Road Network Analysis Report: Zhongguanchun, Beijing

Introduction

This report outlines the process of creating and analyzing a routable road network within the Zhongguanchun area in Beijing, China. Utilizing OpenStreetMap data through the OSMnx library, this report details the steps taken from data acquisition to the visualization of the road network and the determination of the shortest path between two arbitrary points. The ultimate goal is to demonstrate the capabilities of OSMnx and NetworkX in urban planning and transportation engineering contexts.

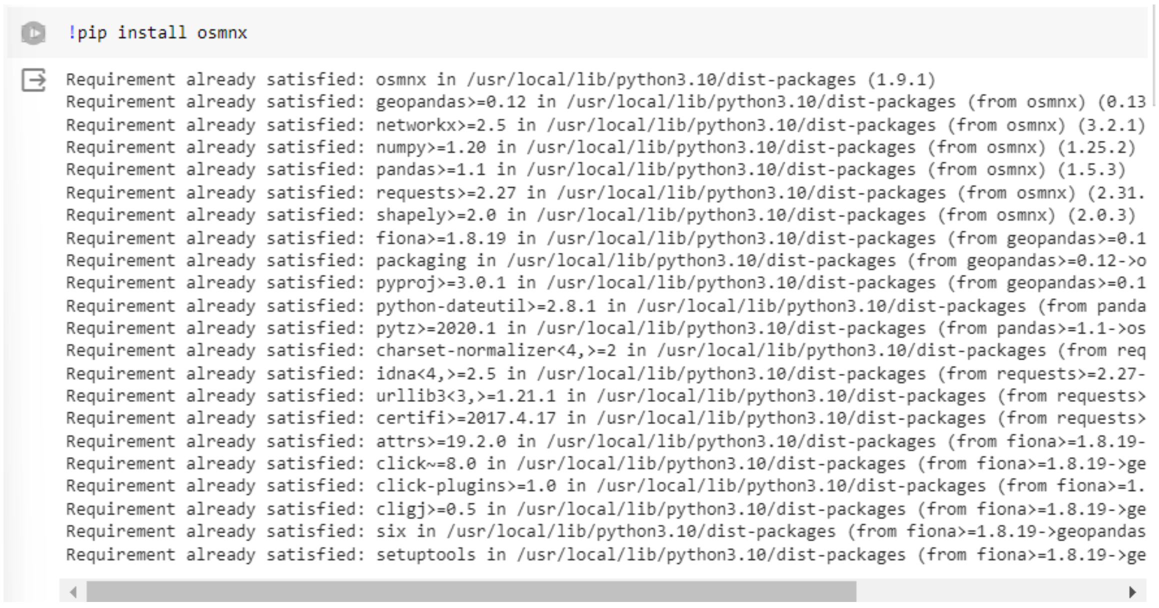
Methodology

The process flow followed in this report is outlined as follows:

1. **Installation and Setup**: This involves setting up the Python environment and installing necessary libraries, including OSMnx, NetworkX and Matplotlib
2. **Data Acquisition**: The road network data for Zhongguanchun, Beijing, is retrieved from OpenStreetMap via OSMnx.
3. **Network Analysis**: Using NetworkX, we analyze the road network to determine the shortest path between two arbitrarily selected nodes.
4. **Visualization**: The road network and the shortest path are visualized using the plotting capabilities of OSMnx.

