The normal distribution

Koohayr Pooladvand

In this lab, you'll investigate the probability distribution that is most central to statistics: the normal distribution. If you are confident that your data are nearly normal, that opens the door to many powerful statistical methods. Here we'll use the graphical tools of R to assess the normality of our data and also learn how to generate random numbers from a normal distribution.

Getting Started

Load packages

In this lab, we will explore and visualize the data using the **tidyverse** suite of packages as well as the **openintro** package.

Let's load the packages.

```
library(tidyverse)
library(openintro)
```

The data

This week you'll be working with fast food data. 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.

Either you can use `glimpse` like before, or `head` to do this.

```
library(tidyverse)
library(openintro)
data("fastfood", package='openintro')
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  Artisan G~      380      60         7       2       0            95
## 2 Mcdonalds  Single Ba~      840     410        45      17       1.5         130
## 3 Mcdonalds  Double Ba~     1130     600        67      27       3           220
## 4 Mcdonalds  Grilled B~      750     280        31      10       0.5         155
## 5 Mcdonalds  Crispy Ba~      920     410        45      12       0.5         120
## 6 Mcdonalds  Big Mac          540     250        28      10       1            80
## # i 9 more variables: sodium <dbl>, total_carb <dbl>, fiber <dbl>, sugar <dbl>,
## #   protein <dbl>, vit_a <dbl>, vit_c <dbl>, calcium <dbl>, salad <chr>
```

You'll see that for every observation there are 17 measurements, many of which are nutritional facts.

You'll be focusing on just three columns to get started: restaurant, calories, calories from fat.

Let's first focus on just products from McDonalds and Dairy Queen.

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

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?

Insert your answer here

#1: Koohyar's answer

First let's plot McDonals and then we can plot dairy queen

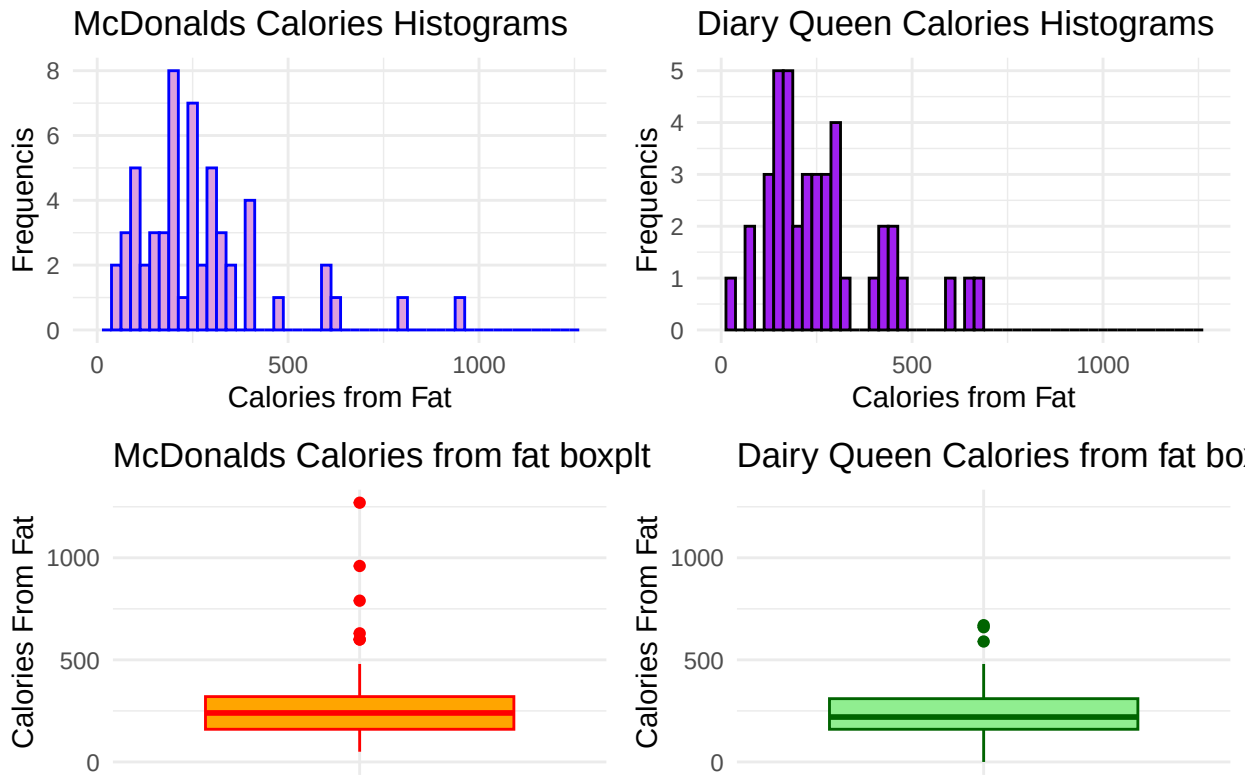
```
library(ggplot2)
#install.packages("gridExtra")
library(gridExtra)

#find max value to us i nthe graph below
max_value <-  max(max(dairy_queen$cal_fat),max(mcdonalds$cal_fat))

# Create a histogram using ggplot2
his_mc_p1 <- mcdonalds |> ggplot(aes(x = cal_fat)) +
  geom_histogram(binwidth= 25, fill = "plum", color = "blue", position = "dodge") +
  labs(title = "McDonalds Calories Histograms", x = "Calories from Fat", y = "Frequencis") +
  theme_minimal() +
  xlim (0, max_value)

box_mc_p2 <- mcdonalds|>ggplot(aes( x = "", y = cal_fat)) +
  geom_boxplot( fill = "orange", color = "red") +
  labs (title = "McDonalds Calories from fat boxplt", x = "", y = "Calories From Fat") + theme_minimal()
  ylim (0, max_value)

#print some data statistic for the MC
mcdonalds %>% select(cal_fat) %>%
  summarize (
    mc_sd       =  sd(cal_fat),
    mc_mean   = mean(cal_fat),
    mc_iqr    = IQR(cal_fat)
    )
```

```
## # A tibble: 1 x 3
##   mc_sd mc_mean mc_iqr
##   <dbl>   <dbl>  <dbl>
## 1  221.    286.    160
```

```r
# Create a histogram using ggplot2
his_dq_p1 <- dairy_queen |> ggplot(aes(x = cal_fat)) +
  geom_histogram(binwidth= 25, fill = "purple", color = "black", position = "dodge") +
  labs(title = "Diary Queen Calories Histograms", x = "Calories from Fat", y = "Frequencis") +
  theme_minimal() +
  xlim (0, max_value)

box_dq_p2 <- dairy_queen|>ggplot(aes( x = "", y = cal_fat)) +
  geom_boxplot( fill = "lightgreen", color = "darkgreen") +
  labs (title = "Dairy Queen Calories from fat boxplt", x = "", y = "Calories From Fat") + theme_minimal
  ylim (0, max_value)

#print some data statistic for the MC
dairy_queen %>% select(cal_fat) %>%
  summarize (
    dq_sd       =  sd(cal_fat),
    dq_mean   = mean(cal_fat),
    dq_iqr    = IQR(cal_fat)
    )
```

```
## # A tibble: 1 x 3
##   dq_sd dq_mean dq_iqr
##   <dbl>   <dbl>  <dbl>
## 1  156.    260.    150
```

```r
gridExtra::grid.arrange(his_mc_p1,his_dq_p1, box_mc_p2, box_dq_p2, nrow = 2, ncol = 2)
```



I used **gridExtra** to plot histograms close to the box plot to be able to see the average and outliers as well.

Looking into the plots, it seems the average for both is close to around **250 calories in fat**, but the range of McDonald's extends up to a maximum of **1270 calories**, while DQ stays below **700 calories**.

DQ shows a betetr bell shape distribution compared to McDonalds.

On the other hand, McDonald's distribution is skewed to the left with more outliers compared to DQ. DQ has two outliers, but McDonald's has about five.

Both distributions do not appear to be normal, but DQ shows a better and closer histogram to a normal distribution.

The normal distribution

In your description of the distributions, did you use words like *bell-shaped*or *normal*? It's tempting to say so when faced with a unimodal symmetric distribution.

