A summary of all the good, the bad and the ugly

Outline

Learning Objectives

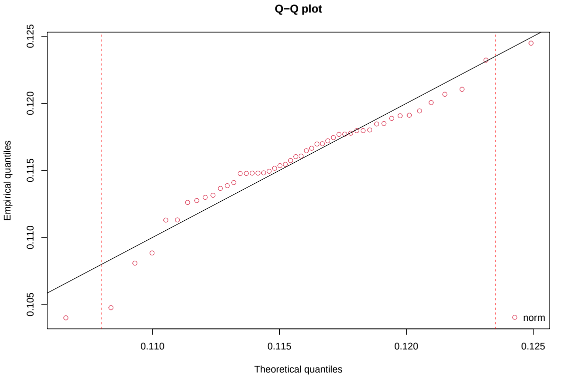
1. Learn methods used to identify outliers.
2. Understand simple wrapper functions.
3. Appreciate the key concepts discussed in this set of e-learning materials.

Outliers

Detecting outliers

Detecting outliers: Examine empirical and theoretical quantiles

- Examine the *empirical* and *theoretical* quantiles of our data.
  - In selecting a distribution $p(x|\theta)$ or $f(x|\theta)$, we assume that the population data follows the distribution.
  - E.g., Q-Q plot for `seatbelts_4$PetrolPrice` fitted to the normal distribution.

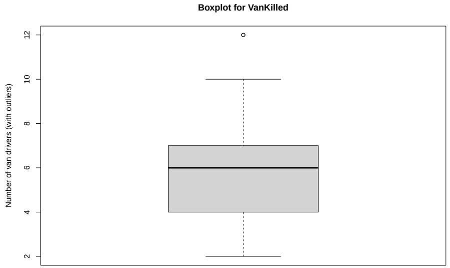


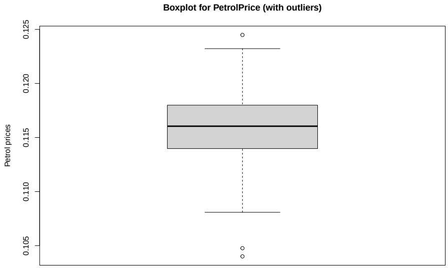
Detecting outliers: The 1.5 IQR rule

- The 1.5 (interquartile range) IQR rule.
  - We can use the `boxplot()` function to make boxplots.

```
boxplot(seatbelts_4$VanKilled,
        main = "Boxplot for VanKilled", ylab="Number of van drivers")
boxplot(seatbelts_4$PetrolPrice,
        main = "Boxplot for PetrolPrice", ylab="Petrol prices")
```

  - Outliers are taken to be data points 1.5 IQR from the first and third quartiles.

