# Answer Sketch and Rubric for Lab 03

431 Staff and Professor Love

Last Edited 2020-09-17 00:30:41

## Contents

0.1 R Setup . . . 1
0.2 An Introduction . . . 2

**1 Question 1** **3**
1.1 Comments . . . 3

**2 Question 2** **4**
2.1 Comments . . . 5

**3 Question 3** **5**
3.1 Comments . . . 5

**4 Question 4** **6**
4.1 Comments . . . 6
4.2 On Assessing Skew . . . 8
4.3 Z Scores for Most Outlying Values . . . 9
4.4 Histogram vs. Expectation under a Normal Distribution . . . 9
4.5 Approach from Slides (Classes 7 and 8) . . . 12

**5 Question 5** **13**
5.1 The Empirical Rule . . . 13

**6 Question 6** **14**
6.1 On Correlation . . . 15
6.2 On Rounding . . . 15

**7 Question 7** **15**

**8 Questions 8 and 9** **16**
8.1 An Alternative Model for the `faithful` Data . . . 18

**9 Question 10** **19**

**10 On Grading Lab 03** **19**

**11 Session Information** **20**

### 0.1 R Setup

Here's the complete R setup we used to build this answer sketch.

```
knitr::opts_chunk$set(comment=NA)
options(width = 60)

library(MASS)
library(broom)
library(knitr)
library(magrittr)
library(patchwork)
library(tidyverse)
## make sure these packages are installed in R

theme_set(theme_light())
```

## 0.2 An Introduction

This answer sketch borrows liberally from a case study entitled *Eruptions of the Old Faithful Geyser* from Chatterjee S Handcock MS Simonoff JS *A Casebook for a First Course in Statistics and Data Analysis* Wiley, 1995.

---

A geyser is a hot spring that occasionally becomes unstable and erupts hot water and steam into the air. The Old Faithful geyser at Yellowstone National Park in Wyoming is probably the most famous geyser in the world. Visitors to the park try to arrive at the geyser site to see it erupt without having to wait too long; the name of the geyser comes from the fact that eruptions follow a relatively stable pattern. The National Park Service web site which streams a live feed of the geyser includes a time frame during which the next eruption is predicted to occur. Thus, it is of interest to understand and predict the interval time until the next eruption. The main part of this assignment considers the `faithful` data frame, which describes eruption durations and waiting times for the Old Faithful geyser.

```
lab03 <- tibble(faithful)

summary(lab03)
```

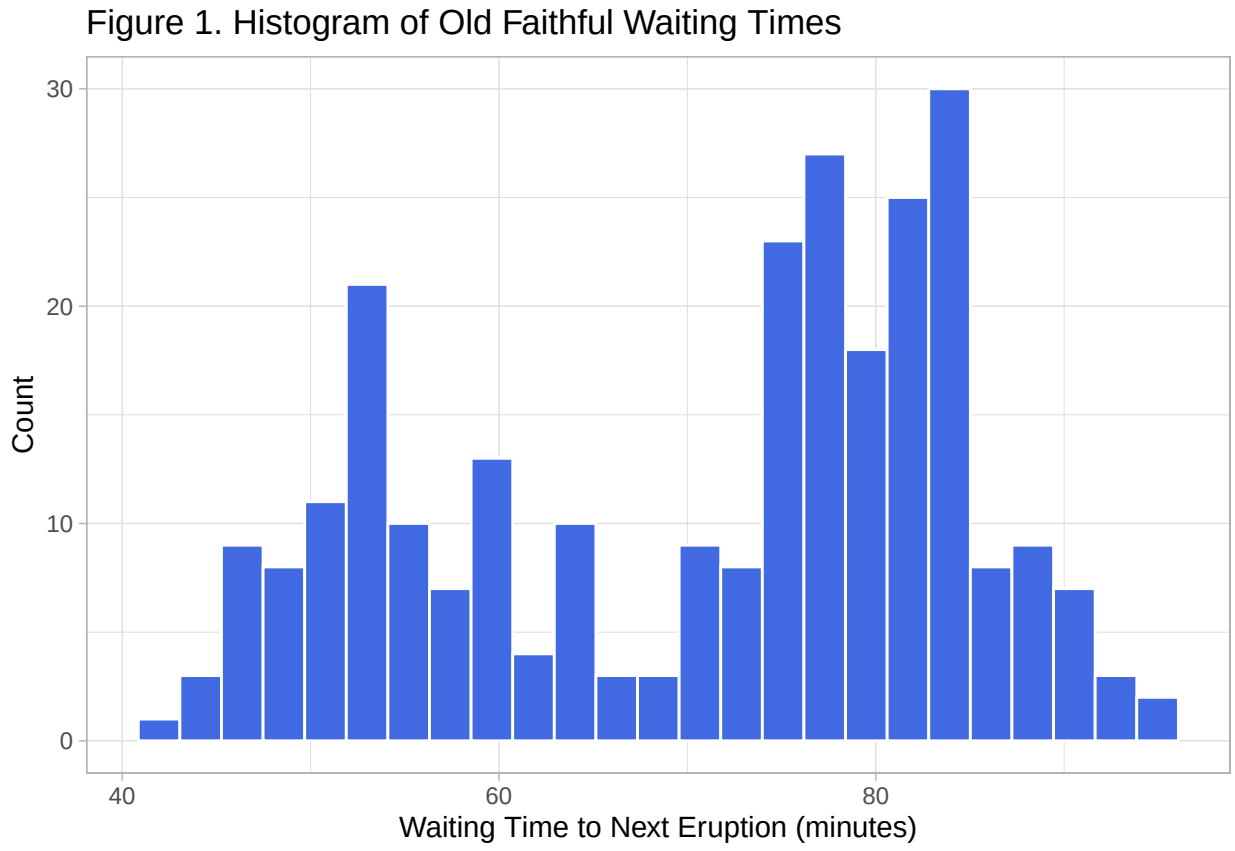
```
   eruptions        waiting
 Min.   :1.600   Min.   :43.0
 1st Qu.:2.163   1st Qu.:58.0
 Median :4.000   Median :76.0
 Mean   :3.488   Mean   :70.9
 3rd Qu.:4.454   3rd Qu.:82.0
 Max.   :5.100   Max.   :96.0
```

# 1 Question 1

> Plot a histogram or other summary plot which meaningfully describes the distribution of the waiting times. Be sure it is very clearly labeled.

The first step in any data analysis is simply to look at the data. A histogram gives a good deal of information about the distribution of eruption times, suggesting some interesting structure. Interval times are in the general range of 40 to 100 minutes, but there are apparently two subgroups in the data, centered at roughly 55 minutes, and 80 minutes, respectively, with a gap in the middle.

```
ggplot(lab03, aes(x = waiting)) +
  geom_histogram(bins = 25, fill = "royalblue", color = "white") +
  labs(title = "Figure 1. Histogram of Old Faithful Waiting Times",
       x = "Waiting Time to Next Eruption (minutes)", y = "Count")
```




## 1.1 Comments

This relatively simple histogram is just one of many possible plots we could use to describe the center, spread and shape of a distribution of data.

- We might consider a **stem-and-leaf display** to show the actual data values while retaining the shape of a histogram.

```
lab03 %$% stem(waiting)
```

```
  The decimal point is 1 digit(s) to the right of the |
```

```
  4 | 3
  4 | 55566666777788899999
  5 | 00000111111222223333333444444444
  5 | 555555666677788889999999
  6 | 00000022223334444
  6 | 555667899
  7 | 00001111123333333444444
  7 | 55555555666666666777777777777888888888888888899999999999
  8 | 00000000111111111111222222222223333333333333334444444444
  8 | 55555566666677888888999
  9 | 00000012334
  9 | 6
```

- We might consider a **boxplot** or **box-and-whiskers plot** (as we'll see later), or perhaps a **violin plot**.
- If we wanted to compare the distribution of the data to what we might expect from a Normal distribution, we might develop a histogram with an overplotted Normal density function or a **Normal Q-Q plot** to facilitate such a comparison.

# 2 Question 2

> What appears to be a typical waiting time? Compare the mean, median and 80% trimmed mean (mean of the middle 80% of the observed waiting times.)