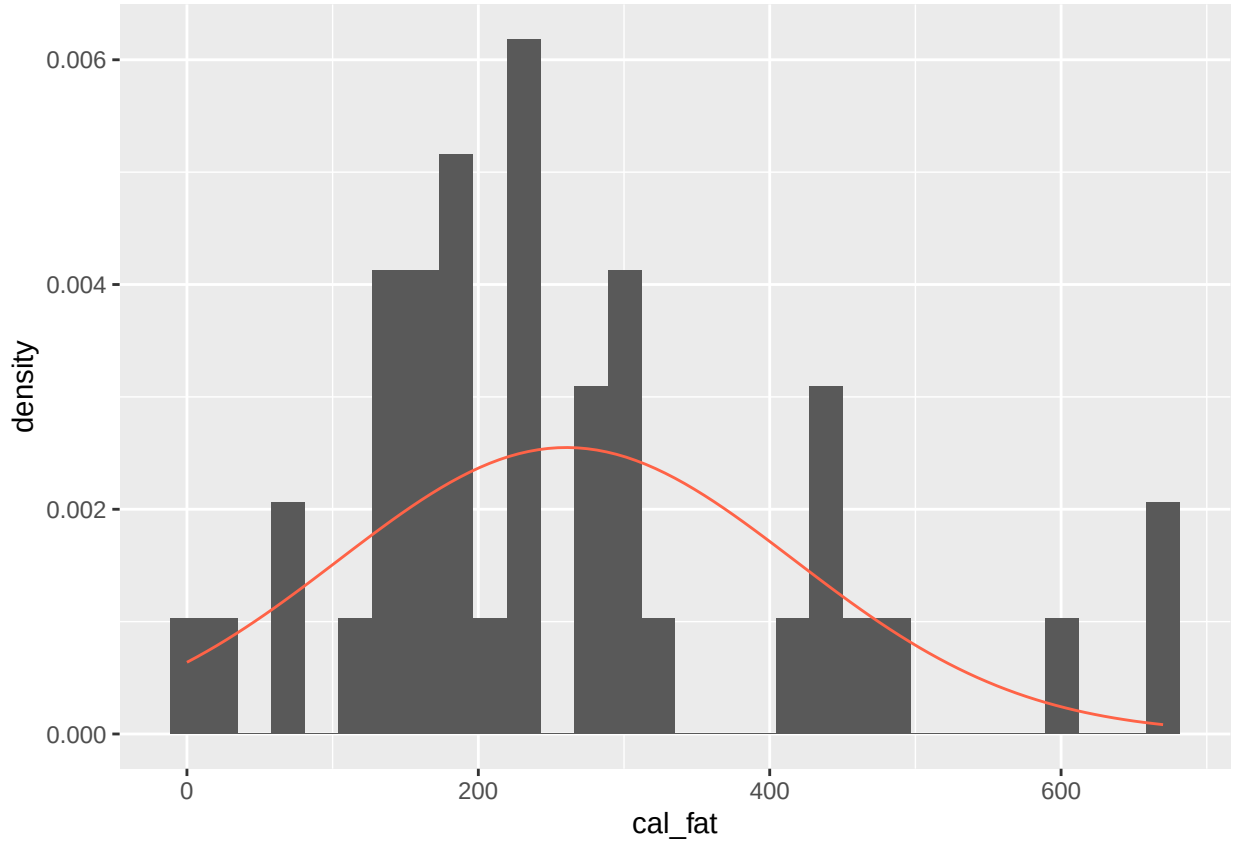
To see how accurate that description is, you can plot a normal distribution curve on top of a histogram to see how closely the data follow a normal distribution. This normal curve should have the same mean and standard deviation as the data. You'll be focusing on calories from fat from Dairy Queen products, so let's store them as a separate object and then calculate some statistics that will be referenced later.

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

Next, you make a density histogram to use as the backdrop and use the `lines` function to overlay a normal probability curve. The difference between a frequency histogram and a density histogram is that while in

a frequency histogram the *heights* of the bars add up to the total number of observations, in a density histogram the *areas* of the bars add up to 1. The area of each bar can be calculated as simply the height *times* the width of the bar. Using a density histogram allows us to properly overlay a normal distribution curve over the histogram since the curve is a normal probability density function that also has area under the curve of 1. Frequency and density histograms both display the same exact shape; they only differ in their y-axis. You can verify this by comparing the frequency histogram you constructed earlier and the density histogram created by the commands below.

```
ggplot(data = dairy_queen, aes(x = cal_fat)) +
        geom_blank() +
        geom_histogram(aes(y = ..density..)) +
        stat_function(fun = dnorm, args = c(mean = dqmean, sd = dqsd), col = "tomato")
```



After initializing a blank plot with `geom_blank()`, the `ggplot2` package (within the `tidyverse`) allows us to add additional layers. The first layer is a density histogram. The second layer is a statistical function – the density of the normal curve, `dnorm`. We specify that we want the curve to have the same mean and standard deviation as the column of fat calories. The argument `col` simply sets the color for the line to be drawn. If we left it out, the line would be drawn in black.

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

Insert your answer here

#2: Koohyar's answer

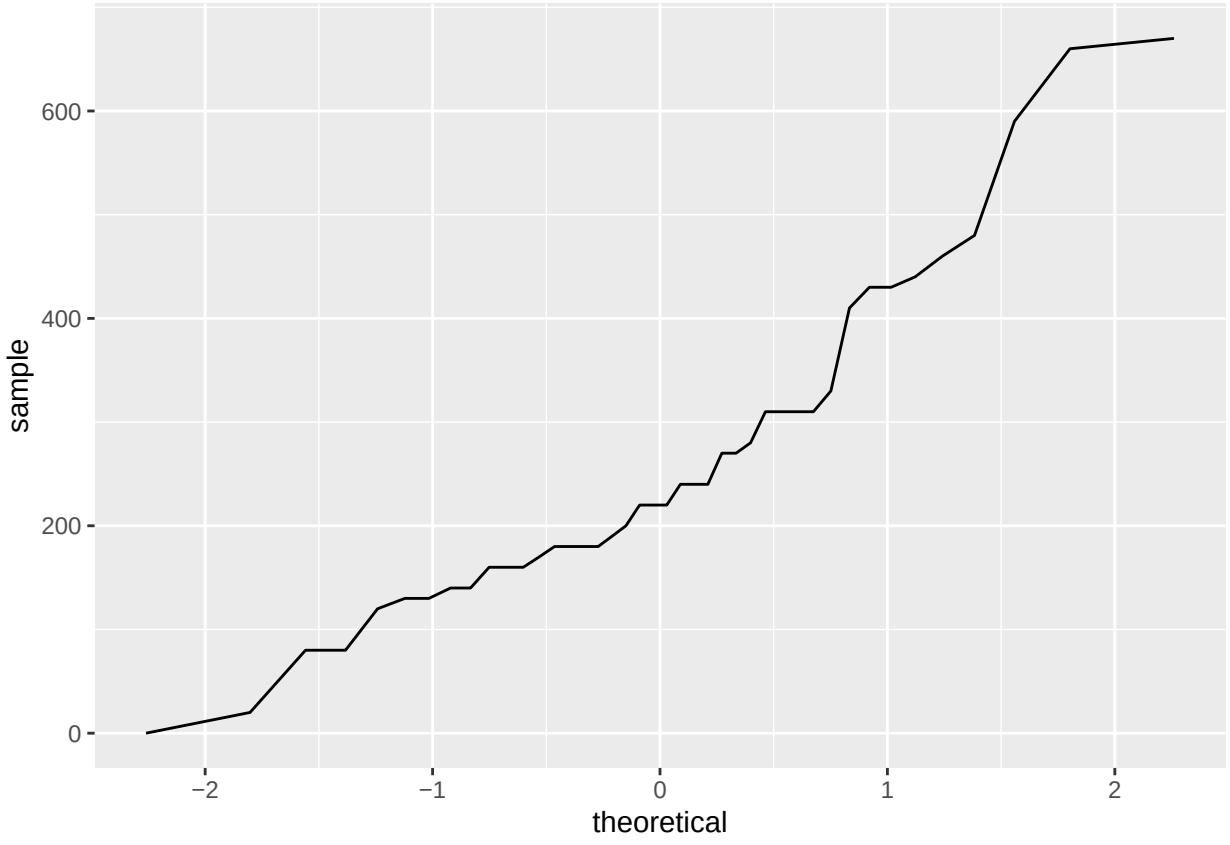
It does not appear that the distribution entirely meets normal distribution requirements. Although the general trend shows higher values in the middle and lower values as we move towards the extremes, the distribution does not precisely match the expected density. There are empty regions and low values in the middle of the graph, as well as some high points exceeding the expected value at the quartal ranges.

