Answer Sketch for Homework D

431 Staff and Professor Love

*Due **2019-09-20** at 2 PM. Last Edited 2019-09-25 22:38:46*

Contents

R Setup

Here's the complete R setup we used.

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

library(janitor); library(magrittr); library(tidyverse)
```

Then we read in the data set, which we'd stored in the project directory.

```
LBWunicef <- read_csv("unicef_lbw.csv") %>%
  clean_names()
```

```
Parsed with column specification:
cols(
  iso3_code = col_character(),
  nation = col_character(),
  pct_low_birthweight = col_double(),
  unicef_subregion = col_character(),
  least_developed = col_double(),
  pct_no_birthweight = col_double()
)
```

We could use `glimpse` to take a look at the data...

```
glimpse(LBWunicef)
```

```
Observations: 202
Variables: 6
$ iso3_code           <chr> "AFG", "ALB", "DZA", "AND",...
$ nation              <chr> "Afghanistan", "Albania", "...
$ pct_low_birthweight <dbl> NA, 4.59, 7.25, 7.45, 15.26...
$ unicef_subregion    <chr> "South Asia", "Eastern Euro...
$ least_developed     <dbl> 1, 0, 0, 0, 1, 0, 0, 0, 0, ...
$ pct_no_birthweight  <dbl> 86.27, 13.17, 11.43, 3.01, ...
```

Or we could just list the tibble, as a check on what we've done...

```
LBWunicef
```

```
# A tibble: 202 x 6
   iso3_code nation pct_low_birthwe~ unicef_subregion
   <chr>     <chr>             <dbl> <chr>
 1 AFG       Afgha~             NA    South Asia
 2 ALB       Alban~              4.59 Eastern Europe ~
 3 DZA       Alger~              7.25 Middle East and~
 4 AND       Andor~              7.45 Western Europe
 5 AGO       Angola             15.3  Eastern and Sou~
 6 AIA       Angui~             NA    Latin America a~
 7 ATG       Antig~              9.05 Latin America a~
 8 ARG       Argen~              7.35 Latin America a~
 9 ARM       Armen~              8.98 Eastern Europe ~
10 AUS       Austr~              6.52 East Asia and P~
# ... with 192 more rows, and 2 more variables:
#   least_developed <dbl>, pct_no_birthweight <dbl>
```

Here's the data description from the assignment for the variables we'll actually use:

- `iso3_code` = three-letter code for each nation
- `nation` = the nation's name
- `pct_low_birthweight` = the nation's low birth weight percentage estimate from 2015 (updated June 2019) from https://data.unicef.org/wp-content/uploads/2014/10/Low-birthweight-data-2000-2015.xlsx
- `least_developed` = whether or not the nation is regarded by the United Nations High Representative for the Least Developed Countries, Landlocked Developing Countries and Small Island Developing States as one of the "least developed" countries on Earth (note that `least_developed` = 1 if the nation is in the "least developed countries" group and is 0 otherwise.)

Reordering the Variables

The problem with that listing of our `LBWunicef` tibble is that it shows a variable I don't need (`unicef_subregion`) and hides one I will need (`least_developed`). So, I'll use `select` (along with the `everything()` function) to re-order the variables, putting the ones I'll need for this assignment at the beginning, so that when I print a tibble, the variables I need will appear in this answer sketch.

```
LBWunicef <- LBWunicef %>%
  select(nation, pct_low_birthweight, least_developed, everything())

LBWunicef
```

```
# A tibble: 202 x 6
   nation pct_low_birthwe~ least_developed iso3_code
   <chr>             <dbl>           <dbl> <chr>
 1 Afgha~             NA                 1 AFG
 2 Alban~              4.59              0 ALB
 3 Alger~              7.25              0 DZA
 4 Andor~              7.45              0 AND
 5 Angola             15.3               1 AGO
 6 Angui~             NA                 0 AIA
 7 Antig~              9.05              0 ATG
 8 Argen~              7.35              0 ARG
 9 Armen~              8.98              0 ARM
10 Austr~              6.52              0 AUS
# ... with 192 more rows, and 2 more variables:
#   unicef_subregion <chr>, pct_no_birthweight <dbl>
```

OK. Much better for our purposes.

Question 1

> How many nations have non-missing low birth weight percentage estimates?

While there are `nrow(LBWunicef)` nations listed in the data set, only 147 have non-missing low birth weight percentage estimates.

Note: The standard applied here, revealed in the Notes on the data at https://data.unicef.org/topic/nutrition/low-birthweight/, is low birth weight is defined as less than 2,500 grams (up to and including 2,499 grams.)

Using `dim` or `nrow` and `summary`

We can use the `dim` function, or the `nrow` function to determine the number of rows in the `LBWunicef` data, and we can use the `summary` function to see if there are any missing values in the `LBWunicef` data:

```
dim(LBWunicef)
```

```
[1] 202   6
```

```
nrow(LBWunicef)
```

```
[1] 202
```

```
summary(LBWunicef)
```

```
    nation          pct_low_birthweight least_developed
```

```
 Length:202         Min.   : 2.410      Min.   :0.0000
 Class :character   1st Qu.: 6.350      1st Qu.:0.0000
 Mode  :character   Median : 8.980      Median :0.0000
                    Mean   : 9.829      Mean   :0.2327
                    3rd Qu.:12.210      3rd Qu.:0.0000
                    Max.   :27.810      Max.   :1.0000
                    NA's   :55
  iso3_code         unicef_subregion   pct_no_birthweight
 Length:202         Length:202         Min.   : 0.000
 Class :character   Class :character   1st Qu.: 1.552
 Mode  :character   Mode  :character   Median : 6.455
                                       Mean   :18.734
                                       3rd Qu.:28.402
                                       Max.   :91.770
                                       NA's   :42
```

There are 55 missing values in `pct_low_birthweight`, as indicated by the summary. Since we have 202 nations in the data, we must have 202 - 55 = 147 nations with a value of `lbw.pct`.

Using `favstats`

Could we could use the `favstats` function from the `mosaic` package to obtain this directly?

```
mosaic::favstats(~ pct_low_birthweight, data = LBWunicef)
```

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

```
  min   Q1 median    Q3   max     mean       sd   n missing
 2.41 6.35   8.98 12.21 27.81 9.829048 4.492624 147      55
```

Yes. The `n` here indicates the number of non-missing values in the low birth weight percentage data.

Using `skim` from `skimr`

```
skimr::skim_with(numeric = list(hist = NULL))
## did that just to leave out the sparkline histograms

skimr::skim(LBWunicef)
```

```
Skim summary statistics
 n obs: 202
 n variables: 6

-- Variable type:character ---------------------------------------------------------------------------
         variable missing complete   n min max empty
        iso3_code       0      202 202   3   3     0
           nation       0      202 202   4  37     0
 unicef_subregion       0      202 202  10  31     0
 n_unique
      202
      202
        9
```

```
-- Variable type:numeric -------------------------------------------------------------------------------
            variable missing complete   n  mean    sd   p0
     least_developed       0      202 202  0.23  0.42 0
 pct_low_birthweight      55      147 202  9.83  4.49 2.41
  pct_no_birthweight      42      160 202 18.73 24.84 0
  p25  p50   p75  p100
 0    0     0     1
 6.35 8.98 12.21 27.81
 1.55 6.46 28.4  91.77
```

Question 2

> Which nations have the three largest low birth weight percentages? Are each of these considered by the UN to be "least developed" nations or not?

The three largest low birth weight percentages in the data are Bangladesh (27.81%), Comoros (23.70%), and Nepal (21.81%). Of these three nations, all three falls in the "least developed nations" category.

Using `dplyr` and the tidyverse

We can use `dplyr`, specifically the `arrange` function, to show a tibble that has been sorted in descending order of `pct_low_birthweight`. R Studio's cheat sheet for Data Transformation at https://www.rstudio.com/resources/cheatsheets/ is very helpful here.

```
LBWunicef %>% arrange(desc(pct_low_birthweight))
```

```
# A tibble: 202 x 6
   nation pct_low_birthwe~ least_developed iso3_code
   <chr>             <dbl>           <dbl> <chr>
 1 Bangl~             27.8               1 BGD
 2 Comor~             23.7               1 COM
 3 Nepal              21.8               1 NPL
 4 Guine~             21.1               1 GNB
 5 Phili~             20.2               0 PHL
 6 Seneg~             18.5               1 SEN
 7 Lao P~             17.3               1 LAO
 8 Moroc~             17.3               0 MAR
 9 Madag~             17.1               1 MDG
10 Mauri~             17.1               0 MUS
# ... with 192 more rows, and 2 more variables:
#   unicef_subregion <chr>, pct_no_birthweight <dbl>
```

And, if we wanted to view just the first three rows, we could arrange and then slice...

