SIT718 Real World Analytics - Prac. 06
Fitting aggregation functions to empirical data - Solutions

1. The Kei Hotels rating data is a 56×10 table where the first column indicates Kei's ratings for each hotel (out of 100) and columns 2 to 9 are the ratings of similar users.

   (i) Download the `KeiHotels.txt` file and save it to your `R` working directory.

   (ii) Assign the data to a matrix, e.g. using

```
kei.data <- as.matrix(read.table("KeiHotels.txt"))
```

(iii) Define a function to measure the similarity between Kei and the other online users. (The Euclidean distance can be defined using the Minkowski distance with $p = 2$)

(iv) Which of the users is *most similar* to Kei? Investigate using scatterplots, his-tograms, the correlation and similarity between Kei and the other users.

The Minkowski distance (with Manhattan distance as the default) can be defined using

```
minkowski <- function(x,y,p=1) (sum(abs(x-y)^p))^(1/p)
```

Using $p = 2$, i.e.

```
minkowski(kei.data[,1],kei.data[,2],2)
```

gives the Euclidean distance between Kei and user 2, while

```
minkowski(kei.data[,1],kei.data[,3])
```

gives the Manhattan distance between Kei and user 3.

We can also use correlation (which is one of the standard statistical func-tions in `R`).

```
cor(kei.data[,1],kei.data[,3])
```

gives the Pearson correlation, while

```
cor(kei.data[,1],kei.data[,3], method = "spearman")
```

gives Spearman correlation.

To calculate all values at once (rather than one at a time), we can use

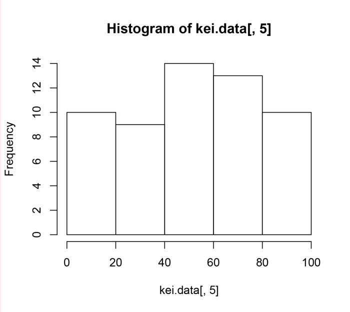
```
d2 <- array(0,9)
for(i in 1:9) {
d2[i] <- minkowski(kei.data[,1],kei.data[,i+1],2) }
```

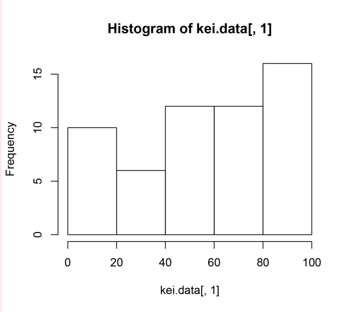
We will obtain the following results.

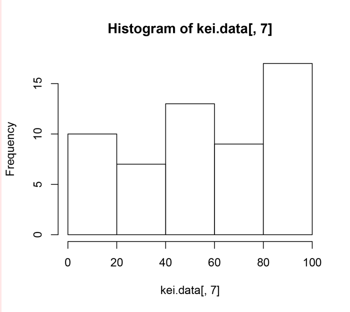
sim	2	3	4	5	6	7	8	9	10
Euclidean	103.407	119.633	116.837	**73.239**	75.478	127.902	119.105	94.557	107.476
Manhattan	541	564	677	**358**	385	691	552	481	551
Pearson	0.879	0.865	0.853	**0.947**	0.941	0.855	0.851	0.921	0.886
Spearman	0.867	0.840	0.835	**0.932**	0.932	0.829	0.820	0.903	0.884

Remember that the lower the distances, the more similar, while the corre-lation should be higher for higher similarity. So the most similar user from these indicators is user 5. However there are other ways that we could con-sider similarity.