Overall, the graph is not a perfect representation of a normal distribution, but it is not severely skewed or entirely out of order. We can still identify some signs of a normal distribution in the data.

Evaluating the normal distribution

Eyeballing the shape of the histogram is one way to determine if the data appear to be nearly normally distributed, but it can be frustrating to decide just how close the histogram is to the curve. An alternative approach involves constructing 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`.

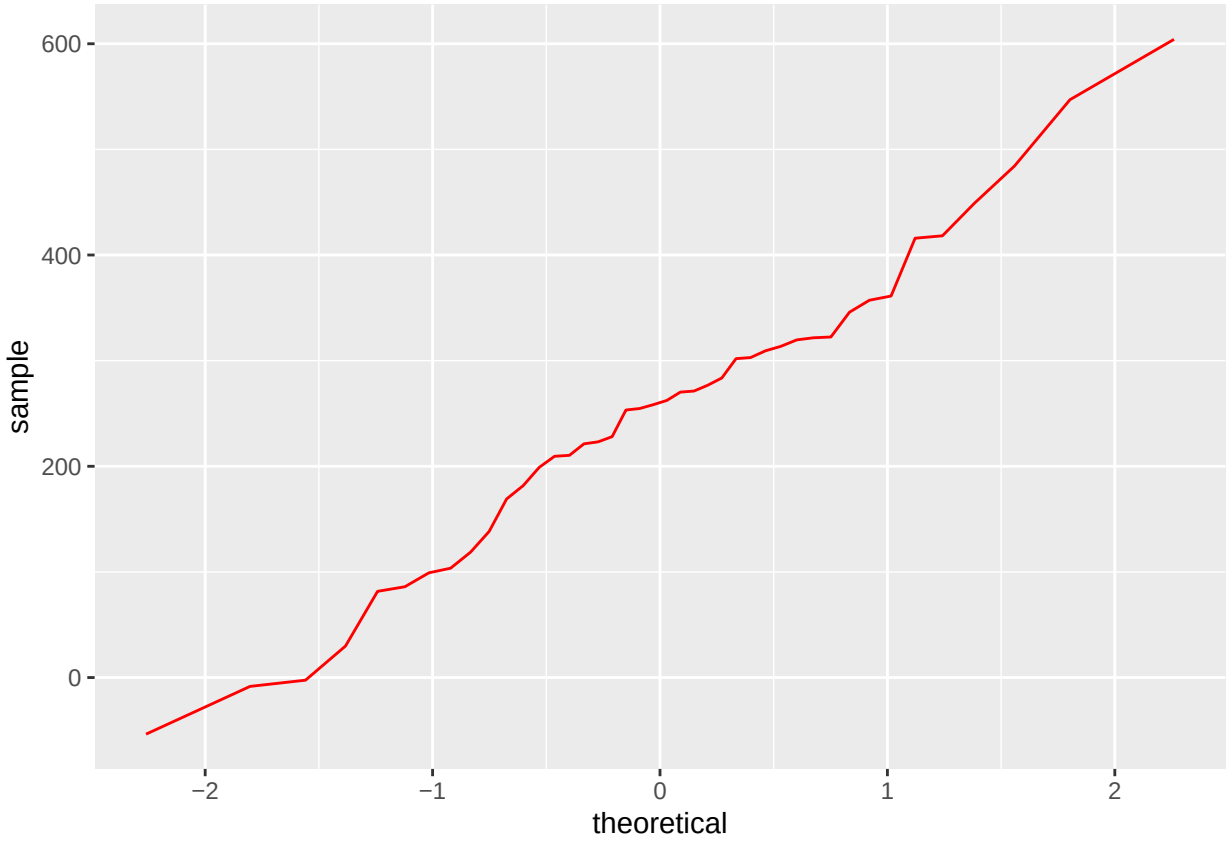
```
sim_norm <- rnorm(n = nrow(dairy_queen), mean = dqmean, sd = dqsd)
```

The first argument indicates how many numbers you'd like to generate, which we specify to be the same number of menu items in the `dairy_queen` data set using the `nrow()` function. The last two arguments determine the mean and standard deviation of the normal distribution from which the simulated sample will be generated. You can take a look at the shape of our simulated data set, `sim_norm`, as well as its normal probability plot.

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 data frame, it can be put directly into the `sample` argument and the `data` argument can be dropped.)

Insert your answer here

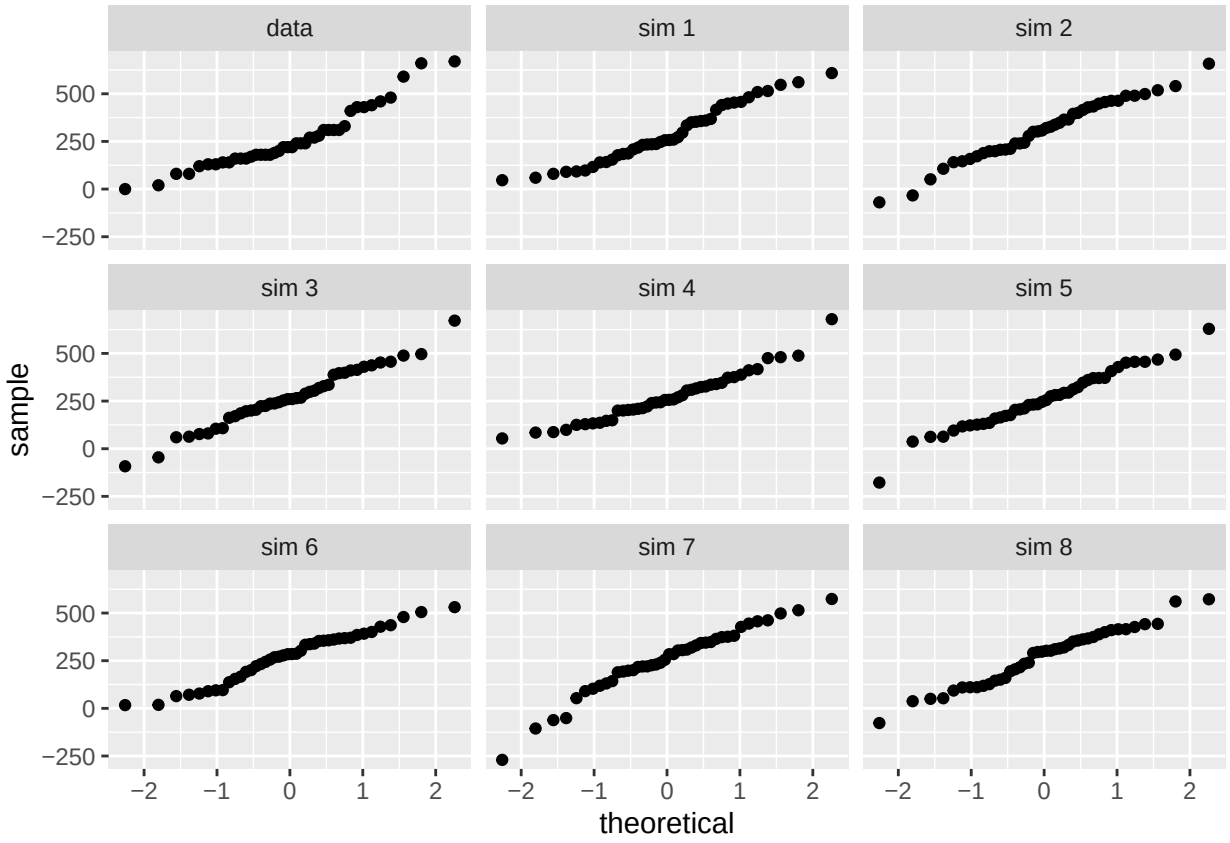
#3: Koohyar's answer



Upon closer examination of the graph, we observe that not all points align perfectly with the line. Some points deviate downward, while others rise above it. Specifically, there is an evident "S" curve in the middle section. Additionally, certain points dip below the linear line before ascending again. Although the overall shape appears linear at the beginning and end, the slopes of these segments differ.

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.

```
qqnormsim(sample = cal_fat, data = dairy_queen)
```



4. Does the normal probability plot for the calories from fat look similar to the plots created for the simulated data? That is, do the plots provide evidence that the calories are nearly normal?

Insert your answer here

#4: Koohyar's response

Exploring the additional normal simulation distribution is intriguing. It reveals how simulated data can vary widely from one iteration to another, and not all of these simulated datasets follow similar trends.

