

# Week 10 - Homework

STAT 420, Fall 2024, Banghao Chi

11/03/2024

---

## Exercise 1 (Writing Functions)

(a) Write a function named `diagnostics` that takes as input the arguments:

- `model`, an object of class `lm()`, that is a model fit via `lm()`
- `pcol`, for controlling point colors in plots, with a default value of `grey`
- `lcol`, for controlling line colors in plots, with a default value of `dodgerblue`
- `alpha`, the significance level of any test that will be performed inside the function, with a default value of 0.05
- `plotit`, a logical value for controlling display of plots with default value `TRUE`
- `testit`, a logical value for controlling outputting the results of tests with default value `TRUE`

The function should output:

- A list with two elements when `testit` is `TRUE`:
  - `p_val`, the p-value for the Shapiro-Wilk test for assessing normality
  - `decision`, the decision made when performing the Shapiro-Wilk test using the `alpha` value input to the function. “Reject” if the null hypothesis is rejected, otherwise “Fail to Reject.”
- Two plots, side-by-side, when `plotit` is `TRUE`:
  - A fitted versus residuals plot that adds a horizontal line at  $y = 0$ , and labels the  $x$ -axis “Fitted” and the  $y$ -axis “Residuals.” The points and line should be colored according to the input arguments. Give the plot a title.
  - A Normal Q-Q plot of the residuals that adds the appropriate line using `qqline()`. The points and line should be colored according to the input arguments. Be sure the plot has a title.

Consider using this function to help with the remainder of the assignment as well.

```
diagnostics = function(model, pcol = "grey", lcol = "dodgerblue",
                        alpha = 0.05, plotit = TRUE, testit = TRUE) {
  resids = residuals(model)
  fitted_vals = fitted(model)

  if(testit) {
    sw_test = shapiro.test(resids)
    output = list(
      p_val = sw_test$p.value,
      decision = ifelse(sw_test$p.value < alpha, "Reject", "Fail to Reject")
    )
  }

  if(plotit) {
    par(mfrow = c(1, 2))
```

```

plot(fitted_vals, resid,
     col = pcol,
     xlab = "Fitted",
     ylab = "Residuals",
     main = "Residuals vs Fitted Values")
abline(h = 0, col = lcol, lwd = 2)

qqnorm(resids,
       col = pcol,
       main = "Normal Q-Q Plot")
qqline(resids, col = lcol, lwd = 2)
}

if(testit) {
  return(output)
}
}

