Kolmogorov-Smirnov Goodness of Fit Test

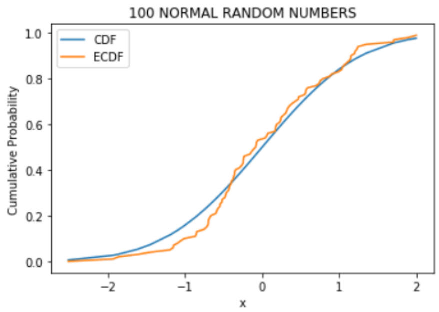
The Kolmogorov–Smirnov test is a nonparametric **goodness-of-fit test** and is used to determine whether two distributions differ, or whether an underlying probability distribution differs from a hypothesized distribution. It is used to decide if a sample comes from a population with a specific distribution.

The Kolmogorov-Smirnov (K-S) test is based on the empirical distribution function (**ECDF**). Given N ordered data points $\mathbf{X_1, X_2, \ldots, X_N}$, the ECDF is defined as

$$\mathbf{E_N = n(i)/N}$$

where $\mathbf{n(i)}$ is the number of points less than $\mathbf{X_i}$ and the $\mathbf{X_i}$ are ordered from smallest to largest value. This is a step function that increases by 1/N at the value of each ordered data point.

```
<matplotlib.legend.Legend at 0x1e646fa26c8>
```



The graph shown is a plot of the empirical distribution function with a normal cumulative distribution function for 100 normal random numbers. The K-S test is based on the maximum distance between these two curves.

Definition.

Suppose that we have an i.i.d. sample $X_1, \ldots, X_n$ with some unknown distribution P and we would like to test the hypothesis that P is equal to a particular distribution P_0, i.e., decide between the following hypotheses:

$$\mathbf{H_0\text{: The sample comes from the population } P_0\ (P = P_0)}$$

$$\mathbf{H_a\text{: The sample does not come from the population } P_0\ (P \neq P_0)}$$

We already know how to test this hypothesis using chi-squared goodness-of-fit test. If distribution P_0 is continuous, we had to group the data and consider a weaker discretized null hypothesis. We will now consider a different test for H_0 based on a quite different idea that avoids this discretization.

As stated before, we want to compare the empirical distribution function of the data, F_{obs}, with the cumulative distribution function associated with the null hypothesis, F_{exp} (expected CDF).

The Kolmogorov-Smirnov test statistic is:

$$D_n = max_x|F_{exp}(x) \ - \ F_{obs}(x)|$$

Steps to Perform KS Test

1. Order the data that has been given and compute the Empirical Distribution Function, **F_{obs}.**
2. For each observation x_i, we compute $F_{exp}(x_i) \ = \ P(Z \ \leq \ x_i)$. Since, the expected distribution function is standard normal, we can use the normal distribution table to get those values.
3. Compute the absolute differences between F_{obs} and F_{exp} for each observation in the data available. The maximum of the absolute differences for each observation is the KS test statistic (D_n).
4. Using the KS table, find out the critical value $D_{n,\alpha}$ corresponding to the number of observations **n** and the significance level $\boldsymbol{\alpha}$**.**
5. Compare the values of critical value $D_{n,\alpha}$ and observed test statistic $\boldsymbol{D_n}$.
   i. Reject null hypothesis H_0 if $D_{crit} < \ D_n$.
   ii. Do not reject null hypothesis H_0 if $D_{crit} > \ D_n$.
6. **P-value approach**
   P-value = $P(D_{m,n} \geq D_{obs}|H_0)$
   If p-value > $\boldsymbol{\alpha}$, we fail to reject the null hypothesis whereas if p-value < $\boldsymbol{\alpha}$, we reject the null hypothesis.

Advantages of KS Test

1. An attractive feature of this test is that the distribution of the K-S test statistic itself does not depend on the underlying cumulative distribution function being tested.
2. Another advantage is that it is an exact test (the chi-square goodness-of-fit test depends on an adequate sample size for the approximations to be valid).
3. It does not assume that data are sampled from Gaussian distributions (or any other defined distributions).

4. The results will not change if you transform all the values to logarithms or reciprocals or any transformation. The KS test reports the maximum difference between the two cumulative distributions and calculates a P value from that and the sample sizes. A transformation will stretch (even rearrange if you pick a strange transformation) the X axis of the frequency distribution but cannot change the maximum distance between two frequency distributions.

