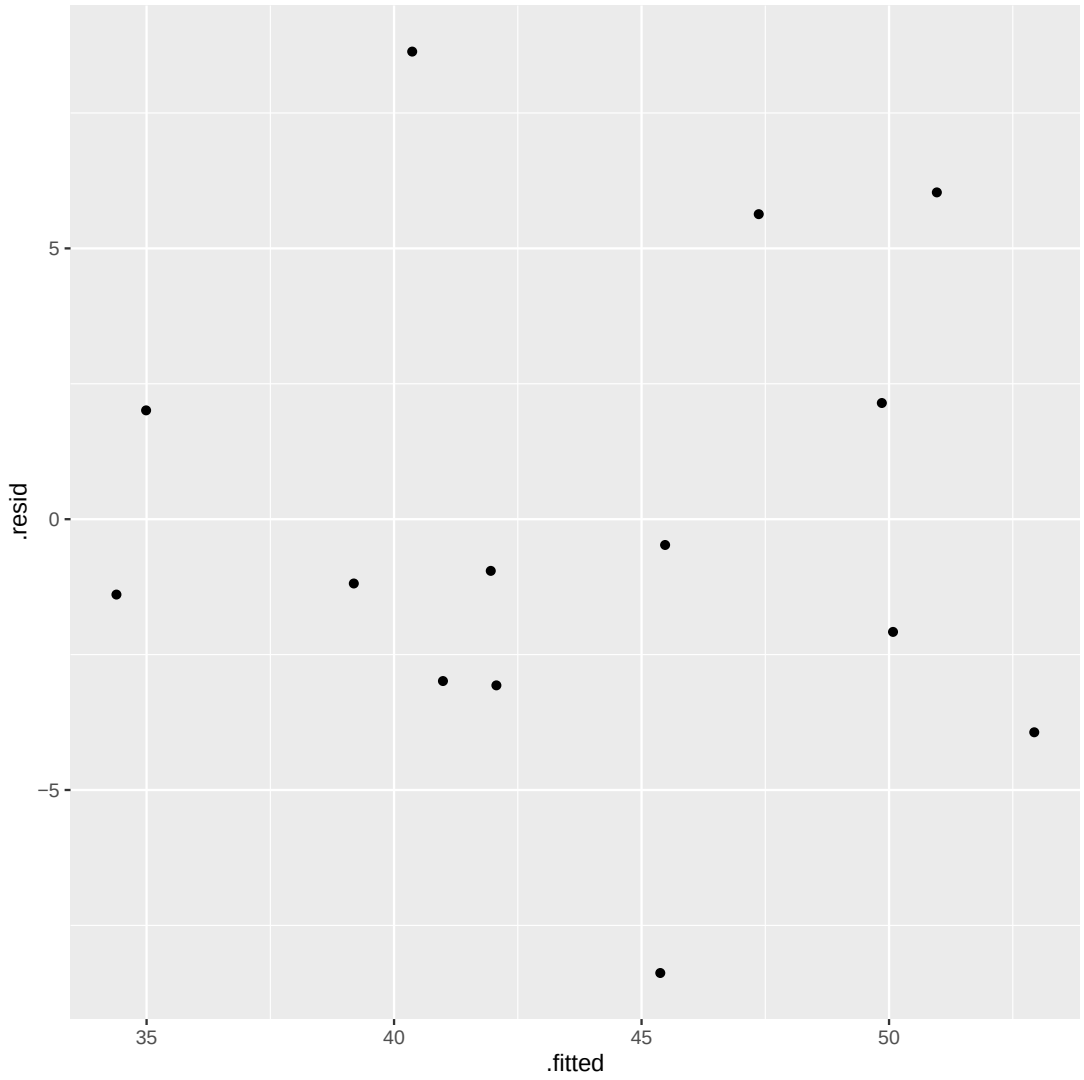
Here we go, then:

```
ggplot(gradrate.2, aes(sample=.resid)) + stat_qq() + stat_qq_line()
```



Slightly long-tailed, but not at all bad. I think we can find something more problematic than that. Residuals against fitted:

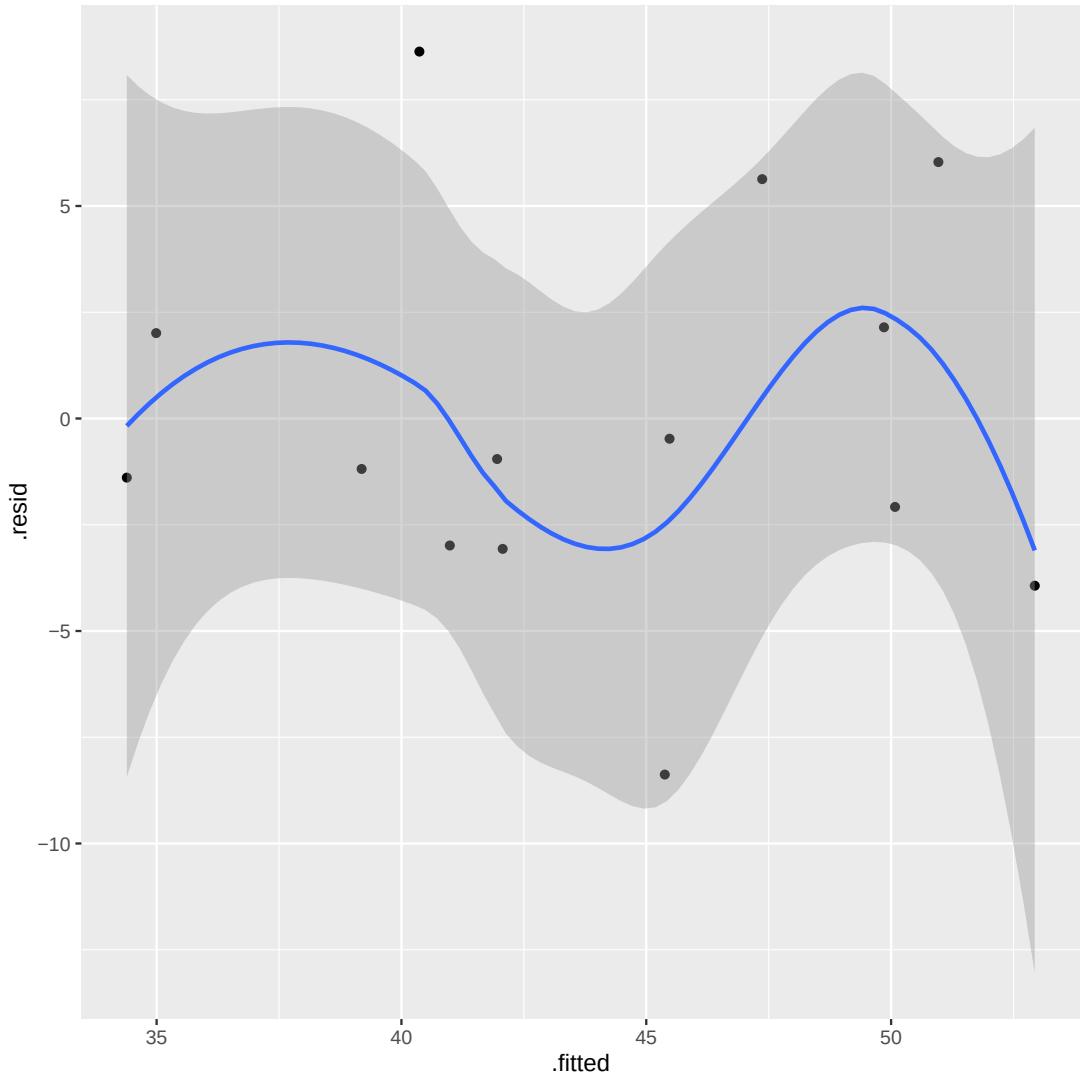
```
ggplot(gradrate.2, aes(x=.fitted, y=.resid)) + geom_point()
```



