Linear Regression - Getting Standard Errors Right

Social Media and Web Analytics @ TiSEM

Lachlan Deer

Preliminary Draft, Last updated: 26 April, 2021

Motivation

- Recall the 6 assumptions we need for the OLS estimator to be unbiased and have the minimum variance:
    1. Our **sample** (the x_k's and y_i) was **randomly drawn** from the population.
    2. y is a **linear function** of the β_k's and u_i.
    3. There is **no perfect multicollinearity** in our sample.
    4. The explanatory variables are **exogenous**: $\boldsymbol{E}[u|X] = 0 \, (\implies \boldsymbol{E}[u] = 0)$.
    5. The disurbances have **constant variance** σ^2 and **zero covariance**, *i.e.*, - $\boldsymbol{E}\left[u_i^2|X_i\right] = \text{Var}(u_i|X_i) = \sigma^2 \implies \text{Var}(u_i) = \sigma^2$ - $\text{Cov}(u_i,\, u_j|X_i,\, X_j) = \boldsymbol{E}[u_i u_j|X_i,\, X_j] = 0$ for $i \neq j$
    6. The disturbances come from a **Normal** distribution, *i.e.*, $u_i \overset{\text{iid}}{\sim} \text{N}\left(0, \sigma^2\right)$.
- While (4) - exogeneity - is by far the most important for getting an unbiased estimate, violations of (5) will lead to misguided our statistical inference
    - Why? (5) effects the standard errors, which are the basis of hypothesis testing and confidence intervals
    - If (5) is violated, then we might be making the wrong conclusions
- This note looks at two violations of (5):
    1. Heteroskedasticity: The variance of the error term is different for different observations

$$\boldsymbol{E}\left[u_i^2\right] = \sigma_i^2$$

    2. Clustered Standard Errors: The variance of the error term is correlated across observations

$$\boldsymbol{E}[u_i u_j] \neq 0 \quad \text{for some } i \neq j$$

- Dealing with violations of (5) is an part of every day life in marketing analytics
    - We need to know what to do when we see it

Heteroskedasticity

- Problem we face: **heteroskedasticity**

$$\boldsymbol{E}\left[u_i^2\right] = \sigma_i^2$$

    - This means that the variance of the error term is different for different observations
- **Heteroskedasticity** is present when the variance of u changes with any combination of our explanatory variables
- Questions we want to answer:

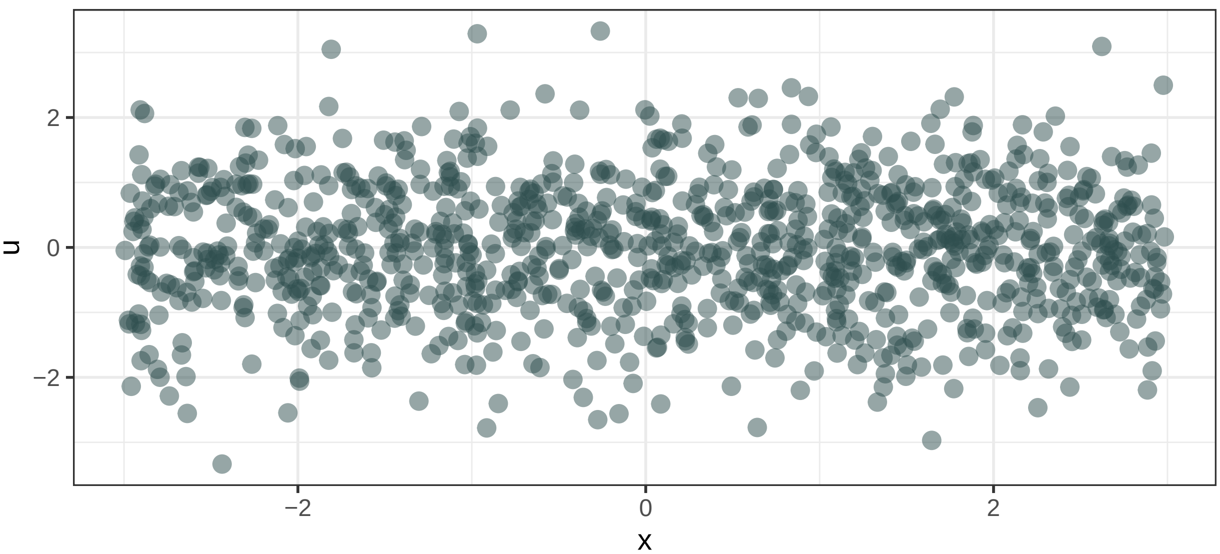
- How can we detect heteroskedasticity?
- What do we do if we detect it?

