Bike-UAE Research Project Midterm Report

2025-06-30

Max Moran

Introduction

This summer I've been assisting Alexander Christou, a Transportation Expert with the Center for Interacting Urban Networks @ NYU Abu Dhabi, with a research project tackling 'Micro Mobility Usage & Challenges in the UAE'. The goal of this research is to assist the people and governments throughout the United Arab Emirates in taking back their cities from cars and help make educated decisions to improve the mobility within each community.

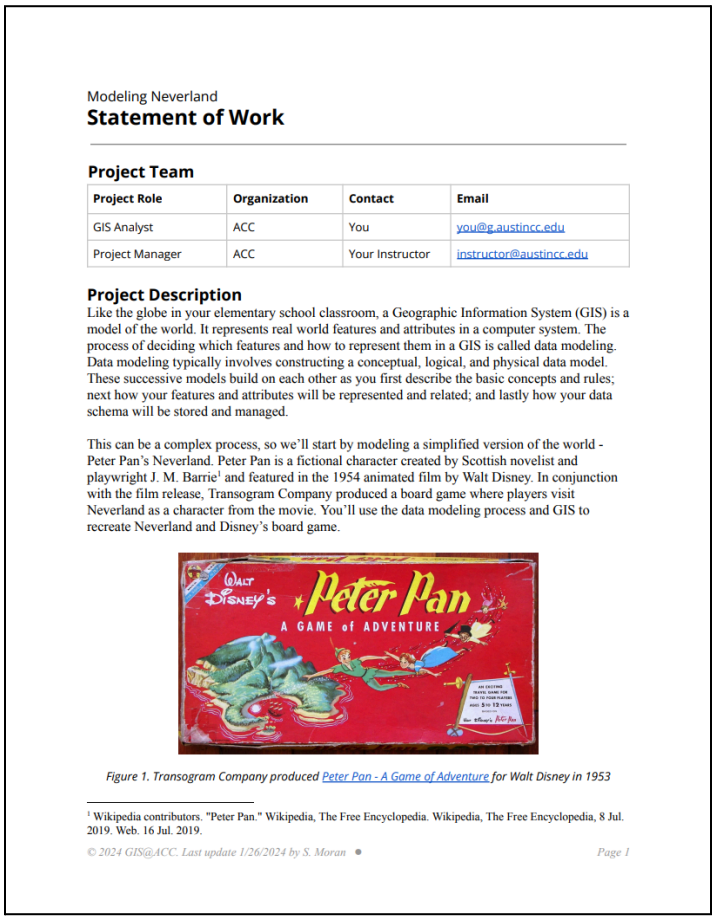
The UAE is set to spend millions of dollars in the next few years creating and improving micro mobility infrastructure. With a plan to build 1200+ km of bike lanes by 2028, it is our job to make sure they're implemented in the most effective way possible.

Throughout this project, we'll be focusing on micro mobility, which at its core is simply traveling from one place to another using small forms of transportation (i.e. bikes, scooters, skateboards, etc.).

I've spent the first half of this internship setting the foundation for some great analysis. I started with an introductory ARCGIS project to supplement future bike lane maps/zones in-need, processing/cleaning the survey data collected this spring, and began some early insights into the provided 'Strava Metro' data.

Introductory ARCGIS Project

To get me settled in and polish up my mapping skills, I completed a short project that got me up to speed with ARCGIS Pro, a common mapping program used to visualize spatial data in effective ways.



Modeling Neverland
Statement of Work

Project Team

Project Role	Organization	Contact	Email
GIS Analyst	ACC	You	you@g.austincc.edu
Project Manager	ACC	Your Instructor	instructor@austincc.edu

Project Description

Like the globe in your elementary school classroom, a Geographic Information System (GIS) is a model of the world. It represents real world features and attributes in a computer system. The process of deciding which features and how to represent them in a GIS is called data modeling. Data modeling typically involves constructing a conceptual, logical, and physical data model. These successive models build on each other as you first describe the basic concepts and rules; next how your features and attributes will be represented and related; and lastly how your data schema will be stored and managed.

This can be a complex process, so we'll start by modeling a simplified version of the world - Peter Pan's Neverland. Peter Pan is a fictional character created by Scottish novelist and playwright J. M. Barrie[1] and featured in the 1954 animated film by Walt Disney. In conjunction with the film release, Transogram Company produced a board game where players visit Neverland as a character from the movie. You'll use the data modeling process and GIS to recreate Neverland and Disney's board game.

Figure 1. Transogram Company produced Peter Pan - A Game of Adventure for Walt Disney in 1953

[1] Wikipedia contributors. "Peter Pan." Wikipedia, The Free Encyclopedia. Wikipedia, The Free Encyclopedia, 8 Jul. 2019. Web. 16 Jul. 2019.

Modeling Neverland, Statement of Work

The project was based around modeling Neverland, an old board game with varying types of data and rulesets.

I learned how to take in different types of files, including elevation, coordinates, images, data tables, etc.

This multi-day project taught me how to use the fundamentals of ARCGIS and should be a great tool for helping me map out ideal bike lanes & zones later this summer.

You can find the statement of work here and my completed map below:

Neverland

A Game of Adventure

LOCATION	NUMBER	OUTCOME
Rock Island	1	Danger - loose a turn
Rock Island	2	All is well, take an extra turn
Rock Island	3	A storm is brewing, go back 5 spaces
Rock Island	4	Smooth sailing go ahead 3 spaces
Crocodile Creek	1	The coast is clear, advance 4 spaces
Crocodile Creek	2	Crocodile chases you back 4 spaces
Crocodile Creek	3	You kill a crocodile, take an extra turn
Crocodile Creek	4	The crocodiles surround you, lose a turn
Hangman's Tree	1	You are well rested, take an extra turn
Hangman's Tree	2	You tell the boys a story lose a turn
Hangman's Tree	3	You look for food, go ahead 5 spaces
Hangman's Tree	4	Tinker Bell warns you of danger, go back 2 spaces
Mermaid Lagoon	1	Capt. Hook is approaching, go back 3 spaces
Mermaid Lagoon	2	The Mermaids call to you, advance 6 spaces
Mermaid Lagoon	3	Wait till Capt. Hook leaves, lose a turn
Mermaid Lagoon	4	The Mermaids arent home, take an extra turn
Skull Rock	1	As you fight Capt. Hook, lose a turn
Skull Rock	2	Capt. Hook is beaten, take an extra turn
Skull Rock	3	Sharp rocks around skull, go back 1 space
Skull Rock	4	A heavy wind pushes you ahead go ahead 1 space
Captian Hook's Ship	1	You are very brave, advance 2 spaces
Captian Hook's Ship	2	The boys fight the Pirates, go back 6 spaces
Captian Hook's Ship	3	You escape, take an extra turn
Captian Hook's Ship	4	You walk the plank, lose a turn

LOCATION	NUMBER	OUTCOME
Pirates Cove	1	You are being chased by Pirates, advance 1 space
Pirates Cove	2	You lost your way, go back 3 spaces
Pirates Cove	3	You are captured by the Pirates, lose one turn
Pirates Cove	4	You escape Pirates, take one extra turn
Indian Camp	1	Indians make you a chief, take an extra turn
Indian Camp	2	You are tired, rest and lose 1 turn
Indian Camp	3	Indians send you back 2 spaces
Indian Camp	4	Indian Chief awards you, advance 2 spaces
Lost Boy's Forest	1	Foxy chases you back 1 space
Lost Boy's Forest	2	You see Chubby, run ahead 3 spaces
Lost Boy's Forest	3	You escape Foxy, take an extra turn
Lost Boy's Forest	4	Pathi is rough, lose one turn
Bear Canyon	1	Bear blocks path, lose one turn
Bear Canyon	2	You are not tired, take an extra turn
Bear Canyon	3	Bear chases you, advance 4 spaces
Bear Canyon	4	Hide in cave, move 6 spaces back
Skull Rock	1	The Pirates spot you, run ahead 5 spaces
Skull Rock	2	The tide start to rise, lose 5 spaces
Skull Rock	3	Captain Hook arrives, lose a turn
Skull Rock	4	You assist in fight, take an extra turn
Captian Hook's Ship	1	You sneak overboard, take an extra turn
Captian Hook's Ship	2	You fall asleep, lose a turn
Captian Hook's Ship	3	You fall overboard, lose 4 spaces
Captian Hook's Ship	4	You swim away, advance 6 spaces

