GETTING STARTED

Install the `car`, `heplots`, `lmtest`, and `sandwich` packages in `R`

Create a new project called `week-14-lecture` in ShareLaTeX

Download the image file `sushi.png` from the course website and save it on your desktop.

CHRISTOPHER PRENER, PH.D.
FALL, 2017

WEEK 14
LECTURE 15

QUANTITATIVE ANALYSIS

MULTIPLE REGRESSION (2)

AGENDA

1. Front Matter
2. Images with LaTeX
3. Regression Diagnostics
4. Adjusting Models
5. Back Matter

1 FRONT MATTER

ANNOUNCEMENTS

PS-10 is due next week along with Lab-14.

This week is the last week of content that is needed for the final project!

2 IMAGES WITH LATEX

USING THE GRAPHICX PACKAGE


```
\usepackage{graphicx}
```

Include in your preamble.

SETTING PATH TO IMAGES


```
\graphicspath{{imagesDir/}}
```

Parameters:

- *imagesDir* should be the name of a subdirectory inside your project where all images are stored.

SETTING PATH TO IMAGES


```
\graphicspath{{imagesDir/}}
```

I use a subdirectory named images for all of my projects:

```
\graphicspath{{images/}}
```

The double braces are required! Include in your preamble after loading the `graphicx` package.

INCLUDING IMAGES

```
\begin{figure}[!h]
\includegraphics[scale= val]{“imageFile”}
\end{figure}
```

Include an image named “sushi.png” at half scale:

```
\begin{figure}[!h]
\includegraphics[scale= .5]{“sushi”}
\end{figure}
```

Scale values require experimentation. A caption can also be used with each image using `\caption{}`. Include it after `\begin{figure}`.

EXERCISE

Week 14 Exercise - Images in LaTeX

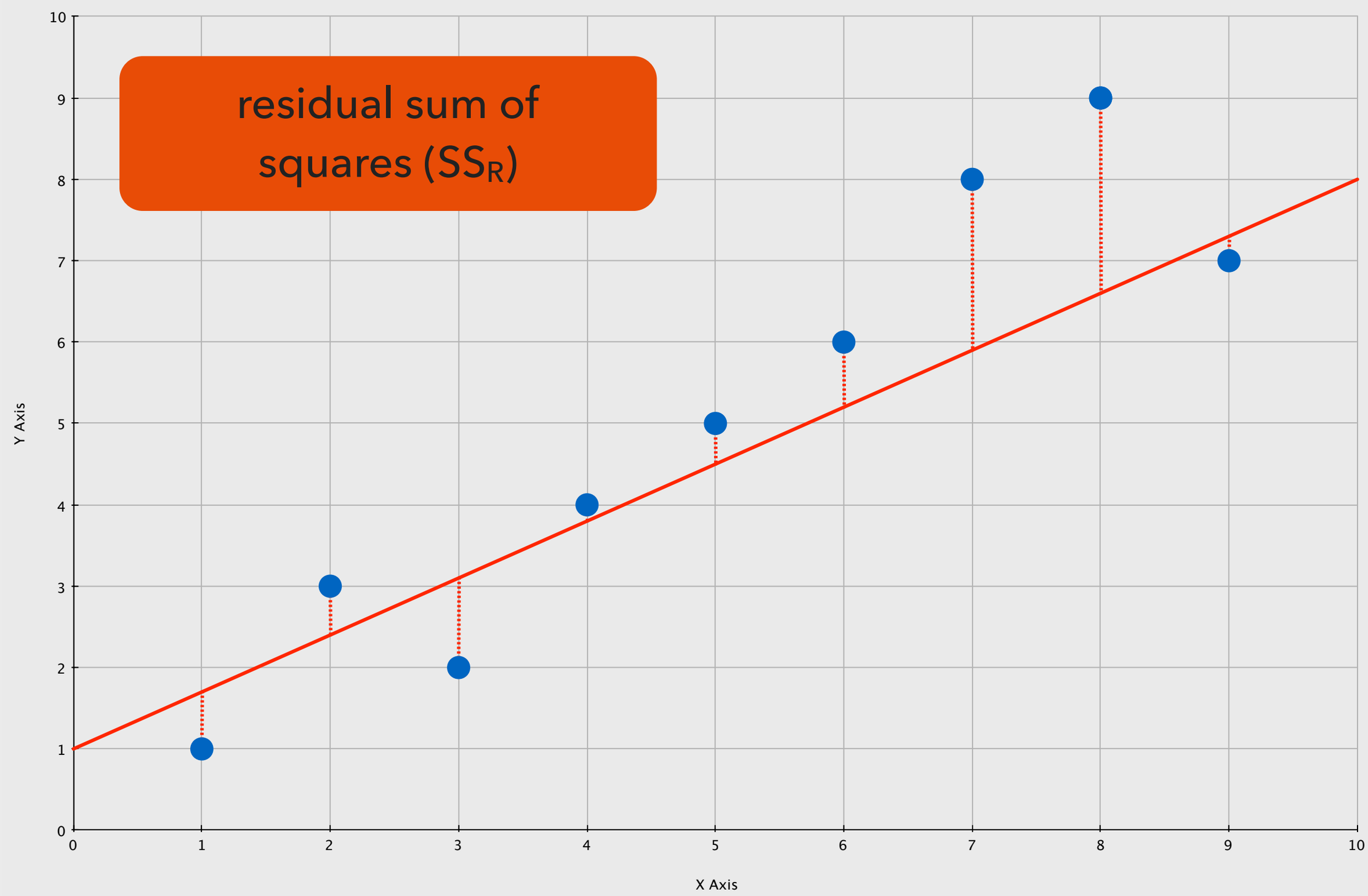
Christopher Prener, Ph.D.

November 27, 2017

3 REGRESSION DIAGNOSTICS

THE GOAL OF OLS REGRESSION

residual sum of squares (SS_R)



IN OTHER WORDS...

We want to explain as much of the variation in y as we can while also minimizing the residual error in the regression line.

BASIC ASSUMPTIONS

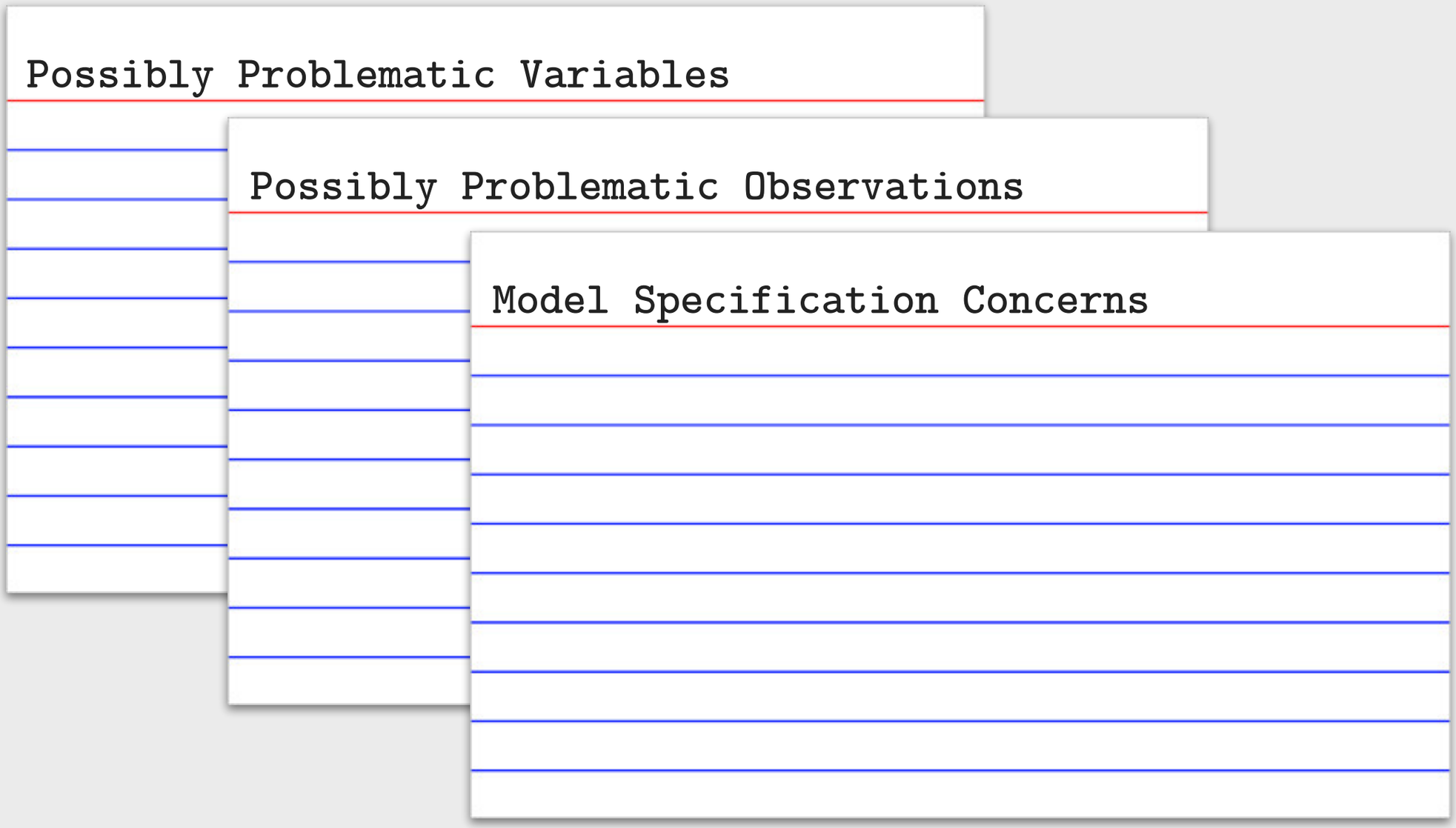
- y must be continuous*
- x can be:
  - binary
  - ordinal
  - continuous
- x variables must have a variance > 0
- Relationships between x variables and y are linear
- y should be normally distributed
- There should be no significant outliers in x and y

MODEL SET-UP

```
> library(ggplot2)

> library(car)

> autoData <- mpg

> model <- lm(hwy ~ displ+cyl, data = autoData)
```

DOCUMENTATION SET-UP

Keep a running list of the variables, observations, and model issues that may be problematic. Summarize these in your notebook at the end of the analysis.



PRINTING ROW NAMES

f(x) `which(`x`)`

Parameters:

- x is a

Available in `base`
Included in standard distributions of `R`

PRINTING ROW NAMES

`which(`x`)`

Parameters:

- x is an object and an expression paired together

PRINTING ROW NAMES

`which(`x`)`

Using the `hwy` variable from `ggplot2`'s `mpg` data:

```
> highmpg <- which(mpg$hwy > 40)
> highmpg
[1] 213 222 223
```

Row numbers are based on the sort order of your data and will change if you re-sort or subset them!

MATCHING VALUES

f(x) `%in%`

Parameters:

- x is a

Available in `base`
Included in standard distributions of `R`

MATCHING VALUES

`%in%`

Using `ggplot2`'s `mpg` data and the `highmpg` list created previously:

```
> filter(mpg, row_number() %in% highmpg)
```

<<<<< OUTPUT OMITTED >>>>>

The `row_number()` function is part of `dplyr`.

CREATING ID NUMBERS

f(x) `rowid_to_column(`*varName*`)`

Parameters:

- *varI* ...nber
  valu...

Available in `tibble`
Download via CRAN, part of `tidyverse`

CREATING ID NUMBERS

f(x) `rowid_to_column(`*varName*`)`

Parameters:

- *varName* is the name of new variable that will contain the row number values.

CREATING ID NUMBERS

f(x) `rowid_to_column(`*varName*`)`

Add a variable named `id` that contains row numbers:

```
> rowid_to_column(“id”)
```

Row numbers are based on the sort order of your data and will change if you re-sort or subset them!

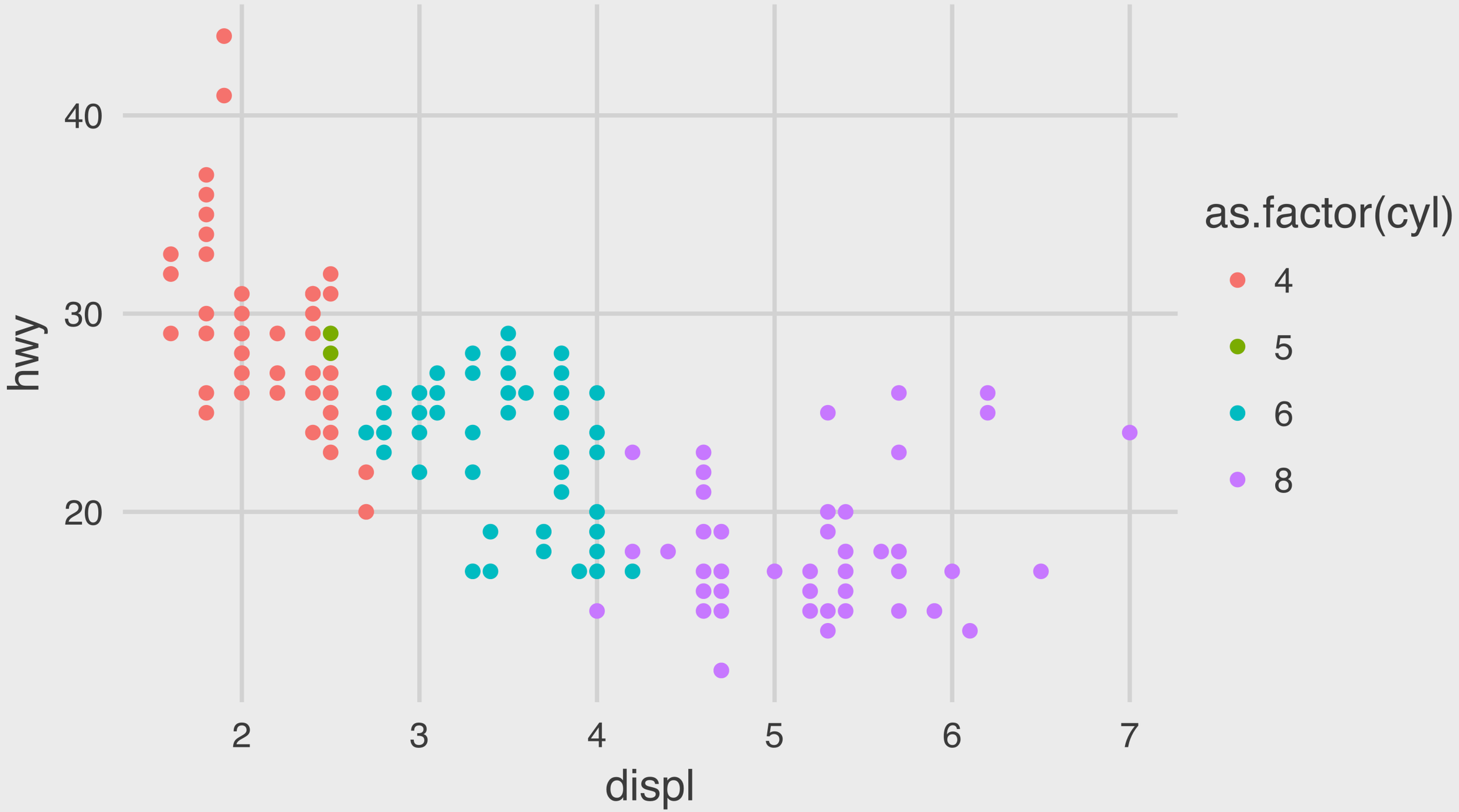
CREATING ID NUMBERS

```
> autoData %>%
    rowid_to_column("id") %>%
    select(id, everything()) -> autoData
```

```
> range(autoData$id)
[1]   1 234

> nrow(autoData)
[1] 234
```

a NON-
LINEARITYY

“MULTI-VARIATE” PLOTS



COMPONENT RESIDUAL PLOTS

f(x) `crPlots(model)`

Parameters:

- *model*
  func

Available in `car`
Download via CRAN

COMPONENT RESIDUAL PLOTS

`crPlots(model)`

Parameters:

- *`model`* is the model object created with the output from the `lm()` function

COMPONENT RESIDUAL PLOTS

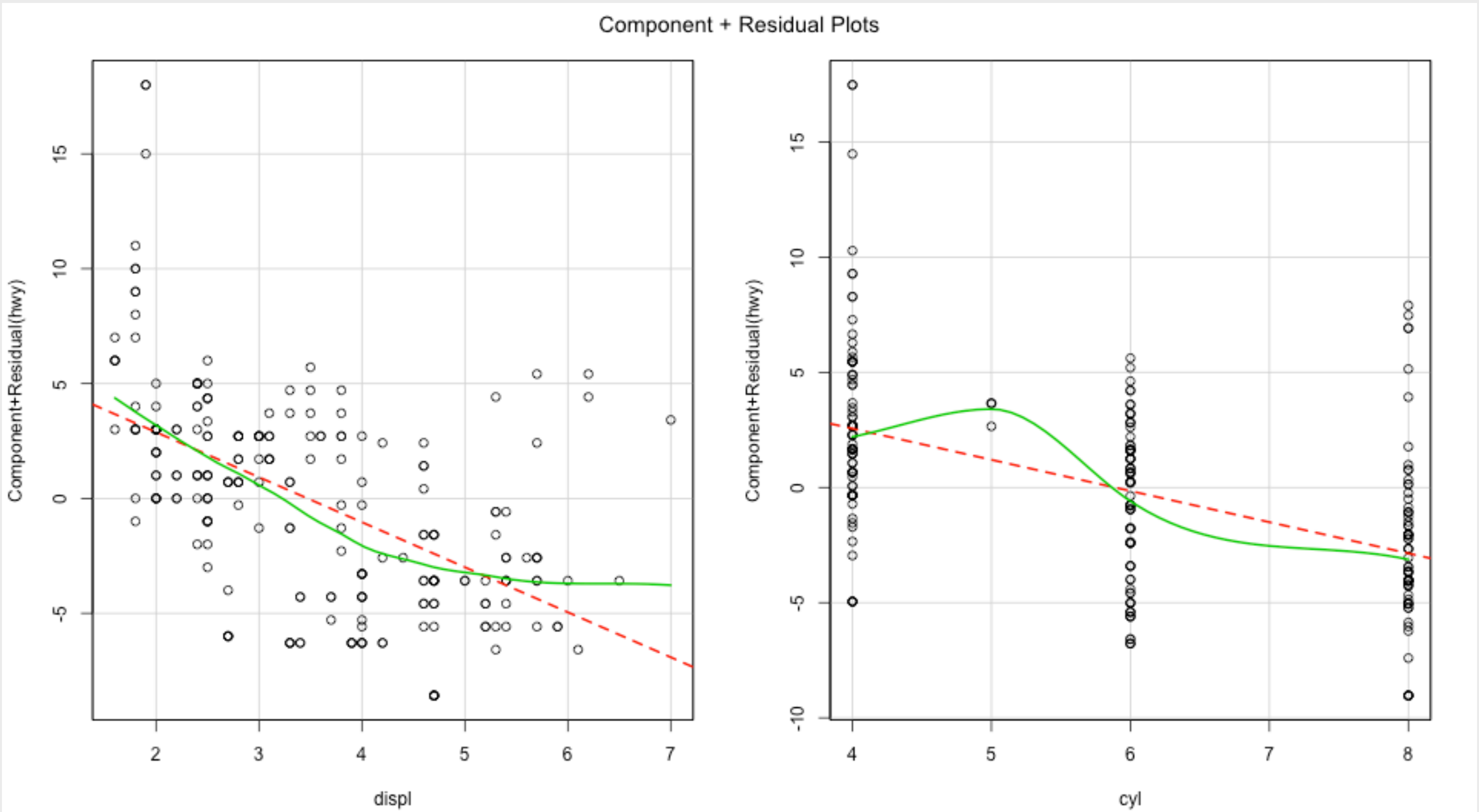
f(x)

`crPlots(model)`

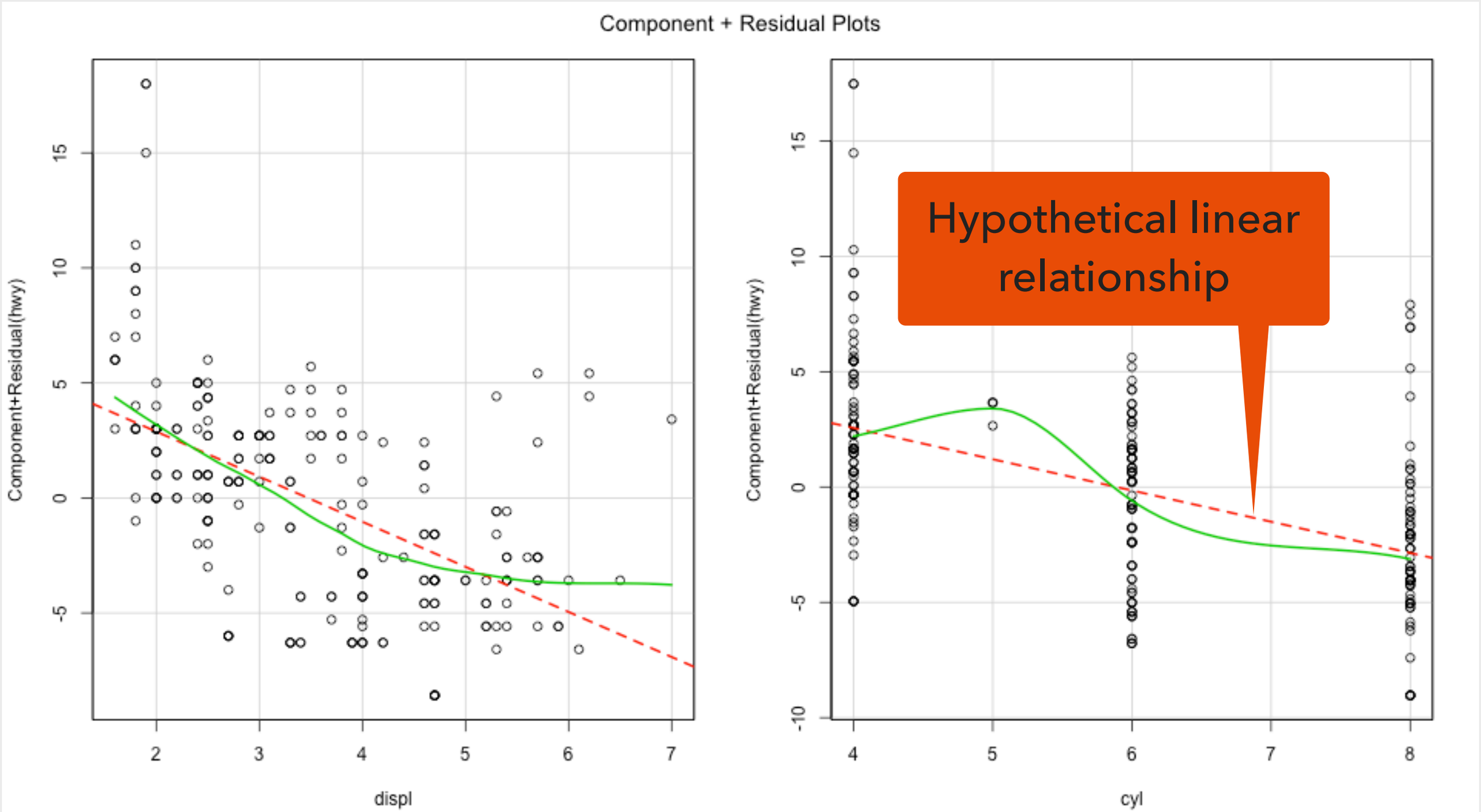
Using the lecture model from `ggplot2`'s `mpg` data:

```
> crPlots(model)
```

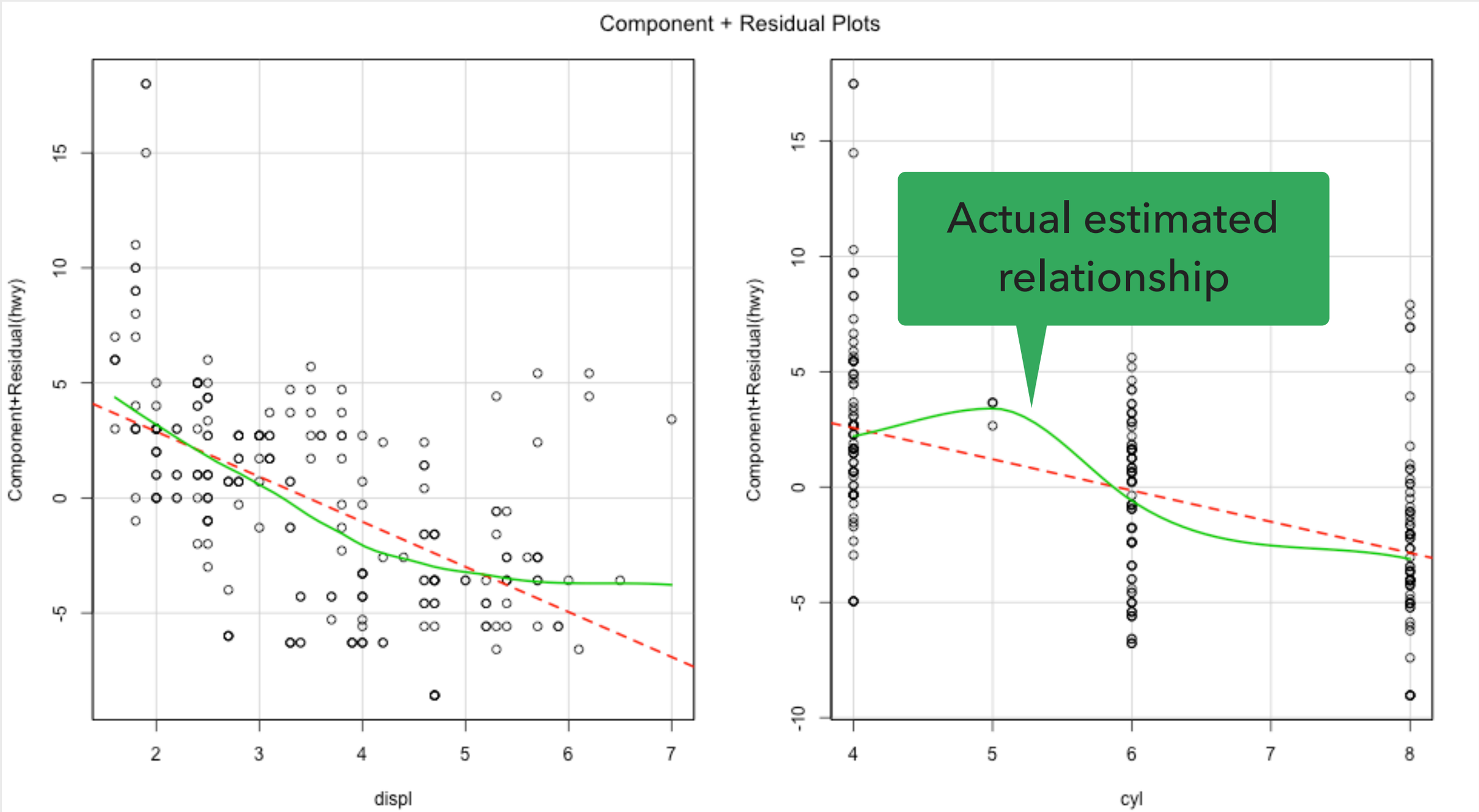
COMPONENT RESIDUAL PLOTS



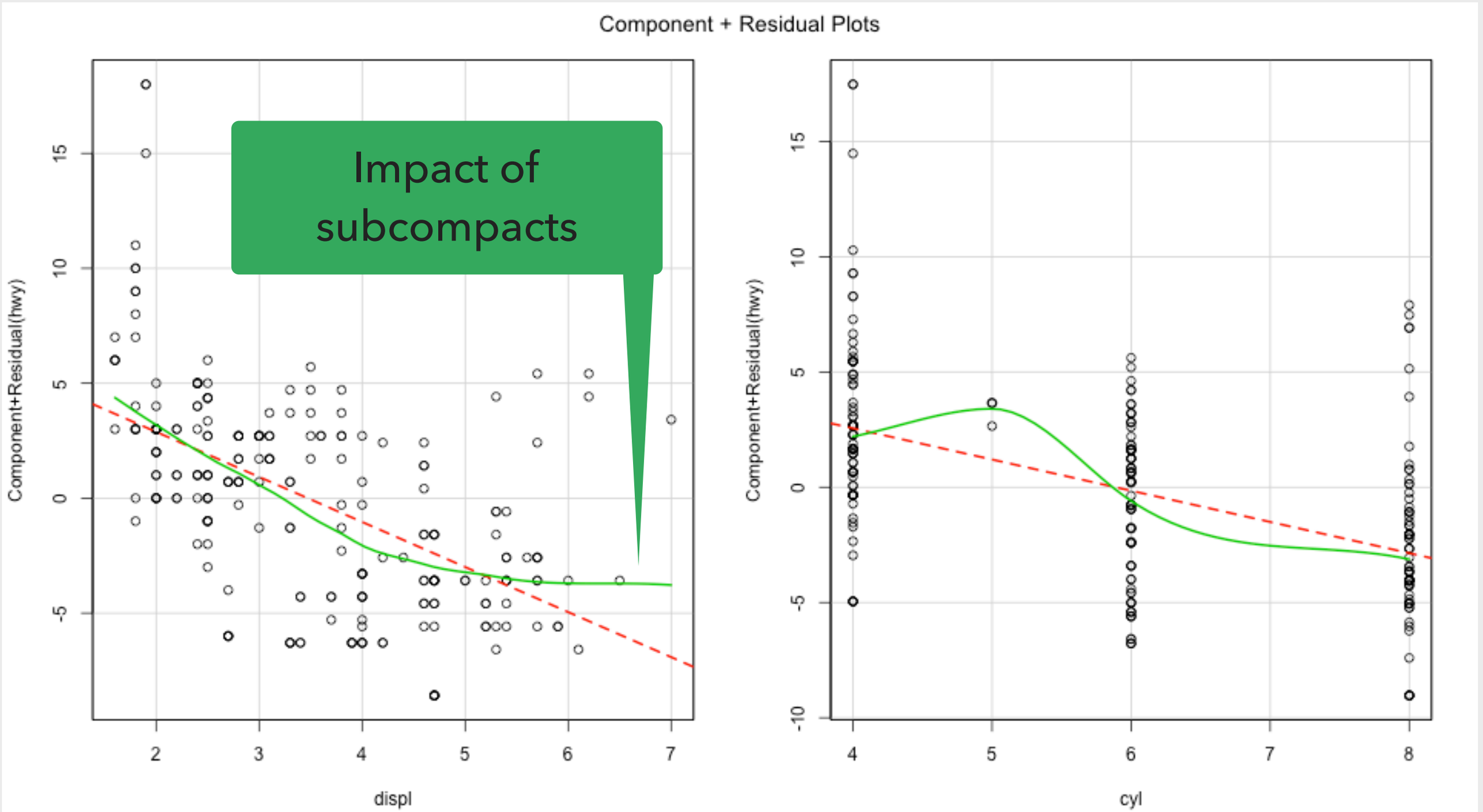
COMPONENT RESIDUAL PLOTS



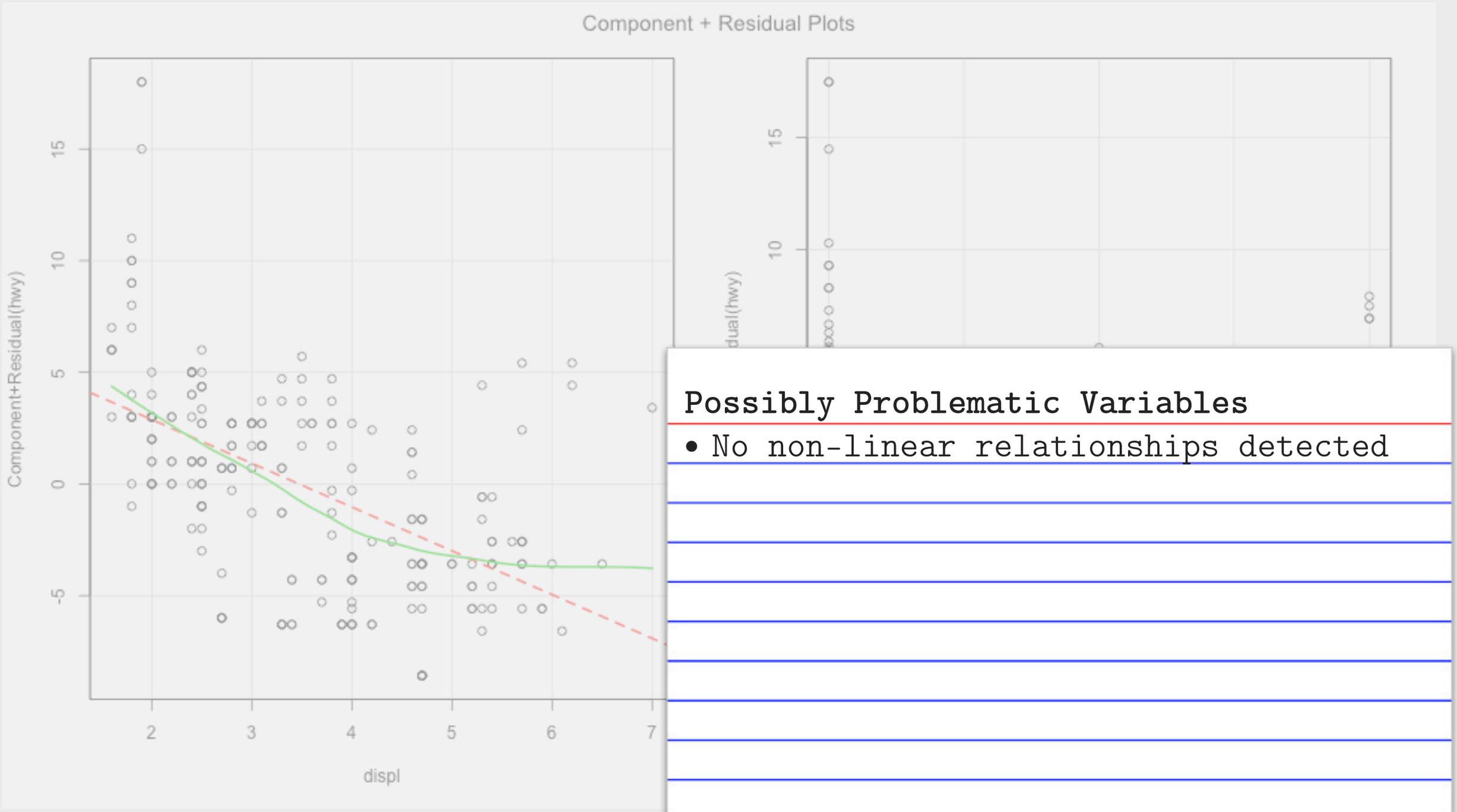
COMPONENT RESIDUAL PLOTS



COMPONENT RESIDUAL PLOTS



COMPONENT RESIDUAL PLOTS



Possibly Problematic Variables

- No non-linear relationships detected

SINGLE PLOTS

f(x) `crPlot(`*`model`*`, variable="`*`varName`*`")`

Parameters:

- *`model`* is the model object created with the output from the `lm()` function
- *`varName`* is the variable you want to focus on

SINGLE PLOTS

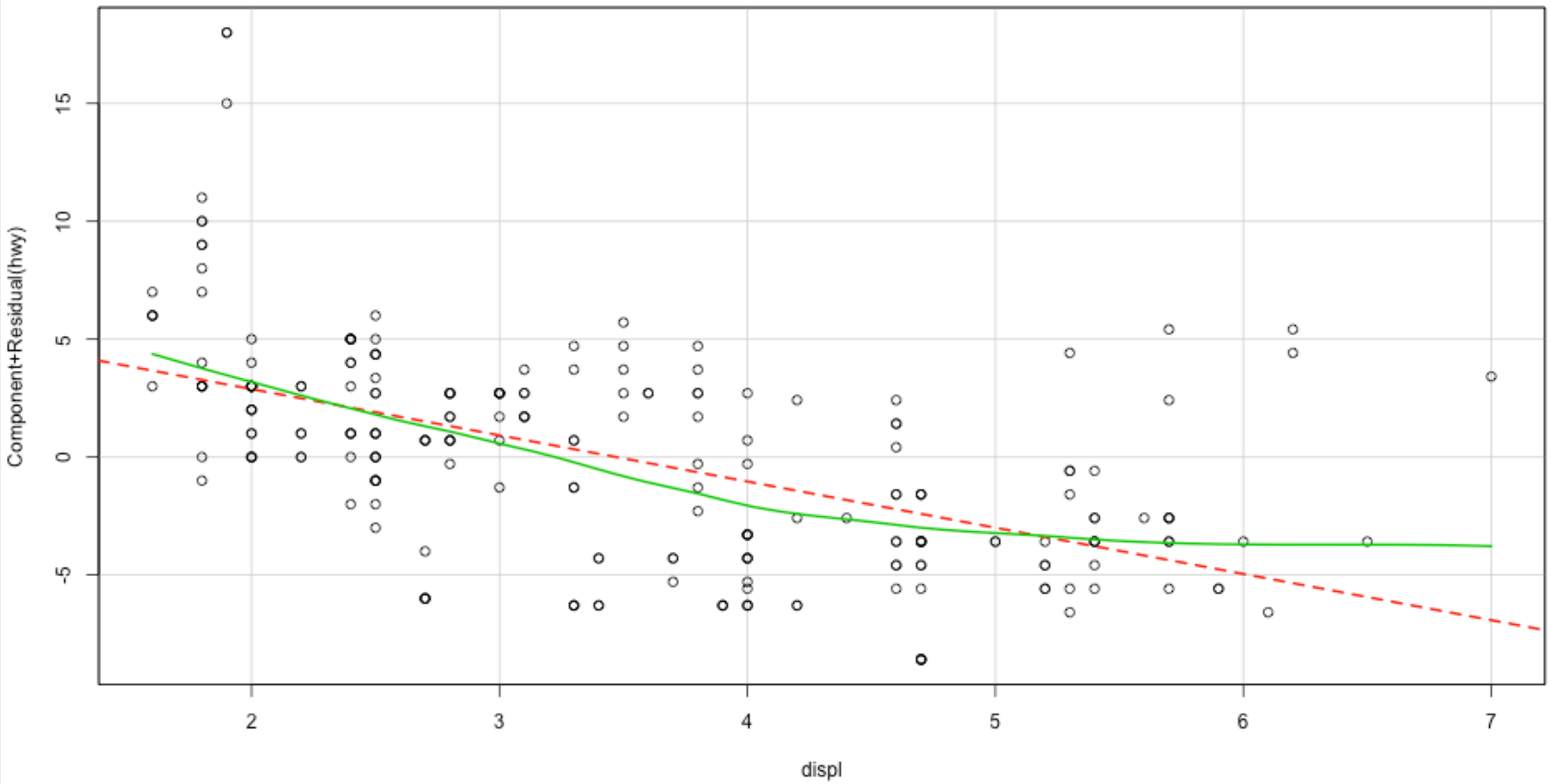
f(x)

```
crPlot(model, variable="varName")
```

Using the lecture model from `ggplot2`'s `mpg` data:

```
> crPlot(model, variable="displ")
```

SINGLE PLOTS



OUTLIER DETECTION

BONFERONNI OUTLIER TEST

f(x) `outlierTest(`*model*`)`

Parameters:

- *mode*
  func

Available in `car`
Download via CRAN

BONFERONNI OUTLIER TEST

f(x) `outlierTest(`*`model`*`)`

Parameters:

- *`model`* is the model object created with the output from the `lm()` function

BONFERONNI OUTLIER TEST

f(x) `outlierTest(model)`

Using the lecture model from `ggplot2`'s `mpg` data:

```
> outlierTest(model)
     rstudent unadjusted p-value Bonferonni p
213 4.127028         5.1401e-05     0.012028
222 4.127028         5.1401e-05     0.012028
```

Will print the row numbers of observations identified as outliers.

BONFERONNI OUTLIER TEST

```
> filter(autoData, row_number() %in% c(213, 222))
# A tibble: 2 x 12
     id manufacturer      model displ  year   cyl      trans   drv   cty   hwy    fl       class
  <int>        <chr>      <chr> <dbl> <int> <int>      <chr> <chr> <int> <int> <chr>       <chr>
1   213   volkswagen      jetta   1.9  1999     4 manual(m5)     f    33    44     d     compact
2   222   volkswagen new beetle   1.9  1999     4 manual(m5)     f    35    44     d subcompact
```

BONFERONNI OUTLIER TEST

```
> filter(autoData, row_number() %in% c(213, 222))
# A tibble: 2 x 12
     id manufacturer      model displ  year   cyl      trans   drv   cty   hwy    fl       class
  <int>        <chr>      <chr> <dbl> <int> <int>      <chr> <chr> <int> <int> <chr>       <chr>
1   213   volkswagen      jetta   1.9  1999     4 manual(m5)     f    33    44     d     compact
2   222   volkswagen new beetle   1.9  1999     4 manual(m5)     f    35    44     d subcompact
```

Possibly Problematic Observations

- Outliers - 213, 222

BASIC ASSUMPTIONS

- y must be continuous*
- x can be:
  - binary
  - ordinal
  - continuous
- x variables must have a variance > 0
- Relationships between x variables and y are linear
- y should be normally distributed
- There should be no **unusual observations** in x and y

UNUSUAL OBSERVATIONS

BIG PICTURE

- Unusual observations are those that have greater-than-expected impact on the slope of the regression line.
- Outliers are one type of unusual observation.
- We can also look for greater-than-expected impact using two other approaches:
  - Observations with high *leverage* values
  - Observations with high *influence* values, known as Cook's Distance
- Put differently, if these observations are removed from the model, we would expect the model to change (sometimes dramatically)

LEVERAGE CUTOFFS

Let:

- p = number of parameters including the intercept
- n = sample size

$$\frac{2 * p}{n}$$

$$\frac{3 * p}{n}$$

LEVERAGE POINTS

f(x) `hatvalues(model)`

Parameters:

- *model*
  func

Available in `stats`
Installed with R distributions

LEVERAGE POINTS

`hatvalues(`*`model`*`)`

Parameters:

- *`model`* is the model object created with the output from the `lm()` function

LEVERAGE POINTS

f(x) `hatvalues(model)`

Using the lecture model from `ggplot2`'s `mpg` data:

```
> which(hatvalues(model) > (2*3)/234)
```

<<<< OUTPUT OMITTED >>>>>

Should be combined with the x2 and/or x3 cutoff values calculated using the appropriate equation.

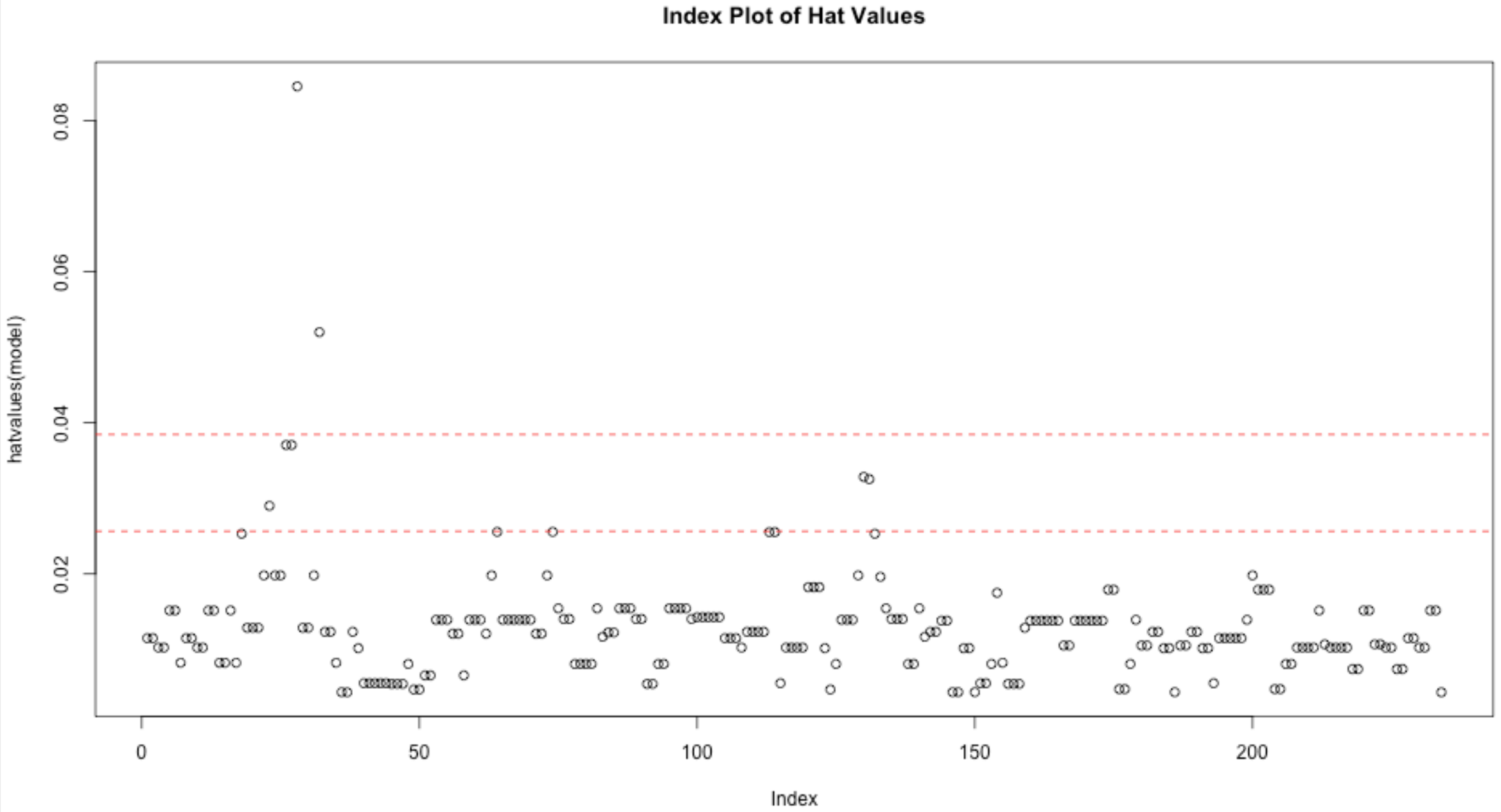
LEVERAGE POINTS

```
> leveragePoints <- which(hatvalues(model) > (2*3)/234)

> filter(autoData, row_number() %in% leveragePoints)
# A tibble: 7 x 12
     id manufacturer                model displ  year   cyl      trans   drv   cty   hwy    fl   class
  <int>        <chr>                <chr> <dbl> <int> <int>      <chr> <chr> <int> <int> <chr>   <chr>
1    23    chevrolet c1500 suburban 2wd   6.0  2008     8   auto(l4)     r    12    17     r     suv
2    26    chevrolet           corvette   6.2  2008     8 manual(m6)     r    16    26     p 2seater
3    27    chevrolet           corvette   6.2  2008     8   auto(s6)     r    15    25     p 2seater
4    28    chevrolet           corvette   7.0  2008     8 manual(m6)     r    15    24     p 2seater
5    32    chevrolet    k1500 tahoe 4wd   6.5  1999     8   auto(l4)     4    14    17     d     suv
6   130         jeep grand cherokee 4wd   6.1  2008     8   auto(l5)     4    11    14     p     suv
7   131   land rover        range rover   4.0  1999     8   auto(l4)     4    11    15     p     suv
```

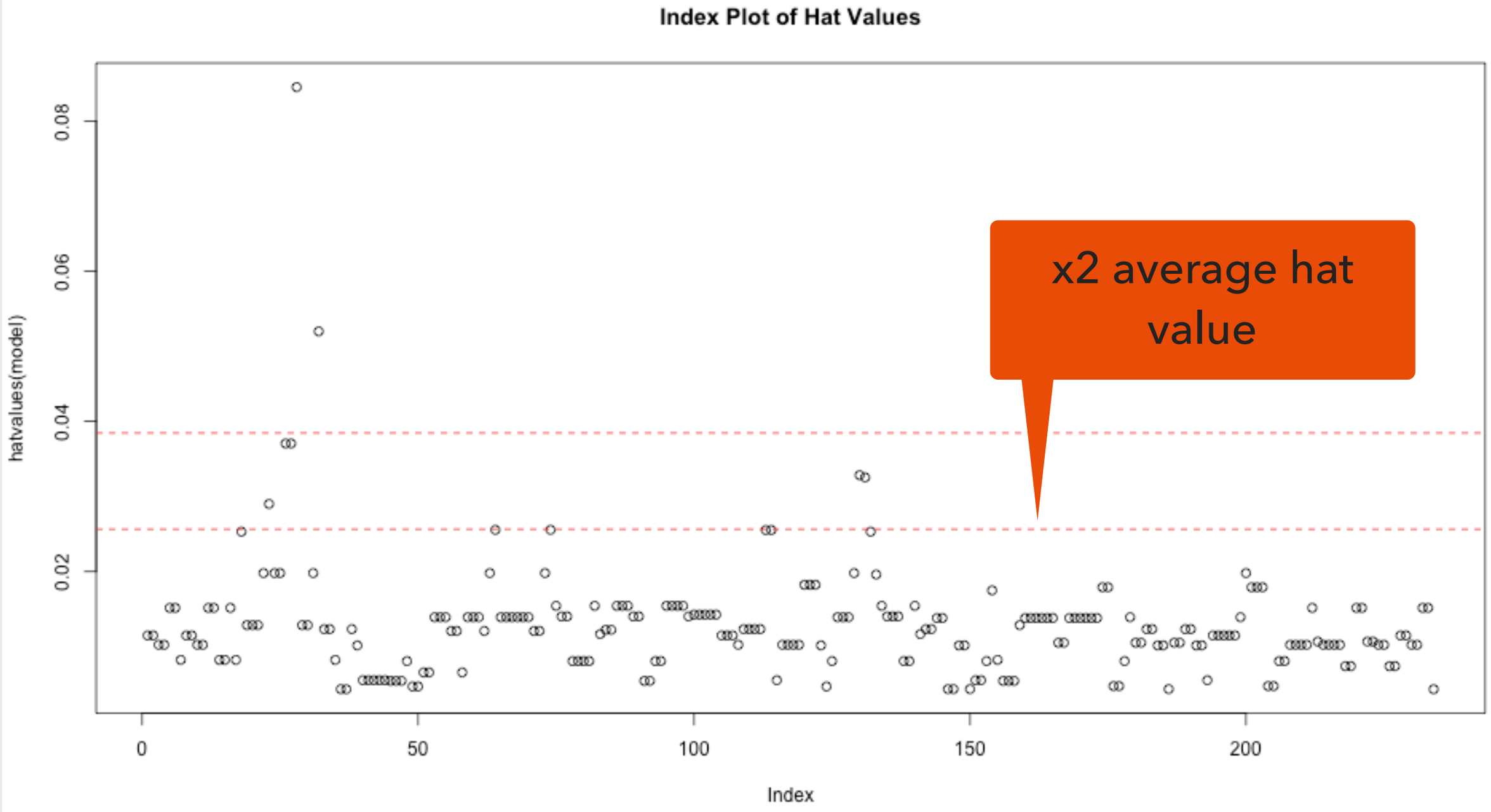
LEVERAGE POINTS

```
> leveragePoints <- which(hatvalues(model) > (2*3)/234)

> filter(autoData, row_number() %in% leveragePoints)
# A tibble: 7 x 12
     id manufacturer                model displ  year   cyl      trans   drv   cty   hwy    fl   class
  <int>        <chr>                <chr> <dbl> <int> <int>      <chr> <chr> <int> <int> <chr>   <chr>
1    23    chevrolet c1500 suburban 2wd   6.0  2008     8   auto(l4)     r    12    17     r     suv
2    26    chevrolet           corvette   6.2  2008     8 manual(m6)     r    16    26     p 2seater
3    27    chevrolet           corvette   6.2  2008     8   auto(s6)     r    15    25     p 2seater
4    28    chevrolet           corvette   7.0  2008     8 manual(m6)     r    15    24     p 2seater
5    32    chevrolet    k1500 tahoe 4wd   6.5  1999     8   auto(l4)     4    14    17     d     suv
6   130         jeep grand cherokee 4wd   6.1  2008     8   auto(l5)     4    11    14     p     suv
7   131   land rover        range rover   4.0  1999     8   auto(l4)     4    11    15     p     suv

> plot(hatvalues(model))

> abline(h = c(2,3)*3/234, col="red", lty=2)
```