Detecting Heteroskedasticity

- Two approaches:
  1. Formal statistical tests
  2. "Eye-conometrics"
- We'll focus on "Eye-conometrics" - i.e. looking for it from visualizing data
  - It means we need to do less statistical analysis[1]
- We can visually detect if the residual,

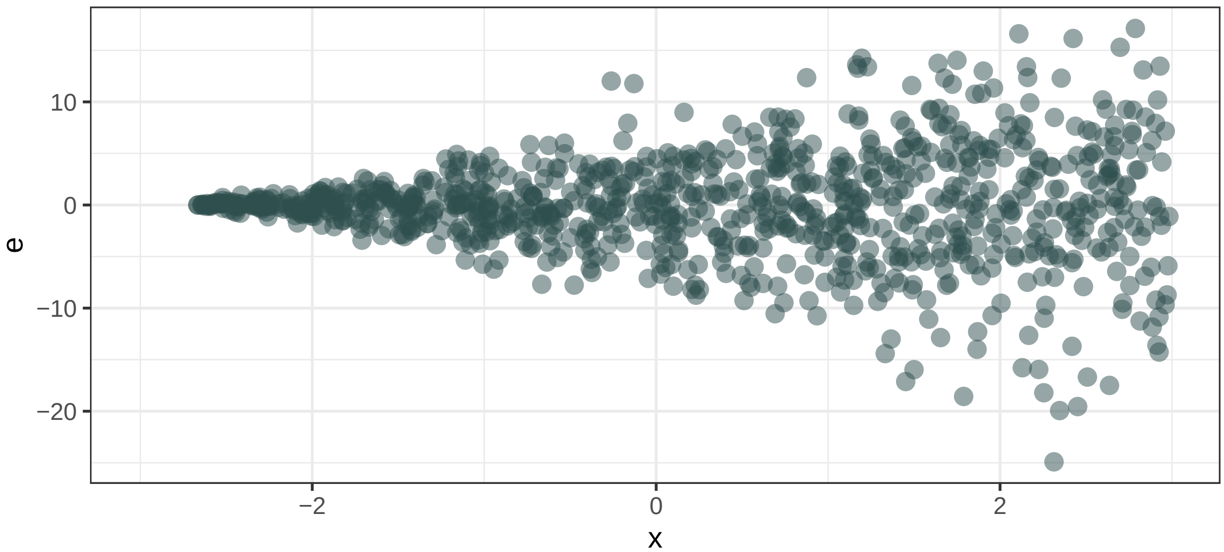
$$e_i = y_i - \hat{\beta}_0 - \hat{\beta}_1 x_{i1} - \hat{\beta}_2 x_{i2} - ...$$

  seems to look non-constant when plotted against either:
  (a) One or any of the x variables
  (b) Against the fitted values of the regression
    - Why? fitted values are just a specific combination of the x's.
- Here's what the errors should look like when there is **no heteroskedasticity**