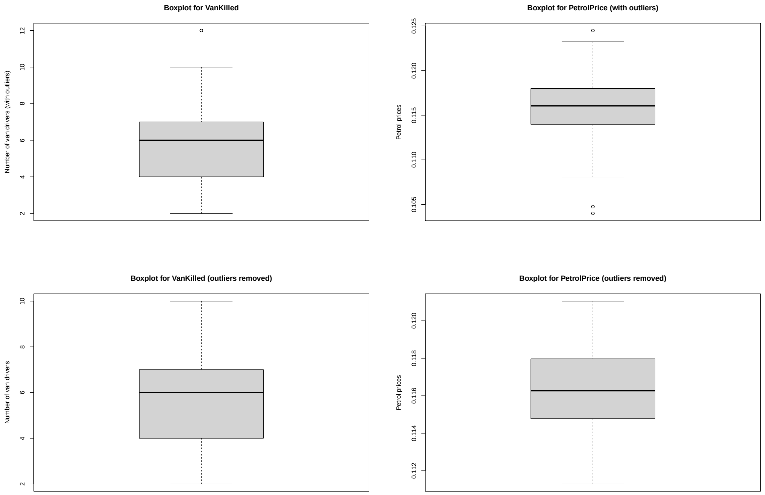




Detecting outliers: The 1.5 IQR rule

cont'd

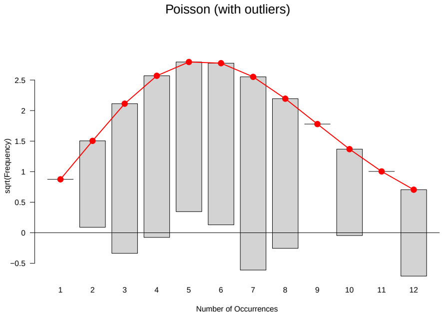
- Let us now remove the outliers.



Sensitivity to outliers: Discrete variables

Sensitivity to outliers: Discrete variables

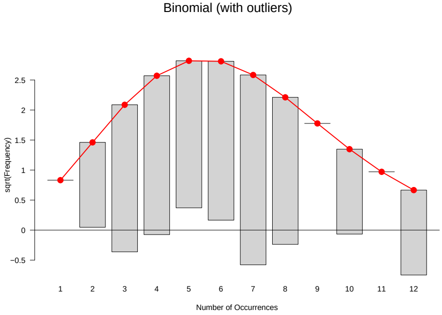
- Let us revisit our example involving `seatbelts_4$VanKilled`.
- For each model, let us briefly compare a pair of rootograms.
- Let us start with the *Poisson* model.

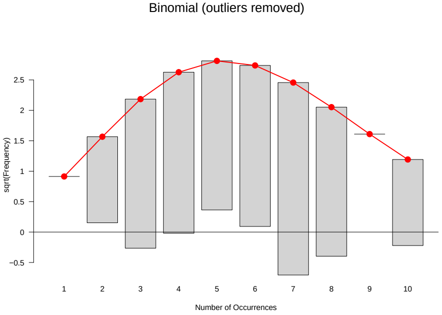


Sensitivity to outliers: Discrete variables

cont'd

- Let us do the same for the *binomial* model.

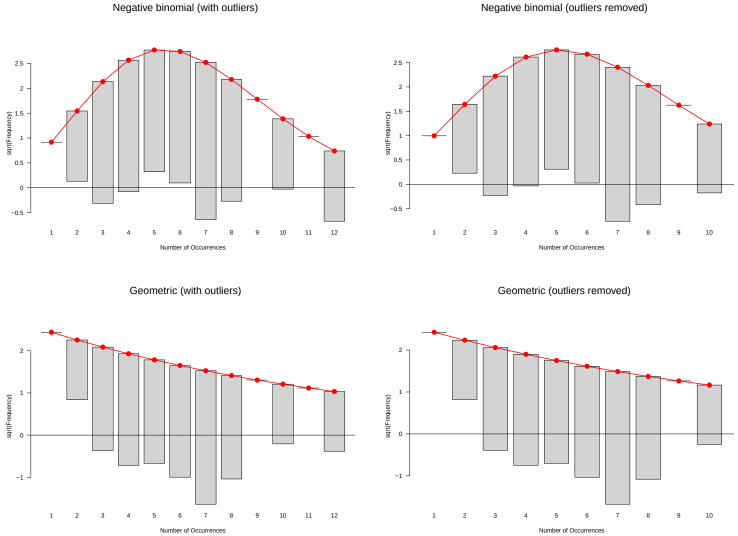




Sensitivity to outliers: Discrete variables

cont'd

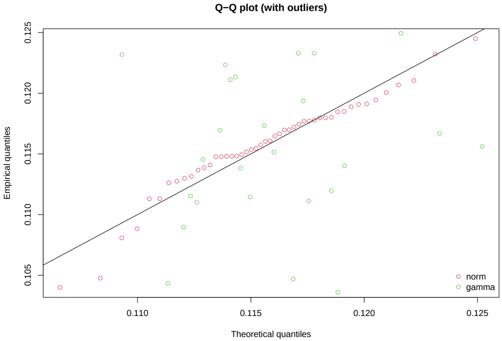
- Let us do the same for the *negative binomial* and *geometric* models.
- We will not always see an improvement upon removing outliers.
- In may cases, outliers can be important.

