False Positive Results Visualized: A Simulation in R

mb

2022-07-13

Let's assume a simple theory where one variable (x) is related to one other variable (y), but in fact the theory is false. That is, in reality there is no relationship between x and y. However, if enough studies are run that test this relationship, some of them are bound to find it. A typical study in social science research might include 50 participants (N). Parts of the code are adapted from this great resource.

Setup and generating data

```
library(tidyverse)
library(broom)
N <- 50
s <- 10000
```

Next, the rnorm() function is used to sample 500,000 values for x and y. The defaults are mean = 0 and sd = 1. Everything is collected in a data frame (tibble) including a label for the study. So, the first 50 lines in sim_data will all be from study 1, rows 51 to 100 from study 2, and so on.

```
set.seed(42)
sim_data <- tibble(study = rep(1:s, each=N),
                   x = rnorm(N * s),
                   y = rnorm(N * s))
sim_data
```

```
## # A tibble: 500,000 x 3
##    study       x       y
##    <int>   <dbl>   <dbl>
##  1     1  1.37   -0.0965
##  2     1 -0.565   1.06
##  3     1  0.363   0.0569
##  4     1  0.633   0.0659
##  5     1  0.404  -0.151
##  6     1 -0.106   0.134
##  7     1  1.51    0.577
##  8     1 -0.0947 -0.982
##  9     1  2.02   -1.42
## 10     1 -0.0627 -2.55
## # ... with 499,990 more rows
```

The true correlation is zero

Now we can look at the bivariate correlation between x and y across all rows/cases. This disregards the nested structure of the data (i.e., the fact that it's not 500,000 cases in one study but 50 cases each in 10,000 studies), but gives us a value very close to the true relationship. The true relationship is zero because the two randomly generated variables are independent. We also see that the means of `x` and `y` are close to zero and their standard deviations close to 1.

```
sim_data %>% select(x,y) %>% cor()
```

```
##             x           y
## x 1.000000000 0.002662993
## y 0.002662993 1.000000000
```

```
sim_data %>% summarise(
  count = n(),
  mean.x = mean(x), sd.x = sd(x),
  mean.y = mean(y), sd.y = sd(y)
)
```

```
## # A tibble: 1 x 5
##    count     mean.x  sd.x  mean.y  sd.y
##    <int>      <dbl> <dbl>   <dbl> <dbl>
## 1 500000 -0.0000404 0.999 0.00119  1.00
```

Within each of the 10,000 studies, things can look quite different though.

```
sim_data %>% group_by(study) %>%
  summarise(
  count = n(),
  mean.x = mean(x), sd.x = sd(x),
  mean.y = mean(y), sd.y = sd(y)
)
```

```
## # A tibble: 10,000 x 6
##    study count   mean.x  sd.x  mean.y  sd.y
##    <int> <int>    <dbl> <dbl>   <dbl> <dbl>
##  1     1    50 -0.0357  1.15  -0.128  1.12
##  2     2    50  0.101   0.925 -0.0571 1.04
##  3     3    50 -0.151   0.928 -0.236  1.03
##  4     4    50 -0.0237  0.885  0.0927 0.935
##  5     5    50  0.00794 0.988 -0.237  1.04
##  6     6    50 -0.0287  1.05   0.0780 1.02
##  7     7    50 -0.0615  0.795  0.256  0.964
##  8     8    50  0.127   0.949 -0.0282 1.02
##  9     9    50 -0.119   0.996  0.0144 1.07
## 10    10    50 -0.116   1.05  -0.147  0.968
## # ... with 9,990 more rows
```

Next, we'll focus on the correlation within each study (new variable `r`) sorted by the size of the coefficient. There are quite a few studies in which the correlation is clearly not zero.

```
res <- sim_data %>%
  group_by(study) %>%
  summarize(r = cor(x, y)) %>%
  arrange(desc(r))
res
```

```
## # A tibble: 10,000 x 2
##    study     r
##    <int> <dbl>
##  1  3111 0.525
##  2  6154 0.492
##   3  3989 0.460
##   4  7003 0.449
##   5  2733 0.444
##   6  4521 0.442
##   7   477 0.441
##   8  7170 0.439
##   9  8109 0.436
##  10  8150 0.435
## # ... with 9,990 more rows
```

By the way, the mean of the individual study correlations is not (necessarily) the same as the overall correlation.

```
res %>% summarise(average.cor = mean(r))
```

```
## # A tibble: 1 x 1
##   average.cor
##         <dbl>
## 1     0.00280
```

```
sim_data %>% summarise(global.cor = cor(x,y))
```

```
## # A tibble: 1 x 1
##   global.cor
##        <dbl>
## 1    0.00266
```

