Welcome to

MATHS 7107 Data Taming

Week 5, Trimester 1, 2024

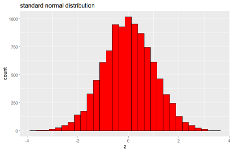
[Our Don Bradman]

Transforming Data

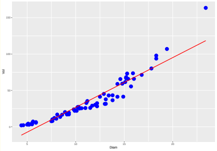
- A lot of statistical techniques rely on:
  - Histogram of single variable being like **normal/Gaussian** distribution "bell-curve"
  - Plot of 2 related variables being a **straight line**.
- But this isn't always true.
- But often we can **transform our data**.
- Standard transformations:
  - Standardisation
  - Min-max Scaling
  - Log transformation.
  - Box-Cox transformation.

We will look at two types of data:

- **Univariate**: measuring a single variable like height or age or number of cars
  - Use **mean**, **standard deviation**, **histograms**,...



- **Univariate**: measuring a connected pair of variables, like (height, age) or (number of cars, time of day).
  - Use **correlation**, **line-of-best-fit**, **scatterplots**,...



Univariate data

We'll start with univariate data

Standarisation

- Putting different variables on the same scale to compare scores between different types of variables.
- Let $(x_1, \ldots, x_n)$ be your sample observations
  - x_j be the observed value,
  - $\bar{x}$ be the mean observed value,
  - s be the sample standard devation
- Then the **standardised values**, **standard scores** or **z-scores** are

$$z_j = \frac{x_j - \bar{x}}{s}$$

- z_j has the same mean and standard deviation as standard normal distribution.

Standardisation

Eg. what is the more extreme food creation?

- The Octuple burger in Las Vegas
- Or Adelaide's largest pizza slice?

Let's say that we took a sample of 200 burgers

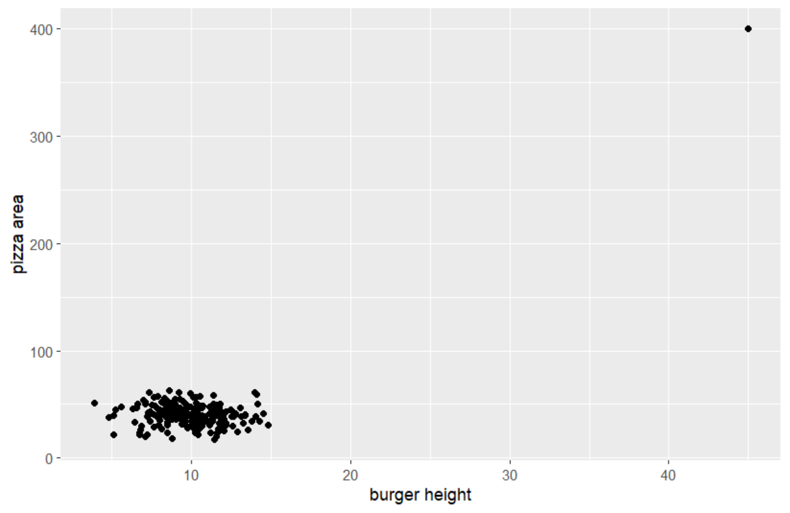
- $(b_1, \ldots, b_{200})$: sample burger heights (in cm)
- $\bar{b} = 10.17$: mean burger height
- $s_b = 3.19$: sample standard deviation of burger height
- The Octuple burger is $45cm$ high.

and 200 pizza slices

- $(p_1, \ldots, p_{200})$: sample pizza slice areas (cm^2)
- $\bar{p} = 41.39$: mean pizza slice area
- $s_p = 27.23$: sample standard deviation of pizza slice areas
- Adelaide's largest pizza slice is $400cm^2$

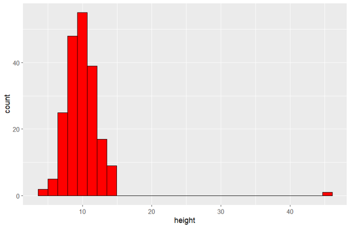
Standardisation

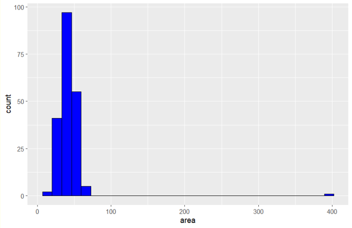
Scatterplot is no use. Why?