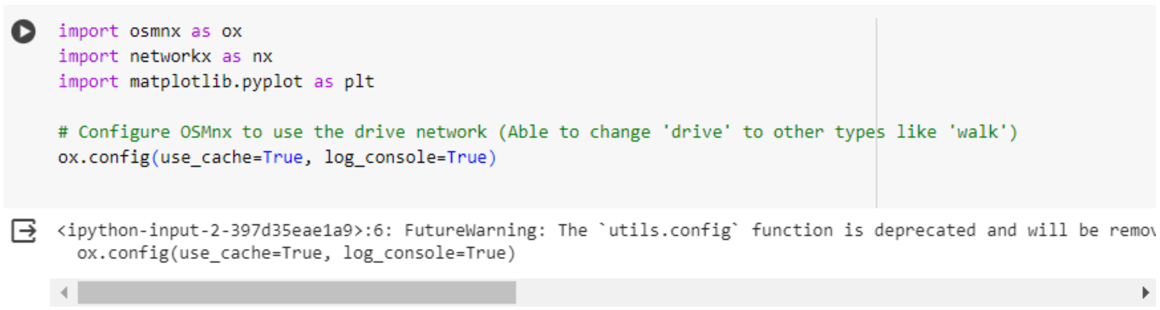
Installation and Setup

The required Python libraries were installed using the following command:



```
!pip install osmnx
```

```
Requirement already satisfied: osmnx in /usr/local/lib/python3.10/dist-packages (1.9.1)
Requirement already satisfied: geopandas>=0.12 in /usr/local/lib/python3.10/dist-packages (from osmnx) (0.13
Requirement already satisfied: networkx>=2.5 in /usr/local/lib/python3.10/dist-packages (from osmnx) (3.2.1)
Requirement already satisfied: numpy>=1.20 in /usr/local/lib/python3.10/dist-packages (from osmnx) (1.25.2)
Requirement already satisfied: pandas>=1.1 in /usr/local/lib/python3.10/dist-packages (from osmnx) (1.5.3)
Requirement already satisfied: requests>=2.27 in /usr/local/lib/python3.10/dist-packages (from osmnx) (2.31.
Requirement already satisfied: shapely>=2.0 in /usr/local/lib/python3.10/dist-packages (from osmnx) (2.0.3)
Requirement already satisfied: fiona>=1.8.19 in /usr/local/lib/python3.10/dist-packages (from geopandas>=0.1
Requirement already satisfied: packaging in /usr/local/lib/python3.10/dist-packages (from geopandas>=0.12->o
Requirement already satisfied: pyproj>=3.0.1 in /usr/local/lib/python3.10/dist-packages (from geopandas>=0.1
Requirement already satisfied: python-dateutil>=2.8.1 in /usr/local/lib/python3.10/dist-packages (from panda
Requirement already satisfied: pytz>=2020.1 in /usr/local/lib/python3.10/dist-packages (from pandas>=1.1->os
Requirement already satisfied: charset-normalizer<4,>=2 in /usr/local/lib/python3.10/dist-packages (from req
Requirement already satisfied: idna<4,>=2.5 in /usr/local/lib/python3.10/dist-packages (from requests>=2.27-
Requirement already satisfied: urllib3<3,>=1.21.1 in /usr/local/lib/python3.10/dist-packages (from requests>
Requirement already satisfied: certifi>=2017.4.17 in /usr/local/lib/python3.10/dist-packages (from requests>
Requirement already satisfied: attrs>=19.2.0 in /usr/local/lib/python3.10/dist-packages (from fiona>=1.8.19-
Requirement already satisfied: click~=8.0 in /usr/local/lib/python3.10/dist-packages (from fiona>=1.8.19->ge
Requirement already satisfied: click-plugins>=1.0 in /usr/local/lib/python3.10/dist-packages (from fiona>=1.
Requirement already satisfied: cligj>=0.5 in /usr/local/lib/python3.10/dist-packages (from fiona>=1.8.19->ge
Requirement already satisfied: six in /usr/local/lib/python3.10/dist-packages (from fiona>=1.8.19->geopandas
Requirement already satisfied: setuptools in /usr/local/lib/python3.10/dist-packages (from fiona>=1.8.19->ge
```



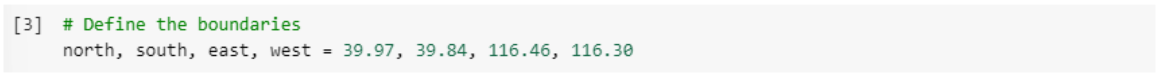
```
import osmnx as ox
import networkx as nx
import matplotlib.pyplot as plt

# Configure OSMnx to use the drive network (Able to change 'drive' to other types like 'walk')
ox.config(use_cache=True, log_console=True)
```

```
<ipython-input-2-397d35eae1a9>:6: FutureWarning: The `utils.config` function is deprecated and will be remov
  ox.config(use_cache=True, log_console=True)
```

The libraries were then imported into the Python script for use in the analysis.