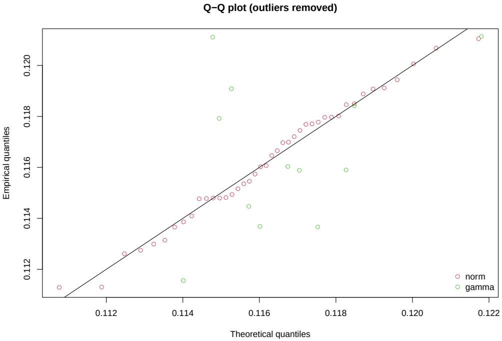


Sensitivity to outliers: Continuous variables

Sensitivity to outliers: Continuous variables

- Let us revisit our example involving `seatbelts_4$PetrolPrice`.
- For each model, let us plot a Q-Q plot.
- In the absence of outliers, model-fit for the normal distribution (in red) improved slightly.
- Outliers do not always impact the end result in an huge way.
- Outliers should be treated on a case-to-case basis.





Some wrapper functions

Some wrapper functions: a simple example

- In the following weeks, we will be using several wrapper functions.
- You are not expected to understand *all* components of these functions.
- Aim to be comfortable using these functions.
- Recall the following code:

```
norm_petrol <- fitdist(data = seatbelts_4$PetrolPrice,
                       distr = "norm")
qqcomp(norm_petrol)
```

- What if you want to choose a different model?
- How about a different dataset and variable?
  - You have to change the `distr` argument.
- What if you have many more steps?

Some wrapper functions: a simple example

cont'd

- We can create a custom function that is like a template.
- The following function will carry out two steps sequentially.

```
qqplot_dist <- function(vec, distname){
  fitted <- fitdist(data = vec, # Set up the data
                    distr = distname) # Select the model
  qqcomp(fitted) # Plot Q-Q plot
}
```

- This function is very similar to the earlier syntax:

```
norm_petrol <- fitdist(data = seatbelts_4$PetrolPrice,
                       distr = "norm")
qqcomp(norm_petrol)
```

- To obtain the Q-Q plot from before, we have to correctly specify the arguments `vec` and `distname`.

```
qqplot_dist(vec = seatbelts_4$PetrolPrice,
            distname = "norm")
```

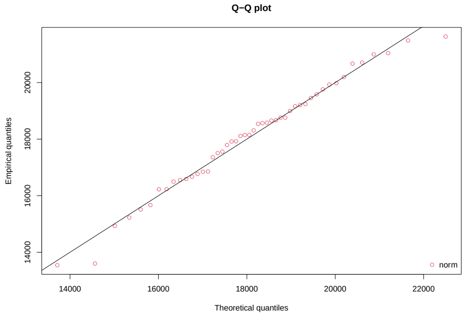
Some wrapper functions: a simple example (cont'd)

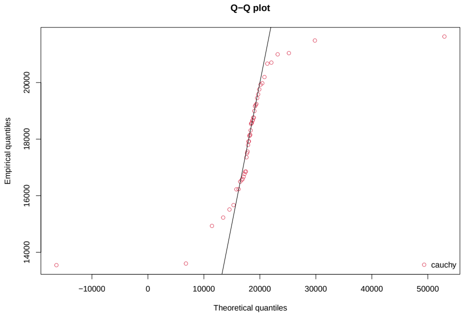
- We can then easily plot the Q-Q plot for a different variable, say, for `kms`.

```
qqplot_dist(vec = seatbelts_4$kms,
            distname = "norm")
```

- We can do the same for a different model, say, for the *Cauchy* distribution.

```
qqplot_dist(vec = seatbelts_4$kms,
            distname = "cauchy")
```





Some wrapper functions: a slightly more complicated example

- We can modify the `qqplot_dist()` function so that we can plot a Q-Q plot for several models at once.

