HW week 12

w203 teaching team
w203: Statistics for Data Science

Arisa Nguyen, Ayman Bari, Emanuel Mejía, Jorge Bonilla

```
library(tidyverse)
library(ggplot2)
library(lmtest)
library(sandwich)
library(stargazer)
library(janitor)
library(car)
```

```
d <- load_and_clean(input = 'videos.txt')
```

```
## Rows: 9618 Columns: 9

## -- Column specification --------------------------------------------------------
## Delimiter: "\t"
## chr (3): video_id, uploader, category
## dbl (6): age, length, views, rate, ratings, comments

##
## i Use ‘spec()‘ to retrieve the full column specification for this data.
## i Specify the column types or set ‘show_col_types = FALSE‘ to quiet this message.
```

```
d %>% summarise(across(everything(),~sum(is.na(.))))
```

```
## # A tibble: 1 x 10
##   video_id uploader   age category length views average_rating count_of_ratings
##      <int>    <int> <int>    <int>  <int> <int>          <int>            <int>
## 1        0        9     9        9      9     9              9                9
## # ... with 2 more variables: comments <int>, log_of_average_rating <int>
```

```
d <- d %>% filter(!is.na(length))
d <- d %>% filter(length < 660)
mean_rating <- mean(d$average_rating, na.rm = TRUE)
d[d$average_rating == 0, "average_rating"] <- mean_rating
```

```
# Count the number of videos where the video id is invalid.
sum(d$video_id == "#NAME?")
```

```
## [1] 128
```

```r
# Count the number of videos that do no have a length of eleven characters.
sum(nchar(as.character(d$video_id)) != 11)
```

```
## [1] 128
```

```r
# Count the number of views recorded as NA
sum(is.na(d$views))
```

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

Regression analysis of YouTube dataset

You want to explain how much the quality of a video affects the number of views it receives on social media. In a world where people can now buy followers and likes, would such an investment increase the number of views that their content receives? **This is a causal question.**

You will use a dataset created by Cheng, Dale and Liu at Simon Fraser University. It includes observations about 9618 videos shared on YouTube. Please see this link for details about how the data was collected.

You will use the following variables:

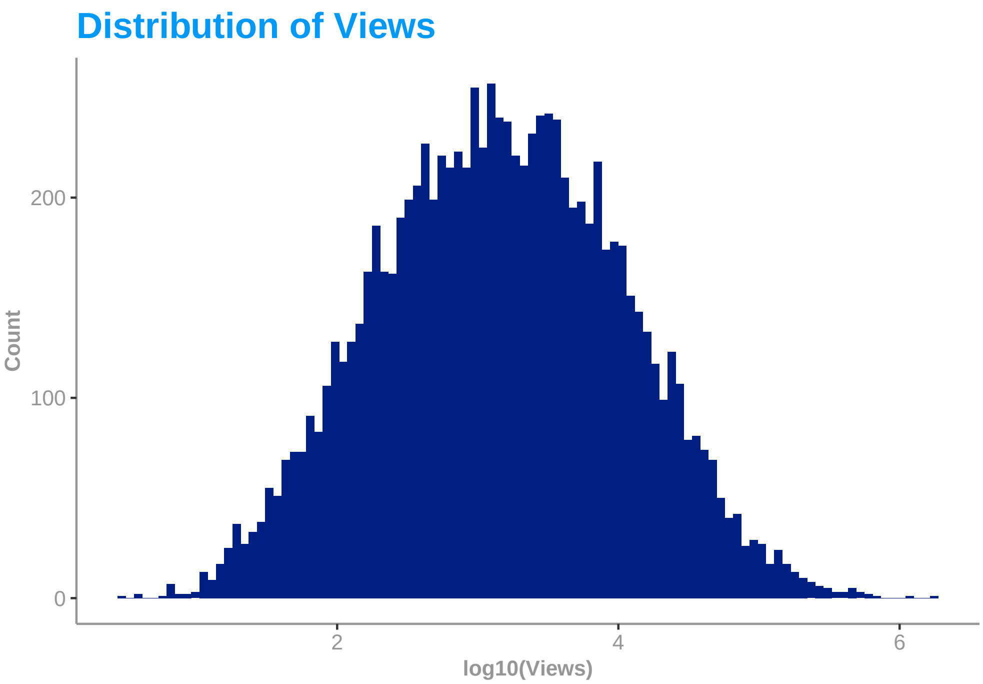
- `views`: the number of views by YouTube users.
- `average_rating`: This is the average of the ratings that the video received, it is a renamed feature from `rate` that is provided in the original dataset. (Notice that this is different from `cout_of_ratings` which is a count of the total number of ratings that a video has received.
- `length:` the duration of the video in seconds.

a. Perform a brief exploratory data analysis on the data to discover patterns, outliers, or wrong data entries and summarize your findings.

