Regression Example: Bank salaries

Paul Johnson

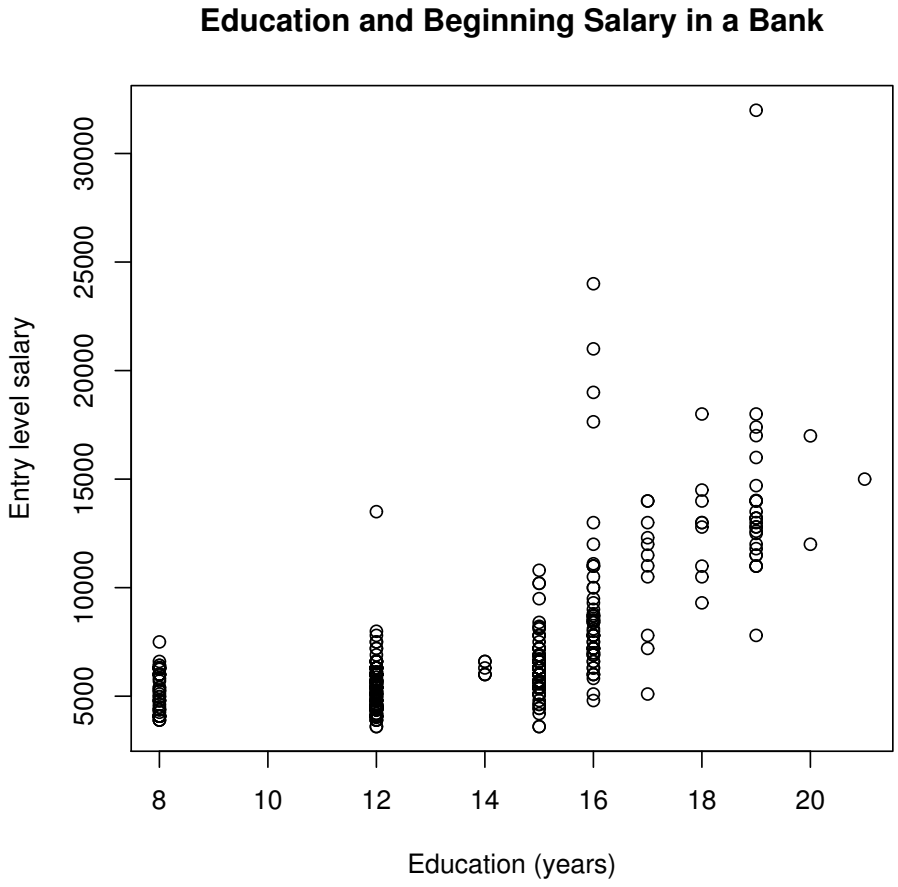
Sept˙21¸2004

1 Introduction

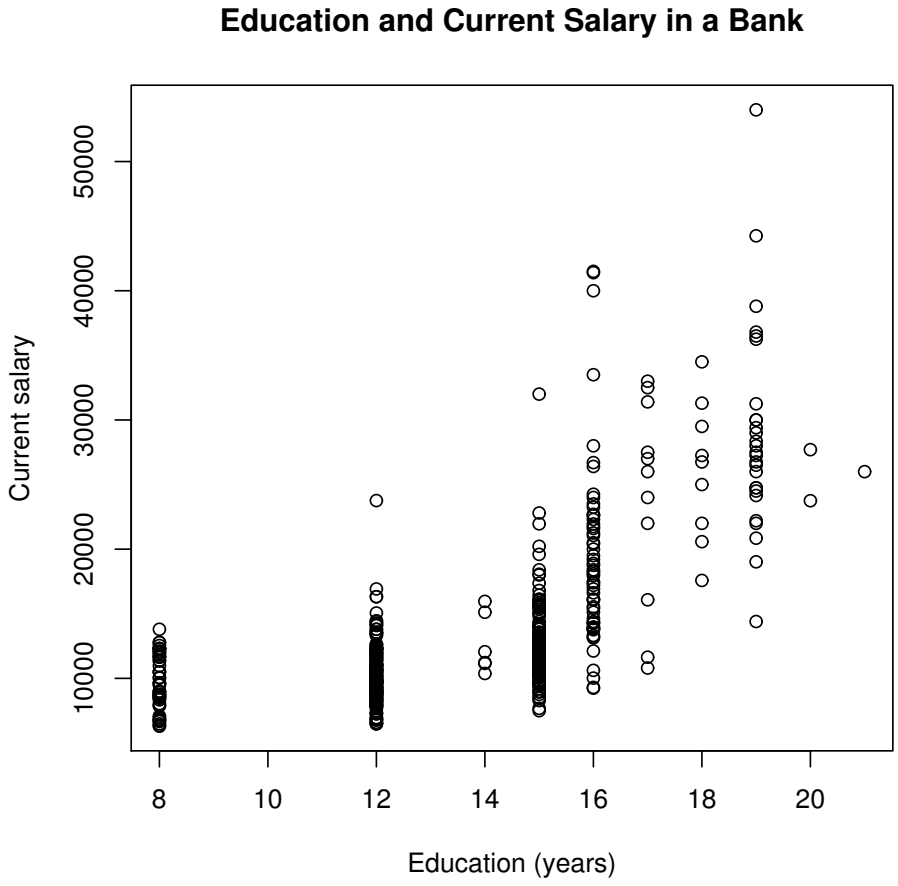
Bivariate Regression¨The Famous SPSS "bank dataset" is bank.sav. I've been seeing this dataset for more than 20 years. I think I've got it this time!

There is an accompanying R program "bankReg.R". It requires some user interaction at 2 points, so you can't just run it all through at once. Instead, you have to step through. If you do, it will create all the figures presented here, plus some more.