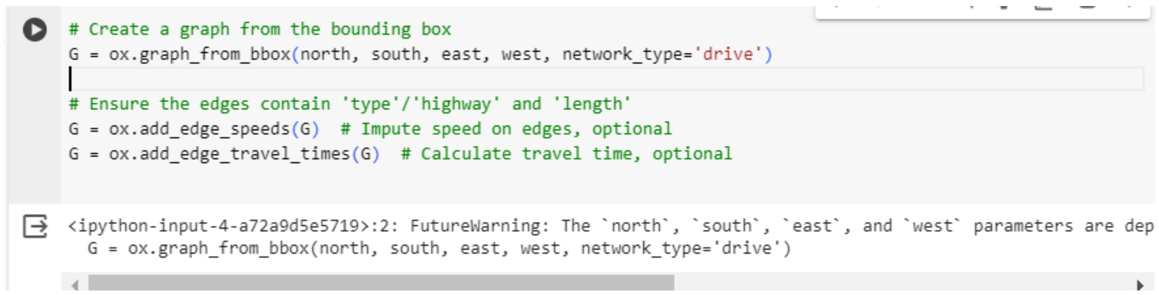
Data Acquisition

Define the boundaries of study area using the latitude and longitude values provided.



```
[3]  # Define the boundaries
     north, south, east, west = 39.97, 39.84, 116.46, 116.30
```

Create the graph from the defined bounding box.

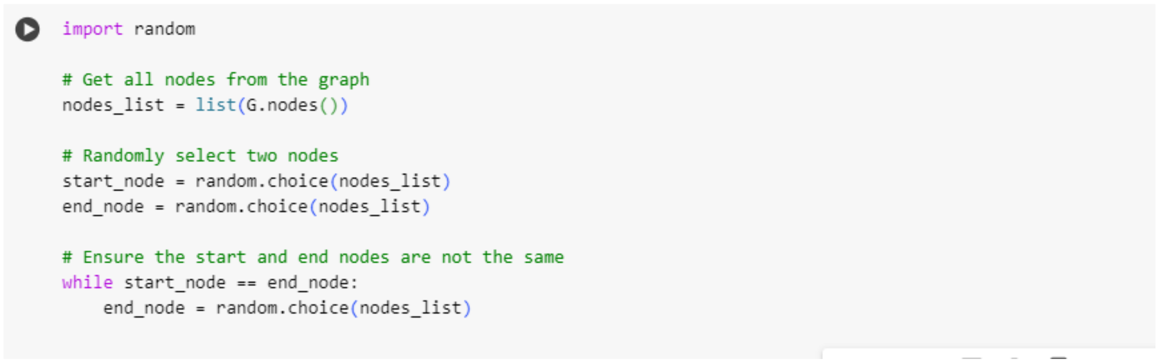


```
# Create a graph from the bounding box
G = ox.graph_from_bbox(north, south, east, west, network_type='drive')

# Ensure the edges contain 'type'/'highway' and 'length'
G = ox.add_edge_speeds(G)  # Impute speed on edges, optional
G = ox.add_edge_travel_times(G)  # Calculate travel time, optional
```

```
<ipython-input-4-a72a9d5e5719>:2: FutureWarning: The `north`, `south`, `east`, and `west` parameters are dep
  G = ox.graph_from_bbox(north, south, east, west, network_type='drive')
```

