<u>Goodness of Fit Tests</u>

Student-Gosset Group

Aditya Garg	2019A8PS0309P
Agneya Bharadwaj	2019A8PS0297P
Harsh Solanki	2019A1PS0670P
Josh Wadhwa	2019A1PS0824P
Parth Saboo	2019A4PS0457P
Sagar Jain	2019A2PS0902P
Surya Rathi	2019A7PS0128P
Udbhav Tripathi	2019A4PS0323P
Yash Gupta	2019A7PS1138P

Birla Institute of Technology and Science (BITS), Pilani

October, 2021

Introduction

A *goodness of fit* test is used to test the hypothesis that a population has a specific historical or theoretical probability distribution. It tells how well the data fits the hypothesized distribution. *Goodness-of-fit* tests **never involve parametric modelling**.

A typical problem involving the application of GoF tests is: Let X be a random variable and we obtained 'n' number of identical and independent random variables from a random sampling. Does X have a distribution ~ N(0,1) or does X follow the student's T-distribution ?

Parametric And Non-Parametric Tests

Parametric Tests: Evaluates hypotheses for a population parameter and has complete information about the population parameter or that can make assumptions about the parameters of the population distribution from which the samples are drawn.

- Prior information of the population is completely known.
- Test statistics are formed based on the assumption of normal probabilistic distribution.

Examples include Z-test, student's T-test, F-test, and ANOVA test..

Non-parametric Tests: Researcher has no prior knowledge about the population parameter, neither can they make specific assumptions. Still they are required to test the hypothesis of the population.

- Test statistics are formed based on arbitrary population distribution assumptions.

  Examples: Kolmogorov Smirnov test , Chi-squared test , Anderson-Darling test, and Cramér-von Mises test.

Brief detail of some important nonparametric goodness of fit tests are as follows:

Anderson-Darling Test: It is the modification of the KS test, which gives more weight to the tails. Unlike the KS test, it makes use of the specified distribution in calculating the critical values. This has the advantage of allowing a more sensitive test and the disadvantage of critical values being calculated for each distribution.

The Anderson-Darling test statistic is defined as:

$$A^2 = -N - S,$$

where S is

$$\sum_{i=1}^{N} \frac{(2i-1)}{N}[\ln F(Y_i) + \ln(1-F(Y_{N+1-i}))]$$

and F is the CDF of the distribution which is to be tested

It is a one sided test, and the critical values depend upon the distribution to be tested. If the value of test statistic A is greater than the critical value, then the hypothesis of the distribution being of a specific form (normal) is rejected.

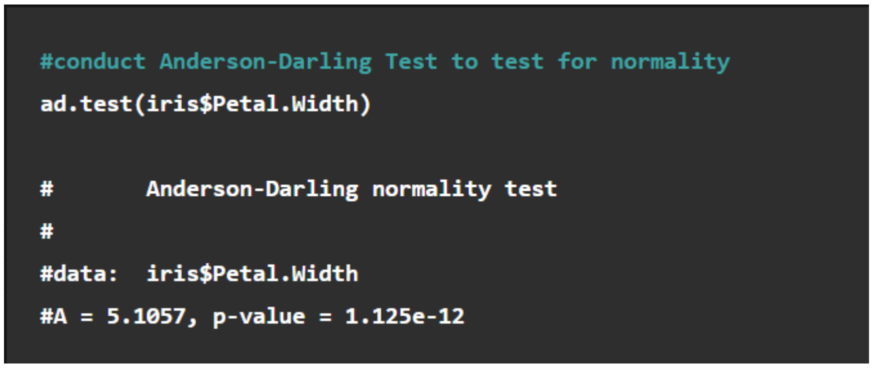
Example:

- H0 : The dataset follows Normal distribution
- H1 : The dataset does not follow the normal distribution

We refer to the Iris dataset, which consists of 150 samples. We observe the distribution of only the Petal_width feature of the dataset. Below histogram strongly suggests that normality is not there. We used an in-built function in R - ad.test() -to run the Anderson-Darling test..

Results: A = 5.107 , p-value = 1.125e-12

Conclusion: Since p-value of the test is very much less than the significance level of 0.05, thus we have sufficient evidence to reject the null hypothesis. We conclude that Petal_width does not follow a normal distribution.



```
#conduct Anderson-Darling Test to test for normality
ad.test(iris$Petal.Width)

#       Anderson-Darling normality test
#
#data:  iris$Petal.Width
#A = 5.1057, p-value = 1.125e-12
```

Shapiro -Wilk Test for Normality

It calculates a **W** statistic that determines whether a random sample $\{X_1, X_2, ..., X_n\}$ comes from a normal distribution. Smaller values of **W** denote departure from normality. This test outperforms many other goodness of fit tests

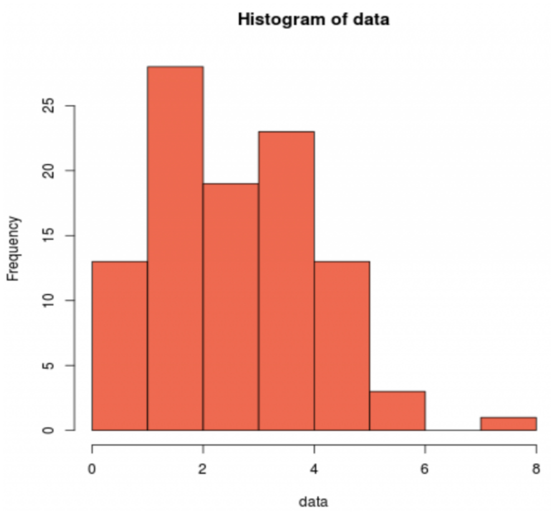
Here, the $x_{(i)}$ are the sample values of the order statistic and the a_i are constants generated from the means, variances & covariances of the order statistics of a sample (size = n) from a normal distribution.

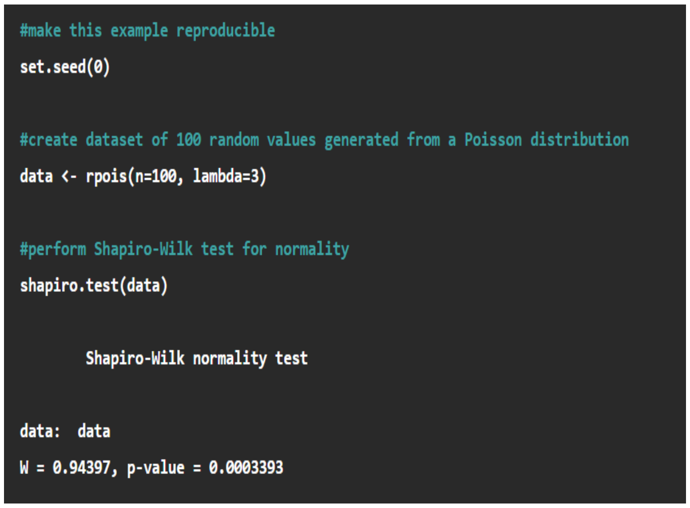
$$W = \frac{\left(\sum_{i=1}^{n} a_i x_{(i)}\right)^2}{\sum_{i=1}^{n} (x_i - \bar{x})^2}$$

Example:

Consider a dataset of size n=100 whose values are randomly generated from a Poisson distribution.

- H0 : The data follows gaussian distribution.
- H1: The data does not follow





```
#make this example reproducible
set.seed(0)

#create dataset of 100 random values generated from a Poisson distribution
data <- rpois(n=100, lambda=3)

#perform Shapiro-Wilk test for normality
shapiro.test(data)

        Shapiro-Wilk normality test

data:  data
W = 0.94397, p-value = 0.0003393
```

Method: We implemented using rpois() in-built function in R on the given dataset.
Results: Since the obtained p-value of the test is **0.0003393** which is less than 0.05 significance level, we have sufficient evidence to say that the sample data does not come from a Gaussian distribution.
Conclusion: From histogram, it is clear that the graph is **right-skewed,** and it matches the results of the Shapiro-Wilko test.

CHI-SQUARED GOODNESS-OF-FIT TEST

The Chi-squared goodness-of-fit test is a non-parametric test used to test whether a sample of data came from a specific distribution. Generally, it is used to determine whether the sample data can be used to represent the population or not. The chi-square goodness-of-fit test can be applied to any univariate distribution for which we can calculate the cumulative distribution function. The chi-square goodness-of-fit test is applied to binned data, that is, data put into classes. In the case of non-binned data, one can simply calculate a histogram or frequency table before generating the chi-square test.

Assumptions:

- The variable under consideration is assumed to be univariate categorical.
- The sampling method used is simple random sampling.

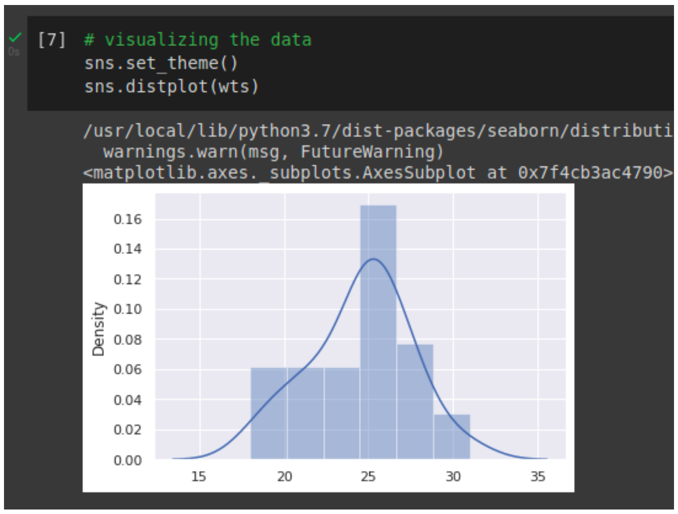
The data is divided into mutually exclusive and exhaustive classes, or bins. This test is mathematically represented as

$$\chi^2_c = \Sigma \frac{(O_i - E_i)^2}{E_i}$$

where, $\mathbf{O_i}$ = the observed frequency of the ith class, and
$\mathbf{E_i}$ = expected frequency of the ith class, based on the hypothesized distribution.

An advantage of the chi-square goodness-of-fit test is that it can be applied to discrete distributions - for instance, the binomial and the Poisson distributions - unlike the Kolmogorov-Smirnov and Anderson-Darling tests that are restricted to continuous distributions. However, it is dependent on how the data is binned and requires a sufficient sample size in order for the chi-square approximation to be valid.
Example:- Let us understand the application of Chi-squared goodness of fit test to decide whether a given sample comes from a population with a normal distribution or not.

Problem at Hand:
A team of pediatric researchers are studying the distribution of weights (in kg) for kindergarten students. They have a pretty good idea that the population follows a normal distribution. To test whether their hypothesis of the population being

