DataFest 2020

Change in Flights Offered and Passengers Enplaned due to COVID-19 and Their Impact on the Airline Industry

Catherine Jennifer

5/5/2020

Step 1: Creating and cleaning the dataset

1A — COVID dataset

```
covid_data <- read.csv("covid-us.csv")
head(covid_data)
```

```
##         date cases deaths
## 1 2020-01-21     1      0
## 2 2020-01-22     1      0
## 3 2020-01-23     1      0
## 4 2020-01-24     2      0
## 5 2020-01-25     3      0
## 6 2020-01-26     5      0
```

```
tail(covid_data)
```

```
##           date   cases deaths
## 108 2020-05-07 1263943  75734
## 109 2020-05-08 1291528  77308
## 110 2020-05-09 1316443  78762
## 111 2020-05-10 1336754  79693
## 112 2020-05-11 1354350  80682
## 113 2020-05-12 1376683  82350
```

1B — Commercial and total flights datasets

```
# Commercial flights
com_flights <- read.csv("commercial-flights.csv")
com_flights_old <- read.csv("commercial-flights-old.csv")
com_flights_old <- com_flights_old[1:9,]
com_flights <- rbind(com_flights_old, com_flights)
rm(com_flights_old)
colnames(com_flights) <- c("Date", "7_Day_Moving_Average", "Number_of_Flights")
head(com_flights)
```

```
##         Date 7_Day_Moving_Average Number_of_Flights
## 1 2020-02-05               104168            103257
## 2 2020-02-06               103129            104450
## 3 2020-02-07               102282            106693
## 4 2020-02-08               102029             94955
```

```
## 5 2020-02-09                101403            92966
## 6 2020-02-10                100733           100456
```

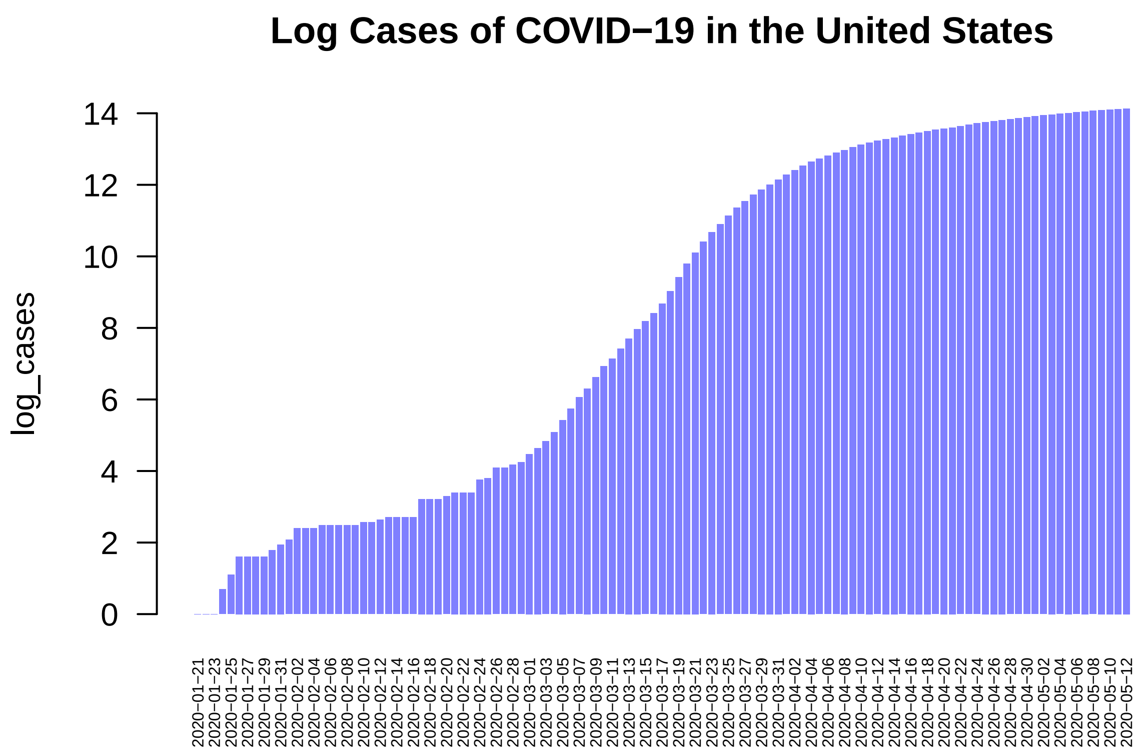
```
# Total flights
total_flights <- read.csv("total-flights.csv")
total_flights_old <- read.csv("total-flights-old.csv")
total_flights_old <- total_flights_old[1:9,]
total_flights <- rbind(total_flights_old, total_flights)
rm(total_flights_old)
colnames(total_flights) <- c("Date", "7_Day_Moving_Average", "Number_of_Flights")
head(total_flights)
```

```
##         Date 7_Day_Moving_Average Number_of_Flights
## 1 2020-02-05               172705            174865
## 2 2020-02-06               170661            171934
## 3 2020-02-07               170184            181611
## 4 2020-02-08               172443            169881
## 5 2020-02-09               171584            152651
## 6 2020-02-10               168731            159472
```