```
LBWunicef %>%
  arrange(desc(pct_low_birthweight)) %>%
  slice(1:3)
```

```
# A tibble: 3 x 6
  nation pct_low_birthwe~ least_developed iso3_code
  <chr>             <dbl>           <dbl> <chr>
1 Bangl~             27.8               1 BGD
2 Comor~             23.7               1 COM
```

```
3 Nepal              21.8               1 NPL
# ... with 2 more variables: unicef_subregion <chr>,
#   pct_no_birthweight <dbl>
```

A fast, one-line alternative with `rank`

With missing values, this can be a challenge. If we restrict the data to those nations with complete data on `pct_low_birthweight`, we can then do this in one (additional) line.

```
LBW_noNA <- LBWunicef %>% filter(complete.cases(pct_low_birthweight))

## The fastest one-line alternative I know
LBW_noNA[which(rank(LBW_noNA$pct_low_birthweight) > length(LBW_noNA$pct_low_birthweight) - 3),]
```

```
# A tibble: 3 x 6
  nation pct_low_birthwe~ least_developed iso3_code
  <chr>             <dbl>           <dbl> <chr>
1 Bangl~             27.8               1 BGD
2 Comor~             23.7               1 COM
3 Nepal              21.8               1 NPL
# ... with 2 more variables: unicef_subregion <chr>,
#   pct_no_birthweight <dbl>
```

`sort`, `which` and brute force

Clearly, we could solve this problem through simple brute force, inspecting the data until we find the largest values, and then associating them with Nations. The `sort` and `which` commands can help us here.

```
LBWunicef %$% sort(pct_low_birthweight)
```

```
  [1]  2.41  3.26  3.40  3.50  4.12  4.19  4.34  4.49  4.51
 [10]  4.51  4.53  4.59  4.93  4.95  5.00  5.06  5.10  5.26
 [19]  5.26  5.34  5.37  5.43  5.44  5.45  5.53  5.63  5.63
 [28]  5.69  5.77  5.81  5.89  5.94  6.12  6.15  6.15  6.25
 [37]  6.32  6.38  6.47  6.52  6.52  6.54  6.60  6.65  6.95
 [46]  6.96  7.22  7.25  7.25  7.28  7.32  7.35  7.44  7.45
 [55]  7.47  7.48  7.60  7.62  7.78  7.82  7.87  7.87  8.02
 [64]  8.09  8.18  8.21  8.27  8.38  8.39  8.60  8.75  8.77
 [73]  8.90  8.98  9.05  9.05  9.10  9.24  9.40  9.49  9.56
 [82]  9.64  9.86  9.96  9.97 10.09 10.30 10.32 10.50 10.52
 [91] 10.54 10.67 10.76 10.84 10.90 10.91 10.96 11.18 11.29
[100] 11.35 11.40 11.47 11.60 11.61 11.66 11.67 11.72 11.91
[109] 11.96 12.12 12.30 12.39 12.63 12.68 13.14 13.14 13.78
[118] 13.82 14.16 14.21 14.21 14.42 14.46 14.54 14.58 14.61
[127] 14.66 15.13 15.26 15.46 15.53 15.62 15.63 15.89 16.09
[136] 16.75 16.89 17.06 17.14 17.28 17.31 18.46 20.15 21.08
[145] 21.81 23.70 27.81
```

OK. So the three largest values have `pct_low_birthweight` greater than 21.5. How do we identify which nations those are?

```
LBWunicef %$% which(pct_low_birthweight > 21.5)
```

```
[1]  15  40 126
```

And now that we know which row numbers are the top 3, we can show all of the available data related to those three row numbers (including their names) using `slice` to identify specific rows in the data.

```
LBWunicef %>% slice(c(15, 40, 126))
```

```
# A tibble: 3 x 6
  nation pct_low_birthwe~ least_developed iso3_code
  <chr>             <dbl>           <dbl> <chr>
1 Bangl~             27.8               1 BGD
2 Comor~             23.7               1 COM
3 Nepal              21.8               1 NPL
# ... with 2 more variables: unicef_subregion <chr>,
#   pct_no_birthweight <dbl>
```

Question 3

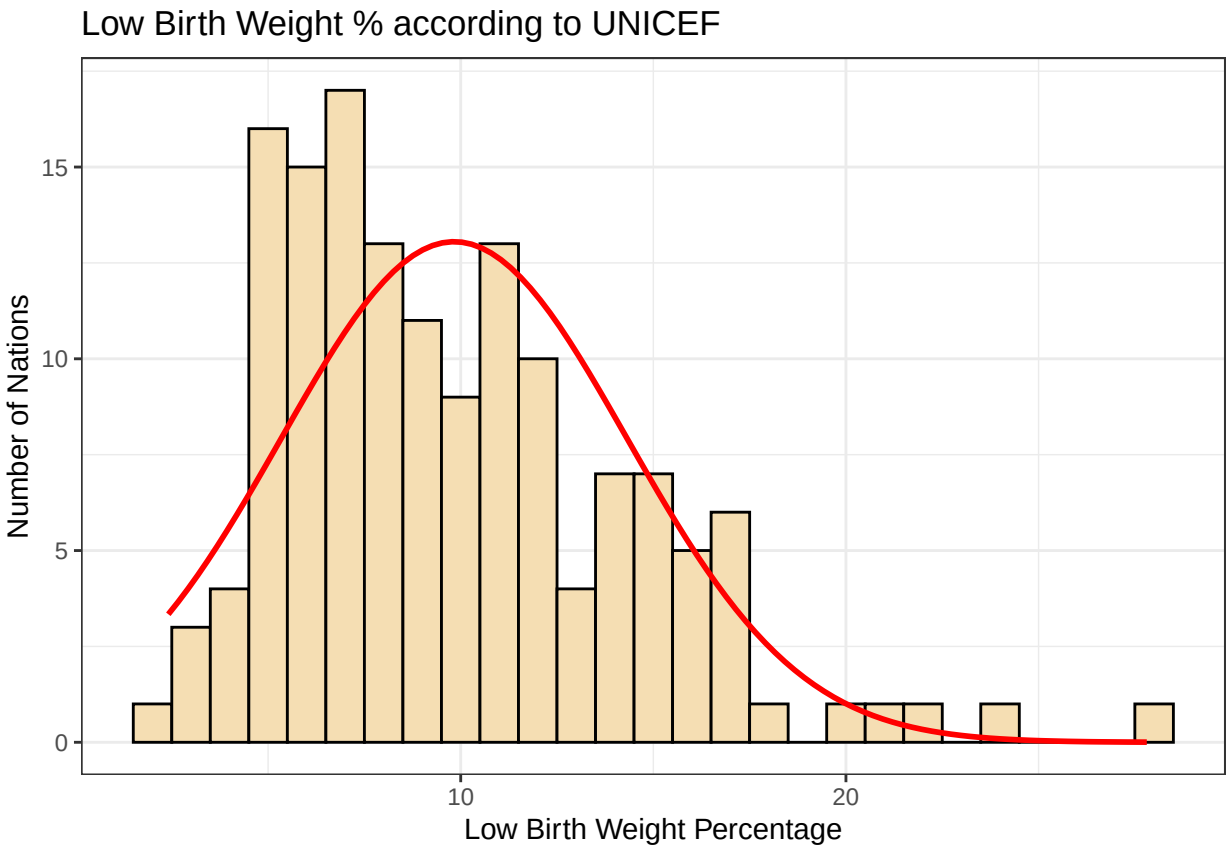
> Create a histogram of the low birth weight percentages, then superimpose a normal density function with the same mean and standard deviation in red. Based on your plot, is the standard deviation or the inter-quartile range a more appropriate measure of variation in the low birth weight rates? Why?

Here's one approach. We definitely will be helped by filtering our sample to include only those cases with complete data on low birth weight percentage estimate.

