Heteroskedasticity

EC 339

Marcio Santetti

Fall 2022

Motivation

The road so far

- Over the past three weeks, we have learned:
  - That omitting relevant variables from a model causes **bias**;
  - That deterministic/strong stochastic linear relationships between two independent variables harm regression **standard errors**, and, therefore, OLS **inference**;
  - That if the *error term* shows linear relationships across its own observations, OLS standard errors will be affected, also harming **inference**.
- This week, we will study the last violation of CLRM Assumptions: **Heteroskedasticity**.

Defining heteroskedasticity

Defining heteroskedasticity

Recall **CLRM Assumption V**:

> *"The error term has a constant variance."*

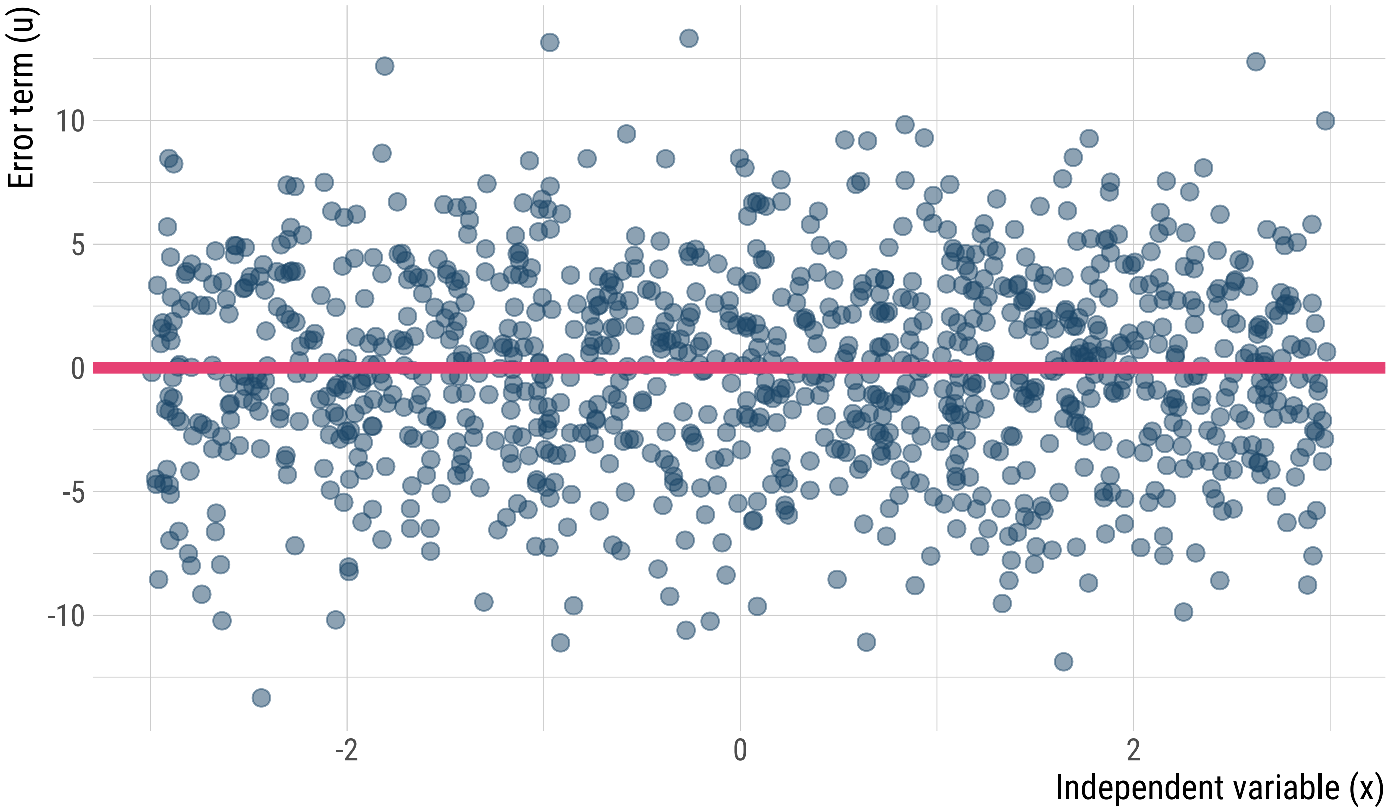
Mathematically...

$$Var(u|x) = \sigma^2$$

In words, this assumption implies that the error term has the **same variance** for each value of the independent variable.

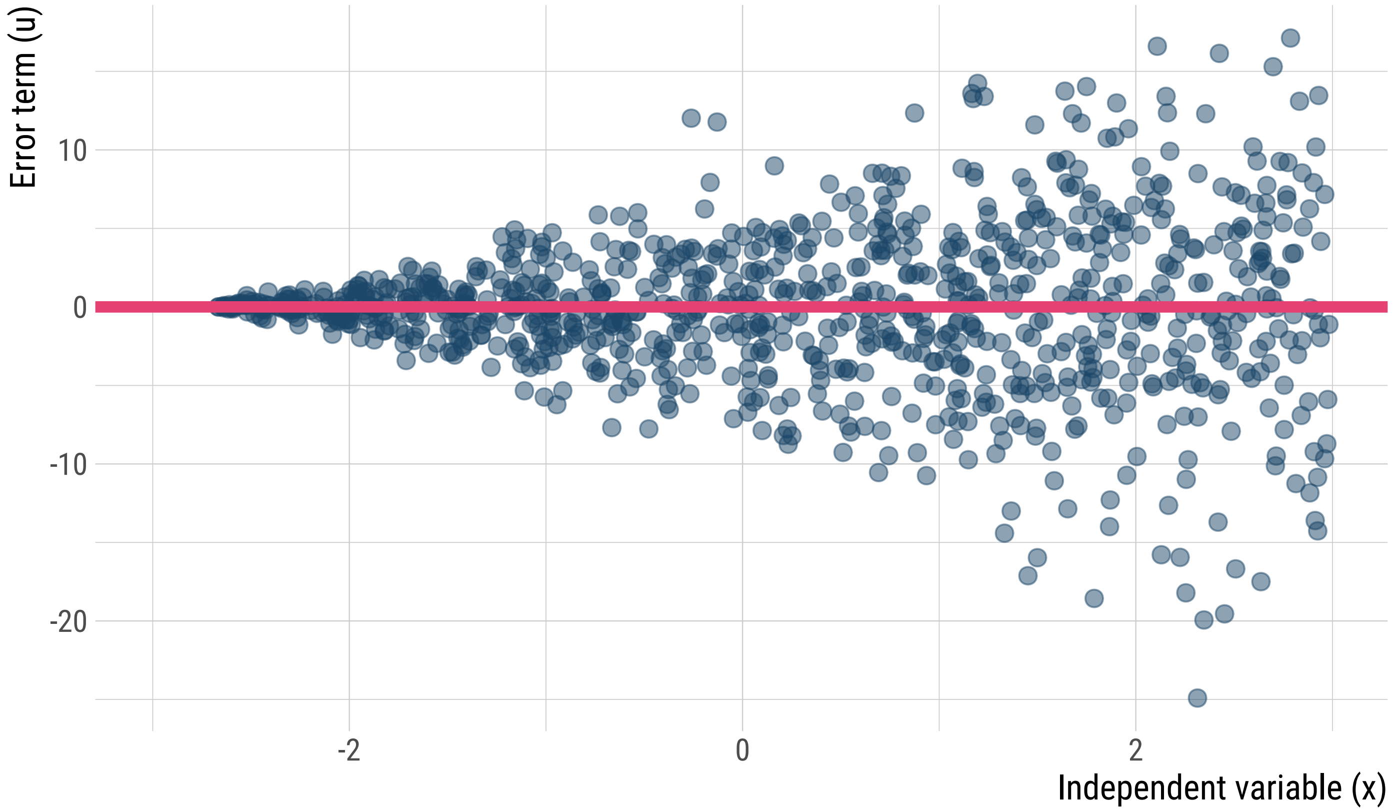
Defining heteroskedasticity

- *Homoskedastic* residuals:



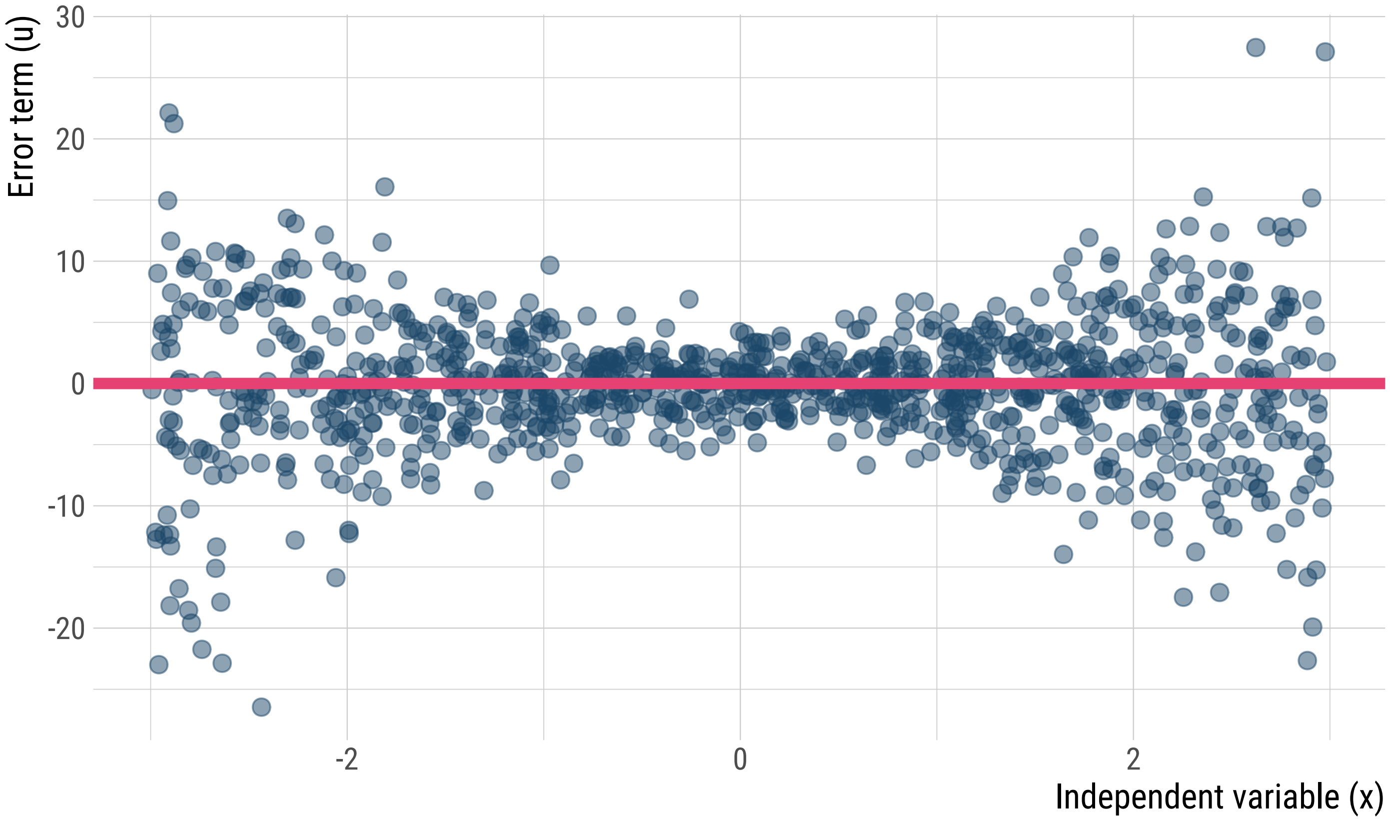
Defining heteroskedasticity

- *Heteroskedastic* residuals (1):



Defining heteroskedasticity

- *Heteroskedastic* residuals (2):



Consequences of heteroskedasticity

Consequences of heteroskedasticity

First of all, heteroskedasticity **does not** cause bias to OLS coefficients.

Similar to multicollinearity and serial correlation, heteroskedasticity affects OLS **standard errors**.

As a consequence, confidence intervals and hypothesis testing procedures become **unreliable**.

Therefore, how can we trust in our models' **inference**?

We **can't**!

Testing for heteroskedasticity

Testing for heteroskedasticity

Here, we will study **two** different statistical tests for heteroskedasticity.

- The **Breusch-Pagan** test;
- The **White** test.

We will study these procedures through an example.

The Breusch-Pagan test

As we have been studying for the past few weeks, all statistical tests involve **auxiliary regression models**.

For the **Breusch-Pagan** test, this is also the case. This time, it involves the regression's **squared residuals**.

The **recipe** 👩‍🍳 👨‍🍳:

1. Estimate the regression model via OLS, storing its residuals;

2. Square the estimated residuals, obtaining $\hat{u}_i^2$;

3. Estimate an *auxiliary regression*, with $\hat{u}_i^2$ as the dependent variable, on *all* independent variables from the original model;