```

(b) Run the following code.

```

set.seed(40)

data_1 = data.frame(x = runif(n = 30, min = 0, max = 10),
                    y = rep(x = 0, times = 30))
data_1$y = with(data_1, 2 + 1 * x + rexp(n = 30))
fit_1 = lm(y ~ x, data = data_1)

data_2 = data.frame(x = runif(n = 20, min = 0, max = 10),
                    y = rep(x = 0, times = 20))
data_2$y = with(data_2, 5 + 2 * x + rnorm(n = 20))
fit_2 = lm(y ~ x, data = data_2)

data_3 = data.frame(x = runif(n = 40, min = 0, max = 10),
                    y = rep(x = 0, times = 40))
data_3$y = with(data_3, 2 + 1 * x + rnorm(n = 40, sd = x))
fit_3 = lm(y ~ x, data = data_3)

diagnostics(fit_1, plotit = FALSE)$p_val

## [1] 0.005613

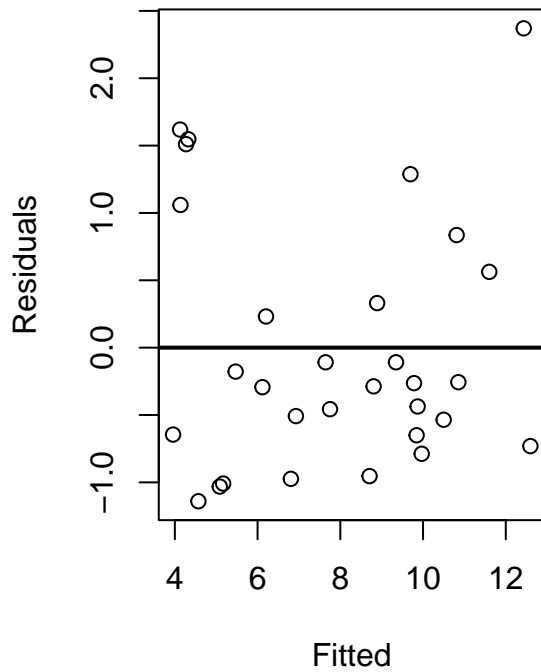
diagnostics(fit_2, plotit = FALSE)$decision

## [1] "Fail to Reject"

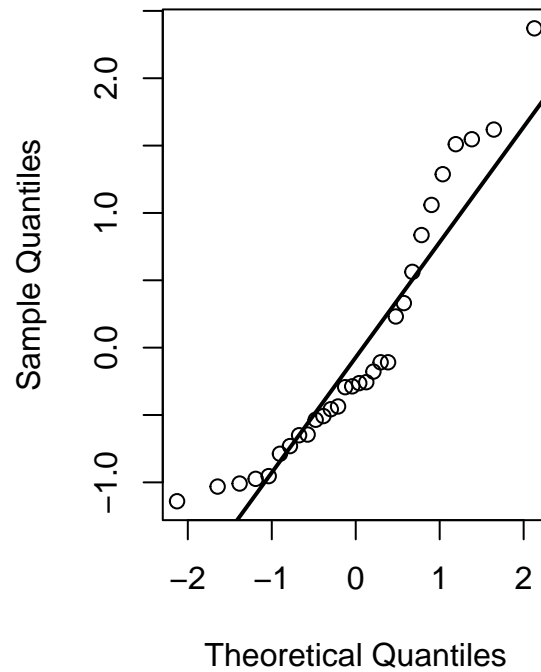
diagnostics(fit_1, testit = FALSE, pcol = "black", lcol = "black")