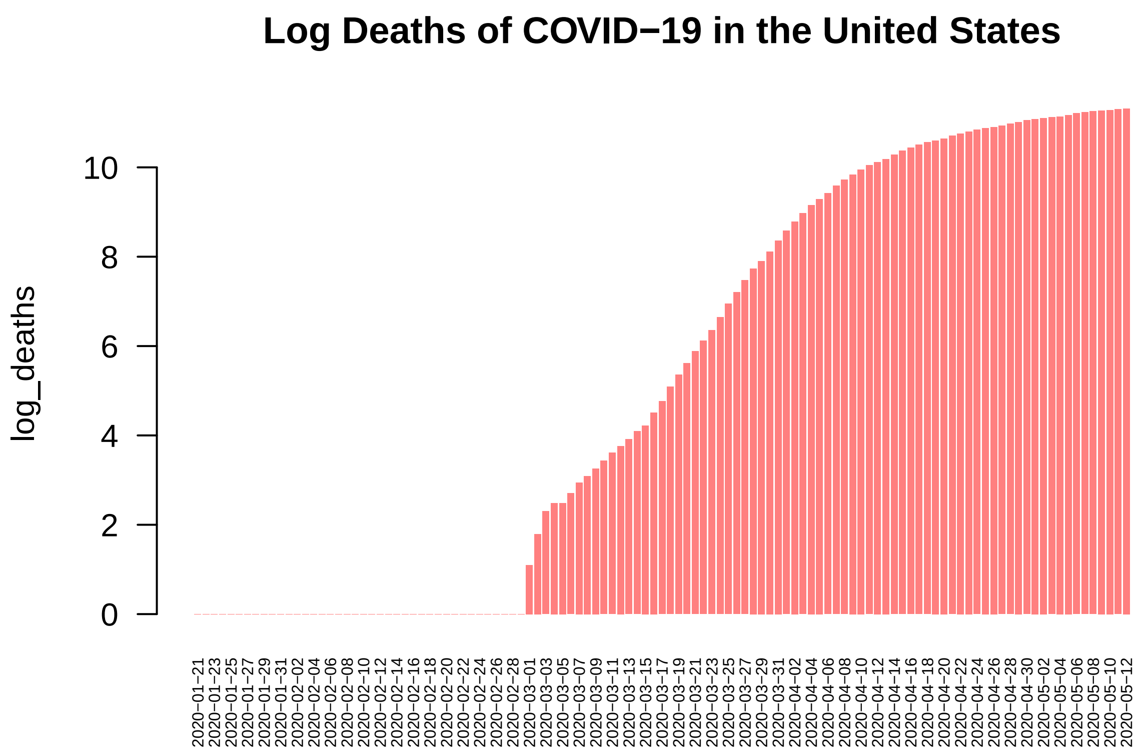
Step 2: Exploratory data analysis

2A — Plotting `covid_data`

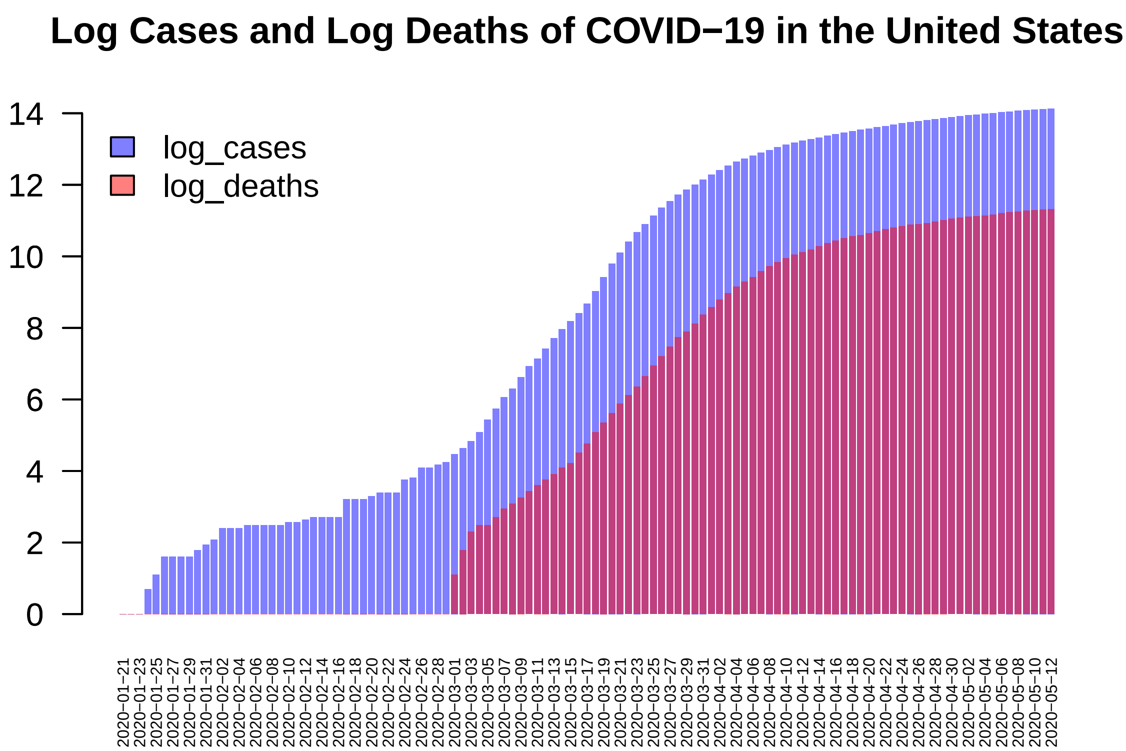
```
# log of COVID cases
log_cases <- log(covid_data$cases)

# setting two semi-transparent colors
blue <- rgb(0, 0, 1, alpha = 0.5)
red <- rgb(1, 0, 0, alpha = 0.5)

barplot(log_cases, main = "Log Cases of COVID-19 in the United States",
        ylab = "log_cases",
        names.arg = covid_data$date,
        col = blue,
        border = NA,
        cex.names = 0.5,
        las = 2
)
```




```
# log of COVID deaths
log_deaths <- log(covid_data$deaths)

# remove -Inf from converting to log scale
log_deaths[c(1:39)] <- 0

barplot(log_deaths, main = "Log Deaths of COVID-19 in the United States",
        ylab = "log_deaths",
        names.arg = covid_data$date,
        col = red,
        border = NA,
        cex.names = 0.5,
        las = 2
)
```



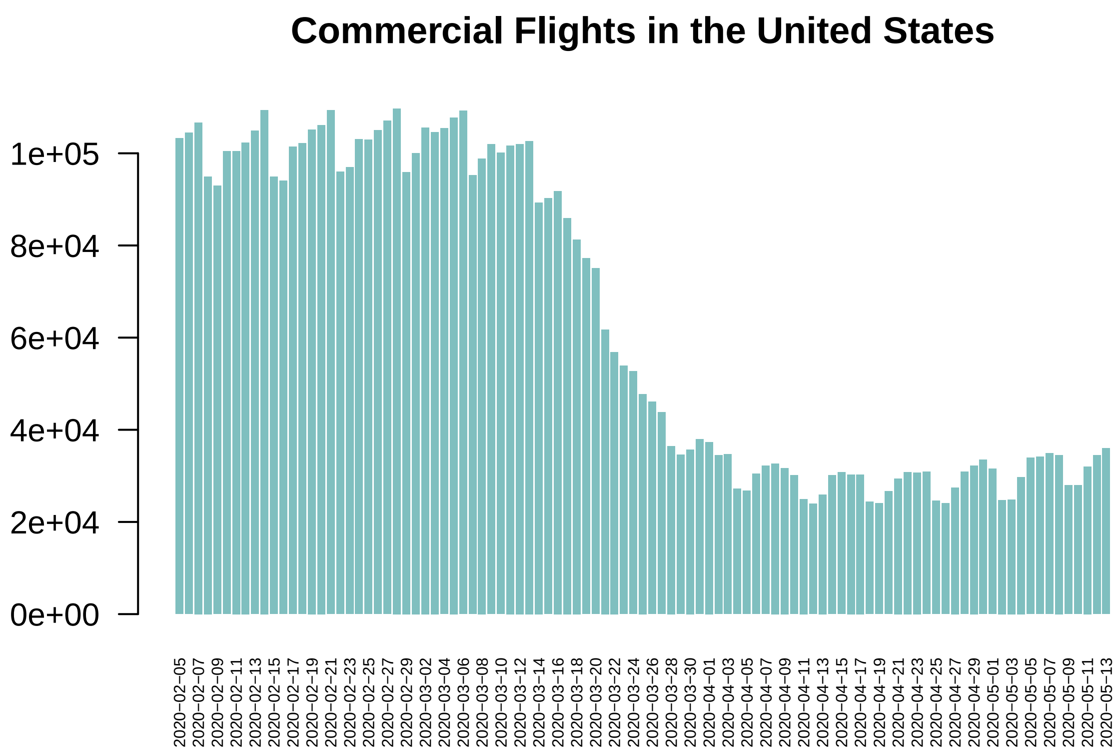

```r
# plot the two datasets together
barplot(log_cases, main = "Log Cases and Log Deaths of COVID-19 in the United States",
        col = blue,
        border = NA,
        names.arg = covid_data$date,
        cex.names = 0.5,
        las = 2
)
barplot(log_deaths,
        col = red,
        border = NA,
        las = 1,
        add = TRUE
)
legend("topleft",
       legend = c("log_cases", "log_deaths"),
       fill = c(blue, red),
       bty = "n")
```



