Church Challenge

Course DS 460

Landon Davis

Using the church challenge dataset I was able to pull data from both Utah and Georgia. Looking at the differences and similarities of people who went to those buildings such as how long they where there and what other locations or brands they went after. This data then was used to help answer these questions.

Question 1

What differences are there between iPhone and Android users when comparing visits to The Church of Jesus Christ buildings of Latter-day Saints in Utah and Georgia?

I was able to find that more people in Utah visit Church of Jesus Christ of Latter Day Saint bulldings then in Georgia. However in Georgia more iOS devices went to buildings then android and it is almost double that of android devices. Compared to in Utah there are more android devices going to church buildings but it is only slightly more then iOS by a couple devices.

Code Snippet:

```
#Creates a chart using altair. This Chart displays the diffrences between diffrent devices users and there visits to LDS church buildings.
visit = Poi_devices_visists.sample(750000)
alt.data_transformers.enable('json')
alt.Chart(visit).mark_bar().encode(
    alt.X('device_type', title = "Device"),
    alt.Y('visits_by_day', title= "Amount of Visits to Church Building Daily"),
    color = "device_type",
    column='region'
).transform_filter(alt.FieldEqualPredicate(field='location_name', equal="The Church of Jesus Christ of Latter day Saints" )).properties(title=  "Daily Visits to a Church B
```

Visualization:

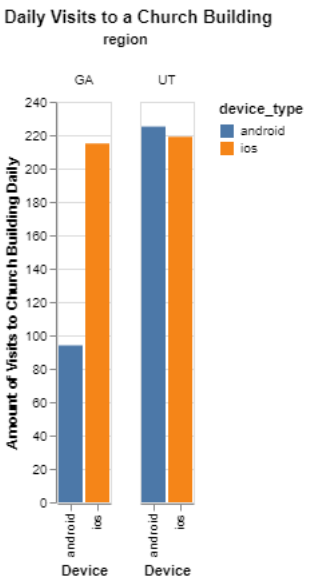


Chart:

placekey	location Name	city	region	postal_code	startDate	endDate	device_type	amount_of_that_device
zzw-224@8g6-y2j-yy9	Ravi Zacharias International Ministries	Alpharetta	GA	30022	2021-10-01 00:00:00+00:00	2021-11-01 00:00:00+00:00	android	0
zzw-224@8g6-y2j-yy9	Ravi Zacharias International Ministries	Alpharetta	GA	30022	2021-10-01 00:00:00+00:00	2021-11-01 00:00:00+00:00	ios	0
zzw-224@8g6-y2j-yy9	Ravi Zacharias International Ministries	Alpharetta	GA	30022	2021-12-01 00:00:00+00:00	2022-01-01 00:00:00+00:00	android	4
zzw-224@8g6-y2j-yy9	Ravi Zacharias International Ministries	Alpharetta	GA	30022	2021-12-01 00:00:00+00:00	2022-01-01 00:00:00+00:00	ios	0
zzw-224@8g6-y2j-yy9	Ravi Zacharias International Ministries	Alpharetta	GA	30022	2021-10-01 00:00:00+00:00	2021-11-01 00:00:00	+00:00	android

Question 2

Compare hourly usage patterns between The Church of Jesus Christ of Latter-day Saints and the other churches in each state.

From this chart we can see that there is a larger variety of church's in Georgia then there is in Utah. We can also see that hourly usage patterns vary greatly but are all very similar with 3 or 4 chruch's with large amounts of hourly usage patterns. We can see The Church of Jesus Christ of

Ladder-day Saints has the largest hourly usage in Utah. While Belmont baptist church has the largest in Georgia and when you find the mean in Georgia other church's would have a higher hourly usage rate.

