# CDS 101 – Final Project Report

JAD - John D Villanueva, Akshaya Pabbisetty, David Brown

December 10, 2025

## Contents

**1. Problem Definition** **2**

**2. Data Acquisition & Description** **2**

**3. Data Cleaning & Preprocessing** **2**

**4. Exploratory Data Analysis (EDA)** **3**

**5. Visualization Quality and Storytelling** **6**

**6. Modeling Approach** **8**

**7. Model Implementation & Evaluation** **8**

**8. Conclusions & Recommendations** **11**

**9. Code Quality & Reproducibility** **11**

Set up. . . . . 11
Cleaning data, only getting the needed values . . . . . 12
Clarity and cut vs prices point plots . . . . . 13
Carat vs price heat plot . . . . . 13
Linear regression model for carat . . . . . 14
Linear regression model for cut . . . . . 15
Linear regression model for clarity . . . . . 15
Graphing Carat vs price in relation to its Model . . . . . 16
Graphing cut vs price in relation to its Model . . . . . 17
Graphing clarity vs price in relation to its Model . . . . . 18
Residual histogram plots of all 3 models. . . . . 19
Creating the models and numerical squared residuals. . . . . 22
Dataframing the combined residuals . . . . . 23
Residual histogram plots of the models. . . . . 23

# 1. Problem Definition

We are going to be investigating a variety of factors that may affect the price of a diamond. Diamonds are highly sought-after gems, used in all sorts of precious jewelry. With a dataset containing information on many aspects of diamonds, we will determine the particular effects of the clarity, weight in carats, and the cut quality of any diamond upon its price. We believe that the clarity and carat of a diamond will be strong predictors of pricing while the cut quality will be a weaker predictor.

# 2. Data Acquisition & Description

The dataset is, quite simply, called "diamonds". It was obtained from the website Kaggle at the link https://www.kaggle.com/datasets/ayeshaseherr/diamonds.

There are ten variables creating each column; three are categorical, these being the cut quality, color grade, and clarity rating. The remaining seven are carat, depth, table, x, y, z, and price.

The variables x, y, and z are the length, width, and depth in millimeters, carats are a measure of diamond weight, table is the top width percentage of each diamond, and price is standardly measured in USD.

There are just under 54000 entries/rows in the dataset, making it very large overall.

```
library(readr)
diamond_df <- read_csv("data/diamonds.csv")
```

```
## Rows: 53940 Columns: 10
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr (3): cut, color, clarity
## dbl (7): carat, depth, table, x, y, z, price
##
## 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.
```

# 3. Data Cleaning & Preprocessing

```
library(tidyverse)
```

```
## -- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 --
## v dplyr     1.1.4     v purrr     1.1.0
## v forcats   1.0.0     v stringr   1.5.1
## v ggplot2   4.0.0     v tibble    3.3.0
## v lubridate 1.9.4     v tidyr     1.3.1
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
## i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become errors
```

```r
diamond_1 <- diamonds %>%
  select(carat, cut, clarity, price) %>%
  mutate(
    cut_n = as.numeric(factor(
      cut,
      levels = c("Fair", "Good", "Very Good", "Premium", "Ideal")
    )
  ),
    clarity_n = as.numeric(factor(
      clarity,
      levels = c("I1", "SI2", "SI1", "VS2", "VS1", "VVS2", "VVS1", "IF")
    ))
  )
```

Cut and Clarity do need to be changed into numeric values with as.numeric(factor()), and I have used levels = c() to make sure that each category is properly organized in ascending quality.

## 4. Exploratory Data Analysis (EDA)

```r
library(broom)
library(tidyverse)
library(modelr)
Diamond_model_a <- lm(price ~ carat, data = diamond_1)
Diamond_model_a %>%
  glance()
```

```
## # A tibble: 1 x 12
##   r.squared adj.r.squared sigma statistic p.value    df   logLik     AIC     BIC
##       <dbl>         <dbl> <dbl>     <dbl>   <dbl> <dbl>    <dbl>   <dbl>   <dbl>
## 1     0.849         0.849 1549.   304051.       0     1 -472730. 945467. 945493.
## # i 3 more variables: deviance <dbl>, df.residual <int>, nobs <int>
```

```r
Diamond_model_u <- lm(price ~ cut_n, data = diamond_1)
Diamond_model_u %>%
  glance()
```

```
## # A tibble: 1 x 12
##   r.squared adj.r.squared sigma statistic  p.value    df   logLik     AIC    BIC
##       <dbl>         <dbl> <dbl>     <dbl>    <dbl> <dbl>    <dbl>   <dbl>  <dbl>
## 1   0.00286       0.00284 3984.      155. 1.76e-35     1 -523698.  1.05e6 1.05e6
## # i 3 more variables: deviance <dbl>, df.residual <int>, nobs <int>
```

```r
Diamond_model_y <- lm(price ~ clarity_n, data = diamond_1)
Diamond_model_y %>%
  glance()
```

```
## # A tibble: 1 x 12
##   r.squared adj.r.squared sigma statistic   p.value    df   logLik    AIC    BIC
```

```
##       <dbl>         <dbl> <dbl>     <dbl>     <dbl> <dbl>    <dbl>  <dbl>  <dbl>
## 1    0.0216        0.0215 3946.     1188. 1.59e-257     1 -523188. 1.05e6 1.05e6
## # i 3 more variables: deviance <dbl>, df.residual <int>, nobs <int>
```

```
diamond_ma_df <- diamond_1 %>%
  add_predictions(Diamond_model_a) %>%
  add_residuals(Diamond_model_a)
diamond_mu_df <- diamond_1 %>%
  add_predictions(Diamond_model_u) %>%
  add_residuals(Diamond_model_u)
diamond_my_df <- diamond_1 %>%
  add_predictions(Diamond_model_y) %>%
  add_residuals(Diamond_model_y)
```

When glancing at each model, the squared residual for the Carat Model is much closer to one in comparison to Cut's model and Clarity's model. As well as the creation of the dataframes of said models.

```
diamond_2 <- diamond_1 %>%
  mutate(
    Weight_com_Clar = clarity_n / carat,
    Weight_com_Cut = cut_n / carat
  )
```

```
Diamond_model_au <- lm(price ~ Weight_com_Cut, data = diamond_2)
```

```
Diamond_model_au %>%
  tidy()
```

```
## # A tibble: 2 x 5
##   term           estimate std.error statistic p.value
##   <chr>             <dbl>     <dbl>     <dbl>   <dbl>
## 1 (Intercept)       7819.     23.9       327.       0
## 2 Weight_com_Cut    -557.      2.86     -195.       0
```

```
Diamond_model_au %>%
  glance()
```

```
## # A tibble: 1 x 12
##   r.squared adj.r.squared sigma statistic p.value    df   logLik      AIC    BIC
##       <dbl>         <dbl> <dbl>     <dbl>   <dbl> <dbl>    <dbl>    <dbl>  <dbl>
## 1     0.413         0.413 3058.    37873.       0     1 -509430. 1018866. 1.02e6
## # i 3 more variables: deviance <dbl>, df.residual <int>, nobs <int>
```

```
Diamond_model_ay <- lm(price ~ Weight_com_Clar, data = diamond_2)
```

```
Diamond_model_ay %>%
  tidy()
```

```
## # A tibble: 2 x 5
##   term            estimate std.error statistic p.value
##   <chr>              <dbl>     <dbl>     <dbl>   <dbl>
## 1 (Intercept)        6640.     22.6       294.       0
## 2 Weight_com_Clar    -351.      2.27     -155.       0
```

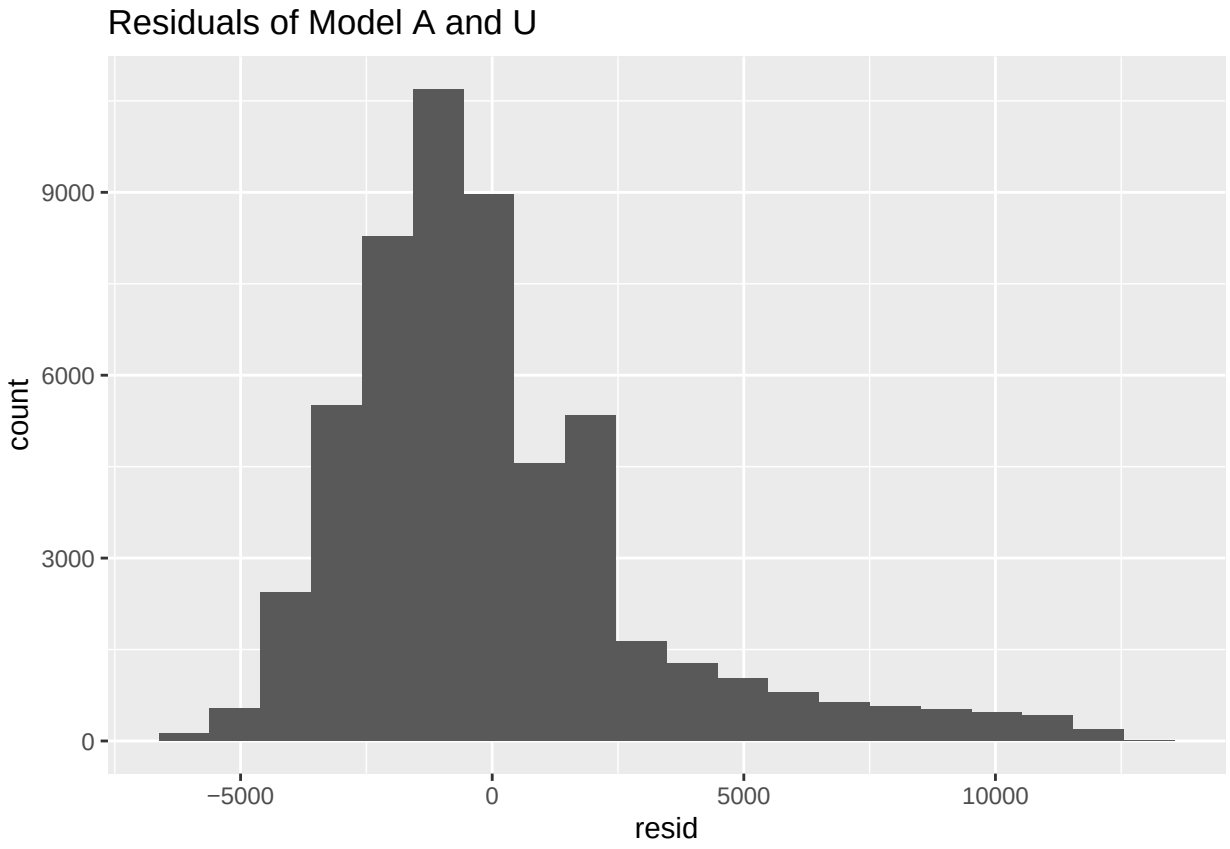
```
Diamond_model_ay %>%
  glance()
```

```
## # A tibble: 1 x 12
##   r.squared adj.r.squared sigma statistic p.value    df   logLik      AIC    BIC
##       <dbl>         <dbl> <dbl>     <dbl>   <dbl> <dbl>    <dbl>    <dbl>  <dbl>
## 1     0.307         0.307 3321.    23900.       0     1 -513883. 1027772. 1.03e6
## # i 3 more variables: deviance <dbl>, df.residual <int>, nobs <int>
```

