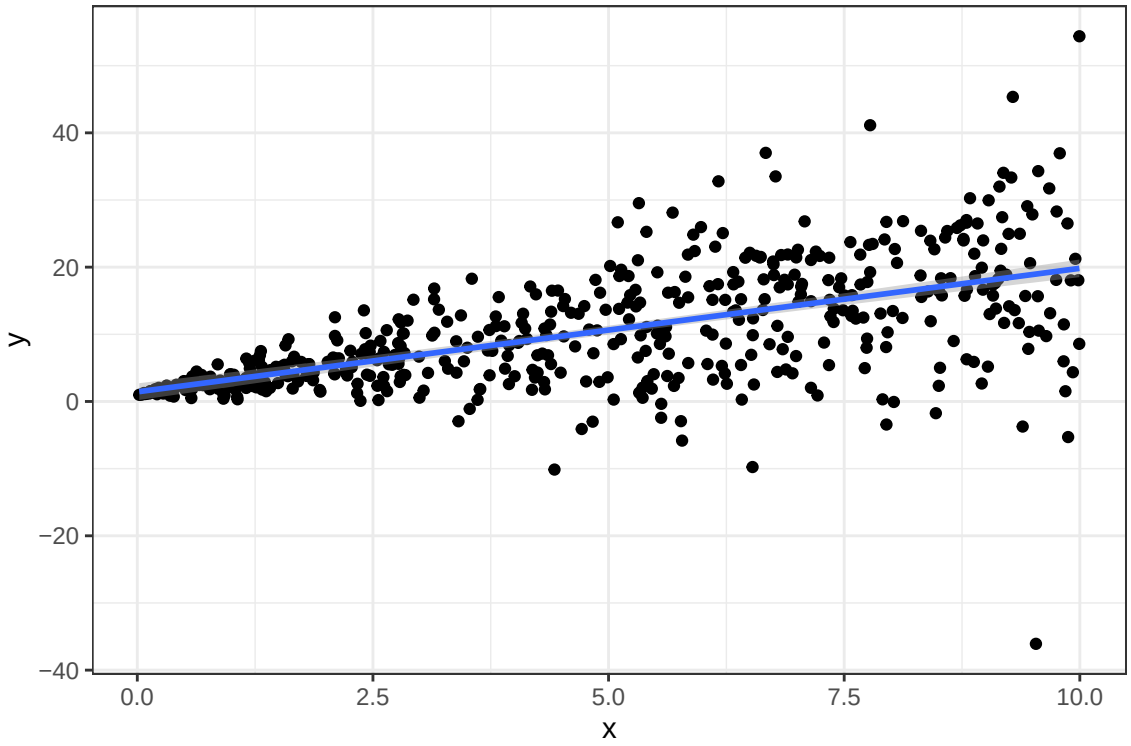
Extending GLMs

```
n <- 500
x <- runif(n,0,10)
beta0 <- 1
beta1 <- 2
y <- rnorm(n, mean = beta0 + x * beta1, sd = x *sqrt(2))

tibble(y = y, x = x) %>% ggplot(aes(y = y, x = x)) +
  geom_point() + theme_bw() +
  geom_smooth(formula = 'y~x', method ='lm')
```

Heteroscedastic Models



Stan code can be written to estimate the variance as a function of x.

```
data {
  int<lower=0> N;
  vector[N] y;
  vector[N] x;
}
```

```
reg_ncv <- stan("heteroskedastic_regression.stan", data=list(N = n, y=y, x = x), refresh = 0)
```

```
print(reg_ncv)
```

```
## Inference for Stan model: heteroskedastic_regression.
## 4 chains, each with iter=2000; warmup=1000; thin=1;
## post-warmup draws per chain=1000, total post-warmup draws=4000.
##
##           mean se_mean   sd     2.5%      25%      50%      75%    97.5% n_eff
## beta0     0.98    0.00 0.02     0.94     0.97     0.98     1.00     1.03  3951
## beta1     2.01    0.00 0.06     1.88     1.97     2.01     2.05     2.14  3106
## sigma     1.37    0.00 0.04     1.29     1.35     1.37     1.40     1.47  3231
## lp__  -1023.72    0.03 1.23 -1027.00 -1024.29 -1023.40 -1022.82 -1022.33  2025
##       Rhat
## beta0    1
## beta1    1
## sigma    1
## lp__     1
##
## Samples were drawn using NUTS(diag_e) at Mon Feb  1 14:03:32 2021.
## For each parameter, n_eff is a crude measure of effective sample size,
## and Rhat is the potential scale reduction factor on split chains (at
## convergence, Rhat=1).
```