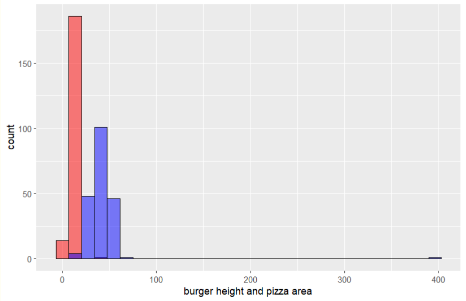


Standardisation

Try histograms







Standardisation

So calculate **z-scores** for every element

$$z_{b,j} = \frac{b_j - \bar{b}}{s_b}, \qquad z_{p,j} = \frac{p_j - \bar{p}}{s_p}$$

Now we have

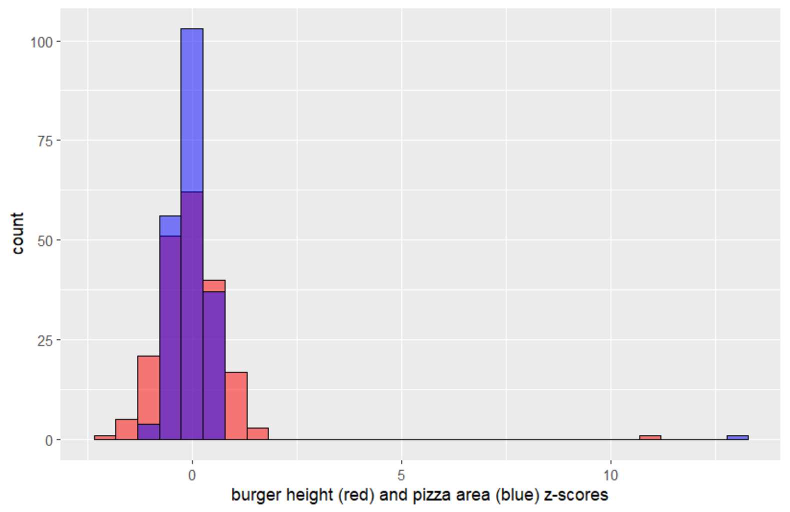
- $(z_{b,1}, \ldots, z_{b,200})$: sample burger heights (in dimensionless units)
- mean of $z_{b,j}$: $(\bar{z}_b) = 0$
- sample standard deviation of $z_{b,j}$: $s_{zb} = 1$
- z-score of Octuple burger: 10.92

and

- $(z_{p,1}, \ldots, z_{p,200})$: sample pizza areas (in dimensionless units)
- mean of $z_{p,j}$: $(\bar{z}_p) = 0$
- sample standard deviation of $z_{p,j}$: $s_{zp} = 1$
- z-score of Adelaide's pizza slice: 13.17

Standardisation

Now compare histograms. Preserves shape. (This is a **linear transformation**.)



Standardisation

Positives

- Shows how the data differs from the mean, both positive and negative direction.
- Standardisation works well for comparing data, especially extreme data (outliers).

Negatives

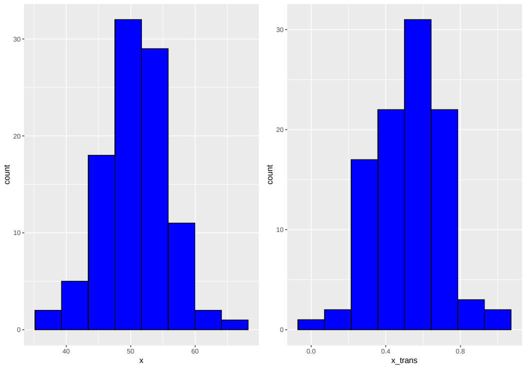
- Outliers can still be arbitrarily far from the centre.
- You may need your data on the exact same range.
- You may need your data to be strictly positive.

Min-max scaling

- Rescaling the range of features to scale the range in $[0, 1]$.

$$x^* = \frac{x - min(x)}{max(x) - min(x)}$$

- Same shape, just rescaled. (This is also a **linear transformation**.)



Min-max scaling

Let's show that this is the only **LINEAR FUNCTION** to rescale like this. (It'll be fun!)

