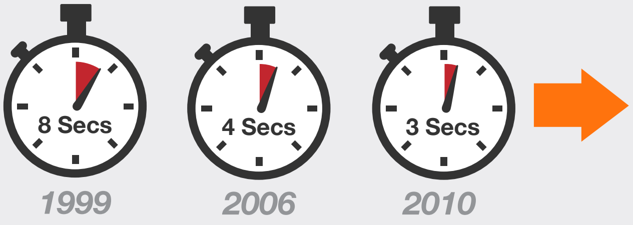
TEST HOW PAGE SPEED IMPACTS YOUR KPI'S

The Importance of Page Speed



Internet users are increasingly demanding
In 1999, the average user was willing to wait 8 seconds for a page to load. By 2010, ***57% of online shoppers said they would abandon a page after 3 seconds.***

Insights + Action = EVOLVE

The one critical aspect that's missing from the analysis above is that companies do not know how page speed **directly impacts** their businesses KPI's.

We've created a script to address this challenge. Implementing the script and using it as part of an A/B test on the Optimizely X platform will give your team the ability to see how an increase or decrease in page speed impacts KPI's that matter to you, such as transactions, revenue per visitor, or lead completions.

Example: Improving average page speed by 1 second led to a +__% in revenue per visitor for the cart page.

The Page Speed Script

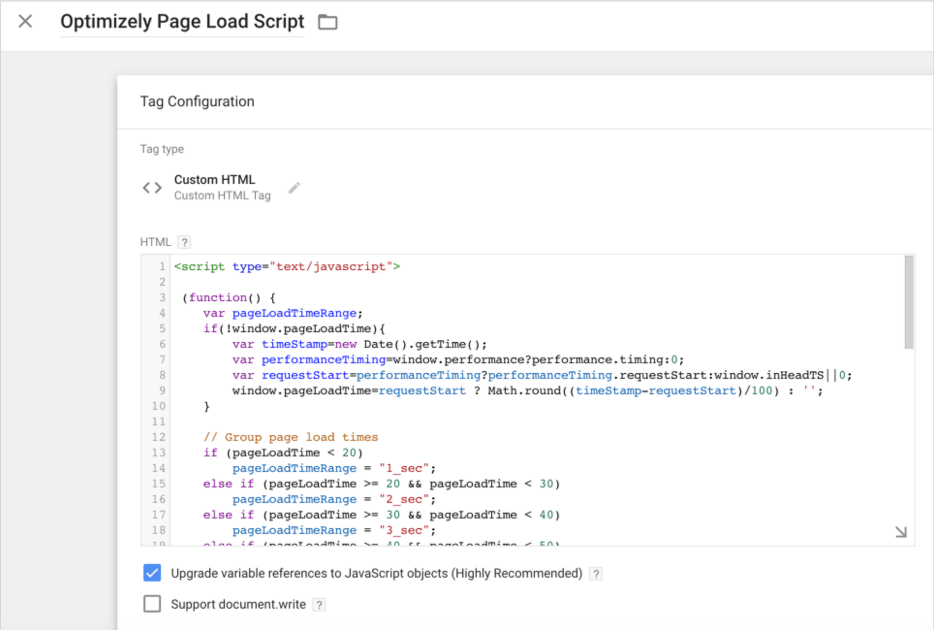
To test how page load speed impacts your business KPI's you will need to:

1. Load our script that calculates page load speed on your site;
2. Configure a custom attribute within Optimizely X to receive this page load data; and
3. Launch an A/B test within Optimizely X

Start Your Journey With Us:
☎ (888) 252-7866 sales@blastam.com | www.blastam.com
Locations: Rocklin, San Francisco, New York, Seattle, Los Angeles, Dallas, Chicago, Boston, London