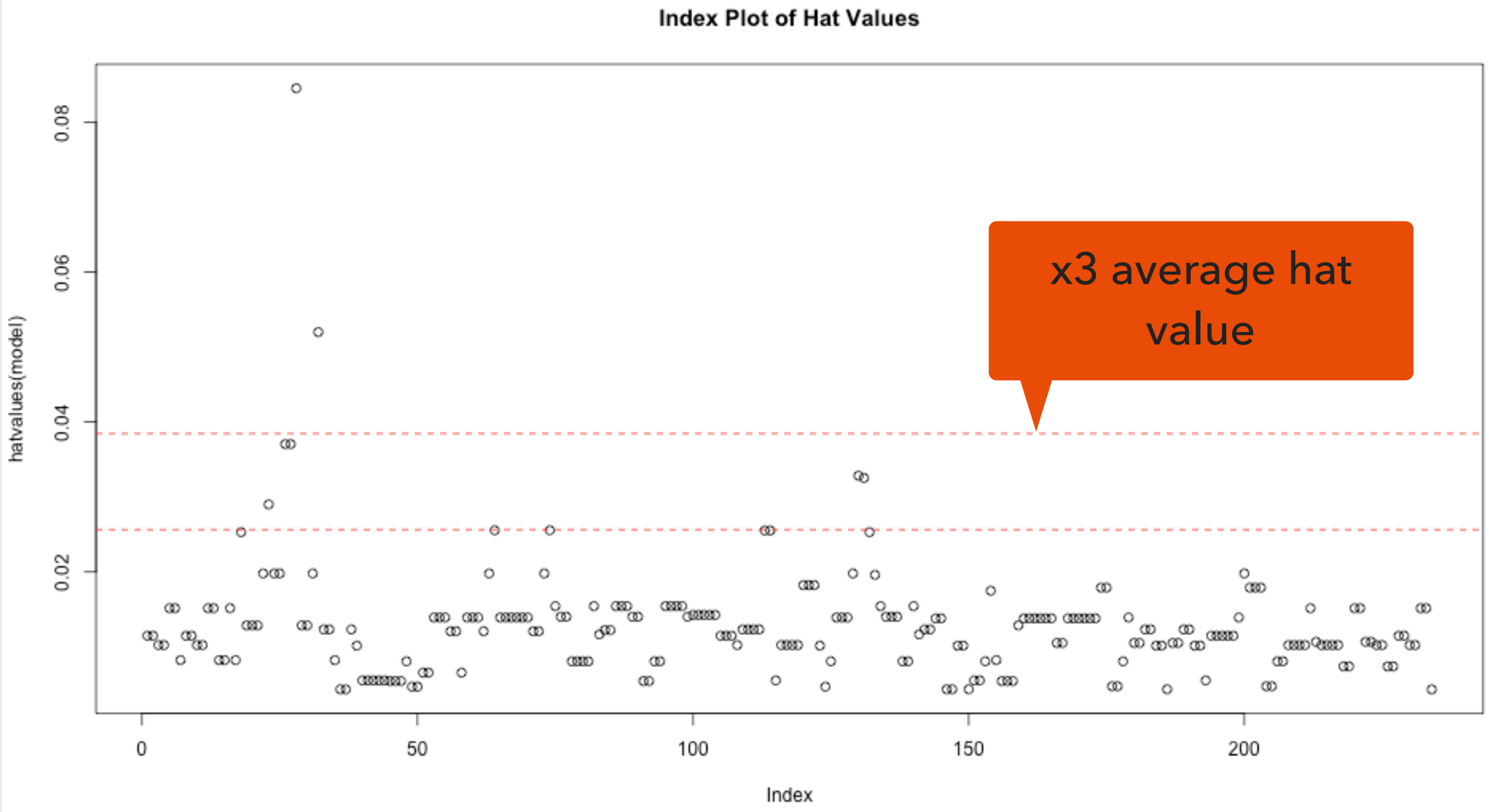
LEVERAGE POINTS



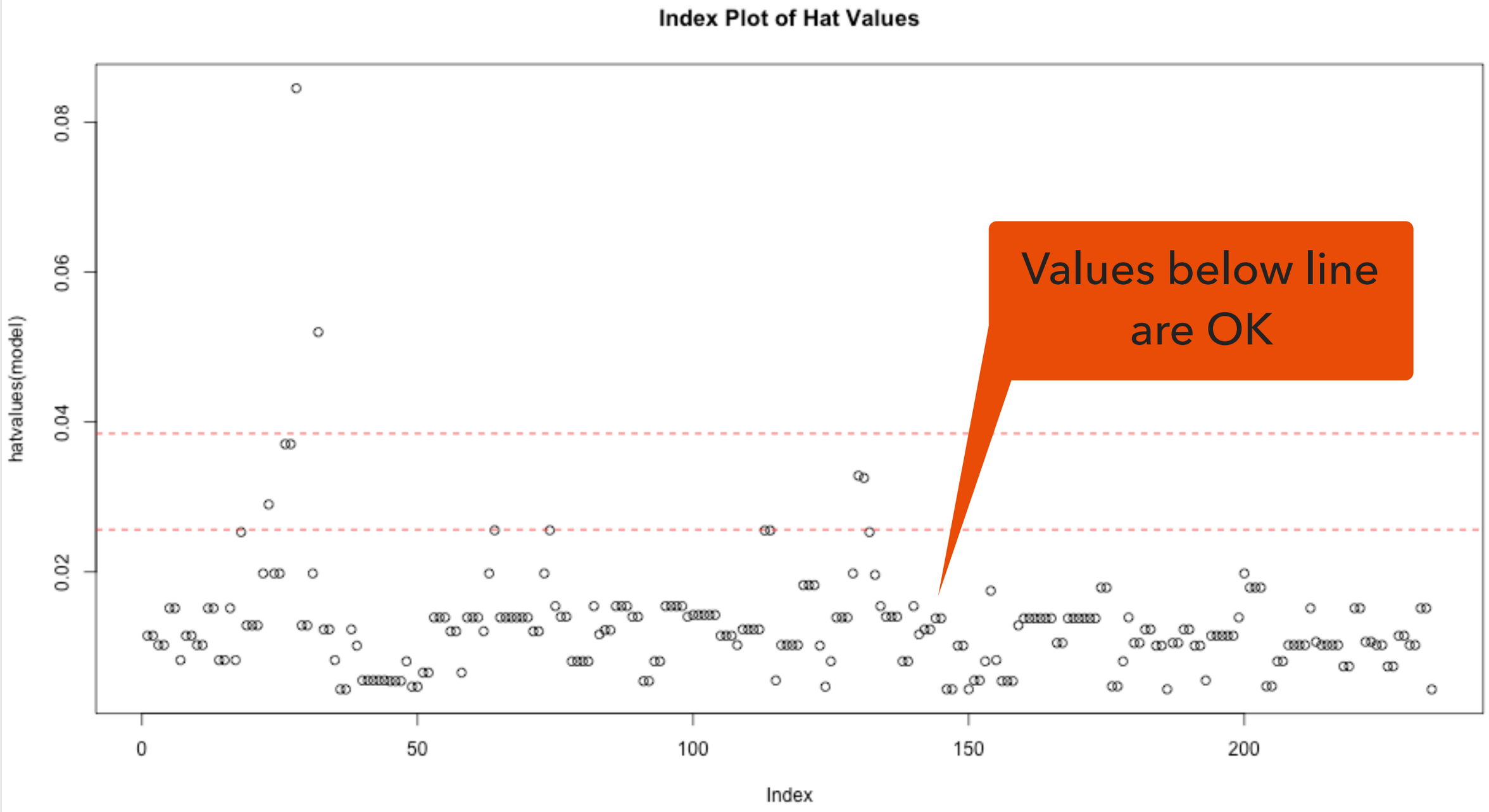
LEVERAGE POINTS



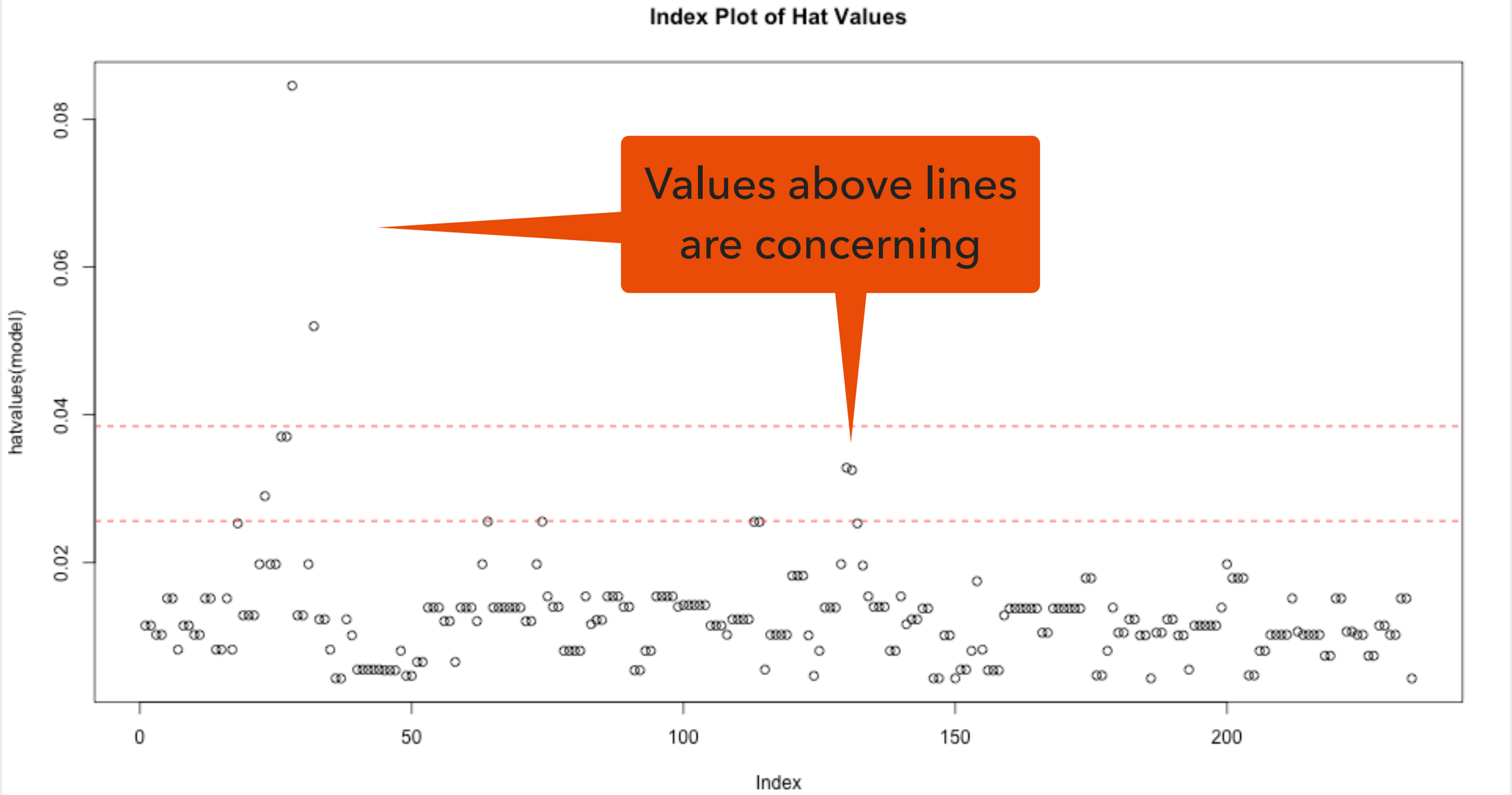
LEVERAGE POINTS



LEVERAGE POINTS



LEVERAGE POINTS



LEVERAGE POINTS

```
> leveragePoints <- which(hatvalues(model) > (2*3)/234)

> filter(autoData, row_number() %in% leveragePoints)
# A tibble: 7 x 12
     id manufacturer                model displ  year   cyl      trans   drv   cty   hwy    fl   class
  <int>        <chr>                <chr> <dbl> <int> <int>      <chr> <chr> <int> <int> <chr>   <chr>
1    23    chevrolet c1500 suburban 2wd   6.0  2008     8   auto(l4)     r    12    17     r     suv
2    26    chevrolet           corvette   6.2  2008     8 manual(m6)     r    16    26     p 2seater
3    27    chevrolet           corvette   6.2  2008     8   auto(s6)     r    15    25     p 2seater
4    28    chevrolet           corvette   7.0  2008     8 manual(m6)     r    15    24     p 2seater
5    32    chevrolet    k1500 tahoe 4wd   6.5  1999     8   auto(l4)     4    14    17     d     suv
6   130         jeep grand cherokee 4wd   6.1  2008     8   auto(l5)     4    11    14     p     suv
7   131   land rover        range rover   4.0  1999     8   auto(l4)     4    11    15     p     suv

> plot(hatvalues(model))

> abline(h = c(2,3)*3/234, col="red", lty=2)
```

LEVERAGE POINTS

```
> leveragePoints <- which(hatvalues(model) > (2*3)/234)

> filter(autoData, row_number() %in% leveragePoints)
# A tibble: 7 x 12
     id manufacturer                model displ  year   cyl      trans   drv   cty   hwy    fl   class
  <int>        <chr>                <chr> <dbl> <int> <int>      <chr> <chr> <int> <int> <chr>   <chr>
1    23    chevrolet c1500 suburban 2wd   6.0  2008     8   auto(l4)     r    12    17     r     suv
2    26    chevrolet             corvette   6.2  2008     8 manual(m6)     r    16    26     p 2seater
3    27    chevrolet             corvette   6.2
4    28    chevrolet             corvette   7.0
5    32    chevrolet    k1500 tahoe 4wd   6.5
6   130         jeep grand cherokee 4wd   6.1
7   131   land rover         range rover   4.0

> plot(hatvalues(model)

> abline(h = c(2,3)*3/234, col="red", lty=2)
```

Possibly Problematic Observations

- Outliers - 213, 222
- x2 Leverage - 23, 26, 27, 28, 32, 130, 131

LEVERAGE POINTS

```
> leveragePoints3 <- which(hatvalues(model) > (3*3)/234)

> filter(autoData, row_number() %in% leveragePoints3)
# A tibble: 2 x 12
     id manufacturer           model displ  year   cyl      trans   drv   cty   hwy    fl   class
  <int>        <chr>           <chr> <dbl> <int> <int>      <chr> <chr> <int> <int> <chr>   <chr>
1    28    chevrolet        corvette   7.0  2008     8 manual(m6)     r    15    24     p 2seater
2    32    chevrolet k1500 tahoe 4wd   6.5  1999     8   auto(l4)     4    14    17     d     suv
```

LEVERAGE POINTS

```
> leveragePoints3 <- which(hatvalues(model) > (3*3)/234)

> filter(autoData, row_number() %in% leveragePoints3)
# A tibble: 2 x 12
     id manufacturer           model displ  year   cyl      trans   drv   cty   hwy    fl   class
  <int>        <chr>           <chr> <dbl> <int> <int>      <chr> <chr> <int> <int> <chr>   <chr>
1    28    chevrolet         corvette   7.0  2008     8 manual(m6)     r    15    24     p 2seater
2    32    chevrolet k1500 tahoe 4wd   6.5  1999     8   auto(l4)     4    14    17     d     suv
```

Possibly Problematic Observations

- Outliers - 213, 222
- x2 Leverage - 23, 26, 27, 28, 32, 130, 131
- x3 Leverage - 28, 32