```
LBW_noNA <-
  LBWunicef %>%
  filter(complete.cases(pct_low_birthweight))

res <- mosaic::favstats(~ pct_low_birthweight, data = LBW_noNA) # save summaries
bin_w <- 1 # specify binwidth

ggplot(LBW_noNA, aes(x = pct_low_birthweight)) +
  geom_histogram(binwidth = bin_w, fill = "wheat",
                 col = "black") +
  theme_bw() +
  stat_function(
    fun = function(x) dnorm(x, mean = res$mean,
                            sd = res$sd) * res$n * bin_w,
    col = "red", size = 1) +
  labs(title = "Low Birth Weight % according to UNICEF",
       x = "Low Birth Weight Percentage",
       y = "Number of Nations")
```



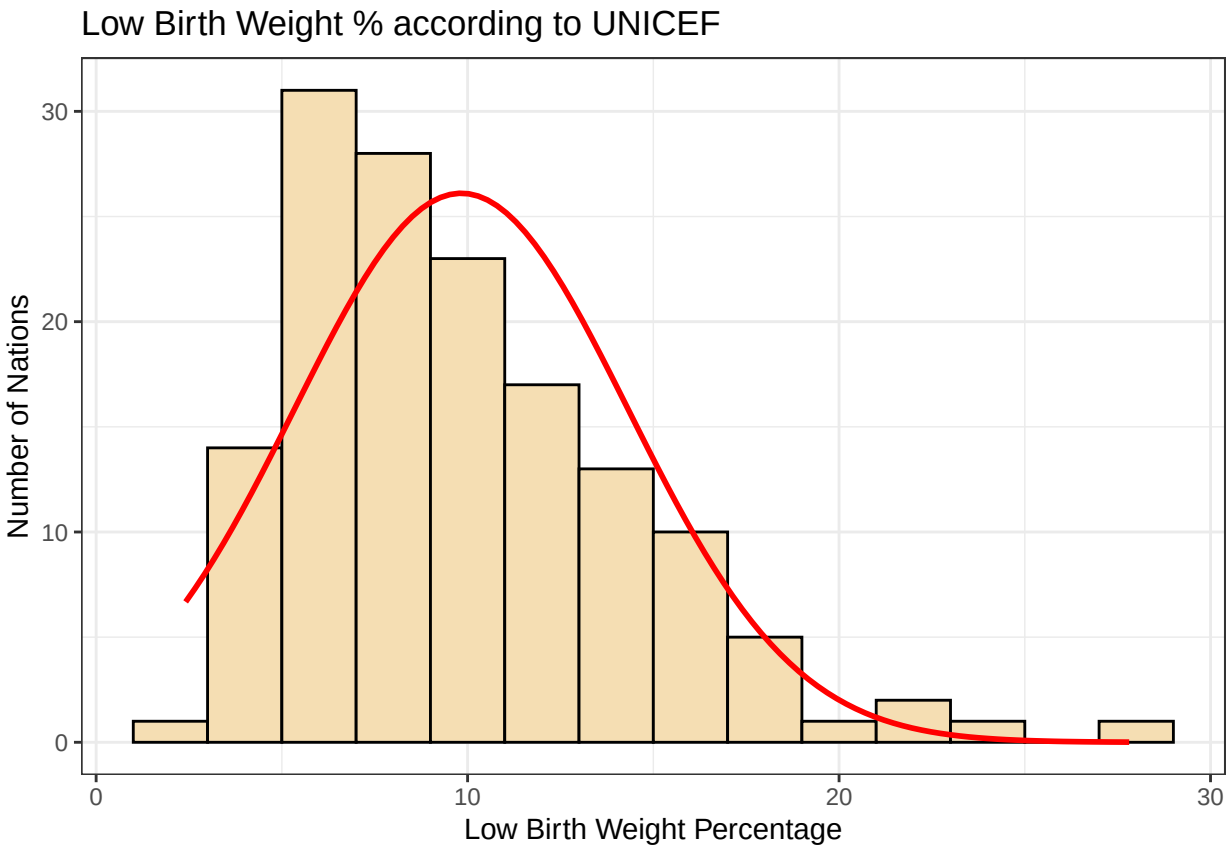
Clearly, the plot shows some right skew, and assuming a Normal model (while not by any means disastrous) doesn't appear to be especially well justified. Under these circumstances, the interquartile range is a more appropriate measure of spread for these data than the standard deviation would be.

Does the story change much if we change the binwidth to be twice as large? Not really, according to the plot below. There's still some signs of right skew here.

```
LBW_noNA <-
  LBWunicef %>%
  filter(complete.cases(pct_low_birthweight))

res <- mosaic::favstats(~ pct_low_birthweight, data = LBW_noNA) # save summaries
bin_w <- 2 # specify binwidth

ggplot(LBW_noNA, aes(x = pct_low_birthweight)) +
  geom_histogram(binwidth = bin_w, fill = "wheat",
                 col = "black") +
  theme_bw() +
  stat_function(
    fun = function(x) dnorm(x, mean = res$mean,
                            sd = res$sd) * res$n * bin_w,
    col = "red", size = 1) +
  labs(title = "Low Birth Weight % according to UNICEF",
       x = "Low Birth Weight Percentage",
       y = "Number of Nations")
```



Note that the IQR and standard deviation are available to us, if we want them, but we have to deal with the missing data somehow.

```
LBWunicef %>%
  summarize(IQR = IQR(pct_low_birthweight, na.rm = TRUE),
            SD = sd(pct_low_birthweight, na.rm = TRUE))
```

```
# A tibble: 1 x 2
    IQR    SD
  <dbl> <dbl>
1  5.86  4.49
```

Another option, of course, is to use a function that automatically restricts its summaries to non-missing values, like `favstats` from the `mosaic` package.

```
mosaic::favstats(~ pct_low_birthweight, data = LBWunicef)
```

```
  min   Q1 median    Q3   max     mean       sd   n missing
 2.41 6.35   8.98 12.21 27.81 9.829048 4.492624 147      55
```

And here, we can then calculate the IQR by subtracting Q1 from Q3.

Question 4

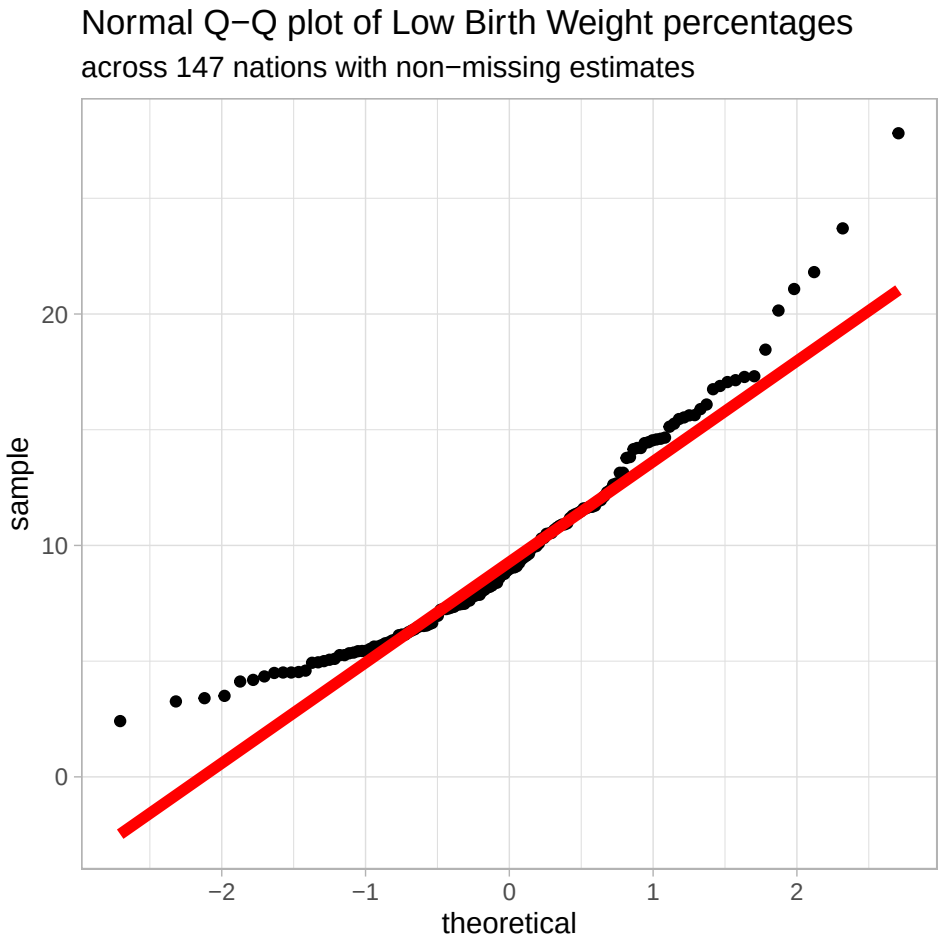
> Create a normal Q-Q plot for the low birth weight percentage estimates. Would you say that the data are approximately Normally distributed, or not approximately Normally distributed?

Justify your answer by interpreting what you see in your plot, and whatever summary statistics you deem to be useful in making your decision.

The data are somewhat right skewed, as indicated previously by the histogram, and now also by the curve in the normal Q-Q plot below.

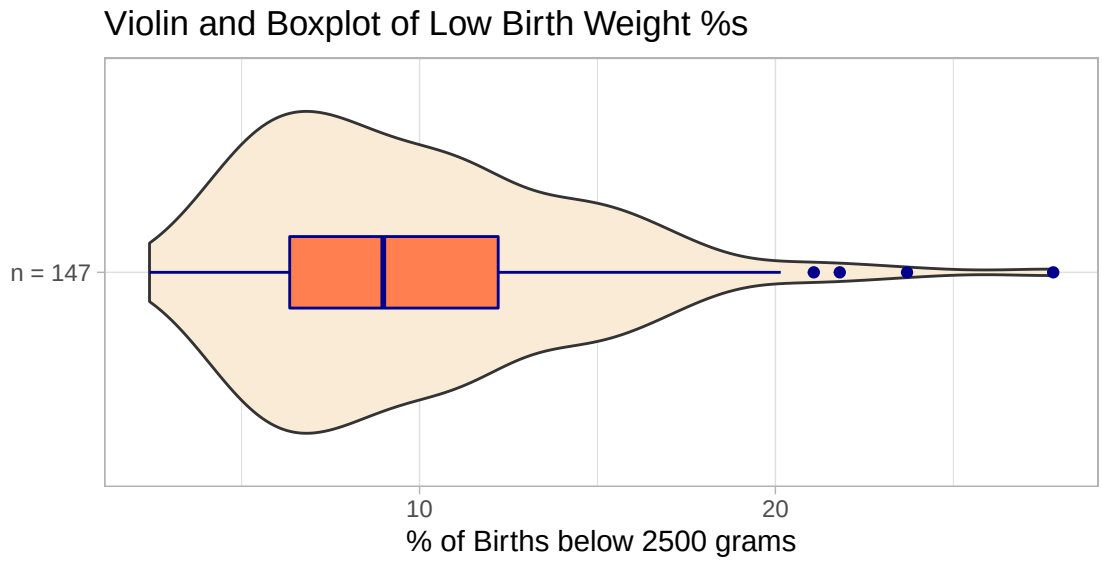
```
LBW_noNA <-
  LBWunicef %>%
  filter(complete.cases(pct_low_birthweight))