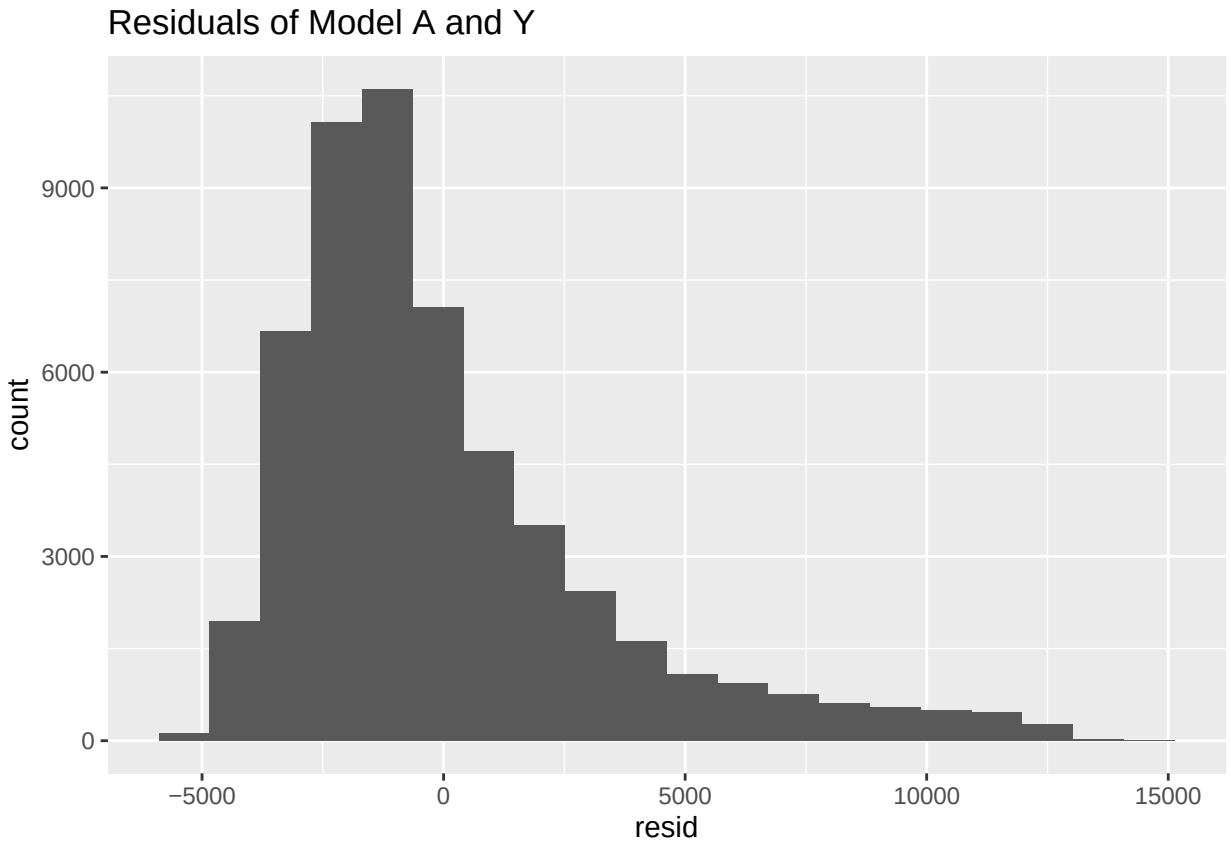
```
diamond_may_df <- diamond_2 %>%
  add_predictions(Diamond_model_ay) %>%
  add_residuals(Diamond_model_ay)

diamond_mau_df <- diamond_2 %>%
  add_predictions(Diamond_model_au) %>%
  add_residuals(Diamond_model_au)
```

```
diamond_mau_df %>%
  ggplot()+
  geom_histogram(mapping = aes(x = resid), bins = 20) +
  labs(title = "Residuals of Model A and U")
```



```r
diamond_may_df %>%
  ggplot()+
  geom_histogram(mapping = aes(x = resid), bins = 20) +
  labs(title = "Residuals of Model A and Y")
```




To further conclude a hierarchy of influence, we did see how Carat and Clarity compares to Weight and Cut as their own variables. The results do show the a Cut of a diamond has a greater influence over Clarity, with Weight_com_Cut's squared residuals being closer to 1.

## 5. Visualization Quality and Storytelling

- Explain why your plot types are appropriate.

We used a heat map to visualize the relationship and the density of weight in Carats in comparison to the price of a diamond. Due to the quantity of entries, there is a high density that would not be able to be explored efficiently with a point plot. The heat map shows the points with a higher density of diamonds and lesser density of diamonds without getting muddled.

- Comment on labels, legends, colors, and overall readability.

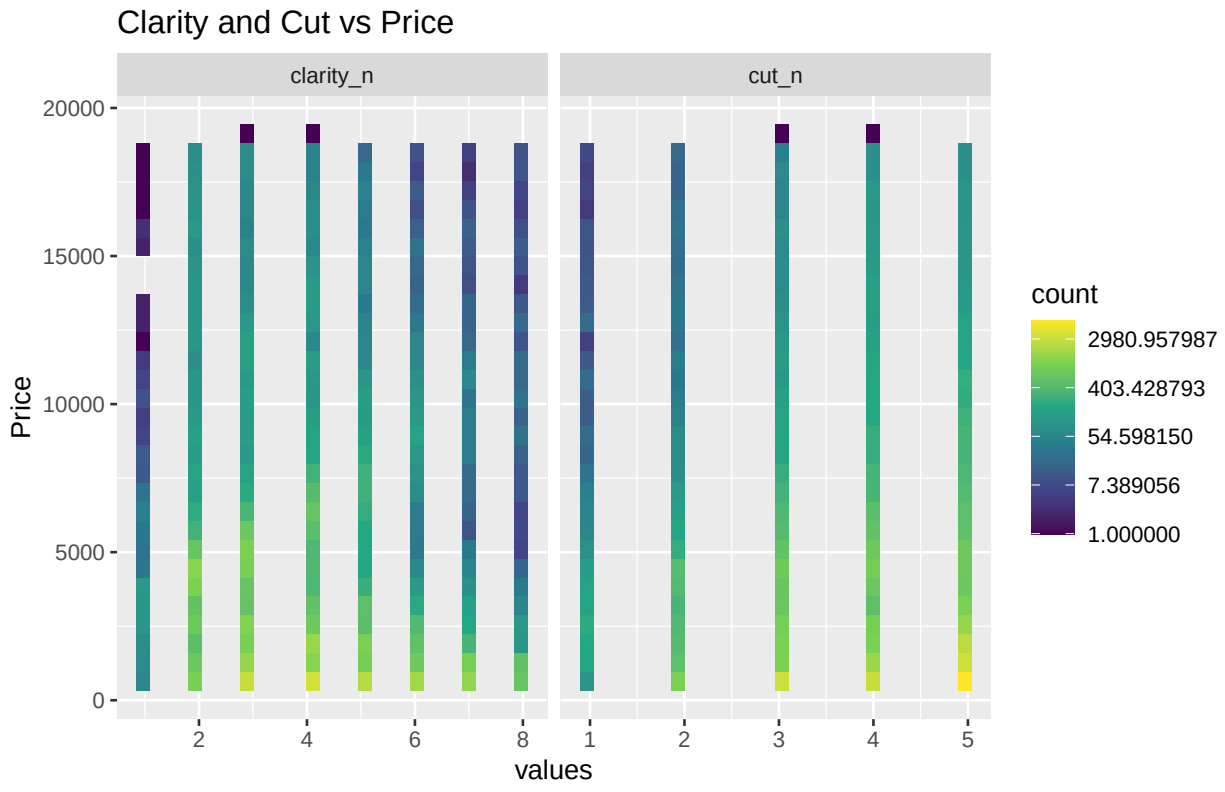
The heatmaps show the density in different colors, making it more readable and understandable of the relationship. All graphs are labeled with accurate titles and labels.

- Mention any steps you took to make plots accessible and interpretable.

We chose to utilize a heat map instead of a normal point plot to make it more interpretable and accessible due to the sheer amount of entries.

```
diamond_1 %>%
  pivot_longer(cols = cut_n|clarity_n,
               names_to = "measurement",
               values_to = "value") %>%
  ggplot() +
  geom_bin2d(aes(x = value, y = price)) +
  labs(
    title = "Clarity and Cut vs. Price",
    x = "values",
    y = "Price"
  ) +
scale_fill_viridis_c(trans = "log") +
  facet_wrap(~ measurement, scales = "free_x") +
  labs(title = "Clarity and Cut vs Price")
```