That looks random, so no problem there. If you like, add a smooth trend to it, but be wary of taking the smooth trend too seriously:

```
ggplot(gradrate.2, aes(x=.fitted, y=.resid)) + geom_point() + geom_smooth()
```

```
## `geom_smooth()` using method = 'loess' and formula 'y ~ x'
```



The smooth trend itself wiggles, but my guideline for these on a residual plot is to ask whether the grey envelope clearly includes zero all the way across. It does, so there is no "significant" deviation of the residuals from zero anywhere.[1] Thus, I call this good.

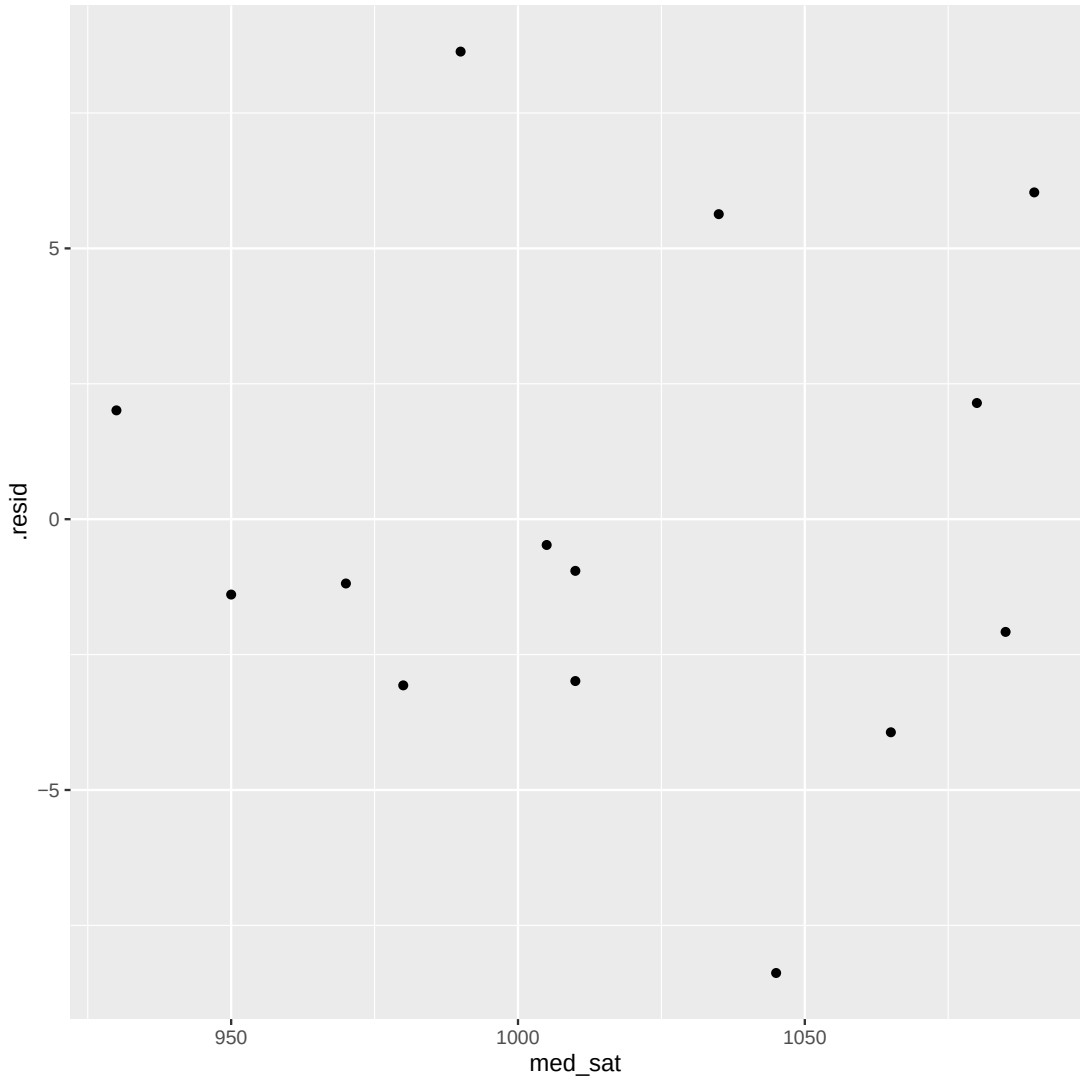
The last two plots to try are the residuals against the two explanatory variables, median SAT score and expenditure. Unfortunately, though, we don't have a data frame with both of these in. The easiest way to proceed is to make one using `augment` from `broom`:

```
library(broom)
gradrate.2 %>% augment(gradrate_x) -> all_stuff
```

The logic of `augment` is that you start with the regression model and then augment it with the data (rather than the other way around).

