Topic 2.3: basics of statistical inference

Quantiles

Sonja Petrović

Created for ITMD/ITMS/STAT 514

Spring 2021.

Theory

Discussion with the whiteboard.

What we learned today:

Definition of a quantile, and of quantile-quantile plot.

An excellent tiny example[1]

Objective

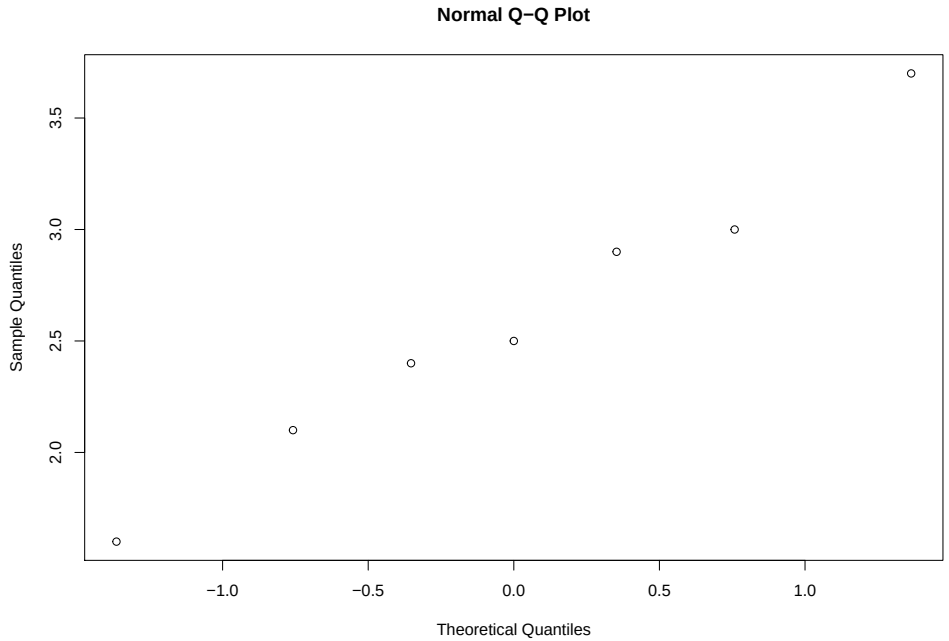
Use R to construct normal scores plots.

- This is done using the function `qqnorm`.

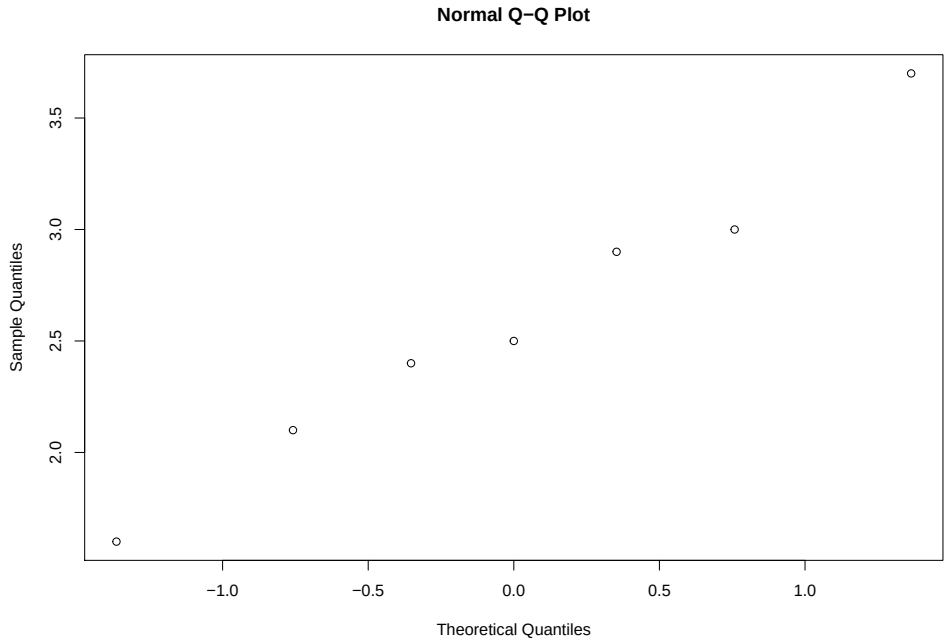
Please read the source pdf linked in the footnote to learn about the differences of output plots between different packages (flipped axes(!), small round-off variation, etc.).