Upon comparing these simulated datasets with the original graph in the top left corner, my perspective has shifted. I now believe that the data may have been approximately normal from the outset. This observation underscores the idea that while simulated normally distributed data can exhibit differences,

they still originate from the assumption of meeting a normal distribution. It emphasizes the importance of conducting thorough analysis rather than relying solely on visual inspection and graphs to draw conclusions

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

Insert your answer here

#5: Koohyar's answer

We will follow the similar appraoch, we find the mean and sd, and will create a random normal distribution. then we laos create a set of normal distribution to compare them together. The goal is to see whether the McDonald's data can represent a normal distribution or not.

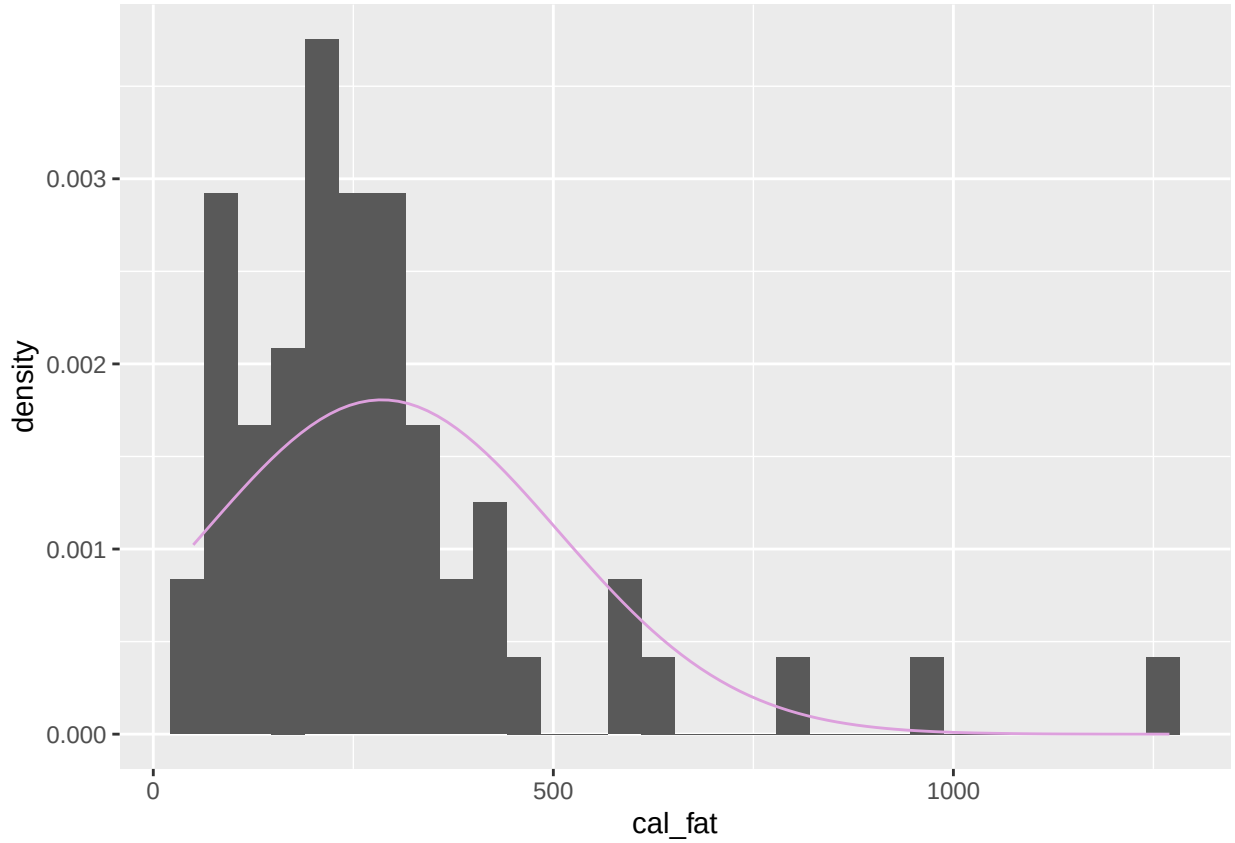
First we start by cacualting the sd adn mean ofthe data, leter we plot the density distribution and coampred it with the normal distrubution.

Then we plot the data agaist the quartile and coapre wither they along, and finally we make a normal dsitribution and comapre the 8 random normal with the atrgetet distribution to see how simialr or disimialr they are.

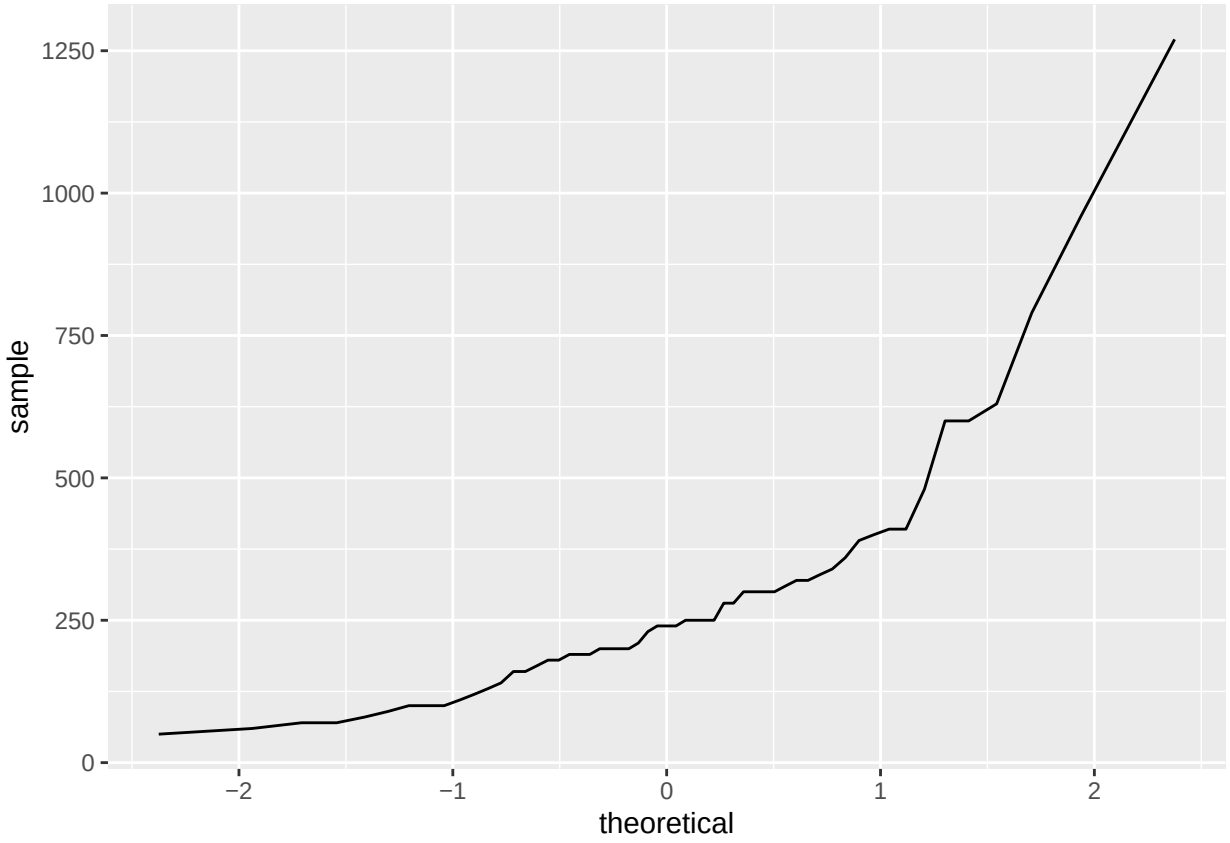
```
#fidn the mean and SD
print("Here are the mean and s for McDonalds calaroes form fat")
```

```
## [1] "Here are the mean and s for McDonalds calaroes form fat"
```

