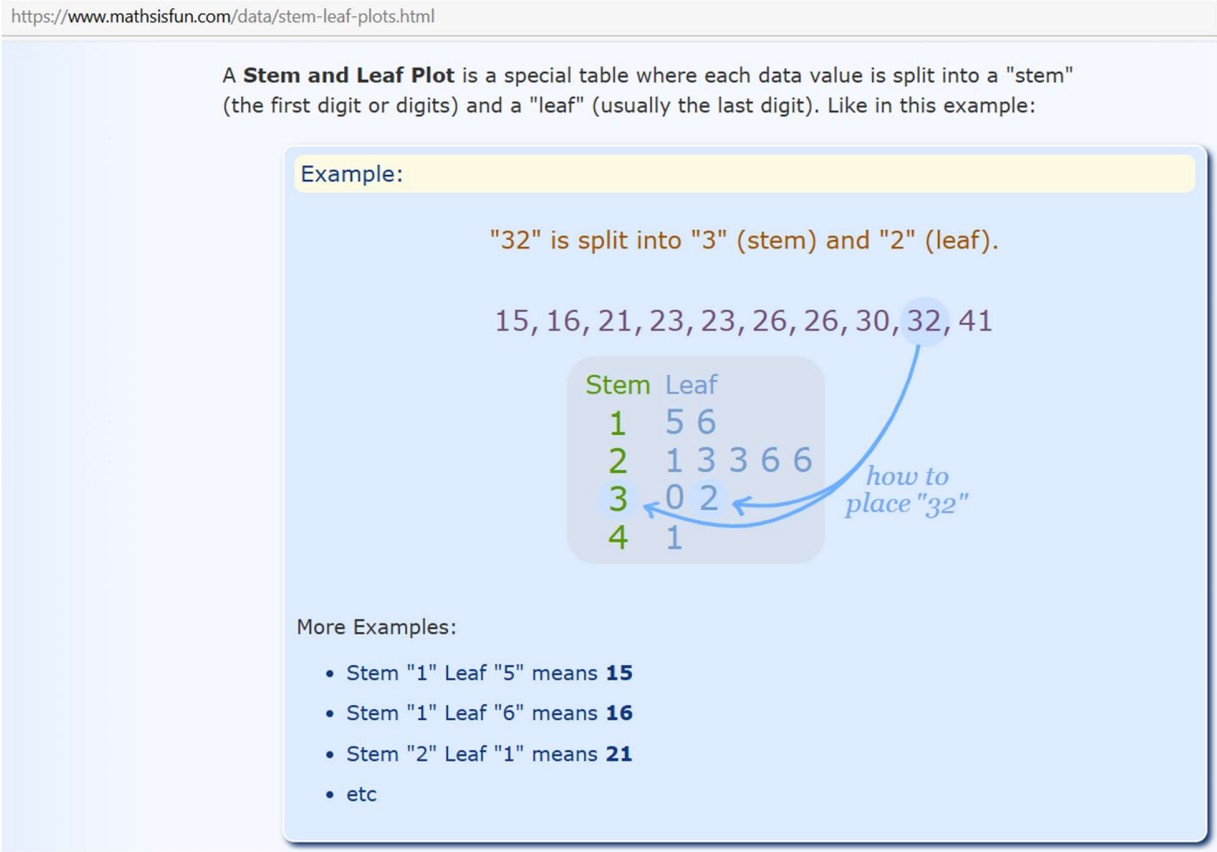
FIN page 1 / ENG page 4

Projektiryhmissä tehtävä piirtotehtävä.

Piirtotehtävä, stem and leaf -kuva

Stem and leaf -kuva on erityinen taulukko, jossa jokainen data-arvo jaetaan ”varteen” (ensimmäinen numero tai numerot) ja ”lehti” (yleensä viimeinen numero). Alla on esimerkki.



Hyöty on siinä, että taulukon rivit sisältävät tarkat lukuarvot, mutta tuottavat myös visuaalisesti pylväsdiagrammin.

