Log 708 - Chapter 4 Solutions

Halvard Arntzen

4.1

a. We can use the test statistic

$$T = \frac{\bar{x} - 290}{S/\sqrt{n}}$$

and use $N(0,1)$ as null distribution. We calculate the observed value either with a calculator or as here, using R

```
t_obs <- (319 - 290)/(100/sqrt(400))
t_obs
```

```
## [1] 5.8
```

The observed value is 5.8. Now, since the test is two-sided, we get the P-value as

$$P = P[T \leq -5.8 \text{ or } T \geq 5.8] = 2 \cdot P[T \leq 5.8]$$

```
pval <- 2*pnorm(-5.8)
pval
```

```
## [1] 6.631492e-09
```

As expected from the observed T value, the P-value is practically 0. We reject H_0 at the given level.

b. For this we use the binomial test. We first note that

$$\hat{p} = \frac{56}{400} = 0.14$$

Here the test statistic and observed value becomes

$$Z_{OBS} = \frac{\hat{p} - 0.20}{\sqrt{\frac{0.20(1-0.20)}{n}}} = \frac{0.14 - 0.20}{\sqrt{\frac{0.20(1-0.20)}{400}}} = -3.0$$

The P-value is now $P = 2 \cdot P[Z \leq -3.0] = 0.003$, so again we reject the null hypothesis. The proportion of cash paying customers is likely to be quite a bit lower than the suggested 0.20.

c. Ok, by assumption in this part, we have 20% paying with cash and they pay on average an estimated 319 NOK. On the other hand, 10% make cash withdrawals at average 400 NOK. I.e. on average, for each withdrawal (-400) there are two cash payments at $2 \cdot 319 = 638$ NOK, so the balance should be OK. (Even with the estimated 14% paying cash, we are OK, because each -400 withdrawal is countered by $1.4 \cdot 319 = 447$ NOK in cash payment.)

4.2

We start by reading the data and look at top rows

```
tripdata <- read.csv("M:/Undervisning/Undervisningh21/Data/Trip_durations.csv")

head(tripdata)
```

```
##   Duration Distance
## 1     12.5     2.74
## 2     33.5    13.60
## 3     33.1    10.99
## 4     37.0    10.31
## 5     18.3     4.93
## 6     29.3     8.15
```

a. We calculate n, $\bar{x}$, S_x as follows. (Note, we can name the code chunks which makes finding errors a little easier sometimes)

```
n <- nrow(tripdata)
x_bar <- mean(tripdata$Duration)
s_x <- sd(tripdata$Duration)
#print the three numbers as a vector (just to save space in output)
c(n, x_bar, s_x)
```

```
## [1] 200.000000  24.310500   6.835471
```

So $n = 200, \bar{x} = 24.31, S_x = 6.84$.

b. The test statistic should be

$$T = \frac{\bar{x} - 24}{S/\sqrt{n}}$$

and the null distribution (the distribution assuming H_0 is true) is a t-distribution with $df = n - 1 = 199$. We can approximate this very well with the $N(0, 1)$ distribution. We can calculate the T_{obs} value:

```
t_obs <- (x_bar - 24)/ (s_x / sqrt(n))
t_obs
```

```
## [1] 0.6424039
```

We get $T_{obs} = 0.64$. The test is two-sided, so we get the P-value as follows.

$$P = P[T \leq -0.64 \text{ or } T \geq 0.64] = 2 * P[T \leq -0.64].$$

Using R and the $N(0, 1)$ distribution, we get

```
p_value <- 2*pnorm(-0.64)
p_value
```

```
## [1] 0.5221726
```

The P-value is about 0.52 and with significance level $\alpha = 0.05$ we can not reject the H_0.

c. Now we can run with the `t.test` function from R.

```
durtest1 <- t.test(tripdata$Duration, mu = 24, alternative = "two.sided")
durtest1$statistic
```

```
##         t
## 0.6424039
```

```
durtest1$p.value
```

```
## [1] 0.5213503
```

We see that we get almost identical values.

d. This means we simply have to change the constant in the test to 23 and repeat. The test statistic is

```r
t_obs <- (x_bar - 23)/ (s_x / sqrt(n))
t_obs
```

```
## [1] 2.711338
```

Using R and the $N(0,1)$ distribution, we get a new P-value:

```r
p_value <- 2*pnorm(-2.71)
p_value
```

```
## [1] 0.006728321
```

And in this case we clearly reject H_0. We can confirm this by testing directly with R:

```r
durtest2 <- t.test(tripdata$Duration, mu = 23, alternative = "two.sided")
durtest2$statistic
```

```
##        t
## 2.711338
```

```r
durtest2$p.value
```

```
## [1] 0.007287267
```

Again, the results are very close, and the same conclusion applies.

e. Since we are to seek evidence for $\mu_s > 10$, this must be the alternative hypothesis H_1, and then we can have $H_0 : \mu_s = 10$ as the null hypothesis. This is then a one-sample t test with a one-sided alternative, that the mean is *greater* than 10, so in R we can do

```r
disttest <- t.test(tripdata$Distance, mu = 10, alternative = "greater")
#now we can chech the whole output.
disttest
```

```
##
##  One Sample t-test
##
## data:  tripdata$Distance
## t = 1.9522, df = 199, p-value = 0.02616
## alternative hypothesis: true mean is greater than 10
## 95 percent confidence interval:
##  10.06158      Inf
## sample estimates:
## mean of x
##  10.40115
```

We get $T_{obs} = 1.95$ and a resulting P-value at 0.026. This is below 0.05, so we reject H_0.

f. One (out of many) ways to get a 95 confidence interval for a mean in R is to run the t.test with a twosided alternative. Since this is the "default" setting for `t.test`, and also the `mu` value is irrelevant for the confidence intervals, we can simply do

```
dur_test <- t.test(tripdata$Duration)
dist_test <- t.test(tripdata$Distance)

dur_test$conf.int
```

```
## [1] 23.35737 25.26363
## attr(,"conf.level")
## [1] 0.95
```

```
dist_test$conf.int
```

```
## [1]  9.995949 10.806351
## attr(,"conf.level")
## [1] 0.95
```

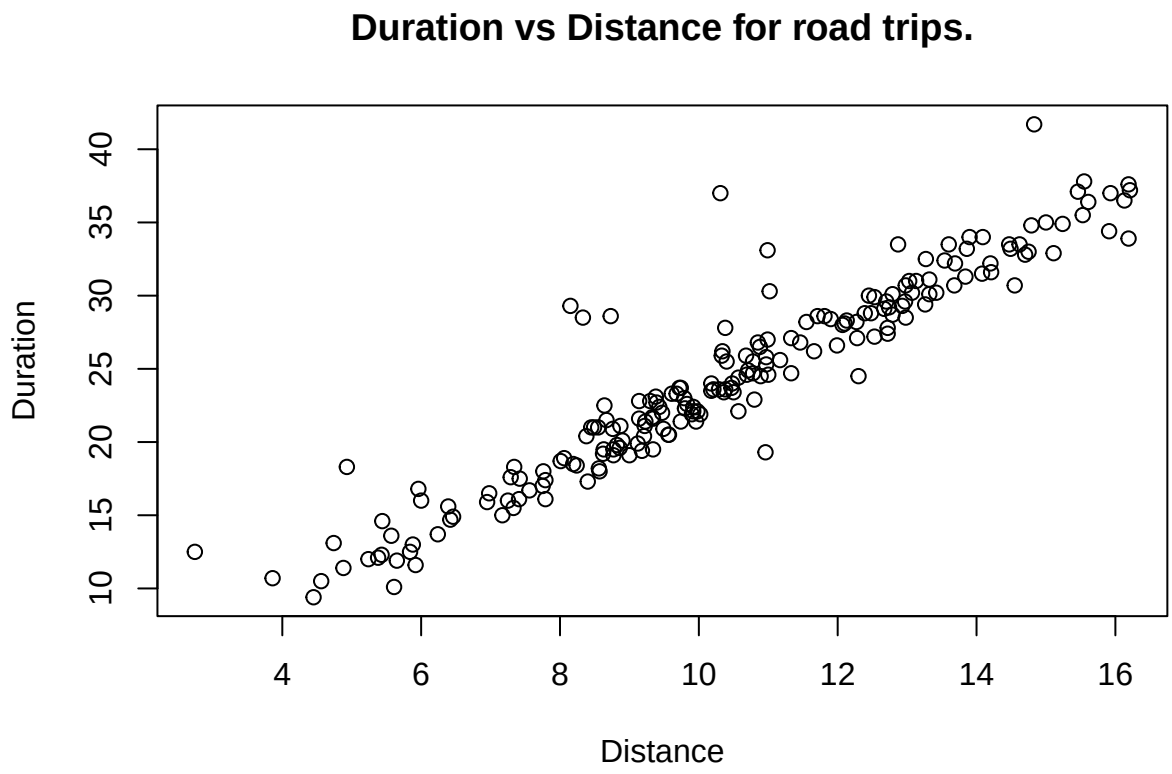
In the case of the duration variable, we see the interval containing 24, but not 23, which explains the different results in b,c contra d. Regarding the distance variable, we see the 95% confidence interval reaching from practically 10.00 to 10.80, so indicating that the μ_s is greater than 10.00, although we get a relatively weak evidence. We also see this since the P-value in this case was not much lower than 0.05.