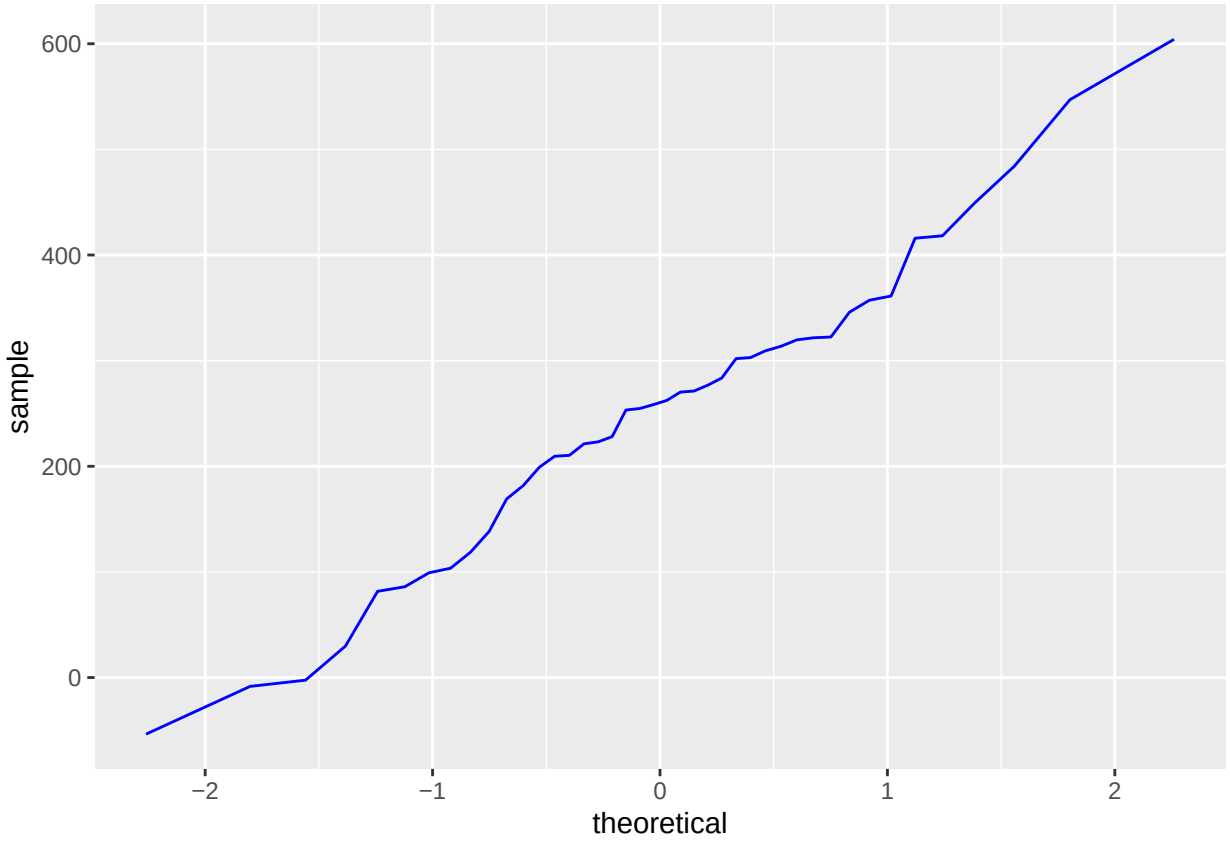
```
mdmean <- mean(mcdonalds$cal_fat)
mdsd   <- sd(mcdonalds$cal_fat)

ggplot(data = mcdonalds, aes(x = cal_fat)) +
        geom_blank() +
        geom_histogram(aes(y = ..density..)) +
        stat_function(fun = dnorm, args = c(mean = mdmean, sd = mdsd), col = "plum")
```



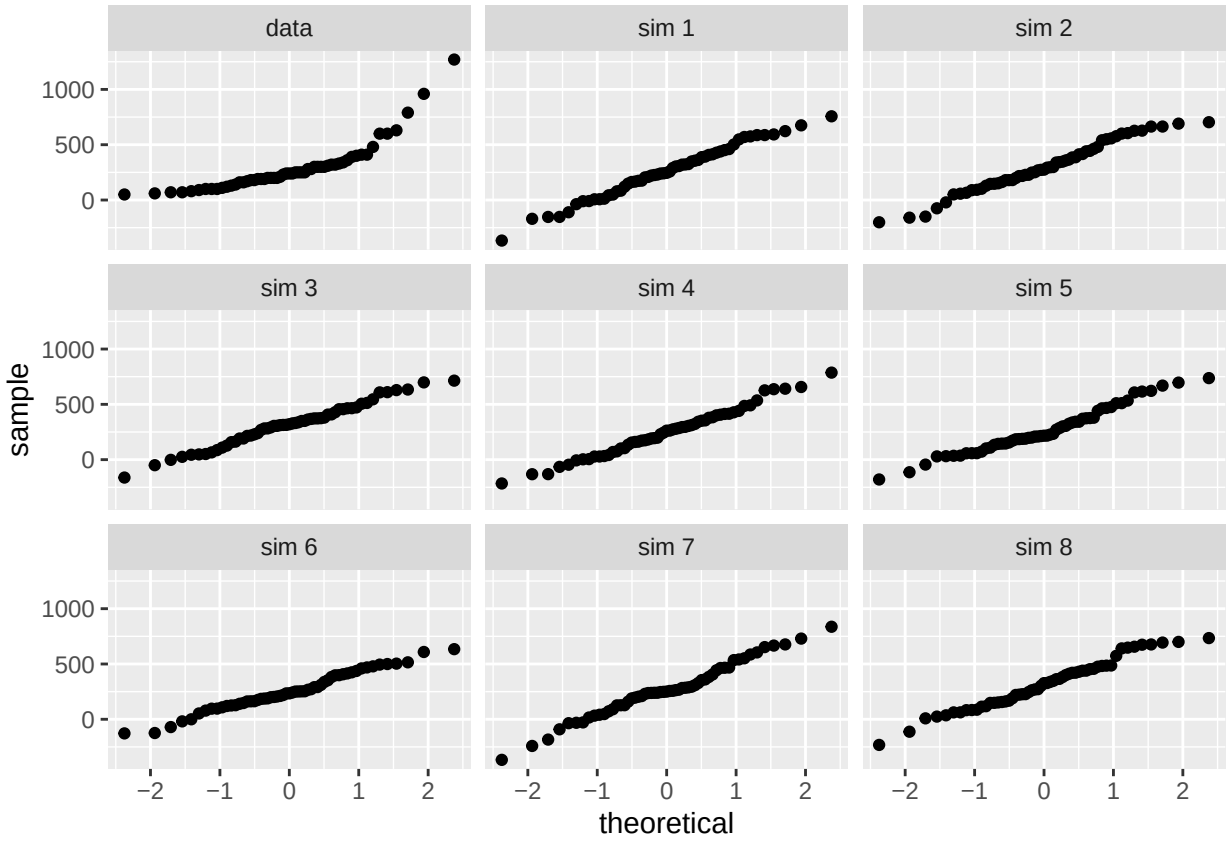

```
ggplot(data = mcdonalds, aes(sample = cal_fat)) +
  geom_line(stat = "qq")
```




```
sim_norm_md <- rnorm(n = nrow(mcdonalds), mean = mdmean, sd = mdsd)

ggplot( , aes(sample = sim_norm)) +
  geom_line(stat = "qq", col = "blue")
```




```
qqnormsim(sample = cal_fat, data = mcdonalds)
```



Let's start by examining the mean and standard deviation (SD). It appears that this dataset has more outliers, and the SD is larger compared to DQ's SD and mean.

When comparing the density plot of `cal_fat` against the normal density plot, we can observe a general trend between the two distributions. However, there are instances where gaps exist between the observed frequencies and the expected normal distribution. Notably, some values extend beyond the expected range, particularly for high-calorie data points. Despite these deviations, considering our previous impressions, it remains plausible that this data can be assumed to follow a normal distribution.

Next, we used the SD and mean to define a random distribution and compared it to the original data. In this scenario, when we compare the graph in the top left corner with the other datasets, a clear difference emerges. The shape of this data resembles a negative bell curve, curving below the straight line. By comparing it to the other eight graphs, we notice significant differences, which could indicate that this data is not normally distributed.

Normal probabilities

Okay, so now you have a slew of tools to judge whether or not a variable is normally distributed. Why should you care?

It turns out that statisticians know a lot about the normal distribution. Once you decide that a random variable is approximately normal, you can answer all sorts of questions about that variable related to probability. Take, for example, the question of, "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()`.

```r
1 - pnorm(q = 600, mean = dqmean, sd = dqsd)
```

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

Note that the function `pnorm()` gives the area under the normal curve below a given value, `q`, with a given mean and standard deviation. Since we're interested in the probability that a Dairy Queen item has more than 600 calories from fat, we have to take one minus that probability.

Assuming a normal distribution has allowed us to calculate a theoretical probability. If we want to calculate the probability empirically, we simply need to determine how many observations fall above 600 and then divide this number by the total sample size.