2 Consider Education and Beginning Salary



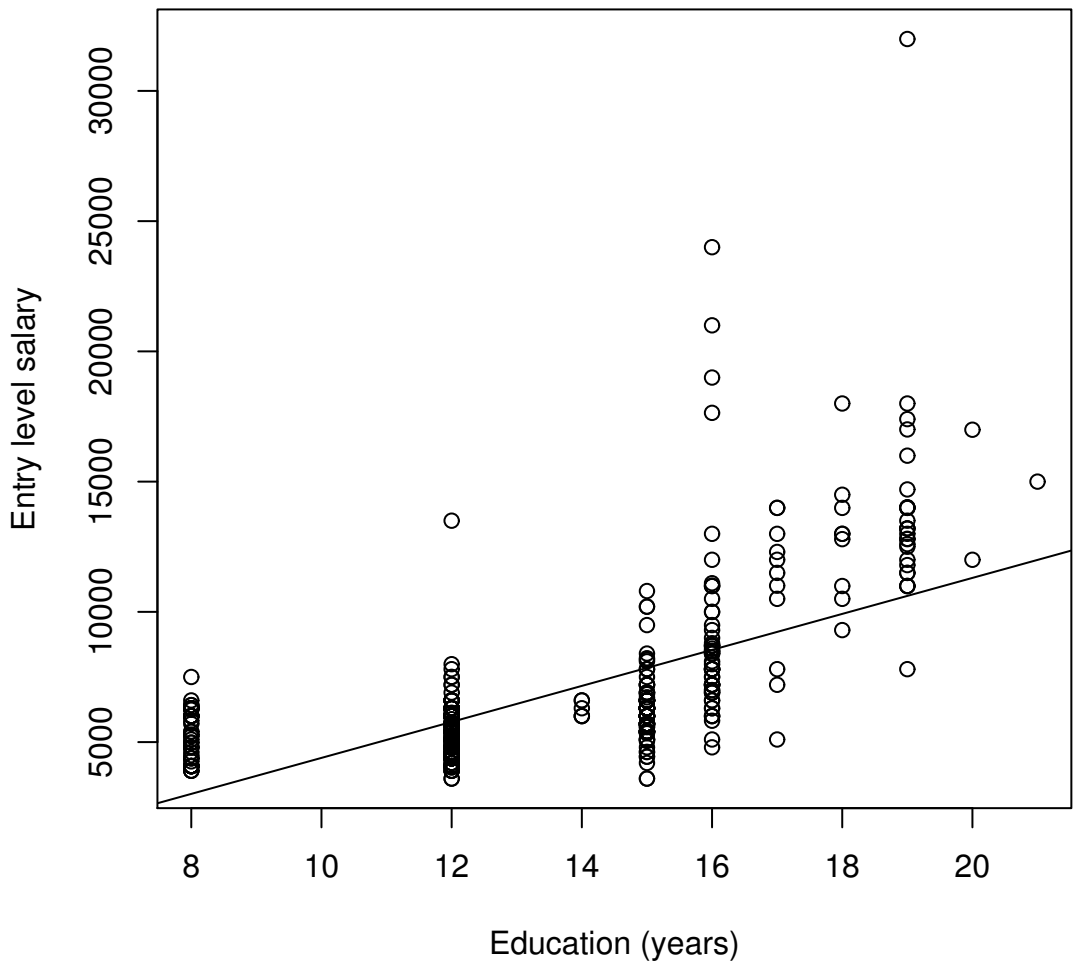
We could plot current salary, but there's no benefit.



3 Ordinary Least Squares

Here's the table that pops out of R's xtable() routine. The only change I made by hand was to add in the R^2, the root mean squared error, *RMSE*, and the sample size *N*.

	Estimate	Std. Error	t value	Pr(>|t|)
(Intercept)	−2516.3869	536.3679	−4.69	0.0000
EDLEVEL	691.0113	38.8788	17.77	0.0000
		$R^2 = 0.40$	RMSE =2439	N = 474



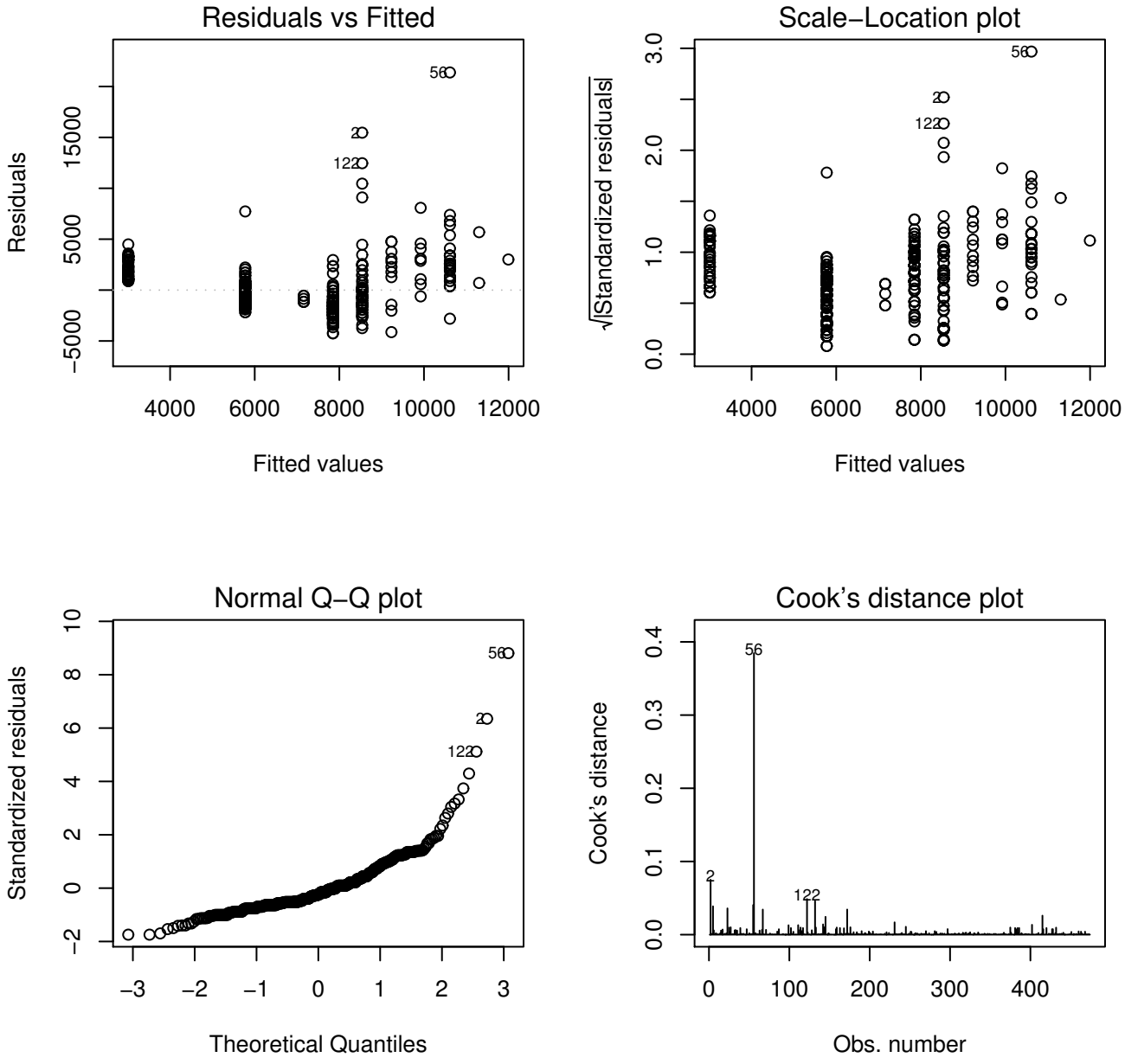
The plot() function, applied to a linear regression model, activates the custom plotting facility built into lm. That will print out four figures, one at a time, and you can either step through those one at a time with commands like this (for a regression *myReg1*).

```
> par(ask=TRUE)
> plot(myReg1)
> par(ask=N)
```

It is important to turn off the "ask" option when you don't want it anymore, because it gets really boring to keep hitting the enter button when you want a graph. I did not bother to print out those graphs, but if you run bankReg.R, you will find them called "importfigs/myReg1Diag01.eps", "importfigs/myReg1Diag02.eps", "importfigs/myReg1Diag03.eps", "importfigs/myReg1Diag04.eps".

