Regression analysis and diagnostics for Albuquerque housing prices

AUTHOR
Michael Dang

PUBLISHED
September 29, 2024

This program reads data on housing prices in Albuquerque, New Mexico in 1993. Find more information in the data dictionary.

This code is placed in the public domain.

Load the tidyverse library

For most of your programs, you should load the tidyverse library. The broom package provides a nice way to compute residuals and predicted values. The messages and warnings are suppressed.

```
library(broom)
library(tidyverse)
```

Read the data and view a brief summary

Use the read_csv function to read the data. The glimpse function will produce a brief summary.

```
alb <- read_csv(
  file="../data/albuquerque-housing.csv",
  col_names=TRUE,
  col_types="nnnnccc",
  na=".")
glimpse(alb)
```

```
Rows: 117
Columns: 7
$ price        <dbl> 205000, 208000, 215000, 215000, 199900, 190000, 180000, 1…
$ sqft         <dbl> 2650, 2600, 2664, 2921, 2580, 2580, 2774, 1920, 2150, 171…
$ age          <dbl> 13, NA, 6, 3, 4, 4, 2, 1, NA, 1, 4, 8, 15, 14, 18, NA, 16…
$ features     <dbl> 7, 4, 5, 6, 4, 4, 4, 5, 4, 3, 5, 6, 3, 5, 8, 3, 4, 3, 4, …
$ northeast    <chr> "yes", "yes", "yes", "yes", "yes", "yes", "yes", "yes", "…
$ custom_build <chr> "yes", "yes", "yes", "yes", "yes", "no", "no", "yes", "no…
$ corner_lot   <chr> "no", "no", "no", "no", "no", "no", "no", "no", "no", "no…
```

m1: regression analysis using features to predict price

You might expect that a house with more features would have a higher sales price. Your first steps are to compute simple descriptive statistics for both the independent variable (features) and the dependent

variable (price). Then you should plot the data.

m1: Calculate descriptive statistics for number of features

```
          alb |>
            summarise(
              features_mn=mean(features, na.rm=TRUE),
              features_sd=sd(features, na.rm=TRUE),
              features_min=min(features, na.rm=TRUE),
              features_max=max(features, na.rm=TRUE),
              n_missing=sum(is.na(features)))
```

```
# A tibble: 1 × 5
  features_mn features_sd features_min features_max n_missing
        <dbl>       <dbl>        <dbl>        <dbl>     <int>
1        3.53        1.41            0            8         0
```

The average number of features is small (3.5) and the standard deviation (1.4) indicates very little variation. At least one house has zero features and no house has all 13 features.

m1: Calculate descriptive statistics for price

```
          alb |>
            summarize(
              price_mn=mean(price, na.rm=TRUE),
              price_sd=sd(price, na.rm=TRUE),
              price_min=min(price, na.rm=TRUE),
              price_max=max(price, na.rm=TRUE),
              n_missing=sum(is.na(price)))
```

```
# A tibble: 1 × 5
  price_mn price_sd price_min price_max n_missing
     <dbl>    <dbl>     <dbl>     <dbl>     <int>
1  106274.   38044.     54000    215000         0
```

The average price is low ($106,000), but the standard deviation ($38,000) shows a fair amount of variation. Note that a dollar sign in R has special meaning. To get it to print normally, you have to put a backslash in front of it.

Question 1

Calculate descriptive statistics (mean, standard deviation, minimum, and maximum for sqft. Interpret these numbers

```
        alb |>
          summarize(
            sqft_mn=mean(sqft, na.rm=TRUE),
            sqft_sd=sd(sqft, na.rm=TRUE),
            sqft_min=min(sqft, na.rm=TRUE),
            sqft_max=max(sqft, na.rm=TRUE),
            n_missing=sum(is.na(sqft)))
```

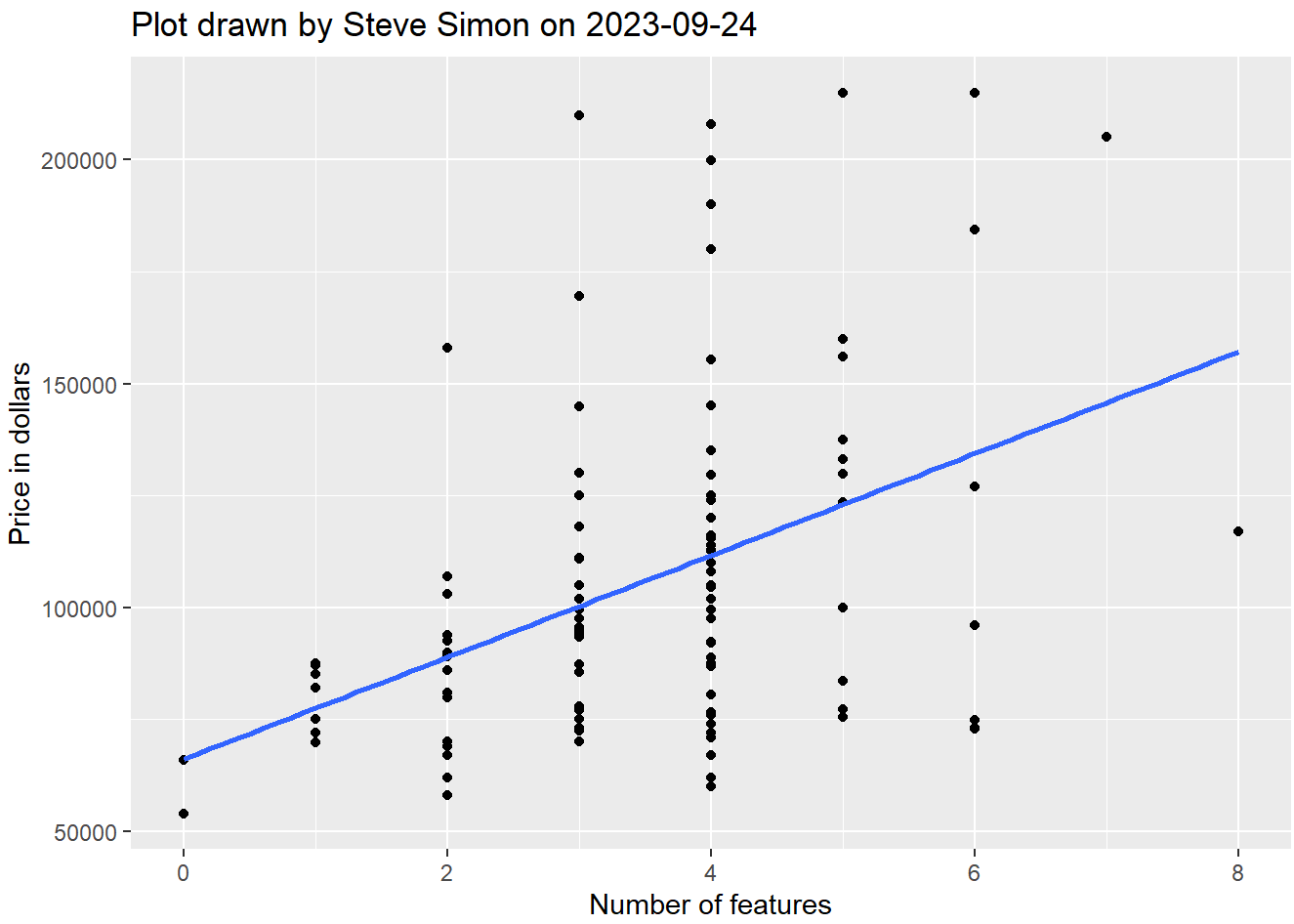
```
# A tibble: 1 × 5
  sqft_mn sqft_sd sqft_min sqft_max n_missing
    <dbl>   <dbl>    <dbl>    <dbl>     <int>
1   1654.    524.      837     3750         0
```

The average sqft per home in this dataset is 1654, but there is significant variability in home sizes, with some being as small as 837 sqft and others as large as 3750 sqft.

m1: Plot features versus price

```
        alb |>
          ggplot(aes(features, price)) +
            geom_point() +
            geom_smooth(method="lm", se=FALSE) +
              ggtitle("Plot drawn by Steve Simon on 2023-09-24") +
              xlab("Number of features") +
              ylab("Price in dollars")
```