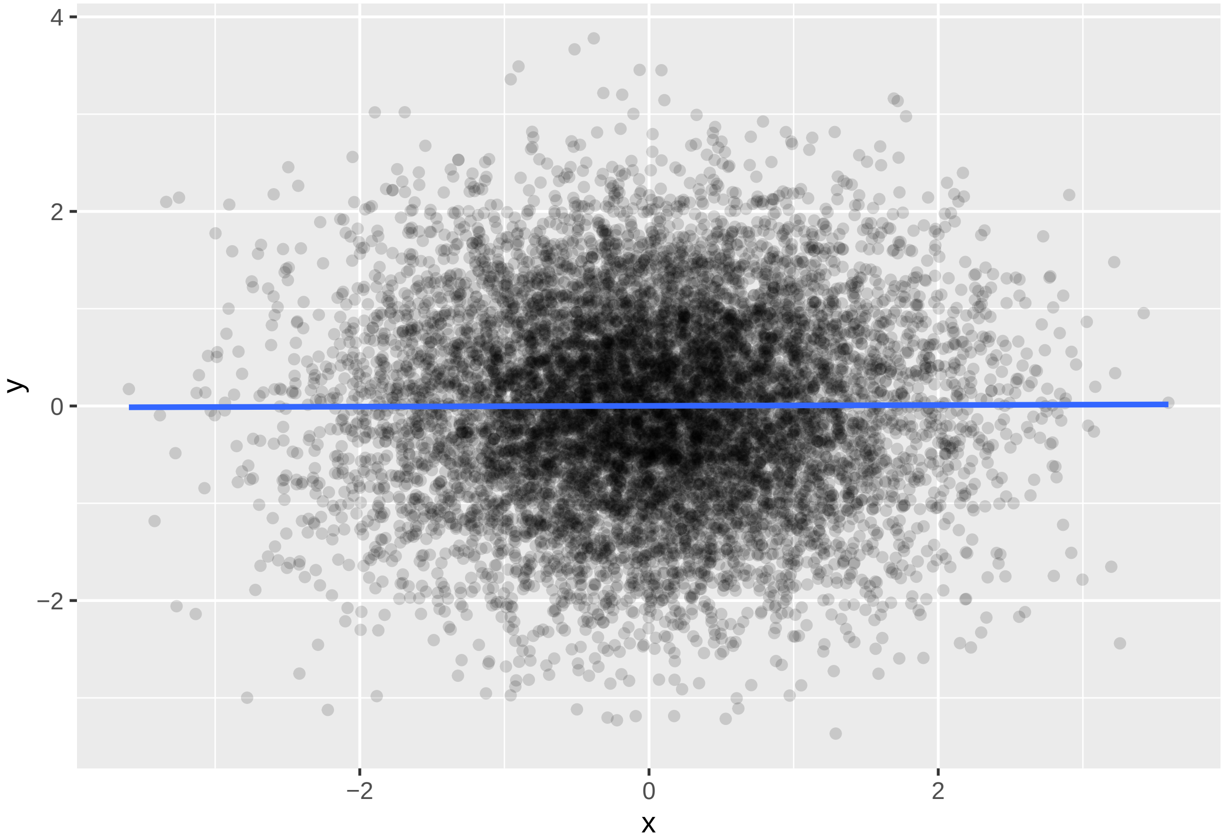
At the other end, we also get negative correlations of roughly the same magnitude.

```
res %>% arrange(r)
```

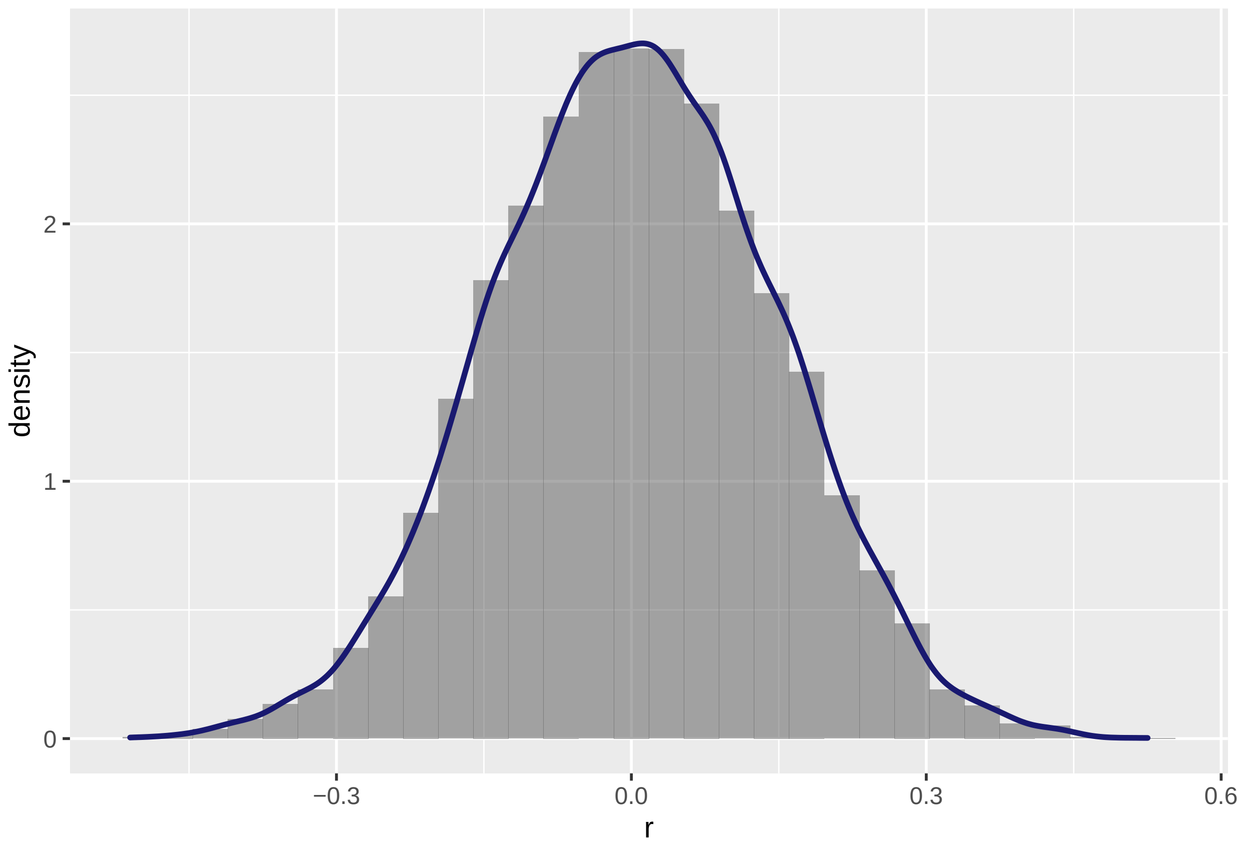
```
## # A tibble: 10,000 x 2
##     study      r
##     <int>  <dbl>
##   1   232 -0.510
##   2  3690 -0.495
##   3  7200 -0.474
##   4  4112 -0.465
##   5  6375 -0.458
##   6  2073 -0.454
##   7   239 -0.452
##   8  9139 -0.440
##   9  4528 -0.437
##  10  7976 -0.435
## # ... with 9,990 more rows
```

