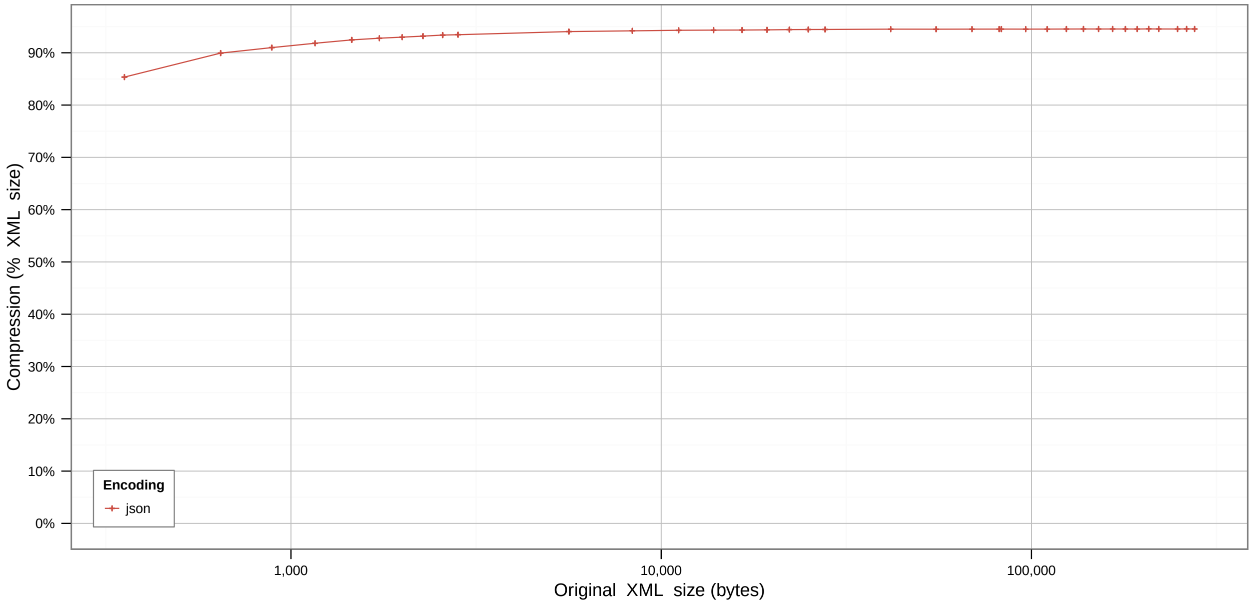
XML/JSON Analysis Template

Results for Automated Identification System (AIS) Use Case

Plaintext Comparisons

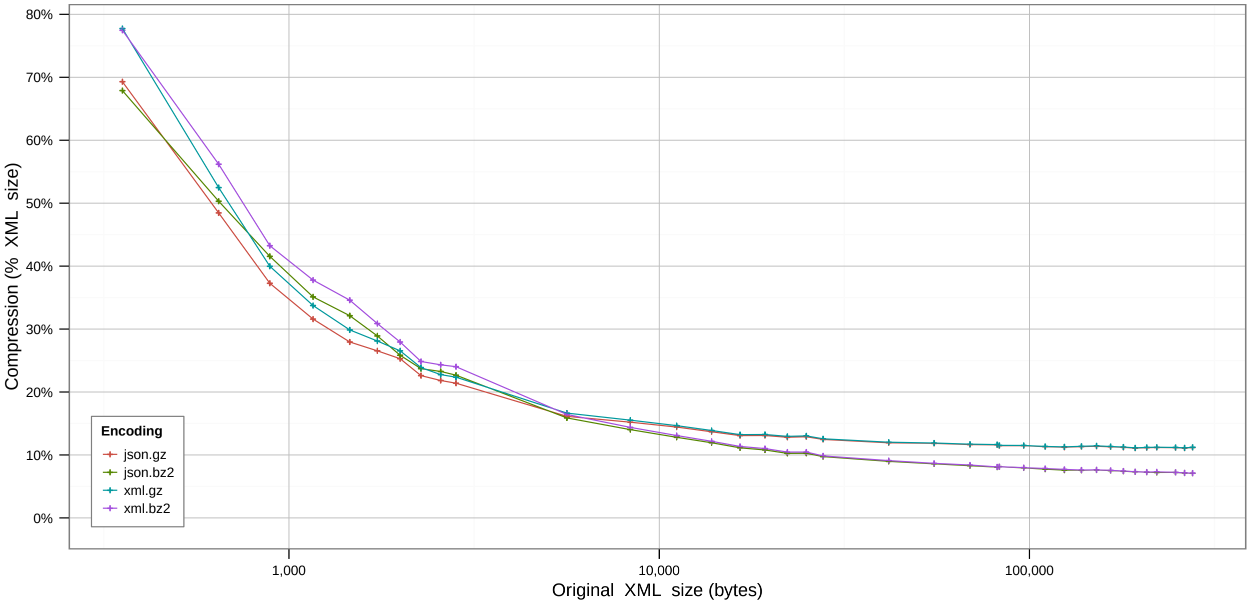
A. How do JSON and XML compare when plaintext-encoded?

```
## [1] "Series:   json"
## [1] "Baseline:  xml"
##        json
##  Min.   :0.8535
##  1st Qu.:0.9346
##  Median :0.9446
##  Mean   :0.9370
##  3rd Qu.:0.9454
##  Max.   :0.9456
```



