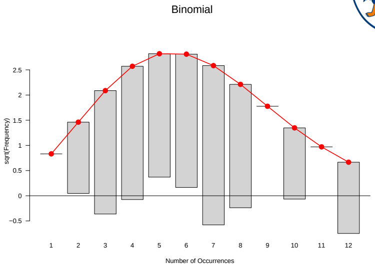
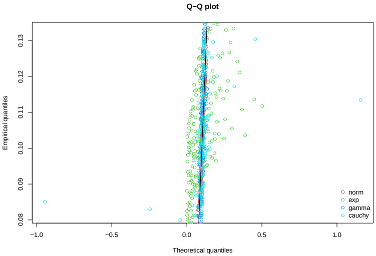
Evaluation of fitted models





Outline

Learning Objectives

1. Learn to use a rootogram for a discrete distribution.
2. Learn to use a quantile-quantile (Q-Q) plot for a continuous distribution.

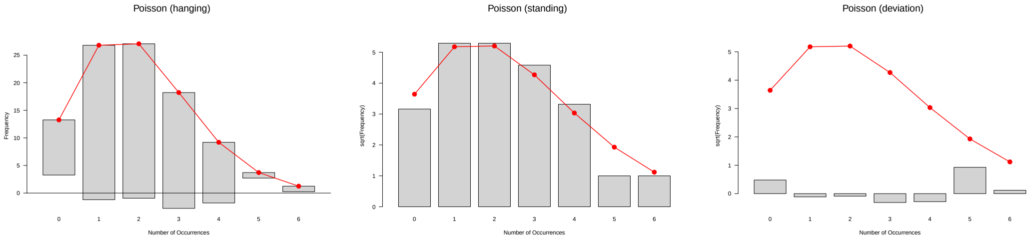
Evaluating fit to discrete distributions : Rootograms

Rootograms

Rootograms

- A rootogram shows if a count variable is well-fitted to a discrete distribution.
- There are three main types: hanging rootograms, standing rootograms, deviation-type rootograms.
- Consider a dummy dataset containing 100 data points, where $X \sim Poisson(x|\lambda = 2)$, generated in the following:

```
set.seed(1)
dummy_data <- rpois(n = 100,# Get 100 random numbers from Poisson
                    lambda = 2) # Set lambda = 2
```



Rootograms

cont'd

- We can use `table()` to obtain the frequency table.

```
(observed <- table(dummy_data))
```

```
##  0  1  2  3  4  5  6
## 10 28 28 21 11  1  1
```

- Suppose we do not know what distribution the data came from.
  - Suppose further that we determined that the Poisson model is appropriate.
  - We can now use `fitdist()` to estimate the parameter λ.

```
pois_par <- fitdist(data = dummy_data, # Use the dummy dataset
                                distr = "pois") # Select Poisson
(lambda <- pois_par$estimate[1]) # Store value of lambda as lambda
```

```
## [1] 2.02
```

Rootograms

cont'd

- Next, we can use dpois() to obtain fitted values.

```
fitted_pois <- dpois(x = 0:6,
                     lambda = lambda) * sum(observed)
```

- To plot the rootograms, we need rootogram() from the vcd package.
- Let us first load this library.

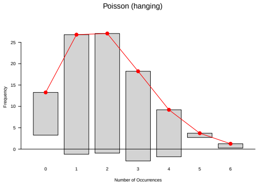
```
library(vcd)
```

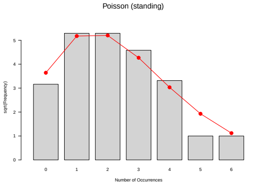
Rootograms (con'd)

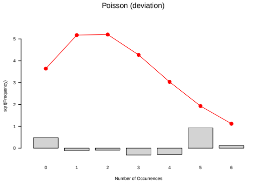
```
rootogram(x = observed,
          fitted = fitted_pois,
          type = "hanging",
          main = "Poisson (hanging)")
```

```
rootogram(x = observed,
          fitted = fitted_pois,
          type = "standing",
          main = "Poisson (standing)")
```

```
rootogram(x = observed,
          fitted = fitted_pois,
          type = "deviation",
          main = "Poisson (deviation)")
```