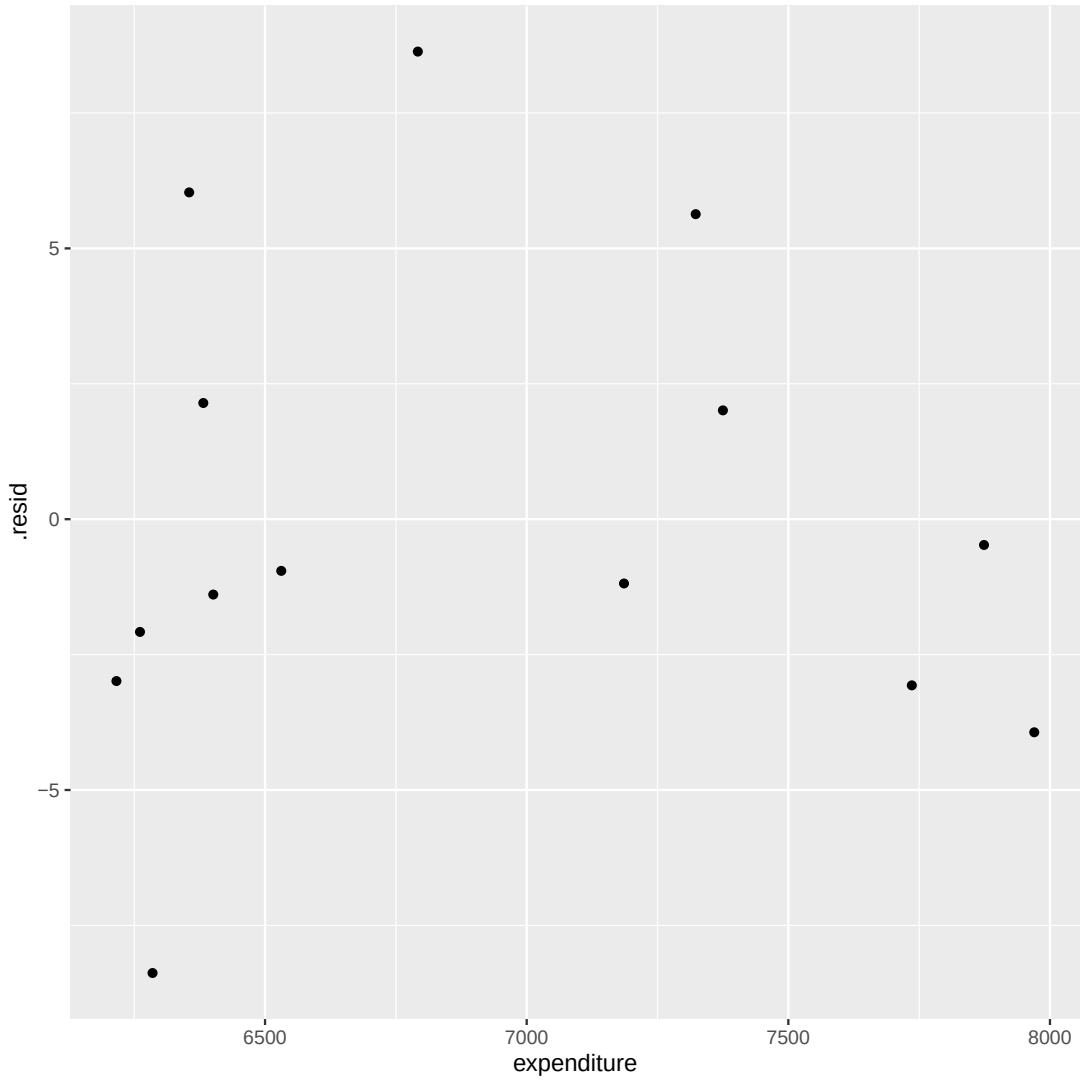
Then you can use what I called `all_stuff` for the plots:

```
ggplot(all_stuff, aes(x=med_sat, y=.resid)) + geom_point()
```



This is also pretty random. So now we're down to the last of our plots:

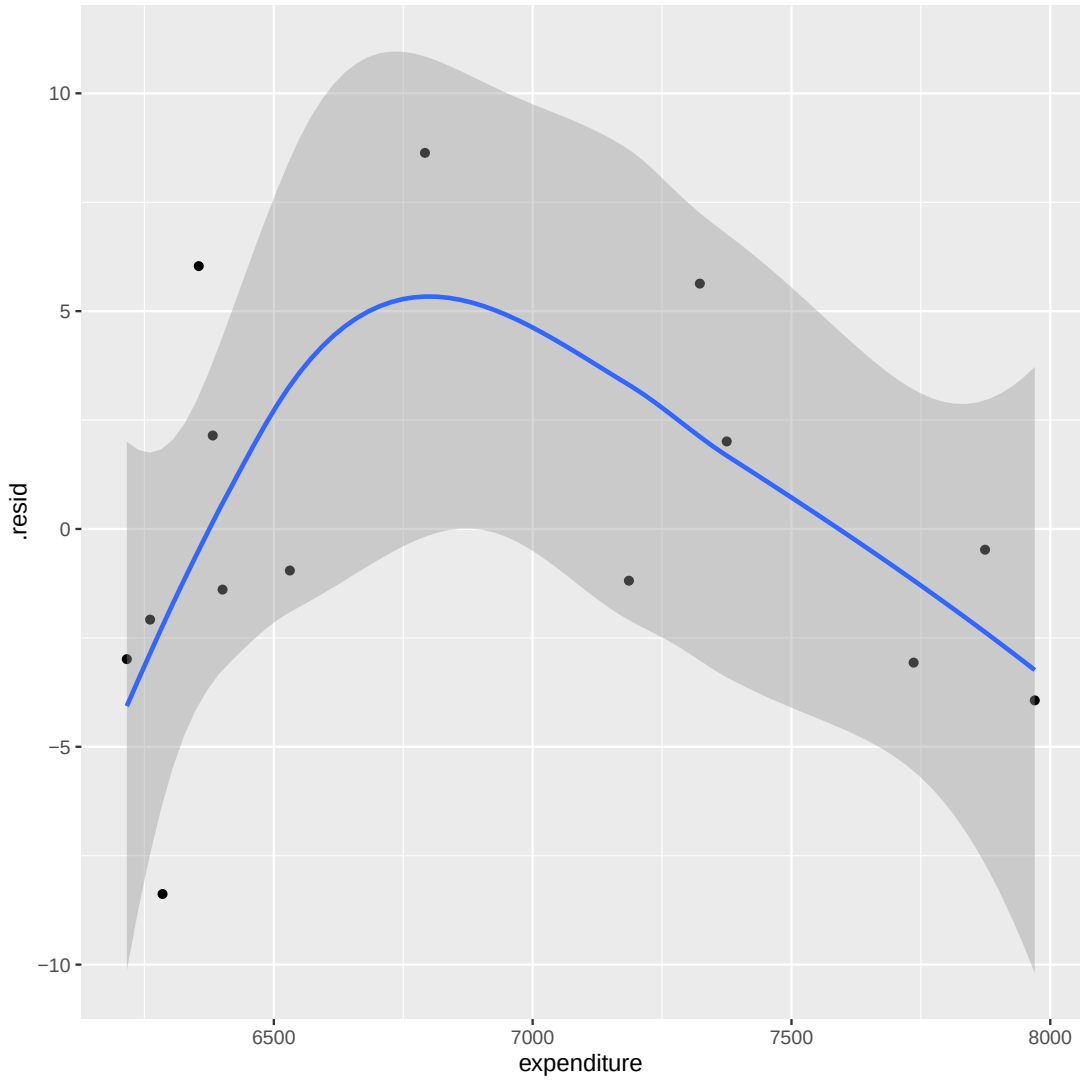
```
ggplot(all_stuff, aes(x=expenditure, y=.resid)) + geom_point()
```



I think here we finally have a problem. The residuals kind of go up and down again in a curved shape. This is perhaps a bit clearer with a smooth trend added:

```
ggplot(all_stuff, aes(x=expenditure, y=.resid)) + geom_point() + geom_smooth()
```

```
## `geom_smooth()` using method = 'loess' and formula 'y ~ x'
```