Instead of printing out all 4 separately, I've positioned them into a single graph.

```
par(mfcol=c(2,2))
plot(myReg1)
dev.copy2eps(file="myReg12by2.eps", horizontal=F)
par(mfcol=c(1,1))
```

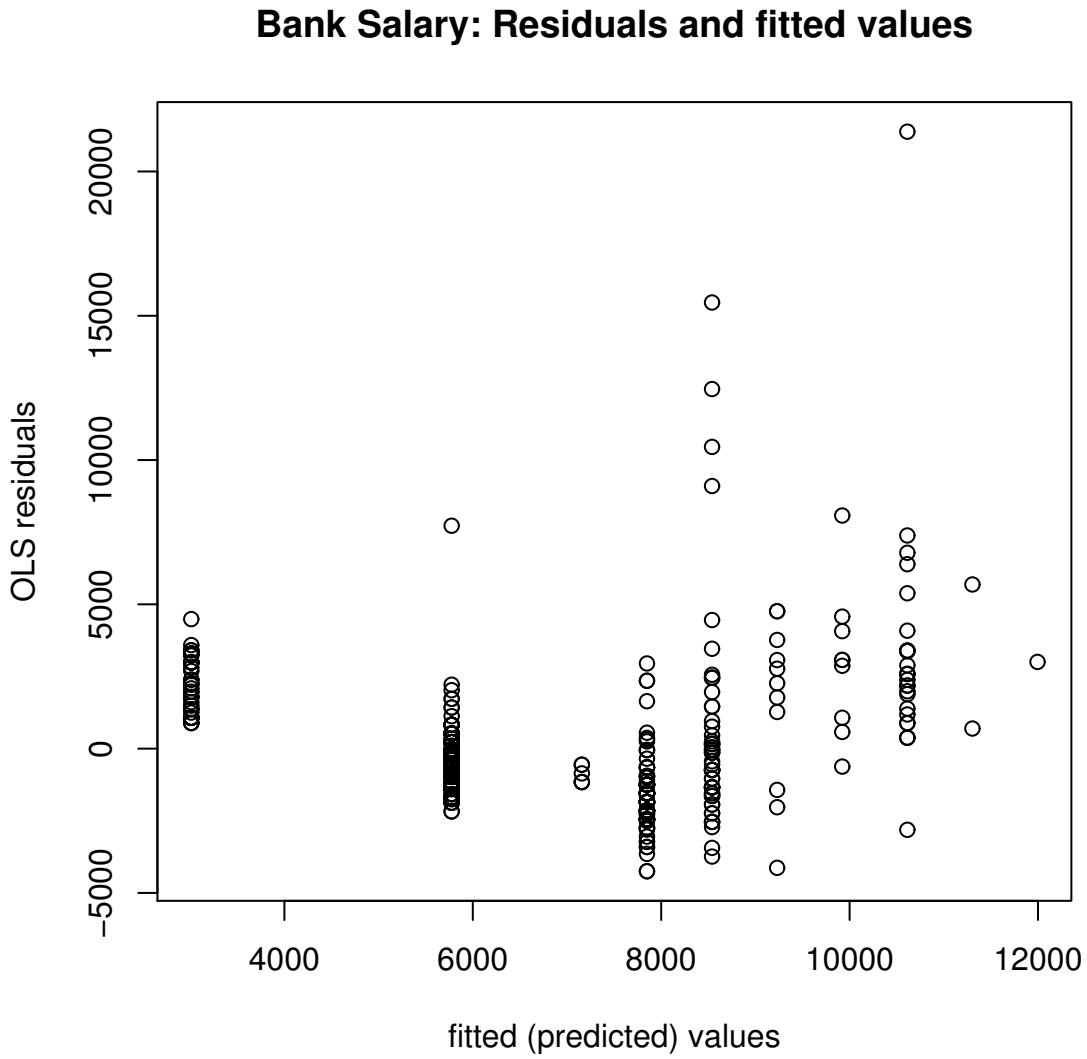


Several problems are obvious. The residuals don't appear to scatter evenly above and below the line as you look from side to side. There are some extremely influential cases, and I've seen better Q-Q plots, come to think of it.

It is a useful exercise to take a model object and then use its attributes to reconstruct graphs of this

sort. The top left is done as:

```
plot(myReg1$fitted.values, myReg1$residuals,
  main="Bank Salary: Residuals and fitted values",
  xlab="fitted (predicted) values",
  ylab="OLS residuals")
```

