Data Transformation
SIT718

Delaram Pahlevani

DEAKIN
UNIVERSITY

The volley ball problem

Student	Sprint 100m	Height(cm)	Serving	Endurance
Mizuho	15.78	148	94	17
Yukie	21.15	147	94	20
Megumi	14.30	134	91	17
Sakura	19.59	174	88	16
Izumi	10.96	145	93	16
Yukiko	19.17	158	83	12
Yumiko	18.35	157	99	20
Kayoko	14.09	177	82	23

Which is better?
Higher or Lower?

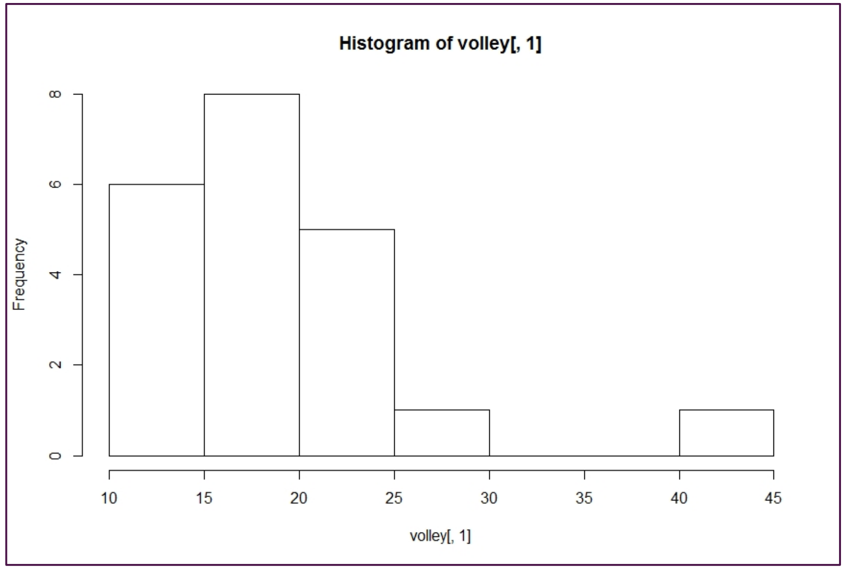
In some cases, it may not be the highest or lowest that becomes ideal but rather a mid-range value. For example, if we are looking at ideal holiday destinations and want to take the climate into account, the best temperature might be described as one that is “not too hot and not too cold”.