Mixture Models

Sometimes a single probability distribution isn't sufficient to model an outcome of interest.

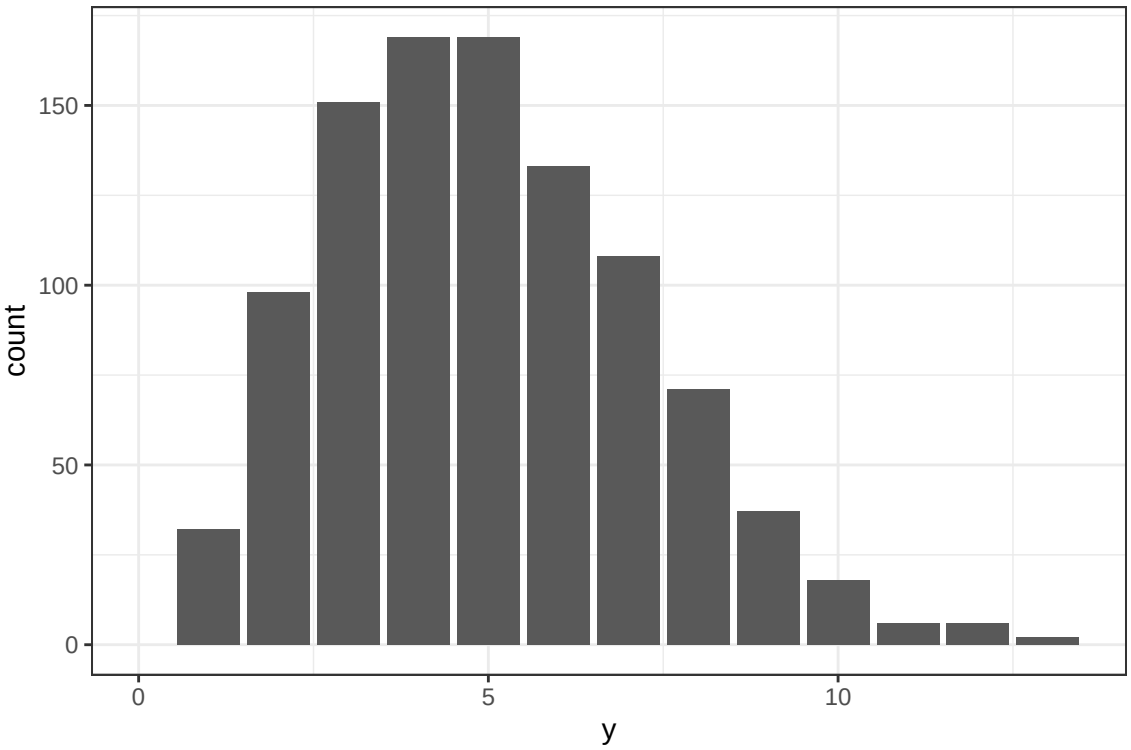
This model could be coded in stan or consider using the `brms` package (bayesian regresion models in stan)

```
zero_prob <- .33
n <- 1000
indicator <- rbinom(n,1,zero_prob)
counts <- tibble(counts = rpois(n, lambda = 15) * (1 - indicator))

zip <- brm(counts ~ 1, data = counts,
           family = zero_inflated_poisson, refresh = 0)
# zip <- brm(counts ~ 1, data = counts,
#family = zero_inflated_poisson, refresh = 0, save_model = 'zip')
```

```
print(zip)
```

```
##  Family: zero_inflated_poisson
##   Links: mu = log; zi = identity
## Formula: counts ~ 1
##    Data: counts (Number of observations: 1000)
## Samples: 4 chains, each with iter = 2000; warmup = 1000; thin = 1;
##          total post-warmup samples = 4000
##
## Population-Level Effects:
##           Estimate Est.Error l-95% CI u-95% CI Rhat Bulk_ESS Tail_ESS
## Intercept     2.72      0.01     2.70     2.74 1.00     3827     2654
##
## Family Specific Parameters:
##    Estimate Est.Error l-95% CI u-95% CI Rhat Bulk_ESS Tail_ESS
## zi     0.35      0.01     0.32     0.38 1.00     3961     2282
##
## Samples were drawn using sampling(NUTS). For each parameter, Bulk_ESS
## and Tail_ESS are effective sample size measures, and Rhat is the potential
## scale reduction factor on split chains (at convergence, Rhat = 1).
```

```
trunc_pois <- tibble(y = rtpois(n, 5, a = 0, b = Inf))

trunc_pois %>% ggplot(aes(x = y)) +
  geom_bar() + theme_bw() + xlim(0, NA)
```




```
truncated_pois <- brm(y ~ 1, data = trunc_pois, family = hurdle_poisson, refresh = 0)
```

```
print(truncated_pois)
```

```