Linear function: $x^* = mx + c$. We must have

$$0 = m\,x_{min} + c \tag{1}$$

$$1 = m\,x_{max} + c. \tag{2}$$

Eq. (1) implies that

$$c = -m\,x_{min} \tag{3}$$

and we substitute this into Eq. (2) to get

$$1 = m\,x_{max} - m\,x_{min} = m\left(x_{max} - x_{min}\right) \Rightarrow m = \frac{1}{x_{max} - x_{min}}.$$

With this expression for m we find Eq. (3) becomes

$$c = -\frac{x_{min}}{x_{max} - x_{min}}$$

Substituting both c and m back into $x^* = mx + c$ we get

$$x^* = \frac{x}{x_{max} - x_{min}} - \frac{x_{min}}{x_{max} - x_{min}} = \frac{x - x_{min}}{x_{max} - x_{min}}.$$

Min-max scaling

Positives

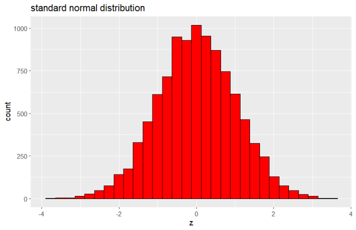
- preserves structure of data
- all rescaled variables have the exact same range
- good for a lot of machine learning algorithms.
  - Eg. house data

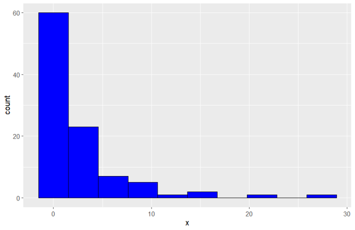
Negatives

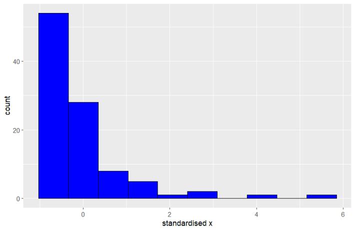
- Extreme values can REALLY effect the scaled data.
- It doesn't include any statistical information (means, standard deviations,...)

Transforming Data for Normality

- Many statistical techniques perform calculations assuming the data is normally distributed.
- But it may be skewed. Eg.







Log transformation.

- A log transformation is a process of applying a logarithm to data to reduce its skew.

$$x^* = log(x) = ln(x) = log_e(x)$$

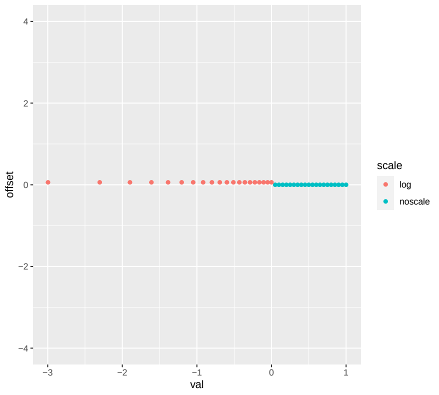
In this class

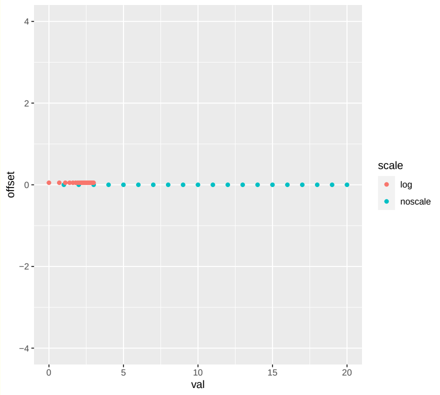
Note that $log(x)$ ALWAYS means the **natural logarithm** $log_e(x)$.

Note : If you have zeros in the data and you can't take the logarithm of zero. In that case you can do $log(x+1)$***.***

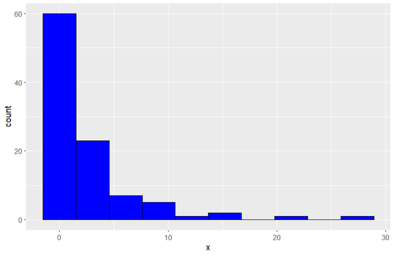
Log transformation.