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

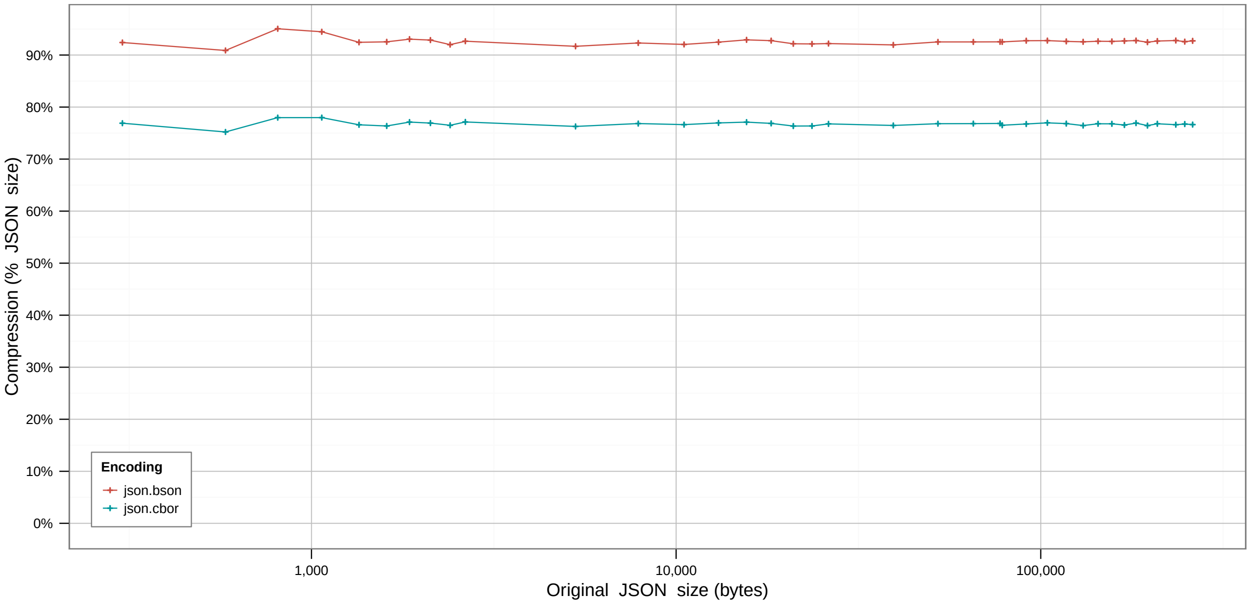
```
## [1] "Series:   json.gz, json.bz2, xml.gz, xml.bz2"
## [1] "Baseline:  xml"
##     json.gz          json.bz2           xml.gz          xml.bz2
##  Min.   :0.1108   Min.   :0.07101   Min.   :0.1112   Min.   :0.07113
##  1st Qu.:0.1131   1st Qu.:0.07589   1st Qu.:0.1135   1st Qu.:0.07633
##  Median :0.1247   Median :0.09735   Median :0.1256   Median :0.09847
##  Mean   :0.1785   Mean   :0.16133   Mean   :0.1861   Mean   :0.17029
##  3rd Qu.:0.2140   3rd Qu.:0.22674   3rd Qu.:0.2236   3rd Qu.:0.24018
##  Max.   :0.6930   Max.   :0.67887   Max.   :0.7775   Max.   :0.77465
```



