BIFX 553 - Discussion 3

Randy Johnson

February 2, 2017

Assessing Model Fit and Assumptions

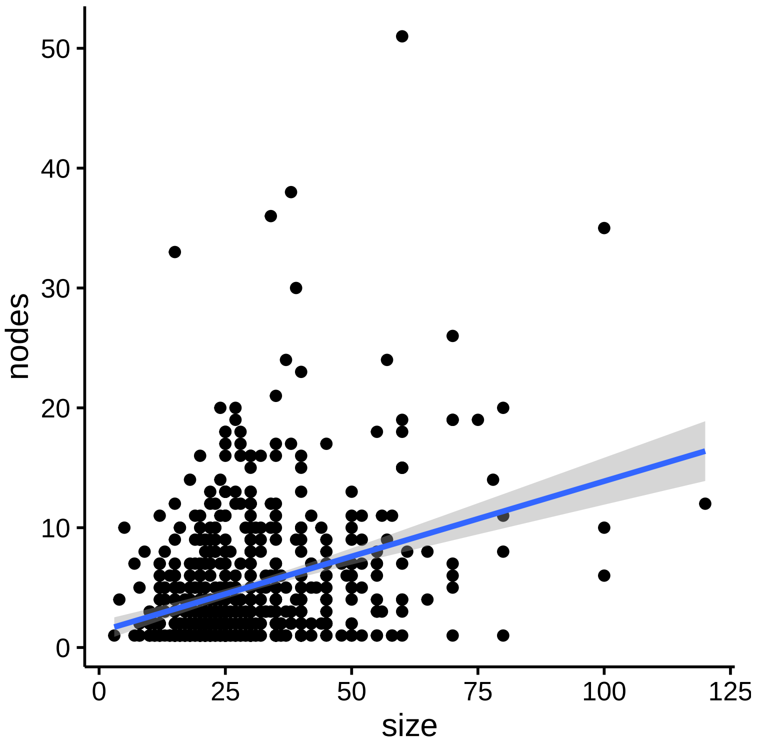
Regression Assumptions

We will primarily use visual inspection and the `car` package for checking regression assumpitions, but there are many other resources in R to do this (e.g. the `rms` and `gvlma` packages).

- Linear relationship
- Multivariate Normality
- No/little multicollinearity
- No autocorrelation
- Homoscedasticity

How does our model hold up?

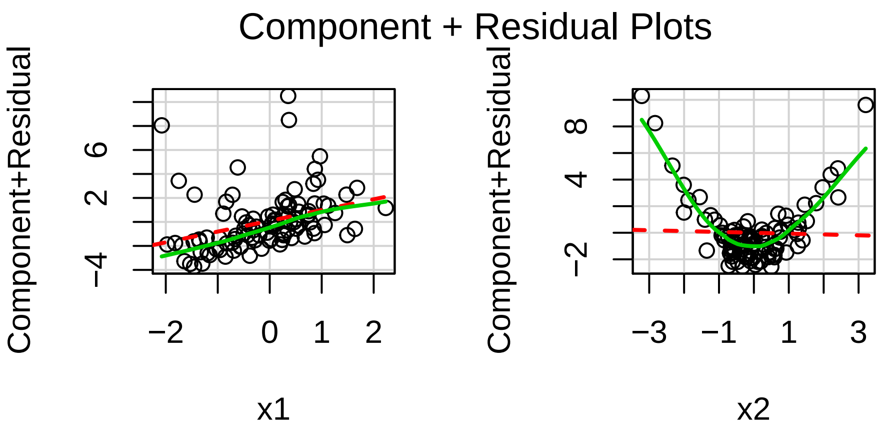
```r
load('../1-26/gbsg.RData')

# model from Discussion2
gbsg.lm <- lm(nodes ~ age + size + grade + meno + pgr + er + hormon, data = gbsg)

ggplot(gbsg, aes(size, nodes)) +
  geom_point() +
  geom_smooth(method = 'lm')
```