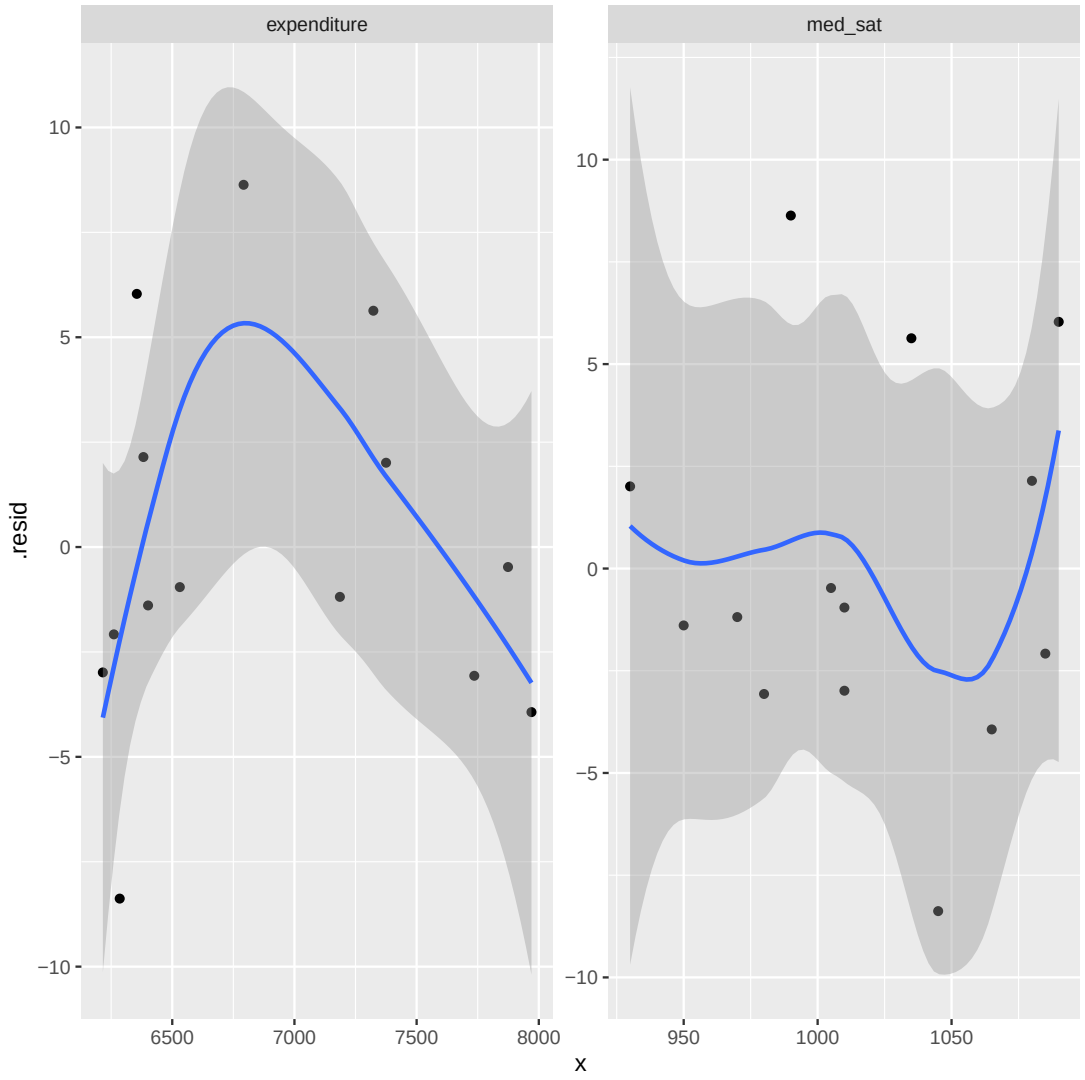


So the last plot, with or without the smooth trend, is what you need to hand in. I don't need to see any of the others. You need to say that there is a curved pattern, indicating a curved relationship (with expenditure).

If you want to do some work to save some work, you can use the facets idea from lecture to get these last two plots both at once:

```
all_stuff %>%
    pivot_longer(med_sat:expenditure, names_to="xname", values_to="x") %>%
    ggplot(aes(x=x, y=.resid)) + geom_point() + geom_smooth() +
    facet_wrap(~xname, scales="free")
```

```
## `geom_smooth()` using method = 'loess' and formula 'y ~ x'
```

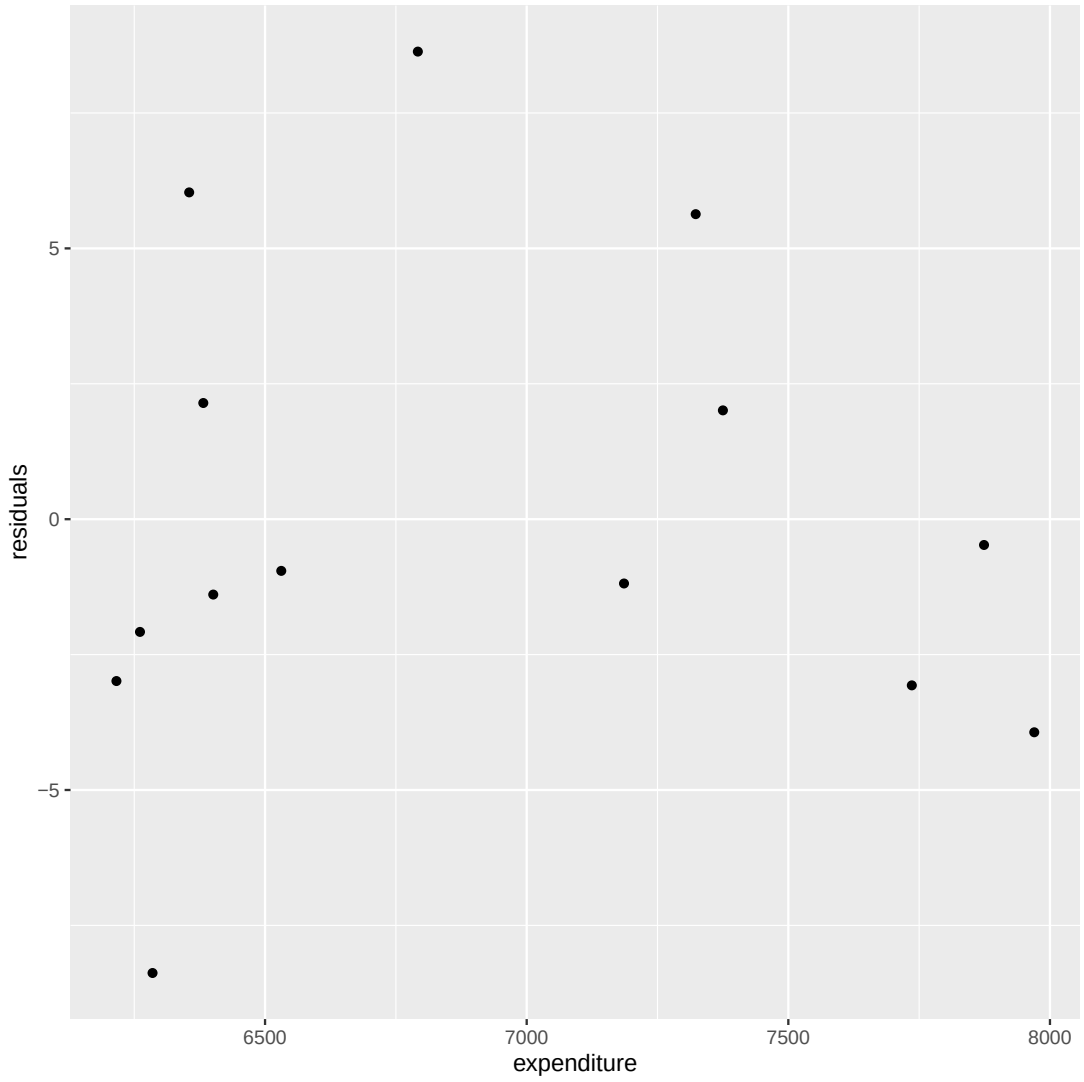


The smooth trends are again optional. The one on the right is definitely not to be taken seriously (see how wide the grey envelope is), but the one on the left is definitely low-high-low, something that the points follow (and thus deserves to be taken seriously). If you go this way on yours, be sure to indicate which one of your two graphs is the problematic one.

