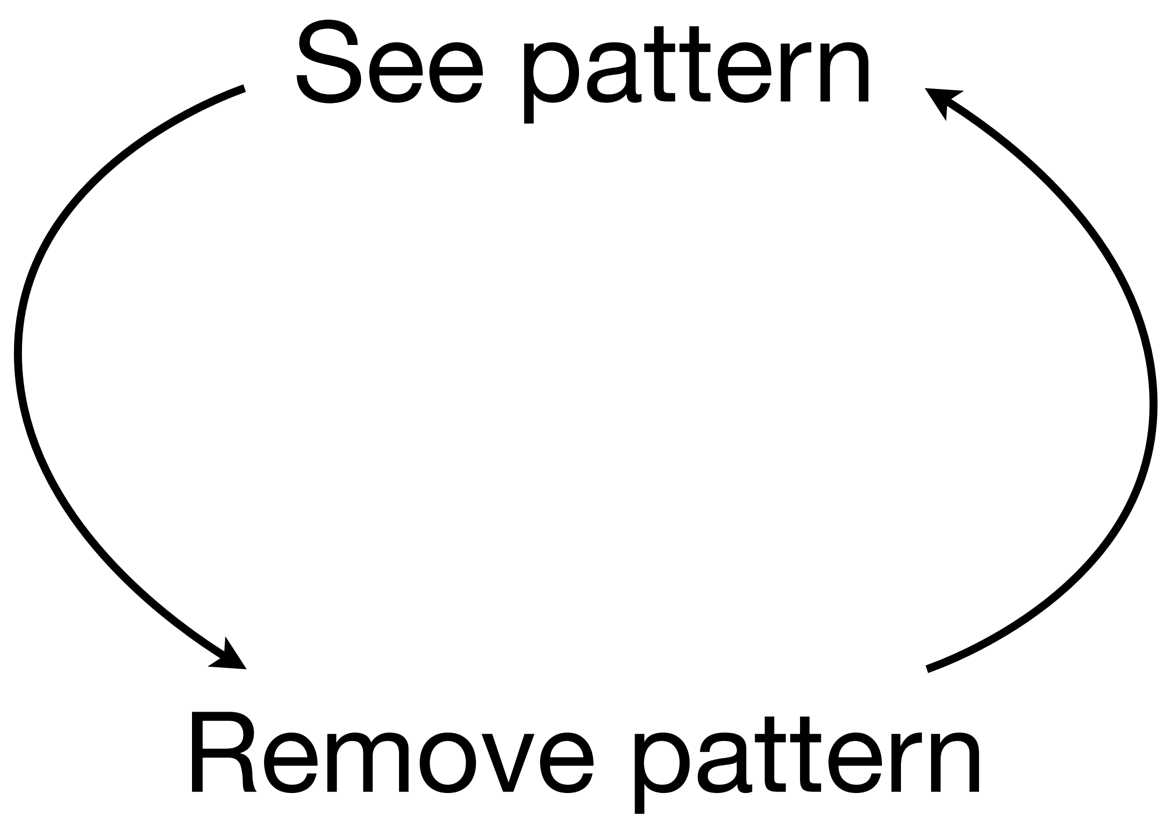
Introduction to modelling

Hadley Wickham
Assistant Professor / Dobelman Family Junior Chair
Department of Statistics / Rice University