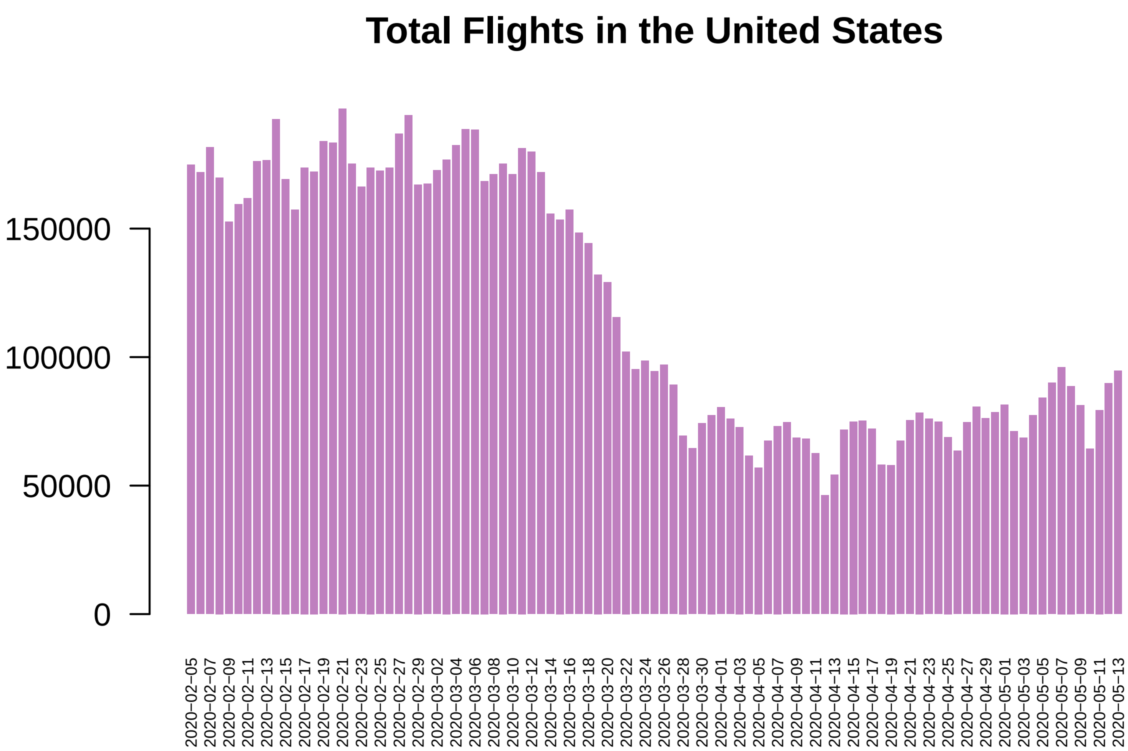
Explain why transforming case and death counts to the logarithmic scale makes sense.

2B — Plotting `com_flights` and `total_flights`

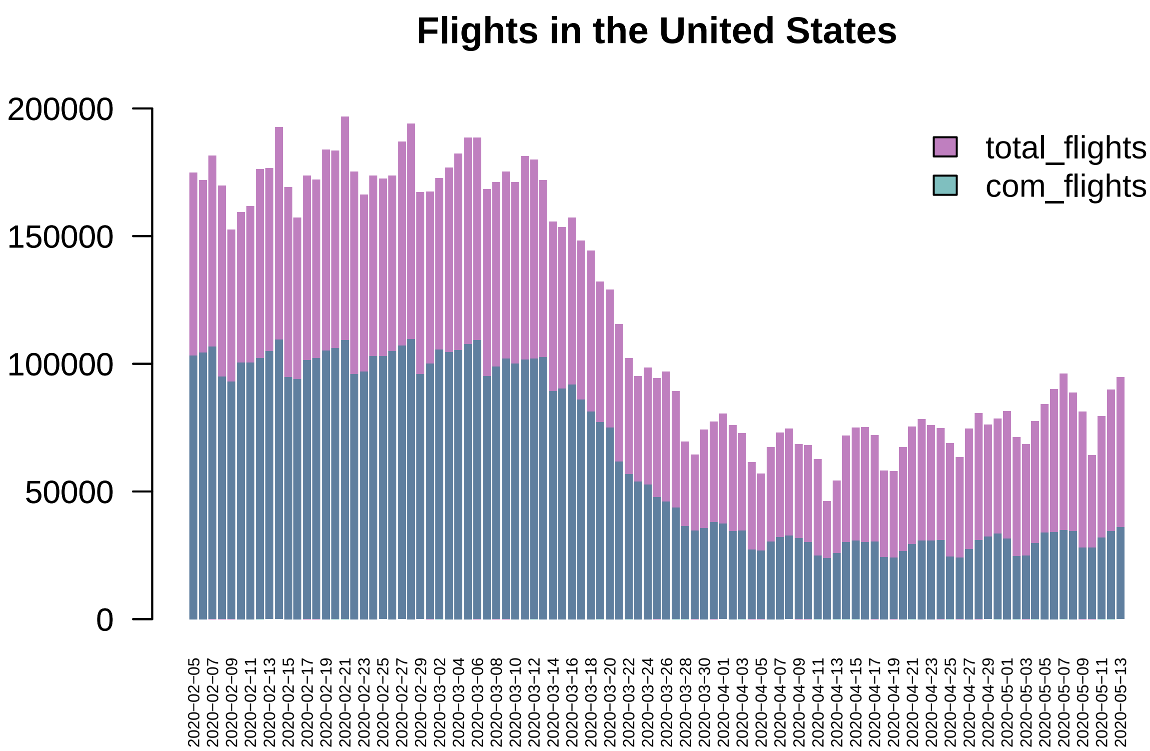
```r
# setting two semi-transparent colors
cyan <- rgb(0, 0.5, 0.5, alpha = 0.5)
magenta <- rgb(0.5, 0, 0.5, alpha = 0.5)

# plotting com_flights
barplot(com_flights$Number_of_Flights, main = "Commercial Flights in the United States",
        names.arg = com_flights$Date,
        col = cyan,
        border = NA,
        cex.names = 0.5,
        las = 2
)
```




```r
# plotting total_flights
barplot(total_flights$Number_of_Flights, main = "Total Flights in the United States",
        names.arg = total_flights$Date,
        col = magenta,
        border = NA,
        cex.names = 0.5,
        las = 2
)
```