The other thing to note here is that the two x-variables are different numbers (completely), so you *need* the `scales="free"` on the `facet_wrap`. (See what happens if you don't put it in.)

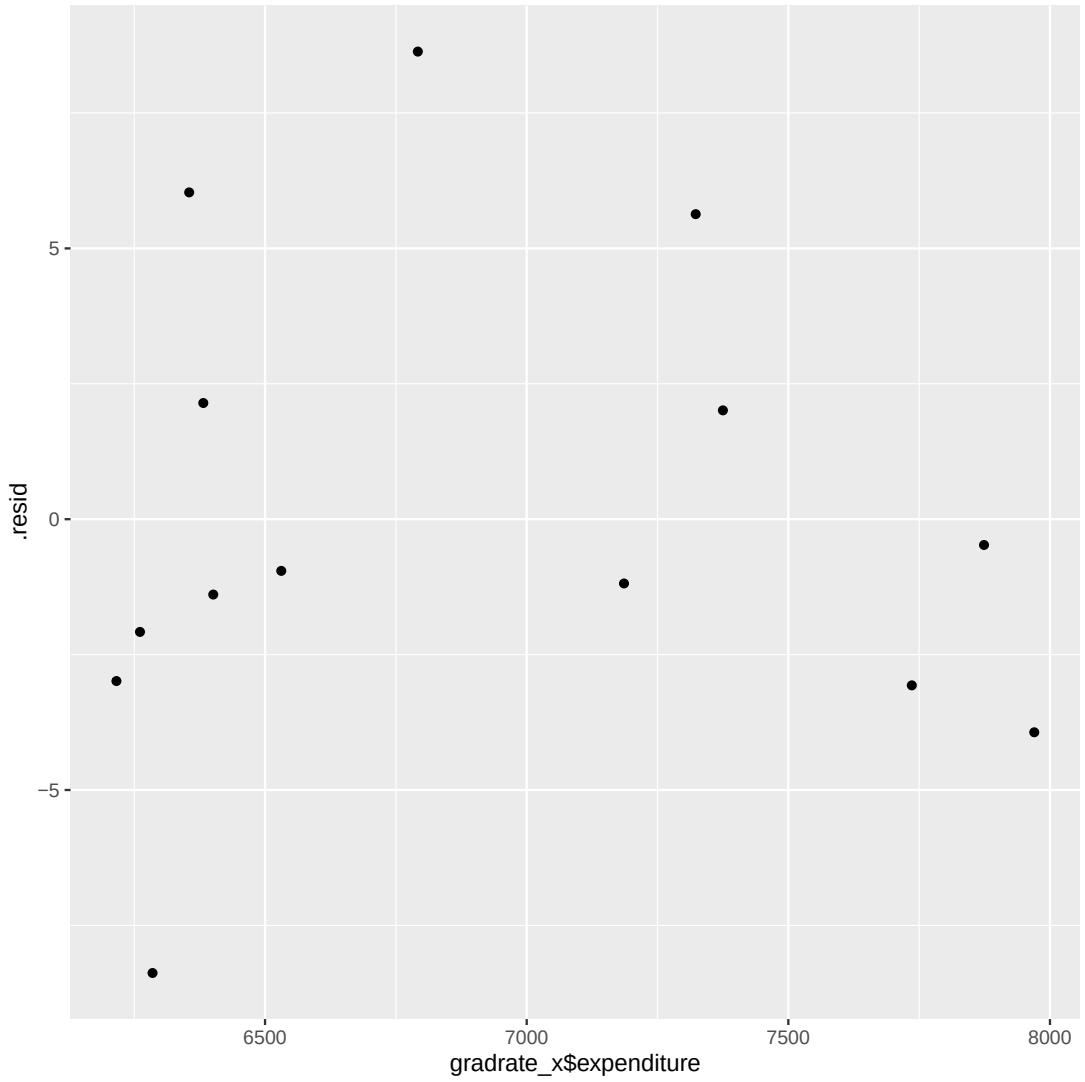
If you don't think of `augment`, you have a couple of other options. One is to make a new data frame taking things from different places:

```
d=tibble(expenditure=gradrate_x$expenditure, residuals=resid(gradrate.2))
ggplot(d, aes(x=expenditure, y=residuals)) + geom_point()
```