As noted previously, the waiting times appear to cluster into two groups: one centered around 55 minutes, and another, larger, group centered near 80 minutes.

The `summary` function in R provides the five-number summary (minimum, 25th, 50th [median] and 75th percentiles, maximum) and the mean, so that gets us two of our three needed summaries. To get the third, we can either use the `describe` function from the `psych` library, or we can calculate the trimmed mean using the `mean` function.

```
# summary provides mean and median, along with quartiles and min/max
lab03 %$% summary(waiting)
```

```
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
   43.0    58.0    76.0    70.9    82.0    96.0
```

```
# describe in the psych library also provides the trimmed mean we're looking for...
lab03 %$% psych::describe(waiting)
```

```
   vars   n mean    sd median trimmed   mad min max range
X1    1 272 70.9 13.59     76    71.5 11.86  43  96    53
    skew kurtosis   se
X1 -0.41    -1.16 0.82
```

```
# this trims 10% from the top and 10% of the bottom
# of the distribution, and then takes the mean of
# what remains, just as psych::describe does
lab03 %$% mean(waiting, trim = 0.1)
```

```
[1] 71.49541
```

## 2.1 Comments

In addition to writing full code chunks, we use Markdown to ask R to fill in the values as we go, rather than inserting them through copy and paste, or retyping. This substantially reduces the chance of errors, and lets us generate a revised document quickly if we find an error in the data.

Look at the Markdown file for this assignment to see, for instance, how we are using code to fill in the values in the next bullet.

- The distribution of 272 waiting times has mean 70.9 minutes, and median 76 minutes, with a trimmed mean of 71.5 minutes.
- Note that `signif` (which is used in the code to generate the previous sentence) is a function which rounds to the specified number of "significant figures" (digits). This has nothing to do with the notion of statistical *significance*.

# 3 Question 3

What is the inter-quartile range, and how does it compare to the standard deviation?

We can obtain the needed summaries directly with the `favstats` function from the `mosaic` package.

```
mosaic::favstats(~ waiting, data = lab03)
```

```
Registered S3 method overwritten by 'mosaic':
  method                           from
  fortify.SpatialPolygonsDataFrame ggplot2
```

```
 min Q1 median Q3 max     mean       sd   n missing
  43 58     76 82  96 70.89706 13.59497 272       0
```

or we can calculate the two statistics like this:

```
lab03 %>% summarize(IQR(waiting), sd(waiting))
```

```
# A tibble: 1 x 2
  `IQR(waiting)` `sd(waiting)`
           <dbl>         <dbl>
1             24          13.6
```

The 25th percentile is 58 and the 75th percentile is 82 so the inter-quartile range is 24 minutes, which is considerably larger than the standard deviation of 13.6 minutes. Specifically, the IQR is about 77% larger than the SD, since $\frac{IQR}{SD} = 24 \ / \ 13.6 = 1.77$.

## 3.1 Comments

- The **range** of the data is just the maximum minus the minimum, or 96 minus 43 or 53. Note that if you ask R for the `range` of the `lab03$waiting` variable with `range(lab03$waiting)`, this yields a vector with two values: the minimum and the maximum, for example 43, 96.
- If the data were Normally distributed, we would expect that about 68% of observations would fall within one standard deviation of the mean. For any distribution, the middle half of the distribution falls within the first and third quartiles. If the data followed a Normal distribution very closely, the IQR would be 25-50% larger than the standard deviation.

- The **median absolute deviation**, or MAD, is another candidate measure of dispersion or scale, which has a more direct relationship with the standard deviation (the population standard deviation is well estimated by the MAD for Normally distributed data).
  - The MAD is defined as the median of the absolute deviations of each observation from the data's median, multiplied by a constant (1.48 by default in R).
  - In this case, the MAD for the waiting times is 11.86 minutes, so the ratio of the standard deviation to the MAD for the Old Faithful waiting times is 13.59 / 11.86 = 1.15.

```
lab03 %>% summarize(mad(waiting))
```

```
# A tibble: 1 x 1
  `mad(waiting)`
           <dbl>
1           11.9
```

# 4 Question 4

Is the distribution multi-modal or unimodal? How do you know?

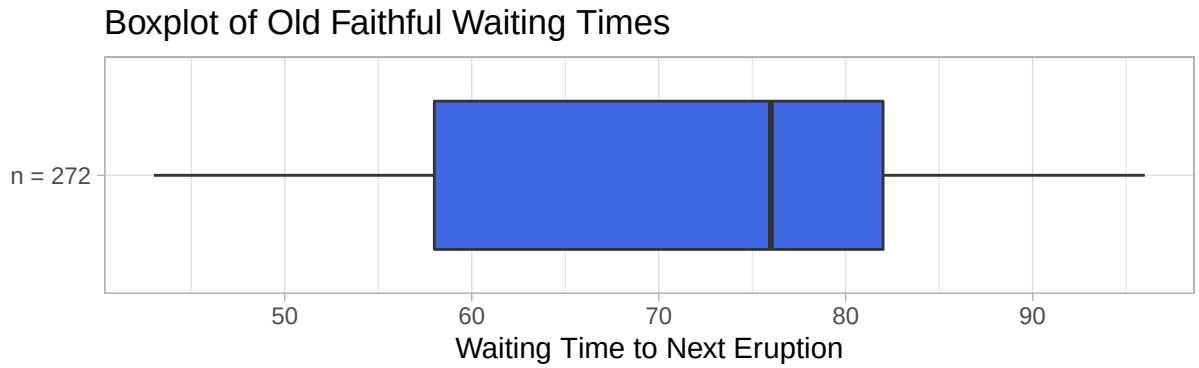
The distribution clearly has one cluster of waiting times centered at 50-55 minutes and another, larger, cluster centered at 80 minutes. The fact that the distribution has multiple local maxima would usually suggest that we interpret this as multi-modal (specifically, because there are two local maxima, we'd say bimodal) data, where a single summary of the center might not be as useful as it would be with unimodal data.

## 4.1 Comments

Not all exploratory techniques are equally effective for these data. A **boxplot** shows that the waiting times are in the general range of 40-100 minutes, but the bimodal distribution is hidden by the form of the plot. Boxplots are mostly used to make comparisons.

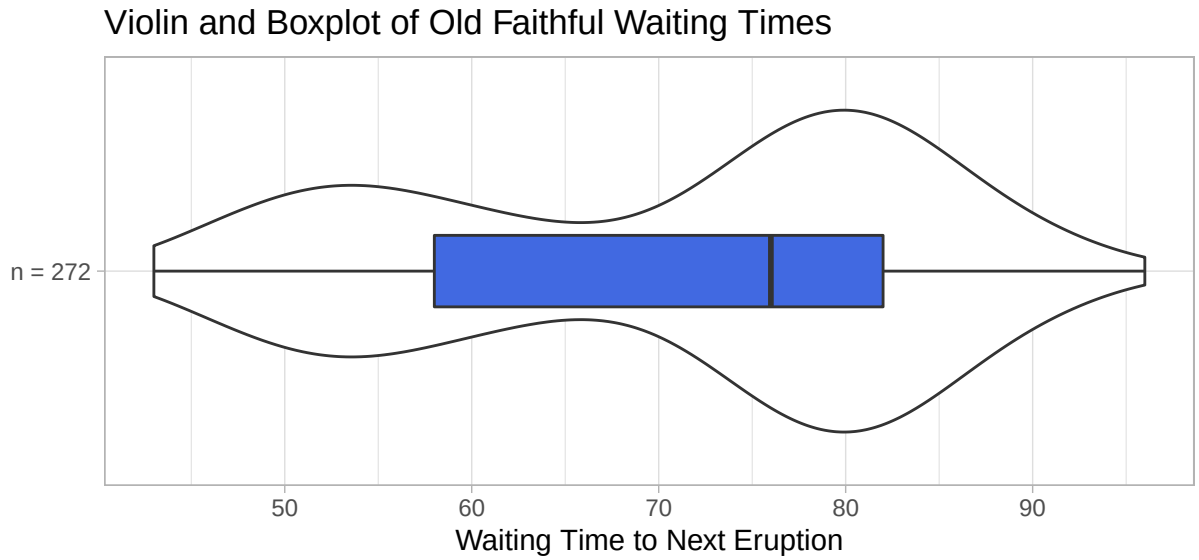
In the notes, we described one way to get a boxplot for a single distribution. That was as follows.

```
ggplot(lab03, aes(x = "n = 272", y = waiting)) +
  geom_boxplot(fill = "royalblue") +
  coord_flip() +
  labs(title = "Boxplot of Old Faithful Waiting Times",
       x = "", y = "Waiting Time to Next Eruption")
```



