Reshaping data - Binary variables

Frank Edwards

2/15/2019

Review HW 3

Reshaping data using the tidyverse

Grouping and summarizing

Evaluating the structure of the data

```
head(gapminder)
```

```
## # A tibble: 6 x 6
##   country     continent  year lifeExp      pop gdpPercap
##   <fct>       <fct>     <int>   <dbl>    <int>     <dbl>
## 1 Afghanistan Asia       1952    28.8  8425333      779.
## 2 Afghanistan Asia       1957    30.3  9240934      821.
## 3 Afghanistan Asia       1962    32.0 10267083      853.
## 4 Afghanistan Asia       1967    34.0 11537966      836.
## 5 Afghanistan Asia       1972    36.1 13079460      740.
## 6 Afghanistan Asia       1977    38.4 14880372      786.
```

How is this data structured?

What natural groupings are present in this data?

Grouping and summarizing: by country

```
gapminder %>% group_by(country) %>% summarise(mean_lifeExp = mean(lifeExp))
```

```
## # A tibble: 142 x 2
##    country     mean_lifeExp
##    <fct>              <dbl>
##  1 Afghanistan         37.5
##  2 Albania             68.4
##  3 Algeria             59.0
##  4 Angola              37.9
##  5 Argentina           69.1
##  6 Australia           74.7
##  7 Austria             73.1
##  8 Bahrain             65.6
##  9 Bangladesh          49.8
## 10 Belgium             73.6
## # ... with 132 more rows
```

Grouping and summarizing: by country (cont.)

```
gapminder %>% group_by(country) %>% summarise(mean_lifeExp = mean(lifeExp),
    max_lifeExp = max(lifeExp), min_lifeExp = min(lifeExp))
```

```
## # A tibble: 142 x 4
##    country     mean_lifeExp max_lifeExp min_lifeExp
##    <fct>              <dbl>       <dbl>       <dbl>
##  1 Afghanistan         37.5        43.8        28.8
##  2 Albania             68.4        76.4        55.2
##  3 Algeria             59.0        72.3        43.1
##  4 Angola              37.9        42.7        30.0
##  5 Argentina           69.1        75.3        62.5
##  6 Australia           74.7        81.2        69.1
##  7 Austria             73.1        79.8        66.8
##  8 Bahrain             65.6        75.6        50.9
##  9 Bangladesh          49.8        64.1        37.5
## 10 Belgium             73.6        79.4        68
## # ... with 132 more rows
```

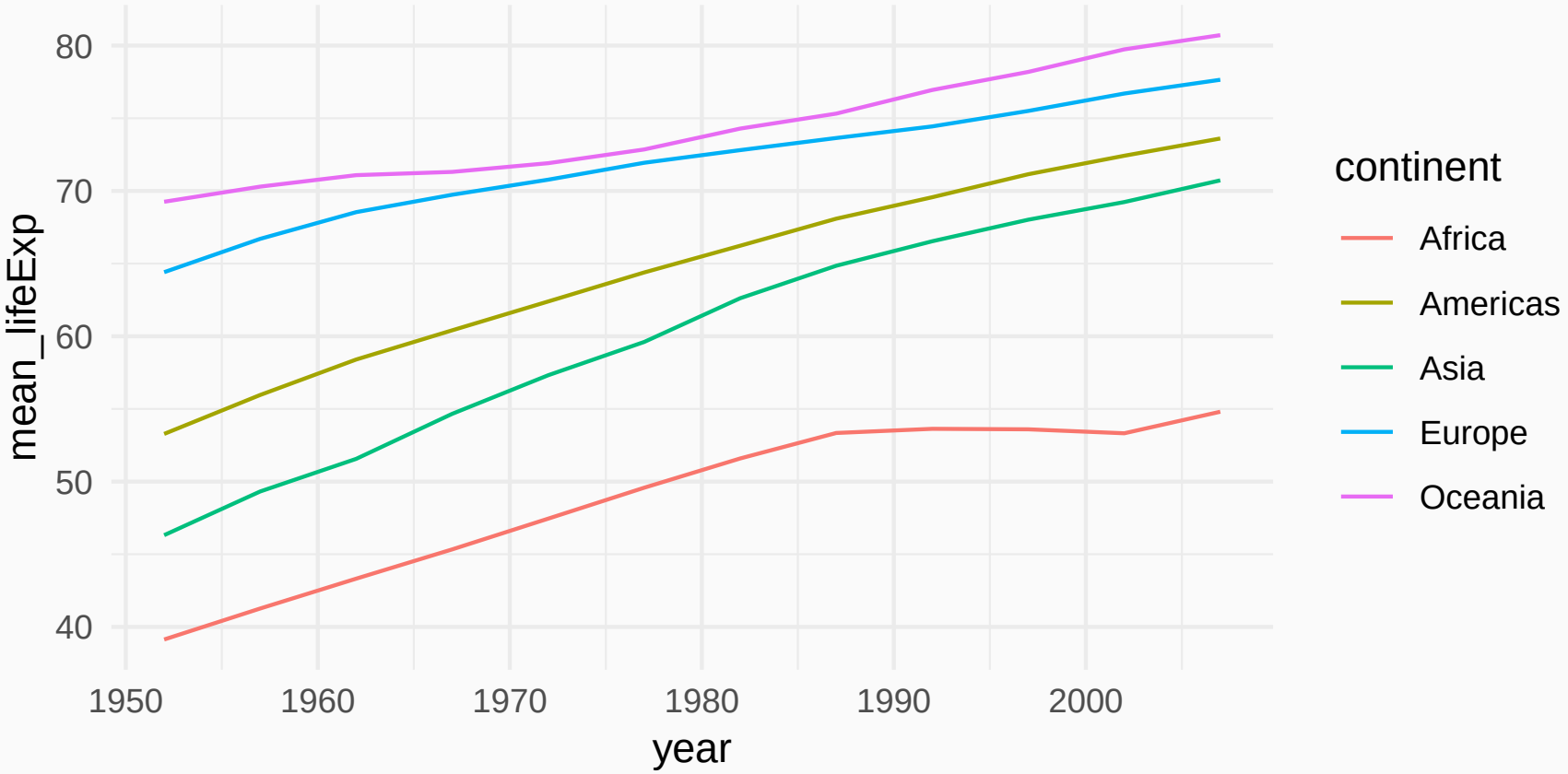
Grouping and summarizing: by continent and year

```
gapminder %>% group_by(continent, year) %>% summarise(mean_lifeExp = mean(lifeExp))
```

```
## # A tibble: 60 x 3
## # Groups:   continent [?]
##    continent  year mean_lifeExp
##    <fct>     <int>        <dbl>
##  1 Africa     1952         39.1
##  2 Africa     1957         41.3
##  3 Africa     1962         43.3
##  4 Africa     1967         45.3
##  5 Africa     1972         47.5
##  6 Africa     1977         49.6
##  7 Africa     1982         51.6
##  8 Africa     1987         53.3
##  9 Africa     1992         53.6
## 10 Africa     1997         53.6
## # ... with 50 more rows
```