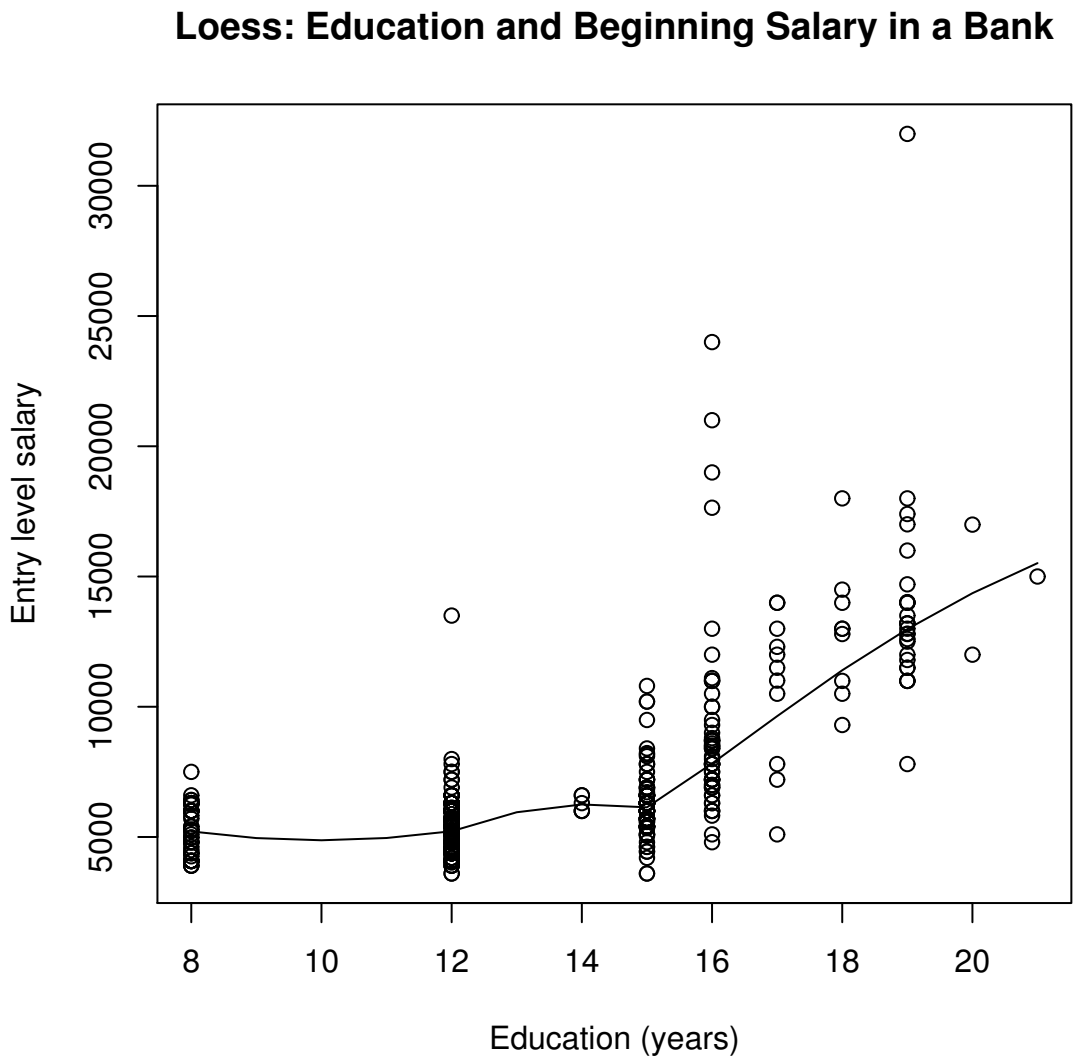


4 Smoothed Curves: Loess, Lowess

In R, there is a "loess" program in the R-base, and there is "lowess" as well. The help page for lowess says it is older and implies that one ought to use loess. I tried both.

4.1 Loess works OK (If you know the secret)

The loess output from my first effort did not look right to me. But this version looks fine:



In contrast to lowess object, a loess will not answer to the simple

lines(myLowess1)

and that was fooling me for a while. But here is the secret recipe:

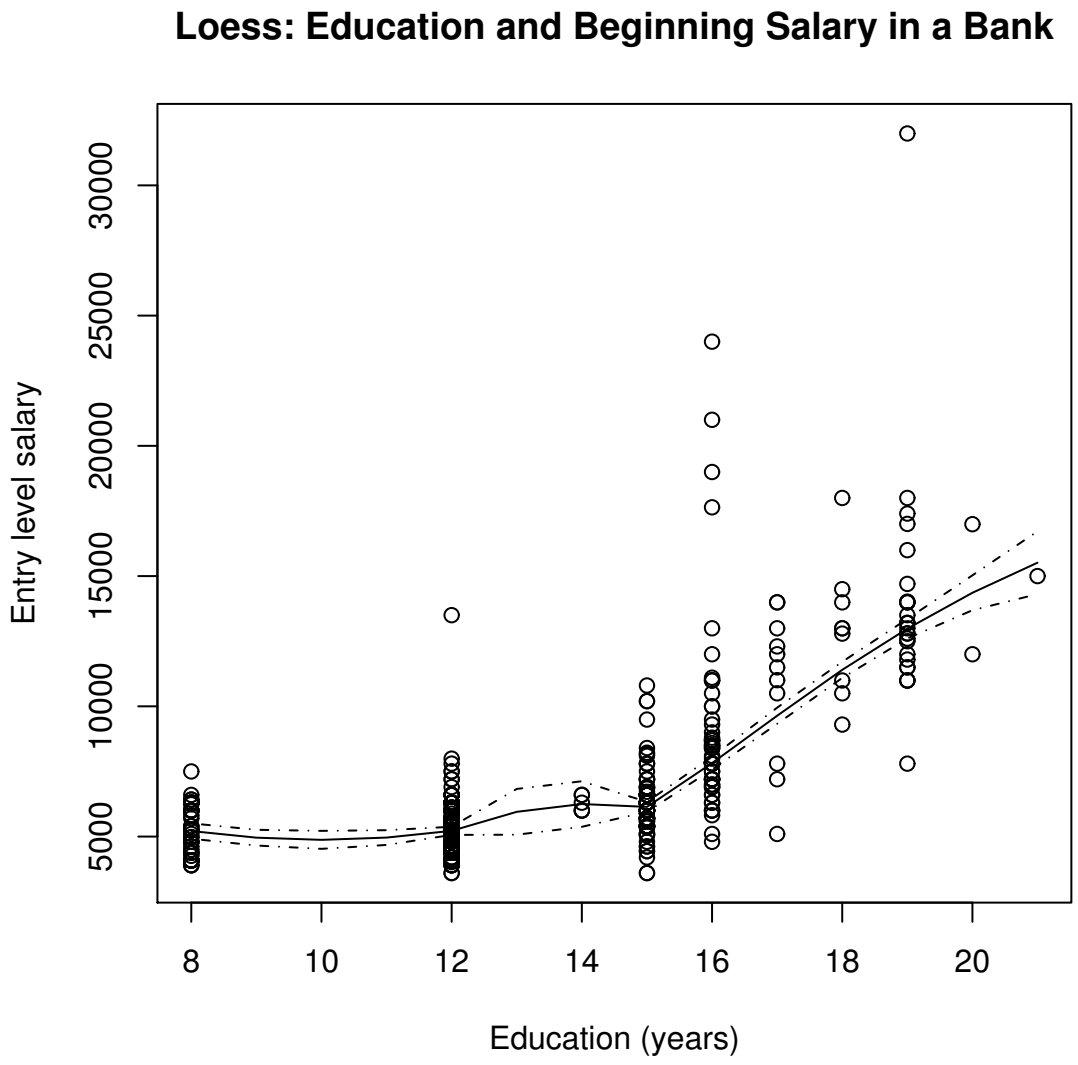
```
myReg.lo <- loess(SALBEG~EDLEVEL, data=bank, span=0.67,
  control=loess.control(surface="direct"), family="symmetric")
EDRange <-  seq(min(EDLEVEL),max(EDLEVEL),1)
lo.pred <- predict(myReg.lo,EDRange, se=TRUE)
plot(EDLEVEL,SALBEG,main="Loess: Education and Beginning
  Salary in a Bank",xlab="Education (years)",
```

```
  ylab="Entry level salary")
lines(EDRange, lo.pred$fit)
```

