Multimedia ANOVA Testing for Website Conversions

Carsten Lange, Cal Poly, Pomona

1 Introduction

Imagine you are in charge of optimizing a company's website. The goal is simple: increase conversions — defined here as users clicking on the ordering form. The marketing team proposes a solution: **Enhance the website with either audio or video content to better engage visitors.**
But will these multimedia upgrades actually drive more clicks, or will the effort fall flat?

To answer this question, the company decides to run a controlled experiment. Every website visitor is randomly redirected to one of three versions of the site:

- the original (**Control**),
- one enhanced with **Audio**, or
- one enhanced with **Video**.

Afterward, the team tracks hourly conversions for each of the three scenarios (*Control*, *Audio*, *Video*). The result generated a randomized dataset, ideal for testing whether the *Audio*, *Video* treatments have a measurable impact on user behavior.

In this article, we will walk through a simulated version of this marketing experiment using artificial data adapted from the well-known Palmer Penguins dataset. The setup mirrors how real companies might structure and evaluate a digital content strategy.

2 Experimental Design

To assess the impact of multimedia content on user behavior, the company conducted a randomized test involving three website variants:

- **Control**: the original site with no enhancements

- **Audio**: the same site augmented with audio content
- **Video**: the site upgraded with embedded video content

Every visitor arriving at the website was randomly assigned to one of these three groups. The primary metric of **conversion** was defined as whether users clicked on the ordering form. Each record reflects one hour of user data for the three scenarios leading to a total of 342 observations.

3 Data Generation and Structure

To simulate this experiment, a proxy dataset was created using the well-known **Palmer Penguins** dataset. In this synthetic version:

- The variable **flipper length** was reinterpreted to represent the **number of conversions** within a one-hour period.
- The **species** of the penguins was repurposed as the **treatment group** (**Control**, **Audio**, or **Video**).

Although artificial, this approach allows us to explore realistic patterns in conversion behavior under different digital content strategies.

The code below shows how the Palmer Penguin dataset was used to generate the dataset and which R libraries were used:

```
library(knitr)
library(kableExtra)
library(tidyverse)
library(palmerpenguins)

DataWeb=penguins |>
       select(Conversions=flipper_length_mm, Treatment=species) |>
       drop_na()
levels(DataWeb$Treatment)=c("Control", "Audio", "Video")
```

Each observation shows for different web design scenarios how many conversions resulted within 1 hour. The first 6 records are shown n Table **??**:

```
set.seed(123)
kable(sample_n(DataWeb,6), caption = "First Six Observations from the Dataset")
```

Table 1: First Six Observations from the Dataset

Conversions	Treatment
215	Video
198	Control
216	Video
201	Audio
189	Control
195	Audio

4 Group and Overall Means

The code below generates group means for the *Control*, *Audio*, and *Video* groups, together with the overall mean (*GrandMean*) across all groups and outputs the results in Table **??**:

```
GrandMean  = mean(DataWeb$Conversions)

# Conditional Means by Treatment
MeansByTreatment <- DataWeb %>%
  group_by(Treatment) %>%
  summarise(Mean = mean(Conversions), N=n())|>
  bind_rows(tibble(
    Treatment = "Overall",
    Mean = GrandMean,
    N = 342
  ))
N=342
# View results
kbl(MeansByTreatment, caption="Overall Mean and Means  of the Treatments")|>
   kable_styling(full_width = FALSE, position = "center")%>%    scroll_box(width = "400px")
```

At a glance, both multimedia-enhanced designs resulted in a higher average number of conversions per hour compared to the control group.

Notably, the *Video* variant had the highest mean, with an average of 217.19 conversions — nearly 27 more than the Control.

The Audio variant also outperformed the Control, though with a smaller gap.

The Control group had the lowest mean but the largest sample size.

Table 2

Table 3: Overall Mean and Means of the Treatments

Treatment	Mean	N
Control	189.9536	151
Audio	195.8235	68
Video	217.1870	123
Overall	200.9152	342

The Grand Mean of 200.92 provides a useful benchmark for understanding overall performance across all treatments, especially when considering statistical comparisons or variance analysis.

Figure **??** provides additional evidence for this impression:

```
ggplot(DataWeb, aes(y=0, x = Conversions, color = Treatment)) +
  geom_jitter(width = 0.1) +
  geom_vline(xintercept = MeansByTreatment[[1,2]], color = "red",linewidth=0.7) +
  geom_vline(xintercept = MeansByTreatment[[2,2]], color = "green",linewidth=0.7) +
  geom_vline(xintercept = MeansByTreatment[[3,2]], color = "blue",linewidth=0.7) +
  geom_vline(xintercept = mean(DataWeb$Conversions), color = "black",linewidth=1)+
  labs(y = "") +              # Set y-axis title
  scale_y_continuous(breaks = NULL)+
  annotate("text", x = MeansByTreatment[[1,2]], y = 0.37,
    label = round(MeansByTreatment[[1,2]],2), color = "red", angle = 90,
    vjust = -0.5, size = 3)+
  annotate("text", x = MeansByTreatment[[2,2]], y = 0.37,
    label = round(MeansByTreatment[[2,2]],2), color = "green", angle = 90,
    vjust = -0.5, size = 3) +
  annotate("text", x = MeansByTreatment[[3,2]], y = 0.37,
    label = round(MeansByTreatment[[3,2]],2), color = "blue", angle = 90,
    vjust = -0.5, size = 3) +
  annotate("text", x = mean(DataWeb$Conversions), y = 0.37,
    label = round(mean(DataWeb$Conversions),2), color = "black", angle = 90, vjust = -0.5, si
```

