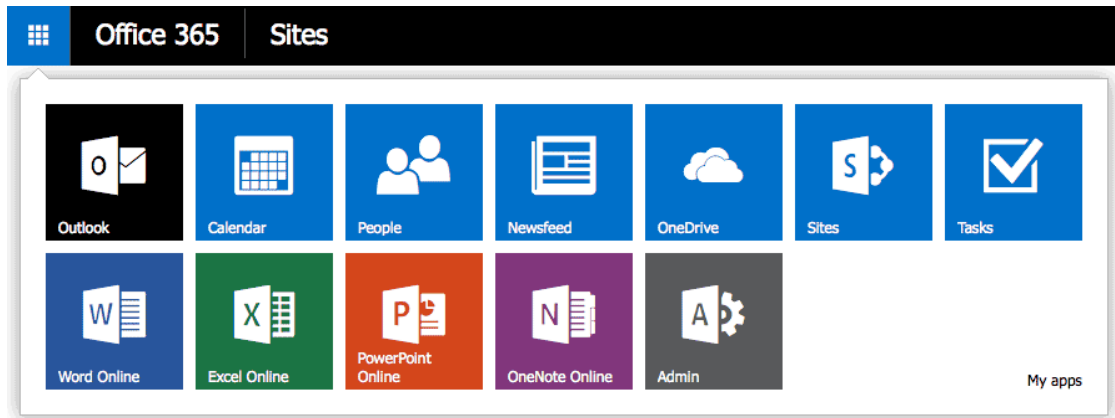


Grading feedback from A4

What the assignment was about



1. Your project's URL
- 2a. Edit code to create a right kind of objective function
- 2b. experiment with different forms of formulas within `groupingCost()`. Why is it `*` and not `/` ?
- 2c. Properties of the objective function
3. What is the effect of k ?

Optimiser's process

1. Generates 30,000 different layout candidates:

```
# Optimization  
winner, winner_score = optimize(30000, random_search, seed_layout, ...
```

2. Applies the same objective function to each of the 30,000 candidates
3. Finds the one ("winner") that has lowest value