1. Load the page speed script via a tag manager tool

Loading the script via Google Tag Manager



```
<script type="text/javascript">

 (function() {
    var pageLoadTimeRange;
    if(!window.pageLoadTime){
        var timeStamp=new Date().getTime();
        var performanceTiming=window.performance?performance.timing:0;
        var requestStart=performanceTiming?performanceTiming.requestStart:window.inHeadTS||0;
        window.pageLoadTime=requestStart ? Math.round((timeStamp-requestStart)/100) : '';
    }

    // Group page load times
    if (pageLoadTime < 20)
        pageLoadTimeRange = "1_sec";
    else if (pageLoadTime >= 20 && pageLoadTime < 30)
        pageLoadTimeRange = "2_sec";
    else if (pageLoadTime >= 30 && pageLoadTime < 40)
        pageLoadTimeRange = "3_sec";
    else if (pageLoadTime >= 40 && pageLoadTime < 50)
```

☑ Upgrade variable references to JavaScript objects (Highly Recommended)

□ Support document.write

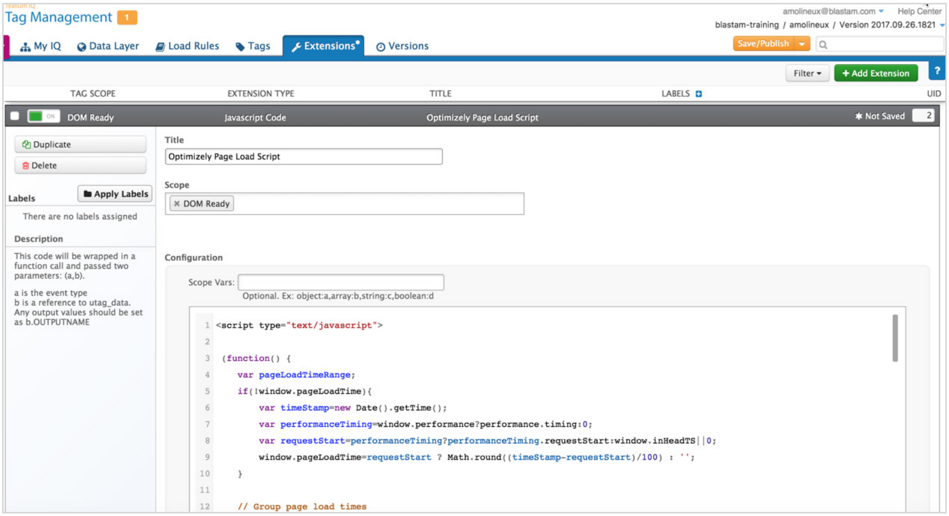
Google Tag Manager

1. Create a new custom HTML tag and name it 'Optimizely Page Load Script'
2. Copy the Blast script into the new tag
3. Setup the Trigger for the tag so that it fires on DOM-ready. The script only works effectively when run on a specific page or a specific type of page (i.e. Product Detail Pages). Update your Trigger to target the specific pages or type of pages you wish to test.
4. Preview your change, ensuring the tag fires on the expected pages
5. Publish the Tag

Loading the script via Tealium iQ

Tealium iQ

1. Create a new Javascript Code Extension
2. Scope the Extension to DOM-ready
3. Copy the Blast script into the new tag
4. The script only works effectively when run on a specific page or a specific type of page (i.e. Product Detail Pages). Update your Trigger to target the specific pages or type of pages you wish to test.
5. Publish the changes to Dev/QA and preview, ensuring the extension loads on the expected pages
6. Publish your changes to Prod



TAG SCOPE	EXTENSION TYPE	TITLE	LABELS	UID
DOM Ready	Javascript Code	Optimizely Page Load Script	✱ Not Saved	2

Title: Optimizely Page Load Script

Scope: × DOM Ready

There are no labels assigned

Description

This code will be wrapped in a function call and passed two parameters: (a,b).

a is the event type
b is a reference to utag_data.
Any output values should be set as b.OUTPUTNAME

Scope Vars:
Optional. Ex: object:a,array:b,string:c,boolean:d

```
<script type="text/javascript">

 (function() {
    var pageLoadTimeRange;
    if(!window.pageLoadTime){
        var timeStamp=new Date().getTime();
        var performanceTiming=window.performance?performance.timing:0;
        var requestStart=performanceTiming?performanceTiming.requestStart:window.inHeadTS||0;
        window.pageLoadTime=requestStart ? Math.round((timeStamp-requestStart)/100) : '';
    }

    // Group page load times
```