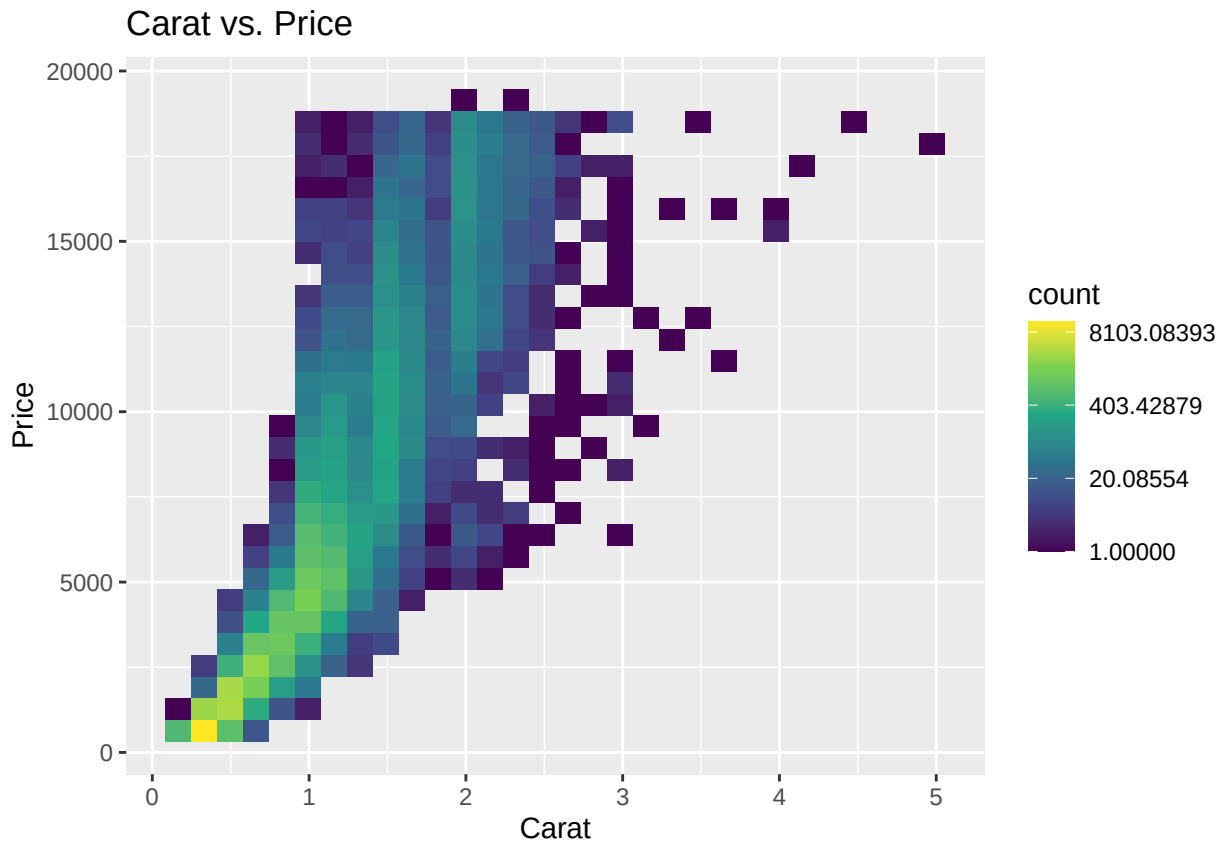
```
## `stat_bin2d()` using `bins = 30`. Pick better value `binwidth`.
```




```
diamond_1 %>%
  ggplot() +
  geom_bin2d(aes(x = carat, y = price)) +
  labs(
    title = "Carat vs. Price",
    x = "Carat",
    y = "Price"
```

```
  ) +
scale_fill_viridis_c(trans = "log")
```

```
## `stat_bin2d()` using `bins = 30`. Pick better value `binwidth`.
```



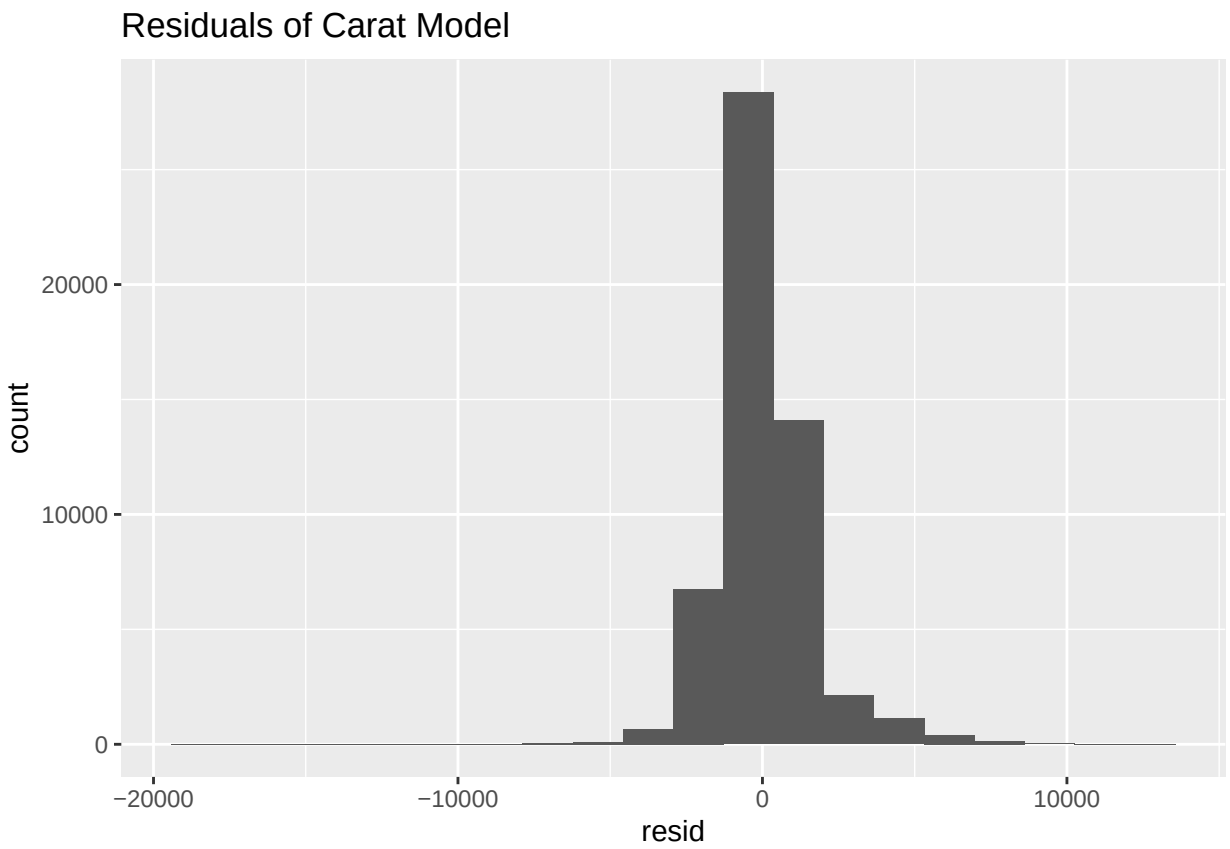

## 6. Modeling Approach

Explain how you framed the problem and which models you chose:

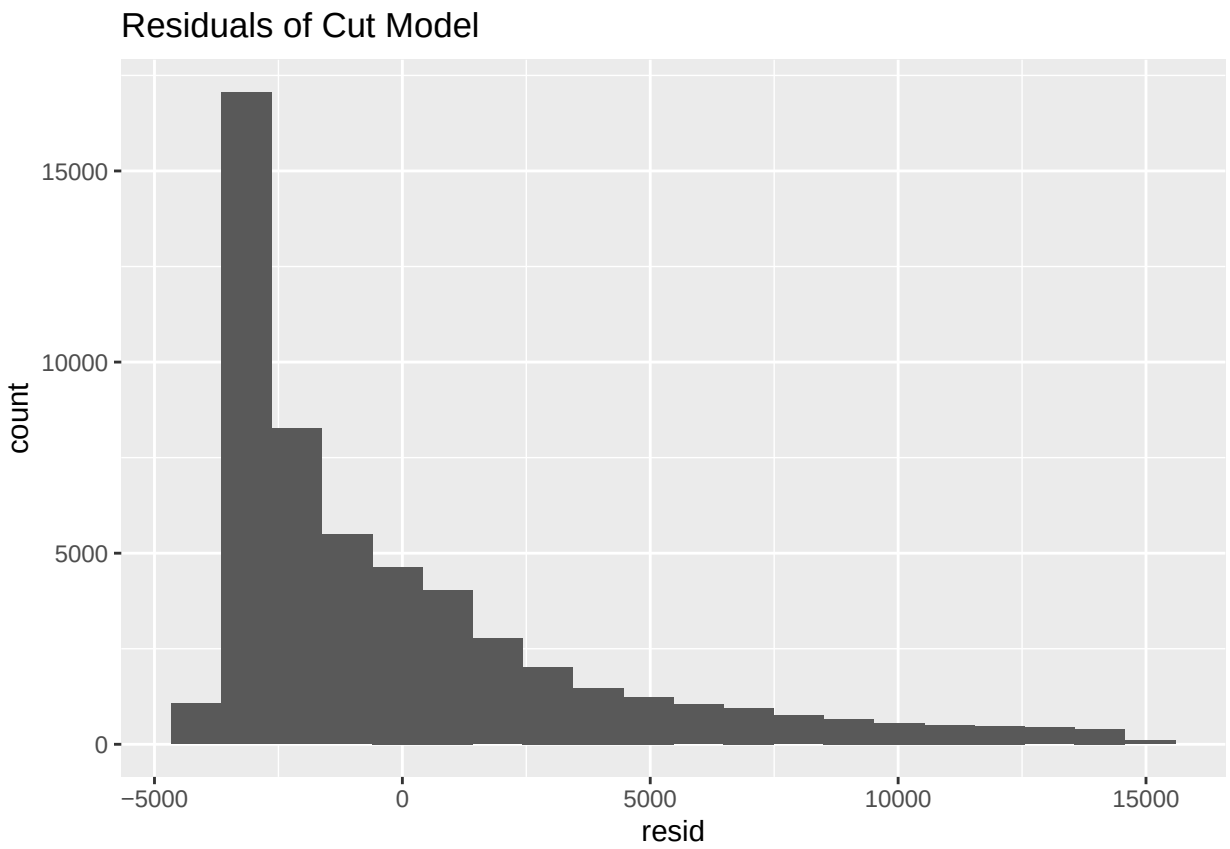
A basic linear regression model was used as it was the most appropriate model to use as our research question was looking at the correlation between three variables and which one affected the pricing of diamonds the most among all entries, and a linear regression model was the most convenient method to do so.

