

STAT 224 Lecture 11

Chapter 4 Model Diagnostics, Part 2

Yibi Huang

Checking Assumptions with Graphs

The Importance of Graphs

Graphs are useful to

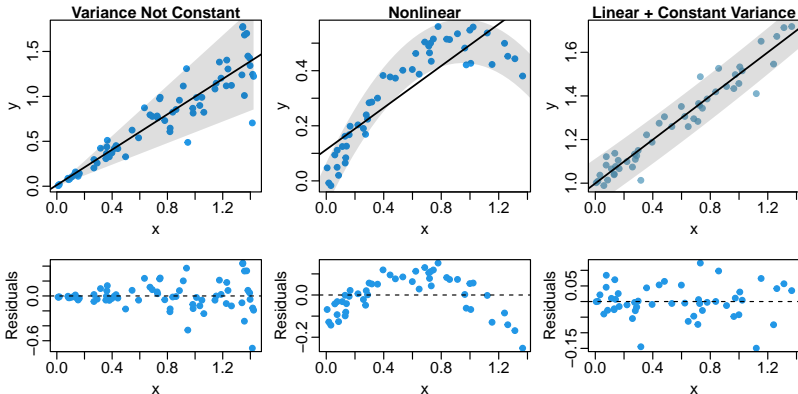
- Detect errors in the data
- Recognize patterns in the data (e.g., clusters, outliers, gaps, etc.)
- Explore relationships among variables
- Discover new phenomena
- Confirm or negate assumptions
- Assess the adequacy of a fitted model
- Suggest remedial actions (e.g., transform the data, redesign the experiment, collect more data, etc.)
- Enhance numerical analyses in general

Checking Assumptions for SLR Using Graphs

For SLR, one can plot

Y against X , Residuals against X , or Residuals against \hat{Y}

and spot problems (nonlinearity, nonconstant variability, outliers, influential points)



Checking Assumptions for MLR Using Graphs

For MLR, $Y = \beta_0 + \beta_1 X_1 + \dots + \beta_p X_p + \varepsilon$ it's much more difficult to check whether the linear form $\beta_0 + \beta_1 X_1 + \dots + \beta_p X_p$ is correctly specified.

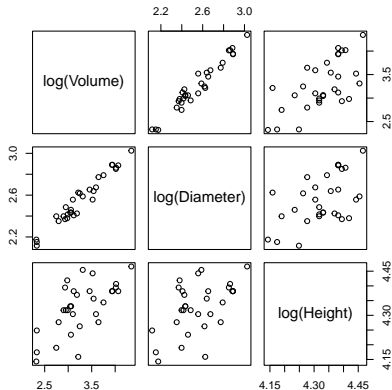
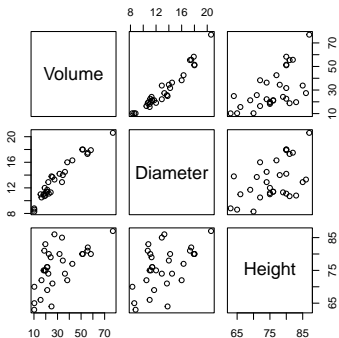
- Y is linear in X_j given other X_k 's
- Do we miss any important predictor, any interactions?
- Does each X_i have linear effect on Y given other predictors?

Tools:

- Pairwise scatterplots = Scatterplot Matrix
- Multi-panel scatterplots using `facet` feature in `ggplot`
- Plotting residuals against each predictors and potential predictors not in the model
- Added variable plots
- Residual plus component plots

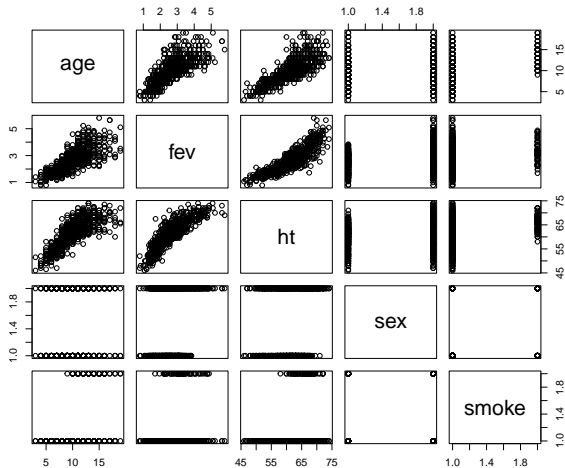
Pairwise Scatterplots

```
data("trees")  
colnames(trees) = c("Diameter", "Height", "Volume")  
pairs(Volume ~ Diameter + Height, data=trees)  
pairs(log(Volume) ~ log(Diameter) + log(Height), data=trees, oma = c(2,
```



