

Course: Geodata

By Dr. Cornelia Schneider

Student: Marjan Aziminezhad

Juni 2023

Inhaltverzeichnis

App identifizieren und Strecke aufzeichnen.....	2
• Identifizieren Sie eine passende App im Google Play Store bzw. Apple App Store. Die App muss in der Lage sein GPS-Daten im 1 Sekundenintervall aufzuzeichnen und als GPX (GPS Exchange Format) zu exportieren	2
• Zeichnen Sie einen GPS-Fußgängertrack von 2 bis 3 Kilometern Länge auf (im 1 Sekundenintervall).....	2
Daten exportieren	2
• Exportieren Sie die aufgezeichnete Fußstrecke als GPX und bringen Sie das File auf Ihren PC	2
Visualisieren in QGIS	3
• Starten Sie QGIS auf Ihrem PC oder am Terminalserver	3
• Visualisieren Sie nun den GPX Track auf einem OpenStreetMap-Layer (HINWEIS: Vektor – GPS-Werkzeuge).....	3
• Berechnen Sie die Länge des GPS Tracks (HINWEIS: Attributtabelle und darin enthaltene Feldrechnerfunktion verwenden).....	4
• Sie kennen nun die genaue Länge Ihres GPS Tracks – berechnen Sie nun die Geschwindigkeit (HINWEIS: Startzeit bzw. -punkt sowie Endzeit bzw. -punkt können über den ersten bzw. letzten Punkt im Track identifiziert werden; die Berechnung kann, muss aber nicht in QGIS durchgeführt werden).....	5
Visualisieren in Google My Maps.....	6
• Loggen Sie sich mit ihrem Google-Account bei Google ein (sollten Sie über keinen Account verfügen legen Sie sich bitte einen Google-Account an)	6
• Visualisieren Sie nun den GPS Track in My Maps	6
• Berechnen/Visualisieren Sie die Länge/Geschwindigkeit des Tracks.....	8
Dokumentation der Übung :Analysen mit Python, MySQL und QGIS	8
• Dokumentieren Sie die in der Lehrveranstaltung durchgeführte Übung (Beschreibung, Screenshots).....	8
Abbildungsverzeichnis	14

App identifizieren und Strecke aufzeichnen

- **Identifizieren Sie eine passende App im Google Play Store bzw. Apple App Store. Die App muss in der Lage sein GPS-Daten im 1 Sekundenintervall aufzuzeichnen und als GPX (GPS Exchange Format) zu exportieren**

Für die Aufzeichnung der Daten kamen zwei Apps in Frage: GEOtracker und GPStracks. Von den beiden wurde GPStracks verwendet, da sie benutzerfreundlicher ist. Allerdings erfordert dies den Kauf der Pro-Version der App, um die Daten auf den PC zu übertragen und sie zu bearbeiten.

- **Zeichnen Sie einen GPS-Fußgängertrack von 2 bis 3 Kilometern Länge auf (im 1 Sekundenintervall)**

Die Strecke wurde in NÖ-Wieselburg aufgezeichnet.

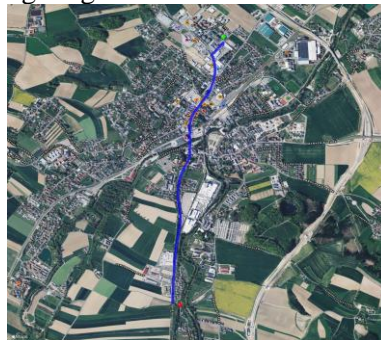


Abbildung 1 Aufzeichnung einer Strecke in Wieselburg

Daten exportieren

- **Exportieren Sie die aufgezeichnete Fußstrecke als GPX und bringen Sie das File auf Ihren PC**

Um die Datei aus der GPStracks App zu exportieren, müssen die folgenden Schritte befolgt werden:

1. Navigieren Sie zur Aufzeichnung, deren Daten Sie exportieren möchten.
2. Suchen Sie nach der Exportfunktion, die normalerweise über ein Symbol oder eine Menüoption zugänglich ist.
3. Tippen Sie auf die Exportoption, um den Exportvorgang zu starten.

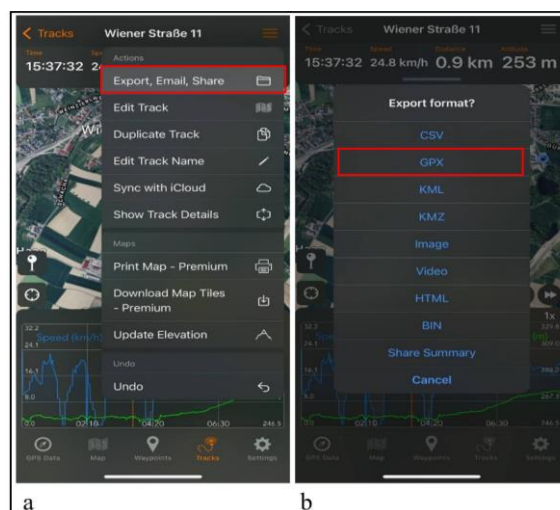


Abbildung 2 a) Exportieren der Datei ,Abbildung 2 b) Auswahl des Formats der zu exportierenden Datei

Visualisieren in QGIS

- **Starten Sie QGIS auf Ihrem PC oder am Terminalserver**
- **Visualisieren Sie nun den GPX Track auf einem OpenStreetMap-Layer (HINWEIS: Vektor – GPS-Werkzeuge)**

Man kann den GPX-Track der aufgezeichneten Strecke auf einem OpenStreetMap-Layer in QGIS 3.30 wie folgt visualisieren:

1. QGIS 3.30 wird geöffnet.
2. Der OpenStreetMap-Layer wird im Browser-Panel geöffnet.(Abb. 3)

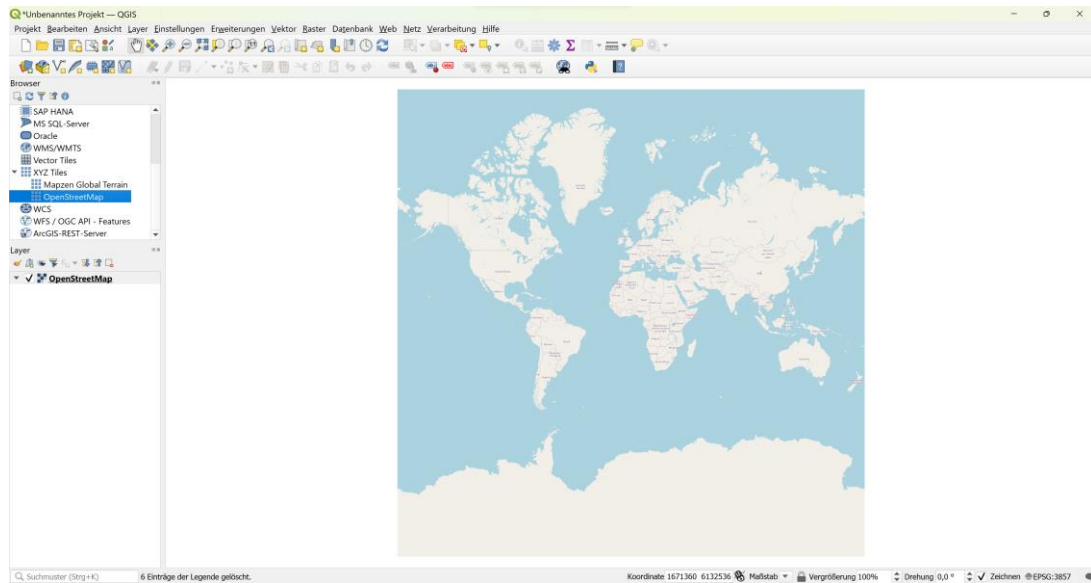


Abbildung 3 Vorbereitung OpenStreetMap

3. Die GPX-Datei der aufgezeichneten Strecke wird importiert, indem man mit der rechten Maustaste auf "Layer" im Browser-Panel klickt und "Neuer Vektorlayer" auswählt.(Abb- 4)

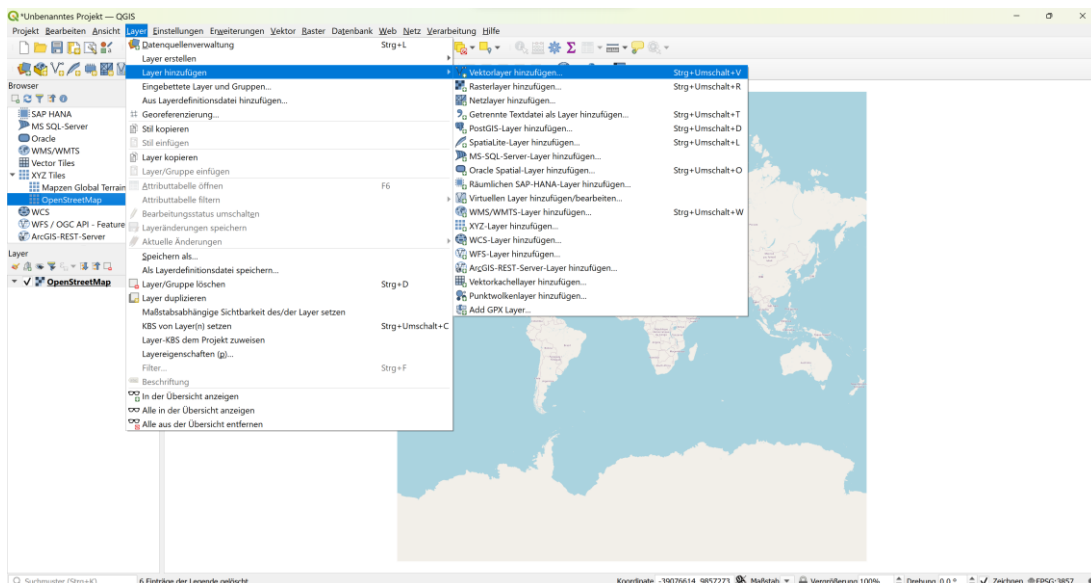


Abbildung 4 Neuer Vektorlayer hinzufügen

- Die GPX-Datei wird ausgewählt und geöffnet, um den GPX-Track als neuen Vektorlayer in der Kartenansicht anzuzeigen.(Abb. 5)

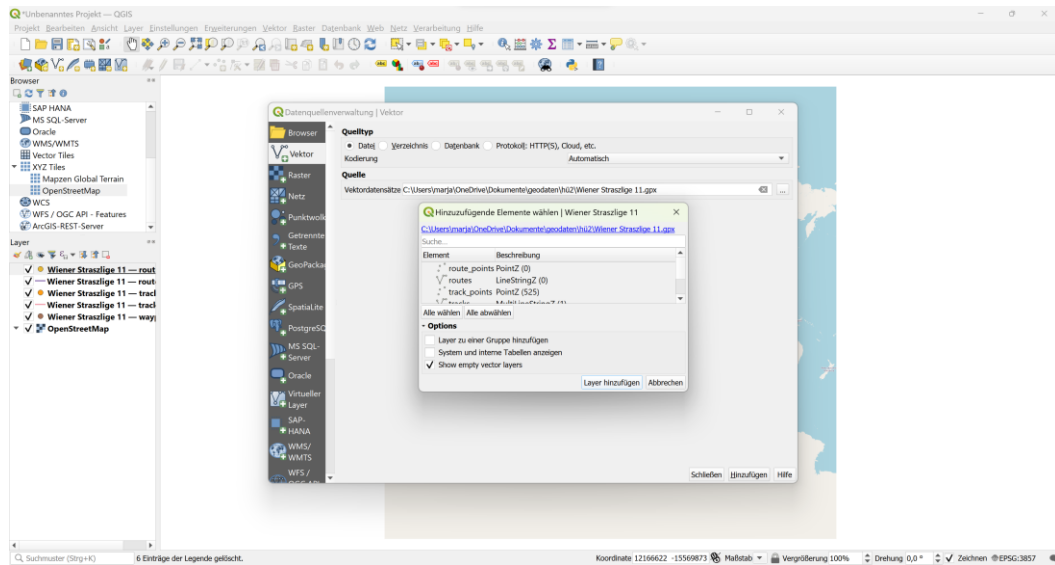


Abbildung 5 Auswahl der GPX Datei

- Durch Verwendung des Zoom-Werkzeugs in der Symbolleiste kann man auf den GPX-Track heranzoomen, um ihn genauer zu betrachten. Mit dem Schiebe-Werkzeug kann man die Karte verschieben, um verschiedene Teile des GPX-Tracks anzuzeigen.

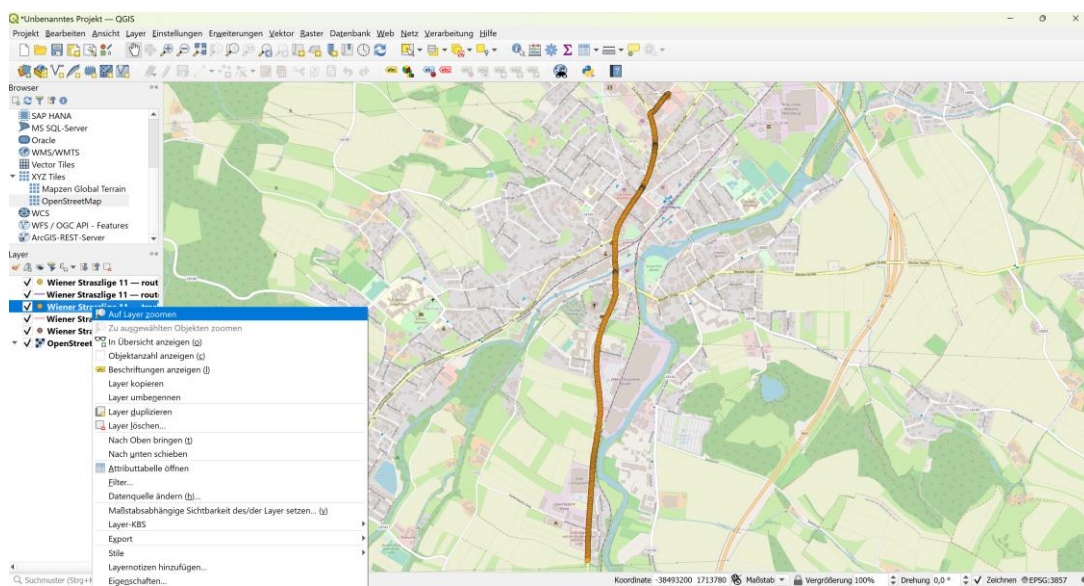


Abbildung 6 Heranzoomen auf den GPX-Track

- Berechnen Sie die Länge des GPS Tracks (HINWEIS: Attributtabelle und darin enthaltene Feldrechnerfunktion verwenden)**

Um die Länge des GPS-Tracks in QGIS zu berechnen, kann man die Attributtabelle und die Feldrechnerfunktion verwenden;

- Mit einem Rechtsklick auf den Layer in der Ebenenliste kann man das Kontextmenü öffnen und " Attributtabelle " auswählen.
- In der Feldrechnerfunktion kann man die Länge des GPS-Tracks berechnen. Man sucht nach das Symbol "Feldrechner" in der Symbolleiste der Attributtabelle, um den Feldrechner zu öffnen

3. Man kann eine neue Spalte hinzufügen. Dazu schreibt man im Feld "Ausgabefeldname" der Name der Spalte ein.
4. Man wählt die entsprechende Funktion zur Berechnung der Länge aus. Je nach den vorhandenen Attributen und der Geometrie des Tracks kann man beispielsweise die Funktion "\$Length" verwenden.
5. Man klickt auf "OK" oder "Berechnen", um die Länge des GPS-Tracks in der Attributtabelle zu berechnen und die Werte in der entsprechenden Spalte anzuzeigen.

Hier in Abbildung 7 wird 2583,37 gezeigt.

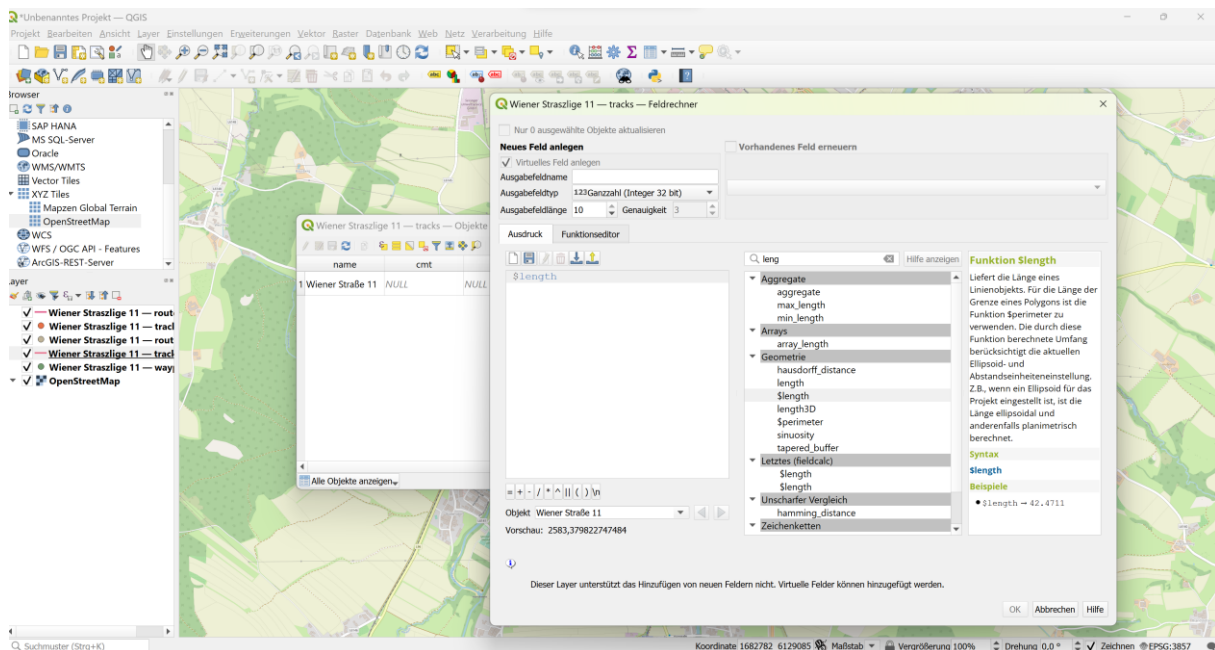


Abbildung 7 Vorschau der Streckenlänge

- Sie kennen nun die genaue Länge Ihres GPS Tracks – berechnen Sie nun die Geschwindigkeit (HINWEIS: Startzeit bzw. -punkt sowie Endzeit bzw. -punkt können über den ersten bzw. letzten Punkt im Track identifiziert werden; die Berechnung kann, muss aber nicht in QGIS durchgeführt werden)

Die Zeitdauer zwischen dem Start- und Endpunkt des Tracks ist laut Abb. 8 zu lesen.

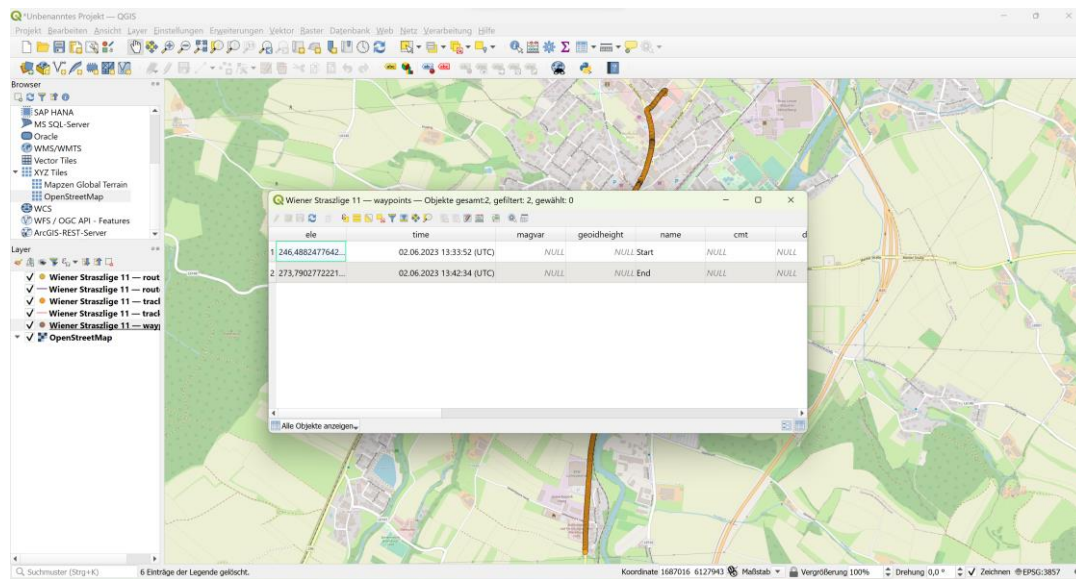


Abbildung 8 Start- und Endzeitpunkt

In diesem Fall beträgt der **Startzeitpunkt 02.06.2023 13:33:52 UTC** und der **Endzeitpunkt 02.06.2023 13:42:34 UTC**. Indem man den Endzeitpunkt vom Startzeitpunkt subtrahiert, erhält man eine Zeitdauer von **8 Minuten und 42 Sekunden**.

Die **Distanz** des Tracks beträgt **2583,3799 m**.

Um die Distanz von Metern in Kilometer umzurechnen, teilen wir die Distanz durch 1000:

-Distanz in Kilometer = $2583,3799 \text{ m} / 1000 = 2,5833799 \text{ km}$

Um die Zeitdauer von Minuten in Stunden umzuwandeln, teilen wir die Minuten durch 60 und die Sekunden durch 3600 und addieren sie zur Stundenangabe:

Zeitdauer in Stunden = $((8 \text{ Minuten} / 60) + (42 \text{ Sekunden} / 3600)) = 0,145 \text{ Stunden}$

Schließlich teilen wir die umgerechnete Distanz in Kilometer durch die Zeitdauer in Stunden, um die Geschwindigkeit in Kilometern pro Stunde zu berechnen:

Geschwindigkeit = Distanz in Kilometer / Zeitdauer in Stunden

Geschwindigkeit = $2,5833799 \text{ km} / 0,145 \text{ Stunden} \approx 17,815 \text{ km/h}$

Visualisieren in Google My Maps

- **Loggen Sie sich mit ihrem Google-Account bei Google ein (sollten Sie über keinen Account verfügen legen Sie sich bitte einen Google-Account an)**
- **Visualisieren Sie nun den GPS Track in My Maps**

1. Google Maps wird in einem Webbrowser geöffnet und zu "My Maps" wird auf der Google Maps-Website navigiert.(Abb. 9)

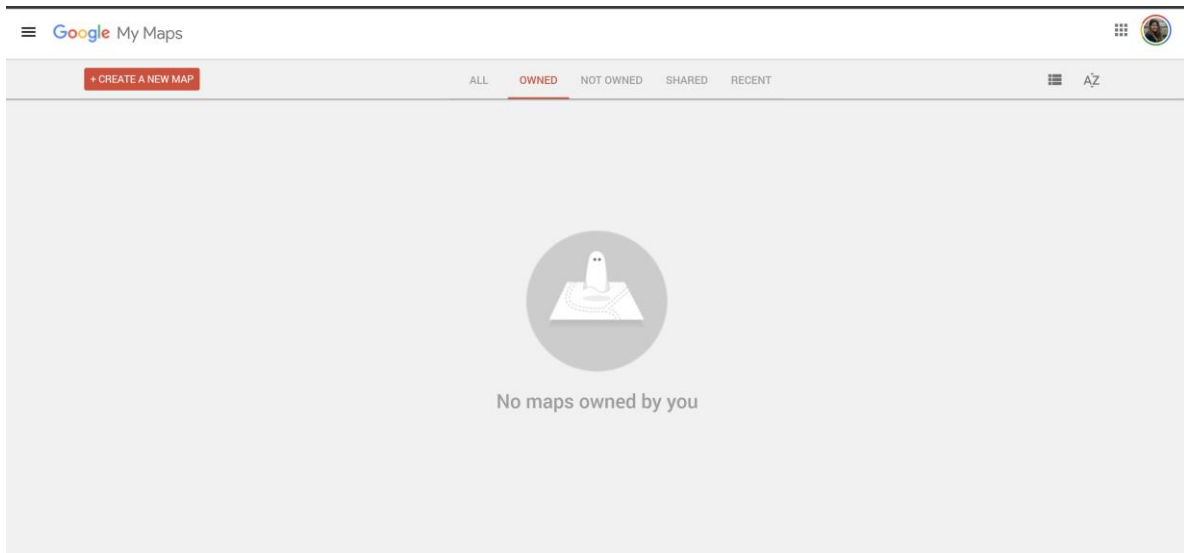


Abbildung 9 MyMaps Webseite

2. Auf "Neue Karte erstellen" wird geklickt danach eine neue Ebene wird hinzugefügt, indem auf die Schaltfläche "Add Layer" geklickt wird.
3. Die Option zum Importieren von Dateien wird aus dem Menü der neuen Ebene ausgewählt.
4. Die Datei mit dem GPS-Track auf dem Computer wird gesucht und ausgewählt.
5. Die Datei wird ausgewählt und auf "Öffnen" geklickt, um den Importvorgang zu starten.(Abb.10)

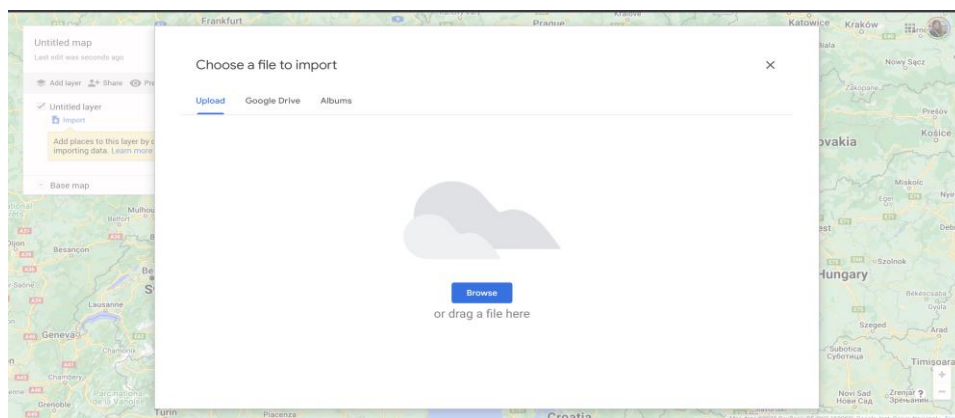


Abbildung 10 Importieren der GPX Datei auf MyMaps Webseite

6. Die Datei mit dem GPS-Track wird von Google Maps hochgeladen und der GPS-Track wird auf der Karte angezeigt.(Abb. 11)
7. Die angezeigte Position des GPS-Tracks auf der Karte wird überprüft.

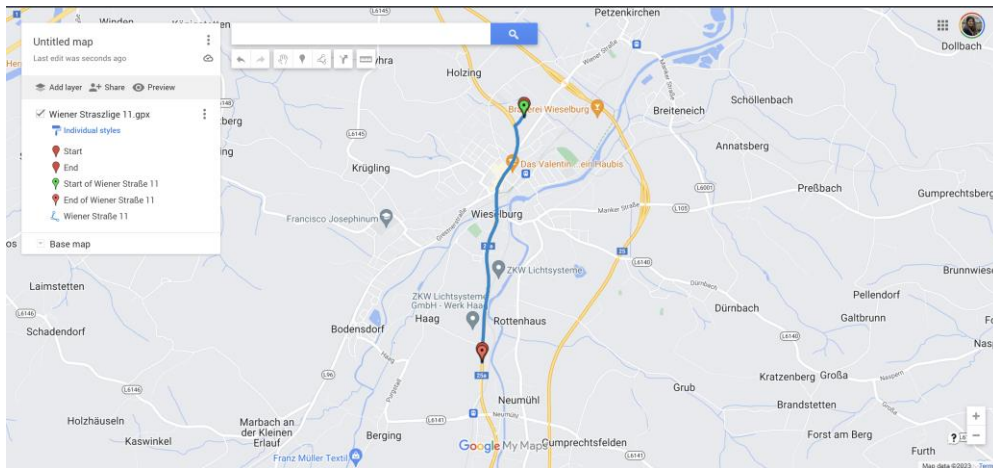


Abbildung 11 Visualisierung der Strecke mittels MyMaps

- **Berechnen/Visualisieren Sie die Länge/Geschwindigkeit des Tracks**

Die berechnete Länge des Tracks und die Geschwindigkeit werden automatisch auf der Karte angezeigt.

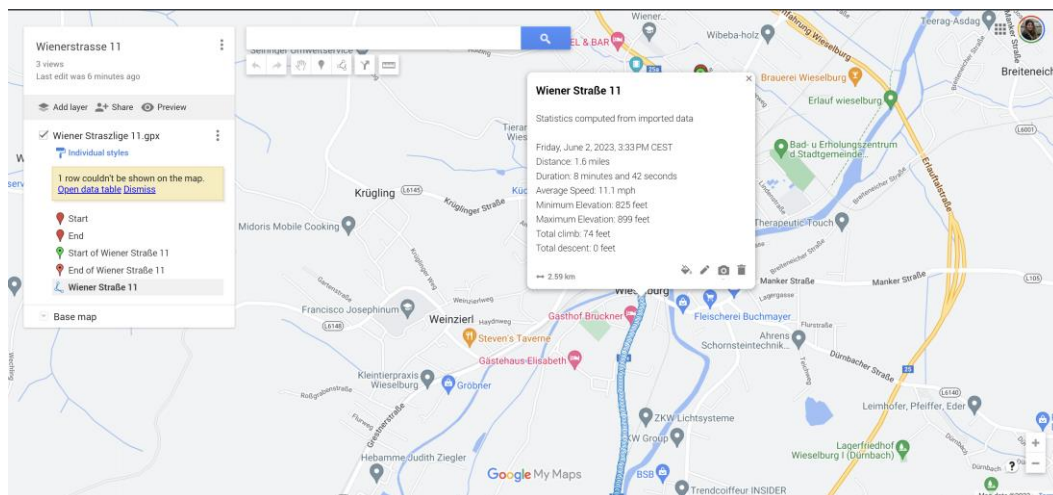


Abbildung 12 Visualisierung der Länge und Geschwindigkeit

Umrechnung der **Distanz**: $1,6 \text{ Meilen} * 1,60934 \text{ km/Mile} = \mathbf{2,57494 \text{ Kilometer}}$

Umrechnung der **Geschwindigkeit**: $11,1 \text{ mph} * 1,60934 \text{ km/h} = \mathbf{17,86474 \text{ km/h}}$

Dokumentation der Übung :Analysen mit Python, MySQL und QGIS

- **Dokumentieren Sie die in der Lehrveranstaltung durchgeführte Übung (Beschreibung, Screenshots)**

Um mit der Übung zu beginnen starten Sie bitte den *QGIS Terminalserver*. In der Lehrveranstaltung haben Sie den *Divvy-Datensatz* kennengelernt. Für die Ausfahrt (trip) mit der Ausfahrtsnummer (trip_id) 487868 sind nachfolgende Aufgaben zu lösen:

- Berechnen Sie (i) die sphärische Distanz zwischen der Ausleih- und Rückgabestation sowie (ii) die Geschwindigkeit mit der diese Strecke zurückgelegt wurde. Nutzen Sie dafür den *MySQL Workbench*.

Der gegebene Code (Abb. 13) berechnet die sphärische Distanz zwischen zwei Divvy-Stationen für eine bestimmte trip_id (487868). Dabei wird die Distanz zwischen der Ausleihstation (id=153) und der Rückgabestation (id=279) ermittelt.

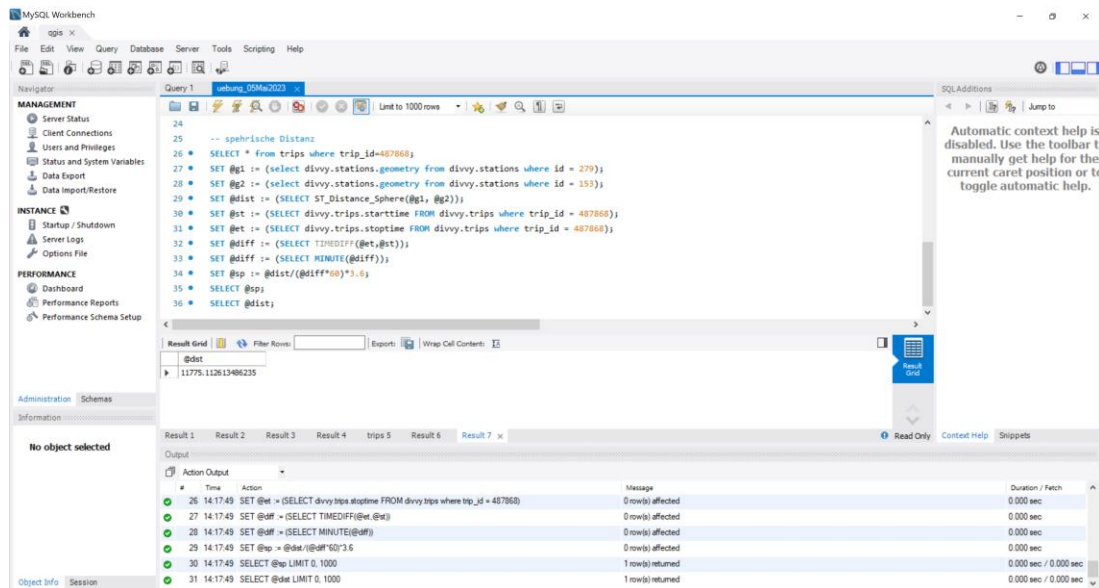


Abbildung 13 MySQL Code zum Berechnen der Distanz zwischen id 279 und id 153

1. Die erste Abfrage (SELECT) ruft die Informationen für die trip_id 487868 aus der Tabelle "trips" ab.
2. Mit den SET-Anweisungen werden Variablen definiert und Werte zugewiesen:
3. @g1: Die Variable @g1 erhält die Geometrie (Koordinaten) der Rückgabestation (id=279) aus der Tabelle "stations".
4. @g2: Die Variable @g2 erhält die Geometrie (Koordinaten) der Ausleihstation (id=153) aus der Tabelle "stations".
5. Die nächste SET-Anweisung berechnet die sphärische Distanz (@dist) zwischen den beiden Stationen mit der Funktion ST_Distance_Sphere.
6. Die folgenden SET-Anweisungen @st und @et rufen die Start- und Endzeit des Trips (trip_id=487868) aus der Tabelle "trips" ab.
7. Mit der SET-Anweisung @diff wird die Differenz zwischen der Endzeit und der Startzeit in Minuten berechnet.
8. Die nächste SET-Anweisung @sp berechnet die Geschwindigkeit (Speed) basierend auf der Distanz, der Zeitdifferenz und der Umrechnungsfaktoren. Die Geschwindigkeit wird in km/h berechnet.
9. Die letzten beiden SELECT-Anweisungen geben die berechnete Geschwindigkeit (@sp) und die sphärische Distanz (@dist) aus.

Tabelle 1 MySQL Ergebnisse der berechneten Distanz und Geschwindigkeit zwischen id 279 und id 153 der trip_id 487868

Geschwindigkeit[km/h]	19.625187689143726
Distanz[m]	11775.112613486235

- Erklären Sie in eigenen Worten was Sie unter einem Polygon verstehen und wofür dieses in der Geoinformatik benötigt wird. Beschreiben Sie, wie man in MySQL ein Polygon erzeugt (siehe MySQL 8 Dokumentation-<https://dev.mysql.com/doc/refman/8.0/en/>).

Ein Polygon ist eine geschlossene Form mit Eckpunkten und Linien, die in der Geoinformatik verwendet wird, um Flächen oder Gebiete darzustellen. Es ermöglicht die Erfassung und Speicherung von geografischen Informationen wie Landgrenzen, Gewässer, Stadtgebiete oder administrative Grenzen. Mit Polygonen können räumliche Analysen durchgeführt, Abfragen erstellt und Berechnungen wie Flächenmessungen und Überlappungen von Gebieten durchgeführt werden. In MySQL kann man ein Polygon erstellen, indem man die Koordinaten der Eckpunkte angibt, um es für räumliche Abfragen und Analysen zu nutzen.

POLYGON((x1 y1, x2 y2, x3 y3, ..., xn yn, x1 y1))

Hierbei repräsentieren x1, y1, x2, y2 usw. die Koordinaten der Eckpunkte des Polygons. Die Koordinaten werden paarweise angegeben, wobei x den Längengrad und y den Breitengrad darstellt.

Ein einfaches Beispiel für die Erstellung eines Quadrats als Polygon mit den Eckpunkten (0 0), (0 1), (1 1) und (1 0) sieht folgendermaßen aus:

POLYGON((0 0, 0 1, 1 1, 1 0, 0 0))

```
mysql> SET @poly =
        'Polygon((0 0,0 3,3 0,0 0),(1 1,1 2,2 1,1 1))';
mysql> SELECT ST_Area(ST_GeomFromText(@poly));
+-----+
| ST_Area(ST_GeomFromText(@poly)) |
+-----+
| 4 |
+-----+
```

Abbildung 14 MySQL Code zur Erstellung eines Polygons

- Berechnen Sie die (i) sphärische Distanz zwischen der Ausleih- und Rückgabestation sowie (ii) die Geschwindigkeiten mit der diese Strecke zurückgelegt wurde. Nutzen Sie dafür Python (verwenden Sie das bereitgestellte Template).
- Berechnen Sie auch die geodätische Distanz. Vergleichen Sie die beiden Ergebnisse? Kommen Sie bei beiden Berechnungsarten auf das selbe Ergebnis? Wenn nein, warum nicht?

HINWEIS: Die sphärische Distanz wird auch Großkreisdistanz („great-circle distance“) genannt. Details zur geopy und great-circle distance finden Sie unter (<https://geopy.readthedocs.io/en/stable/>)

Der bereitgestellte Code als Template (Abb. 15) führt die Berechnungen der sphärischen Distanz und Geschwindigkeit zwischen der Ausleih- und Rückgabestation durch. Es werden auch die geodätische Distanz und Geschwindigkeit berechnet.

```
1  """
2  Distance and speed calculation in Python using geopy and shapely
3  Author: Cornelia Schneider
4  Date: 11.05.2020
5  Version: 1.0
6  """
7
8
9  import mysql
10 import geopy
11 # https://geopy.readthedocs.io/en/stable/#module-geopy.distance
12 from shapely import wkt
13 # https://shapely.readthedocs.io/en/latest/manual.html
14
15 """
16 Get points and timestamps for distance calculation from mysql database
17 """
18
19 # Establish mysql connection
20 connection = mysql.connector.connect(host="localhost",
21                                     user="ggis",
22                                     password="ggis",
23                                     db="divvy")
24
25 # Get cursor for reading data
26 cursor = connection.cursor()
27
28 # Define sql statement for receiving the point geometry (geom) for stations id
29 sql = "select ST_AsText(divvy.stations.geometry) as geom from divvy.stations where id = 279"
30 # Execute sql statement
31 cursor.execute(sql)
32 # Receive result
33 result = cursor.fetchall()
34 # Get geometry from query result -> shapely.geometry.point.Point
35 point_1 = wkt.loads(result[0][0])
36
37 # Define sql statement for receiving the point geometry (geom) for stations id
38 sql = "select ST_AsText(divvy.stations.geometry) as geom from divvy.stations where id = 153"
39 # Execute sql statement
40 cursor.execute(sql)
41 # Receive result
42 result = cursor.fetchall()
43 # Get geometry from query result -> shapely.geometry.point.Point
44 point_2 = wkt.loads(result[0][0])
45
46 # Define sql statement for receiving start timestamp
47 sql = "select divvy.trips.starttime from divvy.trips where trip_id = 487868"
48 # Execute sql statement
49 cursor.execute(sql)
50 # Receive result
51 result = cursor.fetchall()
52 # Start timestamp
53 timestamp_1 = result[0][0]
54
55 # Define sql statement for receiving start timestamp
56 sql = "select divvy.trips.stoptime from divvy.trips where trip_id = 487868"
57 # Execute sql statement
58 cursor.execute(sql)
59 # Receive result
60 result = cursor.fetchall()
61 # Stop timestamp
62 timestamp_2 = result[0][0]
63
64 # Time difference
65 difference = timestamp_2 - timestamp_1
66
67 # Close cursor
68 cursor.close()
69 # Close mysql connection
70 connection.close()
```

Abbildung 15 Der bereitgestellte Code als Template

Die Ergebnisse der sphärischen Distanz werden sowohl mit Shapely als auch mit GeoPy berechnet. Dabei wird die Shapely-Bibliothek verwendet, um die Distanz in Grad zu berechnen, während GeoPy die Geodesic- und Great-Circle-Distanzen in Metern berechnet. Die beiden Ergebnisse können unterschiedlich sein, da die Great-Circle-Distanz auf einer idealen Kugeloberfläche basiert, während die Geodesic-Distanz die tatsächliche Form der Erde berücksichtigt. Daher kann die sphärische Distanz von der geodätischen Distanz abweichen.

```

67
68 """
69 |   Examples for distance calculations
70 |   """
71 # Shapely distance calculation -> result in degrees
72 sDistance = point_1.distance(point_2)
73 print("Shapely distance in degrees = " + str(sDistance) + '\n')
74
75 # GeoPy distance calculation(s)
76 # Points in GeoPy are expected in the format lat/lon so new points have to be created
77 npoint_1 = (point_1.y, point_1.x) # (latitude, longitude)
78 npoint_2 = (point_2.y, point_2.x) # (latitude, longitude)
79
80 # Geodesic distance in meters
81 gDistanceMeter = geopy.distance.geodesic(npoint_1, npoint_2).meters
82 print("GeoPy Geodesic distance in meters = " + str(gDistanceMeter) + '\n')
83 # Geodesic distance in kilometers
84 gDistanceKm = geopy.distance.geodesic(npoint_1, npoint_2).kilometers
85 print("GeoPy Geodesic distance in meters = " + str(gDistanceKm) + '\n')
86 # Geodesic distance in miles
87 gDistanceMiles = geopy.distance.geodesic(npoint_1, npoint_2).miles
88 print("GeoPy Geodesic distance in meters = " + str(gDistanceMiles) + '\n')
89
90 # Great-circle distance in meters
91 gcDistanceMeter = geopy.distance.great_circle(npoint_1, npoint_2).meters
92 print("GeoPy great-circle distance in meters = " + str(gcDistanceMeter) + '\n')
93
94 """
95 |   Speed calculation
96 |   """
97 speed_km_h = gcDistanceMeter / tdifference.seconds * 3.6
98 print("Speed in km/h = " + str(speed_km_h) + '\n')

```

Abbildung 16 Python Code zum Berechnen und der Ausgabe der Ergebnisse

Tabelle 2 Ergebnisse der Distanzberechnungen

	Shapely Distanz (Grad)	GeoPy Geodätische Distanz (Meter)	GeoPy Geodätische Distanz (Kilometer)	GeoPy Geodätische Distanz (Meilen)	GeoPy Great-Circle Distanz (Meter)
Beispiel 1	0.04749966135053793	4981.692402740931	4.981692402740931	3.095480147650801	4984.39477819529
Beispiel 2	0.10646969909321714	11762.611294940722	11.762611294940722	7.3089478041616465	11775.155123011511
Geschwindigkeit (km/h)	N/A	N/A	N/A	N/A	11.962547467668696

Wie auf der Tabelle 2 zu sehen ist, unterscheiden sich die Ergebnisse der sphärischen Distanz (Shapely) und der geodätischen Distanz (GeoPy) voneinander. Die sphärische Distanz in Grad beträgt 0.10646969909321714, während die geodätische Distanz in Metern etwa 4981.692402740931 beträgt. Die beiden Werte sind unterschiedlich, da die geodätische Distanz die Krümmung der Erde berücksichtigt, während die sphärische Distanz dies nicht tut und eine vereinfachte Berechnung auf einer idealen Kugeloberfläche durchführt.

- *Visualisieren Sie die Ausleih- und Rückgabestation von trip_id 487868 in QGIS. Messen Sie auch hier mit einem geeigneten Werkzeug die Distanz zwischen den beiden Punkten.*

Um eine Distanzmatrix für zwei Punkte in QGIS zu erstellen, können Sie die folgenden Schritte befolgen:

1. QGIS wird geöffnet die Punkte werden separat in die Karte geladen.(Abb. 17)
2. Das Menü "Vektor" wird angeklickt und "Distance Matrix" wird ausgewählt.(Abb. 18)
3. Der Eingabe-Layer, der die beiden Punkte enthält, wird ausgewählt.
4. Der Ziel-Layer wird ausgewählt, der den beiden Punkten entspricht. Die IDs der Punkte sind 279 und 153.
5. Der Ausgabepfad und der Name für die Distanzmatrix-Datei werden angegeben.
6. Durch Klicken auf "OK" wird die Berechnung der Distanzmatrix gestartet.
7. Nach Abschluss der Berechnung wird die Distanzmatrix als Tabelle angezeigt.(Abb.19)

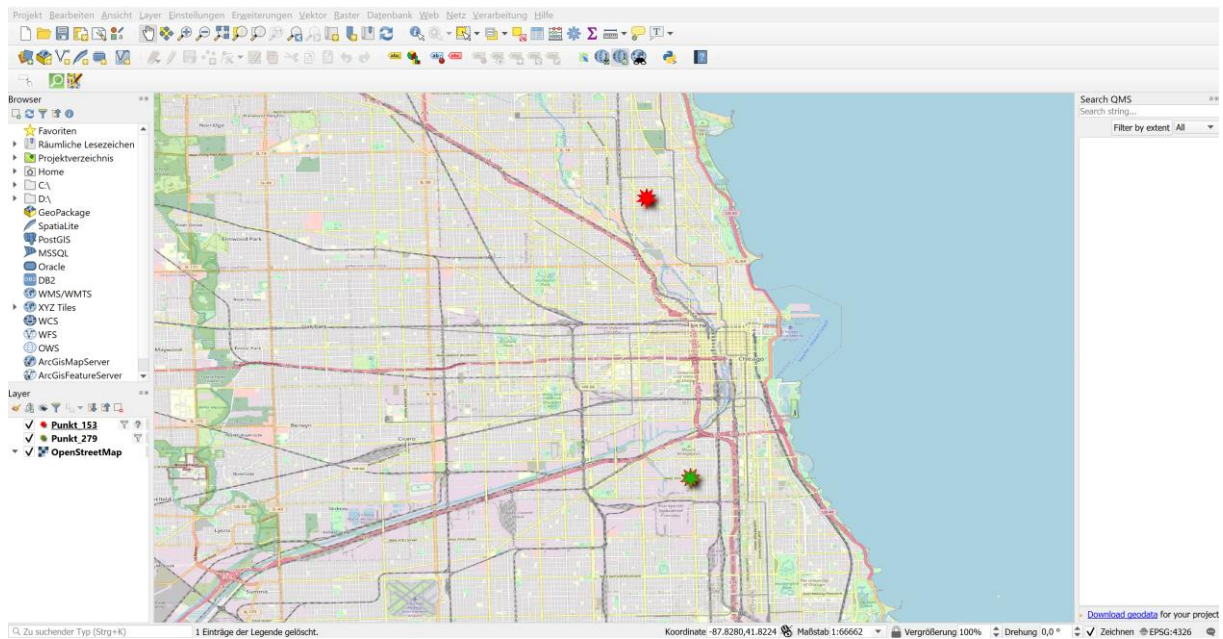


Abbildung 17 Punkt 153 und Punkt 279 auf der OpenStreetMap aufgeladen

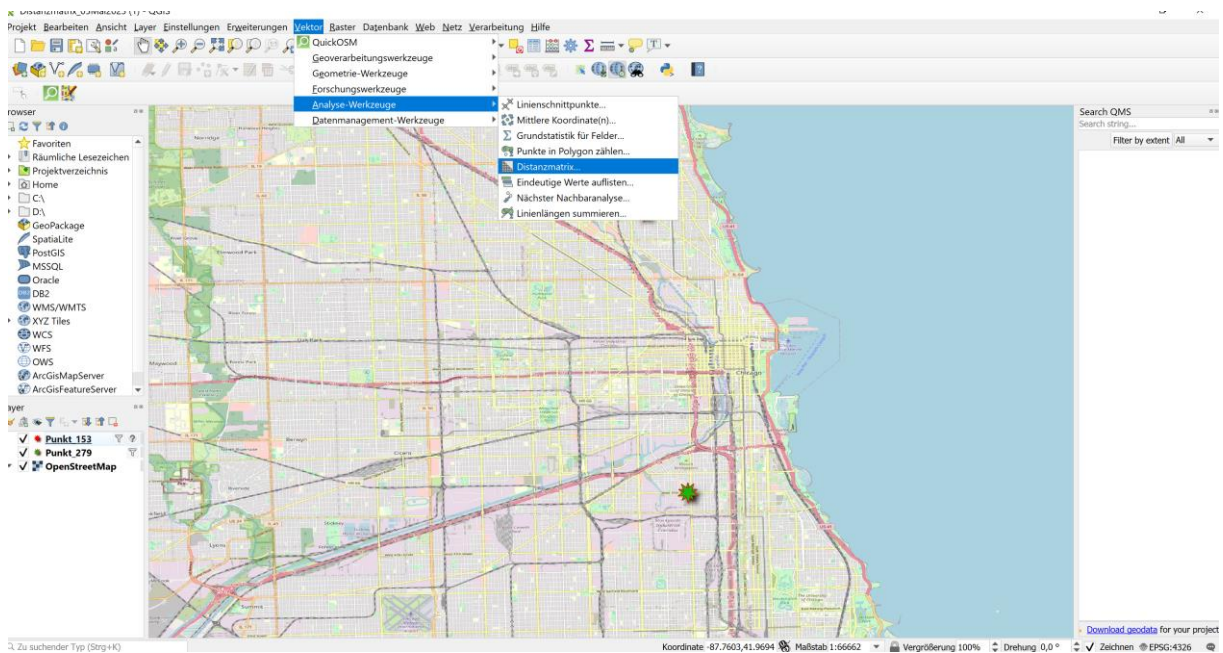


Abbildung 18 Erstellen eines Distanzmatrix

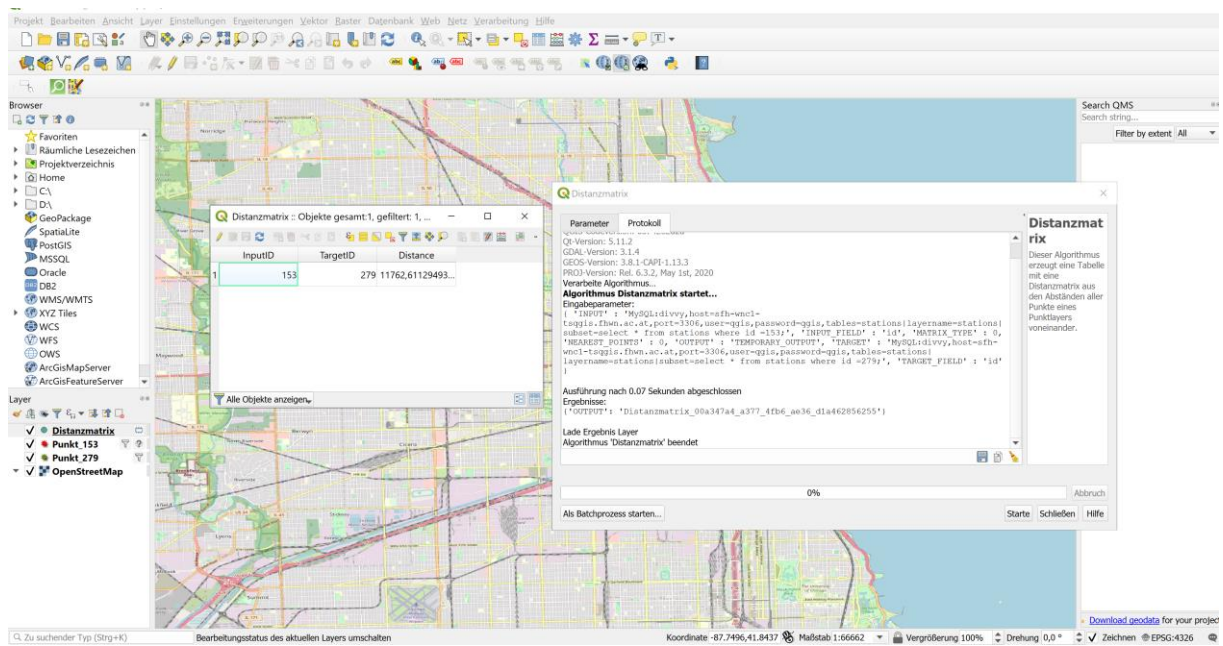


Abbildung 19 Ergebnistabelle des Distanzmatrix

Abbildungsverzeichnis

Abbildung 1	Aufzeichnung einer Strecke in Wieselburg	2
Abbildung 2 a)	Exportieren der Datei	2
Abbildung 3	Vorbereitung OpenStreetMap	3
Abbildung 4	Neuer Vektorlayer hinzufügen	3
Abbildung 5	Auswahl der GPX Datei	4
Abbildung 6	Heranzoomen auf den GPX-Track	4
Abbildung 7	Vorschau der Streckenlänge	5
Abbildung 8	Start- und Endzeitpunkt	6
Abbildung 9	MyMaps Webseite	7
Abbildung 10	Importieren der GPX Datei auf MyMaps Webseite	7
Abbildung 11	Visualisierung der Strecke mittels MyMaps	8
Abbildung 12	Visualisierung der Länge und Geschwindigkeit	8
Abbildung 13	MySQL Code zum Berechnen der Distanz zwischen id 279 und id 153	9
Abbildung 14	MySQL Code zur Erstellung eines Polygons	10
Abbildung 15	Der bereitgestellte Code als Template	10
Abbildung 16	Python Code zum Berechnen und der Ausgabe der Ergebnisse	11
Abbildung 17	Punkt 153 und Punkt 279 auf der OpenStreetMap aufgeladen	12
Abbildung 18	Erstellen einer Distanzmatrix	12
Abbildung 19	Ergebnistabelle der Distanzmatrix	13