Grouping and summarizing: by continent and year (cont.)

```
gapminder %>% group_by(continent, year) %>% summarise(mean_lifeExp = mean(lifeExp)) %>%
    ggplot(aes(x = year, y = mean_lifeExp, col = continent)) + geom_line()
```



Reshaping: gather and spread (long<->wide)

Is this data long or wide?

```
dat <- gapminder %>% group_by(continent, year) %>% summarise(mean_lifeExp = mean(lifeExp))
head(dat)
```

```
## # A tibble: 6 x 3
## # Groups:   continent [1]
##   continent  year mean_lifeExp
##   <fct>     <int>        <dbl>
## 1 Africa     1952         39.1
## 2 Africa     1957         41.3
## 3 Africa     1962         43.3
## 4 Africa     1967         45.3
## 5 Africa     1972         47.5
## 6 Africa     1977         49.6
```

Use spread() to make it wide by continent

```
dat_wide <- dat %>% spread(key = continent, value = mean_lifeExp)
head(dat_wide)
```

```
## # A tibble: 6 x 6
##    year Africa Americas  Asia Europe Oceania
##   <int>  <dbl>    <dbl> <dbl>  <dbl>   <dbl>
## 1  1952   39.1     53.3  46.3   64.4    69.3
## 2  1957   41.3     56.0  49.3   66.7    70.3
## 3  1962   43.3     58.4  51.6   68.5    71.1
## 4  1967   45.3     60.4  54.7   69.7    71.3
## 5  1972   47.5     62.4  57.3   70.8    71.9
## 6  1977   49.6     64.4  59.6   71.9    72.9
```

Use spread() to make it wide by year

```
dat_wide <- dat %>% spread(key = year, value = mean_lifeExp)
head(dat_wide)
```

```
## # A tibble: 5 x 13
## # Groups:   continent [5]
##   continent `1952` `1957` `1962` `1967` `1972` `1977` `1982` `1987` `1992`
##   <fct>      <dbl>  <dbl>  <dbl>  <dbl>  <dbl>  <dbl>  <dbl>  <dbl>  <dbl>
## 1 Africa      39.1   41.3   43.3   45.3   47.5   49.6   51.6   53.3   53.6
## 2 Americas    53.3   56.0   58.4   60.4   62.4   64.4   66.2   68.1   69.6
## 3 Asia        46.3   49.3   51.6   54.7   57.3   59.6   62.6   64.9   66.5
## 4 Europe      64.4   66.7   68.5   69.7   70.8   71.9   72.8   73.6   74.4
## 5 Oceania     69.3   70.3   71.1   71.3   71.9   72.9   74.3   75.3   76.9
## # ... with 3 more variables: `1997` <dbl>, `2002` <dbl>, `2007` <dbl>
```

Use gather() to make wide data long

```
dat_long <- dat_wide %>% gather(key = "year", value = "mean_lifeExp", "1952",
    "1957", "1962", "1967", "1972", "1977", "1982", "1992", "1997", "2002",
    "2007")
head(dat_long)
```

```
## # A tibble: 6 x 4
## # Groups:   continent [5]
##   continent `1987` year  mean_lifeExp
##   <fct>      <dbl> <chr>        <dbl>
## 1 Africa      53.3 1952          39.1
## 2 Americas    68.1 1952          53.3
## 3 Asia        64.9 1952          46.3
## 4 Europe      73.6 1952          64.4
## 5 Oceania     75.3 1952          69.3
## 6 Africa      53.3 1957          41.3
```

