# Q Hacking – Stats for Large Data Programming Challenge

Callithrix Jacchus

2023-09-13

## Overview

In this document, we will outline our proposed solution for the Q Hacking Challenge as part of the Large Data Programming session. We will discuss the code and methods employed, concluding with a report of the results.

## Set Up Packages

```
install.packages("ggplot2", repos = "http://cran.us.r-project.org")
```

```
##
## The downloaded binary packages are in
##  /var/folders/yg/g8v7fq4558580kf8lff98jd40000gn/T//RtmpeO8t5i/downloaded_packages
```

```
install.packages("cowplot", repos = "http://cran.us.r-project.org")
```

```
##
## The downloaded binary packages are in
##  /var/folders/yg/g8v7fq4558580kf8lff98jd40000gn/T//RtmpeO8t5i/downloaded_packages
```

```
install.packages("ggrepel", repos = "http://cran.us.r-project.org")
```

```
##
## The downloaded binary packages are in
##  /var/folders/yg/g8v7fq4558580kf8lff98jd40000gn/T//RtmpeO8t5i/downloaded_packages
```

```
install.packages("htmlwidgets", repos = "http://cran.us.r-project.org")
```

```
##
## The downloaded binary packages are in
##  /var/folders/yg/g8v7fq4558580kf8lff98jd40000gn/T//RtmpeO8t5i/downloaded_packages
```

```
install.packages("plotly", repos = "http://cran.us.r-project.org")
```

```
##
## The downloaded binary packages are in
##  /var/folders/yg/g8v7fq4558580kf8lff98jd40000gn/T//RtmpeO8t5i/downloaded_packages
```

## Loading the Data

We need to load the required covid data.

```
library(dplyr)
```

```
##
## Attaching package: 'dplyr'
```

```
## The following objects are masked from 'package:stats':
##
##     filter, lag
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
data <- read.csv("https://covid.ourworldindata.org/data/owid-covid-data.csv", na.strings = "",header=T)
```

## Selecting a subset of data

As outlined in the guidelines, it's not necessary to conduct p-hacking on the entire dataset. Therefore, we opted to focus on a specific country for our analysis. In this example, we examine new COVID-19 cases in the United States across all available dates. To analyze data from a different country, one can simply modify the location parameter.

```
data_country <- data[which(data$location=="United States"),] # One can modify the location parameter if

data_newcases <- data_country %>% select("new_cases")

new_cases <- data_newcases$new_cases

vl <- length(new_cases)

print(paste("The length of the data selected is:", vl))
```

```
## [1] "The length of the data selected is: 1343"
```

## Let's hack P Value

We opt to create a custom variable by populating the time series with random values. We employ a 'while' loop to continuously regenerate these values until the p-value meets the significance threshold.

```
set.seed(123)

y <- new_cases

p_value <- 1

while (p_value >= 0.05) {

  x <- cumsum(rnorm(vl))

  model <- lm(y ~ x)

  model_summary <- summary(model)

  p_value <- model_summary$coefficients[2, 4]

  print(paste("P is:", p_value))
}
```

```
## [1] "P is: 0.000677210298945013"
```

```
# x should be a variable and y has a "significant" relationship with x
print("Done")
```

```
## [1] "Done"
```

```
print(paste("The final P is:", p_value))
```

```
## [1] "The final P is: 0.000677210298945013"
```

```
print(summary(model))
```

```
##
## Call:
## lm(formula = y ~ x)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -101459  -63992  -39886   26844 1169634
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  98076.8     5485.7  17.879  < 2e-16 ***
## x            -1268.2      372.2  -3.407 0.000677 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 124700 on 1232 degrees of freedom
##   (109 observations deleted due to missingness)
## Multiple R-squared:  0.009336,   Adjusted R-squared:  0.008531
## F-statistic: 11.61 on 1 and 1232 DF,  p-value: 0.0006772
```