Tehtävä. Tee stem and leaf -kuvaaja vapaasti valittavasta aiheesta. (Vähintään 10 arvoa. Vähintään 3 riviä.)

Palautus on pdf-tiedostona, nimeksi tiedostolle RyhmaXX-stem-and-leaf.pdf, missä XX on ryhmän numero.

Esimerkkitoteutus. Ryhma117-stem-and-leaf.pdf

Valitsimme aiheeksi lukua 110 pienemmät alkuluvut. Teimme koodin Google Colabin tekoälyllä. Ensiksi teimme koodin, jolla saa generoitua alkuluvut.

prompt: eratosthenes sieve

```
def eratosthenes_sieve(limit):
    #Generates primes up to limit by Sieve of Eratosthenes.
    prime = [True] * (limit + 1)
    prime[0] = prime[1] = False

    for i in range(2, int(limit**0.5) + 1):
        if prime[i]:
            for multiple in range(i * i, limit + 1, i):
                prime[multiple] = False

    primes = [i for i, is_prime in enumerate(prime) if is_prime]
    return primes
primes=eratosthenes_sieve(40)
print(primes)
```

```
[2, 3, 5, 7, 11, 13, 17, 19, 23, 29, 31, 37]
```

Koodi toimi heti kerralla. Sitten teimme koodin, jolla saa generoitua stem and leaf -kuvaajan

```
def stem_leaf(data):
  stems = []
  leaves = []
  for num in data:
    stem = num // 10
    leaf = num % 10
    if stem not in stems:
      stems.append(stem)
      leaves.append([leaf])
    else:
      leaves[stems.index(stem)].append(leaf)
  for i, stem in enumerate(stems):
    print(f"{stem} | { ' '.join(map(str,sorted(leaves[i]))) }")

primes=eratosthenes_sieve(110)
#print(primes)
stem_leaf(primes)
```

Saimme tulokseksi kuvaajan

```
0 | 2 3 5 7
1 | 1 3 7 9
2 | 3 9
3 | 1 7
4 | 1 3 7
5 | 3 9
6 | 1 7
7 | 1 3 9
8 | 3 9
9 | 7
10 | 1 3 7 9
```

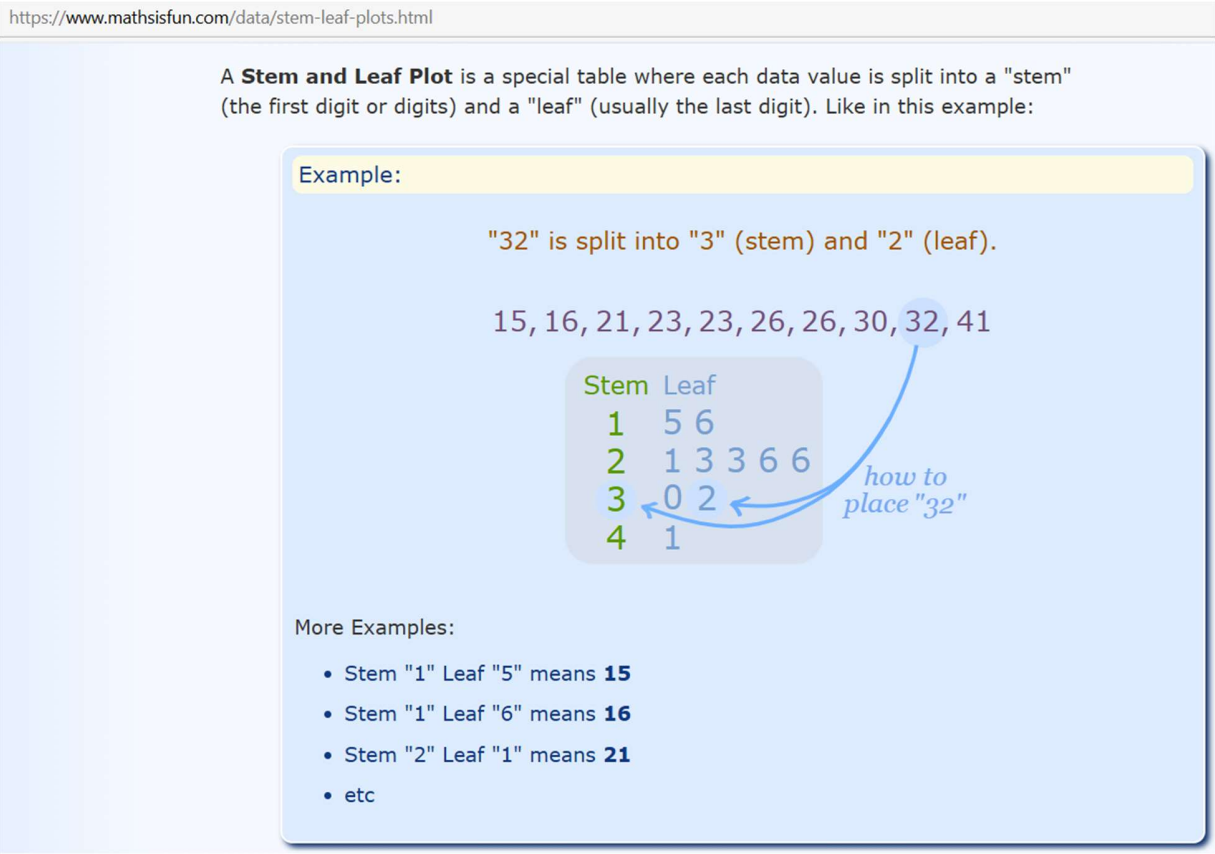
Kuvasta voi näppärästi tarkistaa, vaikkapa että väliltä 50-59 alkulukuja ovat 53 ja 59.