## 7. Model Implementation & Evaluation

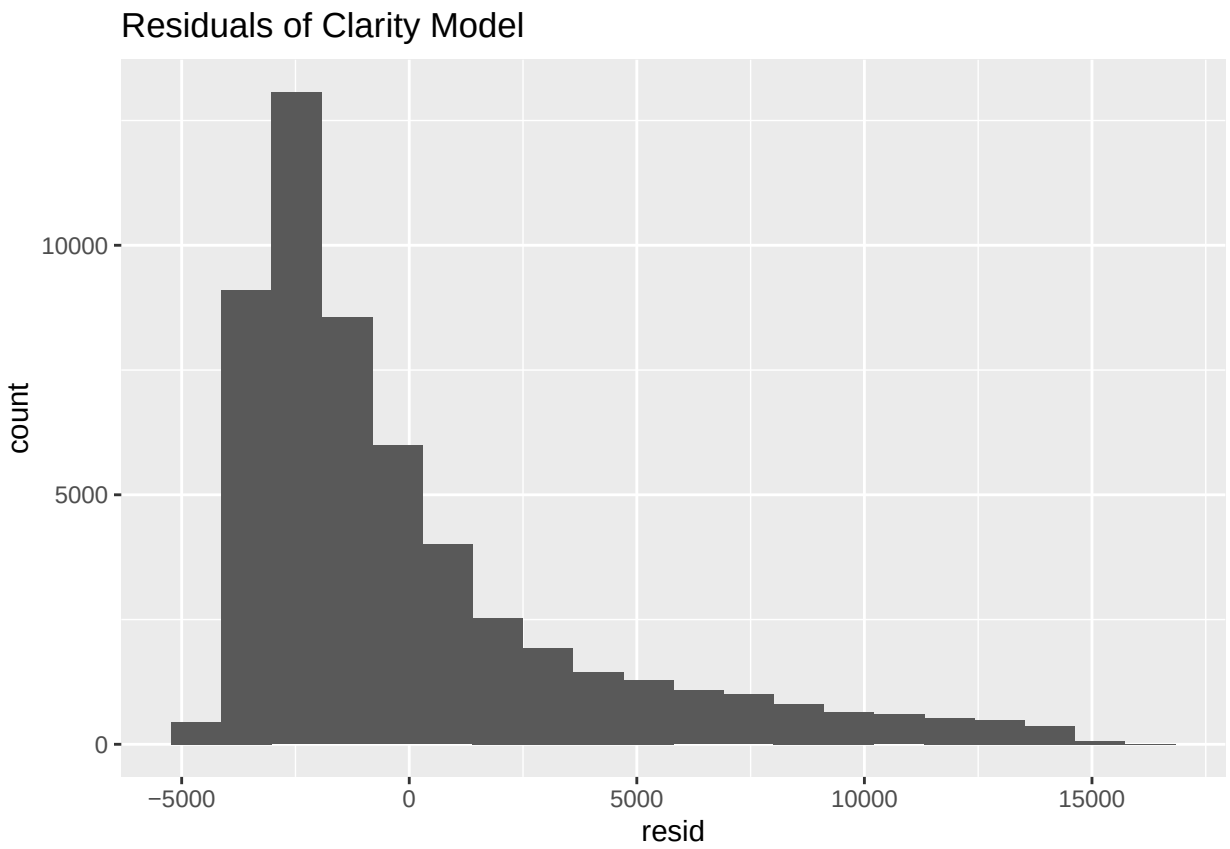
```
diamond_ma_df %>%
  ggplot()+
  geom_histogram(mapping = aes(x = resid), bins = 20) +
  labs(title = "Residuals of Carat Model")
```




```
diamond_mu_df %>%
  ggplot()+
  geom_histogram(mapping = aes(x = resid), bins = 20) +
  labs(title = "Residuals of Cut Model")
```




```
diamond_my_df %>%
  ggplot()+
  geom_histogram(mapping = aes(x = resid), bins = 20) +
  labs(title = "Residuals of Clarity Model")
```




Model for Carat is Skewed left in total, but there is a higher spike at 0, where as Cut and clarity are skewed right.

## 8. Conclusions & Recommendations

Carat seems to have a higher correlation with the Price of a diamond in comparison to Clarity and Cut because of the feature's squared residuals being very close to 1. Since this dataset has nearly 54 thousand entries with the price range approximately 18497 dollars, the sheer amount of data does hinder some comprehension. So taking some smaller sample of the data would help with comprehension. To quote an article by Alicia Briggs from Vrai, in relation to the reasoning behind carat's influence, "Carat weight is a significant factor in determining the total price of a diamond, but it's not the only one. The price of a diamond exponentially rises after 1 carat, with a 2 carat diamond often more than double the cost. Why? Because the higher the carat weight, the bigger the loose diamond it was cut from. This means the diamond is rarer and therefore more valuable."

## 9. Code Quality & Reproducibility

Briefly document how someone else can reproduce your results:

### Set up.

```
## R version 4.5.1 (2025-06-13)
## Platform: aarch64-apple-darwin20
```

```
## Running under: macOS Sequoia 15.5
##
## Matrix products: default
## BLAS:   /Library/Frameworks/R.framework/Versions/4.5-arm64/Resources/lib/libRblas.0.dylib
## LAPACK: /Library/Frameworks/R.framework/Versions/4.5-arm64/Resources/lib/libRlapack.dylib;  LAPACK v
##
## locale:
## [1] en_US.UTF-8/en_US.UTF-8/en_US.UTF-8/C/en_US.UTF-8/en_US.UTF-8
##
## time zone: America/New_York
## tzcode source: internal
##
## attached base packages:
## [1] stats     graphics  grDevices utils     datasets  methods   base
##
## other attached packages:
##  [1] modelr_0.1.11   broom_1.0.9     lubridate_1.9.4 forcats_1.0.0
##  [5] stringr_1.5.1   dplyr_1.1.4     purrr_1.1.0     tidyr_1.3.1
##  [9] tibble_3.3.0    ggplot2_4.0.0   tidyverse_2.0.0 readr_2.1.5
##
## loaded via a namespace (and not attached):
##  [1] bit_4.6.0          gtable_0.3.6       compiler_4.5.1     crayon_1.5.3
##  [5] tidyselect_1.2.1   parallel_4.5.1     scales_1.4.0       yaml_2.3.10
##  [9] fastmap_1.2.0      R6_2.6.1           labeling_0.4.3     generics_0.1.4
## [13] knitr_1.50         backports_1.5.0    pillar_1.11.0      RColorBrewer_1.1-3
## [17] tzdb_0.5.0         rlang_1.1.6        utf8_1.2.6         stringi_1.8.7
## [21] xfun_0.53          S7_0.2.0           bit64_4.6.0-1      viridisLite_0.4.2
## [25] timechange_0.3.0   cli_3.6.5          withr_3.0.2        magrittr_2.0.3
## [29] digest_0.6.37      grid_4.5.1         vroom_1.6.5        rstudioapi_0.17.1
## [33] hms_1.1.3          lifecycle_1.0.4    vctrs_0.6.5        evaluate_1.0.4
## [37] glue_1.8.0         farver_2.1.2       rmarkdown_2.29     tools_4.5.1
## [41] pkgconfig_2.0.3    htmltools_0.5.8.1

## Rows: 53940 Columns: 10
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr (3): cut, color, clarity
## dbl (7): carat, depth, table, x, y, z, price
##
## 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.
```

### Cleaning data, only getting the needed values

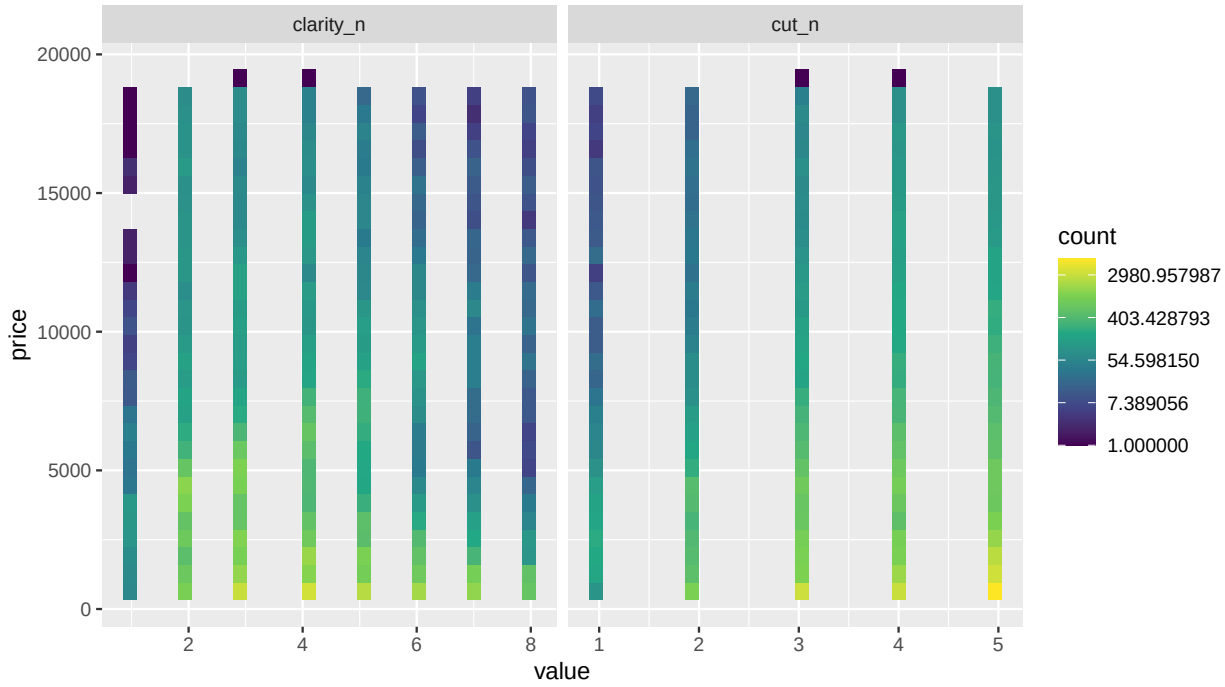
```
diamond_1 <- diamonds %>%
  select(carat, cut, clarity, price) %>%
  mutate(
    cut_n = as.numeric(factor(
      cut,
      levels = c("Fair", "Good", "Very Good", "Premium", "Ideal")
    )
  ),
```

```
    clarity_n = as.numeric(factor(
      clarity,
      levels = c("I1", "SI2", "SI1", "VS2", "VS1", "VVS2", "VVS1", "IF")
    ))
  )
```

## Clarity and cut vs prices point plots

```
diamond_1 %>%
  pivot_longer(cols = cut_n:clarity_n,
               names_to = "measurement",
               values_to = "value") %>%
  ggplot() +
    geom_bin2d(aes(x = value, y = price)) +
    scale_fill_viridis_c(trans = "log") +
    facet_wrap(~ measurement, scales = "free_x")
```

```
## `stat_bin2d()` using `bins = 30`. Pick better value `binwidth`.
```