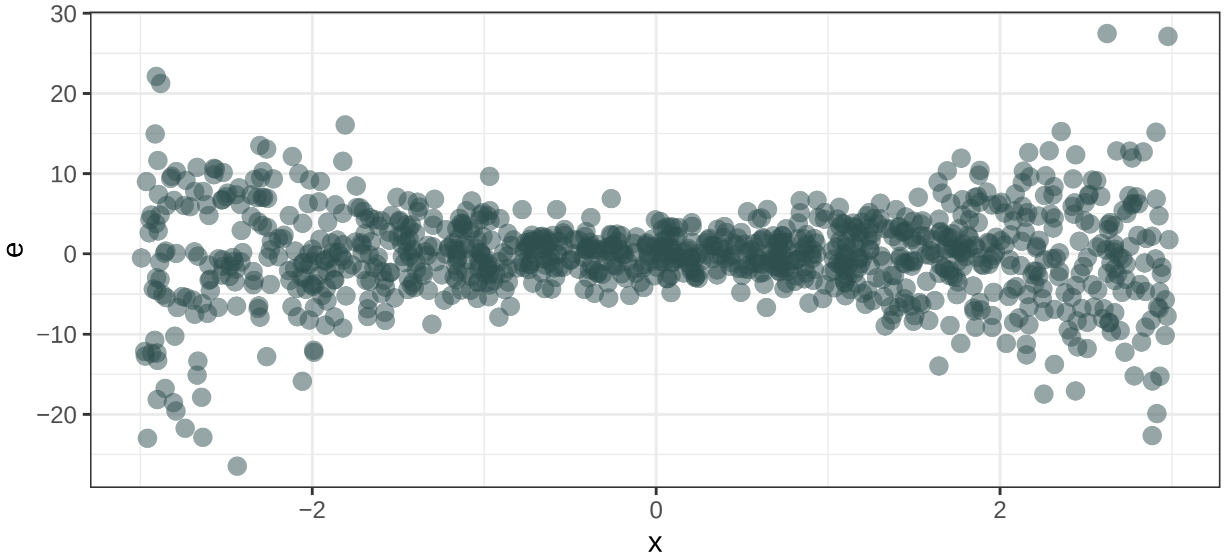


- Here's three examples of what the errors look like when there **is heteroskedasticity**:
  (a) Variance of e increases with x

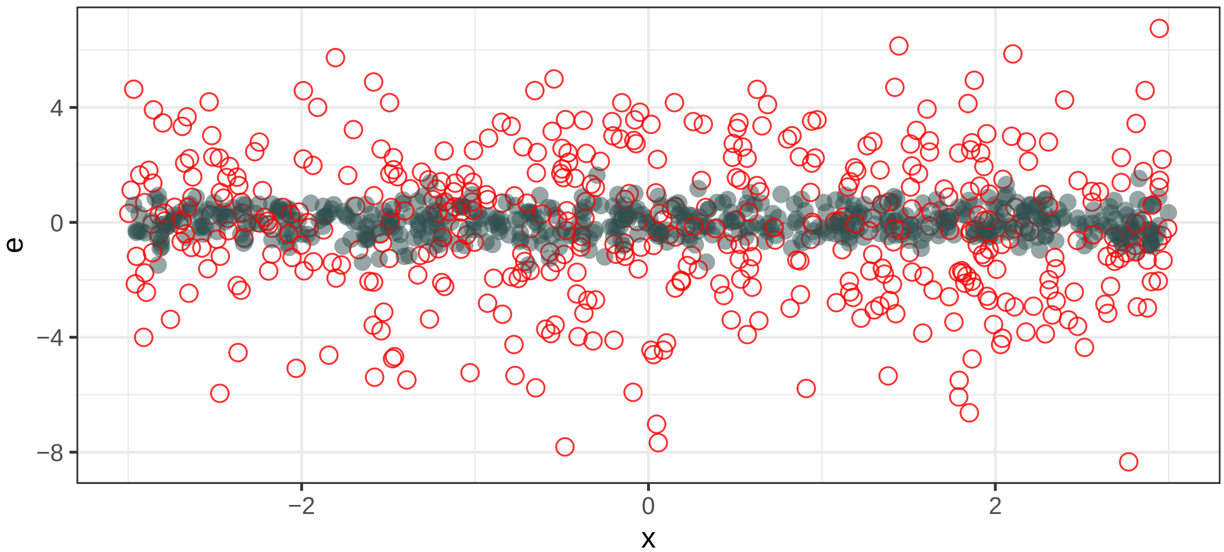
[1] Which for the purpose of this class is useful, though it is not a definitive guarantee we spot heteroskedasticity correctly.



(b) Variance of e increases at the extremes of x



(c) Variance of e differs by group



Living With Heteroskedasticity

- In the presence of heteroskedasticity:
  - The regression **coefficients are still unbiased**
  - The regression **standard errors are biased**
    - Which means confidence intervals and hypothesis tests are going to give potentially wrong conclusions
- What can we do about it?
  - pragmatic answer: find unbiased estimates for the standard errors[2]
    - Unbiased standard errors → 'correct' confidence intervals and hypothesis tests
- Pragmatic Answer: Heteroskedasticity robust standard errors
  - Essentially a different way to estimate the standard errors
  - So that they are "robust" (i.e. unbiased) when there is heteroskedasticity
- How can we do this in `R`?