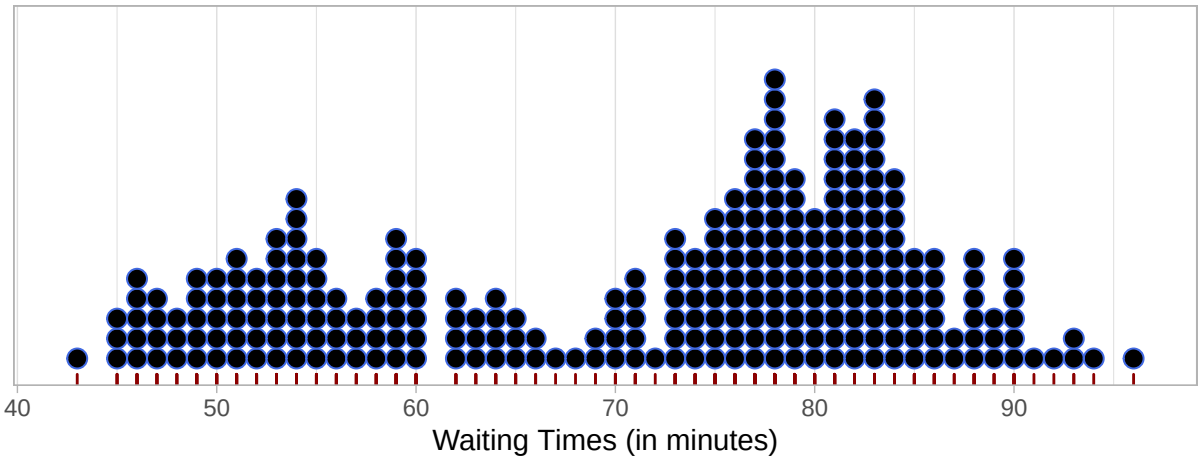

If we wanted to, we could first create a violin plot then add the boxplot:

```
ggplot(lab03, aes(x = "n = 272", y = waiting)) +
  geom_violin() +
  geom_boxplot(width = 0.2, fill = "royalblue") +
  coord_flip() +
  theme_light() +
  labs(title = "Violin and Boxplot of Old Faithful Waiting Times",
       x = "", y = "Waiting Time to Next Eruption")
```




A **dotplot** might help here. Though there are other ways to generate these plots, we like the following approach, which creates a dot plot augmented by a rug plot.

```
ggplot(lab03, aes(x=waiting)) +
  geom_dotplot(binwidth=1, col = "royalblue") +      ## create dot plot
  geom_rug(col = "darkred") +  ## add rug
  scale_y_continuous(breaks=NULL) + ## Remove ticks
  labs(x = "Waiting Times (in minutes)", y = "")
```




> Is the distribution skewed (and if so, in which direction) or is it essentially symmetric? How do you know?

It's a little challenging to see this in a histogram, because of the multiple modes in the data, but I'd call this

a left-skewed distribution, in part because the mean is substantially smaller than the median.

```
lab03 %>%
  summarize(mean(waiting), median(waiting), sd(waiting))
```

```
# A tibble: 1 x 3
  `mean(waiting)` `median(waiting)` `sd(waiting)`
            <dbl>             <dbl>         <dbl>
1            70.9                76          13.6
```

## 4.2 On Assessing Skew

A reasonable measure of skewness or asymmetry in a distribution, sometimes called skew1 or *non-parametric skew*, compares the mean to the median, while using the standard deviation as the unit of measurement.

Skewness is a far more meaningful concept with unimodal data than with multi-modal data like this. We can declare the skew to be positive or negative regardless of whether the data are in fact multi-modal or follow any other particular pattern.

The formula is

$$skew_1 = \frac{mean - median}{SD}$$

where:

- A positive skew1 value indicates right (sometimes called positive) skew where the mean exceeds the median, and
- a negative skew1 value indicates left skew, where the mean is less than the median.
- skew1 = 0 when the mean is equal to the median, an indication of potential symmetry.
- skew1 values exceeding 0.2 in absolute value are sometimes taken to indicate fairly substantial skew (far enough from a Normal distribution to call into question whether the mean and standard deviation alone are sufficient to approximate the data well).
- If skew1 exceeds 0.5 in absolute value, that indicates very strong skew.

In our data, we can calculate skew1 with, for instance:

```
lab03 %>%
  summarize(skew1 = (mean(waiting) - median(waiting))/sd(waiting))
```

```
# A tibble: 1 x 1
   skew1
   <dbl>
1 -0.375
```

Generally, if the mean is more than 20% of a standard deviation away from the median, I would expect a graph of the data to show substantial skew. If the mean is within 20% of a standard deviation of the median, I wouldn't necessarily expect the data to look meaningfully asymmetric.

For the waiting times, skew1 is -0.38. So the mean is about 38% of a standard deviation below the median, indicating fairly substantial left skew.

> Are there any unusual (outlier) values in the distribution, and if so, what are they?

No. For instance, a boxplot of the waiting times (see above) shows no outliers in the distribution.

The boxplot identifies as an outlier any point that is more than 1.5 IQR outside of the middle half of the data. We define the **inner fences** as falling at Q1 - 1.5 IQR and Q3 + 1.5 IQR, and any points outside those fences are identified by the boxplot and considered to be, at the least, outlier candidates. Sometimes we'll define more serious outliers using a tougher standard. We define the **outer fences** as falling at Q1 - 3 IQR and Q3 + 3 IQR, so that any points outside the outer fences are then described as serious outliers.

## 4.3 Z Scores for Most Outlying Values

Another approach to assessing how outlier-prone the data appear to be, in comparison to what we might expect from a Normal distribution, is to calculate the maximum (and minimum) Z scores for the data set.

The Z score for any particular observation X is (X - mean) / SD , so that our skew1 measure, for instance, may be interpreted as the negative of the Z score for the median. If the data were really drawn from a Normal distribution, then we'd expect:

- roughly 10% of observations to have a Z score greater than 1.645 in absolute value.
- roughly 5% of observations to have a Z score greater than 1.96 in absolute value.
- roughly 1% of observations to have a Z score greater than 2.57 in absolute value.
- less than 3 in 1,000 observations to have a Z score greater than 3 in absolute value.
- less than 1 in 10,000 observations to have a Z score greater than 4 in absolute value.

In this case, the maximum observed waiting time was 96, which has a Z score of 1.85.