g. Here we can do

```
with(tripdata, cor(Duration, Distance))
```

```
## [1] 0.9461513
```

```
with(tripdata, plot(Distance, Duration,
                    main = "Duration vs Distance for road trips."))
```



The correlation is about 0.95, and the plot shows strong dependency between the variables.

4.3

a. We read the data in the ordinary way.

```
flats <- read.csv("M:/Undervisning/Undervisningh21/Data/flat_prices.csv")
head(flats)
```

```
##   price area rooms standard situated town distcen age rent
## 1  1031  100     3        2        6    1       5  15 2051
## 2  1129  116     3        1        5    1       4  42 2834
## 3  1123  110     3        2        5    1       3  25 2468
## 4   607   59     2        3        5    1       6  25 1940
## 5   858   72     2        3        4    1       1  17 1611
## 6   679   64     2        2        3    1       3  17 2039
```

From the (updated) exercise text, we find the encoding for `town` as (1, 2, 3) for (Molde, Kristiansund, Ålesund).

we can use the `tapply` function to compute means in the towns as follows.

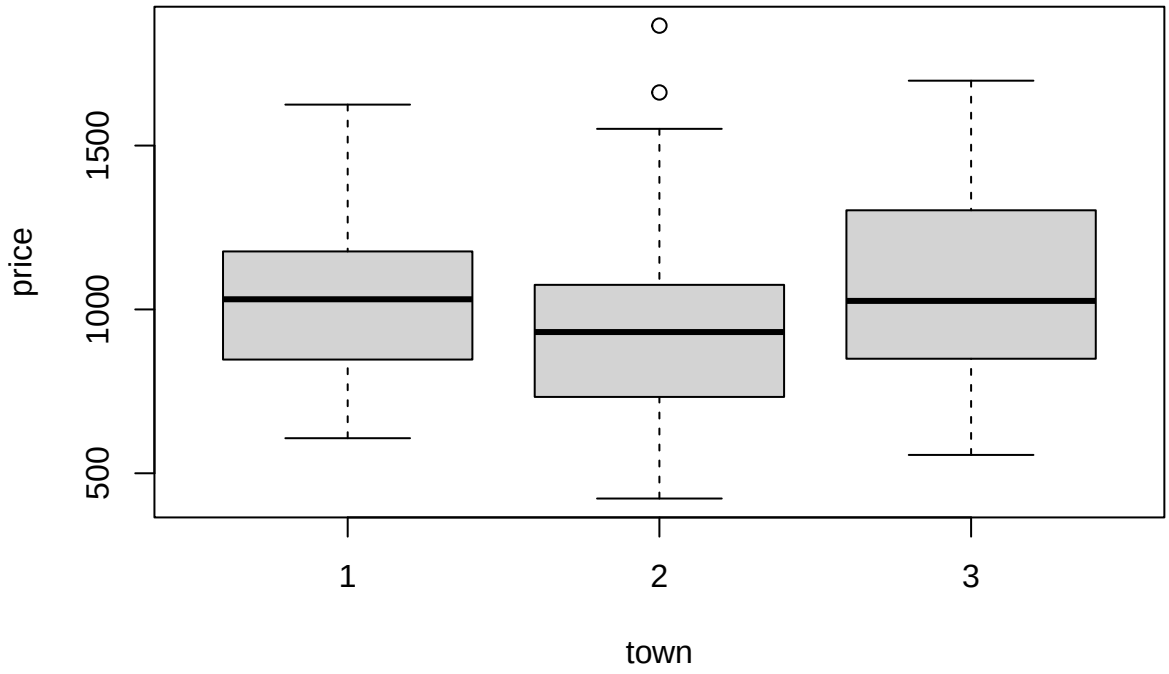
```
with(flats, tapply(price, town, mean))
```

```
##        1        2        3
## 1008.122  949.500 1064.538
```

We see some clear price differences on average.

b. We can do

```
with(flats, boxplot(price ~ town))
```



This points clearly to the same differences, while also showing a slightly more widespread distribution in Ålesund.

c. We want to test whether "true" mean prices are different, i.e. something like

$$H_0 : \mu_M = \mu_K \quad \text{vs} \quad \mu_M \neq \mu_K$$

where the μ's are means in Molde, Kristiansund. We use the `t.test` as suggested, after excluding town 3 from the data. The "two.sided" is the default alternative, and need not be specified.

```r
flats_MK <- subset(flats, town != 3)
with(flats_MK, t.test(price ~  town))
```

```
##
##  Welch Two Sample t-test
##
## data:  price by town
## t = 1.1339, df = 99.663, p-value = 0.2596
## alternative hypothesis: true difference in means between group 1 and group 2 is not e
## 95 percent confidence interval:
##  -43.95448 161.19839
## sample estimates:
## mean in group 1 mean in group 2
##        1008.122         949.500
```

```r
# or: t.test(flatsMK$price ~ flatsMK$town)
```

The relatively high P-value means we can not reject the null hypothesis.

For Ålesund, Kristiansund, we run the same procedure

```r
flats_KA <- subset(flats, town != 1)
with(flats_KA, t.test(price ~  town))
```

```
##
##  Welch Two Sample t-test
##
## data:  price by town
## t = -1.9751, df = 82.279, p-value = 0.05161
## alternative hypothesis: true difference in means between group 2 and group 3 is not e
## 95 percent confidence interval:
##  -230.8975312    0.8206081
## sample estimates:
## mean in group 2 mean in group 3
##         949.500        1064.538
```

Supposing the significance level is 0.05, the P-value here is at the limit, but still does not lead to a rejected null hypothesis.

d. The variable is called `area`. We can use the method from a).

```
with(flats, tapply(area, town, mean))
```

```
##         1         2         3
##  95.92683 100.47143  98.69231
```

So, on average the sample has somewhat different sized flats.

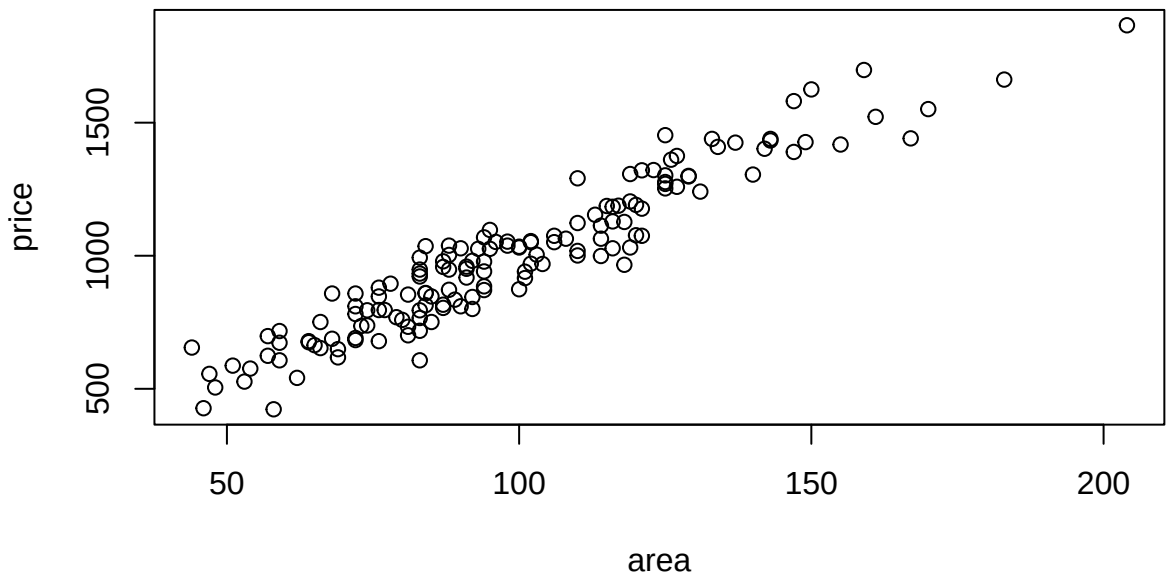
e. For example,

```
with(flats, cor(price, area))
```

```
## [1] 0.9503176
```

