The normal distribution

This data set contains data on 515 menu items from some of the most popular fast food restaurants worldwide. Let's take a quick peek at the first few rows of the data.

```
head(fastfood)
```

```
## # A tibble: 6 x 17
##   restaurant item  calories cal_fat total_fat sat_fat trans_fat cholesterol
##   <chr>      <chr>    <dbl>   <dbl>     <dbl>   <dbl>     <dbl>       <dbl>
## 1 Mcdonalds  Arti~      380      60         7       2       0            95
## 2 Mcdonalds  Sing~      840     410        45      17       1.5         130
## 3 Mcdonalds  Doub~     1130     600        67      27       3           220
## 4 Mcdonalds  Gril~      750     280        31      10       0.5         155
## 5 Mcdonalds  Cris~      920     410        45      12       0.5         120
## 6 Mcdonalds  Big ~      540     250        28      10       1            80
## # ... with 9 more variables: sodium <dbl>, total_carb <dbl>, fiber <dbl>,
## #   sugar <dbl>, protein <dbl>, vit_a <dbl>, vit_c <dbl>, calcium <dbl>,
## #   salad <chr>
```

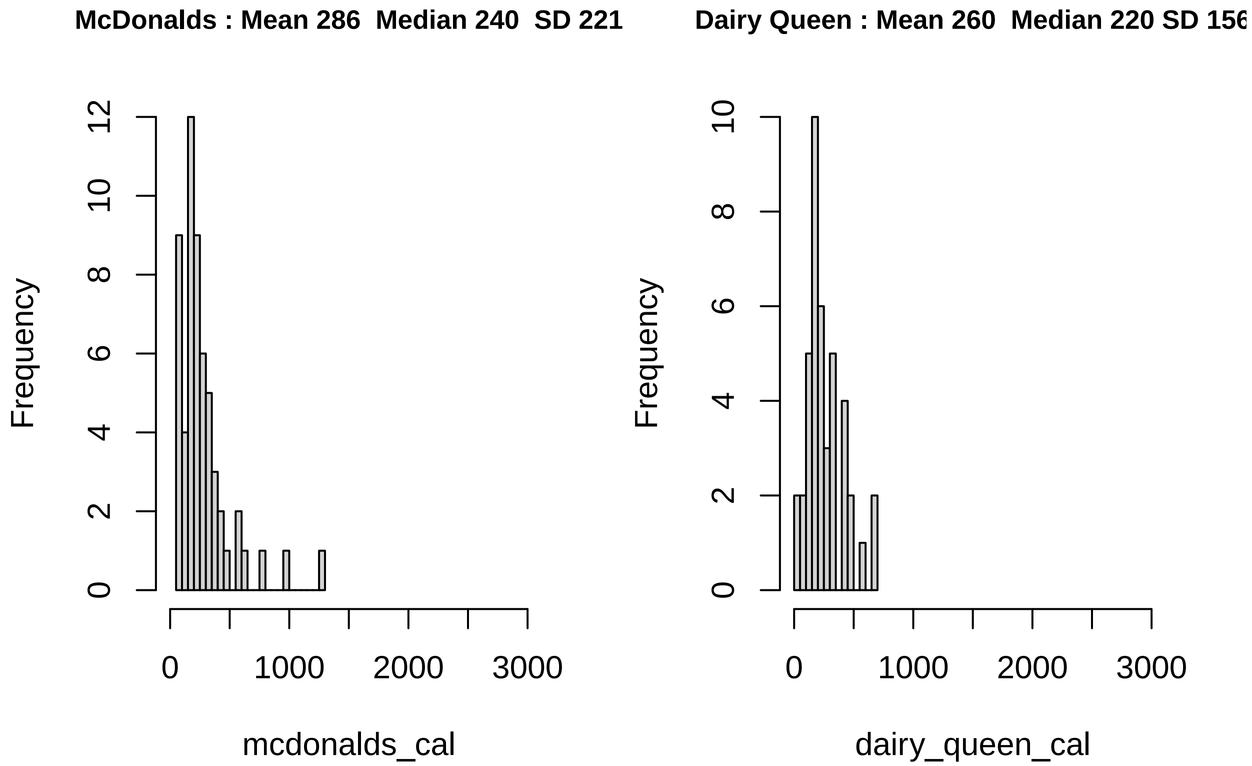
Let's focus on McDonalds and Dairy Queen.

```
mcdonalds <- fastfood %>%
  filter(restaurant == "Mcdonalds")
dairy_queen <- fastfood %>%
  filter(restaurant == "Dairy Queen")
```

1. Make a plot (or plots) to visualize the distributions of the amount of calories from fat of the options from these two restaurants. How do their centers, shapes, and spreads compare?

McDonalds has a wider range, at least it sells a few items with very high caloric content. Both distributions would be considered mutli-modal since there are several bins with low frequencies. So the variance is high but at the same time there is a middle "bulge" that reflects that mean, median, mode are all close.

```
# set up a side by side comparison

mcdonalds_cal<-mcdonalds$cal_fat
dairy_queen_cal<-dairy_queen$cal_fat

mcdonalds_stats<-sprintf("McDonalds : Mean %.0f  Median %.0f  SD %.0f",
                         mean(mcdonalds_cal), median(mcdonalds_cal), sd(mcdonalds_cal) )
dairy_queen_stats<-sprintf("Dairy Queen : Mean %.0f  Median %.0f SD %.0f",
                           mean(dairy_queen_cal), median(dairy_queen_cal), sd(dairy_queen_cal) )


par(mfrow=c(1,2))
hist(mcdonalds_cal, main=mcdonalds_stats, breaks=20, xlim = c(1,3000), cex.main = .8 )
hist(dairy_queen_cal, main=dairy_queen_stats, breaks=20, xlim = c(1,3000), cex.main = .8)
```