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



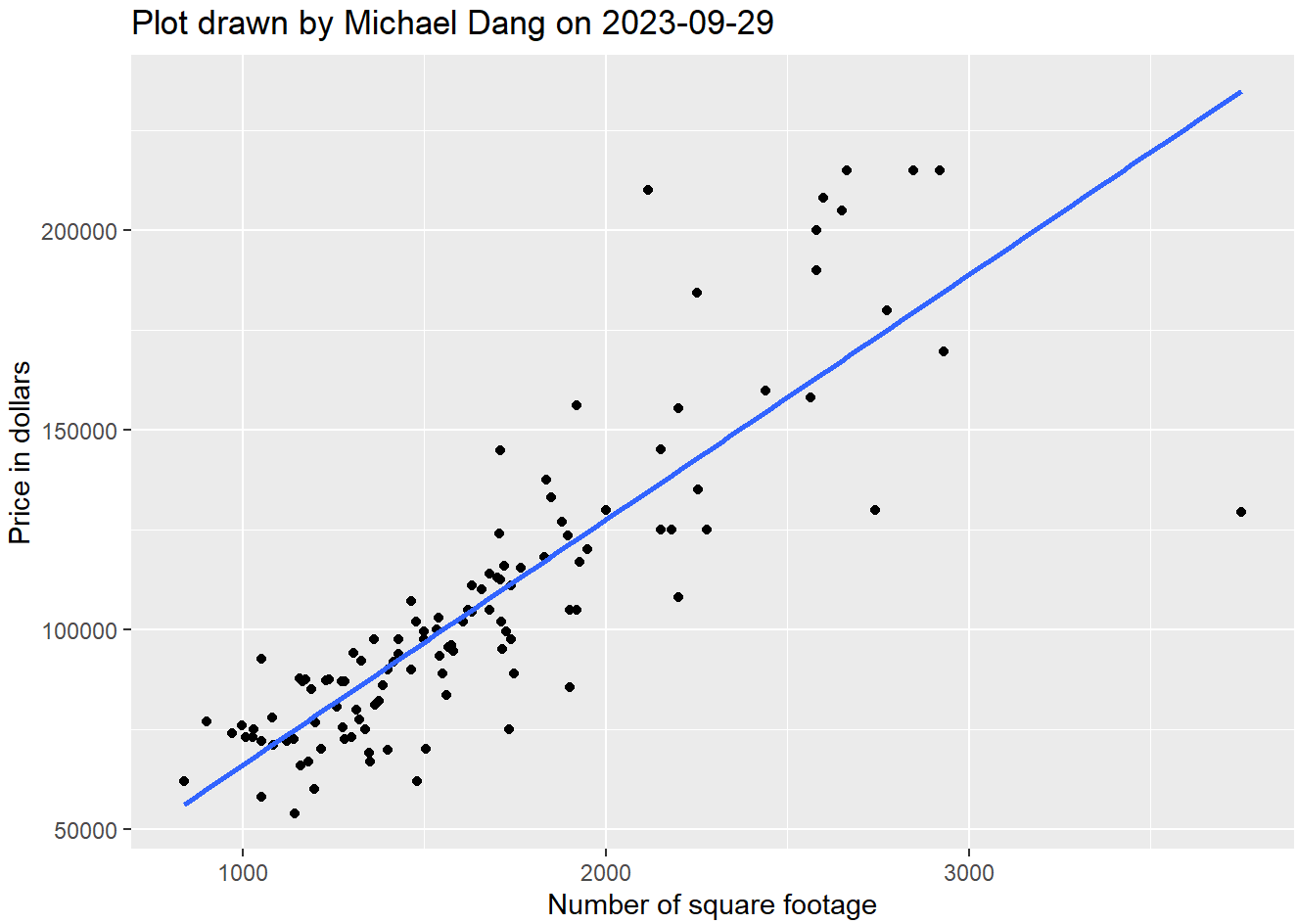
There is a weak positive relationship between the number of features and the price of a house.

Question 2

Draw a plot with price on the y-axis and sqft on the x-axis. Include a linear regression line, but do not extend it beyond the range of the data. Interpret this plot.

```
alb |>
  ggplot(aes(sqft, price)) +
    geom_point() +
    geom_smooth(method="lm", se=FALSE) +
      ggtitle("Plot drawn by Michael Dang on 2023-09-29") +
      xlab("Number of square footage") +
      ylab("Price in dollars")
```

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



The linear regression line captures the overall upward trend and the plot shows a positive relationship between square footage and price, meaning larger homes generally have higher prices.

m1: Use features to predict price

```
m1 <- lm(price~features, data=alb)
m1
```

```
Call:
lm(formula = price ~ features, data = alb)

Coefficients:
(Intercept)     features
      66117        11376
```

The estimated average sales price for a house with no features is $66,000. This not an extrapolation beyond the range of the data. The estimated average sales price increases by $11,000 for each additional feature. This is surprisingly large when you look at what the features are. Perhaps houses with more features are bigger and newer.

Question 3

Calculate a linear regression model using sqft to predict price. Interpret the slope and intercept.

```
m2 <- lm(price~sqft, data=alb)
m2
```

```
Call:
lm(formula = price ~ sqft, data = alb)

Coefficients:
(Intercept)         sqft
    4781.93        61.37
```

The estimated average sales price for a house with no square footage is $4,781.93, though this doesn't have practical meaning since no house would have zero square footage. The estimated average sales price increases by $61.37 for each additional square foot, which seems reasonable given that larger homes typically command higher prices. Other factors, such as location and house features, might also play a role in determining the overall price.

Skip some of the functions for hypothesis tests and p-values

Normally, you would follow this up with various functions like anova(), confint(), or tidy(). This program skips those steps to focus on the diagnostic plots of the residuals.

m1: Calculate residuals and predicted values

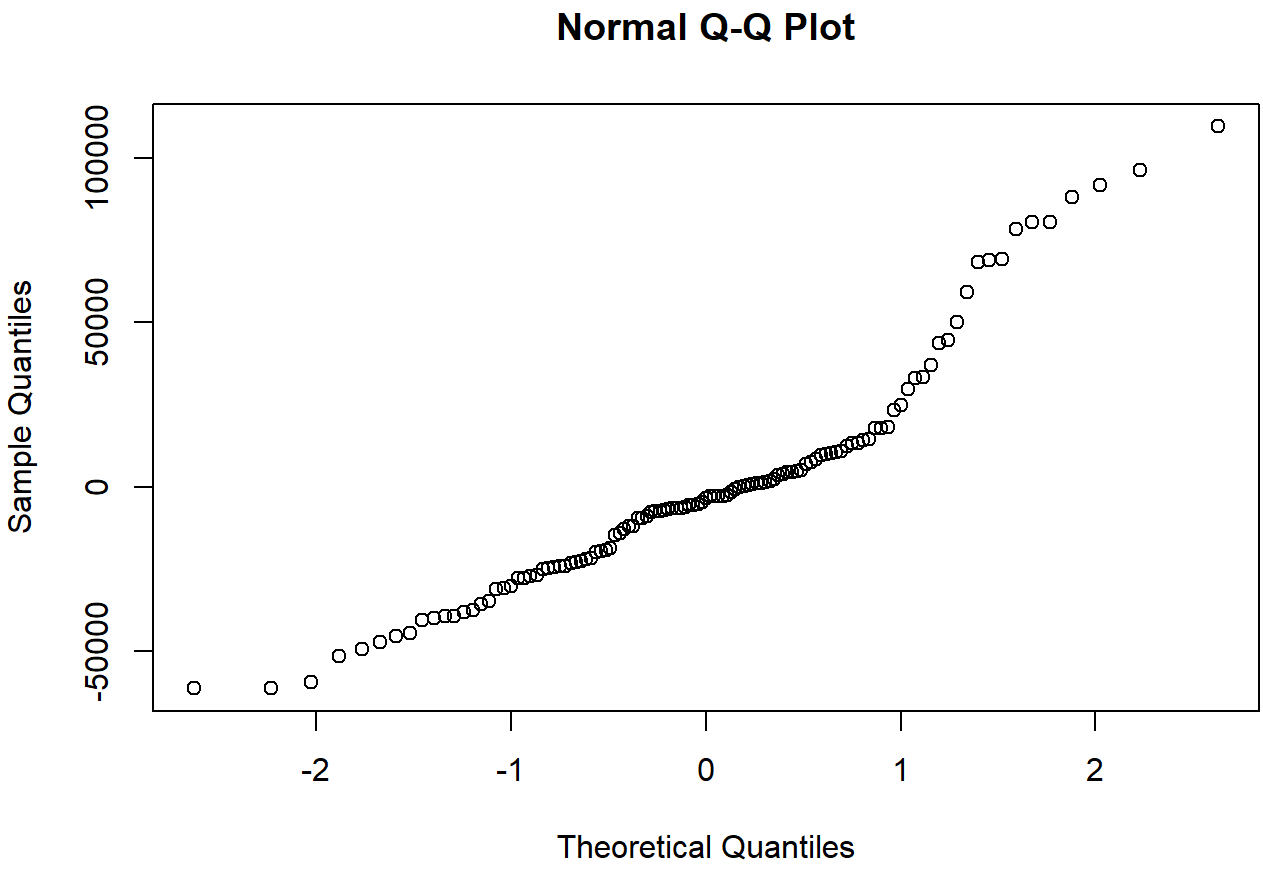
```
r1 <- augment(m1)
glimpse(r1)
```

```
Rows: 117
Columns: 8
$ price      <dbl> 205000, 208000, 215000, 215000, 199900, 190000, 180000, 156…
$ features   <dbl> 7, 4, 5, 6, 4, 4, 4, 5, 4, 3, 5, 6, 3, 5, 8, 3, 4, 3, 4, 3,…
$ .fitted    <dbl> 145748.98, 111621.17, 122997.11, 134373.04, 111621.17, 1116…
$ .resid     <dbl> 59251.0183, 96378.8325, 92002.8944, 80626.9564, 88278.8325,…
$ .hat       <dbl> 0.061096606, 0.009511376, 0.017978366, 0.035173443, 0.00951…
$ .sigma     <dbl> 34348.09, 33620.33, 33719.15, 33963.04, 33816.76, 34032.12,…
$ .cooksd    <dbl> 1.012082e-01, 3.745903e-02, 6.563869e-02, 1.021705e-01, 3.1…
$ .std.resid <dbl> 1.76370021, 2.79316324, 2.67781374, 2.36752756, 2.55841645,…
```

You could have also used the resid() and predict() functions. No interpretation is needed here, as these numbers are better reviewed using various graphical displays.