FIN page 1 / ENG page 4

To be made in project teams.

Drawing assignment, stem and leaf plot

A Stem and Leaf Plot is a special table where each data value is split into a ”stem” (the first digit or digits) and a “leaf” (usually the last digit). Below is an example.



The benefit is that the table rows contain the exact values but also produce visually a bar chart.

Task. Make a stem and leaf plot from your topic of choice. (At least 10 values. At least 3 stems.)

Return as a pdf file. Name the file as RyhmaXX-stem-and-leaf.pdf, where XX is the group number.

Example. Ryhma117-stem-and-leaf.pdf

We chose as our topic the primes less than 110. We generated the code with Google Colab AI. First we made a code to produce the prime numbers.

prompt: eratosthenes sieve

```
def eratosthenes_sieve(limit):
    #Generates primes up to limit by Sieve of Eratosthenes.
    prime = [True] * (limit + 1)
    prime[0] = prime[1] = False

    for i in range(2, int(limit**0.5) + 1):
        if prime[i]:
            for multiple in range(i * i, limit + 1, i):
                prime[multiple] = False

    primes = [i for i, is_prime in enumerate(prime) if is_prime]
    return primes
primes=eratosthenes_sieve(40)
print(primes)

[2, 3, 5, 7, 11, 13, 17, 19, 23, 29, 31, 37]
```

The code worked instantly. Then we made a code which generates a stem and leaf plot.

```
def stem_leaf(data):
  stems = []
  leaves = []
  for num in data:
    stem = num // 10
    leaf = num % 10
    if stem not in stems:
      stems.append(stem)
      leaves.append([leaf])
    else:
      leaves[stems.index(stem)].append(leaf)
  for i, stem in enumerate(stems):
    print(f"{stem} | { ' '.join(map(str,sorted(leaves[i]))) }")

primes=eratosthenes_sieve(110)
#print(primes)
stem_leaf(primes)
```

We got the plot

```
0 | 2 3 5 7
1 | 1 3 7 9
2 | 3 9
3 | 1 7
4 | 1 3 7
5 | 3 9
6 | 1 7
7 | 1 3 9
8 | 3 9
9 | 7
10 | 1 3 7 9
```

The plot allows to check, for example, that the primes between 50 and 60 are exactly 53 and 59.