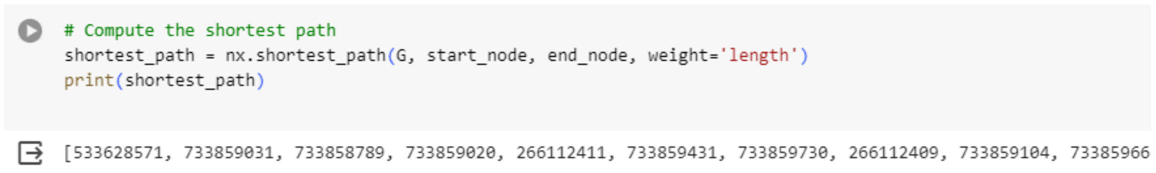
Network Analysis

Two nodes were arbitrarily selected from the network to serve as the start and end points for the shortest path analysis. The shortest path was computed using NetworkX's shortest path algorithm, based on the 'length' attribute of the road segments:



```
import random

# Get all nodes from the graph
nodes_list = list(G.nodes())

# Randomly select two nodes
start_node = random.choice(nodes_list)
end_node = random.choice(nodes_list)

# Ensure the start and end nodes are not the same
while start_node == end_node:
    end_node = random.choice(nodes_list)
```

For the calculation part, I use the shortest_path function from NetworkX, which by default uses Dijkstra's algorithm for weighted graphs.

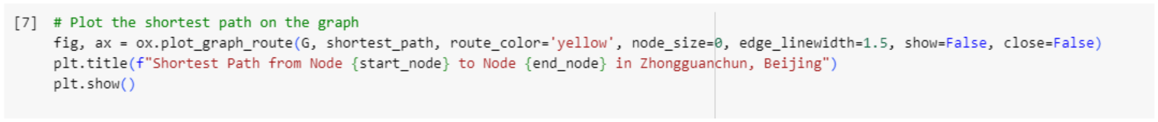


```
# Compute the shortest path
shortest_path = nx.shortest_path(G, start_node, end_node, weight='length')
print(shortest_path)
```

```
[533628571, 733859031, 733858789, 733859020, 266112411, 733859431, 733859730, 266112409, 733859104, 73385966
```