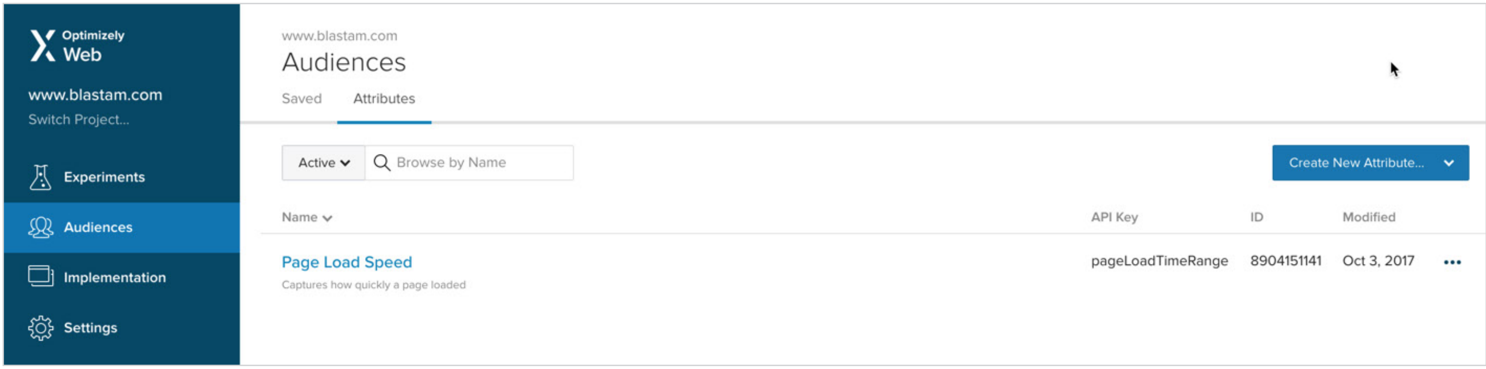
Blast’s Page Speed Script to load:

```
<script type=”text/javascript”>

 (function() {
    var pageLoadTimeRange;
    if(!window.pageLoadTime){
        var timeStamp=new Date().getTime();
        var performanceTiming=window.performance?performance.timing:0;
        var requestStart=performanceTiming?performanceTiming.requestStart:window.inHeadTS||0;
        window.pageLoadTime=requestStart ?
Math.round((timeStamp-requestStart)/100) : ‘’;
    }

    // Group page load times
    if (pageLoadTime < 20)
        pageLoadTimeRange = “1_sec”;
    else if (pageLoadTime >= 20 && pageLoadTime < 30)
        pageLoadTimeRange = “2_sec”;
    else if (pageLoadTime >= 30 && pageLoadTime < 40)
        pageLoadTimeRange = “3_sec”;
    else if (pageLoadTime >= 40 && pageLoadTime < 50)
        pageLoadTimeRange = “4_sec”;
    else if (pageLoadTime >= 50 && pageLoadTime < 60)
        pageLoadTimeRange = “5_sec”;
    else if (pageLoadTime >= 60 && pageLoadTime < 70)
        pageLoadTimeRange = “6_sec”;
    else if (pageLoadTime >= 70 && pageLoadTime < 80)
        pageLoadTimeRange = “7_sec”;
    else
        pageLoadTimeRange = “8_sec_+”;


    //Set Page Load Optimizely Custom Attribute
    window[‘optimizely’] = window[‘optimizely’] || [];
    window[‘optimizely’].push ({
        type:”user”,
        attributes: {
            pageLoadTimeRange: pageLoadTimeRange
        }
    });

})();

</script>
```