[1] http://pages.stat.wisc.edu/~yandell/st571/R/append8.pdf

```r
mydata = c(2.4, 3.7, 2.1, 3, 1.6, 2.5, 2.9)
myquant = qqnorm(mydata)
```



```
mydata = c(2.4, 3.7, 2.1, 3, 1.6, 2.5, 2.9)
myquant = qqnorm(mydata)
```



- If the observations in mydata come from a normal distribution, then the above plot of mydata versus their population quantiles should give a straight line.
- It seems not unreasonable to conclude from this plot that the data come from a normal distribution.

- The object myquant contains the quantiles (`myquant$x`) and the original data (`myquant$y`).
- The quantiles can be viewed by printing the object myquant as a data frame:

```
data.frame(myquant)
```

```
           x   y
1 -0.3529340 2.4
2  1.3644887 3.7
3 -0.7582926 2.1
4  0.7582926 3.0
5 -1.3644887 1.6
6  0.0000000 2.5
7  0.3529340 2.9
```

Example of qqplot

qqplot with birthweight data

Let's begin by loading the packages we'll need to get started

```
library(tidyverse)
```

```
-- Attaching packages --------------------------------------

v tibble  3.0.4     v purrr   0.3.4
v tidyr   1.1.2     v stringr 1.4.0
v readr   1.4.0     v forcats 0.5.1

-- Conflicts ------------------------------------------ tidyve
x dplyr::collapse()          masks nlme::collapse()
x mosaic::count()            masks dplyr::count()
x purrr::cross()             masks mosaic::cross()
x mosaic::do()               masks dplyr::do()
x tidyr::expand()            masks Matrix::expand()
x dplyr::filter()            masks stats::filter()
```

Exploring the birthweight data

We'll begin by doing all the same data processing as in previous lectures

```
# Load data from MASS into a tibble
birthwt <- as_tibble(MASS::birthwt)

# Rename variables
birthwt <- birthwt %>%
  rename(birthwt.below.2500 = low,
         mother.age = age,
         mother.weight = lwt,
         mother.smokes = smoke,
         previous.prem.labor = ptl,
         hypertension = ht,
         uterine.irr = ui,
         physician.visits = ftv,
         birthwt.grams = bwt)
```

```r
# Change factor level names
birthwt <- birthwt %>%
  mutate(race = recode_factor(race, `1` = "white",
                `2` = "black", `3` = "other")) %>%
  mutate_at(c("mother.smokes", "hypertension",
              "uterine.irr", "birthwt.below.2500"),
            ~ recode_factor(.x, `0` = "no", `1` = "yes"))
```

How this fits into the larger picture we're working on

Over the past two lectures we created various tables and graphics to help us better understand the data. Our focus for today is to run hypothesis tests to assess whether the trends we observed last time are statistically significant.

One of the main reasons we want to understand hypothesis testing is that it is important for our tables and figures to convey statistical uncertainty in any cases where it is non-negligible, and where failing to account for it may produce misleading conclusions.

- we will learn a lot more, but for now let's test normality, as that's the focus of today's lecture!