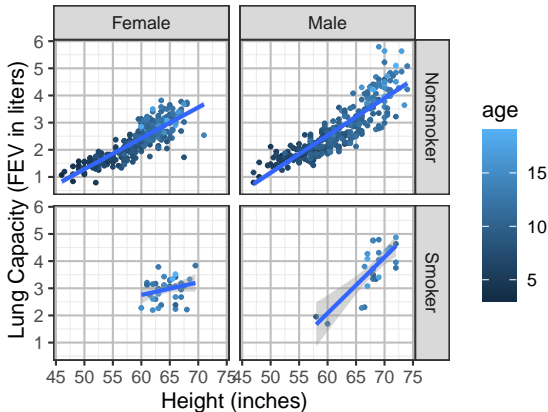
What Do You See From The Plots Below? (FEV data)

```
pairs(fevdata, oma = c(2,2,2,2))
```



A More Informative Plot than Pairwise Scatterplots

```
library(ggplot2)
ggplot(fevdata, aes(x = ht, y = fev, color=age)) +
  geom_point(cex=0.7) + facet_grid(smoke~sex) +
  geom_smooth(method='lm') +
  labs(x="Height (inches)", y="Lung Capacity (FEV in liters)")
```



Pairwise Scatterplots Are Not Very Useful

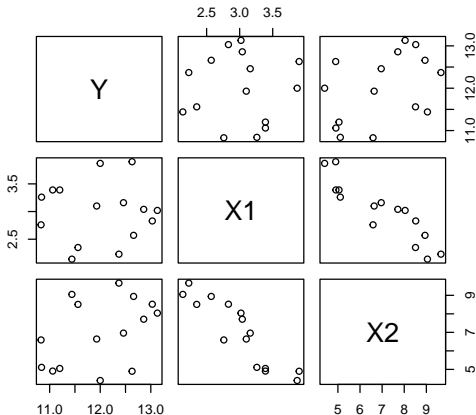
- Pairwise scatterplots only allow us to inspect the relations between variables pairwise, but not 3 or more variables at the same time
- `ggplot` are more useful in inspecting the relations between 3 or more variables as the plot on the previous page
- When there are many predictors, people usually pick the ones with large correlation between the response to begin with. However, this sometimes fails as exemplified by Hamilton's data

Hamilton's Data (p.103 of the textbook)

Download data at:

<http://www.stat.uchicago.edu/~yibi/s224/data/P103.txt>

```
hamilton = read.table("P103.txt", h = T)
pairs(hamilton, oma=c(2,2,2,2))
```



Weird Things of Hamilton's Data (1)

When X1 or X2 is the only predictor, neither of them is significant

```
summary(lm(Y ~ X1, data=hamilton))$coef
      Estimate Std. Error t value Pr(>|t|)
(Intercept) 11.988755    1.2669  9.463133 0.00000034
X1           0.003747    0.4161  0.009007 0.99295064
summary(lm(Y ~ X1, data=hamilton))$r.squared
[1] 0.00000624
```

```
summary(lm(Y ~ X2, data=hamilton))$coef
      Estimate Std. Error t value Pr(>|t|)
(Intercept) 10.6319    0.8109 13.111 0.000000007178
X2           0.1955    0.1125  1.737 0.105959255741
summary(lm(Y ~ X2, data=hamilton))$r.squared
[1] 0.1884
```

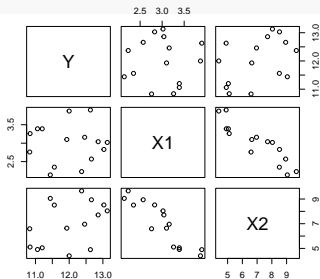
The multiple R^2 for the two models are 0 and 0.1884 only.

Weird Things of Hamilton's Data (2)

X1 and X2 become highly significant when both included, w/ a multiple $R^2 \approx 1$.