4. Then, test the following *null hypothesis*:

H_0: CLRM Assumption V is true
H_a: H_0 is not true

The Breusch-Pagan test

The Breusch-Pagan test's **test statistic** is given by

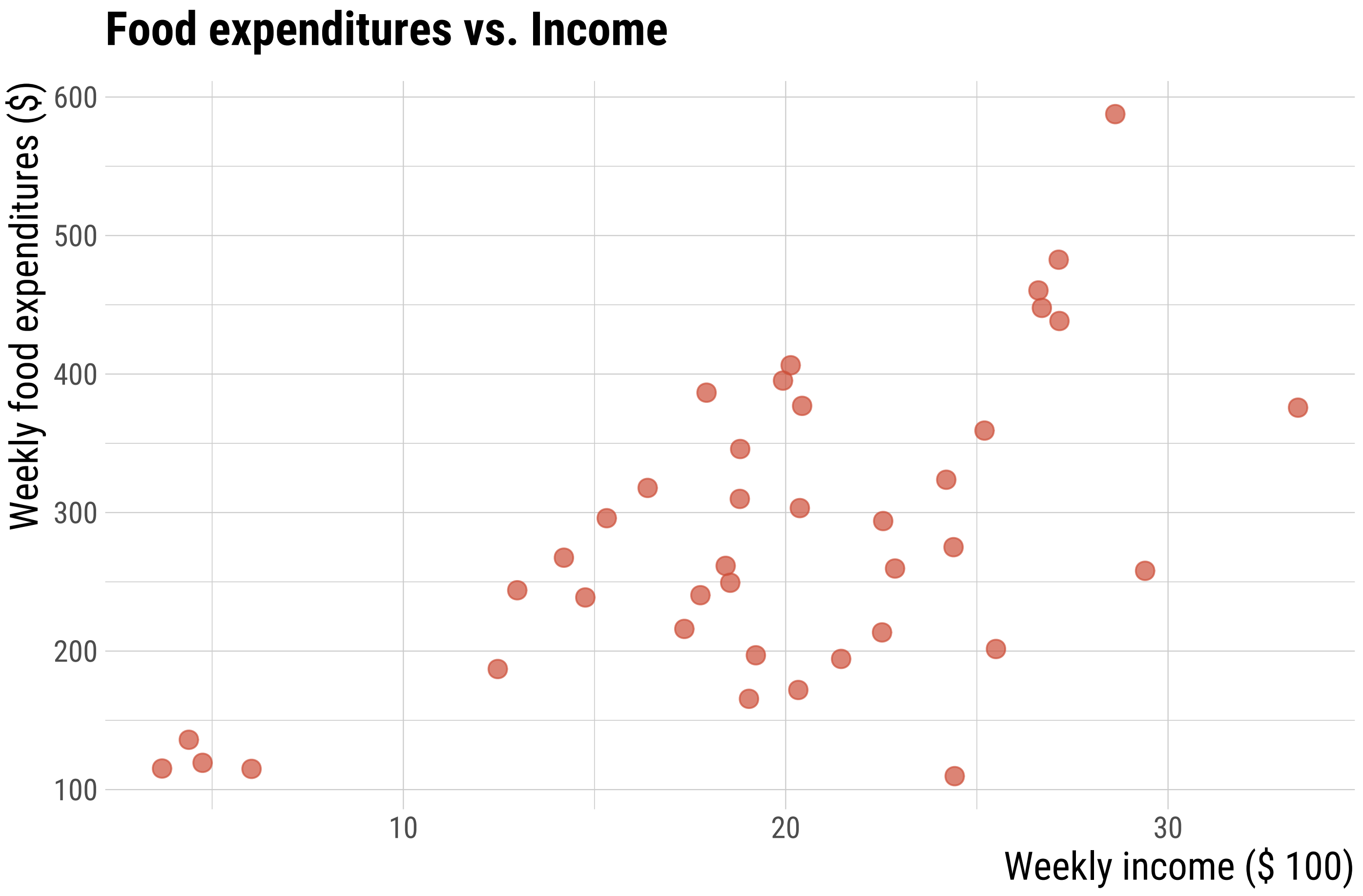
$$LM = n \cdot R^2_{\hat{u}^2}$$

Where n is the sample size, and $R^2_{\hat{u}^2}$ is the coefficient of determination from the auxiliary regression.

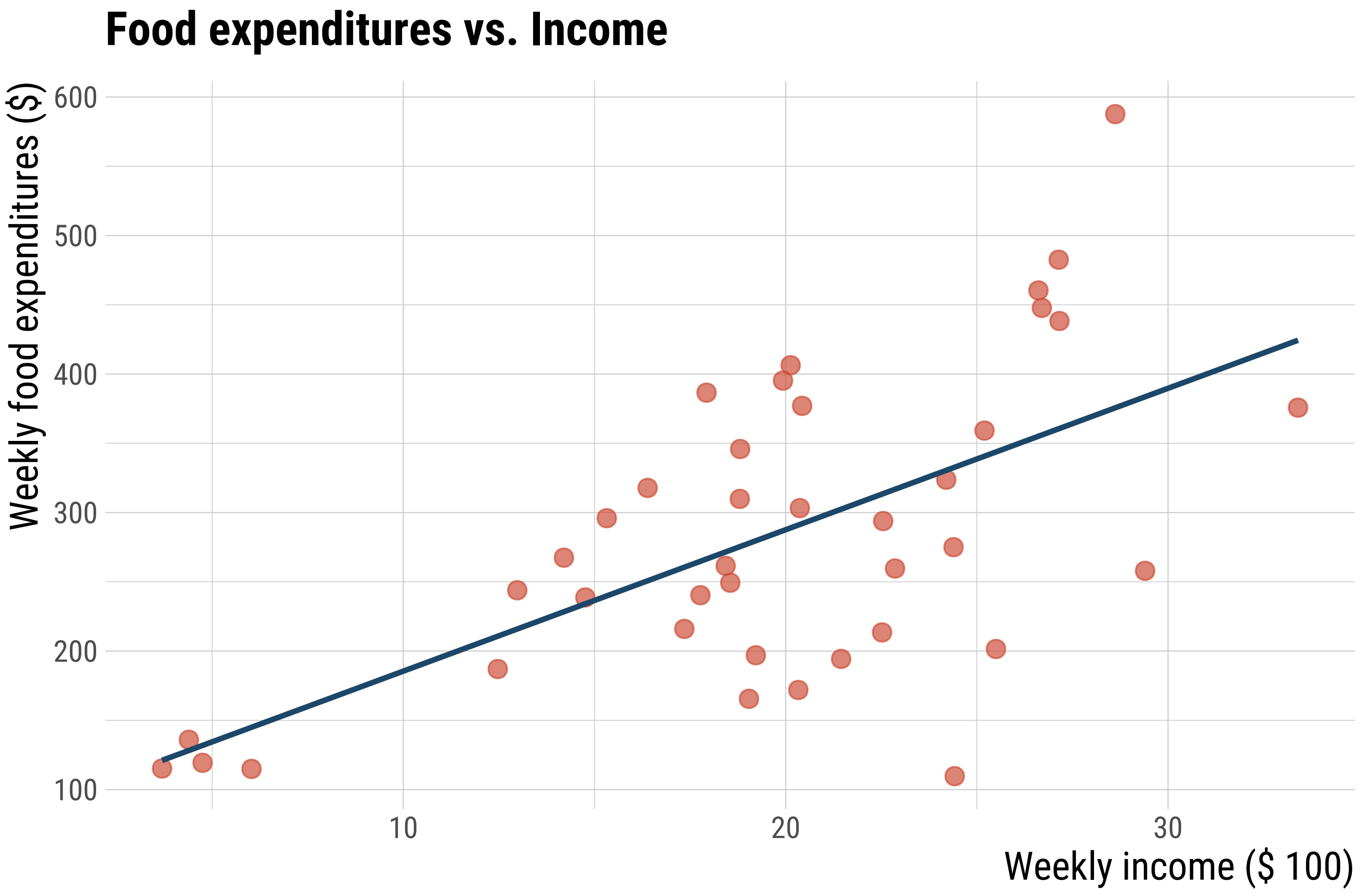
This LM test statistic is Chi-squared distributed, with k degrees-of-freedom.

In case we **reject** the null hypothesis, CLRM Assumption V is **violated** and we have **evidence** of heteroskedasticity in the model's residuals.

The Breusch-Pagan test



The Breusch-Pagan test



The Breusch-Pagan test

In R...

```
food_model <- lm(food_exp ~ income, data = food_data)
food_model %>% tidy()
```

```
#> # A tibble: 2 × 5
#>   term        estimate std.error statistic   p.value
#>   <chr>          <dbl>     <dbl>     <dbl>     <dbl>
#> 1 (Intercept)     83.4     43.4       1.92 0.0622
#> 2 income          10.2      2.09      4.88 0.0000195
```

```
food_model %>% breusch_pagan()
```

```
#> # A tibble: 1 × 5
#>   statistic p.value parameter method                alternative
#>       <dbl>   <dbl>     <dbl> <chr>                 <chr>
#> 1      7.38 0.00658         1 Koenker (studentised) greater
```

What is our **inference**?