Start Here

You Fall Off Cliff Back 3 Spaces

You Are Lost Lose 1 Turn

Iceberg in Way Lose 1 Turn

Big Chief Sends You Ahead 3 Spaces

You Are Captured and Taken to Indian Camp

Crocodile Scares You Back 3 Spaces

You Are Forced Back 3 Spaces

Captain Hook Chases You Ahead 1 Extra Turn

Select Your Character

Whirlpool Throws You Back 3 Spaces

Chubby Runs Away Move On 3 Spaces

Fox Chases You Back 3 Spaces

Bear Canyon

Skull Rock

Mermaid Lagoon

You Admire the View Lose 1 Turn

Mermaids Detain You Back 3 Spaces

Crocodile Chases You to Creek

Fast Current Rushes You Ahead 3 Spaces

Stuck In Bog Lose 1 Turn

Hangman's Tree

Pirates Cove

Caught in Rocks Lose 1 Turn

You Are Lost Lose 1 Turn in Rocks Lose 1 Turn

Run From Pirates 4 Spaces Ahead

Rock Island

Indian Camp

Lost Boys Forest

Captain Hook's Ship

LOCATION	NUMBER	OUTCOME
Rock Island	1	You can't find others, go back 4 spaces
Rock Island	2	All is clear, advance 2 spaces
Rock Island	3	You go around rocks, take an extra turn
Rock Island	4	Rocks block your path, lose a turn
Crocodile Creek	1	You fight a crocodile, lose a turn
Crocodile Creek	2	No crocodiles in sight, take an extra turn
Crocodile Creek	3	You see a familiar sight, advance 3 spaces
Crocodile Creek	4	Stop to investigate, go back 3 spaces
Hangman's Tree	1	You want water, go ahead 4 spaces
Hangman's Tree	2	Search around the tree, go back 2 spaces
Hangman's Tree	3	A good meal delays you, lose a turn
Hangman's Tree	4	Pirates have captured others, take an extra turn
Mermaid Lagoon	1	You fly around, take an extra turn
Mermaid Lagoon	2	You stop to watch the Mermaids
Mermaid Lagoon	3	You like to linger here, go back 1 space
Mermaid Lagoon	4	You run to the Pirates, go ahead 5 spaces
Skull Rock	1	Swim to Captain Hooks ship, go back 6 spaces
Skull Rock	2	Fly to the Pirate's boat, go ahead 6 spaces
Skull Rock	3	Help the others at the boat, take and extra turn
Skull Rock	4	You fight Captain Hook, lose a turn
Captian Hook's Ship	1	You get the Pirates' ship ready, lose a turn
Captian Hook's Ship	2	Throw Captian Hook overboard, take an extra turn
Captian Hook's Ship	3	You are winning the fight, go ahead 1 space
Captian Hook's Ship	4	Go back to rescue the others, lose 5 spaces

LOCATION	NUMBER	OUTCOME
Pirates Cove	1	Pirates don't see you, take an extra turn
Pirates Cove	2	You search for treasure, lose one turn
Pirates Cove	3	You win duel with Pirates, advance 6 spaces
Pirates Cove	4	Hard to walk on sand beach, go back 1 space
Indian Camp	1	You become frightened, go back 2 spaces
Indian Camp	2	Medicine Man sends you ahead 4 spaces
Indian Camp	3	Rest in Wigwam, lose a turn
Indian Camp	4	You make friends with Indians, take an extra turn
Lost Boy's Forest	1	Lost boys decide to return home, lose a turn
Lost Boy's Forest	2	Pirates capture Lost Boys, take and extra turn
Lost Boy's Forest	3	You get lost, go back 3 spaces
Lost Boy's Forest	4	You lead boys into trap, advance 5 spaces
Bear Canyon	1	No bear in sight, advance 3 spaces
Bear Canyon	2	You trip on path, go back 4 spaces
Bear Canyon	3	You help bear cub, take an extra turn
Bear Canyon	4	You hide up a tree, lost a turn
Skull Rock	1	Tinker Bell tells you to take an extra turn
Skull Rock	2	As you chase Captain Hook, you lose a turn
Skull Rock	3	The tide is rising, advance 2 spaces
Skull Rock	4	You slip on the rock and go back 5 spaces
Captian Hook's Ship	1	You need added strength, go back 6 spaces
Captian Hook's Ship	2	Pirates don't see you, go ahead 1 space
Captian Hook's Ship	3	The big fight starts, lose a turn
Captian Hook's Ship	4	You capture the ship, take an extra turn

Rules

1) Two to four players. Each player must choose a character game marker.

2) Shuffle the Adventure Deck and place it face down in the center of the play area. Draw the top five cards and lay them face up to form the Neverland Path.

3) Each player begins the game with three Magic Tokens and one Pixie Dust card, which are placed face up in front of them.

4) Players take turns in clockwise order. On a turn, a player may move their character up to two spaces along the Neverland Path and interact with any face-up card they land on.

5) If a player collects three Lost Boy tokens, they may immediately trade them in for one Wish Star, which brings them closer to victory.

N W E S

0 1,000 2,000 4,000 Feet

Source: Walt Disney and Transogram Company. Map created by Max Moran for ACC Introduction to Geospatial Data on 2025-06-12.