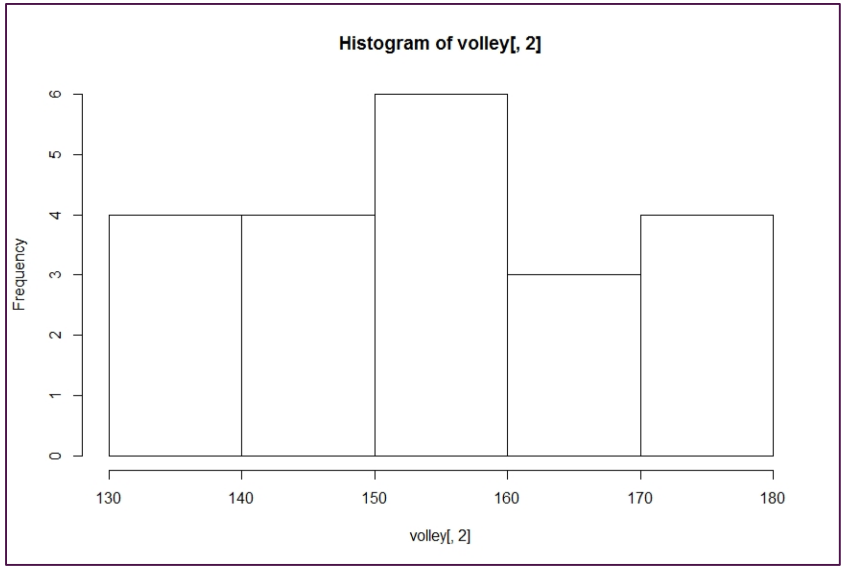
Consistent Scales

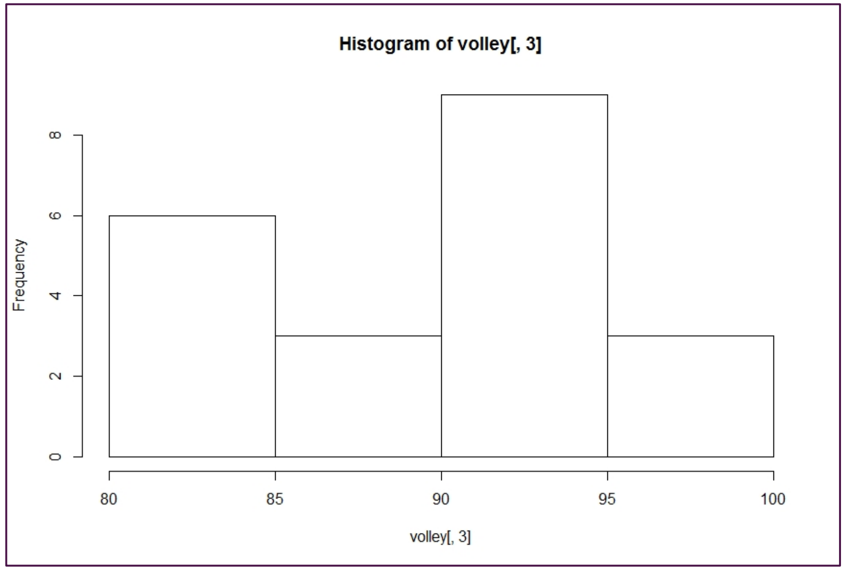
Student	Sprint 100m	Height(cm)	Serving	Endurance	AM
Mizuho	15.78	148	94	17	68.55
Yukie	21.15	147	94	20	70.43
Megumi	14.30	134	91	17	64.36
Sakura	19.59	174	88	16	74.50
Izumi	10.96	145	93	16	66.37
Yukiko	19.17	158	83	12	68.06
Yumiko	18.35	157	99	20	73.44
Kayoko	14.09	177	82	23	73.92

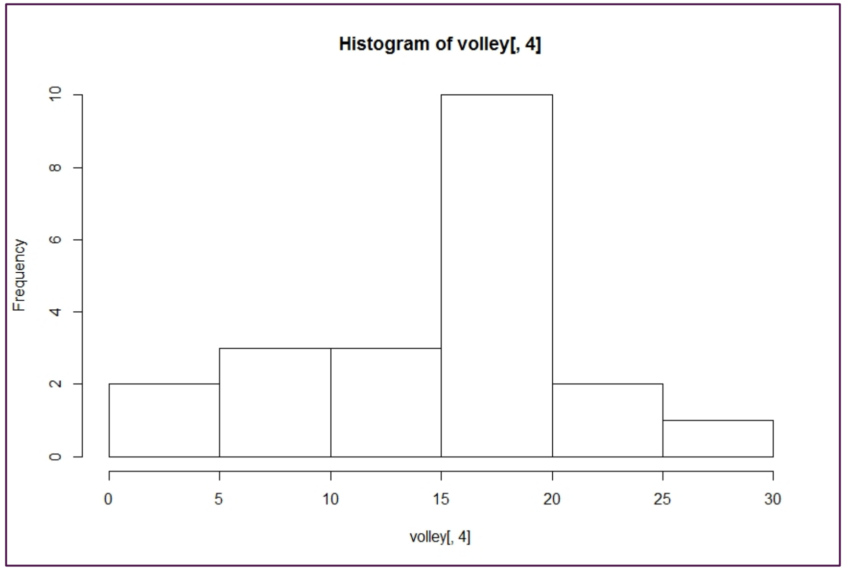
Megumi has the **lowest score** here - but that’s largely due to her height. Not only are the heights higher, but importantly they are much more variable.

Differences in Distribution









Negation and Utility Transformations

Standard negation

Negation functions transform the data so that high values become low and low values become high. If the values are given over the unit interval (between 0 and 1), then the **standard negation** is given by:

$$N(t) = 1 - t$$

Negation and Utility Transformations

Strict Negation:

Informal Definition: A strict negation is a strictly decreasing function of one variable that has a maximum and minimum output that are the same as the domain of the inputs.