JSON-Specific Exploratory

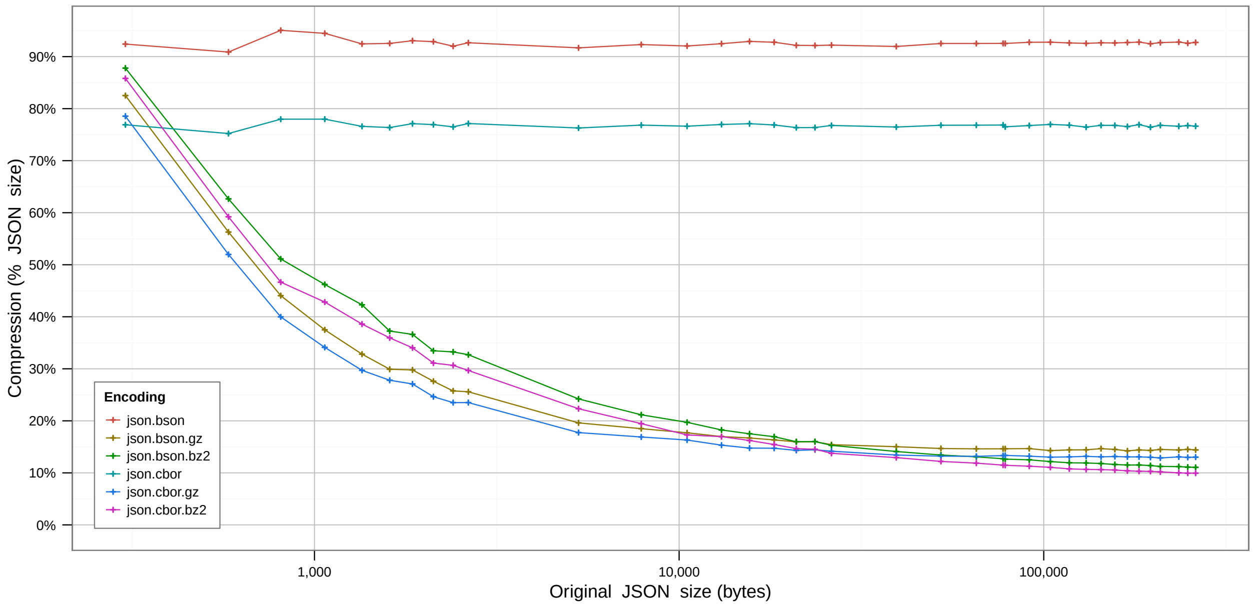
C. Which binary encoding of JSON is most compact?

```
## [1] "Series:   json.bson, json.cbor"
## [1] "Baseline:  json"
##    json.bson         json.cbor     
##  Min.   :0.9088   Min.   :0.7522  
##  1st Qu.:0.9241   1st Qu.:0.7651  
##  Median :0.9254   Median :0.7678  
##  Mean   :0.9259   Mean   :0.7674  
##  3rd Qu.:0.9275   3rd Qu.:0.7690  
##  Max.   :0.9505   Max.   :0.7798  
```