How do I know this? Well. . .

```
lab03 %>% summarize(max(waiting), z = (max(waiting) - mean(waiting)) / sd(waiting))
```

```
# A tibble: 1 x 2
  `max(waiting)`     z
           <dbl> <dbl>
1             96  1.85
```

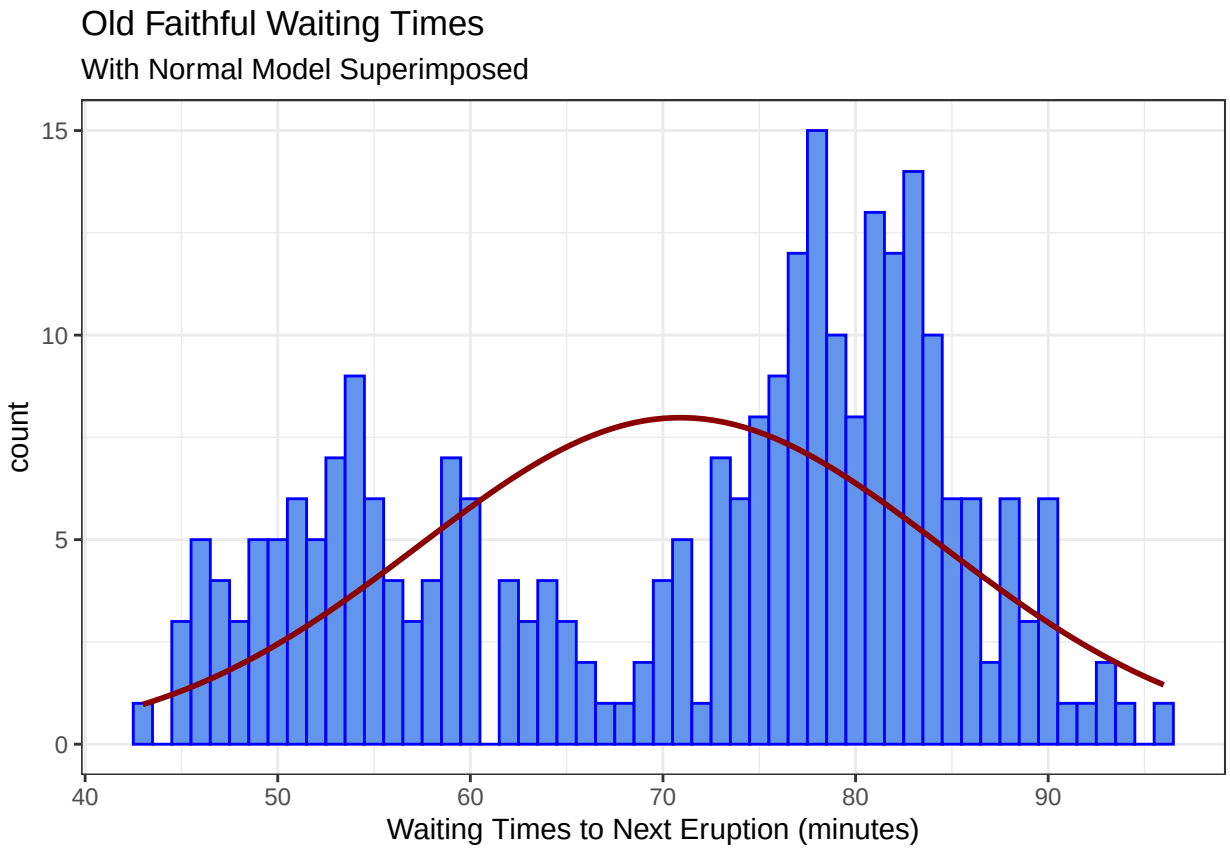
The minimum observed time was 43, which has a Z score of -2.05. With a sample of size 272 these particular values seem to suggest that the data is somewhat **less** outlier-prone than we might expect from a Normal distribution. This is also referred to as the distribution having **lighter tails** than the Normal distribution. But, in essence, we already know that the data don't follow a Normal distribution from our graphs.

## 4.4 Histogram vs. Expectation under a Normal Distribution

Sometimes, it's easier to see *light-tailed* (fewer outlying values than we'd expect from a Normal distribution) vs. *heavy-tailed* (more outliers than a Normal) distributions by directly comparing the histogram to the Normal distribution with the data's mean and standard deviation

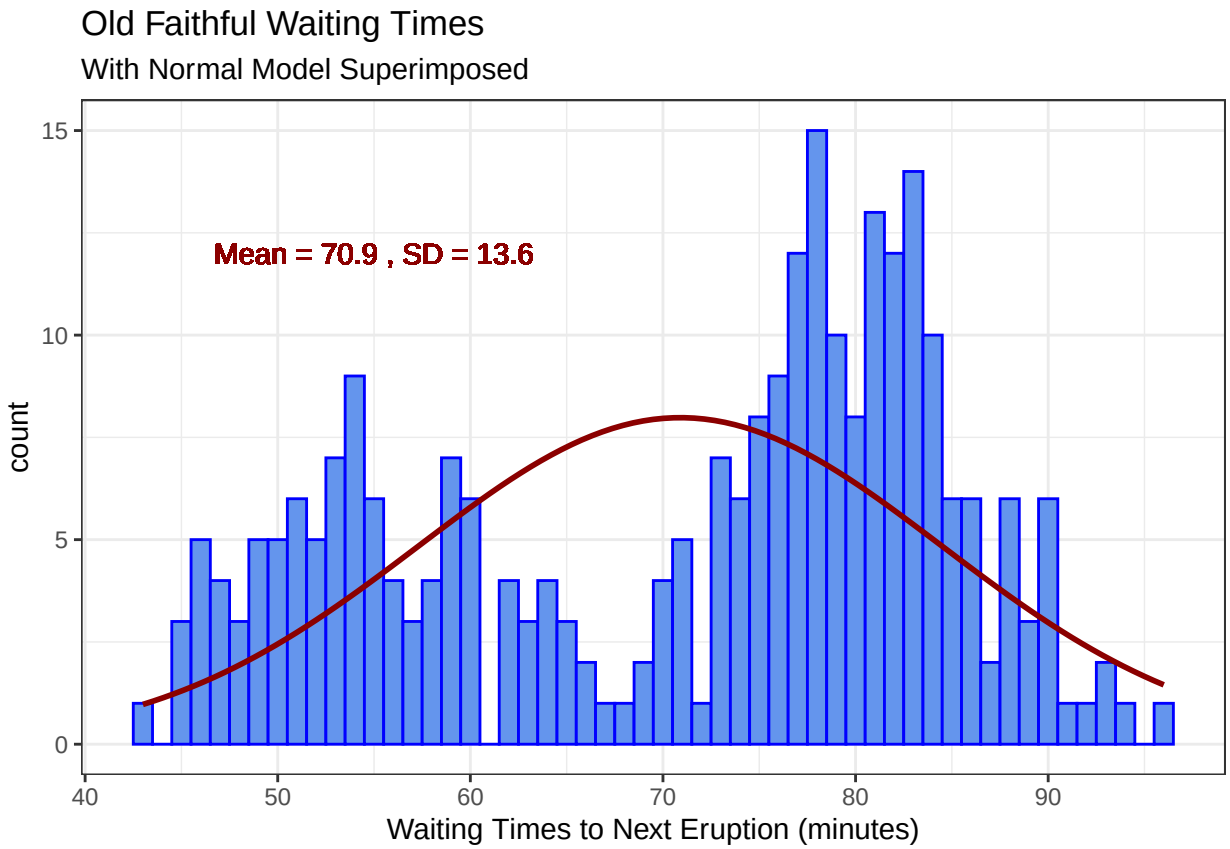
Here's a reasonably clean approach to doing this, that uses the summaries from `favstats` and a pre-specified binwidth for the histogram to generate the plot we want using `geom_histogram` and a function based on `dnorm`.

```
res <- mosaic::favstats(~ waiting, data = lab03) # save summaries
bin_w <- 1 # specify binwidth

ggplot(lab03, aes(x = waiting)) +
  geom_histogram(binwidth = bin_w, fill = "cornflowerblue",
                 col = "blue") +
  theme_bw() +
  stat_function(
    fun = function(x) dnorm(x, mean = res$mean,
                            sd = res$sd) * res$n * bin_w,
    col = "darkred", size = 1) +
labs(title = "Old Faithful Waiting Times",
       subtitle = "With Normal Model Superimposed",
       x = "Waiting Times to Next Eruption (minutes)")
```