ggplot(LBW_noNA, aes(sample = pct_low_birthweight)) +
  geom_qq() + geom_qq_line(col = "red", lwd = 2) +
  labs(title = "Normal Q-Q plot of Low Birth Weight percentages",
       subtitle = "across 147 nations with non-missing estimates") +
  theme_light()
```



We could also have developed a boxplot, perhaps combined with a violin plot.

```
ggplot(LBW_noNA, aes(x = "n = 147", y = pct_low_birthweight)) +
  geom_violin(fill = "antiquewhite") +
  geom_boxplot(width = 0.2, fill = "coral", col = "darkblue") +
  coord_flip() +
  theme_light() +
  labs(title = "Violin and Boxplot of Low Birth Weight %s",
       x = "", y = "% of Births below 2500 grams")
```



Using Numerical Summaries to Assess Normality (if you must)

Assess Normality with plots, whenever possible. Summary statistics should play a supporting role.

Thinking about A Skewness Measure

```
mosaic::favstats(~ pct_low_birthweight, data = LBWunicef)
```

```
  min   Q1 median    Q3   max     mean       sd   n missing
 2.41 6.35   8.98 12.21 27.81 9.829048 4.492624 147      55
```

As for summary statistics, the mean (9.83) is well to the right of the median (8.98), and, since the standard deviation is 4.49. So the skew_1 value is also indicative of right skew, with $\text{skew}_1 = 0.189$, which is quite close to 0.2, the value we usually use as an indicator of substantial right skew.

```
LBW_noNA %>%
  summarize(Mean = mean(pct_low_birthweight),
            Median = median(pct_low_birthweight),
            SD = sd(pct_low_birthweight),
    skew1 = ( Mean - Median ) / SD) %>%
  knitr::kable(digits = 3)
```

Mean	Median	SD	skew1
9.829	8.98	4.493	0.189

Thinking about the Empirical Rule

We've already decided now that the data aren't symmetric enough for a Normal model to be a particularly good choice. If we wanted, we could also determine whether the Empirical Rule holds well for these data, and use that to help guide our understanding of whether the Normal model would work well (although at this point, that seems pretty settled.)

For instance, if a Normal model held, then about 68% of the nations would fall within two standard deviations of the mean. Is that true here?

```
LBW_noNA %>%
  count(mean_pm_1sd = pct_low_birthweight >
          mean(pct_low_birthweight) - sd(pct_low_birthweight) &
        pct_low_birthweight <
          mean(pct_low_birthweight) + sd(pct_low_birthweight) )
```

```
# A tibble: 2 x 2
  mean_pm_1sd     n
  <lgl>       <int>
1 FALSE          45
2 TRUE          102
```

In fact, 102/147 is 69.4% of the nations that fall within 1 SD of the mean. That's a little bit higher than we would expect in data that followed a Normal distribution, but it's awfully close to the expected 68%.

If a Normal model held, then about 95% of the data would fall within two standard deviations of the mean. Is that true?

```
LBW_noNA %>%
  count(mean_pm_2sd = pct_low_birthweight >
          mean(pct_low_birthweight) - 2*sd(pct_low_birthweight) &
        pct_low_birthweight <
          mean(pct_low_birthweight) + 2*sd(pct_low_birthweight) )
```

```
# A tibble: 2 x 2
  mean_pm_2sd     n
  <lgl>       <int>
1 FALSE           5
2 TRUE          142
```

Note that 142/147 is 96.6% of the nations that fall within 2 SD of the mean value of `pct_low_birthweight`. That's also pretty close to the expected 95%, so it appears that the Normal model wouldn't be such a bad choice, in terms of the Empirical Rule fitting the data.

Thinking about Hypothesis Testing (Shapiro-Wilk Test)

A really **bad** idea (if you can avoid it) is to use a hypothesis test to assess Normality. Such a test is essentially valueless without first looking at a plot of the data. But such tests are available. None are good, specifically because they only test for specific types of non-Normality, and most people can visualize several types of non-Normality simultaneously, making that (visualization) a much more powerful tool (even if it seems less "objective").

One of the simplest of such tests to run is the Shapiro-Wilk test of Normality. That test estimates a p value, something that's very easy to misinterpret. In the case of a Shapiro-Wilk test, if you see a p value that is less than a given value (the most common choice is 0.05), then that is meant to suggest that there is some evidence of non-Normality in the way the Shapiro-Wilk test tries to find it, or at least there's more evidence than if the p value were larger. The p value is a conditional probability, so it will always fall between 0 and 1.

```
LBWunicef %$% shapiro.test(pct_low_birthweight)
```

```
    Shapiro-Wilk normality test

data:  pct_low_birthweight
```

```
W = 0.93483, p-value = 2.699e-06
```

Here, the p value is very small, which pushes us slightly further in the direction of concluding that the Normal model isn't a good choice for these data.

Other hypothesis tests are available for assessing non-Normality. Again, none are great. In fact, I can't remember the last time I reported a Shapiro-Wilk test (or any other hypothesis test for non-Normality) in practical work.

Question 5

> Display an effective graph comparing the two development groups (least developed nations vs. all other nations) in terms of their percentages of low birth weight births. What conclusions can you draw about the distribution of low birth weight rates across the two development groups? Be sure to label your graph so it stands alone, and also supplement your graph with separate text discussing your conclusions.

Generally, the low birth weight percentages are higher in the nations which are least developed, but there is considerable overlap.

Preliminaries: Creating a Factor

Before I build my plot, I'll create a new factor variable in the `LBWunicef` data, which I'll call `least_developed` and which will contain the levels No and Yes, for the original numeric 0 and 1.

```
LBWunicef <- LBWunicef %>%
  mutate(least_dev_f =
           fct_recode(factor(least_developed), "Yes" = "1", "No" = "0"))
```

Just as a sanity check, I'll be sure I've recoded appropriately with a frequency table:

```
LBWunicef %>% tabyl(least_developed, least_dev_f)
```

```
 least_developed  No Yes
               0 155   0
               1   0  47
```

We'll want to make sure this appears in the version of LBWunicef that we built omitting the cases with missing low birth weight percentage estimates, too.

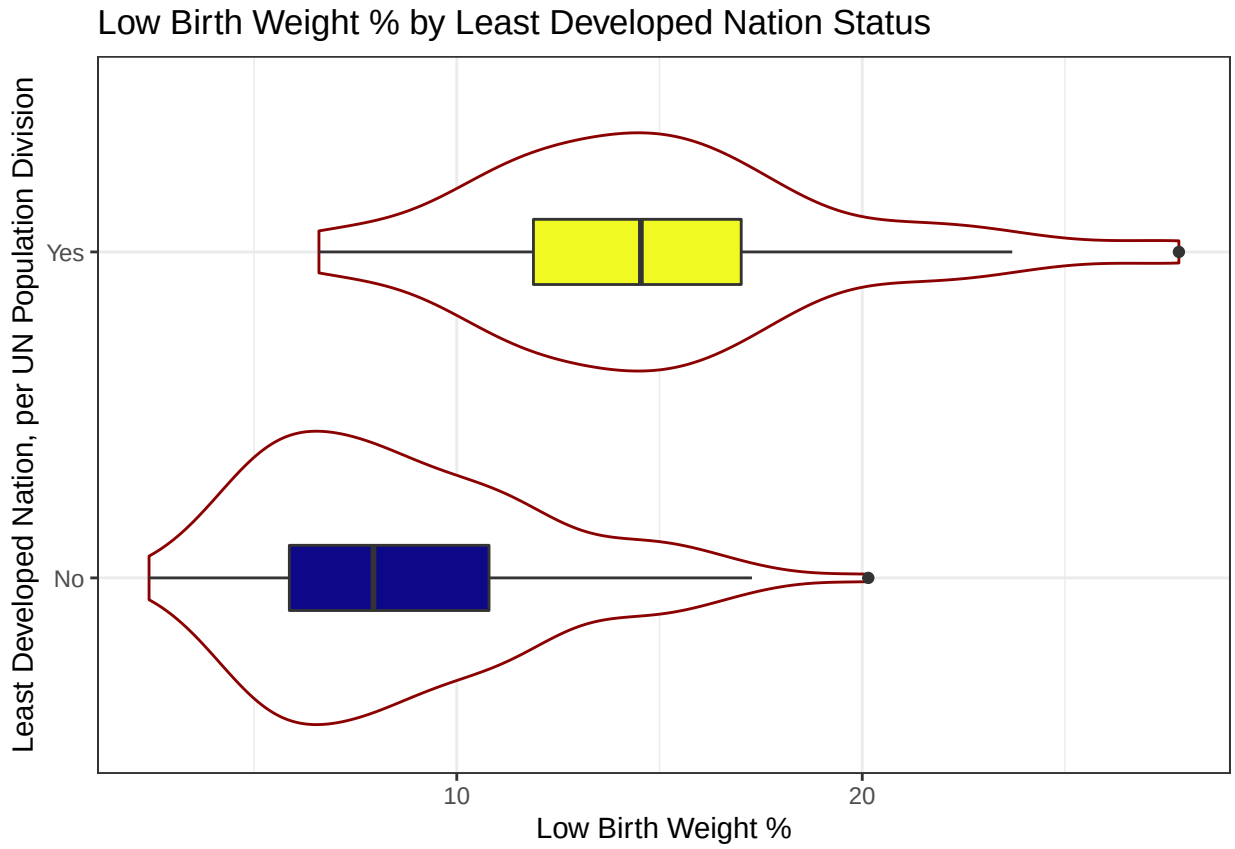
```
LBW_noNA <- LBW_noNA %>%
  mutate(least_dev_f =
           fct_recode(factor(least_developed), "Yes" = "1", "No" = "0"))
```

A Comparison Boxplot (and Violin Plot)

Now, I'll build a comparison boxplot. I'll get a little fancy and create violin plots while I am at it.

```
ggplot(LBW_noNA, aes(x = least_dev_f, y = pct_low_birthweight)) +
  geom_violin(col = "darkred") +
  geom_boxplot(aes(fill = least_dev_f), width = 0.2) +
  guides(fill = FALSE) +
  scale_fill_viridis_d(option = "C") +
  coord_flip() +
  labs(title = "Low Birth Weight % by Least Developed Nation Status",
```

```
      y = "Low Birth Weight %",
      x = "Least Developed Nation, per UN Population Division") +
  theme_bw()
```

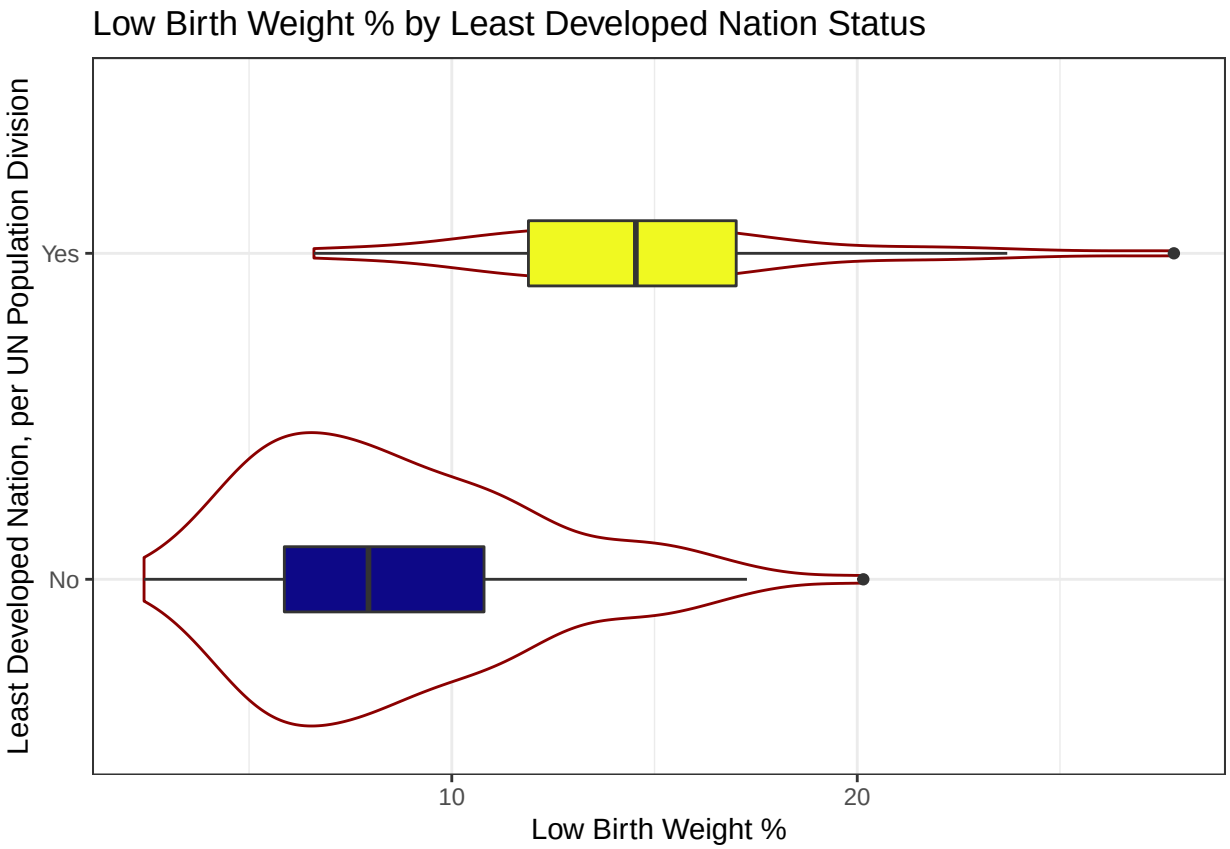


Note: Making the Width of the Violin reflect the sample size

You can set the `scale` parameter to "count" in the `geom_violin()` call to adjust the violins to have areas that are scaled proportionally to the number of observations. Otherwise, they will all have the same area.

Here's an example of that for our data, which shows off the much larger group of No than Yes nations in terms of Least Developed status.