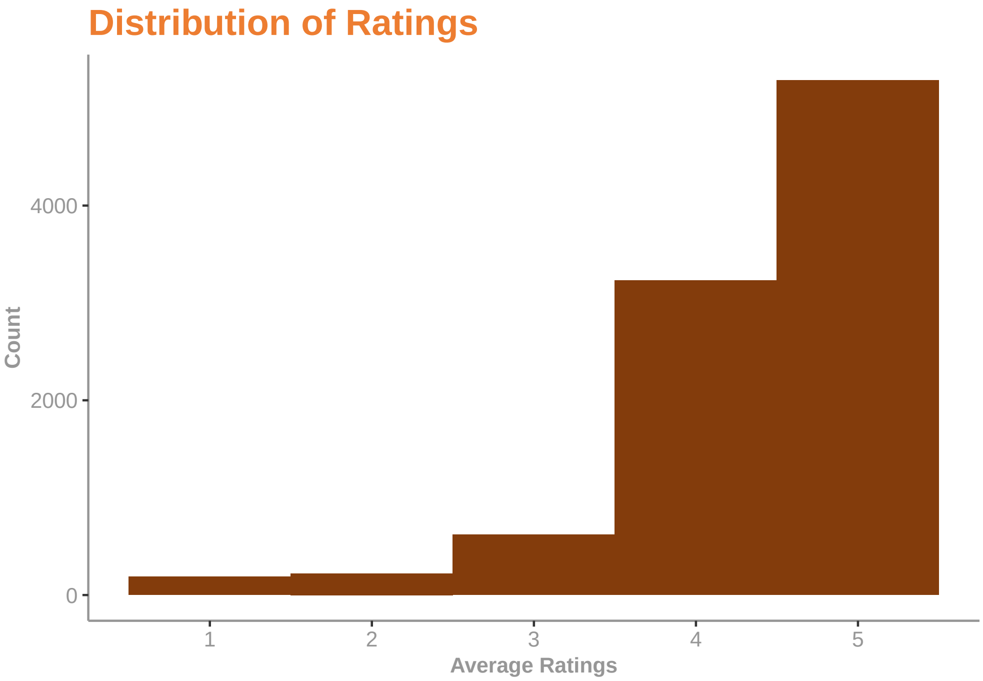
```r
# Data Summary
summary(d)
```

```
##    video_id           uploader              age         category
##  Length:9549        Length:9549        Min.   :   0   Length:9549
##  Class :character   Class :character   1st Qu.: 920   Class :character
##  Mode  :character   Mode  :character   Median :1116   Mode  :character
##                                        Mean   :1045
##                                        3rd Qu.:1227
##                                        Max.   :1258
##      length          views         average_rating  count_of_ratings
##  Min.   :  1.0   Min.   :      3   Min.   :1.000   Min.   :   0.00
##  1st Qu.: 82.0   1st Qu.:    346   1st Qu.:3.740   1st Qu.:   1.00
##  Median :192.0   Median :   1435   Median :4.670   Median :   5.00
##  Mean   :215.1   Mean   :   9300   Mean   :4.323   Mean   :  20.28
##  3rd Qu.:297.0   3rd Qu.:   6100   3rd Qu.:5.000   3rd Qu.:  15.00
##  Max.   :659.0   Max.   :1807640   Max.   :5.000   Max.   :3801.00
##     comments        log_of_average_rating
##  Min.   :   -2.00   Min.   : -Inf
##  1st Qu.:    1.00   1st Qu.:1.218
##  Median :    3.00   Median :1.541
##  Mean   :   19.78   Mean   : -Inf
##  3rd Qu.:   13.00   3rd Qu.:1.609
##  Max.   :13211.00   Max.   :1.609
```