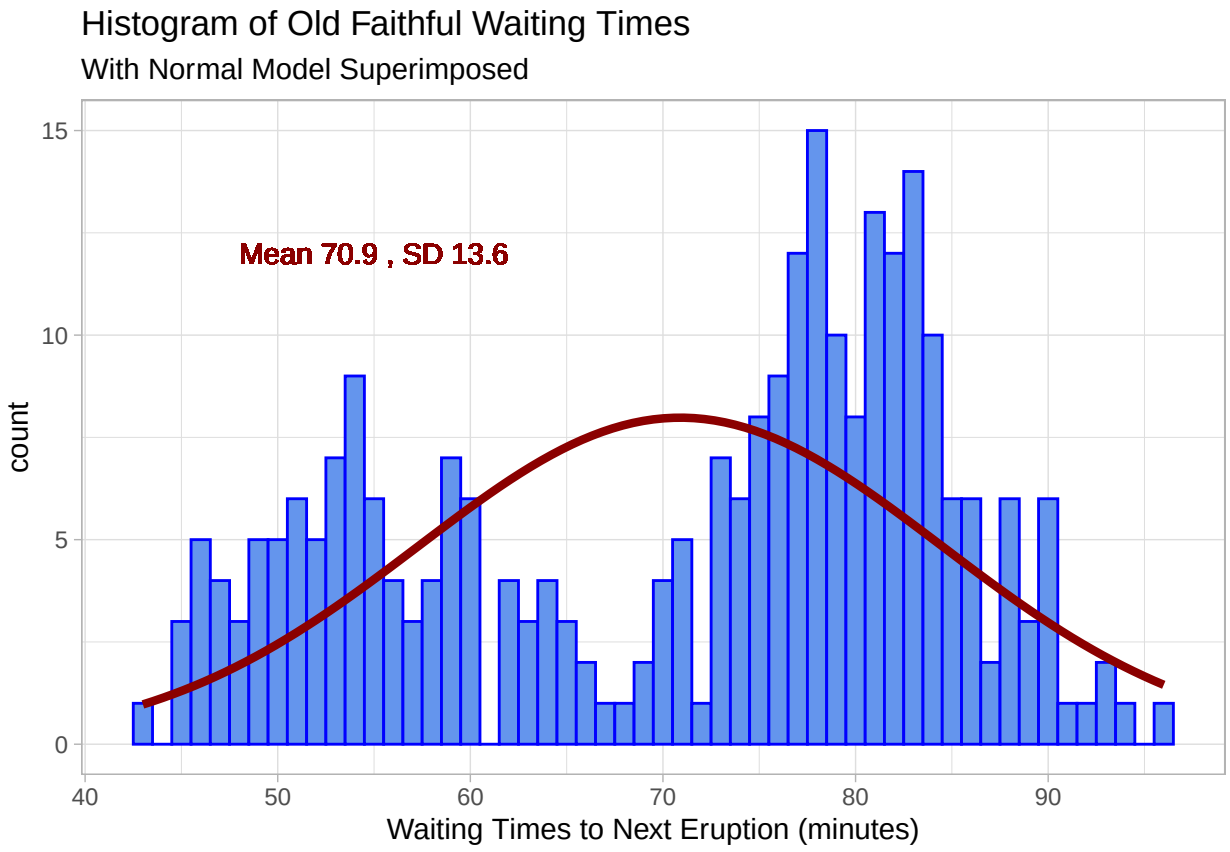

We could also use `geom_text` to add a label that includes the mean and standard deviation of the data.

```
res <- mosaic::favstats(~ waiting, data = lab03) # save summaries
bin_w <- 1 # specify binwidth

ggplot(lab03, aes(x = waiting)) +
  geom_histogram(binwidth = bin_w, fill = "cornflowerblue",
                 col = "blue") +
  theme_bw() +
  stat_function(
    fun = function(x) dnorm(x, mean = res$mean,
                            sd = res$sd) * res$n * bin_w,
    col = "darkred", size = 1) +
    geom_text(aes(
      label = paste("Mean =", round(res$mean,1),
                    ", SD =", round(res$sd,1))),
      x = 55, y = 12, color="darkred") +
labs(title = "Old Faithful Waiting Times",
       subtitle = "With Normal Model Superimposed",
       x = "Waiting Times to Next Eruption (minutes)")
```




You can also do this, if you like, without pre-summarizing the data:

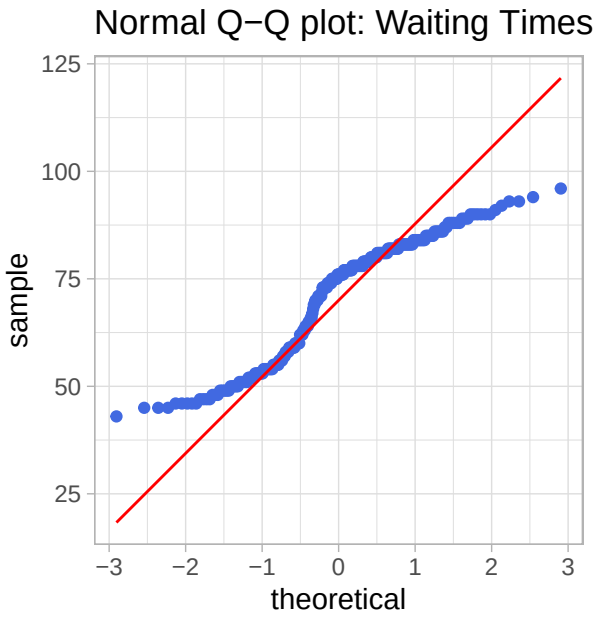
```
## ggplot including histogram of waiting times
## with Normal model superimposed

ggplot(lab03, aes(x = waiting)) +
  geom_histogram(binwidth = 1, fill = "cornflowerblue",
                 col = "blue") +
  stat_function(fun = function(x, mean, sd, n)
    n * dnorm(x = x, mean = mean, sd = sd),
    args = with(lab03,
                c(mean = mean(waiting),
                  sd = sd(waiting),
                  n = length(waiting))),
    col = "darkred", lwd = 1.5) +
    geom_text(aes(label = paste("Mean", round(mean(waiting),1),
                                ", SD", round(sd(waiting),1))),
              x = 55, y = 12, color="darkred") +
  labs(title = "Histogram of Old Faithful Waiting Times",
       subtitle = "With Normal Model Superimposed",
       x = "Waiting Times to Next Eruption (minutes)")
```

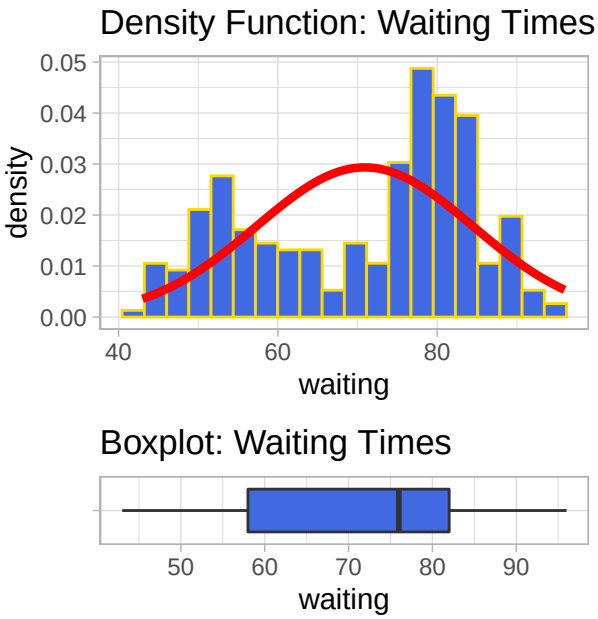



## 4.5 Approach from Slides (Classes 7 and 8)

In the slides for Classes 7 and 8, we proposed another way to explore a distribution, using several plots, plus the `favstats` summary. Here's what that would look like here.

```
p1 <- ggplot(lab03, aes(sample = waiting)) +
  geom_qq(col = "royalblue") + geom_qq_line(col = "red") +
  theme(aspect.ratio = 1) +
  labs(title = "Normal Q-Q plot: Waiting Times")

p2 <- ggplot(lab03, aes(x = waiting)) +
  geom_histogram(aes(y = stat(density)),
                 bins = 20, fill = "royalblue", col = "gold") +
  stat_function(fun = dnorm,
                args = list(mean = mean(lab03$waiting),
                            sd = sd(lab03$waiting)),
                col = "red", lwd = 1.5) +
  labs(title = "Density Function: Waiting Times")

p3 <- ggplot(lab03, aes(x = waiting, y = "")) +
  geom_boxplot(fill = "royalblue", outlier.color = "royalblue") +
  labs(title = "Boxplot: Waiting Times", y = "")

p1 + (p2 / p3 + plot_layout(heights = c(4,1)))
```




```
mosaic::favstats(~ waiting, data = lab03) %>% kable(digits = 2)
```

| min | Q1 | median | Q3 | max | mean | sd | n | missing |
|---|---|---|---|---|---|---|---|---|
| 43 | 58 | 76 | 82 | 96 | 70.9 | 13.59 | 272 | 0 |

# 5 Question 5

> Would a model using the Normal distribution be an appropriate way to summarize the waiting time data? Why or why not?