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

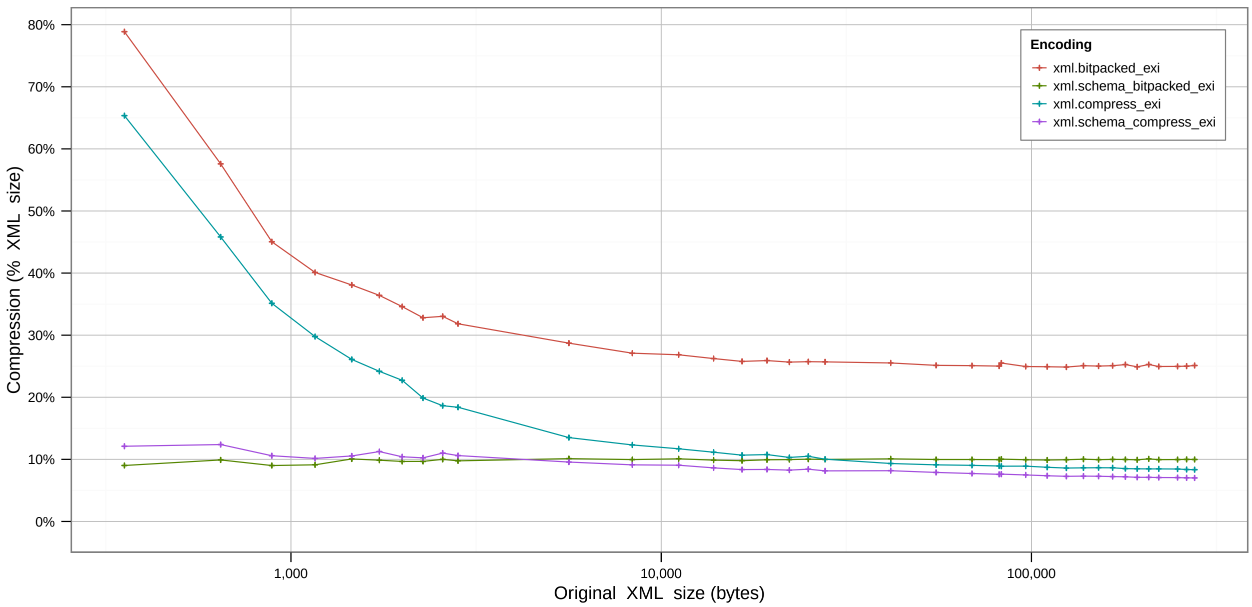
```
## [1] "Series:   json.bson, json.bson.gz, json.bson.bz2, json.cbor, json.cbor.gz, json.cbor.bz2"
## [1] "Baseline:  json"
##    json.bson       json.bson.gz     json.bson.bz2      json.cbor
##  Min.   :0.9088   Min.   :0.1423   Min.   :0.1106   Min.   :0.7522
##  1st Qu.:0.9241   1st Qu.:0.1451   1st Qu.:0.1190   1st Qu.:0.7651
##  Median :0.9254   Median :0.1542   Median :0.1529   Median :0.7678
##  Mean   :0.9259   Mean   :0.2179   Mean   :0.2285   Mean   :0.7674
##  3rd Qu.:0.9275   3rd Qu.:0.2559   3rd Qu.:0.3270   3rd Qu.:0.7690
##  Max.   :0.9505   Max.   :0.8251   Max.   :0.8779   Max.   :0.7798
##   json.cbor.gz    json.cbor.bz2
##  Min.   :0.1287   Min.   :0.09942
##  1st Qu.:0.1308   1st Qu.:0.10667
##  Median :0.1416   Median :0.13760
##  Mean   :0.1989   Mean   :0.21112
##  3rd Qu.:0.2350   3rd Qu.:0.29674
##  Max.   :0.7855   Max.   :0.85809
```