##  Family: hurdle_poisson
##   Links: mu = log; hu = identity
## Formula: y ~ 1
##    Data: trunc_pois (Number of observations: 1000)
## Samples: 4 chains, each with iter = 2000; warmup = 1000; thin = 1;
##          total post-warmup samples = 4000
##
## Population-Level Effects:
##           Estimate Est.Error l-95% CI u-95% CI Rhat Bulk_ESS Tail_ESS
## Intercept     1.60      0.01     1.58     1.63 1.00     2357     2137
##
## Family Specific Parameters:
##    Estimate Est.Error l-95% CI u-95% CI Rhat Bulk_ESS Tail_ESS
## hu     0.00      0.00     0.00     0.00 1.00     2023     1412
##
## Samples were drawn using sampling(NUTS). For each parameter, Bulk_ESS
## and Tail_ESS are effective sample size measures, and Rhat is the potential
## scale reduction factor on split chains (at convergence, Rhat = 1).
```

```
hurdle_pois <- trunc_pois %>% bind_rows(tibble(y = rep(0, n)))
hurdle_poisson <- brm(y ~ 1, data = hurdle_pois, family = hurdle_poisson, refresh = 0)
```

```
## Compiling Stan program...
```

```
## Trying to compile a simple C file
```

```
## Start sampling
```

```
print(hurdle_poisson)
```

```
##  Family: hurdle_poisson
##   Links: mu = log; hu = identity
## Formula: y ~ 1
##    Data: hurdle_pois (Number of observations: 2000)
## Samples: 4 chains, each with iter = 2000; warmup = 1000; thin = 1;
##          total post-warmup samples = 4000
##
## Population-Level Effects:
##           Estimate Est.Error l-95% CI u-95% CI Rhat Bulk_ESS Tail_ESS
## Intercept     1.60      0.01     1.57     1.63 1.00     2620     2284
##
## Family Specific Parameters:
##    Estimate Est.Error l-95% CI u-95% CI Rhat Bulk_ESS Tail_ESS
## hu     0.50      0.01     0.48     0.52 1.00     3232     2611
##
## Samples were drawn using sampling(NUTS). For each parameter, Bulk_ESS
## and Tail_ESS are effective sample size measures, and Rhat is the potential
## scale reduction factor on split chains (at convergence, Rhat = 1).
```

```
zero_prob <- .33
n <- 1000
indicator <- rbinom(n,1,zero_prob)
counts <- tibble(counts = rlnorm(n, meanlog = log(5)) * (1 - indicator))

head(counts)
```

Discrete / Continuous

```
## # A tibble: 6 x 1
##   counts
##    <dbl>
## 1   0
## 2   3.16
## 3   3.00
## 4  25.3
## 5   0
## 6   2.99
```

```
counts %>% ggplot(aes(x = counts)) + geom_histogram(bins = 40)
```