m1: Normal probability plot for residuals

```
qqnorm(r1$.resid)
```

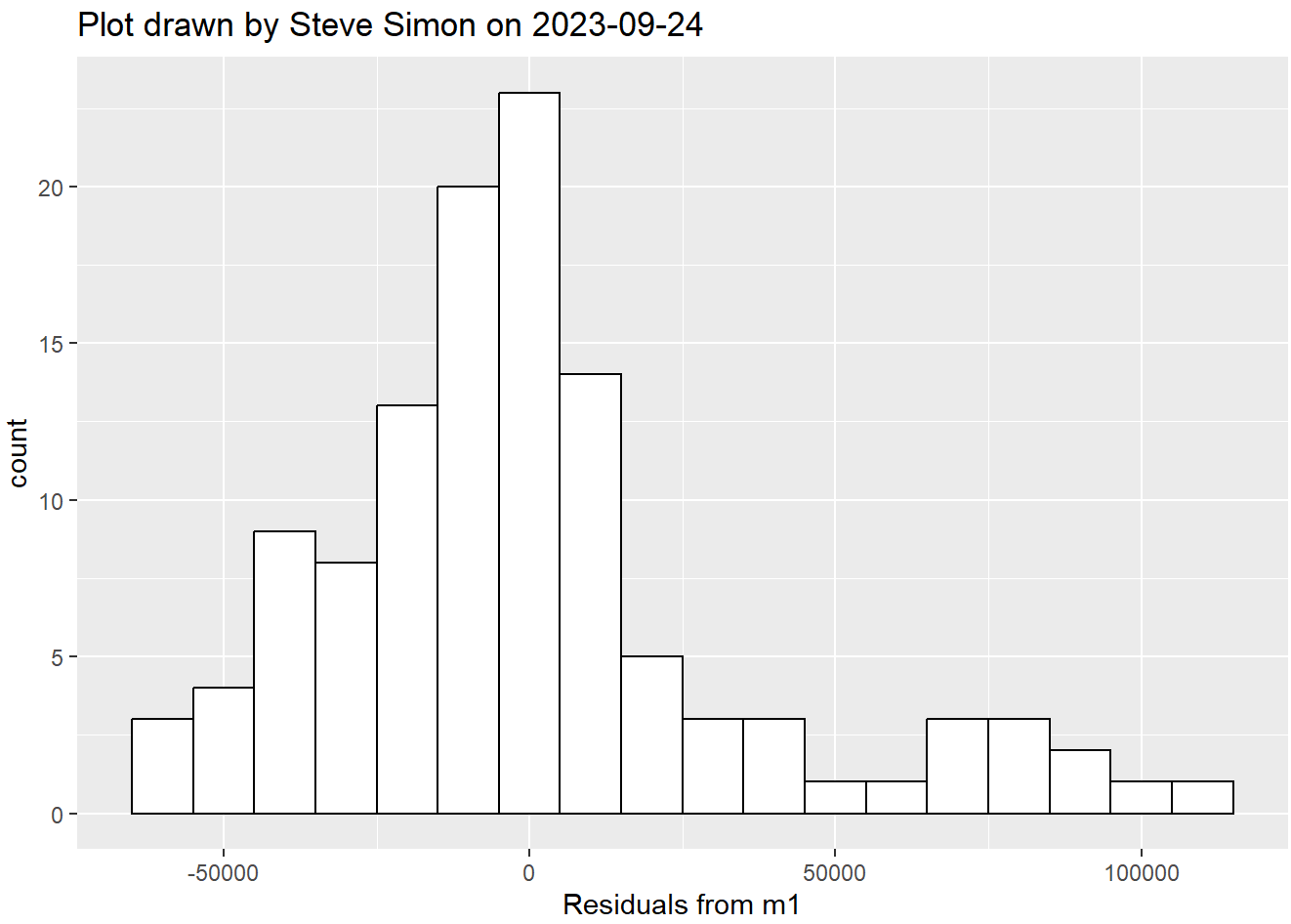


The normal probability plot deviates markedly from a straight line, indicating some possible issues with the normality assumption.

Note that you cannnot use ggtitle, xlab, or ylab with the qqnorm function.

m1: Histogram for residuals

```
r1 |>
  ggplot(aes(.resid)) +
    geom_histogram(
      binwidth=10000,
      color="black",
      fill="white") +
      ggtitle("Plot drawn by Steve Simon on 2023-09-24") +
      xlab("Residuals from m1")
```



The histogram reinforces these concerns. It looks like the data is skewed to the right.

Question 4

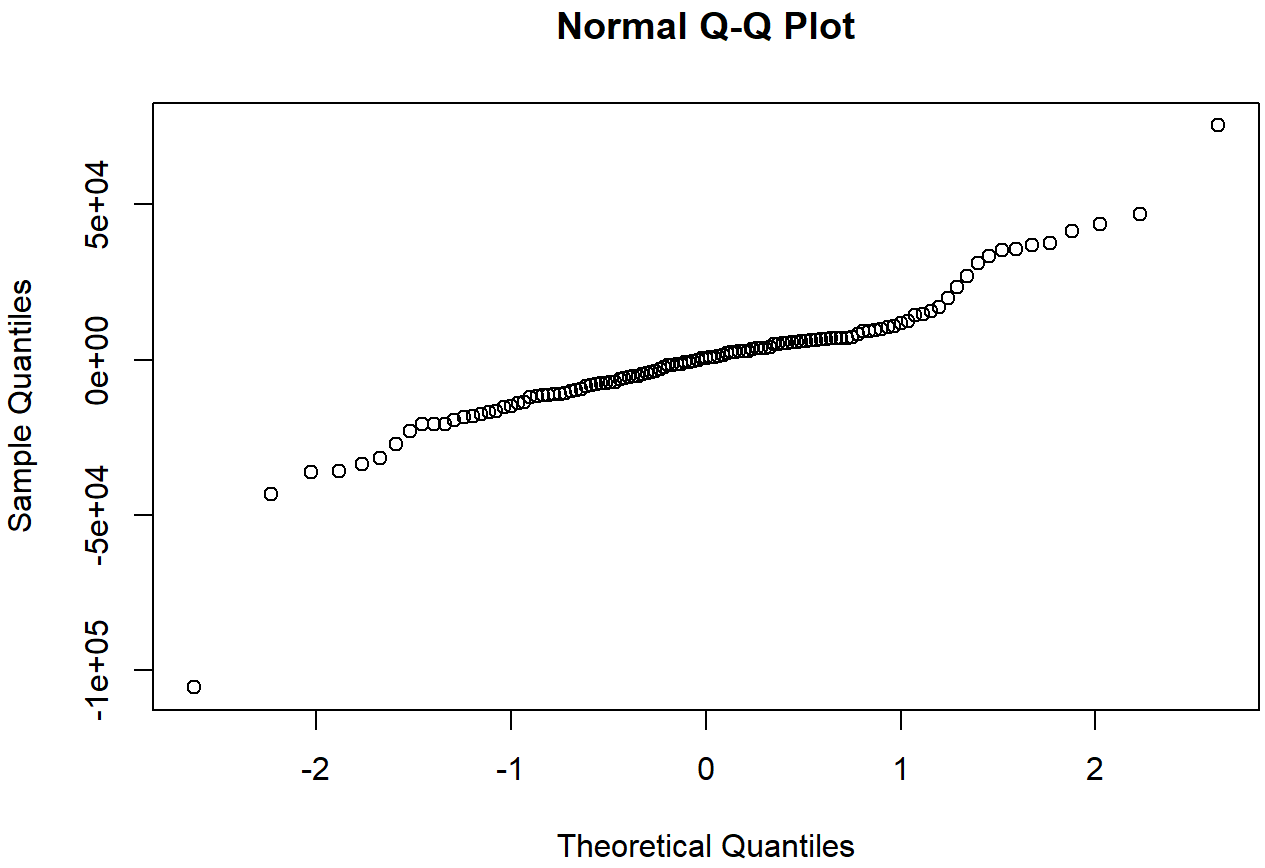
Draw a normal probability plot and a histogram for the residuals (.resid). Interpret these plots.

```
r2 <- augment(m2)
glimpse(r2)
```

```
Rows: 117
Columns: 8
$ price      <dbl> 205000, 208000, 215000, 215000, 199900, 190000, 180000, 156…
$ sqft       <dbl> 2650, 2600, 2664, 2921, 2580, 2580, 2774, 1920, 2150, 1710,…
$ .fitted    <dbl> 167403.63, 164335.30, 168262.77, 184034.01, 163107.97, 1631…
$ .resid     <dbl> 37596.3651, 43664.6991, 46737.2315, 30965.9946, 36792.0327,…
$ .hat       <dbl> 0.039734786, 0.036682514, 0.040617583, 0.059012195, 0.03550…
$ .sigma     <dbl> 20217.75, 20107.41, 20042.38, 20315.77, 20232.60, 20373.81,…
$ .cooksd    <dbl> 7.285696e-02, 9.015129e-02, 1.153048e-01, 7.644246e-02, 6.1…
$ .std.resid <dbl> 1.87655228, 2.17598631, 2.33387459, 1.56136142, 1.83237492,…
```