COOK'S DISTANCE

f(x) `cooks.distance(`*model*`)`

Parameters:

- *mode*
  func

Available in `stats`
Installed with R distributions

COOK'S DISTANCE

f(x) `cooks.distance(`*`model`*`)`

Parameters:

- *`model`* is the model object created with the output from the `lm()` function

COOK'S DISTANCE

f(x) `cooks.distance(model)`

Using the lecture model from `ggplot2`'s `mpg` data:

```
> which(cooks.distance(model) > 1)
named integer(0)
```

Cook's Distance values > 1 are particularly influential observations, > .5 warrant further attention

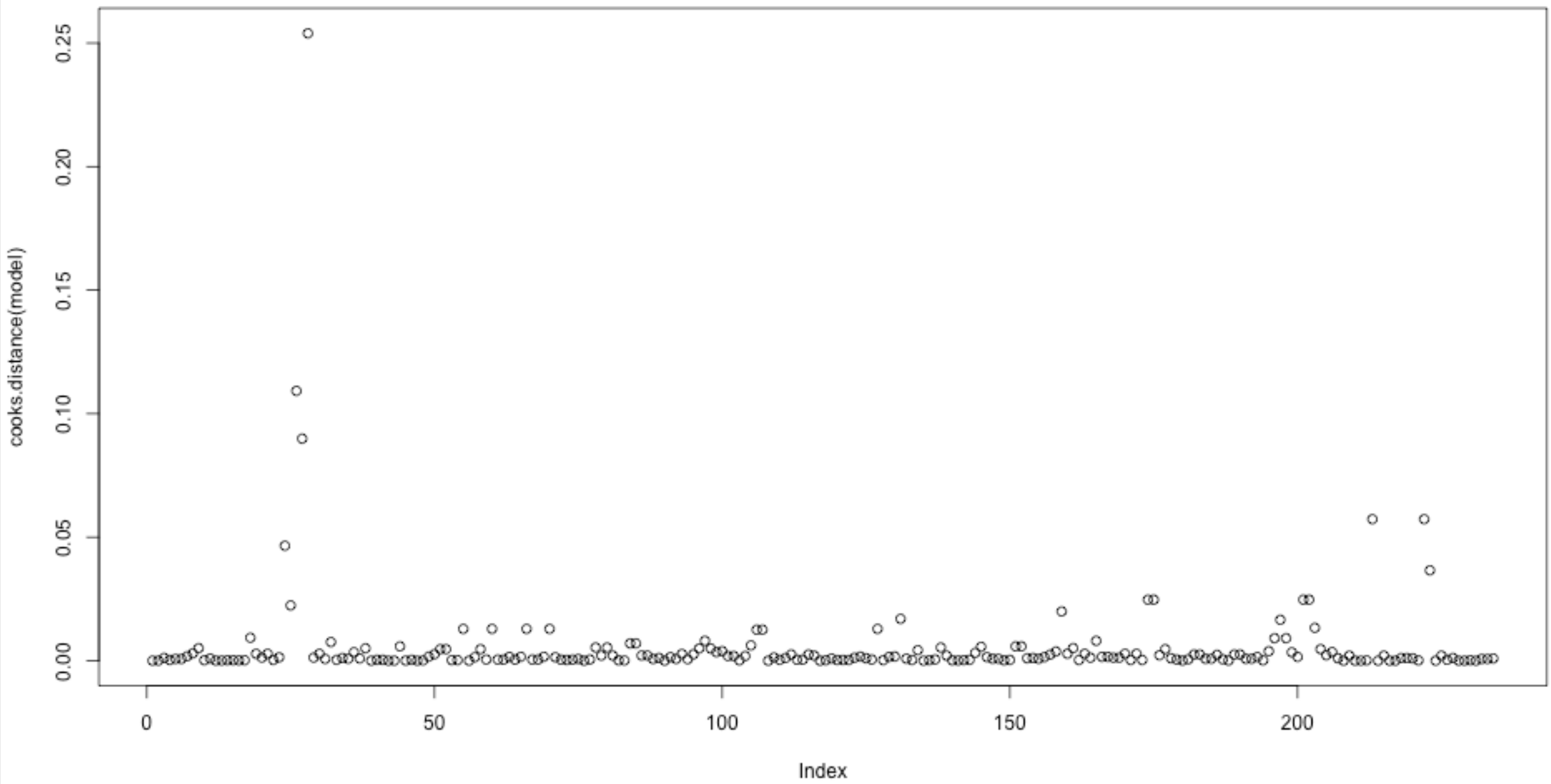
COOK'S DISTANCE

```
> which(cooks.distance(model) > 1)
named integer(0)

> which(cooks.distance(model) > .5)
named integer(0)
```

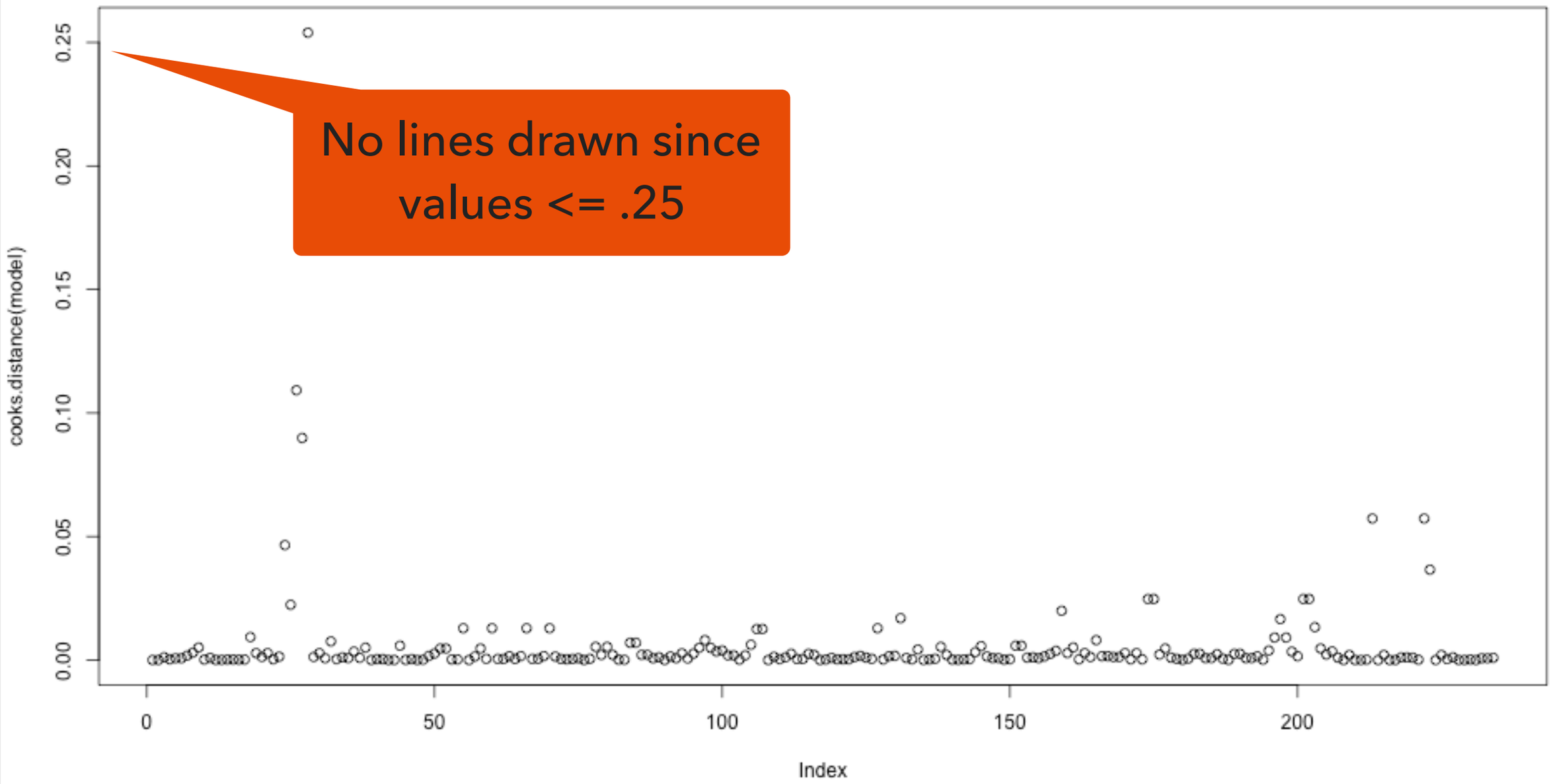
COOK'S DISTANCE

```
> which(cooks.distance(model) > 1)
named integer(0)

> which(cooks.distance(model) > .5)
named integer(0)

> plot(cooks.distance(model))

> abline(h = c(1, .5), col="red", lty=2)
```

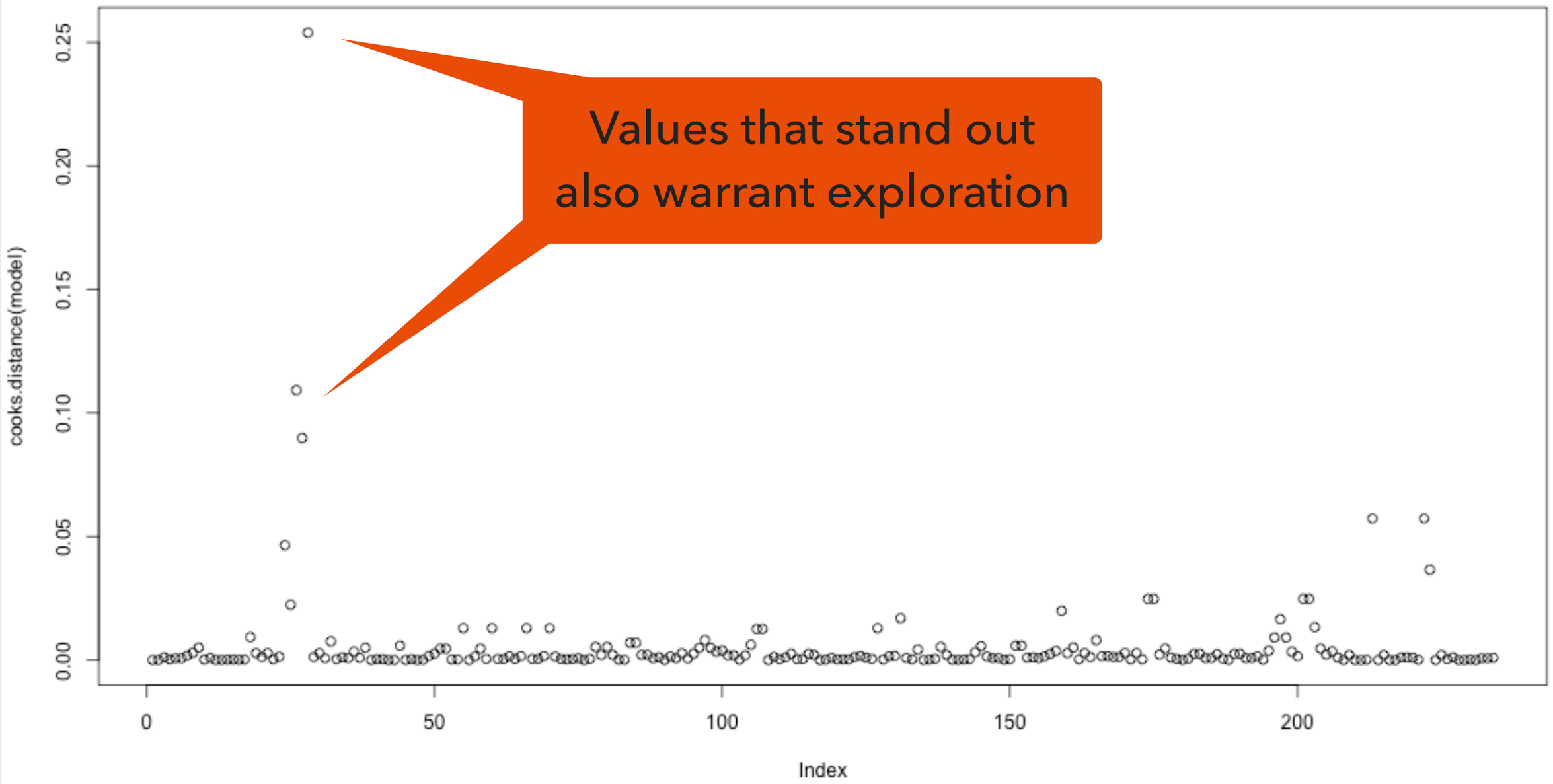
COOK'S DISTANCE



COOK'S DISTANCE



COOK'S DISTANCE



COOK'S DISTANCE

```
> cookPoints <- which(cooks.distance(model) > .08)
```

```
> filter(autoData, row_number() %in% cookPoints)
# A tibble: 3 x 12
     id manufacturer    model displ  year   cyl      trans   drv   cty   hwy    fl   class
  <int>        <chr>    <chr> <dbl> <int> <int>      <chr> <chr> <int> <int> <chr>   <chr>
1    26    chevrolet corvette   6.2  2008     8 manual(m6)     r    16    26     p 2seater
2    27    chevrolet corvette   6.2  2008     8   auto(s6)     r    15    25     p 2seater
3    28    chevrolet corvette   7.0  2008     8 manual(m6)     r    15    24     p 2seater
```

COOK'S DISTANCE

```
> cookPoints <- which(cooks.distance(model) > .08)

> filter(autoData, row_number() %in% cookPoints)
# A tibble: 3 x 12
     id manufacturer    model displ  year   cyl      trans   drv   cty   hwy    fl   class
  <int>        <chr>    <chr> <dbl> <int> <int>      <chr> <chr> <int> <int> <chr>   <chr>
1    26    chevrolet corvette   6.2  2008     8 manual(m6)     r    16    26     p 2seater
2    27    chevrolet corvette   6.2  2008
3    28    chevrolet corvette   7.0  2008
```

Possibly Problematic Observations

- Outliers - 213, 222
- x2 Leverage - 23, 26, 27, 28, 32, 130, 131
- x3 Leverage - 28, 32
- Borderline Cook's D - 26, 27, 28

INFLUENCE PLOT

f(x) `influencePlot(`*`model`*`)`

Parameters:

- *mode*
  func

Available in `car`
Install via CRAN

INFLUENCE PLOT

f(x) `influencePlot(`*`model`*`)`

Parameters:

- *`model`* is the model object created with the output from the `lm()` function

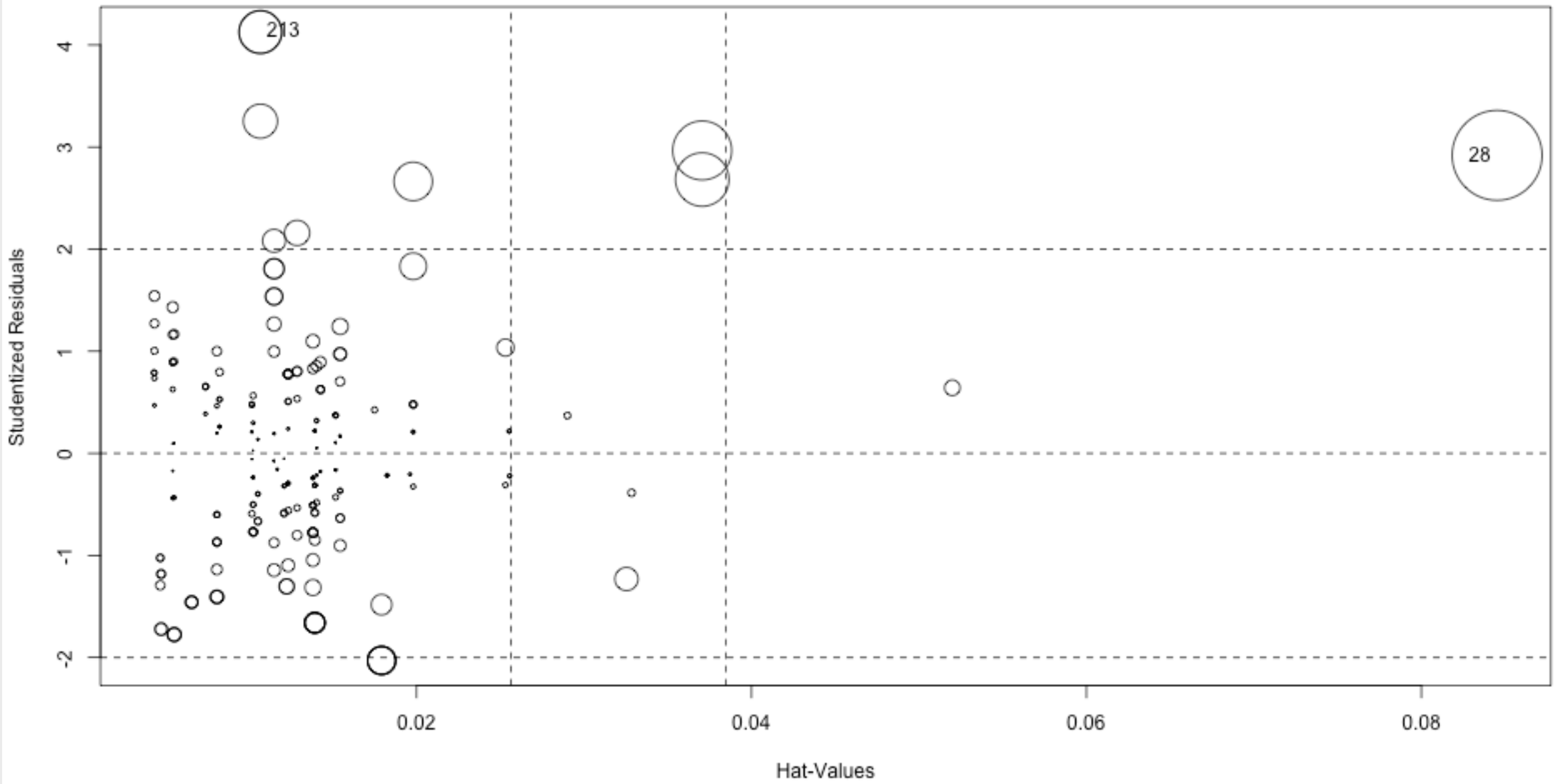
INFLUENCE PLOT

f(x) `influencePlot(`*`model`*`)`

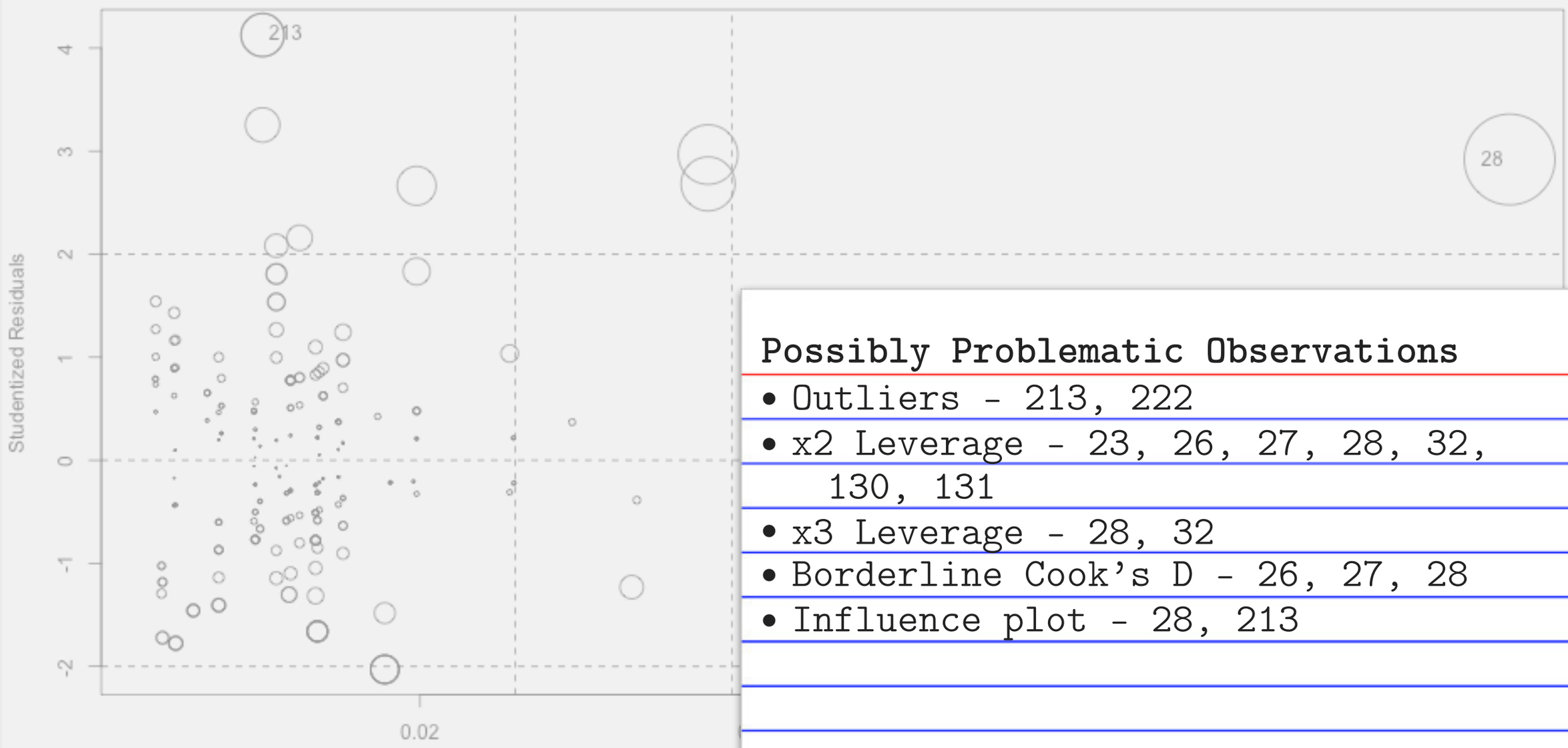
Using the lecture model from `ggplot2`'s `mpg` data:

```
> influencePlot(model)
```

INFLUENCE PLOT



INFLUENCE PLOT



Possibly Problematic Observations

- Outliers - 213, 222
- x2 Leverage - 23, 26, 27, 28, 32, 130, 131
- x3 Leverage - 28, 32
- Borderline Cook's D - 26, 27, 28
- Influence plot - 28, 213

ADVANCED ASSUMPTIONS

- Normality - Residuals should be normally distributed
- Homoskedasticity - the variance of the residuals should be constant
- Autocorrelation - residuals should not be correlated with each other
- Multi-collinearity - predictors should not be highly correlated

NORMALITY OF RESIDUALS

QUANTILE-QUANTILE PLOT

f(x) `qqPlot(model)`

Parameters:

- *model*
  func

Available in `car`
Download via CRAN

QUANTILE-QUANTILE PLOT

f(x) `qqPlot(`*`model`*`)`

Parameters:

- *`model`* is the model object created with the output from the `lm()` function

QUANTILE-QUANTILE PLOT

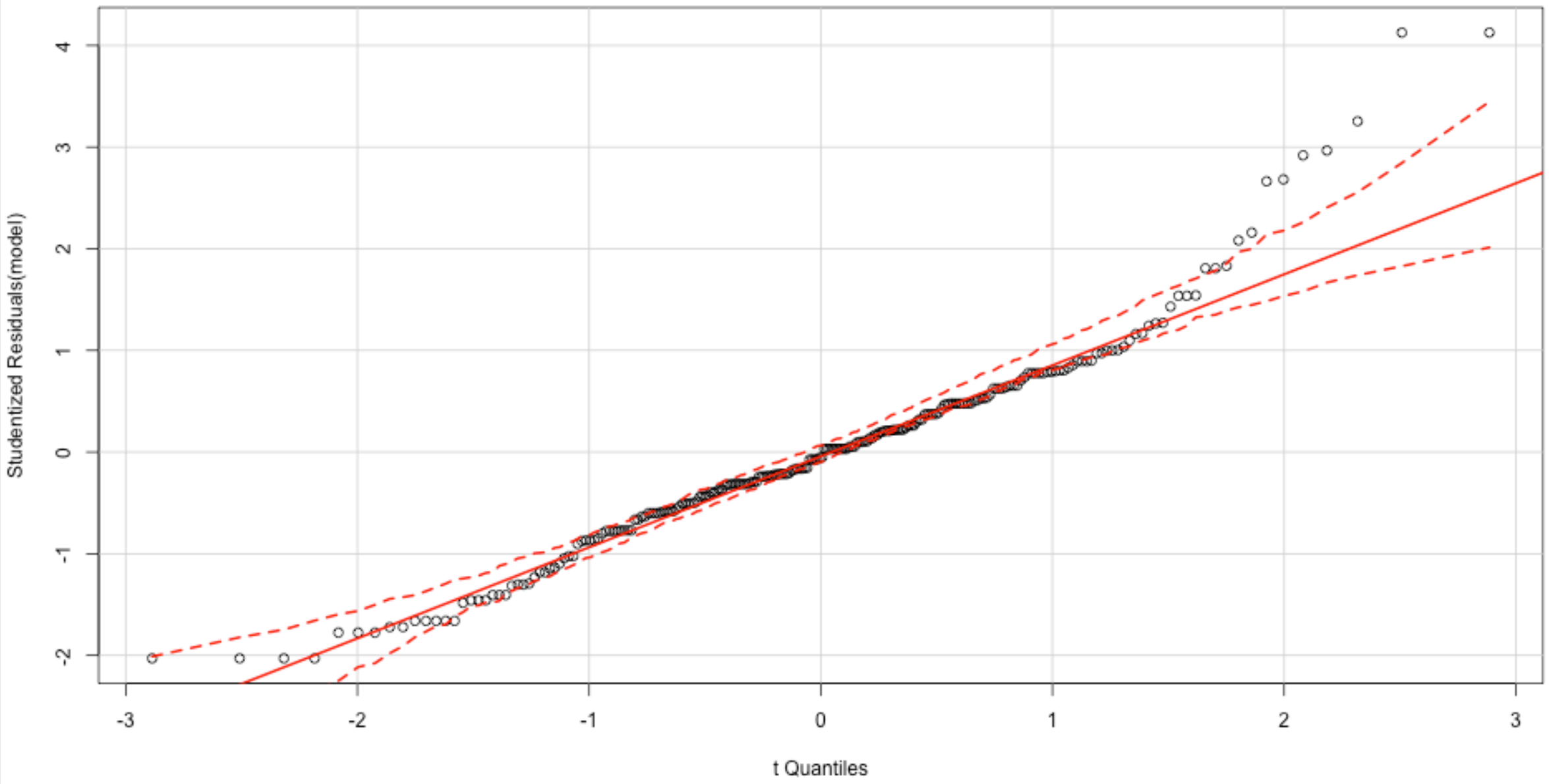
f(x) `qqPlot(`*model*`)`

Using the lecture model from `ggplot2`'s `mpg` data:

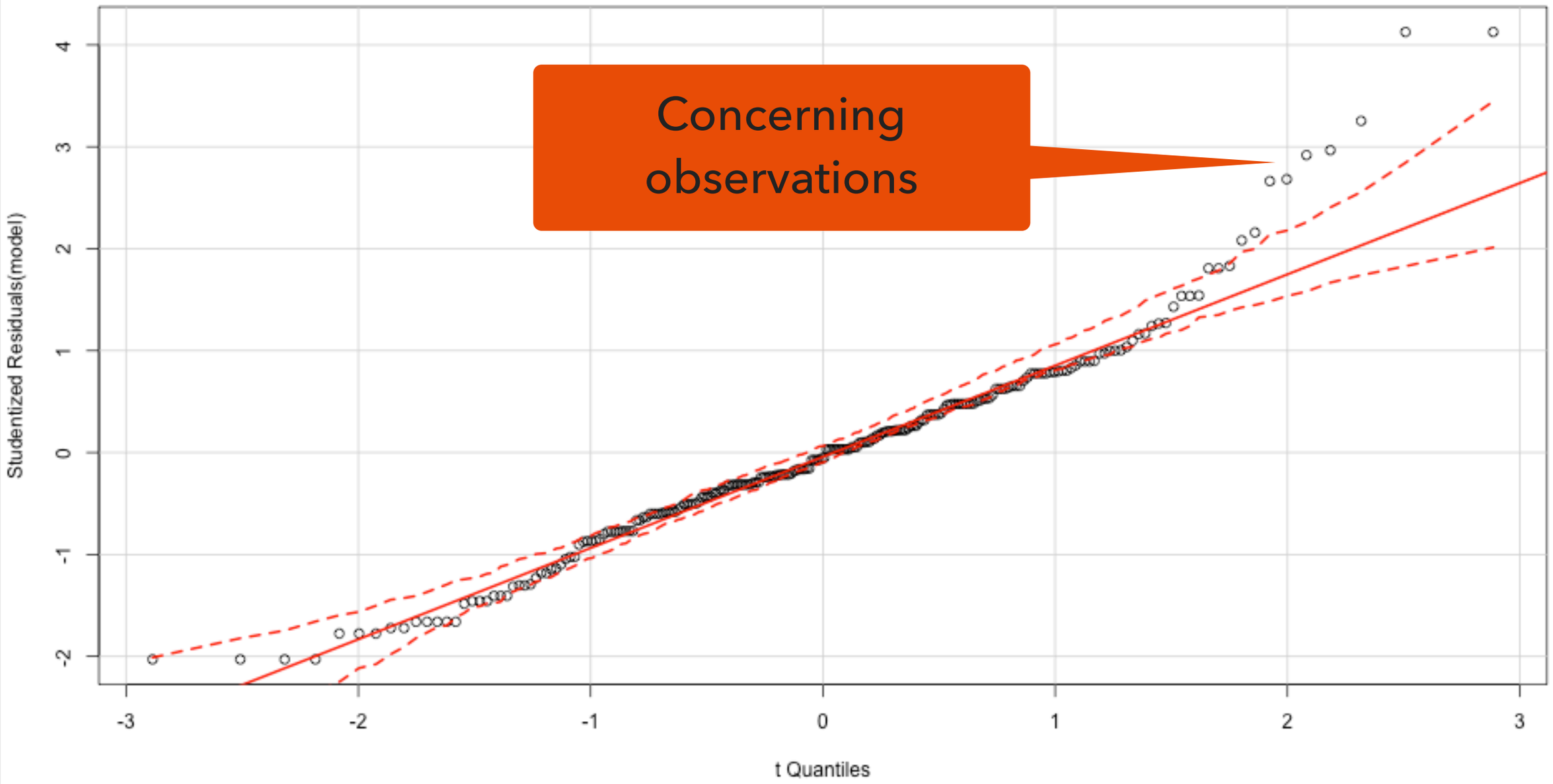
```
> qqPlot(model)
```

Interpret as with other q-q plots, except that we should be particularly concerned with observations that fall outside of the dashed lines.

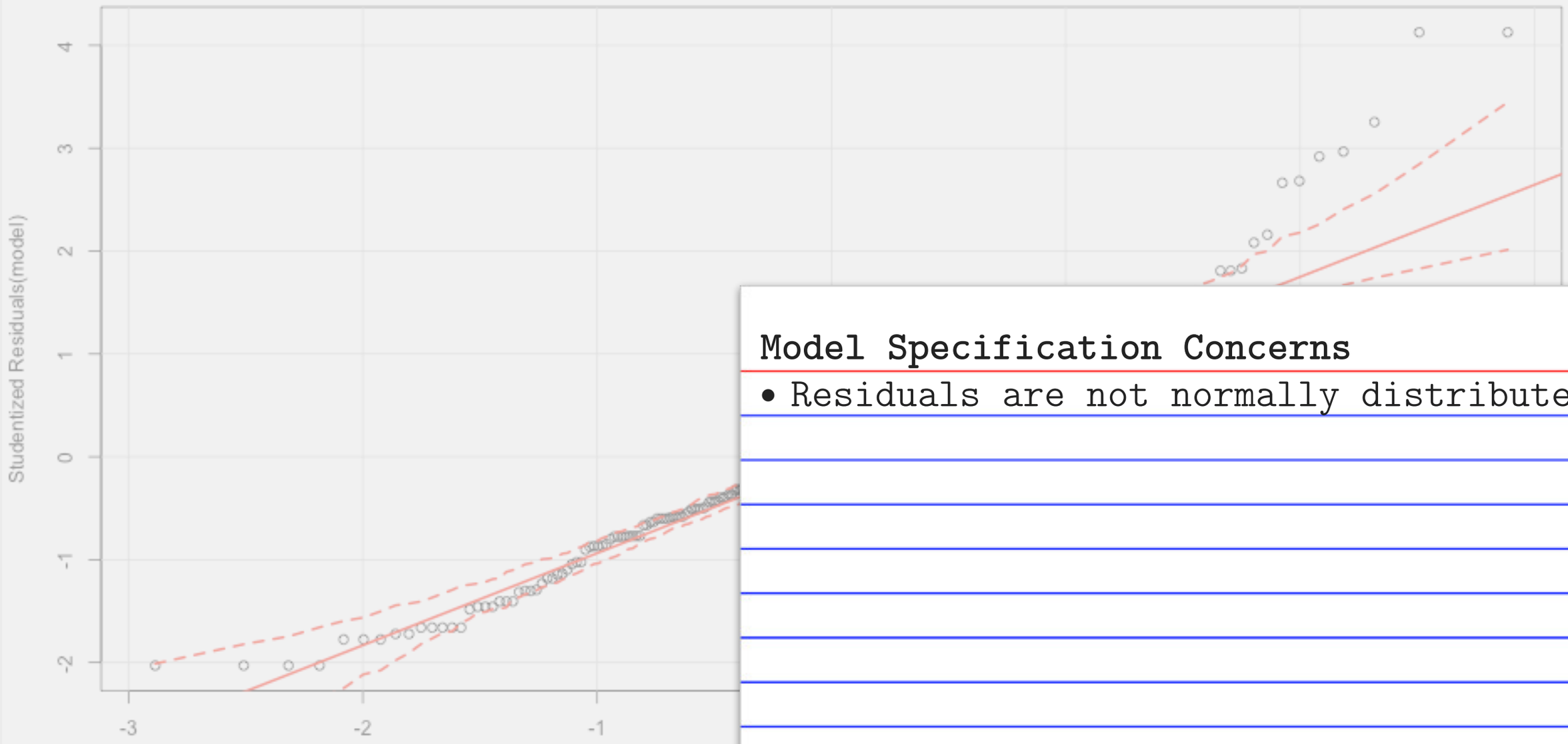
QUANTILE-QUANTILE PLOT



QUANTILE-QUANTILE PLOT



QUANTILE-QUANTILE PLOT



Model Specification Concerns

- Residuals are not normally distributed

e

HOMOSKEDASTIC ERRORS

BREUSCH-PAGAN TEST

f(x) `bptest(model)`

Parameters:

- *model*
  func

Available in `lmtest`
Install via CRAN

BREUSCH-PAGAN TEST

f(x) `bptest(`*`model`*`)`

Parameters:

- *`model`* is the model object created with the output from the `lm()` function

BREUSCH-PAGAN TEST

f(x) `bptest(model)`

Using the lecture model from `ggplot2`'s `mpg` data:

```
> bptest(model)

<<<< OUTPUT OMITTED >>>>>
```

Use this if the residuals are normally distributed. The null hypothesis is that the errors are homoskedastic.