2a. Correct answer

” In the lecture, the following formula for a grouping cost function was presented: "sum of (*distance X association*)"

```
# The objective function that considers two objectives: selection time and YOUR function
def combinedObjectiveFunction (layout, numberOfColumns, inputs):
    # "inputs" is an array of different inputs. This time the array's first element is needed
    # in linear_ST() and the others in yourFunction()

    # You can edit this:
    return linear_ST (layout,numberOfColumns,inputs[0:]) + 0.5 * groupingCost (layout,numberOfColumns,inputs[1:])

def groupingCost (layout, numberOfColumns, inputs):
    cost = 0.0

    # With the two loops, go through each pair of elements in the layout list:
    for i in range(0, len(layout)):
        for j in range(i+1, len(layout)):

            # The association weights have been delivered to us within the "inputs" parameter.
            # We pick each pairwise association score from it. If no element is found, then
            # there is no association between the elements, and we assign 0 to it.
            if inputs[0].get(layout[i]+layout[j]):
                association = inputs[0].get(layout[i]+layout[j])
            elif inputs[0].get(layout[j]+layout[i]):
                association = inputs[0].get(layout[j]+layout[i])
            else:
                association = 0

            # You need to edit this:
            cost += distance(numberOfColumns,i,j) * association

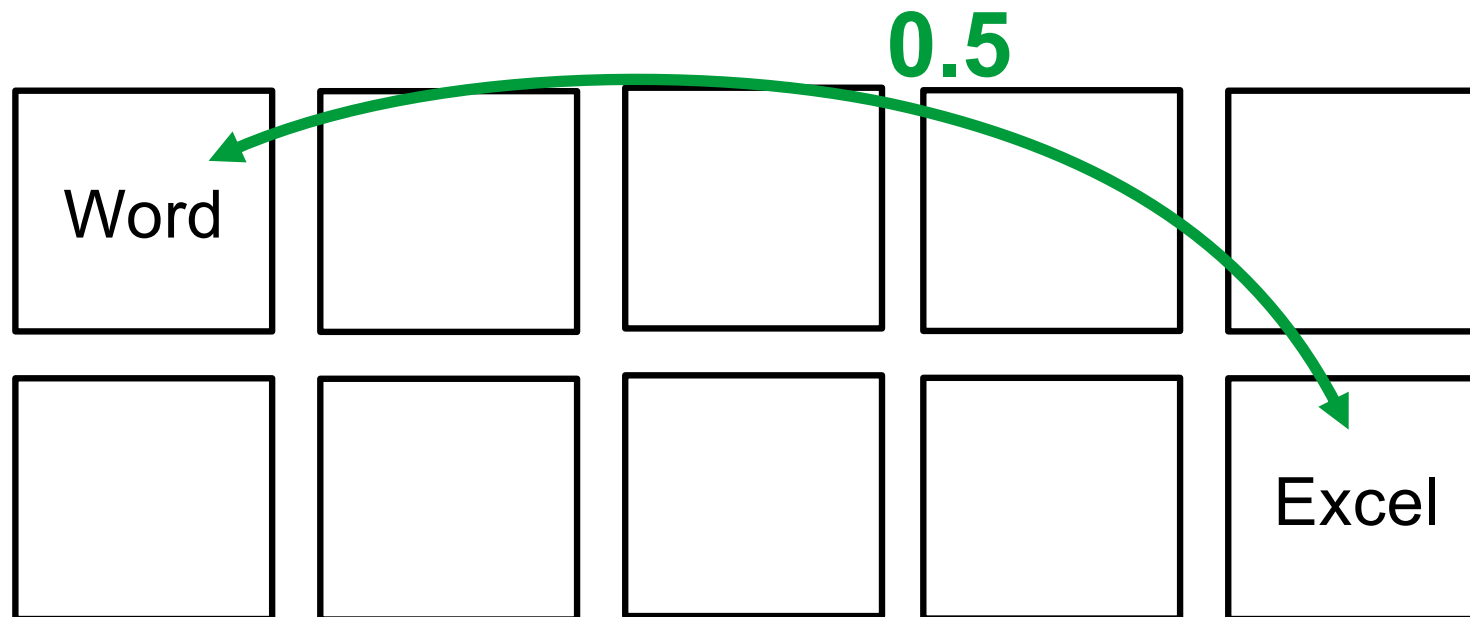
    return cost
```

2b. Why distance * association, not distance / association ?

```
cost += distance(numberOfColumns,i,j) * association
associations = {'WordExcel':0.5, 'WordPPT':0.5, 'MailCal':0.3, 'CalPpl':0
```

If highly associated elements are **far apart**:

- => High distance * High association
- => High grouping cost =>
- => High overall cost
- => Unlikely the most optimal layout