2. Create a Custom Attribute in Optimizely

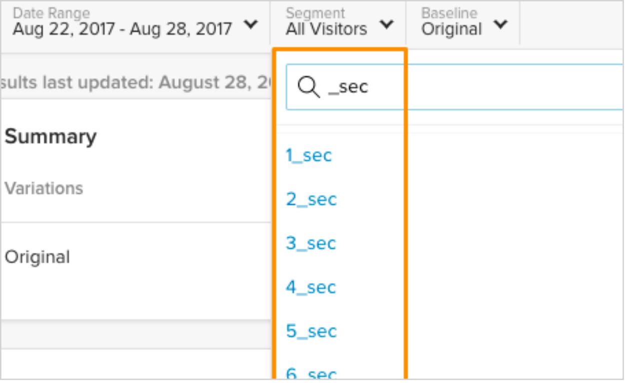
The Blast script is configured to send a custom attribute to Optimizely X. To ensure Optimizely is ready to receive this data you will need to setup a custom attribute in the Optimizely UI.

1. Within Optimizely X go to Audiences and then select Attributes
2. Select 'Create New Attribute'
3. Give this new attribute the 'Attribute Name' of Page Load Speed
4. The API Name must then be 'pageLoadTimeRange'. **Make sure the API Name is correct, including spelling and capitalization.** This API Name is referenced in the Blast script and if there are any misspellings in the Optimizely UI the script will not work.
5. Give the Attribute an appropriate Description and then hit 'Save Attribute'.
6. You're done!



Name	API Key	ID	Modified
Page Load Speed Captures how quickly a page loaded	pageLoadTimeRange	8904151141	Oct 3, 2017

If you have followed these steps then you're good to go! The script you deployed via your tag manager will calculate page load speeds for each user that is bucketed into your experiment and send this data onto the custom attribute setup in Optimizely.



Optimizely X Results Dashboards

3. Launch an A/B Test with the Page Speed Script

After you have the Blast script and accompanying custom attribute implemented, the final step is to set up your A/B test. Your team will want to test a variation that either has a faster or slower page speed than the original page.

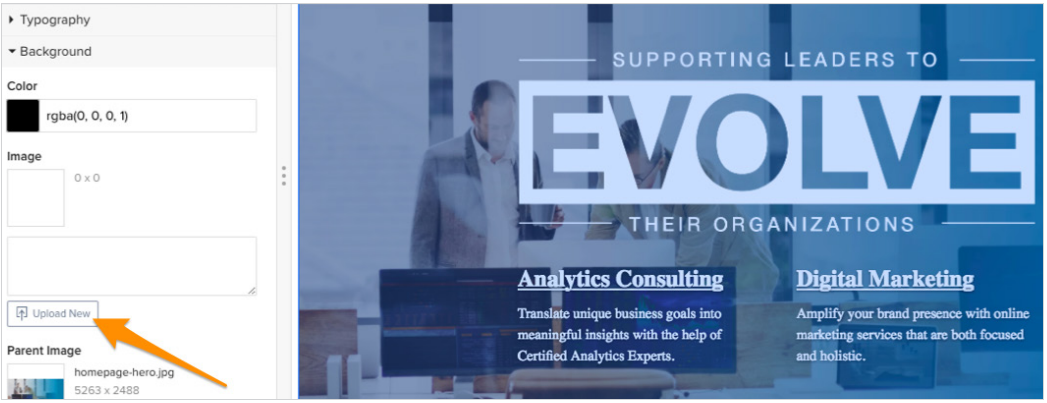
If your team is unsure which one to test, a good place to start is to understand the page load speed for your current site. While it is easy to reference Google Analytics for this information, we often find that the data included in GA's Site Speed report to be misleading. Instead, we would recommend using other tools that provide similar data, such as Pingdom or WebPageTest.

If your site already has a fast page speed (ex. 1 sec), it may be useful to test a variation with a slower page speed. So why would your team want to test a slower page? **The purpose of a page speed test is to understand IMPACT.** While the site in its current state may load quickly, there may be upcoming plans for a redesign or addition of new media assets that can potentially slow down a page. Understanding how a slower page speed affects your business KPI's allows your team to fully assess the impact of future design/UX changes, outside of whether these design changes fit the style of the site.