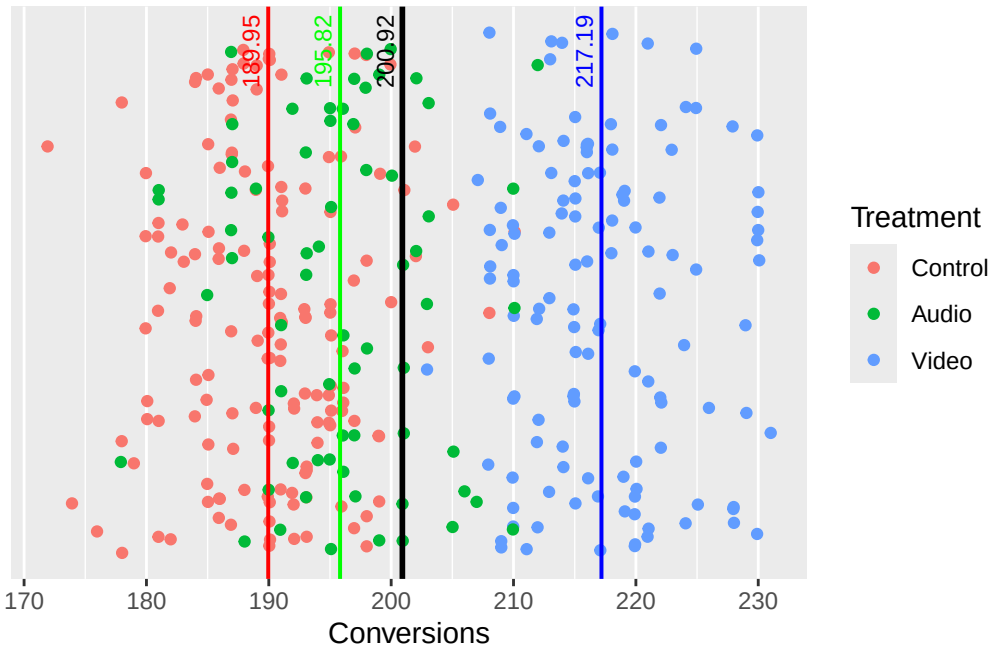

Figure 1: Treatment Means and Grand Means with Data Distribution

Can we scientifically determine whether these observed differences are statistically significant?

5 Why Not Use Pairwise A/B Tests?

A tempting idea in this setting is to run multiple pairwise **A/B Tests** to compare each design against the *Control*:

- **Audio** vs. **Control**
- **Video** vs. **Control**

While intuitive, this approach introduces big statistical concern: increased risk of false positives (*Type I Error*). That is, by running multiple comparisons independently, we raise the chance of incorrectly concluding that a difference exists when it does not.

Even if each individual test maintains a 5% error rate ($\alpha = 0.05$), the combined probability of making at least one false discovery across two tests rises to:

$$1 - (1 - 0.05)^2 = 0.0975 \approx 10\%$$

With more variations, the problem compounds quickly. For example, if we had four treatments in addition to the *Control*, the chance of a false positive somewhere among those comparisons would rise to nearly 20%:

$$1 - (1 - 0.05)^2 = 0.1855 \approx 20\%$$

Clearly, this inflation of error makes multiple pairwise
A/B Tests **statistically unreliable**.

A Better Alternative:
Testing all treatments at once instead of running several separate tests. This is where *Analysis of Variance* or short *ANOVA* comes in:

- *ANOVA* allows to test one or more groups of variables simultaneously
- *ANOVA* avoids the pitfalls of multiple testing

However, if the ANOVA result is significant, we know only that at least one, but not which one(s) of the variables are significant. Afterward, more testing is needed to find out which variable is significant.

6 Regression and ANOVA

One way to approach *ANOVA* is through regression analysis. The idea is simple: if any of the treatment types (*Audio* or *Video*) has an impact on conversions, then including these variables in a regression model should improve its explanatory power.

This is exactly what we explore below. We fit a linear model with **Treatment** as a categorical predictor of **Conversions**:

$$Conv_i = \beta_1 Audio_i + \beta_2 Video_i + \beta_0$$

Here, $Audio_i$ and $Video_i$ are dummy variables for group membership. The **Control group** is omitted and serves as the **reference category**.

The code below runs the analysis in *R*, stores the model in the object `ModelLM`, and displays the summary output in Table **??** with `summary()`.

```
ModelLM = lm(Conversions ~ Treatment, data = DataWeb)

summary(ModelLM) |>
broom::tidy() |>
kable(caption="Output from the Linear Regression") |>
```