Without grouping by study, we already saw that the correlation is essentially zero, which a scatterplot of a random subset of cases (because there are too many to plot nicely) confirms.

```
sim_data %>% slice_sample(n = 10000) %>%
  ggplot(aes(x, y)) +
  geom_point(alpha = 0.15) +
  geom_smooth(method = "lm", se = FALSE)
```

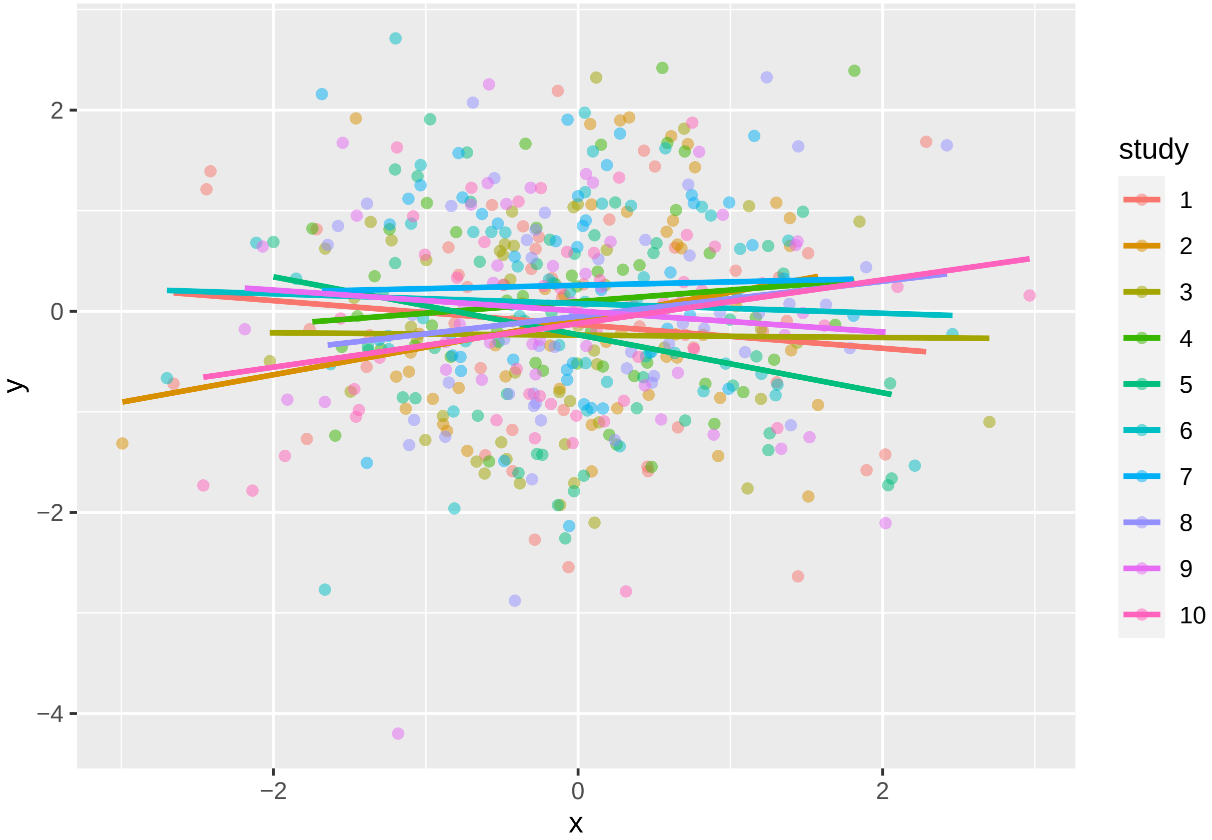


```
res %>%
  ggplot(aes(x = r)) +
  geom_histogram(aes(y = stat(density)), alpha = 0.5) +
  geom_density(color = "midnightblue", lwd = 1)
```