Normal probability plot

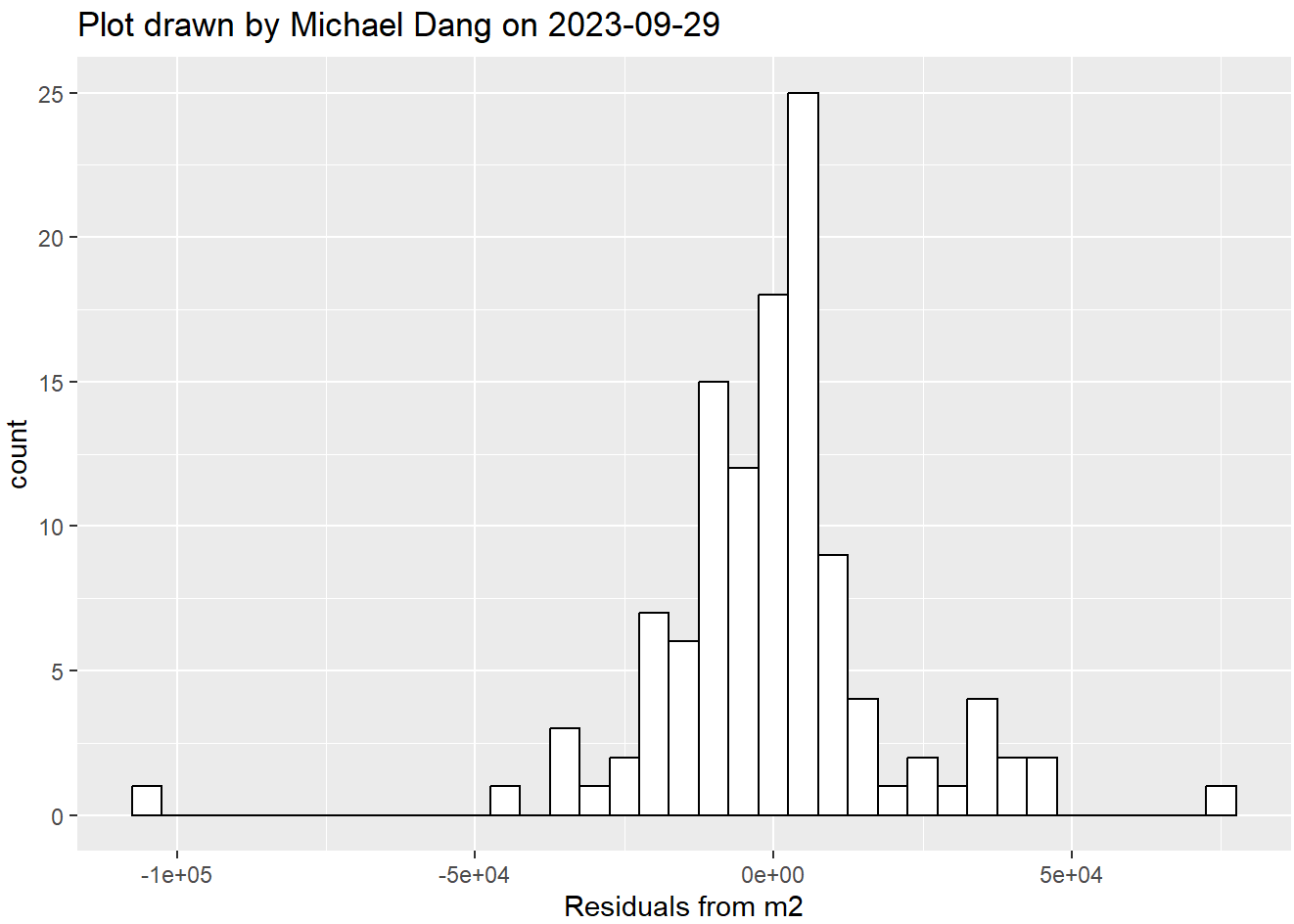
```
qqnorm(r2$.resid)
```



The points mostly follow the straight line, which suggests that the residuals are approximately normally distributed, though there might be slight deviations at the tails.

Histogram for the residuals

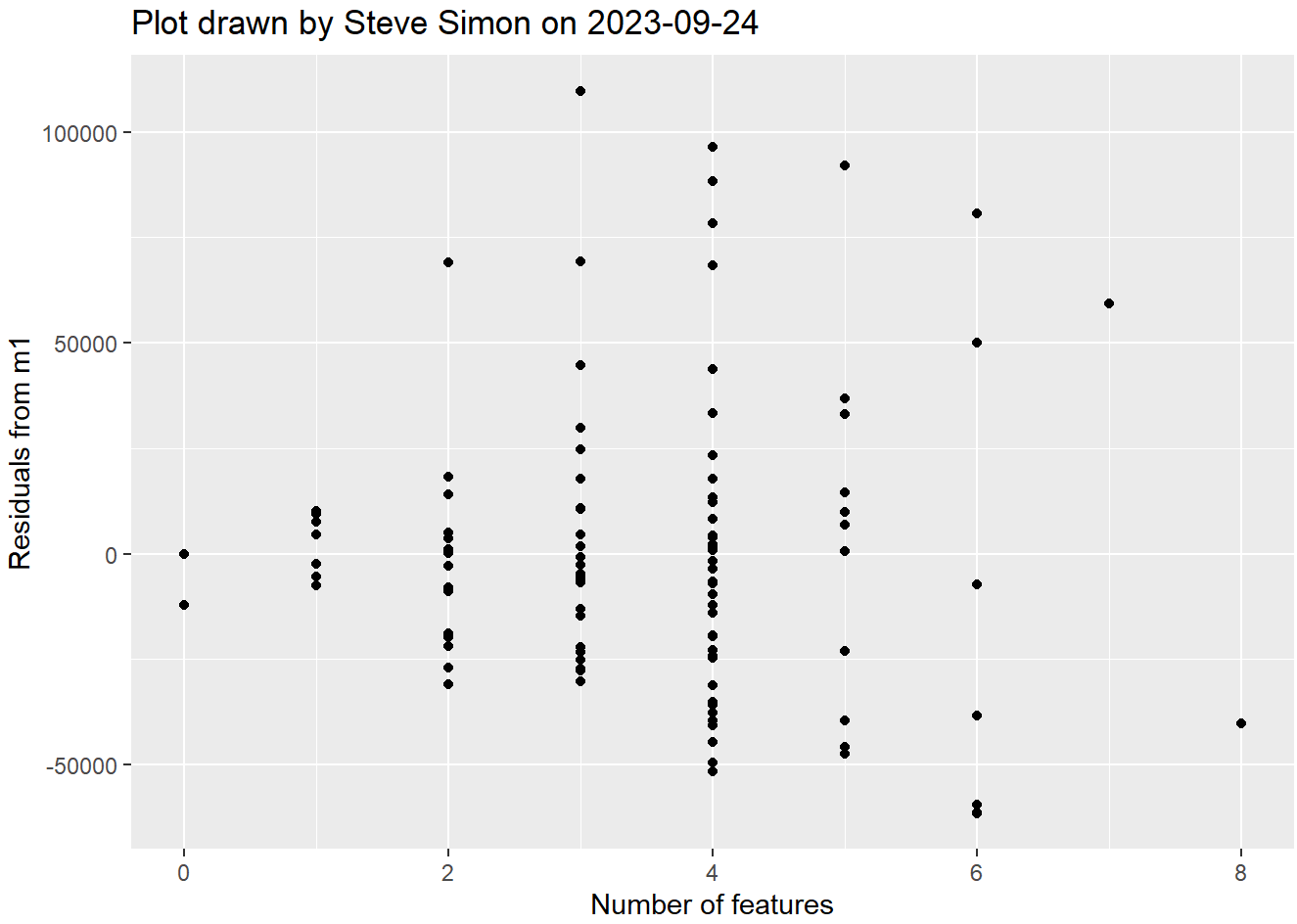
```
r2 |>
  ggplot(aes(.resid)) +
    geom_histogram(
      binwidth=5000,
      color="black",
      fill="white") +
      ggtitle("Plot drawn by Michael Dang on 2023-09-29") +
      xlab("Residuals from m2")
```



The shape of the histogram suggests that most residuals are clustered near zero, but there might be some spread, which could indicate slight deviations from normality.

m1: Plot residuals versus features

```
r1 |>
  ggplot(aes(features, .resid)) +
    geom_point() +
      ggtitle("Plot drawn by Steve Simon on 2023-09-24") +
      xlab("Number of features") +
      ylab("Residuals from m1")
```

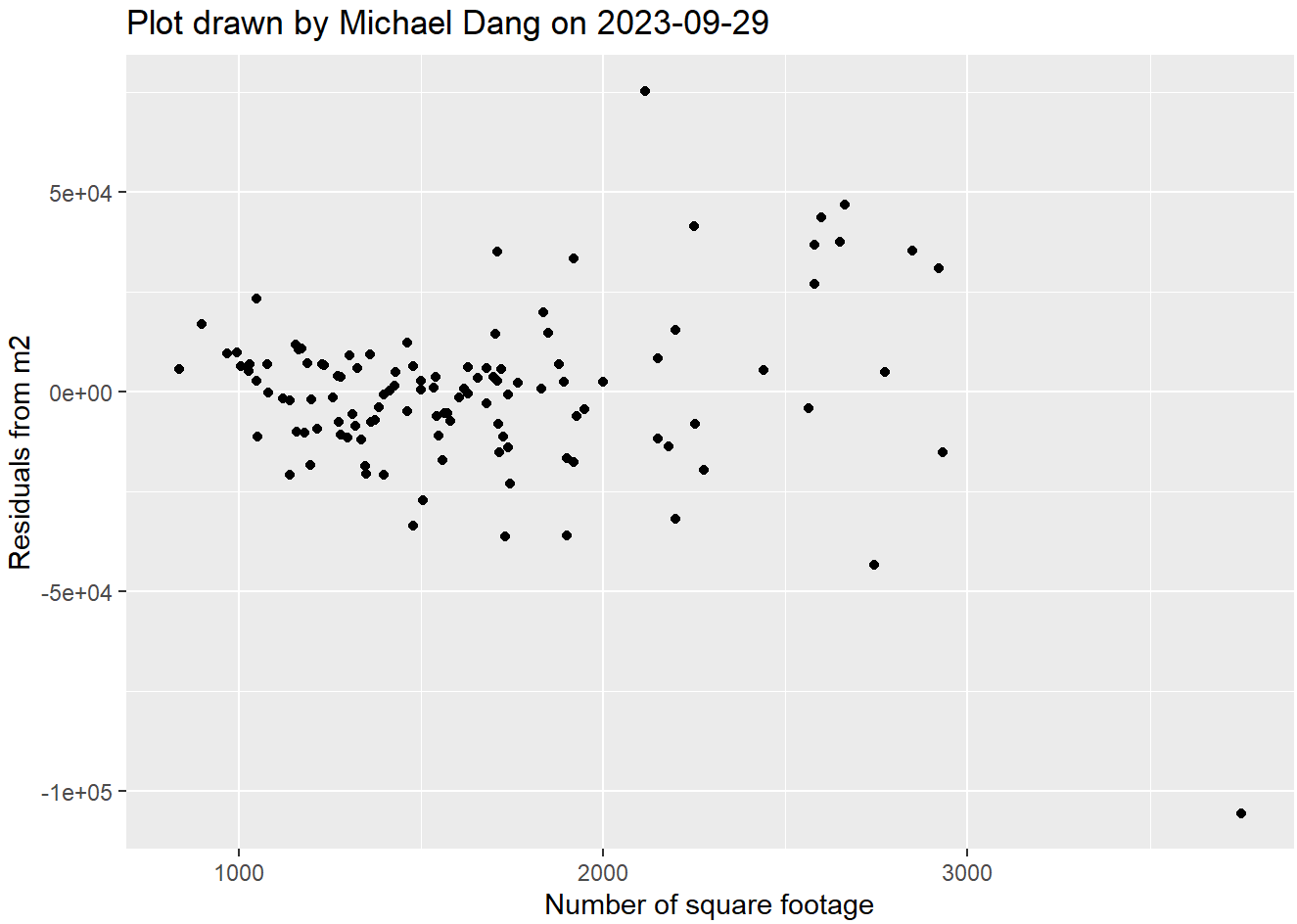


This plot is difficult to interpret. There is some evidence of heterogeneity. It looks, perhaps, like houses with more features also tend to exhibit more variation. There is no evidence of non-linearity.