Further, if note I added the flag se=T for the predict method. That means it will calculate and output the standard error of the predicted value, and those can be added to the plot:

```
lines(EDRange,lo.pred$fit +1.96*lo.pred$se, lty=4)
lines(EDRange,lo.pred$fit -1.96*lo.pred$se, lty=4)
```

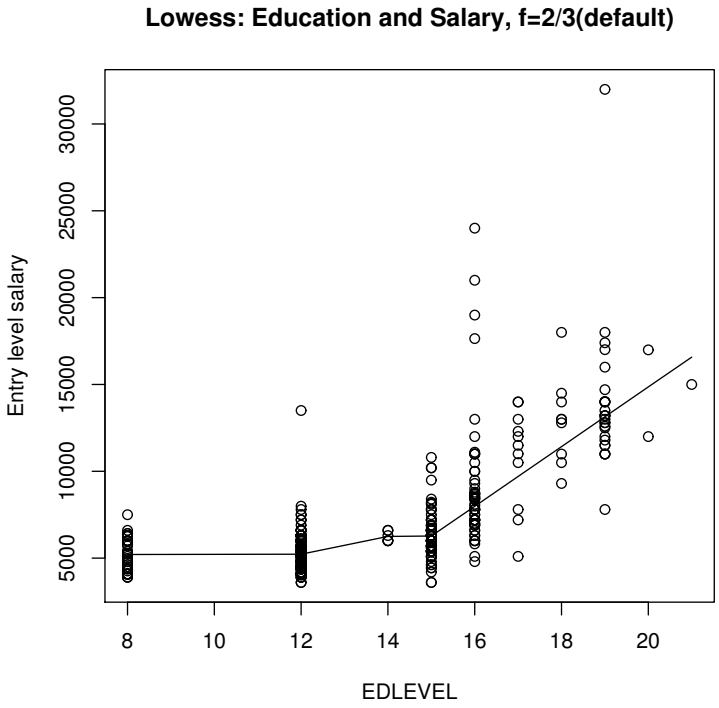
I believe these are the standard error of the fitted value, rather than the standard error about predictions for individual cases. In any case, we obtain:



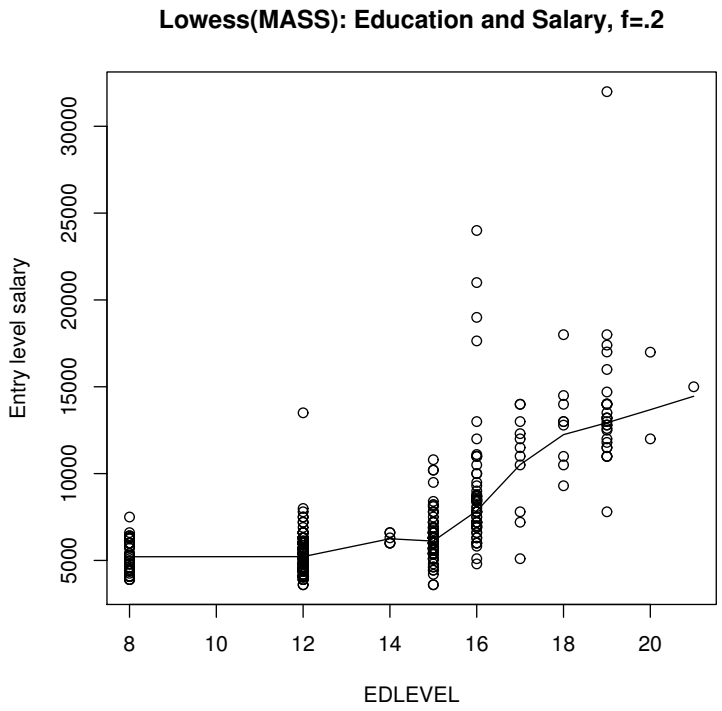
4.2 Lowess is a little easier

Let's compare with "lowess". The main parameter is a setting which controls the proportion of all points that are entered into the "local window" around a point. By default, lowess uses a very broad window. Plotting is slightly easier, since lines() can be directly applied to a lowess object, as in:

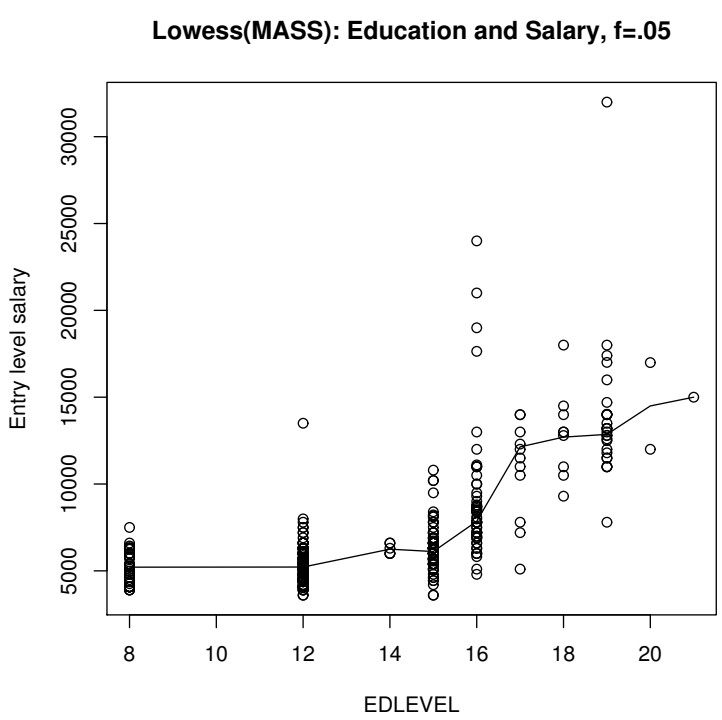
```
myReg1.low <- lowess(EDLEVEL, SALBEG,f=.2)
salBegPlot1("Lowess(MASS): Education and Salary, f=.2")
lines(myReg1.low)
```



If you like to see more abrupt curves, tune that parameter down:


Lowess(MASS): Education and Salary, f=.2
Entry level salary
5000
10000
15000
20000
25000
30000
8
10
12
14
16
18
20
EDLEVEL

Or:


Lowess(MASS): Education and Salary, f=.05
Entry level salary
5000
10000
15000
20000
25000
30000
8
10
12
14
16
18
20
EDLEVEL

5 Put a Knot in there

To my eye, it appears we need to consider the possibility that the relationship between education is not a single "straight line." It appears rather more likely that up to some level of education, say 14, the people who work at the bank don't gain all that much from additional schooling. With additional years, however, the increase is steep.