The normal distribution

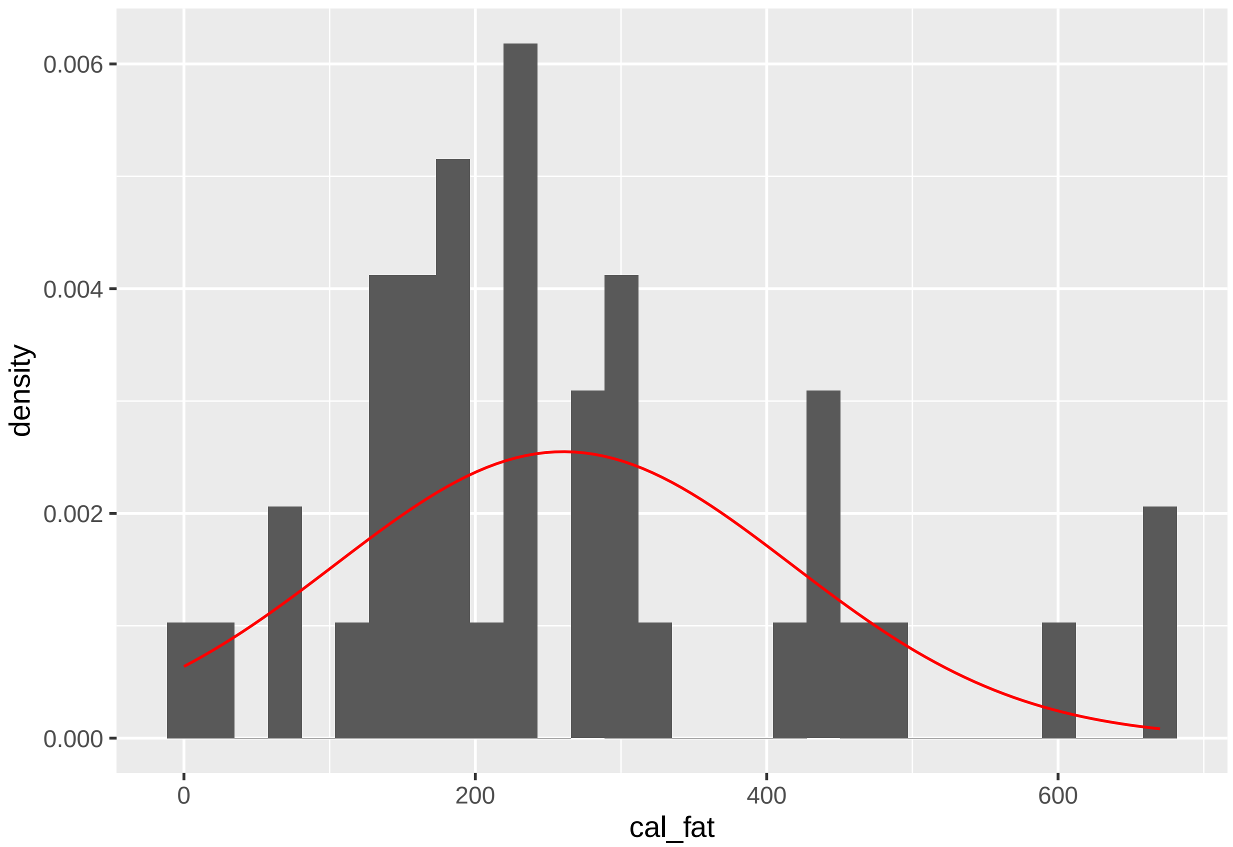
save off the mean and standard deviation

```
dqmean <- mean(dairy_queen$cal_fat)
dqsd   <- sd(dairy_queen$cal_fat)
```

make a density histogram use the `lines` function to overlay a normal probability curve.

note in a density histogram the *areas* of the bars add up to 1. The area of each bar - height *times* the width of the bar.

```
ggplot(data = dairy_queen, aes(x = cal_fat)) +
        geom_blank() +           # initialize a blank plot on top of dairy_queen
        geom_histogram(aes(y = ..density..)) +         # note fun = dnorm
        stat_function(fun = dnorm, args =
                        c(mean = dqmean, sd = dqsd), col = "red")
```



`dnorm`. specifies that the curve to have the same mean and std.dev as dairy_queen$cal_fat

