

Tourenvorschläge für Mountainbiker aus Bewegungsdaten



Bachelorarbeit

Von

**Kessler Sebastian**

Bachelorstudiengang 2014

Abgabedatum 07.09.2017

Studienrichtung Umweltingenieurwesen

Fachkorrektoren:

Dr. Laube, Patrick

ZHAW, Grüental, 8820 Wädenswil

Ratnaweera, Nils

ZHAW, Grüental, 8820 Wädenswil

**Zitiervorschlag:** Kessler, S. (2017). Tourenvorschläge für Mountainbiker aus Bewegungsdaten. Bachelorarbeit, ZHAW Departement N, Wädenswil

**Schlagwörter (Keywords):** Bewegungsdaten, Map-Matching, personalisierter Tourenvorschlag, Mountainbike, Segmentierung

**Adresse:** Kessler Sebastian, 6300 Zug, Email: [sebastian\\_kessler@hotmail.com](mailto:sebastian_kessler@hotmail.com)

## Zusammenfassung

Der Mountainbike-Sport erfreut sich steigender Beliebtheit und ist in Bergregionen längst ein wichtiger Wirtschaftsfaktor. Mountainbike-spezifische Zeitschriften, Internetseiten und Smartphone-Apps bieten unzählige Informationen über Mountainbike-Touren an. Obwohl sich bei Onlineplattformen Suchfilter einsetzen lassen, ist es oft schwierig und zeitaufwändig eine Mountainbike-Tour nach seinem eigenen Geschmack zu finden.

In den verschiedensten Bereichen nutzen Menschen GPS-Sensoren, um ihre Bewegungsdaten mittels GPS aufzuzeichnen. Aus den aufgezeichneten raumzeitlichen Daten können mittels moderner Methoden der Bewegungsanalyse nützliche Informationen gewonnen werden. Im Fall des Mountainbikings können relevante Toureninformationen aus den GPS-Daten abgeleitet und daraus ein personalisiertes Tourenprofil erstellt werden. Dieses kann wiederum verwendet werden, um personalisierte Tourenvorschläge zu unterbreiten.

In dieser Arbeit wurde ein Algorithmus entwickelt, welcher aus GPS-Tracks relevante Tourenparameter ableitet und daraus personalisierte Tourenprofile erstellt. Mittels Tourenprofil kann schliesslich aus einer beliebigen Auswahl an Mountainbike-Touren ein personalisierter Tourenvorschlag unterbreitet werden.

In einem ersten Schritt wurden die GPS-Tracks mit Geodaten, berechneten Umweltfaktoren und Bewegungsparametern angereichert. In einem zweiten Schritt wurden die GPS-Tracks anhand der Umweltfaktoren und Bewegungsparametern segmentiert. Aus den segmentierten GPS-Tracks wurden verschiedene Tourenparameter berechnet aus denen schliesslich ein personalisiertes Tourenprofil erstellt wurde. In einem letzten Schritt wurden zur Auswahl stehende Mountainbike-Touren mit dem Tourenprofil des Mountainbikers verglichen und anhand eines Punktesystems ein personalisierter Tourenvorschlag unterbreitet.

---

## **Abstract**

In the last decade mountain biking has become a very popular sport. For mountain regions it has turned into an important economic factor. Targeted mountain biking magazines, websites and smartphone apps provide a wide range of information about mountain bike trails and tours. Although there is the possibility to apply search filters, it remains difficult to find a mountain bike tour according to one's own preferences.

People use GPS-devices in many different areas to trace their movements while jogging or cycling. This spatio-temporal data can be processed by modern methods of movement analysis. Through the movement analysis, mountain biking specific information can be gained and used to draw up a mountain biker's specific tour profile. A personalised mountain bike tour profile can then be used to develop personalised tour recommendations.

For this thesis a special algorithm was developed which uses GPS tracks to deduce tour parameters, draw up personalised mountain bike tour profiles and make individual tour recommendations.

In step one, the GPS-tracks were enriched with additional geo data, environmental factors and movement parameters. In step two, the GPS-tracks were segmented by the environmental factors and movement parameters. From the segmented GPS-tracks, different mountain biking tour parameters were then calculated and, in the end, merged to personal tour profiles. In the last step, a variety of mountain bike tours were compared with the personalised tour profiles. Based on a special score system, personalised tour recommendations were made.

---

# Inhaltsverzeichnis

1	Einleitung.....	8
1.1	Ausgangslage.....	8
1.2	Problemstellung und Forschungsfragen.....	8
1.3	Ziele.....	9
1.4	Aufbau der Arbeit.....	9
2	Literaturübersicht.....	10
2.1	Relevante Toureninformationen.....	10
2.2	Map-Matching.....	11
2.3	Ähnlichkeit von Trajektorien.....	14
3	Material.....	15
3.1	Datengrundlage.....	15
3.1.1	Bewegungsdaten.....	15
3.1.2	GPS-Daten Mountainbike-Touren.....	15
3.1.3	Geodaten.....	15
3.2	Verwendete Programme.....	16
4	Methoden.....	17
4.1	Ablauf.....	17
4.2	Datenaufbereitung.....	17
4.3	Map-Matching.....	18
4.3.1	Datenaufbereitung für Map-Matching.....	18
4.3.2	Punkte auf kürzeste ermittelte Route einpassen.....	21
4.4	Methoden zur Berechnung zusätzlicher Attribute in R.....	22
4.4.1	Mittlerer Höhenunterschied.....	23
4.4.2	Markierung von Anstiegen, Abfahrten und flachen Abschnitten.....	23
4.4.3	Gefälle.....	24

4.4.4	Verkehrsmittel.....	24
4.4.5	Übergänge.....	25
4.5	Segmentierung .....	25
4.6	Erstellen personalisierter Tourenprofile.....	25
4.7	Personalisierte Tourenvorschläge unterbreiten.....	28
4.8	Validierung der Tourenparameter, personalisierten Tourenprofile und Tourenvorschläge	29
5	Experimente.....	30
5.1	Segmentierung: Anstiege, Abfahrten und flache Abschnitte.....	30
5.2	Vergleich von Gesamtdistanz, Strecke auf Singletrails und Höhenmeter bergauf mit Webdienst .....	30
6	Ergebnisse.....	32
6.1	Segmentierung: Anstiege, Abfahrten und flache Abschnitte.....	32
6.2	Segmentierung der Verkehrsmittel.....	35
6.3	Vergleich von Gesamtdistanz, Strecke auf Singletrails und Höhenmeter bergauf mit Webdienst .....	36
6.4	Personalisierte Mountainbike-Tourenvorschläge.....	38
7	Diskussion .....	42
7.1	Segmentierung: Anstiege, Abfahrten und flache Abschnitte.....	42
7.2	Segmentierung der Verkehrsmittel.....	42
7.3	Vergleich von Gesamtdistanz, Strecke auf Singletrails und Höhenmeter bergauf mit Webdienst .....	43
7.4	Personalisierte Mountainbike-Tourenvorschläge.....	43
8	Schlussfolgerung .....	45
9	Dank .....	46
10	Literaturverzeichnis .....	47
11	Abbildungsverzeichnis.....	50
12	Tabellenverzeichnis.....	53

13 Anhang.....54

# 1 Einleitung

## 1.1 Ausgangslage

Mountainbiking hat sich seit der Jahrtausendwende von einem Trendsport zu einem Massensport entwickelt. Der enorme technische Fortschritt bei den Mountainbikes führte dazu, dass Mountainbiking in immer unwegsamem Gelände möglich wurde und heute sogar auf alpinen Pfaden gefahren werden kann. Über 44% der Schweizerinnen und Schweizer üben Radfahren als Sport aus, wobei 6.3% davon explizit Mountainbiking nennen (Lamprecht, Fischer, & Stamm, 2014). Die stetig wachsende Anzahl an Mountainbiker führt auch dazu, dass immer mehr Mountainbike-Trails gebaut werden und Tourismusregionen spezielle Touren und Angebote für Mountainbiker entwickeln. Zeitschriften, Internetseiten und Smartphone-Apps bieten eine Fülle an Informationen über Mountainbike-Touren an. Obwohl sich bei Onlineplattformen Suchfilter einsetzen lassen, ist es oft schwierig und zeitaufwändig eine Mountainbike-Tour nach seinem eigenen Geschmack zu finden. Apps lassen eine Suche von Touren im Umkreis des aktuellen Standorts zu. Aus den zahlreichen angezeigten Touren ist es dann allerdings mühsam, die passende Tour zu eruieren. Dies auch deshalb, weil Mountainbiking ein sehr dehnbarer Begriff ist. Was einige für eine normale Radtour halten, ist für andere bereits eine Mountainbike-Tour.

In den verschiedensten Bereichen nutzen Menschen ihr Smartphone oder andere Geräte, um ihre Bewegungsdaten mittels GPS aufzuzeichnen. Die Daten werden verwendet, um beispielsweise den Trainingsfortschritt zu dokumentieren oder neu entdeckte Touren mit anderen zu teilen. Krankenkassen ermöglichen ihren Versicherten bereits eine Prämienreduktion, wenn sie ihnen ihre persönlichen Daten zur Verfügung stellen und somit nachweisen können, dass sie sich genügend bewegen. Bewegungsprofile können auch im Bereich des Marketings verwendet werden. Ellersiek, Liebig, Hecker, & Körner (2012) verwendeten beispielsweise Bewegungsprofile von Besuchern im Duisburger Zoo, um eine Standortbewertung neuer und bestehender Gehege durchzuführen. Im Bereich des Mountainbikings könnten die Bewegungsinformationen genutzt werden, um Tourenpräferenzen festzustellen.

In der vorliegenden Arbeit wird versucht, anhand von aufgezeichneten GPS-Tracks verschiedener Mountainbiker jeweils ein Tourenprofil zu erstellen. Die Erstellung der Tourenprofile soll mittels eigens entwickeltem Algorithmus möglichst automatisiert erfolgen. Anschliessend werden die Tourenprofile mit einer Liste von verschiedenen Mountainbike-Touren verglichen. Durch den Vergleich soll die für den jeweiligen Mountainbiker passendste Tour ermittelt werden.

## 1.2 Problemstellung und Forschungsfragen

Für diese Arbeit ergeben sich folgende Forschungsfragen:



- FF1: Wie können Mountainbike-Touren anhand von Geo- und Metadaten kategorisiert werden?
- FF2: Wie können aus analysierten Touren die Präferenzen eines Mountainbikers ermittelt werden?
- FF3: Unterscheiden sich die Tourenprofile verschiedener Mountainbiker signifikant?
- FF4: Wie können aus Präferenzen passende Mountainbike-Touren ausgewählt werden?
- FF5: Wie lässt sich die Ähnlichkeit zwischen Mountainbike-Touren quantifizieren?

### **1.3 Ziele**

In der vorliegenden Arbeit wird versucht, anhand von aufgezeichneten GPS-Tracks verschiedener Mountainbiker jeweils ein Tourenprofil zu erstellen. Die Erstellung der Tourenprofile soll mittels eigens entwickeltem Algorithmus möglichst automatisiert erfolgen. Die Tourenprofile werden anschliessend mit einer Liste von verschiedenen Mountainbike-Touren verglichen. Durch den Vergleich soll die, für den jeweiligen Mountainbiker, passendste Tour ermittelt werden.

### **1.4 Aufbau der Arbeit**

Im Kapitel 2 wird die vorhandene Literatur zu den Themen relevante Toureninformationen, Map-Matching und Ähnlichkeit von Trajektorien vorgestellt. Die Literaturübersicht zum Thema Map-Matching bezieht sich fast ausschliesslich auf das Post-Map-Matching bei dem die Karteneinpassung erst vorgenommen wird, wenn Anfangs- und Endpunkt eines GPS-Tracks bekannt sind. Map-Matching bei dem die Karteneinpassung während der Aufzeichnung der GPS-Punkte stattfindet, ist für die vorliegende Arbeit irrelevant. Kapitel 3 beschreibt die Datengrundlage und es werden die verwendeten Programme aufgelistet. Die detaillierte Datenaufbereitung und Erstellung der Tourenprofile wird im Kapitel 4 erklärt. Auf die Berechnungen einfacher Umweltfaktoren und Bewegungsparameter wird nicht eingegangen. Diese wurden nach dem Vorbild von Tomlinson (2015) berechnet. Das Kapitel 5 beschreibt zwei durchgeführte Experimente. Eines der Beiden dient der Kalibrierung des entwickelten Algorithmus. Das andere Experiment soll die Werte dreier Tourenparameter mit den berechneten Werten eines Webdienstes vergleichen und dadurch die Plausibilität der Tourenparameter prüfen. In den Kapiteln 6 und 7 werden die Resultate des Algorithmus und der Experimente dargestellt und diskutiert. Schlussfolgerungen aus den Ergebnissen und weiterführende Untersuchungen werden im Kapitel 8 festgehalten.

## 2 Literaturübersicht

In einem ersten Teil wurde die Literatur auf relevante Toureninformationen durchforscht, welche schliesslich dem Aufbau des Algorithmus dienen sollte. Als relevante Toureninformationen werden Angaben bezeichnet, aus denen der Mountainbiker Rückschlüsse auf die benötigte Kondition, Fahrtechnik und Ausrüstung machen kann. Wissenschaftliche Artikel zum Thema „relevante Toureninformationen“ existieren allerdings wenig oder gar nicht. Deshalb besteht das nachfolgende Unterkapitel lediglich aus einer Sammlung an Informationen aus Fachbüchern, Magazinen, Websites und Smartphone-Apps. In einem zweiten Teil wird die vorhandene Literatur zum Thema Map-Matching untersucht. Das Map-Matching wird in dieser Arbeit verwendet, um später den Wegtyp aus den Geodaten zu extrahieren. Der dritte Abschnitt befasst sich mit der Ähnlichkeit von Trajektorien. In der vorliegenden Arbeit repräsentieren die GPS-Tracks der Mountainbiker die Trajektorien.

### 2.1 Relevante Toureninformationen

Tabelle 1: Durchsuchte Medien nach relevanten Toureninformationen. Die Toureninformationen sind nach absteigender Häufigkeit des Auftretens in den untersuchten Quellen sortiert.

Quelle	Medium	Distanz	Höhenmeter Aufstieg	Höhenmeter Abfahrt	Fahrzeit	Region	Schwierigkeit	Höhenlage	Beste Jahreszeit	Anfahrtsweg	Aussicht	Gefälle	Wegtyp/Wegtypanteil	Einkehrmöglichkeit	Fahrstil
Buschor, Jürg	Fachliteratur	x	x	x	x	x	x			x					
Swiss Singletrail Map	Fachliteratur	x	x	x	x	x	x	x							
Supertrail Map	Fachliteratur	x	x	x	x	x	x	x							
Ride - Bikestyle Magazin	Fachzeitschrift	x	x						x						
Mountainbike-Magazin	Fachzeitschrift	x	x	x	x			x	x						
Outdooractive	App/Website	x	x	x	x	x	x	x	x	x	x			x	
<a href="http://www.strava.com">www.strava.com</a>	App/Website	x	x	x		x						x			
<a href="http://www.trailforks.com">www.trailforks.com</a>	App/Website	x	x	x	x	x	x	x				x	x		x
<a href="http://www.komoot.com">www.komoot.com</a>	App/Website	x	x	x	x	x	x			x			x		
<a href="http://www.mountainbiker.it">www.mountainbiker.it</a>	Website	x	x	x	x				x						x
<a href="http://www.lenzerheide.ch">www.lenzerheide.ch</a>	Website	x	x	x	x	x	x	x	x	x	x			x	
<a href="http://www.mountainbikeland.ch">www.mountainbikeland.ch</a>	Website	x	x	x						x			x		
<a href="http://www.davos.ch">www.davos.ch</a>	Website	x	x	x											
<a href="http://www.gps-tracks.com">www.gps-tracks.com</a>	Website	x	x	x	x	x	x	x			x				
<a href="http://www.mountainbiker.ch">www.mountainbiker.ch</a>	Website	x	x	x	x	x	x	x	x		x	x			

Die vorherige Tabelle 1 zeigt eine Auflistung der durchforschten Medien und deren verwendeten Toureninformationen.

Die Distanz wird in allen untersuchten Medien angegeben und ist der grundlegendste Tourenparameter. Ein weiterer grundlegender Parameter ist die Höhendifferenz. Meistens wird diese aufgeteilt in Höhenmeter bergauf und bergab. Die beiden Parameter Distanz und Höhenmeter dienen vor allem der Einschätzung der benötigten Kondition und Fahrzeit. In den häufigsten Fällen wird die benötigte Fahrzeit zusätzlich aufgelistet. Selten existiert eine separate Skala für die Ansprüche an die Kondition.

Eine Angabe zum Gefälle gibt es nur bei Strava Inc. (2017), Pinkbike.com (2017) und Mountainbiker.ch (2017). Das Gefälle ist eine Art Schwierigkeitsangabe für Anstiege und Abfahrten, wobei bei Anstiegen die Schwierigkeit mit steigendem Gefälle schneller zunimmt als bei Abfahrten. In der Singletrail-Skala von Schymik, Harald, & Werner (2008) ist das Gefälle eine zentrale Kenngrösse für die fahrtechnische Schwierigkeit. Nebst Angaben zum Gefälle besitzt über die Hälfte der untersuchten Quellen Schwierigkeitsskalen, die die technische Schwierigkeit der Mountainbike-Tour angeben (Buschor, 2011; Komoot GmbH, 2017; Lenzerheide Marketing und Support AG, 2017; Mountainbiker.ch, 2017; Outdooractive GmbH, 2017; Outdooractive GmbH & Co. KG, 2017; Outkomm GmbH, 2014; Pinkbike.com, 2017; Swiss Sports Publishing GmbH, 2015). Gruber, 2017; Pinkbike.com (2017) kategorisieren Touren auch nach dem Fahrstil. Es wird zwischen All Mountain, Cross Country, Enduro, Freeride, Downhill und Bikepark unterschieden. Die Unterscheidung nach Fahrstil ist einerseits eine Empfehlung bezüglich benötigtem Material (Fahrradtyp, Protektoren, etc.), andererseits können daraus die Ansprüche an das Fahrkönnen abgeschätzt werden. Eine weitere indirekte Angabe zum Schwierigkeitsgrad ist der Wegtyp und dessen Anteil an der Mountainbike-Tour. Mit steigendem Anteil an Singletrails, steigt in der Regel auch der technische Schwierigkeitsgrad (Komoot GmbH, 2017; Stiftung SchweizMobil, 2017).

Bei über der Hälfte der untersuchten Medien werden Mountainbike-Touren nach Regionen eingeteilt (Buschor, 2011; Komoot GmbH, 2017; Lenzerheide Marketing und Support AG, 2017; Mountainbiker.ch, 2017; Outdooractive GmbH & Co. KG, 2017; Strava Inc., 2017) und teilweise mit zusätzlichen Informationen wie Aussicht, Anfahrtsweg, Höhenlage und Einkehrmöglichkeiten versehen.

Je nach Zielpublikum der untersuchten Medien erhält der Leser eher Informationen zu den fahrtechnischen Ansprüchen und Wegbeschaffenheit oder erlebnissteigernde Hinweise wie Aussicht oder Angaben zu empfehlenswerten Restaurants und Hotels.

## **2.2 Map-Matching**

In der heutigen Zeit werden immer häufiger Positionsdaten benötigt. Quddus, Ochieng, & Noland (2007) behandeln das Thema Karteneinpassung (engl. Map matching) in Bezug auf Anwendungen und Dienstleistungen im Bereich intelligenter Transportsysteme wie z.B. Busankunftsinformationen oder Wegführung.

Als Karteneinpassung gilt das Einpassen von Positionsdaten auf einem Wegenetz. Positionsdaten werden hierbei von GPS oder GPS in Kombination mit DR (Deduced Reckoning, «Dead» Reckoning) verwendet. Die Karteneinpassung hat das Ziel, Fehler des Positionierungssystems zu korrigieren, um den Aufenthaltsort eines Objekts genauer zu lokalisieren. Voraussetzung dafür ist, dass es sich um ein Objekt handelt, welches sich auf dem verwendeten Wegenetz bewegt. Für eine hohe Genauigkeit der Karteneinpassung sind hochaufgelöste Wegenetzkarten essentiell.

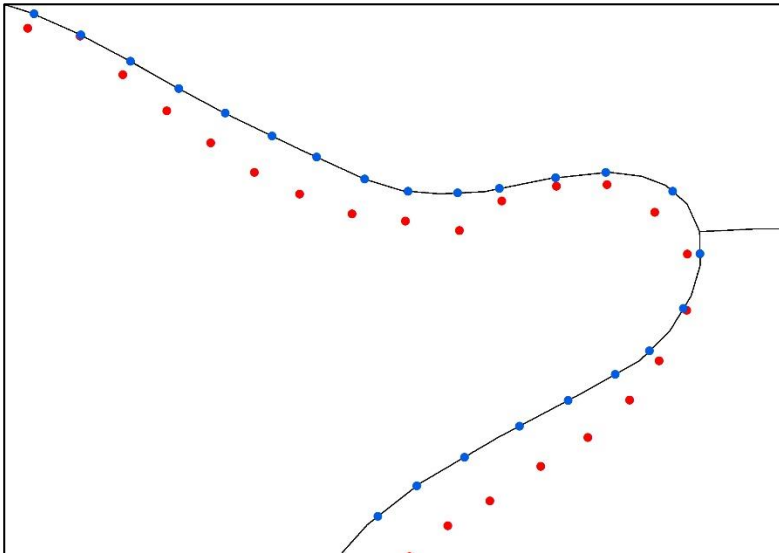


Abbildung 1: Beispiel für eine Karteneinpassung (Map-Matching). In Rot: Original GPS-Punkte. In Blau: Eingepasste GPS-Punkte.

Bei Karteneinpassungsalgorithmen gilt es zwischen Post-Einpassungen und Echtzeit-Einpassungen zu unterscheiden. Quddus et al. (2007) untersuchten aktuelle Map-Matching-Algorithmen für Transportapplikationen. Sie erforschten dabei nur Echtzeit-Karteneinpassungen da Post-Einpassungen in Bezug auf intelligente Transportsysteme unnützlich sind. Weiter werden Karteneinpassungsalgorithmen in simple Suchalgorithmen und fortgeschrittene Suchalgorithmen eingeteilt. Zu den simplen Algorithmen zählen geometrische, topologische und wahrscheinlichkeitsbasierte Algorithmen. Fortgeschrittene Algorithmen verwenden erweiterte Funktionen wie Kalman Filter, Fuzzylogik und Belief Theory.

Marchal, Hackney, & Axhausen (2004) entwickelten einen Map-Matching-Algorithmus für grosse GPS-Datensätze. Sie halten fest, dass intuitive Methoden wie „nächstgelegener Punkt“ oder „nächstgelegene Linie“ schnelle Algorithmen sind. Diese zwei Methoden betrachten jedoch nicht den kompletten Pfad und ignorieren die Beziehungen zwischen den Punkten der einzelnen Untersequenzen. Aus diesem Grund arbeiteten die Autoren einen Algorithmus, der die Netzwerktopologie miteinbezieht indem sie eine „Multiple Hypothesis Technique“ anwendeten. Ihr Ziel war ein Post-Map-Matching Algorithmus, der grosse Datensätze in einer möglichst kurzen Zeit bearbeiten kann. Der Algorithmus sucht die Route aus einem Netzwerk, die den kleinsten Abstand zu den GPS-Punkten hat. In einem ersten Schritt werden die „n“ vielen Abschnitte gesucht, die dem ersten GPS-Punkt am nächsten sind. Die Anzahl Abschnitte n werden Anfangs eingestellt.

Jeder dieser n-Abschnitte ist nun auch eine Route. Jede dieser Routen besitzt einen Entfernungswert. Als Entfernungswert gilt der Abstand eines eingepassten GPS-Punktes zu dem jeweiligen Abschnitt. Abstände der nachfolgenden GPS-Punkten werden jeweils addiert. Als nächster Schritt wird der zweite GPS-Punkt hinzugefügt und die bestehenden n-Routen anhand der Anzahl ausgehender neuer Abschnitte kopiert. Nun wird der Abstand des zweiten GPS-Punktes zu den neuen Abschnitten berechnet und zum Entfernungswert der jeweiligen Route addiert. Falls der Entfernungswert einer kopierten Route jetzt kleiner ist, als der Wert der ursprünglichen Route, wird die Kopie-Route entsprechend der Häufigkeit ihrer ausgehenden Abschnitte kopiert und der Abstand des zweiten GPS-Punktes zu den neuen Abschnitten wird berechnet. Dieser Vorgang wird solange wiederholt, bis der Abstand des zweiten GPS-Punktes zu allen ausgehenden Abschnitten grösser ist, als der Abstand zum Vorherigen. Diese Wiederholungen sind erforderlich, weil Abschnitte auftreten können auf die kein GPS-Punkt eingepasst wird. Anschliessend werden die Entfernungswerte aller bestehenden Routen miteinander verglichen und nur die n-Anzahl Routen behalten, welche die kleinsten Entfernungswerte aufweisen. Alle übrigen Routen werden verworfen. Beim dritten GPS-Punkt werden nun die vom vorherigen Schritt übriggebliebenen n-Routen und alle ihre ausgehenden Abschnitte untersucht. Dabei wird wie beim zweiten GPS-Punkt vorgegangen bis alle GPS-Punkte eingepasst sind. Der vorgestellte Algorithmus wurde in Java programmiert. Skuballa (2015) entwickelte diesen Algorithmus weiter, um gemessene Mobilfunknetzwerkcharakteristiken auf einer Karte einzupassen.

Dalumpines & Scott (2011) erstellten, im Vergleich zu vielen anderen, einen GIS-basierten Karteneinpassungsalgorithmus für Transportforschungsanwendungen. Der Algorithmus nutzt geometrische, Puffer- und Netzwerkfunktionen eines GIS für die nachträgliche Einpassung von GPS-Daten. Der Kern des von Dalumpines & Scott (2011) entwickelten Karteneinpassungsalgorithmus ist der Gebrauch des „Route Analysis“-Tool von ArcGIS® (Network Analyst Erweiterung). Die Einpassung erfolgt in fünf Schritten. Zuerst wird der GPS-Punktepfad in eine Polyline umgewandelt. Der erste und letzte Punkt der Sequenz dienen als Start- und Endpunkt für das „Network Analyst“-Tool in ArcGIS®. Als zweites wird ein selbst definierter Puffer um das Polyline-Element gelegt. In der Regel entspricht die Pufferdistanz dem 5 bis 6fachen der GPS-Genauigkeit. Im dritten Schritt werden Stops und der Rahmen für das „Route-Solver“-Tool festgelegt. Die Stops werden aus dem GPS-Punktepfad generiert. Sie dienen dem „Network Analyst“-Tool als Lokalitäten zwischen denen die kürzeste Route berechnet werden muss. Der Rahmen wird durch die Schnittpunkte der Puffergrenzen mit den umgebenden Teilen des Strassennetzes gegeben. Der Rahmen stellt die Genauigkeit und Effizienz des „Shortest Path“-Algorithmus sicher. Im vierten Schritt wird die betrachtete Route durch den „Shortest Path“-Algorithmus generiert, sprich die Karteneinpassung wird vorgenommen. Im fünften und letzten

Schritt werden die Attributtabelle aktualisiert. Mittels Python-Programmierung wurden die einzelnen Schritte in einem einzigen Tool vereint.

## 2.3 Ähnlichkeit von Trajektorien

Laube (2014) bezeichnet den Vergleich der Ähnlichkeit von Trajektorien und anderen Spuren sich bewegenden Objekten als herausforderndes Problem. Trajektorien können sich signifikant in Länge oder Ausdehnung, Form, Richtung und Granularität unterscheiden. Weiter existieren verschiedene Auffassungen, was als ähnliche Trajektorie bezeichnet werden kann. Man könnte die Spuren von sich bewegenden Objekten als ähnlich bezeichnen, wenn sie die gleiche oder ähnliche Form haben. Die abgeleiteten Bewegungsparameter können sich ähnlich sein, wenn sie beispielsweise die gleiche Durchschnittsgeschwindigkeit haben oder Trajektorien können sich gleichen, wenn sie die gleichen Orte besucht haben.