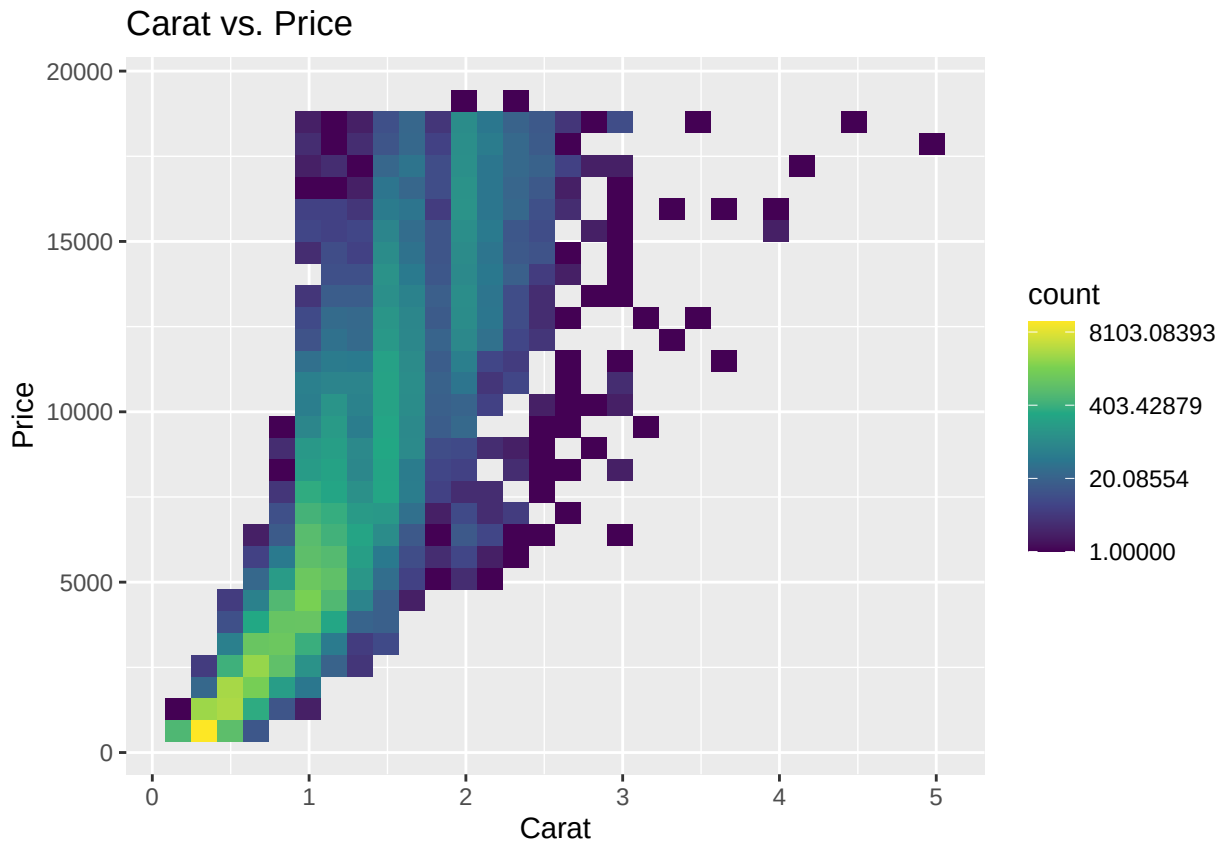



## Carat vs price heat plot

```
diamond_1 %>%
  ggplot() +
  geom_bin2d(aes(x = carat, y = price)) +
  labs(
```

```
    title = "Carat vs. Price",
    x = "Carat",
    y = "Price"
  ) +
scale_fill_viridis_c(trans = "log")
```

```
## `stat_bin2d()` using `bins = 30`. Pick better value `binwidth`.
```




### Linear regression model for carat

```
Diamond_model_a <- lm(price ~ carat, data = diamond_1)
```

```
Diamond_model_a %>%
  tidy()
```

```
## # A tibble: 2 x 5
##   term        estimate std.error statistic p.value
##   <chr>          <dbl>     <dbl>     <dbl>   <dbl>
## 1 (Intercept)   -2256.      13.1     -173.       0
## 2 carat          7756.      14.1      551.       0
```

```
Diamond_model_a %>%
  glance()
```

```
## # A tibble: 1 x 12
##   r.squared adj.r.squared sigma statistic p.value    df   logLik     AIC     BIC
##       <dbl>         <dbl> <dbl>     <dbl>   <dbl> <dbl>    <dbl>   <dbl>   <dbl>
## 1     0.849         0.849 1549.   304051.       0     1 -472730. 945467. 945493.
## # i 3 more variables: deviance <dbl>, df.residual <int>, nobs <int>
```

### Linear regression model for cut

```
Diamond_model_u <- lm(price ~ cut_n, data = diamond_1)
```

```
Diamond_model_u %>%
  tidy()
```

```
## # A tibble: 2 x 5
##   term        estimate std.error statistic  p.value
##   <chr>          <dbl>     <dbl>     <dbl>    <dbl>
## 1 (Intercept)    4679.      62.4      75.0 0
## 2 cut_n          -191.      15.4     -12.4 1.76e-35
```

```
Diamond_model_u %>%
  glance()
```

```
## # A tibble: 1 x 12
##   r.squared adj.r.squared sigma statistic  p.value    df   logLik     AIC    BIC
##       <dbl>         <dbl> <dbl>     <dbl>    <dbl> <dbl>    <dbl>   <dbl>  <dbl>
## 1   0.00286       0.00284 3984.      155. 1.76e-35     1 -523698.  1.05e6 1.05e6
## # i 3 more variables: deviance <dbl>, df.residual <int>, nobs <int>
```

### Linear regression model for clarity

```
Diamond_model_y <- lm(price ~ clarity_n, data = diamond_1)
```

```
Diamond_model_y %>%
  tidy()
```

```
## # A tibble: 2 x 5
##   term        estimate std.error statistic   p.value
##   <chr>          <dbl>     <dbl>     <dbl>     <dbl>
## 1 (Intercept)    5373.      45.1     119.  0
## 2 clarity_n      -356.      10.3     -34.5 1.59e-257
```

```
Diamond_model_y %>%
  glance()
```

```
## # A tibble: 1 x 12
##   r.squared adj.r.squared sigma statistic   p.value    df   logLik    AIC    BIC
##       <dbl>         <dbl> <dbl>     <dbl>     <dbl> <dbl>    <dbl>  <dbl>  <dbl>
## 1    0.0216        0.0215 3946.     1188. 1.59e-257     1 -523188. 1.05e6 1.05e6
## # i 3 more variables: deviance <dbl>, df.residual <int>, nobs <int>
```

### Graphing Carat vs price in relation to its Model

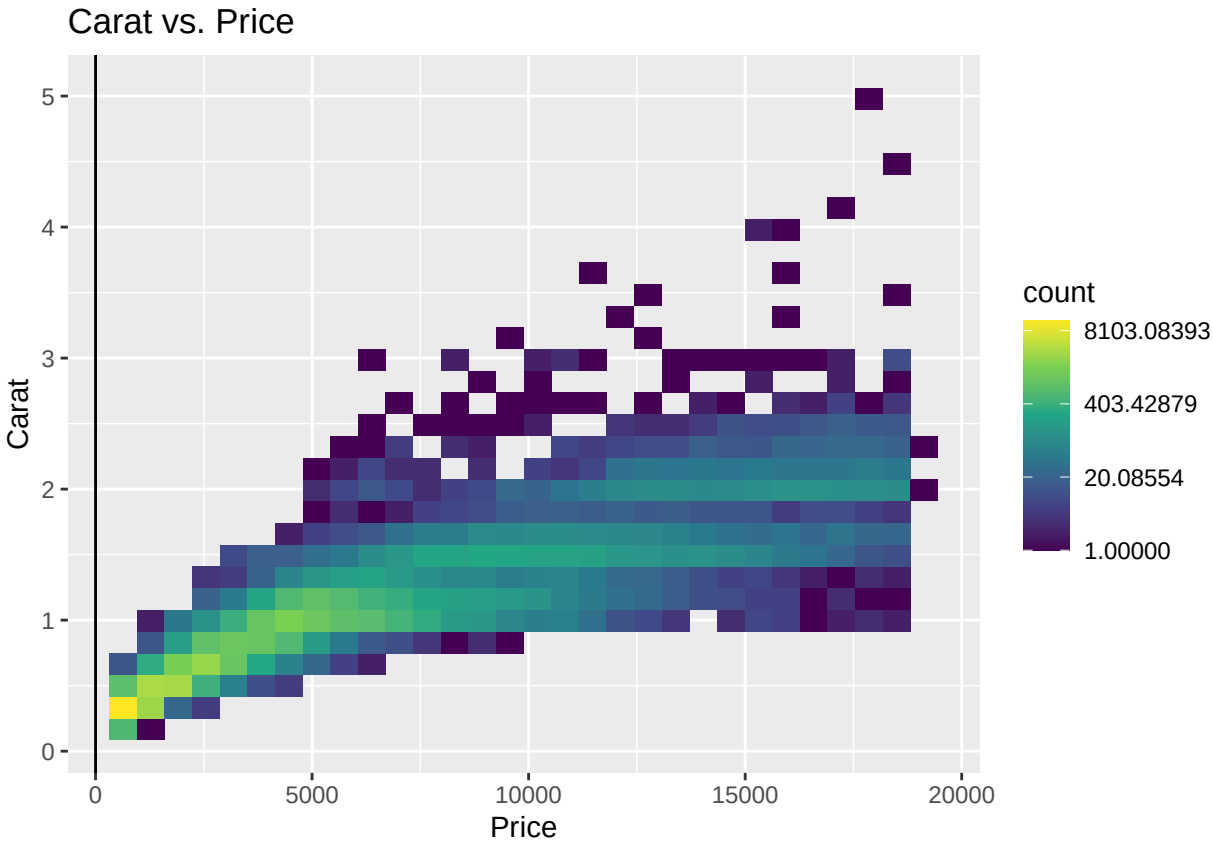
```
diamond_ma_df <- diamond_1 %>%
  add_predictions(Diamond_model_a) %>%
  add_residuals(Diamond_model_a)
```

```
diamond_ma_df
```

```
## # A tibble: 53,940 x 8
##    carat cut       clarity price cut_n clarity_n    pred resid
##    <dbl> <chr>     <chr>   <dbl> <dbl>     <dbl>   <dbl> <dbl>
##  1  0.23 Ideal     SI2       326     5         2 -472.    798.
##  2  0.21 Premium   SI1       326     4         3 -628.    954.
##  3  0.23 Good      VS1       327     2         5 -472.    799.
##  4  0.29 Premium   VS2       334     4         4   -7.00  341.
##  5  0.31 Good      SI2       335     2         2  148.    187.
##  6  0.24 Very Good VVS2      336     3         6 -395.    731.
##  7  0.24 Very Good VVS1      336     3         7 -395.    731.
##  8  0.26 Very Good SI1       337     3         3 -240.    577.
##  9  0.22 Fair      VS2       337     1         4 -550.    887.
## 10  0.23 Very Good VS1       338     3         5 -472.    810.
## # i 53,930 more rows
```

```
ggplot(diamond_ma_df) +
  geom_bin2d(aes(x = price, y = carat)) +
  labs(
    title = "Carat vs. Price",
    x = "Price",
    y = "Carat"
  ) +
scale_fill_viridis_c(trans = "log") +
  geom_abline(slope = Diamond_model_a$coefficients[2], intercept = Diamond_model_a$coefficients[1])
```

```
## `stat_bin2d()` using `bins = 30`. Pick better value `binwidth`.
```