Consider a regression model with a "slope shift" and an "intercept shift". The R command:

```
myReg2 <- lm(SALBEG~EDLEVEL*I(EDLEVEL > 14), data=bank)
```

does the work easily. It estimates a regression with a threshold value set at 14. We estimate slope and intercept shift at 14,

$$\hat{salary} = b_0 + b_1 education + b_2 threshold + b_3 education * threshold$$

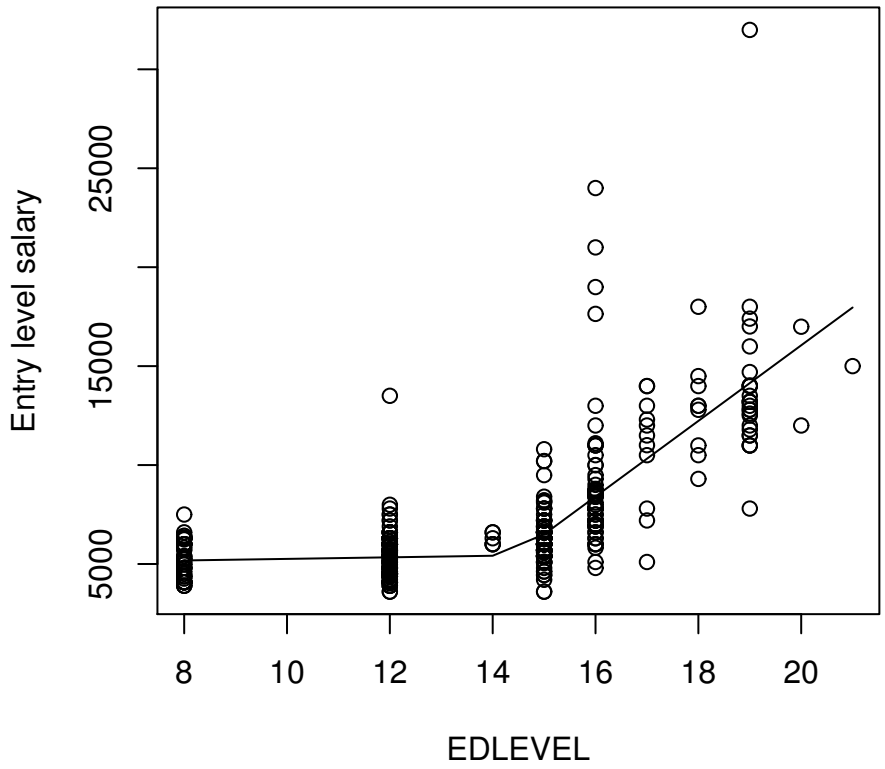
R will "automatically" estimate all of these parameters with the command:

```
myReg4 <- lm(SALBEG~EDLEVEL*I(EDLEVEL > 14), data=bank3)
summary(myReg4)
```

The "indicator" I(EDLEVEL > 14) is a True/False variable, which lm treats as a factor. R creates a "dummy variable", a 0-1 variable, to enter in the model. The R jargon for this is that the coding creates a "contrast."

	Estimate	Std. Error	t value	Pr(>|t|)
(Intercept)	4862.6493	834.1845	5.83	0.0000
EDLEVEL	39.4700	73.6675	0.54	0.5924
I(EDLEVEL > 14)TRUE	−27024.2122	1690.2222	−15.99	0.0000
EDLEVEL:I(EDLEVEL > 14)TRUE	1871.1925	117.3404	15.95	0.0000
		R^2 = 0.61	RMSE = 1965	N = 474

Linear model with 1 knot at 14: note bozosity b/t 14 &

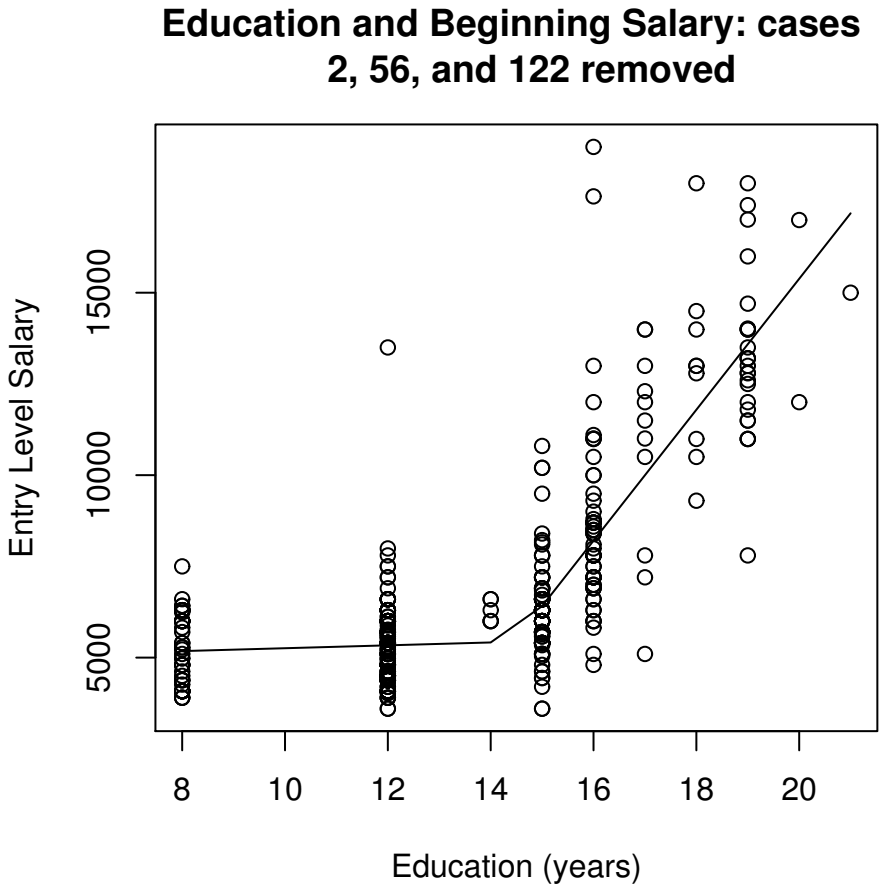


6 Outliers

Rather than worry too much on the fitting of a model with some egregious outliers, let's pursue the wiser course. In the diagnostic plots above, it is apparent that cases 2, 56, and 122 are inordinately influential. As a result, we should get rid of them, see what happens.