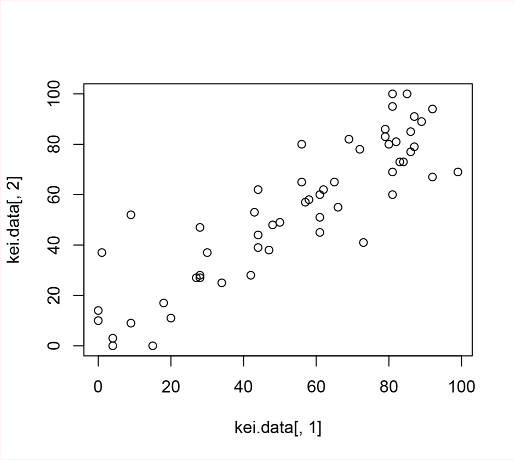
We can compare distributions using histograms. These do not tell us about which hotels were rated, however we can spot whether users tend to rate hotels as consistently high, consistently low, etc. Below are histograms of User 5 (left), Kei (centre) and user 7 (right). Even though User 5 is the most similar user in terms of the other measures, a rough look at these histograms suggests that user 7 may have a more similar pattern in terms of tendency to rate highly or lowly.

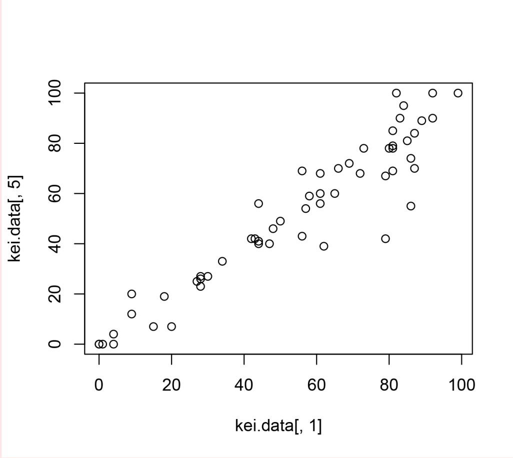






Scatterplots will most likely reflect the values obtained using correlation, however they can also help us pick up whether there could be a non-linear relationship (which would not be reflected in the correlation values). The scatterplots below plot Kei's scores against user 2 (left) and user 5 (right).





2. Download the AggWaFit R file to your working directory and load into the R **workspace using,**

```
source("AggWaFit718.R")
```

(i) Using `fit.QAM`, find the weights for a weighted arithmetic mean that best approximates Kei's ratings from those of the other users.

[hint: You will need to set Kei's data as the last column. You can do this using (e.g. if your data matrix is 'A'), `A <- A[,c(2:9,1)]`

Using

```
fit.QAM(kei.data[,c(2:10,1)])
```

the output stat file gives:

```
RMSE 4.20993003328861
Av. abs error 3.12765137378224



Pearson correlation 0.98951986803197
Spearman correlation 0.983894812270749
i w-i
1 0.0194056285221035
2 0.0390876342948007
3 0.102891478464021
4 0.240427752211895
5 0.281991744682169
6 0.0470443873526975
7 0
8 0.110916927376011
9 0.158234447096302
```

Note that the outputs of this file are 1 off the user identifiers, i.e. what we have been referring to as user 2 corresponds with w_1, user 3 with w_2 etc. Interestingly, fitting in this way the fitted weights allocate more to user 6 (w_5 here) than user 5 (w_4) even though user 5 was more similar to Kei in all of our previous investigations.

(ii) Use `fit.QAM`, `fit.OWA` to find the best weights for

- Weighted power means with $p = 0.5$, and $p = 2$, (the generators required are PM05, invPM05 and QM, invQM).
- A geometric mean (the generators are GMa and invGMa).
- An OWA.

You can also experiment with using only a subset of the variables.

Proceeding in the same manner, e.g. for a power mean we use

```
fit.QAM(kei.data[,c(2:10,1)],g=PM05,g.inv = invPM05)
```

The RMSE and weights are shown for each of the functions below (all val-ues to 3 decimal places)

	PM $(p = 0.5)$	PM $(p = 2)$	GM	OWA
RMSE	4.599	5.375	60.65	3.256
w_1	0.061	0.008	0.000	0.004
w_2	0.000	0.102	0.000	0.000
w_3	0.113	0.034	0.000	0.000
w_4	0.307	0.274	0.000	0.000
w_5	0.337	0.244	1.000	0.945
w_6	0.031	0.054	0.000	0.026
w_7	0.000	0.000	0.000	0.000
w_8	0.121	0.120	0.000	0.000
w_9	0.031	0.163	0.000	0.024

(in some cases the zeros here represent 0, however (especially for the geo-metric mean) some values are just very low.

