

## Náhodný výběr z rovnoměrného rozdělení na intervalu [0, 60]

```
In [163...]: import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
from scipy import stats

In [164...]: #np.random.seed(1) # pro reprodukovatelnost; pokud chcete skutečně náhodný, změ
n = 100 # velikost výběru
low, high = 0.0, 60.0
sample = np.random.uniform(low, high, size=n)
sample

Out[164...]: array([ 7.16742302, 48.75663528, 8.80754241, 15.85784896, 49.14535072,
       18.63523511, 58.94504696, 15.99832213, 32.0192007 , 18.86802067,
       54.64636987, 21.99339863, 26.01553964, 30.73756154, 56.33318864,
       1.85694038, 43.01271982, 53.46113726, 1.63723341, 31.32307484,
       19.55938871, 51.56935925, 33.51099306, 41.41367209, 27.17121 ,
       37.69854225, 17.4058111 , 0.56091468, 34.6053561 , 18.68665285,
       31.03605591, 54.98435118, 25.58848731, 14.8437622 , 22.27762568,
       55.911667 , 56.21210288, 50.65979691, 55.21239086, 13.6740174 ,
       5.24893257, 13.63858418, 18.86259697, 10.48595258, 36.42564972,
       24.8151849 , 48.98109071, 11.1078239 , 42.11259179, 14.42133749,
       34.45314533, 20.93925583, 3.41786397, 13.72882026, 39.84615332,
       29.8350056 , 31.14095904, 10.48320901, 34.24295109, 59.80520551,
       49.01010653, 35.66235743, 58.5593441 , 54.09375508, 35.73647602,
       1.94557952, 5.61462575, 3.9223029 , 27.10398892, 22.52608999,
       58.52100206, 10.07899759, 58.36725552, 46.04849213, 49.45427038,
       37.95694901, 40.12396621, 28.61294001, 0.78818142, 21.18036513,
       29.52430809, 43.80547267, 28.11770056, 27.44429497, 8.25976446,
       0.65332403, 45.49669563, 19.19717028, 59.06300701, 13.21405357,
       20.32248153, 31.43376814, 45.29348742, 27.83146703, 7.48935259,
       18.75008271, 30.27115018, 40.43094048, 46.20899738, 7.82014686])
```

## Standardizace vzorku (odečtení průměru a dělení směrodatnou odchylkou)

```
In [165...]: mean_sample = sample.mean()
sd_sample = sample.std(ddof=1)

In [166...]: mean_sample

Out[166...]: np.float64(29.577269724095014)

In [167...]: sd_sample

Out[167...]: np.float64(17.410037907207307)

In [168...]: z_sample = (sample - mean_sample) / sd_sample
z_sample
```

```
Out[168...]: array([-1.28717966,  1.10162687, -1.19297427, -0.78801786,  1.12395396,
       -0.62848999,  1.68683017, -0.77994934,  0.14025995, -0.61511923,
       1.4399222 , -0.43560336, -0.20457911,  0.06664499,  1.53680992,
      -1.59220385,  0.77170711,  1.37184466, -1.6048234 ,  0.10027578,
      -0.57540834,  1.26318447,  0.22594571,  0.6798608 , -0.13819957,
       0.4664707 , -0.69910581, -1.66664514,  0.28880387, -0.62553666,
       0.08378995,  1.45933522, -0.2291082 , -0.8462651 , -0.4192779 ,
       1.5125985 ,  1.52985498,  1.21094091,  1.47243339, -0.91345306,
      -1.39737416, -0.91548827, -0.61543075, -1.09656953,  0.39335813,
      -0.27352524,  1.11451917, -1.06085041,  0.72000544, -0.87052839,
       0.28006117, -0.49615135, -1.50254732, -0.91030528,  0.58982546,
       0.01480387,  0.08981539, -1.09672712,  0.26798801,  1.73623607,
       1.11618578,  0.34951605,  1.66467612,  1.40818104,  0.35377328,
      -1.58711258, -1.37636943, -1.47357329, -0.14206062, -0.40500657,
       1.66247383, -1.11994427,  1.65364291,  0.94607619,  1.14169772,
       0.4813131 ,  0.60578251, -0.05538929, -1.65359136, -0.48230249,
      -0.00304202,  0.81724135, -0.08383492, -0.12251408, -1.22443761,
      -1.66133732,  0.91438204, -0.59621349,  1.69360558, -0.93987252,
      -0.53157772,  0.1066338 ,  0.90271014, -0.10027564, -1.26868863,
      -0.62189336,  0.0398552 ,  0.62341454,  0.95529532, -1.24968843])
```