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

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

Although the probabilities are not exactly the same, they are reasonably close. The closer that your distribution is to being normal, the more accurate the theoretical probabilities will be.

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?

Insert your answer here

#6: Koohyar's answer

My first question is what is the probability of calories from fat being less than 35% of the total calories in all foods in McDonalds and Burger King?

```r
#add a new column that calculate the calories from fat over the total calories and then calcualte the n

fastfood <- fastfood %>%
  mutate(cal_ratio = cal_fat/calories*100)

md_sum <-  fastfood %>%
  filter( restaurant == "Mcdonalds") %>%
  summarize(
    md_mean     = mean(cal_ratio),
    md_sd       = sd(cal_ratio)
  )

(md_sum)
```

```
## # A tibble: 1 x 2
##   md_mean md_sd
##     <dbl> <dbl>
## 1    42.9  9.42
```

```
print("Parobability of calories from fat to be less than 40% of total calories assuming normal distribu
```

```
## [1] "Parobability of calories from fat to be less than 40% of total calories assuming normal distribu
```

```
(md_prob_n = pnorm(q = 35, mean = md_sum$md_mean, sd = md_sum$md_sd))
```

```
## [1] 0.1996537
```

```
print("Parobability of calories from fat to be less than 35% in MCdonalds data ")
```

```
## [1] "Parobability of calories from fat to be less than 35% in MCdonalds data "
```

```
md_prob_d  <- fastfood %>%
  summarize(
    total_mcdonalds = sum(restaurant == "Mcdonalds"),
    total_mcdonalds_and_35 = sum(restaurant == "Mcdonalds" & cal_ratio <= 35)
  ) %>%
  mutate(
    percent = total_mcdonalds_and_35 / total_mcdonalds * 100
  )

paste("The normal distribuion probablity ", round(md_prob_n*100,2), " % Vs the measured probability of"
```

```
## [1] "The normal distribuion probablity  19.97  % Vs the measured probability of 19.3 % for Mcdonalds
```

```
# Now let's work with Burger king


bk_sum <-  fastfood %>%
  filter( restaurant == "Burger King") %>%
  summarize(
    bk_mean     = mean(cal_ratio),
    bk_sd       = sd(cal_ratio)
  )

(bk_prob_n = pnorm(q = 35, mean = bk_sum$bk_mean, sd = bk_sum$bk_sd))
```

```
## [1] 0.02381765
```

```
bk_prob_d  <- fastfood %>%
  summarize(
    total_bk = sum(restaurant == "Burger King"),
    total_bk_and_35 = sum(restaurant == "Burger King" & cal_ratio <= 35)
  ) %>%
  mutate(
```

```
    percent = total_bk_and_35 / total_bk * 100
  )

paste("The normal distribuion probablity ", round(bk_prob_n*100,2), " % Vs the measured probability of"
```

```
## [1] "The normal distribuion probablity  2.38  % Vs the measured probability of 1.43 % for Burger King
```

What is the probability of foods having sugar more than 10 in Mcdonalds and DQ?

```
print("Let's find the normal dsitribution and measured probability of Mcdonalds or Dairy Queen to have m
```

```
## [1] "Let's find the normal dsitribution and measured probability of Mcdonalds or Dairy Queen to have
```

```
md_sum_1 <-  fastfood %>%
  filter( restaurant == "Mcdonalds") %>%
  summarize(
    md_mean     = mean(sugar),
    md_sd       = sd(sugar)
  )

md_prob_n = 1-pnorm(q = 10, mean = md_sum_1$md_mean, sd = md_sum_1$md_sd)


dq_sum_1 <-  fastfood %>%
  filter( restaurant == "Dairy Queen") %>%
  summarize(
    dq_mean     = mean(sugar),
    dq_sd       = sd(sugar)
  )

dq_prob_n = 1-pnorm(q = 10, mean = dq_sum_1$dq_mean, sd = dq_sum_1$dq_sd)

print("Parobability of sugar to be more than 10 in MCdonalds data")
```

```
## [1] "Parobability of sugar to be more than 10 in MCdonalds data"
```

```
md_prob_d  <- fastfood %>%
  summarize(
    total_mcdonalds = sum(restaurant == "Mcdonalds"),
    total_mcdonalds_and_10 = sum(restaurant == "Mcdonalds" & sugar > 10)
  ) %>%
  mutate(
    percent = total_mcdonalds_and_10 / total_mcdonalds * 100
  )

paste("The normal distribuion probablity ", round(md_prob_n*100,2), " % Vs the measured probability of"
```

```
## [1] "The normal distribuion probablity  53.2  % Vs the measured probability of 42.11 % for Mcdonalds
```

```r
dq_prob_d  <- fastfood %>%
  summarize(
    total_dq = sum(restaurant == "Dairy Queen"),
    total_dq_and_10 = sum(restaurant == "Dairy Queen" & sugar > 10)
  ) %>%
  mutate(
    percent = total_dq_and_10 / total_dq * 100
  )

paste("The normal distribuion probablity ", round(dq_prob_n*100,2), " % Vs the measured probability of"
```

```
## [1] "The normal distribuion probablity  23.43  % Vs the measured probability of 9.52 % for Mcdonalds
```

More Practice

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

Insert your answer here

#7: Koohyar's Answer

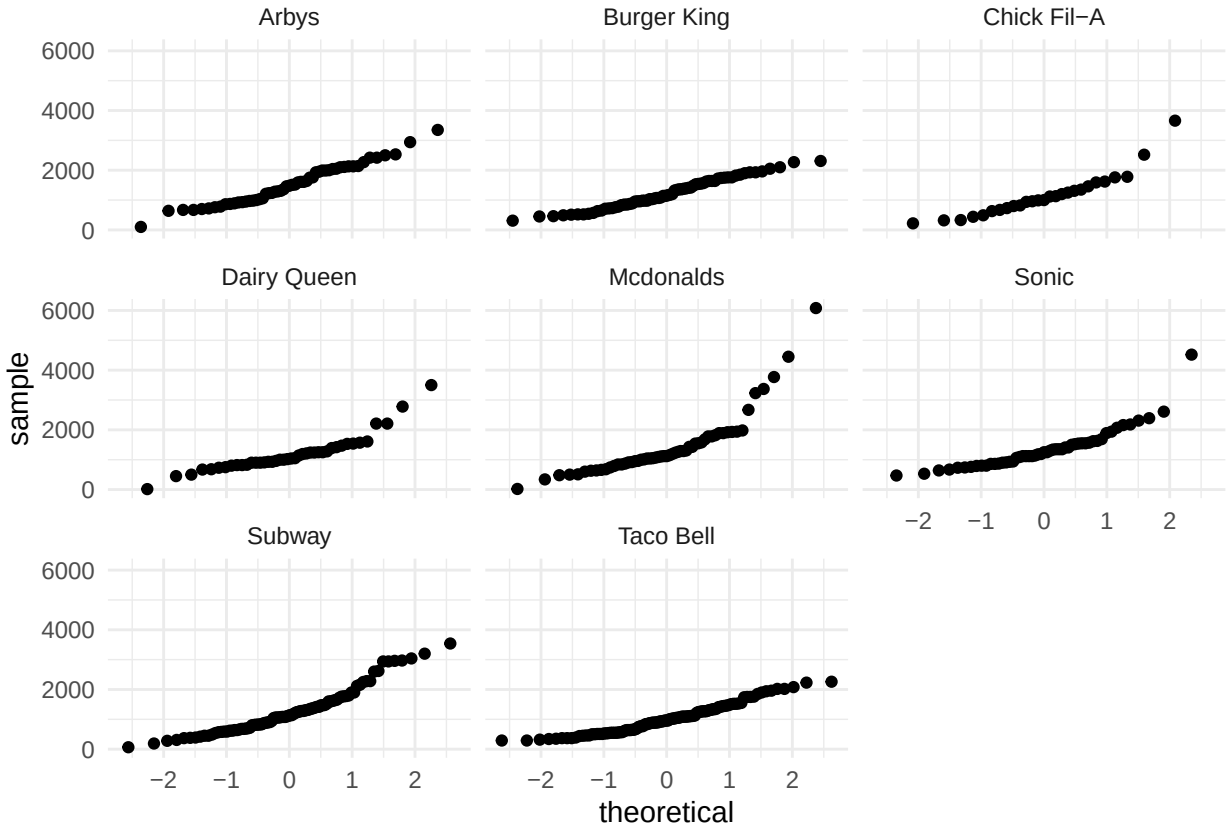
To determine whether a specific distribution is normal or not, we can follow the steps mentioned above and create Q-Q plots. Additionally, I have opted for an alternative approach by calculating summary statistics such as mean, median, skewness, and kurtosis to assess the normality of the data.

```r
# calculate summary statistics such as mean, median, skewness, and kurtosis.
#install.packages("moments")
library(moments)

sum_stat <-  fastfood %>%
  group_by(restaurant) %>%
  summarize(
    mean     = mean(sodium),
    sd       = sd(sodium),
    median   = median(sodium),
    skewness = skewness(sodium),
    kurtosis = kurtosis(sodium)
  ) %>%
  arrange(skewness & kurtosis)

(sum_stat)
```