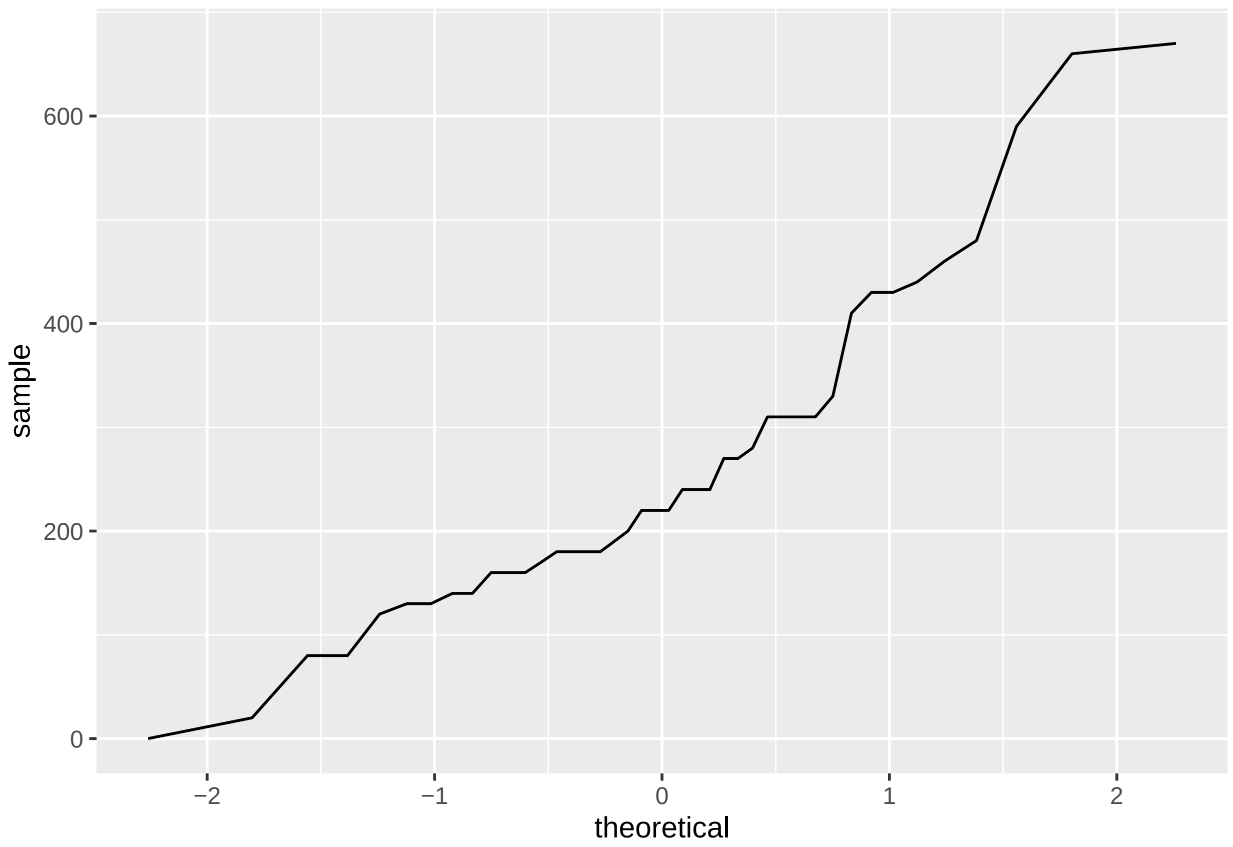
Based on the this plot, does it appear that the data follow a nearly normal distribution?

It peaks where the red line peaks, but I think there are too many gaps to call it a normal distribution.

Evaluating the normal distribution

lets construct a normal probability plot, also called a normal Q-Q plot for "quantile-quantile".

```
ggplot(data = dairy_queen, aes(sample = cal_fat)) +
  geom_line(stat = "qq")
```



This time, you can use the `geom_line()` layer, while specifying that you will be creating a Q-Q plot with the `stat` argument. It's important to note that here, instead of using `x` instead `aes()`, you need to use `sample`.

The x-axis values correspond to the quantiles of a theoretically normal curve with mean 0 and standard deviation 1 (i.e., the standard normal distribution). The y-axis values correspond to the quantiles of the original unstandardized sample data. However, even if we were to standardize the sample data values, the Q-Q plot would look identical. A data set that is nearly normal will result in a probability plot where the points closely follow a diagonal line. Any deviations from normality leads to deviations of these points from that line.

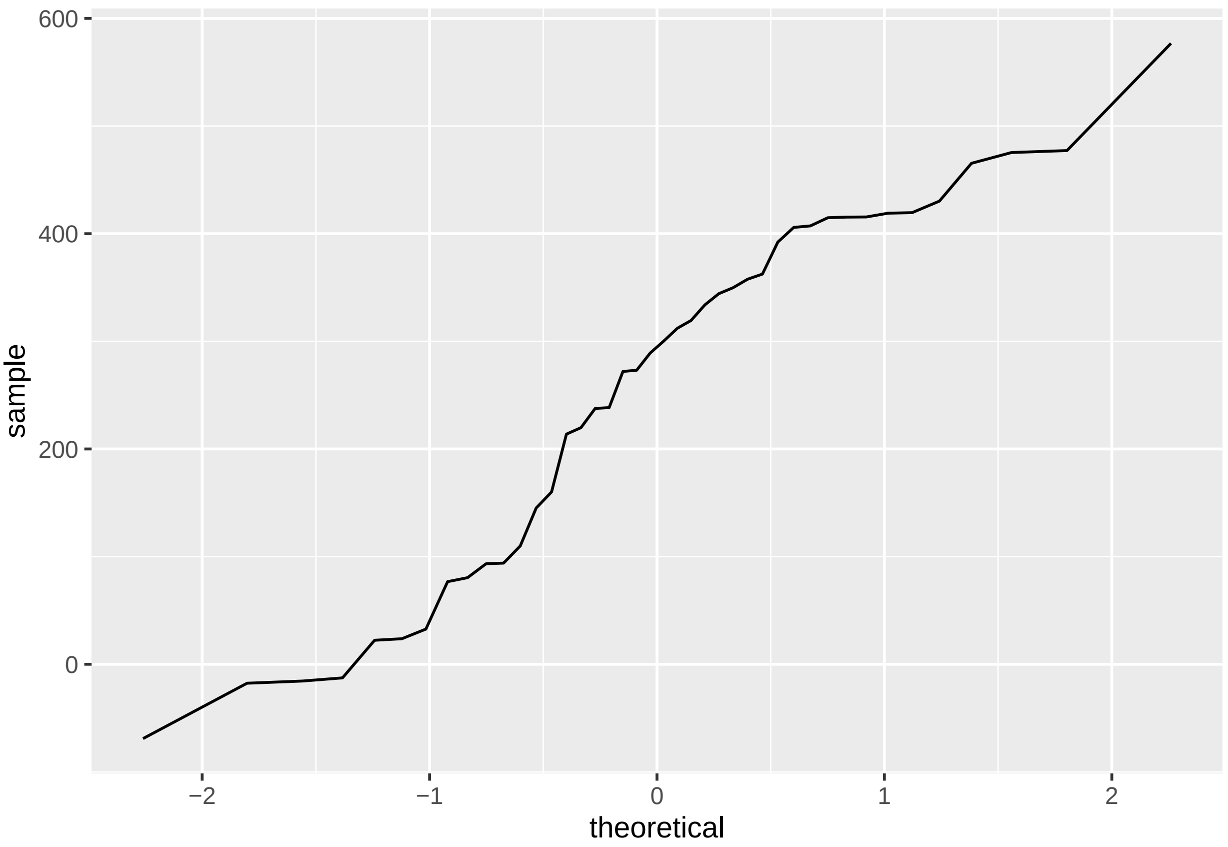
The plot for Dairy Queen's calories from fat shows points that tend to follow the line but with some errant points towards the upper tail. You're left with the same problem that we encountered with the histogram above: how close is close enough?