EXI Exploratory

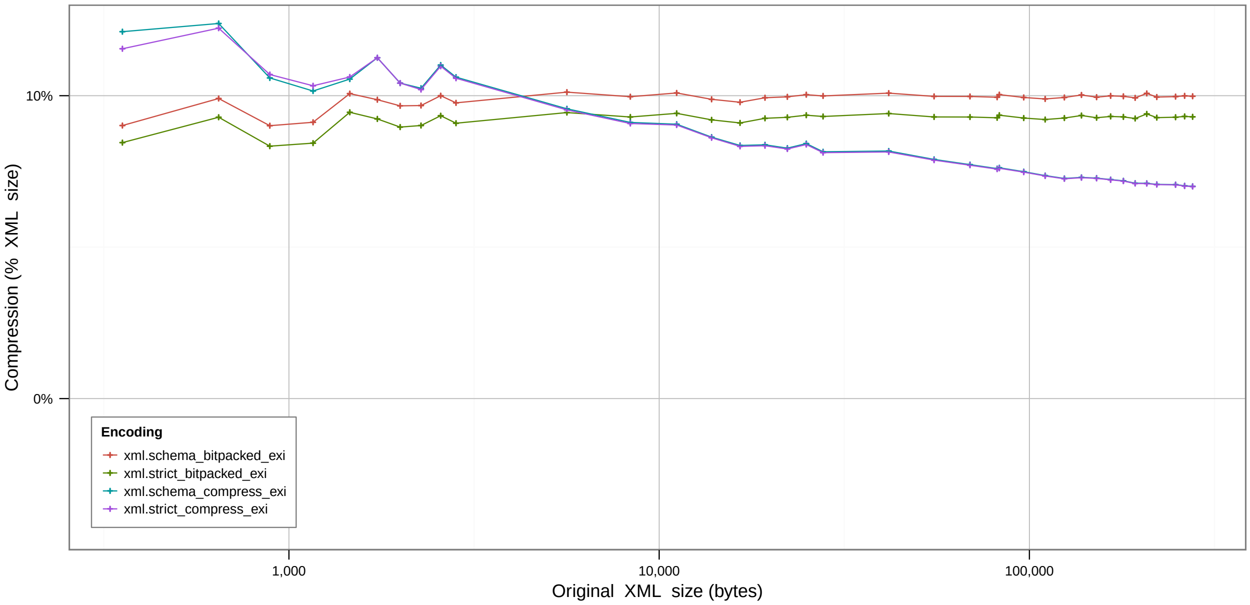
E. How do the primary EXI modes compare for schemaless & schema-informed encodings?

```
## [1] "Series:   xml.bitpacked_exi, xml.schema_bitpacked_exi, xml.compress_exi, xml.schema_compress_ex
## [1] "Baseline:  xml"
##  xml.bitpacked_exi xml.schema_bitpacked_exi xml.compress_exi
##  Min.   :0.2487    Min.   :0.09009          Min.   :0.08339
##  1st Qu.:0.2508    1st Qu.:0.09892          1st Qu.:0.08643
##  Median :0.2566    Median :0.09962          Median :0.10028
##  Mean   :0.3021    Mean   :0.09878          Mean   :0.15230
##  3rd Qu.:0.3184    3rd Qu.:0.09994          3rd Qu.:0.18394
##  Max.   :0.7887    Max.   :0.10117          Max.   :0.65352
##  xml.schema_compress_exi
##  Min.   :0.07008
##  1st Qu.:0.07280
##  Median :0.08173
##  Mean   :0.08643
##  3rd Qu.:0.10155
##  Max.   :0.12384
```



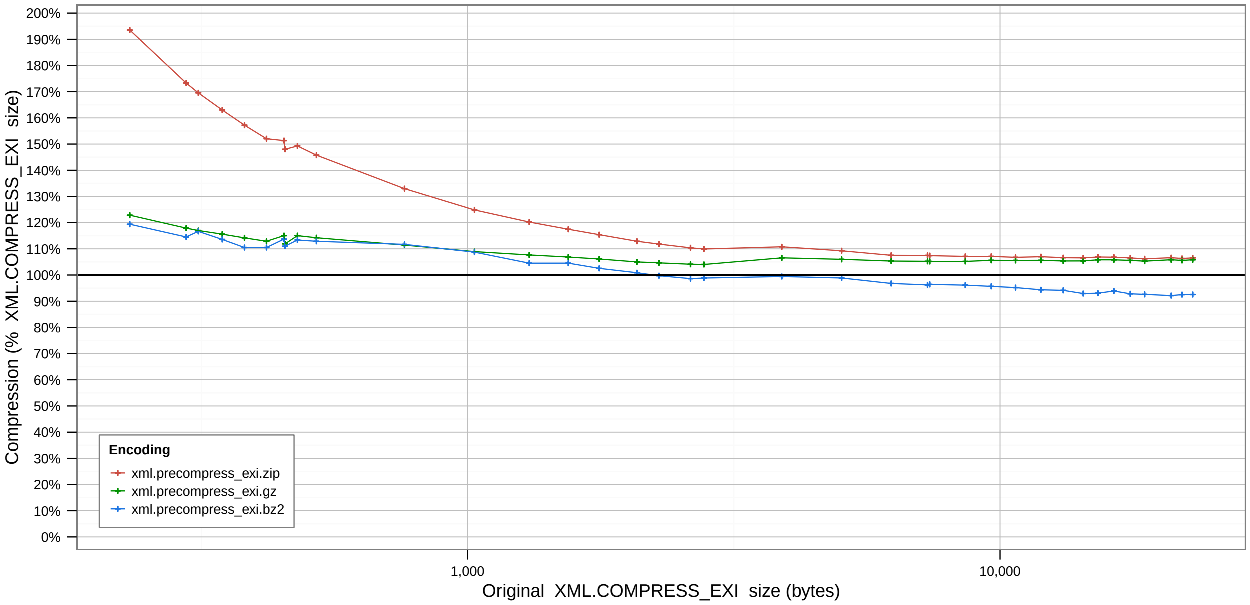
F. Does the ‘strict’ option significantly improve compaction for schema-informed encodings?

```
## [1] "Series:   xml.schema_bitpacked_exi, xml.strict_bitpacked_exi, xml.schema_compress_exi, xml.stric
## [1] "Baseline:  xml"
##  xml.schema_bitpacked_exi xml.strict_bitpacked_exi xml.schema_compress_exi
##  Min.   :0.09009          Min.   :0.08333          Min.   :0.07008
##  1st Qu.:0.09892          1st Qu.:0.09233          1st Qu.:0.07280
##  Median :0.09962          Median :0.09288          Median :0.08173
##  Mean   :0.09878          Mean   :0.09208          Mean   :0.08643
##  3rd Qu.:0.09994          3rd Qu.:0.09316          3rd Qu.:0.10155
##  Max.   :0.10117          Max.   :0.09452          Max.   :0.12384
##  xml.strict_compress_exi
##  Min.   :0.07001
##  1st Qu.:0.07270
##  Median :0.08142
##  Mean   :0.08616
##  3rd Qu.:0.10202
##  Max.   :0.12229
```