Limitations of KS Test

1. It only applies to continuous distributions.
2. It tends to be more sensitive near the center of the distribution than at the tails.
3. Perhaps the most serious limitation is that the distribution must be fully specified. That is, if location, scale, and shape parameters are estimated from the sample data, the critical region of the K-S test is no longer valid. It typically must be determined by simulation.

Application of the KS Test

Consider the ongoing Cardiovascular Study data obtained for 848 people, randomly sampled from the city of Massachusetts. The dataset (test.csv) provides the patient's information including the sex, age, heart rate, BMI etc.

The data for the BMI of each patient has been extracted from the given dataset and all the null values have been excluded from the dataset using Python. The BMI values for each patient follow a continuous distribution.

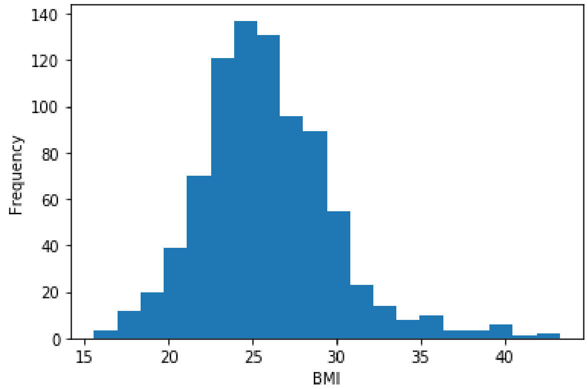
We observe that our experiment conditions and data fulfill the criteria of the KS-test, and hence we can apply this test to verify if our sample conforms to a certain distribution.

Hypothesis Test for Normal Distribution

We would like to check if our sample comes from a normal distribution or not. Based on studies conducted on BMI in the United States [6], we obtain μ=25.3 and σ=3.9 where μ and σ are the mean and standard deviation for all the citizens in the United States of America. Hence, we formulate a null and alternate hypothesis:

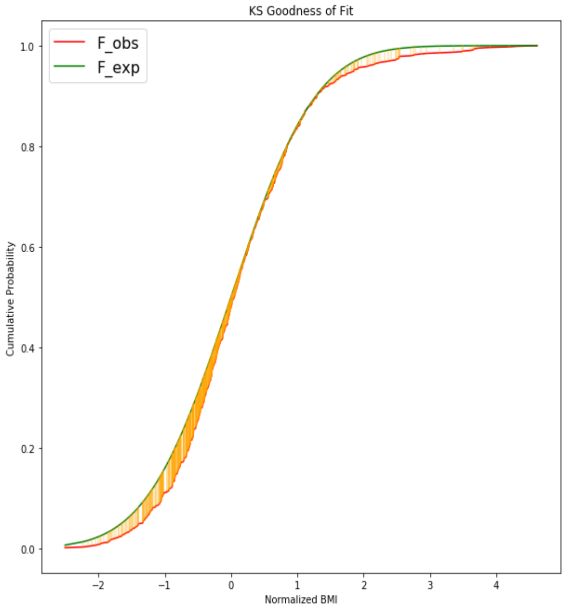
H_0: The sample comes from a normal population with $\boldsymbol{\mu}$=25.3 and $\boldsymbol{\sigma}$=3.9

H_a: The sample does not come from a normal population with $\boldsymbol{\mu}$=25.3 and $\boldsymbol{\sigma}$=3.9



The data is plotted in the form of a histogram in the figure given after distributing the data into 20 equal divisions.

Using a Python code provided in the references, the critical value of the test statistic and the observed test statistic value are obtained, for the significance level α=0.05.



$$D_n=0.06654647919876744$$
$$D_{crit}=0.04684088814164377$$

A KS curve has been plotted using Python showing the margins between the $F_{exp}(x)$ and $F_{obs}(x)$ curves.

We observe that $D_{crit} < D_n$, this implies that we have to reject the null hypothesis, hence the sample does not belong to a normal distribution with μ=25.3 and σ=3.9.

If we are to solve it using P-value approach, it is seen that the P-value= 0.000831. This indicates that the p-value $<$ α, we reject the null hypothesis. ($\alpha = 0.05$)

Textual References

[1] (2008) Kolmogorov–Smirnov Test. In: The Concise Encyclopedia of Statistics. Springer, New York, NY. https://doi.org/10.1007/978-0-387-32833-1_214

[2] Chakravarti, Laha, and Roy, (1967). Handbook of Methods of Applied Statistics, Volume I, John Wiley and Sons, pp. 392-394.