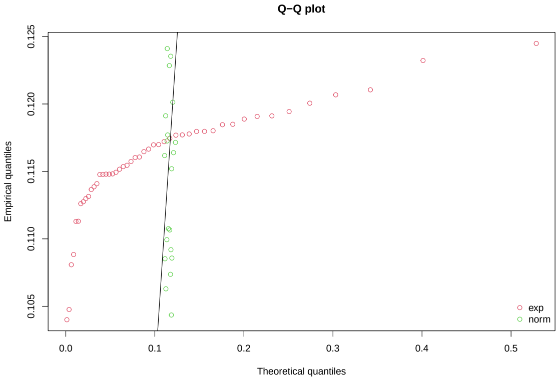
```
qqplot_dist_multi <- function(vec, distname_multi){
  vec <- as.vector(vec) # Ensure that vec is a vector
  distname_multi <- as.list(as.character(distname_multi))
  fitted <- lapply(distname_multi, fitdist, data=vec)
  qqcomp(fitted)
}
```

Some wrapper functions: a slightly more complicated example

cont'd

- To plot the Q-Q plot for both exponential and normal distributions, we can use this function.

```
qqplot_dist_multi(vec = seatbelts_4$PetrolPrice,
                  distname_multi = list("exp", "norm"))
```



Summary of concepts

The need for a vocabulary of distributions

The need for a vocabulary of distributions

The good:

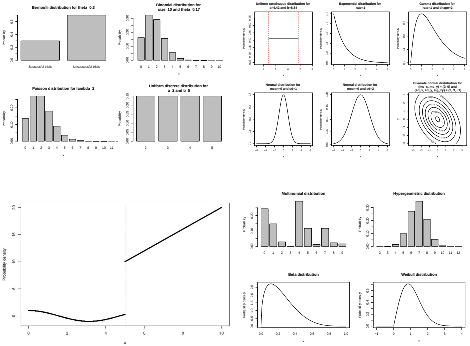
- We have seen several distributions.
- Strive to familiarise yourself with more common distributions.

The bad:

- Not all distributions can be mathematically formalised.
- Not always possible to describe a distribution in terms of parameters and the support.

The ugly:

- When encountering an unfamiliar distribution, focus on its parameters and support.



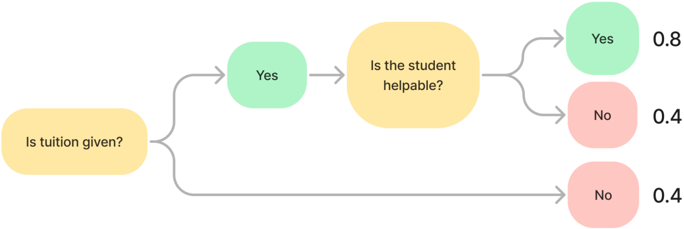
Looking forward:

- We will be using different distributions as building blocks for various simulation models.

A simple simulation

A simple simulation

- Suppose there are 4 students under a certain special programme.
  - Past experience tells us that 40% of students will get the bursary without extra help.
  - Let us then take 0.4 as the *empirical* probability of getting the bursary.
  - Each student can be treated as an individual Bernoulli trial.
  - Then, we expect 1.6 out of 4 students to obtain the bursary.
- Let we further suppose that we want to increase every student's chance of getting the bursary.
  - There are exactly two students are helpable.
    - ★ If one-to-one tuition is provided, the probability of obtaining the bursary for these students can be increased to 0.8.
  - However, there are only two tutors available.
- Is the strategy of providing tuition to *two randomly selected students* a good strategy?



A simple simulation: Using a wrapper function

- We shall use a wrapper function called `sim_students()` to simulate selecting two students randomly.

```
sim_students <- function(outcomes=FALSE){
students <- c("a", "b", "c", "d")
selected.students <- sample(x = students, size = 2)
pa <- 0.4;pb <- 0.4;pc <- 0.4;pd <- 0.4
if(selected.students[1] == "a"){pa <- 0.8}
if(selected.students[2] == "b"){pb <- 0.8}
if(selected.students[1] == "b"){pb <- 0.8}
if(selected.students[2] == "a"){pa <- 0.8}
if(outcomes == FALSE){
sum(rbinom(4, size = 1, prob = c(pa, pb, pc, pd)))
}else{rbinom(4, size = 1, prob = c(pa, pb, pc, pd))}
}
```

A simple simulation: Running 1 simulation

- Let us run the simulation once, and set the seed as 1.
  - Setting `outcomes = TRUE` allows us to view the outcomes.

```
set.seed(1)
sim_students(outcomes = TRUE)
```

```
## [1] 0 0 1 1
```

  - Let us run this again, this time without setting up the argument.

```
set.seed(1)
sim_students()
```

```
## [1] 2
```

A simple simulation: Running 100 simulations

- Using the `replicate()` function, let us now run the simulation 100 times, and set the seed as 123.

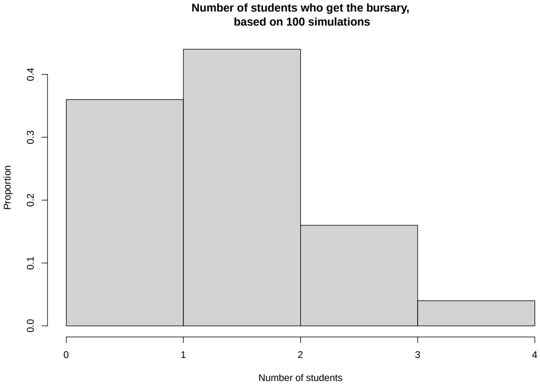
```
set.seed(123)
sim_100 <- replicate(n = 100,
                     sim_students())
```

- Next, we can compute the mean number of students who got the bursary.

```
mean(sim_100)
```

```
## [1] 1.84
```

- This means that 1.84 out of 4 students are expected to obtain the bursary with extra help given.
- Let us also plot the histogram to get a better idea on how the simulated values are distributed.



A simple simulation: Running 100 000 simulations

- Law of large numbers: running more simulations will bring the empirical probabilities closer to their theoretical counterparts.
- Let us now run 100 000 simulations.

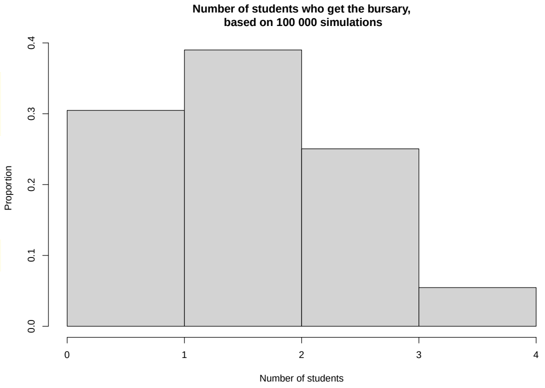
```
sim_100000 <- replicate(n = 100000,
                        sim_students())
```

- Again, we can compute the mean number of students who got the bursary.

```
mean(sim_100000)
```

```
## [1] 2.00241
```

- This means that 2 out of 4 students are expected to obtain the bursary with extra help given.
- Let us also plot the histogram to get a better idea on how the simulated values are distributed.



Summary

Summary

In this video, we have:

- Learned some methods used to identify outliers.
- Learned to use simple wrapper functions.
- Discussed the good, bad and ugly aspects of model-fitting.
- Discussed a simple simulation.

References

R-data — seatbelts dataset.

Hoaglin, D. C., Iglewicz, B., and Tukey, J. W. (1986).
Performance of some resistant rules for outlier labeling.
Journal of the American Statistical Association, 81(396):991–999.

Wasserman, L. (2004).
All of statistics springer new york.