```
ggplot(LBW_noNA, aes(x = least_dev_f, y = pct_low_birthweight)) +
  geom_violin(col = "darkred", scale = "count") +
  geom_boxplot(aes(fill = least_dev_f), width = 0.2) +
  guides(fill = FALSE) +
  scale_fill_viridis_d(option = "C") +
  coord_flip() +
  labs(title = "Low Birth Weight % by Least Developed Nation Status",
       y = "Low Birth Weight %",
       x = "Least Developed Nation, per UN Population Division") +
  theme_bw()
```

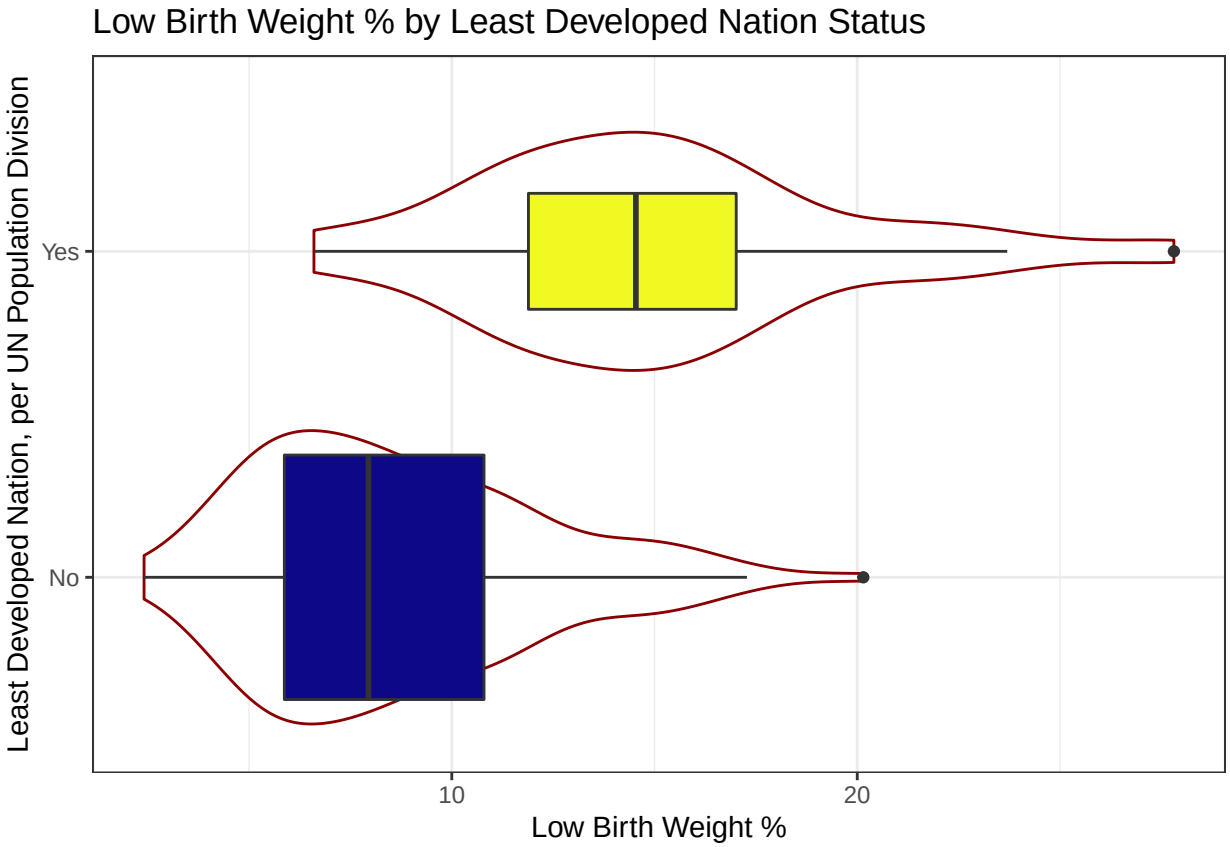


So we see that there are considerably more No than Yes nations.

What if you wanted the boxplots to indicate the size of the data?

You could use `varwidth = TRUE` in the `geom_boxplot` call, like this:

```
ggplot(LBW_noNA, aes(x = least_dev_f, y = pct_low_birthweight)) +
  geom_violin(col = "darkred") +
  geom_boxplot(aes(fill = least_dev_f), varwidth = TRUE) +
  guides(fill = FALSE) +
  scale_fill_viridis_d(option = "C") +
  coord_flip() +
  labs(title = "Low Birth Weight % by Least Developed Nation Status",
       y = "Low Birth Weight %",
       x = "Least Developed Nation, per UN Population Division") +
  theme_bw()
```

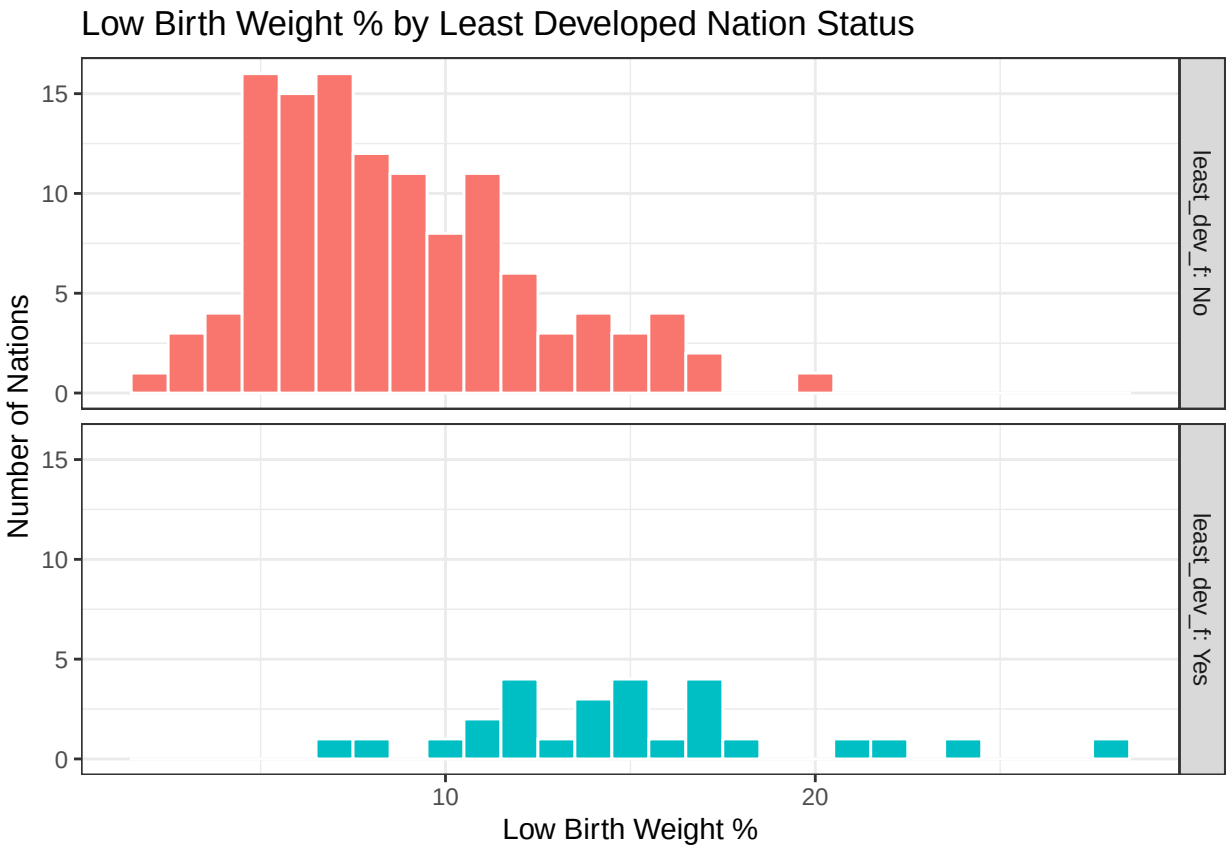


That approach makes the width (so height in this case, because we've flipped the coordinates) of the boxplot proportional to the square root of the sample size. Again, there are more No than Yes.

Another Reasonable Choice: Faceted Histograms

You could certainly have built a set of faceted histograms instead, but ideally, you'd have them arranged so that the distributions were easy to compare (the two histograms on top of each other, as these boxplots are, rather than just plotted next to each other.) That's part of the reason I flipped those boxplots. Here's our attempt.

```
ggplot(LBW_noNA,
       aes(x = pct_low_birthweight, fill = least_dev_f)) +
  geom_histogram(binwidth = 1, col = "white" ) +
  facet_grid(least_dev_f ~ ., labeller = "label_both") +
  guides(fill = FALSE) +
  labs(title = "Low Birth Weight % by Least Developed Nation Status",
       y = "Number of Nations",
       x = "Low Birth Weight %") +
  theme_bw()
```



This does convey a bit more effectively that the "least developed" nations comprise less than 20% (27/147) of the nations with low birth weight percentage estimates, but I think on the whole I prefer the boxplot here.