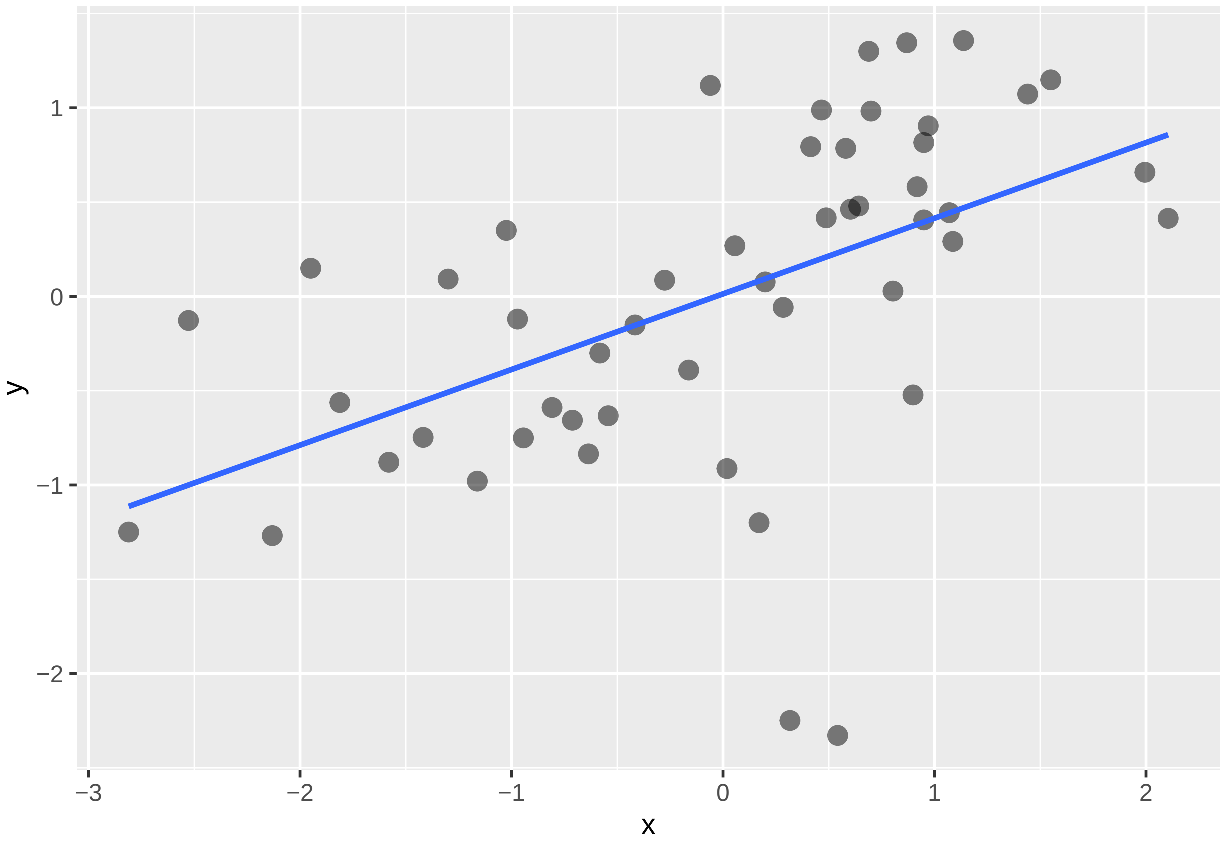
Plotting a subset of studies – here just the first 10 – we see that some fit lines have a positive and some a negative slope.

```
sim_data %>% filter(study %in% 1:10) %>%
  mutate(study = as_factor(study)) %>%
  ggplot(aes(x, y, color = study)) +
  geom_point(alpha = 0.5) +
  geom_smooth(method = "lm", se = FALSE)
```



