# Modelling Heteroskedasticity

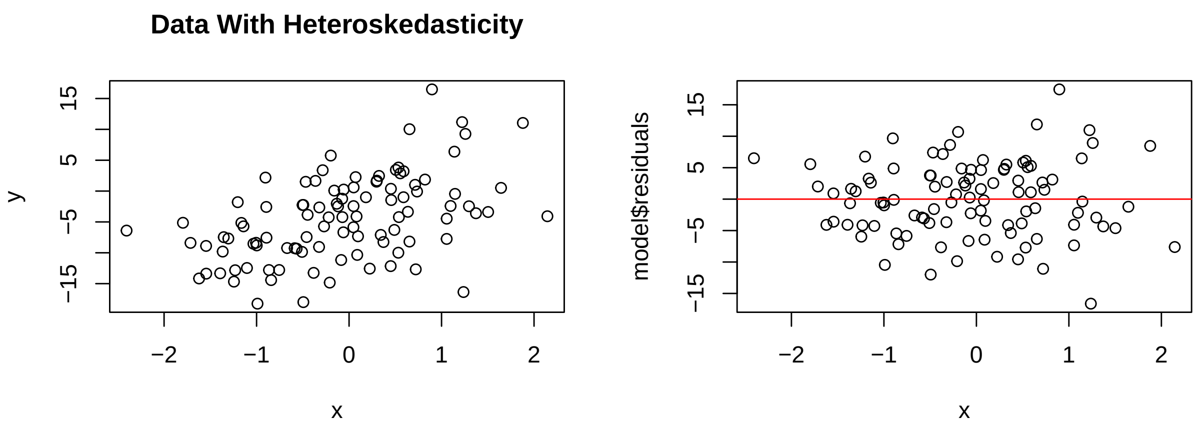
Caden Hewlett

## Introduction

...

## Example: Simulation

Here, we establish the ground truth relation between $x$ and $y$ and aim to see if we can recover it. Specifically, the true relation is $y_i = 3x - 4 + \varepsilon_i$, where $\varepsilon_i \sim N(0, 6 + \frac{3}{2}x_i)$ . Below, we fit a classical frequentist simple linear regression, which has an evident pattern in the residual plots.




## Breusch-Pagan Test

We now conduct a Breusch-Pagan test to verify the heteroskedasticity. Let $\hat{\varepsilon}_i$ be the $i$-th estimated residual from the model. Recall by maximum likelihood estimation, $\hat{\sigma}^2 = \frac{\text{RSS}}{n} = \dfrac{\sum_{i=1}^{n} \varepsilon_i}{n}$ . We thus define $g_i = \hat{\varepsilon_i}^2 / \hat{\sigma}^2$ and fit the linear model $g_i = \gamma_0 + \gamma_1 x_i + \eta_i$. Then, from this model, the test statistic is given as follows $T_{\text{BP}} = \frac{1}{2}(\text{TSS} - \text{RSS}) = \frac{1}{2}\big( \sum_{i=1}^{n} (g_i - \bar{g})^2 -$

$\sum_{i=1}^{n}(g_i - \hat{g}_i)^2)$ and is asymptotically $\chi_p^2$ where $p$ is the number of predictor variables (use $p-1$ if considering the intercept as a predictor 'variable.') The null hypothesis is that there is no evidence of heteroskedasticity in the data.

```
Breusch-Pagan p-value: 0.016239
```

The Breusch-Pagan test is correctly identifying heteroskedasticity at $\alpha = 0.05$. The question becomes... how can we use the Bayesian framework to capture this relationship?

## Hierarchical Model

We adopt the framework of Bayesian Normal regression; however, we attempt to paramaterize $\sigma$ as well. We place a somewhat-sparse hyperprior $\varsigma_i$ on the standard deviation of each normally-distributed covariate $\{\gamma_0, \gamma_1, \beta_0, \beta_1\}$.

$$\begin{aligned}
\{\varsigma_\ell\}_{\ell=0}^{3} &\sim \mathrm{Exp}(0.1) \\
\{\gamma_j\}_{j=0}^{1} &\sim N(0, \varsigma_j^2) \\
\{\beta_j\}_{j=0}^{1} &\sim N(0, \varsigma_{j+2}^2) \\
\mu_i &= \beta_0 + \beta_1 x_i \\
\sigma_i^2 &= \exp(\gamma_0 + \gamma_1 x_i) \\
y_i \mid x_i &\sim N(\mu_i, \sigma_i^2)
\end{aligned}$$

We implement the above in Stan, below:

```stan
data {
  int<lower=1> n;
  vector[n] x;
  vector[n] y;
}

parameters {
  // hyper-prior
  real<lower=0> s_0;
  real<lower=0> s_1;
  real<lower=0> s_2;
  real<lower=0> s_3;

  // coefficients for mean
  real b_0;
  real b_1;
```

```
  // coefficients for log-variance
  real g_0;
  real g_1;
}


model {
  // --------------- //
  // - Hyperpriors - //
  // --------------- //
  s_0 ~ exponential(1);
  s_1 ~ exponential(1);
  s_2 ~ exponential(1);
  s_3 ~ exponential(1);
  // ------------ //
  // --- Mean --- //
  // ------------ //
  b_0 ~ normal(0, s_0);
  b_1 ~ normal(0, s_1);
  // ---------------- //
  // - Log-Variance - //
  // ---------------- //
  s_0 ~ normal(0, s_2);
  s_1 ~ normal(0, s_3);
  // -------------- //
  // - Likelihood - //
  // -------------- //
  for (i in 1:n) {
    // mean
    real mu_i = b_0 + b_1 * x[i];
    // sigma = exp( (g_0 + g_1*x[i]) / 2 )
    real sigma_i = exp(0.5 * (g_0 + g_1 * x[i]));
    // normal likelihood
    y[i] ~ normal(mu_i, sigma_i);
  }
}
```

Now that we have defined the model, we run the MCMC and extract the fit.

```
fit <- readRDS("mod_b.rds")
model_b <- rstan::extract(fit)
```

```r
# mean parameters
b_0 <- model_b$b_0
b_1 <- model_b$b_1
# variance parameters
g_0 <- model_b$g_0
g_1 <- model_b$g_1

# regression means
mu_post <- outer(b_0, rep(1, length(x)), "+") + outer(b_1, x, "*")
# regression variance
sigma_post <- exp(0.5 * (outer(g_0, rep(1, length(x)), "+") + outer(g_1, x, "*")))
# posterior means
mu_mean <- colMeans(mu_post)
sigma_mean <- colMeans(sigma_post)

# credible intervals
mu_lower <- apply(mu_post, 2, quantile, probs = 0.025)
mu_upper <- apply(mu_post, 2, quantile, probs = 0.975)

# estimated predictive intervals
y_lower <- mu_mean - 1.96 * sigma_mean
y_upper <- mu_mean + 1.96 * sigma_mean
# plot it
plot(x, y, ylim = c(-30,30))
lines(x, y_lower, col = 'red')
lines(x, y_upper, col = 'red')
lines(x, mu_mean, col = 'red', lty = 'dotted')
```