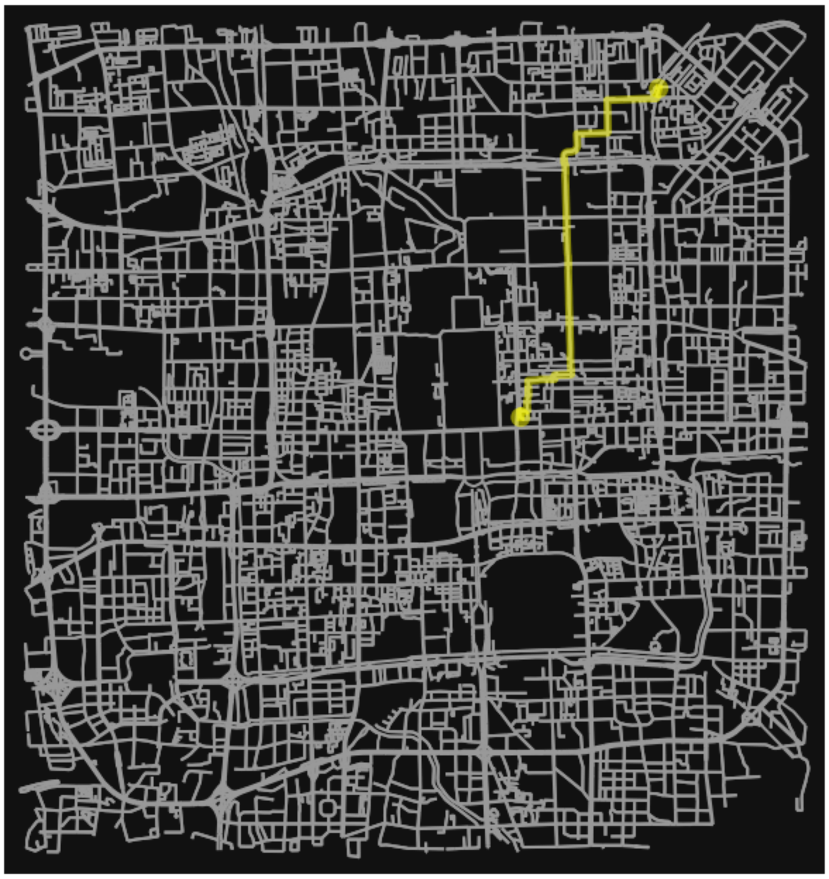
Visualization and Validation

Visualize the road network and the shortest path to validate the results



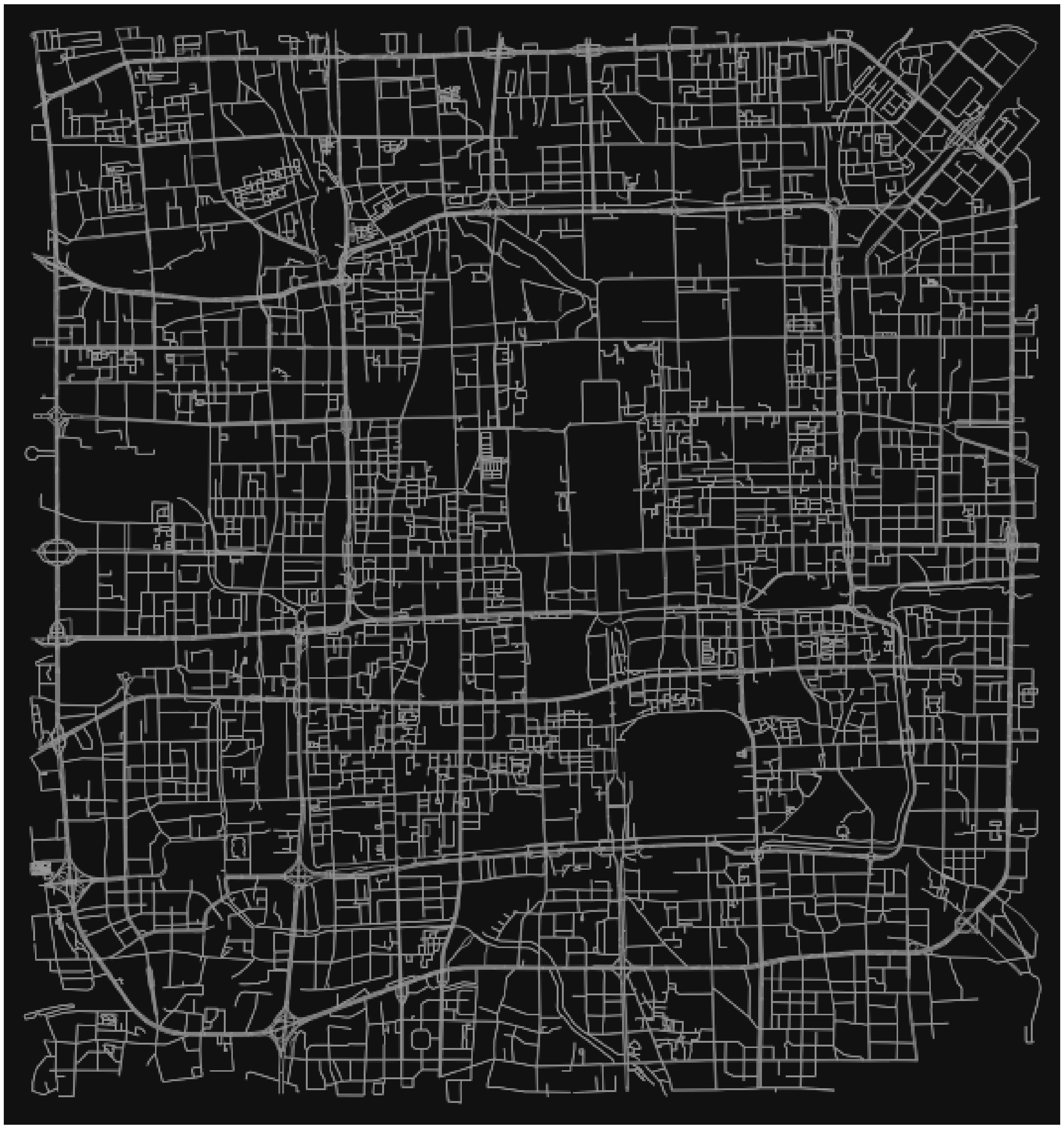
```
[7] # Plot the shortest path on the graph
    fig, ax = ox.plot_graph_route(G, shortest_path, route_color='yellow', node_size=0, edge_linewidth=1.5, show=False, close=False)
    plt.title(f"Shortest Path from Node {start_node} to Node {end_node} in Zhongguanchun, Beijing")
    plt.show()
```