Dodge, Weibel, & Laube (2009) erforschten die Ähnlichkeiten von dynamischen Verhalten von sich bewegenden Objekten. Die Absicht war die Entwicklung eines Konzepts um Gemeinsamkeiten von Trajektorien sich bewegendem Objekte mittels quantitativen Analysen zu extrahieren. Dabei wurde eine 4-Schritte-Methode entwickelt um Ähnlichkeit von Trajektorien zu evaluieren. Im ersten Schritt wurden die Grundzüge der Bewegung an sich betrachtet und beschrieben. Bewegungsparameter wie Geschwindigkeit, Beschleunigung, Drehung, Geradlinigkeit, etc. wurden als Fundament für die konzeptionelle Bewegungsstruktur verwendet. Konzeptionelle Bewegungsstruktur meint erstens alle Elemente, welche die Bewegung von verschiedenen Objekten beschreiben, und zweitens die Klassifikationen und Definitionen von Bewegungsmustern. Der zweite Schritt der 4-Schritte-Methode befasst sich mit der Extraktion der Bewegungselemente. Dazu wurden die Daten von Rauschen und Positionsfehlern befreit und durch Filterung, Glättung und Re-sampling aufbereitet. Danach wurden globale Parameter berechnet und lokale Eigenschaften extrahiert. Für die Extraktion der lokalen Eigenschaften wurden die Trajektorien in bedeutsame Segmente gleichförmiger Bewegungscharakteristiken zerlegt. Im 3. Schritt wurde eine Methode zur Suche von raum-zeitlichen Ähnlichkeiten entwickelt. Hierfür wurden Bewegungsprofile in einem multidimensionalen Raum einander gegenübergestellt und der Euklidische Abstand der Profile zueinander berechnet. Überschreitet der Abstand einen Schwellenwert, so werden die zwei Trajektorien als ungleich angesehen.

## 3 Material

### 3.1 Datengrundlage

#### 3.1.1 Bewegungsdaten

Die Bewegungsdaten stammen von verschiedenen männlichen Mountainbiker unterschiedlichen Alters. Die jeweiligen Personen wählten jeweils um die 10 für sie typische Mountainbike-Touren aus. Um die Daten aufzuzeichnen, verwendeten sie verschiedene Geräte mit unterschiedlichen Aufzeichnungsgeschwindigkeiten. Alter und verwendeter Gerätetyp sind nicht bekannt. Die Mountainbiker stammen aus unterschiedlichen Schweizer Kantonen. Es ist bekannt, dass Mountainbiker *B, C, E, F, G* und *H* aus dem Kanton Zug stammen, Mountainbiker *A, J* und *K* aus dem Kanton Bern und Mountainbiker *D* aus dem Kanton Zürich. Mountainbiker *B, C, E, F,* und *H* sind fast wöchentlich auf gemeinsamen Touren unterwegs. Ihre aufgezeichneten Touren sind deshalb teilweise identisch. Für jedes GPS-Gerät der Mountainbiker wurde aus den aufgezeichneten GPS-Tracks die durchschnittliche Aufzeichnungsrate berechnet (Tabelle 2).

*Tabelle 2: Durchschnittliche Aufzeichnungsrate der GPS-Geräte von jedem Mountainbiker in Anzahl GPS-Punkte pro Minute.*

Mountainbiker	Anzahl aufgezeichnete Touren	Aufzeichnungsrate (GPS-Punkte/Minute)
A	10	14
B	10	16
C	9	60
D	11	13.6
E	9	21.1
F	10	59.6
G	10	14
H	12	60
J	10	10.6
K	10	25.9

#### 3.1.2 GPS-Daten Mountainbike-Touren

Die GPS-Tracks für die Tourenvorschläge stammen von der Webseite [www.outdooractive.com](http://www.outdooractive.com) (Outdooractive GmbH & Co. KG, 2017). Die Tourenvorschläge stammen alle aus der Gegend Flims im Kanton Graubünden. Bei der Wahl der Touren wurde auf eine gute Durchmischung von Länge, Schwierigkeit und Höhenmeter geachtet.

#### 3.1.3 Geodaten

Nachfolgende Tabelle 3 gibt Aufschluss über die verwendeten Geodaten.

Tabelle 3: Verwendete Geodaten; in der Tabelle ist der Name des Datensatzes aufgelistet, der Datentyp, Beschreibung zum Datensatz und der Datenherr.

Datensatz	Datentyp	Beschreibung	Datenherr
pixelkarte25	Raster	Übersichtskarte	Swisstopo
swissALTI3D	Raster	Höhenmodell	Swisstopo
swissTLM3D_Siedlungsname	Vektor	Namen von Ortschaften. Polygone zeigen den Gültigkeitsbereich des Ortsnamen an.	Swisstopo
swissTLM3D_Strassen	Vektor	Strassen und Wege	Swisstopo
swissTLM3D_Eisenbahn	Vektor	Eisenbahnen	Swisstopo
swissTLM3D_UEBRIGE_BAHN	Vektro	Seilbahnen, etc.	Swisstopo
Highway	Vektor	Strassen und Wege	Openstreetmap

### 3.2 Verwendete Programme

Für die Bearbeitung der GPS-Daten und deren anschliessende Verarbeitung wurden die Programme ArcGIS (ESRI, 2015) und RStudio (RStudio, 2016) verwendet.



## 4 Methoden

Im Kapitel 4 werden die Methoden zur Auf- und Vorbereitung und Datenanreicherung beschrieben sowie die anschliessende Verarbeitung der Daten.

### 4.1 Ablauf

Die entwickelte Methode lässt sich in 8 Teilschritte gliedern (siehe Abbildung 2).

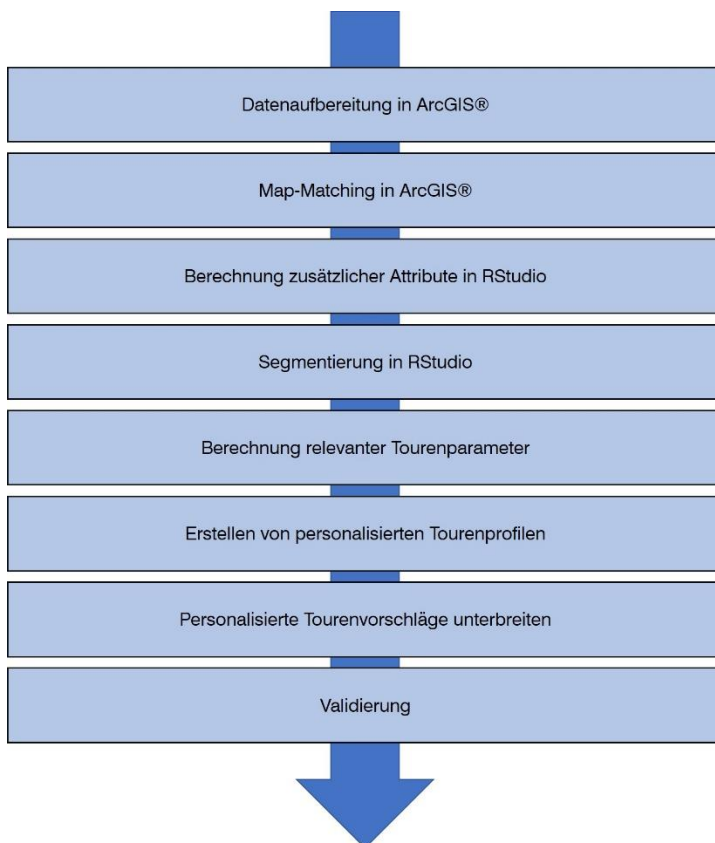


Abbildung 2: Ablauf der Teilschritte mit Angaben zu den verwendeten Programmen.

### 4.2 Datenaufbereitung

In einem ersten Schritt wurden die Rohdaten der verschiedenen Mountainbiker in das gewünschte Dateiformat überführt, um weitere Manipulationen an den Daten vorzunehmen. Mithilfe des Tools «GPX to Features» in ArcGIS wurden alle GPS-Tracks in Punkte-Shapefiles überführt.

Mittels Tool «Project» wurden die Koordinaten vom Format WGS 1984 in das Schweizer Format CH1903\_LV03 umgewandelt. Weiter wurden die Datensätze mit den X- und Y-Koordinaten angereichert.

Die Höhenangaben wurden aus dem Höhenmodell swissALTI3D extrahiert und ebenfalls allen Datensätzen angefügt. Bei Daten bei denen das Anfügen der swissALTI3D Werte nicht möglich war, weil sich der Mountainbiker beispielsweise im Ausland bewegte, wurden die Höhenangaben des GPX-Files verwendet.

Aus dem Vektordatensatz swissTLM\_Siedlungsnamen wurde mittels „Polygon to Raster“-Tool ein Raster mit den Attributen „1“ (Siedlung) und „0“ (keine Siedlung) erstellt. Das Siedlungsattribut wurde dann mit dem „Extract Multi Values to Points“-Tool extrahiert und den GPS-Tracks angefügt. Die Information Siedlung Ja/Nein wird später für die Ermittlung von Übergängen verwendet.

Die GPX-Daten bestehen aus Trackpoints (TRKPT) und Waypoints (WPT). Trackpoints stellen die kontinuierliche Aufzeichnung der räumlichen Position dar. Waypoints verfügen ebenfalls über eine räumlich-zeitliche Komponente. Allerdings werden ihnen vom Benutzer zusätzliche Informationen angefügt wie beispielsweise der Name des Restaurants an dieser Position. Für die weitere Verwendung der Daten sind lediglich die kontinuierlich aufgezeichneten Trackpoints relevant. Waypoints können in weiteren Datenaufbereitungsschritten zu Störungen führen indem sie teilweise die Punkteabfolge durcheinander bringen und wurden deshalb entfernt. Die Trackpoints wurden anschliessend zum Sicherstellen der korrekten Reihenfolge anhand des Aufzeichnungszeitpunkts neu geordnet.

Im letzten Schritt wurden die Datensätze manuell auf Ausreisser und Fehler überprüft und mit der Information über den Wegtyp angereichert. Dies erforderte ein Map-Matching, welches in einem separaten Unterkapitel beschrieben wird.

### **4.3 Map-Matching**

Das Map-Matching wurde lediglich verwendet, um den Wegtypinformation aus den Geodaten zu extrahieren. Der Wegtyp wird als einer der wichtigsten Parameter für die Kategorisierung von Mountainbike-Touren betrachtet, da er ein grundlegendes Merkmal für die technische Schwierigkeit einer Tour ist. Je schmaler und je weniger befestigt ein Weg ist, desto schwieriger wird es für einen Mountainbiker sich auf dem Weg fortzubewegen. Des Weiteren wird der Wegtyp bei der Bestimmung von Übergängen verwendet (siehe Kapitel 4.4.5). Die Wegtyp-Information stammt aus dem Vektordatensatz „swissTLM3D\_Strassen“ von Swisstopo.

Das Map-Matching wurde nach der Methode von Dalumpines & Scott (2011) durchgeführt (siehe Kapitel 2.2). Als Datengrundlage diente dabei „swissTLM3D\_Strassen“ von Swisstopo und „Highway“ von Openstreetmap.

#### **4.3.1 Datenaufbereitung für Map-Matching**

Um einerseits das „Route Solver“-Tool zu beschleunigen und um andererseits bessere Ergebnisse zu erzielen, wurde nur jeder zehnte GPS-Punkt als Wegpunkt im „Route Solver“-Tool verwendet. Zu diesem Zweck wurden von den aufbereiteten Datensätzen Kopien erstellt. Bei den kopierten Datensätzen wurde schliesslich jeder zehnte GPS-Punkt beibehalten und die übrigen Punkte gelöscht. Dies wurde in ArcGIS durchgeführt (siehe Modelle im Anhang).

Abbildung 3 und Abbildung 4 zeigen die kürzesten ermittelten Routen (blau) mit den jeweiligen Wegpunkten (rot) auf dem Wegenetz (grau). Bei Abbildung 4 ist zu erkennen, dass zusätzliche Strassen ausgewählt wurden, die mit grosser Wahrscheinlichkeit nicht vom Mountainbiker befahren wurden. Dies könnte im weiteren Verlauf des Prozesses zu einem falschen Map-Matching führen.



Abbildung 3: Kürzeste Route mit nur jedem zehnten GPS-Punkt.



Abbildung 4: Kürzeste Route mit allen GPS-Punkten.

Dalumpines & Scott (2011) nutzten Barrieren, um die Auswahl an möglichen Routen zu verringern. Der Abstand der Barriere zu den Punkten entsprach dem möglichen Positionierungsfehler des GPS (ca. 50m). Bei mehreren Versuchen wurde festgestellt, dass das Verwenden von Barrieren zu Problemen beim „Route-Solver“-Tool führen kann. Wenn sich der Mountainbiker auf einem nicht kartierten Weg aufhielt, konnte es sein, dass auf Grund des Fehlens einer Verbindung keine Route ermittelt werden konnte. Es besteht die Möglichkeit, den Abstand der Barrieren zu den Punkten grösser zu wählen, sodass immer eine Verbindung zwischen den einzelnen Punkten möglich wäre. Da die Anzahl Wegpunkte bei 10% der Gesamtzahl GPS-Punkte immer noch sehr gross war und das „Route Solver“-Tool deshalb so oder so nur eine kleine Auswahl an möglichen Strassen und Wegen zu Verfügung hat, wurde auf das Erstellen von Barrieren verzichtet.

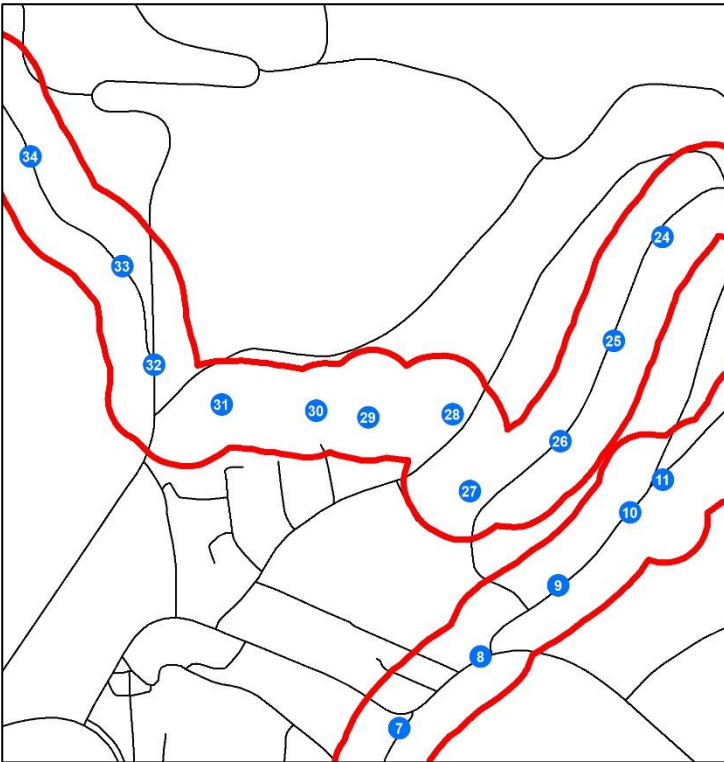


Abbildung 5: Durch Barriere (rot) verursachtes Fehlen von Verbindungen zwischen den Wegpunkten (blau) 26 und 32.

#### 4.3.2 Punkte auf kürzeste ermittelte Route einpassen

Um die GPS-Punkte auf das Strassen- und Wegenetz einzupassen, wurde das „Snap“-Tool verwendet. Als „Snap Environment“ wurde jeweils die mittels „Route Solver“ gelöste Route benutzt. Es wurden jeweils alle Punkte des GPS-Tracks auf die gelöste Route eingepasst. Als Suchradius für das nächstgelegene Liniensegment wurde eine Distanz von 20 Metern verwendet. Wenn sich kein Liniensegment innerhalb des Suchradius befand, wurde angenommen, dass sich der Mountainbiker tatsächlich nicht auf dem Wegenetz befindet. Mittels „Spatial Join“ wurde schliesslich das Wegtyp-Attribut aus dem Strassendatensatz extrahiert und jedem Punkt des GPS-Tracks angeheftet.

Die verwendete Map-Matching-Methode ist fähig anhand der Reduktion der Matching-Auswahl das richtige Segment zu finden. Allerdings ist die Methode nicht fähig, Punkte entlang des ausgewählten Segments zu verschieben. Dadurch entstehen ungewollt Punkteansammlungen falsche Abstände zwischen den Punkten (siehe Abbildung 6). Um weniger Positionsfehler zu generieren, welche schliesslich Auswirkungen auf später berechnete Bewegungsparameter hätten, wurden die Wegtyp-Attribute mittels „Join“-Tool einer Kopie des jeweiligen GPS-Tracks angefügt. Somit wurden die original Positionen die vom Mountainbiker aufgezeichnet wurden behalten. Das Map-Matching erfolgt also nur, um den Wegtyp zu extrahieren, nicht um die GPS-Positionsgenauigkeit zu erhöhen.

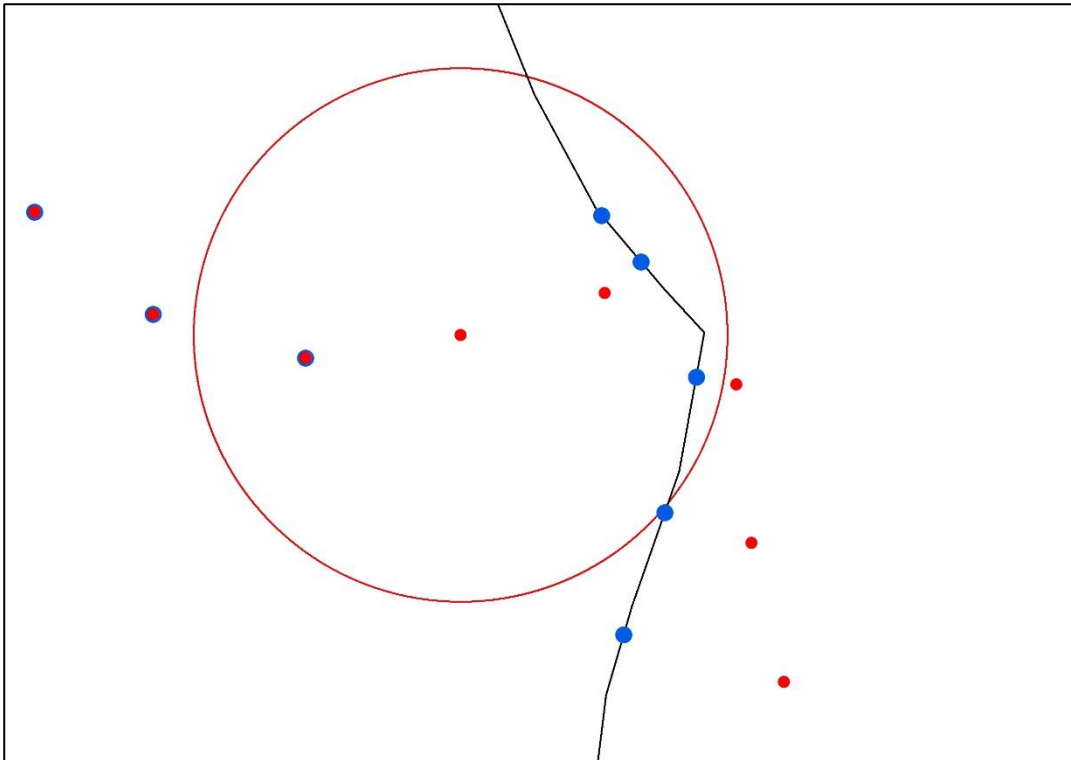


Abbildung 6: Fehler, die bei der verwendeten Map-Matching-Methode entstehen können. Hier im Bild werden 4 Punkte auf das Wegenetz eingepasst, obwohl sie sehr wahrscheinlich neben dem Weg sind, da hier ein nicht-kartierter Singletrail befahren wurde. Rote Punkte: Original GPS-Position. Blaue Punkte: Eingepasste Punkte. Roter Kreis: Snapping Radius (20m). Schwarze Linie: Teile des Wegenetz.

#### 4.4 Methoden zur Berechnung zusätzlicher Attribute in R

Um in weiteren Schritten die GPS-Tracks segmentieren zu können und um die tourenrelevanten Parameter zu berechnen, wurden in R zusätzliche Attribute berechnet.

Zusätzliche berechnete Attribute:

- Planare Distanz: Beschreibt die kürzeste Verbindung zwischen zwei Punkten im 2dimensionalen Raum
- Euklidische Distanz: Beschreibt die kürzeste Verbindung zwischen zwei Punkten im 3dimensionalen Raum
- Höhenunterschied zwischen zwei Punkten
- Höhenmeter bergauf: Summe der Höhenmeter bergauf
- Höhenmeter bergab: Summe der Höhenmeter bergab
- Mittlerer Höhenunterschied: Mittelwert des Höhenunterschieds über n-Punkte berechnet
- Geschwindigkeit
- Anstiege: Identifiziert ununterbrochene Anstiege
- Abfahrten: Identifiziert ununterbrochene Abfahrten
- Flach: Identifiziert ununterbrochene flache Abschnitte
- Gefälle

- Mittleres Gefälle: Mittelwert des Gefälles über n-Punkte berechnet
- Gefälleklassen: Einteilung des gemittelten Gefälles in verschiedene Klassen
- Verkehrsmittel: Ermittelt, ob beispielsweise eine Bergbahn benutzt wurde
- Übergänge: Identifiziert Abschnitte, welche nicht zur Mountainbike-Tour dazugezählt werden

Die Berechnung ausgewählter Attribute wird in den folgenden Unterkapiteln näher beschrieben. Die Parameter planare Distanz, Euklidische Distanz, Geschwindigkeit und Gefälle wurden von Tomlinson (2015) bereits beschrieben. Für die Bewerkstellung eines „Moving Window“ wurde ebenfalls die Methode von Tomlinson (2015) verwendet.

#### 4.4.1 Mittlerer Höhenunterschied

Der mittlere Höhenunterschied besteht aus dem Mittelwert des Höhenunterschieds über 13 Punkte (Anzahl n-Punkte) berechnet und wurde mittels „Moving Window“ ermittelt. Dabei wird bei jedem GPS-Punkt der Mittelwert aus dem Höhenunterschied der 6 vorangegangenen, der 6 nachfolgenden und des eigenen Wertes berechnet. Der mittlere Höhenunterschied wurde für die Einteilung von Abschnitten in Anstiege, Abfahrten und flache Passagen benötigt. Er verhinderte, dass zu kleine Abschnitte gemacht wurden.

Die Anzahl n-Punkte wurde im Experiment „Segmentierung: Anstiege, Abfahrten und flache Abschnitte“ (Kapitel 5.1) ermittelt.

#### 4.4.2 Markierung von Anstiegen, Abfahrten und flachen Abschnitten

Die Anstiege-Abfahrten-Funktion ordnet jedem GPS-Punkt den Wert „1“ (Anstieg), „2“ (Abfahrt) oder „0“ (flach) zu. Das Anstiegs-Abfahrts-Attribut wird später für die Segmentierung verwendet.

GPS-Punkte erhalten den Wert „1“ (Anstieg), wenn sie sich in einem Abschnitt befinden, bei dem eine bestimmte Anzahl Höhenmeter bergauf überschritten und keine Höhenmeter bergab zurückgelegt wurden. Da Mountainbiker es auch als zusammenhängenden Anstieg empfinden, wenn es zwischendurch kurze flache Passagen oder kurze Abschnitte bergab gibt, wurde an jedem Punkt der mittlere Höhenunterschied über n-Punkte als Bergauf/Bergab-Indikator verwendet.

Beispiel 1: Der mittlere Höhenunterschied bei Punkt i ist grösser gleich 0 und der tatsächliche Höhenunterschied bei Punkt i ist ebenfalls grösser gleich 0. Der mittlere Höhenunterschied des vorherigen Punktes ist allerdings kleiner 0. Die überwundene Anzahl Höhenmeter bergauf ist gleich dem Höhenunterschied an Punkt i.

Beispiel 2: Der mittlere Höhenunterschied und der tatsächliche Höhenunterschied bei Punkt i sind grösser gleich 0. Der mittlere Höhenunterschied des vorherigen Punktes ist ebenfalls grösser

gleich 0. Die überwundene Anzahl Höhenmeter bergauf ist die Summe der überwundenen Anzahl Höhenmeter des vorherigen Punktes plus der tatsächliche Höhenunterschied bei Punkt i.

Beispiel 3: Der mittlere Höhenunterschied von Punkt i ist grösser gleich 0. Der tatsächliche Höhenunterschied ist kleiner 0. Die überwundene Anzahl Höhenmeter bergauf wird, der des vorherigen Wertes gleichgesetzt.

Beispiel 4: Der mittlere Höhenunterschied von Punkt i ist kleiner 0. Die überwundene Anzahl Höhenmeter bergauf wird auf 0 gesetzt. Wird die überwundene Anzahl Höhenmeter bergauf auf 0 gesetzt, wird der vorherige Wert mit dem Schwellenwert verglichen. Ist er grösser oder gleich dem Schwellenwert, werden alle vorangegangenen Punkte mit einem Wert grösser 0 als „1“ (Anstieg) markiert.

Abfahrten werden nach dem gleichen Prinzip markiert. Nur, dass da das Kleingleich-Zeichen benutzt wird.

Abschnitte werden als „0“ (flach) markiert, wenn die Bedingungen für Anstieg oder Abfahrt nicht erfüllt sind.

#### **4.4.3 Gefälle**

Das Gefälle wurde zum einen für jeden GPS-Punkt einzeln in Prozent berechnet, und zum anderen wurde das mittlere Gefälle (%) über drei GPS-Punkte berechnet. Letzteres glättet die Werte und mindert extrem Werte, welche durch die GPS-Ungenauigkeit hervorgerufen werden. Das mittlere Gefälle wurde verwendet, um Gefälleklassen zu bilden. Für Abfahrten wurden die Gefälleklasseneinteilung nach der Singletrail-Skala von Schymik et al. (2008) gewählt. Für Anstiege wurde eine eigene Klasse gebildet. Für Abfahrten wurde das mittlere Gefälle in  $\leq 40\%$ ,  $40\%$  bis  $70\%$  und in  $>70\%$  eingeteilt. Für Anstiege wurden die Klassen  $\leq 15\%$  und grösser  $>15\%$ . Die Gefälleklassen für Anstiege wurden nach eigenen Erfahrungen festgelegt, nach denen eine Steigung  $>15\%$  als unangenehm empfunden wird.

#### **4.4.4 Verkehrsmittel**

Das Attribut „Verkehrsmittel“ soll die Verwendung von Bergbahnen anzeigen. Sich bewegende Objekten können anhand von Geschwindigkeit, Beschleunigung, Sinuosität, etc. bestimmten Bewegungscharakteristiken zugeordnet werden (Dodge, Weibel, & Forootan, 2009). Im vorliegenden Fall wurde die im Vergleich zu einem Mountainbiker hohe Geschwindigkeit bei einem positiven Gefälle verwendet, um Bergbahnen zu erkennen. Zu diesem Zweck wurde in einem ersten Schritt eine Tranchierung über zwei Schwellenwerte vorgenommen. Wenn beide Schwellenwerte überschritten werden, wurde der jeweilige Punkt mit „1“ markiert. Wurden die Schwellenwerte unterschritten, wurde der Punkt mit „0“ gekennzeichnet. Als Schwellenwerte



wurden eine Geschwindigkeit von  $\geq 4$  m/s und ein mittleres positives Gefälle von  $\geq 15\%$  festgelegt. Im zweiten Schritt wurde an jedem Punkt mittels „Moving Window“ der Mittelwert über 21 Punkte berechnet. Beträgt der Mittelwert mehr als 0.1, so wurde der jeweilige Punkt definitiv als „Verkehrsmittel“ markiert.