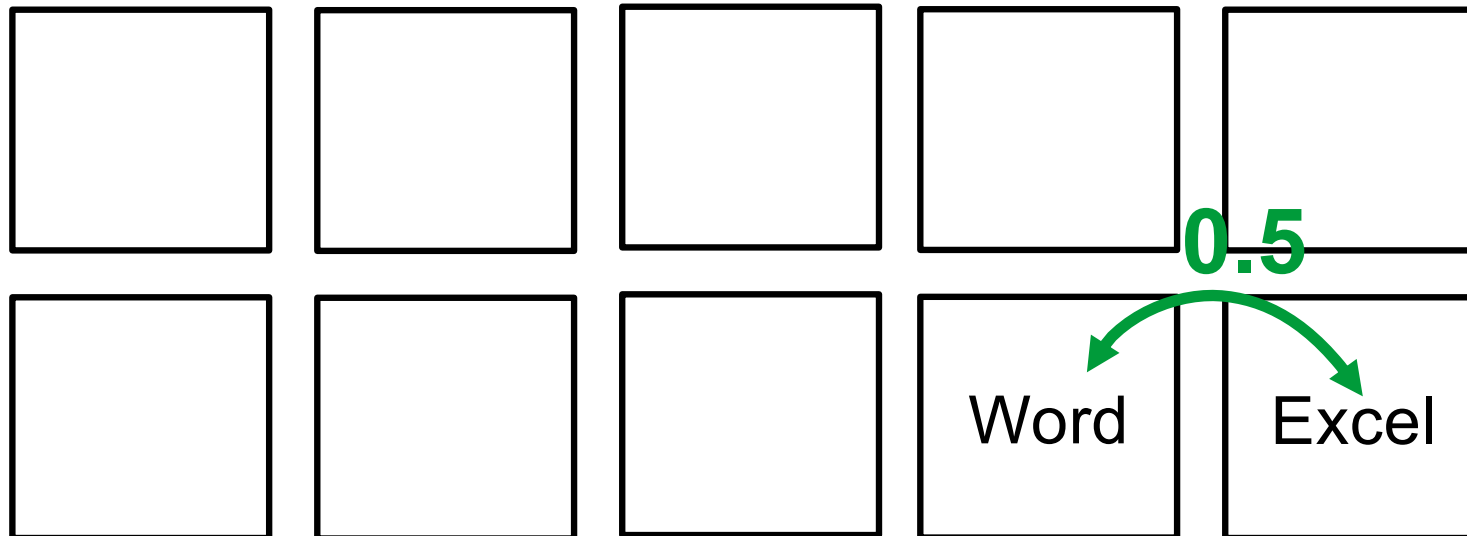


Formula explanation continued...

```
cost += distance(numberOfColumns,i,j) * association
associations = {'WordExcel':0.5, 'WordPPT':0.5, 'MailCal':0.3, 'CalPpl':0
```

If highly associated elements are **close to each other**:

- => Low distance * High association
- => Low grouping cost =>
- => Low overall cost
- => Can be part of optimal layout



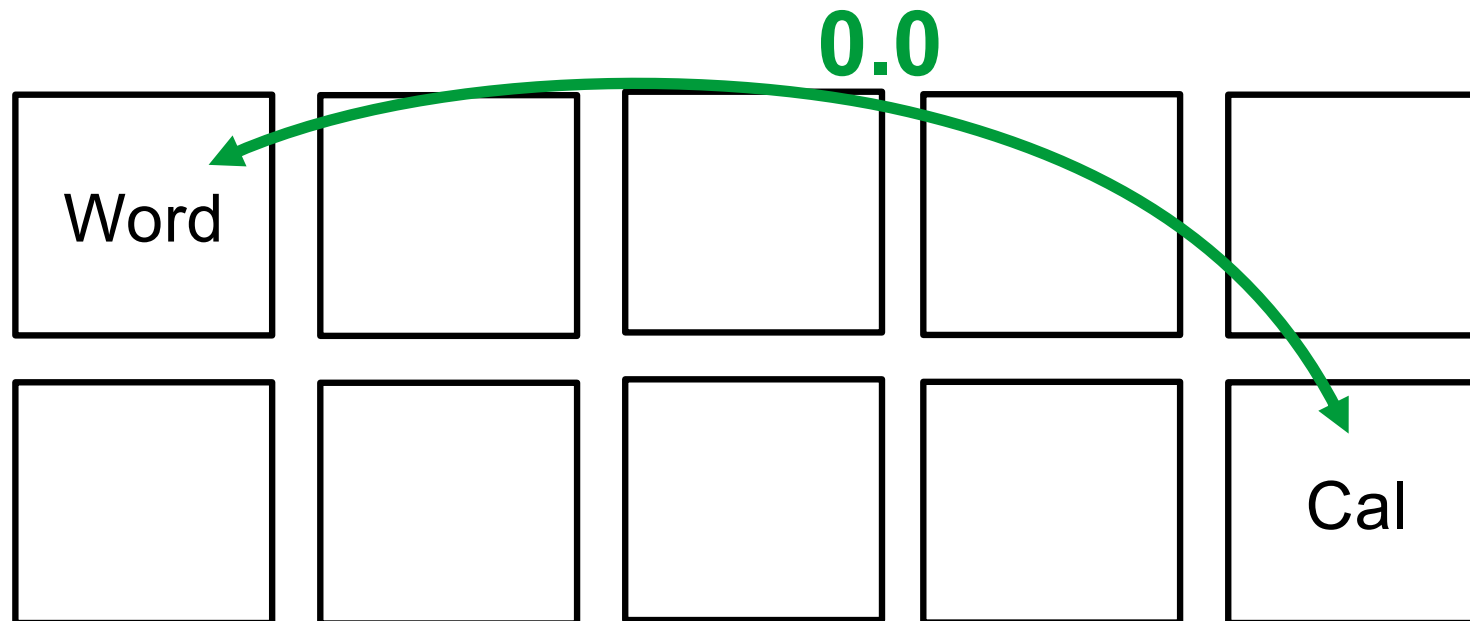
Formula explanation continued...

```
cost += distance(numberOfColumns,i,j) * association
```

```
associations = {'WordExcel':0.5,'WordPPT':0.5,'MailCal':0.3,'CalPpl':0
```

