

Testy normalności



❖ Test zgodności

Test zgodności

- ▶ jest testem nieparametrycznym;



❖ Test zgodności

Test zgodności

- ▶ jest testem nieparametrycznym;
- ▶ służy do weryfikacji hipotezy mówiącej o **zgodności**
 - ▶ **rozkładu, z którego pochodzą analizowane dane** (tj. dane z próby losowej)
 - ▶ z wybranym **rozkładem teoretycznym**.



❖ Test zgodności

Test zgodności

- ▶ jest testem nieparametrycznym;
- ▶ służy do weryfikacji hipotezy mówiącej o **zgodności**
 - ▶ **rozkładu, z którego pochodzą analizowane dane** (tj. dane z próby losowej)
 - ▶ z wybranym **rozkładem teoretycznym**.

przypomnienie:

hipoteza zerowa jest postaci: $H_0 : F = F_0$, tzn. “dystrybuantą badanej cechy jest F_0 ”



❖ Test normalności

Test normalności

- ▶ to **szczególny rodzaj testu zgodności**;
- ▶ służy do weryfikacji hipotezy mówiącej o **zgodności**
 - ▶ **rozkładu, z którego pochodzą analizowane dane**
 - ▶ z **rozkładem normalnym**.



❖ Test normalności

Test normalności

- ▶ to **szczególny rodzaj testu zgodności**;
- ▶ służy do weryfikacji hipotezy mówiącej o **zgodności**
 - ▶ **rozkładu, z którego pochodzą analizowane dane**
 - ▶ z **rozkładem normalnym**.

hipoteza zerowa jest postaci: H_0 : “badana cecha ma rozkład normalny $\mathcal{N}(m, \sigma^2)$ ”



Test normalności Shapiro-Wilka



❖ Test normalności Shapiro-Wilka W

Test normalności Shapiro-Wilka W

- ▶ Jest jednym z **najbardziej efektywnych** testów normalności.



❖ Test normalności Shapiro-Wilka W

Test normalności Shapiro-Wilka W

- ▶ Jest jednym z **najbardziej efektywnych** testów normalności.
- ▶ Jego ogromną zaletą jest **możliwość stosowania go w przypadku małych prób**.



❖ Test normalności Shapiro-Wilka W

Test normalności Shapiro-Wilka W

- ▶ Jest jednym z **najbardziej efektywnych** testów normalności.
- ▶ Jego ogromną zaletą jest **możliwość stosowania go w przypadku małych prób**.
- ▶ Charakteryzuje się jednak **wysoką mocą również w przypadku dużych prób**.



Test normalności Shapiro-Wilka w *Pythonie*

```
177
178     default="y",
179 )
180 global_scale_setting = FloatProperty(
181     name="Scale",
182     min=0.01, max=1000.0,
183     default=1.0,
184 )
185
186 def execute(self, context):
187     # get the folder
188     folder_path = (os.path.dirname(self.filepath))
189
190     # get objects selected in the viewport
191     viewport_selection = bpy.context.selected_objects
192
193     # get export objects
194     obj_export_list = viewport_selection
195     if self.use_selection_setting == False:
196         obj_export_list = [i for i in bpy.context.scene.objects]
197
198     # deselect all objects
199     bpy.ops.object.select_all(action='DESELECT')
200
201     for item in obj_export_list:
202         item.select = True
203         if item.type == 'MESH':
204             file_path = os.path.join(folder_path, "{}.obj".format(item.name))
205             bpy.ops.export_scene.obj(filepath=file_path, use_selection=True,
206                 axis_forward=self.axis_forward_setting,
207                 axis_up=self.axis_up_setting,
208                 use_animation=self.use_animation_setting,
209                 use_mesh_modifiers=self.use_mesh_modifiers_setting,
210                 use_edges=self.use_edges_setting,
211                 use_smooth_groups=self.use_smooth_groups_setting,
212                 use_smooth_groups_bitflags=self.use_smooth_groups_bitflags_setting,
213                 use_normals=self.use_normals_setting,
214                 use_uv=self.use_uv_setting,
215                 use_materials=self.use_materials_setting,
```

Źródło: <https://unsplash.com/> (Pixabay)



❖ Test normalności Shapiro-Wilka w *Pythonie* – przykład

Przykład

Zebrano i uporządkowano w kolejności rosnącej 12 obserwacji dotyczących wskaźnika masy ciała (BMI) pacjentów.

20,6	22,5	23,6	24,5	24,6	25,5	25,9	26,2	26,4	27,1	30,4	32,2
------	------	------	------	------	------	------	------	------	------	------	------

Aby przeprowadzić analizę statystyczną, musimy sprawdzić, czy rozkład tej cechy jest normalny.



❖ Test normalności Shapiro-Wilka w *Pythonie* – przykład

```
1 import numpy as np
2 from scipy import stats
3
4 u = np.array([20.6, 22.5, 23.6, 24.5, 24.6, 25.5, 25.9, 26.2, 26.4, 27.1, 30.4, 32.2])
5
6 print(stats.shapiro(u))
```

ShapiroResult(statistic=0.9541850651024538, pvalue=0.6987224777880876)



❖ Bibliografia

Wykład i slajdy zostały przygotowane w oparciu o następującą literaturę:

- [1] P.L. Gatti, *Probability Theory and Mathematical Statistics for Engineers*, Spon Press, London and New York 2005.
- [2] S.C. Gupta, V.K. Kapoor, *Fundamentals of Mathematical Statistics (A Modern Approach)*, Sultan Chand & Sons, New Delhi 2000.
- [3] J. Koronacki, J. Mielniczuk, *Statystyka dla studentów kierunków technicznych i przyrodniczych*, Wydawnictwa Naukowo-Techniczne, Warszawa 2001.
- [4] **W. Krysicki, J. Bartos, W. Dyczka, K. Królikowska, M. Wasilewski, *Rachunek prawdopodobieństwa i statystyka matematyczna w zadaniach, część II, Statystyka matematyczna*, Wydawnictwo naukowe PWN, Warszawa 1998.**
- [5] H. Pishro-Nik, *Introduction to Probability, Statistics, and Random Processes*, available at <https://www.probabilitycourse.com/>, Kappa Research LLC, 2014.
- [6] Marian Płaszczycyca, strona internetowa "Statystyka od A do Z", <https://www.statystyka.eu/testy.php>.
- [7] **M. Sobczyk, *Statystyka*, Wydawnictwo Naukowe PWN, Warszawa 1999.**

