

benford_konstante

August 21, 2021

```
[1]: import numpy as np
import matplotlib.pyplot as plt

from konstante import get_constants

from benford_helper_functions import benford, get_first_digit,
    ↪check_benford_scaling, normalize

[2]: import sys

sys.path.append('../..')
from plotting.matplotlib_setup import configure_latex, savefig,
    ↪set_size_decorator, savefig

tex_dir, images_dir = 'porocilo/main.tex', 'porocilo/images'

configure_latex(style=['science', 'notebook'], global_save_path=images_dir)

%config InlineBackend.figure_format = 'pdf'

[3]: consts = get_constants()

[4]: first_digits = get_first_digit(consts)

[5]: bins, n = np.unique(first_digits, return_counts=True)
n = n / np.sum(n)

ben = np.vectorize(benford)(np.arange(1, 10, 1))
ben = ben / np.sum(ben)

fig, ax = set_size_decorator(plt.subplots, fraction=0.5, ratio='4:3')(1, 1)

ax.bar(bins, n, align='center', alpha=1, color='C0')

ax.plot(bins, n, '-o', c='C1', lw=1, ms=3, label='podatki')
ax.plot(bins, ben, '-o', c='C2', lw=1, ms=3, label='Benford')
```

```

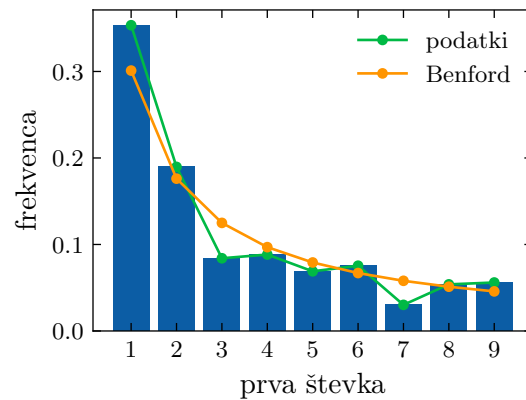
ax.set_xticks(bins)
ax.set_xticklabels(bins)
ax.minorticks_off()

ax.set_xlabel('prva številka')
ax.set_ylabel('frekvenca')

ax.legend()

# savefig('benford_konstante_prva_stevka', tight_layout=False)
plt.show()

```



```

[6]: fig, ax = set_size_decorator(plt.subplots, fraction=0.5, ratio='4:3')(1, 1)

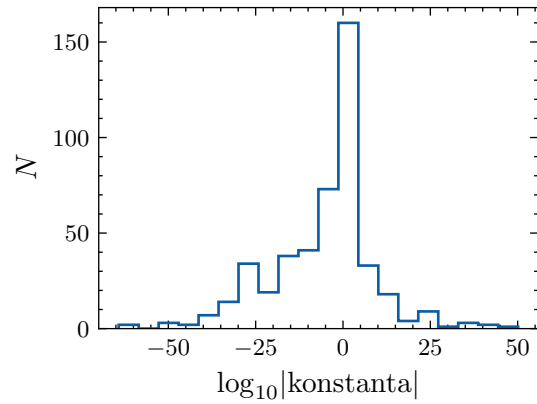
x = np.log10(np.abs(consts))

ax.hist(x, bins=20, histtype='step', lw=1)

ax.set_xlabel(r'\log_{10}$$|konstanta$|')
ax.set_ylabel('$N$')

# savefig('benford_konstante_log', tight_layout=False)
plt.show()

```



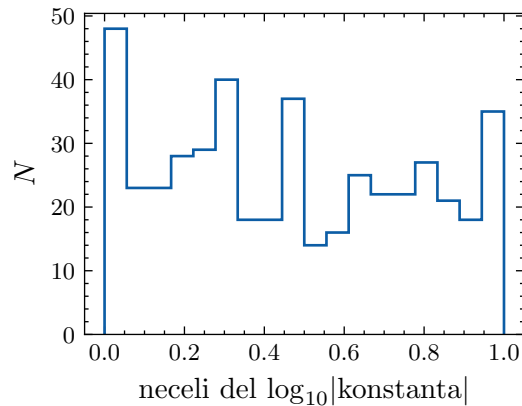
```
[7]: fig, ax = set_size_decorator(plt.subplots, fraction=0.5, ratio='4:3')(1, 1)

fracs = np.log10(np.abs(consts)) % 1

ax.hist(frac, bins=18, histtype='step', lw=1)

ax.set_xlabel(r'neceli del $\log_{10}|\text{konstanta}|$')
ax.set_ylabel('$N$')

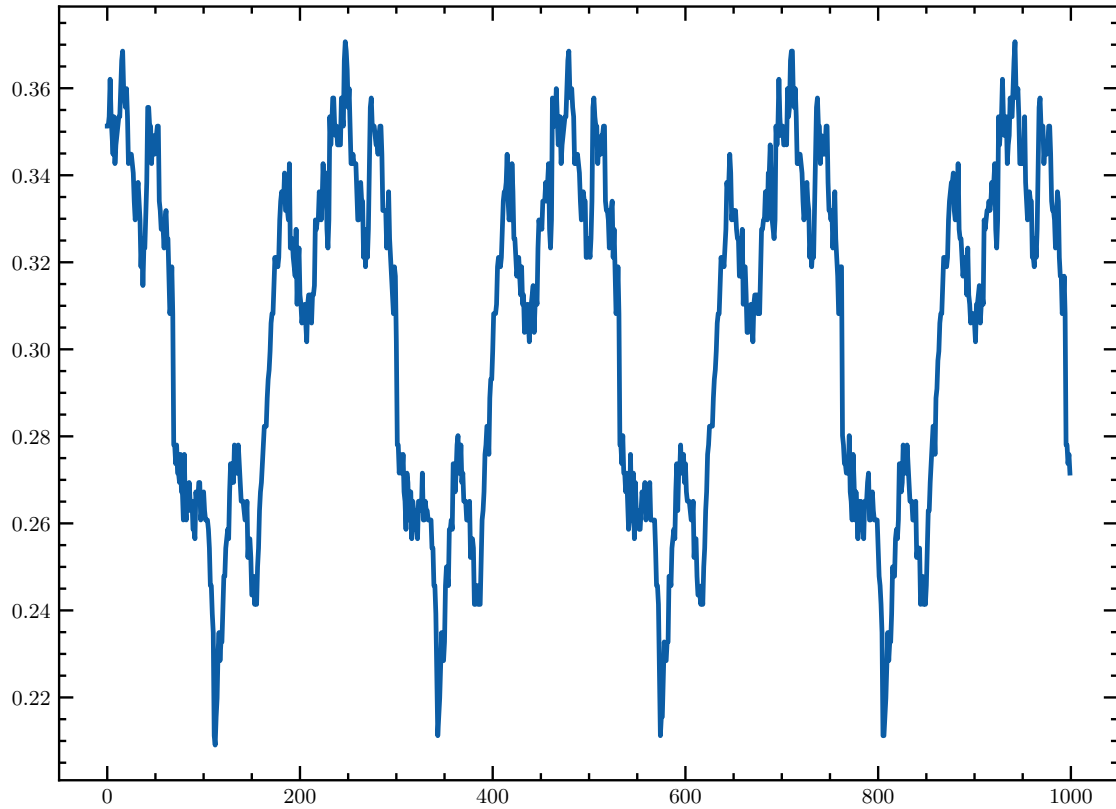
# savefig('benford_konstante_neceli', tight_layout=False)
plt.show()