CN

The Survey

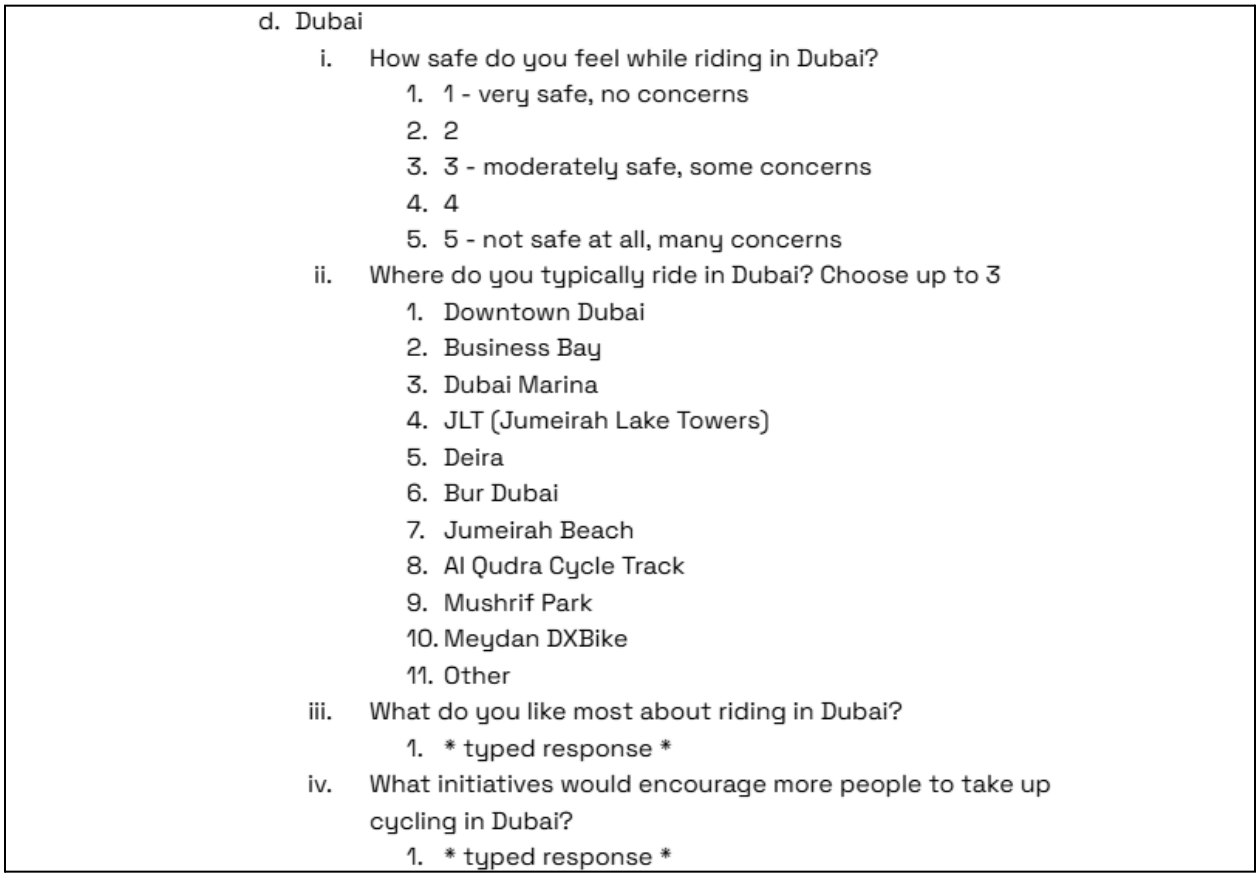
Next came the online questionnaire that Mr. Christou put together. The survey had more than 1000 respondents, all UAE residents aged 18+ who used some form of micro mobility. The data was collected from April to May of this year distributed UAE-wide by Wolfi's.

The data was collected and stored with Qualtrics, an online service to help collect and analyze survey results. You can see the format of the Qualtrics data in the following dashboard.

Q11 - Which Emirate do you call home?	Q15 - Which of the following do you use the most?	Q16 - How often do you cycle?
Abu Dhabi	Bicycle	2-3 times a week
Abu Dhabi	Bicycle	Daily
Dubai	Bicycle	4-6 times a week
Abu Dhabi	Bicycle	2-3 times a week
Abu Dhabi		
Abu Dhabi	Bicycle	4-6 times a week
Abu Dhabi	Bicycle	4-6 times a week
Abu Dhabi	Bicycle	2-3 times a week
Abu Dhabi: Al Ain	Bicycle	Once a week

Survey Data through Qualtrics

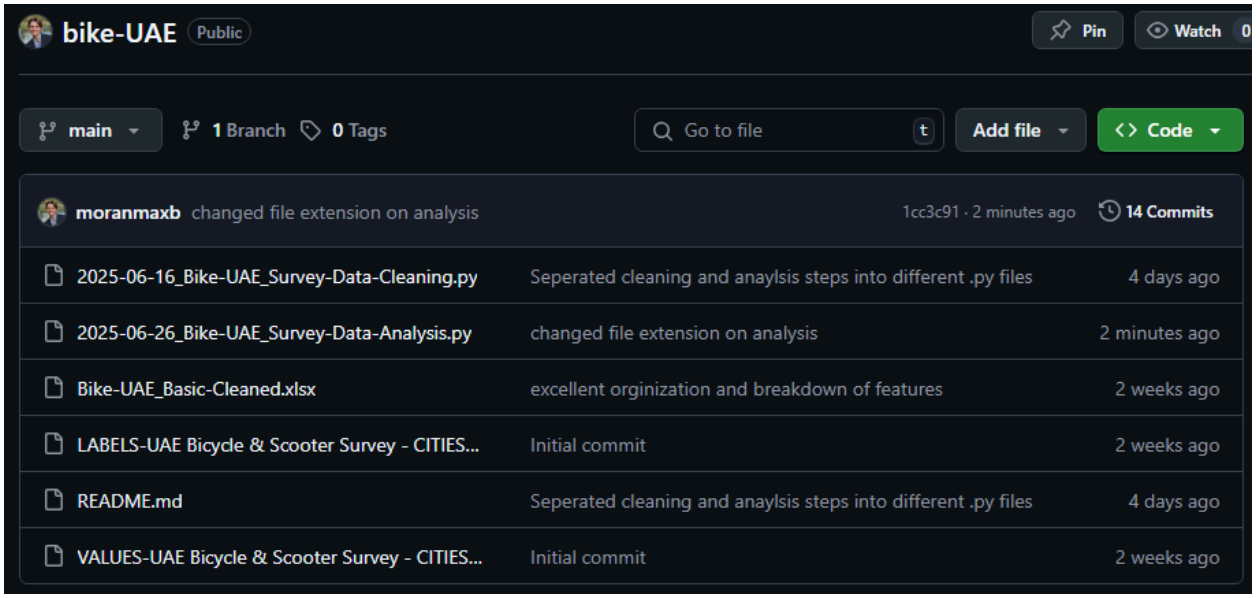
The questionnaire was complex, in that it had almost every type of response possible including multiple choice, text entry, multi select, numeric input, etc. It also featured nested questions, which makes sense for the user and writer but makes understanding the data challenging. Therefore, my first step was to rewrite the survey in a road map style, helping me understand what questions were asked in what order. This resulted in a list of 100+ variables of varying types, with each user response requiring 38 questions to be answered.



d. Dubai
   i. How safe do you feel while riding in Dubai?
      1. 1 - very safe, no concerns
      2. 2
      3. 3 - moderately safe, some concerns
      4. 4
      5. 5 - not safe at all, many concerns
   ii. Where do you typically ride in Dubai? Choose up to 3
      1. Downtown Dubai
      2. Business Bay
      3. Dubai Marina
      4. JLT (Jumeirah Lake Towers)
      5. Deira
      6. Bur Dubai
      7. Jumeirah Beach
      8. Al Qudra Cycle Track
      9. Mushrif Park
      10. Meydan DXBike
      11. Other
   iii. What do you like most about riding in Dubai?
      1. * typed response *
   iv. What initiatives would encourage more people to take up cycling in Dubai?
      1. * typed response *

Survey Questions Road Map

Before the coding started I knew I needed to be extremely organized throughout the development process. I brushed up on my GitHub fundamentals and ran through some practice projects for a technical refresh. A tool like this is crucial to the safety and completeness of your data analysis, preventing the loss of work and/or files. The repository is public so anyone can see the work I've completed so far here.



bike-UAE GitHub

Now that I had established a safe and reliable place for managing my work, it was ready to start the first step in data analysis; cleaning. From Qualtrics I downloaded the .csv file for the survey responses. Below you can see the raw output that was produced.

