

## Chapter 8: heteroskedasticity

We start again by loading the data and filtering it so that we restrict it to properties that host at most 6 people. Also, as in Chapter 3, we again create a variable `review_scores_rating_standardized` with the standardized review score.

```
# Load the datasets from the RData file
load("../dataCreated/listings_clean.RData")

listings_clean_filtered <- listings_clean %>%
  filter(accommodates <= 6)

# Standardize review_scores_rating
listings_clean_filtered <- listings_clean_filtered %>%
  mutate(review_scores_rating_standardized =
    (review_scores_rating - mean(review_scores_rating, na.rm = TRUE)) /
    sd(review_scores_rating, na.rm = TRUE))
```

Our point of departure for Chapter 8 is, as for Chapter 4, the richer model from Chapter 3 where we regress the log price on `review_scores_rating`, `accommodates`, and 4 neighborhood characteristics.

```
estimatesFullModel <- lm(log(price) ~ review_scores_rating_standardized + accommodates + Centrality + Q
summary(estimatesFullModel)
```

```
##
## Call:
## lm(formula = log(price) ~ review_scores_rating_standardized +
##     accommodates + Centrality + Quietness + Coolness + Fanciness,
##     data = listings_clean_filtered)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.48845 -0.24997  0.01026  0.23530  0.94151
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)      3.10015    0.49275   6.292   8e-10 ***
## review_scores_rating_standardized  0.05000    0.01887   2.650  0.00837 **
## accommodates      0.21641    0.01480  14.625 < 2e-16 ***
## Centrality        0.08591    0.04406   1.950  0.05186 .
## Quietness         0.12896    0.04451   2.897  0.00397 **
## Coolness          0.09606    0.07834   1.226  0.22079
## Fanciness        -0.14765    0.05382  -2.744  0.00634 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.3834 on 414 degrees of freedom
## Multiple R-squared:  0.3494, Adjusted R-squared:  0.34
## F-statistic: 37.05 on 6 and 414 DF, p-value: < 2.2e-16
```

The reported point estimates are unbiased (Chapter 3) and consistent (Chapter 5) under MLR1-MLR4. The standard errors are valid when we make the additional assumption of homoskedasticity.

Next, we test the null of homoskedasticity.

```
bptest(estimatesFullModel)
```

```
##
## studentized Breusch-Pagan test
##
## data: estimatesFullModel
## BP = 19.232, df = 6, p-value = 0.00379
```

The null of homoskedasticity is rejected.

We can directly look at results with robust standard errors.

```
coeftest(estimatesFullModel, vcov=hccm)
```

```
##
## t test of coefficients:
##
##
```

	Estimate	Std. Error	t value	Pr(> t )	
## (Intercept)	3.100154	0.495124	6.2614	9.558e-10	***
## review_scores_rating_standardized	0.049999	0.019180	2.6067	0.009471	**
## accommodates	0.216411	0.015438	14.0180	< 2.2e-16	***
## Centrality	0.085908	0.042210	2.0353	0.042462	*
## Quietness	0.128962	0.050107	2.5737	0.010408	*
## Coolness	0.096064	0.076005	1.2639	0.206973	
## Fanciness	-0.147651	0.058117	-2.5406	0.011432	*

```
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Some are smaller, some are bigger.

Next, we do weighted least squares. First, we need to estimate the residuals from the full model.

```
# Obtain residuals
residuals_full_model <- residuals(estimatesFullModel)
```

Then, we specify that

$$\text{var}(u|x_1, x_2, x_3, x_4, x_5, x_6) = \sigma^2 \cdot \exp(\delta_0 + \delta_1 x_1 + \delta_2 x_2 + \delta_3 x_4 + \delta_4 x_4 + \delta_5 x_5).$$

This means that we can estimate the parameters  $\delta_0, \dots, \delta_5$  by regressing the log of the squared residuals on the explanatory variables (see slides 21 and 23):

$$\log(\hat{u}_i^2) = \tilde{\delta}_0 + \delta_1 x_1 + \delta_2 x_2 + \delta_3 x_4 + \delta_4 x_4 + \delta_5 x_5 + w,$$

where  $\tilde{\delta}_0$  is equal to  $\log(\sigma^2) + \delta_0$ .

```
# Step 1 was to run the regression above
# Step 2: Regress squared residuals on predictors to model heteroskedasticity
log_squared_residuais <- log(residuals_full_model^2)
model_resid_squared <- lm(log_squared_residuais ~ review_scores_rating_standardized + accommodates
+ Centrality + Quietness + Coolness + Fanciness, data = listings_clean_filtered)
```

Next, we calculate the weights as the inverse of the square root of the fitted values from the regression above. We have to pay attention to also un-do taking the log of the squared residuals.<sup>1</sup>

<sup>1</sup>See also Section 6-4c in the Wooldridge book on p. 205ff on predicting the level of the dependent variable when one runs a regression that has it in logs on the left hand side. This raises an issue related to the constant term that we can however ignore here. In brief, things related to the smearing factor will only affect the constant term, which is not important here.