```



```
[8]: ones = check_benford_scaling(consts, f=1.01)

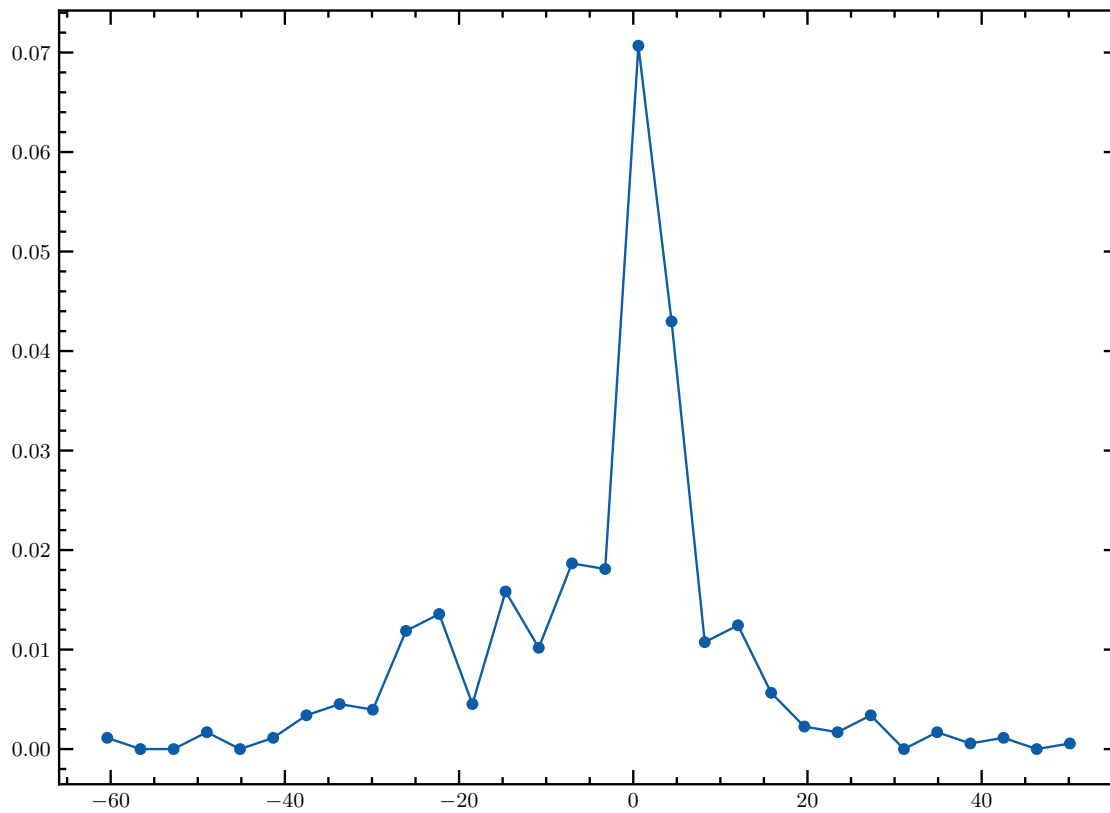
plt.plot(range(len(ones)), ones / len(consts))
```

```
[8]: [<matplotlib.lines.Line2D at 0x7f3b5524a6a0>]
```



```
[9]: x = np.log10(np.abs(consts))  
pdf, bins = np.histogram(x, bins=30, density=True)  
plt.close()  
bins = bins[1:]  
  
plt.plot(bins, pdf, '-o', ms=4, lw=1)
```

```
[9]: [<matplotlib.lines.Line2D at 0x7f3abe200c40>]
```

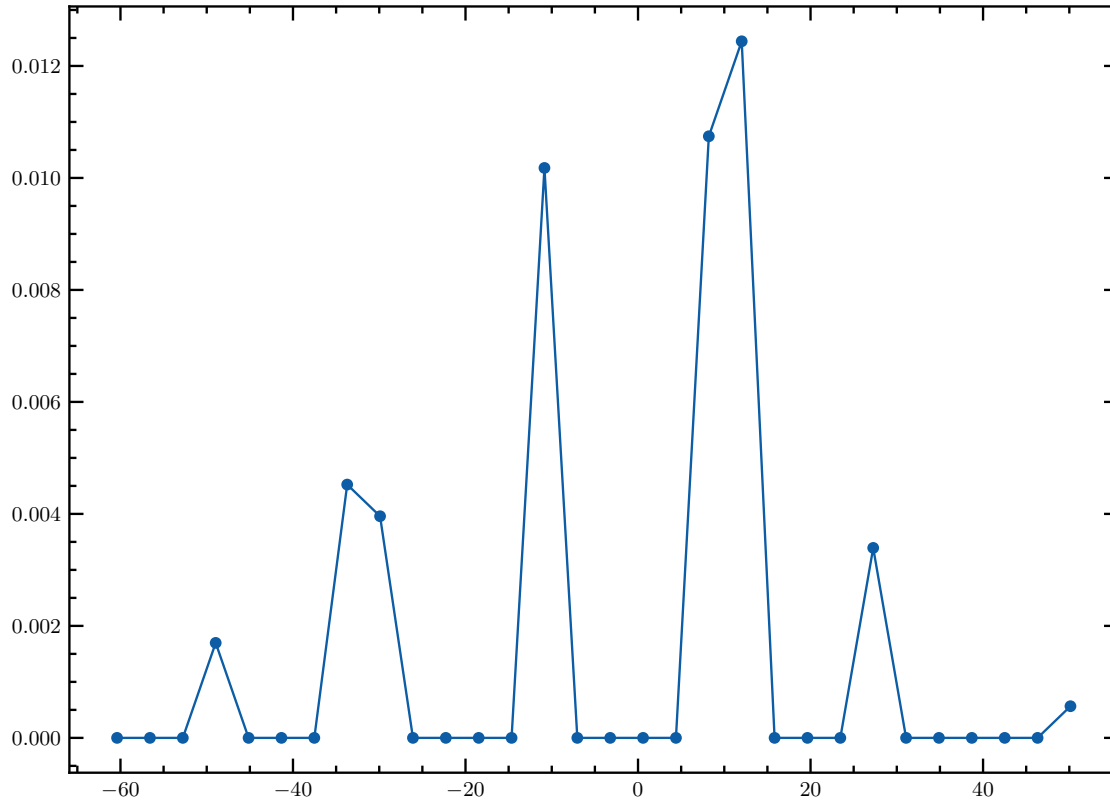


```
[10]: from benford_helper_functions import make_sampling_function
```

```
[11]: sf = make_sampling_function(bins, -80, 80)
```

```
[12]: plt.plot(bins, pdf * sf, '-o', ms=4, lw=1)
```

```
[12]: [<matplotlib.lines.Line2D at 0x7f3afda2ba30>]
```



```
[13]: from benford_helper_functions import shift_multiply_integrate
```

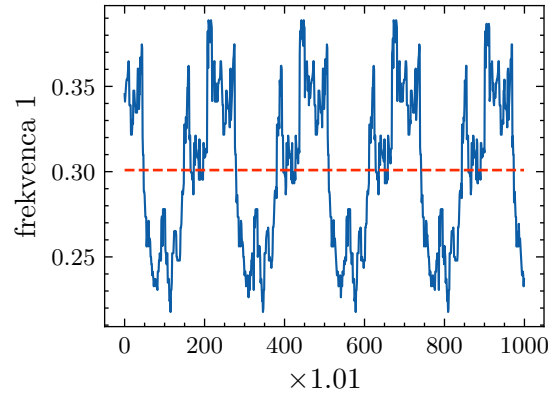
```
[14]: log_input = np.log10(np.abs(consts))

ost = shift_multiply_integrate(log_input, f=1.01, n_shift=1000)
```

```