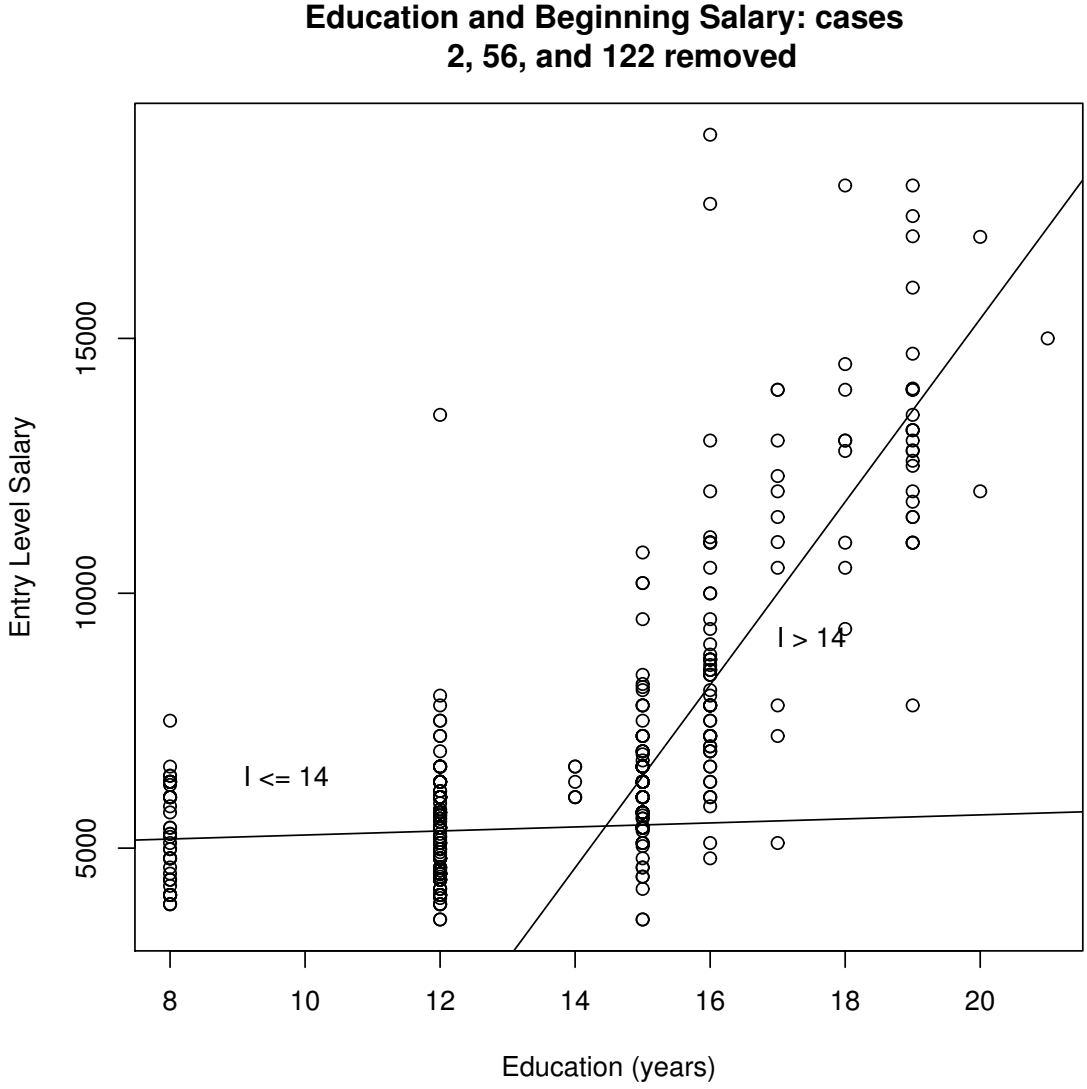
The linear model appears to be much more desirable!

	Estimate	Std. Error	t value	Pr(>|t|)
(Intercept)	4862.6493	645.1940	7.54	0.0000
EDLEVEL	39.4700	56.9776	0.69	0.4888
I(EDLEVEL > 14)TRUE	−25371.0827	1316.0651	−19.28	0.0000
EDLEVEL:I(EDLEVEL > 14)TRUE	1755.0430	91.2908	19.22	0.0000
	R^2 = 0.69	adj-R^2=0.69	RMSE = 1520	N = 471

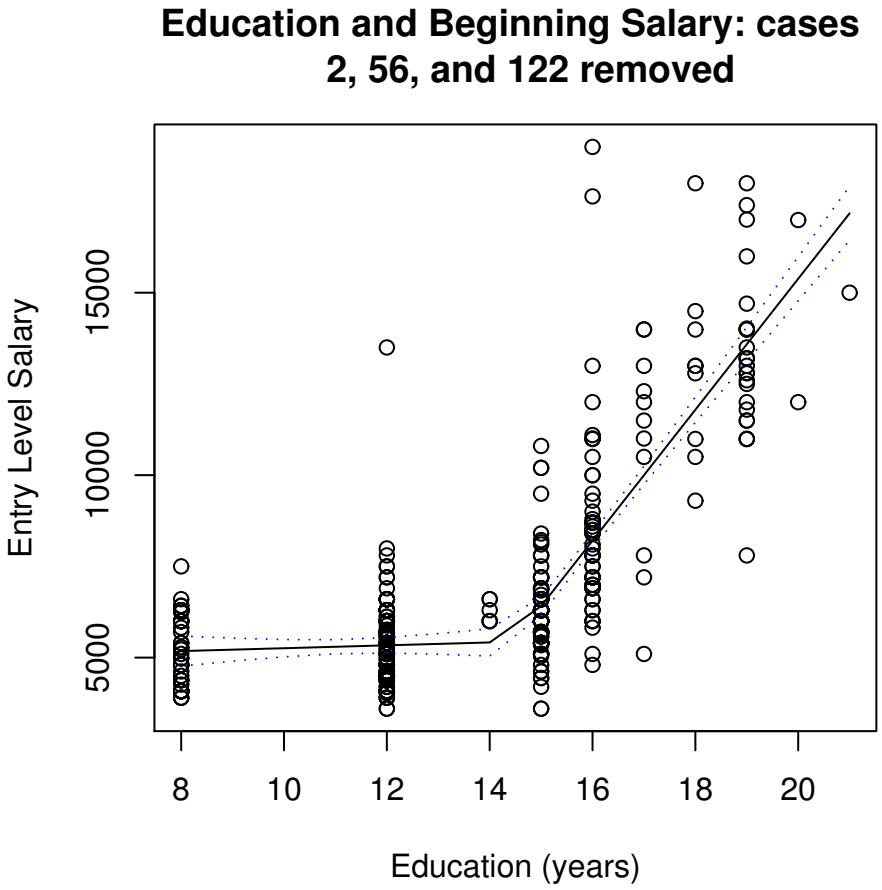


The only problem here is that the graph is a little bogus because, between 14 and 15, the line looks like it has a 3rd slope, and they don't.