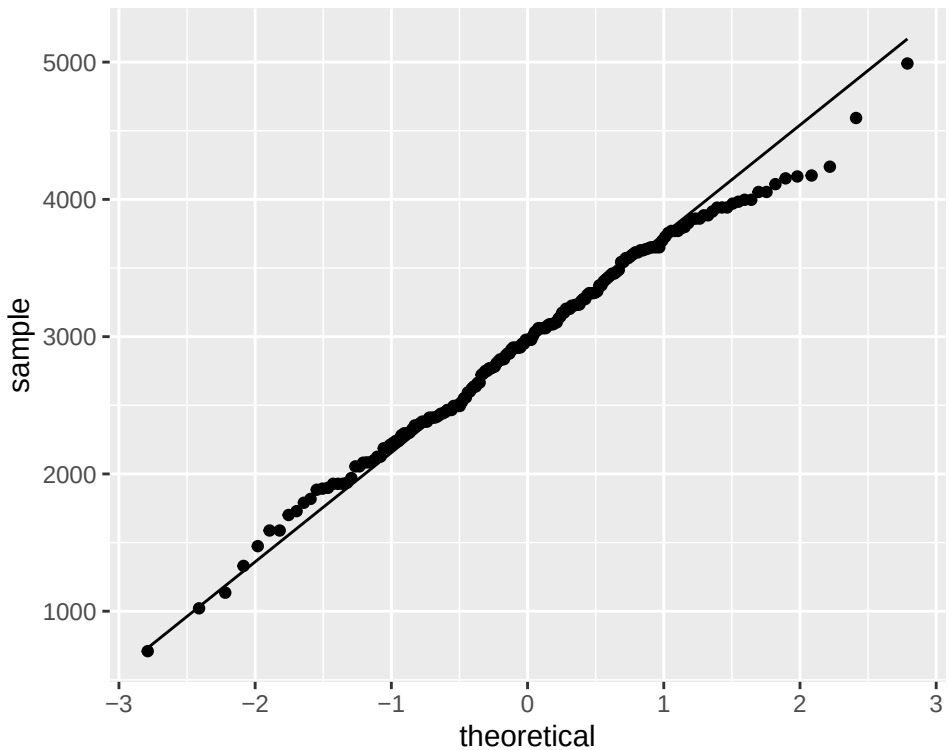
Is the data normal?

I would recommend using a non-parametric test when the data appears highly non-normal and the sample size is small. If you really want to stick to t-testing, it's good to know how to diagnose non-normality.

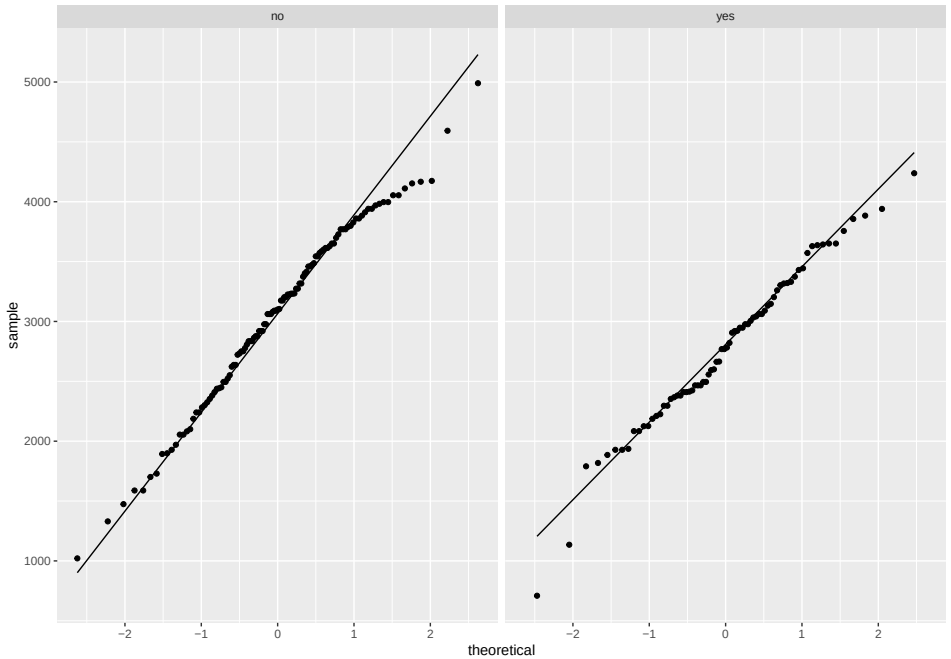
qq-plot

The simplest thing to look at is a normal qq plot of the data. This is obtained using the `stat_qq()` function.