Shortest Path from Node 533628571 to Node 2692298071 in Zhongguanchun, Beijing

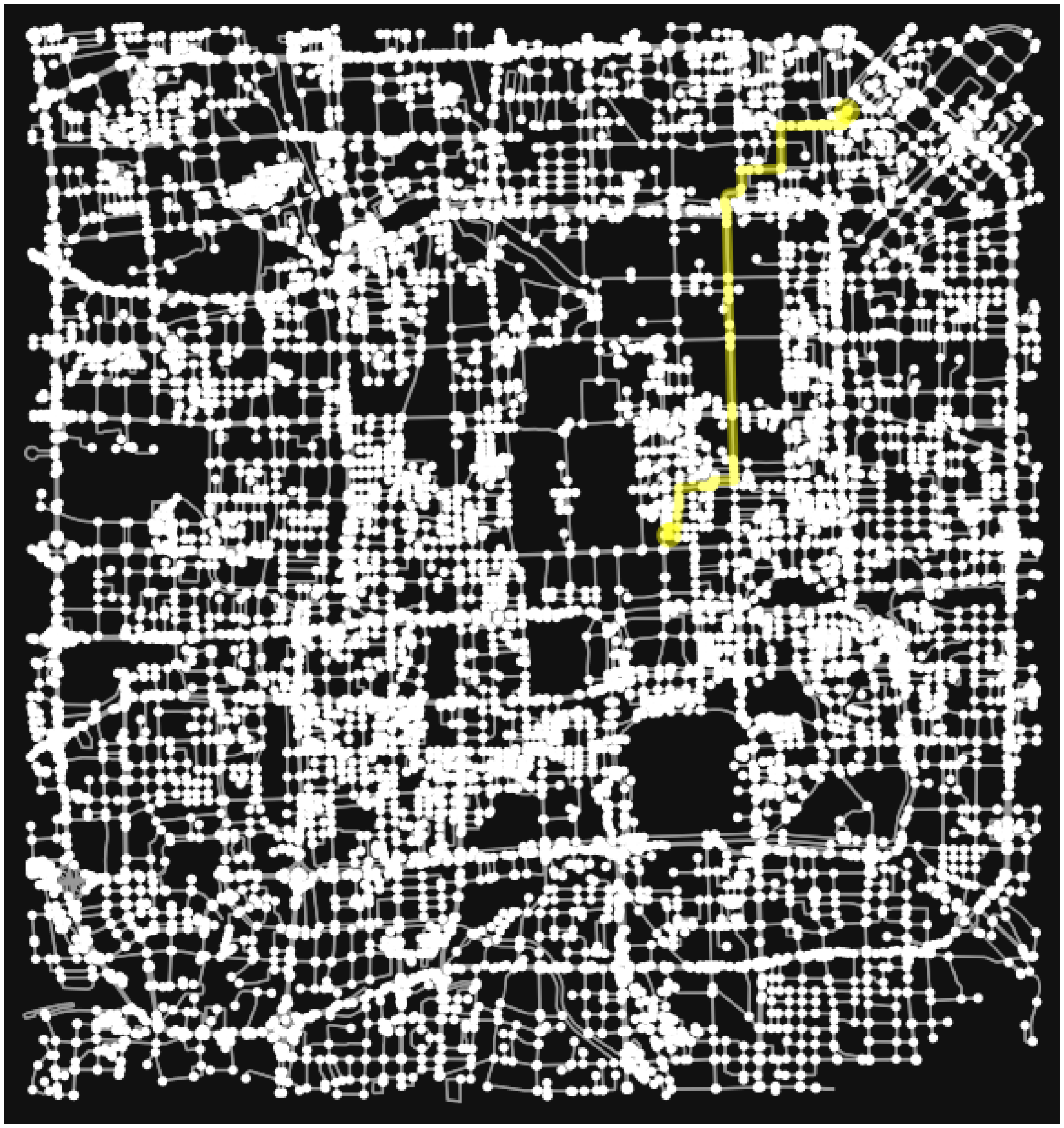


```
# Visualize the network
fig, ax = ox.plot_graph(G, node_size=0, edge_linewidth=0.5, show=False, close=False)
plt.title("Basic Road Network in Zhongguanchun, Beijing")
plt.show()

# Visualize the shortest path
fig, ax = ox.plot_graph_route(G, shortest_path, route_color='yellow', show=False, close=False)
plt.title("Shortest Path in Zhongguanchun, Beijing")
plt.show()
```