The Breusch-Pagan test

In `Stata`...

```
. reg food_exp income

      Source |       SS           df       MS      Number of obs   =        40
-------------+----------------------------------   F(1, 38)        =     23.79
       Model |   190626.98         1   190626.98   Prob > F        =    0.0000
    Residual |  304505.173        38  8013.29403   R-squared       =    0.3850
-------------+----------------------------------   Adj R-squared   =    0.3688
       Total |  495132.153        39  12695.6962   Root MSE        =    89.517

------------------------------------------------------------------------------
    food_exp | Coefficient  Std. err.      t    P>|t|     [95% conf. interval]
-------------+----------------------------------------------------------------
      income |   10.20964   2.093263     4.88   0.000     5.972052    14.44723
       _cons |   83.41601   43.41016     1.92   0.062    -4.463272    171.2953
------------------------------------------------------------------------------
```

The Breusch-Pagan test

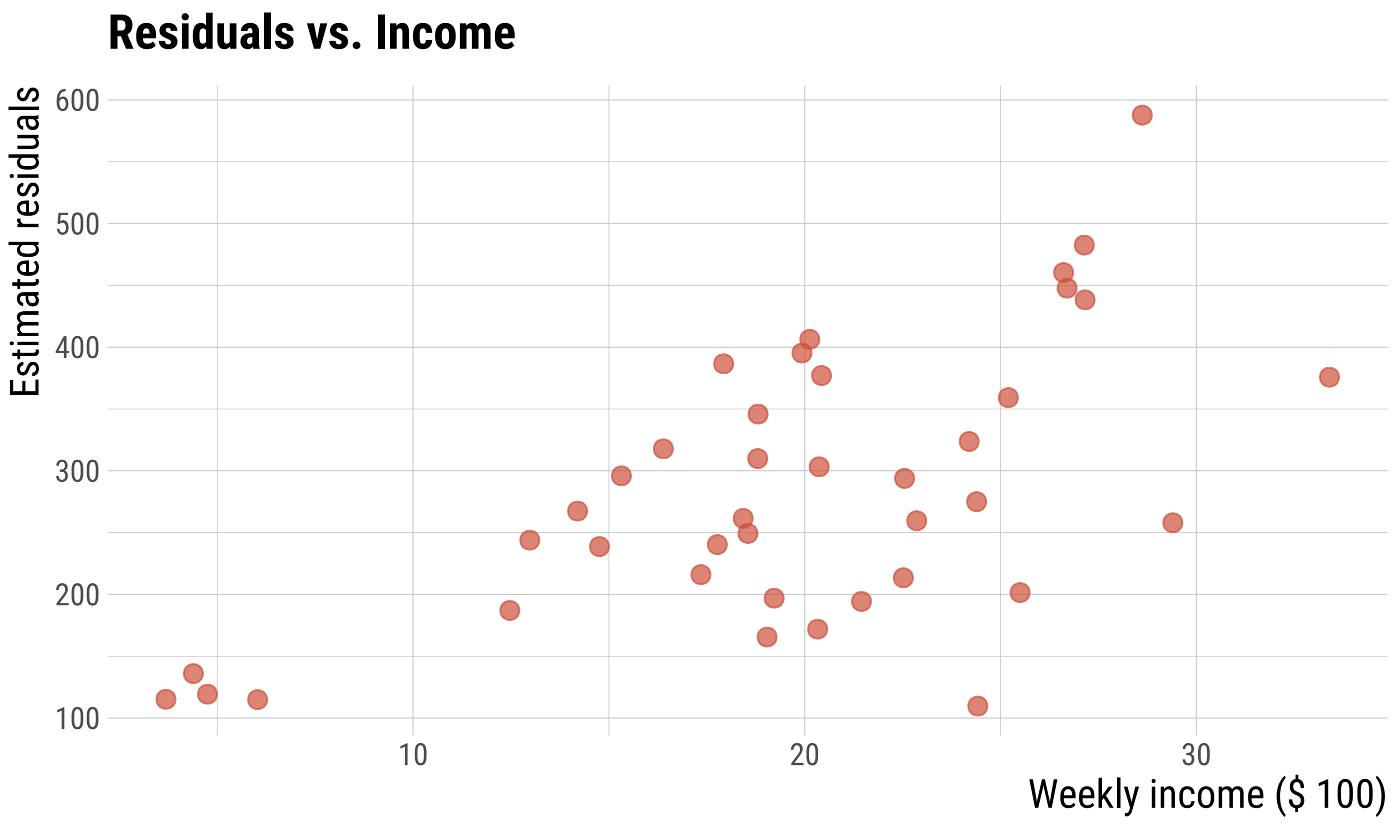
In Stata…

```
. estat hettest, iid

Breusch–Pagan/Cook–Weisberg test for heteroskedasticity
Assumption: i.i.d. error terms
Variable: Fitted values of food_exp

H0: Constant variance

    chi2(1) =   7.38
Prob > chi2 = 0.0066
```

What is our **inference**?

The Breusch-Pagan test

A quick look at this model's **residuals**:



The Breusch-Pagan test

Sometimes, a solution for heteroskedasticity is to **log-transform** the dependent variable.

- Why?

- It reduces the variable's **variance**.

Let's see.

```
food_model2 <- lm(log(food_exp) ~ income, data = food_data)
food_model2 %>% breusch_pagan()
```

```
#> # A tibble: 1 × 5
#>   statistic p.value parameter method                alternative
#>       <dbl>   <dbl>     <dbl> <chr>                 <chr>
#> 1      1.71   0.191         1 Koenker (studentised) greater
```

What happened?