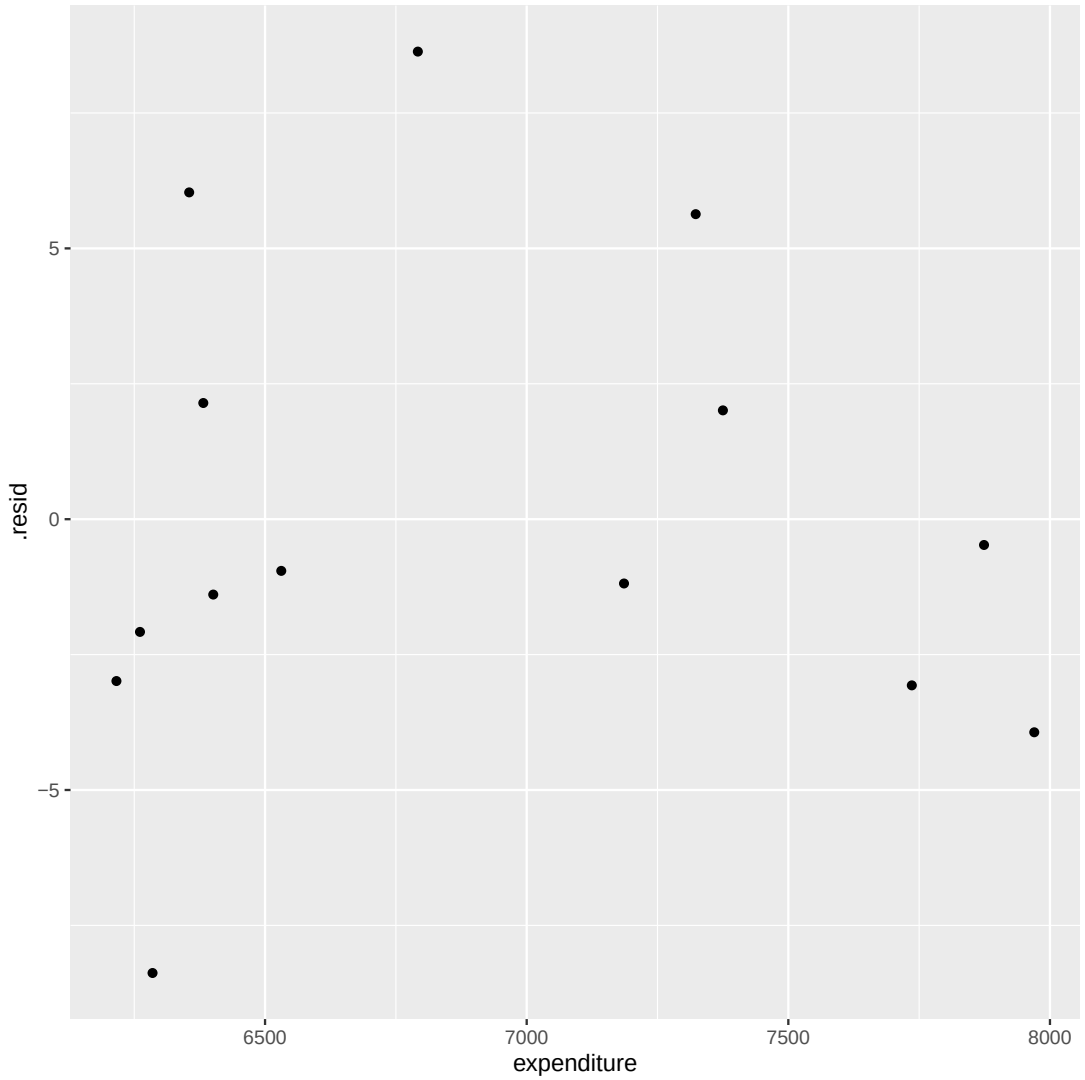
and another is to use the dollar sign *within* `ggplot=` to grab the right things. In this case, it's easier to use the regression object as your "base":

```
ggplot(gradrate.2, aes(x=gradrate_x$expenditure, y=.resid)) + geom_point()
```



This gets you a strange-looking label on the x-axis, but for this problem I'm not worried about that. If it bothers *you*:

```
ggplot(gradrate.2, aes(x=gradrate_x$expenditure, y=.resid)) + geom_point() +
  xlab("expenditure")
```



Extra 1: I was worried at first that the curved trend on this plot was driven mainly by that point by itself with the most positive residual, but on reflection there is enough of an upward trend in the points to the left of it and enough of a downward trend in the points to its right to justify calling this a curve. Compare this residual plot to the one in the final part.

Extra 2: what the regression `gradrate.2` and that last residual plot are indicating is that there is no straight-line relationship with `expenditure`, but there might be a curved one instead. The prospect of a curved relationship is worth looking into, which we do next.

(i) (4 marks) How might you modify your previous regression model to take care of the problem you found in the previous part? Make that modification, re-fit the model, and describe the principal change that you see in the results.

Solution:

The problem is with one of the explanatory variables rather than the response, so the obvious

thing to try is a squared term in `expenditure`. The problem is with expenditure rather than with graduation rate, so fixing up graduation rate will not help as much.

If we are going to add expenditure-squared, we also have to keep `expenditure` itself. The term in `expenditure` controls the left-right position on the page, and if you leave it out, the maximum (here) or possibly minimum (in general) has to be at zero, which it is clearly not for these data. (Note that the linear term in `expenditure` below is also significantly different from zero.)

```
gradrate.3=lm(grad_rate~med_sat+expenditure+I(expenditure^2), data=gradrate_x)
summary(gradrate.3)
```

```
##
## Call:
## lm(formula = grad_rate ~ med_sat + expenditure + I(expenditure^2),
##     data = gradrate_x)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -6.6013 -1.4709  0.3242  1.7163  5.0917
##
## Coefficients:
##                     Estimate Std. Error t value Pr(>|t|)
## (Intercept)      -8.109e+02  2.410e+02  -3.365  0.00718 **
## med_sat           1.579e-01  2.490e-02   6.341 8.44e-05 ***
## expenditure       1.948e-01  6.447e-02   3.022  0.01286 *
## I(expenditure^2) -1.354e-05  4.551e-06  -2.975  0.01392 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3.743 on 10 degrees of freedom
## Multiple R-squared:  0.8015,Adjusted R-squared:  0.742
## F-statistic: 13.46 on 3 and 10 DF,  p-value: 0.0007613
```

The principal change from the last model is that the expenditure-squared term is significant (so there *is* a relationship with expenditure after all, just not a linear one). Or, if you like, note that R-squared has gone way up, so that this model fits much better.

Extra: mathematically, a quadratic is of the form $y = ax^2 + bx + c$. A curve of this type bends exactly once, so it can do one of two things: open upwards, with a minimum value, or open downwards, with a maximum. The way you tell the difference is to look at whether a is plus or minus: if it's plus, it opens upwards, and if minus, it opens downwards.[2] Here, a is the cofficient of expenditure-squared, -0.0000135, which is negative. So the relationship between graduation rate and expenditure opens downwards (has a maximum). That means that the highest graduation rate could go with a *middling* expenditure, not necessarily the highest or lowest one.

Another way of looking at this is to do some predictions. (I didn't do this in lecture, so won't ask you to do it on the assignment. It comes up properly in D29.) To do that, we need some values of median SAT and expenditure to predict from:

```
med_sats=c(900, 950, 1000, 1050, 1100)
expenditures=c(6000, 6500, 7000, 7500, 8000)
```

I chose some values that spanned the data.

Now we need all possible combinations of these:

```
new=crossing(med_sat=med_sats, expenditure=expenditures)
new
```

```
## # A tibble: 25 x 2
##    med_sat expenditure
##      <dbl>       <dbl>
##  1     900        6000
##  2     900        6500
##  3     900        7000
##  4     900        7500
##  5     900        8000
##  6     950        6000
##  7     950        6500
##  8     950        7000
##  9     950        7500
## 10     950        8000
## # ... with 15 more rows
```

and now we can predict these:

```
pred=predict(gradrate.3, new)
new %>% bind_cols(pred_grad_rate=pred) -> d
d
```

```
## # A tibble: 25 x 3
##    med_sat expenditure pred_grad_rate
##      <dbl>       <dbl>          <dbl>
##  1     900        6000           12.6
##  2     900        6500           25.4
##  3     900        7000           31.4
##  4     900        7500           30.6
##  5     900        8000           23.1
##  6     950        6000           20.5
##  7     950        6500           33.3
##  8     950        7000           39.3
##  9     950        7500           38.5
## 10     950        8000           31.0
## # ... with 15 more rows
```

One of the reasons for doing these was to understand how our model behaved. In particular, which combination of the two explanatory variables gives the highest predicted graduation rate?