[3] David Steinsaltz, Introduction to Probability Theory and Statistics for Psychology and Quantitative Methods for Human Sciences, University of Oxford. http://www.stats.ox.ac.uk/~filippi/Teaching/psychology_humanscience_2015/lecture_notes.pdf

[4] 1.3.5.16. Kolmogorov-Smirnov Goodness-of-Fit Test. https://www.itl.nist.gov/div898/handbook/eda/section3/eda35g.htm

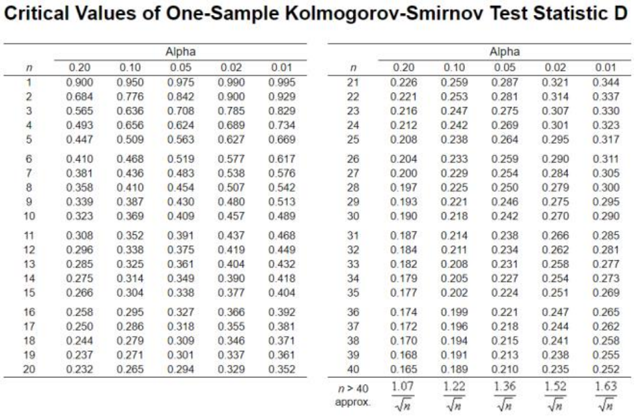
[5] Statistics for Applications Course, Massachusetts Institute of Technology. https://ocw.mit.edu/courses/mathematics/18-443-statistics-for-applications-fall-2006/lecture-notes/lecture14.pdf

[6] Block JP, Subramanian SV, Christakis NA, O'Malley AJ. Population trends and variation in body mass index from 1971 to 2008 in the Framingham Heart Study Offspring Cohort. *PLoS One*. 2013;8(5):e63217. Published 2013 May 10. doi:10.1371/journal.pone.0063217

[7] Massey, F. J. (1951). The Kolmogorov-Smirnov Test for Goodness of Fit. Journal of the American Statistical Association, 46(253), 68. doi:10.2307/2280095

Data for Reference

1. KS Table

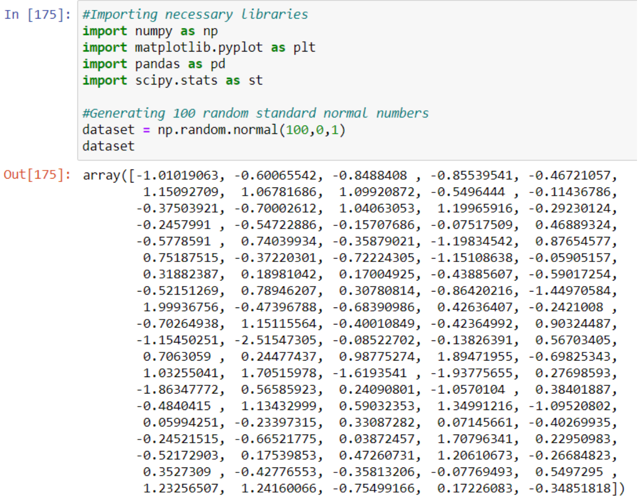


Critical Values of One-Sample Kolmogorov-Smirnov Test Statistic D

	Alpha				
n	0.20	0.10	0.05	0.02	0.01
1	0.900	0.950	0.975	0.990	0.995
2	0.684	0.776	0.842	0.900	0.929
3	0.565	0.636	0.708	0.785	0.829
4	0.493	0.656	0.624	0.689	0.734
5	0.447	0.509	0.563	0.627	0.669
6	0.410	0.468	0.519	0.577	0.617
7	0.381	0.436	0.483	0.538	0.576
8	0.358	0.410	0.454	0.507	0.542
9	0.339	0.387	0.430	0.480	0.513
10	0.323	0.369	0.409	0.457	0.489
11	0.308	0.352	0.391	0.437	0.468
12	0.296	0.338	0.375	0.419	0.449
13	0.285	0.325	0.361	0.404	0.432
14	0.275	0.314	0.349	0.390	0.418
15	0.266	0.304	0.338	0.377	0.404
16	0.258	0.295	0.327	0.366	0.392
17	0.250	0.286	0.318	0.355	0.381
18	0.244	0.279	0.309	0.346	0.371
19	0.237	0.271	0.301	0.337	0.361
20	0.232	0.265	0.294	0.329	0.352
21	0.226	0.259	0.287	0.321	0.344
22	0.221	0.253	0.281	0.314	0.337
23	0.216	0.247	0.275	0.307	0.330
24	0.212	0.242	0.269	0.301	0.323
25	0.208	0.238	0.264	0.295	0.317
26	0.204	0.233	0.259	0.290	0.311
27	0.200	0.229	0.254	0.284	0.305
28	0.197	0.225	0.250	0.279	0.300
29	0.193	0.221	0.246	0.275	0.295
30	0.190	0.218	0.242	0.270	0.290
31	0.187	0.214	0.238	0.266	0.285
32	0.184	0.211	0.234	0.262	0.281
33	0.182	0.208	0.231	0.258	0.277
34	0.179	0.205	0.227	0.254	0.273
35	0.177	0.202	0.224	0.251	0.269
36	0.174	0.199	0.221	0.247	0.265
37	0.172	0.196	0.218	0.244	0.262
38	0.170	0.194	0.215	0.241	0.258
39	0.168	0.191	0.213	0.238	0.255
40	0.165	0.189	0.210	0.235	0.252
$n > 40$ approx.	$\frac{1.07}{\sqrt{n}}$	$\frac{1.22}{\sqrt{n}}$	$\frac{1.36}{\sqrt{n}}$	$\frac{1.52}{\sqrt{n}}$	$\frac{1.63}{\sqrt{n}}$