Heteroskedasticity Robust Standard Errors in R

- We will use the `estimatr` package to compute heteroskedasticity robust standard errors:

```
library(estimatr)
library(broom) # to make our results look tidy
```

- Let's first download some data: from the NBA
  - i.e. basketball data from the US
  - Statistics about average player performance for each player in each year of their career (1946 - 2009)

```
url <- "https://bit.ly/3sO4hrD"

out_file <- "data/nba_data.csv"
download.file(url,
              destfile = out_file,
              mode = "wb")
```

- Read in the data and tidy it up a bit:

[2] There are other approaches, but this is the simplest and most widely used.

```r
library(readr)
# you may get "parsing failure" warnings ... ignore them
nba <- read_csv(out_file)

# clean up the data a little
nba <-
    nba %>%
    rename(
        points = pts,
        player_id = ilkid
    ) %>%
    # keep only those who played "enough" in a year
    filter(minutes > 2) %>%
    select(player_id, points, minutes)
```

- Let's run the following regression:

$$points_i = \beta_0 + \beta_1 minutes_i + u_i$$

i.e, does average points per game for a player in a given season vary depending on the number of minutes? (Likely, yes - expect β_1 to be positive)

- The 'standard' way that assumes **no heteroskedasticity**

```r
ols1 <- lm(points ~ minutes,
           data = nba)
tidy(ols1, conf.int = TRUE)
```

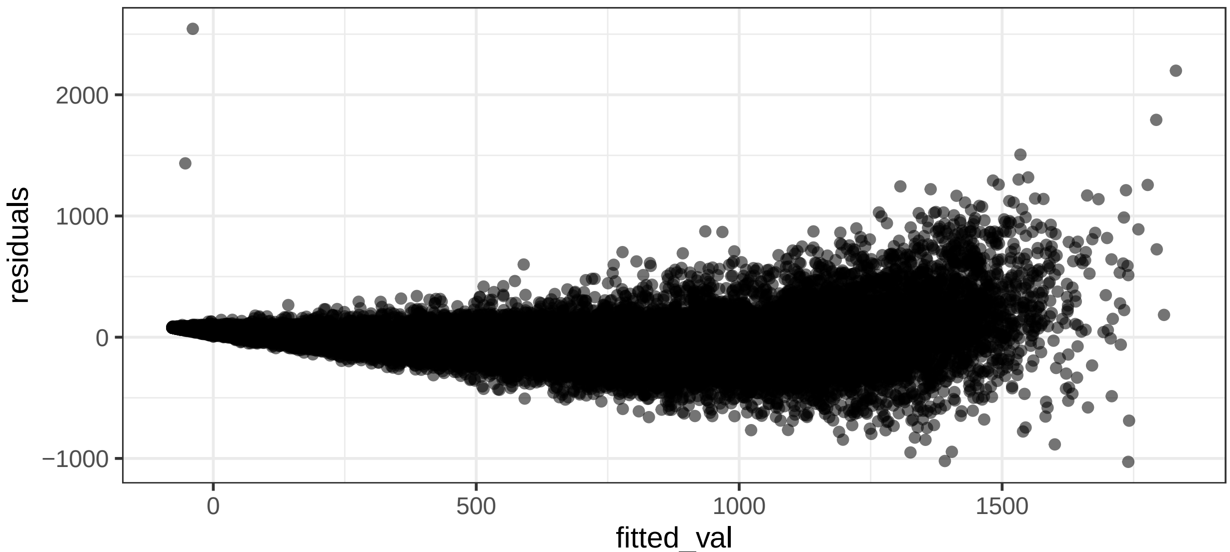
```
## # A tibble: 2 x 7
##   term        estimate std.error statistic   p.value conf.low conf.high
##   <chr>          <dbl>     <dbl>     <dbl>     <dbl>    <dbl>     <dbl>
## 1 (Intercept)  -79.3     2.22        -35.8 6.62e-272  -83.7     -75.0
## 2 minutes        0.492   0.00139     353.  0            0.489     0.495
```