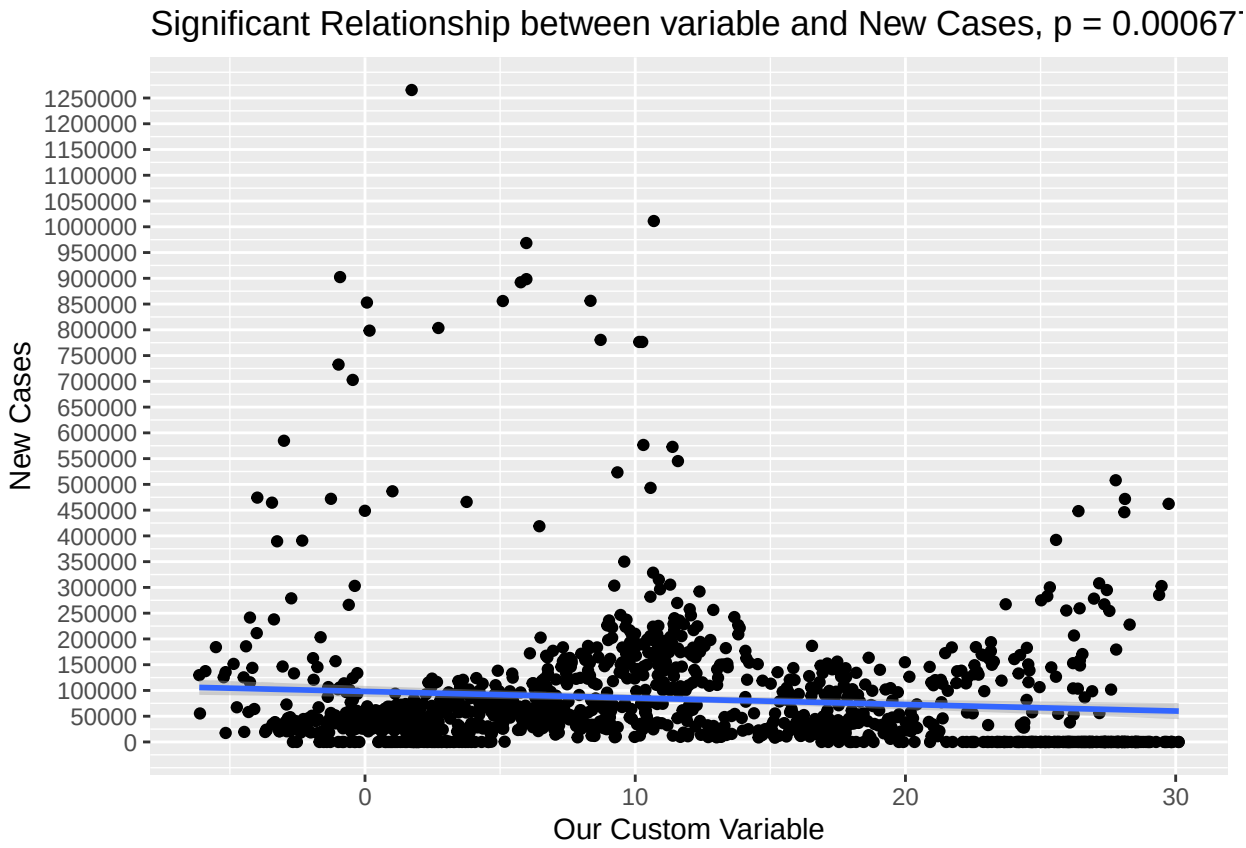
## Plotting

```
library(ggplot2)

data_to_plot <- data.frame("Ours" = x, "Case" = y)

ggplot(data_to_plot, aes(x = Ours, y = Case)) +
  geom_point() +
  geom_smooth(method = "lm") +
  scale_y_continuous(breaks = seq(min(data_to_plot$Case, na.rm = TRUE), max(data_to_plot$Case, na.rm = T
  ggtitle(paste("Significant Relationship between variable and New Cases, p =", p_value)) +
  xlab("Our Custom Variable") +
  ylab("New Cases")
```

```
## `geom_smooth()` using formula = 'y ~ x'
```

```
## Warning: Removed 109 rows containing non-finite values (`stat_smooth()`).
```

```
## Warning: Removed 109 rows containing missing values (`geom_point()`).
```


Significant Relationship between variable and New Cases, p = 0.00067
New Cases
1250000
1200000
1150000
1100000
1050000
1000000
950000
900000
850000
800000
750000
700000
650000
600000
550000
500000
450000
400000
350000
300000
250000
200000
150000
100000
50000
0
0
10
20
30
Our Custom Variable