Linear Relationship

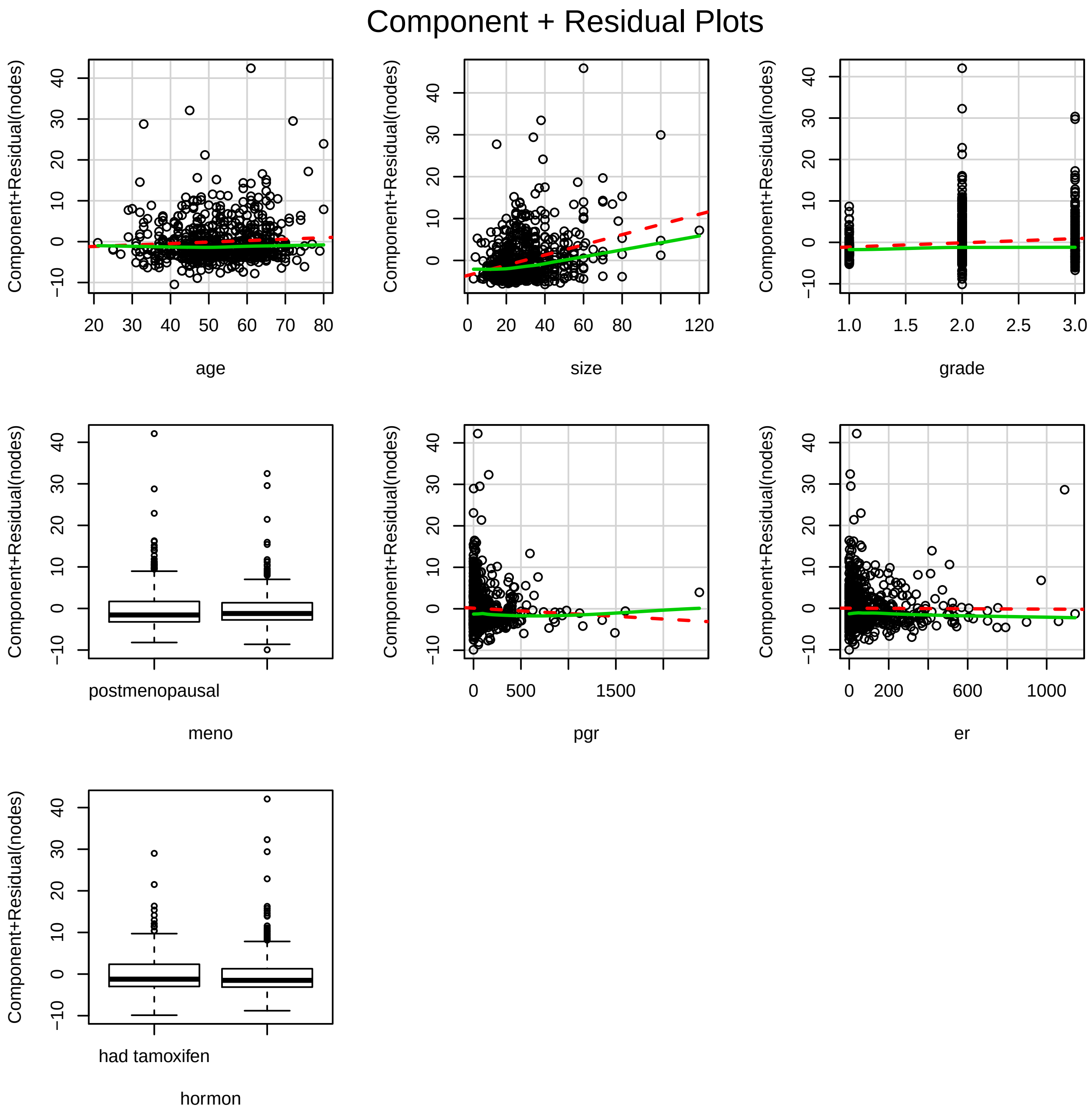
Component residual plots are a way to see if the predictors have a linear relationship with the outcome variable. The red, dashed line in the figures below represents the best fit of each preidictor and the residuals, and the solid, green line is a running, smoothed average along the x-axis. In this example,

```r
tmp <- data_frame(x1 = rnorm(100),
                  x2 = rnorm(100),
                  y = x1 + x2^2 + rnorm(100))
lm(y ~ x1 + x2, data = tmp) %>%
  crPlots()
```