If non-associated elements are far apart:

- => High distance * Zero association
- => Zero grouping cost =>
- => Low overall cost
- => Can be part of optimal layout



2c. Properties of the objective function

Be able to counter-balance the individual elements' weights and their belonging to clusters

Generally:

Be well-behaving (i.e., smooth, no extreme values such as singularities)

Tractable (i.e., whose behaviour is understandable)

Computationally not too heavy

3. What does the k do?

” The objective function whose minimization can yield good UI designs has the following type of a formula: *selection time + k X grouping cost*.

```
# You can edit this:  
return linear_ST (layout,numberOfColumns,inputs[0:]) + 0.5 * groupingCost (layout,numberOfColumns,inputs[1:])
```

_____ selection time _____ k _____ grouping cost _____

Different values of k define how important it is to group associated elements together.

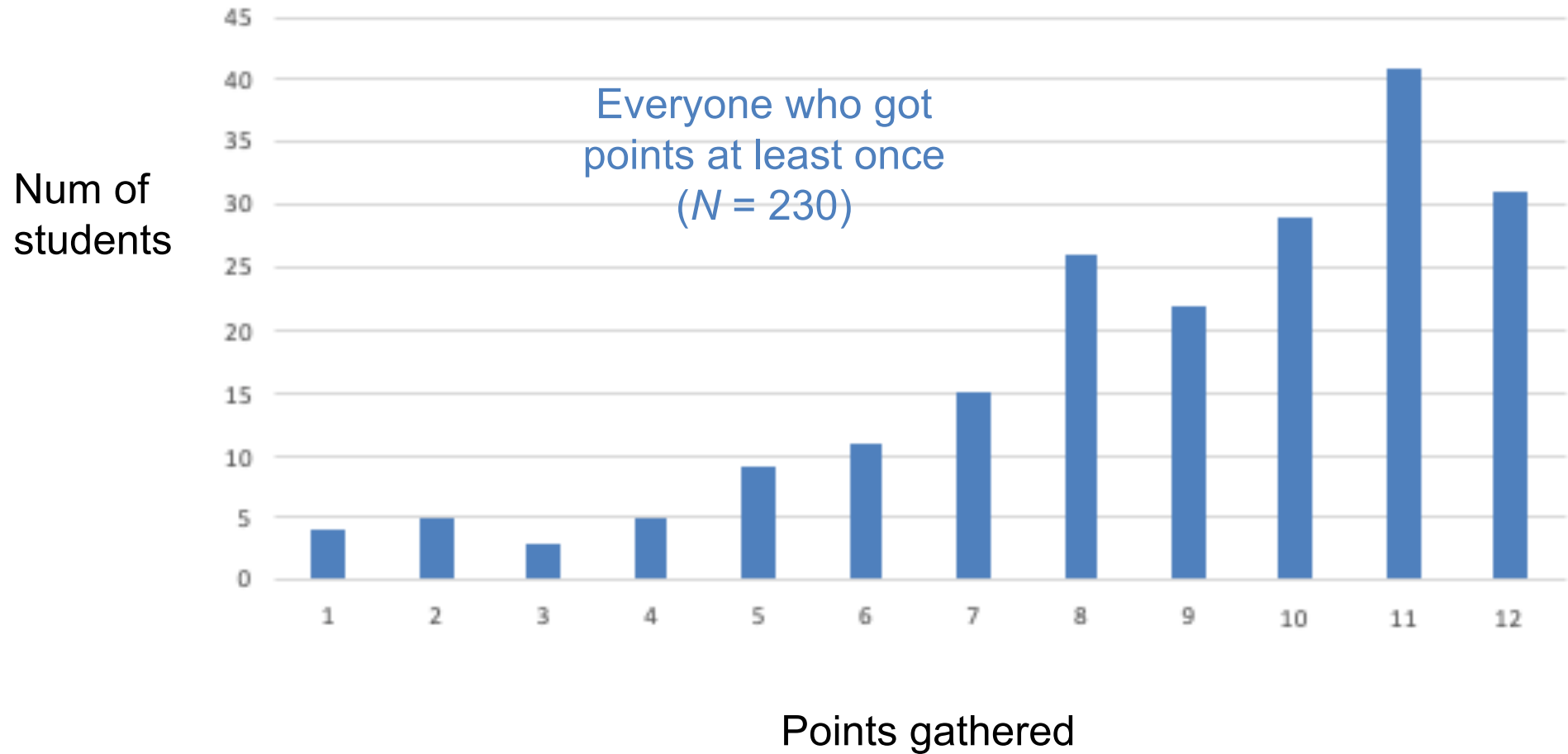
If $k = 0$, it means that only individual elements' weights matter and there is no grouping