Strong correlation, at 0.95. In this connection, showing a scatterplot does not hurt.

```
with(flats, plot(area, price))
```



That looks more or less as expected.

f. We can do the computation as follows. Square meter prices are x1000 NOK

```
flats$sqmprice <- flats$price / flats$area
head(flats)
```

```
##   price area rooms standard situated town distcen age rent  sqmprice
## 1  1031  100     3        2        6    1       5  15 2051 10.310000
## 2  1129  116     3        1        5    1       4  42 2834  9.732759
## 3  1123  110     3        2        5    1       3  25 2468 10.209091
## 4   607   59     2        3        5    1       6  25 1940 10.288136
## 5   858   72     2        3        4    1       1  17 1611 11.916667
## 6   679   64     2        2        3    1       3  17 2039 10.609375
```

g. So, we do a t test again. Note, we need to recalculate the subset to include the `sqmprice` variable. So we can do

```
flats_MK <- subset(flats, town != 3)
with(flats_MK, t.test(sqmprice ~  town))
```

```
##
##  Welch Two Sample t-test
##
## data:  sqmprice by town
## t = 6.6785, df = 75.642, p-value = 3.581e-09
## alternative hypothesis: true difference in means between group 1 and group 2 is not e
## 95 percent confidence interval:
##  0.7935811 1.4681197
## sample estimates:
## mean in group 1 mean in group 2
##        10.63212         9.50127
```

The P-value is almost 0, and much less than 0.05, so we reject the null hypothesis. We conclude that square meter prices are significantly higher in Molde than in Kristiansund.

h. When comparing nominal prices, we do not take into account that while the prices were on average higher in Molde, the flats were also smaller in Molde. So, comparing nominal prices can be misleading when we don't control for the fact that sizes differ on average. Looking at square meter prices is one way to make a more "fair" comparison as the size is taken into account.

i. So, we can make a few more subsets, and run similar tests:

```
flats_KA <- subset(flats, town != 1)
flats_MA <- subset(flats, town != 2)

with(flats_KA, t.test(sqmprice ~ town))
```

```
##
##  Welch Two Sample t-test
##
## data:  sqmprice by town
## t = -8.0705, df = 72.688, p-value = 1.066e-11
## alternative hypothesis: true difference in means between group 2 and group 3 is not e
## 95 percent confidence interval:
##  -1.695627 -1.023977
## sample estimates:
## mean in group 2 mean in group 3
##         9.50127        10.86107
```

```
with(flats_MA, t.test(sqmprice ~ town))
```

```
##
##  Welch Two Sample t-test
##
## data:  sqmprice by town
## t = -1.1576, df = 77.974, p-value = 0.2505
## alternative hypothesis: true difference in means between group 1 and group 3 is not e
## 95 percent confidence interval:
##  -0.6226931  0.1647894
## sample estimates:
## mean in group 1 mean in group 3
##        10.63212        10.86107
```

We see H_0 rejected when comparing Kristiansund and Ålesund, while not when comparing Molde and Ålesund.

As a final remark, when working with categorical variables like `town` here, it can be worthwile to convert to a "factor". Some code for this is shown in section 7.5 in the compendium. If we do this and rerun question a, we get

```
with(flats, tapply(price, town, mean))
```

```
##    Molde   Krsund  Alesund
## 1008.122  949.500 1064.538
```

So instead of constantly trying to remember what was 1, 2, 3 - we now get the actual names in the output.

4.4

a. If `flats` is not already in your working environment, read it from file.

```
flats <- read.csv("M:/Undervisning/Undervisningh21/Data/flat_prices.csv")
head(flats)
```

Following the suggested way in the exercise, we go as follows.

```
#make new variable indicating small/not-small values
flats$small <- (flats$rooms < 3)

#count small/not-small
countsm <- table(flats$small)
countsm
```

```
##
## FALSE  TRUE
##    91    59
```

b. Since we want the `TRUE` count to be in position 1, when using the `binom.test` function, we can "reverse" the count vector for example as follows, and then do the test.

```
countsm <- countsm[2:1]
testresult <- binom.test(countsm, p = 0.5, alternative = "less")
```

Then we can for example inspect the whole `testresult` object:

```
testresult
```

```
##
##  Exact binomial test
##
## data:  countsm
## number of successes = 59, number of trials = 150, p-value = 0.005561
## alternative hypothesis: true probability of success is less than 0.5
## 95 percent confidence interval:
##  0.0000000 0.4634347
## sample estimates:
## probability of success
##              0.3933333
```

From this we see that we get the correct count of 59 "successes". (We always want p to be the "success probability" even though the word "success" is not always very meaningful, like here...)

Further, we find $\hat{p} = 0.393$, which is on the critical side. We get the p-value about 0.006, so we clearly reject H_0. We find strong evidence that the proportion of "small" flats is less than 0.5 (or 50%).

4.5

We continue with the same dataframe `flats` as above.

a. One typical reason why a variable Y can be approximately normal, is when Y can be thought of as the sum of many relatively small (or equal-sized) independent factors, like

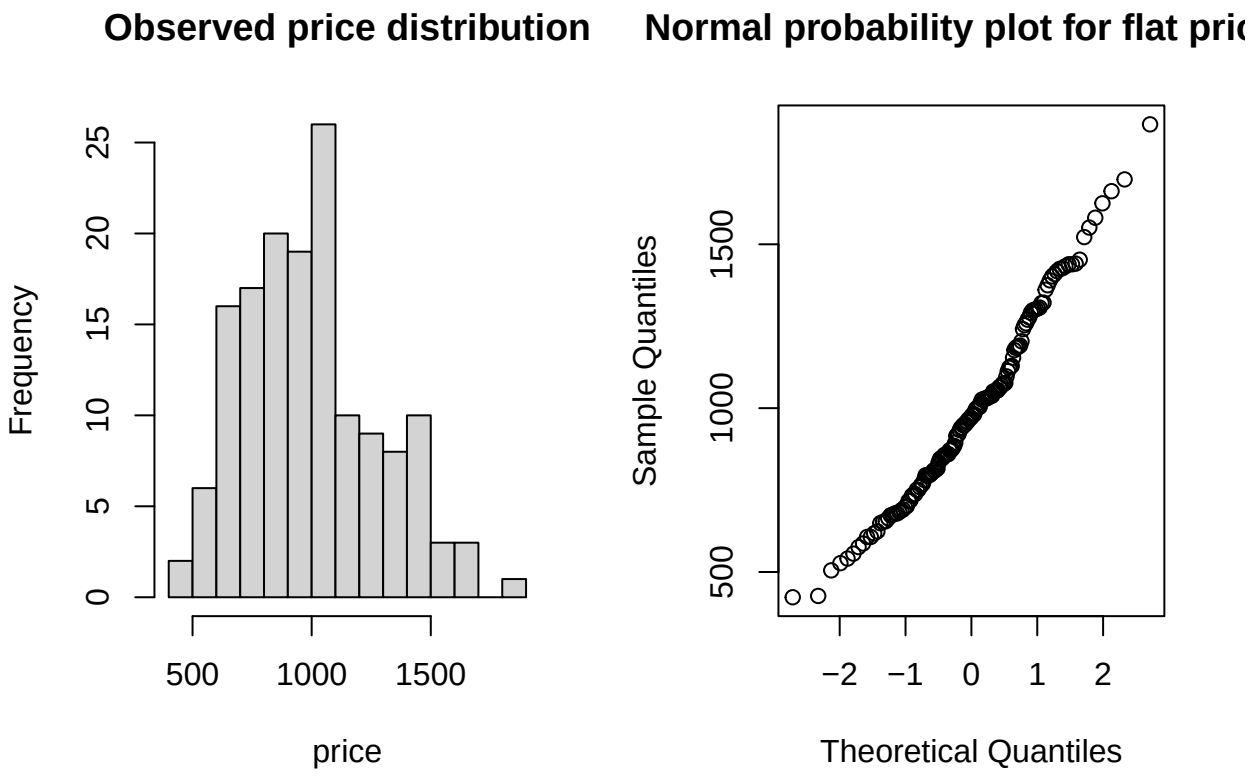
$$Y = X_1 + X_2 + X_3 + \cdots + X_N$$

So on the one hand, we could argue that the flat price is affected by a lot of different factors, which ultimately lead to something like a normal distribution for prices of a bunch of flats.