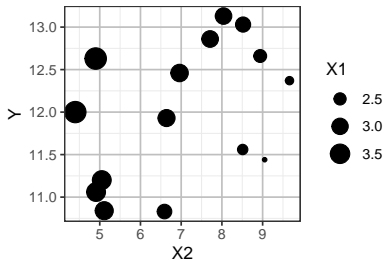
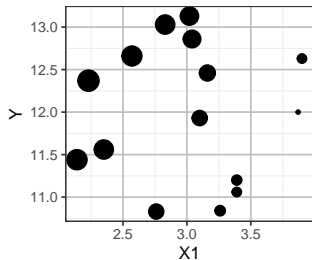
```
summary(lm(Y ~ X1+X2, data=hamilton))$coef
      Estimate Std. Error t value Pr(>|t|)
(Intercept)  -4.515    0.061142  -73.85 2.528e-17
X1             3.097    0.012274  252.31 1.011e-23
X2             1.032    0.003684  280.08 2.888e-24
summary(lm(Y ~ X1+X2, data=hamilton))$r.squared
[1] 0.9998
```

The pairwise scatterplots cannot tell us why



ggplots Can Show X2 Effect on Y After Accounting For X1

```
ggplot(hamilton, aes(x = X1, y = Y, size=X2)) + geom_point()  
ggplot(hamilton, aes(x = X2, y = Y, size=X1)) + geom_point()
```

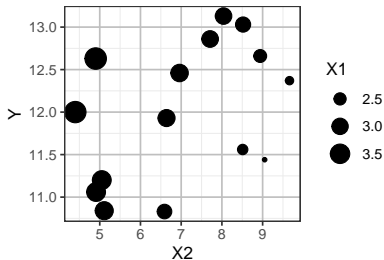
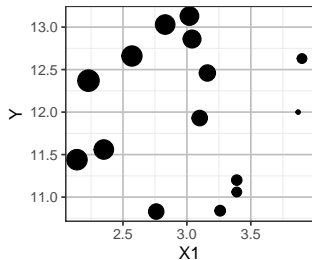


For points with similar X1, points with larger size (higher values of X2) have larger Y values. Hence we can see X2 has an effect on Y after accounting for X1.

Recall β_2 is the effect of X2 on Y when X1 is hold constant.

ggplots Can Show X2 Effect on Y After Accounting For X1

```
ggplot(hamilton, aes(x = X1, y = Y, size=X2)) + geom_point()  
ggplot(hamilton, aes(x = X2, y = Y, size=X1)) + geom_point()
```

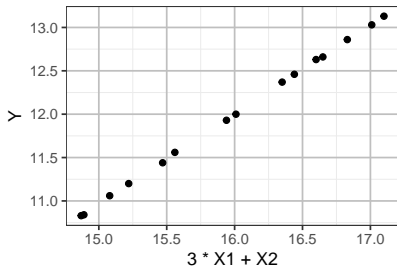


For points with similar X1, points with larger size (higher values of X2) have larger Y values. Hence we can see X2 has an effect on Y after accounting for X1.

Recall β_2 is the effect of X2 on Y when X1 is hold constant.

Can you tell whether X1 has a positive or negative effect on Y after

```
ggplot(hamilton, aes(x = 3*X1+X2, y = Y)) + geom_point()
```



In fact, Y and $3 * X1 + X2$ are highly correlated.

Checking Interactions of Two Numerical Variables

The model

$$Y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \varepsilon$$

assumes that

- Y is linear in X_1 for each given X_2 and the slope of X_1 doesn't change w/ X_2
- Y is linear in X_2 for each given X_1 and the slope of X_2 doesn't change w/ X_1

Example: The Trees Data

Recall the `trees` data are measurements of the diameter, height and volume of timber in 31 felled black cherry trees. The variables are

- `Girth`: Tree diameter (rather than girth, actually) in inches measured at 4 ft 6 in above the ground
- `Height`: Height in ft
- `Volume`: Volume of timber in cubic ft

The `trees` data are build-in in R. One can load the the data by the command

```
data("trees")
```

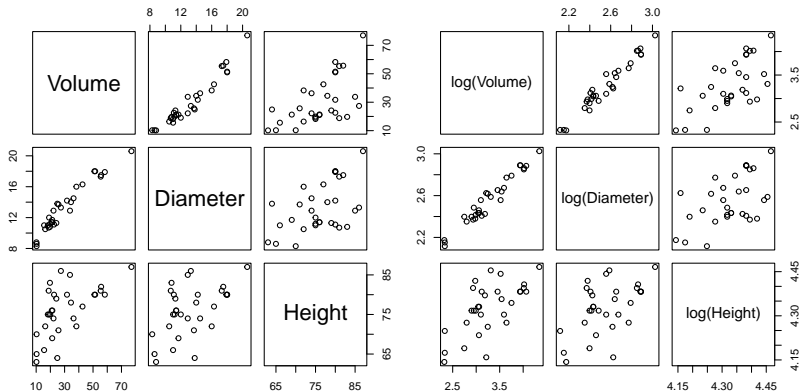
Let's rename the misleading `Girth` variable as `Diameter`

```
trees$Diameter = trees$Girth
```

```

pairs(Volume ~ Diameter + Height, data=trees, oma = c(2,2,2,2))
pairs(log(Volume) ~ log(Diameter) + log(Height), data=trees, oma = c(2,