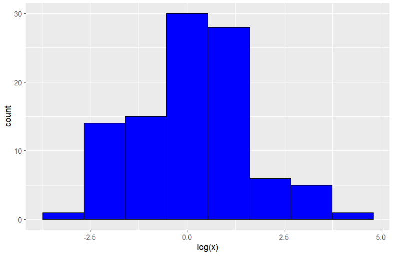
- *Log* function makes
  - smaller values spread out
  - bigger values bunch up





Log transformation.





- How do we measure "goodness" of histograms?
- Using the **skewness** Skew:
  - $\mathrm{Skew} > 0$ means skewed to the right
  - $\mathrm{Skew} < 0$ means skewed to the left
  - $\mathrm{Skew} \approx 0$ means not skewed
- Use `skewness()` in the `moments` package:
  - `skewness(x) = 5.20`
  - `skewness(log(x)) = -0.0682`

Bivariate data

Now we'll look at bivariate data

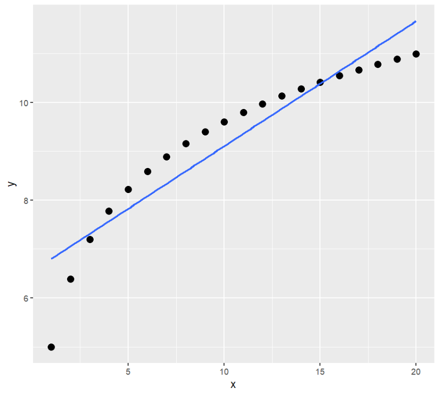
- Although it will involve some univariate analysis as well.

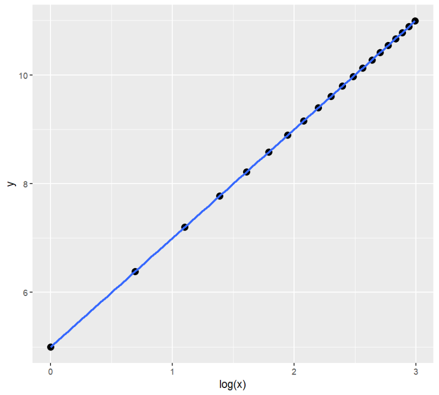
Non-linear relationships

- Here we look at curved relationships between:
  - **2 quantitative variables.**
- 3 of the most common non-linear relationships:
  - Logarithmic functions
  $$y = a\log(x) + b$$
  - Exponential functions
  $$y = \beta e^{ax}$$
  - Polynomial functions
  $$y = \beta x^{a}$$
- We want to make them linear.

Logarithmic functions

$$y = a\log(x) + b$$





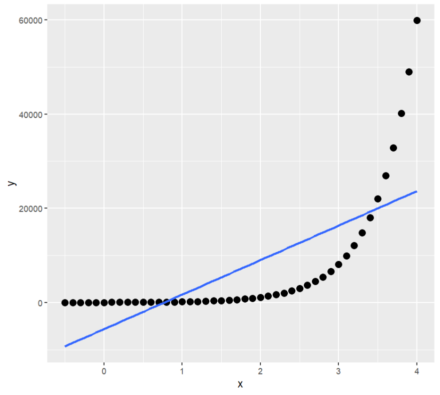
- A "linear–log" plot can get the parameters a and b.
  - a is the gradient (slope)
  - b is the vertical intercept

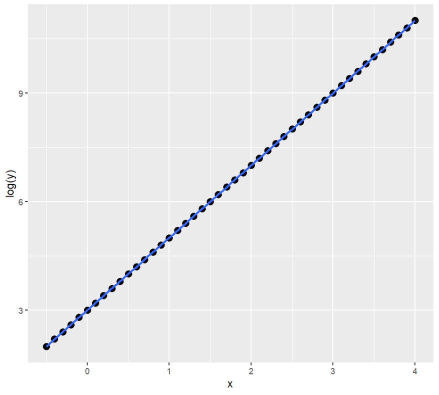
Exponential functions

$$y = \beta e^{ax}$$

Exponential functions

$$y = \beta e^{ax} \quad \Rightarrow \quad \log(y) = ax + b$$





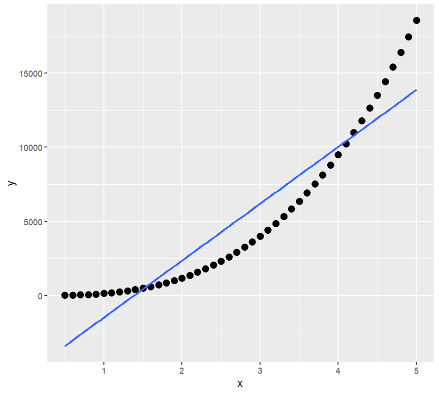
- A "log–linear" plot can get the parameters a and β.
  - a is the gradient (slope)
  - $\beta = e^b$, where b is the vertical intercept