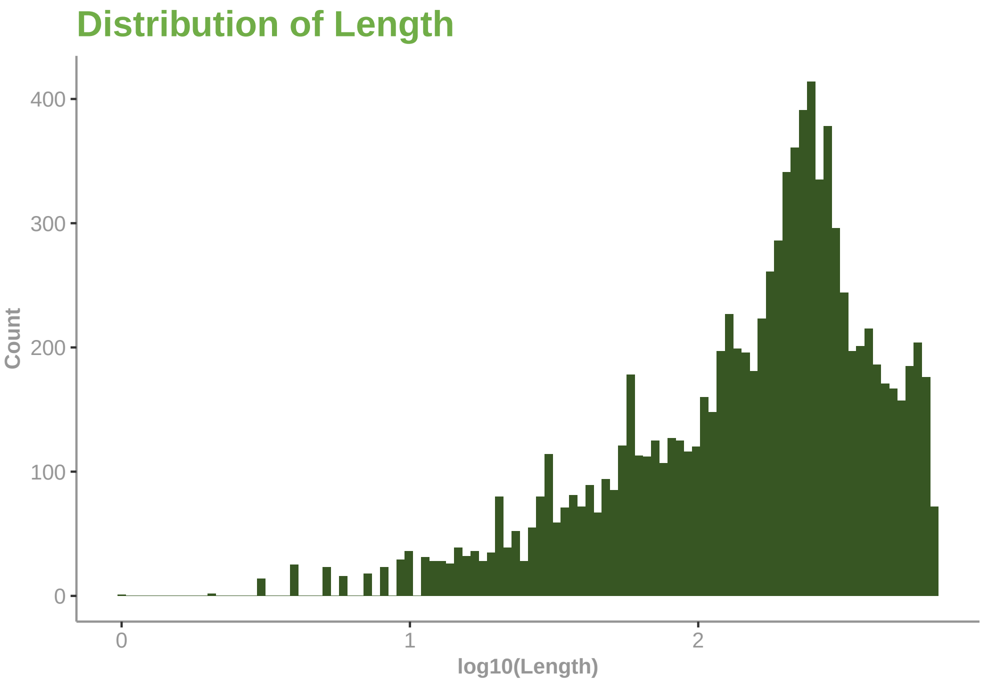
```
# Views
hist_views <- d %>%
  ggplot() +
  aes(x = log10(views)) +
  geom_histogram(bins = 100, fill = "#001F82") +
  labs(title = 'Distribution of Views', x = 'log10(Views)', y = 'Count') +
  theme_classic() +
    theme(
      plot.title = element_text(color = "#0099F8",
                                size = 17,
                                face = "bold"),
      plot.subtitle = element_text(size = 13, face = "italic"),
      axis.title = element_text(color = "#969696",
                                size = 10,
                                face = "bold"),
      axis.text = element_text(color = "#969696", size = 10),
      axis.line = element_line(color = "#969696")
    )

hist_views
```



```
# Rating
hist_rate <- d %>%
  ggplot() +
  aes(x = average_rating) +
  geom_histogram(breaks = seq(0.5,5.5), fill = "#833C0C") +
  labs(title = 'Distribution of Ratings', x = 'Average Ratings', y = 'Count') +
  theme_classic() +
    theme(
      plot.title = element_text(color = "#ED7D31",
                                size = 17,
                                face = "bold"),
      plot.subtitle = element_text(size = 13, face = "italic"),
      axis.title = element_text(color = "#969696",
                                size = 10,
                                face = "bold"),
      axis.text = element_text(color = "#969696", size = 10),
      axis.line = element_line(color = "#969696")
    )
hist_rate
```



```
# Length
hist_length <- d %>%
  ggplot() +
  aes(x = log10(length)) +
  geom_histogram(bins = 100, fill = "#375623") +
  labs(title = 'Distribution of Length', x = 'log10(Length)', y = 'Count') +
  theme_classic() +
    theme(
      plot.title = element_text(color = "#70AD47",
                                size = 17,
                                face = "bold"),
      plot.subtitle = element_text(size = 13, face = "italic"),
      axis.title = element_text(color = "#969696",
                                size = 10,
                                face = "bold"),
      axis.text = element_text(color = "#969696", size = 10),
      axis.line = element_line(color = "#969696")
    )
hist_length
```