```



From the pairwise scatter plots above, the two models below don't seem to differ too much.

$$V = \beta_0 + \beta_1 D + \beta_2 H + \varepsilon$$

$$\log(V) = \beta_0 + \beta_1 \log(D) + \beta_2 \log(H) + \varepsilon$$

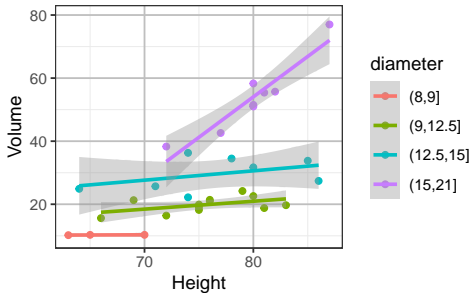
The model $V = \beta_0 + \beta_1 D + \beta_2 H + \varepsilon$ implies the slope of H stay the same for each level of D.

```
trees$diameter = cut(trees$Diameter, breaks=c(8,9,12.5,15,21))  
ggplot(trees, aes(x=Height, y=Volume, color=diameter)) +  
  geom_point() + geom_smooth(method='lm', formula='y~x')
```

Observe the slopes of **Height** increases as **Diameter** increases, which means the model

$$V = \beta_0 + \beta_1 D + \beta_2 H + \varepsilon$$

isn't appropriate.

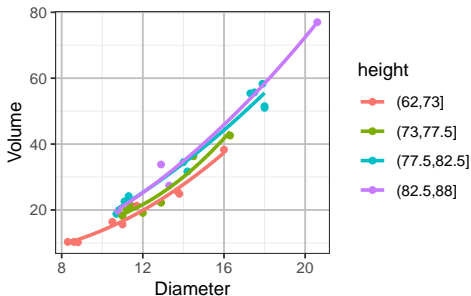


```
trees$height = cut(trees$Height, breaks=c(62,73,77.5,82.5,88))
ggplot(trees, aes(x=Diameter, y=Volume, color=height)) +
  geom_point() +
  geom_smooth(method='lm', formula='y~x+I(x^2)', se=F)
```

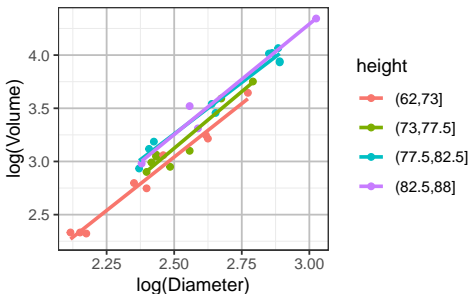
Observe the slopes of Diameter also increases as Height increases, which means the model

$$V = \beta_0 + \beta_1 D + \beta_2 H + \varepsilon$$

isn't appropriate.



```
ggplot(trees, aes(x=log(Diameter), y=log(Volume), color=height)) +  
  geom_point() + geom_smooth(method='lm', formula='y~x', se=F)
```

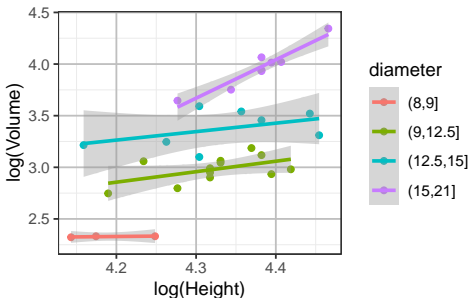


The model

$$\log(V) = \beta_0 + \beta_1 \log(D) + \beta_2 \log(H) + \varepsilon$$

is more appropriate since the slopes of $\log(\text{Diameter})$ on $\log(\text{Volume})$ changes little with Height.

```
ggplot(trees, aes(x=log(Height), y=log(Volume), color=diameter)) +  
  geom_point() + geom_smooth(method='lm', formula='y~x')
```