A useful way to address this question is to rephrase it as: what do probability plots look like for data that I *know* came from a normal distribution? We can answer this by simulating data from a normal distribution using `rnorm`.

```
sim_norm <- rnorm(n = nrow(dairy_queen), mean = dqmean, sd = dqsd)
```

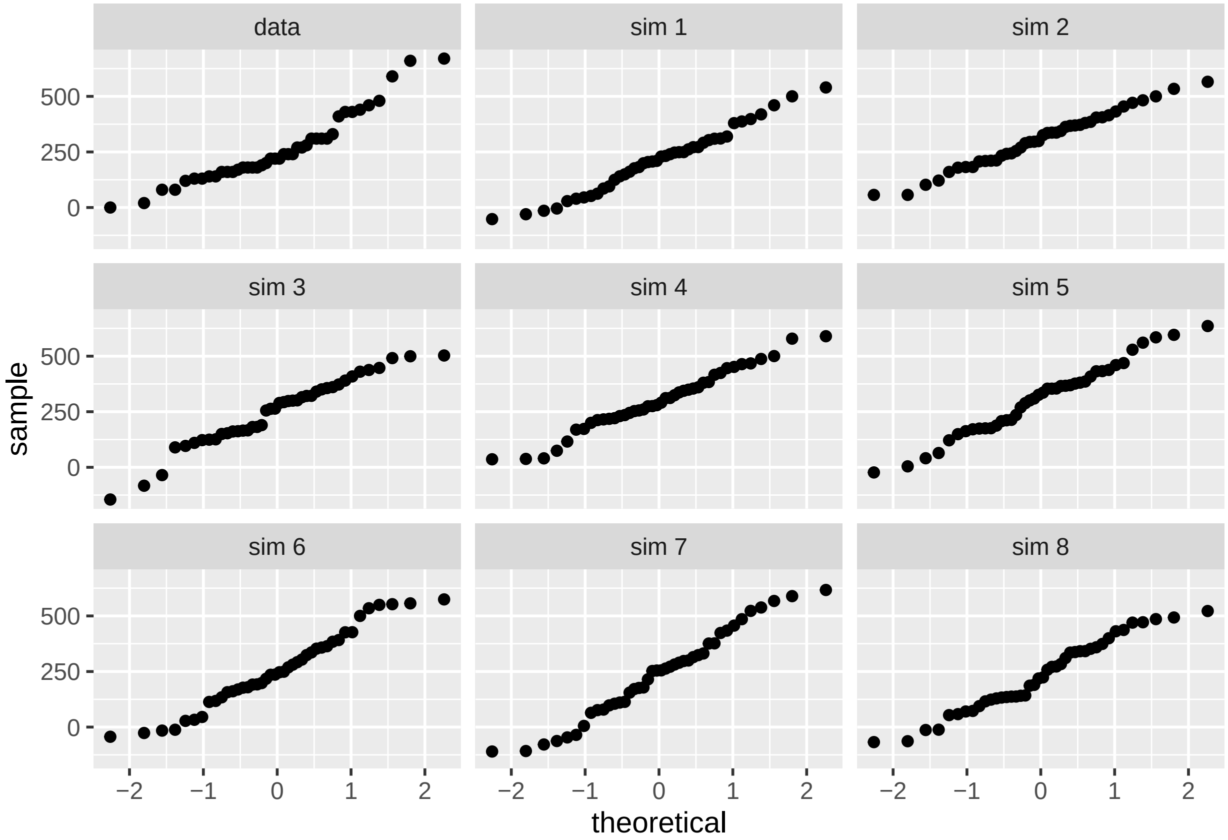
3. Make a normal probability plot of `sim_norm`. Do all of the points fall on the line? How does this plot compare to the probability plot for the real data? (Since `sim_norm` is not a dataframe, it can be put directly into the `sample` argument and the `data` argument can be dropped.)

```
sim_norm_df<-as.data.frame(sim_norm)

ggplot(data=sim_norm_df, aes(sample = sim_norm)) +
  geom_line(stat = "qq")
```



Even better than comparing the original plot to a single plot generated from a normal distribution is to compare it to many more plots using the following function. It shows the Q-Q plot corresponding to the original data in the top left corner, and the Q-Q plots of 8 different simulated normal data. It may be helpful to click the zoom button in the plot window.

```
dairy_queen_cal_df<-as.data.frame(dairy_queen_cal)
colnames(dairy_queen_cal_df)<-c("cals")
qqnormsim(cals, dairy_queen_cal_df)
```