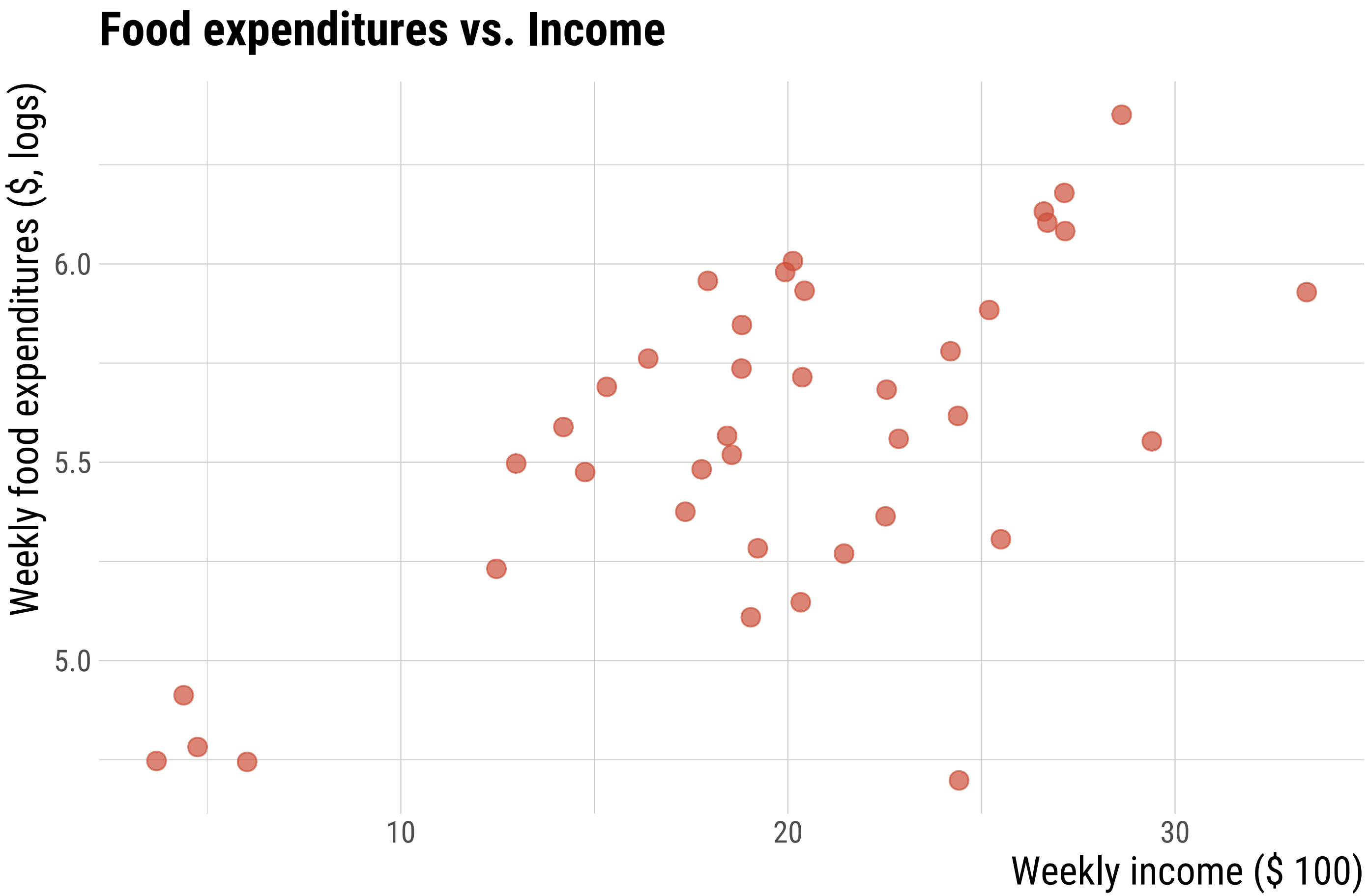
The Breusch-Pagan test

Sometimes, a solution for heteroskedasticity is to **log-transform** the dependent variable.

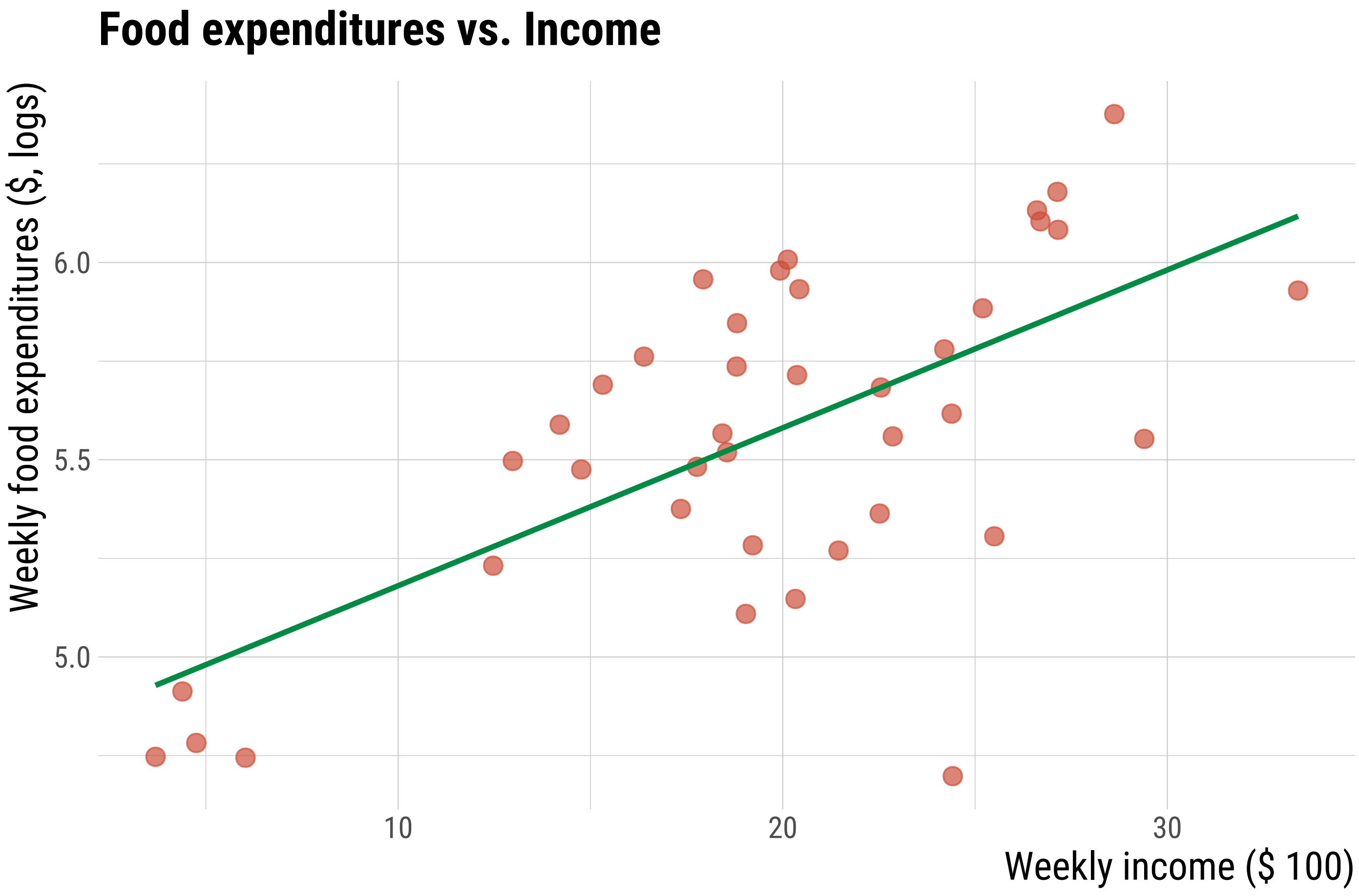
```
. quietly reg log_food_exp income

.
.
. estat hettest, iid

Breusch–Pagan/Cook–Weisberg test for heteroskedasticity
Assumption: i.i.d. error terms
Variable: Fitted values of log_food_exp

H0: Constant variance

    chi2(1) =   1.71
Prob > chi2 = 0.1909
```

What happened?

The Breusch-Pagan test



The Breusch-Pagan test



The White test

The **White test** for heteroskedasticity is a more **general form** of the Breusch-Pagan test.

Basically, it allows $\hat{u}^2$ to be correlated with further functional forms of the independent variables, such as squares, cubes, interactions, etc.

The **recipe** 👩‍🍳 👨‍🍳:

1. Run steps 1 and 2 from the Breusch-Pagan test;

2. Estimate an *auxiliary regression*, with $\hat{u}_i^2$ as the dependent variable, on *all* independent variables from the original model and desired functional forms;

3. Then, test the following *null hypothesis*:

H_0: CLRM Assumption V is true
H_a: H_0 is not true

The White test

Now, let's **apply** this test to our food expenditure models:

- Original model (with *food expenditures* in levels):

```r
food_model %>% white_lm(interactions = TRUE)
```

```
#> # A tibble: 1 × 5
#>   statistic p.value parameter method        alternative
#>       <dbl>   <dbl>     <dbl> <chr>         <chr>
#> 1      7.56  0.0229         2 White's Test greater
```

What is our **inference**?

