Lecture 2

In [2]:

```
import matplotlib.pyplot as plt
import math
```

In [3]:

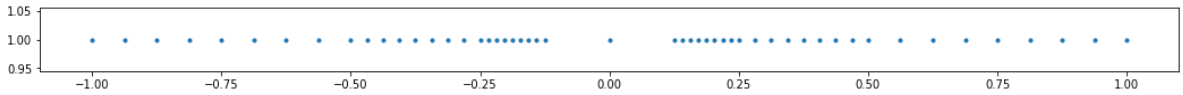
```
from lec02 import get_all_floats, get_denormalized_floats
```

In [4]:

```
def plot_floats(base, precision, l, u, denorm=False):
    all_floats = get_all_floats(BASE=base,
                                PRECISION=precision,
                                L=l,
                                U=u)
    if denorm:
        all_floats += get_denormalized_floats(
                            BASE=base,
                            PRECISION=precision,
                            L=l,
                            U=u)
    plt.figure(None, figsize=(18, 1))
    lt1 = [float(x) for x in all_floats]
    lt1 = [x for x in lt1 if abs(x) <= 1]
    plt.plot(lt1, len(lt1) * [1], ".")
```

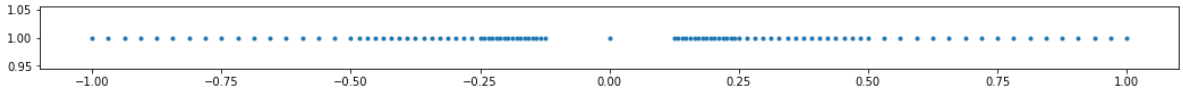
In [5]:

```
plot_floats(base=2, precision=4, l=-3, u=3)
```



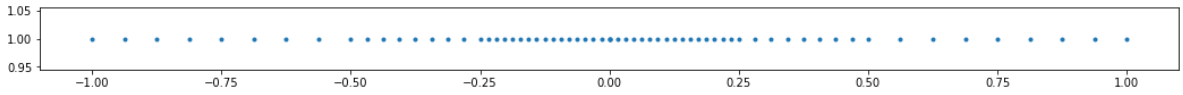
In [7]:

```
plot_floats(base=2, precision=5, l=-3, u=3)
```



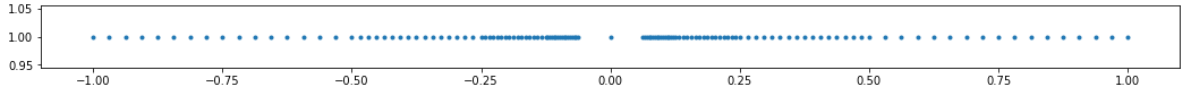
In [6]:

```
plot_floats(base=2, precision=4, l=-3, u=3, denorm=True)
```



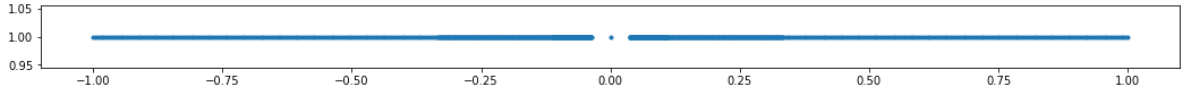
In [8]:

```
plot_floats(base=2, precision=5, l=-4, u=4)
```



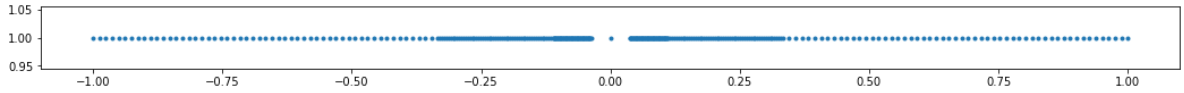
In [9]:

```
plot_floats(base=3, precision=5, l=-3, u=3)
```



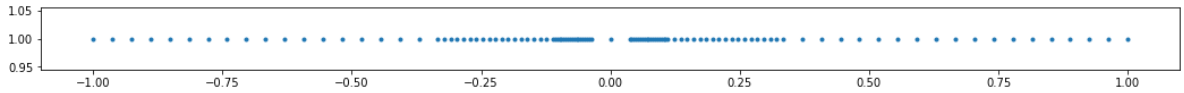
In [10]:

```
plot_floats(base=3, precision=4, l=-3, u=3)
```



In [11]:

```
plot_floats(base=3, precision=3, l=-3, u=3)
```



In [12]:

```
# compute e^x by summing the first n terms of its taylor series expansion
def e(x, n):
    return sum(pow(x,i)/math.factorial(i) for i in range(n))
```

In [14]:

```
# let's comput e^(-40)

for n in range(300):
    print(n, e(-40, n))
```

```
0 0
1 1.0
2 -39.0
3 761.0
4 -9905.666666666666
5 96761.0
6 -756572.3333333334
7 4932316.555555556
8 -27575619.95238095
9 134964062.58730158
10 -587434526.4779541
11 2302159829.7830687
12 -8205456011.166106
13 26819930125.33114
14 -80950488756.19885
15 226964993762.45825
16 -594142959620.6274
17 1458626923837.0867
18 -3371419860769.299
```

In [13]:

```
math.exp(-40)
```

Out[13]:

```
4.248354255291589e-18
```

In [ ]:

```

```