Formal Definition: A strict negation *N* defined over a real interval [a,b] is a function that:

is monotone decreasing, i.e. if $x < y \; then \; N(x) > N(y)$; and

Satisfies boundary conditions $N(a) = b \; and \; N(b) = a$

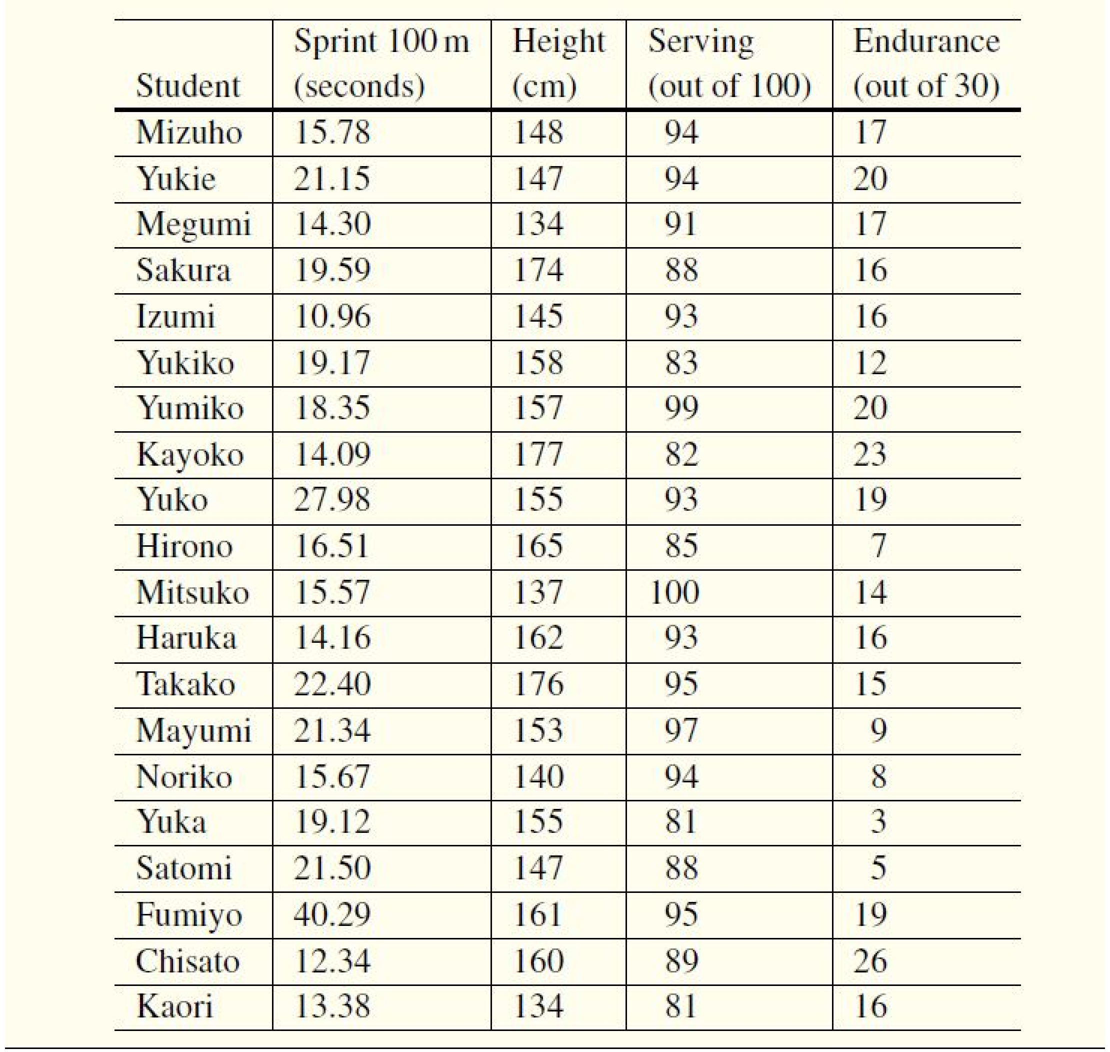
Negation and Utility Transformations

Strong Negation:

In research literature, there is also the concept of a strong negation, which is one that satisfies the property of “involution”. This means that if we perform a negation of the negation then we get the original value.