L	M	N	O	P	Q	R	S	T	U	V	W	X	Y
Q11	Q15	Q17	Q17_5_TE	Q16	Q30	Q34	Q34_5_TE	Q35	Q36	Q37	Q37_5_TE	Q38	Q39
Which Em	Which of t	Pedal pow	Pedal pow	How often	Which mo	Based on	Based on	How often	Which mo	You have	You have	How often	Which mo
{"ImportId	{"ImportId	{"ImportId	{"ImportId	{"ImportId	{"ImportId	{"ImportId	{"ImportId	{"ImportId	{"ImportId	{"ImportId	{"ImportId	{"ImportId	{"ImportId
Dubai													
Dubai	Bicycle	Competitive/Racing		Daily	January,February,March,April,May,June,July,August,September,October,November,December								
Dubai	Bicycle	Recreational/Leisure		2-3 times	January,February,March,September,October,November,December								
Dubai	Electric Scooter									Commuting/Mode of		A few time	January,C
Dubai	Bicycle	Fitness,Competitive		2-3 times	January,February,March,April,May,June,July,August,September,October,November,December								
Dubai	Bicycle	Recreational/Leisure		Once a we	January,February,November,December								
Abu Dhab	Bicycle	Recreational/Leisure		Once a we	January,February,March,April,September,October,November,December								
Abu Dhab	Bicycle	Fitness		2-3 times	January,February,March,April,May,June,July,August,September,October,November,December								
Abu Dhab	Bicycle	Recreational/Leisure		2-3 times	January,February,March,April,May,June,July,August,September,October,November,December								
Dubai	Bicycle	I want to be the next		4-6 times	January,February,March,April,May,June,July,August,September,October,November,December								
Dubai	Bicycle	Recreational/Leisure		Once a we	January,February,March,April,May,October,November,December								
Dubai	Bicycle	Recreational/Leisure		4-6 times	January,February,March,April,May,June,July,August,September,October,November,December								

Raw .csv survey data

The file included 100+ variables with little to no organization as the default output. I knew I would need to clean up inputs, column names, and a multitude of other issues it had.

Trimming & Cleaning Features

The easiest and first step was to trim down the fat and drop some columns that were not needed. Empty columns were selected and removed, including a nested question tree on 'Skateboarding' that seemed to be hidden from the survey entirely, or at least not a single person selected it.

Next came the removal of unnecessary context rows that won't be helpful in analysis. These included items like StartDate, ResponseID, DistributionChannel, etc. This data isn't gone but it won't be helpful for what we're trying to accomplish.

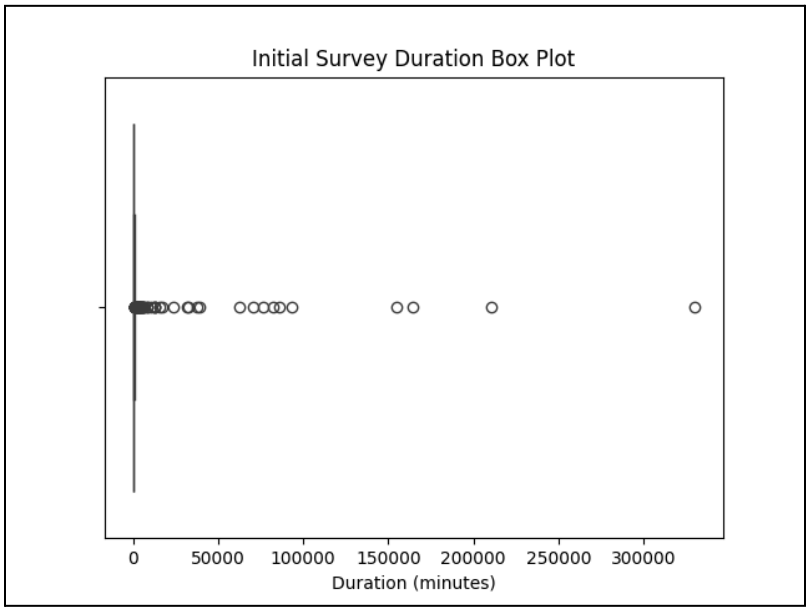
Then, using the road map I had created earlier, I renamed all the columns to help with readability and cut down on lengthy or confusing names. For example, Q11 isn't helpful as a column name without referencing further documentation, but 'Emirate', representing which Emirate the user resides in, is far more effective. Other examples include column names that are simply the question asked in their entirety: *'Pedal power! Based on your previous response, you don't need a motor to get places. Why do you cycle? (select all that apply)'* can simply be written as *'Bike-Why'*. This isn't the fault of Mr. Christou, it is simply how the data was formatted as it was exported.

Dealing with Spam

A common source of error in survey data is spam responses. Upon completion, participants could enter a giveaway, so there are almost certainly going to be individuals who click through the survey quickly without reading any of the questions.

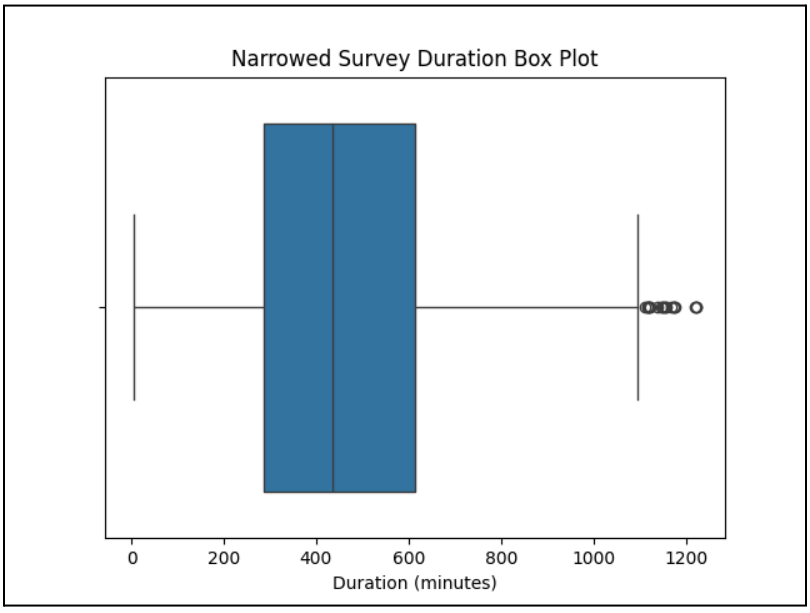
An established way to deal with this is to see if the user's responses are all extreme in one direction or the other, flagging the row as potential 'fraud'. These questions however aren't suited for such a method but thankfully we do have % completion along with duration variables to play with.

Before removing spam responses, let's take a look at the distribution of durations taken in the survey. Below you can find a boxplot (a representation for each quartile) of the data as it comes straight from Qualtrics.



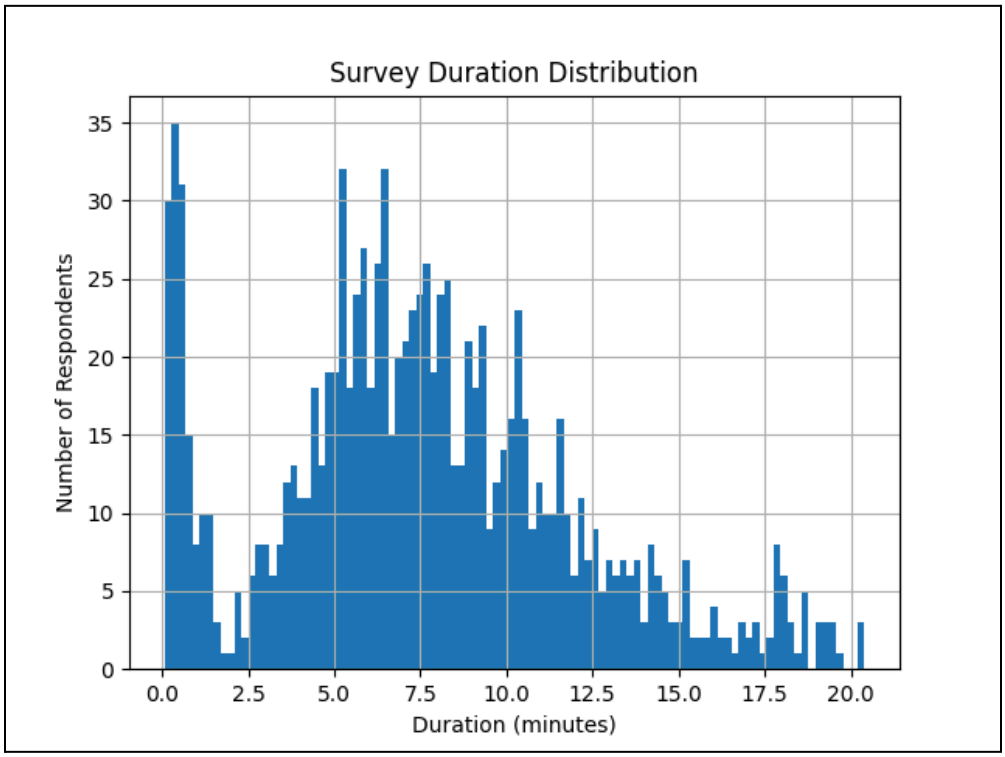
Boxplot of raw data.