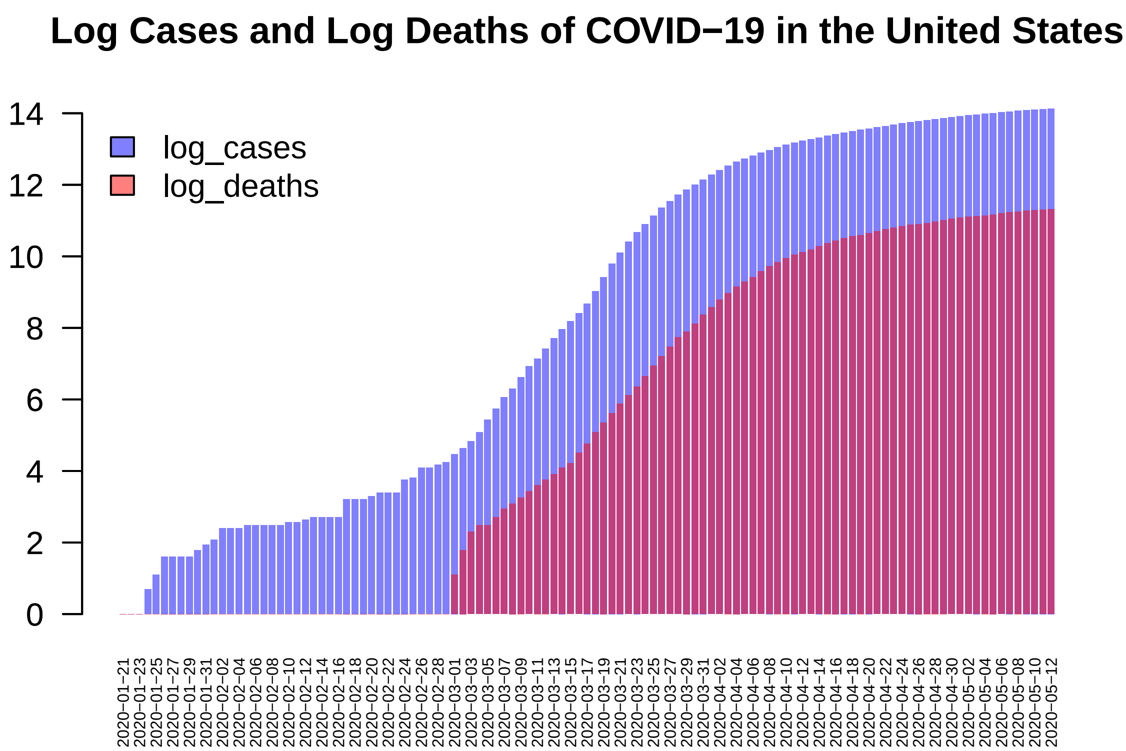

```r
# plotting com_flights and total_flights together
barplot(total_flights$Number_of_Flights,
        main = "Flights in the United States",
        col = magenta,
        border = NA,
        names.arg = total_flights$Date,
        cex.names = 0.5,
        las = 2,
        ylim = c(0,200000)
)
barplot(com_flights$Number_of_Flights,
        col = cyan,
        border = NA,
        las = 1,
        add = TRUE
)
legend("topright",
       legend = c("total_flights", "com_flights"),
       fill = c(magenta, cyan),
       bty = "n")
```



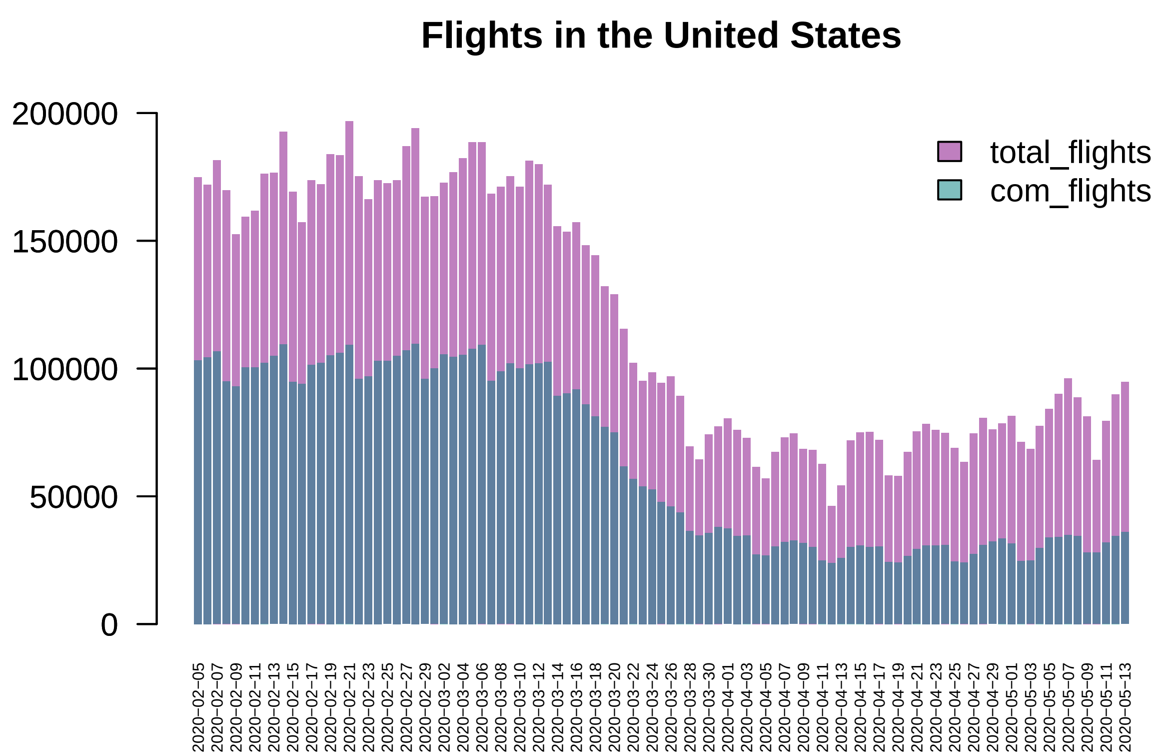
Step 3: Comparing COVID and Flights

3A — Initial visualization

```
barplot(log_cases, main = "Log Cases and Log Deaths of COVID-19 in the United States",
        col = blue,
        border = NA,
        names.arg = covid_data$date,
        cex.names = 0.5,
        las = 2
)
barplot(log_deaths,
        col = red,
        border = NA,
        las = 1,
        add = TRUE
)
legend("topleft",
       legend = c("log_cases", "log_deaths"),
       fill = c(blue, red),
       bty = "n")
```




```r
barplot(total_flights$Number_of_Flights,
        main = "Flights in the United States",
        col = magenta,
        border = NA,
        names.arg = total_flights$Date,
        cex.names = 0.5,
        las = 2,
        ylim = c(0,200000)
)
barplot(com_flights$Number_of_Flights,
        col = cyan,
        border = NA,
        las = 1,
        add = TRUE
)
legend("topright",
       legend = c("total_flights", "com_flights"),
       fill = c(magenta, cyan),
       bty = "n")
```