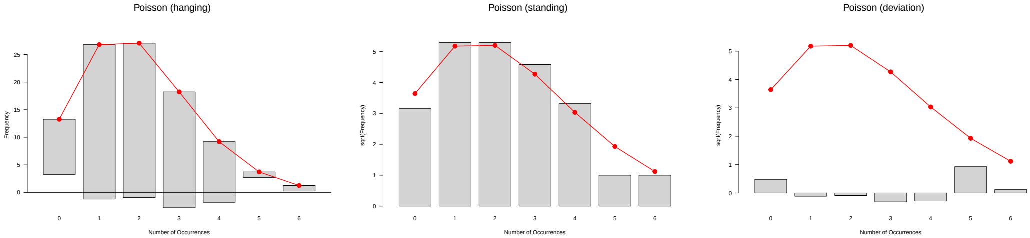






Rootograms

cont'd



- Hanging and standing types: Length of bar is proportional to the square root of each observed count.
- Hanging type: Conveys information about the deviation of fitted counts from observed counts.
  - When the bar starts from **above** the horizontal axis, the model **over-predicts**.
  - When the bar starts from below the horizontal axis, the model **under-predicts**.
  - A perfect fit will have all the bars start from the horizontal axis.
- We shall focus on **hanging** rootograms.

An example: Fitting `seatbelts_4$VanKilled`

An example: Fitting `seatbelts_4$VanKilled`

- Let us focus on the `VanKilled` variable.
- First, let us obtain the observed counts.

```
observed_killed <- tabulate(seatbelts_4$VanKilled)
names(observed_killed) <- 1:12
observed_killed
```

```
##  1  2  3  4  5  6  7  8  9 10 11 12
##  0  2  6  7  6  7 10  6  0  2  0  2
```

- Let us fit this to a *Poisson* model, and estimate the parameter λ.

```
pois_killed <- fitdist(data = seatbelts_4$VanKilled,
                       distr = "pois")
lambda <- pois_killed$estimate[1]
```

- Next, we can obtain the fitted values for $x = 1, 2, \cdots, 12$.

```
fitted_pois <- dpois(x = 1:12,
       lambda = lambda)* sum(observed_killed)
```

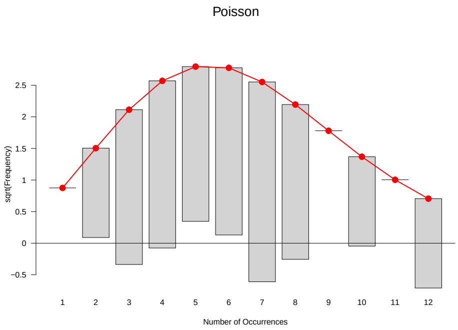
An example: Fitting `seatbelts_4$VanKilled`

cont'd

- Finally, use `rootogram()` to plot the rootogram.

```
rootogram(x = observed_killed,
          fitted = fitted_pois,
          main = "Poisson")
```

- Most of the bars start near the horizontal axis.
- Are there better distributions to fit our data to?

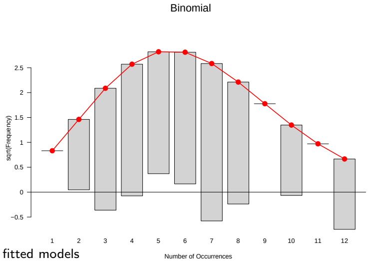


An example: Fitting seatbelts_4$VanKilled (cont'd)

Let us carry out the same steps for the *binomial* model.

```
binom_killed <- fitdist(data = seatbelts_4$VanKilled,
                        distr = "binom",
                    fix.arg=list(size=120),
                    start = list(prob=0.2))
theta <- binom_killed$estimate[1]
fitted_binom <- dbinom(x = as.numeric(names(observed_killed)),
       prob = theta,
       size = 120)* sum(observed_killed)
```

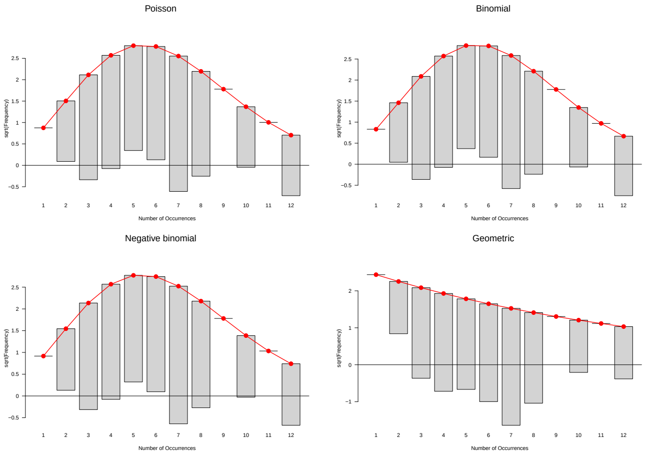
```
rootogram(x = observed_killed,
          fitted = fitted_binom,
          main = "Binomial")
```



An example: Fitting `seatbelts_4$VanKilled`

cont'd

- We shall do the same for *negative binomial* and the *geometric* models.
- Which of these models has the best fit?
  - None of these are perfect fits.
  - Will you be more willing to under-predict or over-predict?
  - Will you rather over-predict large values, or under-predict small values?

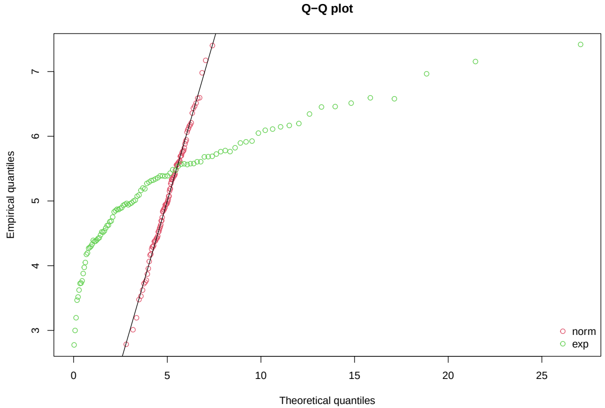


Evaluating fit to continuous distribution: Quantile-quantile (Q-Q) plots

Quantile-quantile (Q-Q) plots

Quantile-quantile (Q-Q) plots

- A quantile-quantile (Q-Q) plot visualises, for each data point, the empirical (observed) and theoretical (expected) quantiles.
- A small difference between observed and expected quantiles indicate a good fit for that data point.
- If most of the data points fit the distribution well, then they will populate the vicinity of a straight line with slope 1 and intercept at 0.
- Here, the data is better fitted to the normal distribution as compared to the exponential one.



Quantile-quantile (Q-Q) plots

cont'd

- Consider a dummy dataset containing 100 data points, where $X \sim \mathit{Normal}(x|\mu = 5, \sigma = 1)$:

```
set.seed(1)
dummy_data_cont <- rnorm(n = 100, # Get 100 random numbers from norm
                    mean = 5,
                    sd = 1)
```

- Suppose that we do not know what distribution should be selected.
  - Using `fitdist()`, let us fit dummy_data_cont to a normal distribution.

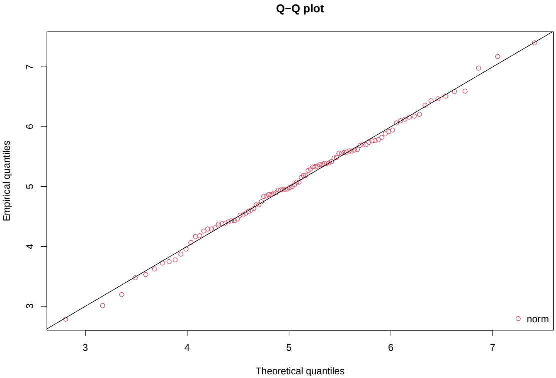
```
dummy_fitted_norm <- fitdist(data = dummy_data_cont,
       distr = "norm")
```

Quantile-quantile (Q-Q) plots

cont'd

- To plot the corresponding Q-Q plot, we can use the `qqcomp()` function from the `fitdistrplus` library.

```
qqcomp(dummy_fitted_norm)
```

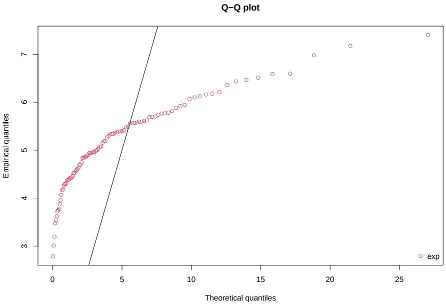


Quantile-quantile (Q-Q) plots

cont'd

- We can carry out the same steps to plot a Q-Q plot for a fit to the exponential distribution.

```
dummy_fitted_exp <- fitdist(data = dummy_data_cont,
        distr = "exp")
qqcomp(dummy_fitted_exp)
```

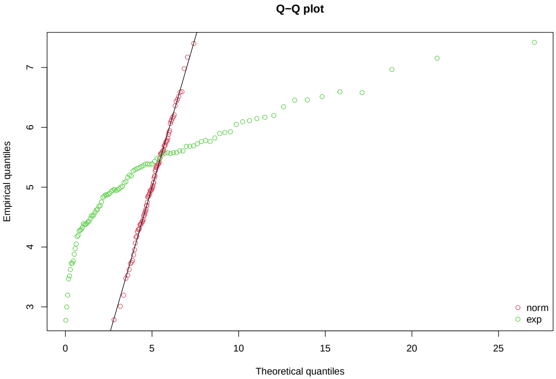


Quantile-quantile (Q-Q) plots

cont'd

- We can place both Q-Q plots in the same diagram by putting the two fitted models in a `list`.

```
qqcomp(list(dummy_fitted_norm, dummy_fitted_exp))
```