This is impossible to read due to some extremely slow people (likely started and left open). These will be fine for feature analysis but for visualization let's temporarily remove the outliers so we can see the data more clearly.



Boxplot of narrowed data.

That's better; now we can see a right skew in the data with a couple outliers on the far right. Let's plot this narrowed data with a histogram to get a better grasp of what we're dealing with.



Narrowed Survey Duration Histogram

With 1000+ responses, we'd expect something similar to a normal distribution with a single hump in the middle (with perhaps a slight right skew to account for slow/afk folks). But above we can see a second hump in the data, with a large percentage of responses happening in under 2 minutes. With 38 questions, some with significant text, these have to be spam responses.

To solve this, I wrote a function to handle these answers we don't want to use. We need a way to mathematically sort out those who simply click through the survey without reading. These values can be tweaked later, but for now I set:

Minimum Questions for 100% Completion = **38**
Minimum Seconds per Question = **5**

Now with those constants established, we can write in a function of progress and duration to figure out who submitted a spam response. This is done by comparing the duration logged to the minimum total time to complete the survey multiplied by the percent completed.

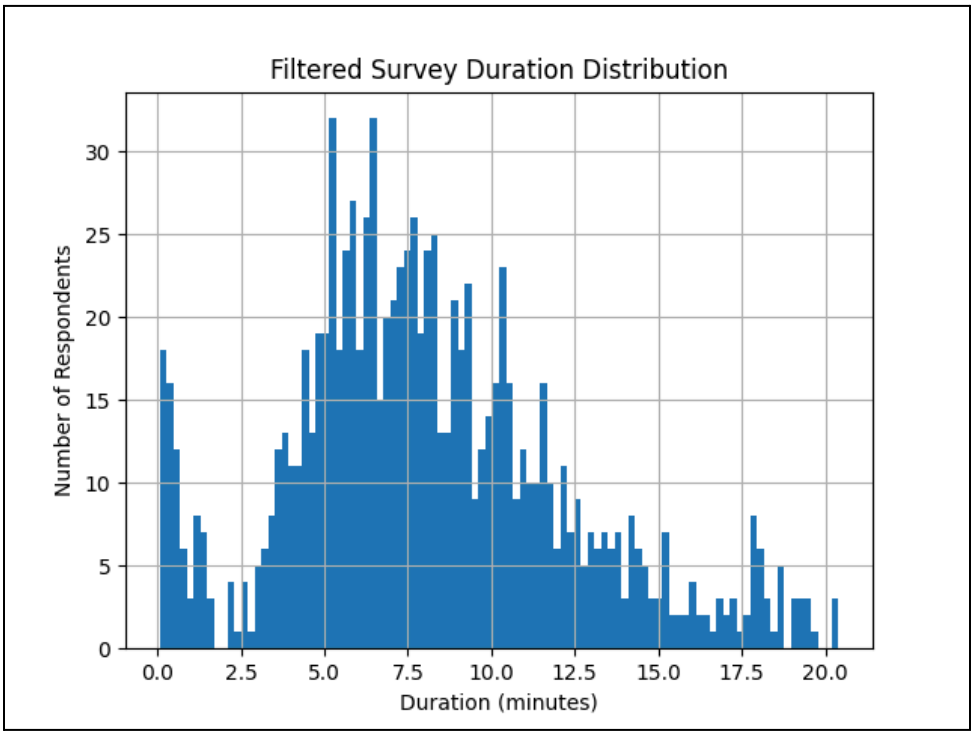
```
### Handle spam responses
# 38 questions (minQuestions) for 100% completion, assume ~5 seconds a question (minSecQ)
# So, any entries with a completion % that was submitted to fast will be dropped (likely spammed through questions)
preSpamFilterLen = len(df)
minQuestions = 38 # this is the minimum number of questions to complete 100% of the survey
minSecQ = 5 # set this variable to how many seconds it takes on average per question
minTotTime = minQuestions * minSecQ

df = df[~(df['Duration'] < (minTotTime*df['Progress'])/100)]
print("Dataframe without spam responses:")
print(df,"\n")
print("Removed",preSpamFilterLen-len(df),"spam responses.\n")

# Convert duration to minutes
df['Duration'] = df['Duration'] / 60
```

Code Snip-it to handle spam responses

If the user submitted a response too fast, then we filter it out and we're just left with the honest answers. To avoid false positives, I've made the minimum time to answer only 5 seconds to account for fast readers. After running, you can see below a much more reasonable distribution of durations.



Filtered Survey Duration Histogram

There is still a slight hump present under 2 minutes, but a portion of these are incomplete responses. So if someone only answered 4 questions and it took them over 20 seconds then that is fine for our uses. Some spam may still have slipped through but this is a great improvement regardless.

The Multi Select Issue

Once I had done some basic cleaning and filtering, I thought I was ready to jump into some analysis. I started with an ANOVA in an attempt to see if Gender affected the type of bikes owned. However, I quickly realized an issue in how the data was collected/formatted.

The survey had many multi select questions (ex: 'Select which types of bikes you own: Road, BMX, Mountain, etc.'), which are great for collecting data by essentially asking 6 questions in 1, but make analysis messy. Rows become cluttered with similar yet horrible to work with responses. For example:

Which months do you ride in?

January, February, April, May
vs
February, March, April
vs
January, March, April, May

All of these responses differ only slightly but show up as a different kind of response for each combination of months. For questions like these with 12 months, there are 12! theoretical combinations of a potential response! This was a serious problem that I had never dealt with before in raw data.