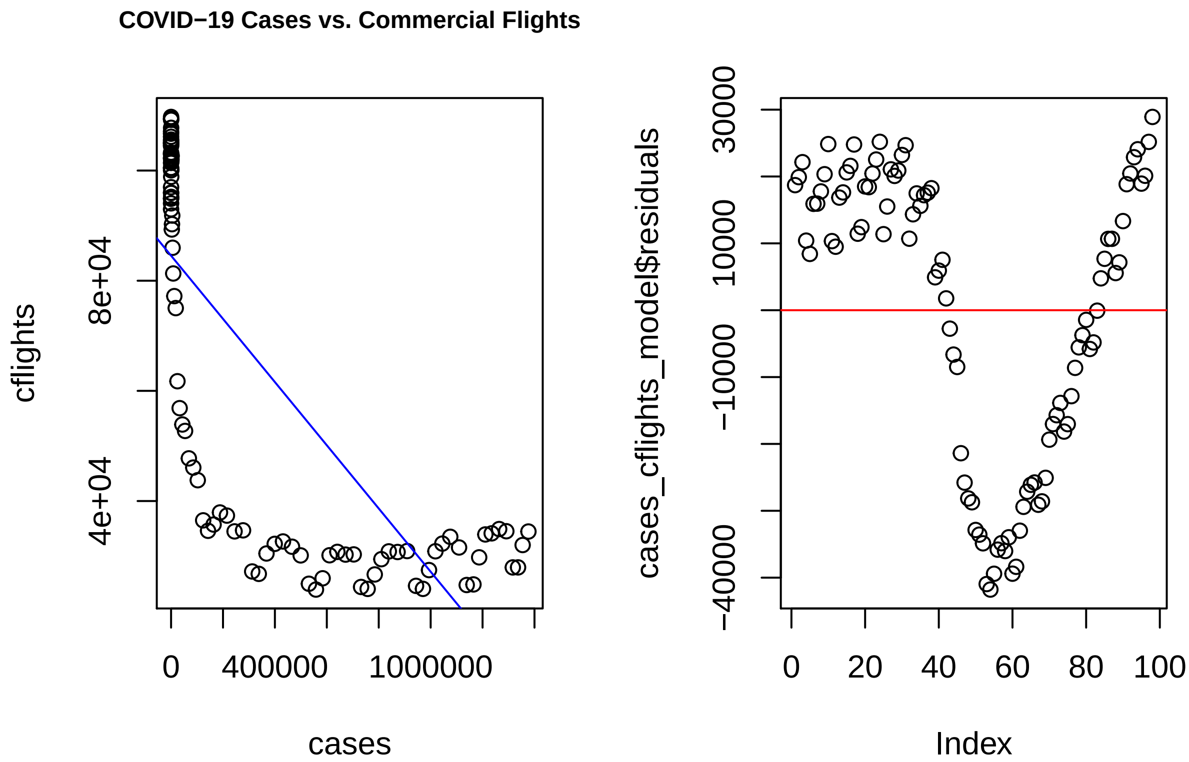
3B — Create master dataset

```r
master <- cbind(covid_data[16:113,],
                com_flights$Number_of_Flights[1:98],
                total_flights$Number_of_Flights[1:98])
colnames(master) <- c("date","cases","deaths","cflights","tflights")
row.names(master) <- NULL
head(master)
```

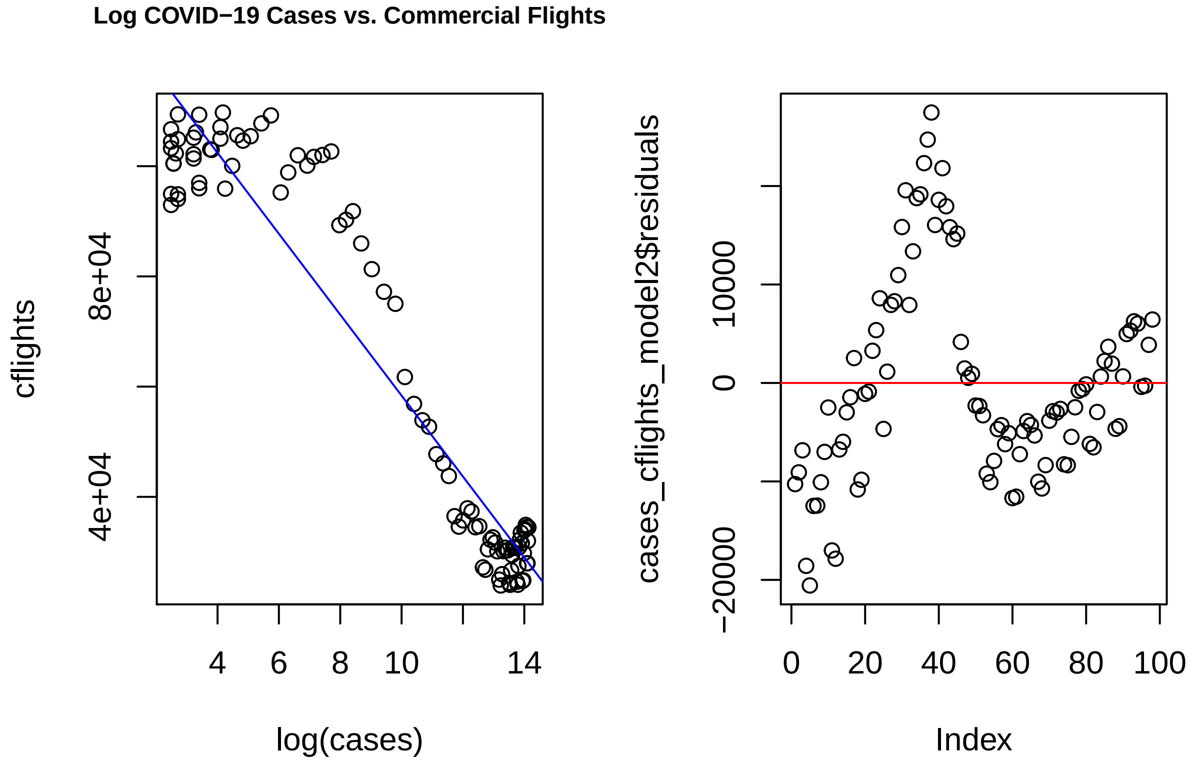
```
##         date cases deaths cflights tflights
## 1 2020-02-05    12      0   103257   174865
## 2 2020-02-06    12      0   104450   171934
## 3 2020-02-07    12      0   106693   181611
## 4 2020-02-08    12      0    94955   169881
## 5 2020-02-09    12      0    92966   152651
## 6 2020-02-10    13      0   100456   159472
```

3C — Diagnose relationship between cases and cflights, i.e. if cases affect cflights

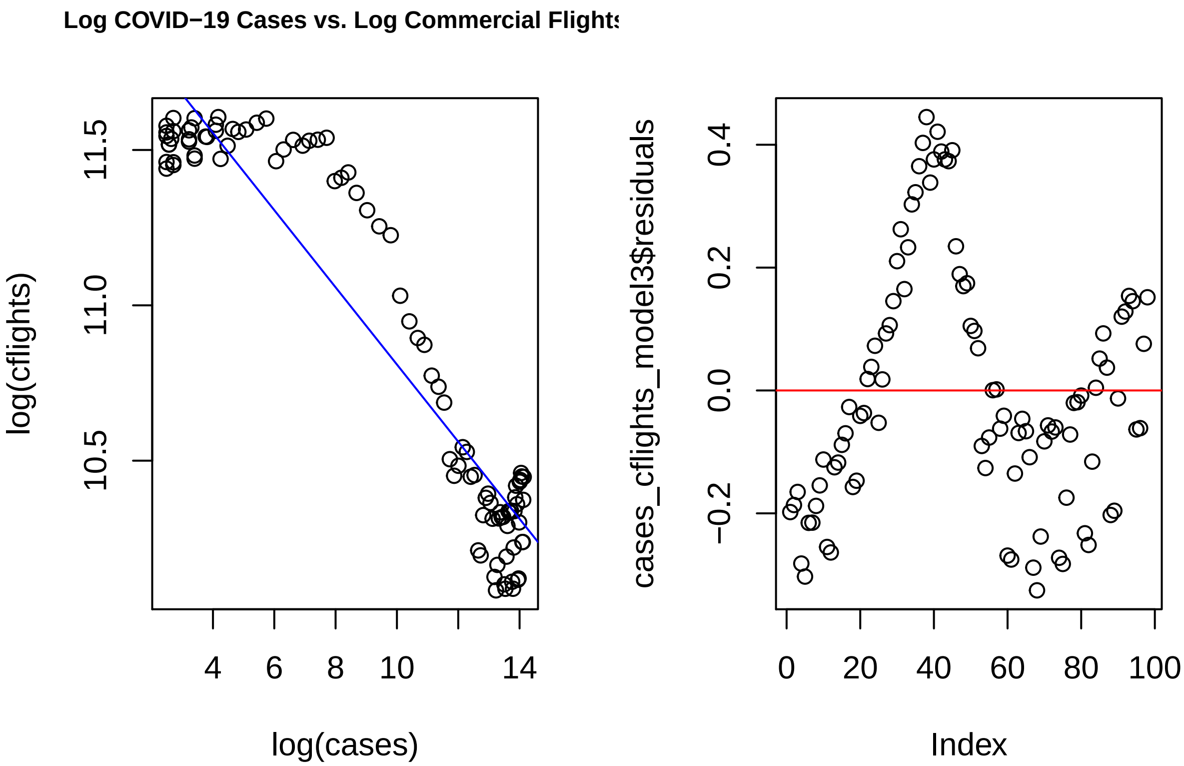
```r
# linear model, no transform
par(mfrow = c(1,2))
plot(cflights ~ cases, data = master,
     main = "COVID-19 Cases vs. Commercial Flights",
     cex.main = 0.75)
cases_cflights_model <- lm(cflights ~ cases, data = master)
abline(cases_cflights_model, col = "blue")
plot(cases_cflights_model$residuals)
abline(h = 0, col = "red")
```