2. Cardiovascular Study Data, test.csv
   https://www.kaggle.com/christofel04/cardiovascular-study-dataset-predict-heart-disea

3. Standard Normal Numbers Dataset generated using Python on Jupyter Notebook.



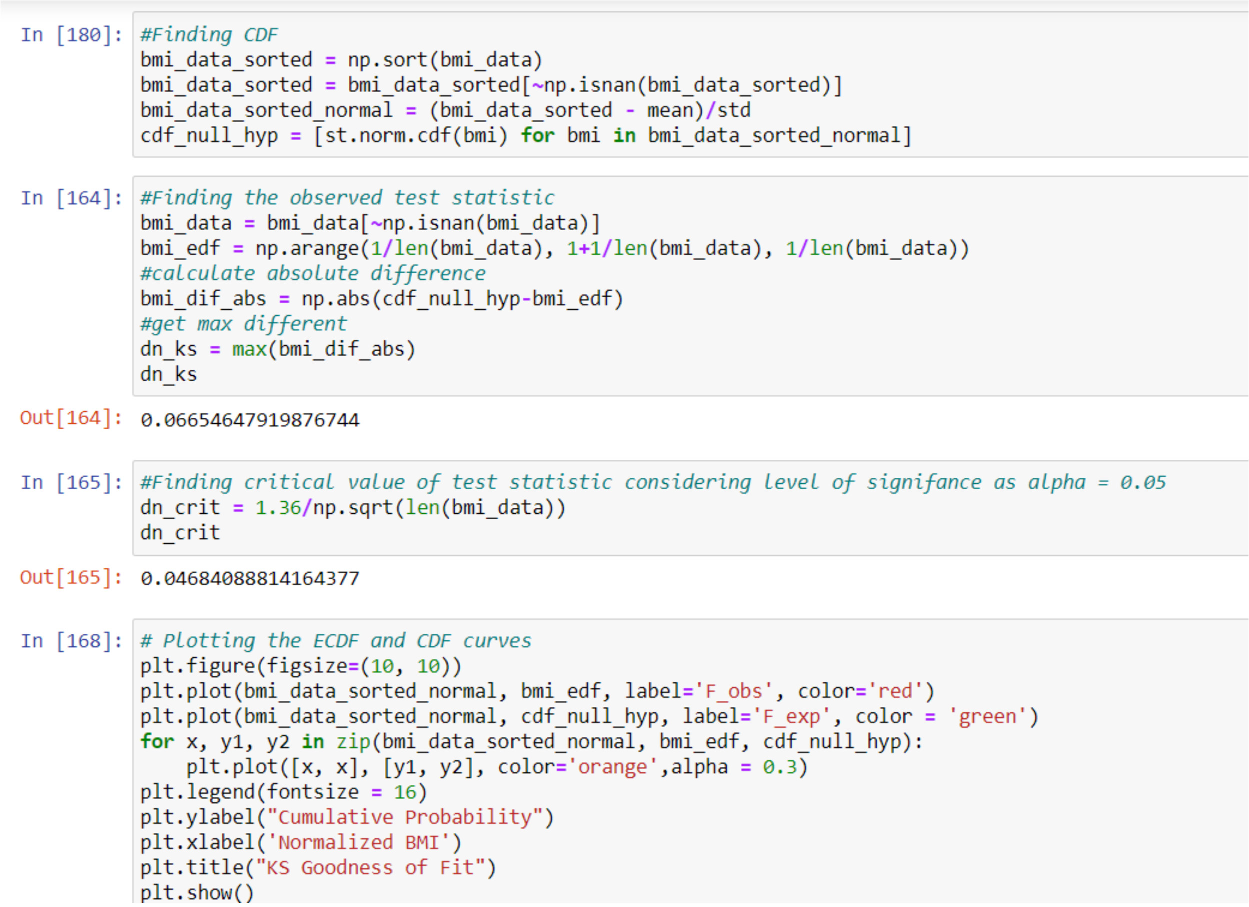
```
In [175]: #Importing necessary libraries
          import numpy as np
          import matplotlib.pyplot as plt
          import pandas as pd
          import scipy.stats as st

          #Generating 100 random standard normal numbers
          dataset = np.random.normal(100,0,1)
          dataset

Out[175]: array([-1.01019063, -0.60065542, -0.8488408 , -0.85539541, -0.46721057,
                  1.15092709,  1.06781686,  1.09920872, -0.5496444 , -0.11436786,
                 -0.37503921, -0.70002612,  1.04063053,  1.19965916, -0.29230124,
                 -0.2457991 , -0.54722886, -0.15707686, -0.07517509,  0.46889324,
                 -0.5778591 ,  0.74039934, -0.35879021, -1.19834542,  0.87654577,
                  0.75187515, -0.37220301, -0.72224305, -1.15108638, -0.05905157,
                  0.31882387,  0.18981042,  0.17004925, -0.43885607, -0.59017254,
                 -0.52151269,  0.78946207,  0.30780814, -0.86420216, -1.44970584,
                  1.99936756, -0.47396788, -0.68390986,  0.42636407, -0.2421008 ,
                 -0.70264938,  1.15115564, -0.40010849, -0.42364992,  0.90324487,
                 -1.15450251, -2.51547305, -0.08522702, -0.13826391,  0.56703405,
                  0.7063059 ,  0.24477437,  0.98775274,  1.89471955, -0.69825343,
                  1.03255041,  1.70515978, -1.6193541 , -1.93775655,  0.27698593,
                 -1.86347772,  0.56585923,  0.24090801, -1.0570104 ,  0.38401887,
                 -0.4840415 ,  1.13432999,  0.59032353,  1.34991216, -1.09520802,
                  0.05994251, -0.23397315,  0.33087282,  0.07145661, -0.40269935,
                 -0.24521515, -0.66521775,  0.03872457,  1.70796341,  0.22950983,
                 -0.52172903,  0.17539853,  0.47260731,  1.20610673, -0.26684823,
                  0.3527309 , -0.42776553, -0.35813206, -0.07769493,  0.5497295 ,
                  1.23256507,  1.24160066, -0.75499166,  0.17226083, -0.34851818])
```