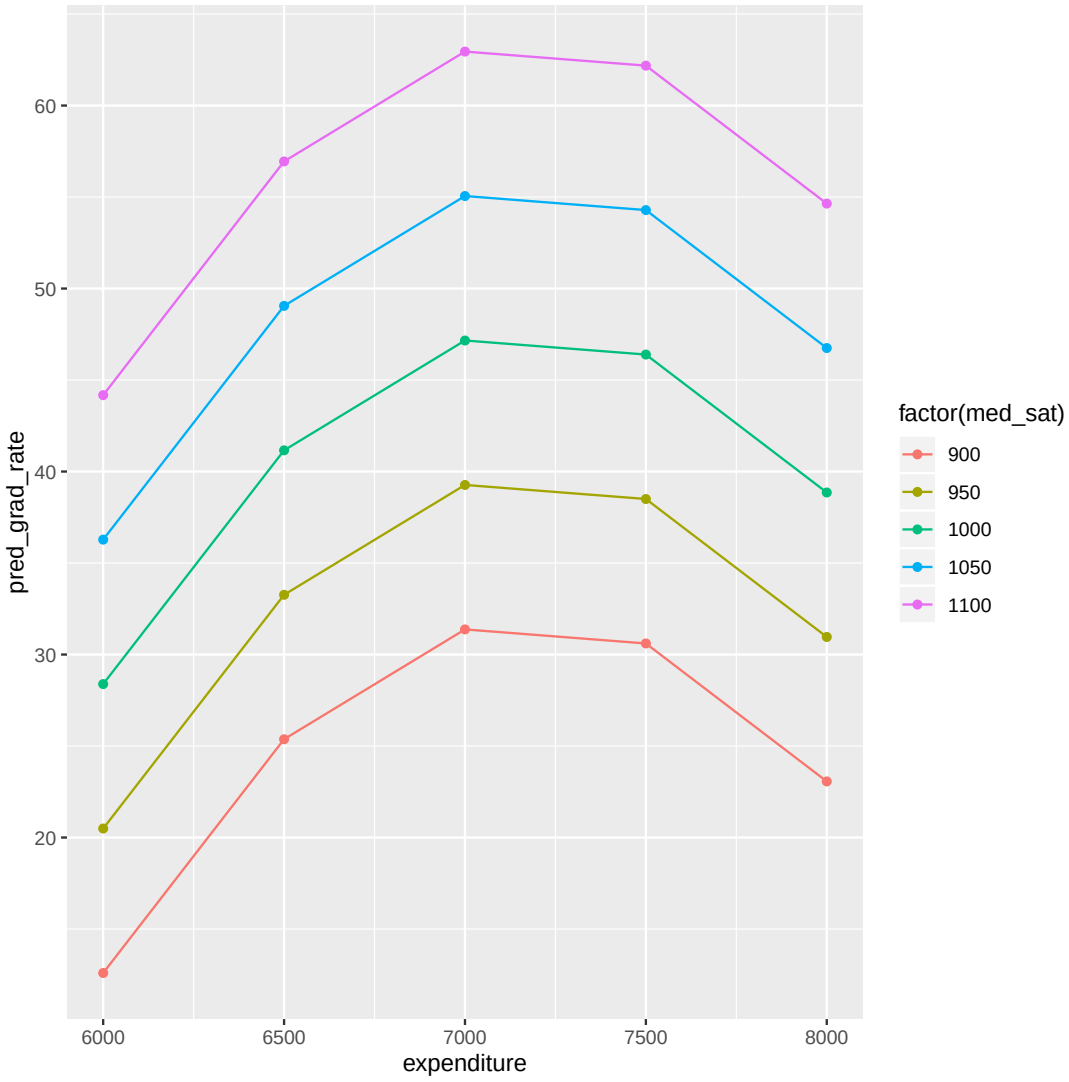
```
d %>% arrange(desc(pred_grad_rate))
```

```
## # A tibble: 25 x 3
##    med_sat expenditure pred_grad_rate
##      <dbl>       <dbl>          <dbl>
##  1    1100        7000           63.0
##  2    1100        7500           62.2
##  3    1100        6500           56.9
##  4    1050        7000           55.1
##  5    1100        8000           54.6
##  6    1050        7500           54.3
##  7    1050        6500           49.1
##  8    1000        7000           47.2
##  9    1050        8000           46.7
## 10    1000        7500           46.4
## # ... with 15 more rows
```

The highest predicted graduation rate goes with the highest median SAT score (as you would expect, since the relationship was linear with a positive slope), but *neither the highest nor lowest expenditure*, since that relationship was a curve with the maximum in between.

I wanted to show you a graph, but there are three quantitative variables, and we don't know how to deal with that. So let's pretend median SAT score is categorical, and use it for colour. I have to make it into a fake-categorical variable to use it like this:

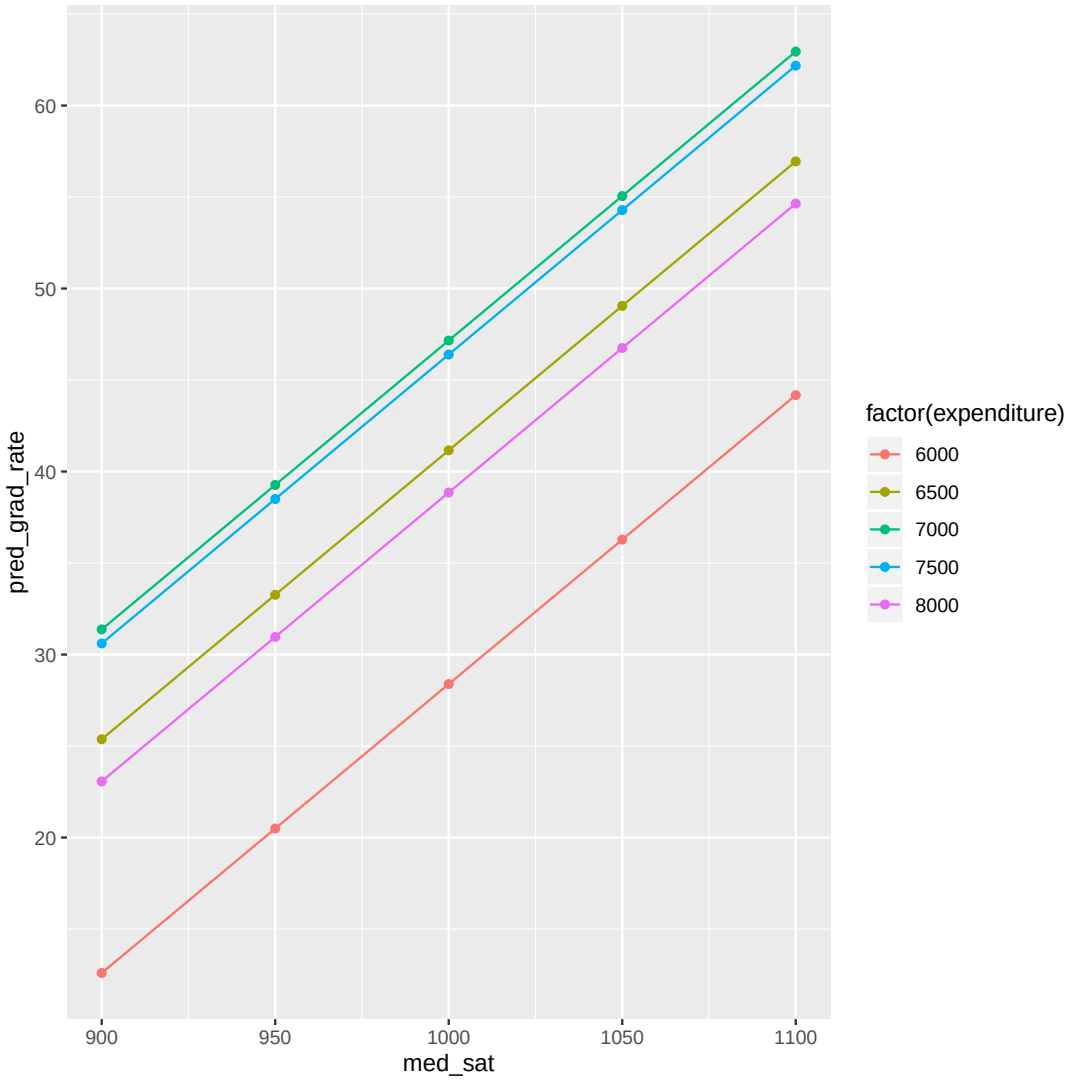
```
ggplot(d, aes(x=expenditure, y=pred_grad_rate, colour=factor(med_sat))) +
        geom_point() + geom_line()
```