$$N\left(N(t)\right) = t$$

Data Transformation for Volleyball team



Student	Sprint 100 m (seconds)	Height (cm)	Serving (out of 100)	Endurance (out of 30)
Mizuho	15.78	148	94	17
Yukie	21.15	147	94	20
Megumi	14.30	134	91	17
Sakura	19.59	174	88	16
Izumi	10.96	145	93	16
Yukiko	19.17	158	83	12
Yumiko	18.35	157	99	20
Kayoko	14.09	177	82	23
Yuko	27.98	155	93	19
Hirono	16.51	165	85	7
Mitsuko	15.57	137	100	14
Haruka	14.16	162	93	16
Takako	22.40	176	95	15
Mayumi	21.34	153	97	9
Noriko	15.67	140	94	8
Yuka	19.12	155	81	3
Satomi	21.50	147	88	5
Fumiyo	40.29	161	95	19
Chisato	12.34	160	89	26
Kaori	13.38	134	81	16

$$N(t) = 40.29 - t + 10.96$$

$$N(t) = 51.24 - t$$

```
#negation transformations:
Neg = matrix(1:21, nrow=21, ncol=1)
for (i in Neg){
Neg[i,1] = (51.24-volley[i,1])
}
neg2=sort(Neg, decreasing = TRUE)
plot(neg2)
```

R Exercise:

Generate 100 random values between 10 and 50:

rawData = runif(100, 10, 50)

and plot this data:

```
rawdata = runif(100, 10, 50)
transformed.data = 50 - rawdata +10
plot(rawdata, col = "red")
plot(transformed.data, col= "blue")
rawdata
```

Apply a negation transformation:

transformedData = 50-rawData+10

Plot for the transformed Data

Scaling, Standardization and Normalization

If all our data can take values over a consistent range and have a consistent interpretation, for example, when we are finding the average measurement for a group with respect to one variable then there would usually be no need to change the scale We can take an average and the output can be interpreted in the same units. On the other hand, if the source and type of inputs vary then this is no longer possible. If we do not have a consistent scale, more varied inputs may have an undue influence in the aggregation step. We should also bear in mind that, whether or not the scale is consistent, if the type of inputs differs we should be careful about how we interpret our aggregated value.

Linear Feature Scaling

For a set of values $x_j = \{x_{1j}, x_{2j}, \dots, x_{mj}\}$ relating to a single feature, we let $a = \min(x_j)$ and $b = \max(x_j) - \min(x_j)$:

$$f(t) = \frac{t - a}{b}$$

```
#linear feature scalling
volley = read.table("volley.txt")
scaling = matrix(1:20, nrow=20, ncol=1)
for(i in scaling){
  scaling[i,1] = (volley[i,2]-134)/43
}
scaling
plot(scaling, col = "purple")
```

Standardization

For an input vector $x_j = \{x_{1j}, x_{2j}, \ldots, x_{mj}\}$ where $SDx_j = \sqrt{\sum_{i=1}^{n} \frac{(x_{ij}-\mu)^2}{n-1}}$ is the sample standard deviation of x_j or the true mean and standard deviation may be known *a priori* for the wider population of data observations), standardization involves transforming each x_{ij} using $x'_{ij} = f(x_{ij})$, where

$$f(t) = \frac{t\ - AM(x_j)}{SD(x_j)}$$

Rank Scaling

For an input vector $x_j = \{x_{1j}, x_{2j}, \dots, x_{mj}\}$, let $O_j(x_{ij})$ denote the rank of x_{ij} with respect to the other entries in x_j, so that $O_j(x_{ij}) = 1$ means that x_{ij} is the ‘best’ or highest score, $O_j(x_{ij}) = 2$ means x_{ij} is the second highest, and so on. We can transform each x_{ij} into a score out of 1 using $x'_{ij} = f(x_{ij})$, where