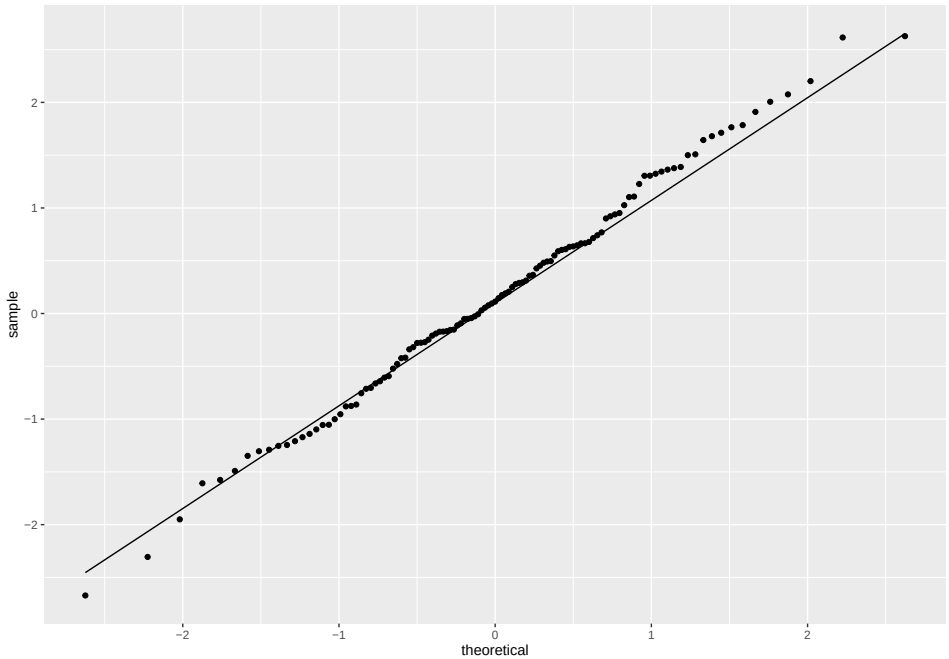
```r
# qq plot
p.birthwt <- ggplot(data = birthwt,
                    aes(sample = birthwt.grams))

p.birthwt + stat_qq() + stat_qq_line()
```



```
# Separate plots for different values of smoking status
p.birthwt + stat_qq() + stat_qq_line() +
  facet_grid(. ~ mother.smokes)
```



```r
# qq plot for 115 observations of truly normal data
df <- data.frame(x = rnorm(115))
ggplot(data= df, aes(sample = x))+ stat_qq()+ stat_qq_line()
```