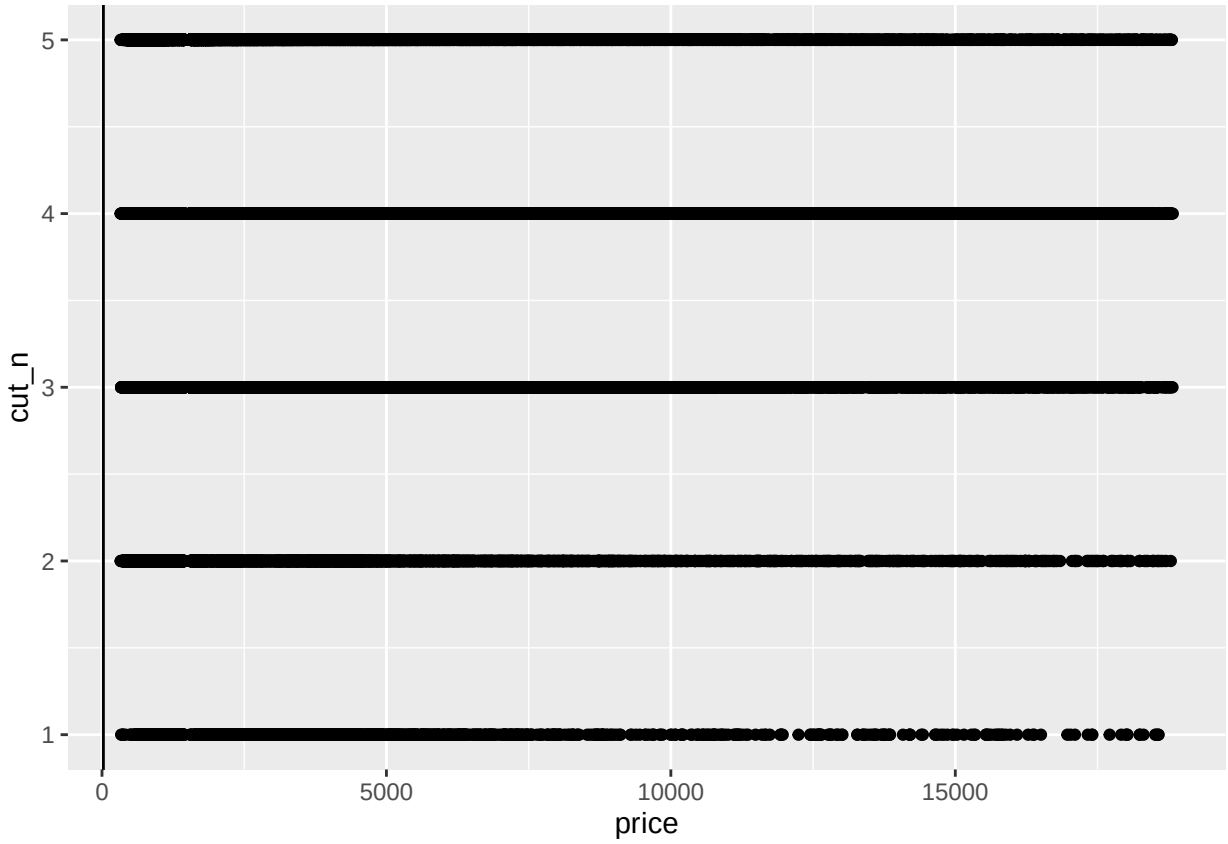

### Graphing cut vs price in relation to its Model

```
diamond_mu_df <- diamond_1 %>%
  add_predictions(Diamond_model_u) %>%
  add_residuals(Diamond_model_u)
```

```
diamond_mu_df
```

```
## # A tibble: 53,940 x 8
##    carat cut       clarity price cut_n clarity_n  pred  resid
##    <dbl> <chr>     <chr>   <dbl> <dbl>     <dbl> <dbl>  <dbl>
##  1  0.23 Ideal     SI2       326     5         2 3723. -3397.
##  2  0.21 Premium   SI1       326     4         3 3914. -3588.
##  3  0.23 Good      VS1       327     2         5 4297. -3970.
##  4  0.29 Premium   VS2       334     4         4 3914. -3580.
##  5  0.31 Good      SI2       335     2         2 4297. -3962.
##  6  0.24 Very Good VVS2      336     3         6 4106. -3770.
##  7  0.24 Very Good VVS1      336     3         7 4106. -3770.
##  8  0.26 Very Good SI1       337     3         3 4106. -3769.
##  9  0.22 Fair      VS2       337     1         4 4488. -4151.
## 10  0.23 Very Good VS1       338     3         5 4106. -3768.
## # i 53,930 more rows
```

```
ggplot(diamond_mu_df) +
  geom_point(mapping = aes(x = price, y = cut_n)) +
  geom_abline(slope = Diamond_model_u$coefficients[2], intercept = Diamond_model_u$coefficients[1])
```




### Graphing clarity vs price in relation to its Model

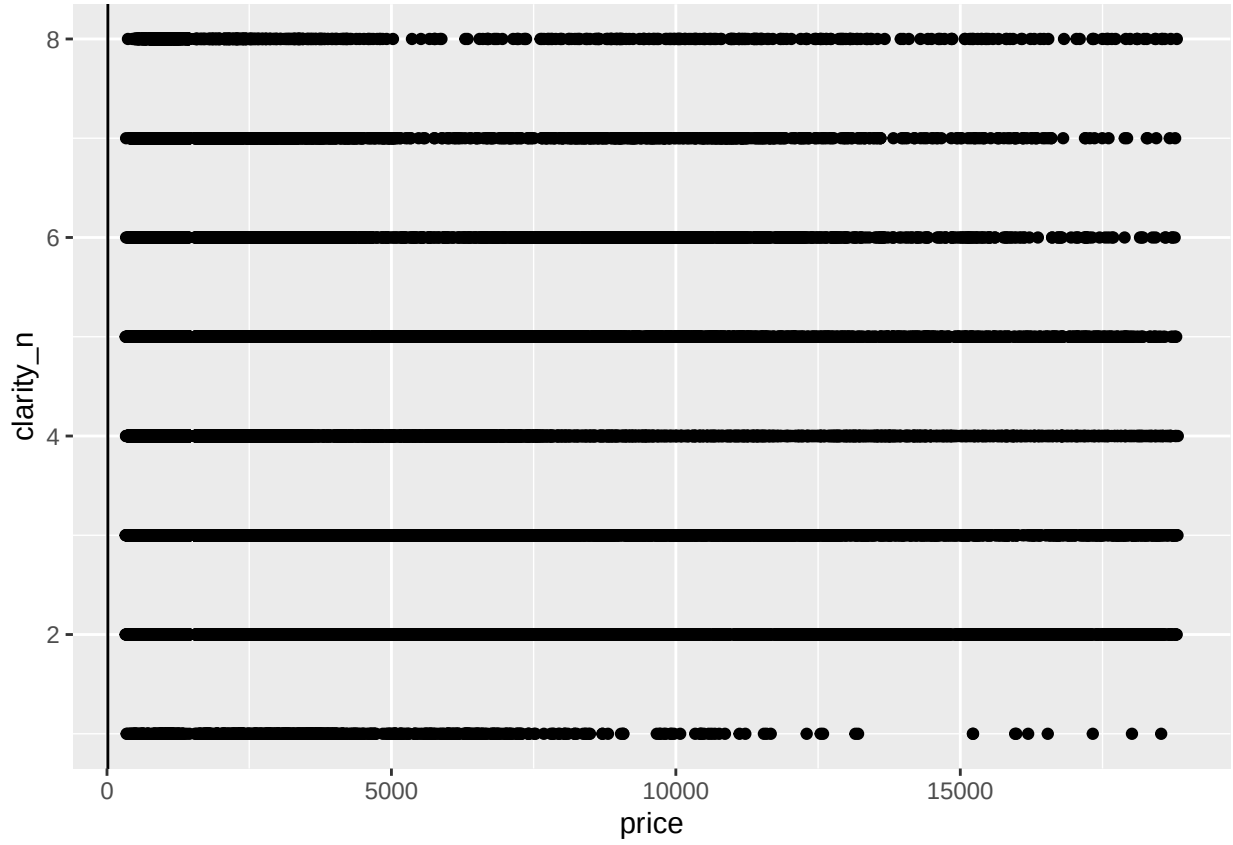
```
diamond_my_df <- diamond_1 %>%
  add_predictions(Diamond_model_y) %>%
  add_residuals(Diamond_model_y)
```

```
diamond_my_df
```

```
## # A tibble: 53,940 x 8
##    carat cut       clarity price cut_n clarity_n  pred  resid
##    <dbl> <chr>     <chr>   <dbl> <dbl>     <dbl> <dbl>  <dbl>
##  1  0.23 Ideal     SI2       326     5         2 4662. -4336.
##  2  0.21 Premium   SI1       326     4         3 4306. -3980.
##  3  0.23 Good      VS1       327     2         5 3595. -3268.
##  4  0.29 Premium   VS2       334     4         4 3951. -3617.
##  5  0.31 Good      SI2       335     2         2 4662. -4327.
##  6  0.24 Very Good VVS2      336     3         6 3240. -2904.
##  7  0.24 Very Good VVS1      336     3         7 2884. -2548.
```