No, a Normal distribution would not be an appropriate way to summarize this distribution, as the data are multi-modal, and substantially left skewed. Based on the histogram's appearance, the distribution might be well described as a mixture of two different (and perhaps close to Normal) distributions, one centered at 50-55 minutes, and another (which would be a more frequently observed component of the mixture), centered at about 80 minutes.

The mean waiting time of about 71 minutes, for example, seems informative, but it doesn't actually describe a typical result in either subgroup.

## 5.1 The Empirical Rule

A useful idea is that roughly 95% of the observations will lie within two standard deviations of the mean when the data follow a Normal distribution.

Here, that means within the range of 70.9 - 2(13.6) = 43.7 to 70.9 + 2(13.6) = 98.1 minutes. In this case, all but one (the minimum value of 43) of the 272 waiting times fall in this range, which is more than we would expect if the waiting times were Normally distributed.

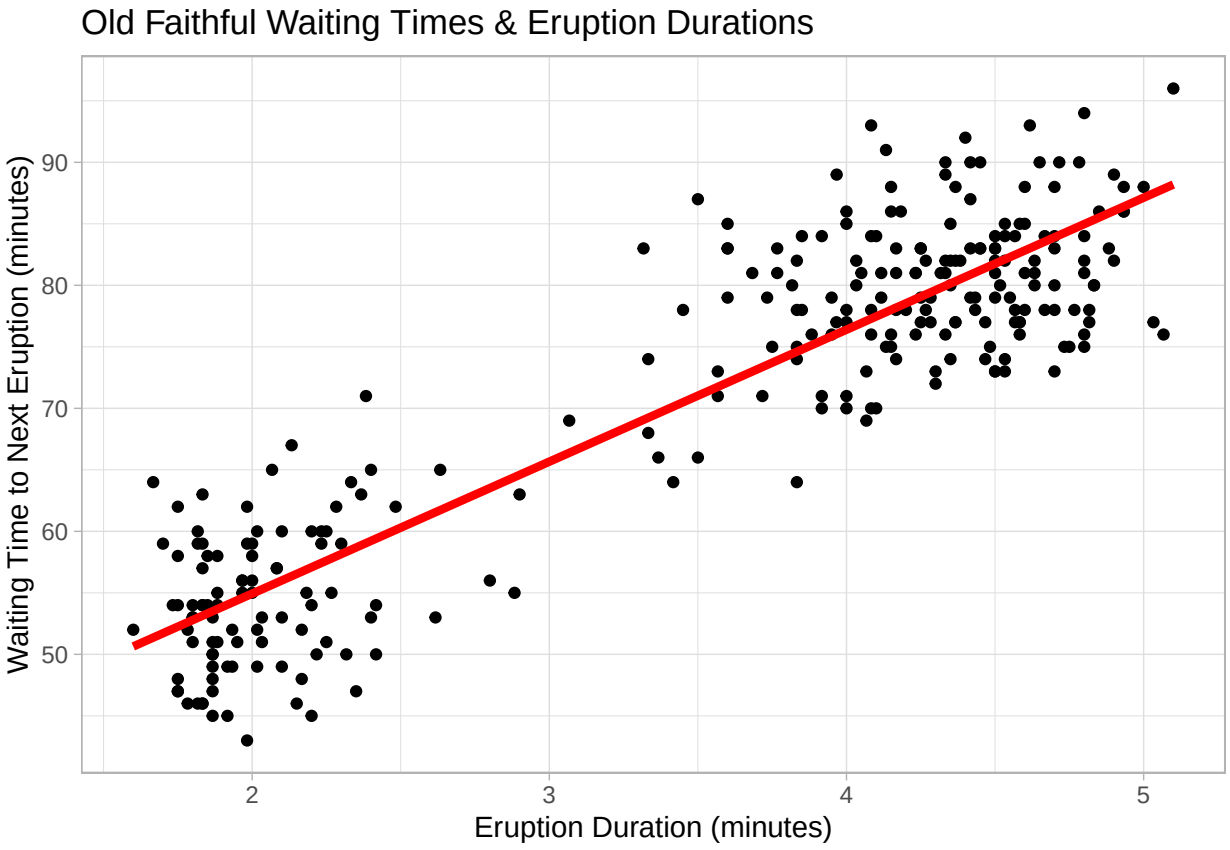
# 6 Question 6

> Plot a scatterplot of the waiting times (y-axis) vs. the eruption durations (x-axis), and be sure your plot is very clearly labeled. Include the result of fitting a straight line regression model in your plot. Then describe your general impression of the plot in a sentence or two: what sort of relationship do you see?

How, then, can we help the tourists? We need more information. One readily available characteristic of the geyser is the duration of the previous eruption. We can think of the `faithful` data as pairs of the form (eruption duration, time to next eruption) and then build a scatterplot of those pairs.

The plot reveals two clusters: it appears that eruption durations of 1.5 to 2.5 minutes are followed by shorter waiting times of 50-65 minutes, while longer eruption durations (of roughly 4 to 5 minutes) are followed by longer waiting times of 75-95 minutes.

```
ggplot(lab03, aes(x = eruptions, y = waiting)) +
  geom_point() +
  geom_smooth(method = "lm", formula = y ~ x, se = FALSE,
              color = "red", lwd = 1.5) +
  labs(title = "Old Faithful Waiting Times & Eruption Durations",
       y = "Waiting Time to Next Eruption (minutes)",
       x = "Eruption Duration (minutes)")
```



The existence of two subgroups in this type of data is rare, but not unheard of. J. S. Rinehart, in a 1969 paper in the *Journal of Geophysical Research*, provides a mechanism for this pattern based on the temperature level of the water at the bottom of a geyser tube at the time the water at the top reaches the boiling temperature. That a shorter eruption would be followed by a shorter waiting time (and a longer eruption would be followed by a longer waiting time) is also consistent with Rinehart's model, since a short eruption is characterized by having more water at the bottom of the geyser heated short of boiling temperature, and left in the tube. This water has been heated somewhat, however, so that it takes less time for the next eruption to occur. A long eruption results in the tube being emptied, so the water must be heated from a colder temperature, which takes longer.

We didn't ask, but you might ask: what is the correlation of waiting time with eruption duration? How would you interpret this result?

The Pearson correlation coefficient is 0.901.

```
lab03 %$%
  cor(waiting, eruptions)
```

```
[1] 0.9008112
```

This indicates a strong positive (or direct) and nearly linear relationship between eruption duration and waiting time.

### 6.1 On Correlation

Any two variables can be correlated. Any correlation that is not zero indicates some degree of correlation. Also, correlation is unitless: it's not a percentage of anything. The correlation is 0.9 here: undoubtedly a strong positive correlation. A perfect correlation would be +1 or -1 (depending on the direction of the relationship) and whether a correlation is strong depends powerfully on the context. For now, it's probably best to suggest that any correlation above about 0.5 in absolute value is usually fairly strong, and any correlation below 0.3 in absolute value is usually fairly weak.

### 6.2 On Rounding

The waiting time data in the `faithful` data frame are rounded to the nearest integer number of minutes. It is therefore silly to claim substantially more precision than 0 decimal places in evaluating summary statistics based on those data. Adding a single additional significant figure in summarizing data is usually somewhat justifiable, but any more than that is not.

Specifying a standard deviation, or a correlation coefficient to more than one decimal place in this case is likely to be inappropriate. The standard deviation of waiting times was about 14 minutes, or maybe 13.6 minutes, but not really 13.5949738 minutes. The correlation of waiting time and eruption duration is about 0.9, but not really 0.9008112.

Borrowing from a great line by John Tukey in a slightly different context:

> Be approximately right, rather than exactly wrong.