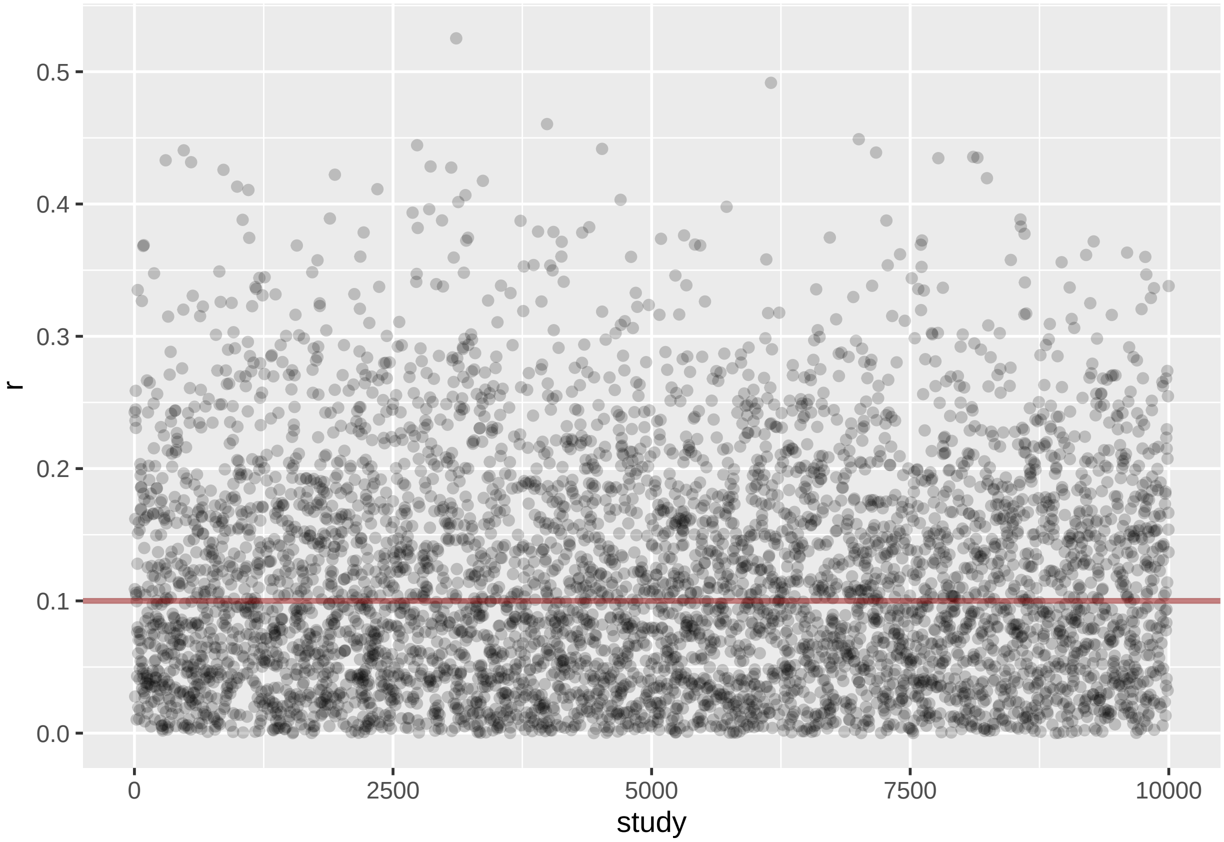
And here is the scatterplot for just the study with the highest correlation.

```
sim_data %>% filter(study == res$study[which.max(res$r)]) %>%
  ggplot(aes(x, y)) +
  geom_point(size = 3, alpha = 0.5) +
  geom_smooth(method = "lm", se = FALSE)
```



Assuming in the theoretical context of the relationship between `x` and `y` a positive correlation of 0.1 or larger is considered substantive, how many of the 10,000 studies yield and `r` of 0.1 or more? All the studies above the red line would have led us to falsely conclude that `x` and `y` are substantively related.

```
co_r <- 0.1
res %>%
  filter(r >= 0) %>%
  ggplot(aes(study,r)) +
  geom_point(alpha=0.2) +
  geom_hline(yintercept = co_r,
             color = "darkred", lwd = 1, alpha = 0.5)
```



Linear model

So far, we have assumed an undirected relationship, but more likely, we might expect `x` to predict `y` which can be modeled in a linear regression. The standardized regression coefficient will be the same as `r` in this bivariate example, but we get a p-value and assume the directionality. For the total data and for the first study in the simulated data the model and output look like this:

```
sim_data %>%
  lm(y ~ x, data = .) %>%
  tidy()
```

```
## # A tibble: 2 x 5
##   term        estimate std.error statistic p.value
##   <chr>          <dbl>     <dbl>     <dbl>   <dbl>
## 1 (Intercept)  0.00119   0.00142     0.838  0.402
## 2 x            0.00267   0.00142     1.88   0.0597
```

```
sim_data %>%
  filter(study == 1) %>%
  lm(y ~ x, data = .) %>%
  tidy()
```

```
## # A tibble: 2 x 5
##   term        estimate std.error statistic p.value
##   <chr>          <dbl>     <dbl>     <dbl>   <dbl>
## 1 (Intercept)   -0.132     0.158    -0.832   0.409
## 2 x             -0.118     0.139    -0.853   0.398
```