Basic Road Network in Zhongguanchun, Beijing



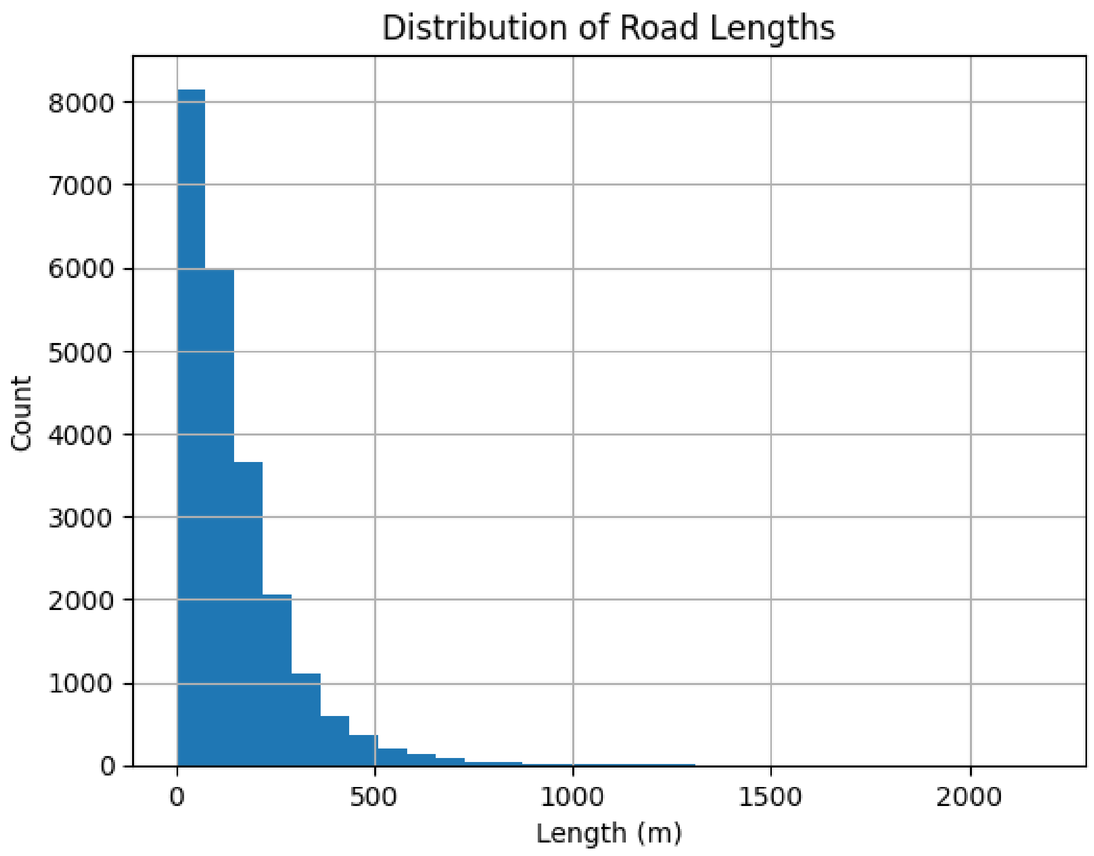
Shortest Path in Zhongguanchun, Beijing



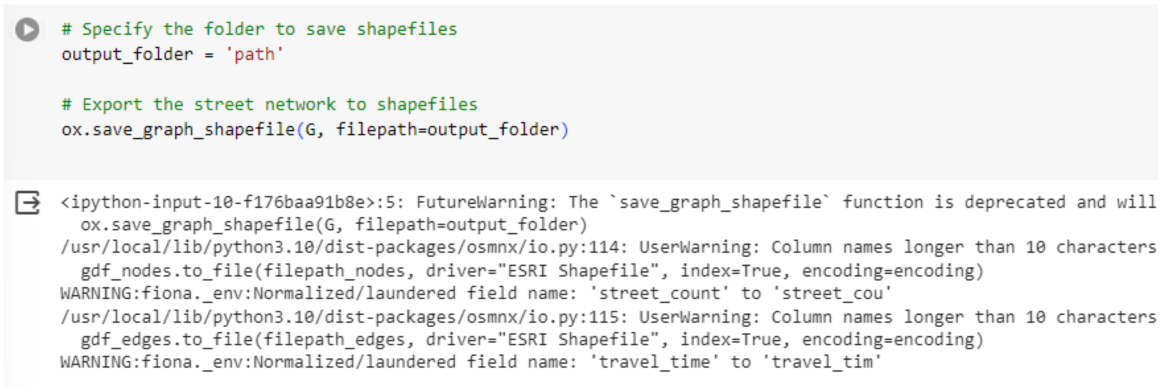
Additional Visualizations-- Distribution of Road Lengths:

I calculate and visualize the distribution of road lengths within the network. This are done using histograms.

```
#additional
import pandas as pd

# Convert graph edges to a DataFrame
edges = ox.graph_to_gdfs(G, nodes=False)

# Plot distribution of road lengths
edges['length'].hist(bins=30)
plt.title('Distribution of Road Lengths')
plt.xlabel('Length (m)')
plt.ylabel('Count')
plt.show()
```

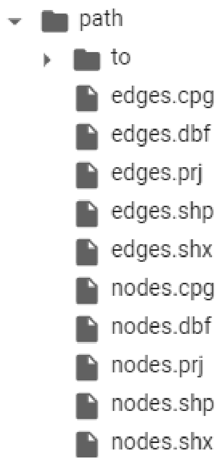


Exporting Data for Use in QGIS



```
# Specify the folder to save shapefiles
output_folder = 'path'

# Export the street network to shapefiles
ox.save_graph_shapefile(G, filepath=output_folder)
```

```
<ipython-input-10-f176baa91b8e>:5: FutureWarning: The `save_graph_shapefile` function is deprecated and will
  ox.save_graph_shapefile(G, filepath=output_folder)
/usr/local/lib/python3.10/dist-packages/osmnx/io.py:114: UserWarning: Column names longer than 10 characters
  gdf_nodes.to_file(filepath_nodes, driver="ESRI Shapefile", index=True, encoding=encoding)
WARNING:fiona._env:Normalized/laundered field name: 'street_count' to 'street_cou'