```
LBW_noNA %>% tabyl(least_dev_f)
```

```
 least_dev_f   n   percent
          No 120 0.8163265
         Yes  27 0.1836735
```

Question 6 - When is "more data" not necessarily a good thing?

We don't write answer sketches for essay questions. We're looking for a clear, coherent piece of writing, written in complete English sentences, that describes a relevant example effectively. We'll gather a few of the more interesting and enlightening responses, and share de-identified excerpts with the group after grading.

Question 7

Generate a "random" sample of 75 observations from a Normal distribution with mean 100 and standard deviation 10 using R. The `rnorm` function is likely to be helpful. Now, display a normal Q-Q plot of these data, using the `ggplot2` package from the `tidyverse`. How well does the Q-Q plot approximate a straight line?

Repeat this task for a second sample of 150 Normally distributed observations, again with a mean of 100 and a standard deviation of 10. Then repeat it again for samples of 25 and 225 Normally

distributed observations with a different mean and variance. Which of the four Q-Q plots you have developed better approximates a straight line and what should we expect the relationship of sample size with this phenomenon to be?

We're going to first draw a random sample of 75 observations from a Normal distribution with mean 100 and standard deviation 10.

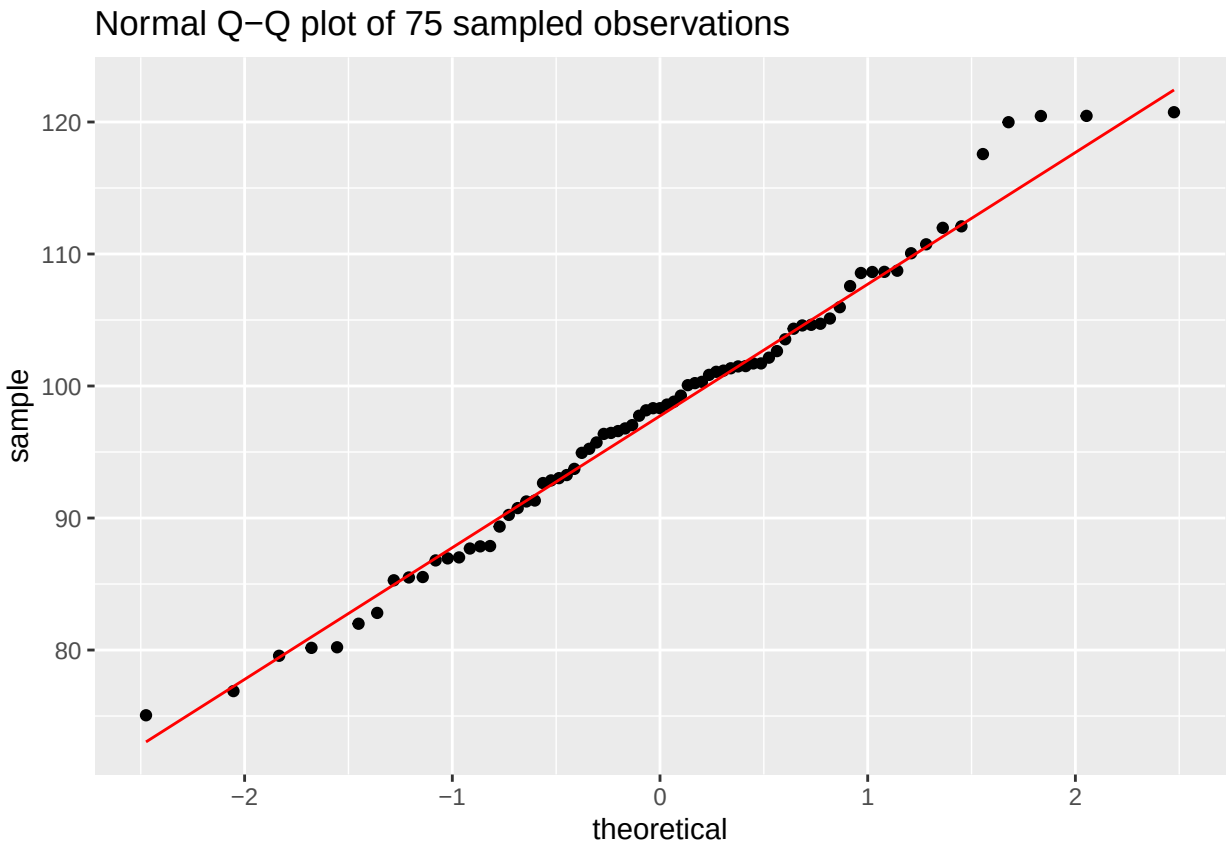
```
set.seed(20190920)
sample_75 <- rnorm(n = 75, mean = 100, sd = 10)
```

Then we'll put that sample into a tibble.

```
q7a <- tbl_df(sample_75)
```

Now we'll draw a Normal Q-Q plot of those data.

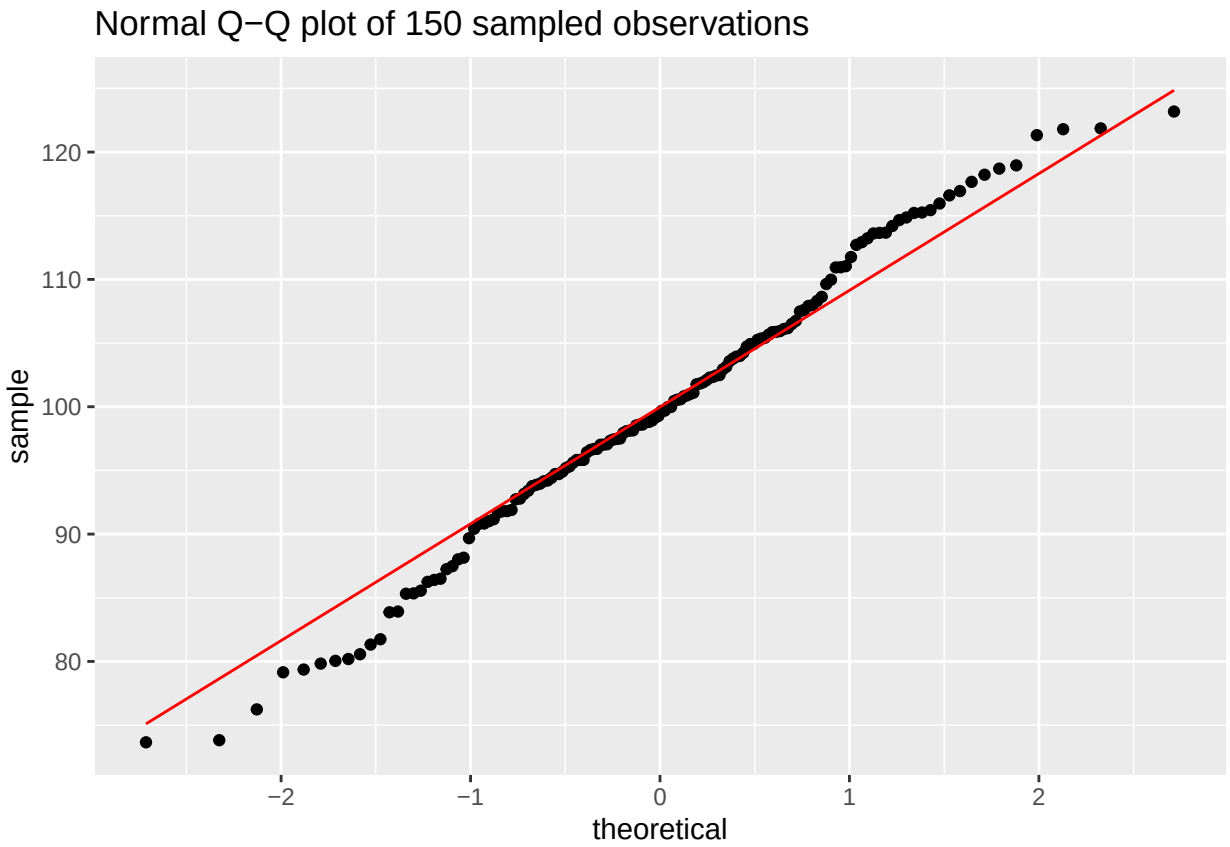
```
ggplot(q7a, aes(sample = sample_75)) +
  geom_qq() + geom_qq_line(col = "red") +
  labs(title = "Normal Q-Q plot of 75 sampled observations")
```



Now, we'll do this again for a new sample of 150 observations, also drawn from a Normal distribution with mean 100 and standard deviation 10.

```
sample_150 <- rnorm(n = 150, mean = 100, sd = 10)
q7b <- tbl_df(sample_150)

ggplot(q7b, aes(sample = sample_150)) +
  geom_qq() + geom_qq_line(col = "red") +
  labs(title = "Normal Q-Q plot of 150 sampled observations")
```



Next, we'll do this again for samples of first 25 and then 225 observations from a Normal distribution with a different mean (we'll use 400) and standard deviation (we'll use 100)

```
sample_25 <- rnorm(n = 25, mean = 400, sd = 100)
q7c <- tbl_df(sample_25)