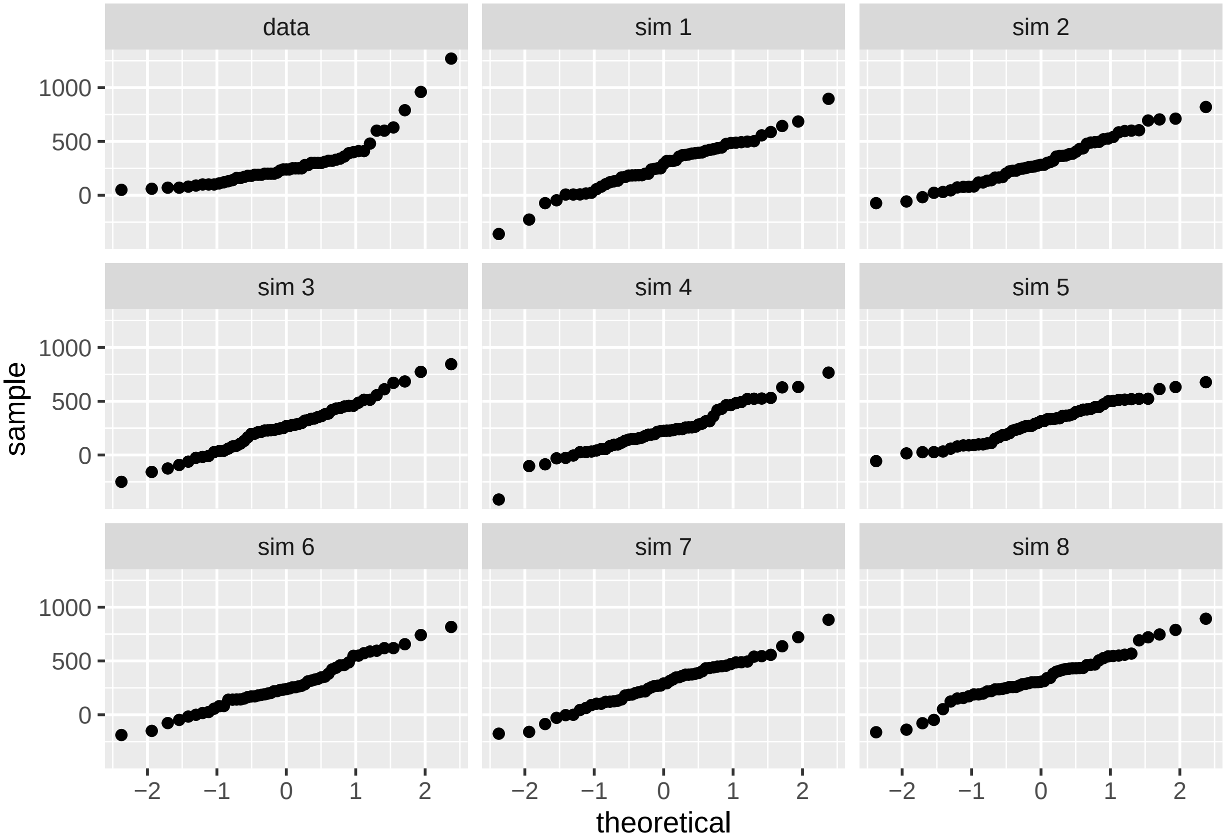


Do the plots provide evidence that the dairy queen calories are nearly normal?

If the defintion of nearly normal is A probability plot where the points closely follow a diagonal line. then I suppose it would be nearly normal.

5. Using the same technique, determine whether or not the calories from McDonald's menu appear to come from a normal distribution.

```
mcdonalds_cal_df<-as.data.frame(mcdonalds_cal)
colnames(mcdonalds_cal_df)<-c("cals")
qqnormsim(cals, mcdonalds_cal_df)
```



McDonalds had some high caloric items, so it makes sense that some of these plots refelct the outliers.

Normal probabilities

if you know a random variable is approximately normal, you can answer all sorts of questions For example, "What is the probability that a randomly chosen Dairy Queen product has more than 600 calories from fat?"

If we assume that the calories from fat from Dairy Queen's menu are normally distributed (a very close approximation is also okay), we can find this probability by calculating a Z score and consulting a Z table (also called a normal probability table). In R, this is done in one step with the function `pnorm()`.

```
# note : lower.tail=FALSE is another way to get the over probability
1 - pnorm(q = 600, mean = dqmean, sd = dqsd)
```

```
## [1] 0.01501523
```

calculate probability empirically (2/42 = .476)

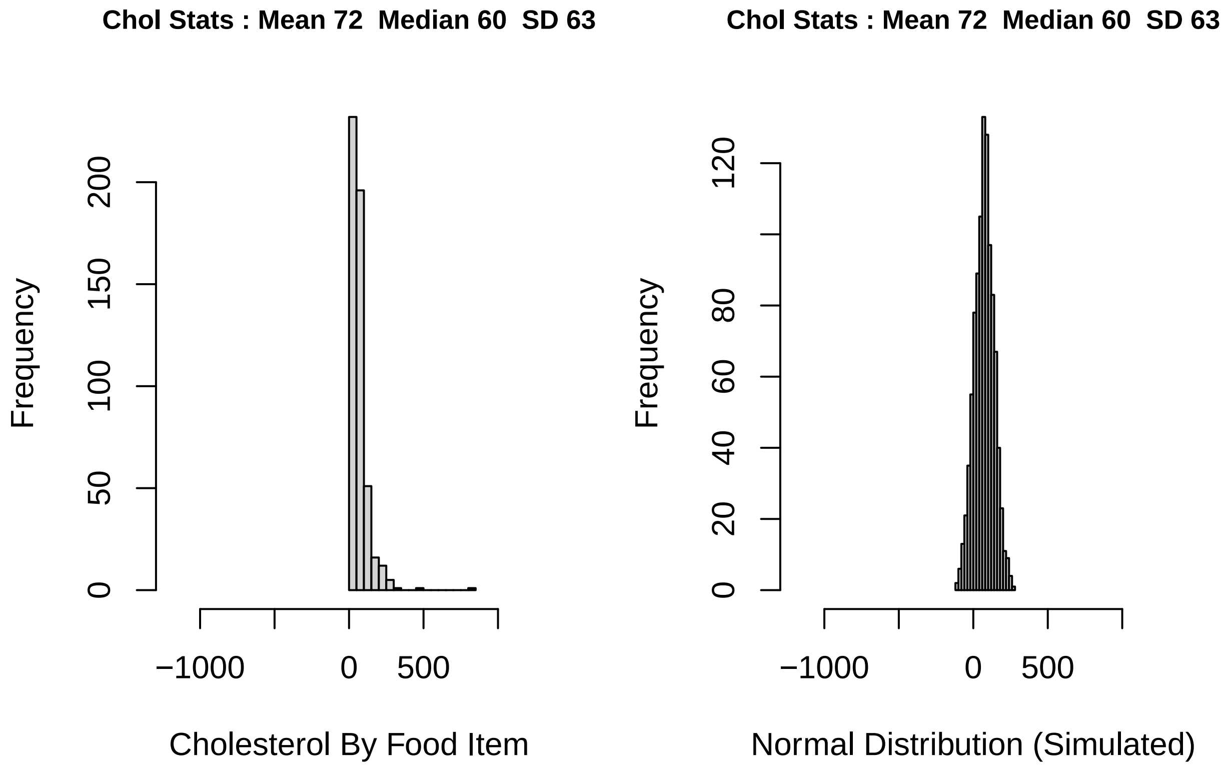
```
dairy_queen %>%
  filter(cal_fat > 600) %>%
  summarise(percent = n() / nrow(dairy_queen))
```

```
## # A tibble: 1 x 1
##   percent
##     <dbl>
## 1  0.0476
```