Code Snippet:

```
#Creates a chart using altair. This chart displays the hourly usage patterns of churchs in each state.
Poi_hour= Poi_hours.sample(100)
alt.data_transformers.enable('json')
alt.Chart(Poi_hour).mark_bar().encode(
    alt.X('location_name', title = "Church Name"),
    alt.Y('popularity_by_hour',title= "Amount of People Per Hour"),
    color = "region" ,
).properties(title= "Hourly Usage")
```

Visualization:

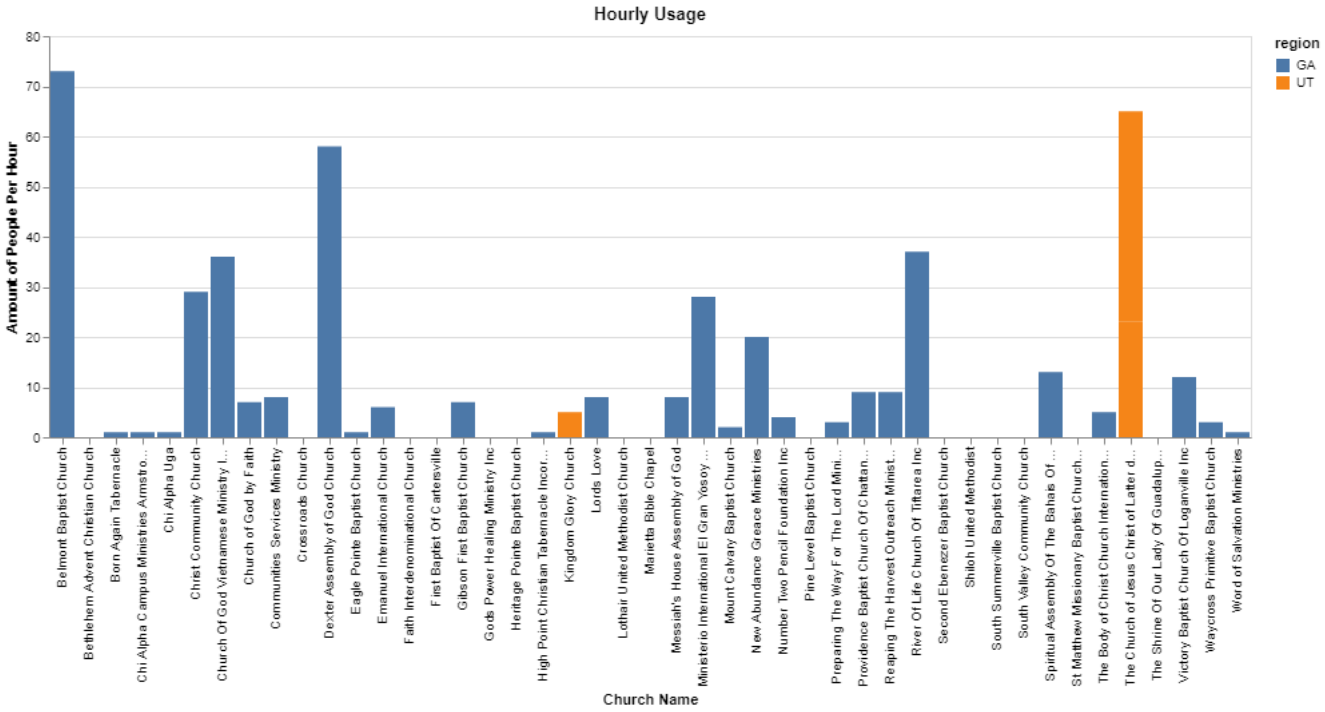


Chart:

placekey	location Name	city	region	postal_code	startDate	endDate	hour	Popularity_by_hour
zw-224@8g6-y2j-yy9	Ravi Zacharias International Ministries	Alpharetta	GA	30022	2021-10-01 00:00:00+00:00	2021-11-01 00:00:00+00:00	1	0.0
zzw-224@8g6-y2j-yy9	Ravi Zacharias International Ministries	Alpharetta	GA	30022	2021-10-01 00:00:00+00:00	2021-11-01 00:00:00+00:00	2	0.0
zzw-224@8g6-y2j-yy9	Ravi Zacharias International Ministries	Alpharetta	GA	30022	2021-10-01 00:00:00+00:00	2021-11-01 00:00:00+00:00	3	0.0
zzw-224@8g6-y2j-yy9	Ravi Zacharias International Ministries	Alpharetta	GA	30022	2021-10-01 00:00:00+00:00	2021-11-01 00:00:00+00:00	4	0.0

placekey	location Name	city	region	postal_code	startDate	endDate	hour	Popularity_by_hour
zzw-224@8g6-y2j-yy9	Ravi Zacharias International Ministries	Alpharetta	GA	30022	2021-10-01 00:00:00+00:00	2021-11-01 00:00:00+00:00	5	0.0

Question 3

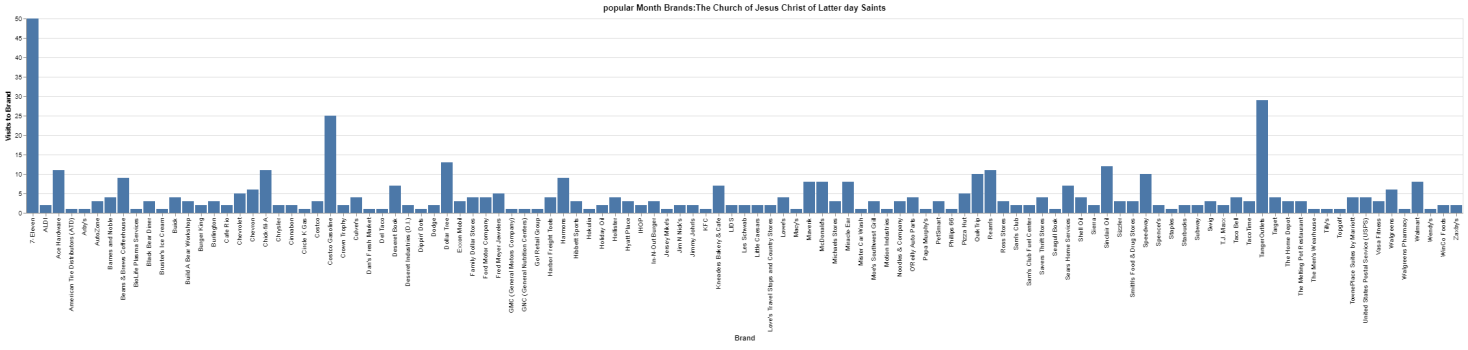
Contrast the related_same_day_brand brands between those who visit the Church of Jesus Christ of Latter-day Saints and those who visit other churches.

When looking at the same day brands of those who visit The Church of Jesus Christ of Latter-day Saints and other Churches. We can see simularities and some differences. For example the brands most visisted after a church building is a gas station. Then in second are grocery stores and finally fastfood and other brands. These brands are slightly different such as after visiting a latter-day saint building the most popular gas station was 7-Eleven. While for other churches it was Maverick. This could be due to there location in different states and overall popularity among brands. This help us to see what they needed to do next or what was important for them to buy after visiting a church building.

Code Snippet:

```
#Creates a chart using altair.This chart shows all the same day brand visits to LDS buildings.
alt.data_transformers.enable('json')
alt.Chart(popular_brands_day).mark_bar().encode(
    alt.X('related_same_day_brand', title= "Brand"),
    alt.Y('same_day_brand_visits', title= "Visits to Brand")).properties(title = "popular Month Brands").transform_filter(alt.FieldEqualPredicate(field='location_name', eq
#Creates a chart using altair.This chart shows all the same day brand visits to churchs.
    alt.data_transformers.enable('json')
alt.Chart(popular_brands_day).mark_bar().encode(
    alt.X('related_same_day_brand', title= "Brand"),
    alt.Y('same_day_brand_visits', title= "Visits to Brand")).properties(title = "popular Month Brands")).properties(title = "popular Month Brands")
```

Visualizations:



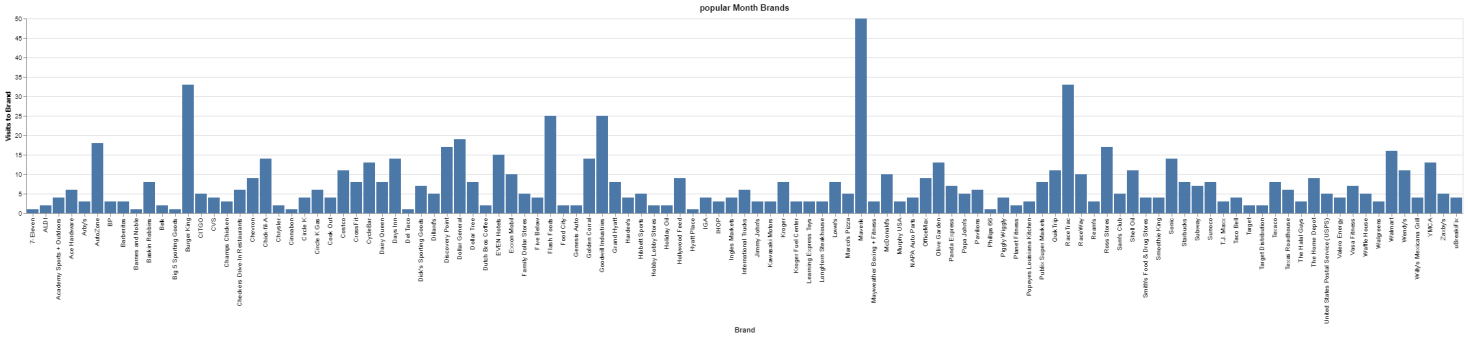


Chart:

placekey	location Name	city	region	postal_code	startDate	endDate	related_same_day_brand	same_day_
zzw-224@8g6-y2j-yy9	Ravi Zacharias International Ministries	Alpharetta	GA	30022	2021-10-01 00:00:00+00:00	2021-11-01 00:00:00+00:00	Dunkin'	
zzw-224@8g6-y2j-yy9	Ravi Zacharias International Ministries	Alpharetta	GA	30022	2021-10-01 00:00:00+00:00	2021-11-01 00:00:00+00:00	Wendy's	
zzw-224@8g6-y2j-yy9	Ravi Zacharias International Ministries	Alpharetta	GA	30022	2021-10-01 00:00:00+00:00	2021-11-01 00:00:00+00:00	BP	
zzw-224@8g6-y2j-yy9	Ravi Zacharias International Ministries	Alpharetta	GA	30022	2021-10-01 00:00:00+00:00	2021-11-01 00:00:00+00:00	Dunkin'	
zzw-224@8g6-y2j-yy9	Ravi Zacharias International Ministries	Alpharetta	GA	30022	2021-10-01 00:00:00+00:00	2021-11-01 00:00:00+00:00	Wendy's	

APPENDIX A (Additional Python Code)

```
#Import Python Libraries.
import pandas as pd
import os
import pyarrow as pa
import pyarrow.parquet as pap
import altair as alt

#Takes the location of parquet files and lists them out.
path_dir = r"C:\Users\lando\OneDrive\Documents\WO1_ds460\parquet/"
for files in os.listdir(path_dir):
        print(files)

#Reads dwell_times parquet file and creats a dataframe. Then renames the value column to something easier.
bucketed_dwell_times = pd.read_parquet( r"C:\Users\lando\OneDrive\Documents\WO1_ds460\parquet/bucketed_dwell_times.parquet", engine="pyarrow")
bucket_df = pd.DataFrame(bucketed_dwell_times)
buckets_df = bucket_df.rename(columns={"value": "number_of_ws_visits"})
#Reads device_type paquet file and creats a dataframe. Then renames the value column to something easier.
device_type = pd.read_parquet( r"C:\Users\lando\OneDrive\Documents\WO1_ds460\parquet/device_type.parquet", engine="pyarrow")
device_df = pd.DataFrame(device_type)
device_df = device_df.rename(columns={'value': 'amount_of_that_device'})
#Reads open_hour paquet file and creats a dataframe. Then renames The value column to something easier.
open_hours = pd.read_parquet( r"C:\Users\lando\OneDrive\Documents\WO1_ds460\parquet/open_hours.parquet", engine="pyarrow")
open_hours_df = pd.DataFrame(open_hours)
open_hours_df = open_hours_df.rename(columns={'value':'hours_open', 'open_hours': "day_of_Week_open"})
#Reads poi paquet file and creats a dataframe. Then drops extra columns not being used.
poi = pd.read_parquet( r"C:\Users\lando\OneDrive\Documents\WO1_ds460\parquet/poi.parquet", engine="pyarrow")
poi_df = pd.DataFrame(poi)
poi_df =  poi_df.drop(columns=["safegraph_brand_ids", "brands", "raw_visit_counts", "normalized_visits_by_region_naics_visits", "poi_cbg", "norma
"date_range_start", "date_range_end","opened_on", "closed_on", "category_tags", "sub_category", "naics_code", "latitude", "longitude", "street_ad
#Displays first five rows of dataframe
poi_df.head()

#Reads popularity_by_day paquet file and creats a dataframe. Then renames The value column to something easier.
popularity_by_day = pd.read_parquet( r"C:\Users\lando\OneDrive\Documents\WO1_ds460\parquet/popularity_by_day.parquet", engine="pyarrow")
popularity_by_day_df = pd.DataFrame(popularity_by_day)
popularity_by_day_df = popularity_by_day_df.rename(columns={'value': "visits_by_day"})

#Reads popularity_by_hour paquet file and creats a dataframe.
popularity_by_hour = pd.read_parquet( r"C:\Users\lando\OneDrive\Documents\WO1_ds460\parquet/popularity_by_hour.parquet", engine="pyarrow")
popularity_by_hour_df = pd.DataFrame(popularity_by_hour)

#Reads popularity_by_day paquet file and creats a dataframe. Then renames The value column to something easier.
related_same_day_brand = pd.read_parquet( r"C:\Users\lando\OneDrive\Documents\WO1_ds460\parquet/related_same_day_brand.parquet", engine="pyarrow"
related_same_day_brand_df = pd.DataFrame(related_same_day_brand)
related_same_day_brand_df = related_same_day_brand_df.rename(columns={'value': "same_day_brand_visits"})

#Reads same_month_brand paquet file and creats a dataframe. Then renames The value column to something easier.
related_same_month_brand = pd.read_parquet( r"C:\Users\lando\OneDrive\Documents\WO1_ds460\parquet/related_same_month_brand.parquet", engine="pyar
related_same_month_brand_df= pd.DataFrame(related_same_month_brand)
related_same_month_brand_df = related_same_month_brand_df.rename(columns={'value': "popularity_of_same_month_brand"})
#Reads paquet file and creats a dataframe. Then renames The value column to something easier.
visitor_by_country_of_origin = pd.read_parquet( r"C:\Users\lando\OneDrive\Documents\WO1_ds460\parquet/visitor_country_of_origin.parquet", engine=
visitor_by_country_of_origin_df = pd.DataFrame(visitor_by_country_of_origin)
visitor_by_country_of_origin_df = visitor_by_country_of_origin_df.rename(columns={'value': "amount_of_visitors_origin_country"})
#Reads file and creats a dataframe.
visitor_daytime_cbgs = pd.read_parquet( r"C:\Users\lando\OneDrive\Documents\WO1_ds460\parquet/visitor_daytime_cbgs.parquet", engine="pyarrow")
visitor_daytime_cbgs_df = pd.DataFrame(visitor_daytime_cbgs)
#Reads paquet file and creats a dataframe.
visitor_home_aggregation = pd.read_parquet( r"C:\Users\lando\OneDrive\Documents\WO1_ds460\parquet/visitor_home_aggregation.parquet", engine="pyar
visitor_home_aggregation_df = device = pd.DataFrame(visitor_home_aggregation)
#Reads paquet file and creats a dataframe.
visitor_home_cbgs = pd.read_parquet( r"C:\Users\lando\OneDrive\Documents\WO1_ds460\parquet/visitor_home_cbgs.parquet", engine="pyarrow")
visitor_home_cbgs_df = pd.DataFrame(visitor_home_cbgs)
#Reads paquet file and creats a dataframe.
visits_by_day = pd.read_parquet( r"C:\Users\lando\OneDrive\Documents\WO1_ds460\parquet/visits_by_day.parquet", engine="pyarrow")
visits_by_day_df = pd.DataFrame(visits_by_day)

#Merge two dataframes together. Then shows first 5 rows.
Poi_devices= pd.merge(poi_df, device_df, on="placekey")
poi_devices.head()

#Merge two dataframes together. Then shows first 5 rows.
Poi_devices_visists = pd.merge(Poi_devices, visits_by_day_df, on = ["placekey", "startDate", 'endDate'])
```

```
poi_device_visits.head()

#Merge two dataframes together. Then shows first 5 rows.
Poi_hours = pd.merge(poi_df, popularity_by_hour_df, on = 'placekey')
poi_hour.head()

#Merge two dataframes together. Then shows first 5 rows.
Poi_devices_brands_day = pd.merge(Poi_devices, related_same_day_brand_df, on= ["placekey", "startDate", "endDate"])
poi_devices_brands_day.head()

#Merge two dataframes together. Then shows first 5 rows.
Poi_devices_brands_month = pd.merge(Poi_devices, related_same_month_brand_df, on = ["placekey", "startDate", 'endDate'])
poi_devices_brnads_month.head()

#Creates a chart using altair. This chart displays the hourly usage patterns of churchs in each state.
Poi_hour= Poi_hours.sample(100)
alt.data_transformers.enable('json')
alt.Chart(Poi_hour).mark_bar().encode(
    alt.X('location_name', title = "Church Name"),
    alt.Y('popularity_by_hour',title= "Amount of People Per Hour"),
    color = "region" ,
).properties(title= "Hourly Usage")
#Creates a chart using altair. This Chart displays the diffrences between diffrent devices users and there visits to LDS church buildings.
visit = Poi_devices_visists.sample(750000)
alt.data_transformers.enable('json')
alt.Chart(visit).mark_bar().encode(
    alt.X('device_type', title = "Device"),
    alt.Y('visits_by_day', title= "Amount of Visits to Church Building Daily"),
    color = "device_type",
    column='region'
).transform_filter(alt.FieldEqualPredicate(field='location_name', equal="The Church of Jesus Christ of Latter day Saints" )).properties(title=  "I

#Creates a chart using altair. This chart shows the same day brand visits to LDS buildings.
alt.data_transformers.enable('json')
alt.Chart(popular_brands_day).mark_bar().encode(
    alt.X('related_same_day_brand', title= "Brand"),
    alt.Y('same_day_brand_visits', title= "Visits to Brand")).properties(title = "popular Month Brands").transform_filter(alt.FieldEqualPredicate

#Creates a chart using altair.This chart shows all the same day brand visits to churchs.
alt.data_transformers.enable('json')
alt.Chart(popular_brands_day).mark_bar().encode(
    alt.X('related_same_day_brand', title= "Brand"),
    alt.Y('same_day_brand_visits', title= "Visits to Brand")).properties(title = "popular Month Brands").properties(title = "popular Month Brands
```