## 7 Question 7

> Does a linear model seem like an appropriate thing to use in attempting to predict the waiting time given the most recent eruption duration, based on these data? Why or why not? Your response should specify the regression line you fit for Question 6 and also provide a relevant summary statistic (or two) that provide an indication about how well that line fits the data.

Yes, a linear model might well be a useful summary here, as the waiting time for the next eruption shows a nearly linear relationship with eruption duration. The scatter of points tracks with the regression line fairly closely across the range of eruption durations.

The regression equation is specified by the code below.

```
m1 <- lm(waiting ~ eruptions, data = lab03)
tidy(m1) %>% kable(digits = 3)
```

| term | estimate | std.error | statistic | p.value |
|---|---|---|---|---|
| (Intercept) | 33.474 | 1.155 | 28.985 | 0 |
| eruptions | 10.730 | 0.315 | 34.089 | 0 |

```
glance(m1) %>% select(r.squared)
```

```
# A tibble: 1 x 1
  r.squared
      <dbl>
1     0.811
```

The regression equation is `waiting` = 33.47 + 10.73 `eruptions`, and this accounts for just over 81% of the variation in `waiting` times. It seems that there is a strong and direct (positive) linear relation between the two variables.

# 8 Questions 8 and 9

> Investigate questions 6 and 7 again using the* `geyser` *data in the* `MASS` *package, and compare your results appropriately.

```
lab03extra <- tibble(MASS::geyser)
lab03extra
```

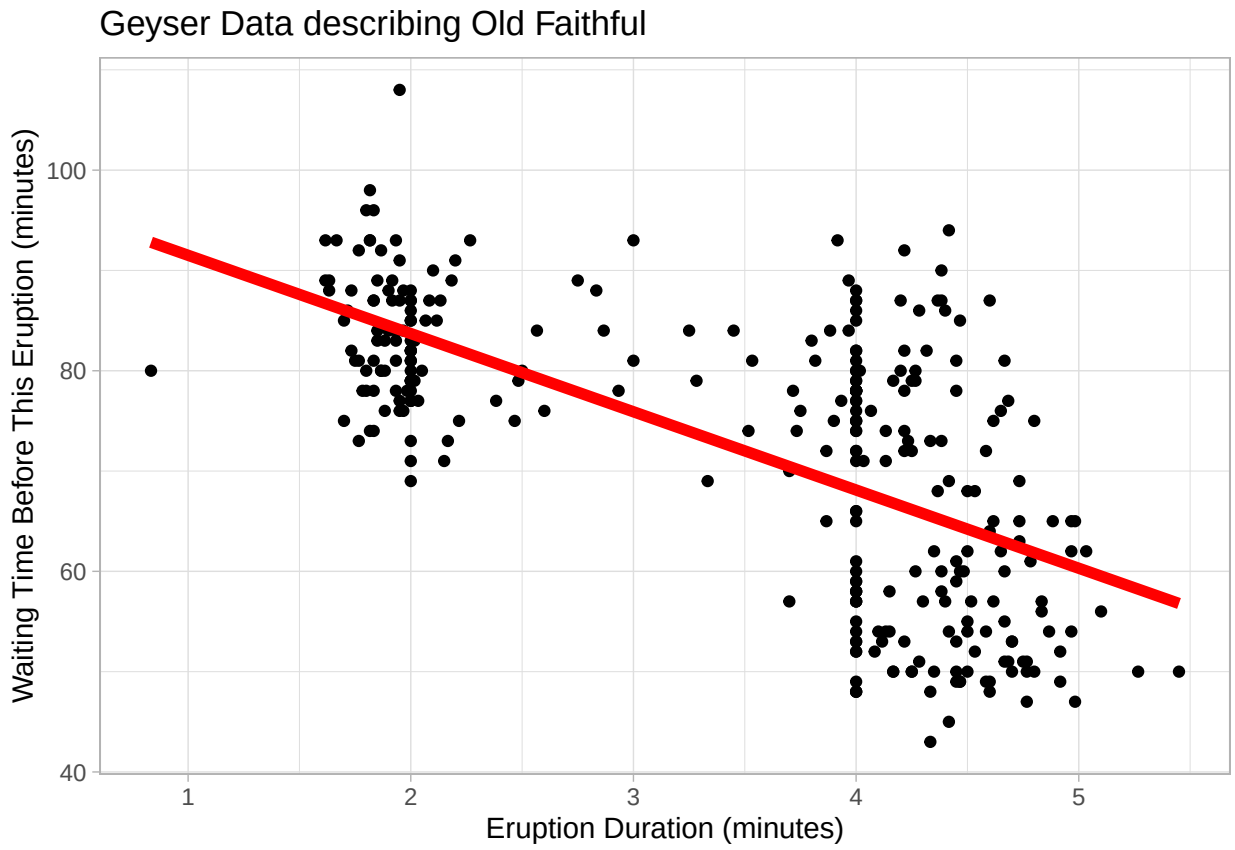
```
# A tibble: 299 x 2
   waiting duration
     <dbl>    <dbl>
 1      80     4.02
 2      71     2.15
 3      57     4
 4      80     4
 5      75     4
 6      77     2
 7      60     4.38
 8      86     4.28
 9      77     2.03
10      56     4.83
# ... with 289 more rows
```

There are several differences between the data frames.

- One difference is that we have 299 observations in the `geyser` data, 27 more than we had in the `faithful` data.
- The second, and more important distinction is that the waiting times now refer to the *current* eruption, so that when we plot the results as they are given, they show the waiting time preceding *this* eruption, rather than the waiting time preceding *the next* eruption.

- Third, we see lots of eruption durations specified as exactly 2 or exactly 4, in the `geyser` data, creating vertical lines in the scatterplot.

```
ggplot(lab03extra, aes(x = duration, y = waiting)) +
  geom_point() +
  geom_smooth(method = "lm", se = FALSE, formula = y ~ x,
              color = "red", lwd = 2) +
  labs(title = "Geyser Data describing Old Faithful",
       x = "Eruption Duration (minutes)",
       y = "Waiting Time Before This Eruption (minutes)")
```




The slope of the regression line is *negative* here, indicating that eruptions of shorter duration (say, 1.5 to 2.5 minutes) were preceded by longer waiting times while eruptions of longer duration (say, 4-5 minutes) were preceded by shorter waiting times. The correlation is -0.645, which indicates a strong negative association between the waiting time for this eruption and its duration.

The regression line is now...

```
m2 <- lm(waiting ~ duration, data = lab03extra)
tidy(m2) %>% kable(digits = 3)
```

| term | estimate | std.error | statistic | p.value |
|---|---|---|---|---|
| (Intercept) | 99.31 | 1.957 | 50.748 | 0 |
| duration | -7.80 | 0.537 | -14.531 | 0 |

```
glance(m2) %>% select(r.squared)
```

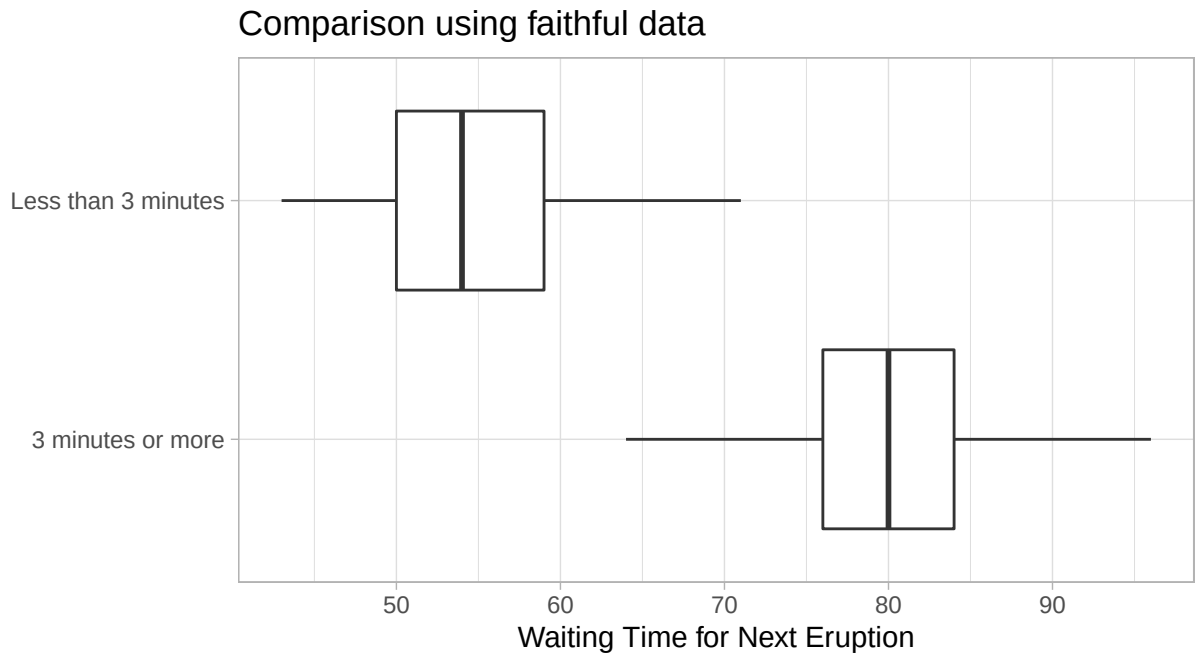
```
# A tibble: 1 x 1
  r.squared
      <dbl>
1     0.416
```