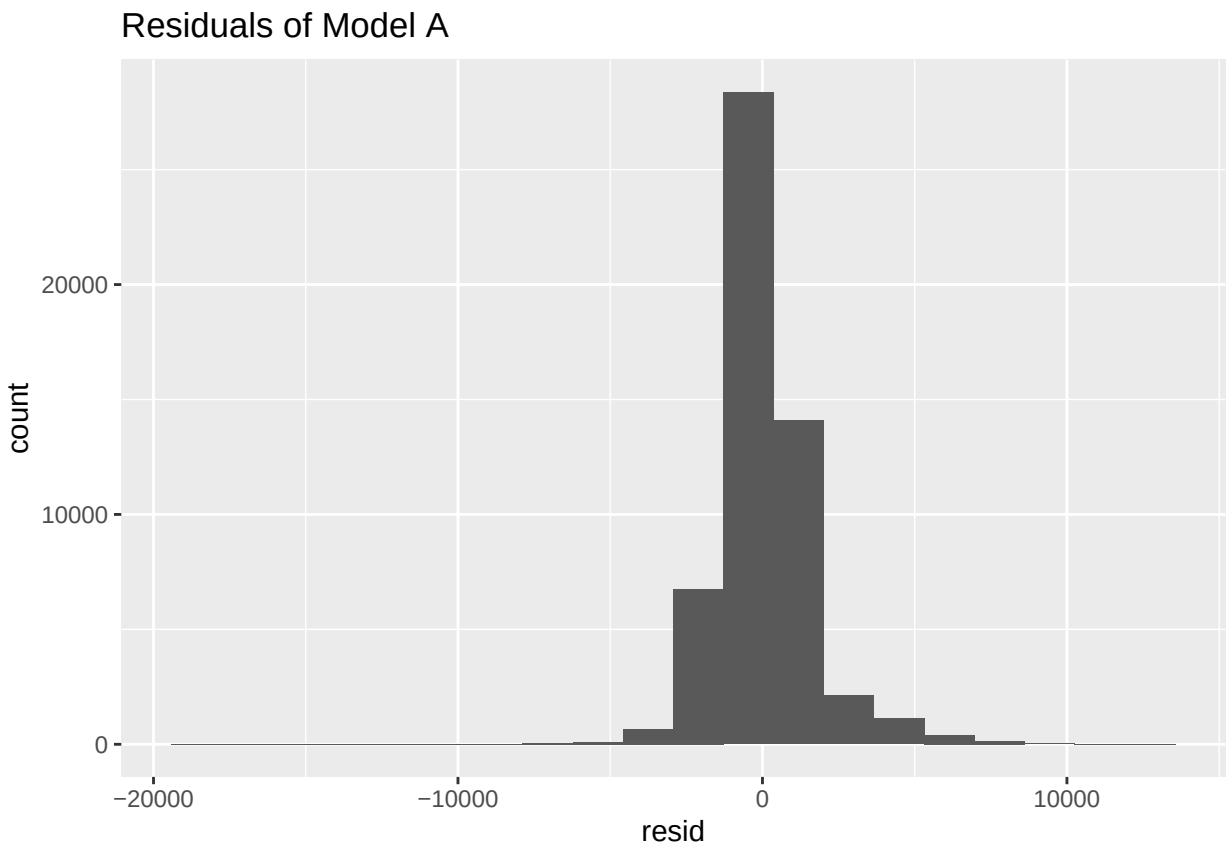
```
##  8  0.26 Very Good SI1       337     3         3 4306. -3969.
##  9  0.22 Fair      VS2       337     1         4 3951. -3614.
## 10  0.23 Very Good VS1       338     3         5 3595. -3257.
## # i 53,930 more rows
```

```
ggplot(diamond_my_df) +
  geom_point(mapping = aes(x = price, y = clarity_n)) +
  geom_abline(slope = Diamond_model_y$coefficients[2], intercept = Diamond_model_y$coefficients[1])
```

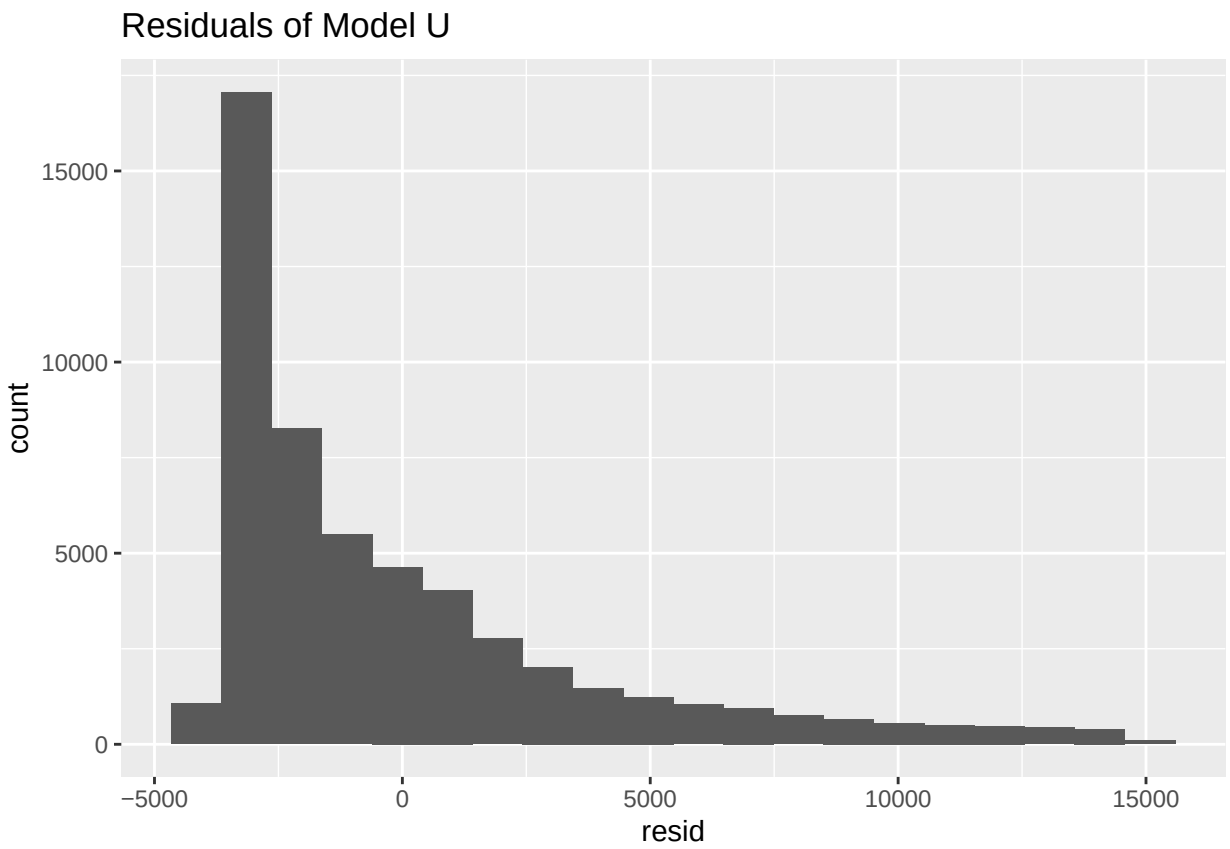



**Residual histogram plots of all 3 models.**

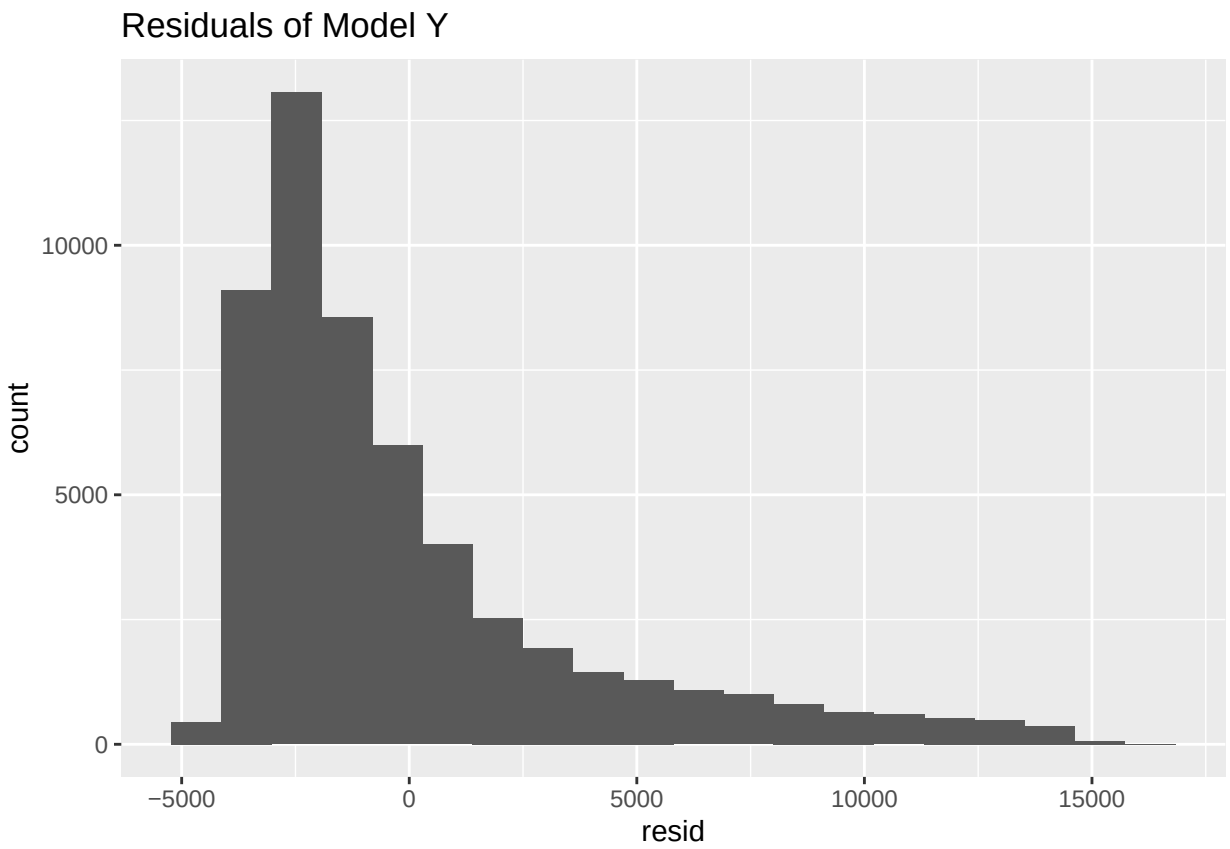
```
diamond_ma_df %>%
  ggplot()+
  geom_histogram(mapping = aes(x = resid), bins = 20) +
  labs(title = "Residuals of Model A")
```




```
diamond_mu_df %>%
  ggplot()+
  geom_histogram(mapping = aes(x = resid), bins = 20) +
  labs(title = "Residuals of Model U")
```




```
diamond_my_df %>%
  ggplot()+
  geom_histogram(mapping = aes(x = resid), bins = 20) +
  labs(title = "Residuals of Model Y")
```