#### 4.4.5 Übergänge

Als Übergänge werden in diesem Dokument Teile eines GPS-Tracks bezeichnet, welche mit grosser Wahrscheinlichkeit nicht zur eigentlichen Mountainbike-Tour gezählt werden können, da es sich mit grosser Sicherheit um den Anfahrts- oder Heimweg handelt oder um einen anderen Abschnitt der nicht als Mountainbike-Tour gewertet werden kann. Punkte, die sich in Siedlungen befinden und nicht dem Wegtyp „Singletrail“ entsprechen, wurden zu Übergängen gezählt. Punkte, die als „Verkehrsmittel“ markiert sind werden ebenfalls als „Übergang“ markiert.

### 4.5 Segmentierung

Die Segmentierung der GPS-Tracks erfolgte anhand von Umwelt- und Bewegungsparametern. Aus den Segmenten werden bei der Erstellung der Tourenprofile Parameter berechnet. Beispielsweise sind Distanzen von gewissen Segmenten wichtig. Auf der anderen Seite werden bestimmte Segmente von den Berechnungen ausgeschlossen (z.B. Übergänge werden von der Berechnung der Länge der Mountainbike-Tour ausgeschlossen). Für die Segmentierung konnten die bereits berechneten zusätzlichen Attribute verwendet werden, welche jedem GPS-Punkt annotiert sind. Die meisten Attribute haben eine Codierung. Beispielsweise „1“ für „JA“ oder „0“ für „NEIN“. Alle GPS-Tracks wurden anhand des Attributs „UphillDownhill“ („Anstieg“ Code 1, „Abfahrt“ Code 2 und „flach“ Code 0) segmentiert. Weiter wurde anhand des Attributs „Übergang“ („ja“ Code 1, „nein“ Code 0) segmentiert (Dies allerdings nicht bei den Tourenvorschlägen. Bei Tourenvorschlägen wird angenommen, dass sie zwar Anfahrtswege enthalten, ansonsten aber auf das Wesentliche reduziert sind. Eine Segmentation anhand von Übergängen wurde also nur bei GPS-Tracks von Mountainbikern gemacht.). Eine weitere Segmentation wurde anhand des Attributs „Verkehrsmittel“ („ja“ Code 1, „nein“ Code 0) segmentiert. Schliesslich wurde noch mit dem Attribut „OBJEKTART“ („Singletrail ja“ Code 16 und 0, „Singletrail nein“ alle Codes ausser 16 und 0) und dem Attribut „Gefälleklasse“ („ $<15\%$ “ Code 0, „ $>15\%$  und  $<40\%$ “ Code 1, „ $>40\%$  und  $<70\%$ “ Code 2, „ $>70\%$ “ Code 3) segmentiert.

### 4.6 Erstellen personalisierter Tourenprofile

Die personalisierten Tourenprofile stellen sich aus den Tourenparametern zusammen, die aus den aufgezeichneten GPS-Tracks berechnet wurden (Tabelle 4 zeigt die Tourenparameter für den GPS-Track Nr. 5 von Mountainbiker G, Abbildung 7 zeigt den GPS-Track Nr. 5 von G auf einer Karte). Für jede aufgezeichnete Tour wurden die folgenden Tourenparameter berechnet:

1. Länge Mountainbike-Tour: Dieser Tourenparameter zeigt an, in welchem Bereich sich die Länge einer Mountainbike-Tour bewegen soll. Der Parameter wird aus der Summe aller euklidischen Distanzen abzüglich der Übergänge berechnet und in Kilometer angegeben.
2. Gesamtdistanz Singletrails: Summe aller Abschnitte in Kilometer, die als Singletrails markiert sind.
3. Anteil der Gesamtdistanz Singletrails an der gesamten Mountainbike-Tour. Als Mountainbike-Tour gilt die Gesamtdistanz des GPS-Tracks minus die Gesamtdistanz aller Übergänge.
4. Cross-Country-Index: Der Cross-Country-Index ist ein Mass für die Häufigkeit von Anstiegen und Abfahrten. War die Mountainbike-Tour komplett flach, so ist der Wert „0“. Je mehr Anstiege und/oder Abfahrten desto grösser der Wert. Die Berechnungsformel lautet  $(\text{Anzahl Anstiege} + \text{Anzahl Abfahrten}) / \text{Gesamtdistanz GPS-Track}$ .
5. Höhenmeter bergauf: Summe aller positiven Höhenmeter, die als „Verkehrsmittel“ markiert sind. Dieser Parameter soll den Bereich für den maximalen Höhenunterschied bergauf darstellen und wird in Meter angegeben.
6. Höhenmeter bergab: Summe aller negativen Höhenmeter. Dieser Parameter wird als Minimum für den Höhenunterschied bergab interpretiert. Das Vorhandensein von Verkehrsmitteln wurde ausser Acht gelassen, da angenommen wird, dass niemand eine Bahn bergab verwendet.
7. Uphill-Downhill-Verhältnis: Dieser Parameter ist das Verhältnis zwischen Höhenmeter bergauf und der absoluten Anzahl Höhenmeter bergab. Der Parameter soll aufzeigen, ob ein Mountainbiker lieber hoch oder abwärts fährt. Der Minimalwert beträgt „0“. Nach oben ist die Skala offen.
8. Gesamtdistanz Anstiege (ohne Übergänge): Gesamtdistanz, die auf Anstiegen zurückgelegt wurde. Punkte, die als Übergänge markiert sind, werden nicht eingerechnet.
9. Gesamtdistanz Singletrail auf Anstiegen
10. Singletrail-Anteil auf Anstiegen
11. Gesamtdistanz von Anstiegen mit einem Gefälle >15% ohne Übergänge: Hier werden auch Punkte miteingerechnet, welche nicht als Singletrail markiert sind solange sie nicht als Übergänge gelten.
12. Anteil Anstiege >15%
13. Gesamtdistanz Abfahrten (ohne Übergänge)
14. Gesamtdistanz Singletrail auf Abfahrten
15. Singletrail-Anteil auf Abfahrten
16. Gesamtdistanz Singletrails auf Abfahrten mit einem Gefälle <40%
17. Anteil Singletrails auf Abfahrten mit einem Gefälle <40%
18. Gesamtdistanz Singletrails auf Abfahrten mit einem Gefälle zwischen 40% und 70%
19. Anteil Singletrails auf Abfahrten mit einem Gefälle zwischen 40% und 70%

20. Gesamtdistanz Singletrails auf Abfahrten mit einem Gefälle >70%
21. Anteil Singletrails auf Abfahrten mit einem Gefälle >70%
22. Gesamtdistanz von flachen Abschnitten
23. Gesamtdistanz von Singletrails auf flachen Abschnitten
24. Anteil von Singletrails auf flachen Abschnitten

Je nach Anzahl und Verschiedenheit der Touren, die für ein Profil verwendet werden, ergibt sich ein mehr oder weniger scharfes Tourenprofil.

*Tabelle 4: Beispiel der berechneten Parameter für den GPS-Track Nr. 5 von Mountainbiker G. Links: Parameter. Mitte: Parameterwert. Rechts: Masseinheit.*

<b>GPS_Track</b>	<b>MTB_G_5</b>	<b>Einheit</b>
gesamt_Dist_GPS_Track	4.866481775	km
CrossCountryIndex	0.410974518	Anzahl Anstieg, Abfahrten / km
Laenge_MTB_Tour	4.470358019	km
Laenge_Uebergange	0.396123757	km
Anteil_Uebergange	8.139838487	%
Dist_Singletrail	1.429443571	km
Anteil_Singletrail	31.97604229	%
gesamt_Hoehe_Uphill	424.4884033	m
gesamt_Hoehe_Downhill	-435.701416	m
Uphill_Downhill_Verhaeltnis	0.974264457	
gesamter_Anstieg	2.421919725	km
Distanz_Singletrail_Anstiege	0.011863359	km
Anteil_Singletrail_Anstieg	0.489832891	%
Anstiege_gr15	698.1375554	m
Anteil_Anstiege_gr15	28.82579254	%
Abfahrt_gesamt	2.048438294	km
Distanz_Singletrail_Abfahrten	1.417580211	km
Anteil_Singletrail_Abfahrten	69.20297358	%
Abfahrten_Singletrail_kl40	1373.328767	m
Anteil_Abfahrten_ST_kl40	67.04272082	%
Abfahrten_ST_gr40_kl70	44.25144467	m
Anteil_Abfahrten_ST_gr40_kl70	2.160252754	%
Abfahrten_ST_gr70	0	m
Anteil_Abfahrten_ST_gr70	0	%
Flach_gesamt	0	km
Distanz_Singletrail_Flach	0	km
Anteil_Singletrail_Flach	0	%

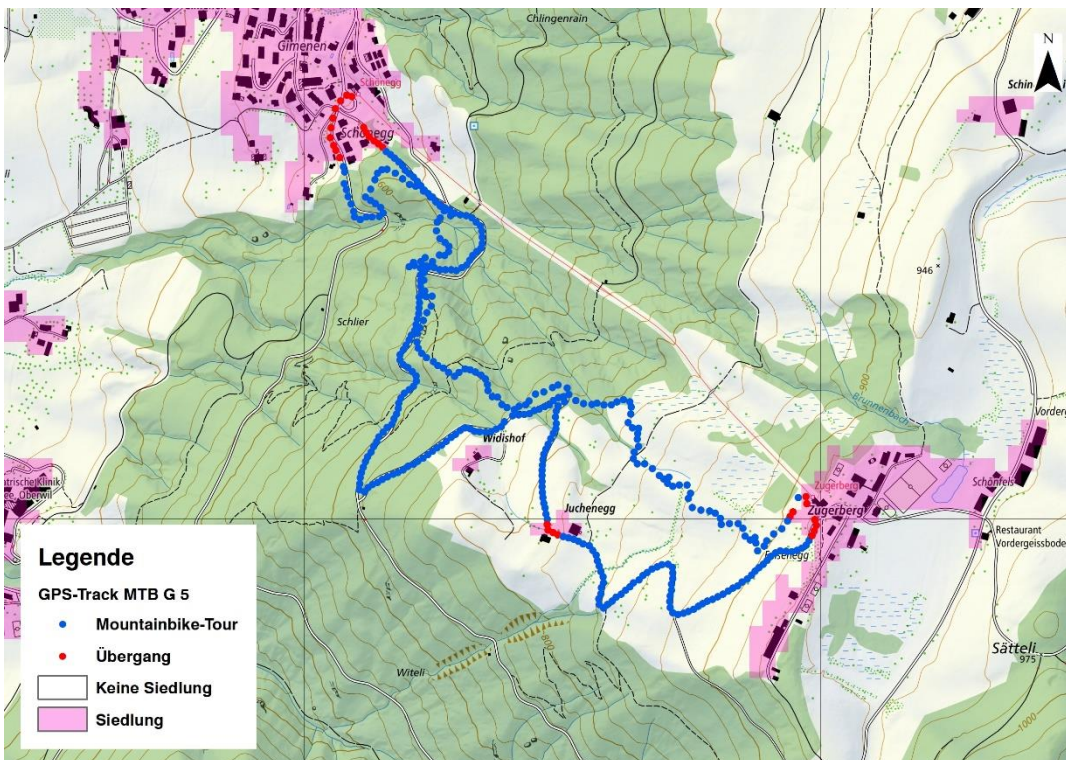


Abbildung 7: GPS-Track Nr. 5 Mountainbiker G mit Segmentierung nach Übergängen (rot).

#### 4.7 Personalisierte Tourenvorschläge unterbreiten

Das Unterbreiten von personalisierten Tourenvorschlägen erfolgt indem die Tourenparameter der vorzuschlagenden Touren mit dem Tourenprofil des jeweiligen Mountainbikers verglichen werden. Dazu wird für jeden der Tourenparameter ein statistischer Vergleichstest durchgeführt. Die Parameterwerte der vorzuschlagenden Tour werden als Referenzwerte und die Parameterwerte des personalisierten Tourenprofils werden als Vergleichswerte verwendet. Alle Tourenparameter werden einzeln miteinander verglichen. Für die vorzuschlagende Tour bedeutet dies pro Parameter ein Wert. Das personalisierte Tourenprofil enthält in der Regel 10 Werte pro Tourenparameter. Unterscheiden sich die Vergleichswerte signifikant (Wilcoxon-Test, Signifikanzniveau  $\alpha = 0.05$ ) von einem der Referenzwerte, so erhält die vorzuschlagende Tour für diesen Parameter null Punkte. Gibt es keinen Unterschied zwischen den Vergleichswerten und dem Referenzwert, so erhält die vorzuschlagende Tour für diesen Parameter die Punktzahl „1“ multipliziert mit einem Gewichtungsfaktor (Gewichtungsfaktor siehe Algorithmus Anhang. Anhand der Gesamtpunktzahlen wird eine Rangliste der vorzuschlagenden Touren erstellt. Als am besten geeignete Tour, der zur auswahlstehenden Touren wird, die mit der höchsten Punktzahl betrachtet.

Die Tourenparameter der vorzuschlagenden Mountainbike-Touren werden nach derselben Methode wie die personalisierten Tourenprofile berechnet. Ausser, dass keine Übergänge berechnet werden. Es wird angenommen, dass die vorzuschlagenden Mountainbike-Touren bereits in „reiner Form“ vorliegen.

## **4.8 Validierung der Tourenparameter, personalisierten Tourenprofile und Tourenvorschläge**

Bis anhin ist nicht bekannt, ob die ausgewählten Tourenparameter eine Unterscheidung der verschiedenen Mountainbiker zulassen. Aus diesem Grund werden Tabellen erstellt, die einen Vergleich der Gesamtpunktzahlen und der Punktzahlen je Tourenparameter zulassen (die Punkte sind aus dem Vergleich zwischen Tourenprofil und Tourenvorschlag).

Um die personalisierten Tourenprofile der einzelnen Mountainbiker zu vergleichen, werden für alle Tourenparameter jedes Mountainbikers Boxplots erstellt. So können Streuung und Lage des Medianwertes eines jeden Tourenparameters von jedem Mountainbiker mit denen der Anderen verglichen werden. Zusätzlich werden die Boxplots mit den Tourenparametern der vorzuschlagenden Mountainbike-Touren angereichert. Dies ermöglicht das Aufzeigen von allfälligen Unterschieden oder Gemeinsamkeiten.

## 5 Experimente

### 5.1 Segmentierung: Anstiege, Abfahrten und flache Abschnitte

Die Segmentierung in Anstiege, Abfahrten und flache Abschnitte ist stark vom mittleren Höhenunterschied und vom gewählten Schwellenwert abhängig. In erster Linie ist der Schwellenwert der bestimmende Faktor für die Abschnitte, denn erst durch die Überschreitung des Schwellenwertes entstehen unterschiedliche Segmente. Beim mittleren Höhenunterschied ist die gewählte Anzahl n-Punkte ausschlaggebend. Wird „n“ zu gross gewählt, wird das Höhenprofil zu stark geglättet. Wird „n“ zu klein gewählt, wird die Aufsummierung der Höhenunterschiede zu oft auf null gesetzt und der Schwellenwert wird bei kleinen Anstiegen/Abfahrten nicht erreicht.

Was als zusammenhängende Abfahrt, Anstieg oder flachen Abschnitt gezählt werden kann, ist subjektiv. Um jedoch die Schwellenwerte und die Anzahl n-Punkte des mittleren Höhenunterschieds möglichst objektiv zu wählen, wurden Anstiege, Abfahrten und flache Abschnitte einer Mountainbike-Tour manuell markiert und mit den Markierungen unterschiedlicher Schwellenwerte und n-Punkte Szenarios verglichen. Mittels Kruskal-Wallis-Test wurde geprüft, ob signifikante Unterschiede zwischen den einzelnen Gruppen existieren. Gruppiert wurde nach Grösse der Anzahl n-Punkte, Grösse des Schwellenwertes und nach Anzahl n-Punkte kombiniert mit jedem Schwellenwert. Der Kruskal-Wallis-Test wurde auf einem Signifikanzniveau von 0.05 durchgeführt (H0: Die Werte unterscheiden sich nicht). Ebenfalls die Prüfung auf Normalverteilung (Shapiro-Wilk-Test: H0: Die Daten sind Normalverteilt).

Für die manuelle Markierung wurden fünf Touren von fünf verschiedenen Mountainbikern ausgewählt, die dem Autor bestens bekannt ist. Eine sechste Tour wurde als Höhenprofil dargestellt und Anstiege, Abfahrten und flache Abschnitte wurden anhand der automatischen Segmentierung eingefärbt. Hier wurden die eingefärbten Abschnitte visuell geprüft und anhand der persönlichen Einschätzung des Autors aussortiert.

### 5.2 Vergleich von Gesamtdistanz, Strecke auf Singletrails und Höhenmeter bergauf mit Webdienst

Die Smartphone-App Komoot (Komoot GmbH, 2017) bietet die Möglichkeit, Mountainbike-Touren zu planen. Dazu kann beispielsweise ein aufgezeichneter Datensatz hochgeladen werden. Der hochgeladene GPS-Track wird dann automatisch auf das Wegenetz von Openstreetmap® eingepasst. Danach wird die Länge der Mountainbike-Tour, Höhendifferenz, Anteil Singletrail, etc. angezeigt.

Diese Funktion von Komoot wurde verwendet, um die Plausibilität der Tourenparameter Gesamtdistanz GPS-Track, Gesamtdistanz Singletrail und Höhenmeter bergauf zu prüfen. Die Plausibilitätsprüfung wurde bei 10 Mountainbike-Touren desselben Mountainbikers durchgeführt.

## 6 Ergebnisse

### 6.1 Segmentierung: Anstiege, Abfahrten und flache Abschnitte

Der Vergleich der in R implementierten Segmentierung der Anstiege, Abfahrten und flachen Abschnitten mit den manuell vorgenommenen Segmentierungen hat gezeigt, dass sich die automatische Segmentierung im Mittel zu 74% mit der manuellen Segmentierung deckt (siehe Anhang C).

In der Abbildung 8 sind die Punktedifferenzen der einzelnen getesteten Kombination einander gegenübergestellt. Getestet wurde eine Anzahl n-Punkte von 13, 25 und 50. Als Schwellenwerte wurden die Werte (in Meter) 10, 25 und 50. Die Differenzen wurden für jede mögliche Kombination dieser Werte geprüft. Der durchgeführte Kruskal-Wallis-Test ergab keine signifikanten Unterschiede bei einem p-Wert von 0.998.

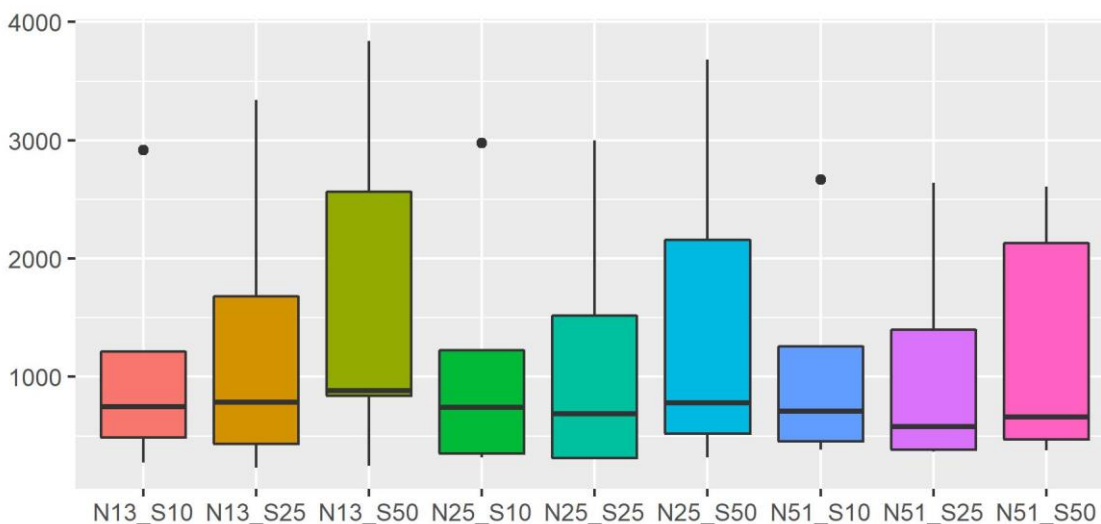


Abbildung 8: Vergleich der Punktedifferenz von manueller Segmentierung zu automatischer Segmentierung aufgeteilt auf jede getestete Kombination von n-Punkte und Schwellenwert. X-Achse: Kombinationsname; N+Zahl entspricht der Anzahl n-Punkte für die Berechnung des mittleren Höhenunterschieds; S+Zahl entspricht der Höhe des Schwellenwertes in Meter. Y-Achse: Punktedifferenz.

Der Vergleich der Punktedifferenzen gruppiert nach n-Punkte und Schwellenwert ergab ebenfalls keine signifikanten Unterschiede (siehe Abbildung 9 und Abbildung 10). Der Kruskal-Wallis-Test ergab für den Vergleich der Punktedifferenzen nach n-Punkte der mittleren Höhendifferenz sortiert einen p-Wert von 0.93 und für die Differenzen sortiert nach Grösse des Schwellenwertes einen p-Wert von 0.74. Bei allen Tests wurde ein Signifikanzniveau von 0.05 verwendet. Die Daten sind nicht normalverteilt.



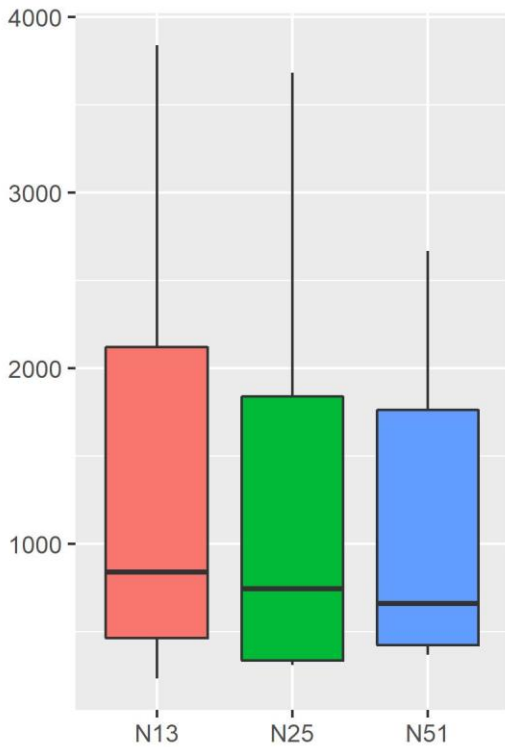


Abbildung 9: Vergleich der Punktedifferenzen nach n-Punkte gruppiert. X-Achse: Gruppen (N13 = mittlere Höhendifferenz über 13 Punkte berechnet, N25 = 25 Punkte, etc.). Y-Achse: Punktedifferenz

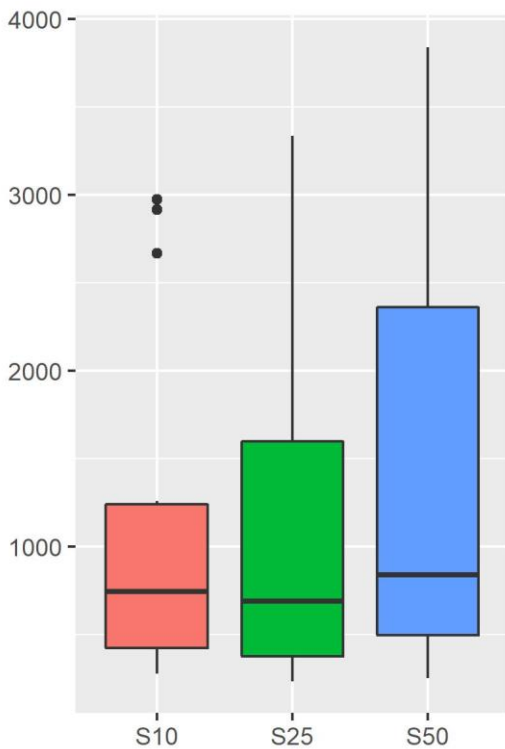


Abbildung 10: Vergleich der Punktedifferenzen nach Schwellenwerte gruppiert. X-Achse: Schwellenwertgruppe (S10 = Schwellenwert 10 Meter, etc.). Y-Achse: Punktedifferenz.

Die visuelle Auswertung des Höhenprofils, welches anhand der verschiedenen Kombinationsmöglichkeiten von Anzahl n-Punkte und Grösse des Schwellenwertes eingefärbt wurde, ergab eine ideale Anzahl n-Punkte von 13 und einen idealen Schwellenwert von 20 Metern

(siehe Abbildung 11). Teilweise waren Abschnitte zu ausgedehnt markiert (siehe Abbildung 12) oder es wurden zu kleinräumige Abschnitte eingefärbt.

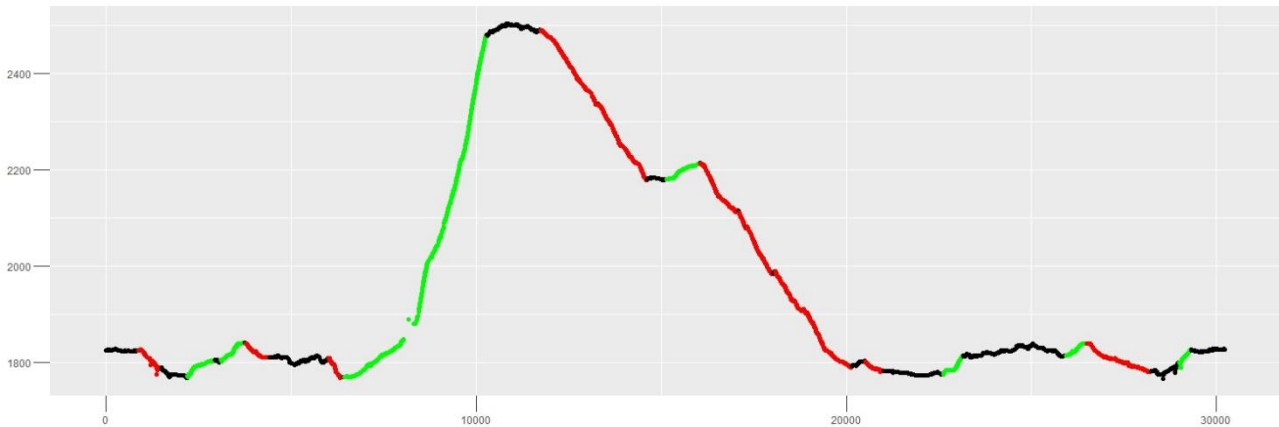


Abbildung 11: Höhenprofil mit markierten Anstiegen (grün), Abfahrten (rot) und flachen Abschnitten (schwarz). Die Segmentierung der Abschnitte erfolgte mit einer Anzahl  $n$ -Punkte von 13 und einem Schwellenwert von 25 Metern. X-Achse: Horizontale Distanz in Metern. Y-Achse: Meter über Meer.