On the other hand, we may suspect that there is a particular "luxury" effect, where a few flats are considerably more expensive than the main bulk of flats, and that there is no corresponding negative effect, since flats in a horrible state would typically be upgraded to a reasonable level before sale. So a guess can be that the prices are a bit right skewed, while you remove the most expensive ones, the remaining "ordinary" flats might have almost normally distributed prices. Since we have the data, we can go on and have a look:

b. Here we are going to lump all prices together, ignoring the fact that we have three separate markets involved. We can make a normal-plot, and run the Shapiro-Wilk test. By the way, it also makes sense to look at a histogram, to get a general feel for how prices are distributed.

```
par(mfrow=c(1,2)) # allow side by side plots
with(flats, hist(price, breaks = 20,
                 main = "Observed price distribution"))

with(flats,  qqnorm(price,
       main="Normal probability plot for flat prices")) # normal plot
```




```
par(mfrow=c(1,1)) # reset plotting parameters
```

(Note that the code above could also be written `hist(flats$price, ...)` and so on. the use of `with(..)` is a matter of "taste".

The plots reveal some tendency of right skewedness, as suggested in a. We can do a test to see whether the effect is significant, so as to reject the H_0 stating that data come from a normal distribution.

```
with(flats, shapiro.test(price))    #or shapiro.test(flats$price)
```

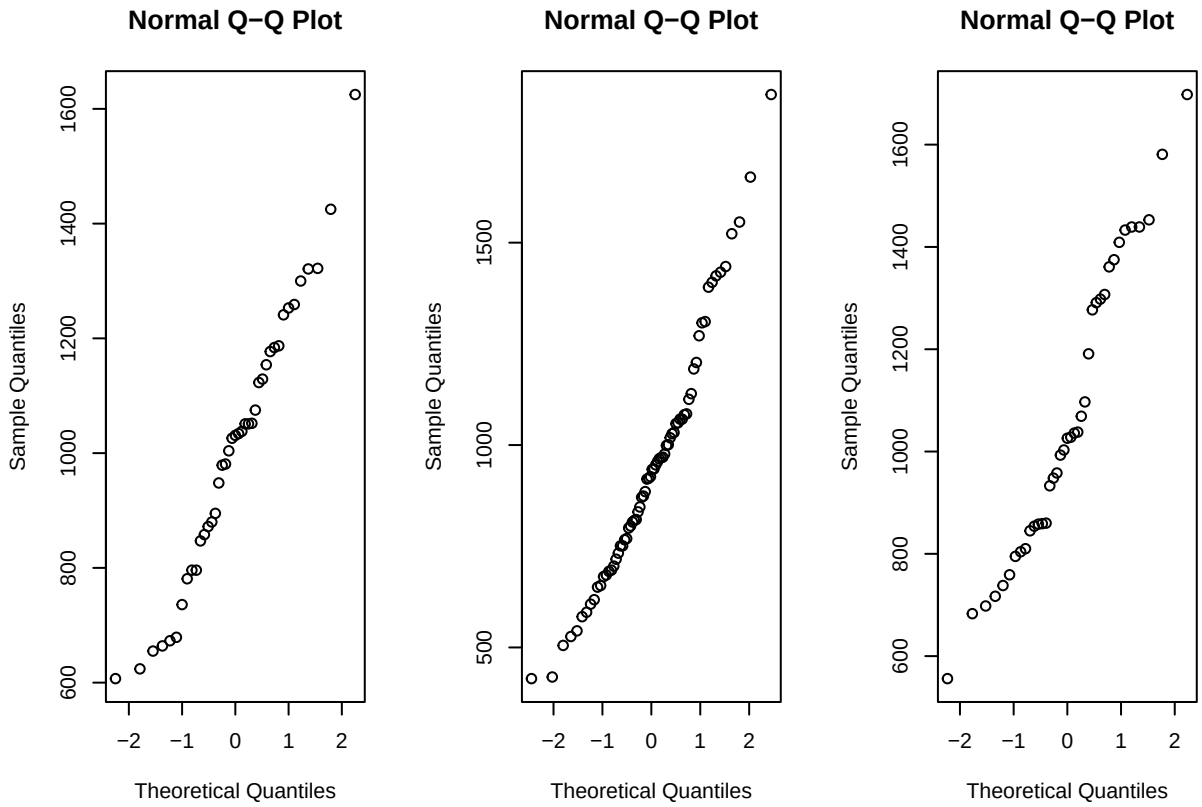
```
##
##  Shapiro-Wilk normality test
##
## data:  price
## W = 0.97897, p-value = 0.02126
```

The p-value is certainly below 0.05, so we will reject H_0 in this case.

c. The lumping together of prices from separate markets can be "dangerous" in this setting. Imagine for example that general price levels were much higher in one town,

but still normally distributed within each town. Then, the mixed price distribution can be very different from normal. We can show some experiments verifying this after answering the questions in the exercise. For the plots and testing town-wise, we can do as follows. Note that the `tapply` function does not work optimally with plots, as some additional output is created. We can stop this by putting the command inside `innvisible(...)`

```
par(mfrow=c(1,3))
invisible(with(flats, tapply(price, town, qqnorm)))
```