‘From this EDA, we learn that there are 9 rows with NA value for views, rating, and length, and should be excluded from further analysis. The minimum value of views is 3 and the maximum is 1,807,640 which is in the realm of possibility. The minimum value of average_ratings is 0 and the maximum value is 5, which is also in the realm of possibility since the Youtube rating system used to range from 0 to 5 stars. The minimum and maximum value for length is 1 second and 5289 seconds (or about 1.5 hours), which also seems plausible. Views and length were both log transformed to normalize its distributions and to eliminate the effect of outliers. After this transformation, views had a normal distribution ranging from 0 to 10ˆ6 views with the peak being at about 10ˆ3 views. Length has a multimodal distribution peaking at about 10ˆ2.4 and 10ˆ2.8 seconds and ranged from 10ˆ0.5 and 10ˆ2.8 seconds. Ratings were distributed between 0 to 5 with the highest peaks occuring at 0 and 5 and smaller peaks at 1, 3, and 4. The high count of videos around whole number ratings may be indicating that many videos receive very few or no rating. Another theory is the ratings tend to cluster around one ratings which may stem from the selection bias associated with viewers having a higher probability to provide low scores to videos they disliked or high ratings to videos they enjoyed. It appears that the most viewers opt for binary ratings as represented in the bimodal distribution of the Ratings histogram. The idea is that dissatisfied viewers and highly satisfied viewers have a disproportionaly higher voting turnout than moderately or lowly satisfied viewers.’

b. Based on your EDA, select an appropriate variable transformation (if any) to apply to each of your three variables. You will fit a model of the type,

$$f(\text{views}) = \beta_0 + \beta_1 g(\text{rate}) + \beta_3 h(\text{length})$$

Where f, g and h are sensible transformations, which might include making *no* transformation.

```
model <- lm(log10(views) ~ average_rating + log10(length), data = d)

stargazer(
   model,
   type = 'text',
   se = list(get_robust_se(model))
   )
```

```
##
## ===============================================
##                         Dependent variable:
##                     ---------------------------
##                            log10(views)
## -----------------------------------------------
## average_rating               0.189***
##                               (0.010)
##
## log10(length)                0.220***
##                               (0.020)
##
## Constant                     1.869***
##                               (0.057)
##
## -----------------------------------------------
## Observations                   9,549
## R2                             0.054
## Adjusted R2                    0.054
## Residual Std. Error      0.844 (df = 9546)
## F Statistic          274.498*** (df = 2; 9546)
## ===============================================
## Note:               *p<0.1; **p<0.05; ***p<0.01
```

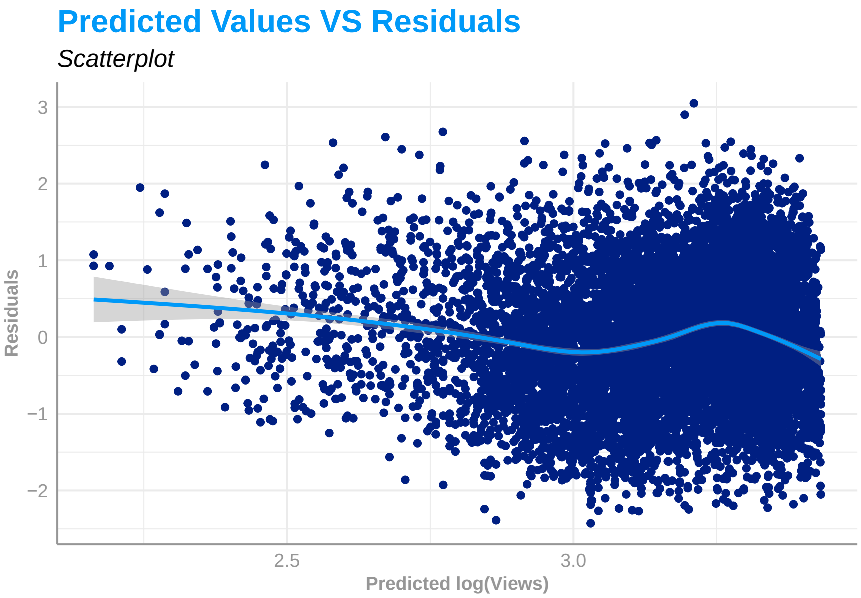
c. Using diagnostic plots, background knowledge, and statistical tests, assess all five assumptions of the CLM. When an assumption is violated, state what response you will take. As part of this process, you should decide what transformation (if any) to apply to each variable. Iterate against your model until your satisfied that at least four of the five assumption have been reasonably addressed.

1. **IID Data:** 'For the CLM to apply, we requrie that our data is generated by an IID sampling process. In this case, a web crawler was used to collect the video data about YouTube videos. The crawler used a directed graph when sampling videos (directional links between videos, where each node represents a video). The initial videos (root 0 in the graph), were preselected by the researchers (from currated pages on the YoueTube platform, such as "Most Viewed", or "Top Rated"). The following videos added to the sample (and to the graph, by the crawler) were selected from the list of videos related to each video already in the sample (adding 20 videos with each step, and going to a depth of 4). While the sample size is large, this method sampling method can not be considered random as we are relying on a currated (non-independent) initial sample, based on some common criteria, then adding further related videos, violiting our requirement for independance in the sample.'