```

**Residuals vs Fitted Values**

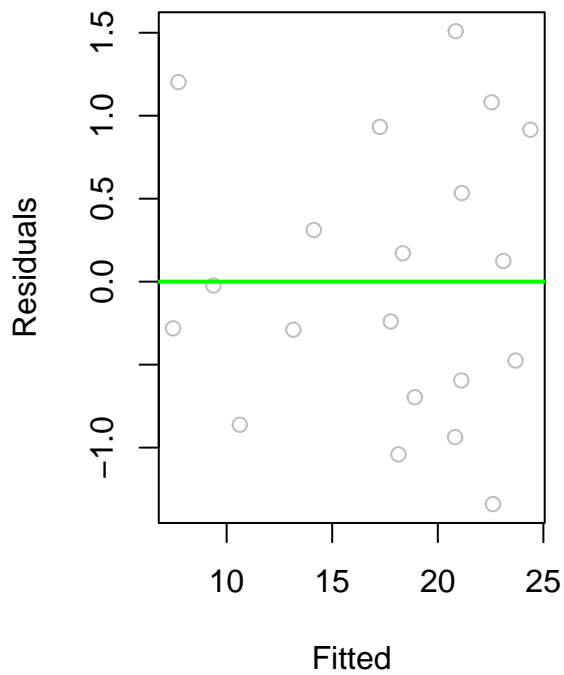


**Normal Q-Q Plot**

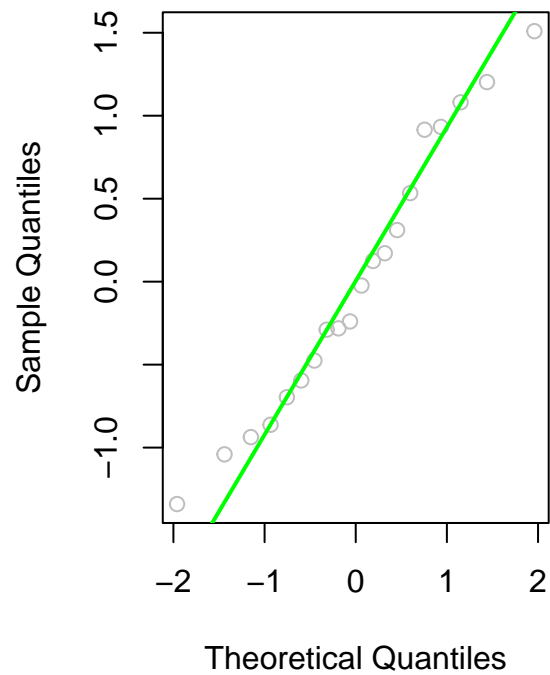


```
diagnostics(fit_2, testit = FALSE, pcol = "grey", lcol = "green")
```

**Residuals vs Fitted Values**

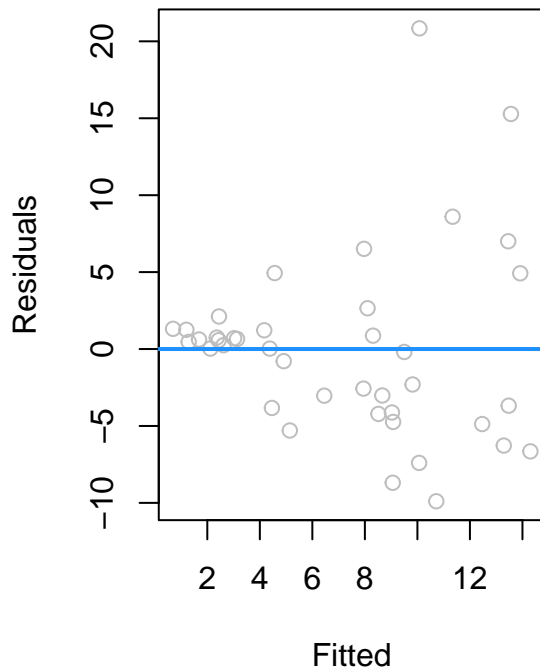


**Normal Q-Q Plot**

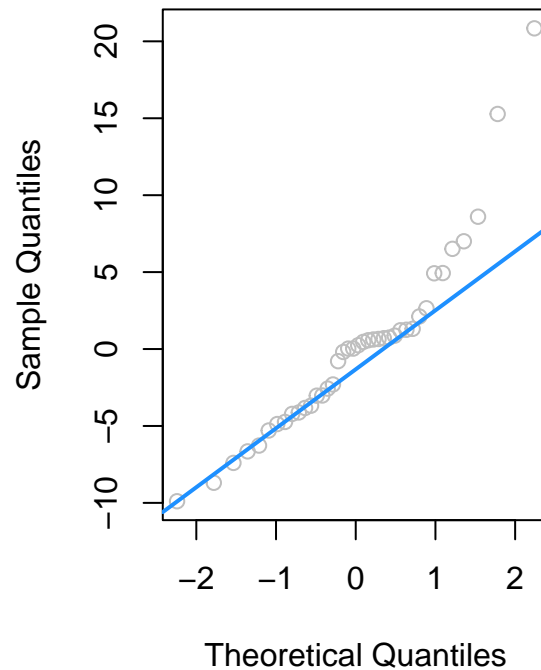


```
diagnostics(fit_3)
```

### Residuals vs Fitted Values



### Normal Q-Q Plot



```
## $p_val
## [1] 0.001608
##
## $decision
## [1] "Reject"
```

---

## Exercise 2 (Prostate Cancer Data)

For this exercise, we will use the `prostate` data, which can be found in the `faraway` package. After loading the `faraway` package, use `?prostate` to learn about this dataset.

```
library(faraway)
```