sample_225 <- rnorm(n = 225, mean = 400, sd = 100)
q7d <- tbl_df(sample_225)
```

OK. So now we have all four samples. Let's put the plots all together in a single figure.

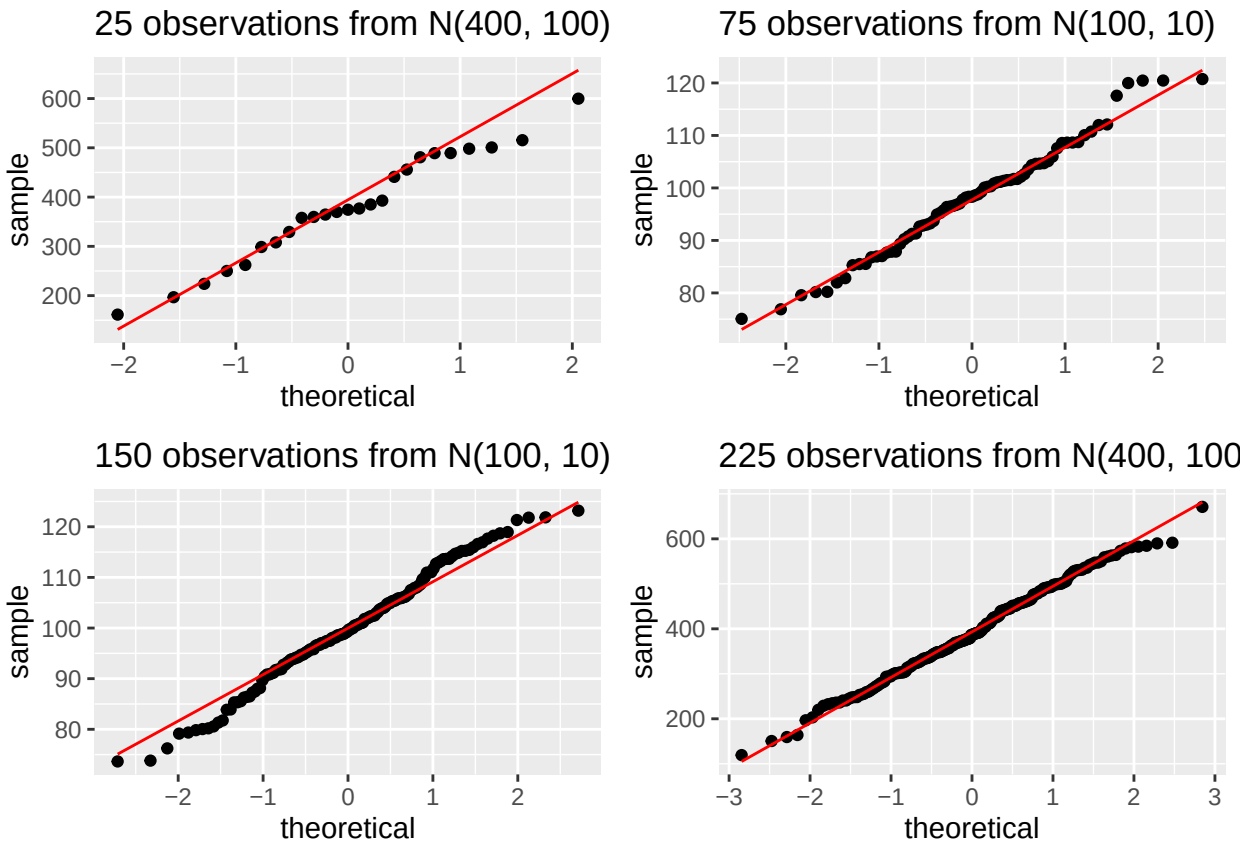
```
plot1 <- ggplot(q7c, aes(sample = sample_25)) +
  geom_qq() + geom_qq_line(col = "red") +
  labs(title = "25 observations from N(400, 100)")

plot2 <- ggplot(q7a, aes(sample = sample_75)) +
  geom_qq() + geom_qq_line(col = "red") +
  labs(title = "75 observations from N(100, 10)")

plot3 <- ggplot(q7b, aes(sample = sample_150)) +
  geom_qq() + geom_qq_line(col = "red") +
  labs(title = "150 observations from N(100, 10)")

plot4 <- ggplot(q7d, aes(sample = sample_225)) +
  geom_qq() + geom_qq_line(col = "red") +
  labs(title = "225 observations from N(400, 100)")
```

```
gridExtra::grid.arrange(plot1, plot2, plot3, plot4, ncol = 2)
```



All four of these plots show fairly modest deviations from what we would expect a Normal distribution to look like, usually in terms of showing a few outlying values.

With larger sample sizes, there's **no real reason** to assume that the plots will improve substantially in terms of eliminating outliers, in fact. Once we have at least 25 points (as in all of these cases) it appears that the results are fairly reasonable (in terms of suggesting that a Normal approximation is generally valid) in each of these plots.

On Grading Homework D

Your grade on Homework D is on a 0-100 scale.

General/Administrative (15 points)

- Award up to 10 points for turning the assignment on time (on time = within 1 hour of the deadline)
  - 10 points for both Markdown and Word/HTML/PDF in on time.
  - 4 points for one of Markdown, Word/HTML/PDF in on time.
  - 0 points if neither is in on time.
  - If a student hasn't submitted either the Markdown or Word/HTML/PDF piece, please identify and pester them via email until they do.
- Award an additional 5 points if there is an on-time answer provided for each of the 7 questions.
- Award zero points on the entire assignment to anyone whose first submission of the assignment is more than 4 hours late, unless excused from the assignment by Professor Love.

Question 1 (5 points)

- 5 points for a correct answer with code to indicate how they got it.
- 2 points for a correct answer but no indication of using code to determine.
- Otherwise, 0.

Question 2 (5 points)

- 5 points for a correct answer (name of each country and correct identification as "least developed" or not is required for a correct answer.)
- 3 points if they got 2/3 correct, or if they named all three countries but did not identify as least developed or not
- Otherwise, 0.

Question 3 (10 points)

Give 10 points if they do all five of the following:

- successfully built the histogram, as required,
- with the Normal distribution plotted as well,
- and labeled it correctly,
- and correctly interpreted it (as not fit well by a Normal model) using complete sentences,
- and answered the question about SD or IQR in a way that matches their interpretation (if they believed the data followed a Normal model well, then they should probably prefer SD or be agnostic, but not prefer IQR)

Drop 2 points for each of those five things that they didn't succeed in doing.

Question 4 (10 points)

Give 10 points if they do all five of the following:

- successfully built a Normal Q-Q plot using `ggplot2`

- include in the plot an appropriate diagonal line,
- correctly interpreted it in terms of Normal vs. Non-Normal using complete sentences
- correctly described what kind of non-Normality they saw (curve = skew, but it's not important for them to specify the direction unless of course they wrote left instead of the correct right skew)
- provided meaningful justification for their read of the plot with either another plot (like a histogram or boxplot) and a meaningful explanation in complete sentences, **or** with numerical summaries (like those I described in the sketch) and a meaningful explanation in complete sentences.

Drop 2 points for each of those five things that they didn't succeed in doing.

Question 5 (10 points)

- 10 points for producing a useful plot, likely a boxplot with or without the violin plot, or a reasonable and correct comparison of two histograms, that does actually report the data appropriately, and for building English sentences that conclude that the "least developed" nations had generally higher rates of low birthweight.
- 6 points for producing a useful plot, but not concluding that the "least developed" nations had generally higher rates of low birthweight.
- 0 points for a useless or incorrect plot.

Question 6 (30 points)

You need to identify (as a group) the 6-8 best essays (of the complete set of 60) that were read by the TAs (so that's choosing from the best two that each of you read, probably). In the Comments to Professor Love, please briefly identify the top 6-8 and specify the topic of these 6-8 best essays so I can read through them before returning them to the students, and select 2-4 to share.

- Award up to 10 points for proper grammar, use of citations, and appropriate length.
  - Take off 5 points if the grammar is consistently poor.
  - Take off 2 points if no citations are used
  - Take off 3 points if the essay does not meet the required length of 200-400 words.
- Award up to 10 points for describing an "example in your own field/work/experience where a *surplus* of information made (or makes) it easier for people dealing with a complex system to cherry-pick information that supports their prior positions."
  - A run of the mill "good" example should receive 7-9 points on this. Reserve 10 points on this for an absolutely excellent essay that was a real pleasure to read.
  - Award no more than 5 points if there is no specific example given.
  - Take off all 10 points if the essay prompt was not followed at all, or the essay is completely off topic.
- Award up to 10 points for describing the implications/lessons learned from the given example.
  - A run of the mill "good" description should receive 7-9 points on this. Reserve 10 points for an absolutely excellent essay.
  - Take off 5 points if no lessons/implications are provided.
  - Take off all 10 points if the essay prompt was not followed at all, or the essay is completely off topic.

The best few essays overall might receive a grade between 28 and 30, but I expect most essays to score between 20 and 26 points. We will just provide the total score on the essay, but the TAs will provide a comment to the students regarding every single essay, regardless of its score. Any score of 21 or higher indicates a fairly good essay, and anything 27 or higher indicates a really excellent one.

Question 7 (15 points)

1. Award 5 points for making the random samples and the four Q-Q plots.

- 3 points for just random samples, but no QQ plots or vice versa.
- 2 points if only the first random sample and QQ plot are completed.

2. Award 10 additional points for answering both questions fully, noting that the plots show only modest deviations from what is expected from a Normal distribution (in my case due to outliers), and for noting that all the plots with 25 points or more approximate a straight line fairly well.

- Deduct 5 points for not getting the point that they are all about equally good, unless their choice of random number was very unfortunate (in which case they should notice whatever they saw in their simulation.)
- Deduct 5 points for only answering one of the two questions posed.
- Deduct 2 points for neglecting to change the mean and SD in the final two plots.
- Deduct 3 points (but don't take this question below 0 points) if they neglected to set a seed for the random numbers. They can use any seed they like.