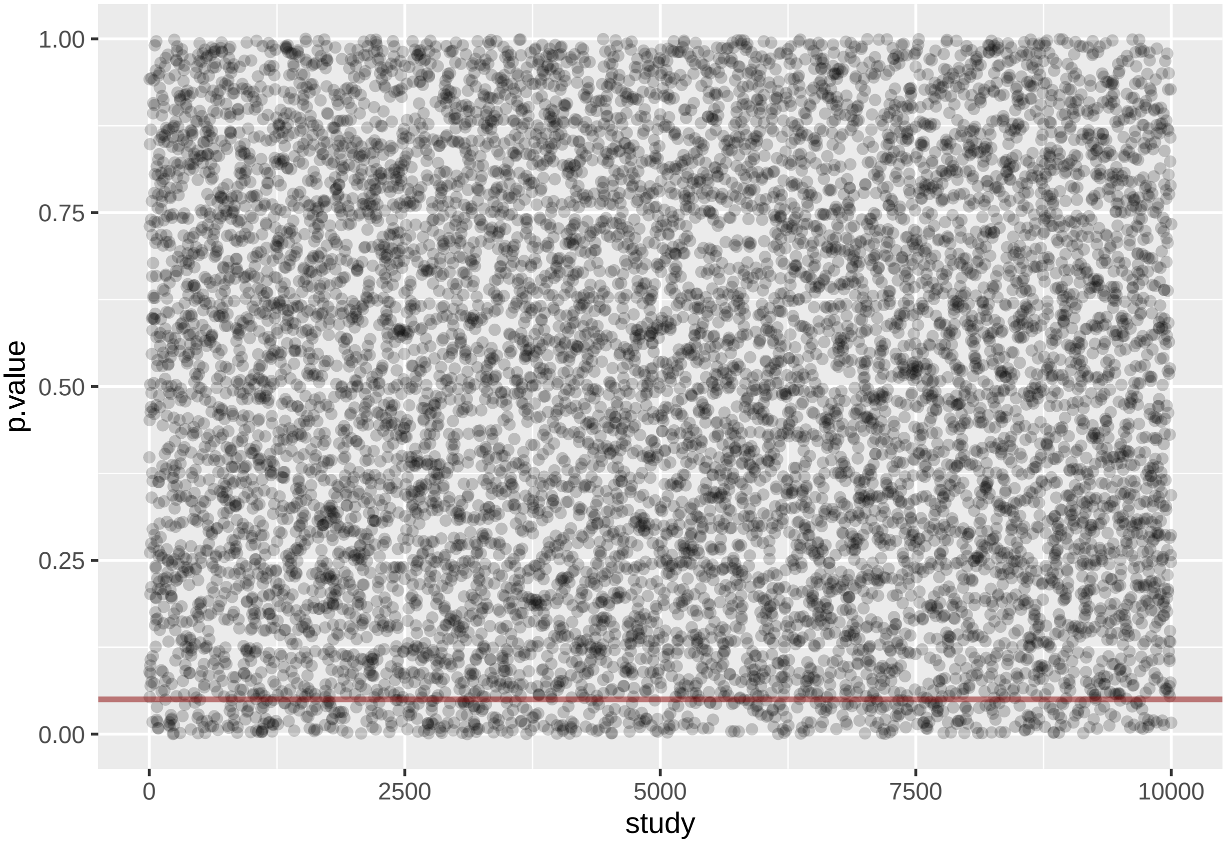
We also learn here that the p-value for the effect of `x` is in the fifth column of the second row of this tidy regression summary. So now let's run the linear model on each of the 10,000 studies and extract the p-values (this takes about 30 seconds on my laptop). We're also attaching the p-values to the first results data frame (the one containing all the correlation coefficients per study).

```
pvalues <- sim_data %>%
  group_by(study) %>%
  summarize(tidy(lm(y ~ x))[2,5]) %>%
  arrange(p.value)

res <- merge(res, pvalues, by = "study")
```

Setting the alpha level at 0.05, we can again see that a lot of studies turned out to be "significant".

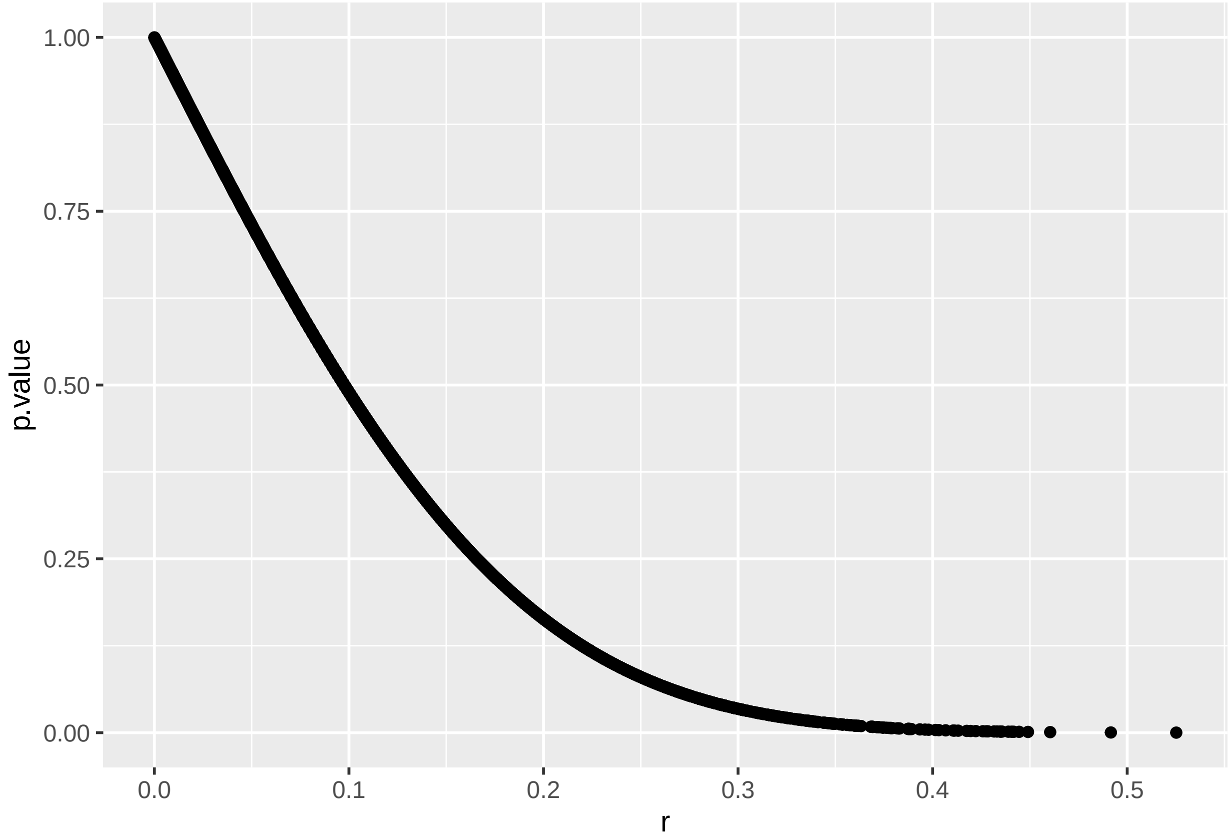
```
co_p <- 0.05
pvalues %>% ggplot(aes(study, p.value)) +
  geom_point(alpha=0.2) +
  geom_hline(yintercept = co_p,
             color = "darkred", lwd = 1, alpha = 0.5)
```