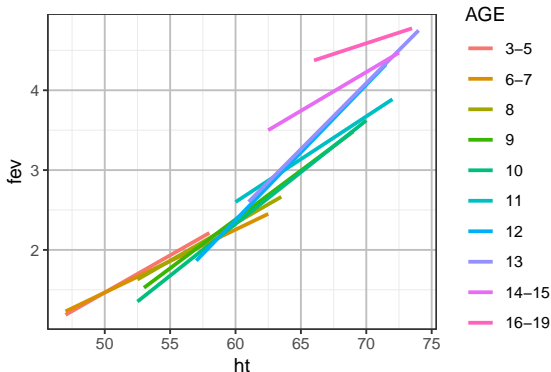
The model

$$\log(V) = \beta_0 + \beta_1 \log(D) + \beta_2 \log(H) + \varepsilon$$

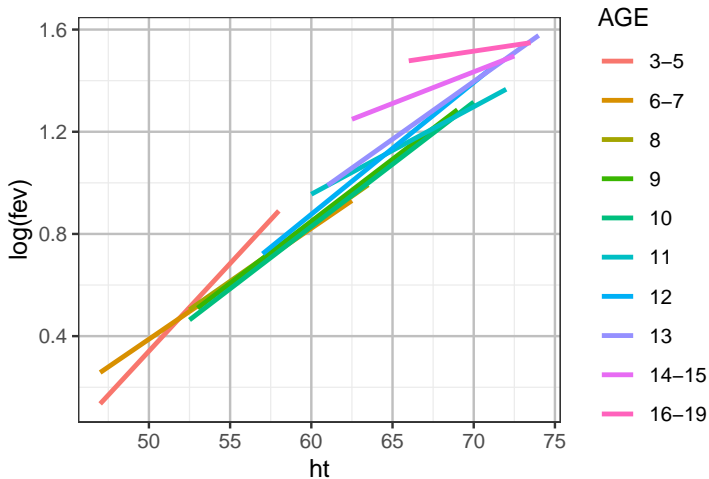
is more appropriate since the slopes of $\log(\text{Height})$ don't change with Diameter as much as before the log-transformation

Example: Lung Capacity Data

```
m.nonsmoker = subset(fevdata, sex=="Male" & smoke=="Nonsmoker")
m.nonsmoker$AGE = cut(m.nonsmoker$age, breaks = c(3,6,8:14,16,20)-0.5)
m.nonsmoker$AGE = factor(m.nonsmoker$AGE,
                          labels=c("3-5", "6-7", as.character(8:13), "14-15", "16-19"))
ggplot(m.nonsmoker, aes(x=ht, y=fev, color=AGE)) +
  geom_smooth(method='lm', formula='y~x', se=F)
```



```
ggplot(m.nonsmoker, aes(x=ht, y=log(fev), color=AGE)) +  
  geom_smooth(method='lm', formula='y~x', se=F)
```



Observe $\text{age} \times \text{ht}$ is highly significant if using `fev` as the response.

```
summary(lm(fev ~ age+ht+age*ht, data=m.nonsmoker))$coef
```

	Estimate	Std. Error	t value	Pr(> t)
(Intercept)	0.367490	0.749808	0.4901	6.244e-01
age	-0.494626	0.084997	-5.8193	1.490e-08
ht	0.026012	0.012766	2.0376	4.245e-02
age:ht	0.009103	0.001292	7.0456	1.226e-11

Observe $\text{age} \times \text{ht}$ become insignificant if `fev` is log-transformed.

```
summary(lm(log(fev) ~ age+ht+age*ht, data=m.nonsmoker))$coef
```

	Estimate	Std. Error	t value	Pr(> t)
(Intercept)	-1.5660268	0.2625189	-5.9654	6.735e-09
age	-0.0156762	0.0297587	-0.5268	5.987e-01
ht	0.0371132	0.0044696	8.3034	3.300e-15
age:ht	0.0006201	0.0004524	1.3709	1.714e-01

Residual Plus Component Plot

Residual Plus Component Plot

A **Residual Plus Component Plot** is a scatterplot of

$$(e + \hat{\beta}_j X_j) \text{ versus } X_j,$$

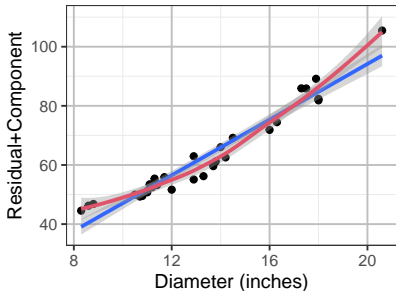
where e and $\hat{\beta}_j$ are from the regression of Y on all predictors, including X_j .