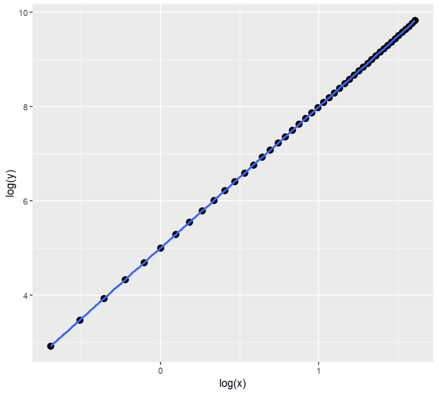
Polynomial functions

$$y = \beta x^a$$

Polynomial functions

$$y = \beta x^a \quad \Rightarrow \quad \log(y) = a \log(x) + b$$





- A "log–log" plot can get the parameters a and β.
  - a is the gradient (slope)
  - $\beta = e^b$, where b is the vertical intercept

Example:

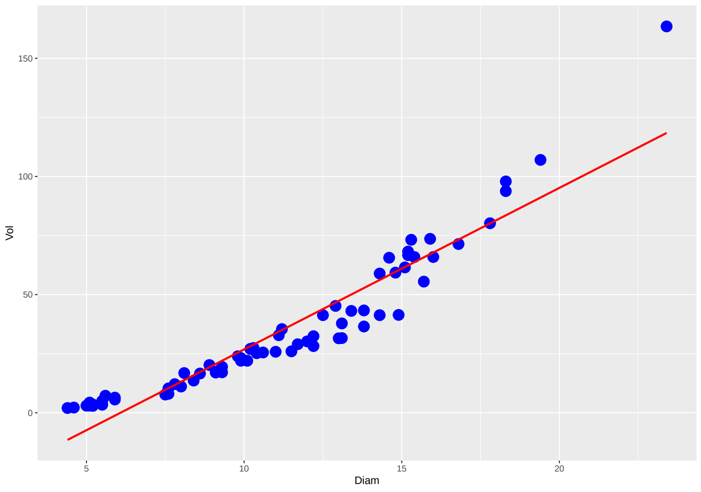
Many different interest groups such as the lumber industry, ecologists, and foresters benefit from being able to predict the volume of a tree just by knowing its diameter. One classic data set (Short Leaf data) concerned the diameter (x, in inches) and volume (y, in cubic feet) of $n = 70$ shortleaf pines.

Question:

- what sort of relationship do you expect?

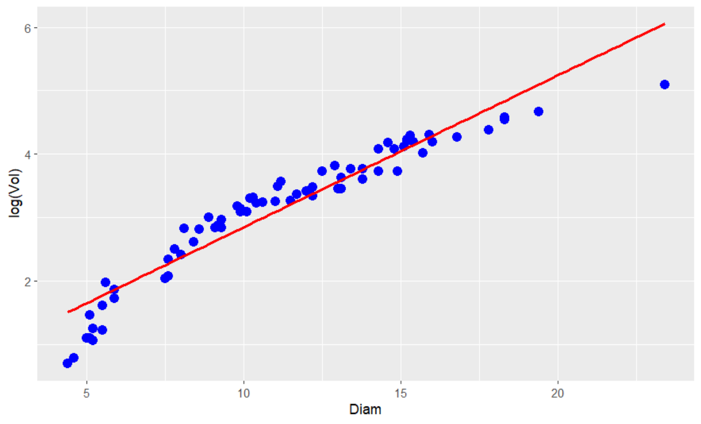
Non-linear relationships

Scatterplot of data



Non-linear relationships

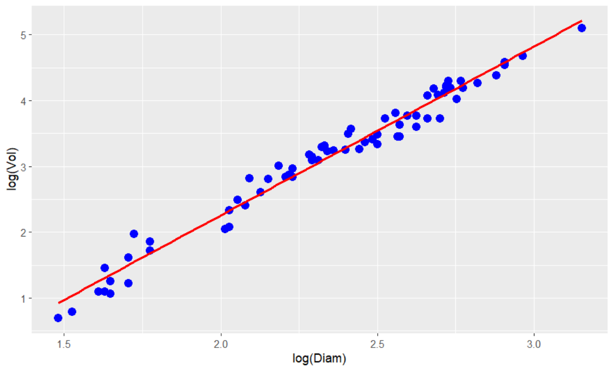
A log–linear plot of `Diam` vs log(`Vol`)



- Not very straight. So probably not **exponential relationship**.
- What about polynomial?