So the solution was to write a function allowing me to expand these questions into multiple boolean variables that simply have *true* or *false* for if the user selected it. For example, months cycled become 12 new boolean columns (January, February, etc.) and the original column is dropped.

```python
################ Tranform Multiselect Columns ################
# function to handle each needed column
def expand_multiselect_column(df, column, delimiter=',', drop_original=False):
    """
    Expands a multi-select column into individual boolean columns.

    Parameters:
        df (pd.DataFrame): Original DataFrame
        column (str): Column name with multi-select strings
        delimiter (str): Separator used between values (default is ',')
        drop_original (bool): If True, drops the original multi-select column

    Returns:
        pd.DataFrame: DataFrame with new boolean columns added
    """
    # Convert strings to lists, handle NaN
    cleaned = df[column].apply(lambda x: [i.strip() for i in str(x).split(delimiter) if i.strip()] if pd.notna(x) else np.nan)

    # Encode non-null rows
    non_null = cleaned.dropna()
    mlb = MultiLabelBinarizer()
    dummies = pd.DataFrame(mlb.fit_transform(non_null), columns=mlb.classes_, index=non_null.index)

    # Create full output with NaNs preserved
    full_dummies = pd.DataFrame(index=df.index, columns=mlb.classes_)
    full_dummies.update(dummies)

    # Convert 1.0/0.0 to True/False, but keep NaN where applicable
    full_dummies = full_dummies.where(full_dummies.isna(), full_dummies.astype(bool))

    # Join to original DataFrame
    df_expanded = pd.concat([df, full_dummies], axis=1)

    if drop_original:
        df_expanded = df_expanded.drop(columns=[column])

    return df_expanded
```

Transform Multiselect Column Function Code Snip-it

This solved my problem but I now had 100+ extra variables to deal with and organize after running the function for each multi select feature. Once again referencing my road map I made earlier (it pays to be thorough), I renamed each newly created variable with a more digestible name. 'A reliable and trustworthy place for maintenance-repairs-and upgrades' under cycler shop benefits can simply be written as 'CSB-ReliableMaintence'.

```python
# CycleShopsBenefit
'A reliable and trustworthy place for maintenance-repairs-and upgrades': 'CSB-ReliableMaintenance',
'A welcoming community and knowledgeable staff who support my cycling journey': 'CSB-WelcomingCommunity',
'Access to premium cycling gear and accessories that improved my performance': 'CSB-PremiumGearAccess',
'High-quality products and expert service that enhanced my cycling experience': 'CSB-HighQualityExpertService',
'Motivation to lead a healthier and more active lifestyle': 'CSB-HealthyMotivation',
# ComfortLocation
'City streets': 'Comfort-CityStreets',
'Cycle paths': 'Comfort-CyclePaths',
'Cycle tracks': 'Comfort-CycleTracks',
'Highways': 'Comfort-Highways',
'Parks': 'Comfort-Parks',
'Sidewalks': 'Comfort-Sidewalks',
'Neighborhood roads': 'Comfort-NeighborhoodRoads',
```

Renaming Multiselect Columns Code Snip-it

When I first realized several questions were nested, I didn't think about how this would affect column count. For example, after being asked about which Emirate the user resides in they were asked to select how safe they felt riding. This results in one column with their answer (ex: Abu Dhabi-3) and 8 empty columns with N/A values.

This is messy and fat, so I wrote a function to consolidate columns that could be merged. So instead of having 9 columns for emirate safety rating, there is only one and you can still see the user's zone in the 'Emirate' column.

```python
### Merge compatiable columns
def merge_string_columns(df, column_groups, separator=', '):
    """
    Concatenates multiple string columns row-wise into a single column.
    Preserves NaN if all values in a row are missing.

    Parameters:
        df (pd.DataFrame): Original DataFrame
        column_groups (dict): {new_col_name: [col1, col2, ...]}
        separator (str): Separator between strings

    Returns:
        pd.DataFrame: DataFrame with new merged string columns
    """
    for new_col, cols_to_merge in column_groups.items():
        # Combine strings row-wise, drop NAs per row
        merged = df[cols_to_merge].apply(
            lambda row: separator.join(row.dropna().astype(str)) if row.notna().any() else np.nan,
            axis=1
        )

        # Drop old columns and insert new one
        df = df.drop(columns=cols_to_merge)
        df[new_col] = merged

    return df
```

Merge Compatible String Columns Code Snip-it

Above is the more complicated merge, accounting for differing column names dependent on previous responses; in addition to this I wrote a function to merge duplicate rows.

For example, each Emirate had columns for months ridden in, so these could simply be consolidated by matching identical names.

```
### Merge exisiting columns (transport and months)
from collections import Counter

# Get all duplicated column names
duplicates = [col for col, count in Counter(df.columns).items() if count > 1]

for col in duplicates:
    # Get all columns with the same name
    same_cols = df.loc[:, df.columns == col]

    # Combine them row-wise using logical OR, preserving NaNs where all are missing
    merged = same_cols.any(axis=1)  # True if any are True

    # If all values in the row are NaN, set result to NaN (instead of False)
    all_nan_mask = same_cols.isna().all(axis=1)
    merged[all_nan_mask] = np.nan

    # Drop original duplicate columns and insert merged one
    df = df.drop(columns=same_cols.columns)
    df[col] = merged
```

Merge Duplicate Columns Code Snip-it

We're almost done! At this point that data is becoming great to work with, but is still hard to understand in table form. After all this cleaning and due to the messy export itself, the output data frame had no structure and was incredibly difficult to follow.

To solve this, I just reordered the columns into logical and readable sections. This has made things much easier when it comes to viewing neighboring sections in the data.

```
### Reorder Columns to logical layout
new_order = [# Starting User Info
             'Finished', 'UserLang', 'Progress', 'Duration',
             # Transport Info
             'TranspoType', 'Type-BMXBike', 'Type-CyclocrossBike', 'Type-ElectricBike',
             'TranspoType-HowOften', 'RiderType', 'RiderType-Other-Text', 'Bikeshare',
             'Why-TourdeFrance', 'Why-Commuting', 'Why-Racing', 'Why-Fitness', 'Why-Lei
             'AltMicro-Bike', 'AltMicro-EBike', 'AltMicro-EScooter', 'AltMicro-ESkatebo
```

Reorder Columns Code Snip-it

The Cleaned Data

Finally, we have not only clean but extremely well formatted data. After all that processing there are now nearly 200 variables we can use. This took a lot of time and effort, but I've learned that it is usually worth it to be kind to your future self and put some work in early.

EmploymentStatus	Salary	DurationInUAE	YearsRiding	January	February	March	April
Working full-time	More than 30k	23	Since childhood	TRUE	TRUE	TRUE	TRUE
Working full-time	20-25k	4	More than 20 years	TRUE	TRUE	TRUE	FALSE
Working full-time	5-10k	3	I am a new rider	TRUE	FALSE	FALSE	FALSE
Retired	More than 30k	2	1-5 years	TRUE	TRUE	TRUE	TRUE
Working full-time	15-20k	40	More than 20 years	TRUE	TRUE	FALSE	FALSE
				TRUE	TRUE	TRUE	TRUE
Working full-time	10-15k	15	15-20 years	TRUE	TRUE	TRUE	TRUE
Working full-time	20-25k	19	10-15 years	TRUE	TRUE	TRUE	TRUE
Working full-time	More than 30k	15	More than 20 years	TRUE	TRUE	TRUE	TRUE
Working full-time	More than 30k	12	More than 20 years	TRUE	TRUE	TRUE	TRUE
Other	More than 30k	1	Since childhood	TRUE	TRUE	TRUE	TRUE
A homemaker or stay-at-home parent	Prefer not to share	0.5	15-20 years	TRUE	TRUE	FALSE	FALSE
Working full-time	More than 30k		1-5 years	TRUE	TRUE	TRUE	TRUE
Working full-time	More than 30k	1	5-10 years	TRUE	TRUE	TRUE	FALSE
Working full-time	More than 30k	8	1-5 years	TRUE	TRUE	TRUE	TRUE
Working full-time	15-20k	2	I am a new rider	FALSE	FALSE	FALSE	FALSE