6. Write out two probability questions that you would like to answer about any of the restaurants in this dataset. Calculate those probabilities using both the theoretical normal distribution as well as the empirical distribution (four probabilities in all). Which one had a closer agreement between the two methods?

The doctor says I should avoid cholesterol. Can I see the total cholestoral numbers, visually, and how normal is that ? What is probability of unwittingly ordering an item with a cholestoral count over 200.

```
chol_risk_level=200             # our tolerance

chol_mean<-mean(fastfood$cholesterol)
chol_median<-median(fastfood$cholesterol)
chol_sd<-sd(fastfood$cholesterol)

chol_df<-as.data.frame(fastfood$cholesterol)


chol_stats<-sprintf("Chol Stats : Mean %.0f  Median %.0f  SD %.0f",
                         chol_mean, chol_median, chol_sd )





par(mfrow=c(1,2))


chol_norm_df<-as.data.frame(rnorm(1000, mean = chol_mean, sd = chol_sd))

colnames(chol_norm_df)<-c("cholesterol")

par(mfrow=c(1,2))
hist(chol_df$`fastfood$cholesterol`, main=chol_stats,cex.main = .8,
     breaks=20, xlim = c(-1200,1200), xlab="Cholesterol By Food Item")
hist(chol_norm_df$cholesterol, main=chol_stats, cex.main = .8,
     breaks=20, xlim = c(-1200,1200), xlab = "Normal Distribution (Simulated)")
```



Well the actual distribution has almost no left tail and there are some outliers to the right so the standard deviation nearly doubles the mean. This worries me.

```
# pnorm(chol_mean+chol_sd, mean = chol_mean, sd = chol_sd)   # 0.841 just checking

risk_on_norm<-pnorm(250, mean = chol_mean,
                    sd = chol_sd, lower.tail = FALSE)

risk_on_empirical=nrow(subset(subset(fastfood, cholesterol >
                                        chol_risk_level)))/nrow(fastfood)

sprintf("The probability of ordering a high cholestoral food")
```

```
## [1] "The probability of ordering a high cholestoral food"
```

```
sprintf("if our data was normalized would be %.2f%s", risk_on_norm * 100,'%')
```

```
## [1] "if our data was normalized would be 0.25%"
```

```
sprintf("However the actual probability is %.2f%s", risk_on_empirical * 100,'%')
```

```
## [1] "However the actual probability is 3.88%"
```

More Practice

7. Now let's consider some of the other variables in the dataset. Out of all the different restaurants, which ones' distribution is the closest to normal for sodium?

Lets first look at the mean, median, standard deviations of each.