The White test

Now, let's **apply** this test to our food expenditure models:

- Original model (with *food expenditures* in levels):

```
. estat imtest, white

White's test
H0: Homoskedasticity
Ha: Unrestricted heteroskedasticity

    chi2(2) =   7.56
Prob > chi2 = 0.0229
```

What is our **inference**?

The White test

- Now, with *food expenditures* in *logs*:

```
food_model2 %>% white_lm(interactions = TRUE)
```

```
#> # A tibble: 1 × 5
#>   statistic p.value parameter method       alternative
#>       <dbl>   <dbl>     <dbl> <chr>        <chr>
#> 1      1.76   0.416         2 White's Test greater
```

And **now**?

The White test

- Now, with *food expenditures* in *logs*:

```
. estat imtest, white

White's test
H0: Homoskedasticity
Ha: Unrestricted heteroskedasticity

    chi2(2) =   1.76
Prob > chi2 = 0.4156
```

And **now**?

Robust standard errors

Robust standard errors

Many times, however, log-transforming variables **does not** guarantee that heteroskedasticity will go away.

A nice solution is to use **heteroskedasticity-robust standard errors**.

By estimating these robust standard errors, we correct the **bias** in a model's standard errors, therefore improving **inference** from our models.

Robust standard errors

Consider the following model:

```r
data("hprice2")

price_model <- lm(lprice ~ lnox + log(dist) + rooms + stratio, data = hprice2)
price_model %>% tidy()
```

```
#> # A tibble: 5 × 5
#>   term        estimate std.error statistic   p.value
#>   <chr>          <dbl>     <dbl>     <dbl>     <dbl>
#> 1 (Intercept)  11.1      0.318       34.8  5.65e-136
#> 2 lnox         -0.954    0.117       -8.17 2.57e- 15
#> 3 log(dist)    -0.134    0.0431      -3.12 1.93e-  3
#> 4 rooms         0.255    0.0185      13.7  1.15e- 36
#> 5 stratio      -0.0525   0.00590     -8.89 1.07e- 17
```

Robust standard errors

Consider the following model:

```
. reg lprice lnox log_dist rooms stratio

      Source |       SS           df       MS      Number of obs   =       506
-------------+----------------------------------   F(4, 501)       =    175.86
       Model |  49.3987586         4  12.3496897   Prob > F        =    0.0000
    Residual |  35.1834663       501   .07022648   R-squared       =    0.5840
-------------+----------------------------------   Adj R-squared   =    0.5807
       Total |   84.582225       505  .167489554   Root MSE        =      .265

------------------------------------------------------------------------------
      lprice | Coefficient  Std. err.      t    P>|t|     [95% conf. interval]
-------------+----------------------------------------------------------------
        lnox |  -.9535388   .1167417    -8.17   0.000    -1.182902   -.7241751
    log_dist |  -.1343395   .0431032    -3.12   0.002    -.2190247   -.0496542
       rooms |   .2545271   .0185303    13.74   0.000     .2181203    .2909338
     stratio |  -.0524511   .0058971    -8.89   0.000    -.0640372    -.040865
       _cons |   11.08386   .3181113    34.84   0.000     10.45887    11.70886
------------------------------------------------------------------------------
```

Robust standard errors

Breusch-Pagan test:

```
price_model %>% breusch_pagan()
```

```
#> # A tibble: 1 × 5
#>   statistic  p.value parameter method                    alternative
#>       <dbl>    <dbl>     <dbl> <chr>                     <chr>
#> 1      69.9 2.42e-14         4 Koenker (studentised) greater
```

White test:

```
price_model %>% white_lm(interactions = TRUE)
```

```
#> # A tibble: 1 × 5
#>   statistic  p.value parameter method       alternative
#>       <dbl>    <dbl>     <dbl> <chr>        <chr>
#> 1      144. 1.15e-23        14 White's Test greater
```

Robust standard errors

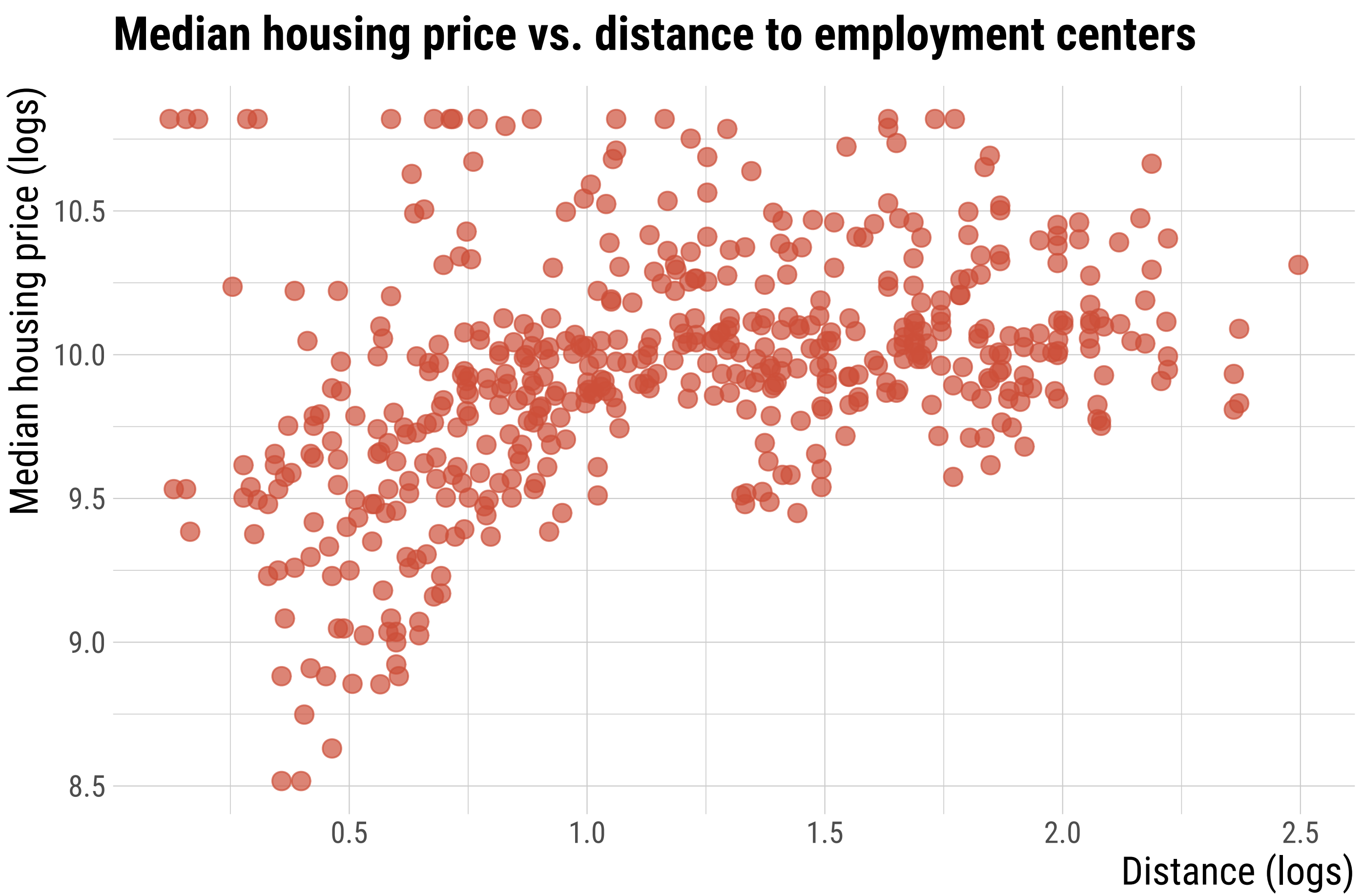
Breusch-Pagan test:

```
. estat hettest, iid

Breusch–Pagan/Cook–Weisberg test for heteroskedasticity
Assumption: i.i.d. error terms
Variable: Fitted values of lprice

H0: Constant variance

    chi2(1) =  37.57
Prob > chi2 = 0.0000
```