- The slope of the graph is $\hat{\beta}_j$.
- This plot is useful to detect **non-linearity** in the partial relationship between Y and X_j .

Residual Plus Component Plot (Trees Data)

For the model `lmtrees1 = lm(Volume ~ Diameter + Height, data=trees)`, the Residual Plus Component plot below for Diameter shows clear *nonlinearity*.

```
ggplot(trees, aes(x=Diameter, y=lmtrees1$res+lmtrees1$coef[2]*Diameter))
  geom_point() + geom_smooth(method='lm') +
  geom_smooth(method='loess', col=2) +
  labs(x="Diameter (inches)", y="Residual+Component")
```

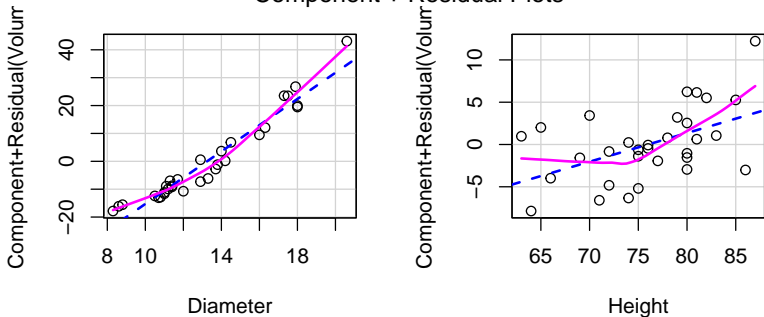


crPlots() in the car Library

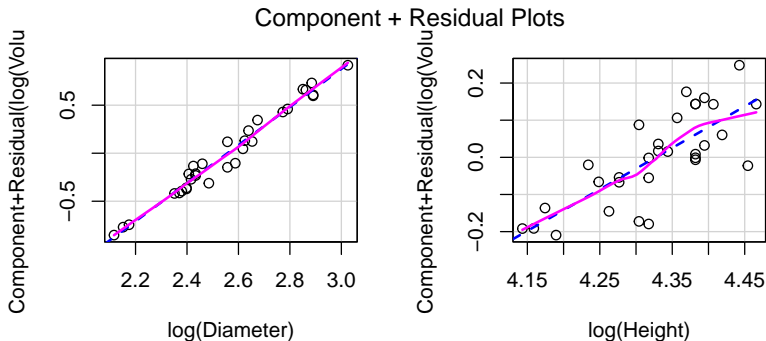
The `crPlots()` function in the `car` library can produce residual plus component plots automatically.

```
library(car)  
crPlots(lmtrees1)
```

Component + Residual Plots



```
lmtree2 = lm(log(Volume) ~ log(Diameter) + log(Height), data=trees)
crPlots(lmtree2)
```



Both plots look linear, meaning the model

$\log(\text{Volume}) = \beta_0 + \beta_1 \log(\text{Diameter}) + \beta_2 \log(\text{Height}) + \varepsilon$ is appropriate.

Residual Plus Component Plot (FEV data)

If we fit the model below,

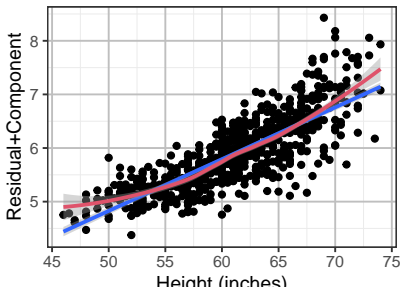
```
lm2 = lm(fev ~ age*smoke + age*sex + ht, data=fevdata)
```

the residual plus component plot for ht is

```
ggplot(fevdata, aes(x=ht, y=lm2$res+lm2$coef[5]*ht)) +  
  geom_point() + geom_smooth(method='lm') +  
  geom_smooth(method='loess', col=2) +  
  labs(x="Height (inches)", y="Residual+Component")
```

We can see

- nonlinearity
- variability not constant



However, `crPlots()` doesn't work if the model includes interactions.

```
library(car)
```

```
#crPlots(lm2, "ht") # not work since lm2 includes interactions
```

```
crPlots(lm(fev ~ age + smoke + ht, data=fevdata), "ht")
```