How does our model look?

```r
crPlots(gbsg.lm2)
```



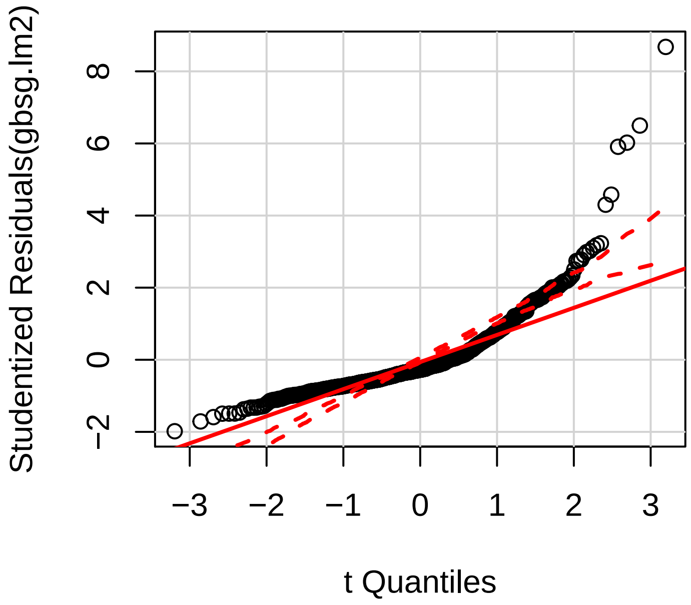
Multivariate Normality

There are a couple of ways we can look at normality. The Sapiro-Wilk test for normality will give you a quantitative measure of whether your residuals are normally distributed, and the QQ plot will give you a graphical way to see what is going on with your residuals.

```
# Shpiro-Wilk test for normality
shapiro.test(augment(gbsg.lm2)$.std.resid)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  augment(gbsg.lm2)$.std.resid
## W = 0.79776, p-value < 2.2e-16
```

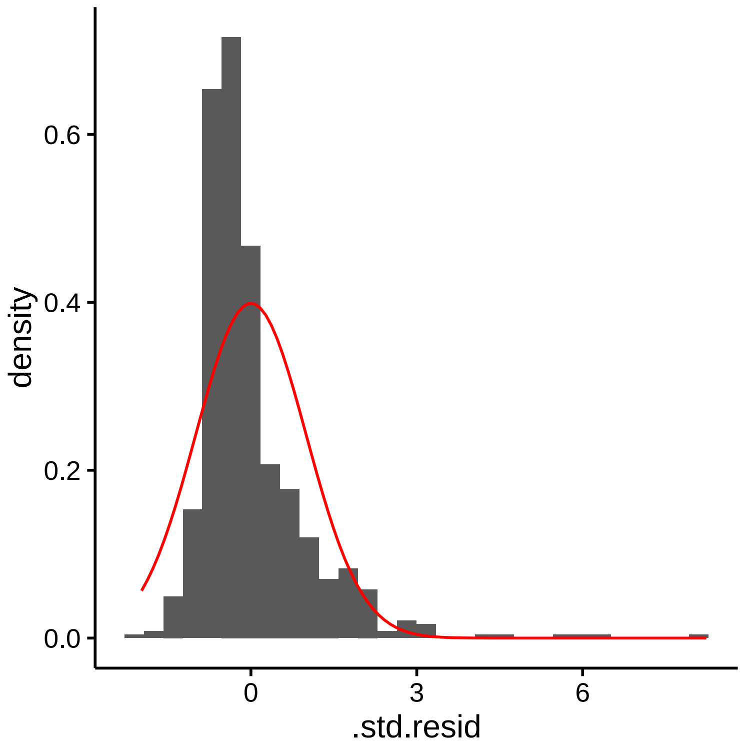
```
# quantile quantile plot
qqPlot(gbsg.lm2)
```



You could also plot a histogram of your residuals along with a normal distribution.

```
# take a look at the residuals
augment(gbsg.lm2) %>%
  ggplot(aes(.std.resid)) + # plot studentized residuals on x-axis
  geom_histogram(aes(y = ..density..)) + # plot histogram as density, rather than frequency
  stat_function(fun = dnorm, color = 'red') # put a N(0,1) density over the top
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



No/little multicollinearity

Variance inflation factors give you a meeasure of how much the residuals are inflated when each predictor is added to the model (compared to the full model minus the predictor in question). This can give us an idea for what predictors might be correlated with eachother. Anything greater than 2 should certainly give us pause.

```
# variance inflation factors
vif(gbsg.lm2)
```

```
##      age     size    grade     meno      pgr       er   hormon
## 2.579321 1.016654 1.050404 2.513984 1.220625 1.327381 1.096945
```

No autocorrelation

Autocorrelation most often occurs when there is a correlation between observations at regular time intervals.The Durbin-Watson test will give us a measure of autocorrelation (the null hypothesis is that there is no autocorrelation).