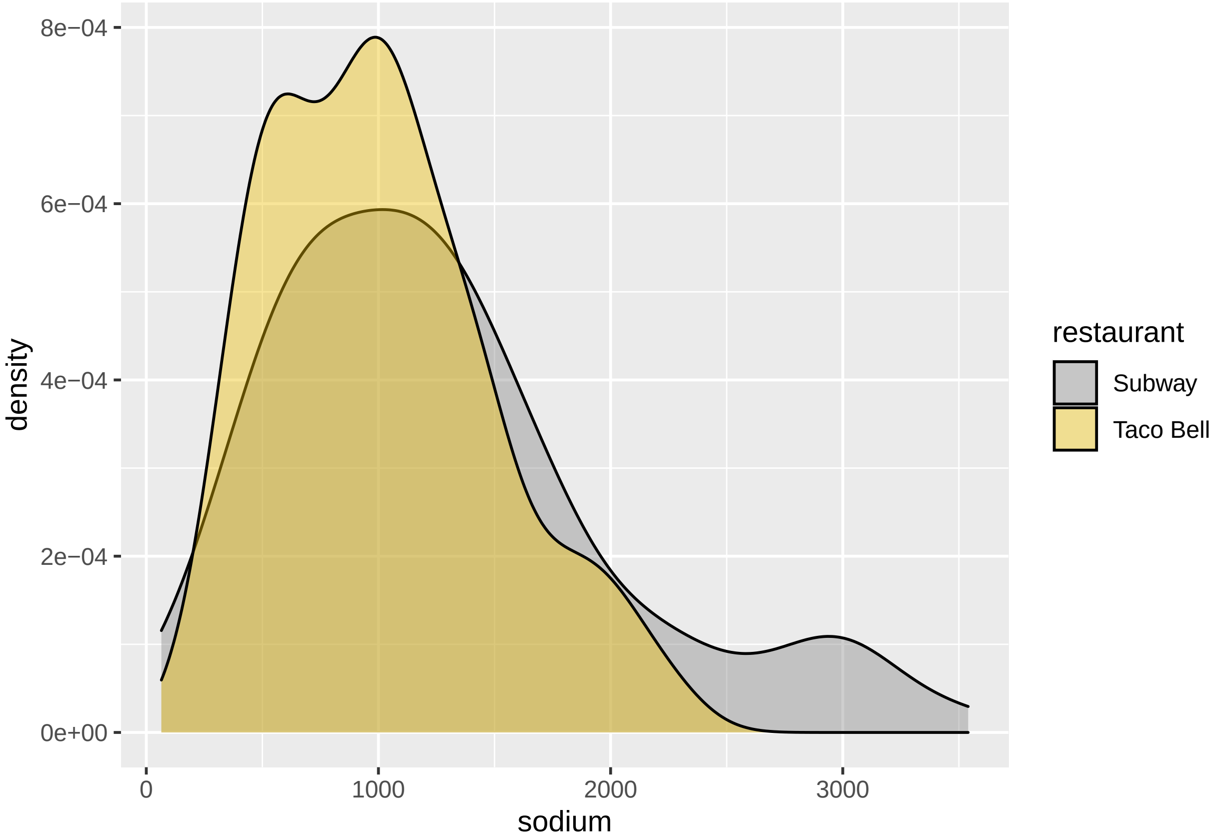
```
sodium_mean=mean(fastfood$sodium)
sodium_median=median(fastfood$sodium)
sodium_sd=sd(fastfood$sodium)



# sprintf("Sodium Stats : Mean %.0f  Median %.0f  SD %.0f", sodium_mean, sodium_median, sodium_sd )


# isolate the restaurants
each_restaurant<-as.factor(unique(fastfood$restaurant))

# create a data frame to hold the stats
rest_stats_df <- data.frame(rest="All",mean=sodium_mean,
                            median=sodium_median, sd=sodium_sd)


for (f in 1:length(each_restaurant)) {
  rest<-as.character(each_restaurant[f])


  rest_data<-subset(fastfood, restaurant==rest)
  rest_sd<-sd(rest_data$sodium)
  rest_median<-median(rest_data$sodium)
  rest_mean<-mean(rest_data$sodium)

  rest_stats_df<-rbind(rest_stats_df,data.frame(rest=rest,
                                                mean=rest_mean, median=rest_median, sd=rest_sd))


}

knitr::kable(rest_stats_df, caption='Distribution Metrics')
```

Table 1: Distribution Metrics

rest	mean	median	sd
All	1246.738	1110	689.9543
Mcdonalds	1437.895	1120	1036.1721
Chick Fil-A	1151.481	1000	726.9203
Sonic	1350.755	1250	665.1340
Arbys	1515.273	1480	663.6651
Burger King	1223.571	1150	499.8841
Dairy Queen	1181.786	1030	609.9398
Subway	1272.969	1130	743.6346
Taco Bell	1013.913	960	474.0544

Taco Bell and Subway have the largest mean to sd ration, which makes them a little more normal than the others. Now plot Taco Bell and Subway together.

```
low_sodium_restaurants<-subset(fastfood, restaurant=="Subway" | restaurant=="Taco Bell")

  a <- ggplot(low_sodium_restaurants, aes(x = sodium))

  # Create 2 curves on one figure, the fill parameter and
  #     the 2 scale functions control the 2 curves
a + geom_density(aes(fill = restaurant), alpha = 0.4) +
   scale_color_manual(values = c("#868686FF", "#EFC000FF"))+
  scale_fill_manual(values = c("#868686FF", "#EFC000FF"))
```



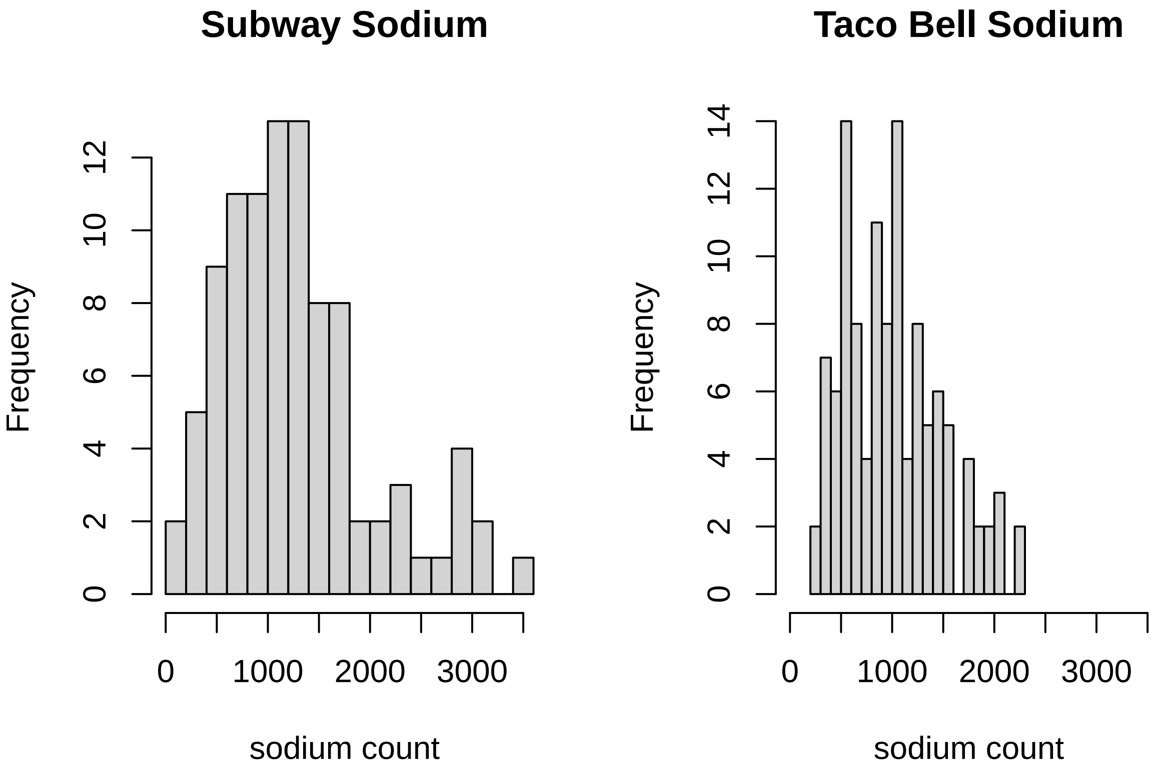
8. Note that some of the normal probability plots for sodium distributions seem to have a stepwise pattern. why do you think this might be the case?