/usr/local/lib/python3.10/dist-packages/osmnx/io.py:115: UserWarning: Column names longer than 10 characters
  gdf_edges.to_file(filepath_edges, driver="ESRI Shapefile", index=True, encoding=encoding)
WARNING:fiona._env:Normalized/laundered field name: 'travel_time' to 'travel_tim'
```



```
path
  to
  edges.cpg
  edges.dbf
  edges.prj
  edges.shp
  edges.shx
  nodes.cpg
  nodes.dbf
  nodes.prj
  nodes.shp
  nodes.shx
```

Results and Discussion

The analysis successfully demonstrated the use of OSMnx and NetworkX in extracting and analyzing a road network from OpenStreetMap data. The shortest path computation between two arbitrary points provides valuable insights into route optimization and urban planning within the Zhongguanchun area. The visualizations offer a clear representation of the road network's complexity and the efficiency of the shortest path algorithm.

Conclusion

This report has presented a comprehensive process for extracting, analyzing, and visualizing a routable road network in Zhongguanchun, Beijing, using OSMnx, NetworkX and Matplotlib. The findings highlight the practical applications of these tools in urban planning and transportation studies. Future research could expand on this foundational work to include traffic data, pedestrian pathways, and other transport modes to create a more detailed and multifaceted analysis of urban mobility.