```
durbinWatsonTest(gbsg.lm2)
```

```
##  lag Autocorrelation D-W Statistic p-value
##    1    -0.004115498      2.005419   0.976
##  Alternative hypothesis: rho != 0
```

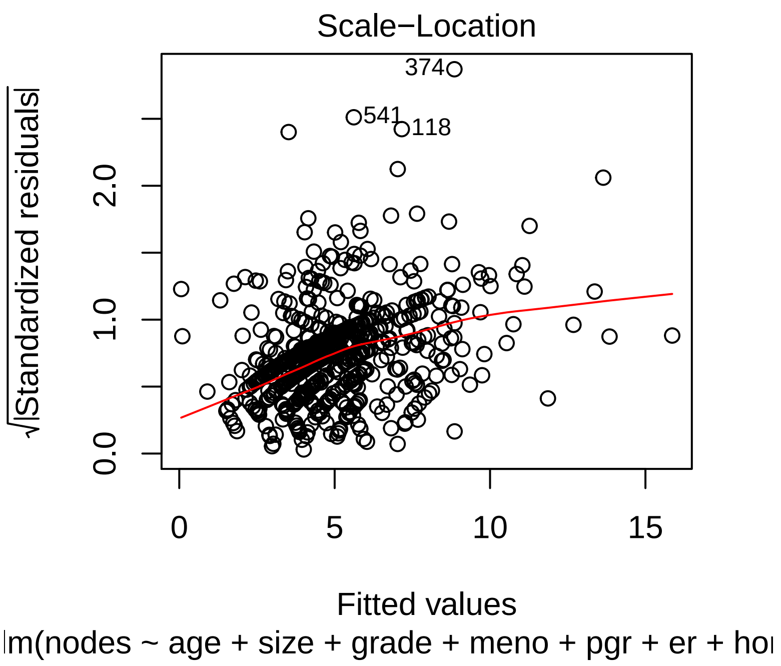
Homoscedasticity

Heteroscedasticity refers to a model where the variance about the predicted value of the outcome changes as the predicted value increases. We can check for this using a score test for non-constant error variance or graphically.

```
# statistical test for heteroscedasticity (null is that they are homoscedastic)
ncvTest(gbsg.lm2)
```

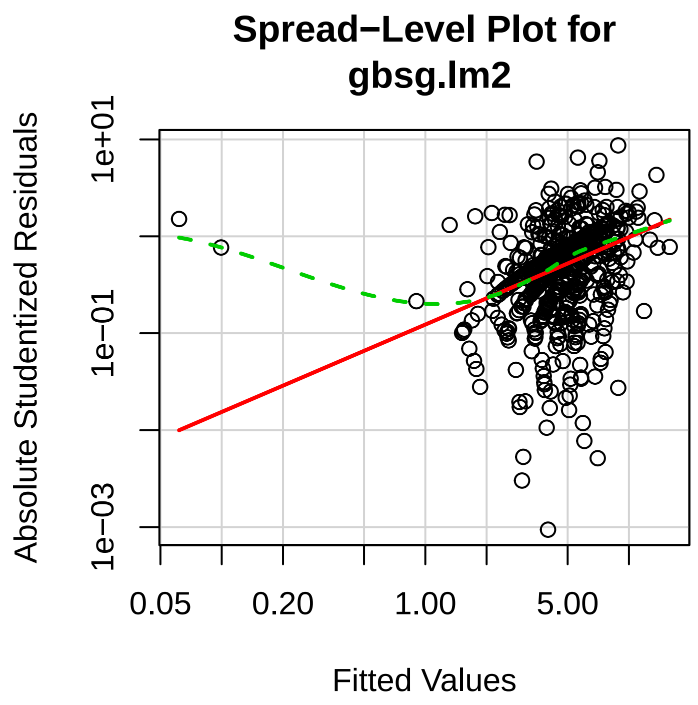
```
## Non-constant Variance Score Test
## Variance formula: ~ fitted.values
## Chisquare = 165.5525    Df = 1     p = 6.928119e-38
```

```
# scale-location plot
plot(gbsg.lm2, which = 3)
```



The `spreadLevelPlot` function gives us a slightly nicer picture of this. The solid red line is what we would hope to see under a homoscedastic model, and the green line is what is actually observed. These two plots (the one just above and the one just below) are represenations of the exact same data. The plot below, however, is scaled differently to highlight departures from the expected.