Use gather() to make wide data long with less code

```
dat_long <- dat_wide %>% gather(key = "year", value = "mean_lifeExp", -continent)
head(dat_long)
```

```
## # A tibble: 6 x 3
## # Groups:   continent [5]
##   continent year  mean_lifeExp
##   <fct>     <chr>        <dbl>
## 1 Africa    1952          39.1
## 2 Americas  1952          53.3
## 3 Asia      1952          46.3
## 4 Europe    1952          64.4
## 5 Oceania   1952          69.3
## 6 Africa    1957          41.3
```

Break

Binary/Bernoulli data

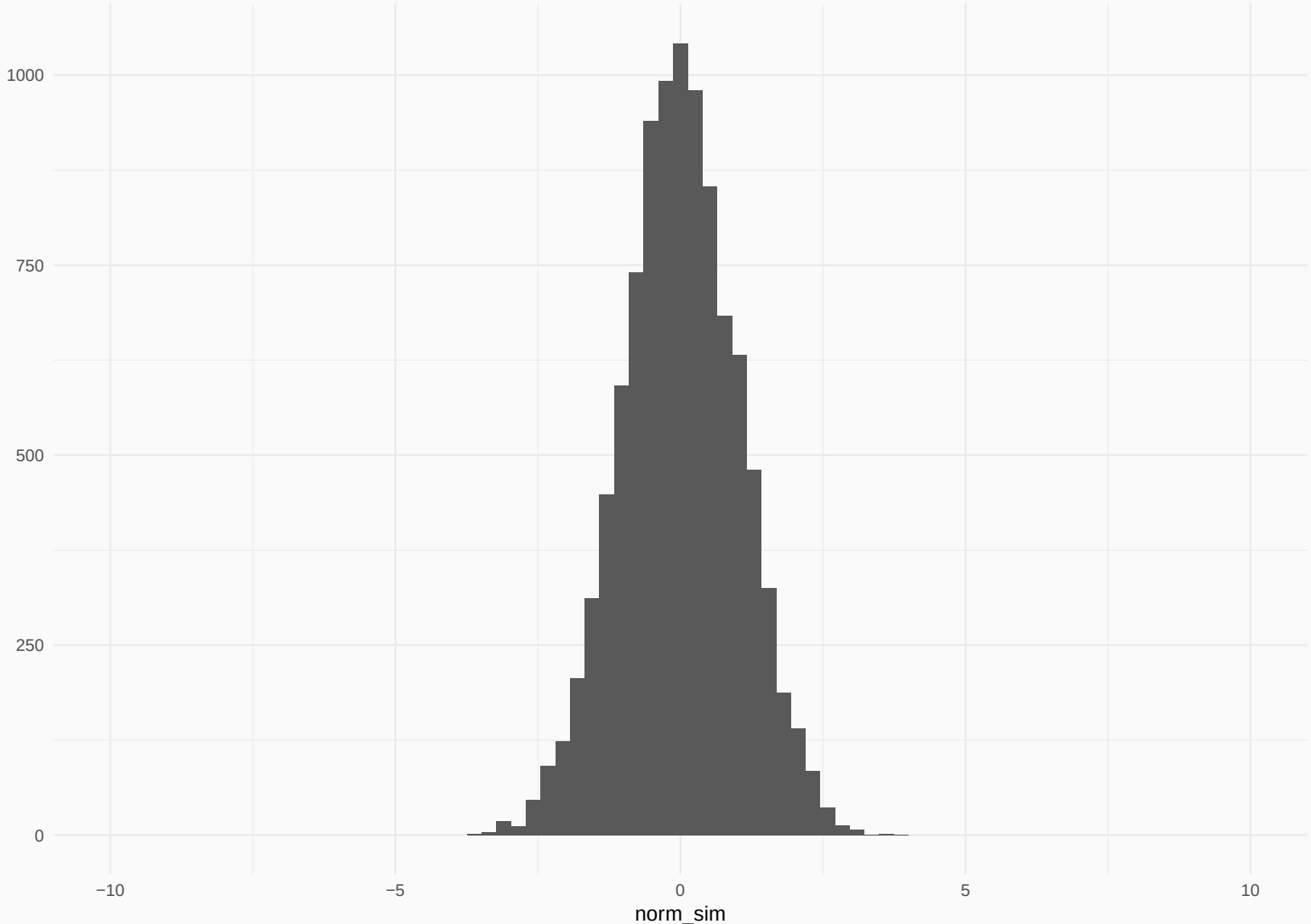
Variables are sampled from probability distributions

Recall that a normally distributed random variable y with mean μ and variance σ^2 can be expressed as:

$$y \sim Normal(\mu, \sigma^2)$$

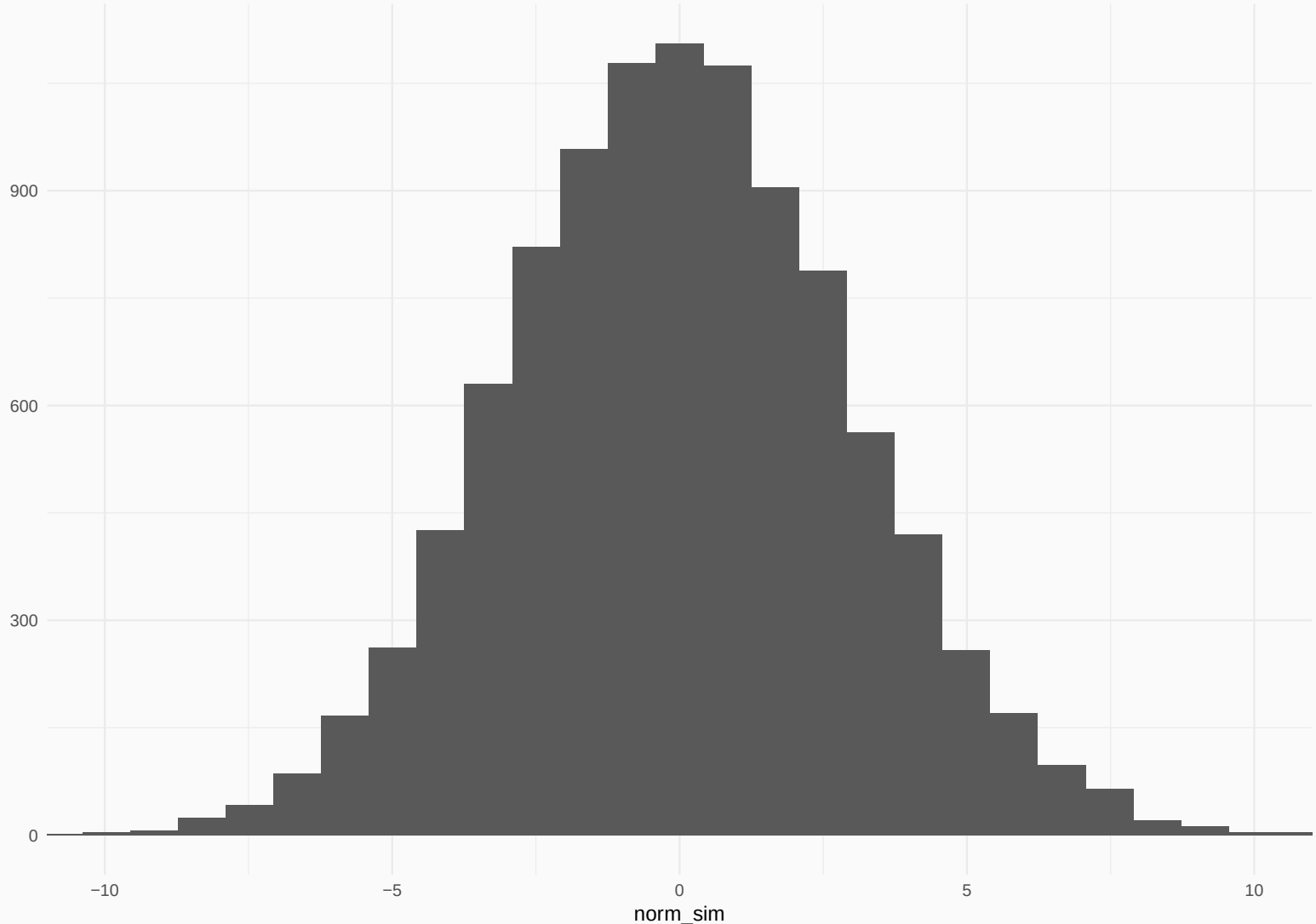
Parameters and shape

```
norm_sim <- rnorm(10000, mean = 0, sd = 1)
qplot(norm_sim) + coord_cartesian(xlim = c(-10, 10))
```