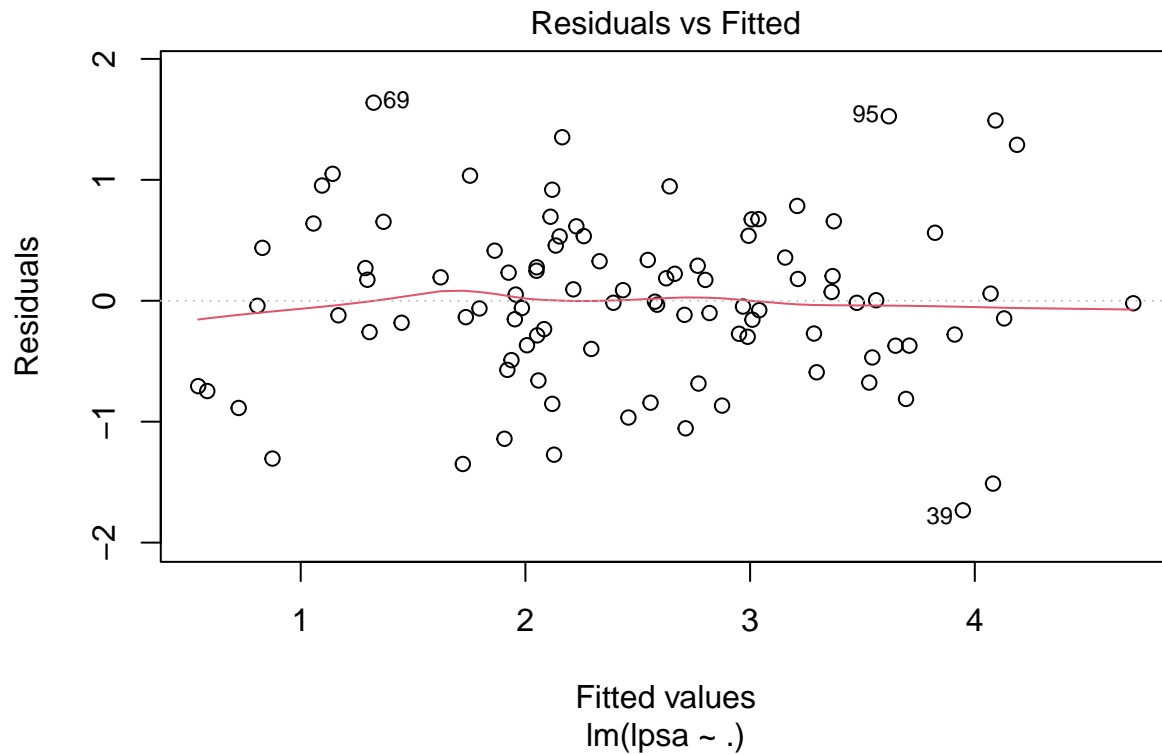
(a) Fit an additive multiple regression model with `lpsa` as the response and the remaining variables in the `prostate` dataset as predictors. Report the  $R^2$  value for this model.

```
model = lm(lpsa ~ ., data = prostate)
r_squared = summary(model)$r.squared
```

The  $R^2$  value for this model is 0.6548.

(b) Check the constant variance assumption for this model. Do you feel it has been violated? Justify your answer.