[15]: fig, ax = set_size_decorator(plt.subplots, fraction=0.5, ratio='4:3')(1, 1)

m = np.mean(ost)
ax.plot(range(len(ost)), ost, lw=0.8)
ax.plot(range(len(ost)), [m]*len(ost), c='C3', lw=1, ls='--')
ax.set_xlabel(r'$\times 1.01$')
ax.set_ylabel(r'frekvencia 1')
# savefig('benford_konstante_freq_1')
m
```

```
[15]: 0.3009103807471266
```



```
[16]: from scipy.stats import lognorm

def lognormMu(x, mu, s):
    """https://stackoverflow.com/questions/66912677/
    ↳scipy-stats-lognormal-distribution-obtain-pdf-with-given-lognormal-distribution"""
    tempX = x / np.exp(mu)
    return lognorm.pdf(tempX, s)

x = np.logspace(0, 6, 10000)[1:]

s = [0.5, 0.8, 1.2]
lognorm_pdf1 = lognormMu(x, mu=7.1, s=s[0])
lognorm_pdf2 = lognormMu(x, mu=7.1, s=s[1])
lognorm_pdf3 = lognormMu(x, mu=7.1, s=s[2])
```

```
[17]: fig, ax = set_size_decorator(plt.subplots, fraction=0.5, ratio='4:3')(1, 1)

lognorm_pdf1 = normalize(lognorm_pdf1, np.log10(x))
lognorm_pdf2 = normalize(lognorm_pdf2, np.log10(x))