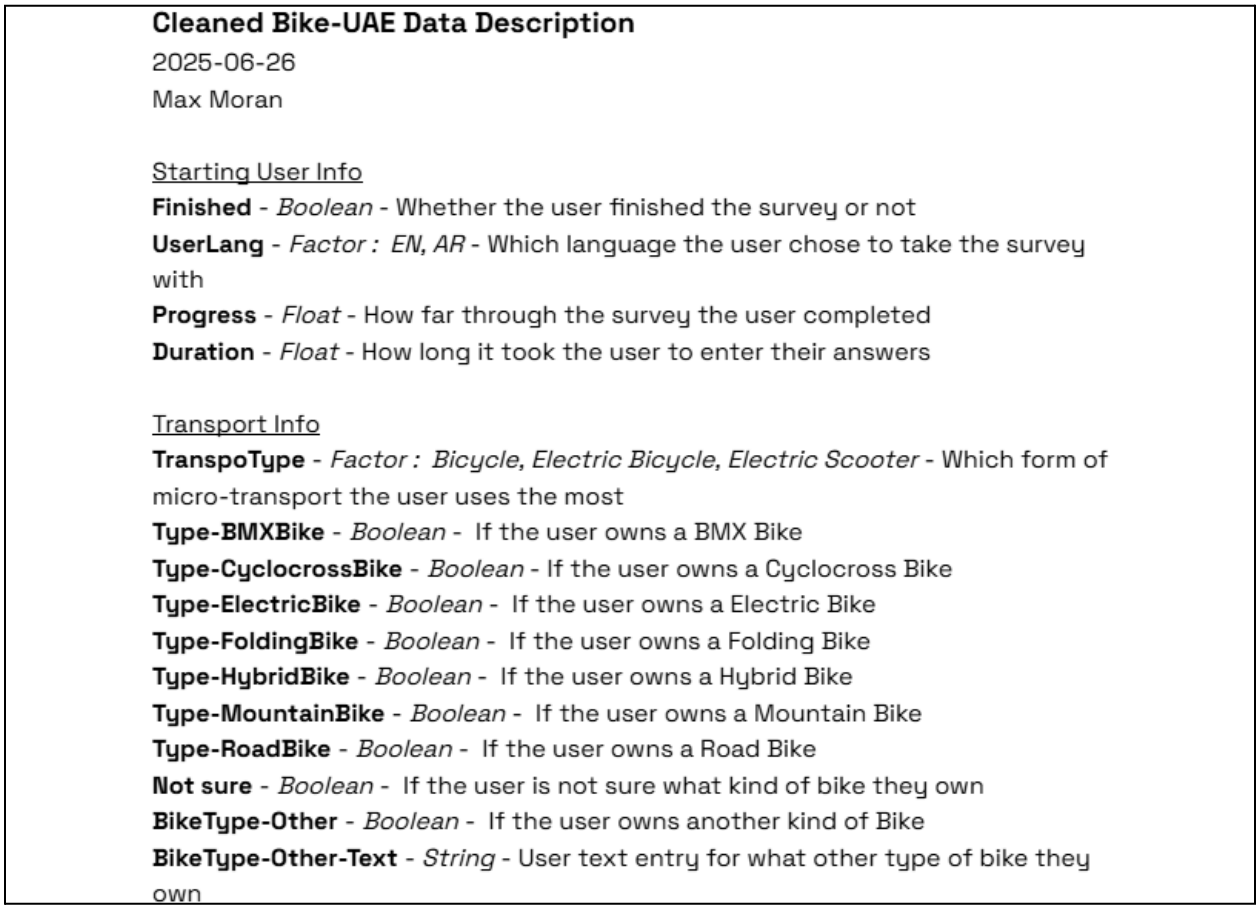
Cleaned Survey Data

The only remaining kind of data processing I could think to do would be handling the 'Other' responses. A decent amount of them are just people who did not read the answers and wrote in an answer that was already listed. Some people input slight variations of already present answers, for example 'Track Riding' and 'Circuit Cycling' are essentially the same thing.

There are not a lot of these however so I will tackle them at a later date. I would like to handle the unique other responses though. I think my approach will be based around using word clouds to establish which phrases/responses were the most common. I could go in manually to handle these but that would be poor practice.

I also need to consult with Mr. Christou about several responses that were in Arabic that I very much would appreciate help translating.

Once again, I want this work to be seen by more than just me and Xander so I've written a companion document that lists every feature in the cleaned data (organized by similar variables) along with their data type, factors for single select multiple choice, and a short description. Each item also appears in the exact order that you'll find in the cleaned excel file.



Cleaned Bike-UAE Data Description
2025-06-26
Max Moran

Starting User Info
Finished - *Boolean* - Whether the user finished the survey or not
UserLang - *Factor : EN, AR* - Which language the user chose to take the survey with
Progress - *Float* - How far through the survey the user completed
Duration - *Float* - How long it took the user to enter their answers

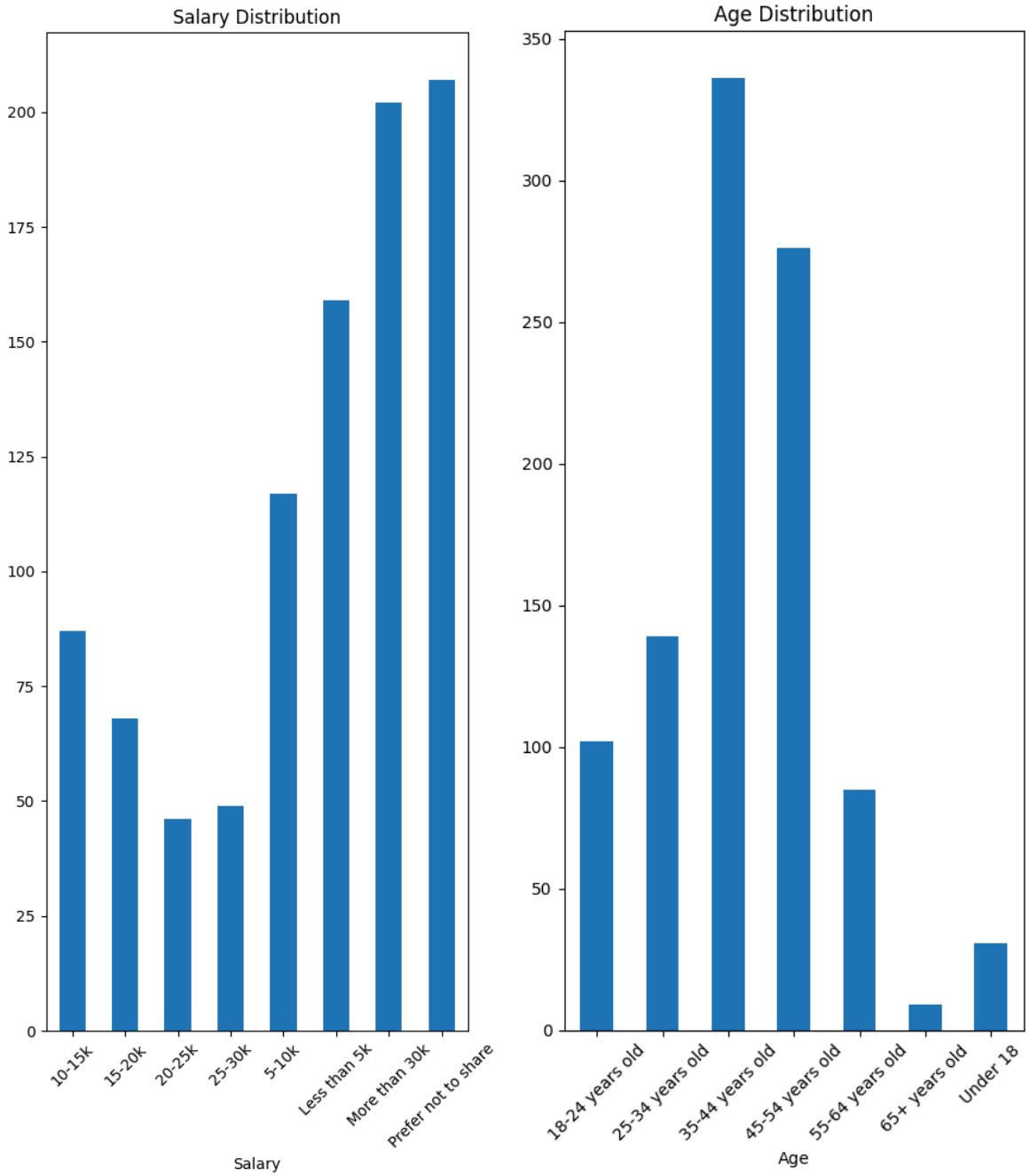
Transport Info
TranspoType - *Factor : Bicycle, Electric Bicycle, Electric Scooter* - Which form of micro-transport the user uses the most
Type-BMXBike - *Boolean* - If the user owns a BMX Bike
Type-CyclocrossBike - *Boolean* - If the user owns a Cyclocross Bike
Type-ElectricBike - *Boolean* - If the user owns a Electric Bike
Type-FoldingBike - *Boolean* - If the user owns a Folding Bike
Type-HybridBike - *Boolean* - If the user owns a Hybrid Bike
Type-MountainBike - *Boolean* - If the user owns a Mountain Bike
Type-RoadBike - *Boolean* - If the user owns a Road Bike
Not sure - *Boolean* - If the user is not sure what kind of bike they own
BikeType-Other - *Boolean* - If the user owns another kind of Bike
BikeType-Other-Text - *String* - User text entry for what other type of bike they own

