HTTP Archive metrics research

Features worth looking at:

```
#standardSQL
SELECT DISTINCT feature, pct_urls
FROM `httparchive.blink_features.usage`
WHERE yyyymmdd >= '20190601'
  AND pct_urls > 1/100 AND pct_urls < 99/100
ORDER BY pct_urls DESC
```

https://gist.github.com/foolip/cffdec7a236bf3c21406bef13383456f creates a table for each feature, which can then be queried all together:

```
#standardSQL
SELECT
  _TABLE_SUFFIX AS feature,
  TRUE AS used,
  COUNT(*) AS pages,
  APPROX_QUANTILES(fcp, 100)[OFFSET(10)] / 1000 AS q10,
  APPROX_QUANTILES(fcp, 100)[OFFSET(50)] / 1000 AS q50,
  APPROX_QUANTILES(fcp, 100)[OFFSET(90)] / 1000 AS q90
FROM `blink-httparchive-research.foolip.pages_mobile_feature_*`
WHERE used IS NOT NULL
GROUP BY feature

UNION ALL

SELECT
  _TABLE_SUFFIX AS feature,
  FALSE AS used,
  COUNT(*) AS pages,
  APPROX_QUANTILES(fcp, 100)[OFFSET(10)] / 1000 AS q10,
  APPROX_QUANTILES(fcp, 100)[OFFSET(50)] / 1000 AS q50,
  APPROX_QUANTILES(fcp, 100)[OFFSET(90)] / 1000 AS q90
FROM `blink-httparchive-research.foolip.pages_mobile_feature_*`
WHERE used IS NULL
GROUP BY feature

ORDER BY feature ASC, used DESC
```

Saved as
`` `blink-httparchive-research.foolip.pages_mobile_fcp_by_feature_use`. ``

Observations from sorting that table by high/low 10th, 50th and 90th percentile:

```
SELECT *
FROM
`blink-httparchive-research.foolip.pages_mobile_fcp_by_feature_use`
WHERE pages > 5140970 * 0.01 AND pages < 5140970 * 0.99
ORDER BY q90 DESC
```

- Percentiles for all pages are 3.5s, 6.1s and 10.6s
- Sync XHR is associated with higher FCP, especially in 90th percentile. However, it’s only used on ~2% of sites, compared to ~5% of page views.
- DocumentUnloadFired is hit more than sync XHR and gets almost as bad FCP. Looks like normally this is hit *after* FCP, but it’s still unclear why it fires more often pages with bad FCP.
- CSSSelectorWebkitMediaControls leads to higher FCP. It looks like this is because it’s used in a common WordPress style sheet, and most likely WordPress sites have a higher FCP than average. (How to confirm?)
- TransformUsesBoxSizeOnSVG leads to higher FCP and is widely used. But why? normalize.css appears in some pages, does that make things slower?
- A bunch of use counters are hit on almost all pages and aren’t interesting
    - *Not* using those very common features is associated with lower FCP, this is probably explained by small sites just using less features, avoiding <form> isn’t itself a plausible explanation.
- Using LinkRelPreload is associated with marginally lower FCP