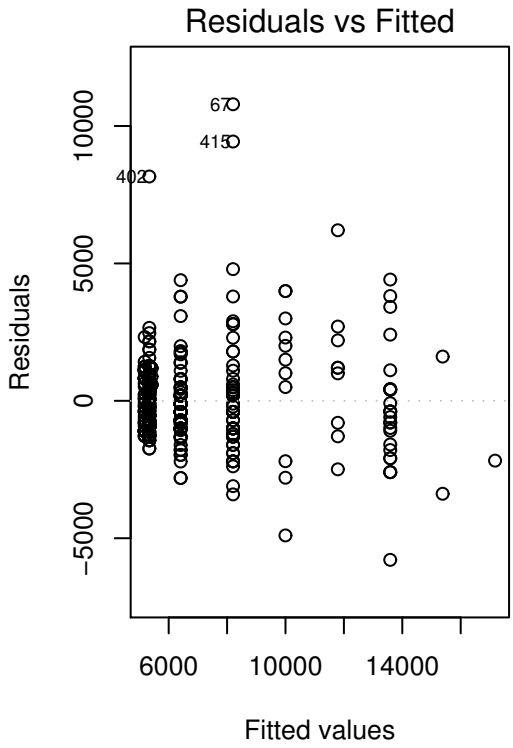
I didn't yet find a perfectly good way to make a line that has a kink. Here are 2 lines to give you the idea of what's going on.

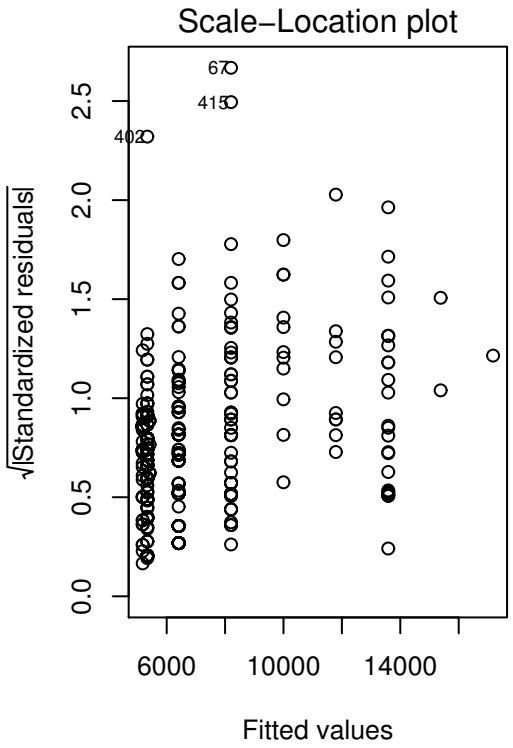


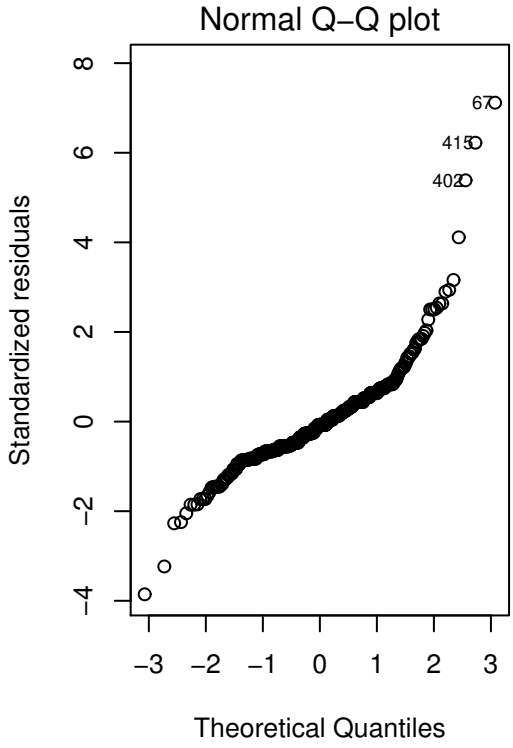
It might be cute to plot the 95% confidence interval on the fitted values. Again, this is the confidence we have in the point prediction, not confidence about predictions on individual cases. It means something like: "With probability 0.95, the estimate of the fitted value would be inside this range." It does not mean: "With probability 0.95, a randomly drawn individual will fit inside this range." In order to get that estimate, a different calculation is required.

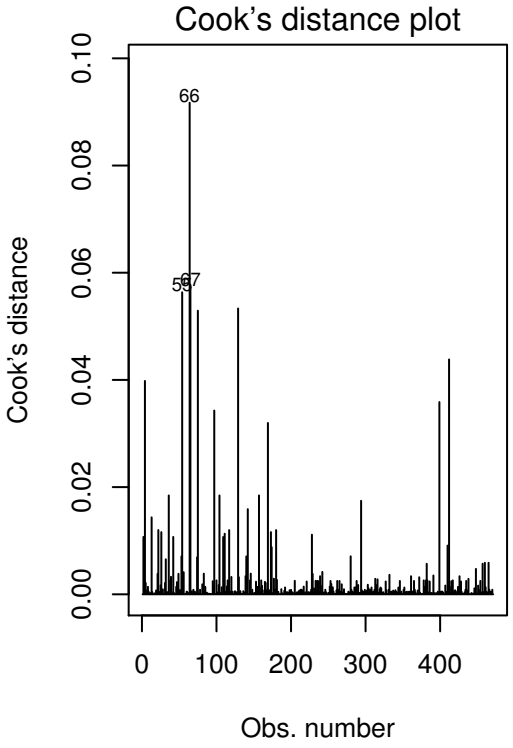


I almost forgot to do a diagnostic plot on this final fitted model. Its not too bad, really.


Residuals vs Fitted
Residuals
Fitted values
67
415
402
−5000
0
5000
10000
6000
10000
14000


Scale−Location plot
√|Standardized residuals|
Fitted values
67
415
402
0.0
0.5
1.0
1.5
2.0
2.5
6000
10000
14000


Normal Q−Q plot
Standardized residuals
Theoretical Quantiles
67
415
402
−4
−2
0
2
4
6
8
−3
−2
−1
0
1
2
3


Cook's distance plot
Cook's distance
Obs. number
66
58
67
0.00
0.02
0.04
0.06
0.08
0.10
0
100
200
300
400