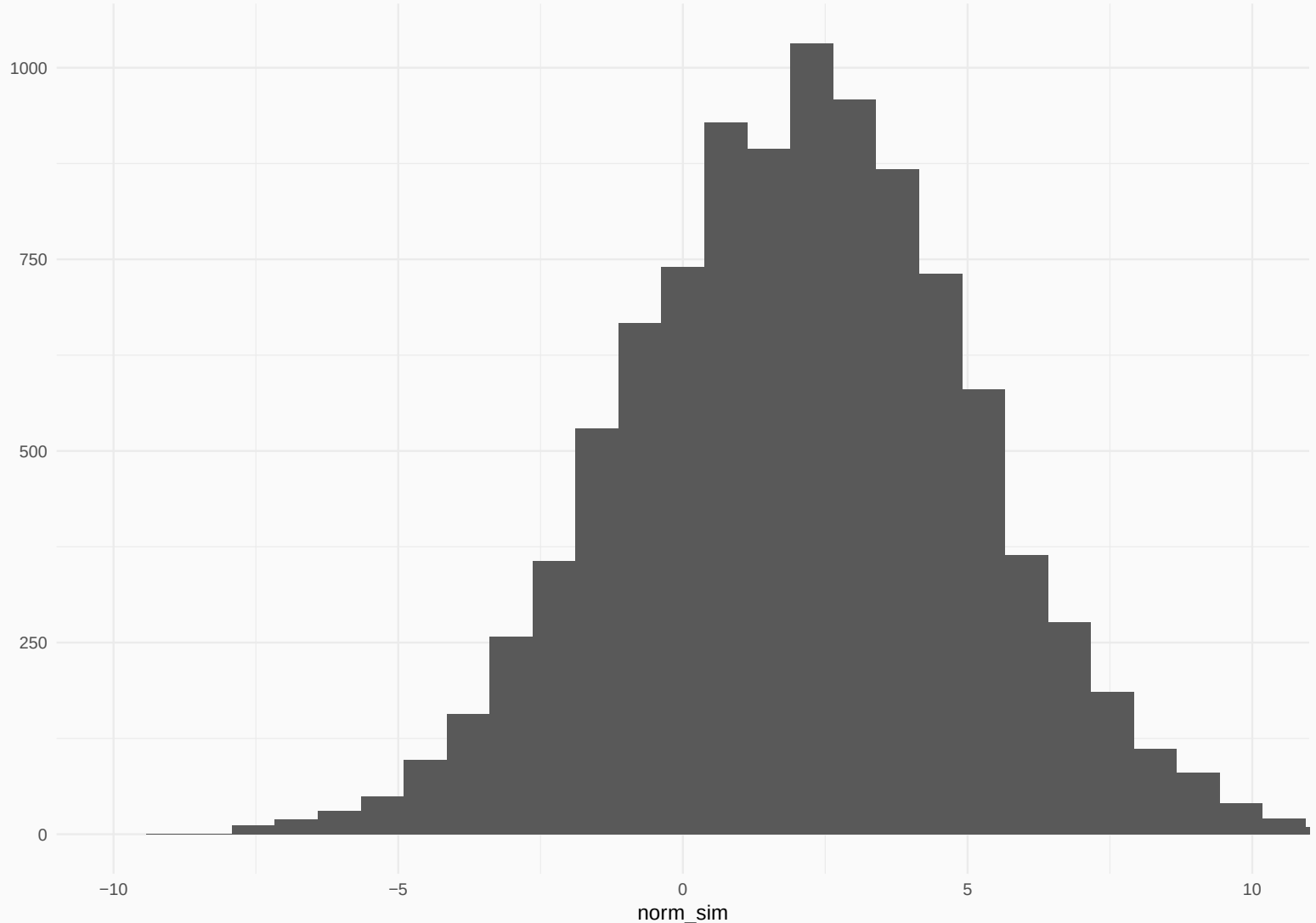
Parameters and shape

```
norm_sim <- rnorm(10000, mean = 0, sd = 3)
qplot(norm_sim) + coord_cartesian(xlim = c(-10, 10))
```



Parameters and shape

```
norm_sim <- rnorm(10000, mean = 2, sd = 3)
qplot(norm_sim) + coord_cartesian(xlim = c(-10, 10))
```



All regressions model outcomes as random variables

Recall that a linear regression treats y as a random variable with mean expectation such that each y_i can be modeled as

$$y_i = X\beta + \varepsilon$$

or

$$y \sim Normal(X\beta, \sigma^2)$$

So each observation y_i is treated as a draw from a Normal distribution with $\mu = X\beta$ and variance σ^2.

Does one size fit all?

Does the normal model describe all phenomena we study well?

An alternative: the Bernoulli distribution for binary data

The Bernoulli distribution for random variable X

$$\Pr(X = 1) = p = 1 - \Pr(X = 0)$$

Paramterization:

$$y \sim Bernoulli(p)$$

Properties of random binary variables

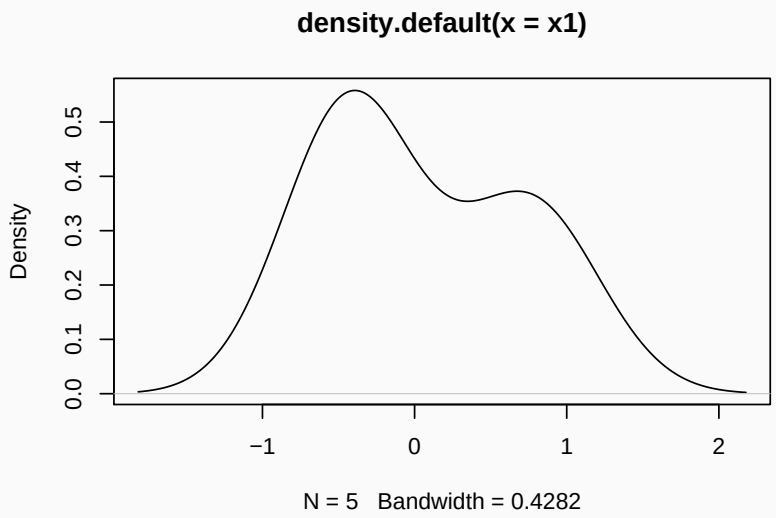
If y is an i.i.d. Bernoulli variable with probability p:

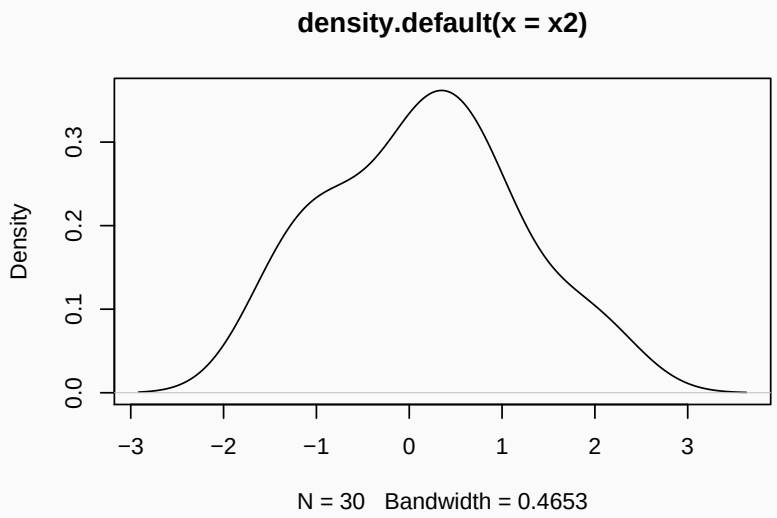
$$y \sim Bernoulli(p)$$

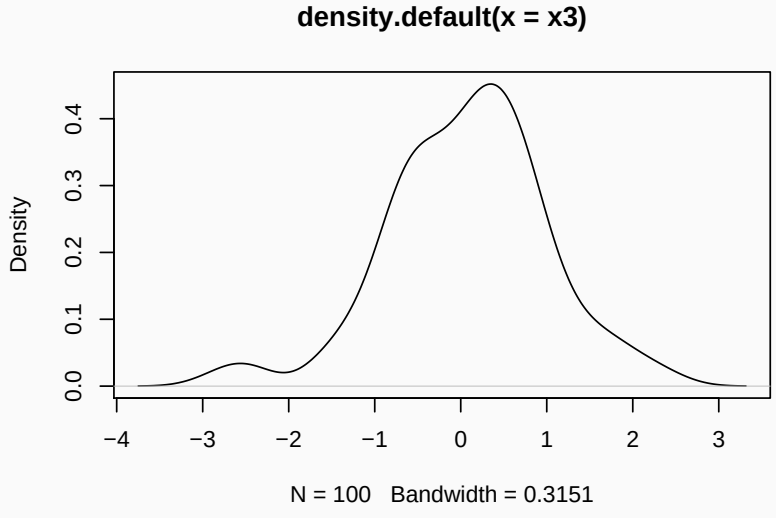
$$E(y) = p$$

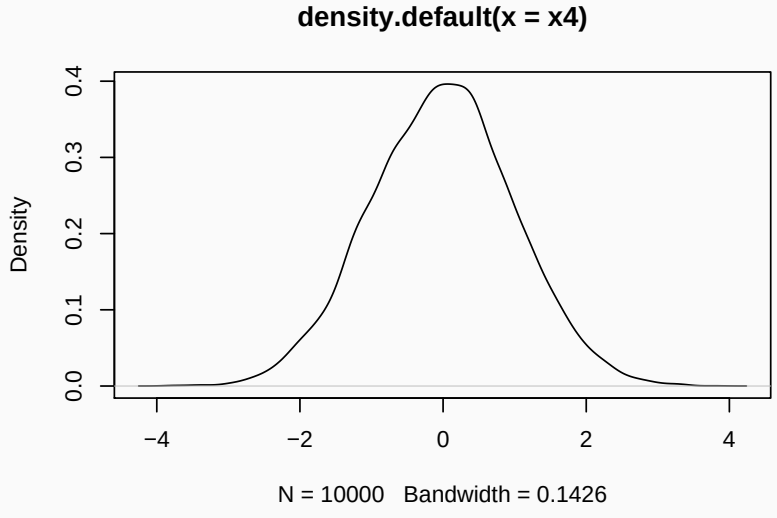
$$Var(y) = p(1 - p)$$

Recall the central limit theorem









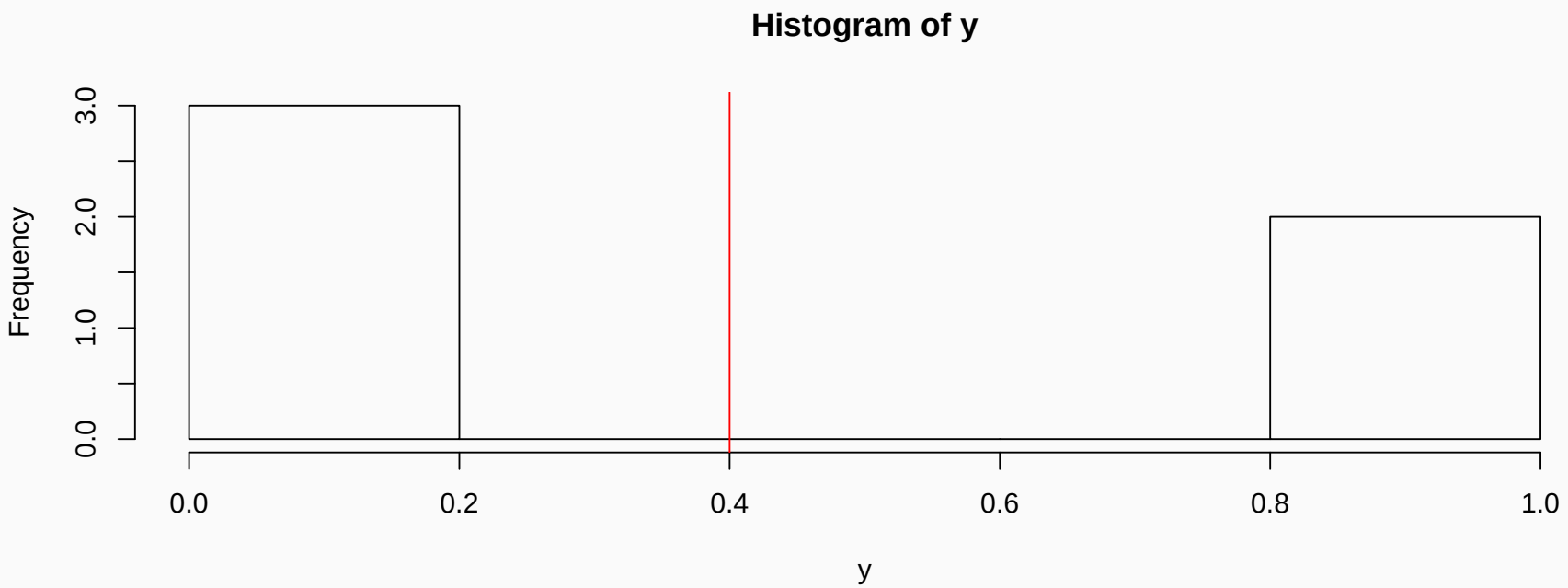
A Bernoulli variable as a coin flip

```
flip_n_coins <- function(n) {
    rbinom(n, 1, 0.5)
}
flip_n_coins(10)
```

```
##  [1] 1 0 0 0 0 0 1 0 1 0
```