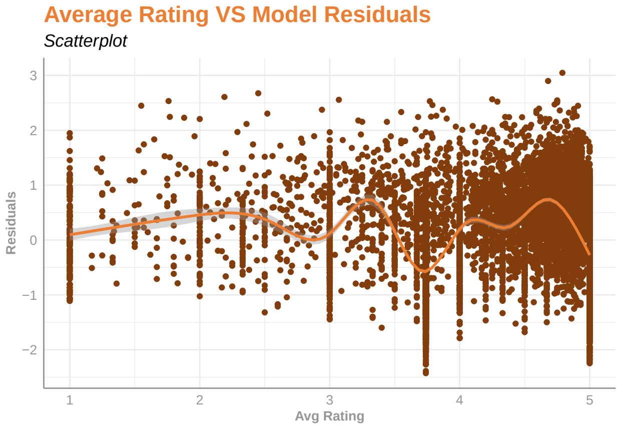
2. **No Perfect Colinearity:** 'We are using two variables, namely average rating and length, which don't seem perfect colinear from their very definition since they don't seem to be different ways to say the same concept, one talks about the perceived quality while the other is the video's duration. When computing their correlation: $\rho = 0.195$, we can notice that indeed they don't seem highly correlated, and therefore neither colinear.'

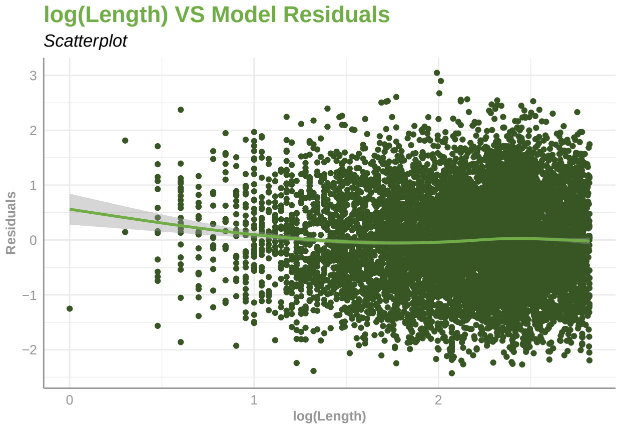
3. **Linear Conditional Expectation:** 'When looking at our scatterplot of Predicted Values and their Residuals, there seems to be a linear conditicional expectation for the residuals. So even when the graph for our first X (Average rating) might not look as linear as desired, we could say that this assumption is fairly met.'

```
d %>% ggplot(aes(x = d$pred, y = d$res)) + geom_point(color = "#001F82") +
  stat_smooth(color = "#0099F8") +
  labs(title = "Predicted Values VS Residuals",
       subtitle = "Scatterplot",
       x = "Predicted log(Views)",
       y = "Residuals") +
  theme_minimal() +
    theme(
      plot.title = element_text(color = "#0099F8",
                                size = 17,
                                face = "bold"),
      plot.subtitle = element_text(size = 13, face = "italic"),
      axis.title = element_text(color = "#969696",
                                size = 10,
                                face = "bold"),
      axis.text = element_text(color = "#969696", size = 10),
      axis.line = element_line(color = "#969696")
    )
```



```
d %>% ggplot(aes(x = x_1, y = d$res)) + geom_point(color = "#833C0C") +
  stat_smooth(color = "#ED7D31") +
  labs(title = "Average Rating VS Model Residuals",
       subtitle = "Scatterplot",
       x = "Avg Rating",
       y = "Residuals") +
  theme_minimal() +
    theme(
      plot.title = element_text(color = "#ED7D31",
                                size = 17,
                                face = "bold"),
      plot.subtitle = element_text(size = 13, face = "italic"),
      axis.title = element_text(color = "#969696",
                                size = 10,
                                face = "bold"),
      axis.text = element_text(color = "#969696", size = 10),
      axis.line = element_line(color = "#969696")
    )

d %>% ggplot(aes(x = x_2, y = d$res)) + geom_point(color = "#375623") +
    stat_smooth(color = "#70AD47") +
  labs(title = "log(Length) VS Model Residuals",
       subtitle = "Scatterplot",
       x = "log(Length)",
       y = "Residuals") +
  theme_minimal() +
    theme(
      plot.title = element_text(color = "#70AD47",
                                size = 17,
                                face = "bold"),
      plot.subtitle = element_text(size = 13, face = "italic"),
      axis.title = element_text(color = "#969696",
                                size = 10,
                                face = "bold"),
      axis.text = element_text(color = "#969696", size = 10),
      axis.line = element_line(color = "#969696")
    )
```