June 2012

1. Model as tools
2. Linear trend
3. Group means

Models as tools


See pattern
Remove pattern

3 tools

Always think about what 0 means

- Remove linear trend
- Remove group means
- *Remove smooth trend*

Graphic	Model
`geom_smooth(` `  method = lm)`	`lm(y ~ x)`

`stat_summary(`
`  fun.y = mean,`
`  geom = "point")`
	`lm(y ~ factor(x))`

`geom_smooth()`	`library(mgcv)` `gam(y ~ s(x))`

Linear trend

```r
# To get started
library(ggplot2)

diamonds$x[diamonds$x == 0] <- NA
diamonds$y[diamonds$y == 0] <- NA
diamonds$y[diamonds$y > 30] <- NA
diamonds$z[diamonds$z == 0] <- NA
diamonds$z[diamonds$z > 30] <- NA

diamonds <- subset(diamonds, carat < 2)

lm_line <- geom_smooth(method = lm, se = F, size = 2)

options(na.action = na.exclude)
```

```
diamonds <- mutate(diamonds,
  volume = x * y * z,
  density = volume / carat)

qplot(carat, volume, data = diamonds) + lm_line

modvol <- lm(volume ~ carat, data = diamonds)
# Slope and intercept:
coef(modvol)

qplot(carat, predict(modvol), data = diamonds)
qplot(carat, resid(modvol), data = diamonds)
```

Your turn

Repeat this technique for x vs y, and x vs. z. For x vs y, how does the result compare to x vs y-x? Why is it different?

```
mody <- lm(y ~ x, data = diamonds)
coef(mody)
# y = 0.05 + 0.99·x

qplot(x, y, data = diamonds)
qplot(x, resid(mody), data = diamonds)
qplot(x, y - x, data = diamonds)

modz <- lm(z ~ x, data = diamonds)
coef(modz)

qplot(x, z, data = diamonds)
qplot(x, resid(modz), data = diamonds)
```

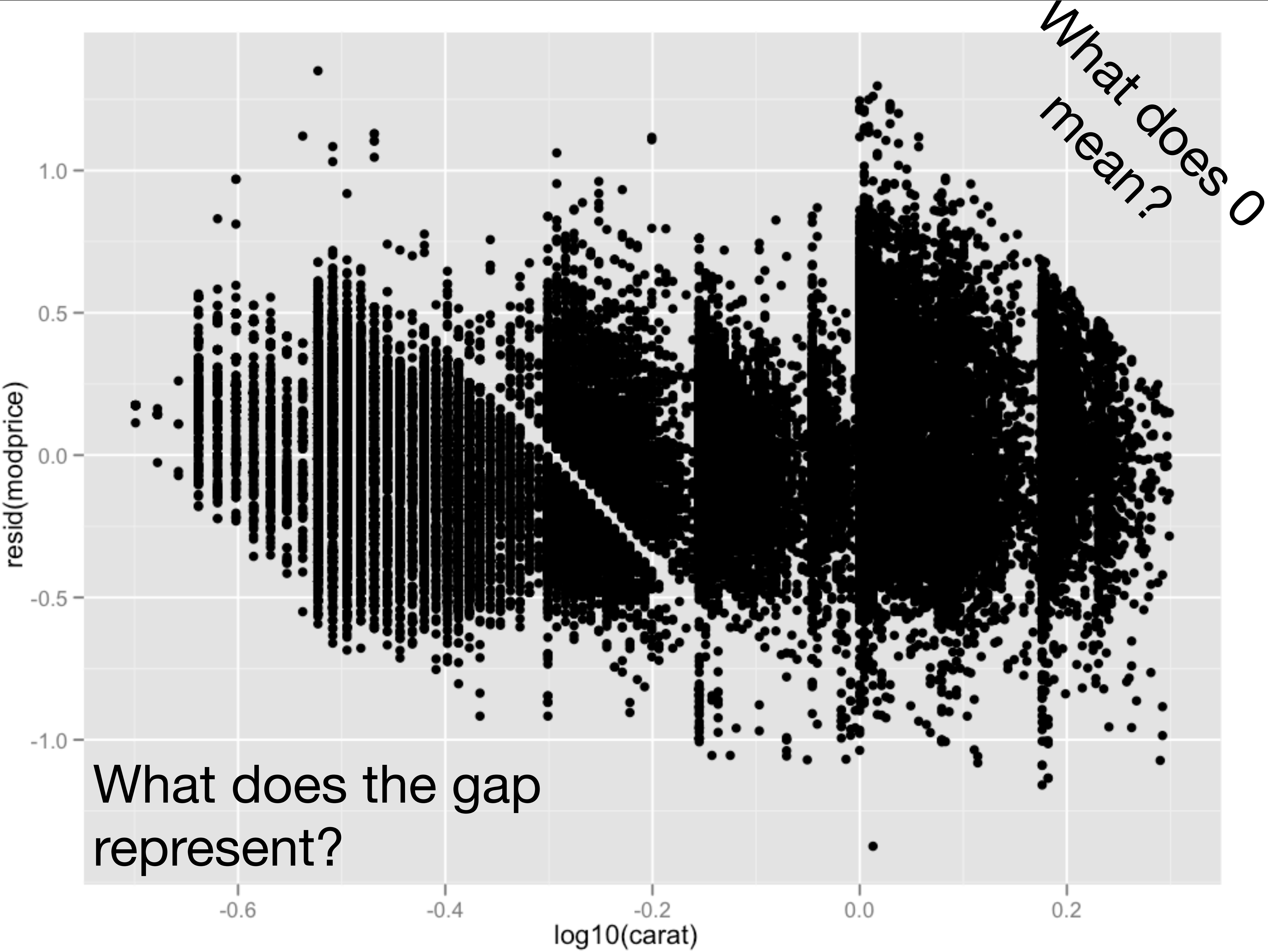
What does 0 mean?