- OK, thats a very small standard error…
- Is there presence of heteroskedasticity?
    - I'll check how the residuals vary with the regression fitted values
    - (you could also do this by looking at residuals vs points)

```r
library(ggplot2) # for plotting
# get residuals and fitted values
nba <-
    nba %>%
    mutate(
        residuals = resid(ols1),
        fitted_val = predict(ols1)
    )

nba %>%
    ggplot(aes(x = fitted_val,
               y = residuals,
               alpha = 0.35)
           ) +
    geom_point() +
```

```r
    theme_bw() +
    theme(legend.position = "none")
```



- Figure above shows definite evidence of a "fan" shape.
    - $\Longrightarrow$ probably heteroskedasticity
- Let's get heteroskedasticity robust standard errors. We use the `lm_robust()` function

```r
# library(estimatr) # already loaded

ols1a <- lm_robust(points ~ minutes,
                   data = nba)
tidy(ols1a, conf.int = TRUE)
```

```
##          term    estimate   std.error statistic p.value    conf.low   conf.high
## 1 (Intercept) -79.3177389 1.619594676 -48.97382       0 -82.4922703 -76.1432075
## 2     minutes   0.4919543 0.001966078 250.22115       0   0.4881006   0.4958079
##      df outcome
## 1 20864  points
## 2 20864  points
```

- Let's compare the standard error on `minutes`:
    - Assuming **no heteroskedasticity**: 0.0013922
    - Assuming **heteroskedasticity**: 0.0019661
    - $\Longrightarrow$ a 41.22 % increase in their magnitude!

Clustered Standard Errors

- Problem we face: **correlated errors** across observations

$$\boldsymbol{E}\left[u_i u_j\right] \neq 0 \quad \text{for some } i \neq j$$

    - i.e. the correlation of the error term between two observations is non-zero
    - Also called **clustered errors**
- Questions we want to answer:
    - What is clustering?
    - What to do if errors are correlated?
    - (It's hard to detect per se)

What is Clustering?

- Often, observations may share important observable and **un**observable characteristics that could influence an outcome variable
  - A sample of individuals, groups of which live in the same province
  - A sample of firms, groups of which are located in the same city
  - and so on...
- We might worry that observations in each of these groups are not independent, and that the regression error terms might be similar (or at least correlated) within the group.
- If there is within group correlation, assumption (5) of the OLS estimator fails
  - And it will impact our analysis

Living with Clustering

- The presence of clustering and its' effects are conceptually similar to when we dealt with heteroskedasticity.
- In the presence of clustered errors:
  - The regression **coefficients may be biased**
    - If we think the clustering effects do not "average out"
    - i.e. clustering might cause violations to exogeneity
    - Which means we have issues interpreting our regression coefficients
  - The regression **standard errors are biased**
    - Which means confidence intervals and hypothesis tests are going to give potentially wrong conclusions
- What can we do about it?
  - Pragmatic answer:
    - Find a way to "de-bias" the regression coefficients
      - So that we can get unbiased regression coefficients
    - Find unbiased estimates for the standard errors[3]
      - Unbiased standard errors → 'correct' confidence intervals and hypothesis tests
- Pragmatic Answer - how to do it:
  - Add Cluster-specific fixed effects to the regression
    - This will hopefully "solve" our endogeneity problem and remove any bias in our coefficients
  - Cluster robust standard errors
    - A different way to estimate the standard errors
    - So that they are "robust" (i.e. unbiased) when there is clustering
- How can we do this in `R`?
  - There will be two approaches:
    (1) Assume clustering does not cause endogeneity $\implies$ only deal with the need to adjust the standard errors
    (2) Assume clustering might be causing endogeneity $\implies$ deal with fixed effects and the need to adjust the standard errors

Cluster Robust Inference in `R`

- Again, let's work with our NBA data, and the points versus minutes regression.

[3] There are other approaches, but this is the simplest and most widely used.