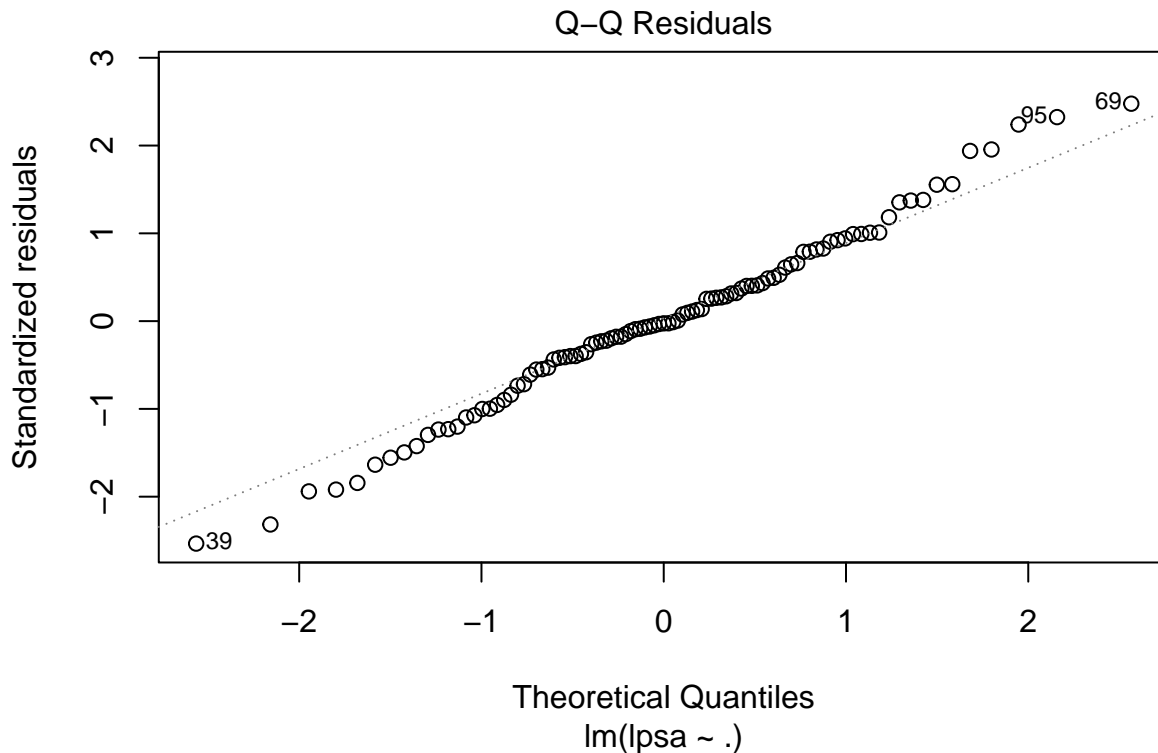
```
plot(model, which = 1)
```



Look at the Fitted vs Residuals plot, I don't think the constant variance assumption for this model is violated, as for any fitted value, the distribution of the residuals seem to be similar (in terms of range and density).

(c) Check the normality assumption for this model. Do you feel it has been violated? Justify your answer.

```
plot(model, which = 2)
```



Look at the Normal Q-Q plot, I think the normality assumption for this model is violated, there seems to be fat tails within the plot.

(d) Check for any high leverage observations. Report any observations you determine to have high leverage.

```
n = nrow(prostate)
p = length(coef(model))
leverage = hatvalues(model)
high_leverage_cutoff = 2 * (p/n)
high_leverage = which(leverage > high_leverage_cutoff)
```

(e) Check for any influential observations. Report any observations you determine to be influential.

(f) Refit the additive multiple regression model without any points you identified as influential. Compare the coefficients of this fitted model to the previously fitted model.

(g) Create a data frame that stores the observations that were “removed” because they were influential. Use the two models you have fit to make predictions with these observations. Comment on the difference between these two sets of predictions.

---

### Exercise 3 (Why Bother?)

**Why** do we care about violations of assumptions? One key reason is that the distributions of the parameter estimators that we have used are all reliant on these assumptions. When the assumptions are violated, the distributional results are not correct, so our tests are garbage. **Garbage In, Garbage Out!**

Consider the following setup that we will use for the remainder of the exercise. We choose a sample size of 50.

```
n = 50
set.seed(420)
x_1 = runif(n, 0, 5)
x_2 = runif(n, -2, 2)
```

