

# Duke Wireless Data

*Applications in Network Infrastructure & Security*

## Undergraduates

*Eidan Jacob | ej68@duke.edu*

*Justina Zou | jmz15@duke.edu*

*Mason Simon | mas166@duke.edu*

## Project Manager

*Alexander Mang | amm194@duke.edu*

## Project Leads

*Richard Biever*

*Jessica Repka*

*Eric Hope*

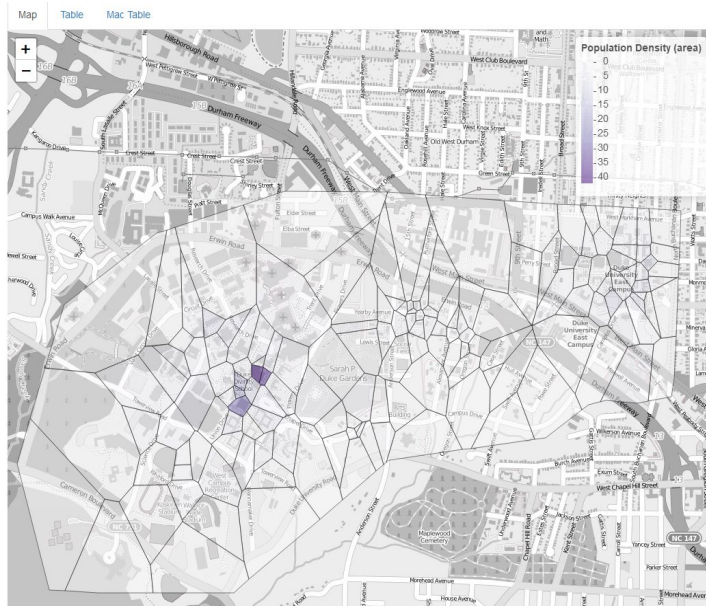
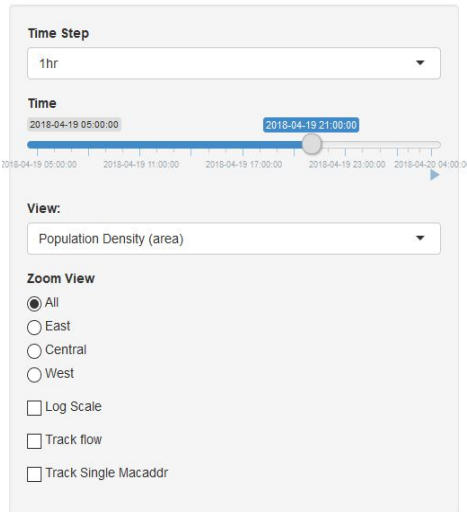
## Overview

Duke's wireless network records millions of interactions with wireless devices each day. We created tools that track these devices as they connect to different wireless access points throughout campus.

Our maps were created using R Shiny and the Leaflet package.

# Crowd Movement

## Duke Wireless Data

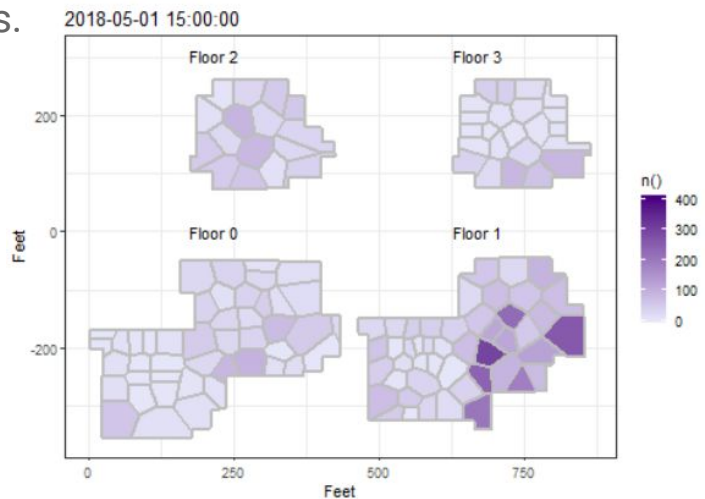


## Campus Population Density Heatmap (left)

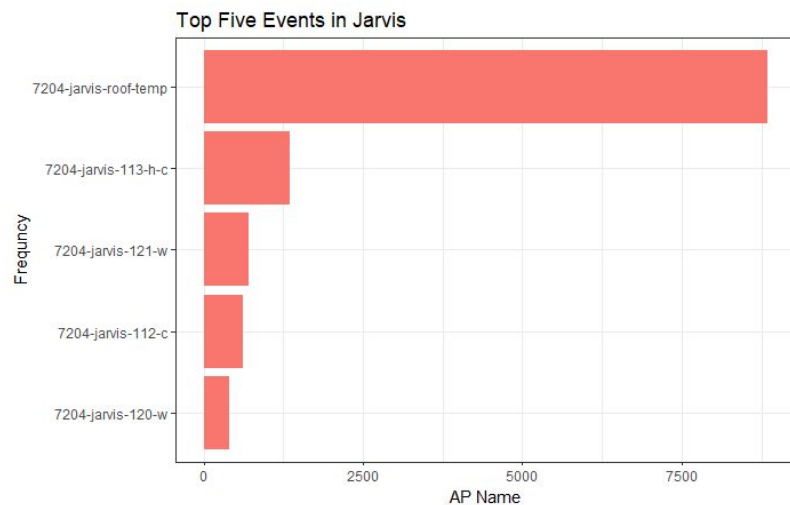
We matched wireless access points to one of 100 locations corresponding to buildings on campus. We generated Voronoi cells based on the building locations and colored them according to a chosen measure of network usage. The map is fully interactive (zoom, pan, etc.) and can be used to display movement lines of devices to examine popular routes.

## Perkins Library (right)

We created an application that uses the shape of a particular building to visualize the relative utilization of wireless access points within that building over time. A public-facing version of this application can be used in real-time by students and faculty to easily avoid areas where internet access may be slowed by heavy use.



# Wireless Infrastructure & Error Detection



This chart shows the top five most “active” access points in Jarvis. The access point on the roof of Jarvis has the most registered events. This, however, can be explained by Jarvis’s geographic location. The East-West bus drives past the dorm, allowing passengers to briefly connect to the access point.

## Pathing

We examined how devices connected to different access points. For example, issues such as frequently switching access points despite no indication of movement could indicate the access points at that location have trouble keeping a device connected.

## Speeding

We wrote scripts that automatically detect when devices jump from two locations that are, for example, across campus in one second. It also detects whether a device appears in two different locations at the same time—an indication of errors either in the data or a suspicious device.