Alternatively, if your site has a slower page speed (ex. 6 sec) then your test variant could load faster. **While we understand generally that improving page speed improves performance, this test will allow your team to prove it.** If test results show a positive impact due to a quicker page speed, your team can leverage this data to make a case for investing resources into improving page speed for the site overall.

Test Variation: *How to Slow Down the Page*

If your team is interested in testing the original page against a slower variation, then one easy way to create this test variation is to target images that are already in use on the page. In Optimizely X's visual editor you can simply swap out current image files with larger versions of the same image.



Typography

Background

Color

rgba(0, 0, 0, 1)

Image

0 x 0

Upload New

Parent Image

homepage-hero.jpg

5263 x 2488

SUPPORTING LEADERS TO

EVOLVE

THEIR ORGANIZATIONS

Analytics Consulting

Translate unique business goals into meaningful insights with the help of Certified Analytics Experts.

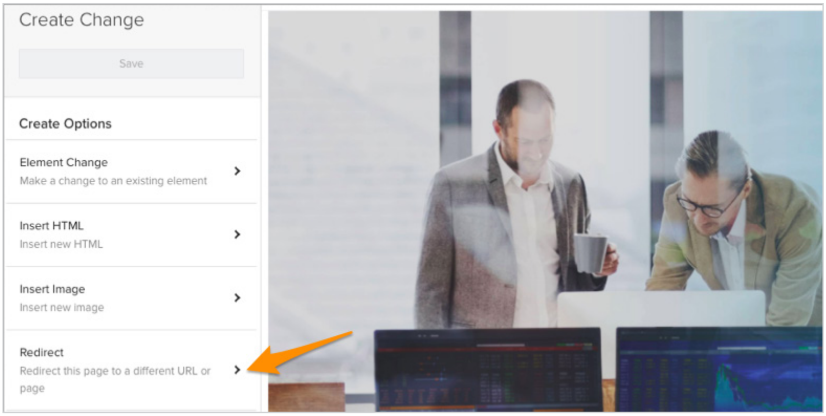
Digital Marketing

Amplify your brand presence with online marketing services that are both focused and holistic.

Test Variation: *How to Speed Up the Page*

Speeding up the variation page can be accomplished in a few different ways. If you utilize the page speed tools mentioned above (ex. Pingdom), you should receive specific recommendations on how to accomplish this, such as:

- Compressing Images
- Leveraging Browser Caching
- Minify Code
- Content Delivery Network (CDN)



Create Change

Save

Create Options

Element Change
Make a change to an existing element

Insert HTML
Insert new HTML

Insert Image
Insert new image

Redirect
Redirect this page to a different URL or page

Similar to slowing down a page, if the targeted page contains images, this is an easy way to impact page speed. In this case, your team would focus on compressing these image files. You can utilize tools, like ImageOptim, to compress images and then upload the new file to the variation via Optimizely X's visual editor.

If your team is looking to implement other recommendations (ex. leverage browser caching), they can build out a separate test page with recommendations implemented, and then use Optimizely X to serve a redirect to this new page.

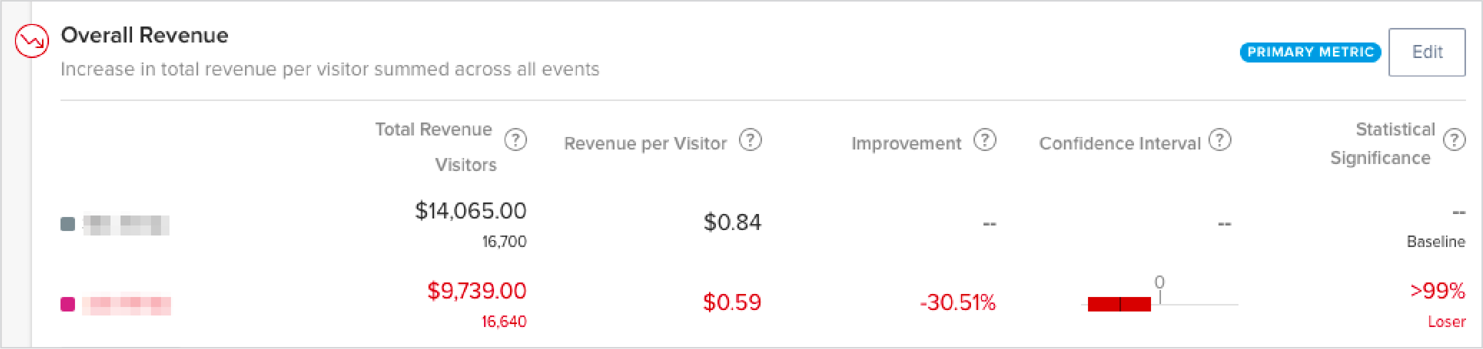
Start Your Journey With Us:

☎ (888) 252-7866 sales@blastam.com | www.blastam.com

Locations: Rocklin, San Francisco, New York, Seattle, Los Angeles, Dallas, Chicago, Boston, London

Track Business KPI's

As part of the test buildout you'll want to ensure that your businesses KPI's are tracked properly. Specifically, if your team is interested in revenue, you'll want to have Optimizely X's revenue tracking in place. This will allow you to see if the test has a significant impact on revenue per visitor.



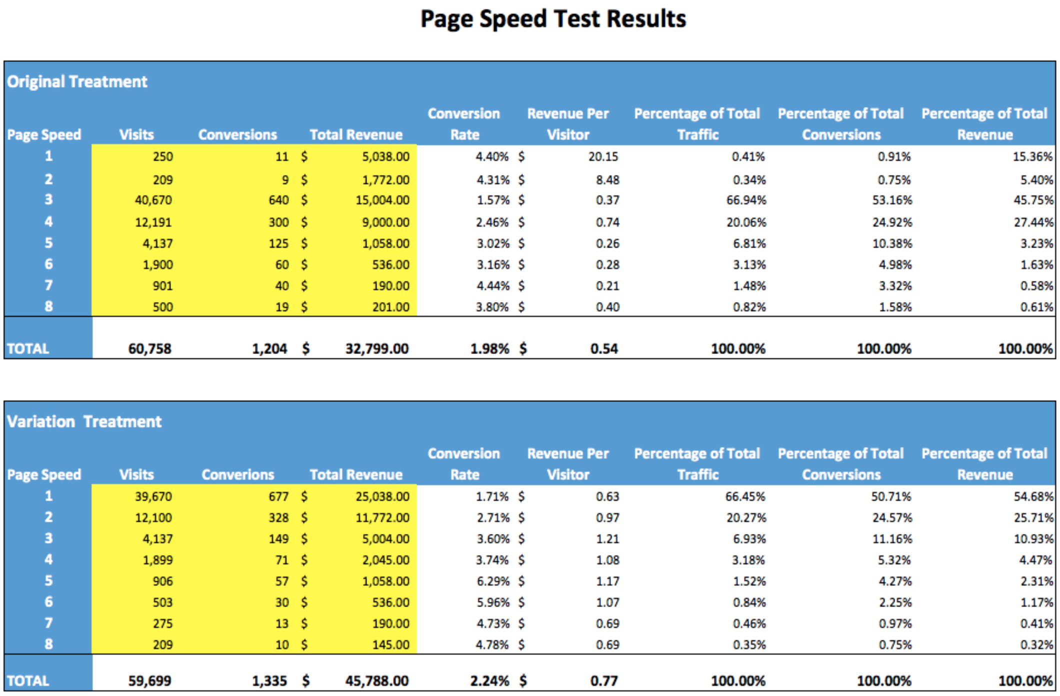
Overall Revenue

Increase in total revenue per visitor summed across all events

	Total Revenue Visitors	Revenue per Visitor	Improvement	Confidence Interval	Statistical Significance
[redacted]	$14,065.00 16,700	$0.84	--	--	-- Baseline
[redacted]	$9,739.00 16,640	$0.59	-30.51%	0	>99% Loser

If it's not possible to track revenue within Optimizely X, or your team prefers to pull revenue data into Google Analytics, then you'll want to make sure 1) the test is properly integrated with GA, and 2) you use our free Revenue Per Visitor Calculator to determine whether there is a statistically significant impact.