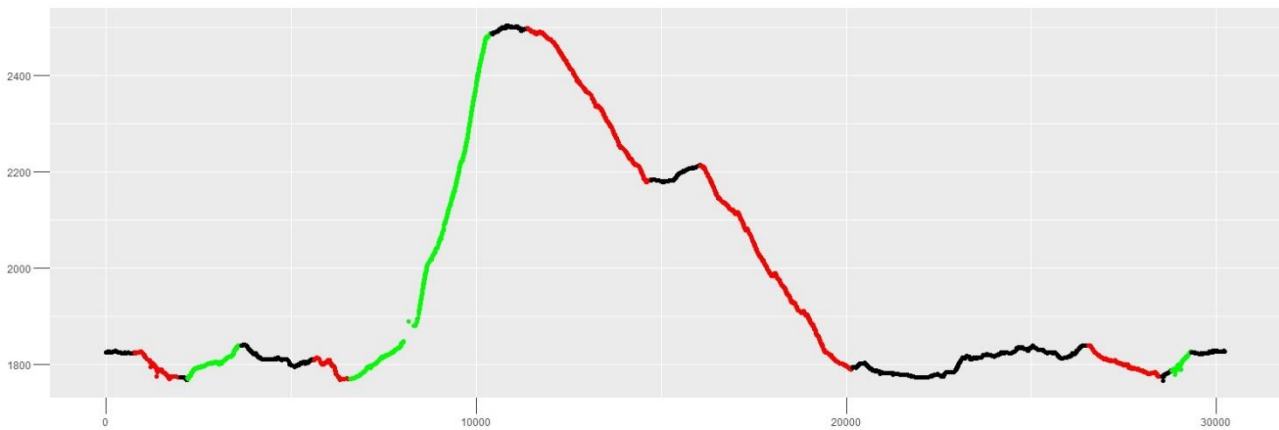


Abbildung 12: Höhenprofil mit markierten Anstiegen (grün), Abfahrten (rot) und flachen Abschnitten (schwarz). Die Segmentierung der Abschnitte erfolgte mit einer Anzahl  $n$ -Punkte von 25 und einem Schwellenwert von 50 Metern. X-Achse: Horizontale Distanz in Metern. Y-Achse: Meter über Meer.

## 6.2 Segmentierung der Verkehrsmittel

Die Abbildung 13 zeigt ein perfektes Beispiel einer Segmentierung nach Verkehrsmitteln. Es ist zu sehen, dass das Umsteigen von der Standseilbahn (roter Bogen) auf die Luftseilbahn (gerader roter Strich) nicht markiert wurde.

Auf Abbildung 14 ist zu erkennen, dass Bahnen nur teilweise oder gar nicht segmentiert wurden.

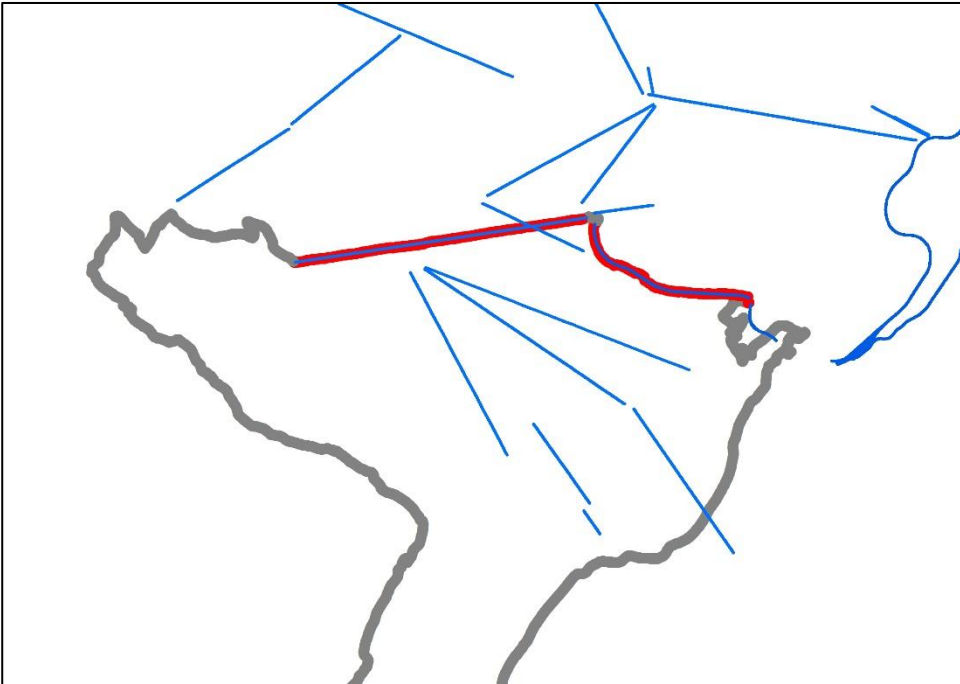


Abbildung 13: Segmentierung der Verkehrsmittel (Raum St. Moritz). In Blau: Standseil-, Luftseil- und Eisenbahnen. Rot: Als Verkehrsmittel segmentierte Abschnitte. Grau: Übrige GPS-Punkte

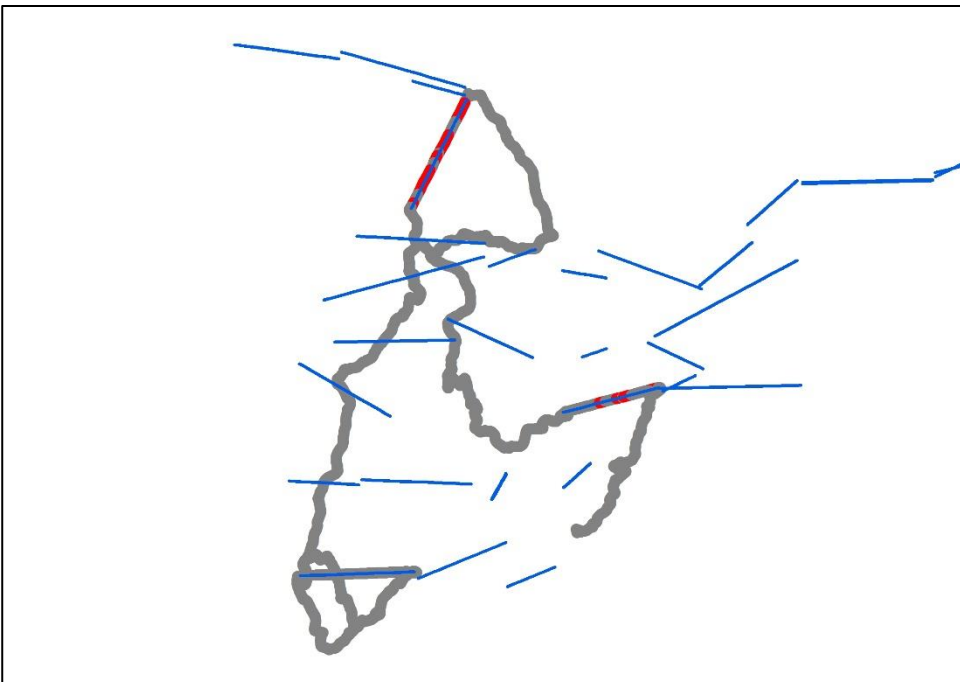


Abbildung 14: Segmentierung der Verkehrsmittel (Raum Lenzerheide). In Blau: Standseil-, Luftseil- und Eisenbahnen. Rot: Als Verkehrsmittel segmentierte Abschnitte. Grau: Übrige GPS-Punkte

### 6.3 Vergleich von Gesamtdistanz, Strecke auf Singletrails und Höhenmeter bergauf mit Webdienst

Abbildung 15 zeigt die berechneten Gesamtdistanzen (in km) von 10 verschiedenen GPS-Tracks verglichen mit den Werten (in km) des Webdienstes Komoot. Die Werte sind jeweils als Punkte dargestellt. Die Verbindungslinien sollen Unterschiede besser hervorheben. Es ist ersichtlich, dass nur minime Unterschiede zwischen den selbst berechneten Distanzen und denen von Komoot bestehen.

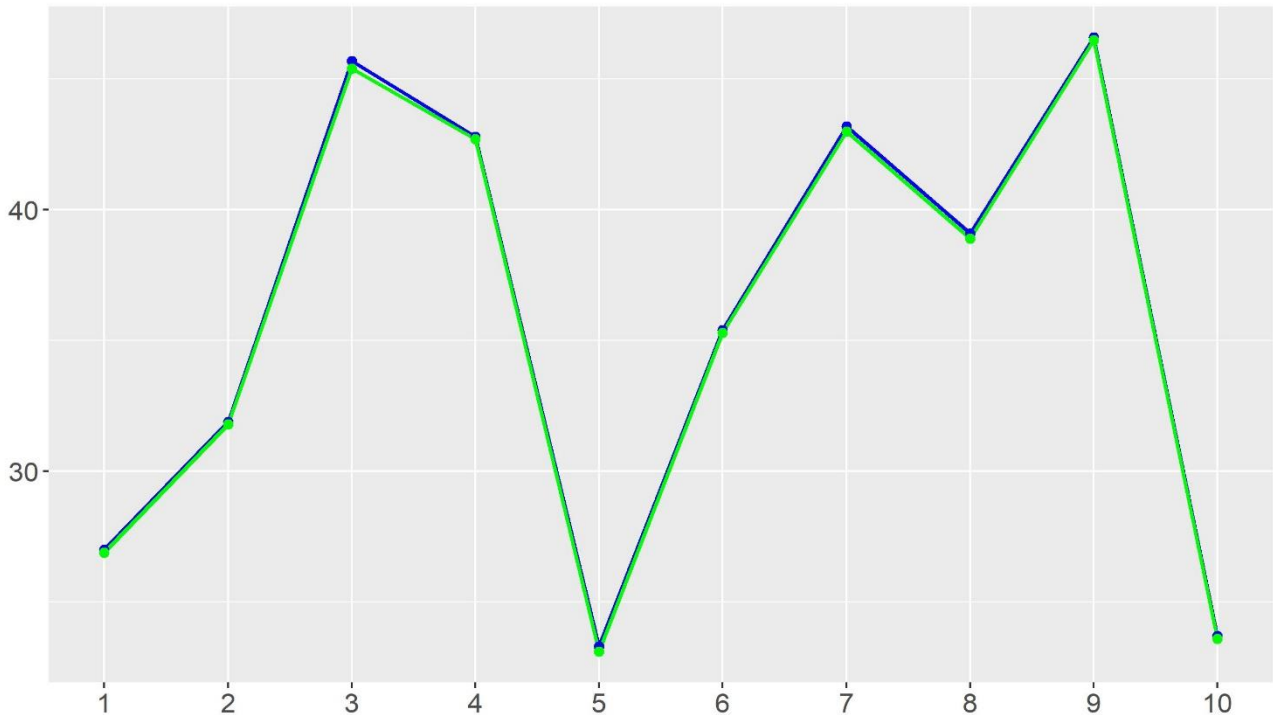


Abbildung 15: Vergleich der berechneten Gesamtdistanzen (blaue Punkte/Linie) mit den Gesamtdistanzen des Webdienstes Komoot (grüne Punkte/Linie) von 10 verschiedenen GPS-Tracks. X-Achse: Nummer des GPS-Tracks. Y-Achse: Distanz in km.

Abbildung 16 zeigt die Unterschiede zwischen Gesamtdistanzen auf Singletrails, Gesamtdistanzen auf Singletrails und Feldwegen und die Gesamtdistanzen des Webdienstes Komoot von 10 verschiedenen GPS-Tracks. Es ist ersichtlich, dass sich die selbst berechneten Distanzen kaum von denen des Webdienstes unterscheiden, wenn zu den Singletrails auch Feldwege hinzugezählt werden. Werden die Gesamtdistanzen auf Singletrails nur mit reinen Singletrails berechnet, entstehen massive Unterschiede von mehreren Kilometern.

Die Abbildung 17 veranschaulicht die Unterschiede zwischen den Höhenmetern bergauf. Es wurden die berechneten Höhenmeter bergauf mit Höhenangaben von swissALTI3D (Swisstopo), mit Höhenangabe aus den GPX-Dateien, mit Höhenangaben von Komoot und mit den Höhenangaben von swissALTI3D wenn nur jeder 10 GPS-Punkt verwendet wird miteinander verglichen. Es zeigt sich, dass je genauer die verwendeten Daten sind, desto grösser wird die Differenz zwischen Werten des Webdienstes Komoot und den Berechneten.

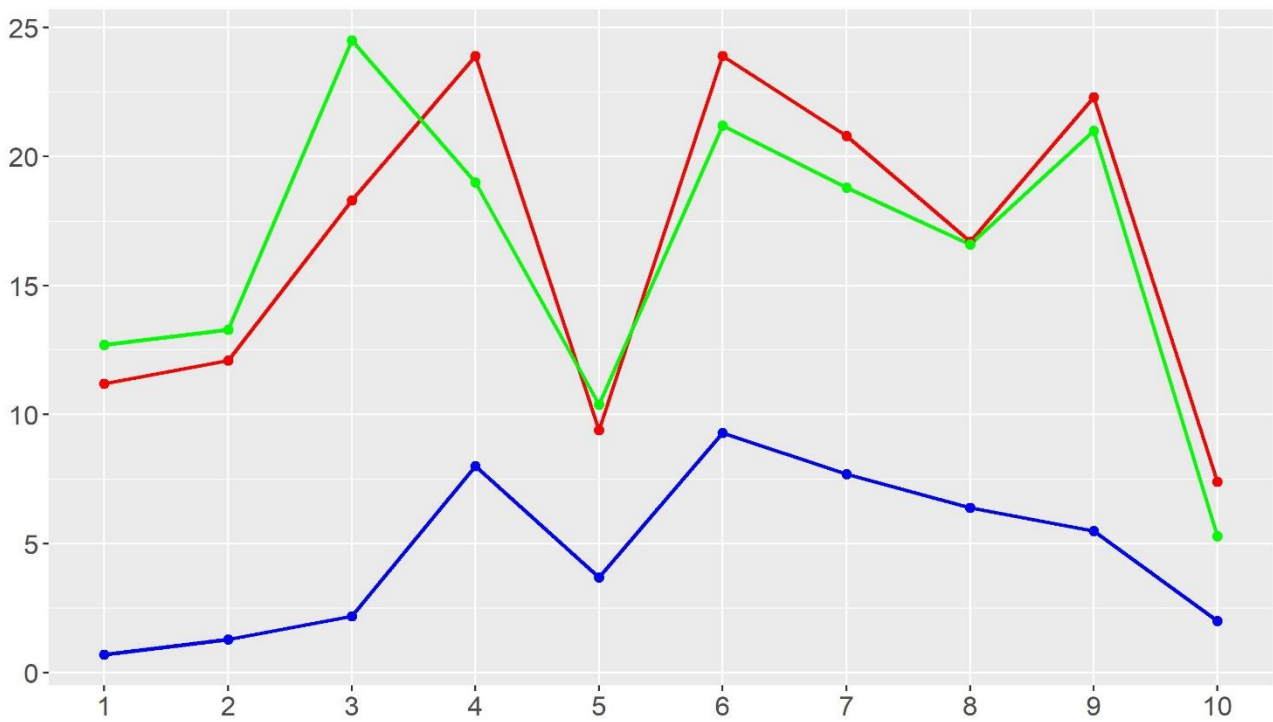


Abbildung 16: Vergleich der Gesamtdistanzen auf Singletrails. Als blaue Punkte und Linien sind die berechneten Gesamtdistanzen auf Singletrails je GPS-Track dargestellt. Als rote Punkte und Linien sind die berechneten Gesamtdistanzen auf Singletrails und Feldwegen je GPS-Track dargestellt. Die grünen Punkte und Linien stellen die Werte des Webdienstes Komoot dar. X-Achse: Nummer des GPS-Tracks. Y-Achse: Distanz in km.

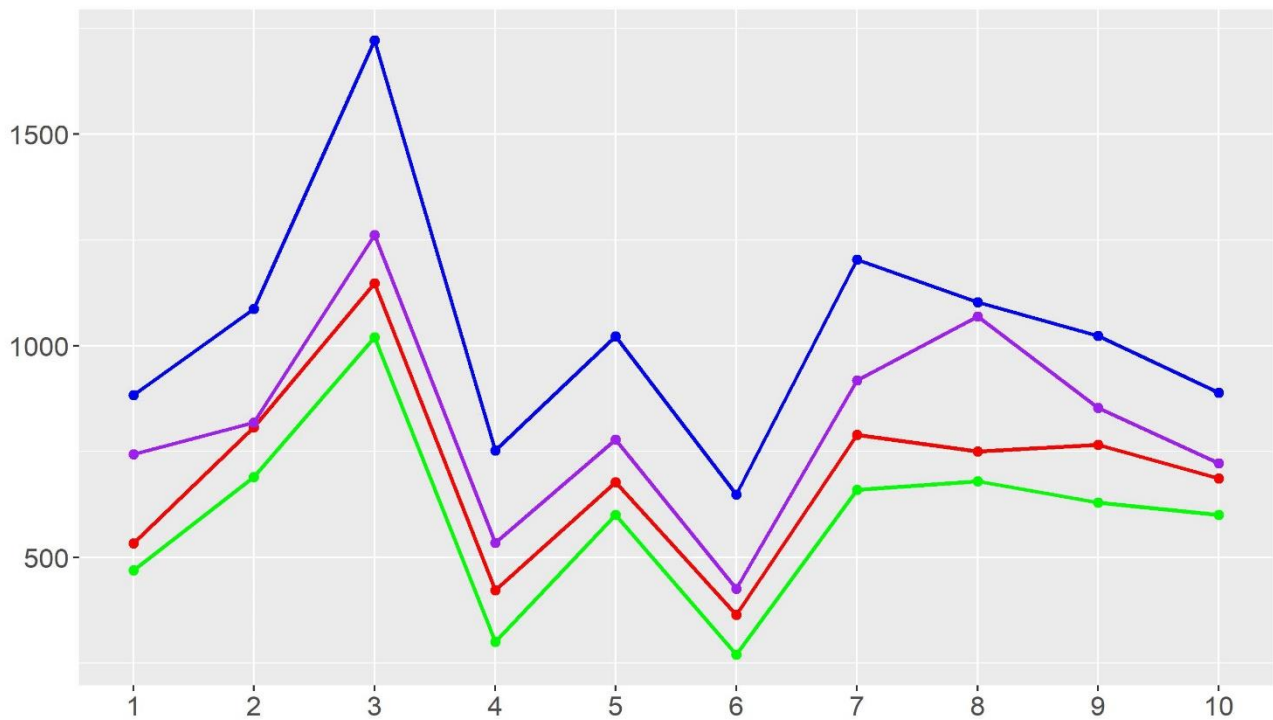


Abbildung 17: Vergleich der berechneten Höhenmeter bergauf mit den Werten des Webdienstes Komoot von 10 GPS-Tracks. Die blauen Punkte und Linien stellen die berechneten Höhenmeter bergauf mit Höhenangaben von Swisstopo (SwissALTI3D) dar. In Violett sind die berechneten Höhenmeter bergauf mit Höhenangaben aus den GPX-Dateien dargestellt. In Rot sind die berechneten Höhenmeter bergauf mit den Höhenangaben von Swisstopo (SwissALTI3D) dargestellt mit dem Unterschied, dass nur jeder zehnte GPS-Punkt für die Berechnung verwendet wurde. In Grün sind die Höhenmeter bergauf des Webdienstes Komoot dargestellt. X-Achse: Nummer des GPS-Tracks. Y-Achse: Höhenmeter bergauf in m.

## 6.4 Personalisierte Mountainbike-Tourenvorschläge

In Tabelle 5 ist für jeden Mountainbiker die vorgeschlagene Mountainbike-Tour (Vorschlag Nr.) aufgelistet. Zusätzlich können die erreichten Punkte für jeden einzelnen Tourenparameter abgelesen werden. Mountainbiker *K* erhielt zwei Tourenvorschläge da beide Vorschläge die gleiche Gesamtpunktzahl erreichten. Die Mountainbiker *B*, *C*, *F*, *H* und *J* erhielten die Tour Nr. 9 als Vorschlag. Drei der Mountainbiker, die die Tour Nr. 9 als Vorschlag erhielten, weisen sogar die Gesamtpunktzahl 26 auf. Alle drei haben jedoch eine andere Punktekombination, welche zur gleichen Gesamtpunktezahl führte. Ebenfalls den gleichen Tourenvorschlag haben Mountainbiker *A* und *D*. Sie erreichten jedoch beide unterschiedliche Gesamtpunktzahlen. Die kleinste Übereinstimmung für eine vorgeschlagene Tour erreichte Mountainbiker *H* mit 33% (13/40 Punkten) für die Nummer 3. Die grösste Übereinstimmung wurde von Mountainbiker *D* mit der Tour Nr. 2 mit 73% (29/40 Punkten) erreicht. Die durchschnittliche Übereinstimmung betrug 60% Prozent.

Tabelle 5: Mountainbiker (A-K) mit jeweiligem Tourenvorschlag, erreichte Gesamtpunktzahl und Anzahl Punkte pro Tourenparameter für den jeweiligen Tourenvorschlag und die Übereinstimmung des Tourenvorschlags mit dem Tourenprofil in %.

Mountainbiker	A	B	C	D	E	F	G	H	J	K	K
Vorschlag Nr.	2	9	9	2	6	9	8	9	9	1	3
Länge MTB-Tour	2	0	2	2	2	0	2	2	2	2	2
Cross-Country-Index	0	2	2	0	0	2	2	0	2	2	0
Gesamtdistanz Singletrail	0	2	2	2	2	2	2	0	0	2	2
Anteil Singletrail	0	2	2	2	2	2	2	0	0	0	2
Gesamthöhe Uphill	1	0	1	1	1	0	1	1	1	1	1
Gesamthöhe Downhill	1	0	1	1	1	0	1	1	1	1	1
Uphill-Downhill-Verhältnis	2	2	2	2	2	2	2	0	2	2	2
Gesamtanstieg	2	0	0	2	0	0	2	0	2	0	0
Distanz Singletrail Anstieg	2	2	2	2	2	2	0	2	0	0	2
Anteil Singletrail Anstieg	2	2	2	2	2	2	2	0	0	0	0
Distanz Anstiege >15%	0	0	0	0	2	0	2	0	2	2	2
Anteil Anstiege >15%	0	2	2	0	0	2	2	2	2	0	2
Gesamtdistanz Abfahrten	2	0	0	2	0	0	2	0	2	2	0
Distanz Singletrail Abfahrten	2	2	2	2	2	0	2	2	2	2	2
Anteil Singletrail Abfahrten	0	2	2	0	2	2	2	0	2	2	2
Distanz Singletrail Abfahrten <40%	1	1	1	1	0	0	1	1	1	1	0
Anteil Singletrail Abfahrten <40%	0	1	1	0	1	1	1	0	1	1	1
Distanz Singletrail Abfahrten >40% und <70%	1	0	0	1	0	0	0	0	0	0	0
Anteil Singletrail Abfahrten >40% und <70%	1	0	0	1	1	0	0	0	0	0	1
Distanz Singletrail Abfahrten >70%	0	1	0	1	0	1	0	0	1	1	1
Anteil Singletrail Abfahrten >70%	0	1	0	1	0	1	0	0	1	1	1
Gesamtdistanz flach	2	0	0	2	0	0	0	0	0	2	0
Distanz Singletrail flach	0	2	0	0	0	2	0	0	0	0	0
Anteil Singletrail flach	0	2	2	2	0	2	0	2	2	0	0
<b>Gesamtpunktzahl</b>	<b>21</b>	<b>26</b>	<b>26</b>	<b>29</b>	<b>22</b>	<b>23</b>	<b>28</b>	<b>13</b>	<b>26</b>	<b>24</b>	<b>24</b>
Übereinstimmung in %	53%	65%	65%	73%	55%	58%	70%	33%	65%	60%	60%

Die Abbildung 18, Abbildung 19, Abbildung 20 und Abbildung 21 zeigen die Verteilung ausgewählter Tourenparameter für jeden Mountainbiker. Die farbigen Punkte zeigen die Wert des

jeweiligen Tourenparameters der verschiedenen Tourenvorschläge an. Sind die Punkte vergrössert, so unterscheiden sie sich nicht signifikant (Signifikanzniveau  $\alpha=0.05$ ) von den Tourenparameter des jeweiligen Mountainbikers und erhalten einen Punkt multipliziert mit dem Gewichtungsfaktor. In Abbildung 18 ist zu sehen, dass die meisten Mountainbike-Touren im Bereich von 10 bis 40 km liegen. Abbildung 19 zeigt den Singletrail-Anteil der Mountainbike-Touren an. Der Hauptanteil, der aufgezeichneten GPS-Tracks bewegt sich um die 25%. Der Cross-Country-Index liegt bei den Touren der Mountainbiker eher höher als bei den Tourenvorschlägen (Abbildung 20). Beim Uphill-Downhill-Verhältnis stechen Mountainbiker *D* und *G* klar heraus. Bei Mountainbiker *G* liegt der Median sogar unter 1. Daraus kann geschlossen werden, dass diese Person mit Sicherheit Bergfahrten meidet.

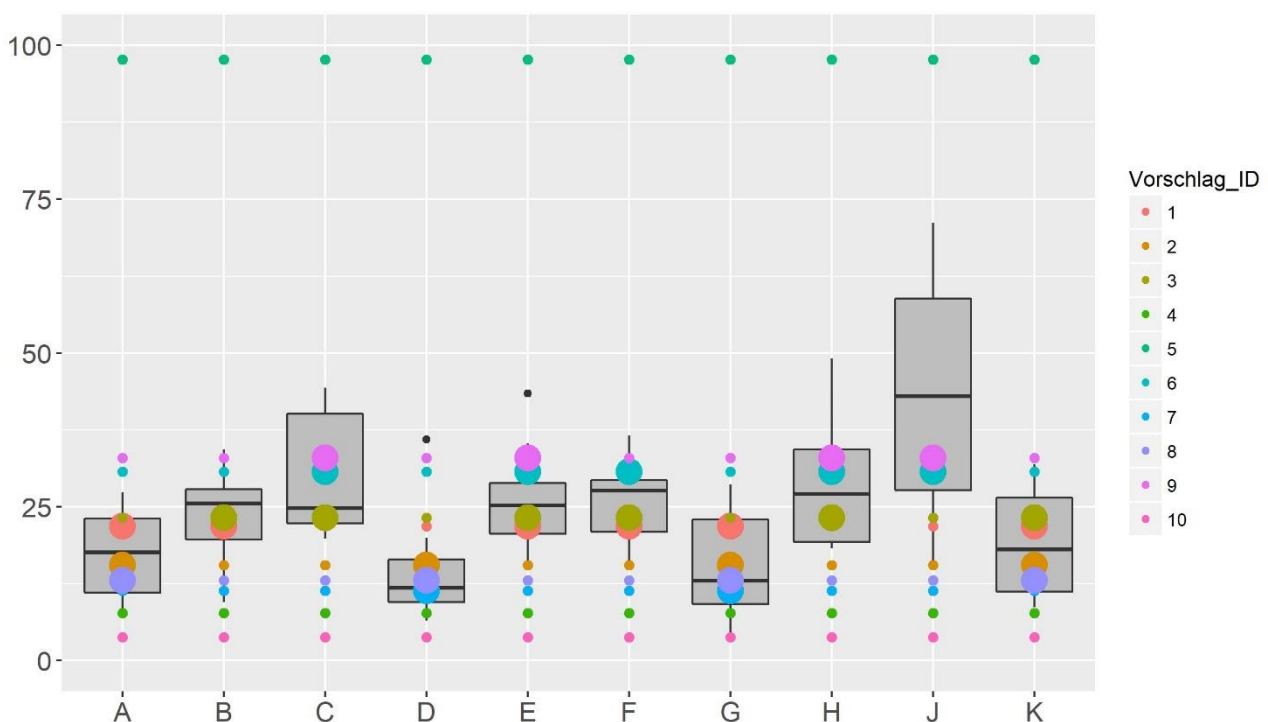


Abbildung 18: Tourenparameter „Länge Mountainbike-Tour“: Die Boxplots stellen die Längenverteilung der Mountainbike-Touren des jeweiligen Mountainbikers dar. Die farbigen Punkte stellen die Länge des jeweiligen Tourenvorschlags dar. Grosse Punkte zeigen die Übereinstimmung mit dem Tourenparameter des jeweiligen Mountainbikers an. X-Achse: Mountainbiker A-K. Y-Achse: Distanz in Kilometer.