```
# another scale-location plot
spreadLevelPlot(gbsg.lm2)
```



```
##
## Suggested power transformation:  0.09857515
```

Outliers / Influential Points

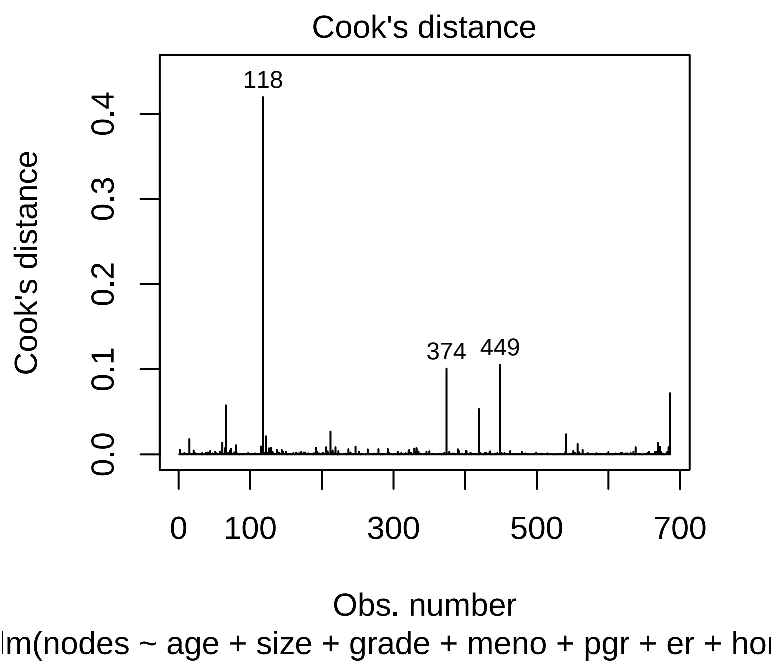
The `outlierTest` function gives us a list of the most significant outliers in a model by row number. The maxumum number of rows to return is controlled by the `n.max` function argument.

```r
# returns the most significant outliers in the residuals
outlierTest(gbsg.lm2)
```

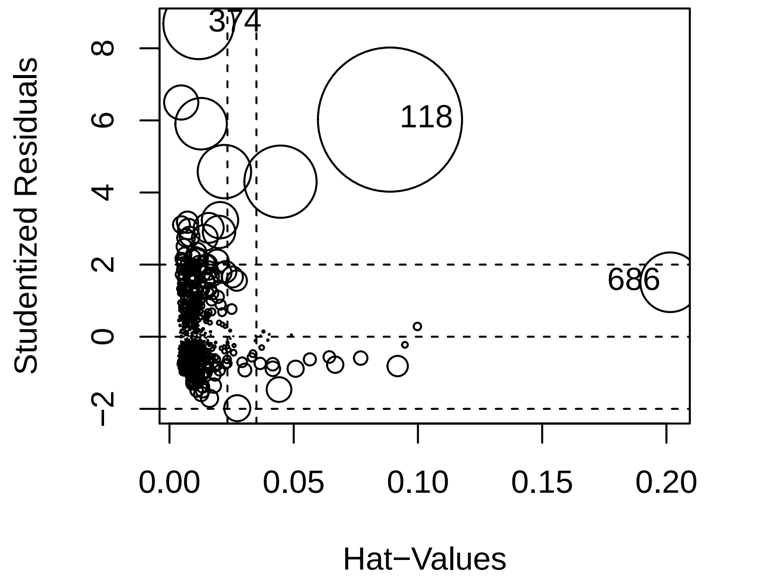
```
##     rstudent unadjusted p-value Bonferonni p
## 374 8.677313         3.0022e-17   2.0595e-14
## 541 6.496241         1.5957e-10   1.0946e-07
## 118 6.021028         2.8416e-09   1.9493e-06
## 419 5.907297         5.5069e-09   3.7777e-06
## 66  4.579290         5.5528e-06   3.8092e-03
## 449 4.299970         1.9594e-05   1.3442e-02
```

We can also graphically inspect the influence that each row of data (or each individual in the sample) has on the overall model. Cook’s distance gives us a measure of how much each row of data influences the model. Ideally we would like them all to be close to the same, but we sometimes have some values that are overly influential (see this illustration of how one data point can influence a regression fit).