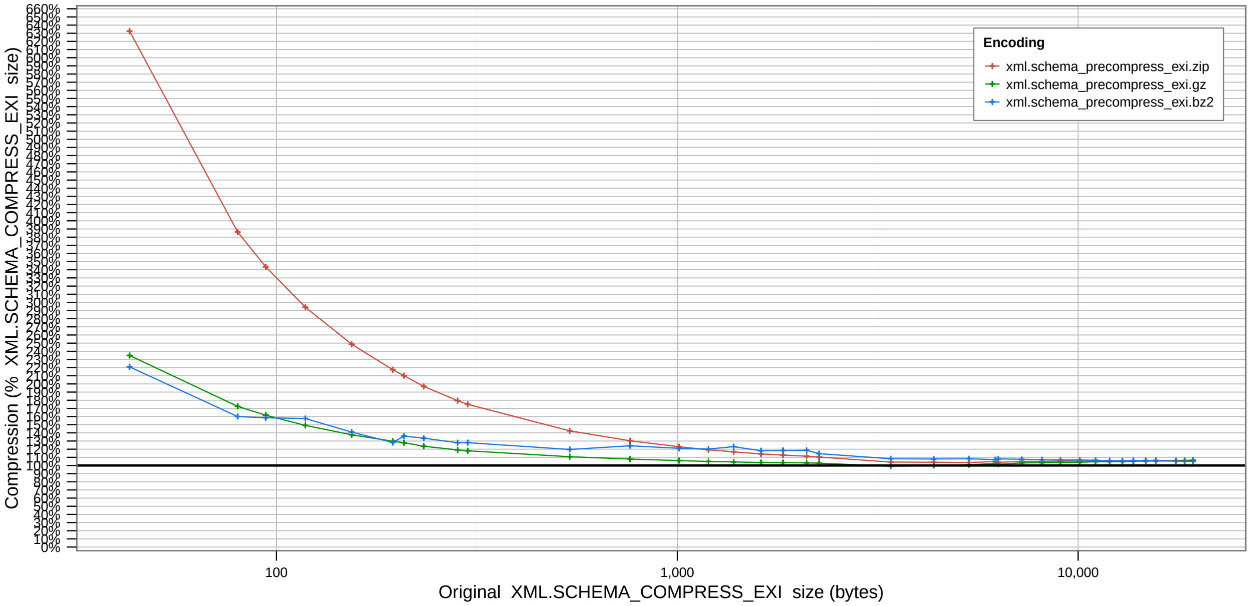
G. Do any of the tested conventional compression algorithms perform better on a schemaless, precompress EXI document than the standard DEFLATE?

```
## [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.   :1.062           Min.   :1.040          Min.   :0.9215
##  1st Qu.:1.069           1st Qu.:1.054          1st Qu.:0.9436
##  Median :1.104           Median :1.058          Median :0.9887
##  Mean   :1.240           Mean   :1.085          Mean   :1.0195
##  3rd Qu.:1.458           3rd Qu.:1.119          3rd Qu.:1.1050
##  Max.   :1.935           Max.   :1.228          Max.   :1.1940
```