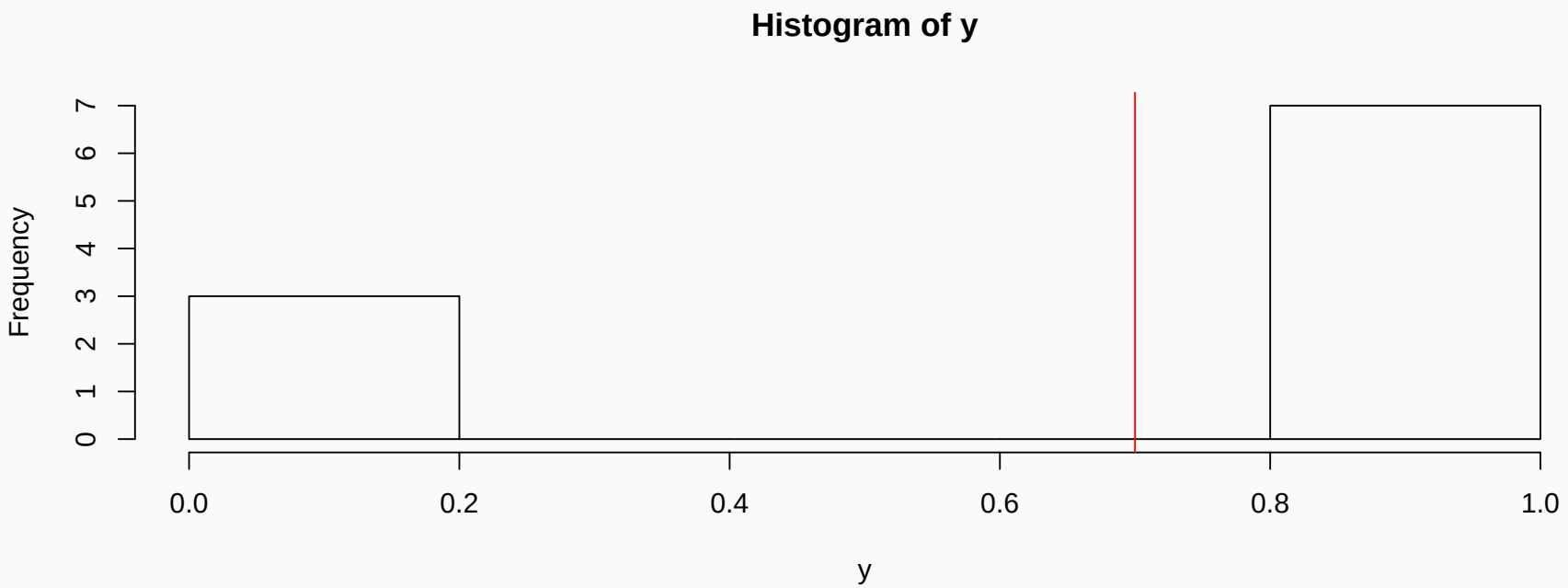
What does the distribution of a binary variable look like?

```
y <- flip_n_coins(5)
hist(y)
abline(v = mean(y), col = 2)
```