The histograms do a better job at revealing the choppiness of the data so see below.

I can only speculate but perhaps there are 2 kinds of consumers, one who wants a modest meal and one who wants a really big meal.

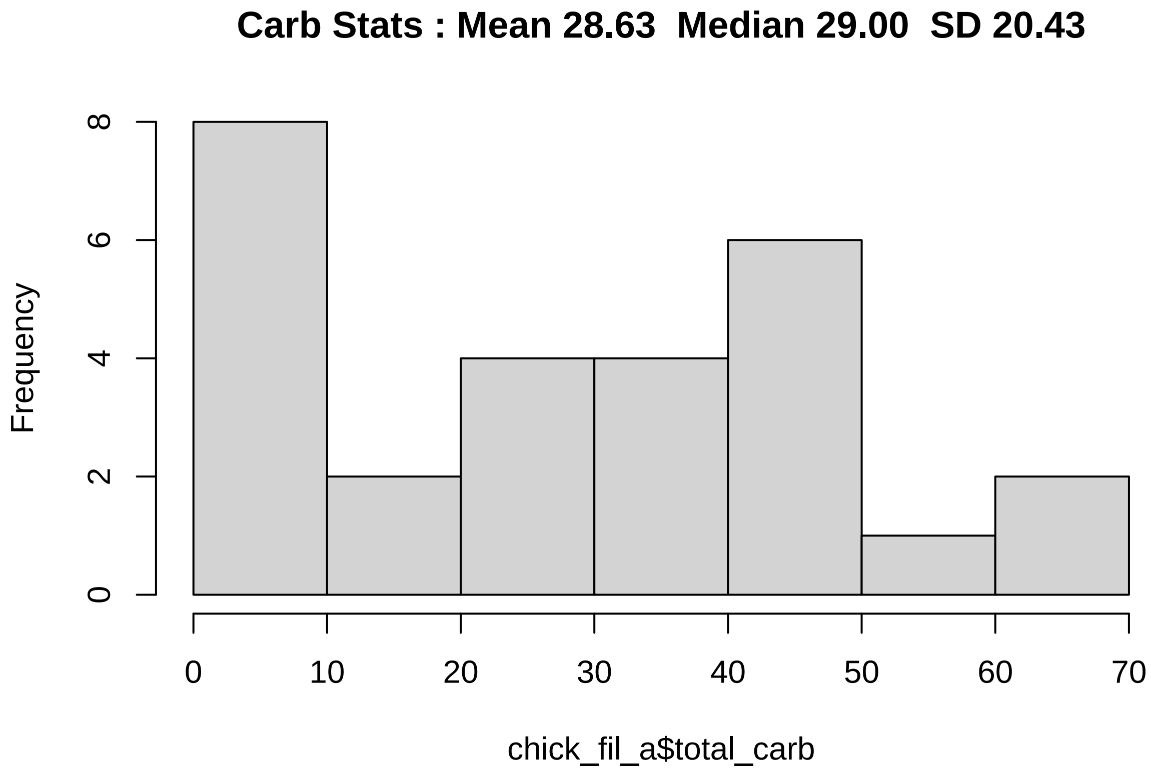
Also if you look at Taco Bell it is up and down a lot. In that regard, I would say there arent enough items and there arent enough ingredients in those items to produce a smooth curve.

```
par(mfrow=c(1,2))
subway<-subset(fastfood, restaurant=="Subway")
taco_bell<-subset(fastfood, restaurant=="Taco Bell")
hist(subway$sodium, main="Subway Sodium", breaks=20, xlim = c(1,3500), xlab="sodium count")
hist(taco_bell$sodium, main="Taco Bell Sodium", breaks=20, xlim = c(1,3500), xlab="sodium count")
```



9. As you can see, normal probability plots can be used both to assess normality and visualize skewness. Make a normal probability plot for the total carbohydrates from a restaurant of your choice. Based on this normal probability plot, is this variable left skewed, symmetric, or right skewed?
Use a histogram to confirm your findings.

```
par(mfrow=c(1,1))
chick_fil_a<-subset(fastfood, restaurant=="Chick Fil-A")


cfa_carb_mean=mean(chick_fil_a$total_carb)
cfa_carb_median=median(chick_fil_a$total_carb)
cfa_carb_sd=sd(chick_fil_a$total_carb)

cfa_title<-sprintf("Carb Stats : Mean %.2f  Median %.2f  SD %.2f", cfa_carb_mean, cfa_carb_median, cfa_

hist(chick_fil_a$total_carb, main=cfa_title, breaks=5)
```



According to this data, 4 chicken nuggets has 1 carb. That doesnt sound right to me Anyway the plot is right skewed due to 8 different nugget items having less than 10 carbs.