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

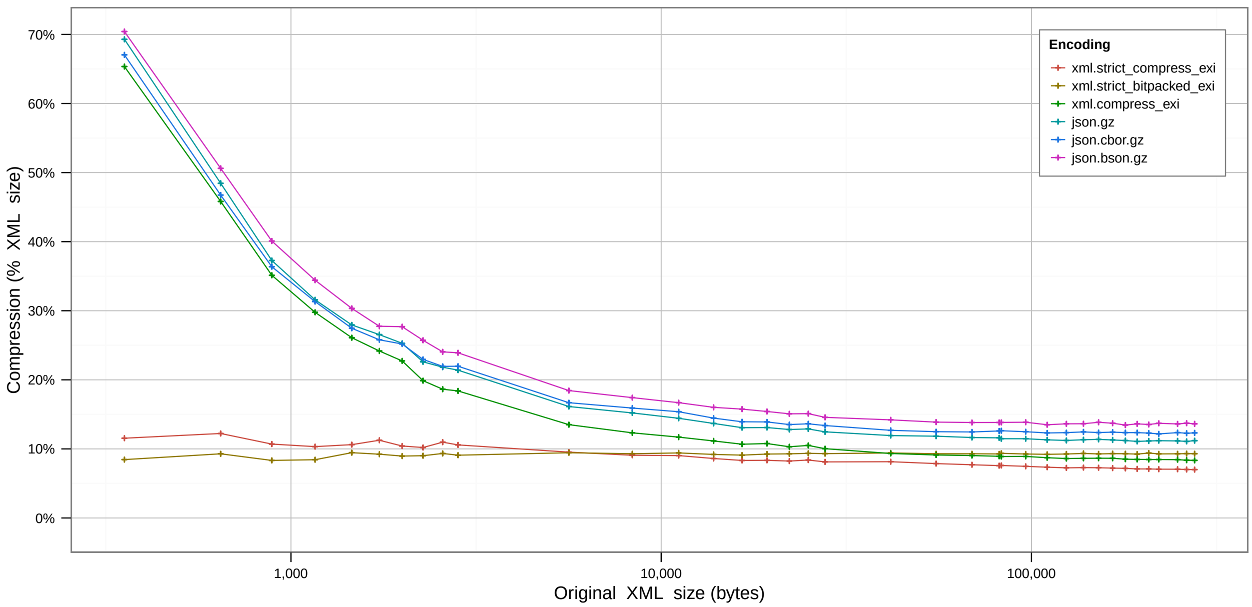
```
## [1] "Series:   xml.schema_precompress_exi.zip, xml.schema_precompress_exi.gz, xml.schema_precompress_
## [1] "Baseline:  xml.schema_compress_exi"
##  xml.schema_precompress_exi.zip xml.schema_precompress_exi.gz
##  Min.   :1.036                  Min.   :0.9915
##  1st Qu.:1.055                  1st Qu.:1.0349
##  Median :1.104                  Median :1.0535
##  Mean   :1.584                  Mean   :1.1588
##  3rd Qu.:1.750                  3rd Qu.:1.1800
##  Max.   :6.326                  Max.   :2.3488
##  xml.schema_precompress_exi.bz2
##  Min.   :1.051
##  1st Qu.:1.068
##  Median :1.145
##  Mean   :1.213
##  3rd Qu.:1.279
##  Max.   :2.209
```