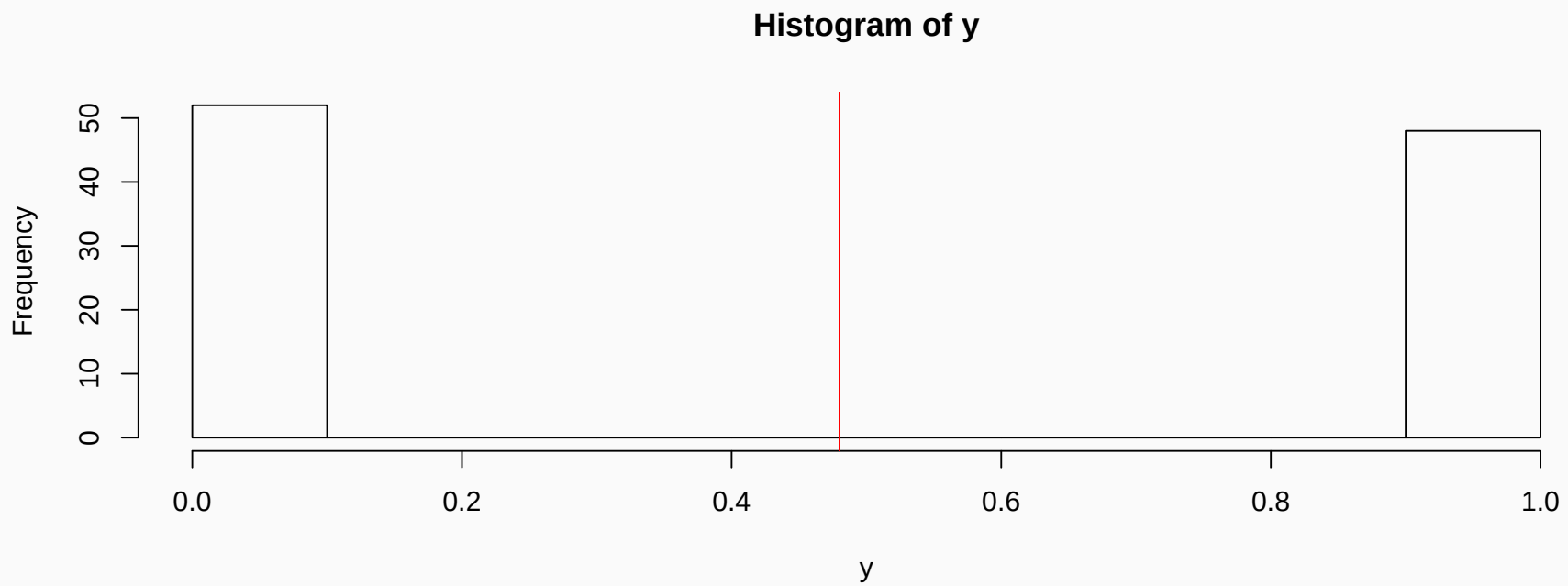
What does the distribution of a binary variable look like?

```
y <- flip_n_coins(10)
hist(y)
abline(v = mean(y), col = 2)
```



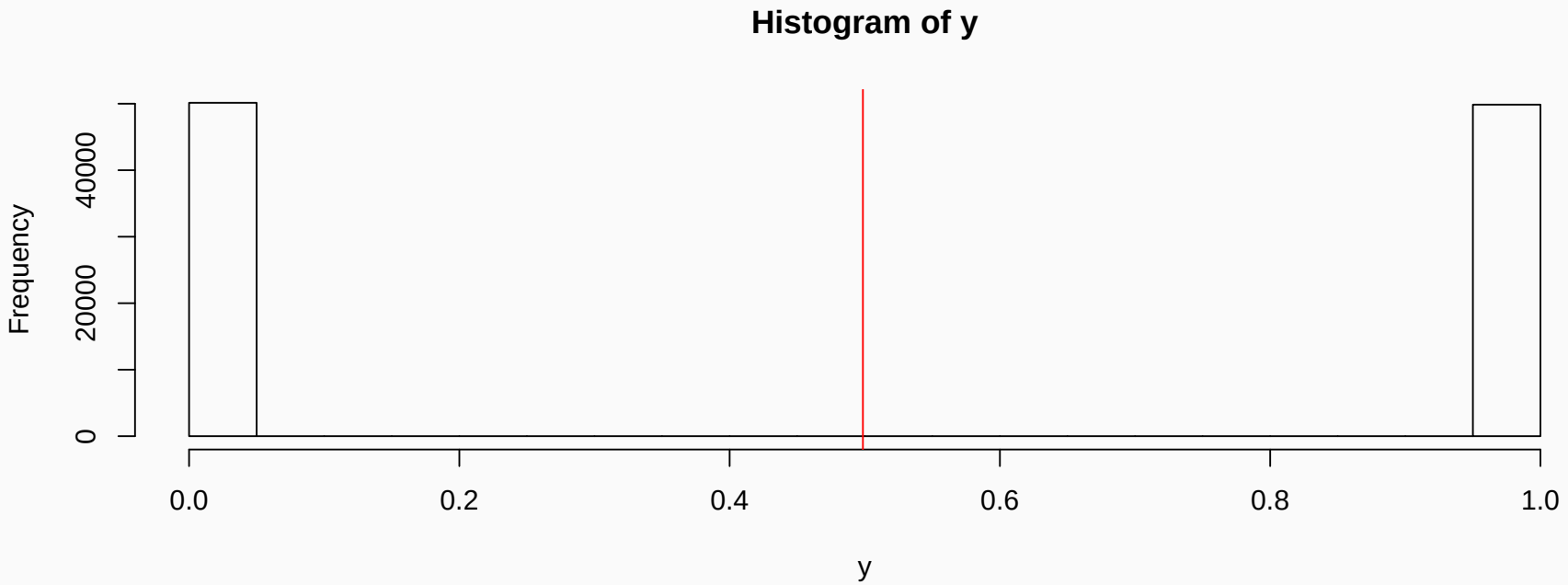
What does the distribution of a binary variable look like?

```
y <- flip_n_coins(100)
hist(y)
abline(v = mean(y), col = 2)
```



What does the distribution of a binary variable look like?

```
y <- flip_n_coins(1e+05)
hist(y)
abline(v = mean(y), col = 2)
```



Recap

A binary variable y takes on the values 1 or 0, with probability

$$Pr(y = 1) = p$$

Let’s look at some data