Question 5

Draw a scatterplot of sqft on the x-axis and the residuals on the y-axis. Is there evidence of non-linearity or heterogeneity?

```
r2 |>
  ggplot(aes(sqft, .resid)) +
    geom_point() +
      ggtitle("Plot drawn by Michael Dang on 2023-09-29") +
      xlab("Number of square footage") +
      ylab("Residuals from m2")
```



For non-linearity, there doesn't appear to be a clear systematic pattern in the residuals that would suggest strong non-linearity. However, some subtle clustering of residuals (near the middle range of square footage) could indicate mild non-linear effects, but it's not a pronounced issue.

For heterogeneity, the spread of the residuals seems to increase slightly as the square footage increases, which suggests some degree of heteroscedasticity.

m1: Leverage values

```
n <- nrow(r1)
r1 |> filter(.hat > 3*2/n)
```

```
# A tibble: 4 × 8
   price features .fitted  .resid   .hat .sigma     .cooksd .std.resid
   <dbl>    <dbl>   <dbl>   <dbl>  <dbl>  <dbl>       <dbl>      <dbl>
1 205000        7 145749.  59251. 0.0611 34348. 0.101          1.76
2 117000        8 157125. -40125. 0.0957 34597. 0.0784        -1.22
3  54000        0  66117. -12117. 0.0629 34803. 0.00438       -0.361
4  66000        0  66117.   -117. 0.0629 34822. 0.000000411   -0.00350
```

There are four data points with high leverage. These correspond to the houses with the most and the fewest features.

Question 6

Display the data (if any) for leverage values greater than 3*2/n. Describe where these leverage values are found relative to the independent and/or dependent variables.

```
n <- nrow(r2)
r2 |> filter(.hat > 3*2/n)
```

```
# A tibble: 4 × 8
   price  sqft .fitted   .resid   .hat .sigma .cooksd .std.resid
   <dbl> <dbl>   <dbl>    <dbl>  <dbl>  <dbl>   <dbl>      <dbl>
1 215000  2921 184034.   30966. 0.0590 20316.  0.0764      1.56
2 169500  2931 184648.  -15148. 0.0598 20482.  0.0186     -0.764
3 215000  2848 179554.   35446. 0.0534 20249.  0.0895      1.78
4 129500  3750 234907. -105407. 0.147  17535.  2.68       -5.58
```

The leverage points are associated with the higher end of square footage values, which suggests that homes with larger square footage are more influential in the model. These leverage points are associated with homes that have larger square footage values (ranging from approximately 2848 to 3750 sqft), and they also tend to have specific characteristics such as being custom-built or located on a corner lot.

m1: Studentized deleted residual

```
r1 |>
  filter(abs(.std.resid) > 3)
```

```
# A tibble: 1 × 8
   price features .fitted  .resid    .hat .sigma .cooksd .std.resid
   <dbl>    <dbl>   <dbl>   <dbl>   <dbl>  <dbl>   <dbl>      <dbl>
1 210000        3 100245. 109755. 0.00977 33255.  0.0499       3.18
```

Only one house, with only an average number of features (3) but with the highest sales price ($215,000), might be considered an outlier.

Question 7

Display the data (if any) for studentized deleted residuals (.std.resid) values greater than 3. Describe where these leverage values are found relative to the independent and/or dependent variables.

```
r2 |>
  filter(abs(.std.resid) > 3)
```

```
# A tibble: 2 × 8
   price  sqft .fitted   .resid   .hat .sigma .cooksd .std.resid
   <dbl> <dbl>   <dbl>    <dbl>  <dbl>  <dbl>   <dbl>      <dbl>
```

```
1 129500  3750 234907. -105407. 0.147  17535.   2.68       -5.58
2 210000  2116 134634.   75366. 0.0153 19263.   0.107       3.71
```

The first point has a sqft of 3750, which is on the higher end, suggesting that the model struggles to predict accurately for very large homes.Also, the first home is priced at $129,500, which is relatively low compared to its large square footage, explaining the large negative residual.

The second point has a sqft of 2116, which is closer to the middle range but still has a large residual, indicating the model's difficulty with this specific case.Also, the second home is priced at $210,000, and the model has under-predicted its price, leading to a large positive residual.

m1: Cook's distance

```
r1 |>
  filter(.cooksd > 1)
```

```
# A tibble: 0 × 8
# ℹ 8 variables: price <dbl>, features <dbl>, .fitted <dbl>, .resid <dbl>,
#   .hat <dbl>, .sigma <dbl>, .cooksd <dbl>, .std.resid <dbl>
```

No houses had a large value for Cook's distance. Even though there are a few high leverage points and one outlier, no single data point has unusually high influence on the predicted values.

Question 8

Display the data (if any) for Cook's distance (.cooksd) values greater than 1. Describe where these leverage values are found relative to the independent and/or dependent variables.

```
r2 |>
  filter(.cooksd > 1)
```

```
# A tibble: 1 × 8
   price  sqft .fitted   .resid  .hat .sigma .cooksd .std.resid
   <dbl> <dbl>   <dbl>    <dbl> <dbl>  <dbl>   <dbl>      <dbl>
1 129500  3750 234907. -105407. 0.147 17535.    2.68      -5.58
```

This particular data point has a Cook's distance of 2.676273, which is significantly greater than 1. This indicates that this point is highly influential in the regression model.

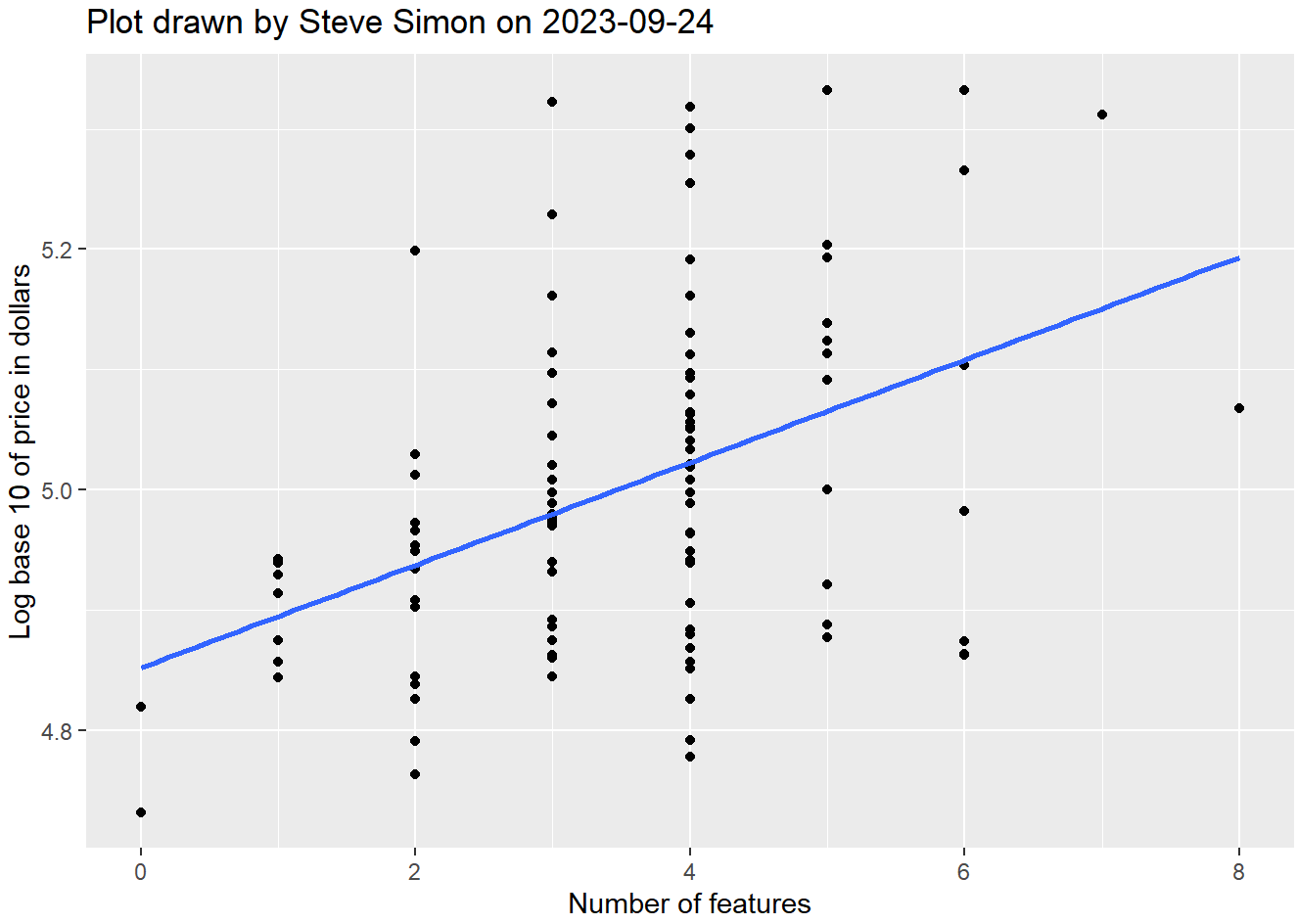
m2: Using features to predict log(price)

Because there are some concerns about non-normality and heterogeneity, you might consider using a log transformation for price. In this example, a base 10 logarithm is a reasonable choice.

m2: scatterplot

```
alb$log_price <- log10(alb$price)
alb |>
  ggplot(aes(features, log_price)) +
    geom_point() +
    geom_smooth(method="lm", se=FALSE) +
      ggtitle("Plot drawn by Steve Simon on 2023-09-24") +
      xlab("Number of features") +
      ylab("Log base 10 of price in dollars")
```

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



There is a weak positive linear relationship between log price and features.