Non-linear relationships

Now a log–log plot:



- Looks good!
- What's the gradient? (Need to extrapolate too far to get intercept.)
- Looks like gradient about 2.25.

Non-linear relationships

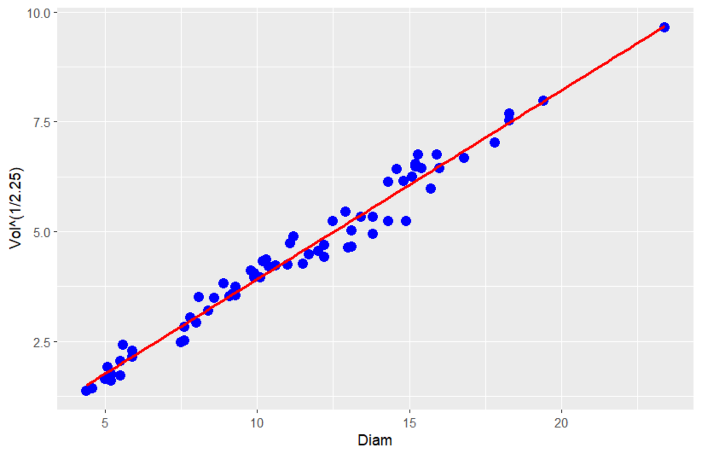
- We used a Log-Log plot:
  - straight line corresponds to **polynomial** $\approx x^a$
  - where gradient is exponent a
- So our data is probably of the form:

$$Vol \approx \beta\, Diam^{2.25} + \gamma$$

- So let's try rescaling `Vol` by the 2.25^{th} root

$$Vol^{1/2.25}$$

Non-linear relationships



- Looks straight enough for a line of best fit.

Non-linear relationships

- So we want to transform our y data by the **inverse** function.
- If $y \approx f(x)$ then we want the transform

$$f^{-1}(y)$$

- Eg. $y \approx x^3$ then we want?

$$y^{1/3}$$

- Eg. $y \approx e^x$ then we want?

$$\log(y)$$

(This is one reason why log transforms are so often useful.)

Box-Cox transformation

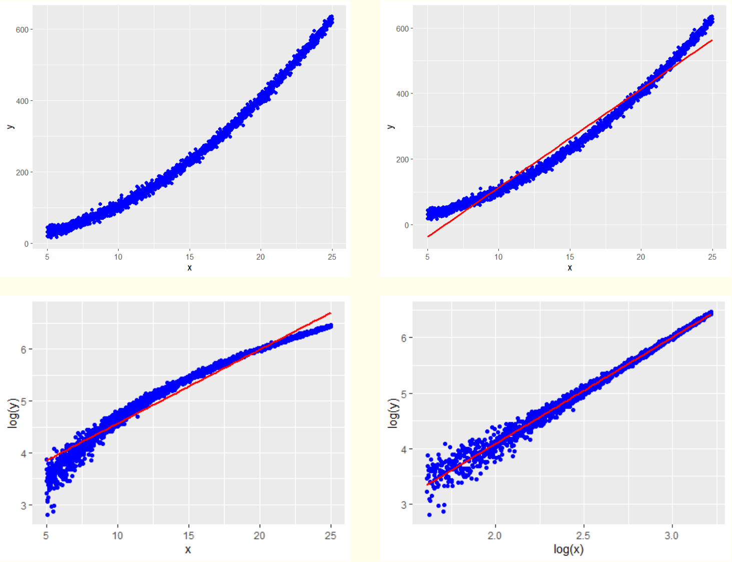
- Is there a quicker way to do this?
- Box-Cox can tell you the “best” transformation for your curved data.
- Automatic transformation using Box–Cox transformation.

$$x^* = \begin{cases} \dfrac{x^\lambda - 1}{\lambda}, & \text{if } \lambda \neq 0 \\ \log(x), & \text{if } \lambda = 0 \end{cases}$$

- So it includes both **logarithmic** and **polynomial** transformations.

Box-Cox transformation

Eg.



Box-Cox transformation

- So we'll try Box-Cox.
- We can use the `caret` package
  - with function `BoxCoxTrans()`.
- Returns best estimate of λ.

Syntax:

```
BoxCoxTrans(x=..., y=..., na.rm=TRUE, [lambda = ...])
```

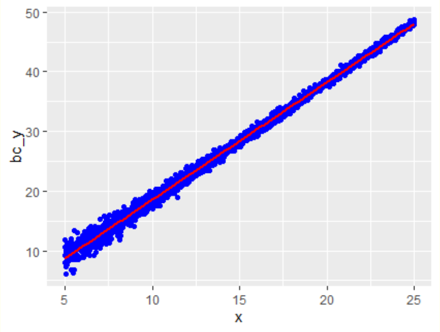
Let's see this in action in `RStudio`.

Box-Cox transformation

- Box-Cox result gave us "best" estimate of $\lambda = 0.5$.
- Then we transformed our data (using `predict()`)

$$y^* = \frac{y^{0.5} - 1}{0.5}$$

- Gives us transformed scatterplot:



Box-Cox transformation

- But what does $\lambda = 0.5$ tell us?

$$y^* = \frac{y^{0.5} - 1}{0.5} \quad \Leftrightarrow \quad \begin{aligned} y &= (0.5y^* + 1)^{1/0.5} \\ &= (0.5y^* + 1)^2 \\ &= (0.5y^*)^2 + y^* + 1 \end{aligned}$$

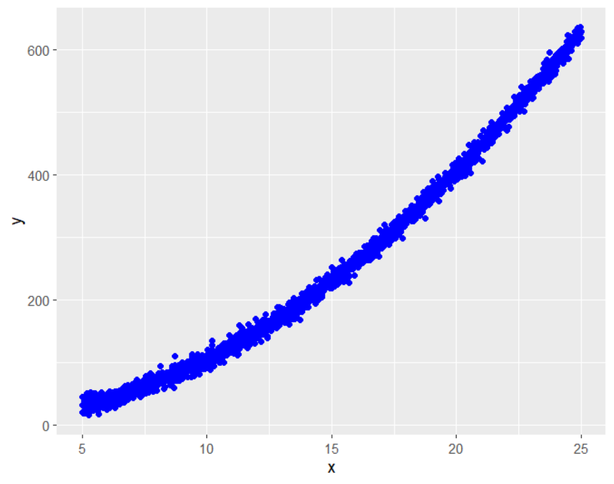
- In general

$$y^* = \frac{y^\lambda - 1}{\lambda} \quad \Leftrightarrow \quad \begin{aligned} y &= (\lambda y^* + 1)^{1/\lambda} \\ &\approx (\lambda y^*)^{1/\lambda} + \text{stuff} \end{aligned}$$

- So our original data must have had exponent (power) $\approx 1/\lambda$
- So in our example, $\lambda = 1/2$ so $1/\lambda = 2$.

Box-Cox transformation

- So let's check. Is it quadratic?



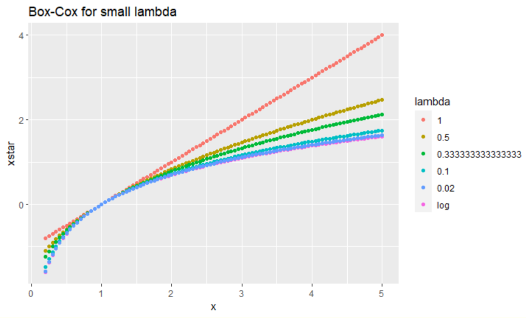
- ? $x \approx 10 \rightarrow y \approx 100$ and $x \approx 20 \rightarrow y \approx 400$.

Box-Cox transformation

Why does Box-Cox change at $\lambda = 0$?

$$x^* = \begin{cases} \dfrac{x^\lambda - 1}{\lambda}, & \text{if } \lambda \neq 0 \\ \log(x), & \text{if } \lambda = 0 \end{cases}$$

- Does it make sense for very small λ? (ie. as $\lambda \to 0$).
- Let's try some numerical evidence!