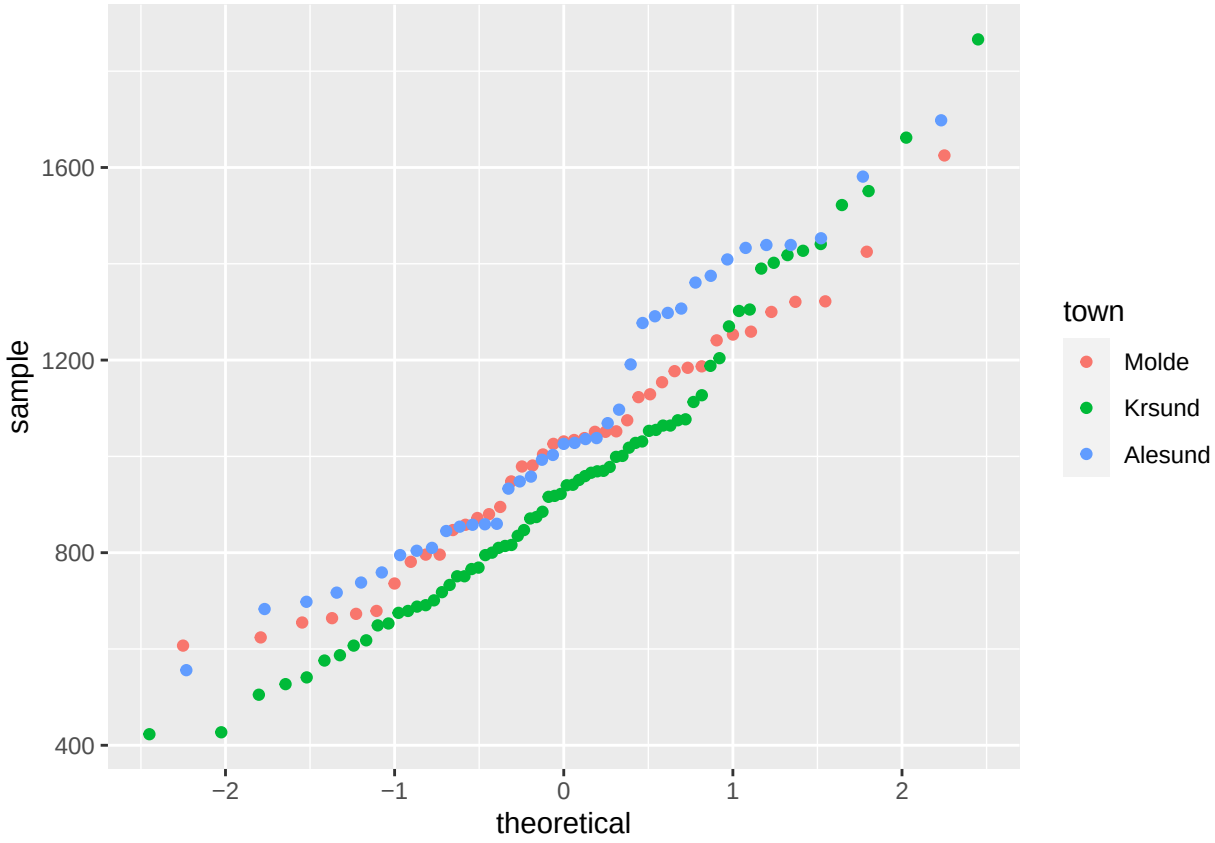

```
par(mfrow=c(1,1))
```

Here is a case where the `ggplot2` package offers superior functionality:

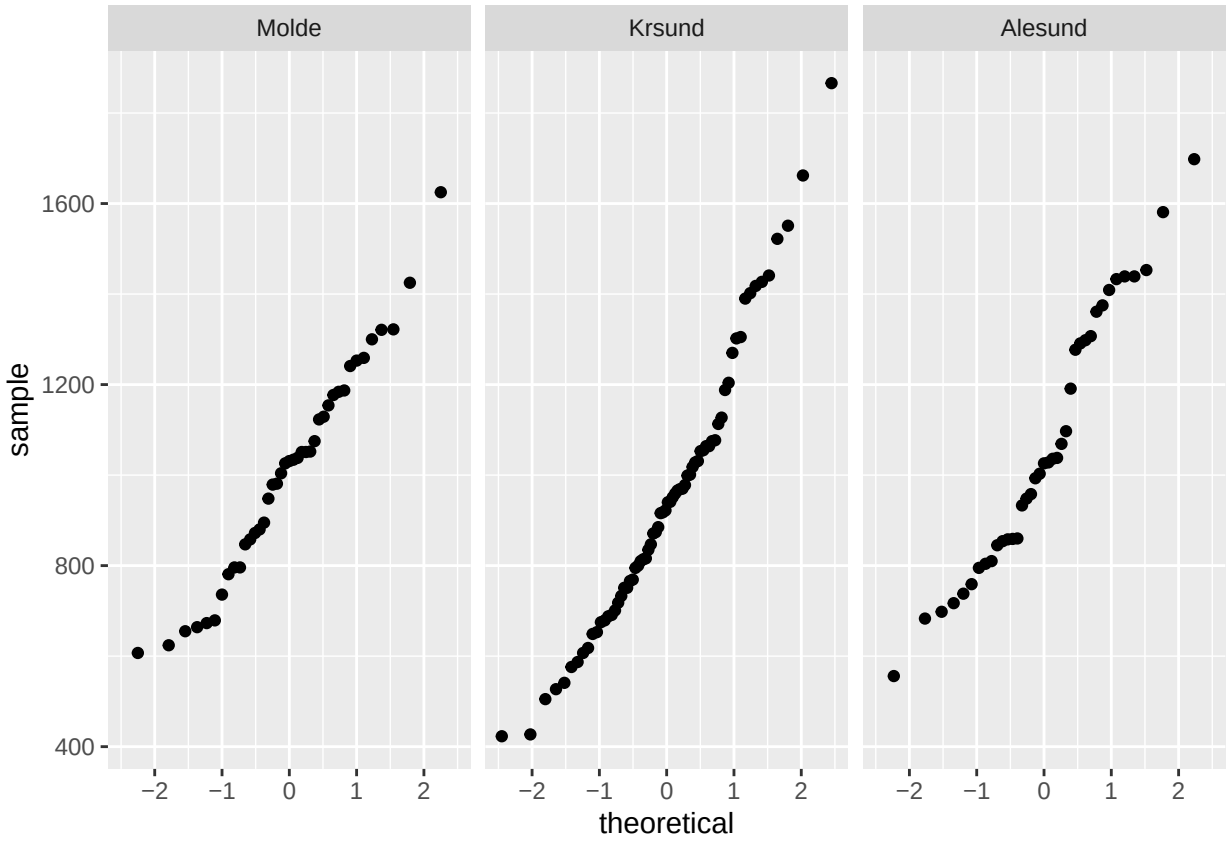
```
library(ggplot2)
```

```
## Warning: package 'ggplot2' was built under R version 4.1.3
```

```
ggplot(flats, aes(sample = price, color = town)) + geom_qq()
```




```
#or

ggplot(flats, aes(sample = price)) + geom_qq() + facet_wrap(~town)
```



We see the curving tendency in each town also. Let's test within towns.

```
with(flats, tapply(price, town, shapiro.test))
```

```
## $Molde
##
##  Shapiro-Wilk normality test
##
## data:  X[[i]]
## W = 0.97683, p-value = 0.5577
##
##
## $Krsund
##
##  Shapiro-Wilk normality test
##
## data:  X[[i]]
## W = 0.96312, p-value = 0.03732
##
##
## $Alesund
```

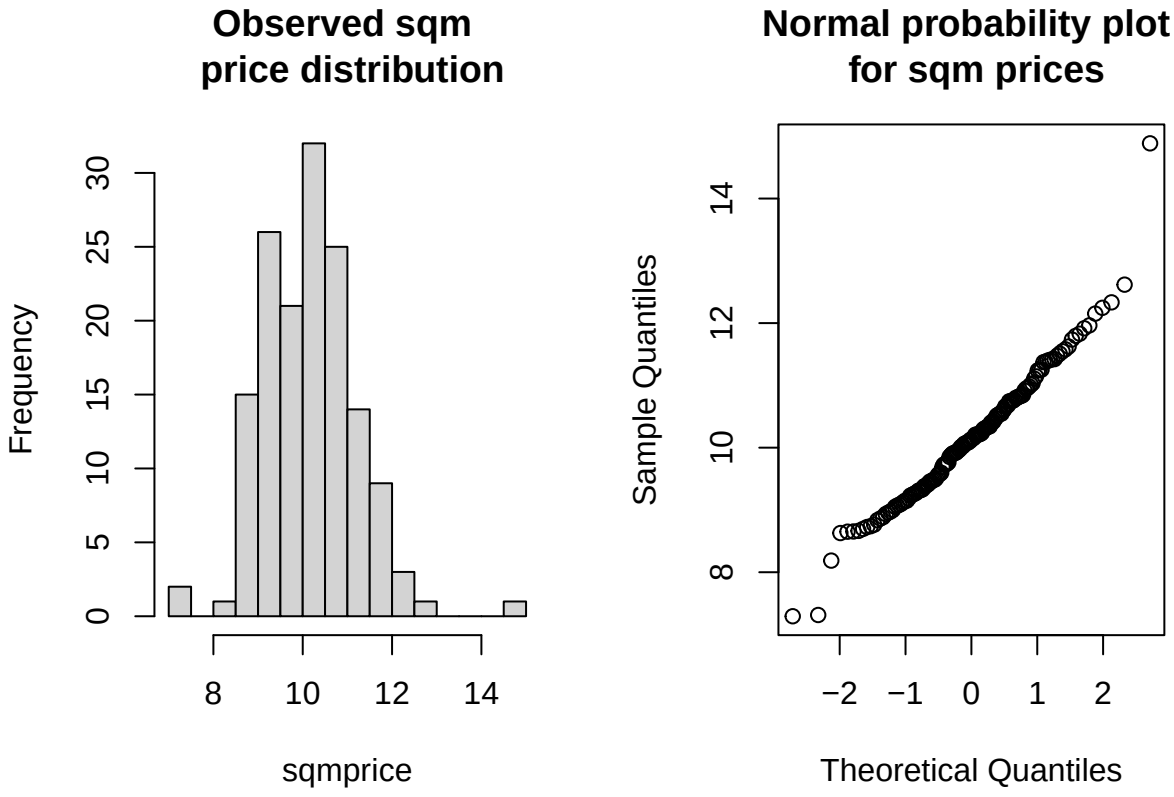
```
##
##  Shapiro-Wilk normality test
##
## data:  X[[i]]
## W = 0.95503, p-value = 0.1215
```

Only in the case of Kristiansund can we now reject the H_0. The most likely explanation for not being able to reject H_0 in Molde, Ålesund is simply that the sample sizes are much smaller when we disaggregate the data, and so we will need a *stronger* effect for it to be significant. Based on the qq-plots, we still should suspect that the data are not perfectly normal in any town, but that the deviation from normal is not dramatic.

d. We could just repeat the process with `sqmprice`.

overall picture: We look first at the aggregated price data.

```
par(mfrow=c(1,2)) # allow side by side plots
with(flats, hist(sqmprice, breaks = 20,
                 main = "Observed sqm \n price distribution"))

with(flats,  qqnorm(sqmprice,
       main="Normal probability plot \n for sqm prices")) # normal plot
```



```
par(mfrow=c(1,1)) # reset plotting parameters
```

Overall, there are a few flats sticking out on high and low square meter prices.

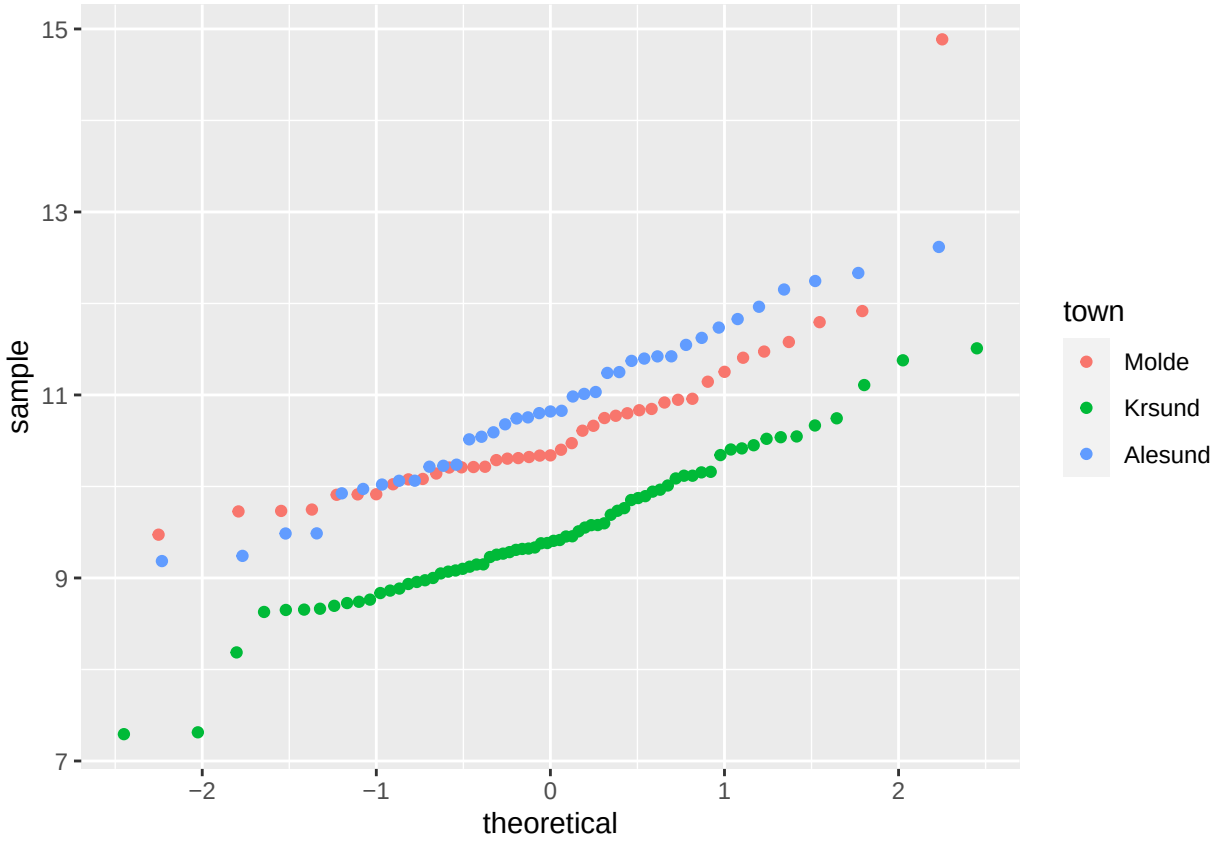
```
with(flats, shapiro.test(sqmprice))
```

```
##
##  Shapiro-Wilk normality test
##
## data:  sqmprice
## W = 0.97183, p-value = 0.003555
```

So, the results point clearly to square meter prices not being normal when considering the combined markets.

disaggregated analysis. We take a look within towns. "Cheating" by using ggplot, we get this:

```
library(ggplot2)
ggplot(flats, aes(sample = sqmprice, color = town)) + geom_qq()
```



Notably, a single flat in Molde is remarkably high, while 3 in Kristiansund fall considerably below the general level there. Running the Shapiro-Wilk test within towns:

```
with(flats, tapply(sqmprice, town, shapiro.test))
```

```
## $Molde
##
##  Shapiro-Wilk normality test
##
## data:  X[[i]]
## W = 0.76513, p-value = 1.088e-06
##
##
## $Krsund
##
##  Shapiro-Wilk normality test
##
## data:  X[[i]]
## W = 0.97158, p-value = 0.112
##
##
## $Alesund
##
##  Shapiro-Wilk normality test
##
## data:  X[[i]]
## W = 0.98426, p-value = 0.8505
```

In light of this, it is only in Molde that the test clearly rejects the H_0. One could further suspect that this is due to the single *outlier* flat that we see. We can check this, by running the test for Molde without that flat, e.g. as follows.

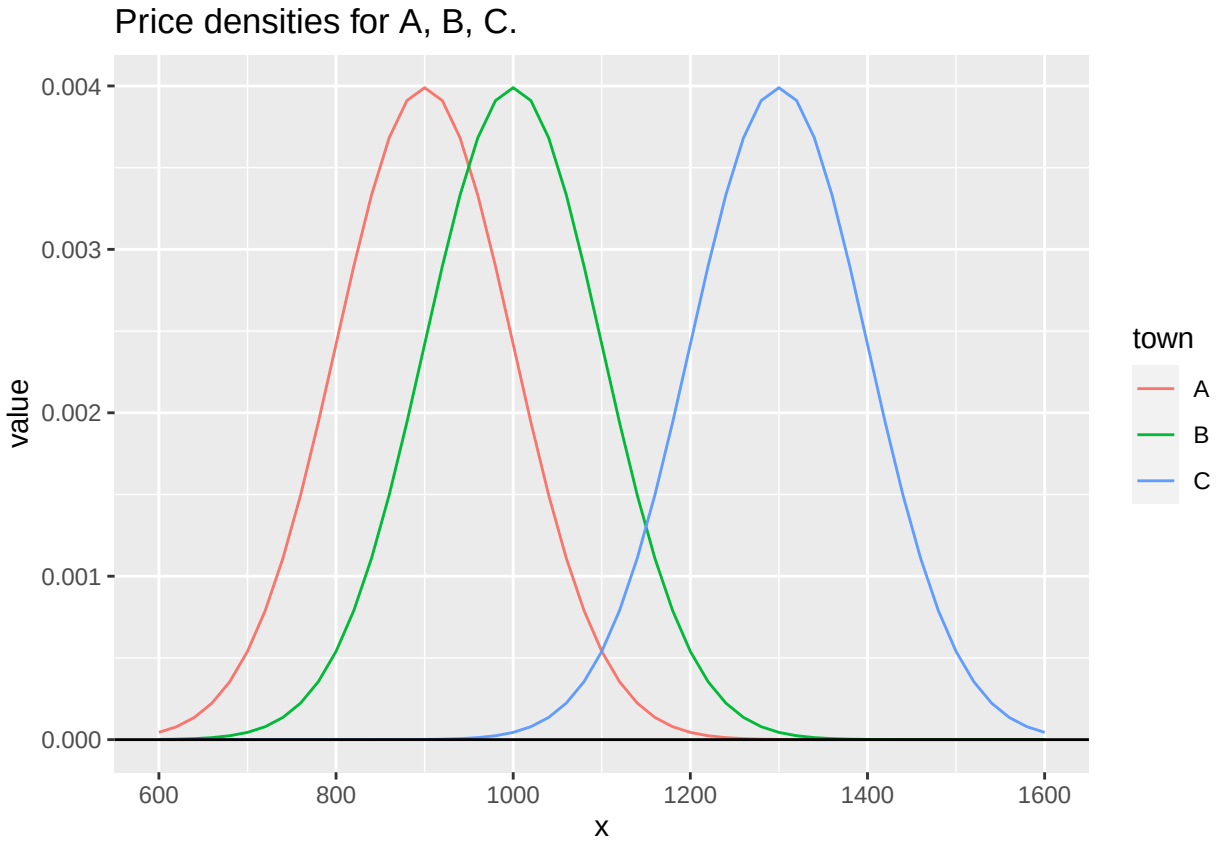
```
M <- max(flats$sqmprice) # find the maximum sqm price

 #make new dataframe with Molde flats of lower than max sqmprice:
df <- subset(flats, town = "Molde" & sqmprice < M)
#test again
with(df, shapiro.test(sqmprice))
```

```
##
##  Shapiro-Wilk normality test
##
## data:  sqmprice
## W = 0.97183, p-value = 0.003555
```

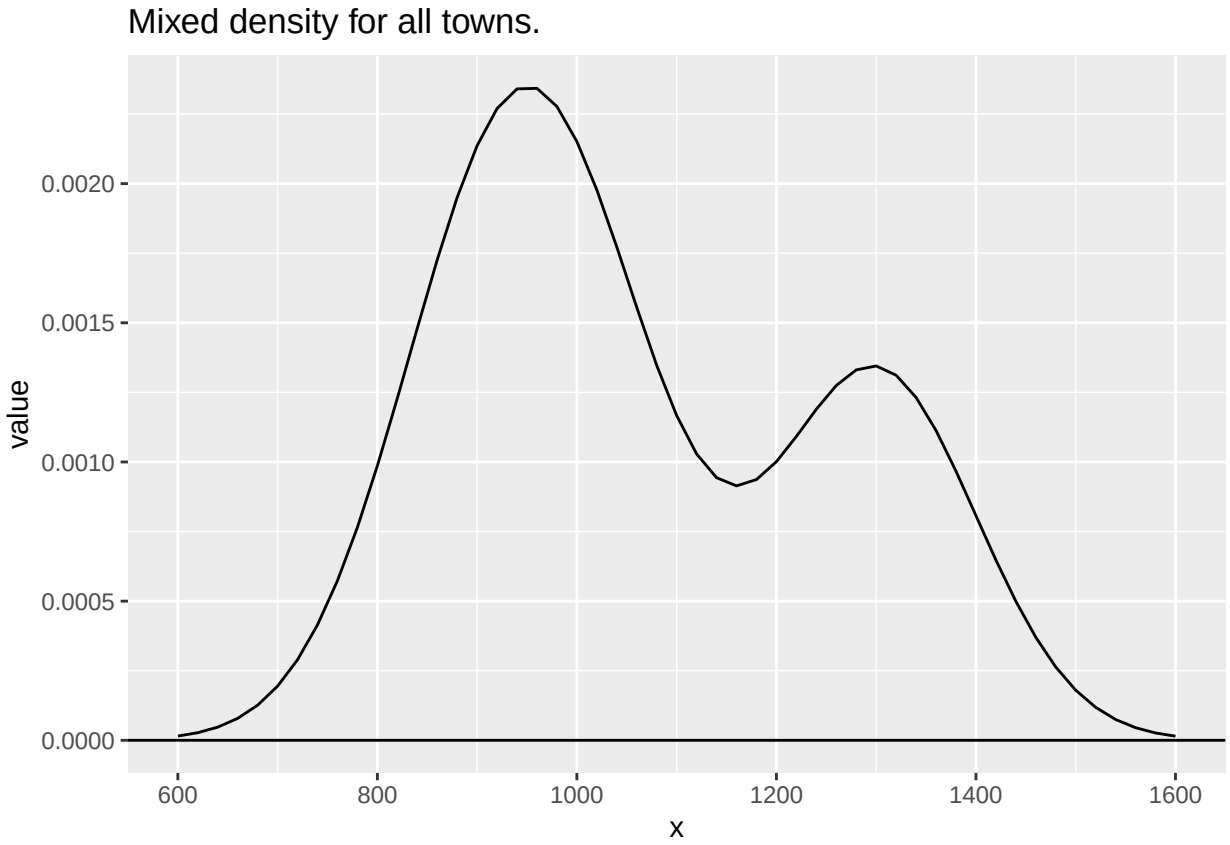
In fact, the pattern persists. We still reject H_0.

Addendum: The dangers of aggregated data.

Since this last exercise is about aggregated data, we may have a look at some general problems that can appear when aggregating data from possibly different probability distributions. Let's stick to the example of three towns A, B, C, and assume the prices within each town is normally distributed, with a standard deviation of 100, but that the mean prices are (say) 900, 1000 and 1300 respectively. Suppose further that exactly 1/3 of the flats are in each town. We can first make a picture of the three individual distributions.



Now look at the mixed density. The interpretation of this is that Y is the price of a randomly selected flat from A, B or C, and then we get a particular probability density for Y. The important thing to note is that this can be very different from a normal density, even though the price within each city is normal. With the particular setup here, we get the following.



So, the basic thing we can learn here, is that a variable can be normally distributed within each group (town), but when we mix the groups and look at the same variable, it will not usually be normal, unless the mean and standard deviation are similar in value.

Precisely the same thing can of course happen when we look at data, so suppose we now sample a set of values from each of the distributions above (A, B, C). Then we would expect to find nice `qqplots` within each town. On the other hand, the mixed data should reveal a non-normal distribution. Let's try.

```
#make data frame DFS with 100 samples from each town:
set.seed(1232) #always set seed before simulations
DFS <- data.frame(A = rnorm(100, mean=900,  sd=100),
                  B = rnorm(100, mean=1000, sd=100),
                  C = rnorm(100, mean=1300, sd=100))
#have a look
head(DFS)
```

```
##           A         B        C
## 1 1052.8871 1012.7667 1188.818
## 2 1030.7375 1067.6928 1284.532
## 3  991.6349  955.6578 1304.833
## 4  845.6964 1190.7237 1330.149
```