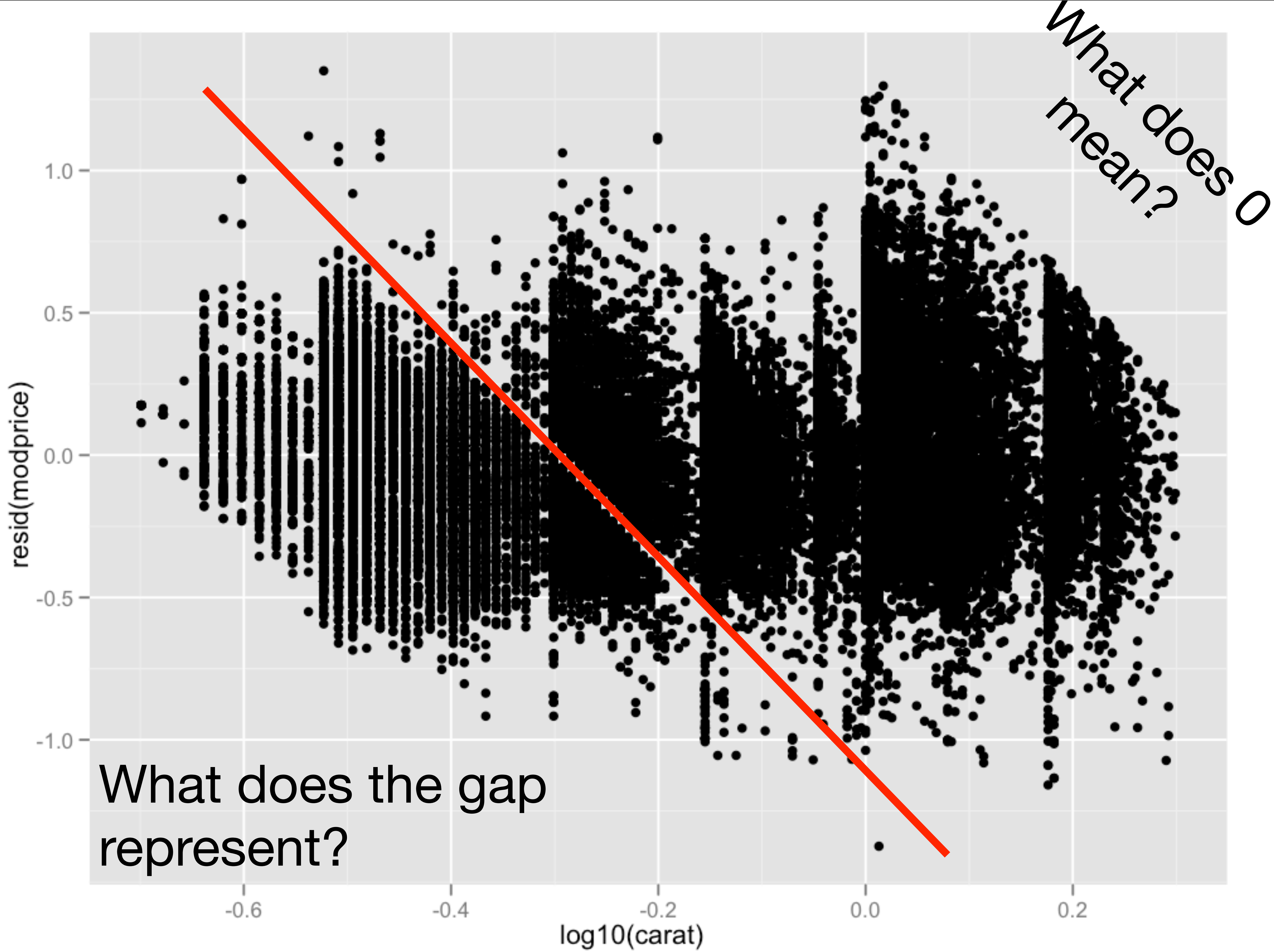
Your turn

Take two minutes to brainstorm with your neighbour on how you might use this technique in a data analysis. Why is it useful?

```
qplot(log10(carat), log10(price), data = diamonds) +
  lm_line

modprice <- lm(log(price) ~ log(carat),
  data = diamonds)

# Intercept and slope of line
coef(modprice)

# Can backtransform to interpret wrt original data
# ln(y) = a + b ln(x)
#    y  = exp(a) x^b
exp(coef(modprice)[1])
```

```
qplot(log10(carat), resid(modprice),
  data = diamonds)

diamonds$price2 <- resid(modprice)
```


What does 0 mean?
What does the gap represent?
resid(modprice)
1.0
0.5
0.0
-0.5
-1.0
-0.6
-0.4
-0.2
0.0
0.2
log10(carat)



What does 0 mean?
1.0
0.5
0.0
-0.5
-1.0
resid(modprice)
-0.6
-0.4
-0.2
0.0
0.2
log10(carat)
What does the gap represent?

Residuals

```
resid(modprice) =
  log(price) - predict(modprice)

exp(resid(modprice)) =
  price / exp(predict(modprice))
```

```
qplot(log10(carat), price2,
  data = diamonds)

qplot(log10(carat), price2, data = diamonds,
  colour = color)

qplot(log10(carat), price2, data = diamonds) +
  facet_wrap(~ color) +
  geom_hline(yintercept = 0, colour = "red")
```

Your turn

Out of colour, cut and clarity, which has the strongest effect on price?

What does 0 mean?

```
qplot(log10(carat), price2, data = diamonds) +
