

# Compression and Decompression Benchmark by graysky

Document version 1.0  
18-Mar-2012

# Purpose and Scope

The purpose of this work was to take a smattering of Arch Linux packages and compare compression/decompression times using the current Arch standard of .tar.xz to .tar.lrz; as well as times to compress/decompress, compare the respective archive file sizes to each other.

- Is it value-added to switch from .tar.xz to .tar.lrz?
- Will there be a savings in archive size and thus transfer sizes to end-users from mirrors?
- Is there an advantage to compression times?

# Details and Method

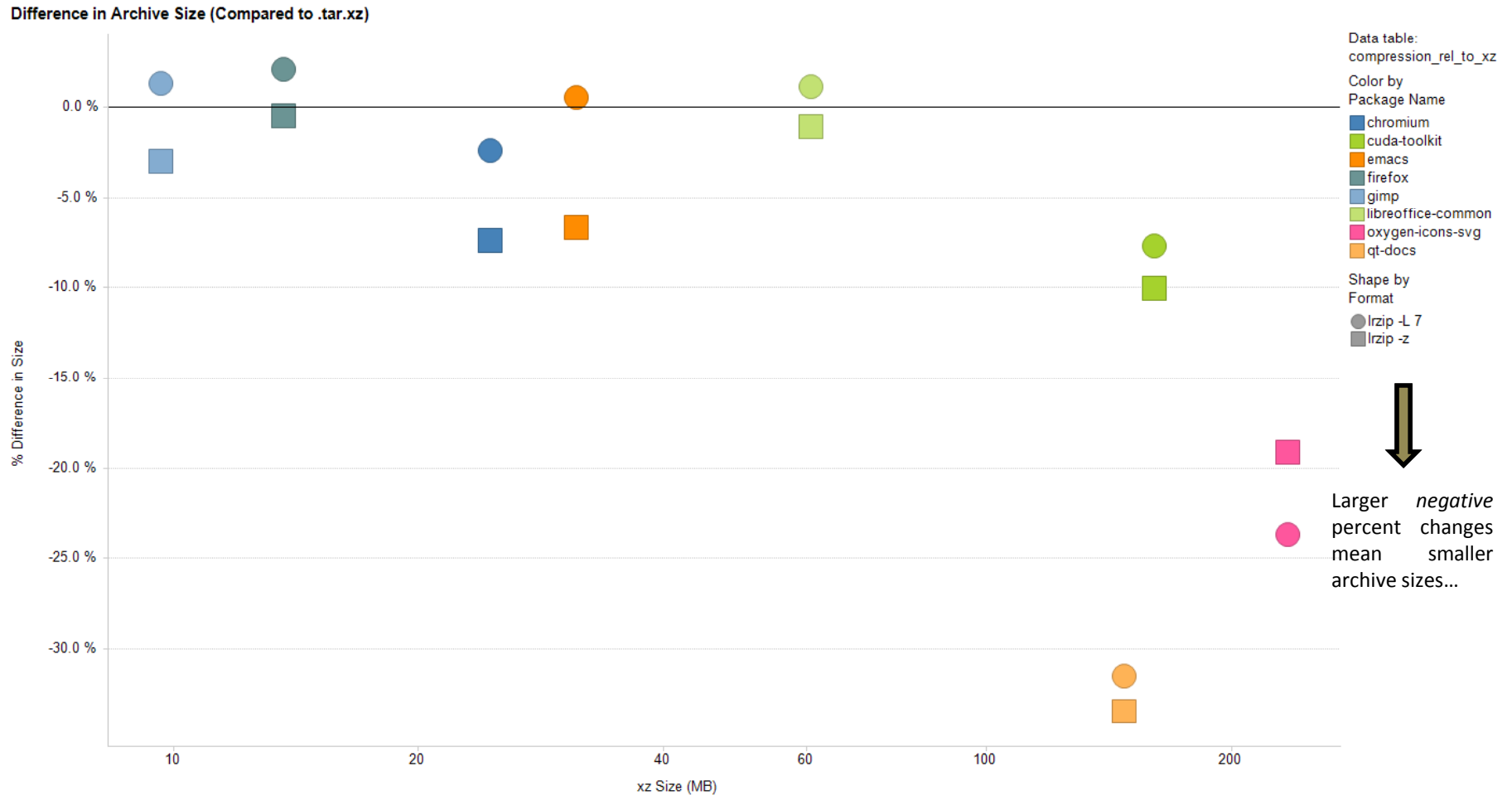
- Source files for benchmark were taken from the official repos and extracted to their own directory for compression benchmarking.
- Packages include (taken from x86\_64):
  - **chromium-17.0.963.79-1**
  - **gimp-2.6.12-1**
  - **cuda-toolkit-4.1.28-2**
  - **Libreoffice-common-3.5.1-1**
  - **emacs-23.4-1**
  - **oxygen-icons-svg-4.8.1-1**
  - **firefox-11.0-2**
  - **qt-doc-4.8.0-1**
- Benchmark was running the below algorithms on each of these dirtrees a total of 3 times recording both the time to compress, and the resulting archive size. Additionally, the times to decompress were also measured 3 times totally. Reported results are the average of the 3 respective runs; error bars are shown when appropriate.
- Formats/algorithms compared:
  - **.tar.xz** via tar Jcf. This is the current Arch Linux standard package format.
  - **.tar.lrz** using the -L 7 switch (lrzip default).
  - **.tar.lrz** using the -z switch (ZPAQ compression to give the best, extreme compression).

- Test machine: quadcore Xeon (X3360) running @ 8.5x400 MHz = 3.40 GHz
- Lrzip v0.612 and tar v1.26 were used.

# All Packages Normalized to .tar.xz Metrics

- The horizontal line is the normalized .tar.xz metrics (by definition=0).
- Each color corresponds to one of the aforementioned packages.
- Circles are lrzip with default settings (-L 7).
- Squares are lrzip using the -z compression flag (ZPAQ compression).

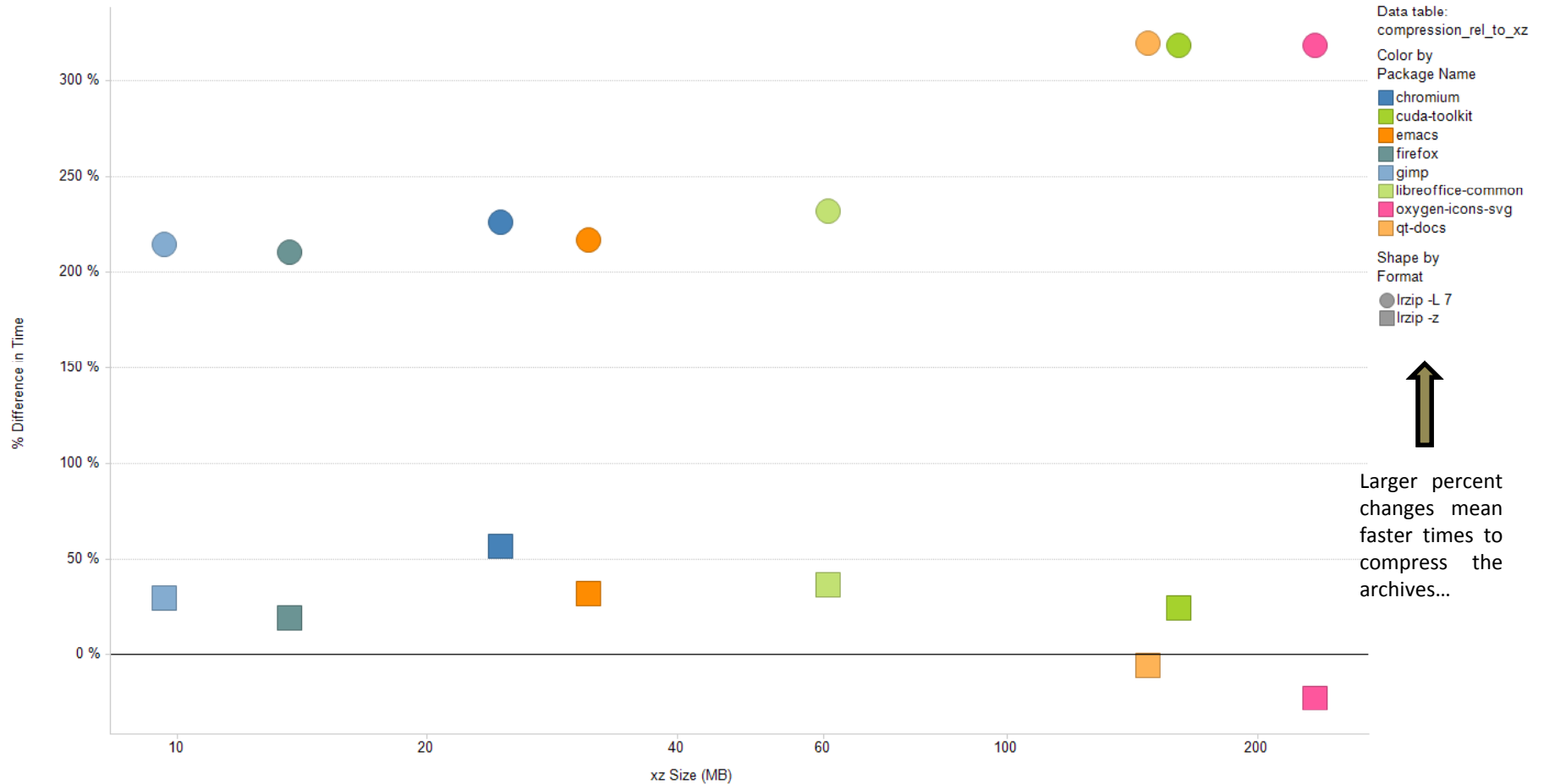
# Normalized Archive Sizes



Plot shows % change in lrzip archive size normalized to xz archive size.

# Normalized Compression Rates

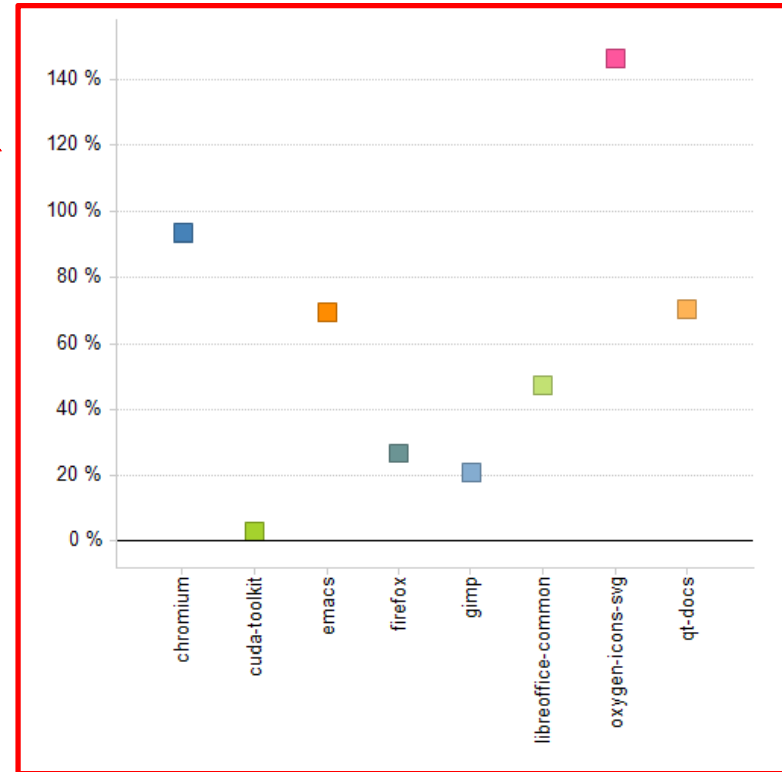
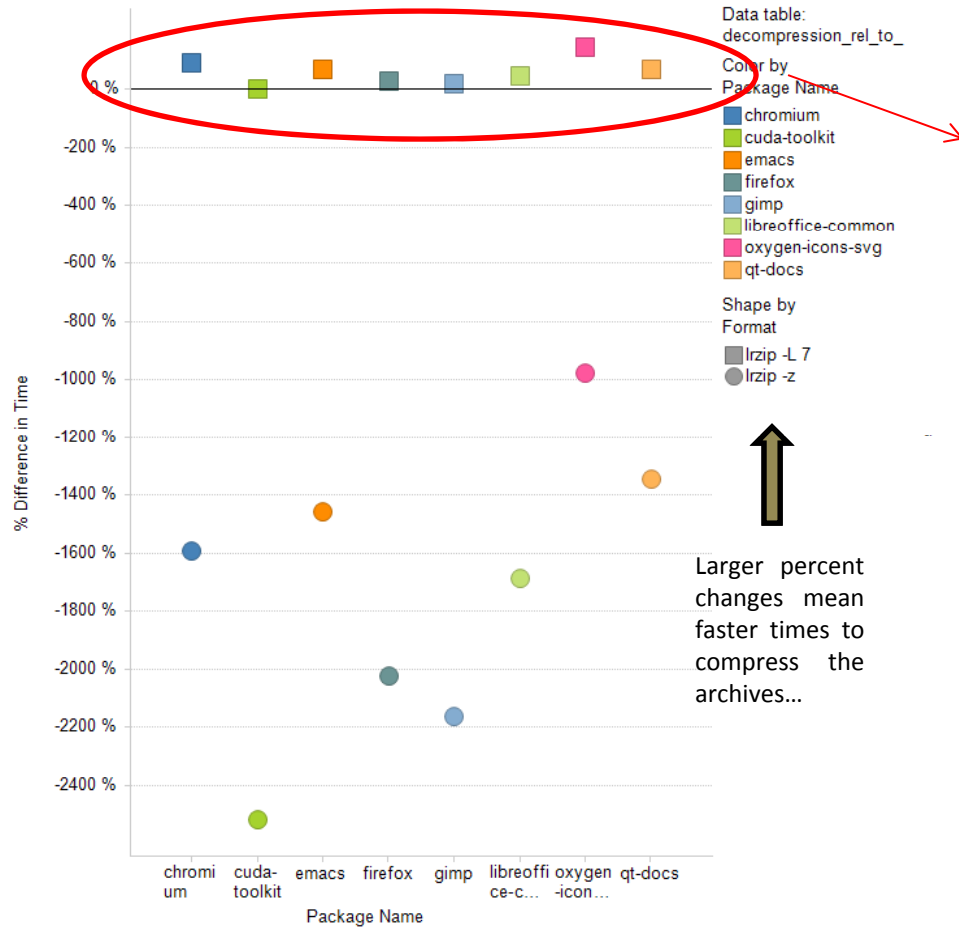
Compression Time Normalized to xz