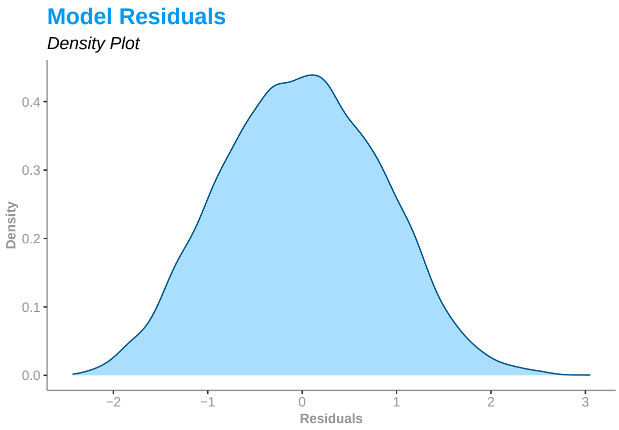
4. **Homoskedastic Errors:** 'When assessing homoskedastic errors, we are looking for evidence that heteroskedacicity exists among the errors. Since the p-value after performing the Breusch-Pagan test is < 2.2e-16 which is smaller than our alpha of 0.05, we reject the null hypothesis that there is no evidence of heteroskedasticity. The "Predicted Values VS Residuals" plot from the previous point also shows that our model suffers from heteroskedastic error because residuals seem further away from zero as the predicted value gets bigger. The assumption of homoskedastic error is not met.'

```
bptest(model)
```

```
##
##  studentized Breusch-Pagan test
##
## data:  model
## BP = 4.3481, df = 2, p-value = 0.1137
```

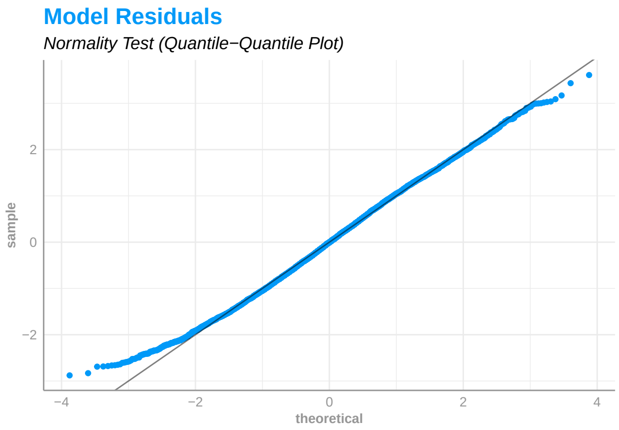
5. **Normally Distributed Errors:** ‘When looking at the distribution of our residuals we find indeed a bell-shaped density, which is confirmed when comparing it with a normal distribution using a Q-Q Plot. This assumption is satisfied in this case’

```
d %>% ggplot(aes(res)) +
    labs(
      title = "Model Residuals",
      subtitle = "Density Plot",
      x = "Residuals",
      y = "Density"
    ) +
    theme_classic() +
    theme(
      plot.title = element_text(color = "#0099F8",
                                size = 17,
                                face = "bold"),
      plot.subtitle = element_text(size = 13, face = "italic"),
      axis.title = element_text(color = "#969696",
                                size = 10,
                                face = "bold"),
      axis.text = element_text(color = "#969696", size = 10),
      axis.line = element_line(color = "#969696")
    ) + geom_density(color = "#00558A", fill = "#71C9FF", alpha = 0.6)

d %>% ggplot(aes(sample = scale(res))) +
  geom_qq(color = "#0099F8") +
  geom_abline(alpha=0.5) +
  labs(title = "Model Residuals",
       subtitle = "Normality Test (Quantile-Quantile Plot)") +
  theme_minimal() +
    theme(
      plot.title = element_text(color = "#0099F8",
                                size = 17,
                                face = "bold"),
      plot.subtitle = element_text(size = 13, face = "italic"),
      axis.title = element_text(color = "#969696",
                                size = 10,
                                face = "bold"),
      axis.text = element_text(color = "#969696", size = 10),
      axis.line = element_line(color = "#969696")
    )
```



Model Residuals

Density Plot



Model Residuals

Normality Test (Quantile−Quantile Plot)