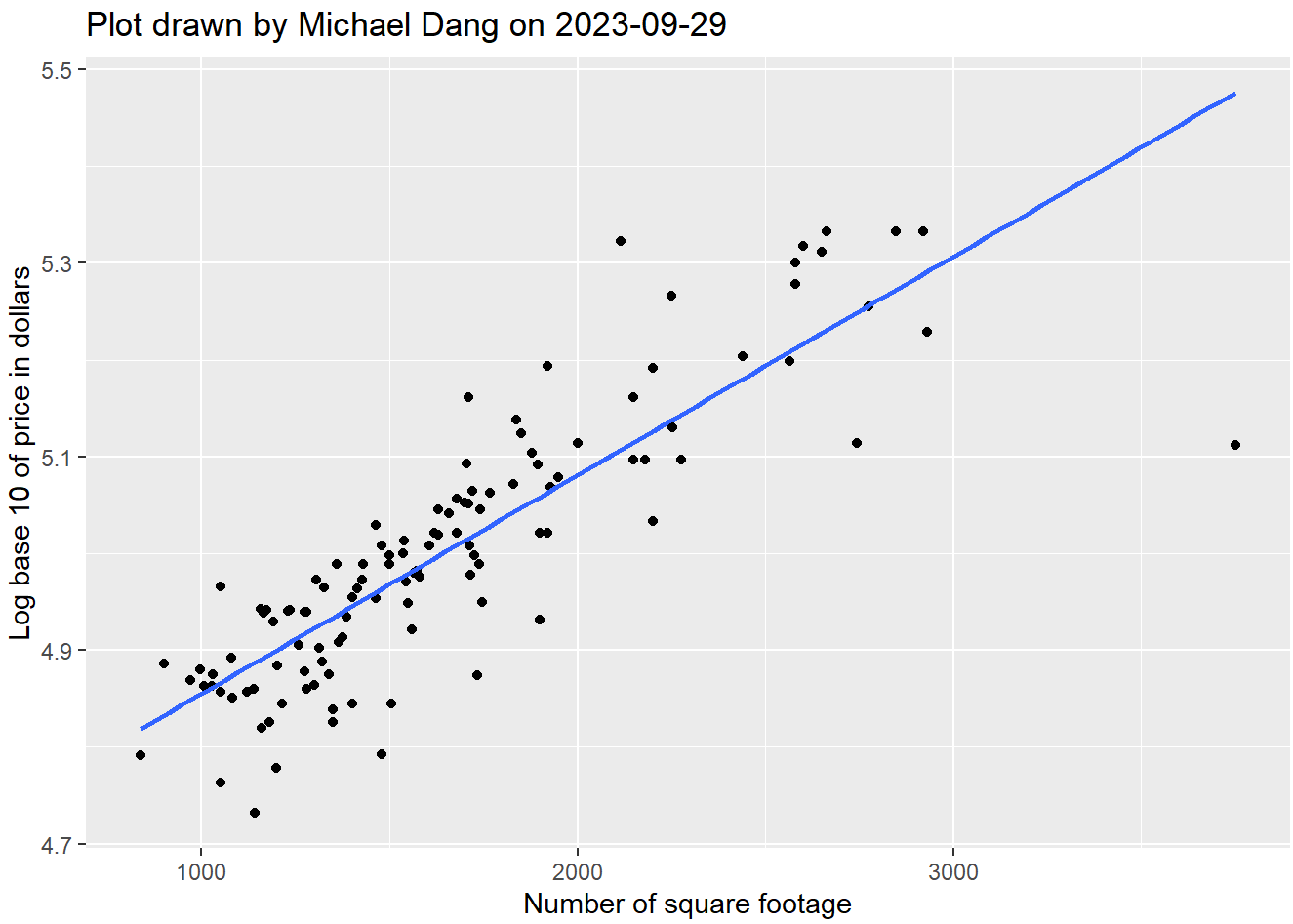
Question 9

Calculate the regression equation predicting log10 of price using sqft. Transform the coefficients back to the original scale of measurement and interpret these values.

```
alb$log_price <- log10(alb$price)
alb |>
```

```
        ggplot(aes(sqft, log_price)) +
          geom_point() +
          geom_smooth(method="lm", se=FALSE) +
            ggtitle("Plot drawn by Michael Dang on 2023-09-29") +
            xlab("Number of square footage") +
            ylab("Log base 10 of price in dollars")
```

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



It look the same from question 2, suggesting that the log transformation did not significantly affect the relationship between square footage and price.

m2: linear regression on log transformed price

```
        m2 <- lm(log_price~features, data=alb)
        m2
```

```
Call:
lm(formula = log_price ~ features, data = alb)

Coefficients:
```

```
(Intercept)     features
    4.85245      0.04263
```

The estimated average log price is 4.8 for a house with no features. The estimated average log price increases by 0.043 for each additional feature. These numbers are easier to interpret when transformed back to the original scale.

m2: Coefficients back transformed to original scale

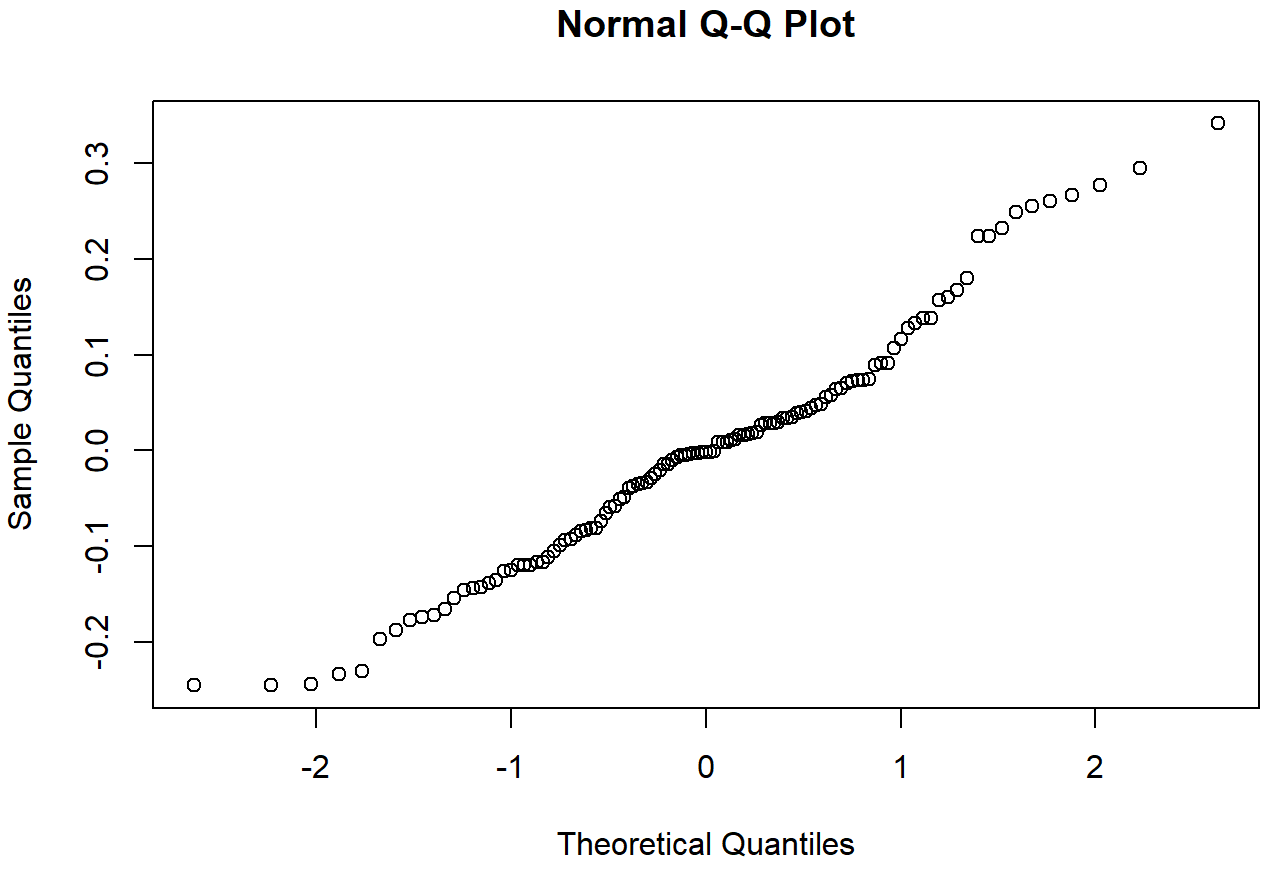
```
10^(coef(m2))
```

```
 (Intercept)     features
71195.269779     1.103135
```

The estimated average price is $71,000 for a house with no features. The estimated average price increases by 1.10 (10%) for each additional feature.