Plot shows % change in lrzip compression rates normalized to xz compression rates.

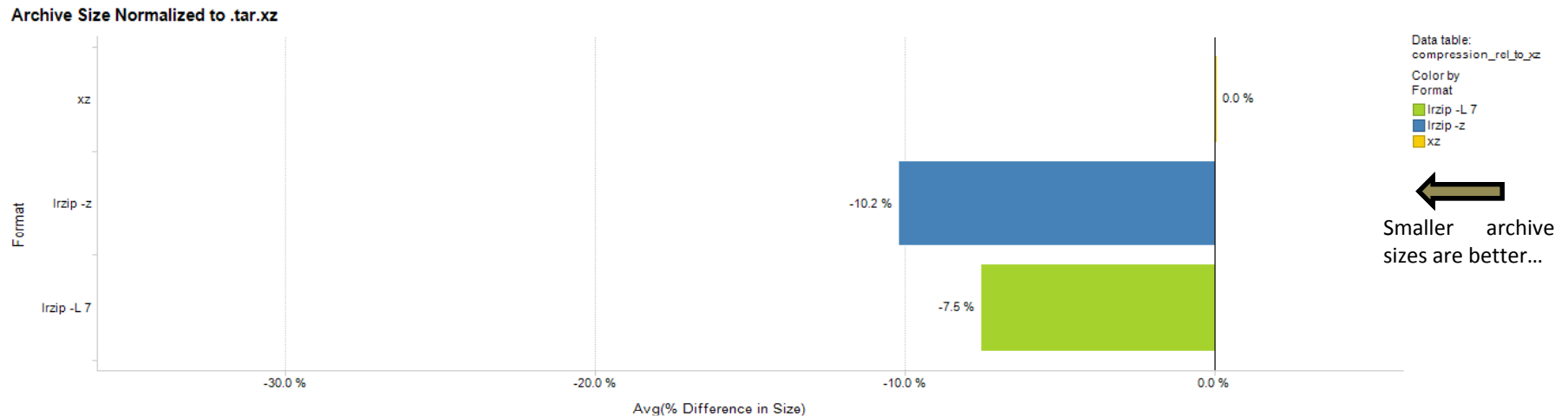
# Normalized Decompression Rates

Decompression Time Normalized to xz



- Plot shows % change in lrzip decompression rates normalized to xz decompression rates.
- Red circle is a zoomed in view of just the lrzip -L 7 data since the lrzip -z decompression times are extremely slow and compress the y-axis!

# Summary of Archive Sizes



- Lrzip gave decent improvements in archive size when normalized to .tar.xz files.

Standard settings:

- Worst case: **2 % increase**
- Best case: **32 % decrease**
- **Average of all cases: 8 % decrease**

Extreme settings:

- Worst case: **1 % decrease**
- Best case: **34 % decrease**
- **Average of all cases: 10 % decrease**



# Summary of Compression Times

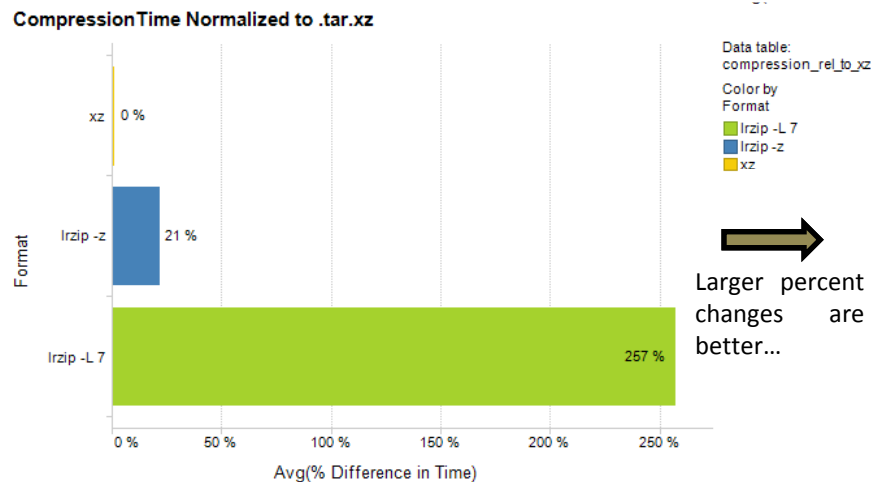
- Lrzip gave marked improvements in compression and decompression times when normalized to .tar.xz files using both standard and extreme settings.

Standard settings:

- Worst case: 210 % decrease
- Best case: 319 % decrease
- **Average of all cases: 267 % decrease**

Extreme settings:

- Worst case: 23 % *increase*
- Best case: 56 % decrease
- **Average of all cases: 21 % decrease**



# Summary of Decompression Times

- Lrzip gave marked improvements in compression and decompression rates when normalized to .tar.xz files using the standard settings, but not the extreme settings.

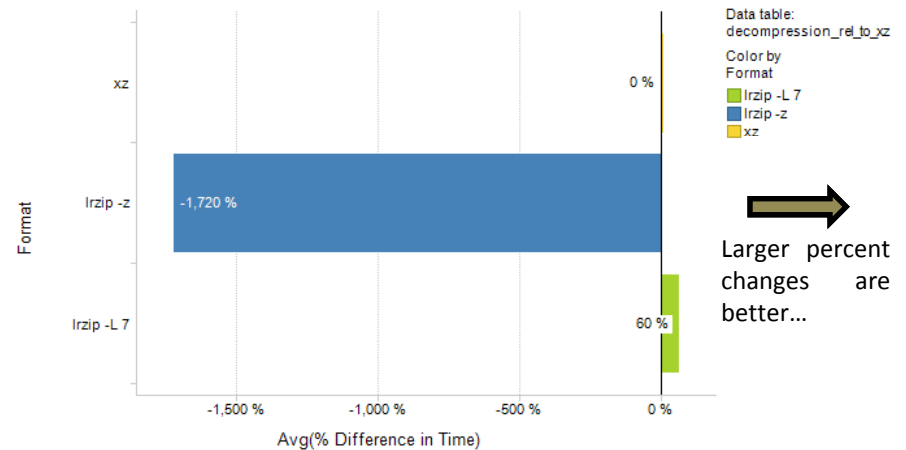
## Standard settings:

- Worst case: **3 % decrease**
- Best case: **146 % decrease**
- **Average of all: 60 % decrease**

## Extreme settings:

- Worst case: **976 % increase**
- Best case: **2,518 % increase**
- **Average of all: 1,720 % increase**

DecompressionTime Normalized to .tar.xz



# Conclusion and Recommendations

Switching from `.tar.xz` to `.tar.lrz` using the default settings within `lrzip (-L 7)` is a value-added proposition in two critical areas:

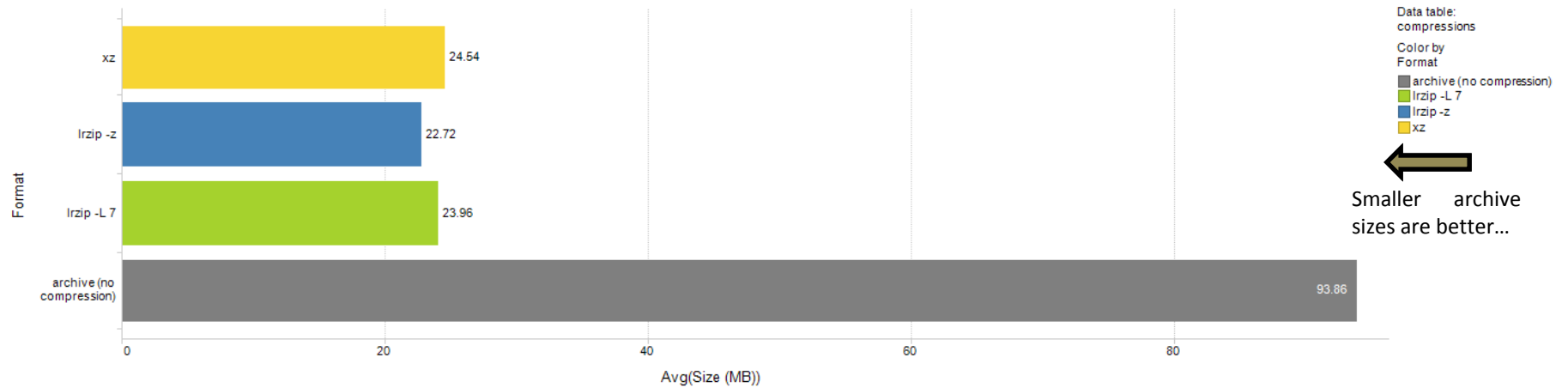
- On the whole, resulting archive sizes are smaller which has positive implications to both storage space on mirrors but more importantly to transfer sizes to end-users.
- On the whole, compression/decompression rates are faster for `lrzip`.

# Appendix

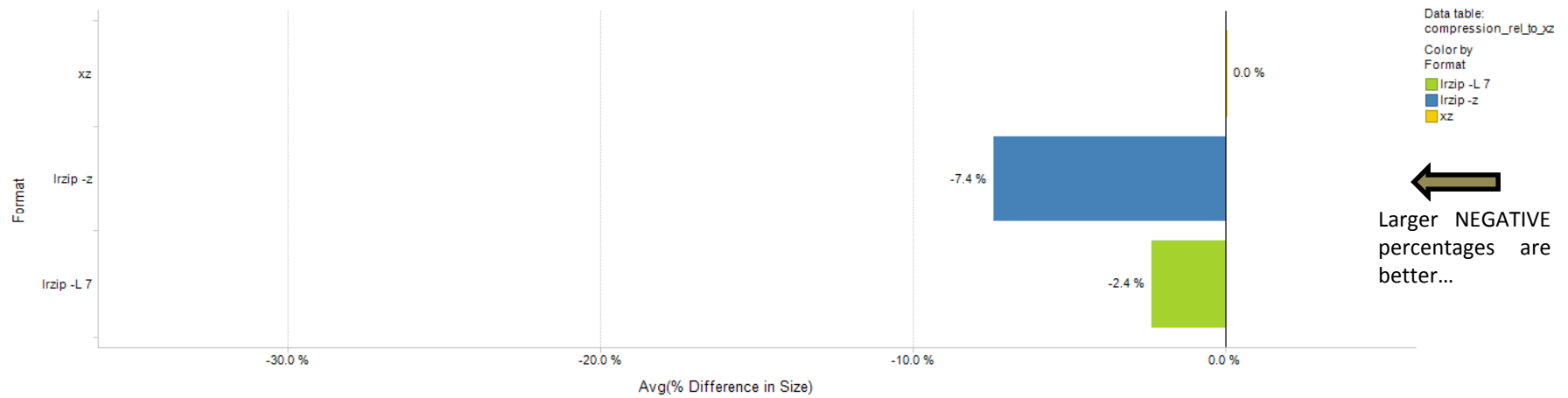
# Archive Size Analysis on a Package-by- Package Basis

# chromium

Compressed File Size (MB)

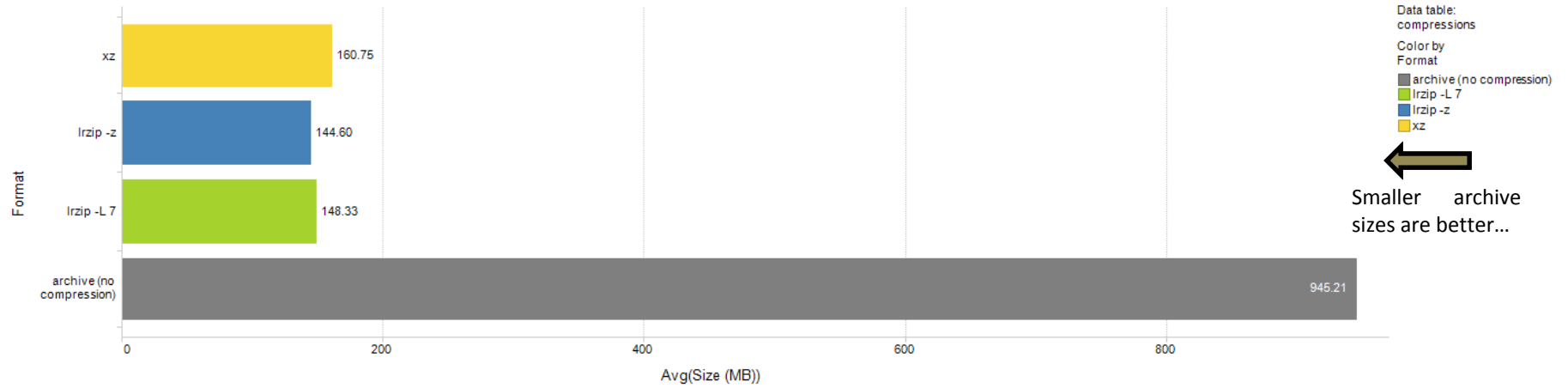


Archive Size Normalized to .tar.xz

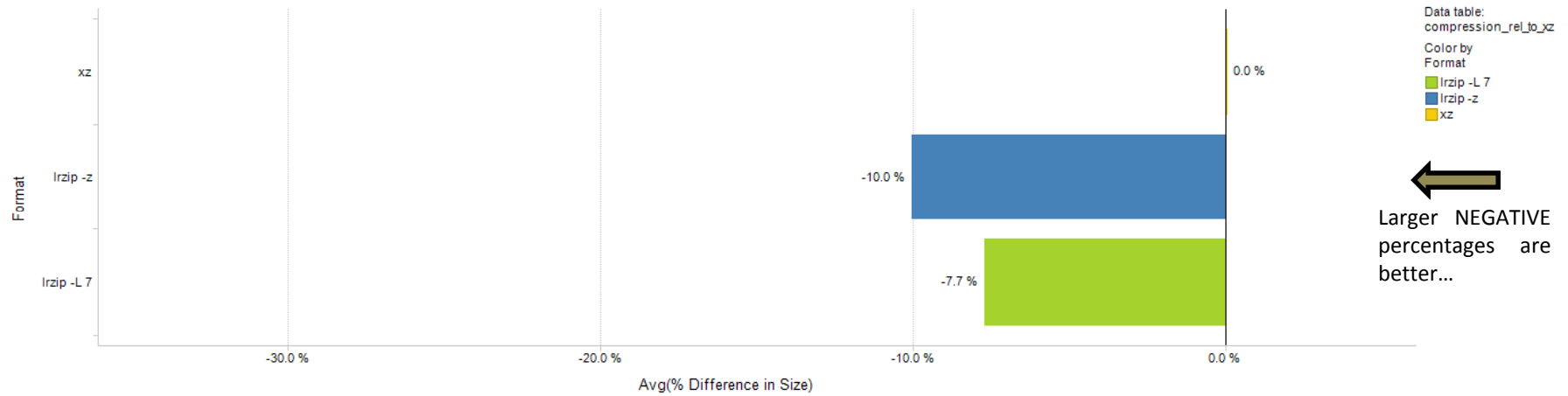


# cuda-toolkit

Compressed File Size (MB)

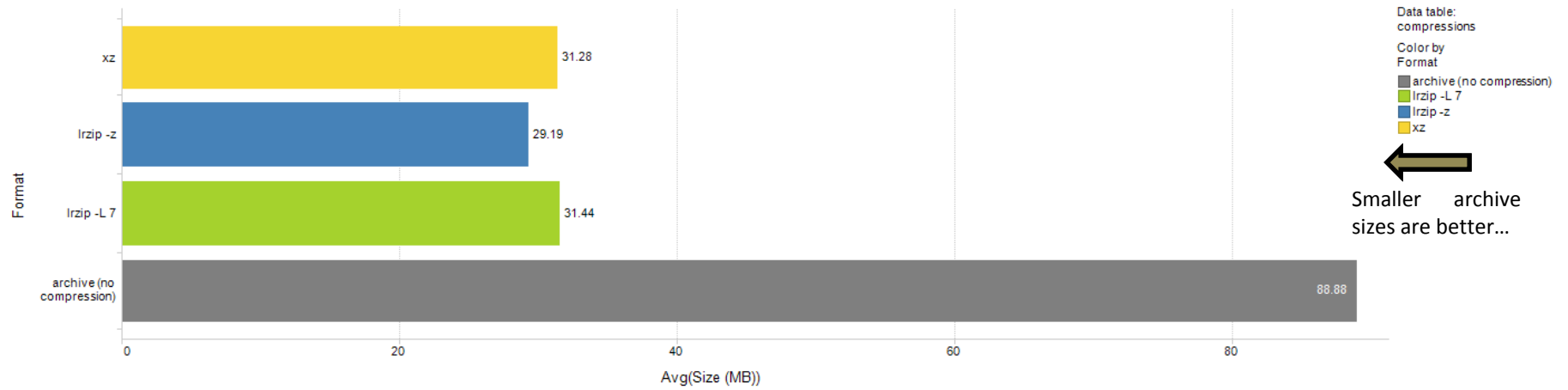


Archive Size Normalized to .tar.xz

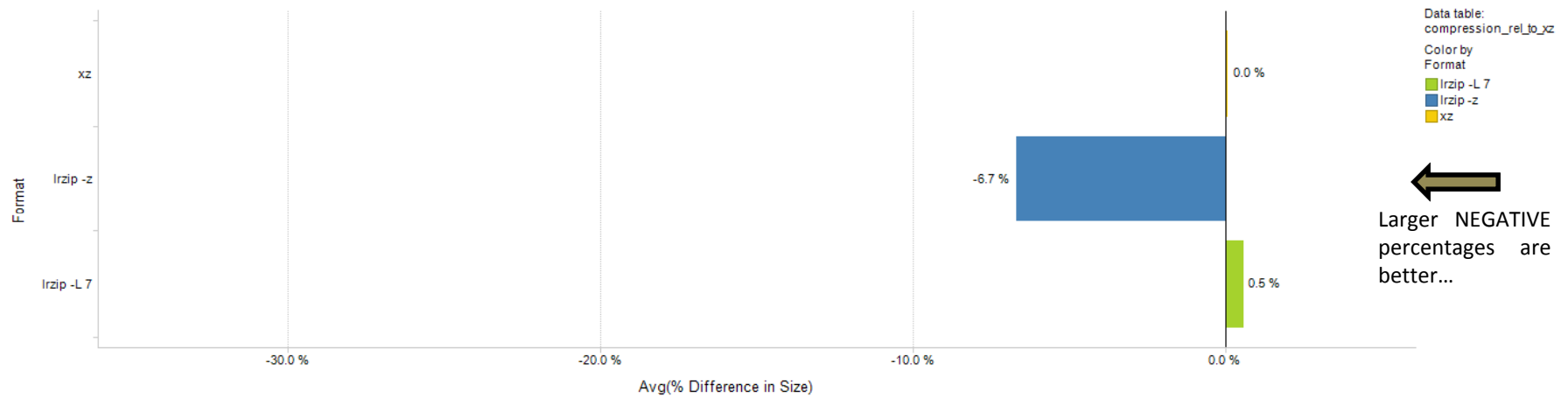


# emacs

Compressed File Size (MB)



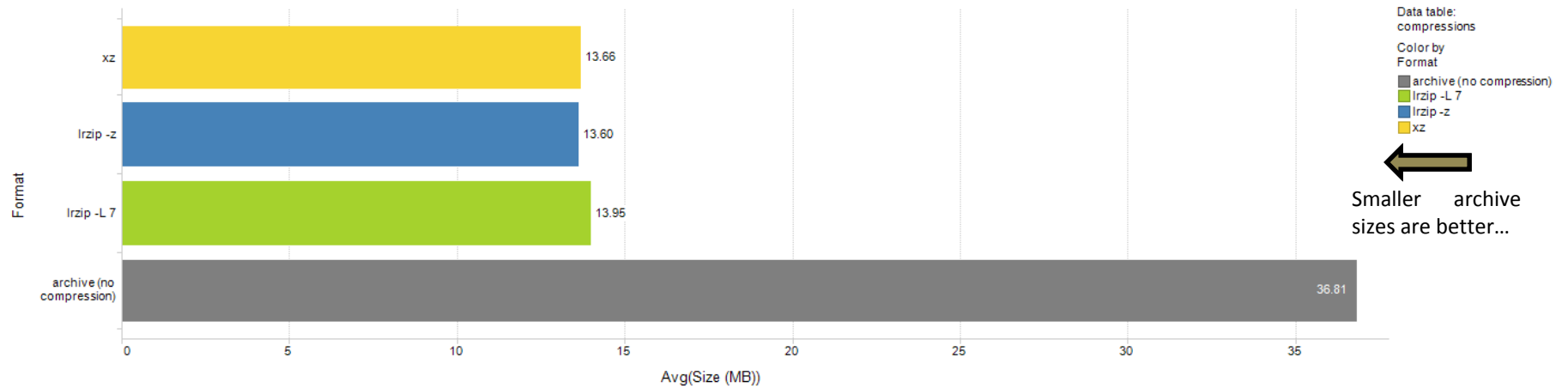
Archive Size Normalized to .tar.xz



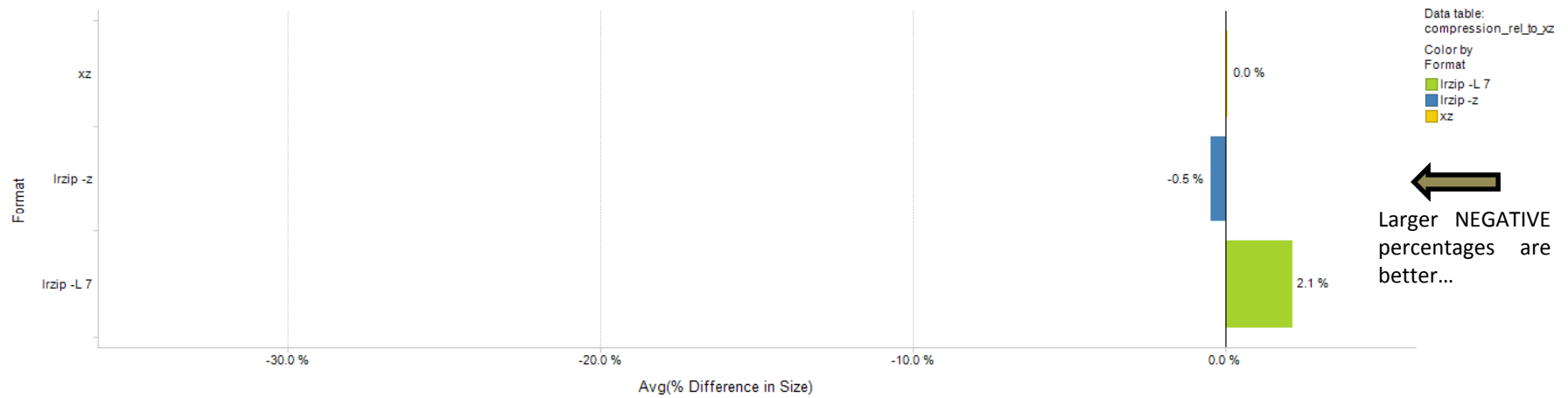


# firefox

Compressed File Size (MB)

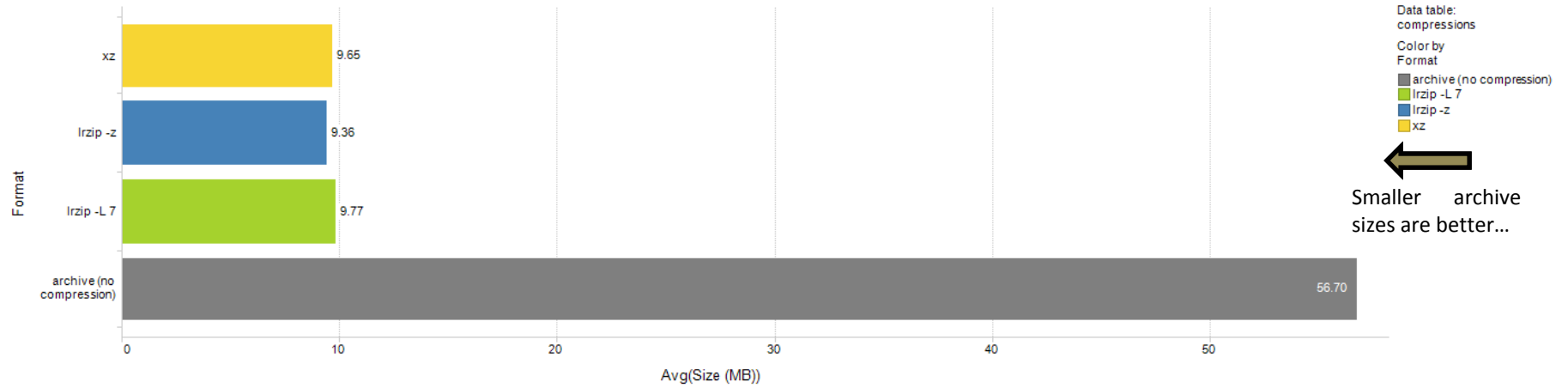


Archive Size Normalized to .tar.xz

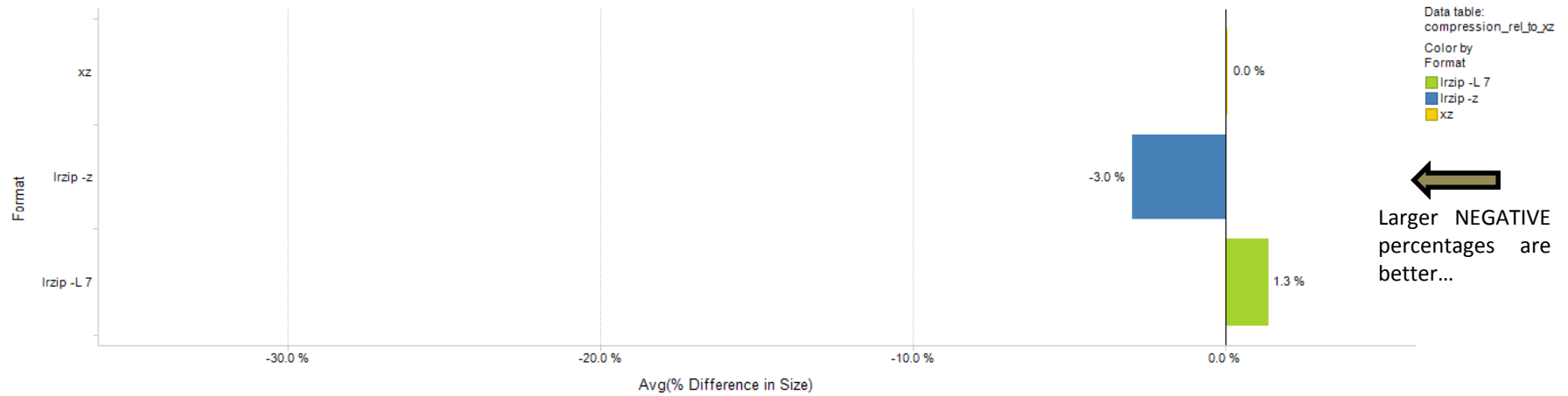


# gimp

Compressed File Size (MB)

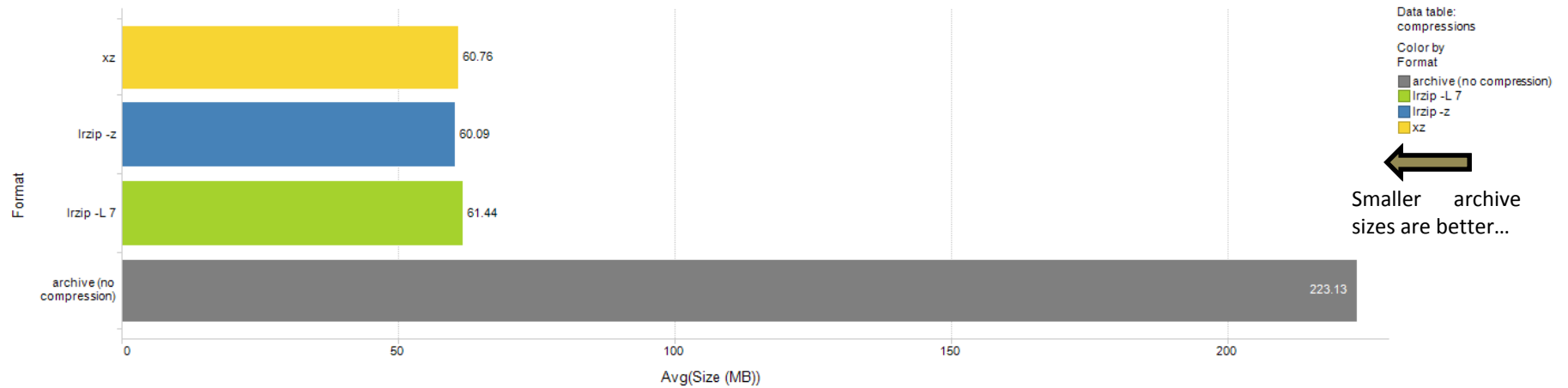


Archive Size Normalized to .tar.xz

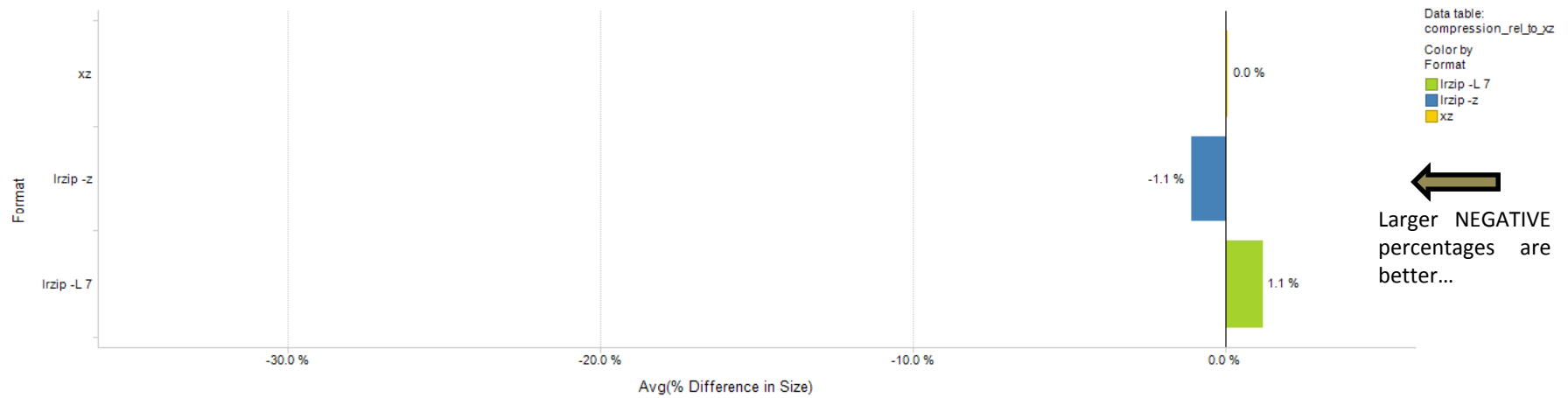


# libreoffice-common

Compressed File Size (MB)

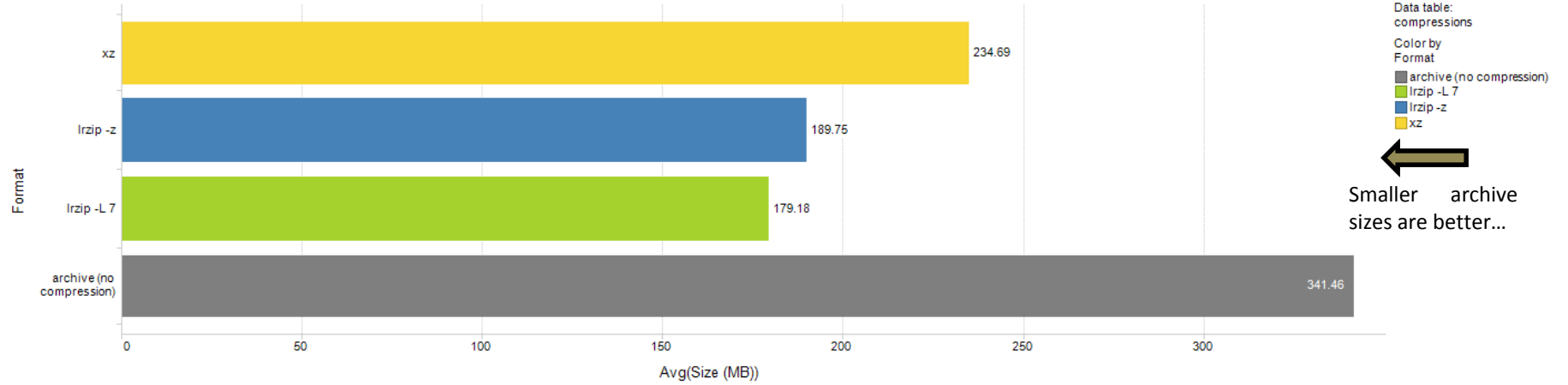


Archive Size Normalized to .tar.xz

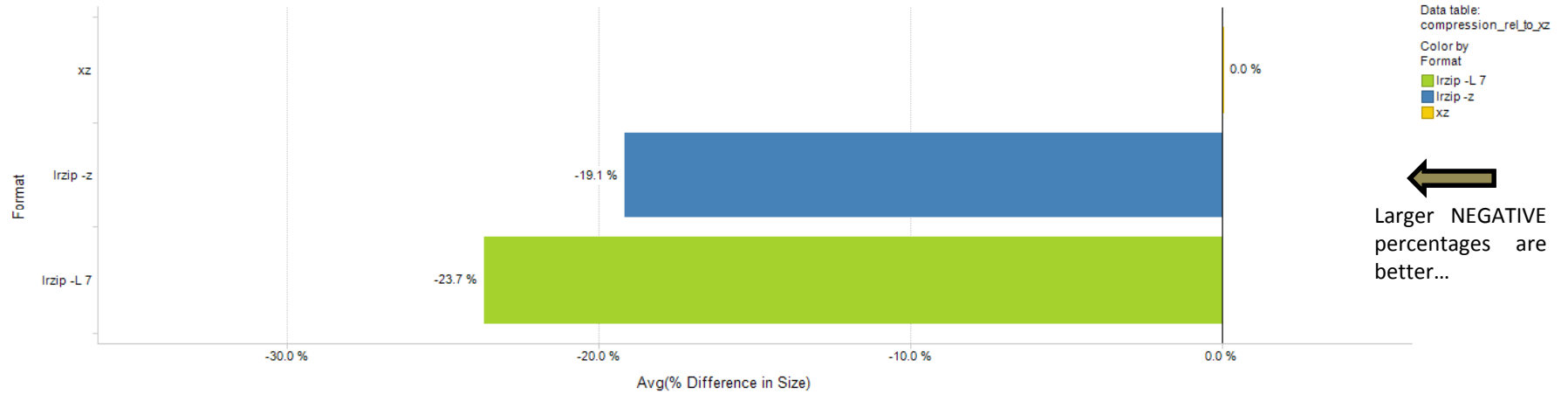


# oxygen-icons-svg

Compressed File Size (MB)

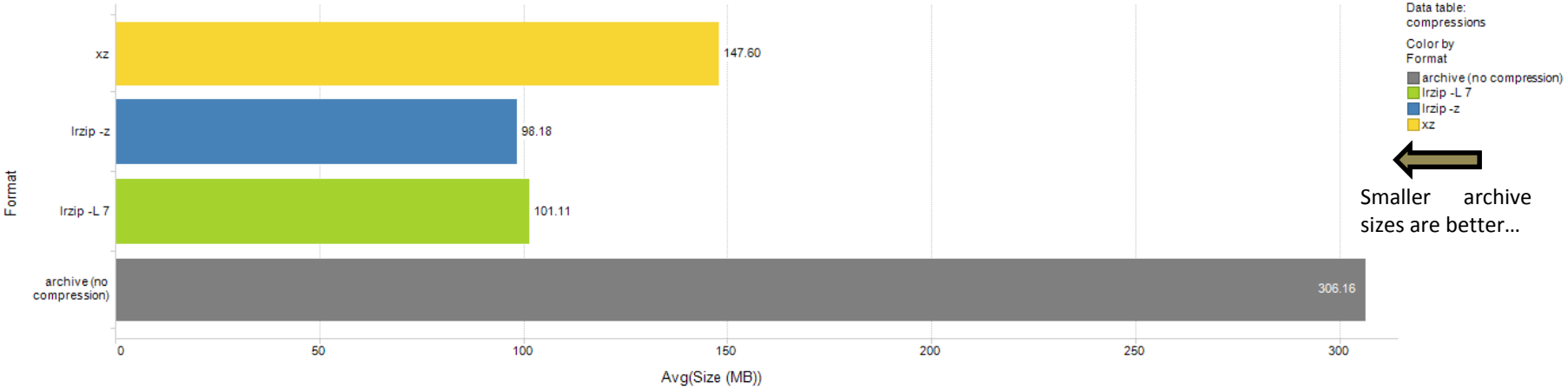


Archive Size Normalized to .tar.xz

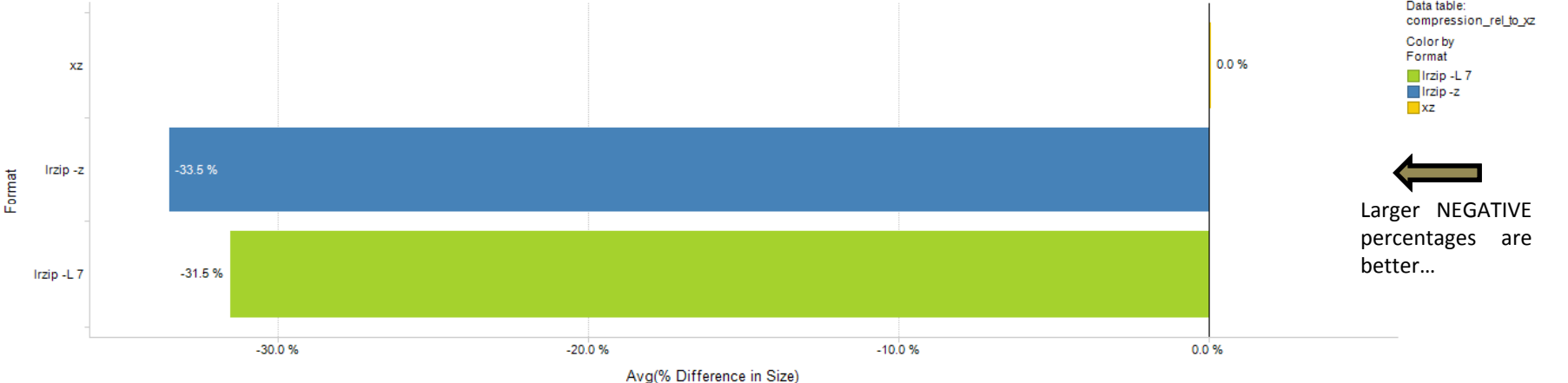


# qt-docs

Compressed File Size (MB)



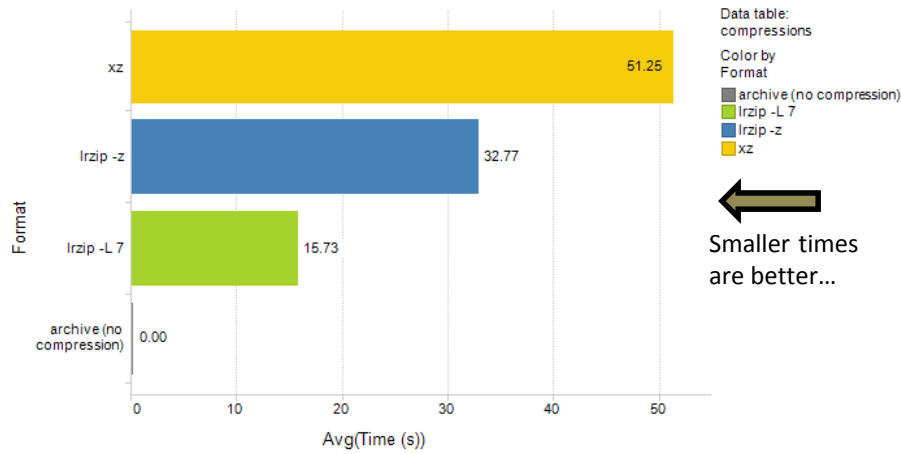
Archive Size Normalized to .tar.xz



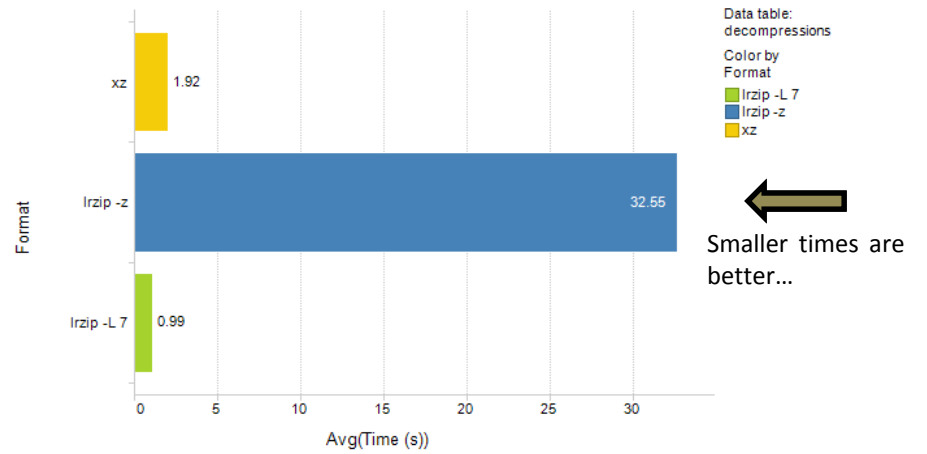
# Compression/Decompression Analysis on a Package-by-Package Basis

# chromium

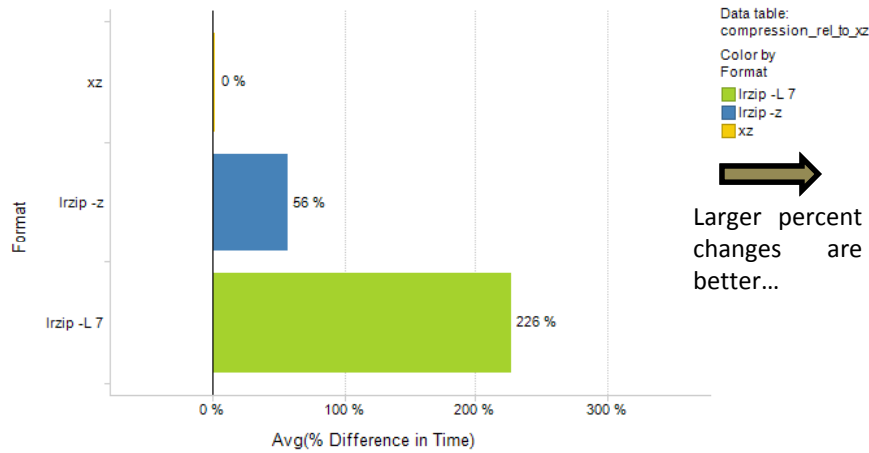
Compression Time (Sec)



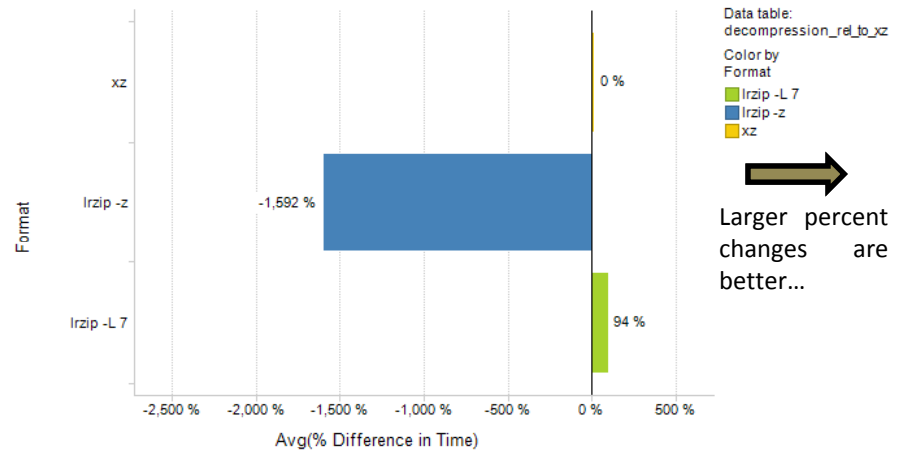
Decompression Time (Sec)



CompressionTime Normalized to .tar.xz

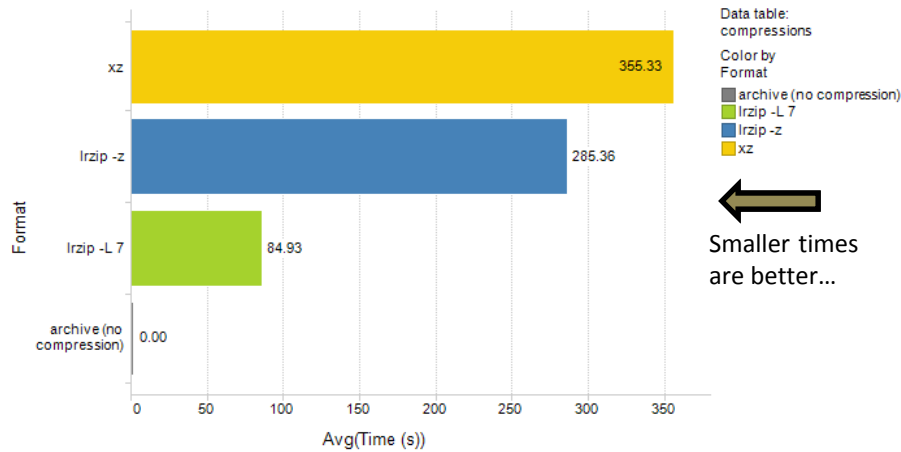


DecompressionTime Normalized to .tar.xz

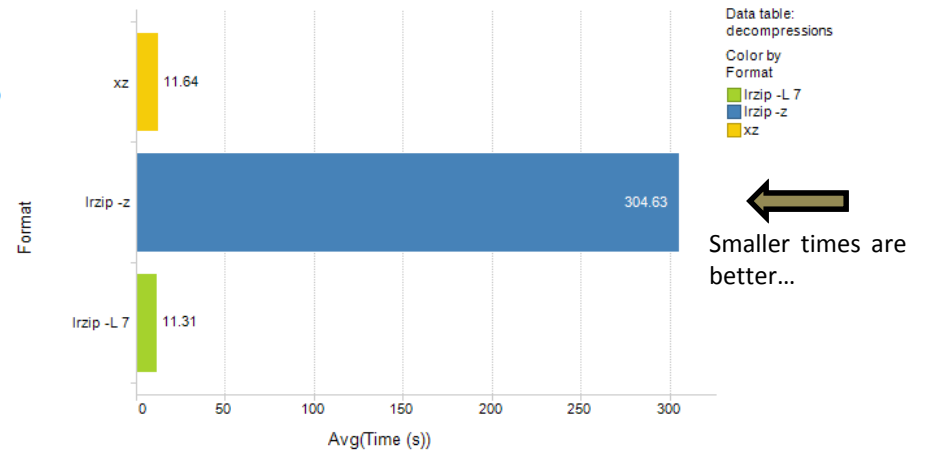


# cuda-toolkit

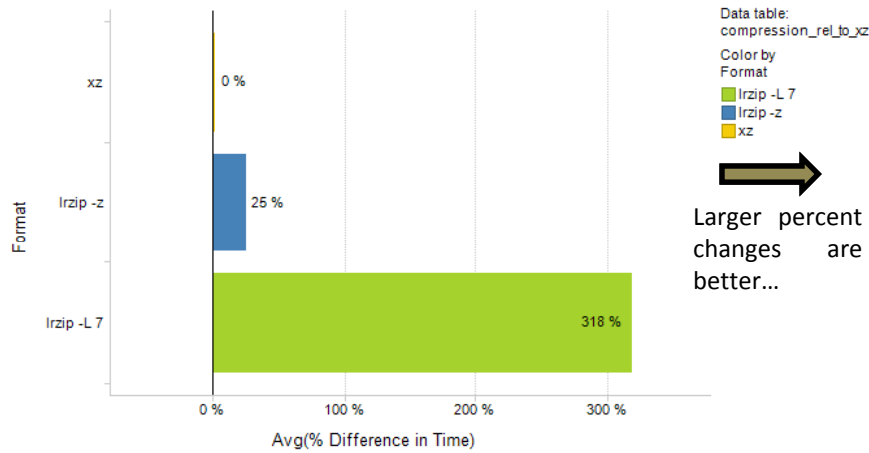
Compression Time (Sec)



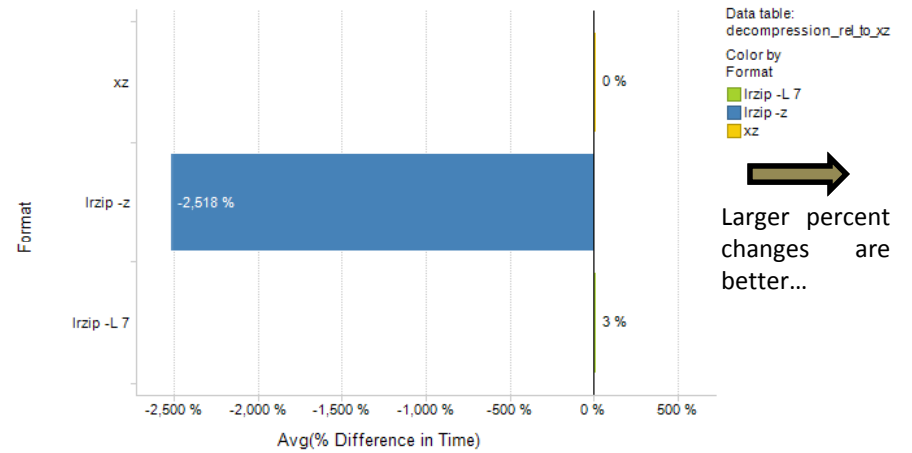
Decompression Time (Sec)



CompressionTime Normalized to .tar.xz



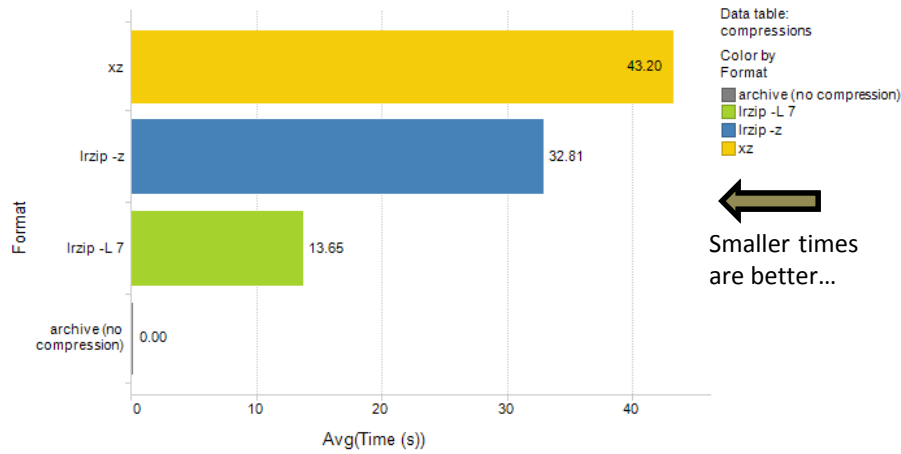
DecompressionTime Normalized to .tar.xz





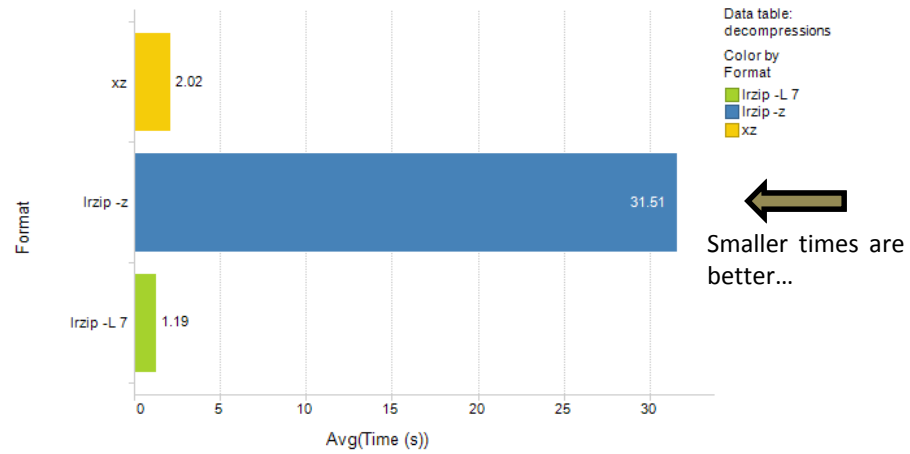
# emacs

Compression Time (Sec)



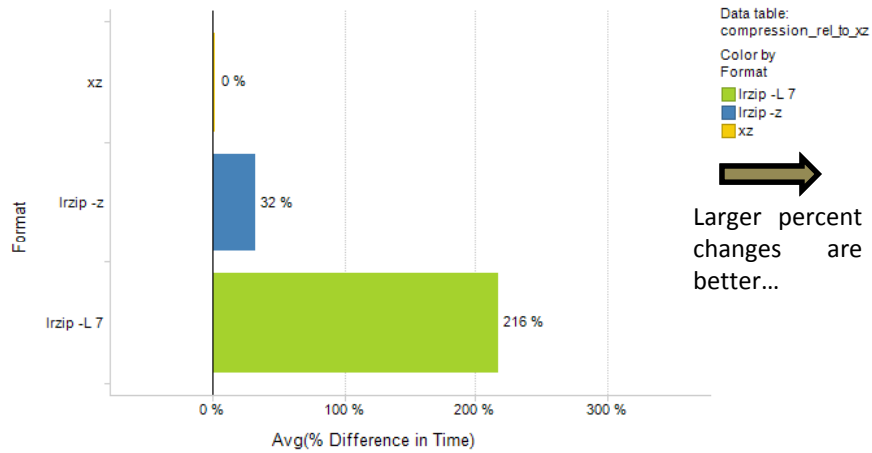
←  
Smaller times are better...

Decompression Time (Sec)



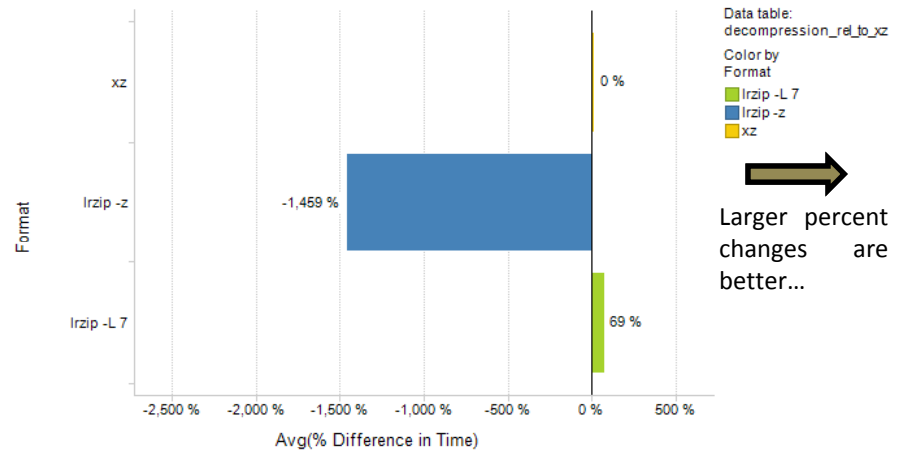
←  
Smaller times are better...

CompressionTime Normalized to .tar.xz



→  
Larger percent changes are better...

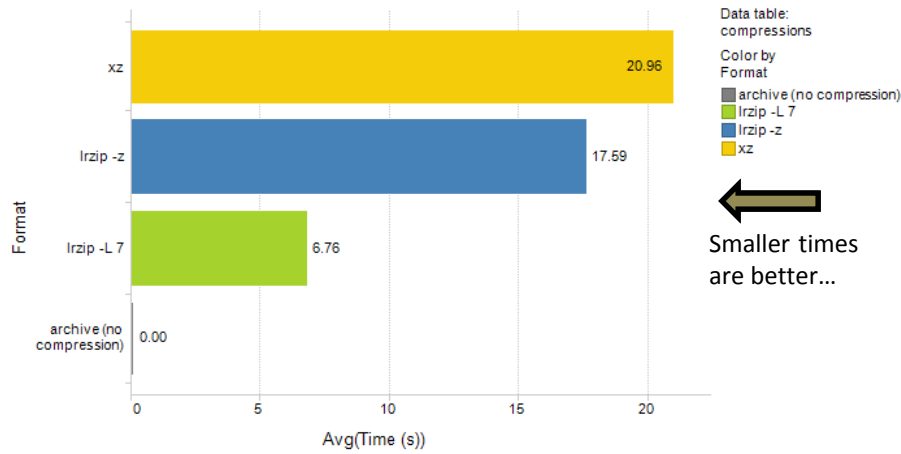
DecompressionTime Normalized to .tar.xz



→  
Larger percent changes are better...

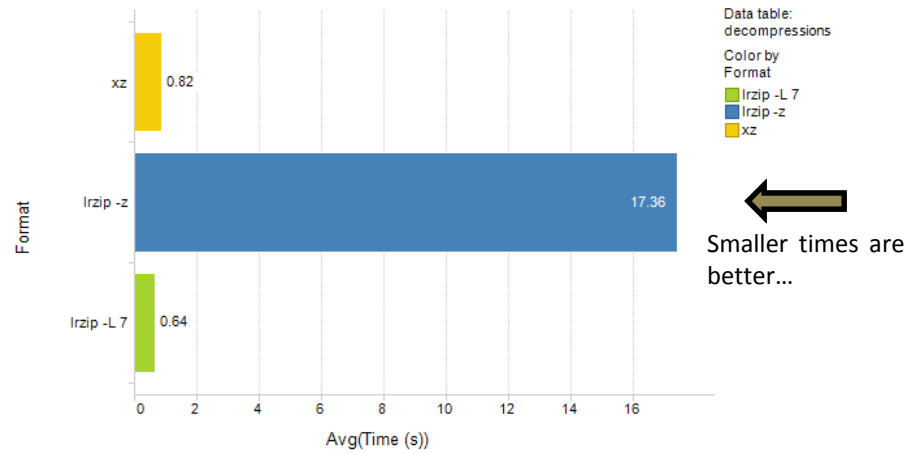
# firefox

Compression Time (Sec)



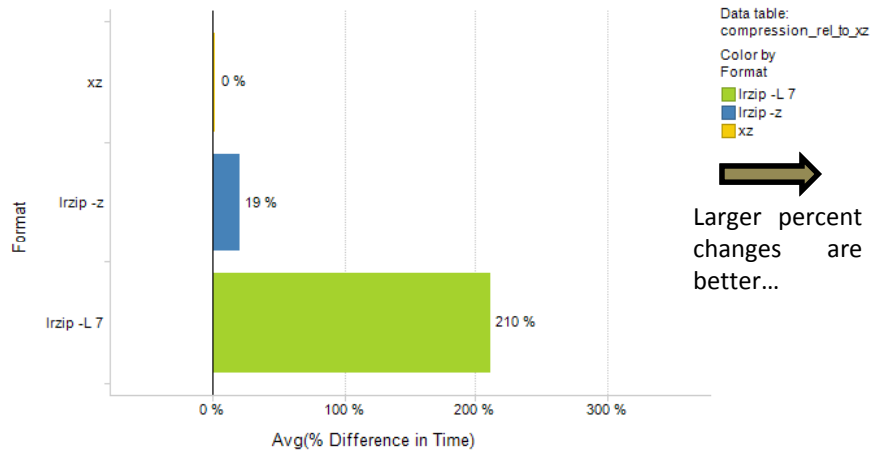
←  
Smaller times are better...

Decompression Time (Sec)



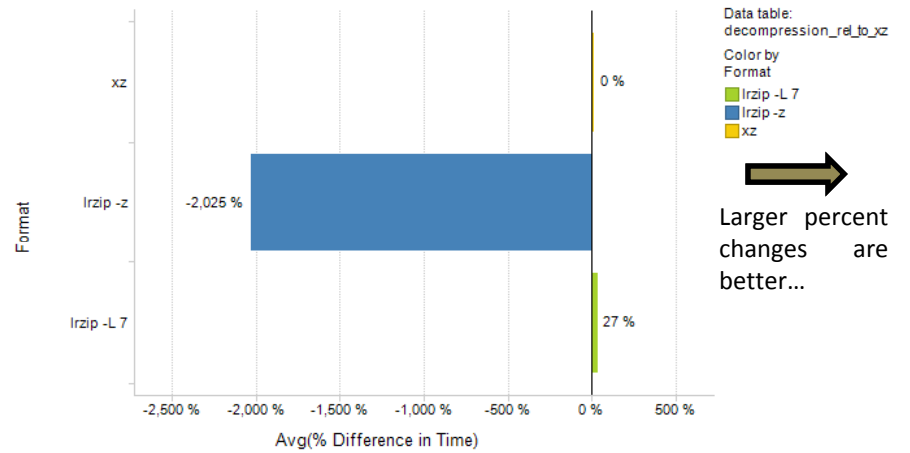
←  
Smaller times are better...

CompressionTime Normalized to .tar.xz



→  
Larger percent changes are better...

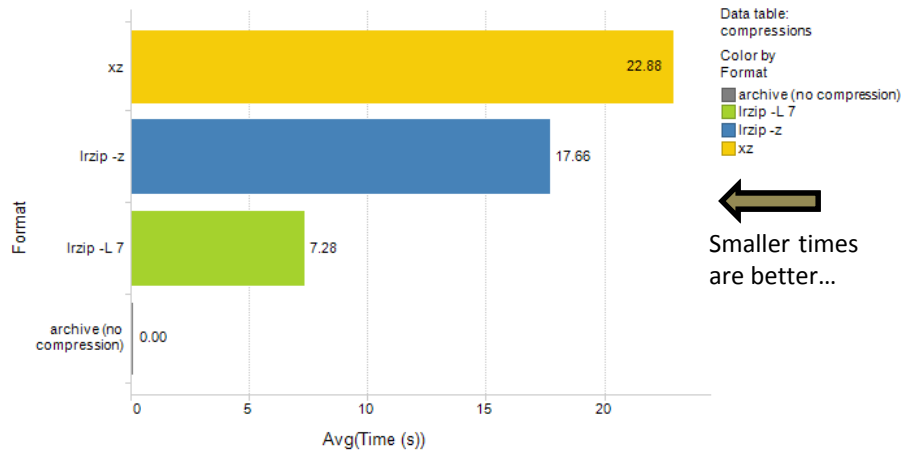
DecompressionTime Normalized to .tar.xz



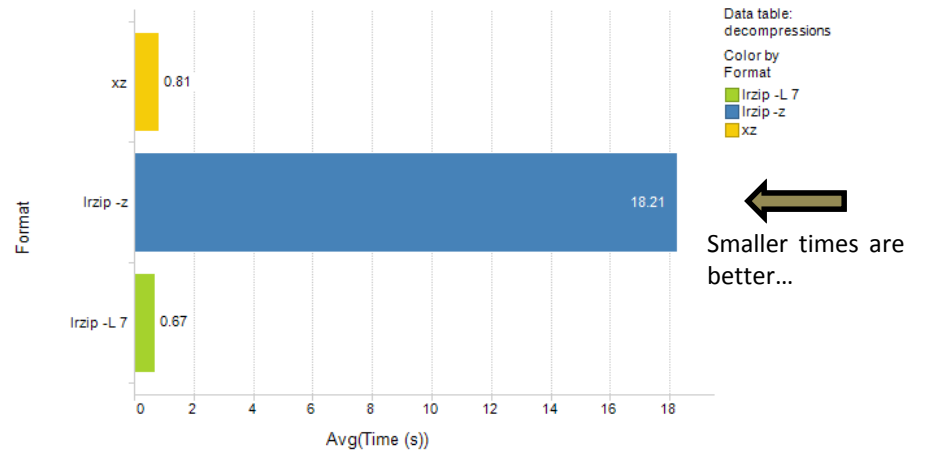
→  
Larger percent changes are better...

# gimp

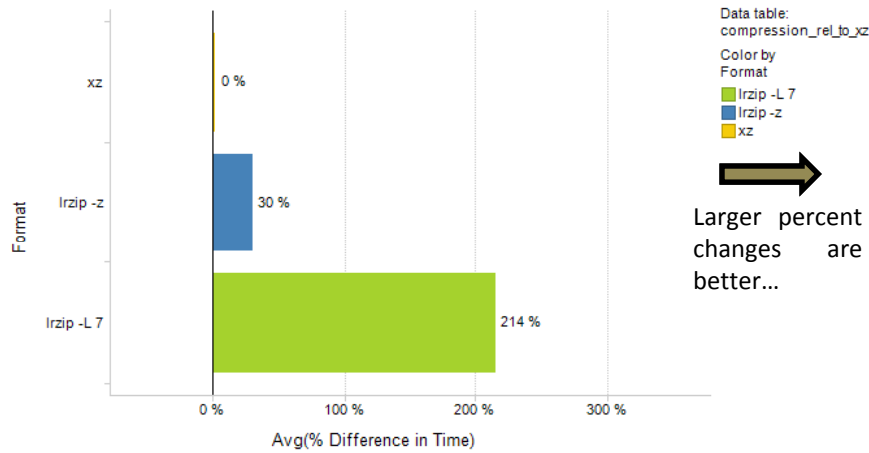
Compression Time (Sec)



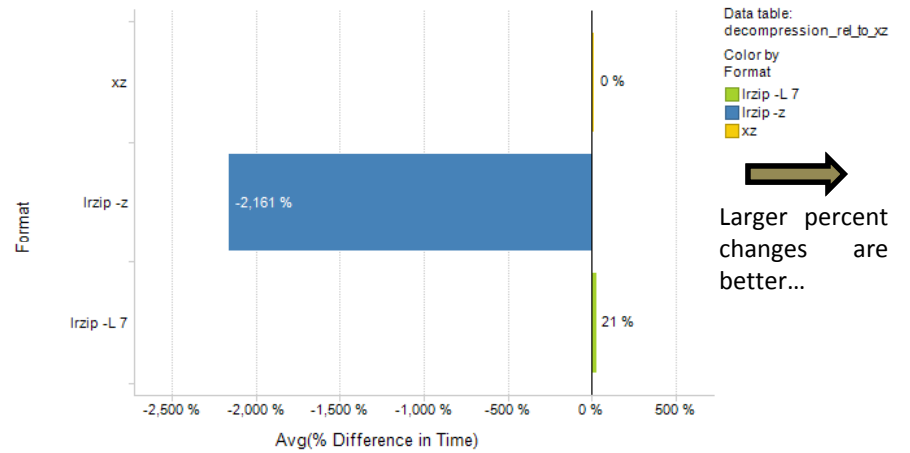
Decompression Time (Sec)



CompressionTime Normalized to .tar.xz

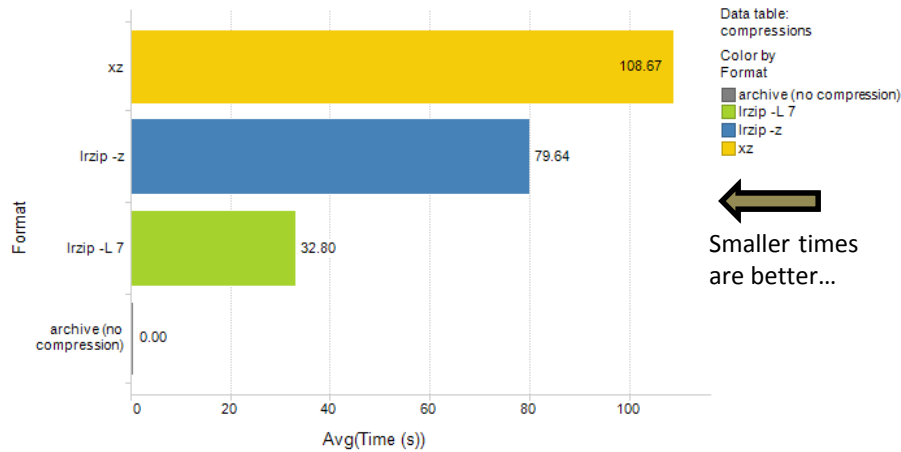


DecompressionTime Normalized to .tar.xz

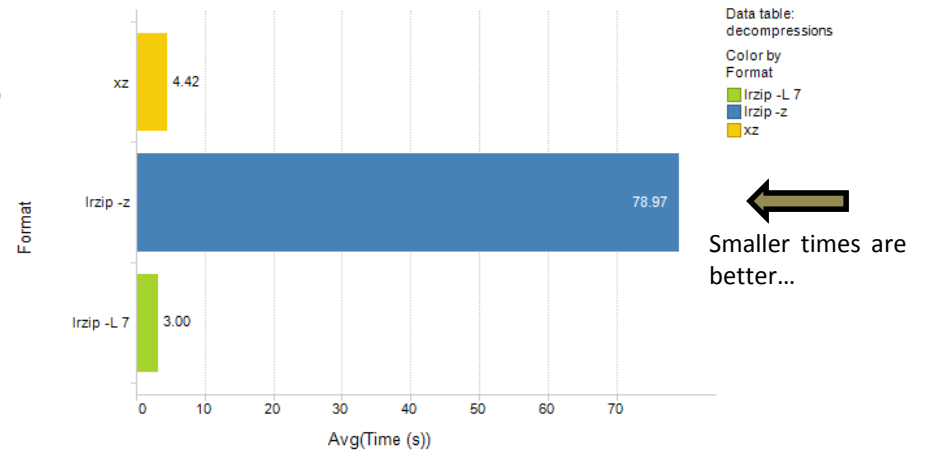


# libreoffice-common

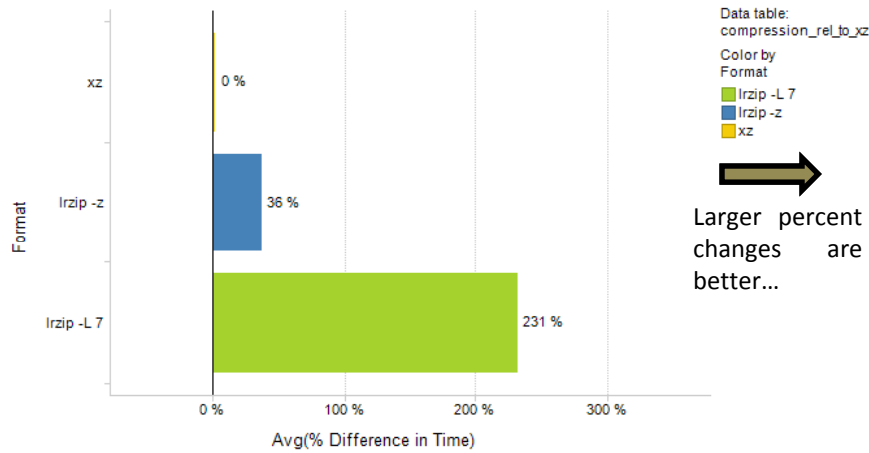
Compression Time (Sec)



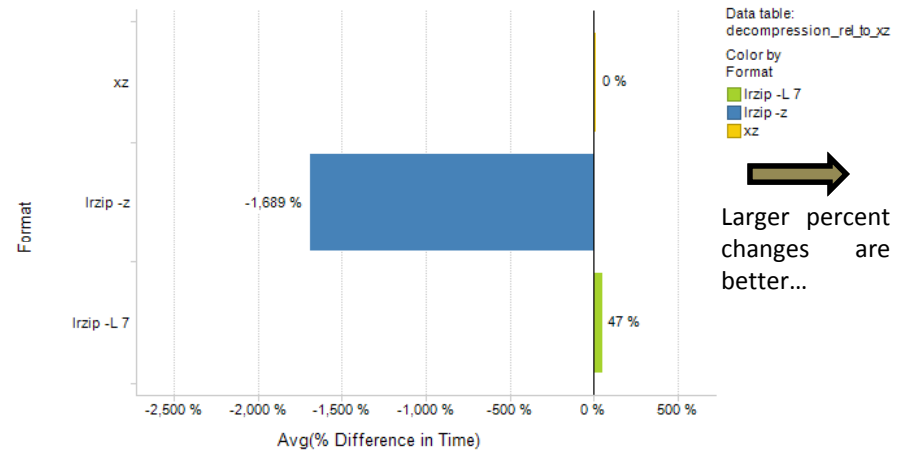
Decompression Time (Sec)



CompressionTime Normalized to .tar.xz

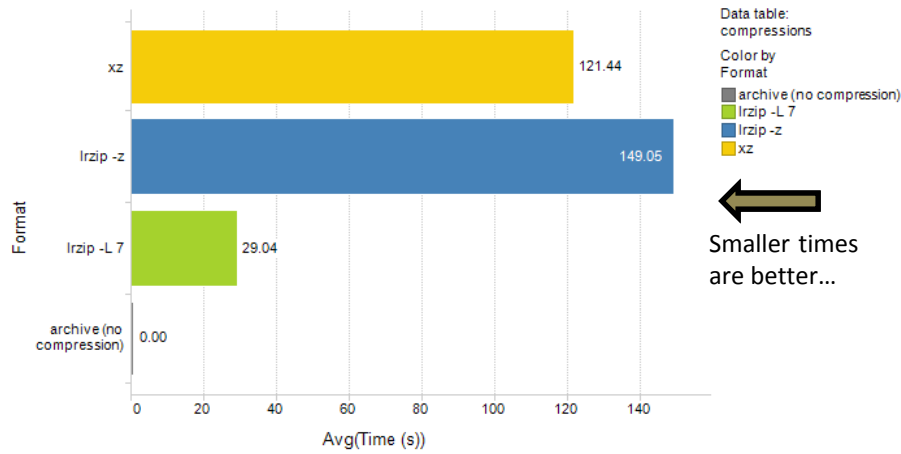


DecompressionTime Normalized to .tar.xz

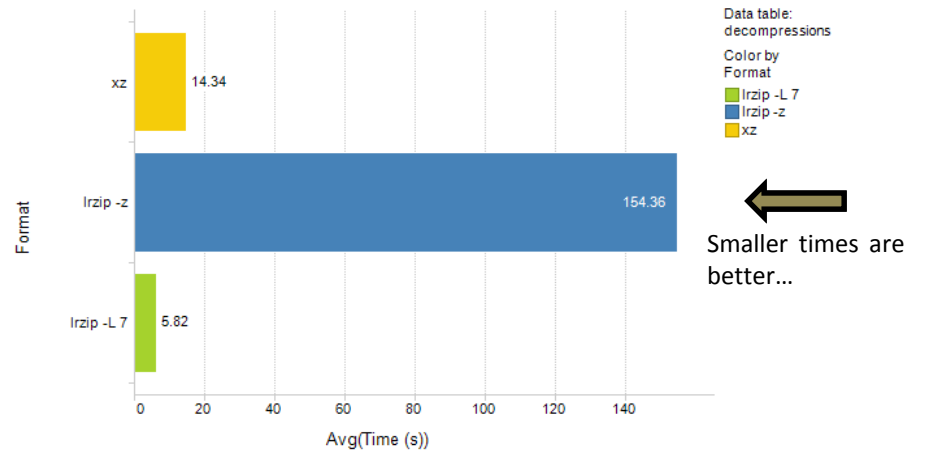


# oxygen-icons-svg

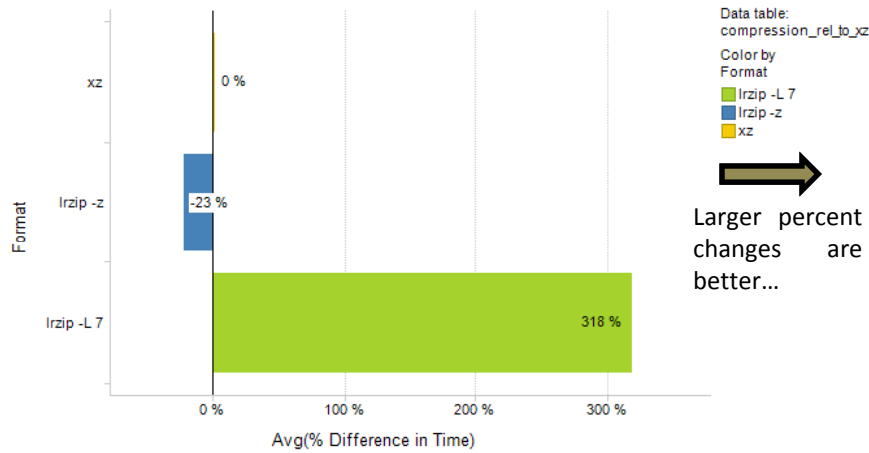
Compression Time (Sec)



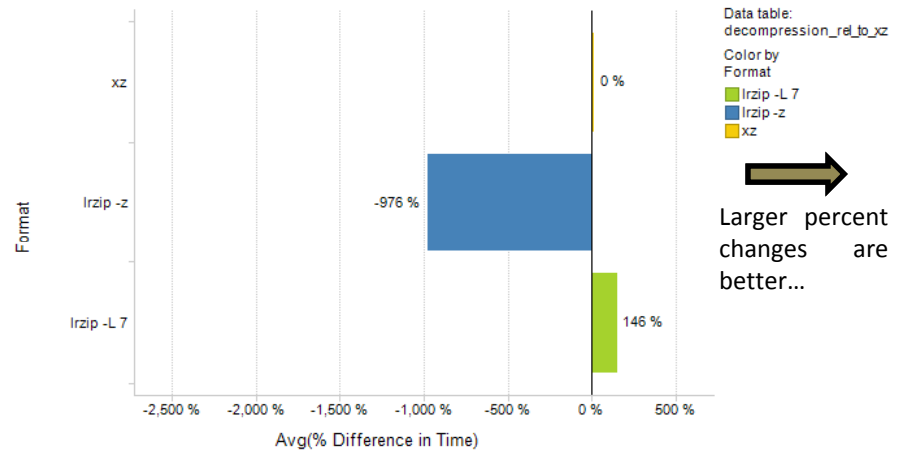
Decompression Time (Sec)



CompressionTime Normalized to .tar.xz

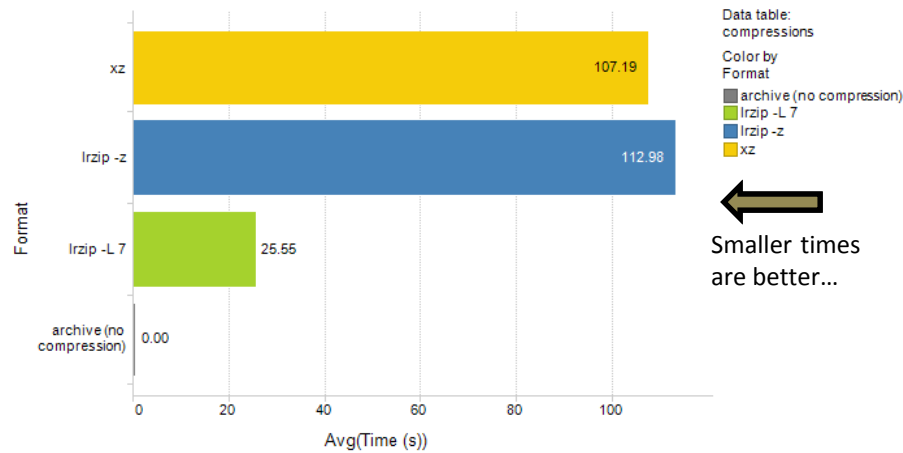


DecompressionTime Normalized to .tar.xz

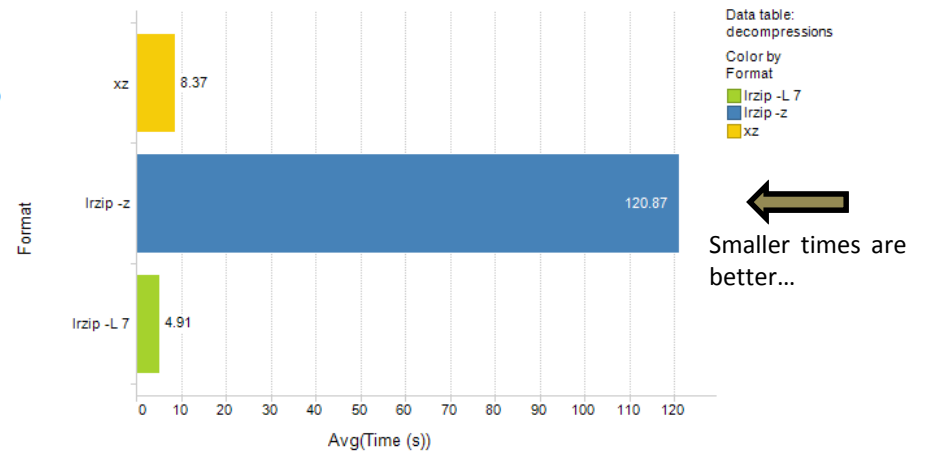


# qt-docs

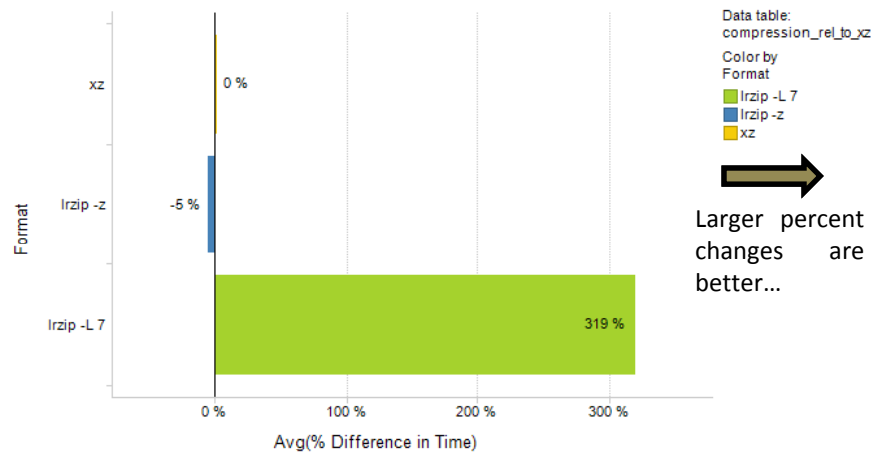
Compression Time (Sec)



Decompression Time (Sec)



CompressionTime Normalized to .tar.xz



DecompressionTime Normalized to .tar.xz