lognorm_pdf3 = normalize(lognorm_pdf3, np.log10(x))

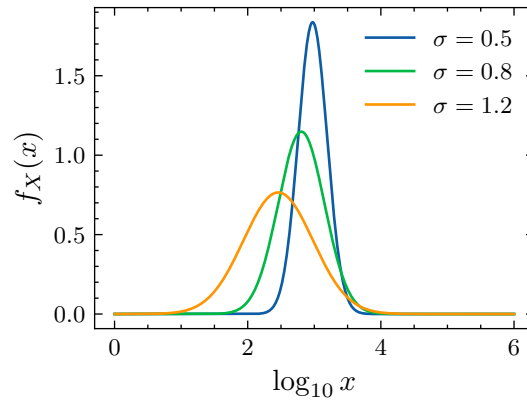
ax.plot(np.log10(x), lognorm_pdf1, lw=1, label=f'\sigma={s[0]}$')
ax.plot(np.log10(x), lognorm_pdf2, lw=1, label=f'\sigma={s[1]}$')
ax.plot(np.log10(x), lognorm_pdf3, lw=1, label=f'\sigma={s[2]}$')

ax.set_xlabel(r'\log_{10}x$')
ax.set_ylabel(r'$f_X(x)$')

ax.legend()

# savefig('lognormal_dist')
```

```
[17]: <matplotlib.legend.Legend at 0x7f3afb959b50>
```



```
[18]: from benford_helper_functions import shift_multiply_integrate_pdf
```

```
[19]: ost = shift_multiply_integrate_pdf(np.log10(x), lognorm_pdf1, f=1.01,
    ↪ n_shift=1000)

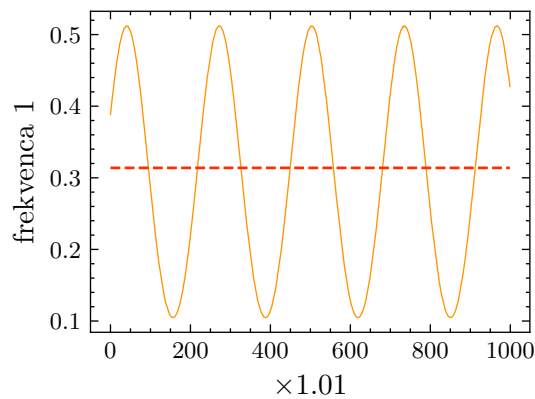
fig, ax = set_size_decorator(plt.subplots, fraction=0.5, ratio='4:3')(1, 1)

m = np.mean(ost)
ax.plot(range(len(ost)), ost, lw=0.5, c='C2')
ax.plot(range(len(ost)), [m]*len(ost), c='C3', ls='--', lw=1)

ax.set_xlabel(r'$\times 1.01$')
ax.set_ylabel(r'frekvenca 1')
# savefig('benford_lognorm_freq_1_mu12')

m
```

```
[19]: 0.3138828472680644
```



[]: