Tempa skladeb různých žánrů

```
Filip Kastl, II. ročník
letní semestr 2021/22
Pravděpodobnost a statistika
```

```
from datetime import datetime
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
%matplotlib inline
import scipy.stats as st
import scipy.special as sp
```

Pro svůj zápočtový projekt jsem si vybral dataset ze stránky Kaggle -- https://www.kaggle.com/datasets/insiyeah/musicfeatures . Obsahuje informace o 1000 úryvcích ze skladeb různých žánrů. Mě bude zajímat tempo skladeb. To se měří v bpm -- beats per minute.

```
DATA_PATH = "data/data.csv"
df = pd.read_csv(DATA_PATH)
```

```
# Počty úryvků jednotlivých žánrů
df["label"].value_counts()
```

```
blues        100
classical    100
country      100
disco        100
hiphop       100
jazz         100
metal        100
pop          100
reggae       100
rock         100
Name: label, dtype: int64
```

Nehodlám objevovat nová obecné fakta. Spíše hodlám zkoumat, z jaké množiny skladeb byly vybírány úryvky do tohoto datasetu. Hypotézy budu zakládat na tom, co bych od datasetu

očekával dle svých zkušeností s hudbou. Budu používat $\alpha = 0.05$.

Je metal rychlý?

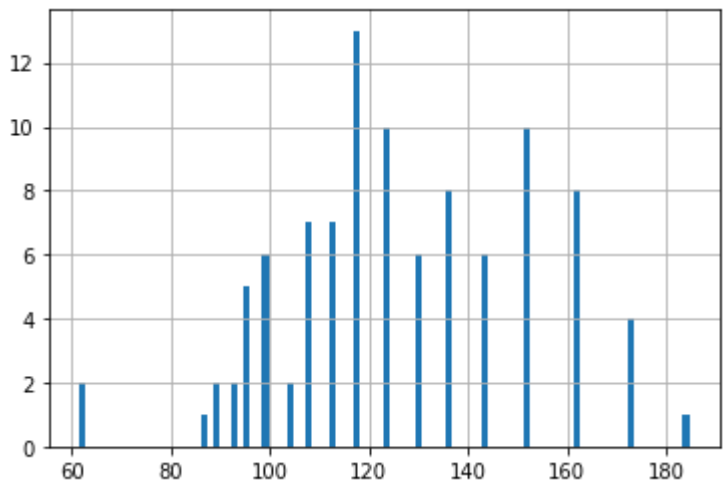
Metal je obecně považován za agresivní a tedy spíše rychlý žánr. Tudíž u náhodně vybrané metalové skladby bude tempo tíhnout spíše k vyšším hodnotám.

Nulovou hypotézou nechť je "Metalové skladby byly vybírány z množiny s průměrem 120 bpm". 120 bpm je považováno za standardní tempo. Pojďme zkusit nulovou hypotézu vyvrátit.

```
ser_metal = df.where(df["label"] == "metal")["tempo"]
```

```
# Pojďme se nejprve podívat na histogram temp metalu
ser_metal.hist(bins=100)
```

```
<AxesSubplot:>
```




```
# Teď vykonáme jednostranný t-test a podíváme se na p-hodnotu
st.ttest_1samp(
    ser_metal,
    120,
    nan_policy="omit",
    alternative="greater"
).pvalue
```

```
0.01298642171896439
```

$0.013 < 0.05$, takže můžeme zamítnout nulovou hypotézu a prohlásit, že dataset čerpá metalové úryvky ze skladeb průměrně rychlejších než 120 bpm.

Je jazz pomalý?

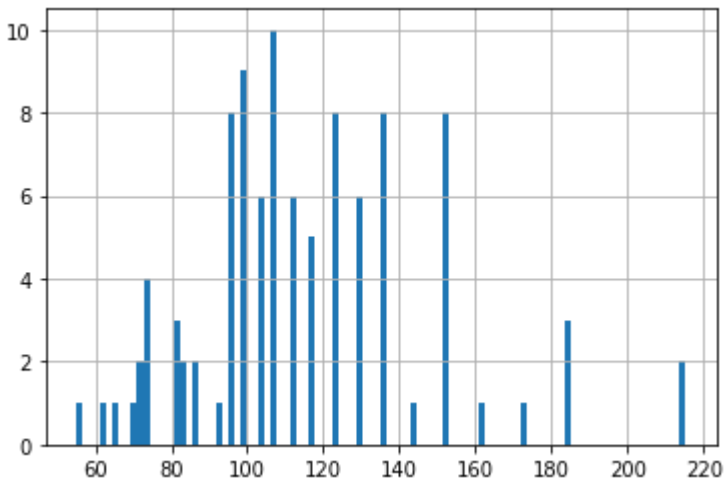
Budeme zkoumat téměř to samé jako v předchozím případě. Tentokrát se však budeme ptát, jestli dataset čerpá jazzové úryvky ze spíše pomalejších skladeb.

Očekávám, že se nám nepovede nic ukázat, jelikož "jazz je pohodová a tedy pomalá hudba" je spíše mýtus.

```
ser_jazz = df.where(df["label"] == "jazz")["tempo"]
```

```
# Pojďme se nejprve podívat na histogram temp jazzu
ser_jazz.hist(bins=100)
```

```
<AxesSubplot:>
```




```
# Teď vykonáme jednostranný t-test a podíváme se na p-hodnotu
st.ttest_1samp(
    ser_jazz,
    120,
    nan_policy="omit",
    alternative="less"
).pvalue
```

```
0.05536641540016916
```

$0.055 > 0.05$, takže nemůžeme zamítnout nulovou hypotézu.

Hraje disco počítač?

Podívejme se teď na něco zajímavějšího. Disco je žánr spoléhající se na syntezátory. V takových žánrech je často za tempo zodpovědný počítač místo lidí.

Jak to vypozorujeme z našich dat? Málokdo navolí na počítači schválně neceločíselné tempo. Tedy úryvky s celočíselným tempem můžeme považovat za hrané počítačem.

Bohužel úryvky byly zpracovány automatickým rozpoznáváním tempa. To nebude absolutně přesné. Abychom se s tím vypořádáme, označíme za celočíselná ta tempa, která jsou ve vzdálenosti 0.1 od celého čísla. Budeme se pak snažit vyvrátit nulovou hypotézu "Celočíselnost disco úryvků se řídí stejným rozdělením jako celočíselnost jiných žánrů."

```
ser_tail = df["tempo"] % 1
df["whole"] = (ser_tail < 0.1) | (ser_tail > 0.9)
df["whole"] = df["whole"].astype("int")
```

```
ser_disco = df.where(df["label"] == "disco")["whole"]
ser_no_disco = df.where(df["label"] != "disco")["whole"]
```

t-test

```
# Vykonáme jednostranný dvouvýběrový t-test
st.ttest_ind(
    ser_disco,
    ser_no_disco,
    nan_policy="omit",
    alternative="greater"
).pvalue
```

```
0.00011976833381252274
```

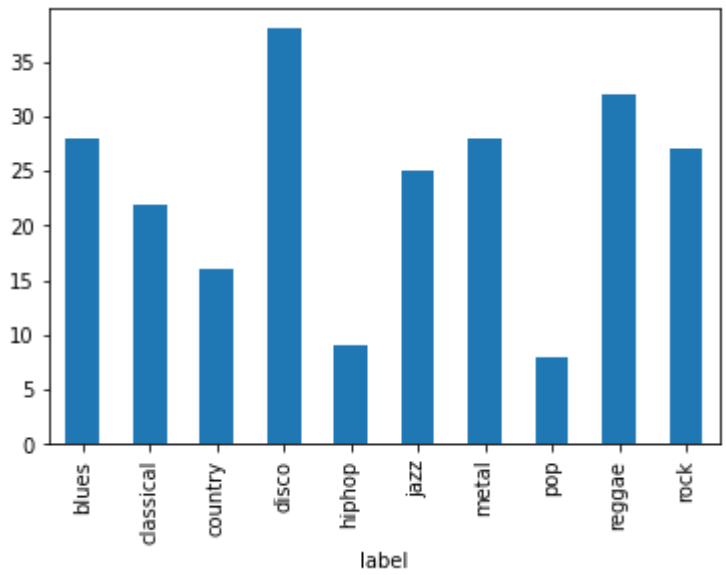
$0.0001 < 0.05$ a to o hodně. Můžeme zamítnout nulovou hypotézu. Tento test nám říká, že úryvky disca byly vybírány ze skladeb mezi kterými bylo více celočíselných temp než mezi

skladbami, z kterých byly vybírány úryvky jiných žánrů.

Vizualizace

```
# Podívejme se, jak vypadají počty celočíselných skladeb napříč žánry
df.groupby("label")["whole"].sum().plot(kind="bar")
```

```
<AxesSubplot:xlabel='label'>
```




```
# Podívejme se na poměry celočíselných skladeb v disco
# úryvcích a v ostatních úryvcích
df["is disco"] = df["label"] == "disco"
df.groupby("is disco")["whole"].mean()
```

```
is disco
False    0.216667
True     0.380000
Name: whole, dtype: float64
```

Kdybychom nepřišli s hypotézou, že disco tíhne k celočíselným bpm, díky předchozím zkušenostem, mohli bychom provést explorativní analýzu třeba na půlce dat, dostali bychom graf a čísla podobná těm výše, zformulovali bychom hypotézu a ověřili ji na druhé půlce dat.

Mimochodem, je zvláštní, že pop v tomto datasetu má nízkou celočíselnost. V produkci dnešního popu hodně figuruje počítač. Jedním možným vysvětlením je, že popové úryvky byly

vybírány ze starších skladeb.

Jiný test: χ^2

Pojďme zkusit vyvrátit předchozí nulovou hypotézu (disco je stejně celočíselné jako jiné žánry) jiným testem -- čistě jen proto, abych si to vyzkoušel.

Za nulové hypotézy by platilo, že mezi neceločíselnými úryvky bude stejný poměr disco úryvků a úryvků jiných žánrů.

```
not_whole_total = df["whole"].value_counts()[0]
print(not_whole_total) # Počet neceločíselných nahrávek
whole_total = df["whole"].value_counts()[1]
print(whole_total) # Počet celočíselných nahrávek
```

```
767
233
```

```
disco_total = 100
other_total = 900
n = disco_total + other_total
a_exp = (disco_total / n) * not_whole_total
print(a_exp) # Předpokládaný počet neceločíselných disco nahrávek
b_exp = (other_total / n) * not_whole_total
print(b_exp) # Předpokládaný počet neceločíselných ne-disco nahrávek
```

```
76.7
690.3000000000001
```

```
a_obs = df.where(df["label"] == "disco")["whole"].value_counts()[0]
print(a_obs) # Pozorovaný počet neceločíselných disco nahrávek
b_obs = df.where(df["label"] != "disco")["whole"].value_counts()[0]
print(b_obs) # Pozorovaný neceločíselných nedisco nahrávek
```

```
62
705
```

Zkusíme použít test dobré shody -- konkrétně χ^2. Máme očekávaný a pozorovaný počet disco skladeb mezi neceločíselnými úryvky. Ovšem na přednášce jsme používali test dobré shody na složitější multinomické rozdělení. Tohle multinomické rozdělení obsahuje pouze 2 hodnoty a tedy nevím, jestli je test vhodný.

```
st.chisquare([a_obs, b_obs], [a_exp, b_exp]).pvalue
```

```
0.07684591597783241
```

S χ^2 testem se nám nulovou hypotézu nepodařilo vyvrátit ($0.077 > 0.05$). Možná na tuto situaci tento test není vhodný -- je v této situaci slabý.

Poznámka: Studentův test

Většinu hypotéz jsme ověřovali pomocí Studentova t-testu. Ovšem nemáme záruku toho, že pracujeme s normálně rozdělenými náhodnými veličinami. Naštěstí pracujeme s velkými výběry ($n \geq 100$), takže se můžeme odvolat na Centrální limitní větu.

Protože neznáme rozptyl náhodných veličin, se kterými pracujeme, používáme t-test místo z-testu.