The conclusions we would draw here, are thus similar to those we developed for the original `faithful` data, but the available information is arranged a bit differently.

## 8.1 An Alternative Model for the `faithful` Data

We noticed two dominant effects in the `faithful` data: there are two different subgroups, and a longer eruption tends to be followed by a longer time interval until the next eruption. Suppose we separate the eruptions by whether the duration is less than three minutes.

```
lab03 <- lab03 %>%
  mutate(timegroup = ifelse(eruptions < 3,
                            "Less than 3 minutes",
                            "3 minutes or more"))

ggplot(lab03, aes(x = timegroup, y = waiting)) +
  geom_boxplot() +
  coord_flip() +
  labs(title = "Comparison using faithful data",
       x = "", y = "Waiting Time for Next Eruption")
```




```
mosaic::favstats(waiting ~ timegroup, data = lab03)
```

```
            timegroup min Q1 median Q3 max     mean
1   3 minutes or more  64 76     80 84  96 79.98857
2 Less than 3 minutes  43 50     54 59  71 54.49485
        sd   n missing
```

```
1 5.994239 175       0
2 5.840098  97       0
```

Based on these summaries, a simple prediction rule would be that an eruption of less than 3 minutes will be followed by a waiting time of about 55 minutes, while an eruption of duration 3 minutes or more will be followed by a waiting time of about 80 minutes. Further, the latter (longer) waiting time would be expected to occur about 2/3 of the time.

# 9 Question 10

> In your reading of Jeff Leek's book so far (you should have completed at least Sections 1-5 and 9-12), what is the most important thing you've learned? In a short essay (or perhaps 100-150 words), identify the relevant passage from Leek's book appropriately (be sure to specify the section of the book and perhaps provide a short quotation), and then provide a brief argument as to why this particular thing is something you value, and how it might apply to your work.

We don't write sketches for essay questions.

# 10 On Grading Lab 03

Your grade on Lab 03 is on a 0-100 scale.

- You'll receive 10 points for providing both the R Markdown file and the knitted HTML file based on that R Markdown file in a timely manner.

The next 70 points will be awarded as follows...

Questions 1, 4, 5, 6, 7, 8 and 9 will each be graded on a scale from 0-10 points. We will not be grading your responses to Questions 2 or 3.

- 10 points for a correct and clearly labeled plot in Question 1.
- 10 points for a correct answer, written in complete sentences for Question 4. We only expected you to interpret the histogram, not build multiple other plots, although if you did, that's great.
- 10 points for a correct answer, written in complete sentences for Question 5.
- 10 points for a correct and clearly labeled plot in Question 6.
- 10 points for an appropriate response and correct regression fit in Question 7, including a comparison between what is shown in Questions 7 and 9..
- 10 points for a correct and clearly labeled plot in Question 8.
- 10 points for an appropriate response and correct regression fit in Question 9, including a comparison between what is shown in Questions 7 and 9.

For these seven questions,

- Responses with a typographical or spelling mistake but nothing else wrong will receive 9 points.
- Generally correct responses with minor problems (including problems with grammar, spelling or syntax) will receive 7-8 points.
- Responses that state things that are incorrect, or with other major problems will receive fewer than 7 points.

Finally, question 10 will be worth 20 points.

- You will receive 17-18 points for a reasonable response to the question, where you meet all specifications, and write in complete, grammatically correct English sentences without spelling mistakes.
- Very strong essays that meet all of those criteria will receive scores of 19, with perhaps the top three or four essays overall receiving a score of 20. These will be especially effective presentations that also meet all parameters.

- Essays that *mostly* fit the requirements, but have a few problems with English or with meeting the request in other ways will usually receive scores of 14-16.
- Weaker essays with multiple problems meeting the specifications, or suffering from more than a few problems with English grammar, syntax, and so forth will receive scores no higher than 14.

# 11 Session Information

```
sessionInfo()
```

```
R version 4.0.2 (2020-06-22)
Platform: x86_64-w64-mingw32/x64 (64-bit)
Running under: Windows 10 x64 (build 19041)

Matrix products: default

locale:
[1] LC_COLLATE=English_United States.1252
[2] LC_CTYPE=English_United States.1252
[3] LC_MONETARY=English_United States.1252
[4] LC_NUMERIC=C
[5] LC_TIME=English_United States.1252

attached base packages:
[1] stats     graphics  grDevices utils     datasets
[6] methods   base

other attached packages:
 [1] forcats_0.5.0   stringr_1.4.0   dplyr_1.0.2
 [4] purrr_0.3.4     readr_1.3.1     tidyr_1.1.2
 [7] tibble_3.0.3    ggplot2_3.3.2   tidyverse_1.3.0
[10] patchwork_1.0.1 magrittr_1.5    knitr_1.29
[13] broom_0.7.0     MASS_7.3-52

loaded via a namespace (and not attached):
 [1] nlme_3.1-148      fs_1.5.0          lubridate_1.7.9
 [4] httr_1.4.2        tools_4.0.2       backports_1.1.7
 [7] utf8_1.1.4        R6_2.4.1          mgcv_1.8-31
[10] DBI_1.1.0         lazyeval_0.2.2    colorspace_1.4-1
[13] withr_2.2.0       tidyselect_1.1.0  gridExtra_2.3
[16] mnormt_2.0.1      leaflet_2.0.3     compiler_4.0.2
[19] cli_2.0.2         rvest_0.3.6       xml2_1.3.2
[22] ggdendro_0.1.21   labeling_0.3      mosaicCore_0.6.0
[25] scales_1.1.1      psych_2.0.8       digest_0.6.25
[28] ggformula_0.9.4   rmarkdown_2.3.3   pkgconfig_2.0.3
[31] htmltools_0.5.0   highr_0.8         dbplyr_1.4.4
[34] htmlwidgets_1.5.1 rlang_0.4.7       readxl_1.3.1
[37] rstudioapi_0.11   farver_2.0.3      generics_0.0.2
[40] jsonlite_1.7.0    crosstalk_1.1.0.1 mosaicData_0.18.0
[43] Matrix_1.2-18     Rcpp_1.0.5        munsell_0.5.0
[46] fansi_0.4.1       lifecycle_0.2.0   stringi_1.4.6
[49] yaml_2.2.1        ggstance_0.3.4    grid_4.0.2
[52] blob_1.2.1        parallel_4.0.2    ggrepel_0.8.2
```

```
[55] crayon_1.3.4      lattice_0.20-41   haven_2.3.1
[58] splines_4.0.2     hms_0.5.3         tmvnsim_1.0-2
[61] pillar_1.4.6      reprex_0.3.0      glue_1.4.2
[64] evaluate_0.14     modelr_0.1.8      vctrs_0.3.3
[67] tweenr_1.0.1      cellranger_1.1.0  gtable_0.3.0
[70] polyclip_1.10-0   assertthat_0.2.1  xfun_0.16
[73] ggforce_0.3.2     mosaic_1.7.0      ellipsis_0.3.1
```