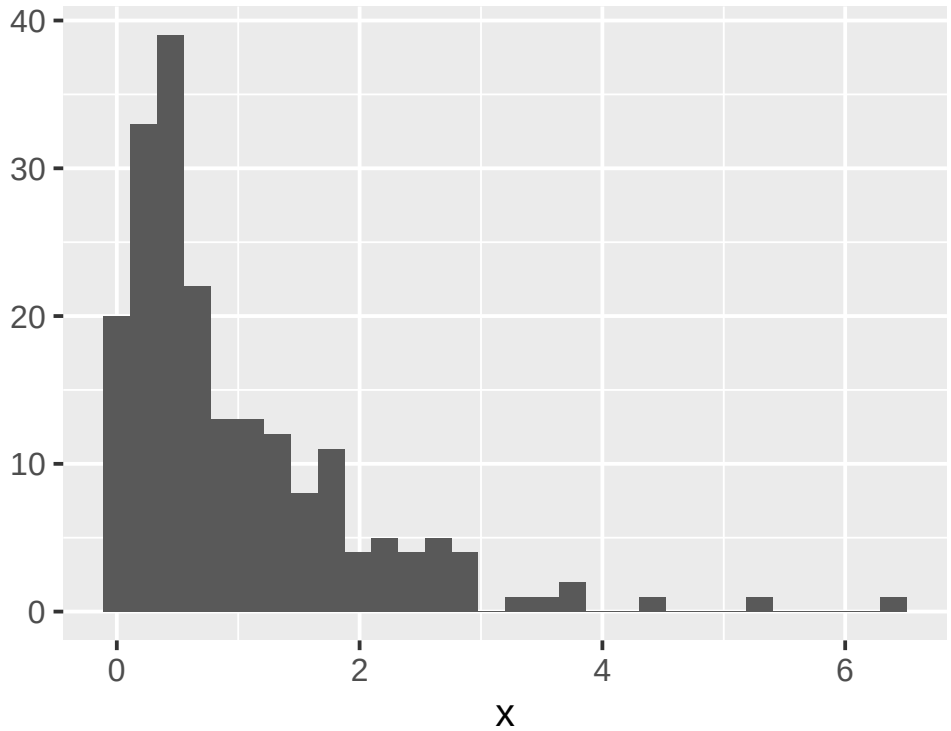


If the data are exactly normal, you expect the points to lie on a straight line. The data we have here are pretty close to lying on a line.

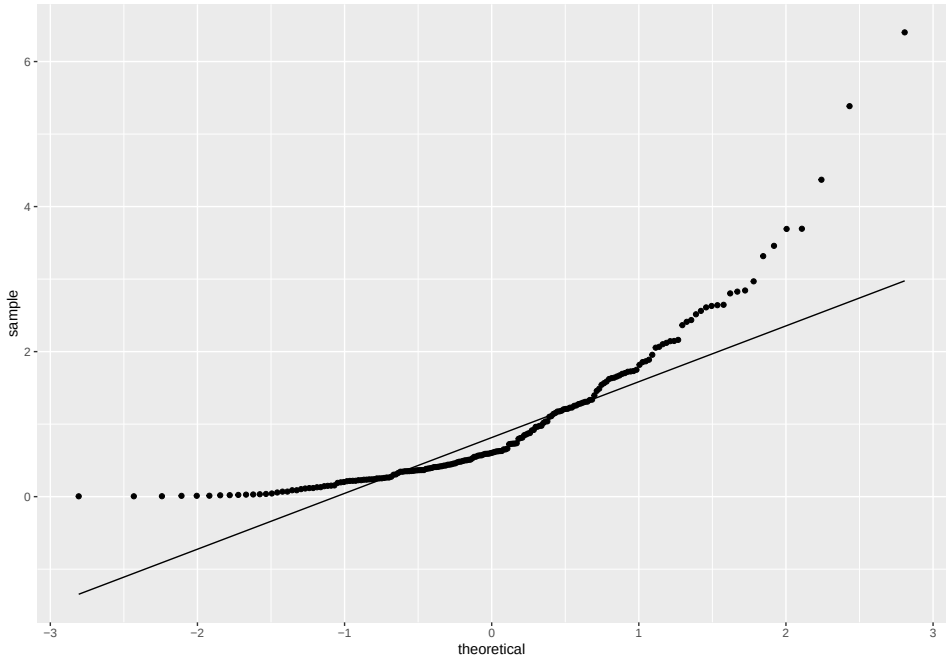
Here's what we would see if the data were right-skewed

```
set.seed(12345)
fake.data <- data.frame(x = rexp(200))
p.fake <- ggplot(fake.data, aes(sample = x))
qplot(x, data = fake.data)
```

```
`stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



```
p.fake + stat_qq() + stat_qq_line()
```



If you construct a qqplot and it looks like this, you should be careful, particularly if your sample size is small.

What's next?

- we will use the above dataset and qqplot when we perform a test of difference in means for this data. Stay tuned!

License

This document is created for ITMD/ITMS/STAT 514, Spring 2021, at Illinois Tech.

The second part of the notes (about qqplot with the birthweight data) is xtracted from Prof. Alexandra Chouldechova at CMU, under the Creative Commons Attribution-NonCommercial-ShareAlike 4.0 International License.