```r
# Cook's Distance
plot(gbsg.lm2, which = 4)
```



```r
influencePlot(gbsg.lm2)
```



```
##       StudRes        Hat      CookD
## 118 6.021028 0.08875591 0.41956588
## 374 8.677313 0.01173853 0.10075428
## 686 1.510113 0.20151716 0.07180519
```

Influence of terms in the model

We also want to make our model as parsemonious as we can. Sometimes we will include a prediction variable because we believe that it is important, but usually we will include preditors only if they appear to be statistically important to our model. To get a quick look at the statistical significance of a predictor, we can just look at the regression output.

```
tidy(gbsg.lm2)
```

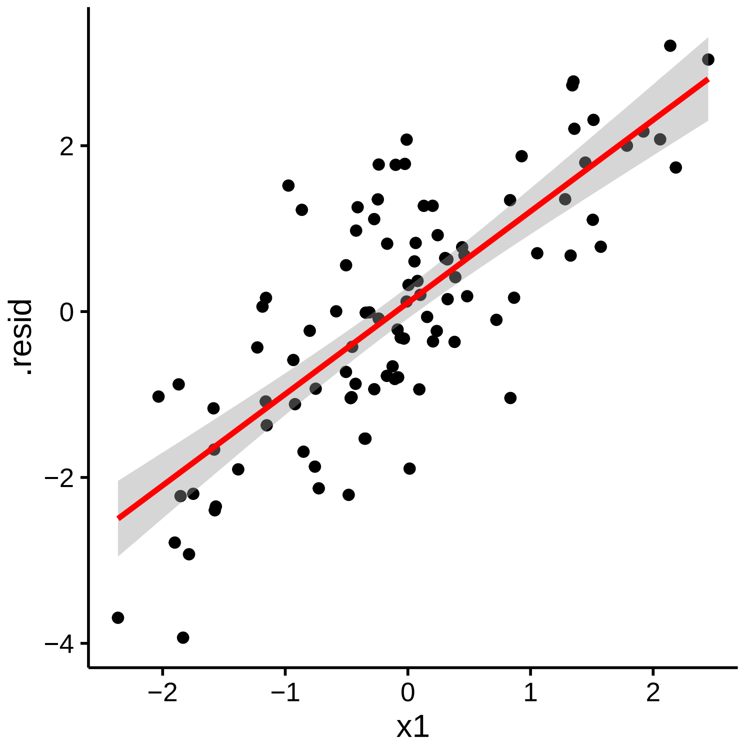
```
##                  term      estimate   std.error  statistic      p.value
## 1         (Intercept) -2.3283622032 2.079881226 -1.1194688 2.633366e-01
## 2                 age  0.0356703062 0.031204218  1.1431245 2.533904e-01
## 3                size  0.1216888424 0.013868785  8.7742975 1.385482e-17
## 4               grade  1.0089048656 0.345800531  2.9175920 3.644177e-03
## 5  menopremenopausal  0.1528321647 0.630688405  0.2423259 8.086009e-01
## 6                 pgr -0.0013100137 0.001073743 -1.2200440 2.228724e-01
## 7                  er -0.0002215252 0.001479929 -0.1496863 8.810566e-01
## 8 hormonno tamoxifen -0.2456421925 0.429119910 -0.5724325 5.672187e-01
```

Added Vairiable Plots provide a graphical approach to identifying how much added predictive value a specific variable will give you. In order to generate an Added Variable Plot, we need to perform two additional regressions.

```
tmp <- data_frame(x1 = rnorm(100),
                  x2 = rnorm(100),
                  x3 = rnorm(100),
                  x4 = rnorm(100),
                  y = x1 + x2 + x3 + rnorm(100))