$$\boldsymbol{f(t) = \frac{m - O_j(t)}{m - 1}}$$

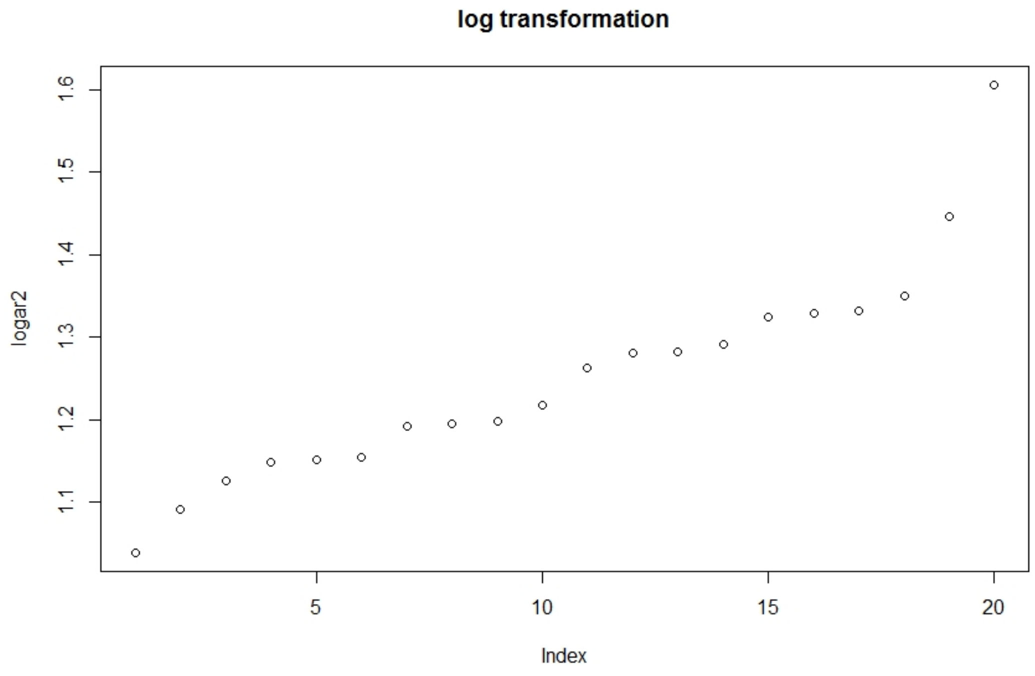
Log and Polynomial Transformations

Other common transformations used as part of some statistical techniques includes the use of increasing functions like

$$f(t) = \ln t \ , \qquad f(t) = \ t^2$$

```
#log and polynominal transformation:
square = matrix (1:20, nrow=20, ncol=1)
for(i in square){
  square[i,1] = (volley[i,1]^2)
}
square

logar = matrix (1:20, nrow=20, ncol=1)
for (i in logar){
  logar[i,1] = (log10(volley[i,1]))
}
logar2= sort(logar)
plot(logar2)
```



Piecewise-Linear Transformations (informal definition)

The piecewise-linear transformations we use will usually be monotone functions where the domain is split into intervals and a different linear function (i.e. $f(t) = mt + c$) is used to transform the data over each interval. The functions should connect on the border of the domains for the transformations to be continuous.

Piecewise-Linear Transformations (formal definition)

For a set of evaluations $x_j = \{x_{1j}, x_{2j}, \ldots, x_{mj}\}$ given over [a,c] we split the domain into two sub-intervals, [a,b), and [b,c]. Let $q \in [0,1]$ be the transformed value we want our variable to take when $x_{ij} = b$. Letting $x'_{ij} = f(x_{ij})$ with the following piecewise function scales the data to the unit interval.

- $f(x) = \begin{cases} q\dfrac{t-a}{b-a}, & a \leq t \leq b \\ q + (1-q)\dfrac{t-b}{c-b} & b \leq t \leq c \end{cases}$

R Exercise

Import the volley.txt data from Week 3 and apply:

Feature scaling to column 1:

V[,1] = V[,1]-min(V[,1]))/(max(V[,1])-min(V[,1])

Normalisation to column 2:

V[,2] =(V[,2]-mean(V[,2]))/sd(V[,2])

R Exercise

Enter in the function and try out a few entries to see that it makes sense and is working correctly.

$$f(x) = \begin{cases} 0.7\dfrac{t}{0.5}\,, & 0 \leq t \leq 0.5 \\ 0.7 + 0.3\dfrac{t-0.5}{0.5} & 0.5 \leq t \leq 1 \end{cases}$$

```
pw.function<-function(t)
{if(t<0.5){0.7*t/0.5}
else{0.7+0.3*(t-0.5)/0.5}
}
```