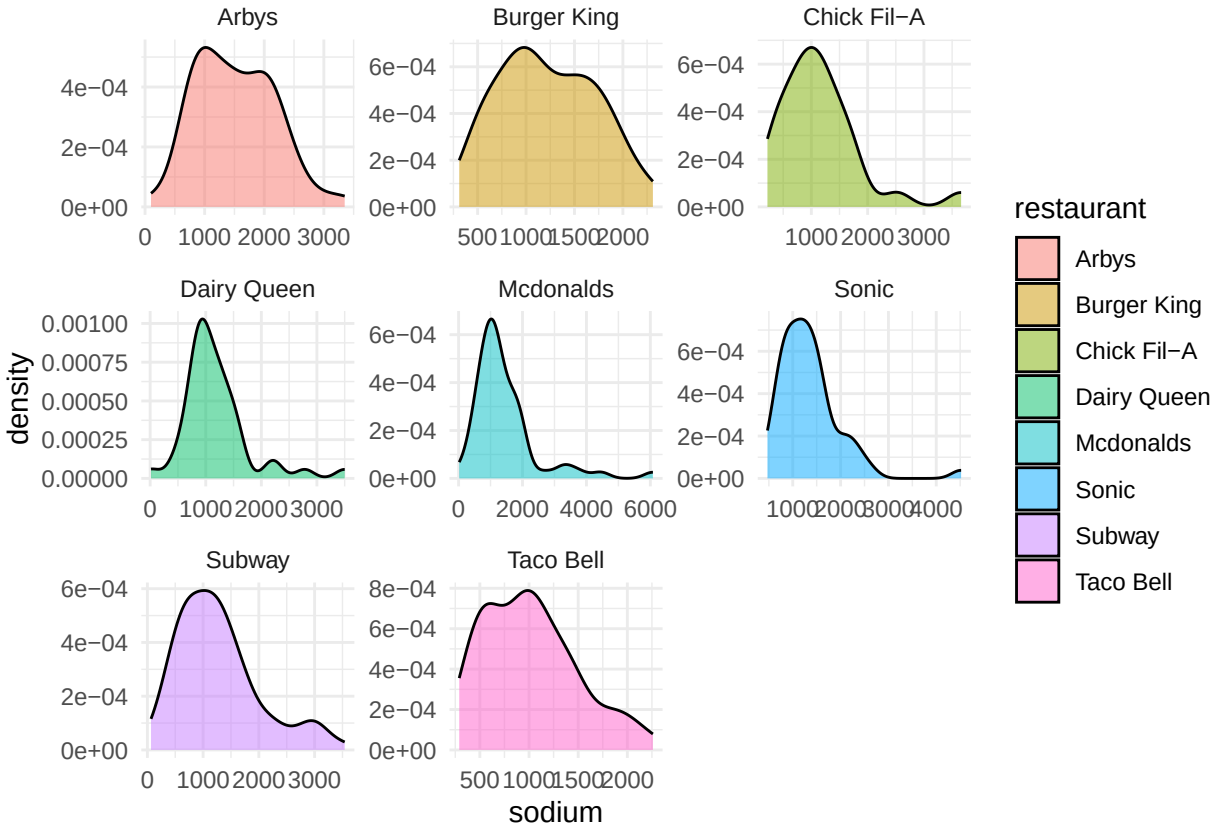
```
## # A tibble: 8 x 6
##   restaurant   mean    sd median skewness kurtosis
##   <chr>       <dbl> <dbl>  <dbl>    <dbl>    <dbl>
## 1 Arbys       1515.  664.   1480    0.394     2.77
## 2 Burger King 1224.  500.   1150    0.192     2.11
```

```
## 3 Chick Fil-A 1151.  727.   1000    1.64      6.55
## 4 Dairy Queen 1182.  610.   1030    1.73      7.30
## 5 Mcdonalds   1438. 1036.   1120    2.34      9.69
## 6 Sonic       1351.  665.   1250    2.24     11.0
## 7 Subway      1273.  744.   1130    1.02      3.65
## 8 Taco Bell   1014.  474.    960    0.602     2.74
```

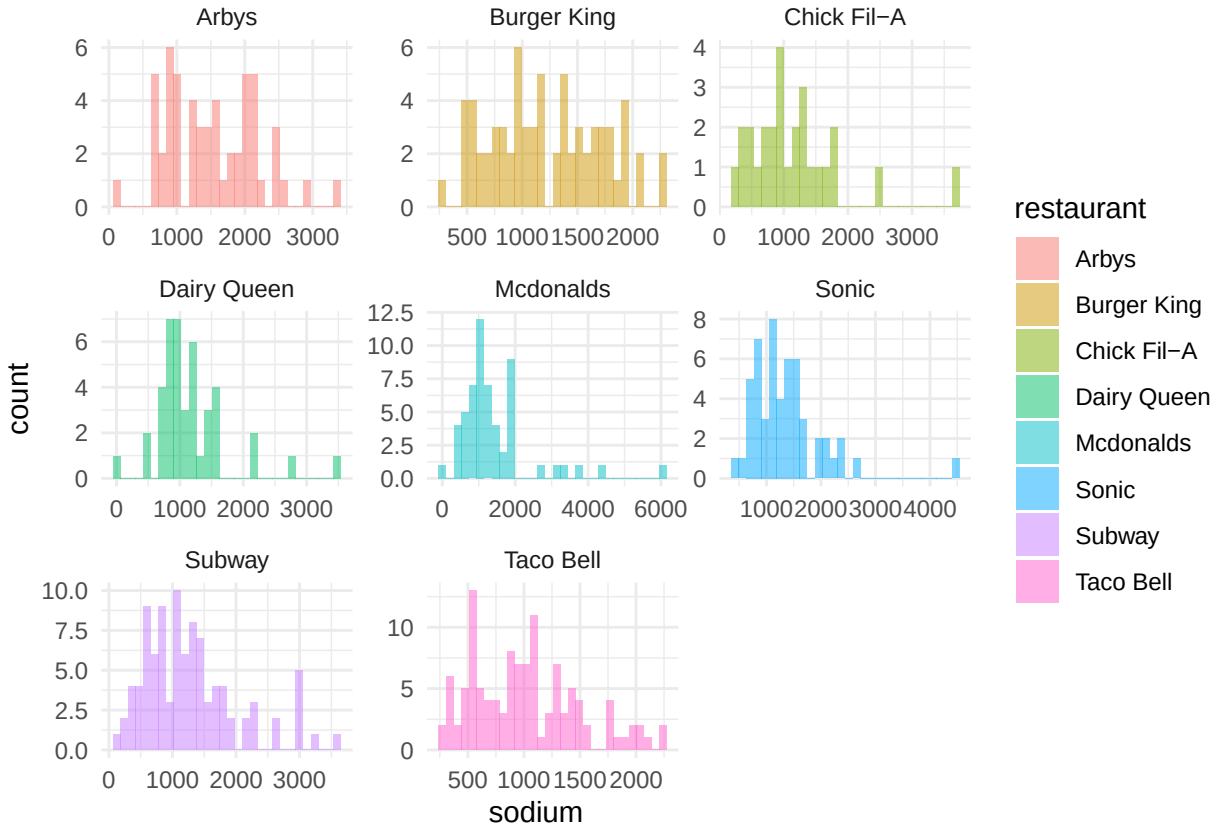
```r
#Let's plot density histograms

# let's plot Q-Q s
ggplot(fastfood, aes(sample = sodium)) +
  geom_qq() +
  facet_wrap(~restaurant)+
    theme_minimal()  # Optional: Adjust plot theme
```




```r
# Let's plot density histograms
ggplot(fastfood, aes(x = sodium, fill = restaurant)) +
  geom_density(alpha = 0.5) +
  facet_wrap(~ restaurant, scales = "free") +
  theme_minimal()
```




```r
# Let's plot histograms
ggplot(fastfood, aes(x = sodium, fill = restaurant)) +
  geom_histogram(alpha = 0.5) +
  facet_wrap(~ restaurant, scales = "free") +
  theme_minimal()
```



Looking into the skewness and kurtosis Looking, it seems Arbys, Burger King, and Taco Bell with skewness close to zero and kurtosis close to 3 have higher chances of being normal distributions.