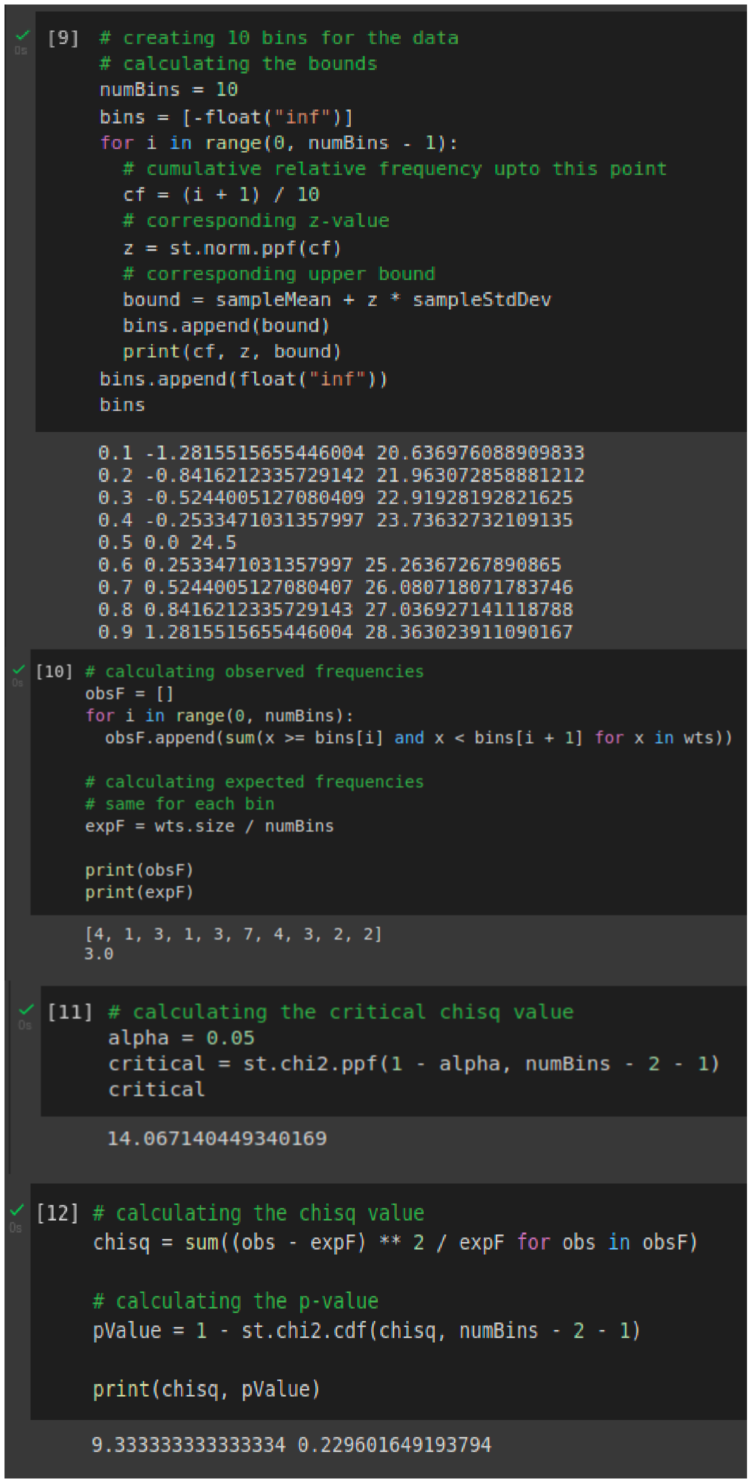


```
[7] # visualizing the data
    sns.set_theme()
    sns.distplot(wts)

    /usr/local/lib/python3.7/dist-packages/seaborn/distributi
      warnings.warn(msg, FutureWarning)
    <matplotlib.axes._subplots.AxesSubplot at 0x7f4cb3ac4790>
```

normal is correct, they select a random sample of weights of 30 children.
The steps involved to test whether the data comes from a normal distribution have been illustrated using Python snippets developed in the Google Colab environment.

Steps for the Solution:

- ➔ The data - the 30 weights - are loaded and hence converted to a numpy array.
- ➔ On plotting the data using seaborn.distplot(), we notice that the sample does not seem to provide any evidence of deviating significantly from the normal distribution.
- ➔ Sample mean and sample standard deviation are calculated. These are used in developing ten bins to put the observed values into.
- ➔ The bins are developed in such a way that we expect each of them to have the same frequencies. However, the observed frequencies - obtained by checking whether a data point in the sample is within the bounds of the bin - might be different for them.
- ➔ We choose to perform the hypothesis testing at a 5% significance level. The corresponding critical chi-squared value is calculated. The degrees of freedom in the test are $k - p - 1$, k being the number of bins and p being the number of parameters (mean and standard deviation).
- ➔ Chi-squared value for the sample data is evaluated using the expression involving expected and observed frequencies.
- ➔ We reject the null hypothesis if the p-value obtained is less than or equal to the alpha value. In our case, we do not have sufficient evidence to reject the (null) hypothesis that the data comes from a population following a normal distribution.



```
[9] # creating 10 bins for the data
    # calculating the bounds
    numBins = 10
    bins = [-float("inf")]
    for i in range(0, numBins - 1):
      # cumulative relative frequency upto this point
      cf = (i + 1) / 10
      # corresponding z-value
      z = st.norm.ppf(cf)
      # corresponding upper bound
      bound = sampleMean + z * sampleStdDev
      bins.append(bound)
      print(cf, z, bound)
    bins.append(float("inf"))
    bins

    0.1 -1.2815515655446004 20.636976088909833
    0.2 -0.8416212335729142 21.963072858881212
    0.3 -0.5244005127080409 22.91928192821625
    0.4 -0.2533471031357997 23.73632732109135
    0.5 0.0 24.5
    0.6 0.2533471031357997 25.26367267890865
    0.7 0.5244005127080407 26.080718071783746
    0.8 0.8416212335729143 27.036927141118788
    0.9 1.2815515655446004 28.363023911090167
```

```
[10] # calculating observed frequencies
     obsF = []
     for i in range(0, numBins):
       obsF.append(sum(x >= bins[i] and x < bins[i + 1] for x in wts))

     # calculating expected frequencies
     # same for each bin
     expF = wts.size / numBins

     print(obsF)
     print(expF)

     [4, 1, 3, 1, 3, 7, 4, 3, 2, 2]
     3.0
```

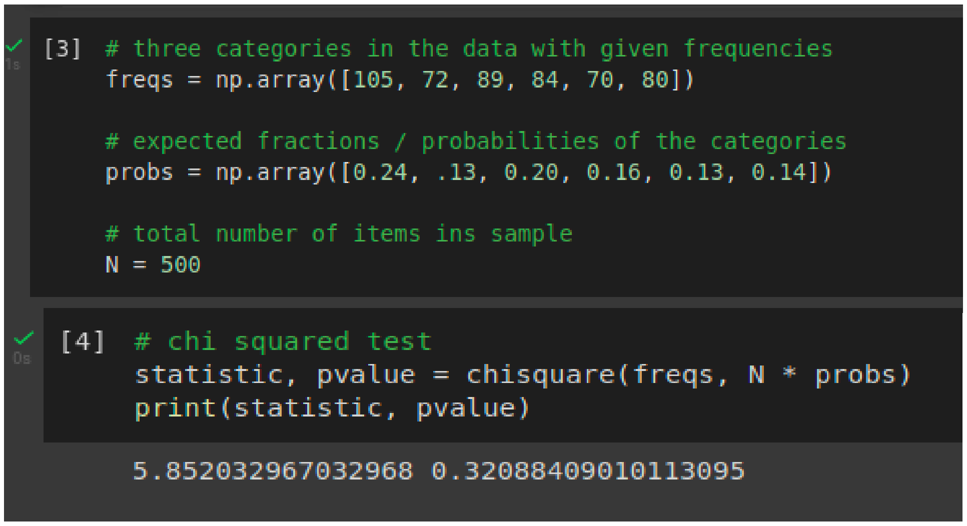
```
[11] # calculating the critical chisq value
     alpha = 0.05
     critical = st.chi2.ppf(1 - alpha, numBins - 2 - 1)
     critical

     14.067140449340169
```

```
[12] # calculating the chisq value
     chisq = sum((obs - expF) ** 2 / expF for obs in obsF)

     # calculating the p-value
     pValue = 1 - st.chi2.cdf(chisq, numBins - 2 - 1)

     print(chisq, pValue)

     9.333333333333334 0.229601649193794
```

Chi-Squared Goodness of Fit Test for Multinomial Distributions

The following is an application of the Chi-squared test to decide whether the data follows a historical or estimated multinomial distribution..

Mars, Inc. produces colored candies, including blue, brown, green, orange, red, and yellow candies. They want them to occur in any packaged bag of candies in a specified multinomial distribution, captured in the array probs. They test their hypothesis that the candies do follow the distribution by opening up random candy bags for a total of 500 candies. The actual number of the candies recorded are given in the array freqs.



```
[3] # three categories in the data with given frequencies
    freqs = np.array([105, 72, 89, 84, 70, 80])

    # expected fractions / probabilities of the categories
    probs = np.array([0.24, .13, 0.20, 0.16, 0.13, 0.14])

    # total number of items ins sample
    N = 500
```

```
[4] # chi squared test
    statistic, pvalue = chisquare(freqs, N * probs)
    print(statistic, pvalue)

    5.852032967032968 0.32088409010113095
```

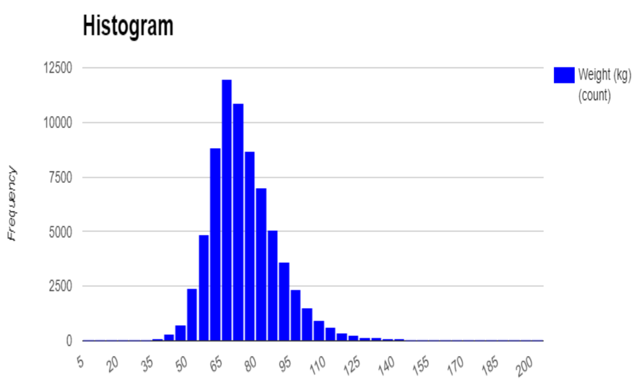
To test the hypothesis, they obtain a chi-squared statistic and a corresponding p-value using the scipy.stats.chisquare() library function.

The null hypothesis is rejected if the p-value is less than or equal to the alpha. In this case, the null hypothesis is not rejected, meaning the candies do seem to follow the specified distribution.

KOLMOGOROV-SMIRNOV GOODNESS OF FIT TEST

The Kolmogorov-Smirnov test is a nonparametric goodness-of-fit test 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 (KS) test is based on the empirical cumulative distribution function (ECDF). Given N ordered data points X_1, X_2, ..., X_N. The ECDF is defined as E_N= n(i)/N where n(i) is the number of points less than X_i and the 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.



Working of the KS Test

Given an independent and identically distributed sample with some unknown distribution, say P, we want to find out whether P is a particular distribution. This is accomplished via hypothesis testing. Hence we come up with the following hypotheses:

- $H_{o:}$ The sample comes from population P_o i.e. $P=P_o$
- $H_{a:}$ The sample doesn't come from population P_o i.e. $P \neq P_o$

Now we compare the empirical distribution function of the data F_{obs} with the cumulative distribution function associated with the null hypothesis F_{exp}. The KS test statistic is $D_n = \max_x |F_{exp}(x) - F_{obs}(x)|$

Steps to Perform the 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 those 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 α.
5. Compare the values of critical value $D_{n,\alpha}$ and observed test statistic D_n
    a. Reject null hypothesis H_o if $D_{crit} < D_n$
    b. Do not reject null hypothesis H_o, if $D_{crit} > D_n$

Advantage of the K-S Test:

- The K-S test is distribution-free, i.e. the distribution of the test statistic is not dependent on the cumulative distribution function being tested.
- There is no restriction on the sample size.
- This is an exact test while the chi-squared goodness-of-fit test depends on adequate sample size for the approximations to be valid.

Limitations of the K-S Test:

- It generally can only be used for continuous distributions and not discrete distributions.
- This test tends to be more sensitive near the center of the distribution than at the tails.
- The location, scale, and shape parameters must be fully specified. If they are calculated from the KS test, the test will become invalid.

Hypothesis Test for Normal Distribution

Here we test whether our sample comes from a normal distribution or not. The dataset consists of records of patient data in 12 features, such as age, gender, weight, systolic blood pressure, diastolic blood pressure, and etc. We take one feature i.e. patient weight and test whether it is normally distributed. The mean obtained is 74.2 and the variance is 207.23.

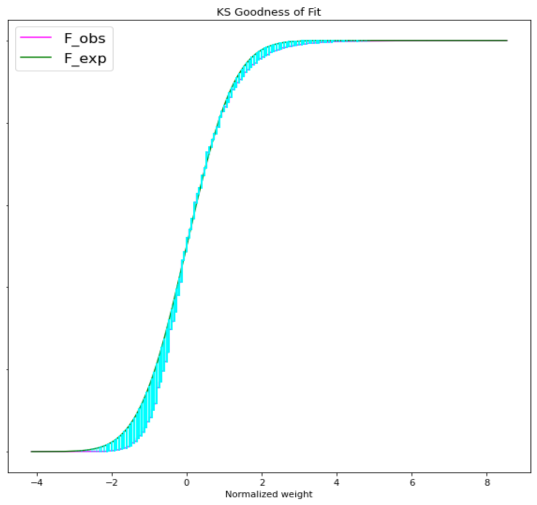
Hence we formulate the null and alternative hypotheses.

- H_o: The sample comes from a normal distribution with mean 74.2 and variance 207.23.
- H_o: The sample doesn't come from a normal distribution with mean 74.2 and variance 207.23.

Fig-1 Using Python, the critical value of the test statistic and the observed test statistic value are obtained, for a significance level of 0.05. The code is provided in the references.

D_n=0.0952553985833967

D_{crit}= 0.0051403168329254905



Since $D_{crit} < D_n$, we will have to reject the null hypothesis. This implies that the sample does not belong to the normal distribution with mean 74.2 and variance 207.23.

A KS curve has been plotted using Python which illustrates the margins between the F_{exp} and F_{obs} curves.

AKAIKE INFORMATION CRITERION

The Akaike Information Criterion (AIC) is a mathematical method used to compare the quality of a set of statistical models to each other and choose which model fits the data better, thereby acting as a model selector. The best fit model according to AIC is the one which explains the greatest amount of variation with minimum number of independent variables.The AIC value of a given model is given by:

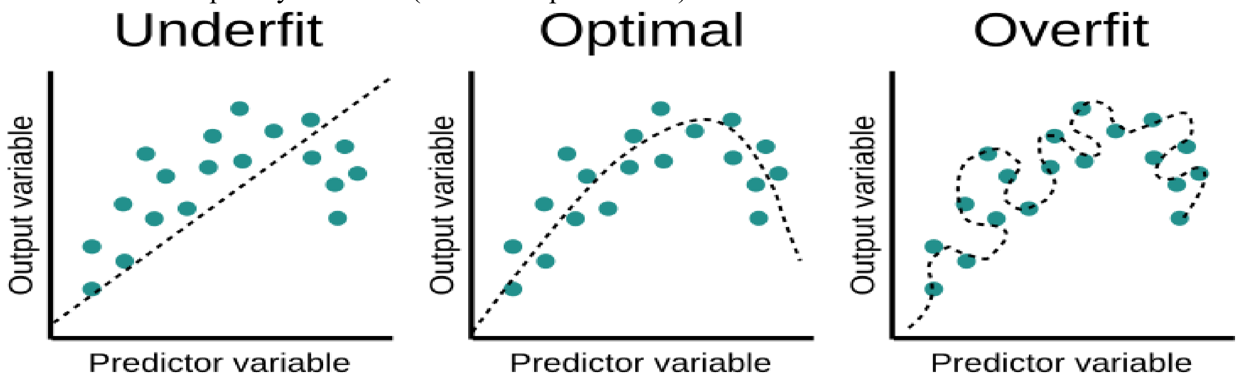
$$AIC = 2k - 2\ln(L)$$

where, k is the number of estimated parameters (independent variables) in the model, and

L is the maximum likelihood estimate of the model.

AIC in Practice:- To apply AIC, we start with potential candidate models, which can approximate the true model for the data, that is, these models will always have some information loss associated with them. We wish to choose the model which has the minimum information loss. The corresponding AIC values are calculated, and the result of selection is drawn from it.

Conclusion:- Lower the AIC score is, the better is the fit of the model, since low scores are associated with high likelihoods, which is desirable. It also adds a penalty for models with high parameter complexity since more parameters means the model is more likely to overfit the data, that is., it prevents generalization of the model. Therefore, AIC deals with the trade off between goodness of fit (MLE) of model and complexity of model (number of parameters).



Therefore, AIC is a measure of relative goodness of fit.

Appendix: Source Codes for the Problems Discussed

Chi-Squared Goodness of Fit Test for Normal Distribution

```
import numpy as np
import pandas as pd
import scipy.stats as st
import seaborn as sns

# reading the data file
df = pd.read_csv("/content/Weights.csv")
df.head()

# storing the sample observations in a numpy array
wts = np.array(df.Weights)
wts

# visualizing the data
sns.set_theme()
sns.distplot(wts)

# calculating the sample statistics
sampleMean, sampleStdDev =  wts.mean(), wts.std(ddof = 1)
print(sampleMean, sampleStdDev)

# creating 10 bins for the data
# calculating the bounds
numBins = 10
bins = [-float("inf")]
for i in range(0, numBins - 1):
  # cumulative relative frequency upto this point
  cf = (i + 1) / 10
  # corresponding z-value
  z = st.norm.ppf(cf)
  # corresponding upper bound
  bound = sampleMean + z * sampleStdDev
  bins.append(bound)
  print(cf, z, bound)
bins.append(float("inf"))
bins

# calculating observed frequencies
obsF = []
for i in range(0, numBins):
  obsF.append(sum(x >= bins[i] and x < bins[i + 1] for x in wts))

# calculating expected frequencies
# same for each bin
expF = wts.size / numBins

print(obsF)
print(expF)

# calculating the critical chisq value
alpha = 0.05
critical = st.chi2.ppf(1 - alpha, numBins - 2 - 1)
critical

# calculating the chisq value
chisq = sum((obs - expF) ** 2 / expF for obs in obsF)

# calculating the p-value
pValue = 1 - st.chi2.cdf(chisq, numBins - 2 - 1)

print(chisq, pValue)

# hypothesis testing
print("Null Hypothesis: Given distr in normal with mean = {mean} and std dev
= {stdDev}.".format(
    mean = sampleMean, stdDev = sampleStdDev
))
print("Alternative Hypothesis: Not normal with given mean and std dev.")

if (pValue <= alpha):
  print("Null hypothesis rejected.")
else:
  print("Null hypothesis not rejected.")
```

Chi-Squared Goodness of Fit Test for Multinomial Distribution

```
import pandas as pd
import numpy as np
from scipy.stats import chisquare

# three categories in the data with given frequencies
freqs = np.array([105, 72, 89, 84, 70, 80])

# expected fractions / probabilities of the categories
```

```
probs = np.array([0.24, .13, 0.20, 0.16, 0.13, 0.14])

# total number of items ins sample
N = 500

# chi squared test
statistic, pvalue = chisquare(freqs, N * probs)
print(statistic, pvalue)

# performing hypothesis test with p-value
alpha = 0.05 # significance level
print("At", alpha, "significance level: ", end = "")
print("Fitting" if pvalue > alpha else "Not Fitting")
```

Kolmogorov-Smirnov Goodness of Fit Test

```
import numpy as np
import matplotlib.pyplot as plt
import pandas as pd
import scipy.stats as st

weight_data=pd.read_csv('/weightdata.txt', header = None, names = ["W"])
weight_data = weight_data["W"]

#Finding CDF
weight_data_sorted = np.sort(weight_data)
weight_data_sorted = weight_data_sorted[~np.isnan (weight_data_sorted)]
weight_data_sorted_normal = (weight_data_sorted- 72)/15
cdf_null_hyp = [st.norm.cdf(weight) for weight in weight_data_sorted_normal]

#Finding the observed test statistic
weight_data = weight_data[-np.isnan(weight_data)]
weight_edf = np.arange(1/len(weight_data),
                        1+1/len(weight_data),
                        1/len(weight_data))

#calculate absolute difference
weight_dif_abs = np.abs(cdf_null_hyp-weight_edf)
#get max different
dn_ks = max(weight_dif_abs)
print(dn_ks)

#Finding critical value of test statistic with alpha = 0.05
dn_crit = 1.36/np. sqrt(len(weight_data))
```

```
print(dn_crit)

# Plotting the ECDF and CDF curves
plt.figure(figsize=(10, 10))
plt.plot(weight_data_sorted_normal,weight_edf,label='F_obs', color='magenta')
plt.plot(weight_data_sorted_normal,cdf_null_hyp,label='F_exp', color='green')
for x, yi, y2 in zip(weight_data_sorted_normal, weight_edf, cdf_null_hyp):
     plt.plot([x, x], [yi, y2], color='cyan',alpha = 0.2)
plt. legend(fontsize = 16)
plt.ylabel("Cumulative Probability")
plt.xlabel('Normalized weight')
plt.title("KS Goodness of Fit")
plt.show()
```

Data Resources Used

- The data for examples to illustrate the chi-squared goodness of fit tests comes from .csv files available on textbook's companion website - https://www.cengage.com/cgi-wadsworth/course_products_wp.pl?fid=M20b&product_isbn_issn=9781305585317&token= - for problems 12.26 (page 538) and 12.22 (page 537).
- The data for KS Test was taken from https://www.kaggle.com/sulianova/eda-cardiovascular-data/data . Only the weight column was used for the test.

References

- Anderson, D. R., Sweeney, D. J., Williams, T. A., Camm, J. D., & Cochran, J. J. (2014). Statistics for Business & Economics (12th ed.). Cengage Learning
- (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
- KOLMOGOROV-SMIRNOV GOODNESS OF FIT TEST https://www.itl.nist.gov/div898/handbook/eda/section3/eda35g.htm
- Akaike information criterion:
- Wikipedia(https://en.wikipedia.org/wiki/Akaike_information_criterion) 15 Oct,2021
- https://www.jmp.com/en_ch/statistics-knowledge-portal/chi-square-test/chi-square-goodness-of-fit-test.html