(iii) Which model fits the data the best?

The OWA has the lowest RMSE and so it seems to be the best fitting func-tion. Remember that the OWA weights are not associated with users but relative values. Here the OWA is almost exactly the same as the median, with 0.945 allocated to the middle weight (whereas the median would have 1 here). The weighted arithmetic mean obtained earlier is better than any of the power means tried here - suggesting that it's better when the output doesn't tend toward high or low values (which is also supported by the fact that the OWA obtained is almost the same as the median).

(iv) Comment on similarities and differences between the users that were found to be the most similar to Kei and whether they had they highest weights allocated in the fitted data models.

Users 5 and 6 (weights w_4 and w_5 respectively) were allocated the most weight by most of the models. Interestingly, user 5 was not always allocated
the most weight and user 7 (w_6) was never allocated the least weight (even though the distance measures make it much higher).

3. Use a subset of any four of the similar users and the `fit.choquet` function to find the fuzzy measure that fits the data best and compare with your previous findings (trying to use more users will result in a very long time to find the values).

Using the best 4 users based on the WAM (i.e. similar user 6,5,10,9 (in that order), the following stat file is obtained. Variable 1, 2, 3 and 4 are user 6, 5, 10 and 9, respectively.

```
RMSE 3.55879402102922
Av. abs error 2.53337971552437

Pearson Correlation 0.992462190684642
Spearman Correlation 0.98957099698045
Orness 0.529772542272563
i Shapley i
```

i	Shapley i	
1	0.273870573870577	The importance of Variable 1 (user 6)
2	0.301551226551394	The importance of Variable 2 (user 5)
3	0.220537795537936	The importance of Variable 3 (user 10)
4	0.204040404040449	The importance of Variable 4 (user 9)

binary number	nam.weights
1	0.179487179487191
2	0.214285714285701
3	0.400000000000108
4	0.117216117216137
5	0.179487179487191
6	0.954545454545463
7	0.954545454545558
8	0.0357142857143169
9	0.892857142856606
10	0.214285714285695
11	1.00000000000009
12	0.142857142857174
13	1
14	1.00000000000027
15	1.00000000000036

#No.	Binary	Variable-set
#1	0001	variable 1
#2	0010	variable 2
#3	0011	variable 1,2
#4	0100	variable 3
#5	0101	variable 1,3
#6	0110	variable 2,3
#7	0111	variable 1,2,3
#8	1000	variable 4
#9	1001	variable 1,4
#10	1010	variable 2,4
#11	1011	variable 1,2,4
#12	1100	variable 3,4
#13	1101	variable 1,3,4
#14	1110	variable 2,3,4
#15	1111	variable 1,2,3,4

Based on the Shapley values, now the most important user is user 5 (then 6 then 10 then 9). Looking at the fuzzy measure, which has an orness suggesting it is fairly neutral (and possibly close to a weighted arith-metic mean), we can spot a few interesting relationships. $v(\{2\}|) = 0.214$ and $v(\{3\}) = 0.117$ (binary numbers 2 = 10 and 4 = 100 respectively), however together their weight is $v(\{2,3\}) = 0.955$ (binary number 6 = 1010). This suggests a complementary relationship. Similarly, 4 by itself is $v(\{4\}) = 0.036$ however with the first variable $v(\{1,4\}) = 0.89$ (binary number 9 = 1001). Meanwhile the variables 4 and three together are fairly additive. The superadditive/positive synergy relationship for some of these pairs simply suggests that if BOTH are high then it is likely that the out-put should be high, whereas in some cases, perhaps where users are more similar to each other, both having good scores does not suggest much about Kei's scores.