Box-Cox transformation

- What about real evidence?
- Calculus saves the day (again)! ***

We want to know if

$$\lim_{\lambda\to 0}\frac{x^\lambda - 1}{\lambda} \stackrel{?}{=} \log(x) \quad \text{but} \quad \lim_{\lambda\to 0} x^\lambda = 1 \quad (x \neq 0) \quad \Rightarrow \quad \lim_{\lambda\to 0}\frac{x^\lambda - 1}{\lambda} \approx \frac{0}{0}$$

So we can use:

$$\text{L'Hôpital's rule :} \quad \lim_{z\to 0}\frac{f(z)}{g(z)} = \lim_{z\to 0}\frac{f'(z)}{g'(z)} = \lim_{z\to 0}\frac{\frac{d}{dz}f(z)}{\frac{d}{dz}g(z)}$$

Derivatives of numerator and denominator:

$$\frac{d}{d\lambda}\lambda = 1, \qquad \frac{d}{d\lambda}x^\lambda - 1 = \frac{d}{d\lambda}e^{\log(x^\lambda)} - 1 = \frac{d}{d\lambda}e^{\lambda\log(x)} - 1$$

$$= \log(x)e^{\lambda\log(x)} = \log(x)e^{\log(x^\lambda)} = \log(x)x^\lambda$$

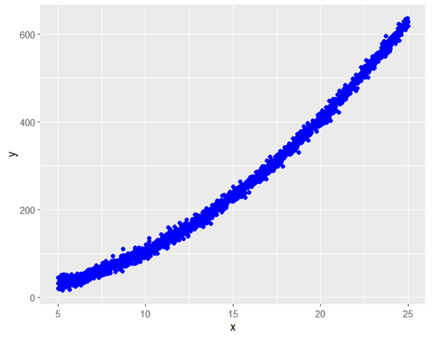
So now

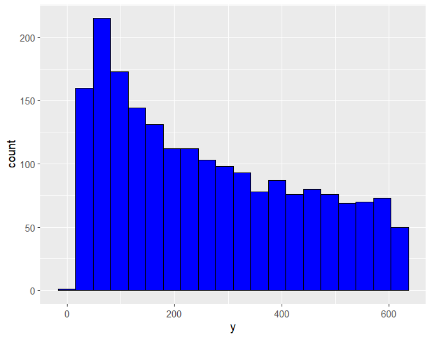
$$\lim_{\lambda\to 0}\frac{x^\lambda - 1}{\lambda} = \lim_{\lambda\to 0}\frac{\frac{d}{d\lambda}x^\lambda - 1}{\frac{d}{d\lambda}\lambda} = \lim_{\lambda\to 0}\frac{\log(x)x^\lambda}{1} = \log(x)\lim_{\lambda\to 0}x^\lambda$$

$$= \log(x) \qquad (\text{for } x \neq 0).$$

Univariate Box-Cox transformation

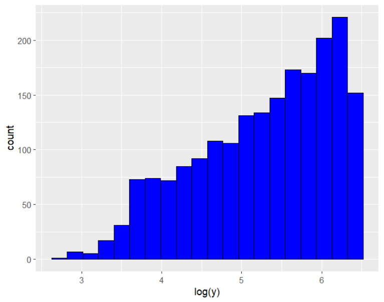
- Box-Cox also works for **univariate** data
  - ie. when you have just a single variable that you're looking at.
- Eg. what does the histogram of our quadratic look like?

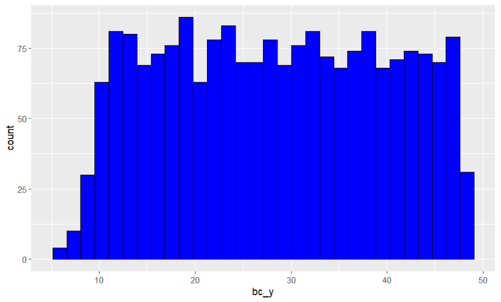




- Histogram: `skewness(y)=0.446`, not very good.

Univariate Box-Cox transformation





- **Log transform** no good:
  - `skewness(log(y)) = -0.543`, even worse!
- **Box-Cox transform** good:
  - `skewness(bc_y) = 0.00341`