Binary-comparisons

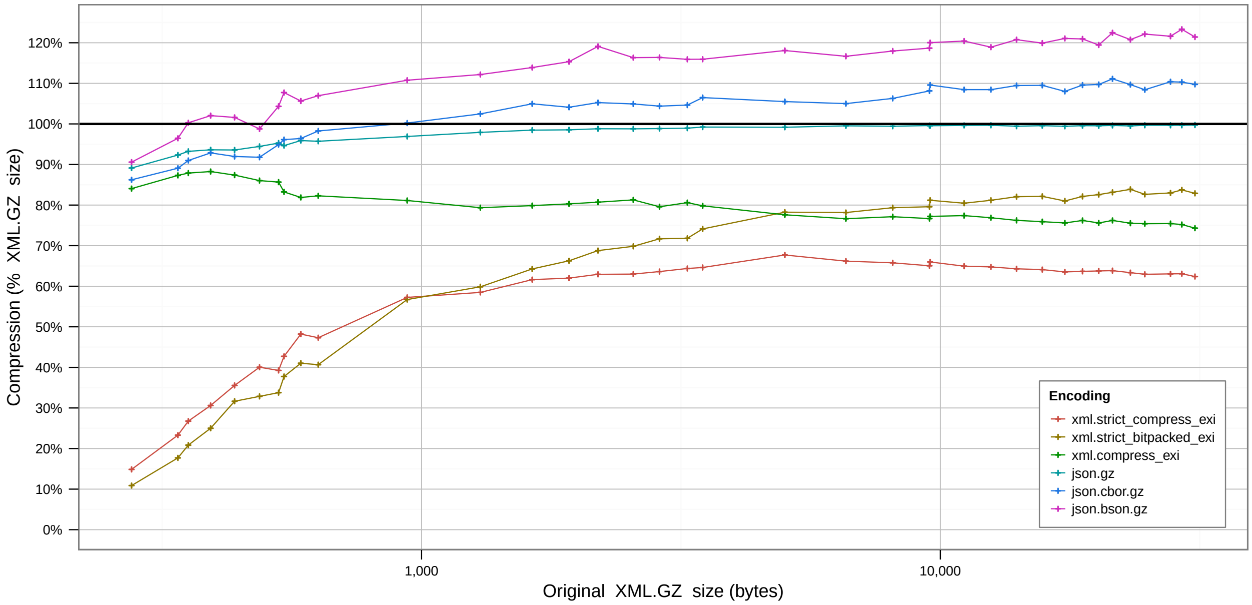
I. Which binary format is the most compact?

```
## [1] "Series:   xml.strict_compress_exi, xml.strict_bitpacked_exi, xml.compress_exi, json.gz, json.cb
## [1] "Baseline:  xml"
##  xml.strict_compress_exi xml.strict_bitpacked_exi xml.compress_exi
##  Min.   :0.07001         Min.   :0.08333          Min.   :0.08339
##  1st Qu.:0.07270         1st Qu.:0.09233          1st Qu.:0.08643
##  Median :0.08142         Median :0.09288          Median :0.10028
##  Mean   :0.08616         Mean   :0.09208          Mean   :0.15230
##  3rd Qu.:0.10202         3rd Qu.:0.09316          3rd Qu.:0.18394
##  Max.   :0.12229         Max.   :0.09452          Max.   :0.65352
##     json.gz        json.cbor.gz     json.bson.gz
##  Min.   :0.1108   Min.   :0.1217   Min.   :0.1345
##  1st Qu.:0.1131   1st Qu.:0.1236   1st Qu.:0.1372
##  Median :0.1247   Median :0.1338   Median :0.1457
##  Mean   :0.1785   Mean   :0.1840   Mean   :0.2018
##  3rd Qu.:0.2140   3rd Qu.:0.2195   3rd Qu.:0.2391
##  Max.   :0.6930   Max.   :0.6704   Max.   :0.7042
```



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

```
## [1] "Series:   xml.strict_compress_exi, xml.strict_bitpacked_exi, xml.compress_exi, json.gz, json.cb
## [1] "Baseline:  xml.gz"
##  xml.strict_compress_exi xml.strict_bitpacked_exi xml.compress_exi
##  Min.   :0.1486          Min.   :0.1087           Min.   :0.7431
##  1st Qu.:0.4821          1st Qu.:0.4103           1st Qu.:0.7621
##  Median :0.6305          Median :0.7412           Median :0.7935
##  Mean   :0.5581          Mean   :0.6386           Mean   :0.7967
##  3rd Qu.:0.6429          3rd Qu.:0.8206           3rd Qu.:0.8188
##  Max.   :0.6770          Max.   :0.8387           Max.   :0.8827
##     json.gz        json.cbor.gz     json.bson.gz
##  Min.   :0.8913   Min.   :0.8623   Min.   :0.9058
##  1st Qu.:0.9590   1st Qu.:0.9826   1st Qu.:1.0773
##  Median :0.9918   Median :1.0526   Median :1.1667
##  Mean   :0.9774   Mean   :1.0334   Mean   :1.1392
##  3rd Qu.:0.9956   3rd Qu.:1.0946   3rd Qu.:1.2042
##  Max.   :0.9972   Max.   :1.1111   Max.   :1.2333
```


Compression (% XML.GZ size)
Original XML.GZ size (bytes)
Encoding
xml.strict_compress_exi
xml.strict_bitpacked_exi
xml.compress_exi
json.gz
json.cbor.gz
json.bson.gz