Joo-ho Kim

Linda Chattin

IEE 380

04. 15. 2025

Hypothesis Test on Mean Compression Time Using Paired t-Test: ZIP vs 7z.

1. Introduction

The widespread use of smartphones has led to a massive increase in photo and image generation. Storing and transferring these images requires significant space and bandwidth, making compression a necessary part of modern digital workflows. Given this overwhelming volume of image data, efficient compression techniques have become increasingly critical.

ZIP and 7z are two common compression formats used for image files. ZIP typically employs the DEFLATE algorithm (Deutsch), whereas 7z uses the LZMA algorithm (Pavlov), which is designed to achieve higher compression ratios. However, processing time may vary depending on file type and system configuration. For companies that process large volumes of image files daily, even a slight difference in compression speed can lead to meaningful time and cost savings. This project aims to compare the compression times of ZIP and 7z to evaluate whether there is a statistically significant difference in performance when compressing image files of similar size.

2. Hypothesis and Theoretical Background

In everyday usage, smartphone users generate many image files that must be stored or transferred efficiently. ZIP and 7z are two widely used compression formats, with ZIP utilizing the DEFLATE algorithm and 7z using the LZMA algorithm. It is commonly assumed that for small files such as daily smartphone images, there is no meaningful difference in compression time between the two methods. However, the LZMA algorithm used in 7z is a dictionary-based compression method that involves

long-range pattern matching, complex probability modeling, and entropy encoding. This structure makes it computationally heavier than the DEFLATE algorithm used in ZIP, which could result in longer compression times—even for relatively small image files.

This study investigates whether ZIP compresses image files significantly faster than 7z. To test this, the following hypotheses are formulated:

a. Null Hypothesis (H_0): There is no difference in mean compression time between ZIP and 7z.

$$H_0: \mu_D = \Delta_0$$

b. Alternative Hypothesis (H_1): ZIP has a significantly lower mean compression time than 7z.

$$H_1: \mu_D < \Delta_0$$

3. Method and Data Collection

This experiment was designed to compare the compression times of ZIP and 7z formats when applied to image files of similar size. A total of 20 JPG image files were selected from a personal photo library. These were photos taken directly by the author using a smartphone, and their file sizes ranged from approximately 1 MB to 2 MB, simulating typical smartphone image data.

The images were stored in a folder named IEE380Project on the desktop. Compression time was measured for each file using both the ZIP format and the 7z format. ZIP compression was performed using zip.exe from GnuWin32 (https://gnuwin32.sourceforge.net/packages/zip.htm), and 7z compression was performed using the official 7z.exe tool from 7-Zip (https://www.7-zip.org) (see Figure 1). Both tools were executed from PowerShell on a Windows machine using the Measure-Command command, which reports the time taken to execute a specific command block (see Figure 2). For each image, two separate timing measurements were taken:

a. Time to compress using ZIP.

b. Time to compress using 7z.

All timing results were recorded in a CSV file using a PowerShell script and later imported into Excel for analysis.

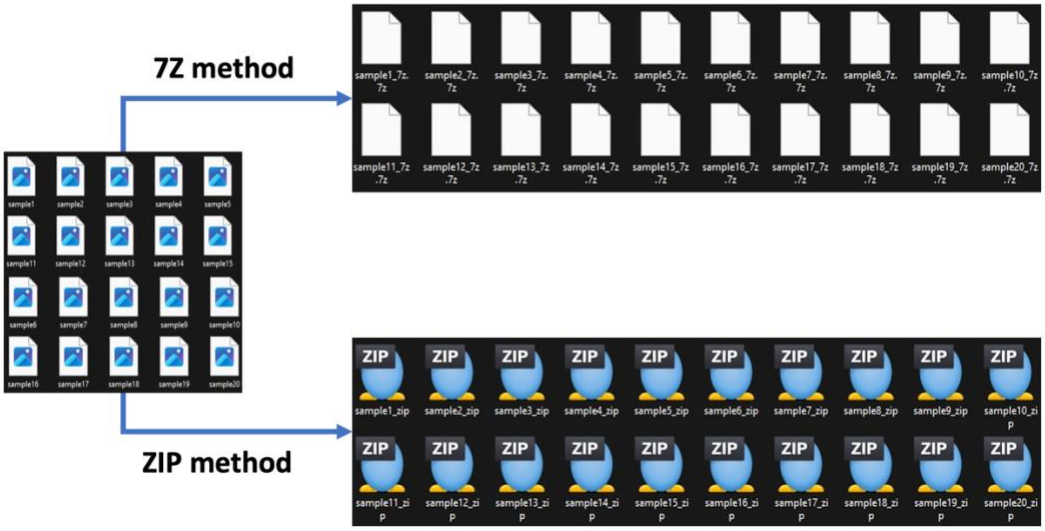

Figure 1. Compression process for 20 image files using 7z (top) and ZIP (bottom), producing .7z and .zip files respectively.

```
PS C:\Users\junok\OneDrive\바탕 화면>  Measure-Command { & "C:\Program Files\7-Zip\7z.exe" a sample1.7z sample1.jpg }


Days              : 0
Hours             : 0
Minutes           : 0
Seconds           : 0
Milliseconds      : 54
Ticks             : 542604
TotalDays         : 6.28013888888889E-07
TotalHours        : 1.50723333333333E-05
TotalMinutes      : 0.00090434
TotalSeconds      : 0.0542604
TotalMilliseconds : 54.2604
```

Figure 2. Sample output from PowerShell using Measure-Command to measure compression time.

4. Statistical Analysis and Confidence Interval

This section presents the statistical analysis performed on the observed differences in compression time between the ZIP and 7z formats. A total of 20 paired observations were collected, each consisting of the compression time (in seconds) using both methods on the same image file.

File	Zip Time	7z Time
sample1.jpg	0.0546	0.0715
sample2.jpg	0.0437	0.0618
sample3.jpg	0.0397	0.0521
sample4.jpg	0.0473	0.0587
sample5.jpg	0.0503	0.0702
sample6.jpg	0.0532	0.0703
sample7.jpg	0.0418	0.0594
sample8.jpg	0.0410	0.0625
sample9.jpg	0.0485	0.0696
sample10.jpg	0.0360	0.0573
sample11.jpg	0.0343	0.0497
sample12.jpg	0.0368	0.0593
sample13.jpg	0.0383	0.0570
sample14.jpg	0.0501	0.0681
sample15.jpg	0.0402	0.0572
sample16.jpg	0.0412	0.0619
sample17.jpg	0.0557	0.0753
sample18.jpg	0.0413	0.0546
sample19.jpg	0.0445	0.0673
sample20.jpg	0.0530	0.0818

*Table 1. Compression time (in **seconds**) for each image file.*

Because the research question focuses on whether ZIP compresses image files faster than 7z (i.e., whether ZIP's average compression time is less than 7z's), a 95% one-sided confidence interval is more appropriate than a two-sided interval. This approach ensures that if the entire interval lies below zero, we have strong evidence that ZIP is faster (Montgomery & Runger). A 95% confidence level was chosen to maintain a conventional balance between Type I and Type II error rates (Fisher).

To estimate the mean difference in compression time, a 95% one-sided confidence interval was constructed using the formula:

a. One-Sided Lower Confidence Interval:

$$\hat{L} = \bar{d} - t_{\alpha,n-1}\frac{S_D}{\sqrt{n}}$$

b. One-Sided Upper Confidence Interval:

$$\hat{U} = \bar{d} + t_{\alpha,n-1}\frac{S_D}{\sqrt{n}}$$

The following table summarizes the symbolic equations and key steps used in constructing the 95% one-sided confidence interval for the mean difference in compression time between ZIP and 7z. It outlines the population parameter, hypotheses, test statistic formula, significance level, computed values, and the resulting confidence interval bounds, all using the notation and protocols prescribed in our class.

Step	Description
1. Population parameter of interest	μ_D
2. Null Hypothesis (H_0)	$H_0: \mu_D = \Delta_0$
3. Alternative Hypothesis (H_1)	$H_1: \mu_D < \Delta_0$
4. Significance level	$\alpha = 0.05$
5. Test statistic	$t_0 = \frac{\bar{d}-\Delta_0}{\frac{S_D}{\sqrt{n}}}$ $\bar{d} = \frac{\sum_{j=1}^{n} d_j}{n}$ $S_D = \sqrt{\frac{\sum_{j=1}^{n}(d_j - \bar{d})^2}{n-1}}$
6. Reject H_0 if	p-value $< \alpha = 0.05$
7. Computations	$\bar{d} = -0.019$ $S_D = 0.004$ $n = 20$ $df = 19$ $t_0 = -21.243$ $p - value = 5.0 * 10^{-15}$
8. Interval	$\hat{L} = \bar{d} - t_{\alpha,n-1}\frac{S_D}{\sqrt{n}}$= -0.021 $\hat{U} = \bar{d} + t_{\alpha,n-1}\frac{S_D}{\sqrt{n}}$= -0.018

Table 2. Symbolic Equation for CI using the equation editor

As the raw measurements were recorded to four decimal places, all summary statistics reported below have been rounded to three decimal places for clarity and consistency (F. Habibzadeh & P. Habibzadeh).

The values of $\bar{d}$ = -0.019, S_D = 0.004, and the t_0 = -21.243 were calculated using a TI-89 calculator. The standard deviation and t-score were computed using the built-in *1-Var Stats* and *Inverse-t* functions. The confidence interval bounds were computed manually using the formula for a one-sample paired t-interval. The p-value of 5.00×10^{-15} was obtained from the *t-cdf* function with input bounds [-10^{99}, -21.243] and degrees of freedom = 19.

The following figure shows the t-distribution for 19 degrees of freedom used in our one-sided hypothesis test. The graph illustrates the rejection region (at $\alpha = 0.05$) shaded in red, while the observed test statistic ($t_0 = -21.243$) is indicated by a red line. The annotated p-value (approximately 5.00×10^{-15}) reinforces the decision to reject the null hypothesis.

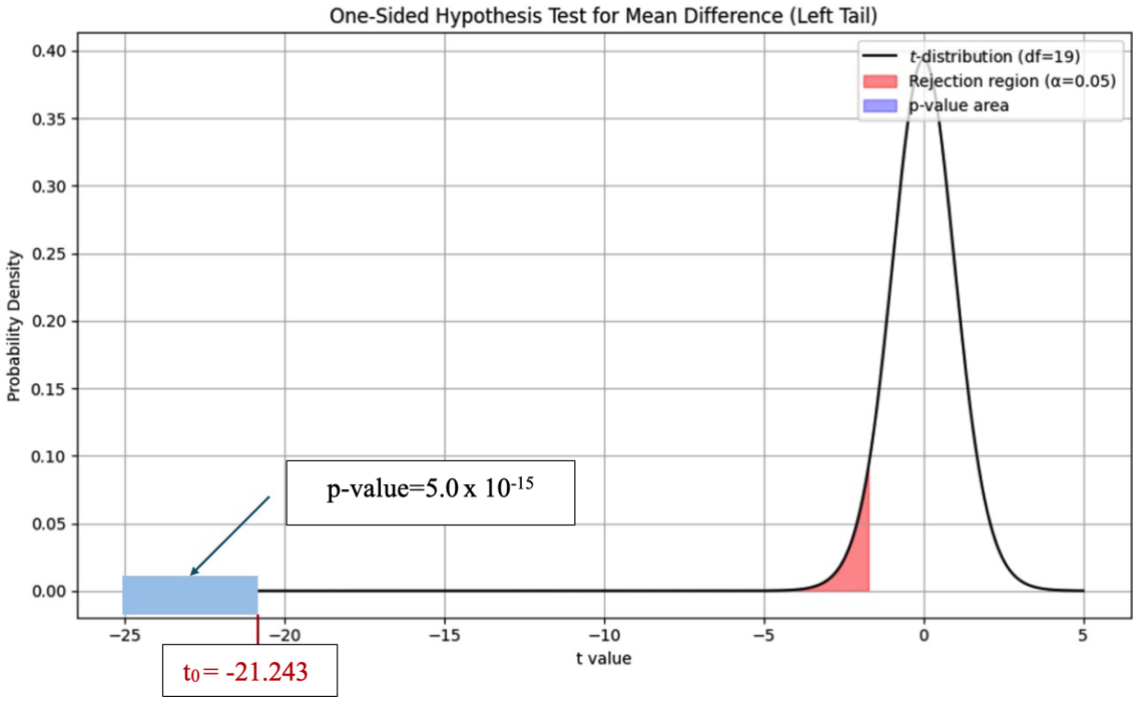

Figure 3. One-Sided t-Distribution Plot by Python

5. Inference and Conclusion

The paired t-test conducted on 20 observations revealed a test statistic of $t_0 = -21.243$ with a corresponding one-tailed p-value of approximately 5.00×10^{-15}. Additionally, the 95% one-sided confidence interval for the mean difference in compression time (ZIP – 7z) was calculated as [−0.021, −0.018]. Since the entire confidence interval lies below zero, these results provide overwhelming statistical evidence to reject the null hypothesis that there is no difference. Consequently, we conclude that ZIP compresses image files significantly faster than 7z.

These findings suggest that, for the range of image files tested, the simpler DEFLATE algorithm employed by ZIP offers superior compression speed compared to the more computationally intensive LZMA algorithm used by 7z. In practical applications—such as systems that process a large volume of image files daily—this speed advantage could translate into significant time and cost savings.

Future studies might extend this analysis to other file types or larger datasets to further explore the trade-offs between compression speed and efficiency. However, based on the current experiment, the faster compression provided by ZIP is clearly advantageous in scenarios where processing time is critical.

References

Pavlov, I. (2001, December 5). 7z format. 7-Zip Documentation. Retrieved April 15, 2025, from https://www.7-zip.org/7z.html

Deutsch, P. (1996, May). *DEFLATE Compressed Data Format Specification version 1.3* (RFC 1951). Internet Engineering Task Force. https://datatracker.ietf.org/doc/html/rfc1951

Montgomery, D. C., & Runger, G. C. (2018). *Applied Statistics and Probability for Engineers (7th ed.).* John Wiley & Sons.

Fisher, R. A. (1925). *Statistical Methods for Research Workers*. Oliver and Boyd.

Habibzadeh F, Habibzadeh P. *How much precision in reporting statistics is enough?* Croat Med J. 2015 Oct;56(5):490–492. doi: 10.3325/cmj.2015.56.490. Retrieved from https://pmc.ncbi.nlm.nih.gov/articles/PMC4679338/