  - The data are annual, and per player, so we might worry that residuals are correlated within each player

Case 1: Only Adjust the Standard Errors

- `estimatr` let's us handle clustering with the `lm_robust` function too
  - But only if there's one source of clustering … correlation within a player is probably the most important, so let's start there:

```
ols2 <- lm_robust(points ~ minutes,
                  clusters = player_id,
                  data = nba)
tidy(ols2, conf.int = TRUE)
```

```
##          term    estimate  std.error statistic       p.value    conf.low
## 1 (Intercept) -79.3177389 3.47020772 -22.85677 9.624695e-104 -86.1230327
## 2     minutes   0.4919543 0.00505906  97.24223  0.000000e+00   0.4820303
##     conf.high       df outcome
## 1 -72.5124452 2160.996  points
## 2   0.5018782 1428.585  points
```

- We see that, by **clustering the standard errors**:
  - The regression coefficient did not change
  - The standard error on minutes increases to 0.0050591
    - $\implies$ a 263.38 % increase in their magnitude!
    - That is **very substantial**

Case 2: Cluster Specific Fixed Effects

- If we think that the errors are correlated within a player and don't "average out" we have to worry about biased regression coefficients and biased standard errors[4]

- Two problems, needs two solutions:

  (1) Fixed Effects at the level of clustering
    - Helps fix out not averaging out to zero problem
    - And tries to "de-bias" the regression coefficients

  (2) Adjusting the standard errors
    - To fix the standard errors

- Easiest way to achieve this is with the `fixest` package. It allows us to estimate linear regressions with fixed effects using the `feols()` package.

- Run the regression, adding fixed effects for each player:

```
ols2a <- feols(points ~ minutes
                |
                # fixed effects for each player
                player_id,
                data = nba)
```

- Let's look at what comes out …

1. If add the fixed effects, but do not worry about making standard errors robust to clustering:

```
tidy(ols2a, se = "standard", conf.int = TRUE)
```

[4] More technically, "averaging out" would be an assumption that the effect of the clustering is zero on average. This is a relatively big assumption to make in most situations.

```
## # A tibble: 1 x 7
##   term    estimate std.error statistic p.value conf.low conf.high
##   <chr>      <dbl>     <dbl>     <dbl>   <dbl>    <dbl>     <dbl>
## 1 minutes    0.465   0.00143      325.       0    0.462     0.468
```

- Regression coefficient of `minutes` decreases to 0.47
  - And our previous estimate of the `minutes` coefficient, 0.4919543 no longer falls in the new confidence interval

2. If we add the effects **and** correct the standard errors for clustering:

```
# by default, feols clusters std errors by the first fixed effect,
# we only have one, so that is by player_id
tidy(ols2a, se = "cluster", conf.int = TRUE)
```

```
## # A tibble: 1 x 7
##   term    estimate std.error statistic p.value conf.low conf.high
##   <chr>      <dbl>     <dbl>     <dbl>   <dbl>    <dbl>     <dbl>
## 1 minutes    0.465   0.00386      121.       0    0.458     0.473
```

- Adding cluster robust standard errors does not change our regression coefficient
  - In the same way that heteroskedasticity robust ones did not either
- The standard error on minutes is 0.0039
  - $\Longrightarrow$ a 177.19 % increase in its' magnitude when compared to the naive OLS estimate (`ols1`)
  - $\Longrightarrow$ a 169.52 % increase in its' magnitude when compared to the estimate with fixed effects (`ols2`)

Bottom Line

- Worrying about assumption (5) - i.e. whether the standard errors have either **heteroskedasticity** or **clustering** is important
  - With **heteroskedasticity** regression coefficients OK, **inference is wrong**
  - With *clustered errors*, regression **coefficients** might **not be OK**, and **inference is wrong**
- Remark: We did not worry about what if "heteroskedasticity and clustering" at the same time
  - Why? cluster robust standard errors will clean up any issues with heteroskedasticity for "free"
  - Then why not always do clustering?
    * We have to take a stand on what variables might be causing the clustering
    * Heteroskedasticity doesn't need us to do this
    * Though, most modern empirical work will cluster the standard errors

Acknowledgements

These notes have used inspiration and some content (sometimes quite liberally) from the following sources:

- Gregory S. Crawford's lecture notes from "Empirical Methods" taught in the Master's programs at the University of Zurich
- Ed Rubin's lecture notes from "Introduction to Econometrics" taught in the Bachelor's program at the University Oregon

License

This work is licensed under a Creative Commons Attribution-ShareAlike 4.0 International License.

Suggested Citation

Deer, Lachlan, 2021. Social Media and Web Analytics: Linear Regression - Getting Standard Errors Right. Tilburg University. url = "https://github.com/tisem-digital-marketing/regression-standard-errors"