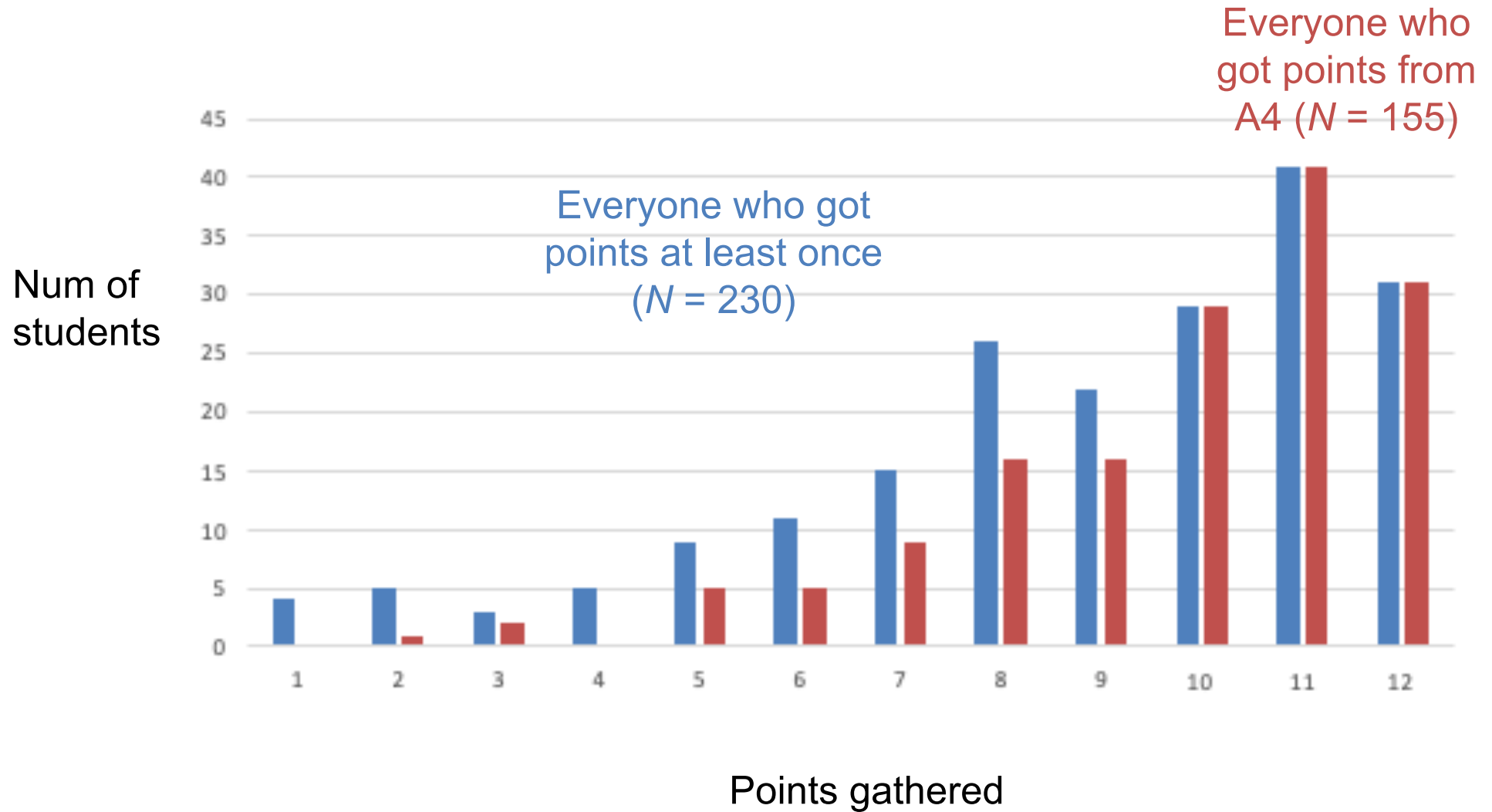
Aalto-yliopisto
Perustieteiden
korkeakoulu