```
# Obtain fitted values from this regression
fitted_values <- exp(fitted(model_resid_squared))
```

```
# Step 3: Calculate weights as inverse of the square root of fitted values
weights <- 1 / sqrt(fitted_values)
```

Finally, we run the WLS regression with these weights.

```
# Step 4: Run the WLS regression with these weights
wls_model <- lm(log(price) ~ review_scores_rating_standardized + accommodates
  + Centrality + Quietness + Coolness + Fanciness,
  data = listings_clean_filtered, weights = weights)
```

```
# Summary of the WLS model
summary(wls_model)
```

```
##
## Call:
## lm(formula = log(price) ~ review_scores_rating_standardized +
##     accommodates + Centrality + Quietness + Coolness + Fanciness,
##     data = listings_clean_filtered, weights = weights)
##
## Weighted Residuals:
##      Min       1Q   Median       3Q      Max
## -4.1537 -0.5811  0.0196  0.5532  2.1320
##
## Coefficients:
##                                Estimate Std. Error t value Pr(>|t|)
## (Intercept)                   3.15303    0.49368   6.387 4.56e-10 ***
## review_scores_rating_standardized 0.05891    0.01847   3.189 0.00154 **
## accommodates                   0.20300    0.01362  14.901 < 2e-16 ***
## Centrality                    0.07760    0.04441   1.747 0.08129 .
## Quietness                     0.12370    0.04616   2.680 0.00766 **
## Coolness                      0.09661    0.08038   1.202 0.23009
## Fanciness                     -0.13630    0.05086  -2.680 0.00766 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.8678 on 414 degrees of freedom
## Multiple R-squared:  0.361, Adjusted R-squared:  0.3518
## F-statistic: 38.99 on 6 and 414 DF, p-value: < 2.2e-16
```

Finally, we use the stargazer package to show OLS results side-by-side OLS results with robust standard errors and weighted least squares results without and with robust standard errors. For the WLS estimates, in theory, one does not need them if one gets the weighting function right.

```
# Calculate robust standard errors for both OLS and WLS models
robust_se_ols <- sqrt(diag(hccm(estimatesFullModel, type = "hc3")))
robust_se_wls <- sqrt(diag(hccm(wls_model, type = "hc3")))

# Compare OLS with standard errors, OLS with robust standard errors, WLS, and WLS with robust SE
stargazer(estimatesFullModel, estimatesFullModel, wls_model, wls_model,
  se = list(NULL, robust_se_ols, NULL, robust_se_wls),
  column.labels = c("OLS", "OLS (Robust SE)", "WLS", "WLS (Robust SE)",
  column.separate = c(1, 1, 1, 1),
  type = "text", keep.stat = c("n", "rsq"))
```

```

##
## =====
##                                     Dependent variable:
##                                     -----
##                                     log(price)
##                                     OLS      OLS (Robust SE)      WLS      WLS (Robust SE)
##                                     (1)        (2)        (3)        (4)
## -----
## review_scores_rating_standardized 0.050***    0.050***    0.059***    0.059***
##                                     (0.019)    (0.019)    (0.018)    (0.017)
##
## accommodates                      0.216***    0.216***    0.203***    0.203***
##                                     (0.015)    (0.015)    (0.014)    (0.015)
##
## Centrality                        0.086*      0.086**    0.078*      0.078**
##                                     (0.044)    (0.042)    (0.044)    (0.039)
##
## Quietness                         0.129***    0.129**    0.124***    0.124**
##                                     (0.045)    (0.050)    (0.046)    (0.049)
##
## Coolness                          0.096       0.096      0.097       0.097
##                                     (0.078)    (0.076)    (0.080)    (0.075)
##
## Fanciness                        -0.148***   -0.148**   -0.136***   -0.136**
##                                     (0.054)    (0.058)    (0.051)    (0.057)
##
## Constant                         3.100***    3.100***    3.153***    3.153***
##                                     (0.493)    (0.495)    (0.494)    (0.485)
##
## -----
## Observations                      421         421         421         421
## R2                                0.349        0.349        0.361        0.361
## =====
## Note:                               *p<0.1; **p<0.05; ***p<0.01

```

Comparing column (1) and (2) shows that standard errors are almost unchanged. They sometimes even get smaller. This can in principle happen, as it does here. Usually, they get bigger. For the WLS estimates, standard errors are bigger when they are robust, but tentatively smaller than the robust OLS ones.