Effect size and significance

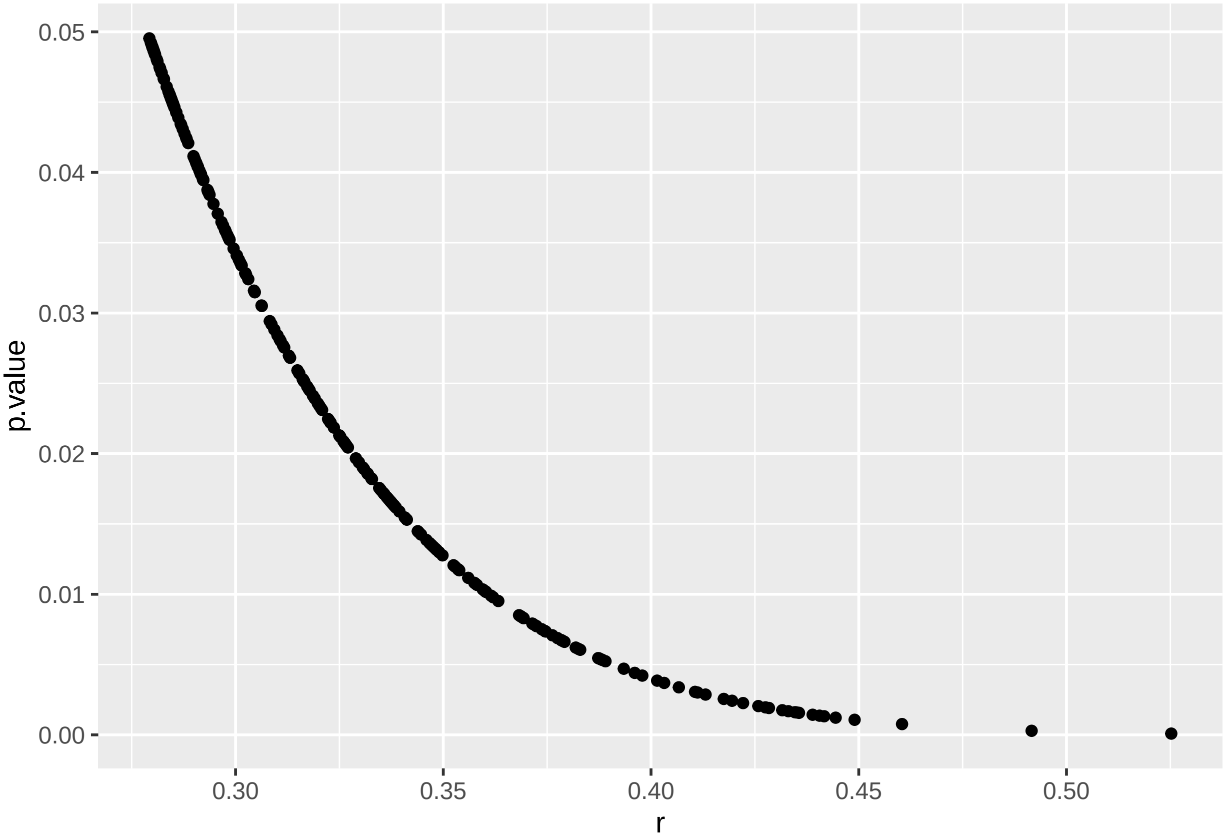
So what is the relationship between effect size or correlation and the p-value (again just looking at the positive values, but it's symmetrical)? Unsurprisingly, larger effects are associated with lower p-values, but not in a linear way.

```
res %>% filter(r >= 0) %>%
  ggplot(aes(r, p.value)) +
  geom_point()
```



We can also zoom in on the bottom right portion of the previous plot by only selecting the substantive and significant effects.