Point distribution summary from A1–A4

Assignment points from A1–A4



Assignment points from A1–A4



Grading principles and examples of answers that were returned

Computational HCI

The objective function (you'll need to edit these)

```
In [23]: # A new objective function that considers two objectives: selection time and YOUR function
def combinedObjectiveFunction (layout, numberOfColumns, inputs):
    # "inputs" is an array of different inputs. This time the array's first element is needed
    # in linear_ST() and the others in yourFunction()

    # You can edit this:
    return linear_ST (layout,numberOfColumns,inputs[0:]) + 0.99 * groupingFunction (layout,numberOfColumns,inputs[1:])

def groupingFunction (layout, numberOfColumns, inputs):
    cost = 0.0
    for i in range(0, len(layout)):
        for j in range(i+1, len(layout)):
            # The association scores have been delivered to us within the "inputs" parameter.
            # We pick each pairwise association score from it. If no element is found, then
            # there is no association between the elements, and we assign 0 to it.
            if inputs[0].get(layout[i]+layout[j]):
                association = inputs[0].get(layout[i]+layout[j])
            elif inputs[0].get(layout[j]+layout[i]):
                association = inputs[0].get(layout[j]+layout[i])
            else:
                association = 0

            # You need to edit this:
            cost += 0

    return cost
```

- 1) Edit the code, link it
- 2) 2b) Division vs. Multiplication 2c) Properties
- 3) 3 purpose of k(weight)

Features of 3-point answer

TASK 1

Submitted a working code file with correct changes to the cost function

TASK 2

2b) Was able to reason that “*distance / association*” gets higher values which produces opposite results than what our cost function is supposed to measure.

2c) Was able to reason about the properties needed for the design function

TASK 3

Was able to deduct that changing the $k(\text{weight})$ changes how much weight is given to the association vs individual weights (=selection time factors)

Observations: Multiplication vs. Division

- Why does it matter? It changes the output to “opposite” direction!
- Many people opted to use just the error message as sole reason why division shouldn't be used. If you stopped here, you didn't get punished if rest of your assignment didn't include mistakes
 - > What's incorrect about this: there's no comparison

Back up your conclusions with evidence and by showing us your process!

K(weight)

- What actually happens when the k-value is changed? Why does it do that and what does that mean?
- Once again, demonstrate the process and provide us with complete answers

Some tips and tricks

If you feel that it's unclear what to do:

- When you begin the assignment, read the instructions carefully from top to bottom and write down what is asked from you
- When your assignment is done, review if you have answered all the questions

It's useful to write a summary at the end of your assignment. This forces you to reflect on your answer and you might notice that something you thought of is actually not very clear.

You can even leave that into the version you return!