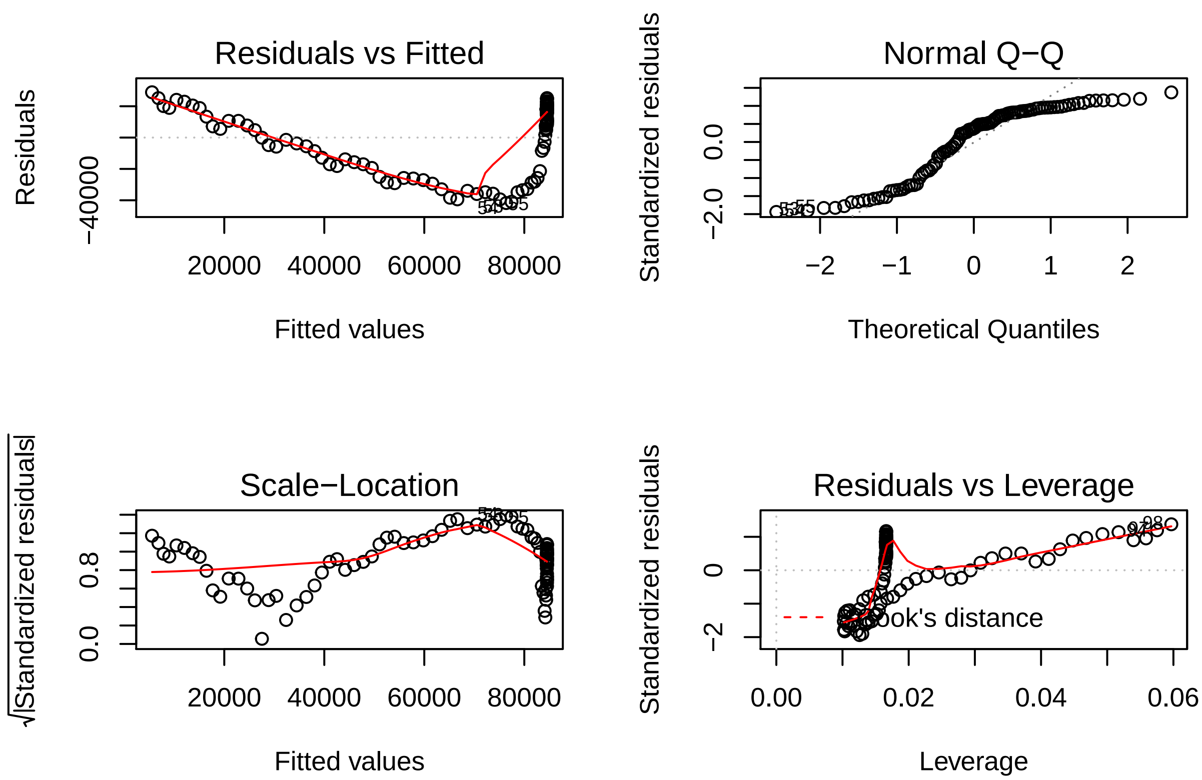

```r
# linear, log transform
plot(cflights ~ log(cases), data = master,
     main = "Log COVID-19 Cases vs. Commercial Flights",
     cex.main = 0.75)
cases_cflights_model2 <- lm(cflights ~ log(cases), data = master)
abline(cases_cflights_model2, col = "blue")
plot(cases_cflights_model2$residuals)
abline(h = 0, col = "red")
```



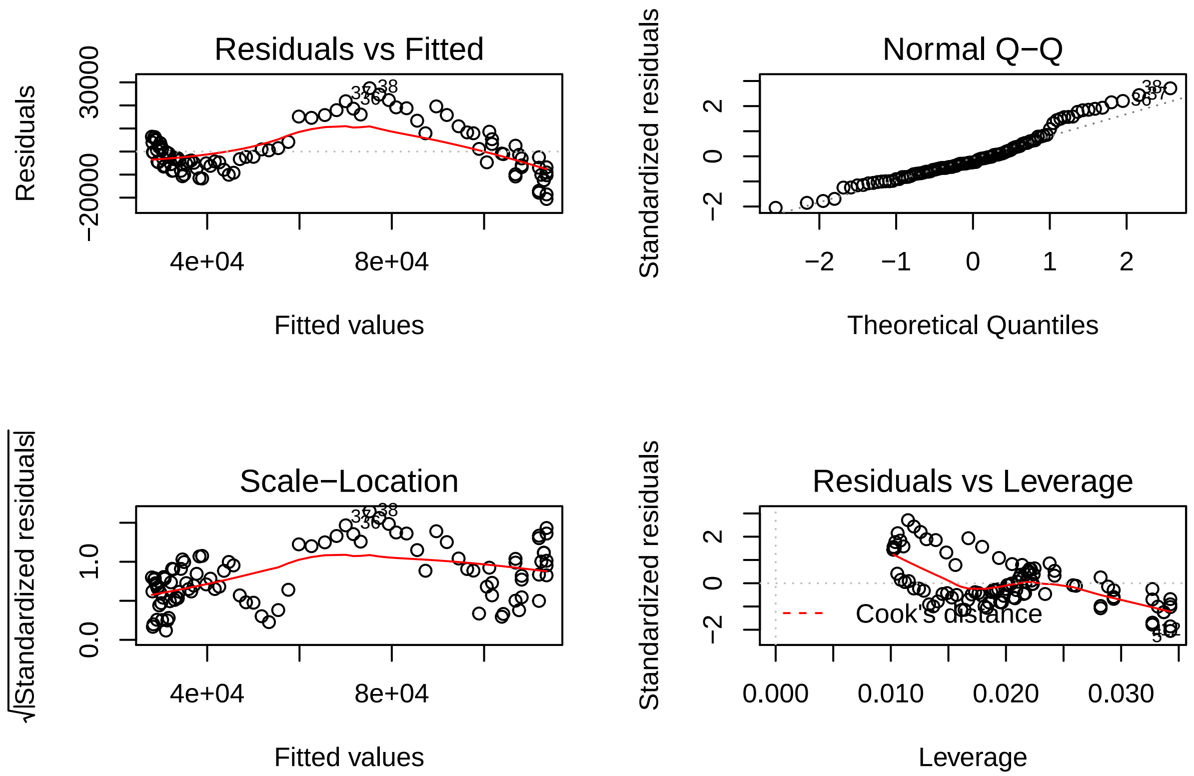

```r
# linear, log-log transform
plot(log(cflights) ~ log(cases), data = master,
     main = "Log COVID-19 Cases vs. Log Commercial Flights",
     cex.main = 0.75)
cases_cflights_model3 <- lm(log(cflights) ~ log(cases), data = master)
abline(cases_cflights_model3, col = "blue")
plot(cases_cflights_model3$residuals)
abline(h = 0, col = "red")
```