## Teoretické kvantily standardního a normálního rozdělení pro $p = (i-0.5)/n$

```
In [169...]: prob = (np.arange(1, n + 1) - 0.5) / n # percentily, posun o 0,5 kvuli vypoctu

# ppf = percent point function – tedy inverzní distribuční funkce (kvantilová funkce)
# Ta nám řekne: "K jaké hodnotě normálního rozdělení odpovídá tato pravděpodobnost"
theoretical_q = stats.norm.ppf(prob) # normální

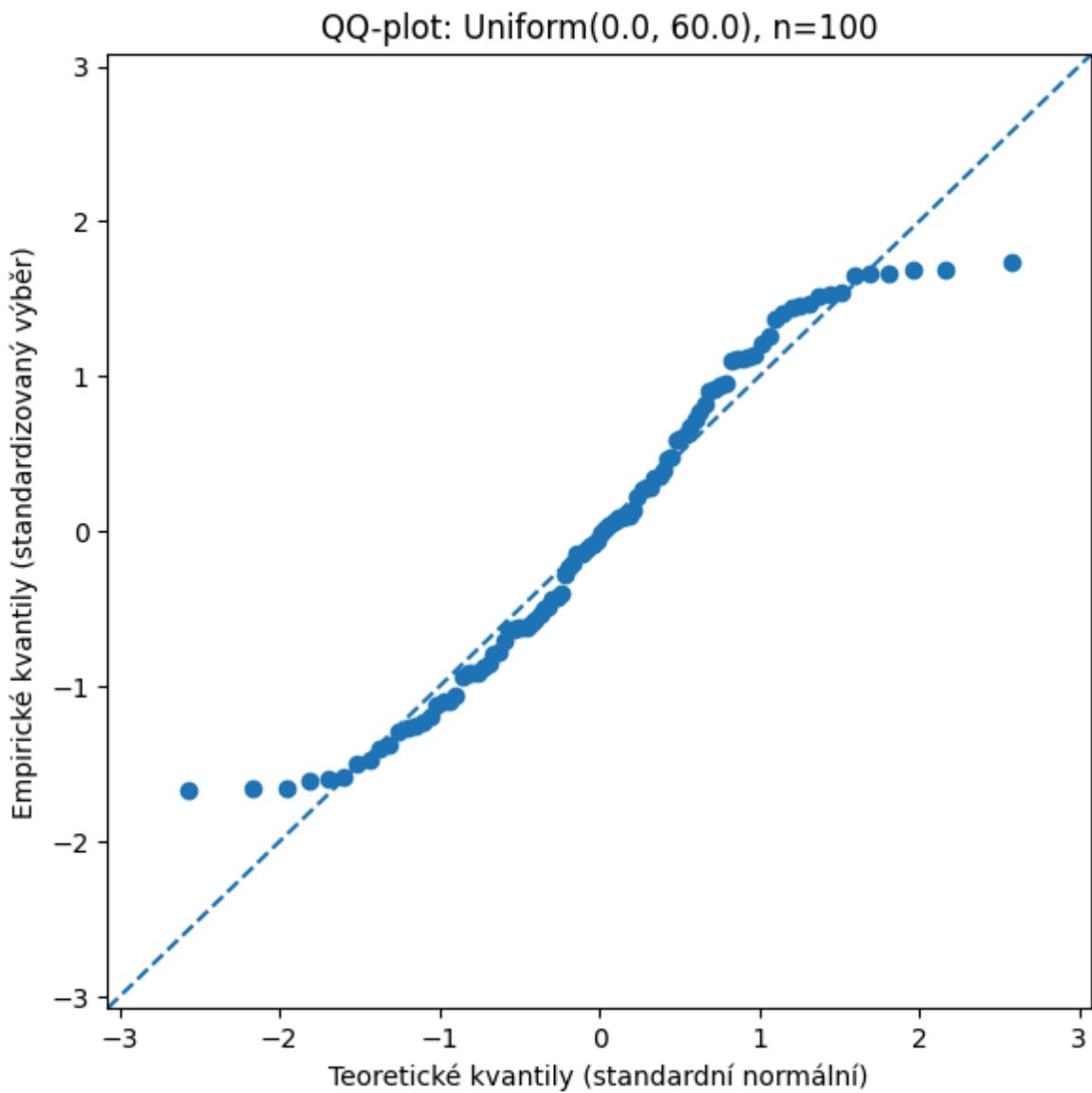
empirical_q = np.sort(z_sample) # standardní
```