BREUSCH-PAGAN TEST

```
> bptest(model)

    studentized Breusch-Pagan test

data:  model
BP = 8.5133, df = 2, p-value = 0.01417
```

$p < .05$ indicates heteroskedastic errors

WHITE'S TEST

f(x)

```
bptest(model, ~ x1 * x2 + I(x1^2) + I(x2^2),
     data = dataFrame)
```

Using the lecture model from `ggplot2`'s `mpg` data:

```
> bptest(model, ~ displ * cyl + I(displ^2) +
    I(cyl^2), data = autoData)
```

<<<< OUTPUT OMITTED >>>>>

Use this if the residuals are not normally distributed. The null hypothesis is that the errors are homoskedastic.

WHITE'S TEST

```
> bptest(model, ~ displ * cyl + I(displ^2) + I(cyl^2),
    data = autoData)

    studentized Breusch-Pagan test

data:  model
BP = 16.022, df = 5, p-value = 0.00678
```

$p < .05$ indicates heteroskedastic errors

WHITE'S TEST

```
> bptest(model, ~ displ * cyl + I(displ^2) + I(cyl^2),
    data = autoData)

    studentized Breusch-Pagan test

data:  model
BP = 16.022, df = 5, p-value =
```

Model Specification Concerns

- Residuals are not normally distributed
- Residuals are heteroskedastic per White's test

RESIDUAL PLOT

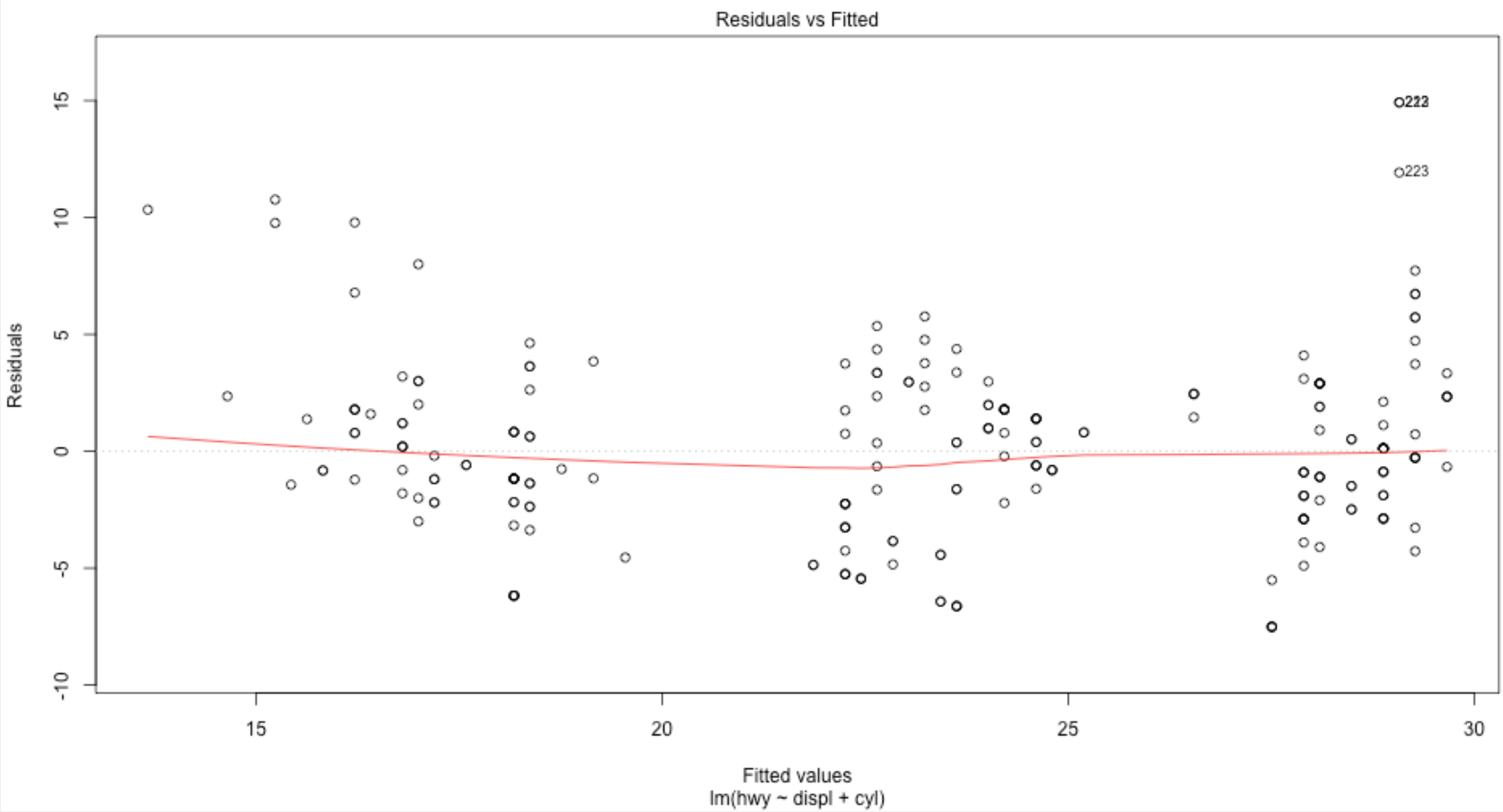
f(x)

```
plot(model, which = 1)
```

Using the lecture model from `ggplot2`'s `mpg` data:

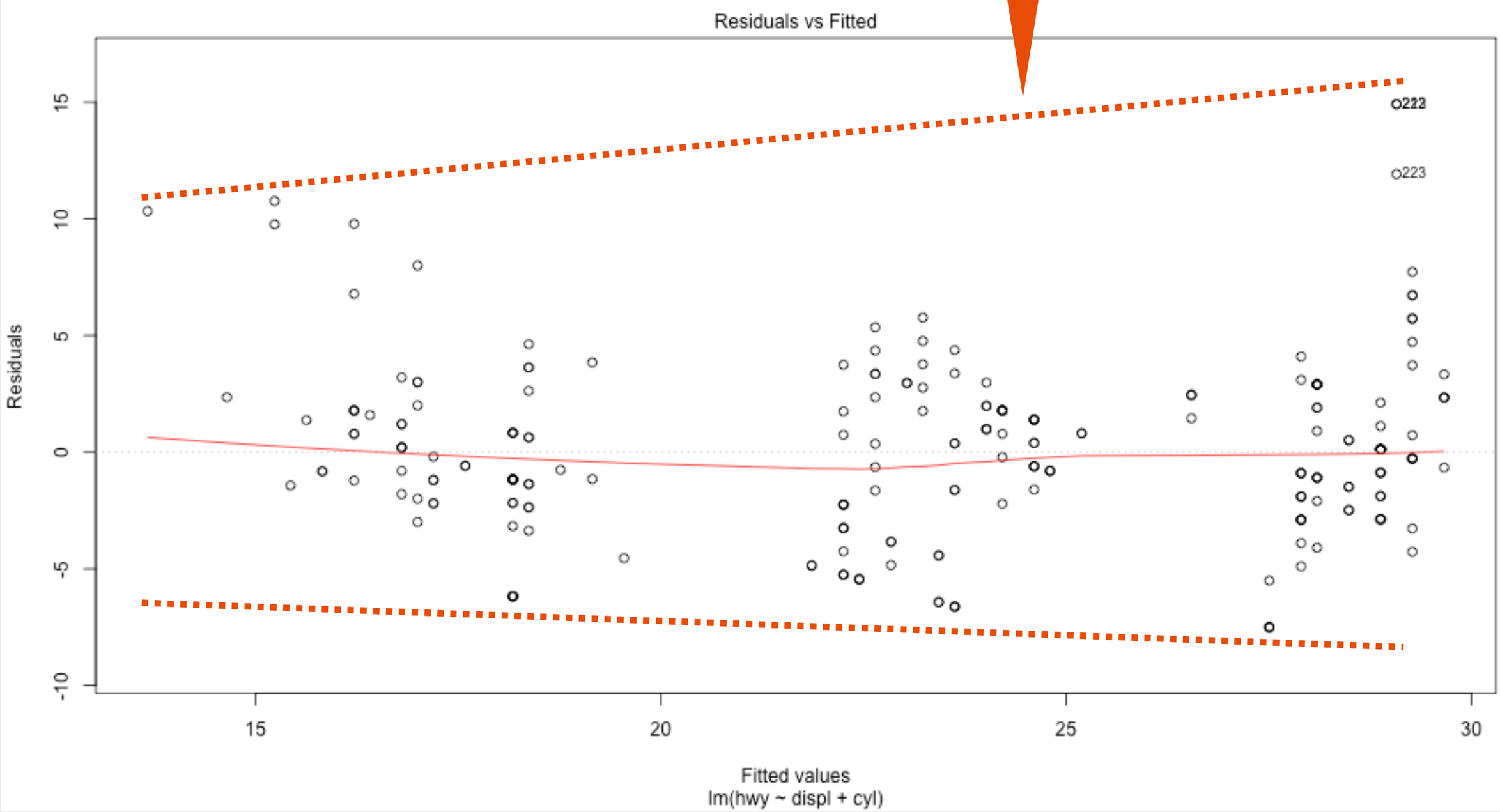
```
> plot(model, which = 1)
```

RESIDUAL PLOT

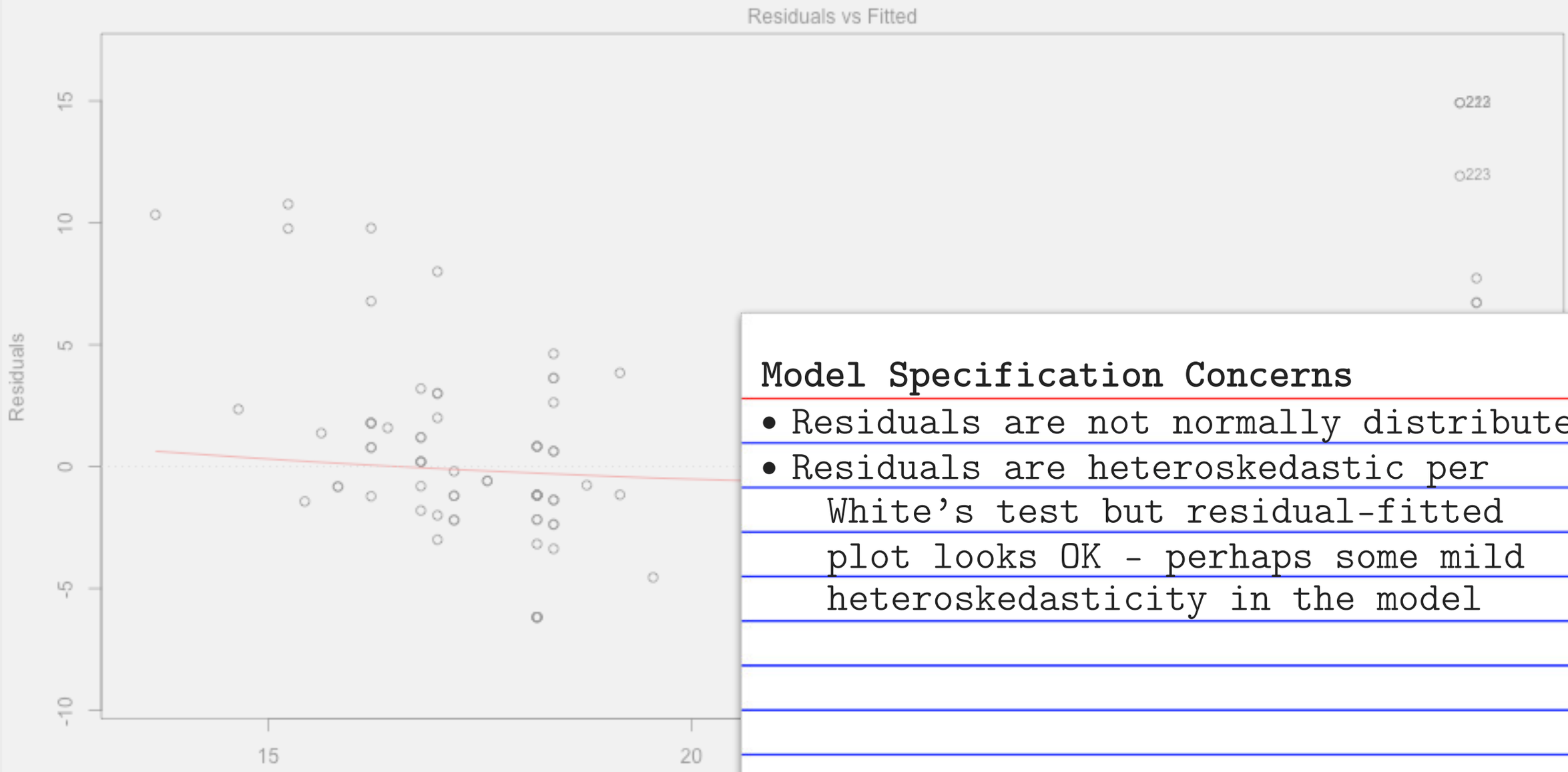


RESIDUAL PLOT

look for narrowing in plot



RESIDUAL PLOT



Model Specification Concerns

- Residuals are not normally distributed
- Residuals are heteroskedastic per White's test but residual-fitted plot looks OK - perhaps some mild heteroskedasticity in the model

AUTO-CORRELATION

DURBIN-WATSON TEST

f(x) `durbinWatsonTest(`*model*`)`

Parameters:

- *mode*
  func

Available in `car`
Install via CRAN

DURBIN-WATSON TEST

f(x) `durbinWatsonTest(`*`model`*`)`

Parameters:

- *`model`* is the model object created with the output from the `lm()` function

DURBIN-WATSON TEST

f(x) `durbinWatsonTest(model)`

Using the lecture model from `ggplot2`'s `mpg` data:

```
> durbinWatsonTest(model)
```

<<<< OUTPUT OMITTED >>>>>

The null hypothesis is that the errors are not correlated (i.e. autocorrelation is not a concern).