# full model
full_model <- lm(y ~ x1 + x2 + x3 + x4, data = tmp)
tidy(full_model)
```

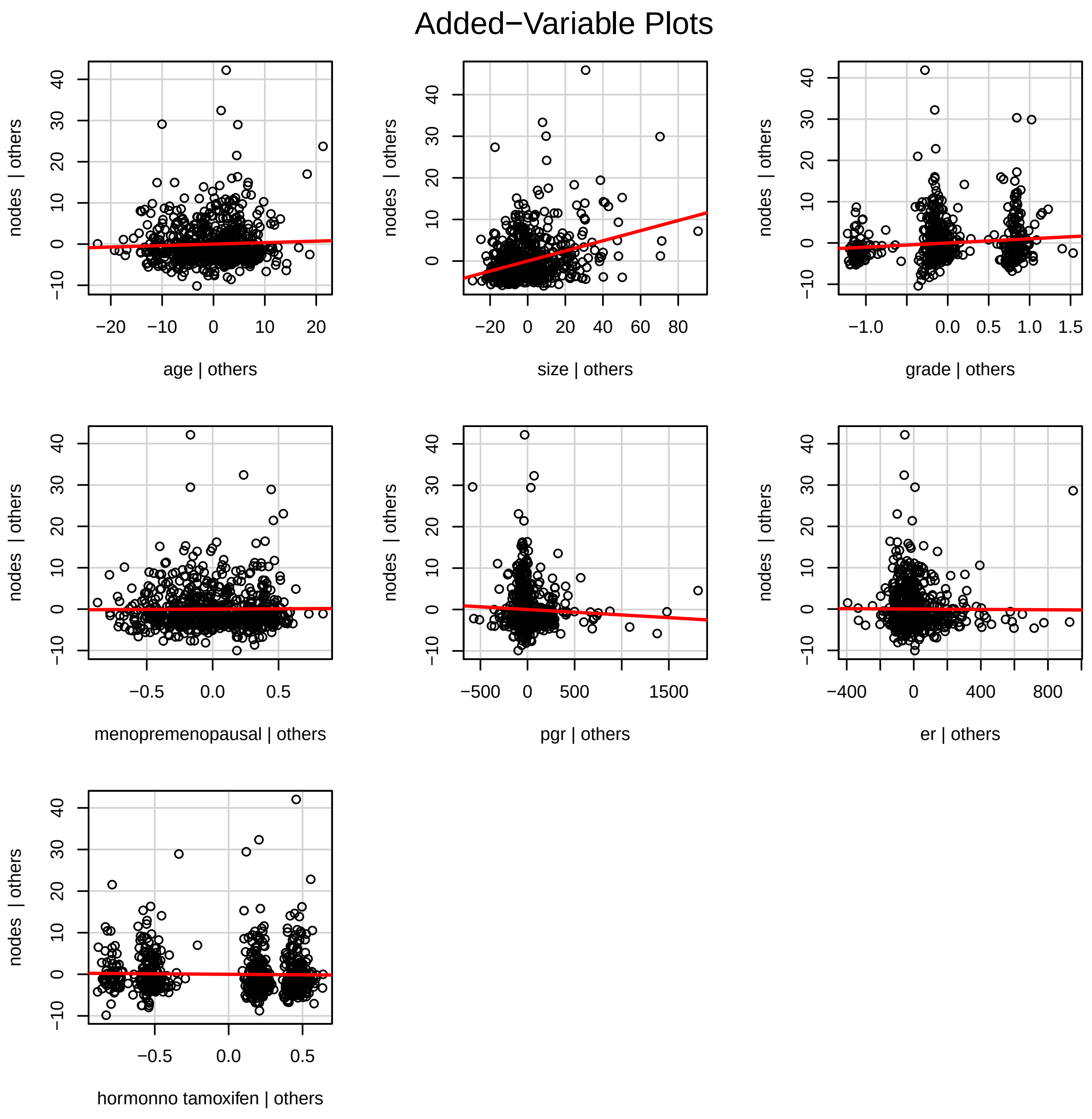
```
##          term    estimate  std.error statistic      p.value
## 1 (Intercept) -0.23597071 0.09731665 -2.424772 1.720874e-02
## 2          x1  1.11429550 0.09345619 11.923185 1.403852e-20
## 3          x2  1.07905134 0.09285788 11.620461 6.041750e-20
## 4          x3  1.00536184 0.10702587  9.393634 3.268010e-15
## 5          x4 -0.03511934 0.10439354 -0.336413 7.373014e-01
```

```
# model without x1
no_x1 <- update(full_model, . ~ . - x1) %>%
  augment()

no_x1$x1 <- tmp$x1 # put back into the data_frame, but not into the model

# now regress x1 onto y conditioning on the other variables
ggplot(data = no_x1, aes(x = x1, y = .resid)) +
  geom_point() +
  geom_smooth(method = 'lm', color = 'red')
```



Now lets take a look at all of the predictors in our model:

```r
# added variable plots
avPlots(gbsg.lm2)
```



Homework

Choose a model as a team for the clinical data in Project 1, checking all model assumptions. Each individual should submit their own description justifying your group's model choice. **Limit: 500 words.**