  facet_wrap(~ color) +
  geom_hline(yintercept = 0, colour = "red")

qplot(log10(carat), price2, data = diamonds) +
  facet_wrap(~ clarity) +
  geom_hline(yintercept = 0, colour = "red")

qplot(log10(carat), price2, data = diamonds) +
  facet_wrap(~ cut) +
  geom_hline(yintercept = 0, colour = "red")
```

What does 1 mean?

```
options(digits = 3)

ddply(diamonds, "cut", summarise,
  effect = mean(exp(price2)))
ddply(diamonds, "color", summarise,
  effect = mean(exp(price2)))
ddply(diamonds, "clarity", summarise,
  effect = mean(exp(price2)))

# Clarity appears to have the biggest effect
```

```
means <- ddply(diamonds, c("clarity", "color"),
  summarise, effect = mean(exp(price2)))

qplot(clarity, effect, data = means, colour = color) +
  geom_line(aes(group = color))

qplot(color, effect, data = means, colour = clarity) +
  geom_line(aes(group = clarity))
```

Group means

```
# If the x variable is a factor, lm models group
# means

# The "intercept" is first level in the factor.
# All other values are relative to that
lm(exp(price2) ~ clarity, data = diamonds)

# Removing intercept makes coefficients easier to
# interpret.  (But predictions/residuals the same)
lm(exp(price2) ~ clarity - 1, data = diamonds)

# Exactly equivalent to ddply results
ddply(diamonds, "clarity", summarise,
   effect = mean(exp(price2)))
```