m2: Normal probability plot

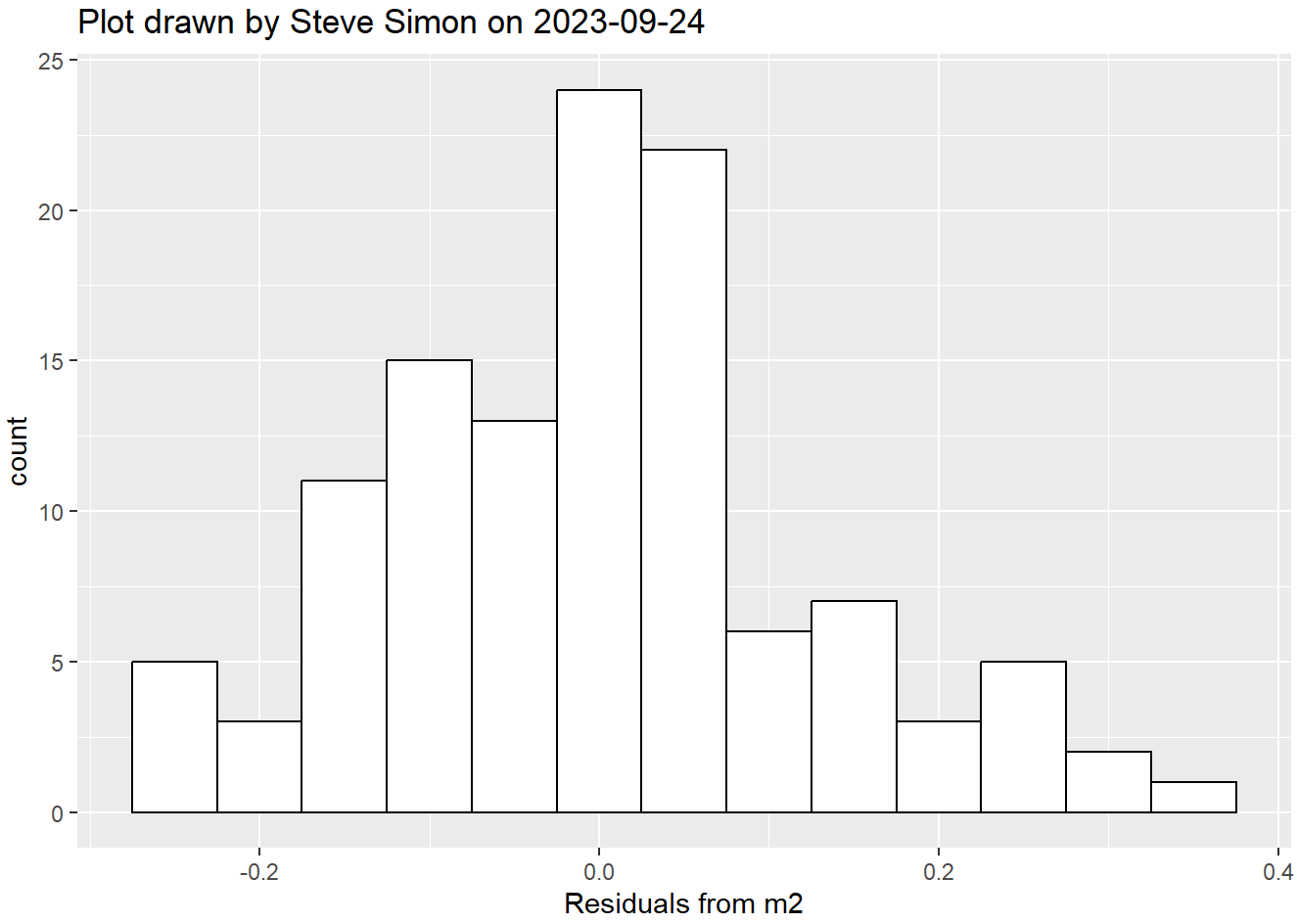
```
r2 <- augment(m2)
qqnorm(r2$.resid)
```



The normal probability plot is close to a straight line, indicating a reasonably close fit to a normal distribution.

m2: Histogram of residuals

```
r2 |>
  ggplot(aes(.resid)) +
    geom_histogram(
      binwidth=0.05,
      color="black",
      fill="white") +
      ggtitle("Plot drawn by Steve Simon on 2023-09-24") +
      xlab("Residuals from m2")
```

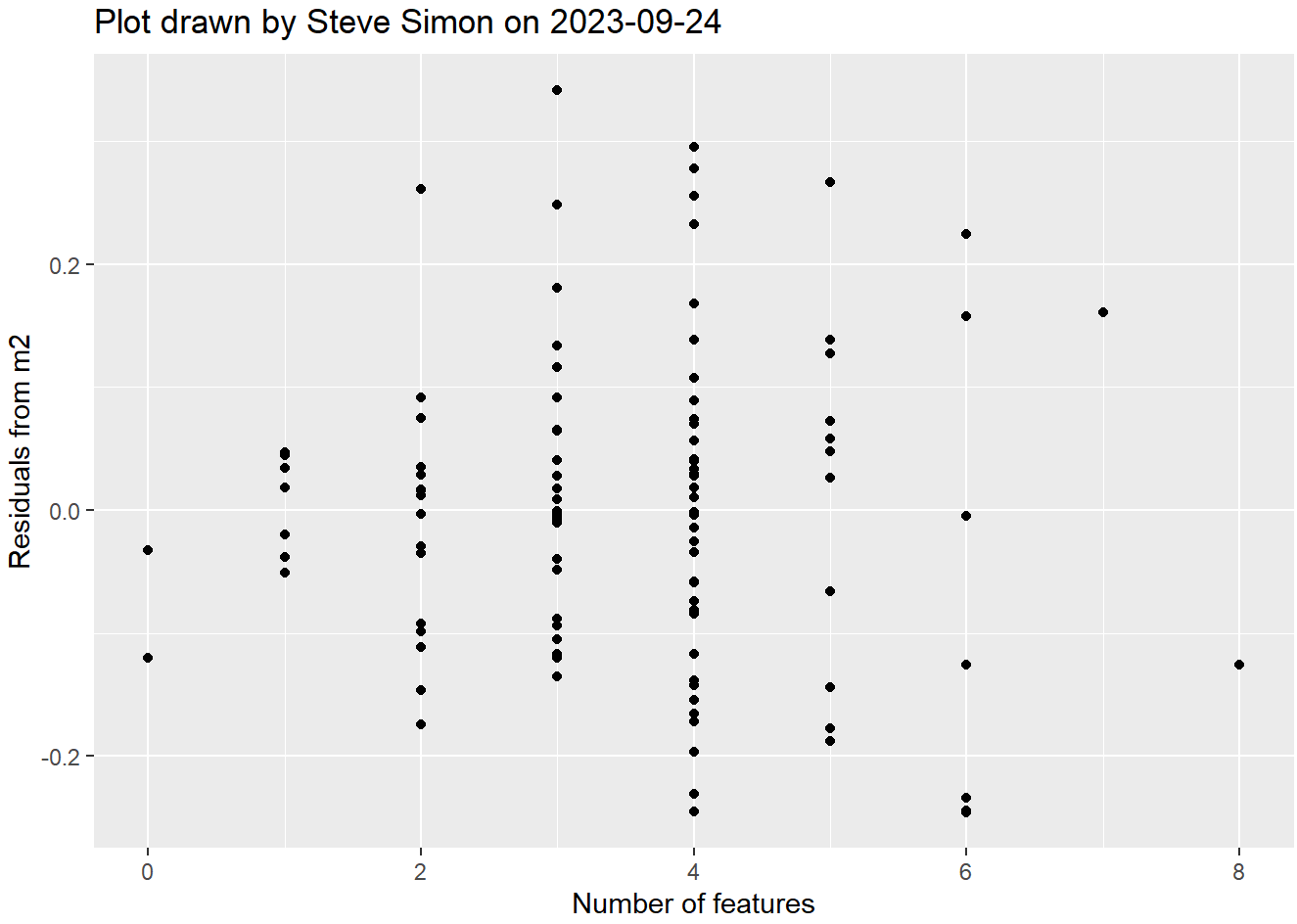


The histogram of residuals also indicates a close fit to a normal distribution. The regression model using log price does a better job meeting the normality assumption.

m2: Plot residuals versus features

```
r2 |>
  ggplot(aes(features, .resid)) +
```

```
  geom_point() +
    ggtitle("Plot drawn by Steve Simon on 2023-09-24") +
    xlab("Number of features") +
    ylab("Residuals from m2")
```



This plot is difficult to interpret. There is certainly no evidence of non-linearity, but perhaps the problems with heterogenity persist even after the log transformation. Houses with zero or one features seem to have less variation than the rest of the data.

Question 10

Calculate diagnostic plots (normal probability plot, histogram, and sqft versus residuals). Do these plots show that a model using log10 price better meets the assumptions for linear regression?

Linear regression on log transformed price

```
m3 <- lm(log_price~sqft, data=alb)
m3
```

```
Call:
lm(formula = log_price ~ sqft, data = alb)
```

```
Coefficients:
(Intercept)         sqft
  4.6294697    0.0002258
```

The estimated average sales log price for a house with no square footage is 4.6, though this doesn't have practical meaning since no house would have zero square footage. The estimated average sales log price increases by 0.0002 for each additional square foot.

Coefficients back transformed to original scale

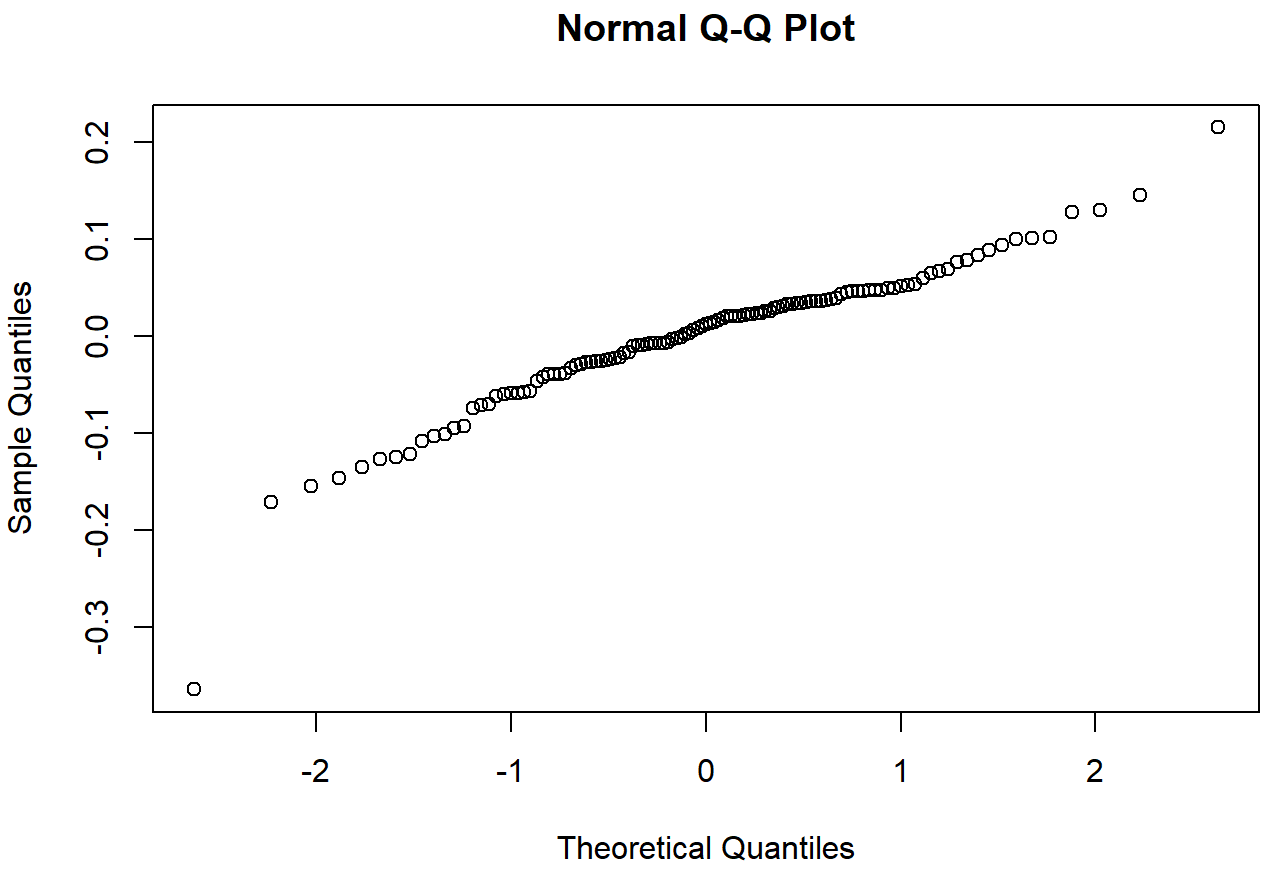
```
10^(coef(m3))
```

```
(Intercept)         sqft
42605.89143     1.00052
```

The estimated average price is $42,605 for a house with no features. The estimated average price increases by 1.00052 (0.52%) for each additional square footage.