Code Used to Perform Tests

1. KS Test



```
In [180]: #Finding CDF
          bmi_data_sorted = np.sort(bmi_data)
          bmi_data_sorted = bmi_data_sorted[~np.isnan(bmi_data_sorted)]
          bmi_data_sorted_normal = (bmi_data_sorted - mean)/std
          cdf_null_hyp = [st.norm.cdf(bmi) for bmi in bmi_data_sorted_normal]

In [164]: #Finding the observed test statistic
          bmi_data = bmi_data[~np.isnan(bmi_data)]
          bmi_edf = np.arange(1/len(bmi_data), 1+1/len(bmi_data), 1/len(bmi_data))
          #calculate absolute difference
          bmi_dif_abs = np.abs(cdf_null_hyp-bmi_edf)
          #get max different
          dn_ks = max(bmi_dif_abs)
          dn_ks

Out[164]: 0.06654647919876744

In [165]: #Finding critical value of test statistic considering level of signifance as alpha = 0.05
          dn_crit = 1.36/np.sqrt(len(bmi_data))
          dn_crit

Out[165]: 0.04684088814164377

In [168]: # Plotting the ECDF and CDF curves
          plt.figure(figsize=(10, 10))
          plt.plot(bmi_data_sorted_normal, bmi_edf, label='F_obs', color='red')
          plt.plot(bmi_data_sorted_normal, cdf_null_hyp, label='F_exp', color = 'green')
          for x, y1, y2 in zip(bmi_data_sorted_normal, bmi_edf, cdf_null_hyp):
              plt.plot([x, x], [y1, y2], color='orange',alpha = 0.3)
          plt.legend(fontsize = 16)
          plt.ylabel("Cumulative Probability")
          plt.xlabel('Normalized BMI')
          plt.title("KS Goodness of Fit")
          plt.show()
```