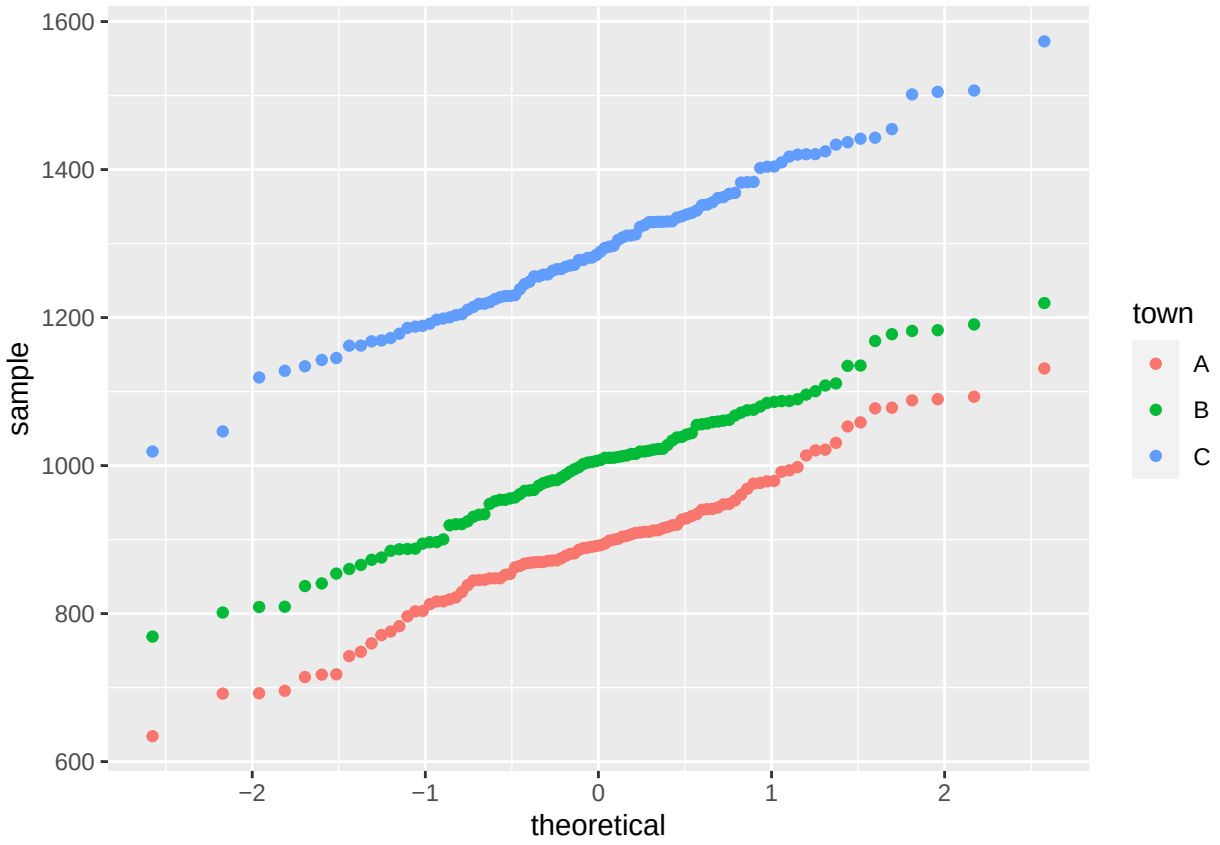
```
## 5  816.5266 1100.4579 1265.413
## 6  695.6770  965.9092 1329.912
```

```
#stack the columns in DFS (ggplot likes stacked data)
DFS2 <- pivot_longer(DFS, cols=A:C, names_to="town")

#have a look
head(DFS2)
```

```
## # A tibble: 6 x 2
##   town  value
##   <chr> <dbl>
## 1 A     1053.
## 2 B     1013.
## 3 C     1189.
## 4 A     1031.
## 5 B     1068.
## 6 C     1285.
```

Now we can make qqplots for each town.

```
ggplot(DFS2, aes(sample = value, color = town)) + geom_qq()
```



And we can do the Shapiro-test within each town.

```
with(DFS2, tapply(value, town, shapiro.test))
```

```
## $A
##
##  Shapiro-Wilk normality test
##
## data:  X[[i]]
## W = 0.98366, p-value = 0.2531
##
##
## $B
##
##  Shapiro-Wilk normality test
##
## data:  X[[i]]
## W = 0.99054, p-value = 0.7082
##
##
## $C
##
##  Shapiro-Wilk normality test
##
## data:  X[[i]]
## W = 0.99411, p-value = 0.9451
```

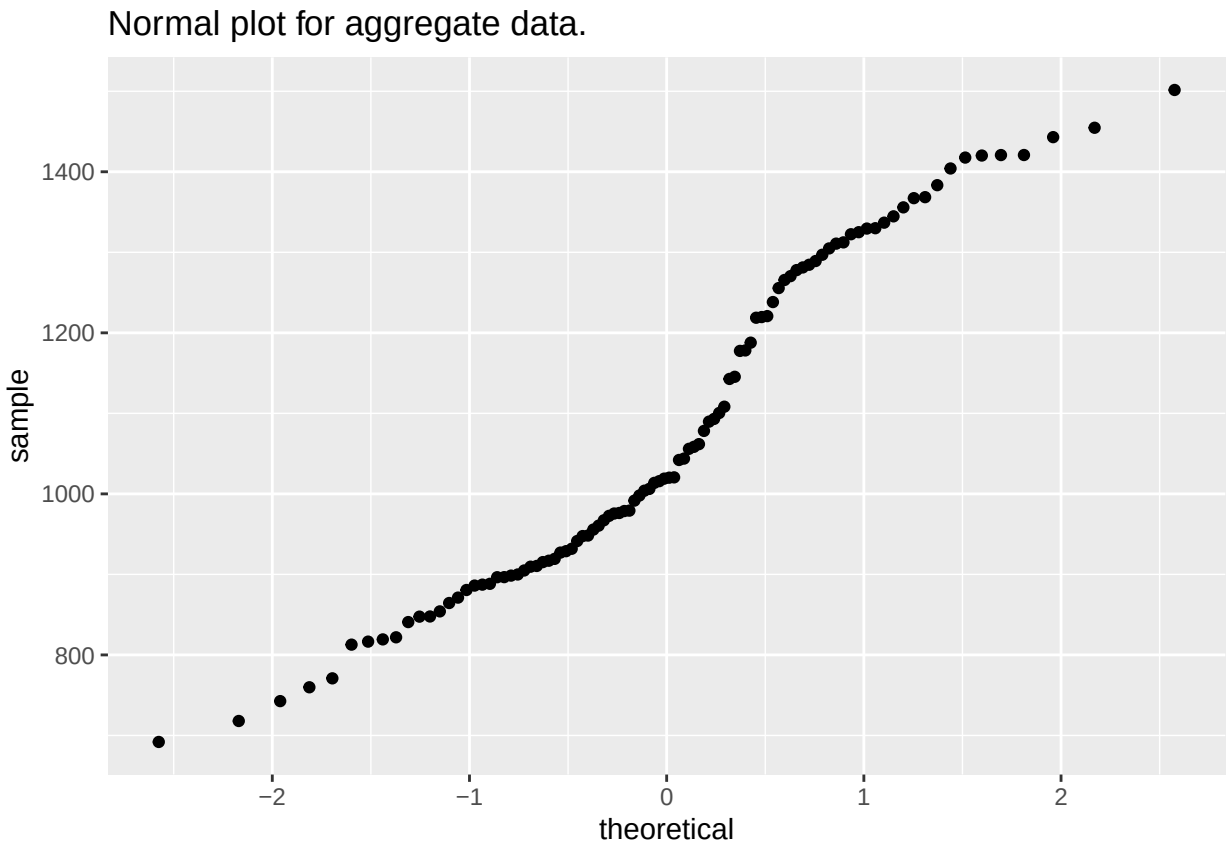
In each town, the null hypothesis survives the test.

Now, let's sample 100 prices from the whole set of data. This can be done by sampling 100 random numbers from 1-300, and pick the corresponding prices from `DFS2`.

```
set.seed(1212)
indices <- sample(1:300, 100)
aggregate_sample <- DFS2[indices, ]
head(aggregate_sample)
```

```
## # A tibble: 6 x 2
##   town  value
##   <chr> <dbl>
## 1 C     1420.
## 2 C     1345.
## 3 A      819.
## 4 A      692.
## 5 C     1285.
## 6 C     1270.
```

```
ggplot(aggregate_sample, aes(sample = value)) + geom_qq() +
  labs(title="Normal plot for aggregate data.")
```



So, as we see the normal plot for aggregate data shows clear deviations from normality. We can finish this discussion with a final Shapiro test:

```
with(aggregate_sample, shapiro.test(value))
```

```
##
##  Shapiro-Wilk normality test
##
## data:  value
## W = 0.94609, p-value = 0.0004628
```

So, based on the aggregate data, we seemingly find "evidence" that the prices are *not* normal.

The moral of this story is something like this: When we group together values of a variable that are recorded in (say) different towns, we risk to ignore systematic differences between the towns. Implicitly assuming that values are homogeneously distributed across towns can lead to seriously flawed conclusions regarding the variable in focus. Such problems extend way beyond the question about normal or non-normal distribution. We will get back to this in chapter 6, in a regression setting, where it is related to the concept of "omission bias".