## QQ-plot (teoretické na x, empirické na y)

```
In [170...]: fig, ax = plt.subplots(figsize=(6, 6))
ax.scatter(theoretical_q, empirical_q)

# reference line y = x
lims = [
    min(theoretical_q.min(), empirical_q.min()) - 0.5,
    max(theoretical_q.max(), empirical_q.max()) + 0.5,
]

ax.plot(lims, lims, linestyle="--")
ax.set_xlim(lims)
ax.set_ylim(lims)
ax.set_xlabel("Teoretické kvantily (standardní normální)")
ax.set_ylabel("Empirické kvantily (standardizovaný výběr)")
ax.set_title(f"QQ-plot: Uniform({low}, {high}), n={n}")
plt.tight_layout()
```



```
In [171...]: # připravíme tabulku s některými hodnotami (vzor, standardizovaná hodnota, pořadí)
df = pd.DataFrame(
{
    "pořadí": np.arange(1, n + 1),
    "hodnota": np.sort(sample),
    "standardizováno (z)": empirical_q,
    "teoretický_q": theoretical_q,
}
)
df.head()
```

	<b>pořadí</b>	<b>hodnota</b>	<b>standardizováno (z)</b>	<b>teoretický_q</b>
<b>0</b>	1	0.560915	-1.666645	-2.575829
<b>1</b>	2	0.653324	-1.661337	-2.170090
<b>2</b>	3	0.788181	-1.653591	-1.959964
<b>3</b>	4	1.637233	-1.604823	-1.811911
<b>4</b>	5	1.856940	-1.592204	-1.695398

## Posud'te, jak se výběrové empirické rozdělení liší od normálního

QQ-graf porovnává dvě rozdělení:

- teoretické kvantily normálního rozdělení (na ose X),
- empirické kvantily tvého výběru (na ose Y).

Každý bod představuje „odpovídající pozici“ ve dvou rozděleních – např. 10. percentil, 50. percentil, 90. percentil atd.

Pokud by výběr byl opravdu normálně rozdělený, body by ležely přibližně na přímce  $y = x$ .

Při porovnání QQ-plotu je vidět:

- střed (kolem mediánu) se může zdát podobný — hodnoty nejsou moc odlišné
- okraje (taily) se ale silně liší: Rovnoměrné rozdělení má konečné meze (např. od 0 do 60). Normální rozdělení se teoreticky rozprostírá do nekonečna.

Proto v levé části (malé kvantily) jsou body nad přímkou a v pravé části (velké kvantily) jsou body pod přímkou => „esovitý“ (S) tvar grafu — typický znak, že rozdělení má kratší ocasys než normální.