

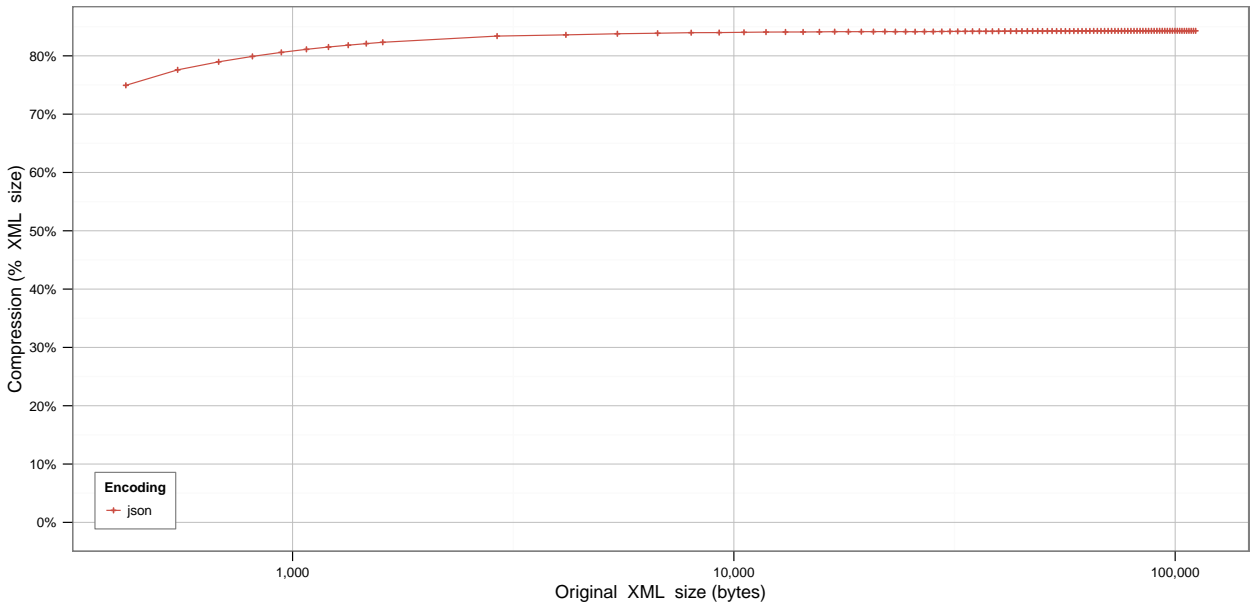
XML/JSON Analysis Template

Results for Global Positioning System XML (GPX) Use Case

Plaintext Comparisons

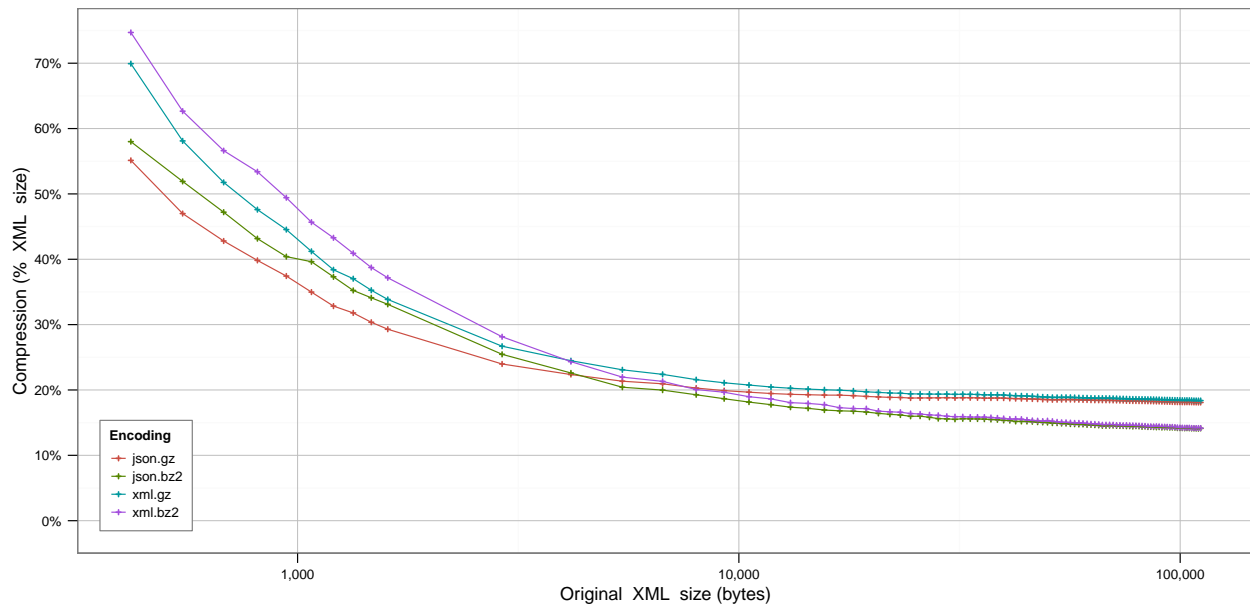
A. How do JSON and XML compare when plaintext-encoded?

```
## [1] "Series:  json"
## [1] "Baseline: xml"
##      json
## Min.   :0.7494
## 1st Qu.:0.8414
## Median :0.8425
## Mean   :0.8377
## 3rd Qu.:0.8428
## Max.   :0.8429
```