## Mutating for other features over carats.

```
diamond_2 <- diamond_1 %>%
  mutate(
    Weight_com_Clar = clarity_n / carat,
    Weight_com_Cut = cut_n / carat
  )
```

### Creating the models and numerical squared residuals.

```
Diamond_model_au <- lm(price ~ Weight_com_Cut, data = diamond_2)
```

```
Diamond_model_au %>%
  tidy()
```

```
## # A tibble: 2 x 5
##   term           estimate std.error statistic p.value
##   <chr>             <dbl>     <dbl>     <dbl>   <dbl>
## 1 (Intercept)       7819.     23.9       327.       0
## 2 Weight_com_Cut    -557.      2.86     -195.       0
```

```
Diamond_model_au %>%
  glance()
```

```
## # A tibble: 1 x 12
##   r.squared adj.r.squared sigma statistic p.value    df   logLik      AIC    BIC
##       <dbl>         <dbl> <dbl>     <dbl>   <dbl> <dbl>    <dbl>    <dbl>  <dbl>
## 1     0.413         0.413 3058.    37873.       0     1 -509430. 1018866. 1.02e6
## # i 3 more variables: deviance <dbl>, df.residual <int>, nobs <int>
```

```
Diamond_model_ay <- lm(price ~ Weight_com_Clar, data = diamond_2)
```

```
Diamond_model_ay %>%
  tidy()
```

```
## # A tibble: 2 x 5
##   term            estimate std.error statistic p.value
##   <chr>              <dbl>     <dbl>     <dbl>   <dbl>
## 1 (Intercept)        6640.     22.6       294.        0
## 2 Weight_com_Clar    -351.      2.27     -155.        0
```

```
Diamond_model_ay %>%
  glance()
```

```
## # A tibble: 1 x 12
##   r.squared adj.r.squared sigma statistic p.value    df   logLik      AIC    BIC
##       <dbl>         <dbl> <dbl>     <dbl>   <dbl> <dbl>    <dbl>    <dbl>  <dbl>
## 1     0.307         0.307 3321.    23900.       0     1 -513883. 1027772. 1.03e6
## # i 3 more variables: deviance <dbl>, df.residual <int>, nobs <int>
```

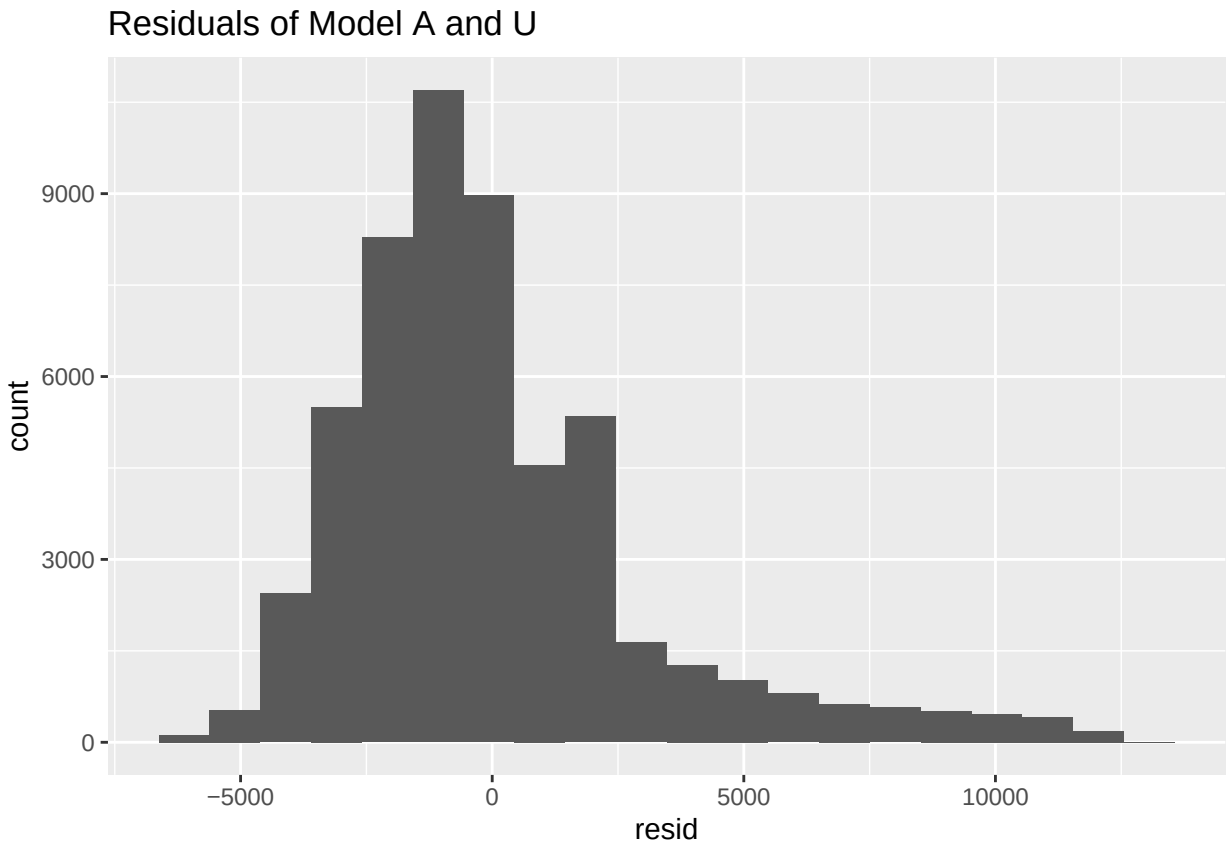
### Dataframing the combined residuals

```
diamond_may_df <- diamond_2 %>%
  add_predictions(Diamond_model_ay) %>%
  add_residuals(Diamond_model_ay)
```

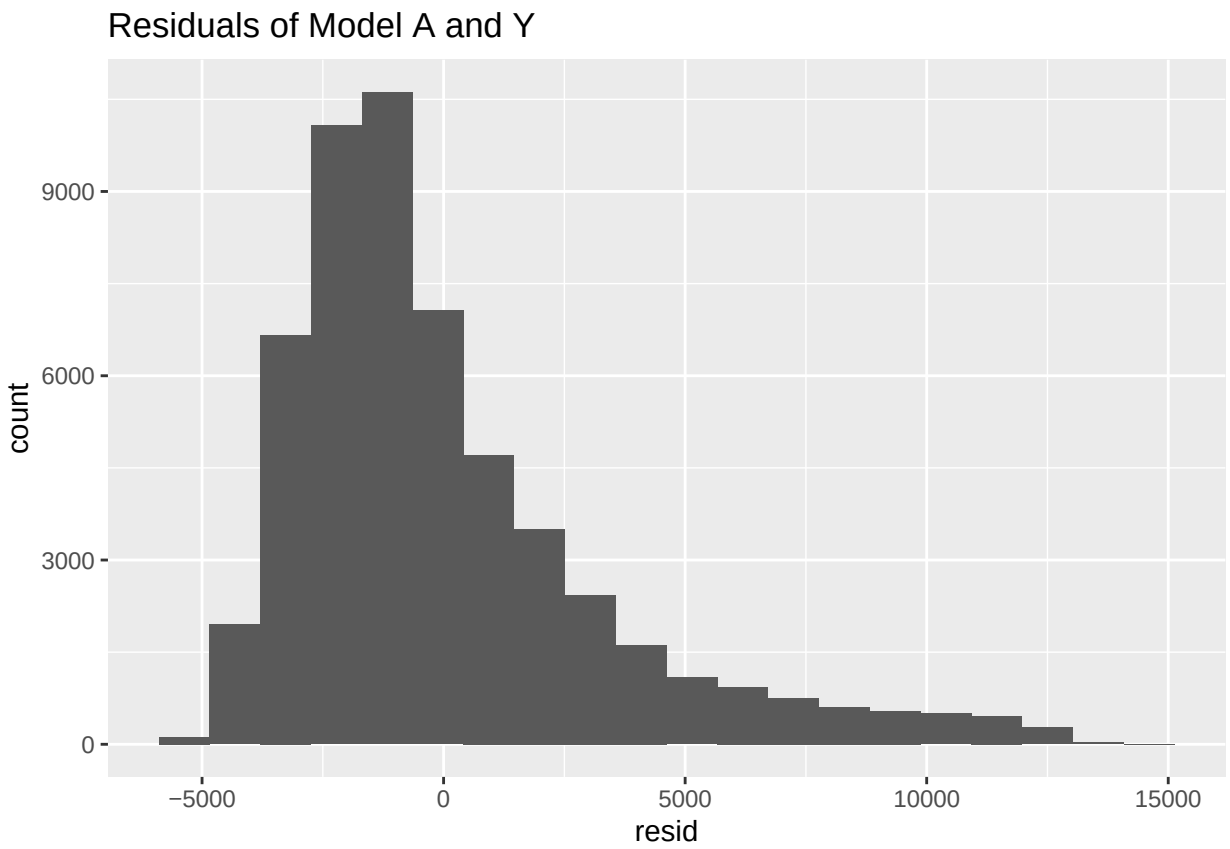
```
diamond_mau_df <- diamond_2 %>%
  add_predictions(Diamond_model_au) %>%
  add_residuals(Diamond_model_au)
```

### Residual histogram plots of the models.

```
diamond_mau_df %>%
  ggplot()+
  geom_histogram(mapping = aes(x = resid), bins = 20) +
  labs(title = "Residuals of Model A and U")
```




```
diamond_may_df %>%
  ggplot()+
  geom_histogram(mapping = aes(x = resid), bins = 20) +
  labs(title = "Residuals of Model A and Y")
```




## 10. References

References:

https://www.vrai.com/journal/post/diamond-carat-price article by Alicia Briggs | February 06, 2023

https://www.kaggle.com/datasets/ayeshaseherr/diamonds.