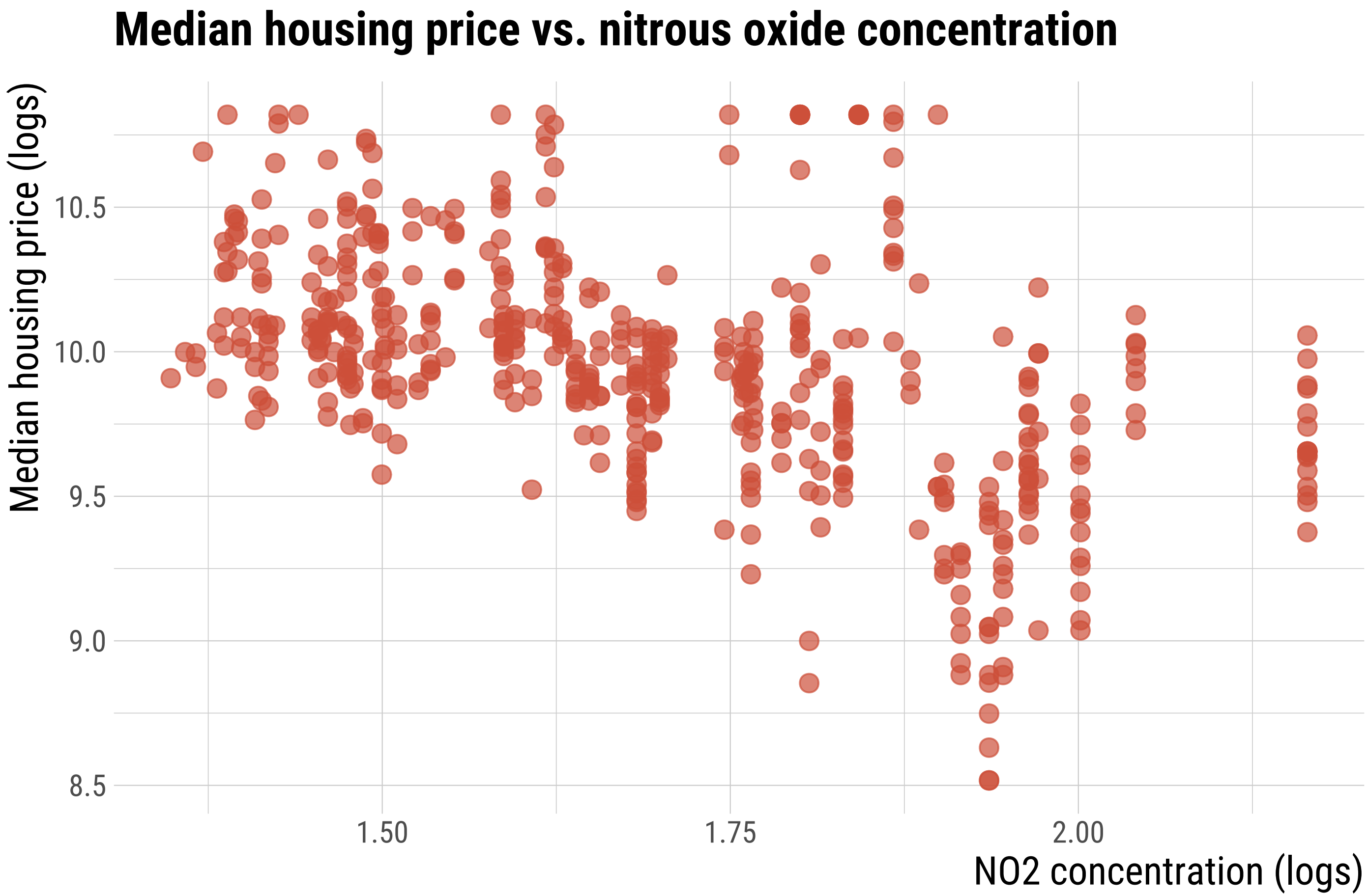
White test:

```
. estat imtest, white

White's test
H0: Homoskedasticity
Ha: Unrestricted heteroskedasticity

   chi2(14) = 143.98
Prob > chi2 = 0.0000
```