The curves get higher up the page as median SAT increases, but each relationship with expenditure is definitely curved. So colleges with the highest graduation rate have the highest median SAT score, but clearly not the highest student-related expenditure.

I could have chosen the other explanatory variable as the colour:

```
ggplot(d, aes(x=med_sat, y=pred_grad_rate, colour=factor(expenditure))) +
        geom_point() + geom_line()
```



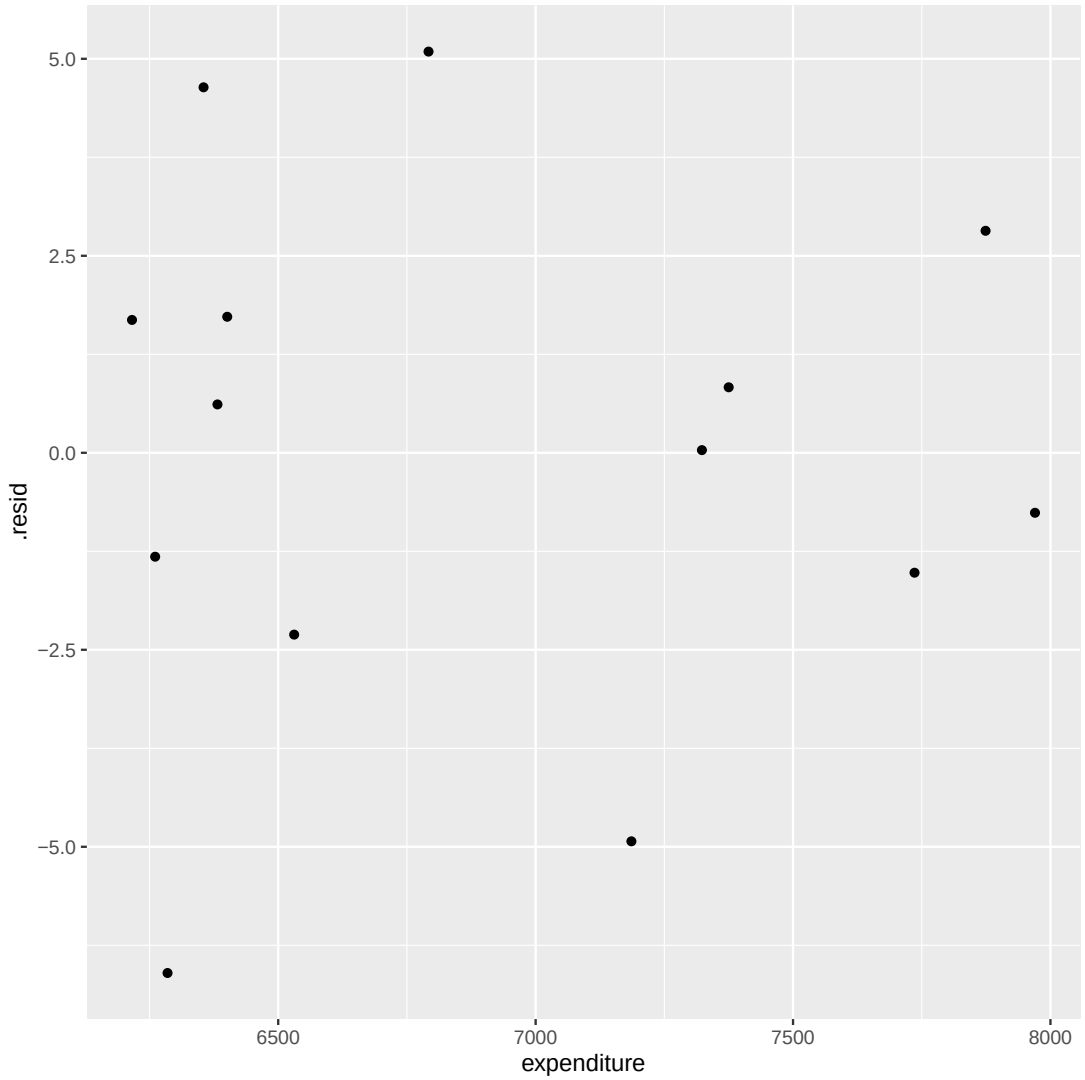
The relationships with SAT score are linear, but the highest line is the green one, going with an expenditure of \$7,000, not the highest or lowest.

I like this way of understanding how models are behaving: do some predictions and then plot them or otherwise look at them. Some of the models we will see in D29 are best understood this way, to my mind.

(j) (3 marks) Plot the residuals from your last regression against expenditure. Does the problem seem to have been solved?

Solution: This again means making a data frame with the residuals and the original data in it. I like `augment` for this. Since we are only making one plot this time,[3] we can use the pipeline thing and pipe the output from `augment` directly into `ggplot`, thus:

```
gradrate.3 %>% augment(gradrate_x) %>%
  ggplot(aes(x=expenditure, y=.resid)) + geom_point()
```



The up-and-down has gone, and this is random, so adding the squared term has solved our problem. Don't get too swayed by that one very positive residual in the middle; before, the ones to the left of it had an upward trend and the ones to the right a downward trend, but that has more or less gone away now.

That's my opinion, at least; if you can make a sufficiently persuasive case that the problem is still there, then that's OK too. If that's what you think, the issue might be one of "the wrong kind of curve", one that a quadratic term doesn't fix. For example, a quadratic curve has to come down at the same rate that it went up. (That is, it has a left-right symmetry.) If you go back to the plot of residuals against expenditure in (h), you could reasonably say that it comes down slower than it went up. This is not well described by a quadratic curve (adding a squared term in expenditure); doing so might improve things, but you could reasonably say that we should still do better.