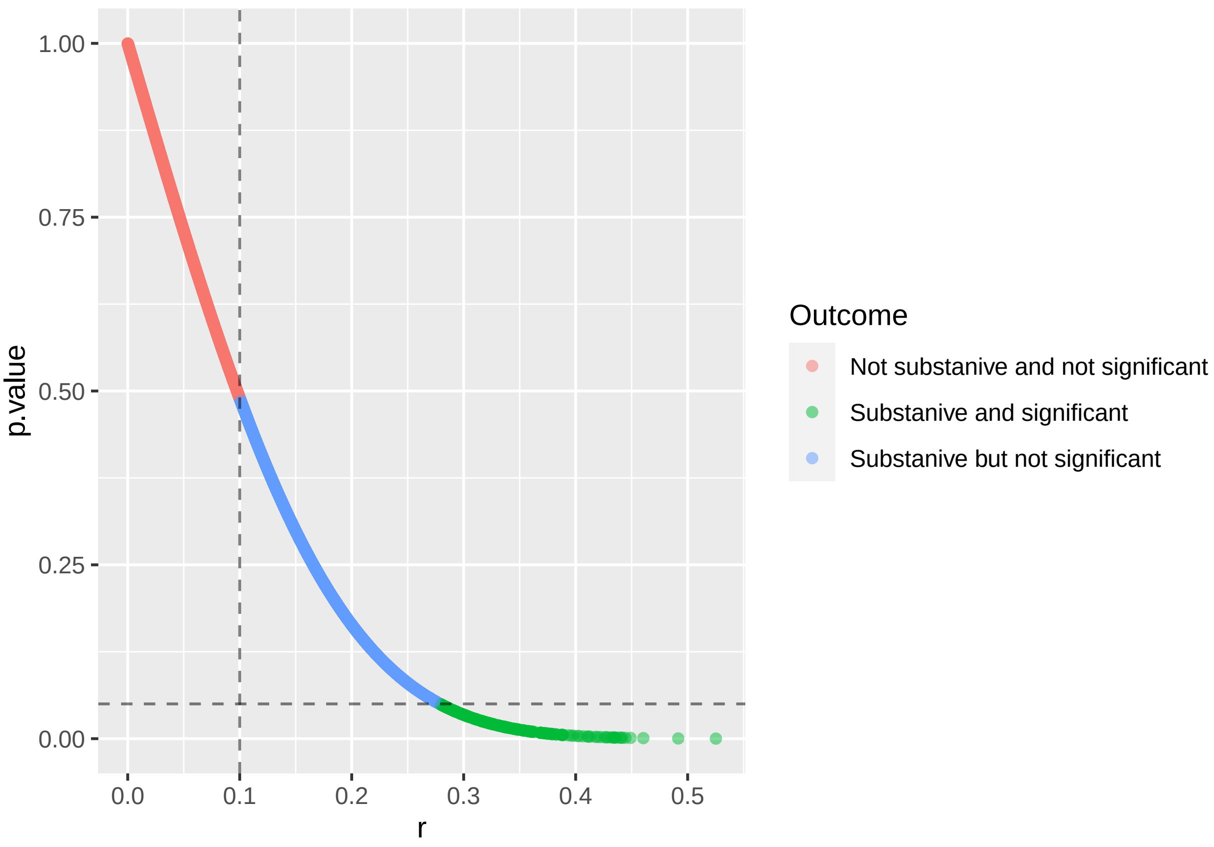
```
res %>% filter(r > co_r & p.value < co_p) %>%
  ggplot(aes(r, p.value)) +
  geom_point()
```



Finally, we can create a new variable `Outcome` to indicate the combinations of effect size and significance cut-offs.

```
res2 <- res %>%
  mutate(Outcome = case_when(
  abs(r) > co_r & p.value < co_p ~ "Substanive and significant",
  abs(r) > co_r & p.value >= co_p ~ "Substanive but not significant",
  abs(r) <= co_r & p.value < co_p ~ "Not substanive but significant",
  abs(r) <= co_r & p.value >= co_p ~ "Not substanive and not significant",
  TRUE ~ "Other"))

res2 %>%
  filter(r >= 0) %>%
  ggplot(aes(r, p.value, color = Outcome)) +
  geom_point(alpha=0.5) +
  geom_hline(yintercept = co_p, lty = 2, alpha = 0.5) +
  geom_vline(xintercept = co_r, lty = 2, alpha = 0.5)
```



False positives

If 10,000 studies were run and only those that find substantive and significant results get published, there is evidently a problem. The probability of finding the “truth”, i.e., a non-substantive effect in a given study, is only about 50%.

```
res2 %>% select(Outcome) %>% table() / s
```

```
## .
## Not substanive and not significant         Substanive and significant
##                             0.5107                             0.0492
##     Substanive but not significant
##                             0.4401
```