DURBIN-WATSON TEST

```
> durbinWatsonTest(model)
 lag Autocorrelation D-W Statistic p-value
   1       0.534405     0.9284804       0
 Alternative hypothesis: rho != 0
```

$p < .05$ indicates autocorrelation is a concern

DURBIN-WATSON TEST

```
> durbinWatsonTest(model)
 lag Autocorrelation D-W Statistic p-value
   1       0.534405     0.9284804       0
 Alternative hypothesis: rho != 0
```

Model Specification Concerns

- Residuals are not normally distributed
- Residuals are heteroskedastic per White's test but residual-fitted plot looks OK - perhaps some mild heteroskedasticity in the model
- Autocorrelation is a concern

g

MULTI-COLLINEARITY

VARIANCE INFLATION FACTOR

f(x) `vif(model)`

Parameters:

- *model*
  func

Available in `car`
Install via CRAN

VARIANCE INFLATION FACTOR

f(x) `vif(model)`

Parameters:

- *`model`* is the model object created with the output from the `lm()` function

VARIANCE INFLATION FACTOR

f(x)

`vif(model)`

Using the lecture model from `ggplot2`'s `mpg` data:

```
> sqrt(vif(model))
   displ      cyl
2.724912 2.724912
```

Individual square root of VIF values should be less than 10

VARIANCE INFLATION FACTOR

f(x) `vif(model)`

Using the lecture model from `ggplot2`'s `mpg` data:

```
> mean(sqrt(vif(model)))
[1] 2.724912
```

Mean square root of VIF values should be less than 1

VARIANCE INFLATION FACTOR

```
> sqrt(vif(model))
   displ      cyl
2.724912 2.724912
```

```
> mean(sqrt(vif(model)))
[1] 2.724912
```

VARIANCE INFLATION FACTOR

```
> sqrt(vif(model))
   displ      cyl
2.724912 2.724912
```

```
> mean(sqrt(vif(model)))
[1] 2.724912
```

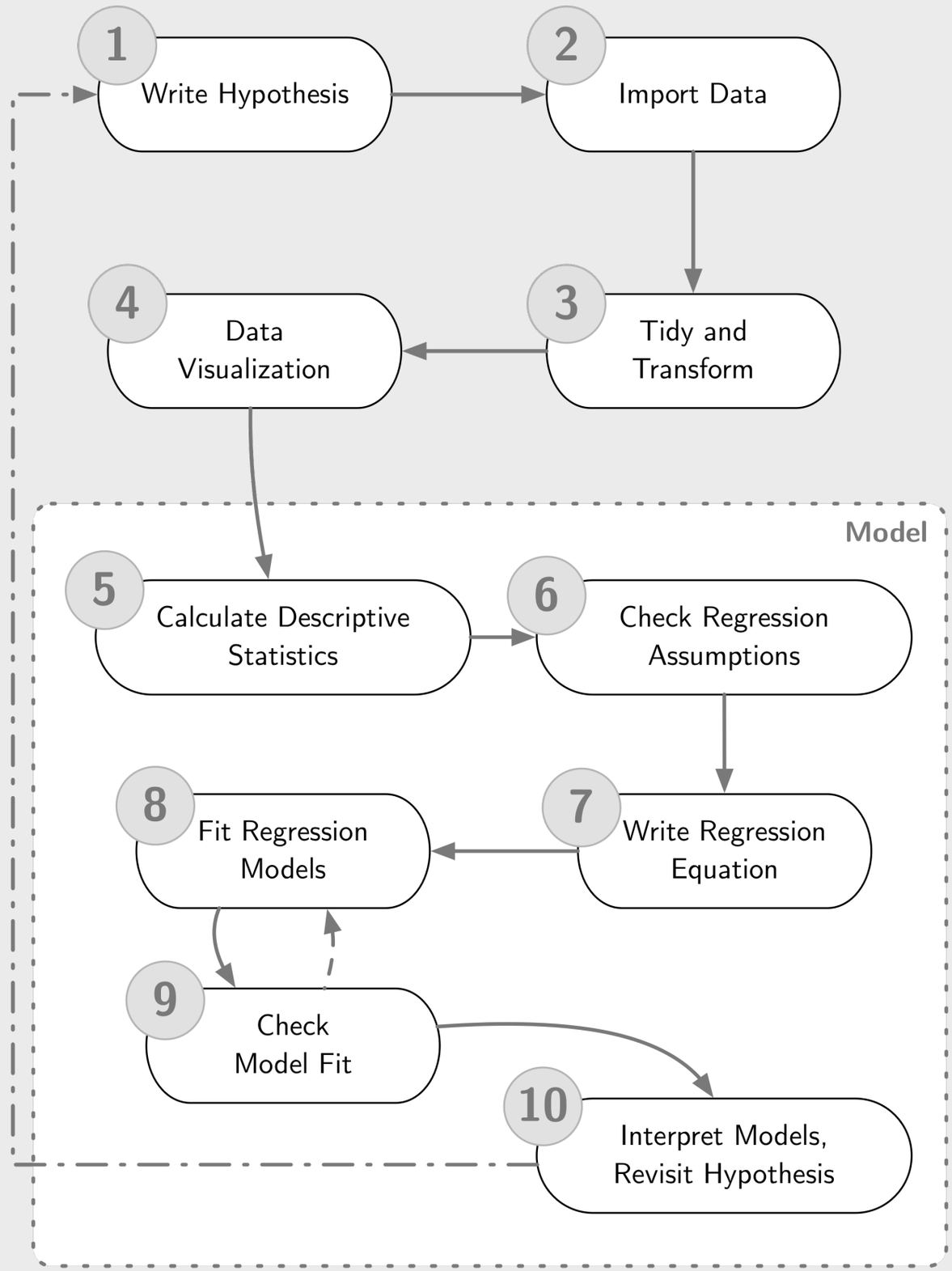
Possibly Problematic Variables

- No non-linear relationships detected
- Perhaps some mild multi-collinearity
  based on mean VIF but individual VIF values OK

4 ADJUSTING MODELS

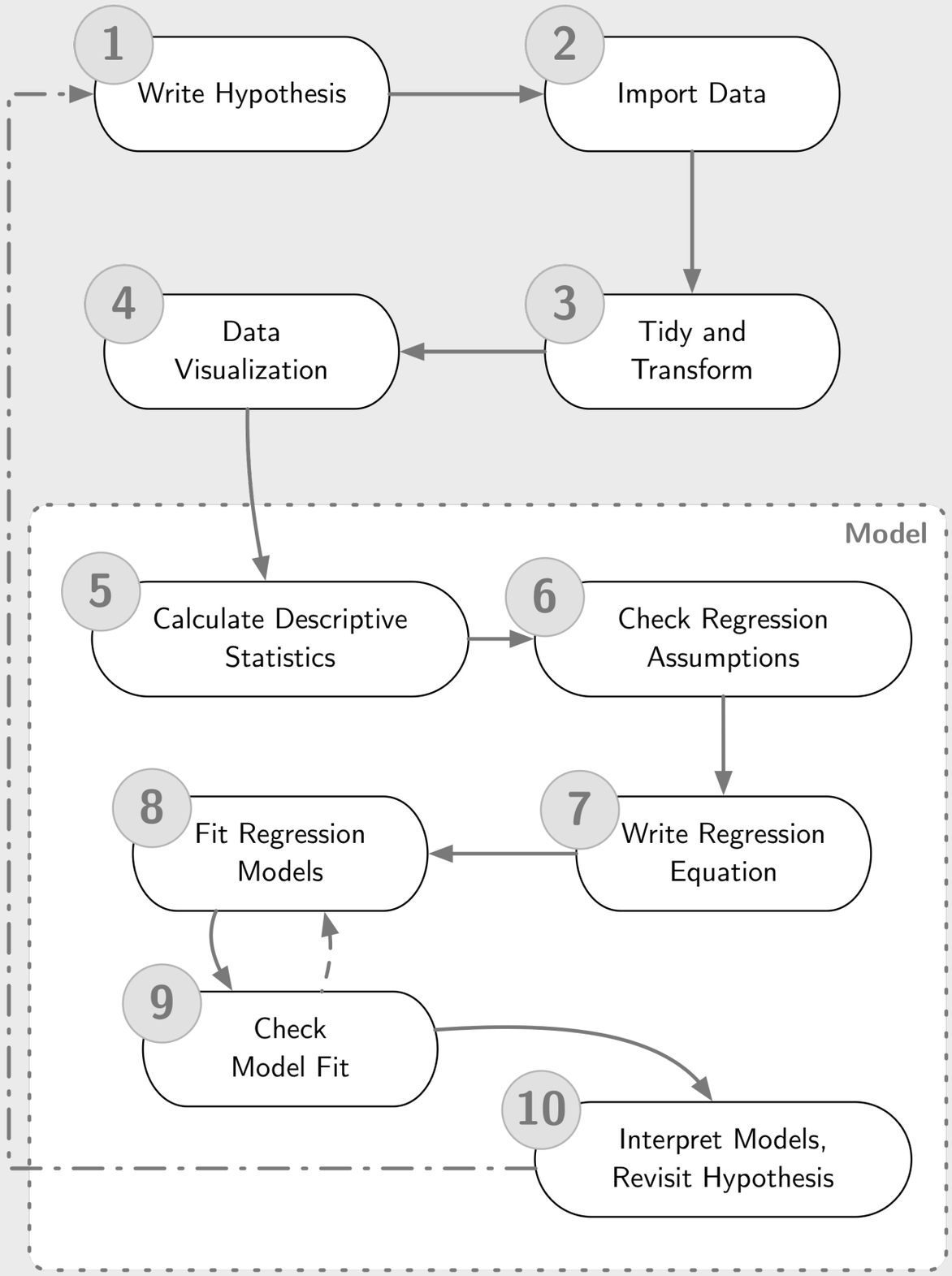
WORKFLOW

- **Before:** Check regression assumptions
  - Levels of measurement
  - Correlation analysis (look for indicators that multi-collinearity may be an issue in the x variables)
  - Check the distribution of y



WORKFLOW

- **After:** Check model fit
  - Check for model specification issues first (non-linearity, residuals, auto-correlation, multi-collinearity)
  - If the model appears to be correctly specified, then check for unusual observations
- **Then:** If warranted, address unusual observations, re-calculate a model on a subsample, and re-check model fit



SUMMARY OF DIAGNOSTICS

Possibly Problematic Variables

- No non-linear relationships detected
- Perhaps some mild multi-collinearity based on mean VIF but individual VIF values OK

Possibly Problematic Observations

- Outliers - 213, 222
- x2 Leverage - 23, 26, 27, 28, 32, 130, 131
- x3 Leverage - 28, 32
- Borderline Cook's D - 26, 27, 28
- Influence plot - 28, 213

Model Specification Concerns

- Residuals are not normally distributed
- Residuals are heteroskedastic per White's test but residual-fitted plot looks OK - perhaps some mild heteroskedasticity in the model
- Autocorrelation is a concern

SUMMARY OF DIAGNOSTICS

Possibly Problematic Variables

- No non-linear relationships detected
- Perhaps some mild multi-collinearity based on mean VIF but individual VIF values OK

Model Specification Concerns

- Residuals are not normally distributed
- Residuals are heteroskedastic per White's test but residual-fitted plot looks OK - perhaps some mild heteroskedasticity in the model
- Autocorrelation is a concern

SUMMARY OF DIAGNOSTICS

- Are omitted variables the underlying cause of specification concerns?
- Does transforming a problematic variable address specification concerns?
- Does removing variables and (optionally) creating a scale address multi-collinearity?
- If concerns remain, use “robust” standard errors.

Possibly Problematic Variables

- No non-linear relationships detected
- Perhaps some mild multi-collinearity based on mean VIF but individual VIF values OK

Model Specification Concerns

- Residuals are not normally distributed
- Residuals are heteroskedastic per White’s test but residual-fitted plot looks OK - perhaps some mild heteroskedasticity in the model
- Autocorrelation is a concern

“ROBUST” STANDARD ERRORS

f(x) `vcovHC(`*`model`*`, “HC1”)`

Parameters:

- *mode*
  func

- *HC3*
  (among a number of options). `HC1` is used by Stata, for example. `HC3` is recommended by Long and Ervin (2000).

Available in `sandwich`
Download via CRAN

“ROBUST” STANDARD ERRORS

f(x) `vcovHC(`*`model`*`, “HC3”)`

Parameters:

- *`model`* is the model object created with the output from the `lm()` function
- *`HC3`* is the specific type of heteroskedasticly-consistent estimator (among a number of options). `HC1` is used by Stata, for example. `HC3` is recommended by Long and Ervin (2000).

“ROBUST” STANDARD ERRORS

`vcovHC(model, “HC3”)`

Using the lecture model from `ggplot2`’s `mpg` data:

```
> vcovHC(model, “HC3”)

<<<<< OUTPUT OMITTED >>>>>
```

Produces what is known as a covariance matrix estimate.

“ROBUST” STANDARD ERRORS

f(x) `coeftest(`*`model`*`, vcov = `*`cme`*`)`

Parameters:

- *mode*
  func

- vcov
  `vcovHC()` function.

Available in `lmtest`
Download via CRAN

“ROBUST” STANDARD ERRORS

`coeftest(`*`model`*`, vcov = `*`cme`*`)`

Parameters:

- *`model`* is the model object created with the output from the `lm()` function
- `vcov` accepts the covariance matrix estimate (cme) created by the `vcovHC()` function.

“ROBUST” STANDARD ERRORS

f(x) `coeftest(model, vcov = cme)`

Using the lecture model from `ggplot2`’s `mpg` data:

```
> coeftest(model, vcov = vcovHC(model, “HC3”))

<<<<< OUTPUT OMITTED >>>>>
```

Produces a more conservative set of standard errors, t-values, and p-values that inference should be made from.

“ROBUST” STANDARD ERRORS

```
> coeftest(model, vcov = vcovHC(model, "HC3"))

t test of coefficients:

             Estimate Std. Error t value  Pr(>|t|)
(Intercept) 38.21615    1.08010 35.3820 < 2.2e-16 ***
displ       -1.95987    0.67666 -2.8964  0.004137 **
cyl         -1.35369    0.48448 -2.7941  0.005642 **
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

“ROBUST” STANDARD ERRORS

```
> robustSE <- tidy(coeftest(model, vcov = vcovHC(model, “HC3”)))
```

```
> stargazer(robustSE, summary = FALSE, rownames = FALSE)
```

```
<<<<< OUTPUT OMITTED >>>>>
```

There is not a way to integrate these output into results automatically using `stargazer`. Your best bet is to create a set of LaTeX output with your models and then copy and paste the appropriate values into it.

SUMMARY OF DIAGNOSTICS

Possibly Problematic Variables

- No non-linear relationships detected
- Perhaps some mild multi-collinearity based on mean VIF but individual VIF values OK

Possibly Problematic Observations

- Outliers - 213, 222
- x2 Leverage - 23, 26, 27, 28, 32, 130, 131
- x3 Leverage - 28, 32
- Borderline Cook's D - 26, 27, 28
- Influence plot - 28, 213

Model Specification Concerns

- Residuals are not normally distributed
- Residuals are heteroskedastic per White's test but residual-fitted plot looks OK - perhaps some mild heteroskedasticity in the model
- Autocorrelation is a concern

SUMMARY OF DIAGNOSTICS

Possibly Problematic Observations

- Outliers - 213, 222
- x2 Leverage - 23, 26, 27, 28, 32, 130, 131
- x3 Leverage - 28, 32
- Borderline Cook's D - 26, 27, 28
- Influence plot - 28, 213

SUMMARY OF DIAGNOSTICS

- Look at observations that appear in multiple categories or have particularly large values in one category

Possibly Problematic Observations

- Outliers - 213, 222
- x2 Leverage - 23, 26, 27, 28, 32, 130, 131
- x3 Leverage - 28, 32
- Borderline Cook's D - 26, 27, 28
- Influence plot - 28, 213

SUMMARY OF DIAGNOSTICS

Possibly Problematic Observations

- Outliers - 213, 222
- x2 Leverage - 23, 26, 27, 28, 32, 130, 131
- x3 Leverage - 28, 32
- Borderline Cook's D - 26, 27, 28
- Influence plot - 28, 213

- Look at observations that appear in multiple categories or have particularly large values in one category
- IDs 26, 27, 28, and 213 all appear in multiple categories

SUMMARY OF DIAGNOSTICS

Possibly Problematic Observations

- Outliers - 213, 222
- x2 Leverage - 23, 26, 27, 28, 32, 130, 131
- x3 Leverage - 28, 32
- Borderline Cook's D - 26, 27, 28
- Influence plot - 28, 213

- Look at observations that appear in multiple categories or have particularly large values in one category
- IDs 26, 27, 28, and 213 all appear in multiple categories
- Additionally, identify the most extreme values in other categories
- IDs 32 and 222 meet this second criteria

CREATING A SUBSAMPLE

```
> flaggedObs <- c(26, 27, 28, 32, 213, 222)

> autoData %>%
    mutate(insample = ifelse(id %in% flaggedObs, TRUE, FALSE) %>%
    filter(insample == FALSE) -> autoDataSub

> modelSub <- lm(hwy ~ displ+cyl, data = autoDataSub)
```

CREATING A SUBSAMPLE

```
> AIC(model)
[1] 1288.779

> AIC(modelSub)
[1] 1194.796

> BIC(model)
[1] 1302.601

> BIC(modelSub)
[1] 1208.514
```

5 EFFECT SIZES

ETA SQUARED

f(x) `etasq(model, partial = FALSE)`

Parameters:

- *model*
  func

- *part* When FALSE, the “full” eta-squared value is calculated.

Available in `heplots`
Download via CRAN

ETA SQUARED

f(x) `etasq(`*`model`*`, partial = FALSE)`

Parameters:

- *`model`* is the model object created with the output from the `lm()` function
- *`partial`* allows for the calculation of a variation of eta-squared. When `FALSE`, the "full" eta-squared value is calculated.

ETA SQUARED

f(x) `etasq(model, partial = FALSE)`

Using the `hwy` variable from `ggplot2`'s `mpg` data:

```
> etasq(model, partial = FALSE)
```

<<<< OUTPUT OMITTED >>>>>

A small effect is ~ .2, a moderate effect is ~ .13, and a large effect is ~ .26.

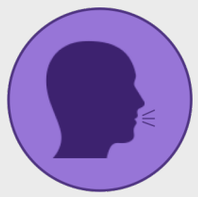
ETA SQUARED

```
> etasq(model, partial = FALSE)
              eta^2
displ     0.05565362
cyl       0.04131008
Residuals         NA
```

How would you interpret this result?

ETA SQUARED

```
> etasq(model, partial = FALSE)
             eta^2
displ     0.05565362
cyl       0.04131008
Residuals       NA
```



Both the effect sizes for displacement and number of cylinders are small.

6 BACK MATTER

WHAT WE COVERED

2. Images with LaTeX

3. Regression Diagnostics

4. Adjusting Models

5. Effect Sizes

REMINDERS

PS-10 is due next week along with Lab-14.

This week is the last week of content that is needed for the final project!