Consider the model,

$$Y = 4 + 1x_1 + 0x_2 + \epsilon.$$

That is,

- $\beta_0 = 4$
- $\beta_1 = 1$
- $\beta_2 = 0$

We now simulate  $y_1$  in a manner that does **not** violate any assumptions, which we will verify. In this case  $\epsilon \sim N(0, 1)$ .

```
set.seed(83)
library(lmtest)

## Loading required package: zoo
##
## Attaching package: 'zoo'
## The following objects are masked from 'package:base':
##
##      as.Date, as.Date.numeric
y_1 = 4 + 1 * x_1 + 0 * x_2 + rnorm(n = n, mean = 0, sd = 1)
fit_1 = lm(y_1 ~ x_1 + x_2)
bptest(fit_1)
```

```
##
## studentized Breusch-Pagan test
##
## data: fit_1
## BP = 4.4, df = 2, p-value = 0.1
```

Then, we simulate  $y_2$  in a manner that **does** violate assumptions, which we again verify. In this case  $\epsilon \sim N(0, \sigma = |x_2|)$ .

```
set.seed(83)
y_2 = 4 + 1 * x_1 + 0 * x_2 + rnorm(n = n, mean = 0, sd = abs(x_2))
fit_2 = lm(y_2 ~ x_1 + x_2)
bptest(fit_2)
```

```
##
## studentized Breusch-Pagan test
##
## data: fit_2
## BP = 4.9, df = 2, p-value = 0.08
```

(a) Use the following code after changing `birthday` to your birthday.

```
num_sims = 2500
p_val_1 = rep(0, num_sims)
p_val_2 = rep(0, num_sims)
birthday = 19081014
set.seed(birthday)
```

Repeat the above process of generating  $y_1$  and  $y_2$  as defined above, and fit models with each as the response 2500 times. Each time, store the p-value for testing,

$$\beta_2 = 0,$$

using both models, in the appropriate variables defined above. (You do not need to use a data frame as we have in the past. Although, feel free to modify the code to instead use a data frame.)

(b) What proportion of the `p_val_1` values is less than 0.01? Less than 0.05? Less than 0.10? What proportion of the `p_val_2` values is less than 0.01? Less than 0.05? Less than 0.10? Arrange your results in a table. Briefly explain these results.

---

## Exercise 4 (Corrosion Data)

For this exercise, we will use the `corrosion` data, which can be found in the `faraway` package. After loading the `faraway` package, use `?corrosion` to learn about this dataset.

```
library(faraway)
```

(a) Fit a simple linear regression with `loss` as the response and `Fe` as the predictor. Plot a scatterplot and add the fitted line. Check the assumptions of this model.

(b) Fit higher order polynomial models of degree 2, 3, and 4. For each, plot a fitted versus residuals plot and comment on the constant variance assumption. Based on those plots, which of these three models do you think are acceptable? Use a statistical test(s) to compare the models you just chose. Based on the test, which is preferred? Check the normality assumption of this model. Identify any influential observations of this model.

---

## Exercise 5 (Diamonds)

The data set `diamonds` from the `ggplot2` package contains prices and characteristics of 54,000 diamonds. For this exercise, use `price` as the response variable  $y$ , and `carat` as the predictor  $x$ . Use `?diamonds` to learn more.

```
library(ggplot2)
```

(a) Fit a linear model with `price` as the response variable  $y$ , and `carat` as the predictor  $x$ . Return the summary information of this model.