An example: Fitting `seatbelts_4$PetrolPrice`

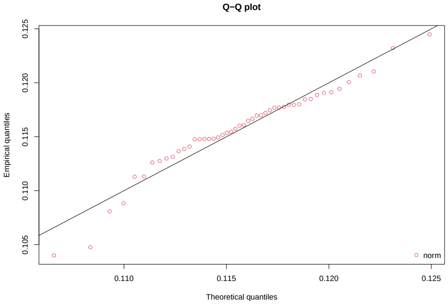
An example: Fitting `seatbelts_4$PetrolPrice`

- Let us focus on the `PetrolPrice` variable.
- We shall fit `seatbelts_4$PetrolPrice` to a *normal distribution*.

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

- Using `qqcomp()`, we can then obtain the Q-Q plot.

```
qqcomp(norm_petrol)
```



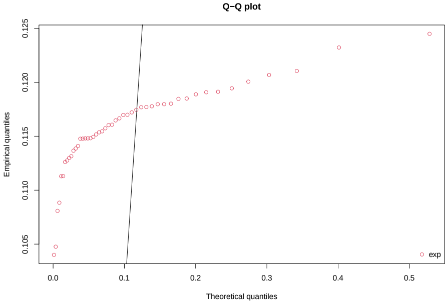
An example: Fitting seatbelts_4$PetrolPrice

cont'd

- We can repeat these steps for the *exponential* model.

```
exp_petrol <- fitdist(data = seatbelts_4$PetrolPrice,
        distr = "exp")
qqcomp(exp_petrol)
```

- The exponential model is unsuitable.



An example: Fitting `seatbelts_4$PetrolPrice`

cont'd

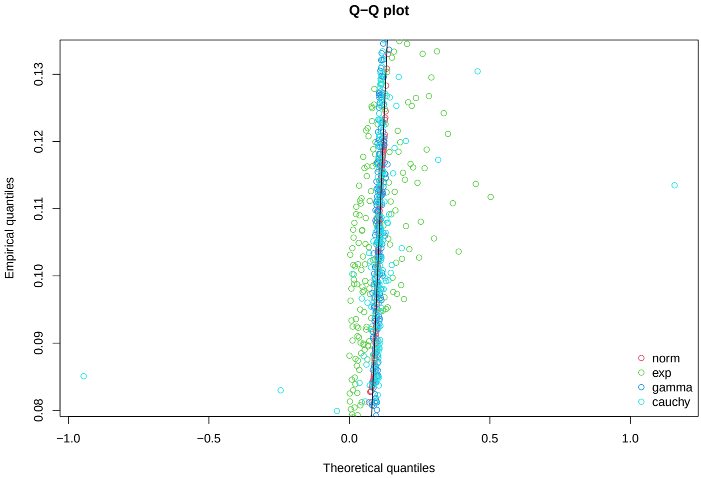
- Let us further repeat these steps for the *gamma* and *Cauchy* distributions.

```
gamma_petrol <- fitdist(data = seatbelts_4$PetrolPrice,
        distr = "gamma")
cauchy_petrol <- fitdist(data = seatbelts_4$PetrolPrice,
        distr = "cauchy")
qqcomp(list(norm_petrol,
            exp_petrol,
            gamma_petrol,
            cauchy_petrol))
```

An example: Fitting `seatbelts_4$PetrolPrice`

cont'd

- Which model is the most appropriate?
- Exponential and Cauchy distributions are *not* good candidates.
- Normal and gamma distributions are good candidates.
- There are other factors.
  - One may be interested in exploiting other properties of the distribution.



Summary

Summary

In this video, we have:

- Evaluated a model fit by plotting a *rootogram* for a *discrete* variable.
  - Visualise the discrepancies between observed and expected *counts*.
- Evaluated a model fit by plotting a *Q-Q plot* for a *continuous* variable.
  - Visualise the discrepancies between observed and expected *quantiles*.

References

R-data — seatbelts dataset.

Kleiber, C. and Zeileis, A. (2016).
Visualizing count data regressions using rootograms.
The American Statistician, 70(3):296–303.

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