Normal probability plot

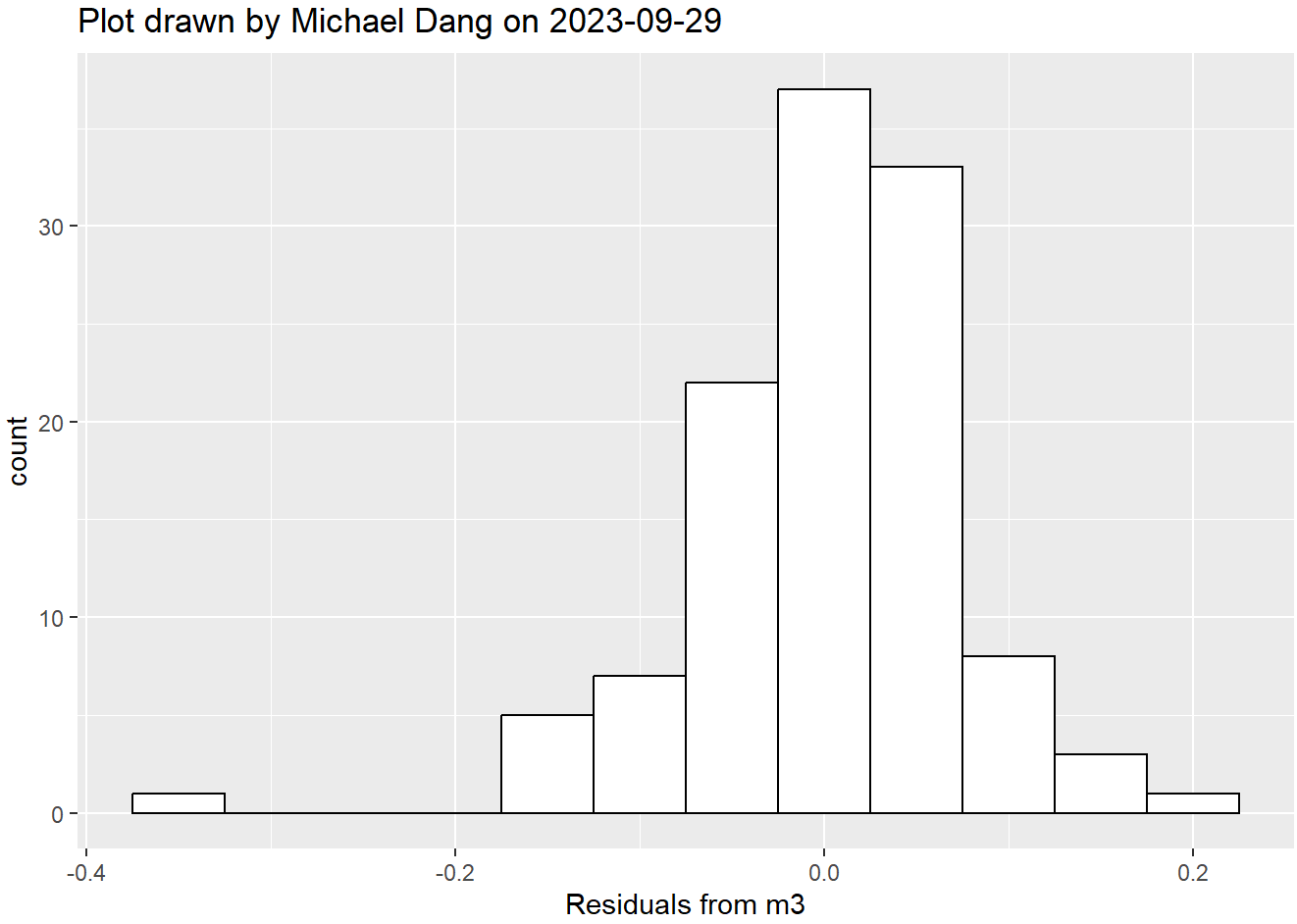
```
r3 <- augment(m3)
qqnorm(r3$.resid)
```



The residuals mostly follow the straight line, though there are slight deviations at the extremes, particularly at the lower end. This indicates that while the residuals are fairly close to a normal distribution, there may be slight deviations from normality at the tails.

Histogram of residuals

```
r3 |>
  ggplot(aes(.resid)) +
    geom_histogram(
      binwidth=0.05,
      color="black",
      fill="white") +
      ggtitle("Plot drawn by Michael Dang on 2023-09-29") +
      xlab("Residuals from m3")
```

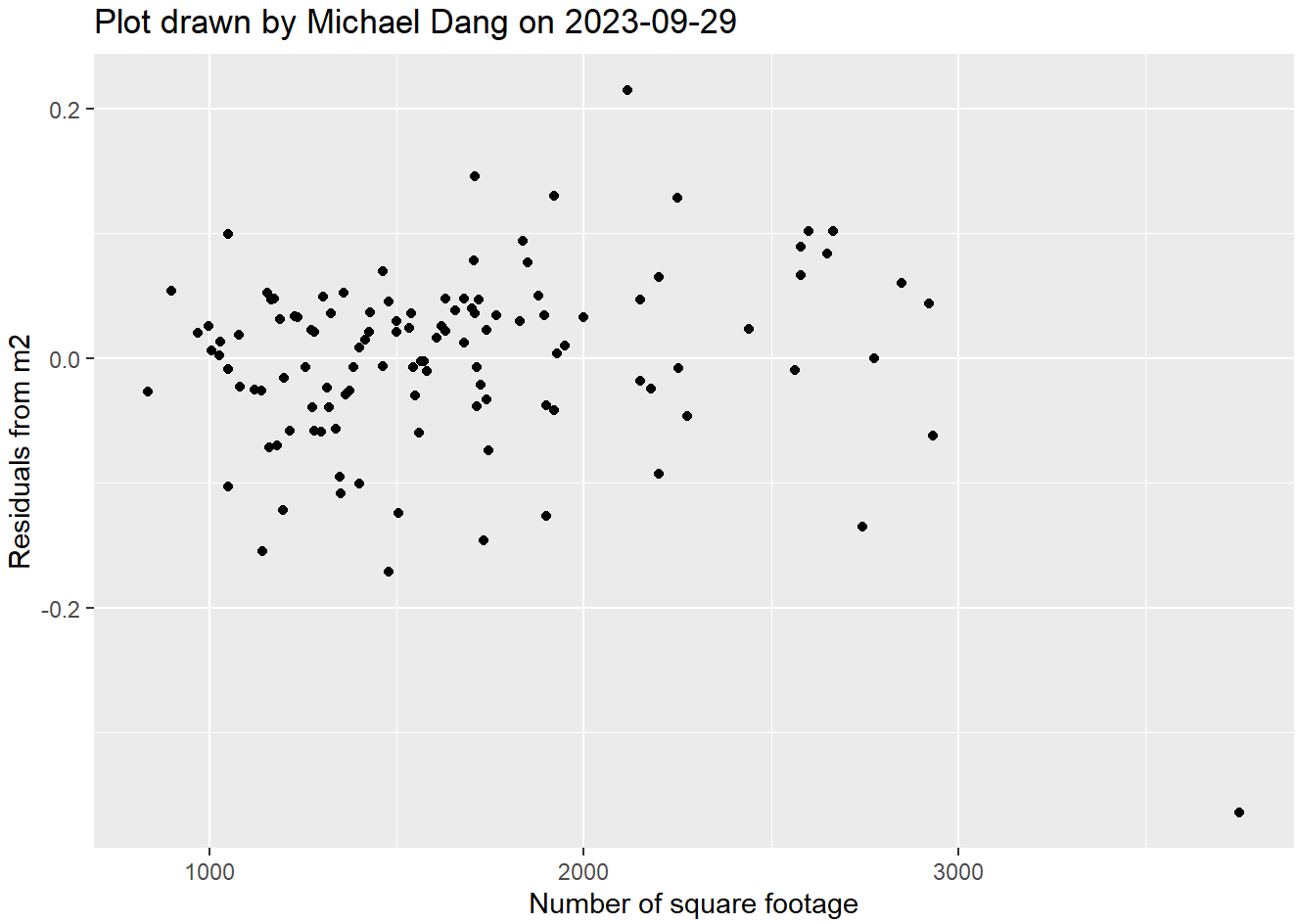


The residuals are fairly symmetric and centered around zero, which suggests a near-normal distribution. There are a few outliers at the extreme ends, but overall, the distribution is more centered than in the original (untransformed) model.

Plot residuals versus feature

```
r3 |>
  ggplot(aes(sqft, .resid)) +
```

```
  geom_point() +
    ggtitle("Plot drawn by Michael Dang on 2023-09-29") +
    xlab("Number of square footage") +
    ylab("Residuals from m2")
```



The residuals appear to be scattered randomly around zero, without a clear pattern or funnel shape. This indicates that heteroscedasticity is not a significant issue, and the variance of the residuals is more consistent compared to the untransformed model.