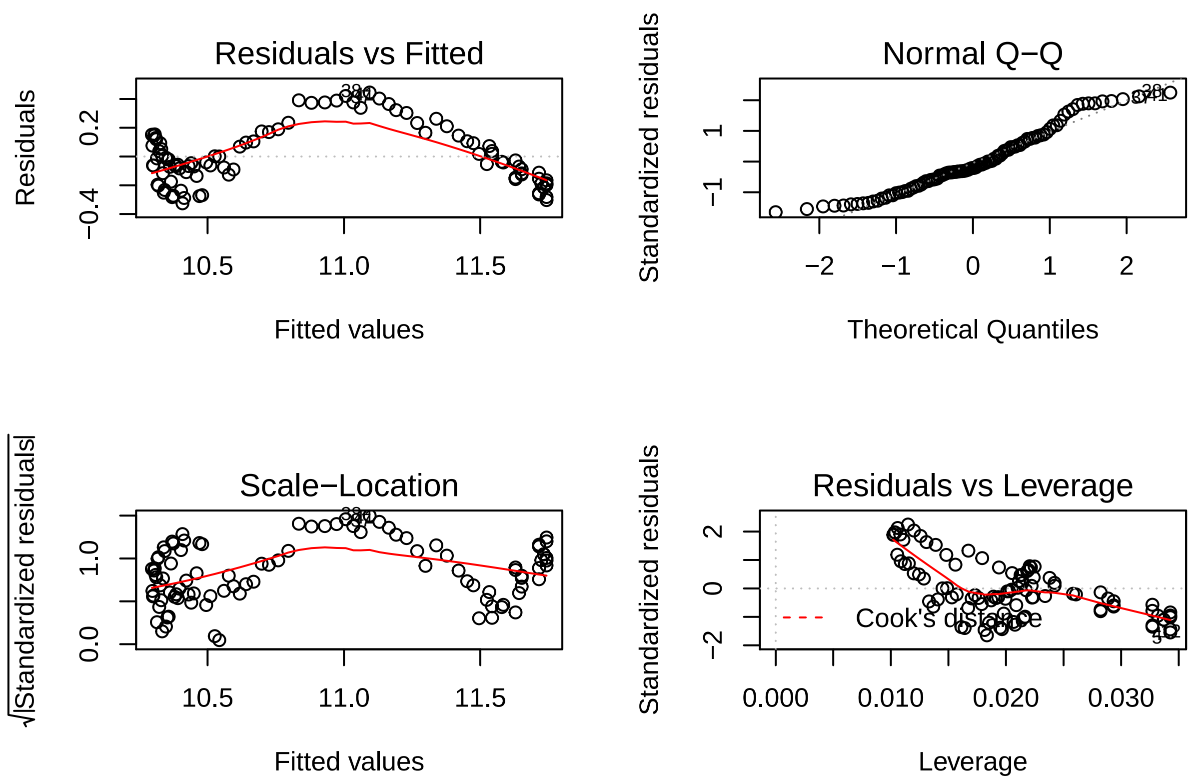

```r
# diagnostic plots
par(mfrow = c(2,2))
plot(cases_cflights_model)
```



```r
plot(cases_cflights_model2)
```




```r
plot(cases_cflights_model3)
```



```r
# summaries
options(digits = 8)
summary(cases_cflights_model)
```

```
##
## Call:
## lm(formula = cflights ~ cases, data = master)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -41782.4 -19057.8   8055.6  18485.6  28909.3
##
## Coefficients:
##                Estimate  Std. Error t value  Pr(>|t|)
## (Intercept)  8.4547e+04  2.7940e+03  30.261 < 2.2e-16 ***
## cases       -5.7380e-02  4.7636e-03 -12.046 < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 21684 on 96 degrees of freedom
## Multiple R-squared:  0.60182,    Adjusted R-squared:  0.59767
## F-statistic:  145.1 on 1 and 96 DF,  p-value: < 2.22e-16
```

```r
summary(cases_cflights_model2)
```

```
##
## Call:
## lm(formula = cflights ~ log(cases), data = master)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -20566.1  -6700.7  -2405.9   5227.7  27468.2
##
## Coefficients:
##               Estimate Std. Error t value  Pr(>|t|)
## (Intercept) 131771.57    2392.94  55.067 < 2.2e-16 ***
## log(cases)   -7340.12     232.75 -31.536 < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 10195 on 96 degrees of freedom
## Multiple R-squared:  0.91197,    Adjusted R-squared:  0.91105
## F-statistic: 994.54 on 1 and 96 DF,  p-value: < 2.22e-16
```

```r
summary(cases_cflights_model3)
```

```
##
## Call:
## lm(formula = log(cflights) ~ log(cases), data = master)
##
## Residuals:
##       Min        1Q    Median        3Q       Max
## -0.325320 -0.143976 -0.041268  0.126410  0.444970
##
## Coefficients:
```

```
##                Estimate Std. Error t value  Pr(>|t|)
## (Intercept) 12.0515255  0.0467368 257.860 < 2.2e-16 ***
## log(cases)  -0.1242017  0.0045459 -27.322 < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.19913 on 96 degrees of freedom
## Multiple R-squared:  0.88605,    Adjusted R-squared:  0.88486
## F-statistic: 746.48 on 1 and 96 DF,  p-value: < 2.22e-16
```