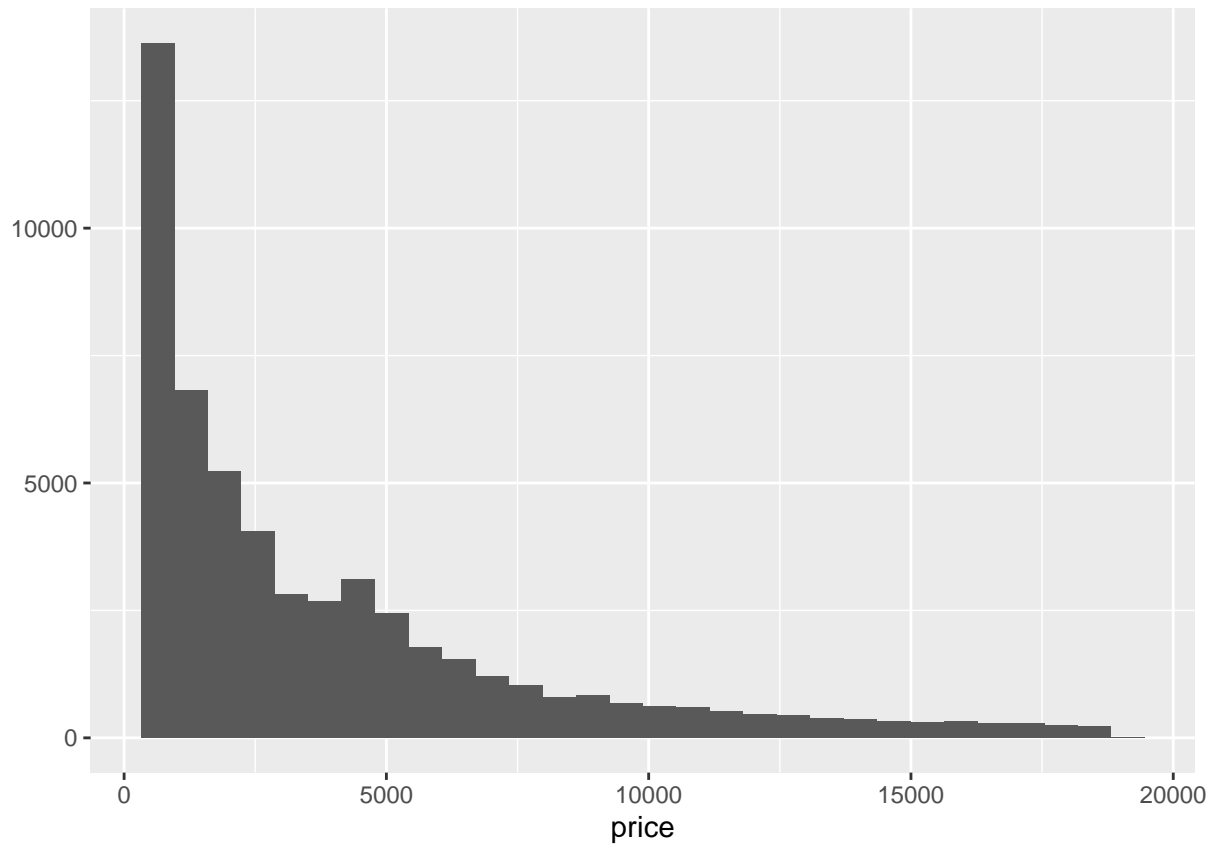
(b) Plot a scatterplot of price versus carat and add the line for the fitted model in part (a). Using a fitted versus residuals plot and/or a Q-Q plot, comment on the diagnostics.

(c) Seeing as the price stretches over several orders of magnitude, it seems reasonable to try a log transformation of the response. Fit a model with a logged response, plot a scatterplot of log-price versus carat and add the line for the fitted model, then use a fitted versus residuals plot and/or a Q-Q plot to comment on the diagnostics of the model.

```
qplot(price, data = diamonds, bins = 30)
```

```
## Warning: `qplot()` was deprecated in ggplot2 3.4.0.
## This warning is displayed once every 8 hours.
## Call `lifecycle::last_lifecycle_warnings()` to see where this warning was
## generated.
```





(d) Try adding log transformation of the predictor. Fit a model with a logged response and logged predictor, plot a scatterplot of log-price versus log-carat and add the line for the fitted model, then use a fitted versus residuals plot and/or a Q-Q plot to comment on the diagnostics of the model.

(e) Use the model from part (d) to predict the price (in dollars) of a 3-carat diamond. Construct a 99% prediction interval for the price (in dollars).