Clean Data Variable Descriptions

This document, in addition to the readme on GitHub, should allow any individual to look over the data and analysis for their own use. I also went this far for my future self, because there is nothing more frustrating than forgetting why you did something that you used to know but no longer can remember.

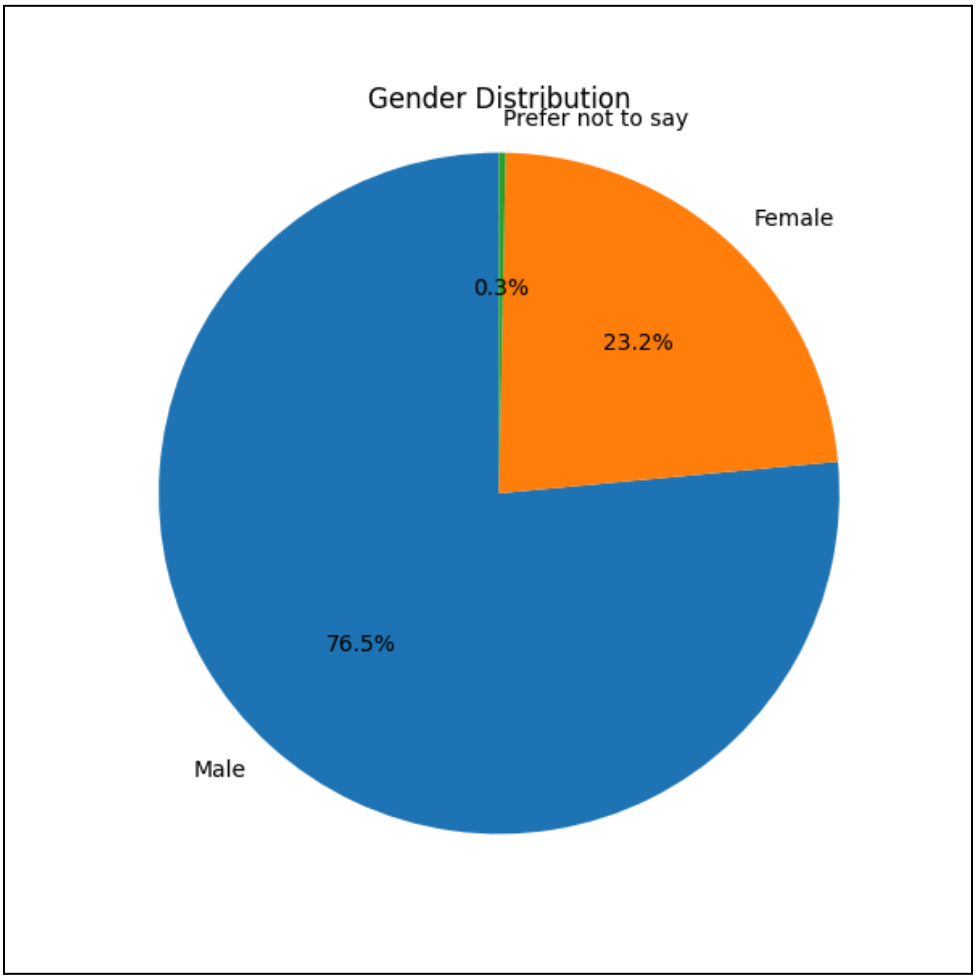
Basic Visualization

After all that processing, let's take a look at just some of what we have to work with. I plan to get into more complex visualization but for now this is sufficient for understanding who took this survey: some wealthier millennials apparently.



Note: future charts will be less confusing

There is certainly some selection bias, with the sampling frame only being those with access to the survey, but we can still get a decent picture of who uses micro mobility in the UAE.



Gender Distribution of Riders

The Beginnings of Analysis

I plan to do all sorts of data analysis in July, but the GIS and data processing took longer than expected, in addition to getting access to the data. So for now here is just a quick contingency table to show what direction I'd like to go in for the analysis.

```
# Check influence of Gender on Owning a Bike
table = pd.crosstab(df['Gender'], df['OwnBike'])
print(table)
chi2, p, dof, expected = chi2_contingency(table)

print()
print("Degrees of Freedom:", dof)
print()
print("Chi-Square Statistic:", chi2)
print("p-value:", p)
```

Gender Analysis Code Snip-it

Differences between genders cycling is important, because if only men bike then the system is failing. Especially in a country where it is less common for women to drive, alternatives to transport are crucial.

OwnBike	**No**	**Yes**
Female	45	182
Male	47	701
Prefer not to say	0	3

Gender vs Owning a Bike Contingency Table

Chi-Square Statistic: **37.77868517966452**
p-value: **6.258391774749355e-09**

From this chi-squared contingency table, we can see a statistically significant difference in owning a bike for women and men. A chi-square statistic of 37 and p-value of 0.0000000063 all but guarantees an imbalance.

With a cleaned and organized data set, I plan to do all sorts of analysis in July, making models and random forest to get down to the bottom of why citizens in the UAE do and don't ride their bike. With these mathematical backings, this survey becomes truly a citable and reliable justification for better and more efficient bike infrastructure.

Strava Data

One of the coolest tools provided in this internship, ‘Strava Metro’ allows us to see the specifics of activity within the UAE. Ranging from trip counts to elevation gain we can do all sorts of analysis for why people ride where they do.

Strava Metro Dashboard Overview (Blurred for data privacy policies)

I spent most of June working on other items, but I’m excited to dig my teeth into this data this July and hopefully discover some really cool statistical trends.

In addition to tabled numerical data, I also have spatial data to work with. Below you can see a heatmap of where users chose to ride in Abu Dhabi.

Strava Metro Abu Dhabi Heatmap (Blurred for data privacy policies)

I plan to reference this data with satellite imagery, infrastructure projects, and personal testimony to see what does and doesn't work across the UAE. Micro mobility infrastructure is only really successful if people use it, so it is important that we understand why projects work and what that means for future construction.

Conclusion

I learned a lot about the processing of real world data the past 5 weeks, along with how to manage working with someone who is halfway across the world. In this coming month, I plan to make a comprehensive analysis of the survey and strava metro data. In addition, I am going to use my GIS skills to create maps for potential future projects that, based on the data, should help the most people as effectively as possible.