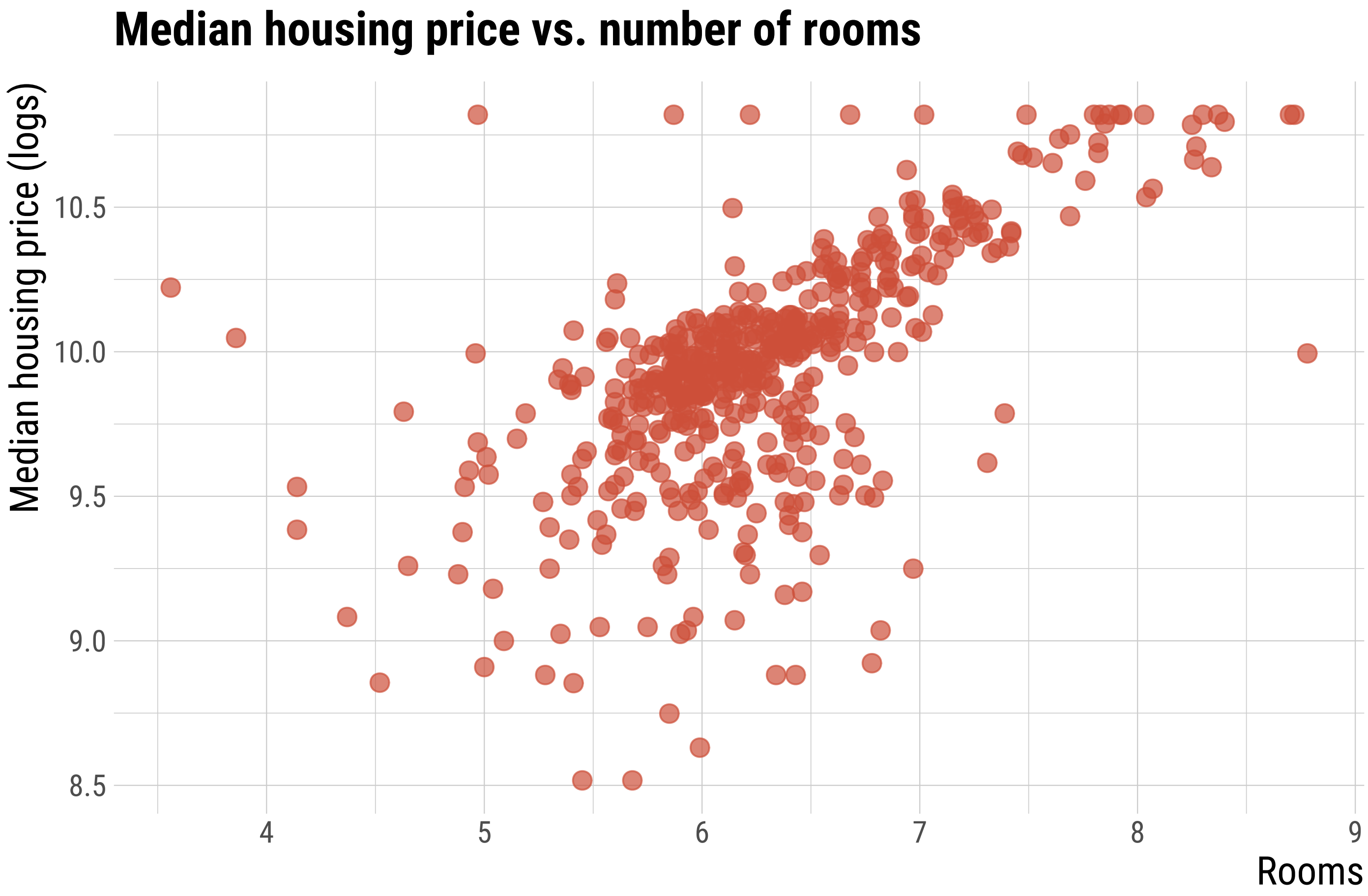
Robust standard errors



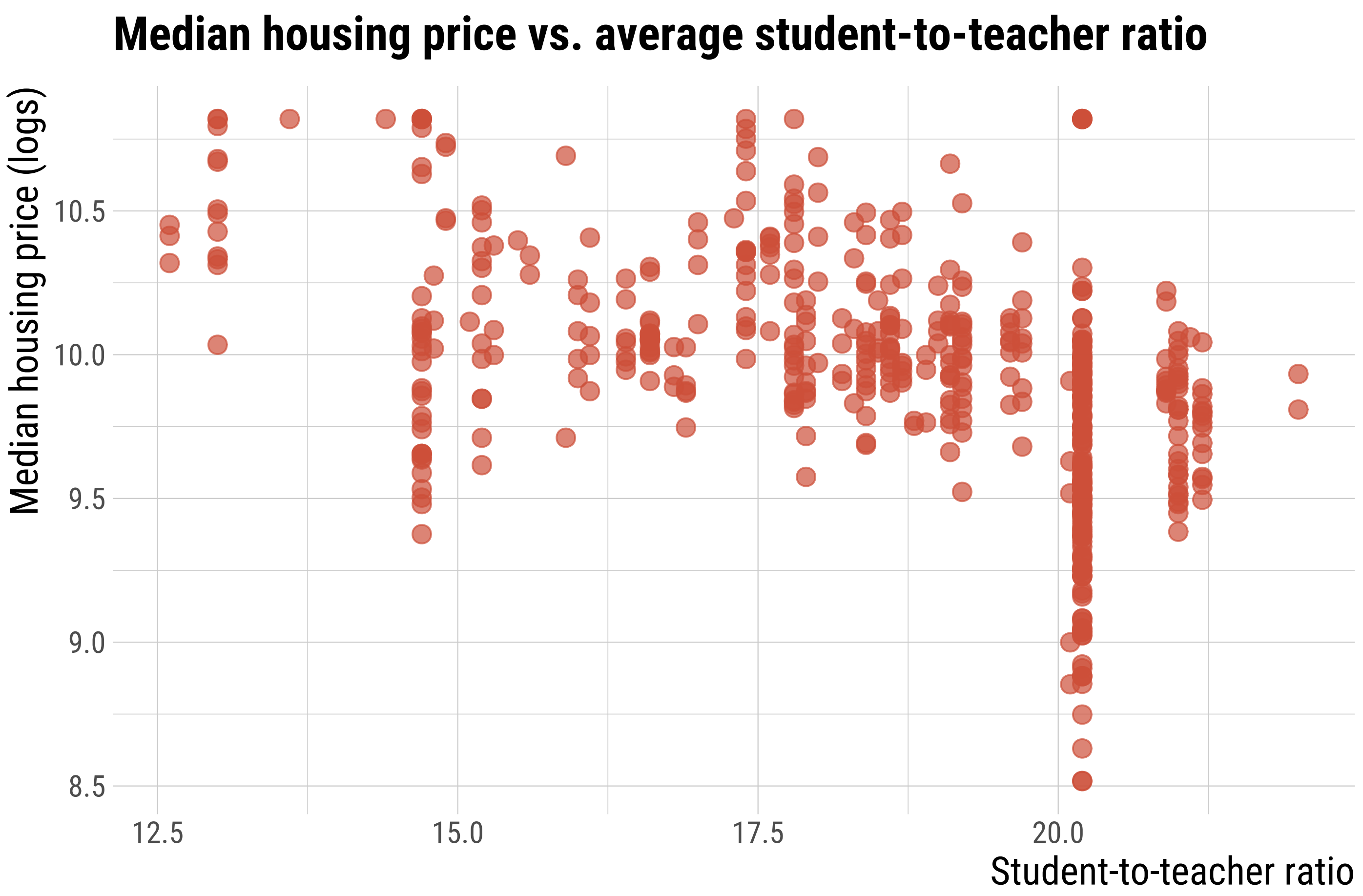
Robust standard errors



Robust standard errors



Robust standard errors



Robust standard errors

Robust (White) standard errors:

```
lm_robust(lprice ~ lnox + log(dist) + rooms + stratio, data = hprice2,
          se_type="HC1")
```

Variable	Coefficient	Standard error	t-statistic	p-value
(Intercept)	11.0838616	0.3772949	29.3771817	0.0000000
lnox	-0.9535388	0.1268005	-7.5199909	0.0000000
log(dist)	-0.1343395	0.0535287	-2.5096731	0.0123986
rooms	0.2545271	0.0247205	10.2962139	0.0000000
stratio	-0.0524511	0.0046082	-11.3821438	0.0000000

Robust standard errors

Robust (White) standard errors:

```
. reg lprice lnox log_dist rooms stratio, robust

Linear regression                               Number of obs     =        506
                                                F(4, 501)         =     146.27
                                                Prob > F          =     0.0000
                                                R-squared         =     0.5840
                                                Root MSE          =       .265

------------------------------------------------------------------------------
             |               Robust
      lprice | Coefficient  std. err.      t    P>|t|     [95% conf. interval]
-------------+----------------------------------------------------------------
        lnox |  -.9535388   .1268005    -7.52   0.000    -1.202665   -.7044125
    log_dist |  -.1343395   .0535287    -2.51   0.012    -.2395078   -.0291711
       rooms |   .2545271   .0247205    10.30   0.000     .2059585    .3030956
     stratio |  -.0524511   .0046082   -11.38   0.000    -.0615049   -.0433974
       _cons |   11.08386   .3772949    29.38   0.000     10.34259    11.82514
------------------------------------------------------------------------------
```

Robust standard errors

In **summary**, whenever interpreting a model with **heteroskedastic** residuals, use **robust standard errors** for inference purposes.

Otherwise, any inferential analysis from our models will not be valid, since violating **CLRM Assumption V** directly affects OLS standard errors.

Next time: Heteroskedasticity in practice