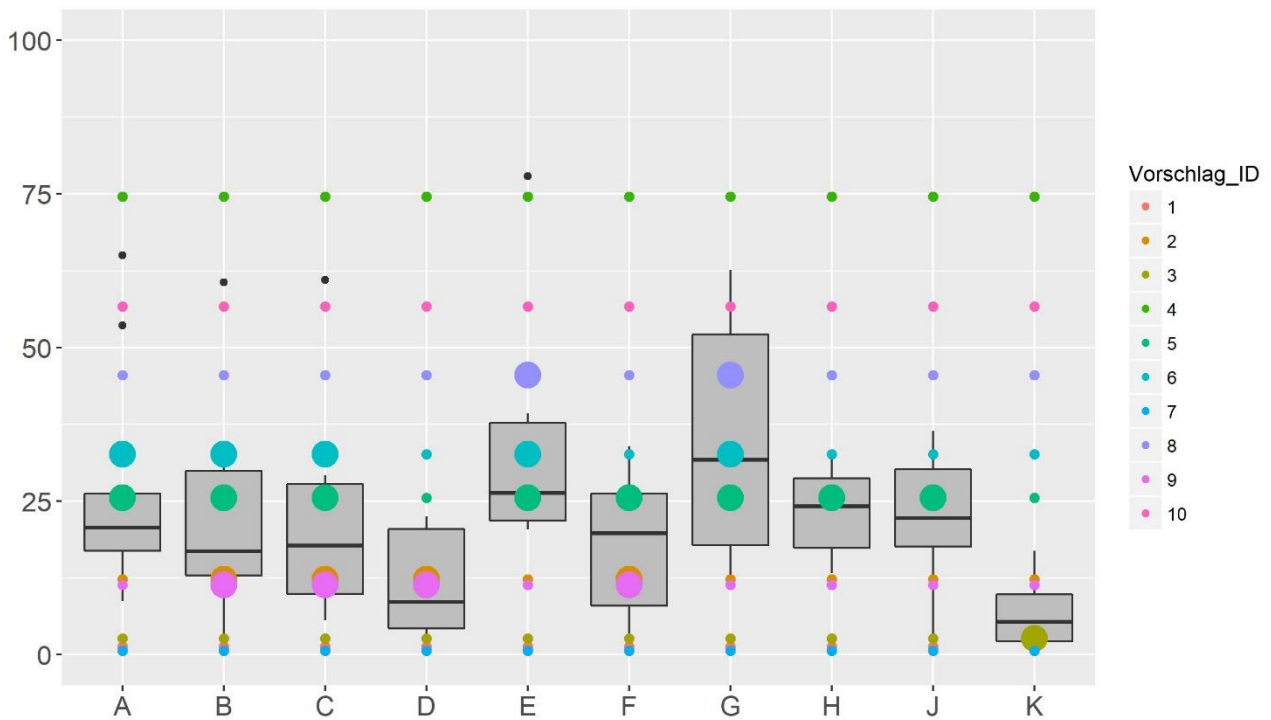


Abbildung 19: Tourenparameter „Anteil Singletrail“: Die Boxplots stellen die Verteilung der Singletrail-Anteile des jeweiligen Mountainbikers dar. Die farbigen Punkte stellen den Anteil an Singletrails des jeweiligen Tourenvorschlags dar. Grosse Punkte zeigen die Übereinstimmung mit dem Tourenparameter des jeweiligen Mountainbikers an. X-Achse: Mountainbiker A-K. Y-Achse: Anteil in %.

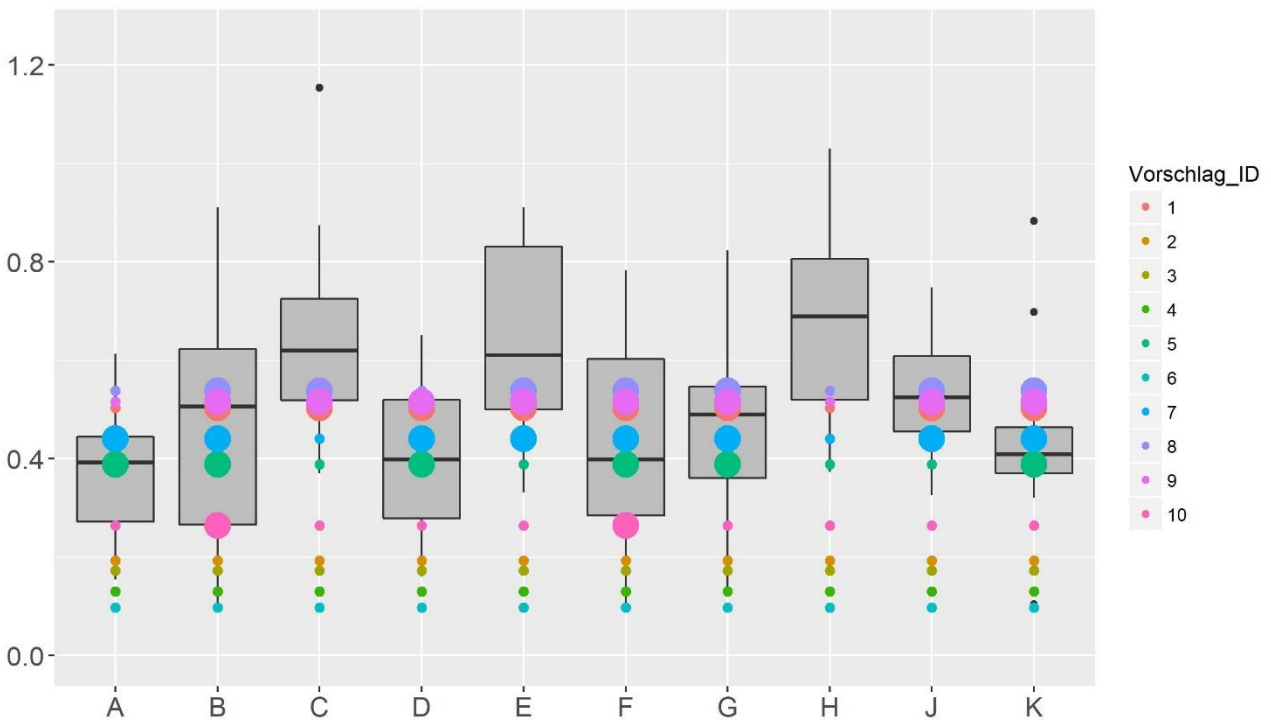


Abbildung 20: Tourenparameter „Cross-Country-Index“: Die Boxplots stellen den Cross-Country-Index des jeweiligen Mountainbikers dar. Die farbigen Punkte stellen den Cross-Country-Index des jeweiligen Tourenvorschlags dar. Grosse Punkte zeigen die Übereinstimmung mit dem Tourenparameter des jeweiligen Mountainbikers an. X-Achse: Mountainbiker A-K. Y-Achse: Cross-Country-Index (Summe der Anzahl Anstiege und Abfahrten pro Kilometer)



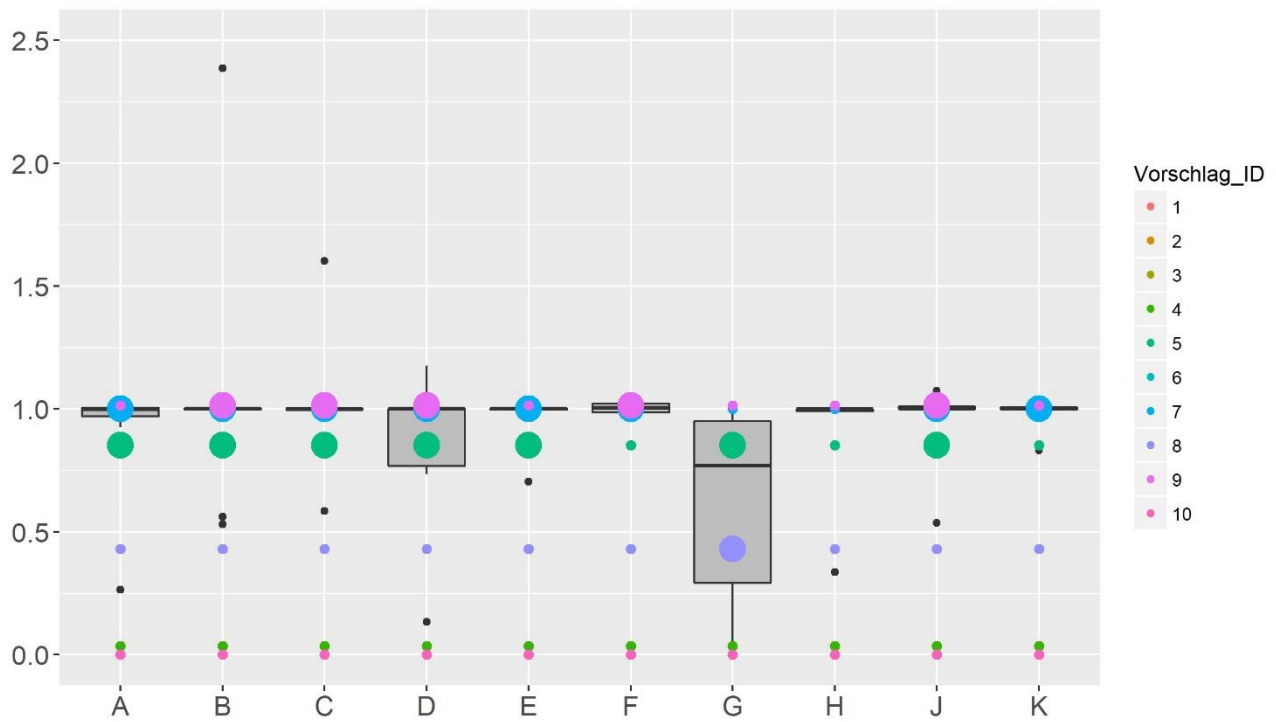


Abbildung 21: Tourenparameter „Uphill-Downhill-Verhältnis“: Die Boxplots zeigen die Verhältnisse von Höhenmeter bergauf zu Höhenmeter bergab (Y-Achse) für jeden Mountainbiker (A-K) (X-Achse) an. Ein Wert >1 bedeutet mehr Höhenmeter bergauf als bergab. 0 bedeutet keine Höhenmeter bergauf. Die farbigen Punkte zeigen das Uphill-Downhill-Verhältnis des jeweiligen Tourenvorschlags an. Grosse Punkte zeigen die Übereinstimmung mit dem Tourenparameter des jeweiligen Mountainbikers an.

## 7 Diskussion

### 7.1 Segmentierung: Anstiege, Abfahrten und flache Abschnitte

Die statistische Auswertung der verschiedenen Kombinationsmöglichkeiten von Anzahl n-Punkte (Grösse des „Moving Window“ für den mittleren Höhenunterschied) und der verschiedenen Schwellenwerte ergab keine signifikanten Unterschiede. Bei der durchgeführten Untersuchung der Segmentation von Anstiegen, Abfahrten und flachen Abschnitten wurde die Aufzeichnungsrate der verschiedenen GPS-Geräte komplett ignoriert. Jede Art von abgeleitetem Bewegungsparameter ist stark durch die Aufzeichnungsrate beeinflusst (Laube & Purves, 2011). Um den Einfluss der Aufzeichnungsrate zu senken, wurde bei Kessler (2017) für die Berechnung der Sinuosität eine fixe Abschnittlänge verwendet. Im vorliegenden Fall wäre die Berechnung des mittleren Höhenunterschieds über eine fixe Abschnittlänge anstatt über eine fixe Anzahl n-Punkte möglich gewesen, technisch jedoch viel aufwändiger.

Bei der visuellen Aussortierung anhand der markierten Abschnitte an einem Höhenprofil konnten grobe Fehler in der Segmentierung erkannt werden (siehe Abbildung 12). Kleinere Fehler konnten allerdings schlechter eruiert werden und sind auch bei der schlussendlich gewählten Kombination von n-Punkten und Schwellenwert vorhanden. Die Fehler treten jedoch vor allem bei kleineren Anstiegen oder Abfahrten auf, welche sich im Bereich von 50 Meter befinden. Grössere Abschnitte werden bei allen verwendeten Kombinationen korrekt markiert (vergleiche Abbildung 11 und Abbildung 12). Die Art der Segmentierung kann als zielführend und gesamtheitlich als richtig betrachtet werden.

### 7.2 Segmentierung der Verkehrsmittel

Bei der Segmentierung nach Verkehrsmitteln wurde festgestellt, dass nicht alle Bahnen erkannt wurden. Nicht erkannte Bahnen mussten von Hand nachgetragen werden. Die Erkennung von Verkehrsmittel wurde anhand der Geschwindigkeit und des Gefälle durchgeführt, was eine relativ einfache Methode darstellt und somit auch einfach zu programmieren ist. Um Verkehrsmittel mit einer grösseren Sicherheit zu segmentieren, könnten weitere Parameter wie Sinuosität oder Ausrichtung verwendet werden. Eine intensive Auseinandersetzung mit der konzeptionellen Bewegungsstruktur, wie sie Dodge et al. (2009) propagieren wäre ratsam. Eine weitere Möglichkeit bestünde darin, die Verkehrsmittel-Information auf die gleiche Weise wie die Wegtyp-Information mittels Map-Matching zu gewinnen. Allerdings könnten dabei Fehler auftreten, wenn sich der Mountainbiker bei der Abfahrt im Falle einer Luftseilbahn unter der Bahn bewegt.

### **7.3 Vergleich von Gesamtdistanz, Strecke auf Singletrails und Höhenmeter bergauf mit Webdienst**

Die Vergleiche der Gesamtdistanzen mit dem Webdienst Komoot haben sehr gute Resultate ergeben. Es hat sich gezeigt, dass die selbstberechneten Werte fast exakt mit denen von Komoot übereinstimmen.

Unterschiede bei den Distanzen von Singletrails haben zeigen ein Definitionsproblem auf. Komoot scheint einen Singletrail anders definiert zu haben oder die Datengrundlage von Komoot enthält Fehler. Da die Datengrundlage von Komoot Openstreetmap ist, muss mit grösseren Fehlern gerechnet werden. Die Datengrundlage für die selbst berechneten Distanzen stammt von Swisstopo, welche eine grosse Zuverlässigkeit aufweist. Allerdings wurden auch hier Mängel festgestellt. Speziell für Mountainbiker gebaute Trails sind oft nicht kartiert.

Bei den Vergleichen der Höhenmeter bergauf, können die Unterschiede auf die Genauigkeit der Daten zurückgeführt werden. Werden die Höhenangaben aus einem sehr feinmaschigen Raster extrahiert, so wird die Differenz zu den Komoot-Werten immer grösser. Bei einem feinmaschigen Raster haben allerdings GPS-Positionsfehler einen grösseren Einfluss.

### **7.4 Personalisierte Mountainbike-Tourenvorschläge**

Die entwickelte Methode zur Erstellung von personalisierten Tourenprofilen kann als einfach und zielführend beschrieben werden. Bewegungs- und Umweltfaktoren werden in 26 verschiedene Tourenparameter abgeleitet. Alle Tourenparameter scheinen nachvollziehbare Werte abzuliefern. Ausser die vier Tourenparameter, welche die Abfahrten auf Singletrails mit einem Gefälle  $>40\%$   $<70\%$  und  $>70\%$  behandeln. Sie enthalten oft keine Angaben. Eine Übereinstimmung mit dem Tourenvorschlag entsteht deshalb oft wenn der Vorschlag ebenfalls keine Angaben hat. Dafür ist der Tourenparameter umso wichtiger bei einem Mountainbiker der hohe Werte bei diesen Tourenparametern besitzt. Durch die Glättung des Höhenprofils durch die mittlere Höhendifferenz werden mit grosser Sicherheit viele solcher Abschnitte „weggeglättet“. Die Glättung ist jedoch auch sinnvoll, da im steilen Gelände GPS-Fehler grosse Auswirkungen auf die Höhenangabe haben (Tomlinson, 2015). Die Mountainbiker *B*, *C*, *E*, *F* und *H* unternehmen oft Mountainbike-Touren zusammen, was sich auch in ihren Tourenprofilen niederschlägt. Und dies obwohl sie unterschiedliche GPS-Geräte verwenden, die sich in ihrer Aufzeichnungsrate teilweise um fast das 4-fache unterscheiden (siehe Tabelle 2 in Kapitel 3.1.1). Dies spricht für die Robustheit des entwickelten Algorithmus. Das Segmentieren der GPS-Tracks der Mountainbiker in Übergänge und Mountainbike-Tour kann als sinnvoll betrachtet werden. Es verhindert, dass der Singletrail-Anteil unterschätzt wird und führt somit zu genaueren Tourenprofilen, was wiederum zu besseren Tourenvorschlägen führt. Bei den Mountainbike-Touren, die als Vorschläge verwendet werden,

wurde bewusst auf die Segmentation von Übergängen verzichtet. Sie sind in der Regel schon optimal zugeschnitten und enthalten keine langen Anfahrtswege.

Bei Tourenvorschlägen wird nicht zwischen passend oder unpassend unterschieden. Es wird eine Rangliste der Vorschläge erstellt. Ein Mountainbiker erhält somit immer einen Tourenvorschlag, solange mindestens ein Tourenparameter mindestens bei einer der zur auswahlstehenden Touren einen Punkt erhält. Ob diese Tour wirklich den Ansprüchen des Mountainbikers entspricht, kann nicht beurteilt werden. Dies auch deshalb, weil weitere Faktoren wie beispielsweise Aussicht, Erreichbarkeit, Verpflegungsmöglichkeiten, etc. für die Tourenwahl von Bedeutung sind. Der entwickelte Algorithmus leitet nur Parameter ab die direkte oder indirekte Indikatoren für Kondition und/oder Fahrtechnik sind.

## 8 Schlussfolgerung

Im Zuge dieser Arbeit wurde ein hoch automatisierter Algorithmus entwickelt, welcher fähig ist, aus verschiedenen GPS-Tracks ein Tourenprofil zu berechnen, es mit anderen Touren zu vergleichen und schliesslich einen personalisierten Tourenvorschlag zu unterbreiten. Die unterschiedlichen Methoden, die zur Aufbereitung der Daten verwendet wurden, lieferten keine fehlerfreien Resultate, sind jedoch gesamtheitlich plausibel und richtig.

Die gewählte Methode zur Durchführung des Map-Matching kann als sehr geeignet angesehen werden. Da sich Mountainbiker relativ oft auf nicht-kartierten Pfaden bewegen, kann ein geeigneter „Snapping“-Radius das Resultat der Karteneinpassung zusätzlich verbessern. Ansonsten würde ein beachtlicher Teil an gefahrenen Singletrails nicht als solche kategorisiert und stattdessen den Wegtyp der nächstgelegenen Strasse erhalten. Die Plausibilitätsprüfung mit dem Webdienst Komoot (siehe Kapitel 6.3) hat gezeigt, dass wenn Feldwege ebenfalls als Singletrails definiert werden, kaum Unterschiede zwischen Webdienst und eigenem Map-Matching bestehen. Daraus kann geschlossen werden, dass Komoot Feldwege ebenfalls als Singletrails einstuft und dass das in dieser Arbeit durchgeführte Map-Matching korrekte Resultate liefert.

Die Segmentierung der GPS-Tracks war nötig, um Anfahrtswege, Heimfahren oder Übergangswegen von bestimmten Berechnungen der Tourenparameter auszuschliessen. Bestimmte Tourenparameter erhielten so eine höhere Genauigkeit wie zum Beispiel die Gesamtdistanz an Singletrails auf der gesamten Mountainbike-Tour. Ebenfalls wurde durch die Segmentierung die Verwendung von Verkehrsmitteln durch den Mountainbiker erkannt. Dies allerdings mit einigen Fehlern (Abbildung 14). Wie die Abbildung 11 und Abbildung 12 zeigten, segmentiert der Algorithmus praktisch einwandfrei die Anstiege, Abfahrten und flachen Abschnitte.

Die Vergabe von Punkten bei den Vergleichen von Tourenparametern des Mountainbikers mit den Tourenparametern der Vorschläge ist simple gelöst, einfach reproduzierbar und statistisch quantifizierbar. Der Wilcoxon-Test eignet sich besonders gut, da er mit Daten arbeitet, welche parameterfrei sind und keine Normalverteilung benötigen.

Die hier entwickelte Methode arbeitet mit Parameter die für den Mountainbiker direkt oder indirekt die Schwierigkeit oder Kondition repräsentieren. Andere Faktoren, wie die Schönheit einer Tour wurden ausser Acht gelassen. Dies könnte bei einer Weiterentwicklung des Algorithmus miteinbezogen werden. Ebenfalls wurden viele Einflussfaktoren noch nicht ausreichend geprüft. Die Auswirkung der Aufzeichnungsrate auf die Segmentierung wurde nicht eruiert. Die Erkennung von Verkehrsmitteln muss unbedingt weiterentwickelt werden. Das wichtigste wäre allerdings eine Studie mit verschiedenen Mountainbikern, die den Algorithmus anwenden und vom Algorithmus gemachte Vorschläge testen und bewerten. Dies würde zeigen, ob der entwickelte Algorithmus praxistauglich ist.

## 9 Dank

Hiermit möchte ich mich herzlichst bei all meinen Unterstützern bedanken. Ohne Euch wäre diese Arbeit nicht möglich gewesen. Ein spezieller Dank geht an meinen Korrektor Dr. Patrick Laube, welcher mich immer mit wertvollen Ratschlägen unterstützt hat und sehr kurzfristig um Hilfe gebeten werden konnte. Ein weiterer Dank geht an meine Freundin Patrizia End und meine Mutter Mirta Kessler, die mich täglich Motiviert und zu Höchstleistungen angetrieben haben. Zum Schluss danke ich allen Mountainbike-Touren-Gebern. Ohne diese hätte ich keine Datengrundlage.

## 10 Literaturverzeichnis

- Buschor, J. (Ed.). (2011). *Singletrails in den schweizer alpen - die 100 schönsten mountainbike-touren*. Aarau und München: AT Verlag.
- Dalumpines, R., & Scott, D. M. (2011). GIS-based map-matching: Development and demonstration of a postprocessing map-matching algorithm for transportation research. *Advancing Geoinformation Science for a Changing World*, , 101-120.
- Dodge, S., Weibel, R., & Forootan, E. (2009). Revealing the physics of movement: Comparing the similarity of movement characteristics of different types of moving objects. *Computers, Environment and Urban Systems*, 33(6), 419-434.
- Dodge, S., Weibel, R., & Laube, P. (2009). Exploring movement-similarity analysis of moving objects. *SIGSPATIAL Special*, 1(3), 11-16.
- Ellersiek, T., Liebig, T., Hecker, D., & Körner, C. (2012). *Analyse von raumzeitlichen bewegungsmustern auf basis von bluetooth-sensoren*
- Gruber, M. (2017). Mountainbiker.it. Retrieved from [www.mountainbiker.it](http://www.mountainbiker.it)
- Kessler, S. (2017). *Kategorisierung von mountainbike-touren mittels bewegungsdaten von mountainbiker*. (Unpublished Semesterarbeit 2). ZHAW Departement N, Wädenswil.
- Komoot GmbH. (2017). Komoot.ch. Retrieved from [www.komoot.de](http://www.komoot.de)
- Lamprecht, M., Fischer, A., & Stamm, H. (2014). Sport schweiz 2014. *Sportaktivität Und Sportinteresse Der Schweizer Bevölkerung*. Zürich: Seismo,
- Laube, P. (2014). *Computational movement analysis* Springer.
- Laube, P., & Purves, R. S. (2011). How fast is a cow? cross-scale analysis of movement data. *Transactions in GIS*, 15(3), 401-418.

- Lenzerheide Marketing und Support AG. (2017). Lenzerheide.ch. Retrieved from [www.lenzerheide.ch](http://www.lenzerheide.ch)
- Marchal, F., Hackney, J., & Axhausen, K. (2004). Efficient map-matching of large GPS data sets- tests on a speed monitoring experiment in zurich. *Arbeitsbericht Verkehrs-Und Raumplanung*, 244
- Mountainbiker.ch. (2017). Mountainbiker.ch. Retrieved from [www.mountainbiker.ch](http://www.mountainbiker.ch)
- Outdooractive GmbH. (2017). Gps-tracks.com. Retrieved from [www.gps-tracks.com](http://www.gps-tracks.com)
- Outdooractive GmbH & Co. KG. (2017). Outdooractive.com. Retrieved from [www.outdooractive.com](http://www.outdooractive.com)
- Outkomm GmbH. (2014). Supertrail-map. Retrieved from [www.supertrail-map.com](http://www.supertrail-map.com)
- Pinkbike.com. (2017). Trailforks.com. Retrieved from [www.trailforks.com](http://www.trailforks.com)
- Quddus, M. A., Ochieng, W. Y., & Noland, R. B. (2007). Current map-matching algorithms for transport applications: State-of-the art and future research directions. *Transportation Research Part C: Emerging Technologies*, 15(5), 312-328.
- Schymik, C., Harald, P. & Werner, D. (2008). Singletrail-skala. Retrieved from [www.singletrail-skala.de](http://www.singletrail-skala.de)
- Skuballa, A. (2015). *Analyse und bereitstellung von positionsbasierten mobilfunkmessungen zur trace-basierten V2X-simulation* (Masterarbeit).
- Stiftung SchweizMobil. (2017). Mountainbikeland.ch. Retrieved from [www.mountainbikeland.ch](http://www.mountainbikeland.ch)
- Strava Inc. (2017). Strava.com. Retrieved from [www.strava.com](http://www.strava.com)
- Swiss Sports Publishing GmbH. (2015). *Swiss singletrail map* Swiss Sports Publishing GmbH.



Tomlinson, D. (2015). *Wie schnell fahren mountainbiker? eine GIS gestützte analyse basierend auf bewegungsdaten von mountainbikern.* (Unpublished Bachelorthesis). ZHAW Departement N, Wädenswil.

## 11 Abbildungsverzeichnis

Abbildung 1: Beispiel für eine Karteneinpassung (Map-Matching). In Rot: Original GPS-Punkte. In Blau: Eingepasste GPS-Punkte. ....	12
Abbildung 2: Ablauf der Teilschritte mit Angaben zu den verwendeten Programmen.....	17
Abbildung 3: Kürzeste Route mit nur jedem zehnten GPS-Punkt. ....	19
Abbildung 4: Kürzeste Route mit allen GPS-Punkten.....	20
Abbildung 5: Durch Barriere (rot) verursachtes Fehlen von Verbindungen zwischen den Wegpunkten (blau) 26 und 32. ....	21
Abbildung 6: Fehler, die bei der verwendeten Map-Matching-Methode entstehen können. Hier im Bild werden 4 Punkte auf das Wegenetz eingepasst, obwohl sie sehr wahrscheinlich neben dem Weg sind, da hier ein nicht-kartierter Singletrail befahren wurde. Rote Punkte: Original GPS-Position. Blaue Punkte: Eingepasste Punkte. Roter Kreis: Snapping Radius (20m). Schwarze Linie: Teile des Wegenetz. ....	22
Abbildung 7: GPS-Track Nr. 5 Mountainbiker G mit Segmentierung nach Übergängen (rot). ....	28
Abbildung 8: Vergleich der Punktedifferenz von manueller Segmentierung zu automatischer Segmentierung aufgeteilt auf jede getestete Kombination von n-Punkte und Schwellenwert. X-Achse: Kombinationsname; N+Zahl entspricht der Anzahl n-Punkte für die Berechnung des mittleren Höhenunterschieds; S+Zahl entspricht der Höhe des Schwellenwertes in Meter. Y-Achse: Punktedifferenz. ....	32
Abbildung 9: Vergleich der Punktedifferenzen nach n-Punkte gruppiert. X-Achse: Gruppen (N13 = mittlere Höhendifferenz über 13 Punkte berechnet, N25 = 25 Punkte, etc.). Y-Achse: Punktedifferenz.....	33
Abbildung 10: Vergleich der Punktedifferenzen nach Schwellenwerte gruppiert. X-Achse: Schwellenwertgruppe (S10 = Schwellenwert 10 Meter, etc.). Y-Achse: Punktedifferenz. ....	33
Abbildung 11: Höhenprofil mit markierten Anstiegen (grün), Abfahrten (rot) und flachen Abschnitten (schwarz). Die Segmentierung der Abschnitte erfolgte mit einer Anzahl n-Punkte von 13 und einem Schwellenwert von 25 Metern. X-Achse: Horizontale Distanz in Metern. Y-Achse: Meter über Meer. ....	34
Abbildung 12: Höhenprofil mit markierten Anstiegen (grün), Abfahrten (rot) und flachen Abschnitten (schwarz). Die Segmentierung der Abschnitte erfolgte mit einer Anzahl n-Punkte von 25 und einem Schwellenwert von 50 Metern. X-Achse: Horizontale Distanz in Metern. Y-Achse: Meter über Meer. ....	34

Abbildung 13: Segmentierung der Verkehrsmittel (Raum St. Moritz). In Blau: Standseil-, Luftseil- und Eisenbahnen. Rot: Als Verkehrsmittel segmentierte Abschnitte. Grau: Übrige GPS-Punkte...	35
Abbildung 14: Segmentierung der Verkehrsmittel (Raum Lenzerheide). In Blau: Standseil-, Luftseil- und Eisenbahnen. Rot: Als Verkehrsmittel segmentierte Abschnitte. Grau: Übrige GPS-Punkte...	35
Abbildung 15: Vergleich der berechneten Gesamtdistanzen (blaue Punkte/Linie) mit den Gesamtdistanzen des Webdienstes Komoot (grüne Punkte/Linie) von 10 verschiedenen GPS-Tracks. X-Achse: Nummer des GPS-Tracks. Y-Achse: Distanz in km.....	36
Abbildung 16: Vergleich der Gesamtdistanzen auf Singletrails. Als blaue Punkte und Linien sind die berechneten Gesamtdistanzen auf Singletrails je GPS-Track dargestellt. Als rote Punkte und Linien sind die berechneten Gesamtdistanzen auf Singletrails und Feldwegen je GPS-Track dargestellt. Die grünen Punkte und Linien stellen die Werte des Webdienstes Komoot dar. X-Achse: Nummer des GPS-Tracks. Y-Achse: Distanz in km.....	37
Abbildung 17: Vergleich der berechneten Höhenmeter bergauf mit den Werten des Webdienstes Komoot von 10 GPS-Tracks. Die blauen Punkte und Linien stellen die berechneten Höhenmeter bergauf mit Höhenangaben von Swisstopo (SwissALTI3D) dar. In Violett sind die berechneten Höhenmeter bergauf mit Höhenangaben aus den GPX-Dateien dargestellt. In Rot sind die berechneten Höhenmeter bergauf mit den Höhenangaben von Swisstopo (SwissALTI3D) dargestellt mit dem Unterschied, dass nur jeder zehnte GPS-Punkt für die Berechnung verwendet wurde. In Grün sind die Höhenmeter bergauf des Webdienstes Komoot dargestellt. X-Achse: Nummer des GPS-Tracks. Y-Achse: Höhemeter bergauf in m.....	37
Abbildung 18: Tourenparameter „Länge Mountainbike-Tour“: Die Boxplots stellen die Längenverteilung der Mountainbike-Touren des jeweiligen Mountainbikers dar. Die farbigen Punkte stellen die Länge des jeweiligen Tourenvorschlags dar. Grosse Punkte zeigen die Übereinstimmung mit dem Tourenparameter des jeweiligen Mountainbikers an. X-Achse: Mountainbiker A-K. Y-Achse: Distanz in Kilometer.....	39
Abbildung 19: Tourenparameter „Anteil Singletrail“: Die Boxplots stellen die Verteilung der Singletrail-Anteile des jeweiligen Mountainbikers dar. Die farbigen Punkte stellen den Anteil an Singletrails des jeweiligen Tourenvorschlags dar. Grosse Punkte zeigen die Übereinstimmung mit dem Tourenparameter des jeweiligen Mountainbikers an. X-Achse: Mountainbiker A-K. Y-Achse: Anteil in %.....	40
Abbildung 20: Tourenparameter „Cross-Country-Index“: Die Boxplots stellen den Cross-Country-Index des jeweiligen Mountainbikers dar. Die farbigen Punkte stellen den Cross-Country-Index des jeweiligen Tourenvorschlags dar. Grosse Punkte zeigen die Übereinstimmung mit dem Tourenparameter des jeweiligen Mountainbikers an. X-Achse: Mountainbiker A-K. Y-Achse: Cross-Country-Index (Summe der Anzahl Anstiege und Abfahrten pro Kilometer) .....	40

Abbildung 21: Tourenparameter „Uphill-Downhill-Verhältnis“: Die Boxplots zeigen die Verhältnisse von Höhenmeter bergauf zu Höhenmeter bergab (Y-Achse) für jeden Mountainbiker (A-K) (X-Achse) an. Ein Wert  $>1$  bedeutet mehr Höhenmeter bergauf als bergab. 0 bedeutet keine Höhenmeter bergauf. Die farbigen Punkte zeigen das Uphill-Downhill-Verhältnis des jeweiligen Tourenvorschlags an. Grosse Punkte zeigen die Übereinstimmung mit dem Tourenparameter des jeweiligen Mountainbikers an. .... 41

## 12 Tabellenverzeichnis

Tabelle 1: Durchsuchte Medien nach relevanten Toureninformationen. Die Toureninformationen sind nach absteigender Häufigkeit des Auftretens in den untersuchten Quellen sortiert.....	10
Tabelle 2: Durchschnittliche Aufzeichnungsrate der GPS-Geräte von jedem Mountainbiker in Anzahl GPS-Punkte pro Minute. ....	15
Tabelle 3: Verwendete Geodaten; in der Tabelle ist der Name des Datensatzes aufgelistet, der Datentyp, Beschrieb zum Datensatz und der Datenherr.....	16
Tabelle 4: Beispiel der berechneten Parameter für den GPS-Track Nr. 5 von Mountainbiker G. Links: Parameter. Mitte: Parameterwert. Rechts: Masseinheit. ....	27
Tabelle 5: Mountainbiker (A-K) mit jeweiligem Tourenvorschlag, erreichte Gesamtpunktzahl und Anzahl Punkte pro Tourenparameter für den jeweiligen Tourenvorschlag und die Übereinstimmung des Tourenvorschlags mit dem Tourenprofil in %.....	38

## 13 Anhang

Anhang A.	Plagiatserklärung .....	55
Anhang B.	Poster .....	56
Anhang C.	Segmentierung .....	57
Anhang D.	Entwickelter Algorithmus um personalisierte Tourenvorschläge zu unterbreiten .....	58

---

## **Anhang A. Plagiatserklärung**

### **ERKLÄRUNG**

**betreffend das selbständige Verfassen einer Bachelorarbeit**

**im Departement Life Sciences und Facility Management**

Mit der Abgabe dieser Bachelorarbeit versichert der/die Studierende, dass er/sie die Arbeit selbständig und ohne fremde Hilfe verfasst hat.


Der/die unterzeichnende Studierende erklärt, dass alle verwendeten Quellen (auch Internetseiten) im Text oder Anhang korrekt ausgewiesen sind, d.h. dass die Bachelorarbeit keine Plagiate enthält, also keine Teile, die teilweise oder vollständig aus einem fremden Text oder einer fremden Arbeit unter Vorgabe der eigenen Urheberschaft bzw. ohne Quellenangabe übernommen worden sind.

Bei Verfehlungen aller Art treten Paragraph 39 und Paragraph 40 der Rahmenprüfungsordnung für die Bachelor- und Masterstudiengänge an der Zürcher Hochschule für Angewandte Wissenschaften vom 29. Januar 2008 sowie die Bestimmungen der Disziplinarmassnahmen der Hochschulordnung in Kraft.

Ort, Datum:

06.09.2017.....

Unterschrift:

  
.....

## Tourenvorschläge für Mountainbiker aus Bewegungsdaten

### Ausgangslage

Mountainbiking hat sich seit der Jahrtausendwende von einem Trendsport zu einem Massensport entwickelt. Der enorme technische Fortschritt bei den Mountainbikes führte dazu, dass Mountainbiking in immer unwegsamere Gelände möglich wurde und heute sogar auf alpinen Plätzen gefahren werden kann. Über 44% der Schweizerinnen und Schweizer üben Radfahren als Sport aus, wobei 6.3% davon explizit Mountainbiking nennen (Lamprecht, Fischer, & Stamm, 2014). Die stetig wachsende Anzahl an Mountainbikern führt auch dazu, dass immer mehr Mountainbike-Tour-gebiets werden und Tourismusregionen spezielle Touren und Angebote für Mountainbiker entwickeln. Zeitschriften, Internetseiten und Smartphone-Apps bieten eine Fülle an Informationen über Mountainbike-Touren an. Obwohl sich bei Onlineplattformen Suchfilter einsetzen lassen, ist es oft schwierig und zeitaufwendig eine Mountainbike-Tour nach einem eigenen Geschmack zu finden. Apps lassen eine Suche von Touren im Umkreis des aktuellen Standorts zu. Aus den zahlreichen angezeigten Touren ist es dann allerdings mühsam, die passende Tour zu eruieren. Dies auch deshalb, weil Mountainbiking ein sehr dehnbarer Begriff ist. Was einige für eine normale Radtour halten, ist für andere bereits eine Mountainbike-Tour.

In den verschiedensten Bereichen nutzen Menschen ihr Smartphone oder andere Geräte, um ihre Bewegungsdaten mittels GPS aufzuzeichnen. Die Daten werden verwendet, um beispielsweise den Trainingsfortschritt zu dokumentieren oder neu entdeckte Touren mit anderen zu teilen. Krankenkassen ermöglichen ihren Versicherten bereits eine Prämienerkennung, wenn sie ihnen ihre persönlichen Daten zur Verfügung stellen und somit nachweisen können, dass sie sich genügend bewegen. Bewegungsprofile können auch im Bereich des Marketings verwendet werden. (Ellersiek, Liebig, Hecker, & Körner, 2012) verwenden beispielsweise Bewegungsprofile von Besuchern im Duisburger Zoo, um eine Standortbewertung neuer und bestehender Gehege durchzuführen. Im Bereich des Mountainbiking könnten die Bewegungsinformationen genutzt werden, um Tourenpräferenzen festzustellen.

In der vorliegenden Arbeit wird versucht, anhand von aufgezeichneten GPS-Tracks verschiedener Mountainbiker jeweils ein Tourenprofil zu erstellen. Die Erstellung der Tourenprofile soll mittels eines entwickelten Algorithmus möglichst automatisiert erfolgen. Anschliessend werden die Tourenprofile mit einer Liste von verschiedenen Mountainbike-Touren verglichen. Durch den Vergleich soll die für den jeweiligen Mountainbiker passende Tour ermittelt werden.

### Methode

In dieser Arbeit wurde ein Algorithmus entwickelt, welcher aus GPS-Tracks relevante Tourenparameter ableitet und daraus personalisierte Tourenprofile erstellt. Mittels Tourenprofil kann schliesslich aus einer beliebigen Auswahl an Mountainbike-Touren ein personalisierter Tourenvorschlag unterbreitet werden.

In einem ersten Schritt wurden die GPS-Tracks mit zusätzlich Geodaten angereichert. Zu diesem Zweck wurde ein Map-Matching durchgeführt, mit welchem der Waypoint aus einem Strassenattributsextrahiert wurde. In einem zweiten Schritt wurden die GPS-Tracks anhand der Umweltfaktoren und Bewegungsparametern segmentiert. Aus den segmentierten GPS-Tracks wurden verschiedene Tourenparameter berechnet aus denen schliesslich ein personalisiertes Tourenprofil erstellt wurde. In einem letzten Schritt wurden zur Auswahl stehende Mountainbike-Touren mit dem Tourenprofil des Mountainbikers verglichen und anhand eines Punktesystems ein personalisierter Tourenvorschlag unterbreitet.

### Resultate

Der Vergleich der in R implementierten Segmentierung der Anstiege, Abfahrten und flachen Abschnitte mit den manuell vorgenommenen Segmentierungen hat gezeigt, dass sich die automatische Segmentierung im Mittel zu 74% mit der manuellen Segmentierung deckt.

In der Abbildung 1 sind die Punktdifferenzen der einzelnen getesteten Kombination einander gegenübergestellt. Getestet wurde eine Anzahl n-Punkte von 13, 25 und 50. Als Schwellenwerte

wurden die Werte (in Meter) 10, 25 und 50. Die Differenzen wurden für jede mögliche Kombination dieser Werte geprüft. Der durchgeführte Kruskal-Wallis-Test ergab keine signifikanten Unterschiede bei einem p-Wert von 0.998. Der Vergleich der Punktdifferenzen gruppiert nach n-Punkte und Schwellenwert ergab ebenfalls keine signifikanten Unterschiede (siehe Abbildung 2 und Abbildung 3). Der Kruskal-Wallis-Test ergab für den Vergleich der Punktdifferenzen nach n-Punkte Tour mittleren Höhendifferenz sortiert einen p-Wert von 0.93 und für die Differenzen sortiert nach Grösse des Schwellenwertes einen p-Wert von 0.74. Bei allen Tests wurde ein Signifikanzniveau von 0.05 verwendet. Die Daten sind nicht normalverteilt. Die Abbildung 5 und Abbildung 6 zeigen die Verteilung ausgewählter Tourenparameter für jeden Mountainbiker. Die farbigen Punkte zeigen die Werte des jeweiligen Mountainbikers und enthalten einen Punkt multipliziert mit dem Gewichtungsfaktor. Abbildung 6 zeigt die berechneten Gesamtdistanzen (in km) von 10 verschiedenen GPS-Tracks verglichen mit den Werten (in km) des Webdienstes Komoot. Die Werte sind jeweils als Punkte dargestellt. Die Verbindungslinien sollen Unterschiede besser hervorheben. Es ist ersichtlich, dass nur minimale Unterschiede zwischen den selbst berechneten Distanzen und denen von Komoot bestehen. Abbildung 6 zeigt den Vergleich der Länge der Mountainbike-Touren mit Werten des Webdienstes Komoot. Es waren keine grosse Unterschiede vorhanden.

### Fazit

Im Zuge dieser Arbeit wurde ein hoch automatisierter Algorithmus entwickelt, welcher fähig ist, aus verschiedenen GPS-Tracks ein Tourenprofil zu berechnen, es mit anderen Touren zu vergleichen und schliesslich einen personalisierten Tourenvorschlag zu unterbreiten. Die unterschiedlichen Methoden, die zur Aufbereitung der Daten verwendet wurden, lieferten keine einheitlichen Resultate, sind jedoch gesamtheitlich plausibel und richtig.

Die gewählte Methode zur Durchführung des Map-Matching kann als sehr geeignet angesehen werden. Da sich Mountainbiker relativ oft auf nicht kartierten Plätzen bewegen, kann ein geeigneter „Snapping“ Radius das Resultat der Kartierung zusätzlich verbessern. Ansonsten würde ein beachtlicher Teil an gefahrenen Singletrails nicht als solche kategorisiert und stattdessen den Wegtyp der nächstgelegenen Strasse erhalten. Die Plausibilitätsprüfung mit dem Webdienst Komoot hat gezeigt, dass wenn Feldwege ebenfalls als Singletrails erfasst und dass dies in dieser Arbeit durchgeführte Map-Matching korrekte Resultate liefert. Die Segmentierung der GPS-Tracks war nötig, um Anfahrtswege, Heimfahrten oder Übergangswegen von bestimmten Berchnungen der Tourenparameter auszuschliessen. Bestimmte Tourenparameter erhielten so eine höhere Genauigkeit wie zum Beispiel die Gesamtdistanz an Singletrails auf der gesamten Mountainbike-Tour. Ebenfalls wurde durch die Segmentierung die Verwendung von Verkehrsmitteln durch den Mountainbiker erkannt. Dies allerdings mit einigen Fehlern. Wie die Abbildungen 4 zeigt, segmentiert der Algorithmus praktisch einwandfrei die Anstiege, Abfahrten und flachen Abschnitte.

Die Vergabe von Punkten bei den Vergleichen von Tourenparametern des Mountainbikers mit den Tourenparametern der Vorschläge ist simple gelöst, einfach reproduzierbar und statistisch quantifizierbar. Der Wilcoxon-Test eignet sich besonders gut, da er mit Daten arbeitet, welche parameterfrei sind und keine Normalverteilung benötigen. Die hier entwickelte Methode arbeitet mit Parameter die für den Mountainbiker direkt oder indirekt die Schwierigkeit oder Kondition repräsentieren. Andere Faktoren, wie die Schönheit einer Tour wurden ausser Acht gelassen. Dies könnte bei einer Weiterentwicklung des Algorithmus miteinbezogen werden. Ebenfalls wurden viele Einflussfaktoren noch nicht ausreichend geprüft. Die Auswirkung der Aufzeichnungsrate auf die Segmentierung wurde nicht eruiert. Die Erkennung von Verkehrsmitteln muss unbedingt weiterentwickelt werden. Das wichtigste wäre allerdings eine Studie mit verschiedenen Mountainbikern, die den Algorithmus anwenden und vom Algorithmus gemachte Vorschläge testen und bewerten. Dies würde zeigen, ob der entwickelte Algorithmus praxistauglich ist.



Abbildung 1

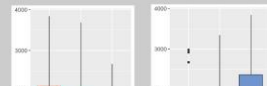


Abbildung 2



Abbildung 3



Abbildung 4

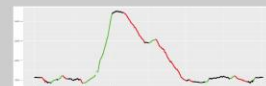


Abbildung 5



Abbildung 6

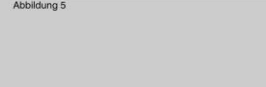
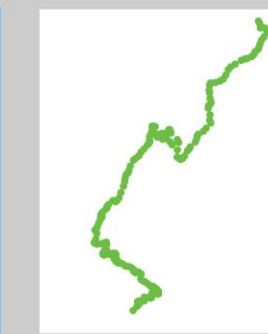
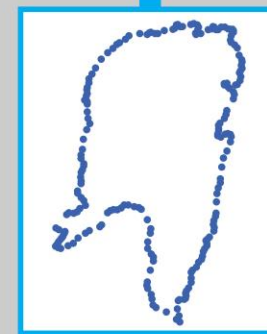
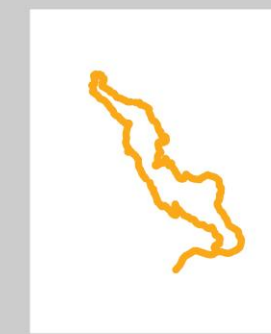
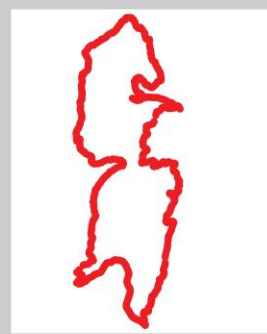


Abbildung 7



Abbildung 8



### Quellen

Lamprecht, M., Fischer, A., & Stamm, H. (2014). Sport Schweiz 2014. Sportaktivität Und Sportinteresse Der Schweizer Bevölkerung Zürich: Salsino.

Ellersiek, T., Liebig, T., Hecker, D., & Körner, C. (2012). Analyse von räumlichen Bewegungsmustern auf Basis von bluetooth-sensoren



## Anhang C. Segmentierung

Name	Tour	N_Punkte	Schwellenwert	Anzahl übereinstimmende Punkte	Total erreichbare Punkte	Differenz	Übereinstimmung in %
N13_S10	B_10	N13	S10	1612	2099	487	77%
N13_S10	E_10	N13	S10	4313	5062	749	85%
N13_S10	H_7	N13	S10	6162	9080	2918	68%
N13_S10	G_2	N13	S10	795	1069	274	74%
N13_S10	C_8	N13	S10	4870	6085	1215	80%
N13_S25	B_10	N13	S25	1664	2099	435	79%
N13_S25	E_10	N13	S25	4275	5062	787	84%
N13_S25	H_7	N13	S25	5742	9080	3338	63%
N13_S25	G_2	N13	S25	836	1069	233	78%
N13_S25	C_8	N13	S25	4405	6085	1680	72%
N13_S50	B_10	N13	S50	1260	2099	839	60%
N13_S50	E_10	N13	S50	4178	5062	884	83%
N13_S50	H_7	N13	S50	5238	9080	3842	58%
N13_S50	G_2	N13	S50	818	1069	251	77%
N13_S50	C_8	N13	S50	3522	6085	2563	58%
N25_S10	B_10	N25	S10	1749	2099	350	83%
N25_S10	E_10	N25	S10	4318	5062	744	85%
N25_S10	H_7	N25	S10	6104	9080	2976	67%
N25_S10	G_2	N25	S10	748	1069	321	70%
N25_S10	C_8	N25	S10	4861	6085	1224	80%
N25_S25	B_10	N25	S25	1787	2099	312	85%
N25_S25	E_10	N25	S25	4373	5062	689	86%
N25_S25	H_7	N25	S25	6080	9080	3000	67%
N25_S25	G_2	N25	S25	761	1069	308	71%
N25_S25	C_8	N25	S25	4568	6085	1517	75%
N25_S50	B_10	N25	S50	1581	2099	518	75%
N25_S50	E_10	N25	S50	4282	5062	780	85%
N25_S50	H_7	N25	S50	5396	9080	3684	59%
N25_S50	G_2	N25	S50	752	1069	317	70%
N25_S50	C_8	N25	S50	3925	6085	2160	65%
N51_S10	B_10	N51	S10	1643	2099	456	78%
N51_S10	E_10	N51	S10	4353	5062	709	86%
N51_S10	H_7	N51	S10	6411	9080	2669	71%
N51_S10	G_2	N51	S10	684	1069	385	64%
N51_S10	C_8	N51	S10	4828	6085	1257	79%
N51_S25	B_10	N51	S25	1732	2099	367	83%
N51_S25	E_10	N51	S25	4480	5062	582	89%
N51_S25	H_7	N51	S25	6439	9080	2641	71%
N51_S25	G_2	N51	S25	687	1069	382	64%
N51_S25	C_8	N51	S25	4686	6085	1399	77%
N51_S50	B_10	N51	S50	1629	2099	470	78%
N51_S50	E_10	N51	S50	4402	5062	660	87%
N51_S50	H_7	N51	S50	6471	9080	2609	71%
N51_S50	G_2	N51	S50	688	1069	381	64%
N51_S50	C_8	N51	S50	3956	6085	2129	65%
<b>Mittelwert</b>							74%
<b>Median</b>							75%

## Anhang D. Entwickelter Algorithmus um personalisierte Tourenvorschläge zu unterbreiten

### Algorithmus\_BA

```
# Arbeitsbereich festlegen
setwd("C:/Users/sebastian/Desktop/ZHAW/Bachelorarbeit/R_Codes")

#####
#####
# Lade Packages
require(rgeos)

## Loading required package: rgeos

## rgeos version: 0.3-23, (SVN revision 546)
## GEOS runtime version: 3.5.0-CAPI-1.9.0 r4084
## Linking to sp version: 1.2-4
## Polygon checking: TRUE

require(rgdal)

## Loading required package: rgdal

## Loading required package: sp

## rgdal: version: 1.2-8, (SVN revision 663)
## Geospatial Data Abstraction Library extensions to R successfully loaded
## Loaded GDAL runtime: GDAL 2.0.1, released 2015/09/15
## Path to GDAL shared files: C:/Users/sebastian/Documents/R/win-library/3.4/rgdal/gdal
## Loaded PROJ.4 runtime: Rel. 4.9.2, 08 September 2015, [PJ_VERSION: 492]
## Path to PROJ.4 shared files: C:/Users/sebastian/Documents/R/win-library/3.4/rgdal/proj
## Linking to sp version: 1.2-5

require(maptools)

## Loading required package: maptools

## Checking rgeos availability: TRUE

require(xlsx)

## Loading required package: xlsx

## Loading required package: rJava

## Loading required package: xlsxjars

require(ggplot2)

## Loading required package: ggplot2

require(RColorBrewer)

## Loading required package: RColorBrewer

#####
#####
# Laden der Shapefiles
layer_names <- c("MTB_A_1", "MTB_A_2", "MTB_A_3", "MTB_A_4", "MTB_A_5", "MTB_A_6", "MTB_A_7", "MTB_A_8",
"MTB_A_9", "MTB_A_10")

layer_names_Vorschlaege <- c("Vorschlag_1", "Vorschlag_2", "Vorschlag_3", "Vorschlag_4", "Vorschlag_5",
"Vorschlag_6", "Vorschlag_7", "Vorschlag_8", "Vorschlag_9", "Vorschlag_10")
```

```

shapefiles <- lapply(layer_names, function(x){readOGR(dsn="C:/Users/sebastian/Desktop/ZHAW/Bachelorarbeit/GIS/Finale_angereicherte_Daten",layer = x)})

shapefiles_vorschlaege <- lapply(layer_names_Vorschlaege, function(x){readOGR(dsn="C:/Users/sebastian/Desktop/ZHAW/Bachelorarbeit/GIS/Tourenvorschlaege/Vorschl_Finale",layer = x)})

# Elemente der Liste benennen
names(shapefiles) <- layer_names
names(shapefiles_vorschlaege) <- layer_names_Vorschlaege

#####
#####
# Zeitformat ändern
changeTimeFormat <- function(shapefile){
  shapefile$DateTime <- as.POSIXct(shapefile$DateTimeS, "%Y-%m-%dT%H:%M:%S",tz="Africa/Algiers")
  return(shapefile)
}
shapefiles <- lapply(shapefiles, changeTimeFormat)
shapefiles_vorschlaege <- lapply(shapefiles_vorschlaege, changeTimeFormat)
#####
#####
# Neue Spalte mit Höhenmeter ohne Datenlücken erstellen

new.altitude <- function(Rastervalue,Z_Punkt,Elevation){
  x=0
  for (i in 1:length(Rastervalue)){
    if (Rastervalue[i]<0 & (is.na(Elevation[i])) & (is.na(Z_Punkt[i]) | Z_Punkt[i]<0)){
      x[i]=NA
    } else if (Rastervalue[i]<0 & (is.na(Z_Punkt[i]) | Z_Punkt[i]<0)){
      x[i]=as.numeric(Elevation[i])
    } else if (Rastervalue[i]<0){
      x[i]=as.numeric(Z_Punkt[i])
    } else {
      x[i]=as.numeric(Rastervalue[i])
    }
  }
  return(x)
}
add.new.altitude <- function(shapefile){
  shapefile$altitude <- new.altitude(shapefile$RASTERVALU,shapefile$POINT_Z,shapefile$Elevation)
  return(shapefile)
}
shapefiles <- lapply(shapefiles,add.new.altitude)
shapefiles_vorschlaege <- lapply(shapefiles_vorschlaege,add.new.altitude)
#####
#####
# planare Distanz zwischen den einzelnen Punkten berechnen
calculateDistplanar <- function(x_Koord,y_Koord){
  dist = x_Koord
  dist = 0
  for (i in 1:(length(x_Koord)-1)){
    dist[i+1] = sqrt(((x_Koord[i] - x_Koord[i+1])^2) + ((y_Koord[i] - y_Koord[i+1])^2))
  }
  return(as.numeric(dist))
}

addplanarDist <- function(shapefile){
  shapefile$planare_Distanz <- calculateDistplanar(shapefile$POINT_X,shapefile$POINT_Y)
  return(shapefile)
}
shapefiles <- lapply(shapefiles, addplanarDist)
shapefiles_vorschlaege <- lapply(shapefiles_vorschlaege, addplanarDist)
#####
#####
# euklidische Distanz zwischen den einzelnen Punkten berechnen
calculateDisteuklid <- function(x_Koord,y_Koord,z_Koord){
  dist = x_Koord
  dist = NA

```

```

    for (i in 1:(length(x_Koord)-1)){
      dist[i+1] = sqrt(((x_Koord[i] - x_Koord[i+1])^2) + ((y_Koord[i] - y_Koord[i+1])^2) + ((z_Koord[i] - z_Koord[i+1])^2))
    }
    return(as.numeric(dist))
  }
}
addeuclidDist <- function(shapefile){
  shapefile$euclid_Distanz <- calculateDisteuklid(shapefile$POINT_X,shapefile$POINT_Y,shapefile$altitude)
  return(shapefile)
}
shapefiles <- lapply(shapefiles,addeuclidDist)
shapefiles_vorschlaege <- lapply(shapefiles_vorschlaege,addeuclidDist)
#####
#####
# Zeitdifferenz zwischen zwei Punkten berechnen

Zeitdiff <- function(Zeit){
  x=0
  for (i in 1:(length(Zeit)-1)){
    x[i+1]=difftime(Zeit[i+1],Zeit[i],unit = "sec")
  }
  return(as.numeric(x))
}
addZeit <- function(shapefile){
  shapefile$Zeitdifferenz <- Zeitdiff(shapefile$DateTime)
  return(shapefile)
}
shapefiles <- lapply(shapefiles,addZeit)

#####
#####
# Höhenunterschied berechnen zwischen zwei Punkten
calculateHoehendif <- function(Hoehe){
  hight <- Hoehe
  hight <- 0
  for (i in 1:(length(Hoehe)-1)){
    hight[i+1] = (Hoehe[i+1] - Hoehe[i])
  }
  return(as.numeric(hight))
}
addHohenunterschied <- function(shapefile){
  shapefile$Hohenunterschied <- calculateHoehendif(shapefile$altitude)
  return(shapefile)
}
shapefiles <- lapply(shapefiles,addHohenunterschied)
shapefiles_vorschlaege <- lapply(shapefiles_vorschlaege,addHohenunterschied)
#####
##
# Geschwindigkeit in m/s berechnen (über 11 Punkte)

Geschw <- function(x_Koord,y_Koord,z_Koord,Zeit,npoint=11){
  x = 0
  for (i in 1:(length(x_Koord)-6)){
    x[i+6]= (sqrt(((x_Koord[i+1] - x_Koord[i+npoint])^2) + ((y_Koord[i+1] - y_Koord[i+npoint])^2) + ((z_Koord[i+1] - z_Koord[i+npoint])^2)))/(as.numeric(difftime((Zeit[i+npoint]),(Zeit[i+1]),unit = "sec"))))
  }
  return(as.numeric(x))
}
}
addGeschw <- function(shapefile){
  shapefile$Geschwindigkeit <- Geschw(shapefile$POINT_X,shapefile$POINT_Y,shapefile$altitude,shapefile$DateTime)
  return(shapefile)
}
shapefiles <- lapply(shapefiles,addGeschw)

```

```

#####
#####
# Höhenunterschied über n-Punkte (moving window) berechnen.
calculateHohenunterschied13 <- function(df,npoint=13){
  Hohenunterschied13 = df
  for (i in 1:(length(df)-6)){
    Hohenunterschied13[i+6]= round(mean(df[(i):(i+(npoint-1))], na.rm = TRUE),digits=2)
  }
  return(as.numeric(Hohenunterschied13))
}
addHohenunterschied13 <- function(shapefile){
  shapefile$Hohenunterschied13 <- calculateHohenunterschied13(shapefile$Hohenunterschied)
  return(shapefile)
}
shapefiles <- lapply(shapefiles,addHohenunterschied13)
shapefiles_vorschlaege <- lapply(shapefiles_vorschlaege,addHohenunterschied13)
#####
#####
# Anstiege, Abfahrten und Geradeausfahrt bestimmen und hinzufügen

# Berechnet den Höhenunterschied vom Anfang zum Ende eines Anstiegs/Abfahrt
Anstiege <- function(Hohenunterschied,mittlerer_Hohenunterschied){
  x = 0
  for (i in 1:(length(Hohenunterschied)-1)){
    if (mittlerer_Hohenunterschied[i]<0 & mittlerer_Hohenunterschied[i+1]>=0 & Hohenunterschied[i+1]>=0){
      x[i+1]=Hohenunterschied[i+1]
    } else if (mittlerer_Hohenunterschied[i+1]<0){
      x[i+1]=0
    } else if (Hohenunterschied[i+1]<0){
      x[i+1]=x[i]
    } else {
      x[i+1]=x[i]+Hohenunterschied[i+1]
    }
  }
  return(as.numeric(x))
}
Abfahrt <- function(Hohenunterschied,mittlerer_Hohenunterschied){
  x = 0
  for (i in 1:(length(Hohenunterschied)-1)){
    if (mittlerer_Hohenunterschied[i]>0 & mittlerer_Hohenunterschied[i+1]<=0 & Hohenunterschied[i+1]<=0){
      x[i+1]=Hohenunterschied[i+1]
    } else if (mittlerer_Hohenunterschied[i+1]>0){
      x[i+1]=0
    } else if (Hohenunterschied[i+1]>0){
      x[i+1]=x[i]
    } else {
      x[i+1]=x[i]+Hohenunterschied[i+1]
    }
  }
  return(x)
}

# Markiert Anstiege/Abfahrten mit "1"
AnstiegeJaNein <- function(df){
  x <- 1
  for (i in 1:(length(df))){
    if ((df[i])!=0){
      x[i]=1
    } else {
      x[i]=0
    }
  }
  return(x)
}
AbfahrtJaNein <- function(df){
  x <- 1

```

```

for (i in 1:(length(df))){
  if ((df[i])!=0){
    x[i]=1
  } else {
    x[i]=0
  }
}
return(x)
}

```

*# Nummeriert die in einem/einer Anstieg/Abfahrte enthaltenen Punkte*

```

AnstiegeNr <- function(df){
  x <- 0
  for (i in 1:(length(df)-1)){
    if (df[i+1]==1){
      x[i+1]=x[i]+df[i+1]
    } else {
      x[i+1]=0
    }
  }
  return(x)
}

```

```

AbfahrtNr <- function(df){
  x <- 0
  for (i in 1:(length(df)-1)){
    if (df[i+1]==-1){
      x[i+1]=x[i]+df[i+1]
    } else {
      x[i+1]=0
    }
  }
  return(x)
}

```

*# Ordnet jedem Punkt in einem Anstiegsabschnitt/Abfahrtsabschnitt den maximalen Höhenunterschied des gesamten Anstiegsabschnitt zu.*

```

Anstiegmax <- function(df,AnstiegeNr){
  x <- 0
  for (i in 1:length(df)){
    if (df[i]>0){
      x[(i-((AnstiegeNr[i])-1)):i]=df[i]
    } else {
      x[i]=0
    }
  }
  return(x)
}

```

```

Abfahrtmax <- function(df,AbfahrtNr){
  x <- 0
  for (i in 1:length(df)){
    if (df[i]<0){
      x[(i-((AbfahrtNr[i])-1)):i]=df[i]
    } else {
      x[i]=0
    }
  }
  return(x)
}

```

*# Vereint alle Anstiegsfunktionen/Abfahrtfunktionen*

```

AlleAufstiegFunktionen <- function(Hoehenunterschied,mittlerer_Hoehenunterschied){
  x = 0
  Anstiege <- c(Anstiege(Hoehenunterschied,mittlerer_Hoehenunterschied))
  AnstiegeJaNein <- c(AnstiegeJaNein(Anstiege))
  AnstiegeNr <- c(AnstiegeNr(AnstiegeJaNein))
  x = c(Anstiegmax(Anstiege,AnstiegeNr))
  return(x)
}

```

```

AlleAbfahrtsFunktionen <- function(Hoehenunterschied,mittlerer_Hoehenunterschied){
  x = 0
  Abfahrt <- c(Abfahrt(Hoehenunterschied,mittlerer_Hoehenunterschied))
  AbfahrtJaNein <- c(AbfahrtJaNein(Abfahrt))
  AbfahrtNr <- c(AbfahrtNr(AbfahrtJaNein))
  x = c(Abfahrtmax(Abfahrt,AbfahrtNr))
  return(x)
}

# Vereint Anstiegs- und Abfahrtsfunktionen in einer einzigen Funktion. Anstiege werden mit "1" ma
rkiert.
# Abfahrten werden mit "2" markiert. Geradeausfahrten werden mit "0" markiert. Für Anstiege und A
bfahrten kann
# ein Schwellenwert definiert werden, der bestimmt ab welchem Höhenunterschied der jeweilige Absc
hnitt als Anstieg
# oder Abfahrt gezählt wird.
AnstiegAbfahrtsFunktion <- function(Hoehenunterschiede,mittlerer_Hoehenunterschied,Schwellenwert_
Aufstieg,Schwellenwert_Abfahrt){
  x <- 1
  Anstiegmaximal <- AlleAufstiegFunktionen(Hoehenunterschiede,mittlerer_Hoehenunterschied)
  Abfahrtmaximal <- AlleAbfahrtsFunktionen(Hoehenunterschiede,mittlerer_Hoehenunterschied)
  for (i in 1:length(Anstiegmaximal)){
    if (Anstiegmaximal[i]>=Schwellenwert_Aufstieg){
      x[i]=1
    } else if (Abfahrtmaximal[i]<=(-Schwellenwert_Abfahrt)){
      x[i]=2
    } else {
      x[i]=0
    }
  }
  return(x)
}

addAnstiegAbfahrtsFunktion <- function(shapefile){
  shapefile$UphillDownhill <- AnstiegAbfahrtsFunktion(shapefile$Hoehenunterschied,shapefile$Hoehe
unterschied13,25,25)
  return(shapefile)
}
shapefiles <- lapply(shapefiles,addAnstiegAbfahrtsFunktion)
shapefiles_vorschlaege <- lapply(shapefiles_vorschlaege,addAnstiegAbfahrtsFunktion)

#####
#####
# Gefälle hinzufügen

Gefaelle <- function(Hoehenunterschied,Distanz_planar){
  x <- 0
  for (i in 1:(length(Hoehenunterschied))){
    x[i]=(Hoehenunterschied[i]/Distanz_planar[i])*100
  }
  return(as.numeric(x))
}
add.Gefaelle <- function(shapefile){
  shapefile$Gefaelle <- Gefaelle(shapefile$Hoehenunterschied,shapefile$planare_Distanz)
  return(shapefile)
}
shapefiles <- lapply(shapefiles,add.Gefaelle)
shapefiles_vorschlaege <- lapply(shapefiles_vorschlaege,add.Gefaelle)
#####
#####
# Gefälle über n-Punkte gemittelt hinzufügen

Gefaelle.nPunkte <- function(Gefaelle,npoint=3){
  x = 0
  for (i in 1:(length(Gefaelle)-1)){
    x[i+1]= round(mean(Gefaelle[(i):(i+(npoint-1))], na.rm = TRUE),digits=2)
  }
}

```

```

    return(as.numeric(x))
}
add.Gefaelle.nPunkte <- function(shapefile){
  shapefile$Gefaelle.ueber.3 <- Gefaelle.nPunkte(shapefile$Gefaelle)
  return(shapefile)
}
shapefiles <- lapply(shapefiles,add.Gefaelle.nPunkte)
shapefiles_vorschlaege <- lapply(shapefiles_vorschlaege,add.Gefaelle.nPunkte)
#####
#####
# NaN-Werte als Null ersetzen

NaN.to.Zero.Gefaelle <- function(shapefile){
  shapefile$Gefaelle[is.nan(shapefile$Gefaelle)]<-0
  return(shapefile)
}
shapefiles <- lapply(shapefiles,NaN.to.Zero.Gefaelle)
NaN.to.Zero.Gefaelle.ueber.3 <- function(shapefile){
  shapefile$Gefaelle.ueber.3[is.nan(shapefile$Gefaelle.ueber.3)]<-0
  return(shapefile)
}
shapefiles <- lapply(shapefiles,NaN.to.Zero.Gefaelle.ueber.3)
shapefiles_vorschlaege <- lapply(shapefiles_vorschlaege,NaN.to.Zero.Gefaelle.ueber.3)
#####
#####
# Gefälleklassen hinzufügen
# Klassen: 1 = Gefälle <40%, 2 = 40%< Gefälle <70%, 3 = Gefälle >70%

Gefaelle.Klassen <- function(Gefaelle.nPunkte){
  x = 0
  for (i in 1:length(Gefaelle.nPunkte)){
    if (abs(Gefaelle.nPunkte[i])<=15){
      x[i]=0
    } else if (abs(Gefaelle.nPunkte[i])<=40 & abs(Gefaelle.nPunkte[i])>15){
      x[i]=1
    } else if (abs(Gefaelle.nPunkte[i])<=70 & abs(Gefaelle.nPunkte[i])>40){
      x[i]=2
    } else if (abs(Gefaelle.nPunkte[i])>70){
      x[i]=3
    }
  }
  return(x)
}
add.Gefaelle.Klassen <- function(shapefile){
  shapefile$Gefaelle.Klassen <- Gefaelle.Klassen(shapefile$Gefaelle.ueber.3)
  return(shapefile)
}
shapefiles <- lapply(shapefiles,add.Gefaelle.Klassen)
shapefiles_vorschlaege <- lapply(shapefiles_vorschlaege,add.Gefaelle.Klassen)
#####
#####
# Identifikation von motorisierten Verkehrsmittel (z.B. Standseilbahnen o. Luftseilbahnen)

# In Spalte Geschwindigkeit und Gefälle über 3 Punkte Werte mit "NA" auf "0" setzen
Loesche.NA.zu.0 <- function(shapefile){
  shapefile$Geschwindigkeit[is.na(shapefile$Geschwindigkeit)]=0
  shapefile$Gefaelle.ueber.3[is.na(shapefile$Gefaelle.ueber.3)]=0
  return(shapefile)
}
shapefiles <- lapply(shapefiles,Loesche.NA.zu.0)

Verkehrsmittel.identifikation.Schritt.1 <- function(Geschwindigkeit,Steigung,Wegtyp){
  x=0
  for (i in 1:length(Steigung)){
    if (Geschwindigkeit[i]>=4 & Steigung[i]>=15 & Wegtyp[i]==0){
      x[i]=1
    }
  }
}

```



```

    } else {
      x[i]=0
    }
  }
  return(x)
}

Durchschnitt.ueber.21Pkt <- function(Geschwindigkeit,Steigung,Wegtyp,npoint=21){
  x=0
  Verkehrsmittel= c(Verkehrsmittel.identifikation.Schritt.1(Geschwindigkeit,Steigung,Wegtyp))
  for (i in 1:(length(Steigung)-11)){
    x[i+11]=mean(Verkehrsmittel[(i+1):(i+npoint)])
  }
  return(as.numeric(x))
}

Verkehrsmittel.identifikation.Schritt.2 <- function(Geschwindigkeit,Steigung,Wegtyp){
  x=0
  Durchschnitt_ueber_21Pkt = c(Durchschnitt.ueber.21Pkt(Geschwindigkeit,Steigung,Wegtyp,npoint=21))
  Durchschnitt_ueber_21Pkt[is.na(Durchschnitt_ueber_21Pkt)]=0
  for (i in 1:length(Steigung)){
    if (Durchschnitt_ueber_21Pkt[i]>=0.1){
      x[i]=1
    } else {
      x[i]=0
    }
  }
  return(x)
}

add.Verkehrsmittel.identifikation.Schritt.2 <- function(shapefile){
  shapefile$Verkehrsmittel <- Verkehrsmittel.identifikation.Schritt.2(shapefile$Geschwindigkeit,s
hapefile$Gefaelle.ueber.3,shapefile$OBJEKTART)
  return(shapefile)
}
shapefiles <- lapply(shapefiles,add.Verkehrsmittel.identifikation.Schritt.2)

#####
#####
# Übergänge hinzufügen.
# Übergänge markieren Teile eines GPS-Tracks, welche nicht zur Mountainbike-Tour
# hinzugezählt werden, da es sich mit grosser Wahrscheinlichkeit um den Anfahrtsweg bis zum eigen
tlichen
# Tourbeginn handelt, oder es handelt sich um den Rückweg von der Tour oder es handelt sich um ei
n
# Verbindungsstück zwischen zwei Touren.

Uebergang.Ja.Nein <- function(Siedlung,Wegtyp,Verkehrsmittel){
  x = 0
  for (i in 1:length(Siedlung)){
    if (Verkehrsmittel[i]==1){
      x[i]=1
    } else if (Siedlung[i]==1 & (Wegtyp[i]!=16 | Wegtyp[i]!=0)){ # 16=Singletrail und 0=nicht
kartierter Singletrail{
      x[i]=1
    } else {
      x[i]=0
    }
  }
  return(as.numeric(x))
}

add.Uebergang.Ja.Nein <- function(shapefile){
  shapefile$Uebergang = Uebergang.Ja.Nein(shapefile$siedlung,shapefile$OBJEKTART,shapefile$Verkeh
rsmittel)
  return(shapefile)
}
Uebergang.manuell.Vorschlaege <- function(ObjektID){

```

```

x=0
for (i in 1:length(ObjektID)){
  x[i]=0
}
return(as.numeric(x))
}
add.Uebergang.manuell <- function(shapefile){
  shapefile$Uebergang = Uebergang.manuell.Vorschlaege(shapefile$OBJECTID)
  return(shapefile)
}
shapefiles <- lapply(shapefiles,add.Uebergang.Ja.Nein)
shapefiles_vorschlaege <- lapply(shapefiles_vorschlaege,add.Uebergang.manuell)
#####
#####
# Höhenmeter bergauf hinzufügen und Höhenmeter bergab hinzufügen (wird für die Berechnung des
# Anstiegs-Abfahrts-Verhältnis verwendet)

# bergauf
Hohenmeter.bergauf <- function(Hoehendifferenz,Verkehrsmittel){
  x = 0
  for (i in 1:length(Hoehendifferenz)){
    if (Hoehendifferenz[i]>0 & Verkehrsmittel[i]==0){
      x[i] = Hoehendifferenz[i]
    } else {
      x[i] = 0
    }
  }
  return(as.numeric(x))
}
add.Hohenmeter.bergauf <- function(shapefile){
  shapefile$Hohenmeter.Uphill <- Hohenmeter.bergauf(shapefile$Hoehenunterschied,shapefile$Verkehrsmittel)
  return(shapefile)
}
# bergauf (für Vorschläge)
Hohenmeter.bergauf.Vorschlaege <- function(Hoehendifferenz){
  x = 0
  for (i in 1:length(Hoehendifferenz)){
    if (Hoehendifferenz[i]>0){
      x[i] = Hoehendifferenz[i]
    } else {
      x[i] = 0
    }
  }
  return(as.numeric(x))
}
add.Hohenmeter.bergauf.Vorschlaege <- function(shapefile){
  shapefile$Hohenmeter.Uphill <- Hohenmeter.bergauf.Vorschlaege(shapefile$Hoehenunterschied)
  return(shapefile)
}
shapefiles <- lapply(shapefiles,add.Hohenmeter.bergauf)
shapefiles_vorschlaege <- lapply(shapefiles_vorschlaege,add.Hohenmeter.bergauf.Vorschlaege)
#bergab
Hohenmeter.bergab <- function(Hoehendifferenz,Verkehrsmittel){
  x = 0
  for (i in 1:length(Hoehendifferenz)){
    if (Hoehendifferenz[i]<0 & Verkehrsmittel[i]==0){
      x[i] = Hoehendifferenz[i]
    } else {
      x[i] = 0
    }
  }
  return(as.numeric(x))
}
add.Hohenmeter.bergab <- function(shapefile){
  shapefile$Hohenmeter.Downhill <- Hohenmeter.bergab(shapefile$Hoehenunterschied,shapefile$Verkehrsmittel)
  return(shapefile)
}

```

```

}
#bergab (für Vorschläge)
Hoehenmeter.bergab.Vorschlaege <- function(Hoehendifferenz){
  x = 0
  for (i in 1:length(Hoehendifferenz)){
    if (Hoehendifferenz[i]<0){
      x[i] = Hoehendifferenz[i]
    } else {
      x[i] = 0
    }
  }
  return(as.numeric(x))
}
add.Hoehenmeter.bergab.Vorschlaege <- function(shapefile){
  shapefile$Hoehenmeter.Downhill <- Hoehenmeter.bergab.Vorschlaege(shapefile$Hoehenunterschied)
  return(shapefile)
}
shapefiles <- lapply(shapefiles,add.Hoehenmeter.bergab)
shapefiles_vorschlaege <- lapply(shapefiles_vorschlaege,add.Hoehenmeter.bergab.Vorschlaege)

#####
#####
# Erstelle Data frame mit Parametern der einzelnen GPS-Tracks
GPS_Track <- layer_names
GPS_Track_Vorschlag <- layer_names_Vorschlaege
DF_Parameter <- data.frame(GPS_Track)
DF_Vorschlaege <- data.frame(GPS_Track_Vorschlag)

#####
# Gesamt zurückgelegte Distanz für jeden einzelnen GPS-Track berechnen und in den neuen
# Data frame einfügen

gesamt_Dist_GPS_Track <- function(shapefile){
  x=0
  x=sum(shapefile$euklid_Distanz,na.rm = TRUE)
  return(x)
}
DF_Parameter$gesamt_Dist_GPS_Track <- lapply(shapefiles,gesamt_Dist_GPS_Track)
DF_Vorschlaege$gesamt_Dist_GPS_Track <- lapply(shapefiles_vorschlaege,gesamt_Dist_GPS_Track)
#####
# Cross-Country-Index berechnen

AnzahlAnstiege <- function(shapefile){
  x <- 0
  for (i in 1:(length(shapefile$UphillDownhill)-1)){
    if (shapefile$UphillDownhill[i+1]==shapefile$UphillDownhill[i]){
      x[i+1] = x[i]
    } else if (shapefile$UphillDownhill[i+1]==1 & shapefile$UphillDownhill[i]!=1){
      x[i+1] = x[i] + 1
    } else {
      x[i+1] = x[i]
    }
  }
  return(max(x))
}

AnzahlAbfahrten <- function(shapefile){
  x <- 0
  for (i in 1:(length(shapefile$UphillDownhill)-1)){
    if (shapefile$UphillDownhill[i+1]==shapefile$UphillDownhill[i]){
      x[i+1] = x[i]
    } else if (shapefile$UphillDownhill[i+1]==2 & shapefile$UphillDownhill[i]!=2){
      x[i+1] = x[i] + 1
    } else {
      x[i+1] = x[i]
    }
  }
  return(max(x))
}

```

```

}
CrossCountryIndex <- function(shapefile){
  x=0
  x=(AnzahlAnstiege(shapefile))+(AnzahlAbfahrten(shapefile))
  return(x)
}
DF_Parameter$CrossCountryIndex <- lapply(shapefiles,CrossCountryIndex)
DF_Vorschlaege$CrossCountryIndex <- lapply(shapefiles_vorschlaege,CrossCountryIndex)
#####
# Gesamt Länge Mountainbike-Tour (Distanz ohne Übergänge) für jeden einzelnen GPS-Track berechnen
und in den neuen
# Data frame einfügen

Dist.MTB.Tour <- function(shapefile){
  x = 0
  for (i in 1:length(shapefile$euklid_Distanz)){
    if (shapefile$Uebergang[i]==0){
      x[i]=shapefile$euklid_Distanz[i]
    } else {
      x[i]=0
    }
  }
  return(sum(x,na.rm = TRUE))
}
DF_Parameter$Laenge_MTB_Tour <- lapply(shapefiles,Dist.MTB.Tour)

#####
# Gesamtlänge der Übergänge für jeden einzelnen GPS-Track berechnen und in den neuen
# Data frame einfügen

Dist.MTB.Uebergaenge <- function(shapefile){
  x = 0
  for (i in 1:length(shapefile$euklid_Distanz)){
    if (shapefile$Uebergang[i]==1){
      x[i]=shapefile$euklid_Distanz[i]
    } else {
      x[i]=0
    }
  }
  return(sum(x,na.rm = TRUE))
}
DF_Parameter$Laenge_Uebergaenge <- lapply(shapefiles,Dist.MTB.Uebergaenge)

#####
# Anteil der Übergänge am gesamten GPS-Track berechnen

Anteil.Uebergaenge <- function(gesamt.Dist.GPS.Track,Laenge.Uebergaenge){
  x=0
  for (i in 1:length(gesamt.Dist.GPS.Track)){
    x[i]=100*as.numeric(Laenge.Uebergaenge[i])/as.numeric(gesamt.Dist.GPS.Track[i])
  }
  return(x)
}

DF_Parameter$Anteil_Uebergaenge <- Anteil.Uebergaenge(DF_Parameter$gesamt_Dist_GPS_Track,DF_Parameter$Laenge_Uebergaenge)

#####
# Gesamtlänge Singletrail für jeden einzelnen GPS-Track berechnen

gesamt.Singletrail <- function(shapefile){
  x=0
  for (i in 1:length(shapefile$euklid_Distanz)){
    if (shapefile$OBJEKTART[i]==16 | shapefile$OBJEKTART[i]==0 & shapefile$Uebergang[i]==0){
      x[i]=as.numeric(shapefile$euklid_Distanz[i])
    }
  }
  return(sum(x,na.rm = TRUE))
}

```

```

}
gesamt.Singletrail.Vorschlaege <- function(shapefile){
  x=0
  for (i in 1:length(shapefile$euklid_Distanz)){
    if (shapefile$OBJEKTART[i]==16 | shapefile$OBJEKTART[i]==0){
      x[i]=as.numeric(shapefile$euklid_Distanz[i])
    }
  }
  return(sum(x,na.rm = TRUE))
}
DF_Parameter$Dist_Singletrail <- lapply(shapefiles,gesamt.Singletrail)
DF_Vorschlaege$Dist_Singletrail <- lapply(shapefiles_vorschlaege,gesamt.Singletrail.Vorschlaege)
#####
# Anteil Singletrail berechnen

Anteil.Singletrail <- function(Laenge_MTB_Tour,Dist_Singletrail){
  x=0
  for (i in 1:length(Laenge_MTB_Tour)){
    x[i]=100*(as.numeric(Dist_Singletrail[i]))/(as.numeric(Laenge_MTB_Tour[i]))
  }
  return(x)
}
Anteil.Singletrail.Vorschlaege <- function(gesamt_Dist_GPS_Track,Dist_Singletrail){
  x=0
  for (i in 1:length(gesamt_Dist_GPS_Track)){
    x[i]=100*(as.numeric(Dist_Singletrail[i]))/(as.numeric(gesamt_Dist_GPS_Track[i]))
  }
  return(x)
}
DF_Parameter$Anteil_Singletrail <- Anteil.Singletrail(DF_Parameter$Laenge_MTB_Tour,DF_Parameter$Dist_Singletrail)
DF_Vorschlaege$Anteil_Singletrail <- Anteil.Singletrail.Vorschlaege(DF_Vorschlaege$gesamt_Dist_GPS_Track,DF_Vorschlaege$Dist_Singletrail)
#####
# Gesamte Anzahl Höhenmeter Uphill für jeden einzelnen GPS_Track berechnen und in den
# neuen DF einfügen

gesamt_Hoehenmeter_Uphill <- function(shapefile){
  x=0
  x=sum(shapefile$Hoehenmeter.Uphill,na.rm = TRUE)
  return(x)
}
DF_Parameter$gesamt_Hoehe_Uphill <- lapply(shapefiles,gesamt_Hoehenmeter_Uphill)
DF_Vorschlaege$gesamt_Hoehe_Uphill <- lapply(shapefiles_vorschlaege,gesamt_Hoehenmeter_Uphill)
#####
# Gesamte Anzahl Höhenmeter Downhill für jeden einzelnen GPS_Track berechnen und in DF einfügen

gesamt_Hoehenmeter_Downhill <- function(shapefile){
  x=0
  x=sum(shapefile$Hoehenmeter.Downhill,na.rm = TRUE)
  return(x)
}
DF_Parameter$gesamt_Hoehe_Downhill <- lapply(shapefiles,gesamt_Hoehenmeter_Downhill)
DF_Vorschlaege$gesamt_Hoehe_Downhill <- lapply(shapefiles_vorschlaege,gesamt_Hoehenmeter_Downhill)
#####
# Verhältnis Höhenmeter Uphill/Downhill

Verhaeltnis.Uphill.Downhill <- function(gesamt_Hoehe_Uphill,gesamt_Hoehe_Downhill){
  x=0
  for (i in 1:length(gesamt_Hoehe_Uphill)){
    x[i]=(as.numeric(gesamt_Hoehe_Uphill[i]))/abs(as.numeric((gesamt_Hoehe_Downhill[i])))
  }
  return(x)
}
DF_Parameter$Uphill_Downhill_Verhaeltnis <- Verhaeltnis.Uphill.Downhill(DF_Parameter$gesamt_Hoehe_Uphill,DF_Parameter$gesamt_Hoehe_Downhill)
DF_Vorschlaege$Uphill_Downhill_Verhaeltnis <- Verhaeltnis.Uphill.Downhill(DF_Vorschlaege$gesamt_Ho

```

```

ehe_Uphill,DF_Vorschlaege$gesamt_Hoehe_Downhill)
#####
# Gesamtlänge Aufstiege (ohne Übergänge) für jeden einzelnen GPS-Track berechnen und in DF einfügen

gesamt.Anstieg <- function(shapefile){
  x=0
  for (i in 1:length(shapefile$euklid_Distanz)){
    if (shapefile$UphillDownhill[i]==1 & shapefile$Uebergang[i]==0){
      x[i]=as.numeric(shapefile$euklid_Distanz[i])
    } else {
      x[i]=0
    }
  }
  return(sum(x,na.rm = TRUE))
}
DF_Parameter$gesamter_Anstieg <- lapply(shapefiles,gesamt.Anstieg)
DF_Vorschlaege$gesamter_Anstieg <- lapply(shapefiles_vorschlaege,gesamt.Anstieg)
#####
# Singletrail an Anstiegen berechnen

Singltrail.Anstiege <- function(shapefile){
  x=0
  for (i in 1:length(shapefile$euklid_Distanz)){
    if (shapefiles$UphillDownhill[i]==1 & (shapefile$OBJEKTART[i]==16 | shapefile$OBJEKTART[i]==0
& shapefile$Uebergang[i]==0)){
      x[i]=shapefile$euklid_Distanz[i]
    } else {
      x[i]=0
    }
  }
  return(sum(x,na.rm = TRUE))
}
DF_Parameter$Distanz_Singletrail_Anstiege <- lapply(shapefiles,Singltrail.Anstiege)
DF_Vorschlaege$Distanz_Singletrail_Anstiege <- lapply(shapefiles_vorschlaege,Singltrail.Anstiege)
#####
# Singletrail-Anteil an Anstiegen berechnen

Anteil.Singletrail.Anstiege <- function(gesamter_Anstieg,Distanz_Singletrail_Anstiege){
  x=0
  for (i in 1:length(gesamter_Anstieg)){
    x[i]=100*(as.numeric(Distanz_Singletrail_Anstiege[i]))/(as.numeric(gesamter_Anstieg[i]))
    if(is.nan(x[i])){
      x[i]=0
    }
  }
  return(x)
}
DF_Parameter$Anteil_Singletrail_Anstieg <- Anteil.Singletrail.Anstiege(DF_Parameter$gesamter_Anstieg,DF_Parameter$Distanz_Singletrail_Anstiege)
DF_Vorschlaege$Anteil_Singletrail_Anstieg <- Anteil.Singletrail.Anstiege(DF_Vorschlaege$gesamter_Anstieg,DF_Vorschlaege$Distanz_Singletrail_Anstiege)
#####
#
# Distanz Anstiege mit einer Steigung >15%

Anstiege.gr15 <- function(shapefile){
  x=0
  for (i in 1:length(shapefile$euklid_Distanz)){
    if (shapefile$Gefaeulle.Klassen[i]!=0 & shapefile$UphillDownhill[i]==1 & shapefile$Uebergang[i]==0){
      x[i]=shapefile$euklid_Distanz[i]
    } else {
      x[i]=0
    }
  }
  return(as.numeric(sum(x,na.rm = TRUE)))
}

```

```

DF_Parameter$Anstiege_gr15 <- lapply(shapefiles,Anstiege.gr15)
DF_Vorschlaege$Anstiege_gr15 <- lapply(shapefiles_vorschlaege,Anstiege.gr15)
#####
#
# Anteil Anstiege >15% berechnen

Anteil.Anstiege.gr15 <- function(gesamter_Anstieg,Anstiege_gr15){
  x=0
  for (i in 1:length(gesamter_Anstieg)){
    x[i]=100*(as.numeric(Anstiege_gr15[i]))/(as.numeric(gesamter_Anstieg[i]))
    if(is.nan(x[i])){
      x[i]=0
    }
  }
  return(x)
}
DF_Parameter$Anteil_Anstiege_gr15 <- Anteil.Anstiege.gr15(DF_Parameter$gesamter_Anstieg,DF_Parameter$Anstiege_gr15)
DF_Vorschlaege$Anteil_Anstiege_gr15 <- Anteil.Anstiege.gr15(DF_Vorschlaege$gesamter_Anstieg,DF_Vorschlaege$Anstiege_gr15)
#####
#
# Gesamtlänge Abfahrten (ohne Übergänge) für jeden einzelnen GPS-Track berechnen und in DF einfügen

gesamt.Abfahrt <- function(shapefile){
  x=0
  for (i in 1:length(shapefile$euklid_Distanz)){
    if (shapefile$UphillDownhill[i]==2 & shapefile$Uebergang[i]==0){
      x[i]=as.numeric(shapefile$euklid_Distanz[i])
    } else {
      x[i]=0
    }
  }
  return(sum(x,na.rm = TRUE))
}
DF_Parameter$Abfahrt_gesamt <- lapply(shapefiles,gesamt.Abfahrt)
DF_Vorschlaege$Abfahrt_gesamt <- lapply(shapefiles_vorschlaege,gesamt.Abfahrt)
#####
#
# Distanz Singeltrails auf Abfahrten berechnen

Singltrail.Abfahrten <- function(shapefile){
  x=0
  for (i in 1:length(shapefile$euklid_Distanz)){
    if (shapefile$UphillDownhill[i]==2 & (shapefile$OBJEKTART[i]==16 | shapefile$OBJEKTART[i]==0)
    & shapefile$Uebergang[i]==0){
      x[i]=shapefile$euklid_Distanz[i]
    } else {
      x[i]=0
    }
  }
  return(sum(x,na.rm = TRUE))
}
DF_Parameter$Distanz_Singltrail_Abfahrten <- lapply(shapefiles,Singltrail.Abfahrten)
DF_Vorschlaege$Distanz_Singltrail_Abfahrten <- lapply(shapefiles_vorschlaege,Singltrail.Abfahrten)
#####
#
# Anteil von Singeltrails auf Abfahrten berechnen

Anteil.Singltrail.Abfahrt <- function(Abfahrt_gesamt,Distanz_Singltrail_Abfahrten){
  x=0
  for (i in 1:length(Abfahrt_gesamt)){
    x[i]=100*(as.numeric(Distanz_Singltrail_Abfahrten[i]))/(as.numeric(Abfahrt_gesamt[i]))
    if(is.nan(x[i])){
      x[i]=0
    }
  }
  return(x)
}

```

```

}
DF_Parameter$Anteil_Singletrail_Abfahrten <- Anteil.Singletrail.Abfahrt(DF_Parameter$Abfahrt_gesamt,DF_Parameter$Distanz_Singletrail_Abfahrten)
DF_Vorschlaege$Anteil_Singletrail_Abfahrten <- Anteil.Singletrail.Abfahrt(DF_Vorschlaege$Abfahrt_gesamt,DF_Vorschlaege$Distanz_Singletrail_Abfahrten)
#####
#####
# Distanz Singletrail auf Abfahrten mit einem Gefälle von <40%

Abfahrten.kl40 <- function(shapefile){
  x=0
  for (i in 1:length(shapefile$euklid_Distanz)){
    if ((shapefile$Gefaelle.Klassen[i]==0 | shapefile$Gefaelle.Klassen[i]==1 ) & (shapefile$OBJEKTART[i]==16
    | shapefile$OBJEKTART[i]==0) & shapefile$Uebergang[i]==0 & shapefile$UphillDownhill[i]==2){
      x[i]=shapefile$euklid_Distanz[i]
    } else {
      x[i]=0
    }
  }
  return(sum(x,na.rm = TRUE))
}
DF_Parameter$Abfahrten_Singletrail_kl40 <- lapply(shapefiles,Abfahrten.kl40)
DF_Vorschlaege$Abfahrten_Singletrail_kl40 <- lapply(shapefiles_vorschlaege,Abfahrten.kl40)
#####
#
# Anteil von Singletrails auf Abfahrten mit einem Gefälle von <40% berechnen

Anteil.Singletrail.Abfahrt.kl40 <- function(Abfahrt_gesamt,Abfahrten_Singletrail_kl40){
  x=0
  for (i in 1:length(Abfahrt_gesamt)){
    x[i]=100*(as.numeric(Abfahrten_Singletrail_kl40[i]))/(as.numeric(Abfahrt_gesamt[i]))
    if(is.nan(x[i])){
      x[i]=0
    }
  }
  return(x)
}
DF_Parameter$Anteil_Abfahrten_ST_kl40 <- Anteil.Singletrail.Abfahrt(DF_Parameter$Abfahrt_gesamt,DF_Parameter$Abfahrten_Singletrail_kl40)
DF_Vorschlaege$Anteil_Abfahrten_ST_kl40 <- Anteil.Singletrail.Abfahrt(DF_Vorschlaege$Abfahrt_gesamt,DF_Vorschlaege$Abfahrten_Singletrail_kl40)
#####
#####
# Distanz Singletrail auf Abfahrten mit einem Gefälle von >40% und <70% berechnen

Abfahrten.gr40.kl70 <- function(shapefile){
  x=0
  for (i in 1:length(shapefile$euklid_Distanz)){
    if (shapefile$UphillDownhill[i]==2 & (shapefile$Gefaelle.Klassen[i]==2 ) & (shapefile$OBJEKTART[i]==16 | shapefile$OBJEKTART[i]==0) & shapefile$Uebergang[i]==0){
      x[i]=shapefile$euklid_Distanz[i]
    } else {
      x[i]=0
    }
  }
  return(sum(x,na.rm = TRUE))
}
DF_Parameter$Abfahrten_ST_gr40_kl70 <- lapply(shapefiles,Abfahrten.gr40.kl70)
DF_Vorschlaege$Abfahrten_ST_gr40_kl70 <- lapply(shapefiles_vorschlaege,Abfahrten.gr40.kl70)
#####
#
# Anteil von Singletrails auf Abfahrten mit einem Gefälle von >40% und <70% berechnen

Anteil.Singletrail.Abfahrt.gr40.kl70 <- function(Abfahrt_gesamt,Abfahrten_ST_gr40_kl70){
  x=0
  for (i in 1:length(Abfahrt_gesamt)){

```



```

x[i]=100*(as.numeric(Abfahrten_ST_gr40_k170[i]))/(as.numeric(Abfahrt_gesamt[i]))
if(is.nan(x[i])){
  x[i]=0
}
}
return(x)
}
DF_Parameter$Anteil_Abfahrten_ST_gr40_k170 <- Anteil.Singletrail.Abfahrt(DF_Parameter$Abfahrt_gesamt,DF_Parameter$Abfahrten_ST_gr40_k170)
DF_Vorschlaege$Anteil_Abfahrten_ST_gr40_k170 <- Anteil.Singletrail.Abfahrt(DF_Vorschlaege$Abfahrt_gesamt,DF_Vorschlaege$Abfahrten_ST_gr40_k170)
#####
#####
# Distanz Singletrail auf Abfahrten mit einem Gefälle von >70% berechnen

Abfahrten.gr70 <- function(shapefile){
  x=0
  for (i in 1:length(shapefile$euklid_Distanz)){
    if (shapefile$UphillDownhill[i]==2 & (shapefile$Gefaelle.Klassen[i]==3 ) & (shapefile$OBJEKTART[i]==16 | shapefile$OBJEKTART[i]==0) & shapefile$Uebergang[i]==0){
      x[i]=shapefile$euklid_Distanz[i]
    } else {
      x[i]=0
    }
  }
  return(sum(x,na.rm = TRUE))
}
DF_Parameter$Abfahrten_ST_gr70 <- lapply(shapefiles,Abfahrten.gr70)
DF_Vorschlaege$Abfahrten_ST_gr70 <- lapply(shapefiles_vorschlaege,Abfahrten.gr70)
#####
#
# Anteil von Singletrails auf Abfahrten mit einem Gefälle von >40% und <70% berechnen

Anteil.Singletrail.Abfahrt.gr70 <- function(Abfahrt_gesamt,Abfahrten_ST_gr70){
  x=0
  for (i in 1:length(Abfahrt_gesamt)){
    x[i]=100*(as.numeric(Abfahrten_ST_gr70[i]))/(as.numeric(Abfahrt_gesamt[i]))
    if(is.nan(x[i])){
      x[i]=0
    }
  }
  return(x)
}
DF_Parameter$Anteil_Abfahrten_ST_gr70 <- Anteil.Singletrail.Abfahrt.gr70(DF_Parameter$Abfahrt_gesamt,DF_Parameter$Abfahrten_ST_gr70)
DF_Vorschlaege$Anteil_Abfahrten_ST_gr70 <- Anteil.Singletrail.Abfahrt.gr70(DF_Vorschlaege$Abfahrt_gesamt,DF_Vorschlaege$Abfahrten_ST_gr70)
#####
#
# Gesamtlänge Flach (ohne Übergänge) für jeden einzelnen GPS-Track berechnen und in DF einfügen
gesamt.Flach <- function(shapefile){
  x=0
  for (i in 1:length(shapefile$euklid_Distanz)){
    if (shapefile$UphillDownhill[i]==0 & shapefile$Uebergang[i]==0){
      x[i]=as.numeric(shapefile$euklid_Distanz[i])
    } else {
      x[i]=0
    }
  }
  return(sum(x,na.rm = TRUE))
}
DF_Parameter$Flach_gesamt <- lapply(shapefiles,gesamt.Flach)
DF_Vorschlaege$Flach_gesamt <- lapply(shapefiles_vorschlaege,gesamt.Flach)
#####
#
# Distanz Singeltrails auf flachen Abschnitten berechnen
Singltrail.Flach <- function(shapefile){
  x=0

```

```

for (i in 1:length(shapefile$euklid_Distanz)){
  if (shapefile$UphillDownhill[i]==0 & (shapefile$OBJEKTART[i]==16 | shapefile$OBJEKTART[i]==0)
& shapefile$Uebergang[i]==0){
    x[i]=shapefile$euklid_Distanz[i]
  } else {
    x[i]=0
  }
}
return(sum(x,na.rm = TRUE))
}
DF_Parameter$Distanz_Singletrail_Flach <- lapply(shapefiles,Singletrail.Flach)
DF_Vorschlaege$Distanz_Singletrail_Flach <- lapply(shapefiles_vorschlaege,Singletrail.Flach)
#####
#
# Anteil von Singletrails auf flachen Abschnitten berechnen

Anteil.Singletrail.Flach <- function(Flach_gesamt,Distanz_Singletrail_Flach){
  x=0
  for (i in 1:length(Flach_gesamt)){
    x[i]=100*(as.numeric(Distanz_Singletrail_Flach[i]))/(as.numeric(Flach_gesamt[i]))
    if(is.nan(x[i])){
      x[i]=0
    }
  }
  return(x)
}
DF_Parameter$Anteil_Singletrail_Flach <- Anteil.Singletrail.Flach(DF_Parameter$Flach_gesamt,DF_Pa
rameter$Distanz_Singletrail_Flach)
DF_Vorschlaege$Anteil_Singletrail_Flach <- Anteil.Singletrail.Flach(DF_Vorschlaege$Flach_gesamt,D
F_Vorschlaege$Distanz_Singletrail_Flach)
# Werte in von Meter in Kilometer umrechnen
Wert.in.KM <- function(Parameterwert){
  x=0
  for (i in 1:length(Parameterwert)){
    x[i]=as.numeric(Parameterwert[i])/1000
  }
  return(as.numeric(x))
}
DF_Parameter$gesamt_Dist_GPS_Track <- Wert.in.KM(DF_Parameter$gesamt_Dist_GPS_Track)
DF_Parameter$Laenge_MTB_Tour <- Wert.in.KM(DF_Parameter$Laenge_MTB_Tour)
DF_Parameter$Laenge_Uebergaenge <- Wert.in.KM(DF_Parameter$Laenge_Uebergaenge)
DF_Parameter$Dist_Singletrail <- Wert.in.KM(DF_Parameter$Dist_Singletrail)
DF_Parameter$gesamter_Anstieg <- Wert.in.KM(DF_Parameter$gesamter_Anstieg)
DF_Parameter$Distanz_Singletrail_Anstiege <- Wert.in.KM(DF_Parameter$Distanz_Singletrail_Anstiege
)
DF_Parameter$Abfahrt_gesamt <- Wert.in.KM(DF_Parameter$Abfahrt_gesamt)
DF_Parameter$Distanz_Singletrail_Abfahrten <- Wert.in.KM(DF_Parameter$Distanz_Singletrail_Abfahrt
en)
DF_Parameter$Flach_gesamt = Wert.in.KM(DF_Parameter$Flach_gesamt)
DF_Parameter$Distanz_Singletrail_Flach <- Wert.in.KM(DF_Parameter$Distanz_Singletrail_Flach)

DF_Vorschlaege$gesamt_Dist_GPS_Track <- Wert.in.KM(DF_Vorschlaege$gesamt_Dist_GPS_Track)
DF_Vorschlaege$Dist_Singletrail <- Wert.in.KM(DF_Vorschlaege$Dist_Singletrail)
DF_Vorschlaege$gesamter_Anstieg <- Wert.in.KM(DF_Vorschlaege$gesamter_Anstieg)
DF_Vorschlaege$Distanz_Singletrail_Anstiege <- Wert.in.KM(DF_Vorschlaege$Distanz_Singletrail_Anst
iege)
DF_Vorschlaege$Abfahrt_gesamt <- Wert.in.KM(DF_Vorschlaege$Abfahrt_gesamt)
DF_Vorschlaege$Distanz_Singletrail_Abfahrten <- Wert.in.KM(DF_Vorschlaege$Distanz_Singletrail_Abf
ahrten)
DF_Vorschlaege$Flach_gesamt = Wert.in.KM(DF_Vorschlaege$Flach_gesamt)
DF_Vorschlaege$Distanz_Singletrail_Flach <- Wert.in.KM(DF_Vorschlaege$Distanz_Singletrail_Flach)

CrossCountryIndex.pro.km<- function(CCI,gesamt_DistanzGPS){
  x=0
  for (i in 1:length(CCI)){
    x[i]=as.numeric(CCI[i])/as.numeric(gesamt_DistanzGPS[i])
  }
  return(x)
}

```

```

}
DF_Parameter$CrossCountryIndex <- CrossCountryIndex.pro.km(DF_Parameter$CrossCountryIndex,DF_Parameter$gesamt_Dist_GPS_Track)
DF_Vorschlaege$CrossCountryIndex <- CrossCountryIndex.pro.km(DF_Vorschlaege$CrossCountryIndex,DF_Vorschlaege$gesamt_Dist_GPS_Track)

#####
#####
# Tourenvorschläge validieren

# Erstelle Data Frame für Eignungswerte
DF_Toureneignung <- data.frame(DF_Vorschlaege$GPS_Track_Vorschlag)

# Vergleiche die Länge der Mountainbike-Tour

Vergleich <- function(DF_Tourenprofil,DF_Vorschlag,Gewichtungsfaktor){
  x=0
  for (i in 1:length(DF_Vorschlag)){
    if (sum(as.numeric(DF_Tourenprofil))==0){
      x=as.numeric(0)
    }else if ((wilcox.test((as.numeric(DF_Tourenprofil)),mu=(as.numeric(DF_Vorschlag[i])),alternative = "two.sided")$p.value)>0.05){
      x[i]=1*Gewichtungsfaktor
    } else {
      x[i]=0
    }
  }
  return(as.numeric(x))
}
DF_Toureneignung$Laenge_MTB_Tour <- Vergleich(DF_Parameter$Laenge_MTB_Tour,DF_Vorschlaege$gesamt_Dist_GPS_Track,2)

# Vergleiche den Cross-Country-Index
DF_Toureneignung$CrossCountryIndex <- Vergleich(DF_Parameter$CrossCountryIndex,DF_Vorschlaege$CrossCountryIndex,2)

# Vergleiche die Gesamtdistanz Singletrail
DF_Toureneignung$Gesamtdistanz_ST <- Vergleich(DF_Parameter$Dist_Singletrail,DF_Vorschlaege$Dist_Singletrail,2)

# Vergleiche den Anteil Singletrail an der gesamten Mountainbiketour
DF_Toureneignung$Anteil_ST <- Vergleich(DF_Parameter$Anteil_Singletrail,DF_Vorschlaege$Anteil_Singletrail,2)

# Vergleiche Gesamthöhe Uphill
DF_Toureneignung$Gesamthoehe_Uphill <- Vergleich(DF_Parameter$gesamt_Hoehe_Uphill,DF_Vorschlaege$gesamt_Hoehe_Uphill,1)

# Vergleiche Gesamthöhe Downhill
DF_Toureneignung$Gesamthoehe_Downhill <- Vergleich(DF_Parameter$gesamt_Hoehe_Downhill,DF_Vorschlaege$gesamt_Hoehe_Downhill,1)

# Vergleiche Uphill-Downhill-Verhältnis
DF_Toureneignung$Up_Down_Verhaeltnis <- Vergleich(DF_Parameter$Uphill_Downhill_Verhaeltnis,DF_Vorschlaege$Uphill_Downhill_Verhaeltnis,2)

# Vergleiche Gesamtanstieg
DF_Toureneignung$Gesamtanstieg <- Vergleich(DF_Parameter$gesamter_Anstieg,DF_Vorschlaege$gesamter_Anstieg,2)

# Vergleiche Distanz Singletrail bei Anstiegen
DF_Toureneignung$Dist_ST_Anstieg <- Vergleich(DF_Parameter$Distanz_Singletrail_Anstiege,DF_Vorschlaege$Distanz_Singletrail_Anstiege,2)

# Vergleiche Anteil Singletrail bei Anstiegen
DF_Toureneignung$Anteil_ST_Anstieg <- Vergleich(DF_Parameter$Anteil_Singletrail_Anstieg,DF_Vorschlaege$Anteil_Singletrail_Anstieg,2)

```

```

# Vergleiche Distanz Anstiege >15% Steigung
DF_Toureneignung$Distanz_Anstiege_gr15 <- Vergleich(DF_Parameter$Anstiege_gr15,DF_Vorschlaege$Anstiege_gr15,2)

# Vergleiche Anteil Anstiege >15%
DF_Toureneignung$Anteil_Anstiege_gr15 <- Vergleich(DF_Parameter$Anteil_Anstiege_gr15,DF_Vorschlaege$Anteil_Anstiege_gr15,2)

# Vergleiche Gesamtdistanz Abfahrten
DF_Toureneignung$Gesamtdistanz_Abfahrten <- Vergleich(DF_Parameter$Abfahrt_gesamt,DF_Vorschlaege$Abfahrt_gesamt,2)

# Vergleiche Distanz Singletrail bei Abfahrten
DF_Toureneignung$Dist_ST_Abfahrten <- Vergleich(DF_Parameter$Distanz_Singletrail_Abfahrten,DF_Vorschlaege$Distanz_Singletrail_Abfahrten,2)

# Vergleiche Anteil Singletrail auf Abfahrten
DF_Toureneignung$Anteil_ST_Abfahrten <- Vergleich(DF_Parameter$Anteil_Singletrail_Abfahrten,DF_Vorschlaege$Anteil_Singletrail_Abfahrten,2)

# Vergleiche Distanz Singletrail auf Abfahrten mit Gefälle <40%
DF_Toureneignung$Dist_ST_Abfahrten_kl40 <- Vergleich(DF_Parameter$Abfahrten_Singletrail_kl40,DF_Vorschlaege$Abfahrten_Singletrail_kl40,1)

# Vergleiche Anteil Singletrail auf Abfahrten mit Gefälle <40%
DF_Toureneignung$Anteil_ST_Abfahrten_kl40 <- Vergleich(DF_Parameter$Anteil_Abfahrten_ST_kl40,DF_Vorschlaege$Anteil_Abfahrten_ST_kl40,1)

# Vergleiche Distanz Singletrail auf Abfahrten mit Gefälle >40% und <70%
DF_Toureneignung$Dist_ST_Abfahrten_gr40_kl70 <- Vergleich(DF_Parameter$Abfahrten_ST_gr40_kl70,DF_Vorschlaege$Anteil_Abfahrten_ST_gr40_kl70,1)

# Vergleiche Anteil Singletrail auf Abfahrten mit Gefälle >40% und <70%
DF_Toureneignung$Anteil_ST_Abfahrten_gr40_kl70 <- Vergleich(DF_Parameter$Anteil_Abfahrten_ST_gr40_kl70,DF_Vorschlaege$Anteil_Abfahrten_ST_gr40_kl70,1)

# Vergleiche Distanz Singletrail auf Abfahrten mit Gefälle >70%
DF_Toureneignung$Dist_ST_Abfahrten_gr70 <- Vergleich(DF_Parameter$Abfahrten_ST_gr70,DF_Vorschlaege$Abfahrten_ST_gr40_kl70,1)

# Vergleich Anteil Singletrail auf Abfahrten mit Gefälle >70%
DF_Toureneignung$Anteil_ST_Abfahrten_gr70 <- Vergleich(DF_Parameter$Anteil_Abfahrten_ST_gr70,DF_Vorschlaege$Anteil_Abfahrten_ST_gr70,1)

# Vergleiche Gesamtdistanz von flachen Abschnitten
DF_Toureneignung$Gesamtdistanz_Flach <- Vergleich(DF_Parameter$Flach_gesamt,DF_Vorschlaege$Flach_gesamt,2)

# Vergleiche Distanz Singletrail auf flachen Abschnitten
DF_Toureneignung$Dist_ST_Flach <- Vergleich(DF_Parameter$Distanz_Singletrail_Flach,DF_Vorschlaege$Distanz_Singletrail_Flach,2)

# Vergleiche Anteil Singletrail auf flachen Abschnitten
DF_Toureneignung$Anteil_ST_flach <- Vergleich(DF_Parameter$Anteil_Singletrail_Flach,DF_Vorschlaege$Anteil_Singletrail_Flach,2)

Gesamtpunktzahl.berechnen <- function(DF){
  x=0
  for (i in 1:length(DF[,1])){
    x[i]=sum(DF[i,2:(length(DF[1,]))])
  }
  return(as.numeric(x))
}
DF_Toureneignung$Gesamtpunktzahl <- Gesamtpunktzahl.berechnen(DF_Toureneignung)

# Touren absteigend nach Gesamtpunktzahl ordnen
#DF_Toureneignung <- DF_Toureneignung[order(DF_Toureneignung$Gesamtpunktzahl,decreasing = TRUE),]

```

```
# Excel Files erstellen
#write.xlsx(DF_Parameter, "DF_MTB_A_20170831.xlsx")
#write.xlsx(DF_Vorschlaege, "DF_Vorschlaege_20170831.xlsx")
#write.xlsx(DF_Toureneignung, "DF_Toureneignung_MTB_A.xlsx")
# Datei in Excel File überführen
#write.xlsx(shapefiles[[1]], "MTB_A_1_20170829.xlsx")
#write.xlsx(shapefiles[[2]], "MTB_A_2_20170829.xlsx")
#write.xlsx(shapefiles[[3]], "MTB_A_3_20170829.xlsx")
#write.xlsx(shapefiles[[4]], "MTB_A_4_20170829.xlsx")
#write.xlsx(shapefiles[[5]], "MTB_A_5_20170829.xlsx")
#write.xlsx(shapefiles[[6]], "MTB_A_6_20170829.xlsx")
#write.xlsx(shapefiles[[7]], "MTB_A_7_20170829.xlsx")
#write.xlsx(shapefiles[[8]], "MTB_A_8_20170829.xlsx")
#write.xlsx(shapefiles[[9]], "MTB_A_9_20170829.xlsx")
#write.xlsx(shapefiles[[10]], "MTB_A_10_20170829.xlsx")

##### ENDE #####
```

---