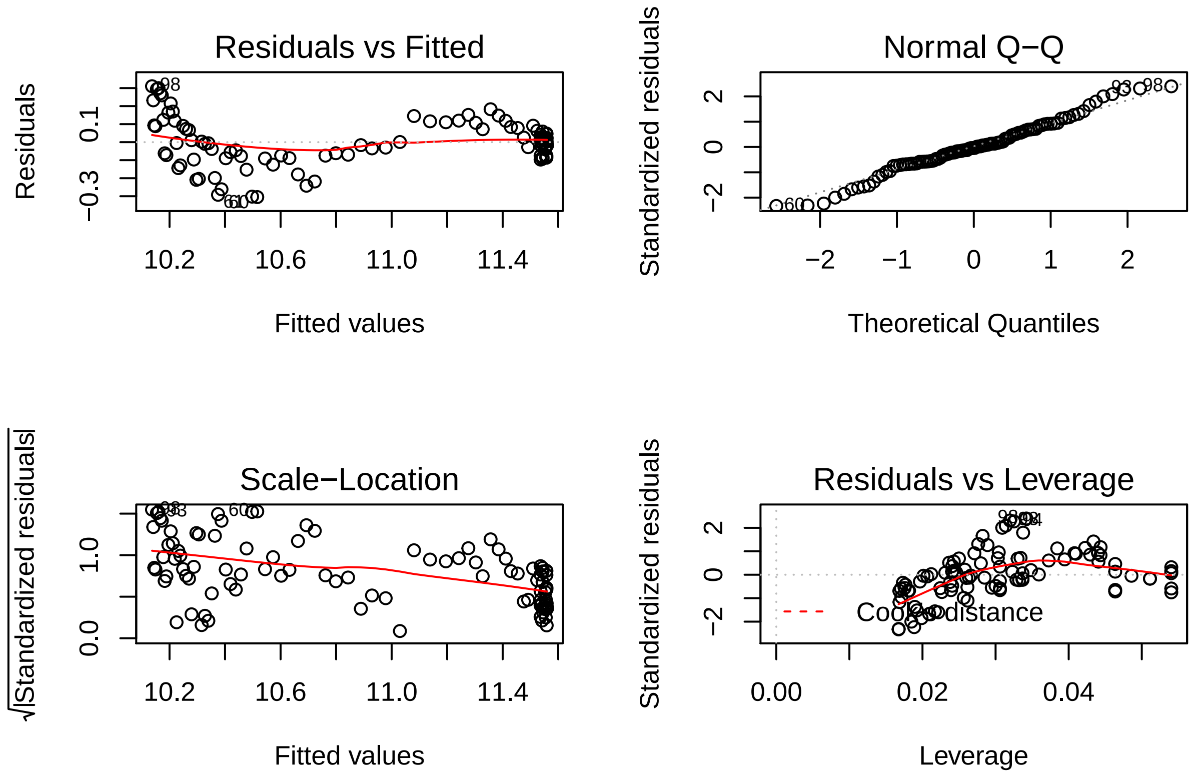
From the plots above, we see that the best transformation out of the three is the log-log model: log `cases` and log `cflights`, which provides easier interpretability and greatly reduced residual standard error. However, the relationship between them are non-linear. Now, let's try fitting a quadratic and a cubic model to the log-log transformed variables.

3C — Fitting a quadratic and a cubic model

```r
# quadratic
quad_model <- lm(log(cflights) ~ log(cases) + I((log(cases))^2), data = master)
summary(quad_model)
```

```
##
## Call:
## lm(formula = log(cflights) ~ log(cases) + I((log(cases))^2),
##     data = master)
##
## Residuals:
##        Min         1Q     Median         3Q        Max
## -0.3046933 -0.0753760 -0.0052846  0.0828606  0.3105973
##
## Coefficients:
##                      Estimate Std. Error  t value  Pr(>|t|)
## (Intercept)       11.3671379  0.0688592 165.0780 < 2.2e-16 ***
## log(cases)         0.1012814  0.0204873   4.9436 3.295e-06 ***
## I((log(cases))^2) -0.0133219  0.0011973 -11.1268 < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.1319 on 95 degrees of freedom
## Multiple R-squared:  0.95053,    Adjusted R-squared:  0.94948
## F-statistic:  912.6 on 2 and 95 DF,  p-value: < 2.22e-16
```

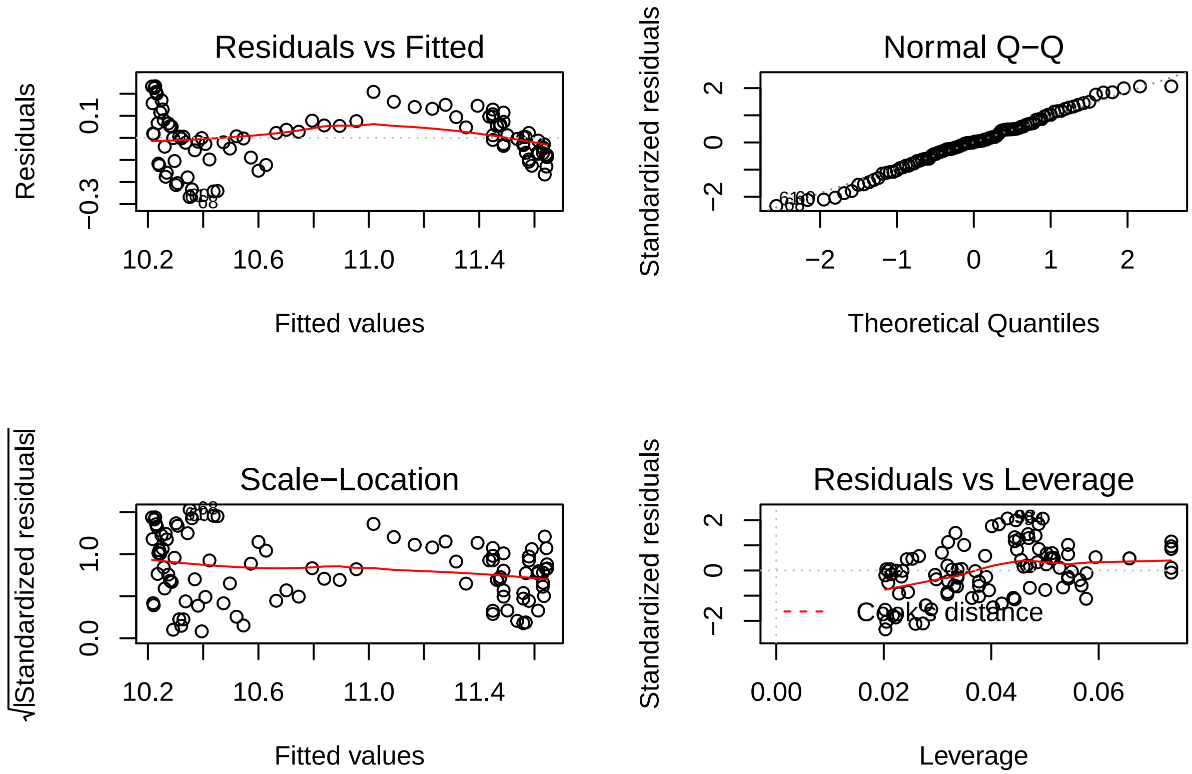
```r
par(mfrow = c(2,2))
plot(quad_model)
```




```r
# cubic
cubic_model <- lm(log(cflights) ~ log(cases) + I((log(cases))^2) +
                    I((log(cases))^3), data = master)
summary(cubic_model)
```

```
##
## Call:
## lm(formula = log(cflights) ~ log(cases) + I((log(cases))^2) +
##     I((log(cases))^3), data = master)
##
## Residuals:
##        Min         1Q     Median         3Q        Max
## -0.2681585 -0.0715173  0.0014353  0.0711179  0.2341932
##
## Coefficients:
##                       Estimate Std. Error t value  Pr(>|t|)
## (Intercept)        10.6699419  0.1425421 74.8547 < 2.2e-16 ***
## log(cases)          0.4550694  0.0679263  6.6995 1.524e-09 ***
## I((log(cases))^2) -0.0617139  0.0090208 -6.8413 7.886e-10 ***
## I((log(cases))^3)  0.0019272  0.0003568  5.4013 4.966e-07 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.11583 on 94 degrees of freedom
## Multiple R-squared:  0.96224,    Adjusted R-squared:  0.96104
## F-statistic: 798.56 on 3 and 94 DF,  p-value: < 2.22e-16
```

```r
par(mfrow = c(2,2))
plot(cubic_model)
```


Residuals vs Fitted
Residuals
Fitted values
Normal Q−Q
Standardized residuals
Theoretical Quantiles
Scale−Location
√|Standardized residuals|
Fitted values
Residuals vs Leverage
Standardized residuals
Cook's distance
Leverage