Finally, your team can utilize our Page Speed Test Results template to calculate the average page load speed for each variation and analyze the percentage change for metrics that are important to your business.



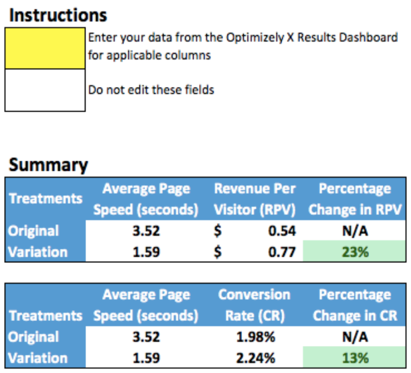
Page Speed Test Results

Original Treatment

Page Speed	Visits	Conversions	Total Revenue	Conversion Rate	Revenue Per Visitor	Percentage of Total Traffic	Percentage of Total Conversions	Percentage of Total Revenue
1	250	11	$ 5,038.00	4.40%	$ 20.15	0.41%	0.91%	15.36%
2	209	9	$ 1,772.00	4.31%	$ 8.48	0.34%	0.75%	5.40%
3	40,670	640	$ 15,004.00	1.57%	$ 0.37	66.94%	53.16%	45.75%
4	12,191	300	$ 9,000.00	2.46%	$ 0.74	20.06%	24.92%	27.44%
5	4,137	125	$ 1,058.00	3.02%	$ 0.26	6.81%	10.38%	3.23%
6	1,900	60	$ 536.00	3.16%	$ 0.28	3.13%	4.98%	1.63%
7	901	40	$ 190.00	4.44%	$ 0.21	1.48%	3.32%	0.58%
8	500	19	$ 201.00	3.80%	$ 0.40	0.82%	1.58%	0.61%
TOTAL	60,758	1,204	$ 32,799.00	1.98%	$ 0.54	100.00%	100.00%	100.00%

Variation Treatment

Page Speed	Visits	Converions	Total Revenue	Conversion Rate	Revenue Per Visitor	Percentage of Total Traffic	Percentage of Total Conversions	Percentage of Total Revenue
1	39,670	677	$ 25,038.00	1.71%	$ 0.63	66.45%	50.71%	54.68%
2	12,100	328	$ 11,772.00	2.71%	$ 0.97	20.27%	24.57%	25.71%
3	4,137	149	$ 5,004.00	3.60%	$ 1.21	6.93%	11.16%	10.93%
4	1,899	71	$ 2,045.00	3.74%	$ 1.08	3.18%	5.32%	4.47%
5	906	57	$ 1,058.00	6.29%	$ 1.17	1.52%	4.27%	2.31%
6	503	30	$ 536.00	5.96%	$ 1.07	0.84%	2.25%	1.17%
7	275	13	$ 190.00	4.73%	$ 0.69	0.46%	0.97%	0.41%
8	209	10	$ 145.00	4.78%	$ 0.69	0.35%	0.75%	0.32%
TOTAL	59,699	1,335	$ 45,788.00	2.24%	$ 0.77	100.00%	100.00%	100.00%



Instructions

🟨	Enter your data from the Optimizely X Results Dashboard for applicable columns
□	Do not edit these fields

Summary

Treatments	Average Page Speed (seconds)	Revenue Per Visitor (RPV)	Percentage Change in RPV
Original	3.52	$ 0.54	N/A
Variation	1.59	$ 0.77	23%

Treatments	Average Page Speed (seconds)	Conversion Rate (CR)	Percentage Change in CR
Original	3.52	1.98%	N/A
Variation	1.59	2.24%	13%

Start Your Journey With Us:

☎ (888) 252-7866 sales@blastam.com | www.blastam.com

Locations: Rocklin, San Francisco, New York, Seattle, Los Angeles, Dallas, Chicago, Boston, London