Also looking into the Q-Q graphs and histograms of all distinct restaurants seems to support that Arbys, Burger King, Taco Bell, and Sonic are among those with a normal distribution of sodium close to normal.

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?

Insert your answer here

#8: Koohyar's Answer

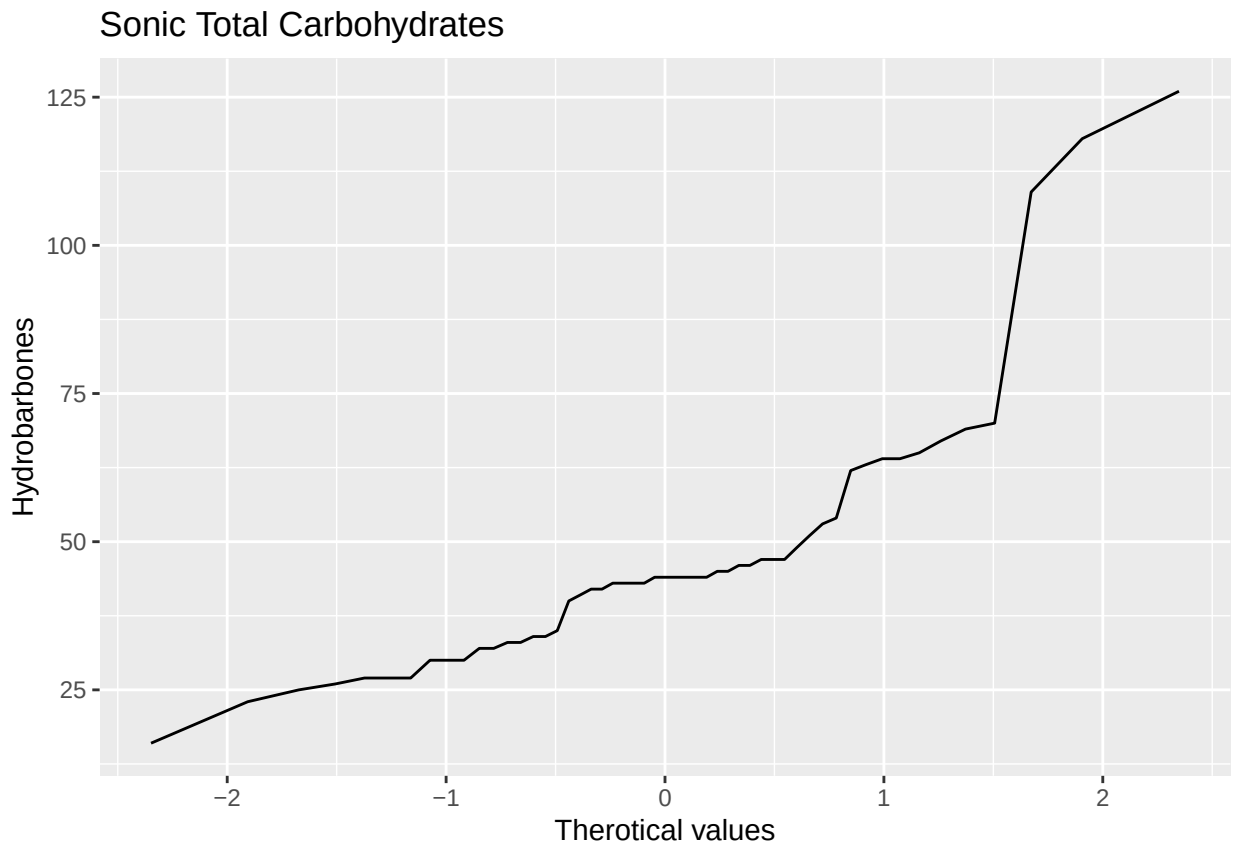
This likely relates to the outliers. Certain food items have notably higher sodium levels, which contribute to the sudden change. Additionally, there appears to be a group of foods with elevated sodium content, and these outliers stand out distinctly in the histograms.

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.

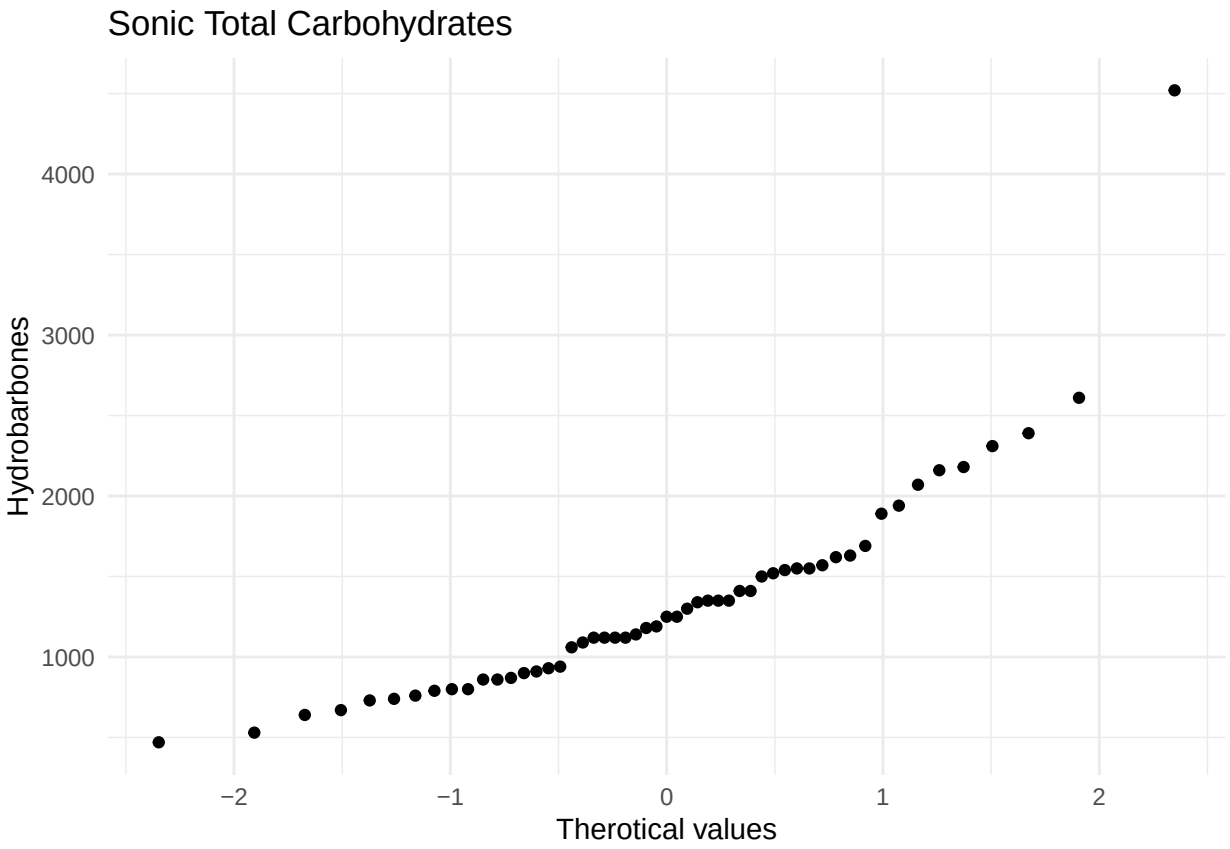
#9: Koohyar's Answer

I have selected Sonic. It has knees around 1.5, which indicates right-skewed data. Additionally, it exhibits a concave appearance that supports the right-skewness

```r
Sonic <- fastfood %>%
  filter(restaurant == "Sonic")

ggplot(data = Sonic, aes(sample = total_carb)) +
  geom_line(stat = "qq") +
  labs(title = "Sonic Total Carbohydrates", x = "Therotical values", y = "Hydrobarbones")
```




```r
#or
ggplot(Sonic, aes(sample = sodium)) +
  geom_qq() +
  theme_minimal() +
   labs(title = "Sonic Total Carbohydrates", x = "Therotical values", y = "Hydrobarbones")
```



Insert your answer here