B. How do JSON and XML compare when compressed with conventional compression algorithms?

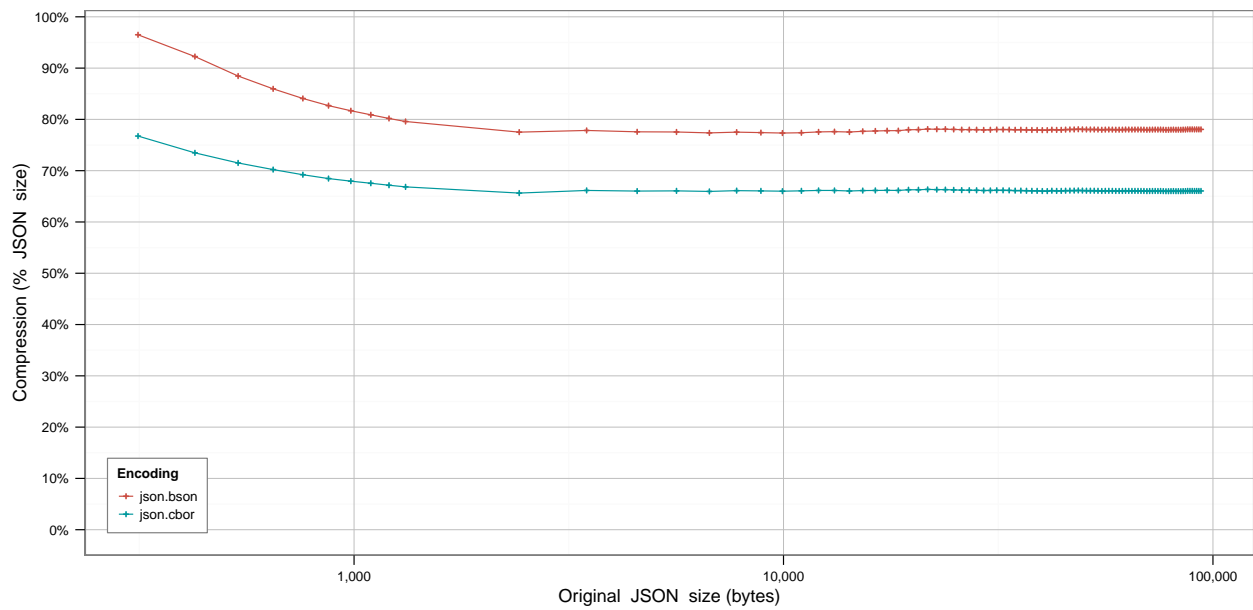
```
## [1] "Series:  json.gz, json.bz2, xml.gz, xml.bz2"
## [1] "Baseline:  xml"
##      json.gz      json.bz2      xml.gz      xml.bz2
## Min.   :0.1808   Min.   :0.1408   Min.   :0.1838   Min.   :0.1416
## 1st Qu.:0.1829   1st Qu.:0.1440   1st Qu.:0.1860   1st Qu.:0.1454
## Median :0.1853   Median :0.1497   Median :0.1890   Median :0.1522
## Mean   :0.2076   Mean   :0.1822   Mean   :0.2202   Mean   :0.1938
## 3rd Qu.:0.1896   3rd Qu.:0.1647   3rd Qu.:0.1965   3rd Qu.:0.1685
## Max.   :0.5513   Max.   :0.5800   Max.   :0.6993   Max.   :0.7470
```



JSON-Specific Exploratory

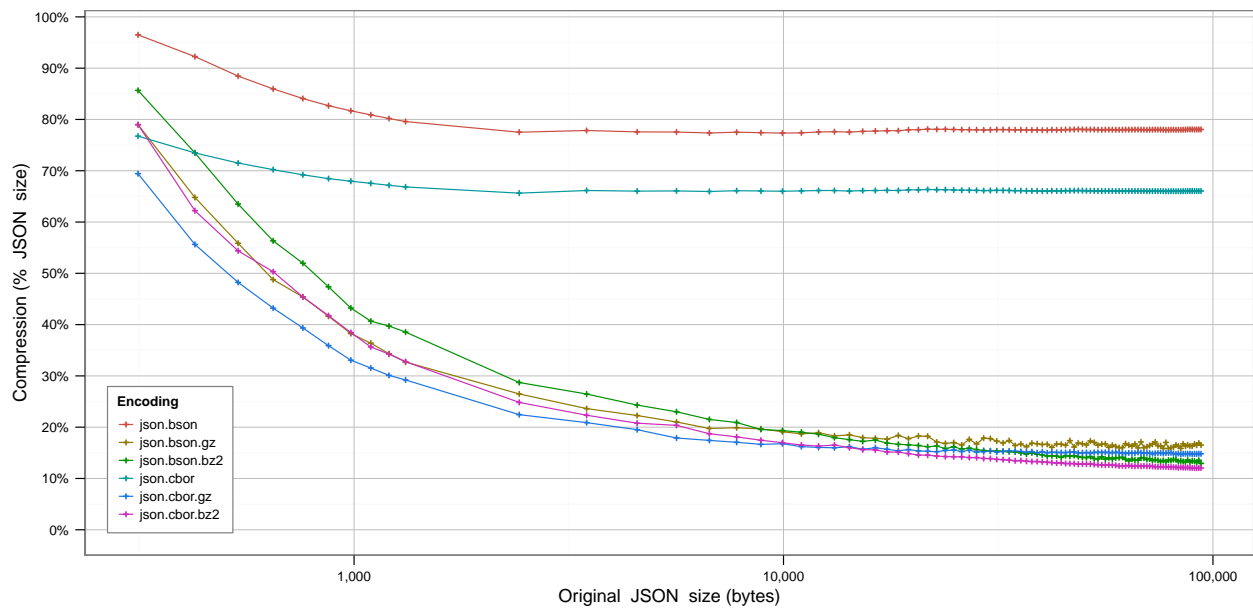
C. Which binary encoding of JSON is most compact?

```
## [1] "Series:  json.bson, json.cbor"
## [1] "Baseline:  json"
##      json.bson      json.cbor
##  Min.   :0.7734   Min.   :0.6564
## 1st Qu.:0.7796   1st Qu.:0.6604
## Median :0.7801   Median :0.6607
## Mean   :0.7869   Mean   :0.6648
## 3rd Qu.:0.7804   3rd Qu.:0.6615
## Max.   :0.9650   Max.   :0.7675
```



D. For binary JSON formats, does post-compression with conventional compression algorithms improve compactness?

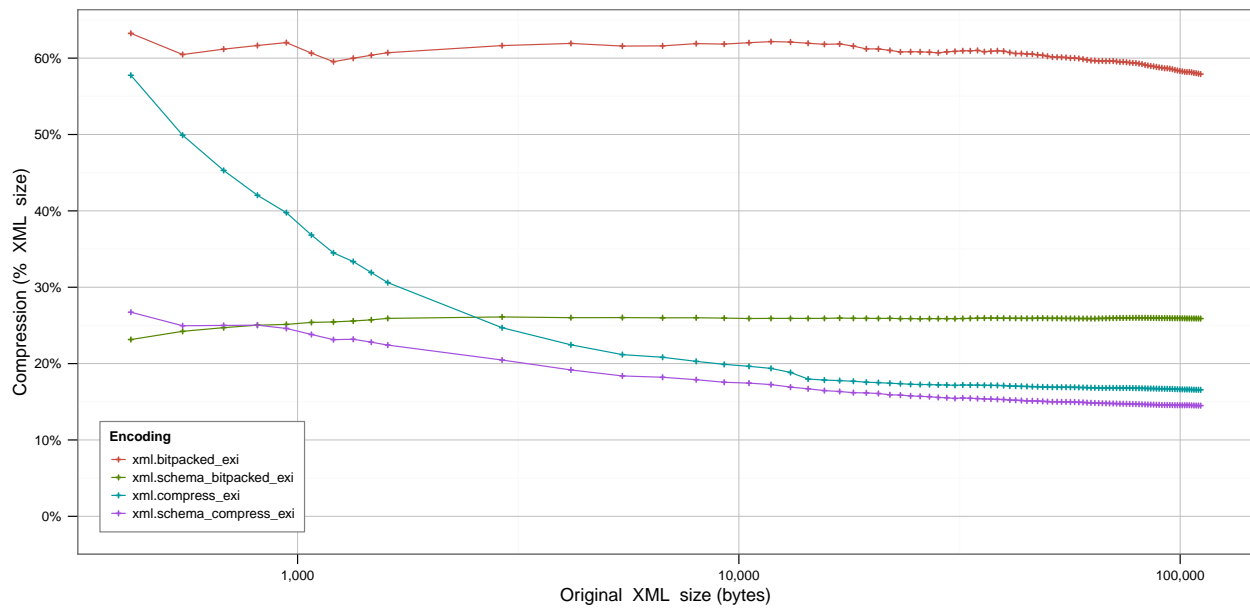
```
## [1] "Series:  json.bson, json.bson.gz, json.bson.bz2, json.cbor, json.cbor.gz, json.cbor.bz2"
## [1] "Baseline:  json"
##      json.bson      json.bson.gz      json.bson.bz2      json.cbor
## Min.   :0.7734    Min.   :0.1584    Min.   :0.1294    Min.   :0.6564
## 1st Qu.:0.7796    1st Qu.:0.1647    1st Qu.:0.1370    1st Qu.:0.6604
## Median :0.7801    Median :0.1677    Median :0.1443    Median :0.6607
## Mean   :0.7869    Mean   :0.2044    Mean   :0.1942    Mean   :0.6648
## 3rd Qu.:0.7804    3rd Qu.:0.1824    3rd Qu.:0.1698    3rd Qu.:0.6615
## Max.   :0.9650    Max.   :0.7898    Max.   :0.8567    Max.   :0.7675
##      json.cbor.gz      json.cbor.bz2
## Min.   :0.1471    Min.   :0.1203
## 1st Qu.:0.1496    1st Qu.:0.1240
## Median :0.1511    Median :0.1307
## Mean   :0.1818    Mean   :0.1731
## 3rd Qu.:0.1569    3rd Qu.:0.1526
## Max.   :0.6943    Max.   :0.7898
```



EXI Exploratory

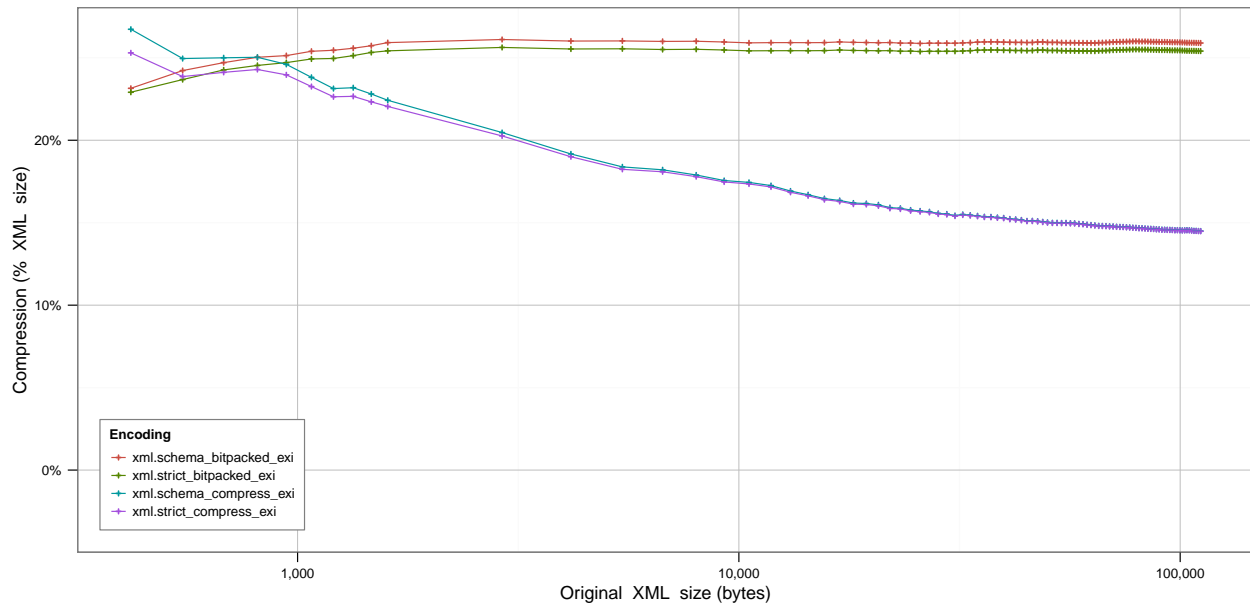
E. How do the primary EXI modes compare for schemaless & schema-informed encodings?

```
## [1] "Series:  xml.bitpacked_exi, xml.schema_bitpacked_exi, xml.compress_exi, xml.schema_compress_exi"
## [1] "Baseline:  xml"
## xml.bitpacked_exi xml.schema_bitpacked_exi xml.compress_exi
## Min.   :0.5791      Min.   :0.2315          Min.   :0.1656
## 1st Qu.:0.5927      1st Qu.:0.2591          1st Qu.:0.1677
## Median :0.6012      Median :0.2593          Median :0.1692
## Mean   :0.6011      Mean   :0.2585          Mean   :0.1973
## 3rd Qu.:0.6092      3rd Qu.:0.2596          3rd Qu.:0.1752
## Max.   :0.6325      Max.   :0.2611          Max.   :0.5776
## xml.schema_compress_exi
## Min.   :0.1450
## 1st Qu.:0.1467
## Median :0.1501
## Mean   :0.1628
## 3rd Qu.:0.1610
## Max.   :0.2673
```



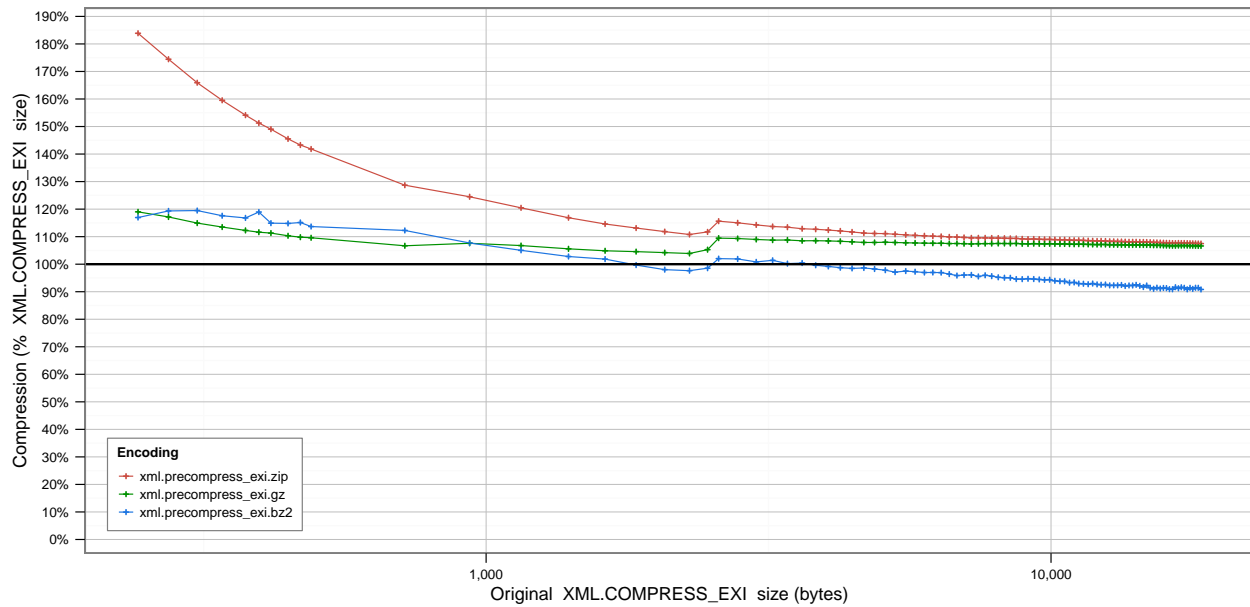
F. Does the ‘strict’ option significantly improve compaction for schema-informed encodings?

```
## [1] "Series:  xml.schema_bitpacked_exl, xml.strict_bitpacked_exl, xml.schema_compress_exl, xml.strict_compress_exl"
## [1] "Baseline:  xml"
## xml.schema_bitpacked_exl xml.strict_bitpacked_exl xml.schema_compress_exl
## Min.   :0.2315          Min.   :0.2291          Min.   :0.1450
## 1st Qu.:0.2591          1st Qu.:0.2542          1st Qu.:0.1467
## Median :0.2593          Median :0.2544          Median :0.1501
## Mean   :0.2585          Mean   :0.2536          Mean   :0.1628
## 3rd Qu.:0.2596          3rd Qu.:0.2547          3rd Qu.:0.1610
## Max.   :0.2611          Max.   :0.2563          Max.   :0.2673
## xml.strict_compress_exl
## Min.   :0.1448
## 1st Qu.:0.1466
## Median :0.1499
## Mean   :0.1617
## 3rd Qu.:0.1604
## Max.   :0.2530
```



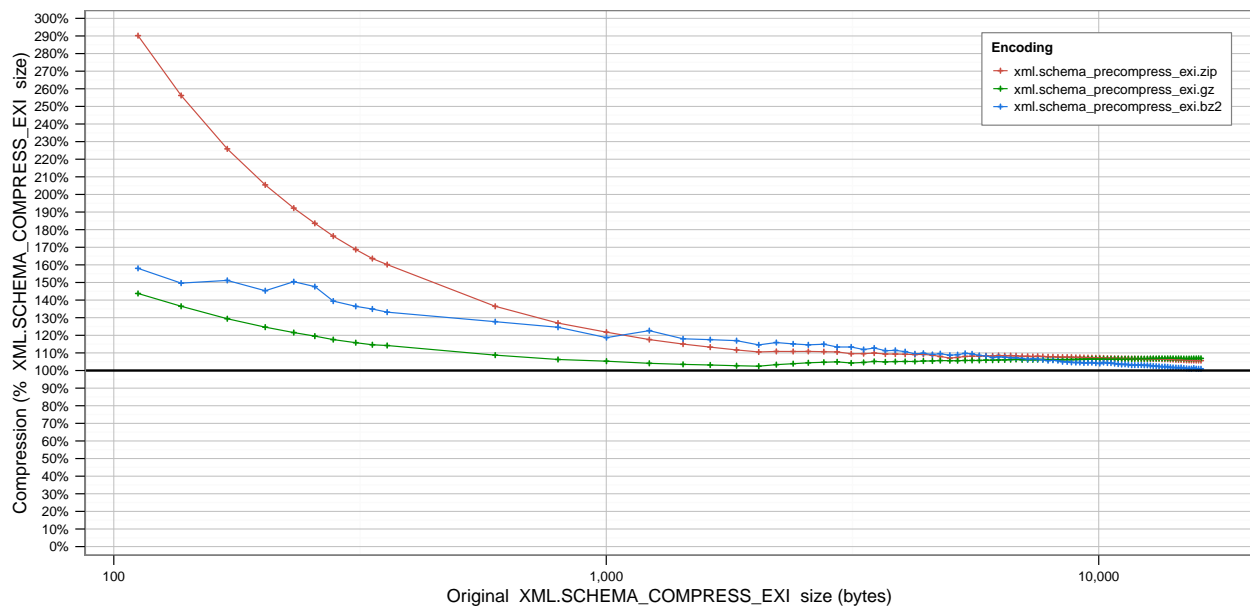
G. Do any of the tested conventional compression algorithms perform better on a schemaless, precompress EXI document than the standard DEFLATE?

```
## [1] "Series:  xml.precompress_exi.zip, xml.precompress_exi.gz, xml.precompress_exi.bz2"
## [1] "Baseline:  xml.compress_exi"
##  xml.precompress_exi.zip xml.precompress_exi.gz xml.precompress_exi.bz2
##  Min.    :1.075          Min.    :1.039          Min.    :0.9082
##  1st Qu.:1.082          1st Qu.:1.069          1st Qu.:0.9229
##  Median :1.094          Median :1.073          Median :0.9487
##  Mean   :1.150          Mean   :1.078          Mean   :0.9755
##  3rd Qu.:1.121          3rd Qu.:1.078          3rd Qu.:0.9884
##  Max.   :1.839          Max.   :1.190          Max.   :1.1948
```



H. Do any of the tested conventional compression algorithms perform better on a schema-informed, precompress EXI document than the standard DEFLATE?

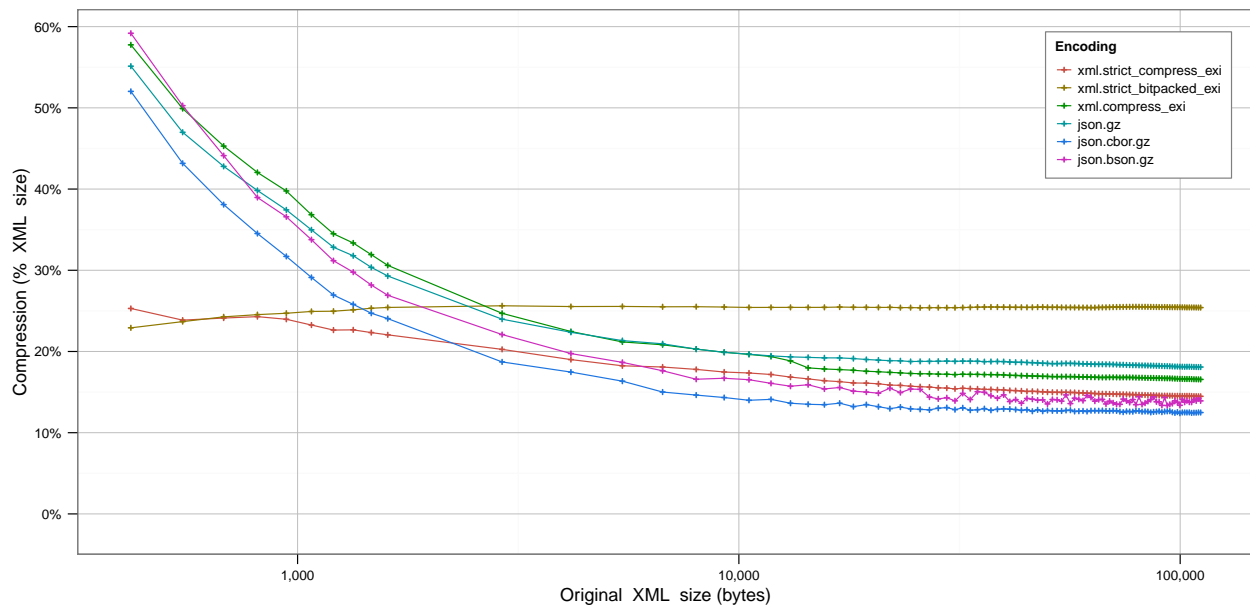
```
## [1] "Series:  xml.schema_precompress_exi.zip, xml.schema_precompress_exi.gz, xml.schema_precompress_exi.bz2"
## [1] "Baseline: xml.schema_compress_exi"
## xml.schema_precompress_exi.zip xml.schema_precompress_exi.gz
## Min.      :1.056                Min.      :1.025
## 1st Qu.:1.067                1st Qu.:1.057
## Median :1.078                Median :1.065
## Mean    :1.184                Mean    :1.078
## 3rd Qu.:1.097                3rd Qu.:1.068
## Max.    :2.902                Max.    :1.438
## xml.schema_precompress_exi.bz2
## Min.      :1.010
## 1st Qu.:1.031
## Median :1.062
## Mean    :1.110
## 3rd Qu.:1.129
## Max.    :1.580
```



Binary-comparisons

I. Which binary format is the most compact?

```
## [1] "Series:  xml.strict_compress_exl xml.strict_bitpacked_exl xml.compress_exl json.gz json.cb
## [1] "Baseline: xml"
## xml.strict_compress_exl xml.strict_bitpacked_exl xml.compress_exl
## Min. :0.1448 Min. :0.2291 Min. :0.1656
## 1st Qu.:0.1466 1st Qu.:0.2542 1st Qu.:0.1677
## Median :0.1499 Median :0.2544 Median :0.1692
## Mean :0.1617 Mean :0.2536 Mean :0.1973
## 3rd Qu.:0.1604 3rd Qu.:0.2547 3rd Qu.:0.1752
## Max. :0.2530 Max. :0.2563 Max. :0.5776
## json.gz json.cbor.gz json.bson.gz
## Min. :0.1808 Min. :0.1240 Min. :0.1335
## 1st Qu.:0.1829 1st Qu.:0.1261 1st Qu.:0.1388
## Median :0.1853 Median :0.1273 Median :0.1413
## Mean :0.2076 Mean :0.1510 Mean :0.1697
## 3rd Qu.:0.1896 3rd Qu.:0.1320 3rd Qu.:0.1535
## Max. :0.5513 Max. :0.5203 Max. :0.5919
```



J. Do any of the binary formats offer improvement for a network already using gzip?

```
## [1] "Series:  xml.strict_compress_exl xml.strict_bitpacked_exl xml.compress_exl json.gz, json.cb
## [1] "Baseline:  xml.gz"
## xml.strict_compress_exl xml.strict_bitpacked_exl xml.compress_exl
## Min.    :0.3618          Min.    :0.3276          Min.    :0.8259
## 1st Qu.:0.7876          1st Qu.:1.2941          1st Qu.:0.8915
## Median :0.7891          Median :1.3460          Median :0.8974
## Mean    :0.7684          Mean    :1.2473          Mean    :0.8985
## 3rd Qu.:0.7961          3rd Qu.:1.3707          3rd Qu.:0.9015
## Max.    :0.8392          Max.    :1.3823          Max.    :0.9471
## json.gz json.cbor.gz json.bson.gz
## Min.    :0.7884 Min.    :0.6604 Min.    :0.7134
## 1st Qu.:0.9647 1st Qu.:0.6724 1st Qu.:0.7399
## Median :0.9802 Median :0.6764 Median :0.7560
## Mean    :0.9598 Mean    :0.6799 Mean    :0.7612
## 3rd Qu.:0.9831 3rd Qu.:0.6792 3rd Qu.:0.7770
## Max.    :0.9838 Max.    :0.7440 Max.    :0.8652
```

